



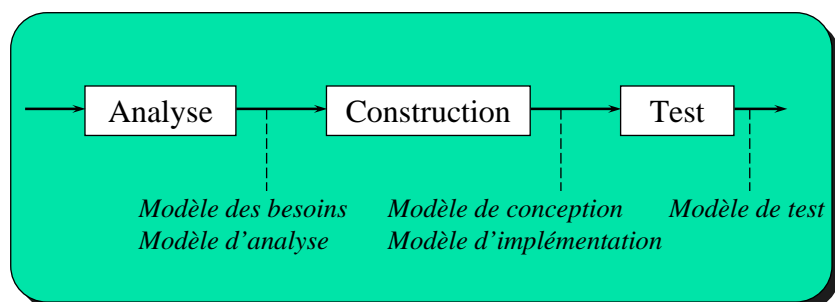
## Modèle des cas d'utilisation

Jolita Ralyté  
Université de Genève



## Origine – Méthode OOSE

- OOSE (Object Oriented Software Engineering) - une méthode orientée-objet fondée sur les cas d'utilisation





## Modèles de OOSE

---

- **Modèle des besoins**
  - la spécification des besoins fonctionnels
- **Modèle d'analyse**
  - la structure objet du système
- **Modèle de conception**
  - la structure objet adoptée et affinée pour l'environnement d'implémentation courant
- **Modèle d'implémentation**
  - l'implémentation du système
- **Modèle de test**
  - la vérification du système



## Modèle des besoins

---

**But :** Définir les limitations du système et spécifier son comportement

**Composants:**

- Modèle de cas d'utilisation
- Descriptions des interfaces
- Modèle de domaine du problème



## Modèle de cas d'utilisation

### But :

- Aider à élucider, comprendre et définir les besoins fonctionnels du système
- Modéliser la vue externe du système

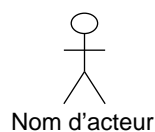
### Concepts :

- **Acteur** : Un rôle joué par un utilisateur ou une chose qui interagit avec le système
- **Cas d'utilisation** : Ce que les utilisateurs devraient être capables de faire avec le système

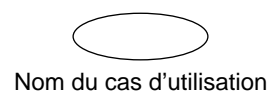


## Représentation des concepts

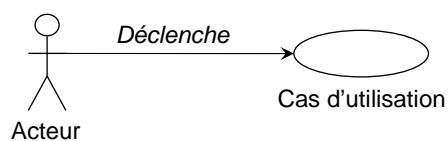
- Acteur



- Cas d'utilisation



- Relation





## Processus de construction d'un modèle de cas d'utilisation

1. Identification des acteurs
2. Identification des cas d'utilisation
3. Description des cas d'utilisation
  - Cas normal
  - Cas exceptionnels
4. Affinement des cas d'utilisation
  - Relation d'extension
5. Abstraction des cas d'utilisation
  - Relation d'utilisation
6. Organisation des cas d'utilisation
  - Relation d'inclusion



## Identification des Acteur

### Un acteur représente :

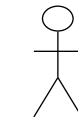
- Un rôle joué par un utilisateur du système ou une chose qui interagit avec le système
- Ce qui existe en dehors du système et qui interagit avec le système (Humain, machine, un autre système)
  - **Acteurs primaires** - les personnes qui utilisent une ou plusieurs des tâches principales du système
  - **Acteurs secondaires** - les personnes qui supervisent et qui maintiennent le système
  - **Matériel externe** - les dispositifs matériels périphériques qui font partie du domaine d'application et qui doivent être utilisés
  - **Autres systèmes** - les systèmes avec lesquels le système doit interagir



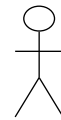
## Identification des acteurs

### Exemple: Bibliothèque

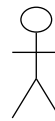
- Qui sont les utilisateurs potentiels de la bibliothèque?
- Qui va superviser la bibliothèque?
- Y aura t-il d'autres systèmes en connexion avec la bibliothèque?



Abonné



Bibliothécaire



Système de connexion  
entre plusieurs bibliothèques

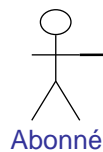
Acteur  $\neq$  Utilisateur

L'utilisateur est la personne qui utilise le système, tandis qu'un acteur représente un certain rôle qu'un utilisateur peut jouer



## Identification des acteurs

### Exemple: Bibliothèque



Abonné



SI de la  
Bibliothèque



Bibliothécaire



Système de connexion  
entre plusieurs bibliothèques

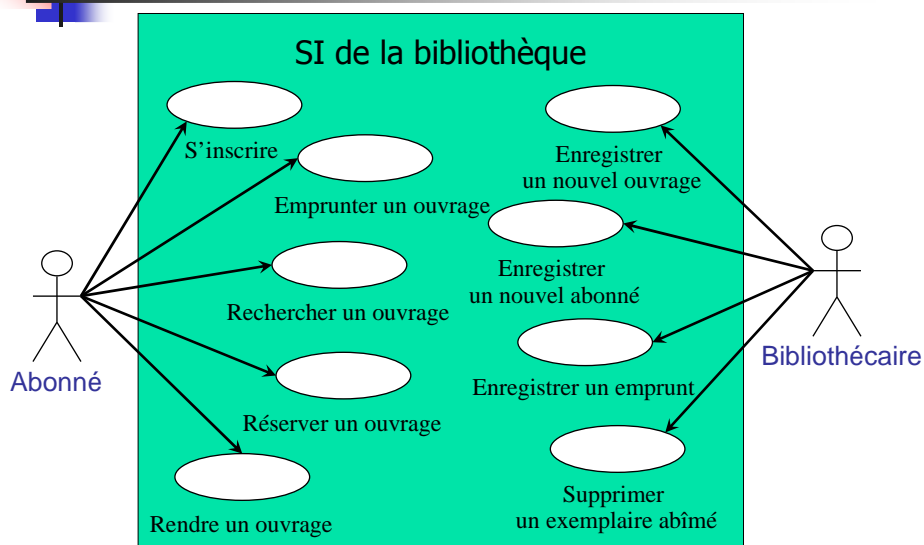


## Identification des Cas d'utilisation

- Un cas d'utilisation représente:
  - Un service que le système doit rendre à ses utilisateurs
  - Une suite complète de transactions entre l'utilisateur et le système
  - Ce qui doit être effectué par le système
- Poser les questions:
  - Quelles sont les principales tâches de chaque acteur?
  - L'acteur aura-t-il lire/écrire/modifier une information du système?
  - L'acteur devra-t-il informer le système des modifications extérieures?
  - L'acteur souhaite-t-il être informé de modifications inopinées?

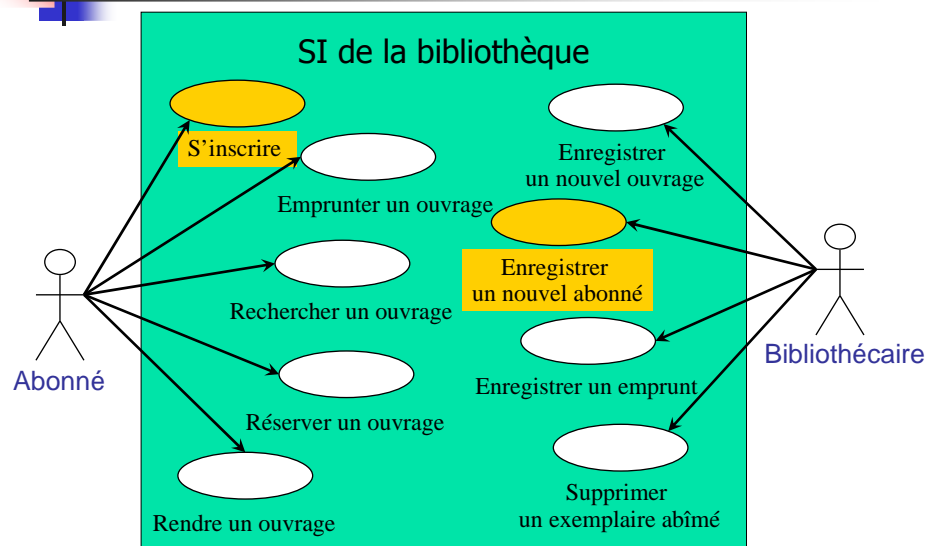


## Cas d'utilisation *Exemple: Bibliothèque*



## Cas d'utilisation

### Exemple: Bibliothèque

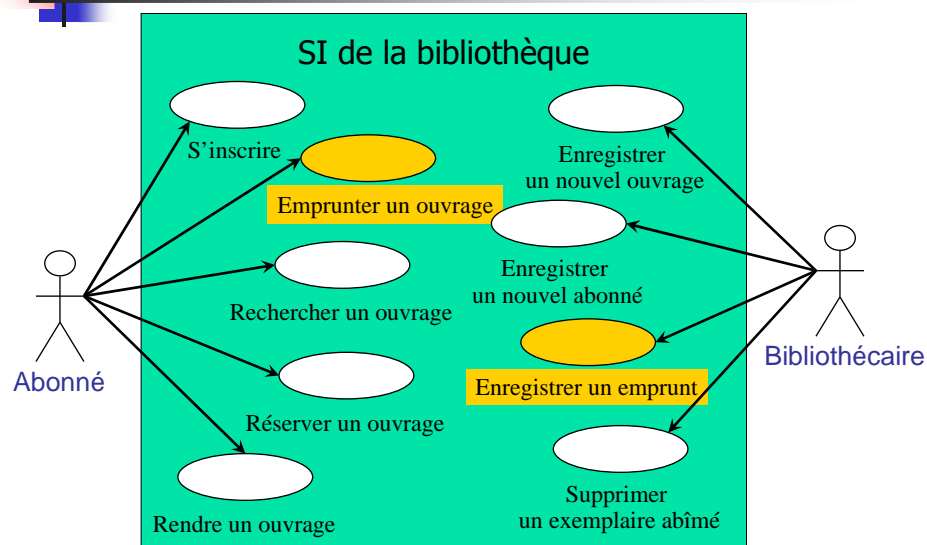


©J. Ralyté, Université de Genève

13

## Cas d'utilisation

### Exemple: Bibliothèque

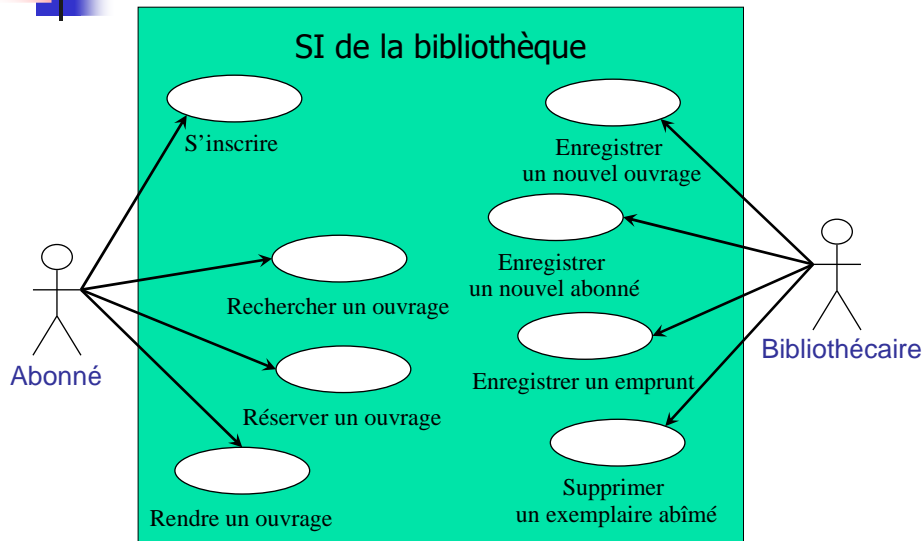


©J. Ralyté, Université de Genève

14

## Cas d'utilisation

### Exemple: Bibliothèque



©J. Ralyté, Université de Genève

15

## Description des cas d'utilisation

- **Scénario de base** : le scénario principal donnant la meilleure compréhension du cas d'utilisation. L'acteur atteint son objectif.
  - Ex. : *Enregistrer un emprunt*
- **Scénario d'exception** : le scénario décrivant une variante du scénario de base correspondant à un fonctionnement exceptionnel.
  - Ex. : *Abonné est suspendu, Le nombre d'emprunts autorisé est atteint, ...*

*Un cas d'utilisation a un scénario de base unique et peut avoir plusieurs scénarios d'exception*

©J. Ralyté, Université de Genève

16





## Description des cas d'utilisation

### Exemple : Bibliothèque

#### Cas d'utilisation: *Enregistrer un emprunt*

##### Scénario de base :

1. La bibliothécaire sélectionne la fonctionnalité '*Enregistrer emprunt*'
2. Le système demande le nom de l'abonné
3. La bibliothécaire inscrit le nom de l'abonné
4. Le système affiche l'information concernant les emprunts en cours et l'état de l'abonné
5. Si l'abonné est actif et le nombre d'emprunts est inférieur à 3 la bibliothécaire inscrit le nouveau emprunt
6. Le système affiche le message '*Emprunt enregistré*'.

##### Scénarios d'exception :

##### *Le nombre d'emprunts autorisé est atteint*

- 5.1 Si l'abonné a déjà 3 livres empruntés le système affiche le message '*Interdiction d'emprunter plus de 3 livres*'.



## Description des cas d'utilisation

### Exemple : Bibliothèque

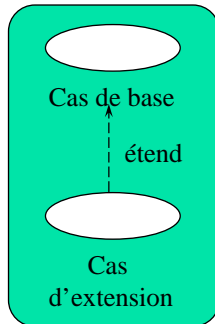
#### Cas d'utilisation: *Rechercher un ouvrage*

##### Scénario de base :

1. L'abonné se connecte sur le système de la bibliothèque.
2. Le système vérifie si les paramètres de connexion sont correctes.
3. Si les paramètres de connexion sont corrects le système affiche tous les services que l'abonné peut utiliser.
4. L'abonné sélectionne le service '*Rechercher par mot clé*'
5. Le système demande d'insérer le mot clé et valider.
6. L'abonné insère le mot clé et valide.
7. Le système sélectionne tous les ouvrages correspondant à ce mot clé et affiche la liste en précisant pour chaque ouvrage s'il est disponible et dans quel endroit il est placé.
8. L'abonné se déconnecte.



## Affinement des cas d'utilisation Association d'extension

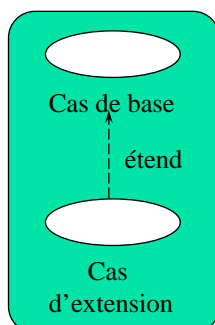


**L'association d'extension** permet d'étendre un cas d'utilisation déjà complet pour modéliser:

- des parties optionnelles du cas d'utilisation
  - des scénarios de remplacement complexes qui n'arrivent que rarement
  - des sous-scénarios qui ne s'exécutent que dans certains cas
  - le fait que différents cas d'utilisation peuvent être insérés dans un cas d'utilisation particulier ou un système de menu avec des options
- **Le cas d'extension définit :**
- le point d'extension – l'endroit dans le cas principal où le cas d'extension doit être inséré
  - la condition d'extension



## Affinement des cas d'utilisation Association d'extension

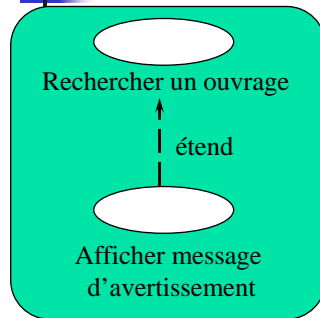


Comment se passe l'exécution d'un cas d'extension :

1. L'exécution du cas de base est interrompu par le cas d'extension si la condition d'extension est satisfaite
2. Le cas d'extension est exécuté complètement
3. L'exécution du cas de base est repris à l'endroit où il a été interrompu
4. Le cas de base est exécuté complètement

## Affinement des cas d'utilisation

### Exemples d'extensions : Bibliothèque



#### Cas d'utilisation: *Rechercher un ouvrage*

Scénario de base :

1. L'abonné se connecte sur le système de la bibliothèque.
2. Le système vérifie si les paramètres de connexion sont correctes.
3. Si les paramètres de connexion sont corrects le système affiche tous les services que l'abonné peut utiliser.
4. L'abonné sélectionne le service *Rechercher par mot clé*.
5. Le système demande d'insérer le mot clé.

#### Cas d'utilisation: *Afficher message d'avertissement*

Point d'extension: après la connexion

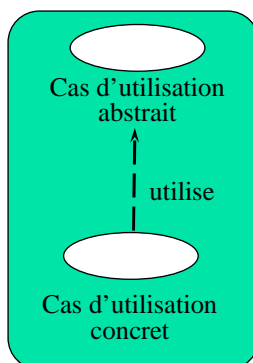
1. S'il existe un message d'avertissement pour cet abonné, le système affiche le message et demande de le valider
2. L'abonné clique sur OK

©J. Ralyté, Université de Genève

21

## Abstraction des cas d'utilisation

### Association d'utilisation



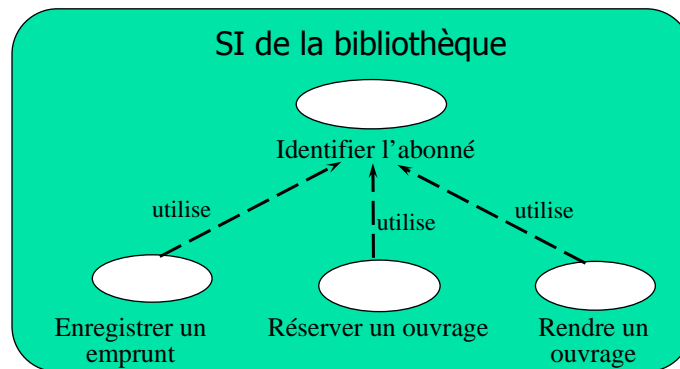
- **Cas d'utilisation abstrait** : description des parties communes à plusieurs cas d'utilisation
  - une description d'un cas d'utilisation abstrait est réutilisée par plusieurs d'autres cas d'utilisation
  - un cas d'utilisation abstrait n'a pas d'instances en tant que tel
- **Cas d'utilisation concret** : un cas d'utilisation qui est réellement exécuté

©J. Ralyté, Université de Genève

22

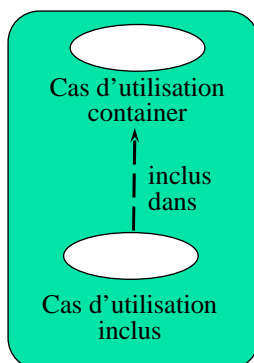
## Abstraction des cas d'utilisation

### Exemple d'association d'utilisation



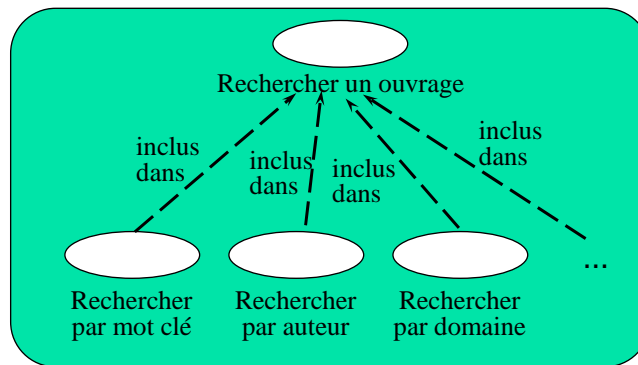
## Organisation des cas d'utilisation

### Association d'inclusion (UML)



- **Cas d'utilisation container** : un cas d'utilisation qui inclut le comportement d'un autre cas d'utilisation.  
Il définit à quel moment et sous quelle condition le cas inclus peut être réalisé.
- **Cas d'utilisation inclus** : un cas d'utilisation qui est réalisé dans le cadre d'un autre cas d'utilisation.

## Organisation des cas d'utilisation Association d'inclusion



## Description des interfaces

Les descriptions d'interfaces spécifient en détail ce à quoi ressemble l'interface utilisateur lorsque l'on exécute les cas d'utilisation

### Moyens :

- description détaillée de l'interface utilisateur pour chaque cas d'utilisation
  - les menus, les boutons, les messages que l'utilisateur verra sur l'écran,
  - les boutons, les fenêtres sur les appareils à utiliser
- Programmation des interfaces :
  - Formes Access
  - Pages HTML
  - Visual Basic
  - Editeurs graphiques
  - ...



## Modèle de domaine du problème

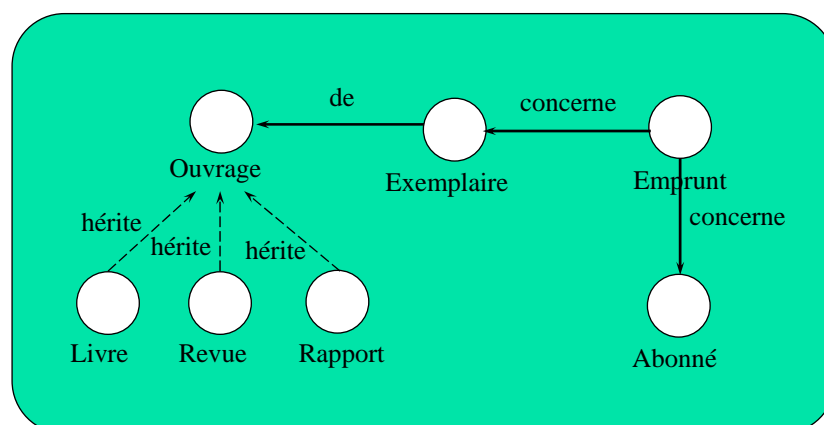
**But :** donner une image conceptuelle et une meilleure compréhension du système

- Identification des objets du domaine du problème
- Identification des associations statiques entre ces objets
- Identification des liens d'héritage entre les objets



## Modèle de domaine du problème

*Exemple: Bibliothèque*



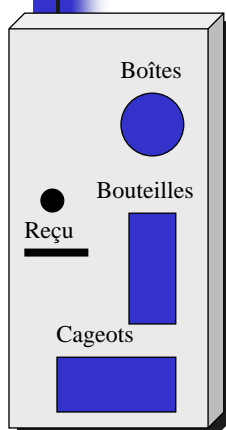


## La suite...

- Validation des cas d'utilisation avec les futurs utilisateurs
- Conception du système à base des cas d'utilisation :
  - Identification des objets pour les modèles statiques
  - Identification des événement, des opérations, des contraintes pour les modèles dynamiques:
    - modèles d'activité
    - modèles de collaboration
    - modèles de cycle de vie d'objet
  - Identification des modules, des procédures, des programmes etc. pour les modèles d'architecture:
    - modèles de composants
    - modèles de déploiement
- Identification des besoins non-fonctionnels



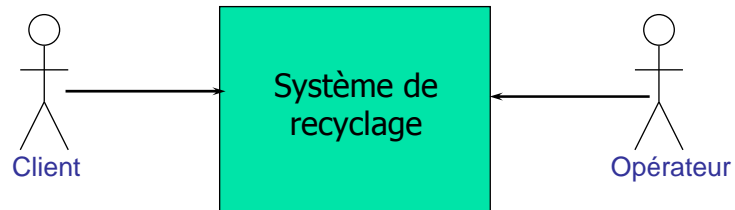
## Exercice: Système de recyclage



- Le système commande une machine de recyclage pour les bouteilles, les boîtes et les cageots consignés.
- Plusieurs clients peuvent utiliser la machine en même temps, et chacun peut rapporter les trois types d'éléments à la fois.
- Le système doit vérifier le type de chaque élément.
- Le système enregistre le nombre d'éléments rapportés par un client et, lorsque le client demande un reçu, le système imprime ce qu'il a reçu, la valeur des éléments rendus et la somme totale du retour, qui sera payée au client.
- Un opérateur utilise le système pour savoir combien d'éléments de chaque type ont été rendus durant la journée et demande une impression du nombre total d'éléments déposés.
- L'opérateur doit être capable de modifier les informations du système, telles que la valeur de dépôt des éléments.
- Un signal d'alarme spécial alerte l'opérateur si, par exemple, une boîte est coincée, ou si le rouleau de papier pour les reçus est terminé.

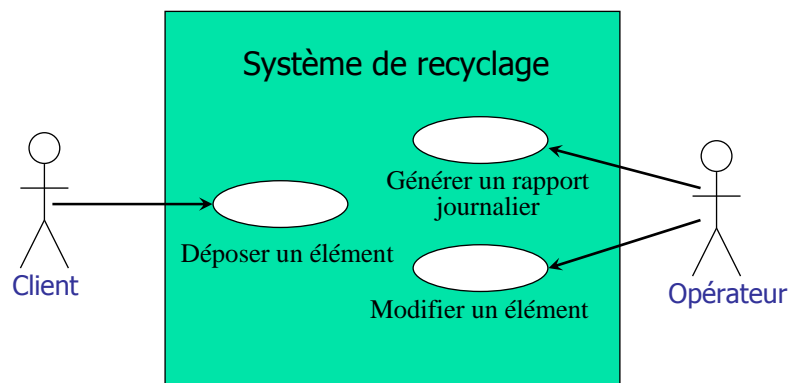
## Exercice : Système de recyclage

### Identification des Acteur



## Exercice : Système de recyclage

### Identification des Cas d'utilisation







## *Exercice : Système de recyclage*

### Description des cas d'utilisation

#### **Cas d'utilisation :** *Déposer un élément*

Scénario de base:

1. Le client dépose un article consigné.
2. Le système mesure l'article et détermine de quel type de bouteille, de boîte ou de cageot il s'agit.
3. Si l'article est accepté, le système incrémente le total du client de même que le total journalier concernant ce type d'élément particulier.
4. Le client appuie sur le bouton de reçu.
5. Le système calcule le total du client et imprime les informations suivantes sur le reçu; pour chaque type d'article: le nom, le nombre de dépôts effectués, la valeur de dépôts pour le type d'article, le total pour ce type, la somme que le client doit recevoir.



## *Exercice : Système de recyclage*

### Description des cas d'utilisation

#### **Cas d'utilisation :** *Déposer un élément*

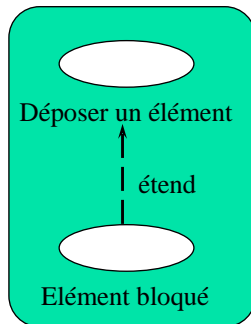
Scénario d'exception: *Article non valide*

1. Si l'article n'est pas accepté, la lumière 'NON VALIDE' s'allume sur le panneau.
2. Le client retire l'article.

## Exercice : Système de recyclage

### Affinement des cas d'utilisation

Lorsqu'un élément est bloqué, le système doit pouvoir donner l'alarme

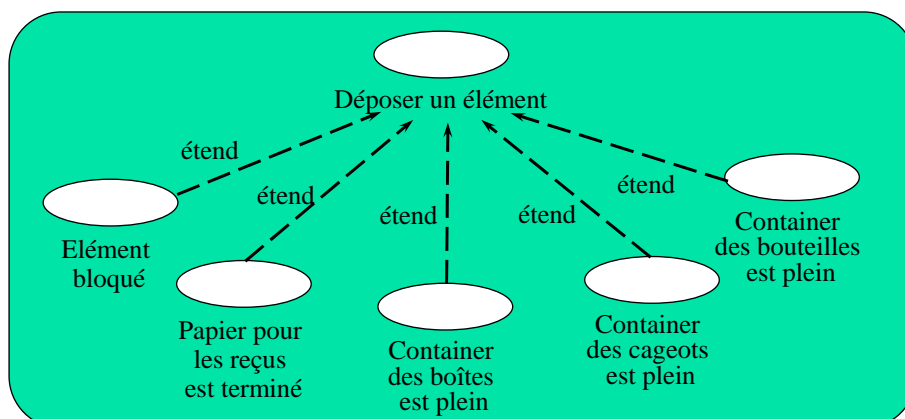


#### Cas d'utilisation: *Elément bloqué*

1. Lorsqu'un article se coince, l'alarme est activée pour appeler l'opérateur.
2. Lorsque l'opérateur a retiré l'article coincé, il réinitialise l'alarme et le client peut continuer à déposer des articles. Le total acquis jusqu'alors par le client est toujours valide. Le client n'acquiert aucun crédit sur l'article coincé.

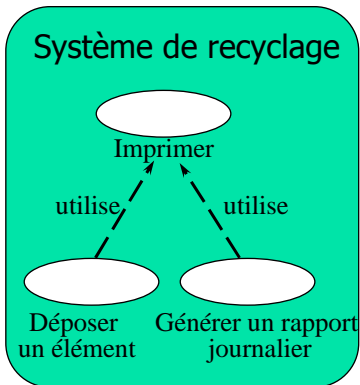
## Exercice : Système de recyclage

### Affinement des cas d'utilisation



## Exercice : Système de recyclage

### Abstraction des cas d'utilisation



- Les cas d'utilisation, '*Déposer un élément*' et '*Générer un rapport journalier*', tous les deux impriment un reçu.
- On identifie un cas d'utilisation abstrait '*Imprimer*' qui est ainsi réutilisé par ces deux cas d'utilisation.

## Exercice : Système de recyclage

### Modèle de domaine du problème

- Les concepts principaux sont: la *Boîte*, la *Bouteille* et le *Cageot*
- Ces objets sont traités de la même manière, on identifie une classe abstraite *Élément consigné*
- Le système doit imprimer un *Reçu* au *Client*

