

# DIAGRAMME DES CAS D'UTILISATION

Analyse et conception orientée objet



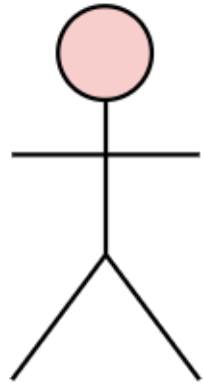
[www.elbahihassan.com](https://www.elbahihassan.com)



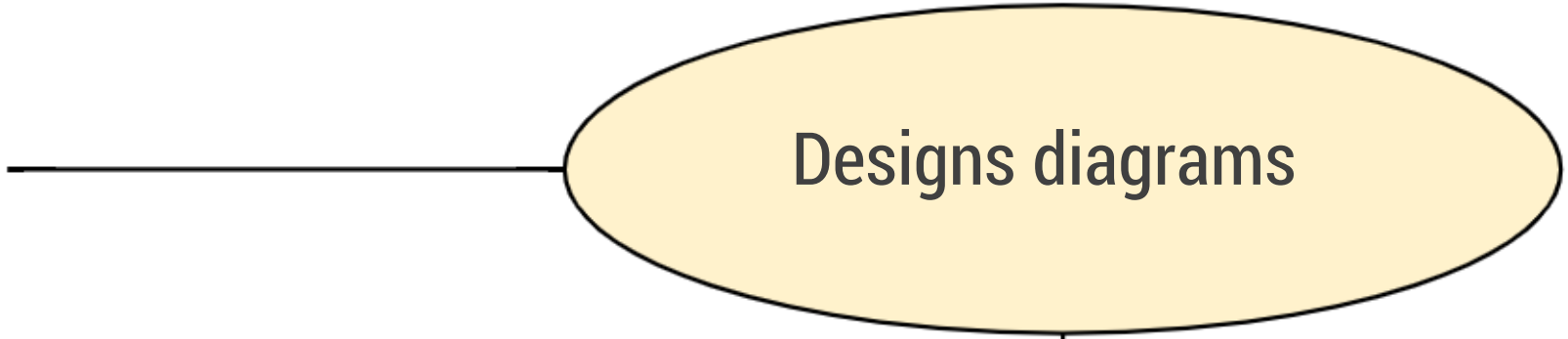
[elbahihassan@gmail.com](mailto:elbahihassan@gmail.com)



ISTA Oulad Teima

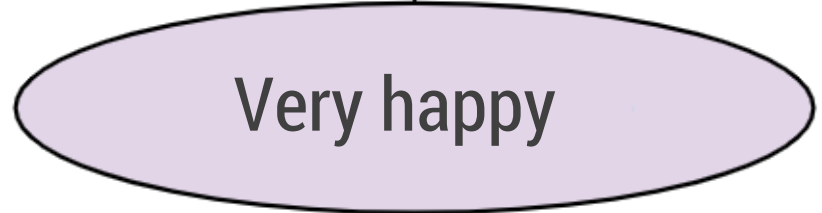


User



<<Include>>

feels



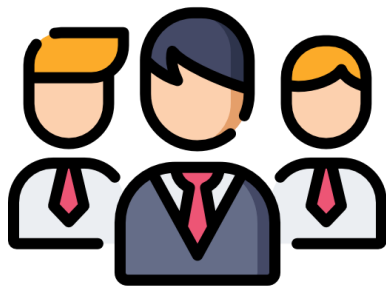
# Qu'est-ce que le diagramme des **cas d'utilisation** ?

Le diagramme des cas d'utilisation (**Use Case Diagram**) constitue la première étape de l'analyse UML en :

- Modélisant les **besoins** des utilisateurs.
- Identifiant les grandes **fonctionnalités** et les limites du système.
- Représentant les **interactions** entre le système et ses utilisateurs.

# Les éléments d'un diagramme des cas d'utilisation

Le diagramme de cas d'utilisation se compose généralement de deux éléments :



**Acteurs**

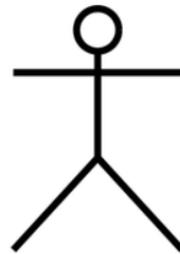


**Cas d'utilisations**

# Les acteurs

Les éléments d'un diagramme des cas d'utilisation

- Avant de rechercher les besoins, la première tâche consiste à identifier les différentes entités intervenants sur le système. Ces entités sont appelés **acteurs**.
- Les acteurs se représentent sous la forme d'un petit personnage (**stick man**) ou sous la forme d'une case rectangulaire (appelé **classeur**) avec le mot clé « actor ». Chaque acteur porte un **nom**.



Client



Un exemple d'un acteur qui porte le nom client

# Les acteurs

Les éléments d'un diagramme des cas d'utilisation

Parmi les acteurs, nous distinguons :

- les **acteurs principaux** agissent directement sur le système. Il s'agit d'entités qui ont des besoins d'utilisation du système. On peut donc considérer que les futurs utilisateurs du logiciel sont les acteurs principaux.
- les **acteurs secondaires** n'ont pas de besoin direct d'utilisation. Ils peuvent être soit consultés par le système à développer, soit récepteur d'informations de la part du système. Cela est généralement un autre système (logiciel) avec lequel le nôtre doit échanger des informations.

# Les acteurs

Les éléments d'un diagramme des cas d'utilisation

Parmi les acteurs, nous distinguons :

- les **acteurs principaux** agissent directement sur le système. Il s'agit d'entités qui ont des besoins d'utilisation du système. On peut donc considérer que les futurs utilisateurs du logiciel sont les acteurs principaux.
- les **acteurs secondaires** n'ont pas de besoin direct d'utilisation. Ils peuvent être soit consultés par le système à développer, soit récepteur d'informations de la part du système. Cela est généralement un autre système (logiciel) avec lequel le nôtre doit échanger des informations.

# Les acteurs

## Les éléments d'un diagramme des cas d'utilisation

**Exemple :** Le **DAB** (Distributeur Automatique de Billet). Nous utiliserons cet exemple tout le long du cours :

- Un DAB permet à tout détenteur de carte bancaire de retirer de l'argent.
- Si le détenteur de carte est un client de la banque propriétaire du DAB, il peut en plus consulter les soldes de ses comptes et effectuer des virements entre ces différents comptes.
- Les transactions sont sécurisées c'est-à-dire :
  - Le DAB consulte le Système d'Information de la banque (S.I. Banque) pour les opérations que désire effectuer un client de la banque (retraits, consultation soldes et virements).
  - Le DAB consulte le Système d'Autorisation Globale Carte Bancaire (Sys. Auto.) pour les retraits des porteurs de cartes non clients de la banque.
- Le DAB nécessite des opérations de maintenance tel que la recharge en billet, la récupération des cartes avalée, etc.

**Question :** Quels sont les différents **acteurs** interagissant avec le DAB ?



# Les acteurs

Les éléments d'un diagramme des cas d'utilisation

- Les acteurs principaux :



**Porteur de carte**



**Client banque**



**Opérateur de maintenance**

- les acteurs secondaires :



**S.I. banque**

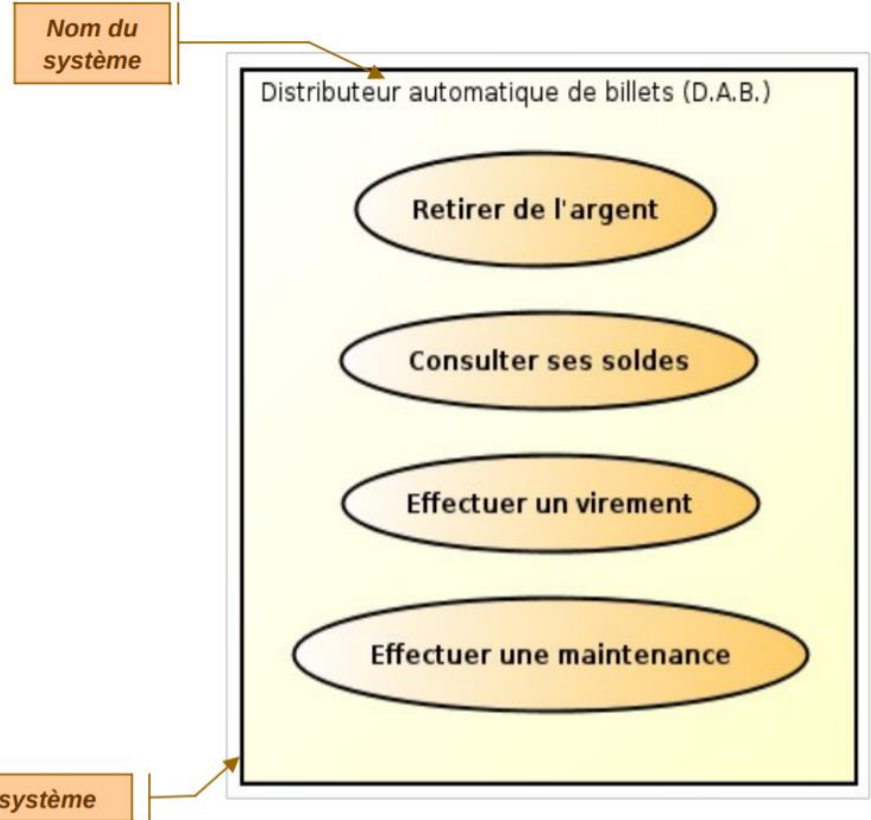


**Sys. Auto**

# Les cas d'utilisation

Les éléments d'un diagramme des cas d'utilisation

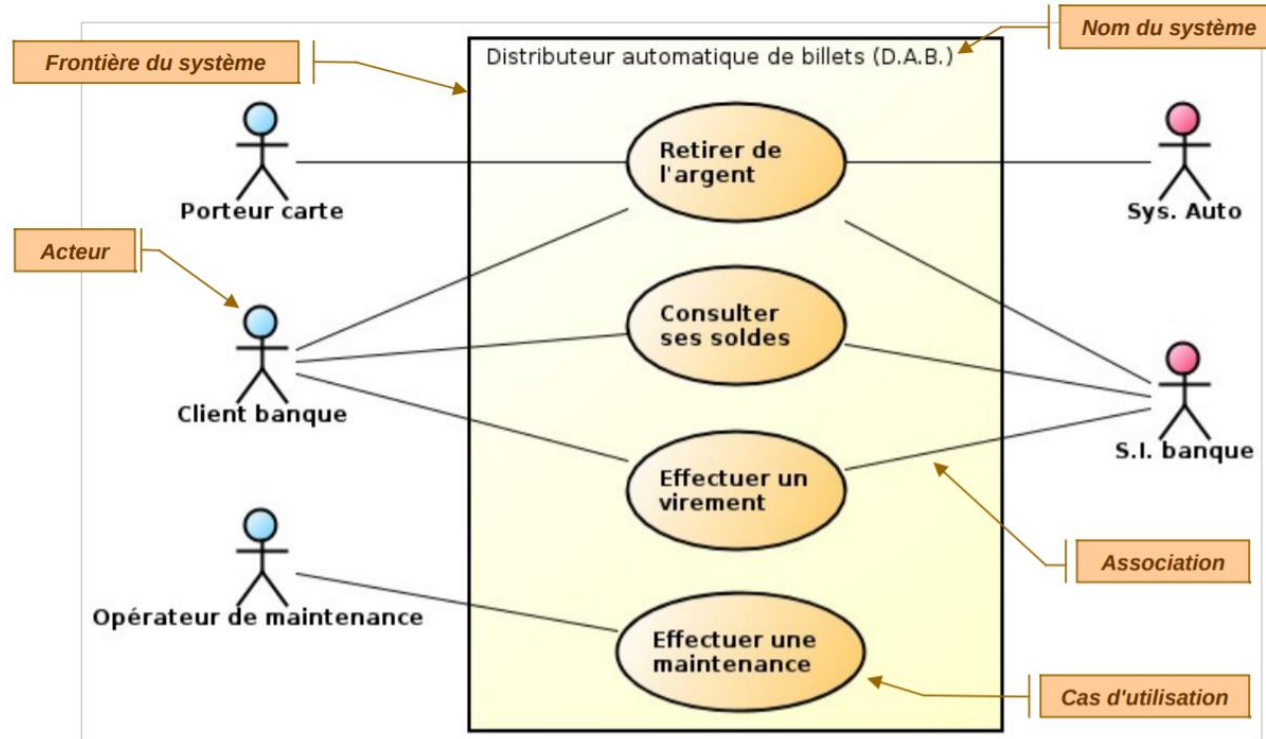
- Le cas d'utilisation représente une **fonctionnalité** du système.
- Un cas d'utilisation se représente par une **ellipse** contenant le nom du cas d'utilisation (phrase commençant par un verbe à l'**infinitif**).
- Les différents cas d'utilisation peuvent être représentés à l'intérieur d'un même **rectangle** représentant les limites du système.



# Relation entre acteurs et cas d'utilisation

Les éléments d'un diagramme des cas d'utilisation

- Chaque acteur est associé un ou plusieurs cas d'utilisations.
- Les acteurs principaux sont placés à gauche et les acteurs secondaires à droite du système.



# Les **relation** entre les cas d'utilisation

---

- Les cas d'utilisation que nous avons découvert dans les diapositives précédentes sont directement liés à un acteur et sont appelés des « **cas d'utilisation principales** ».
- Chacun de ces cas d'utilisation nécessitera un certain nombre d'actions. Il arrive que l'on doive regrouper certaines actions dans un ou plusieurs cas d'utilisation complémentaires qui ne sont pas directement liés à un acteur. On parle alors d'un **cas d'utilisation interne**.
- Les liens (ou relations) entre cas d'utilisation peuvent être divisés en deux groupes :
  - Relation d'inclusion
  - Relation d'extension

# Relation d'inclusion

Les relation entre les cas d'utilisation

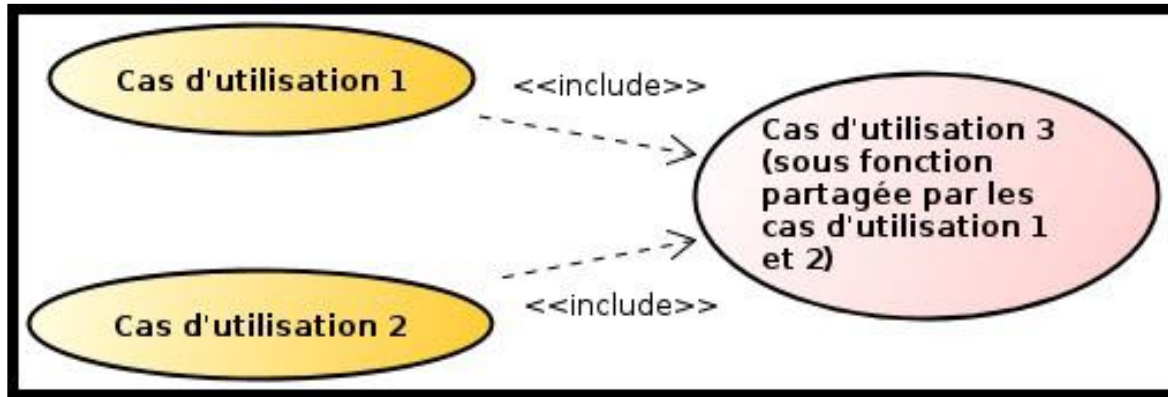
- La relation d'inclusion sert à enrichir un cas d'utilisation par un autre cas d'utilisation (c'est une sous fonction).
- **La relation d'inclusion est impérative et donc systématique.**
- Dans un diagramme des cas d'utilisation, cette relation est représentée par une flèche pointillée reliant les 2 cas d'utilisation et munie du stéréotype « **include** ».

# Relation d'inclusion

Les relation entre les cas d'utilisation

L'inclusion permet de :

1. **Partager** une fonctionnalité commune entre plusieurs cas d'utilisation :

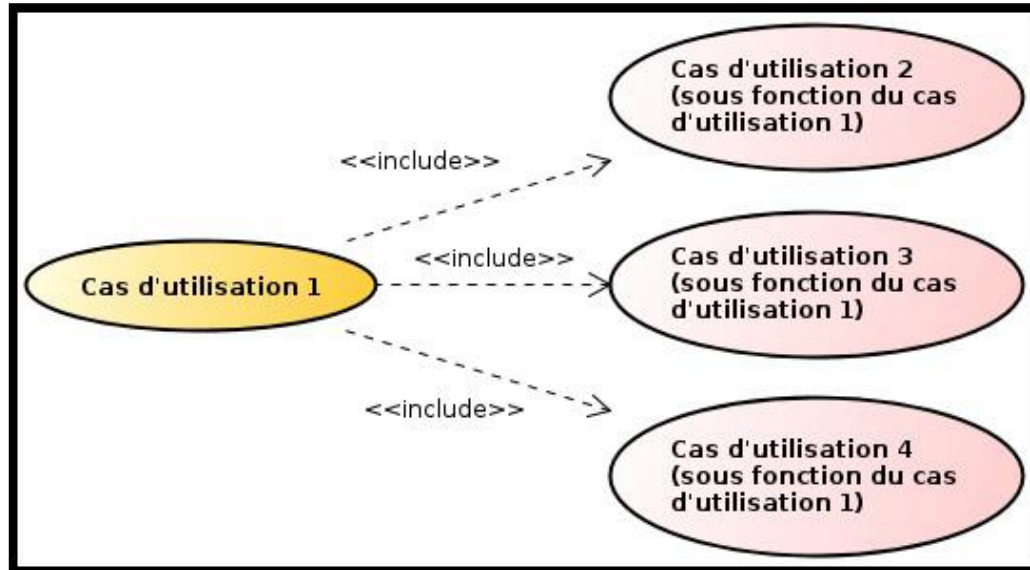


# Relation d'inclusion

Les relation entre les cas d'utilisation

L'inclusion permet de :

2. **Décomposer** un cas d'utilisation complexe en décrivant ses sous fonctions :

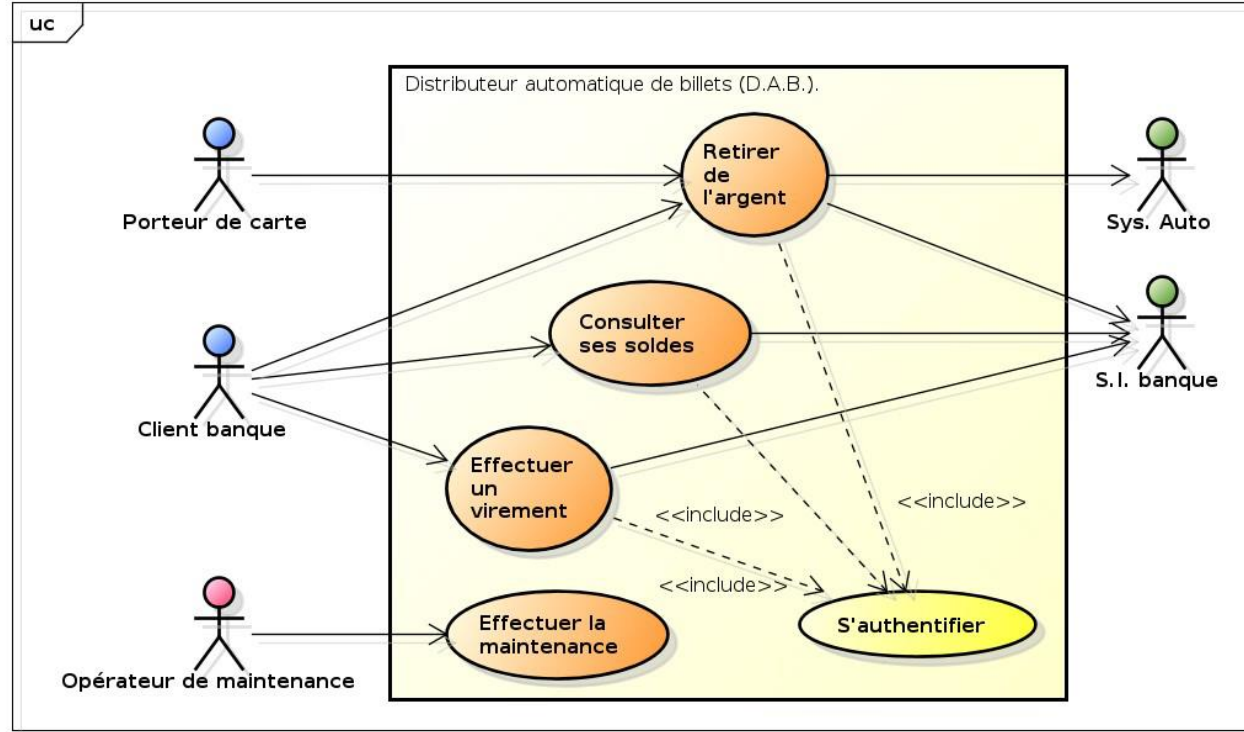


# Relation d'inclusion

Les relation entre les cas d'utilisation

Exemple : le **DAB**.

Après discussion avec l'expert métier, il apparaît que l'une des sous fonctions importantes est l'**authentification** (systématique et commune au 3 cas d'utilisations Retirer de l'argent, Consulter ses soldes et Effectuer un virement).





# Relation d'**extension**

Les relation entre les cas d'utilisation

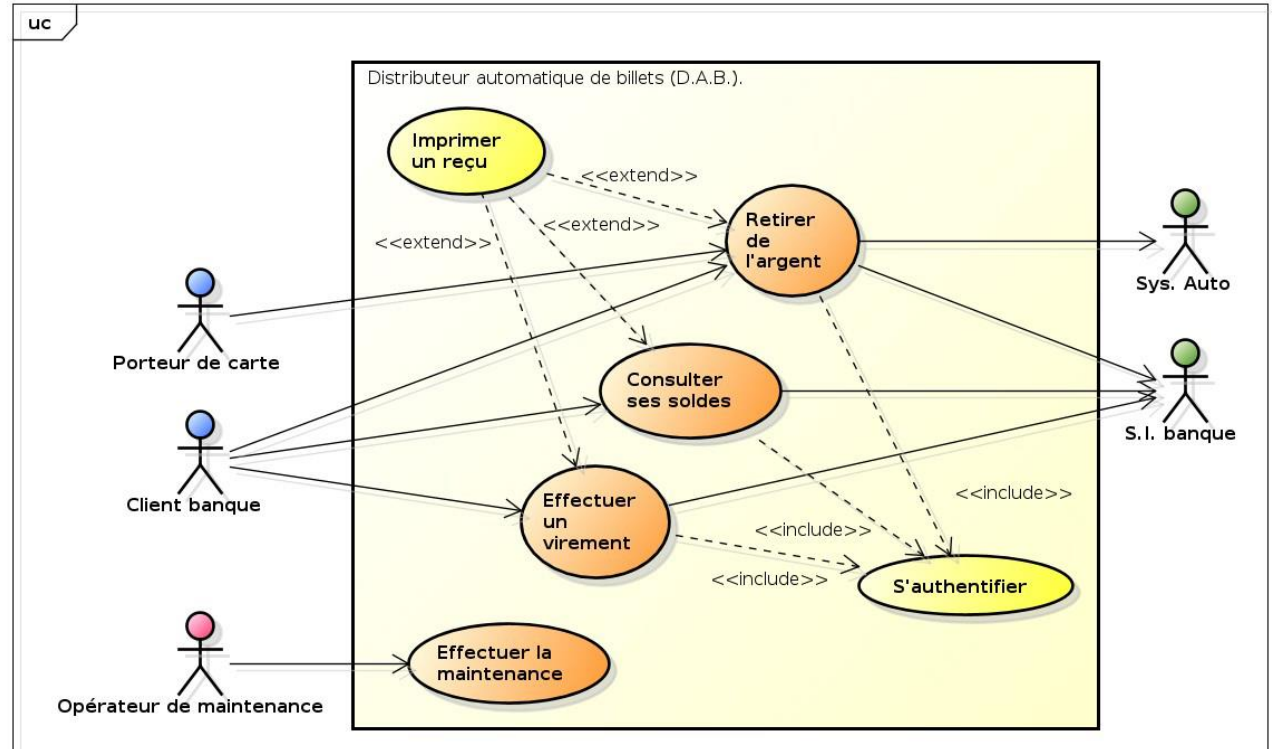
- Comme la relation d'inclusion, la relation d'extension enrichit un cas d'utilisation par un autre cas d'utilisation de sous fonction mais celui-ci est **optionnel**.
- Cette relation est représentée par une flèche en pointillée reliant les 2 cas d'utilisations et munie du stéréotype « **extend** ».



# Relation d'extension

Les relation entre les cas d'utilisation

**Exemple :** Le DAB permet à son utilisateur d'imprimer un reçu s'il le désire.



# Point d'extension

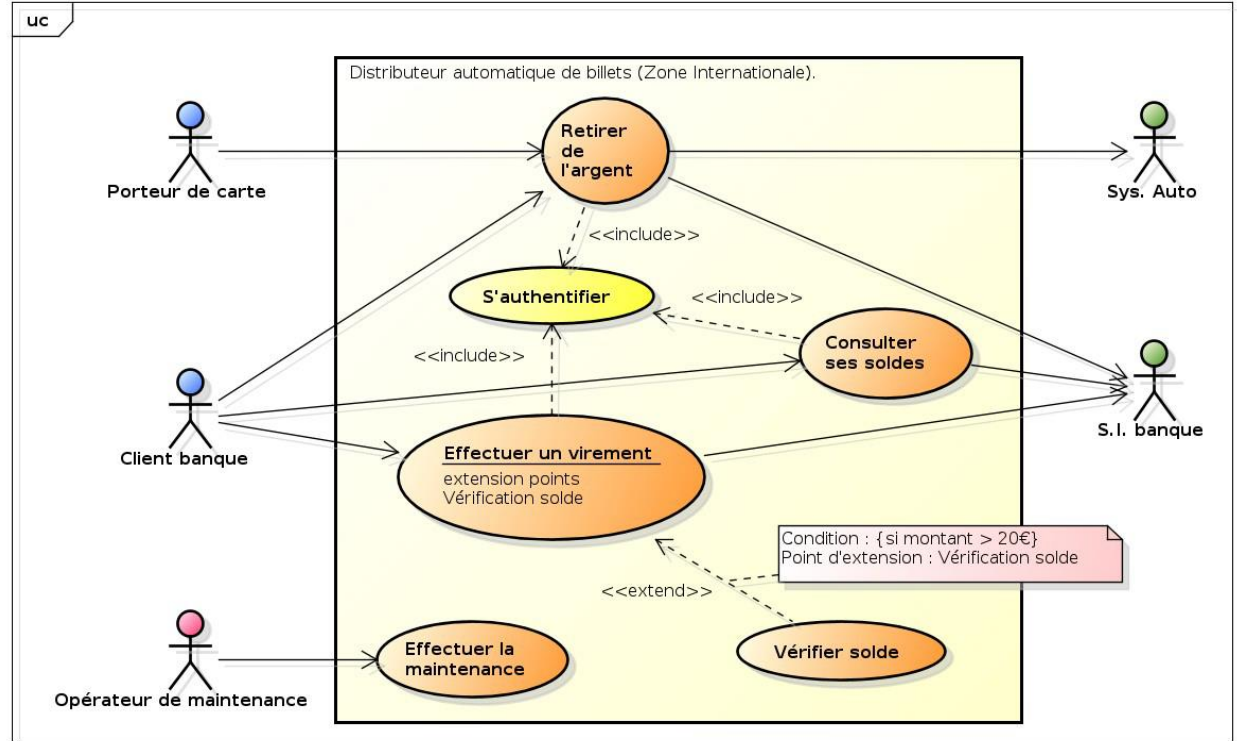
Les relation entre les cas d'utilisation

- L'extension peut intervenir à un point précis du cas étendu. Ce point s'appelle le point d'extension.
- Il porte un nom, qui figure dans un compartiment du cas étendu sous la rubrique point d'extension, et est éventuellement associé à une contrainte indiquant le moment où l'extension intervient. Une extension est souvent soumise à **condition**.

# Point d'extension

Les relation entre les cas d'utilisation

Graphiquement, la condition est exprimée sous la forme d'une note. En reprenant l'exemple du DAB, une vérification du solde du compte éventuelle n'intervient que si la demande de retrait dépasse 20 euros.

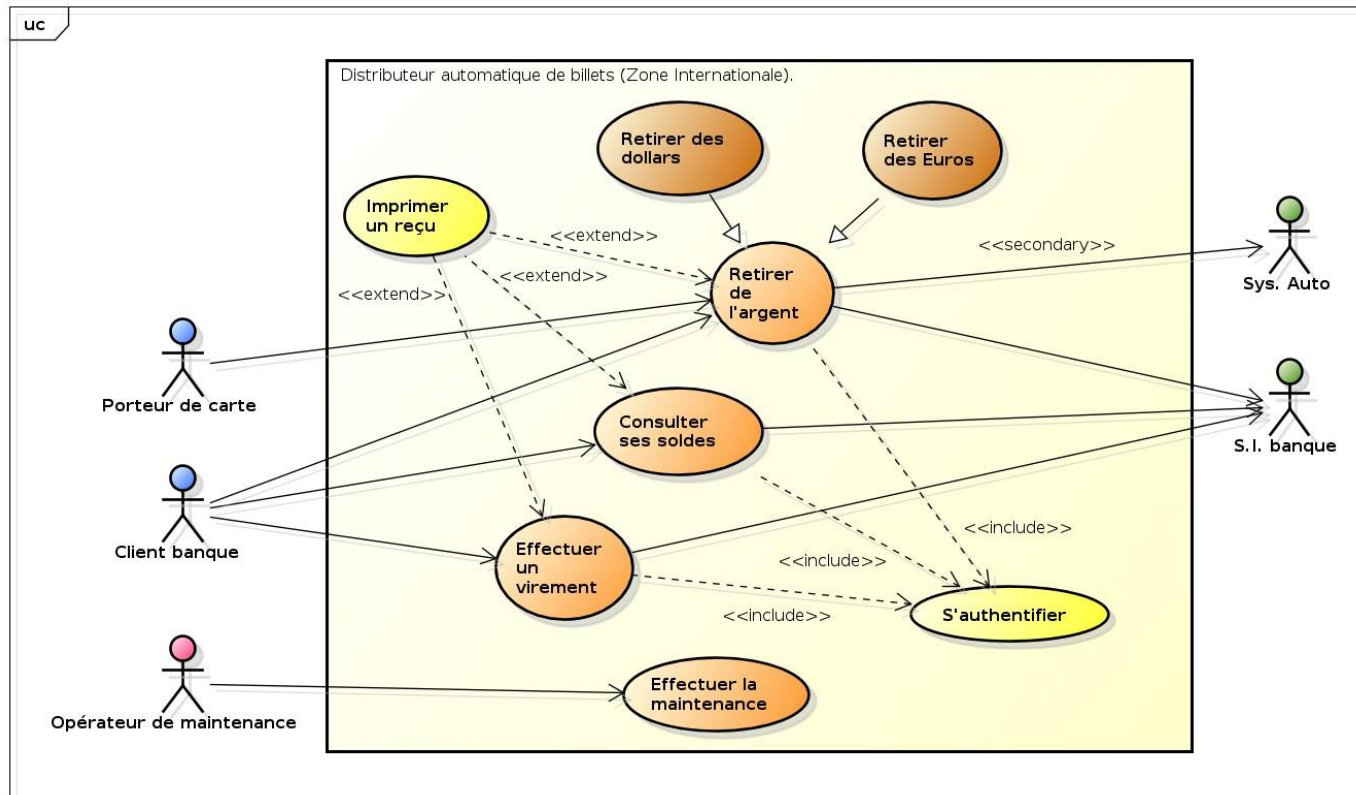


# Relation de **généralisation** ou de **spécialisation** :

- Comme l'orienté d'objet, il est également possible de **spécialiser un cas d'utilisation en un autre cas d'utilisation**. Nous obtenons alors un sous-cas d'utilisation.
- Comme pour les classes, Le sous-cas d'utilisation **hérite** aussi de toutes les associations du sur-cas.
- Quelquefois, le sur-cas d'utilisation est abstrait (c'est-à-dire qu'il ne peut pas être instancié). Il correspond à un comportement partiel et sert uniquement de base pour les sous-cas d'utilisation qui en hériteront.
- La relation de généralisation est représenté par une flèche avec une extrémité triangulaire.

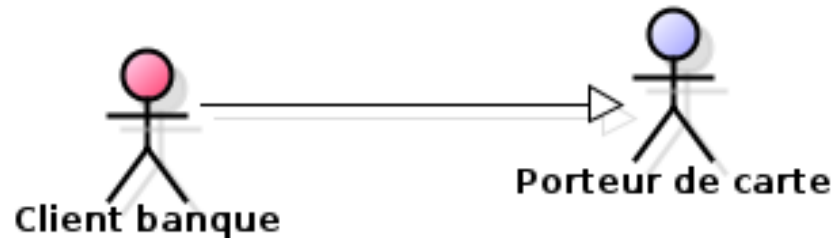
# Relation de généralisation ou de spécialisation :

L'expert métier précise que le DAB sera situé dans une zone internationale et devra donc pouvoir fournir la somme d'argent en Dollars ou en Euros.

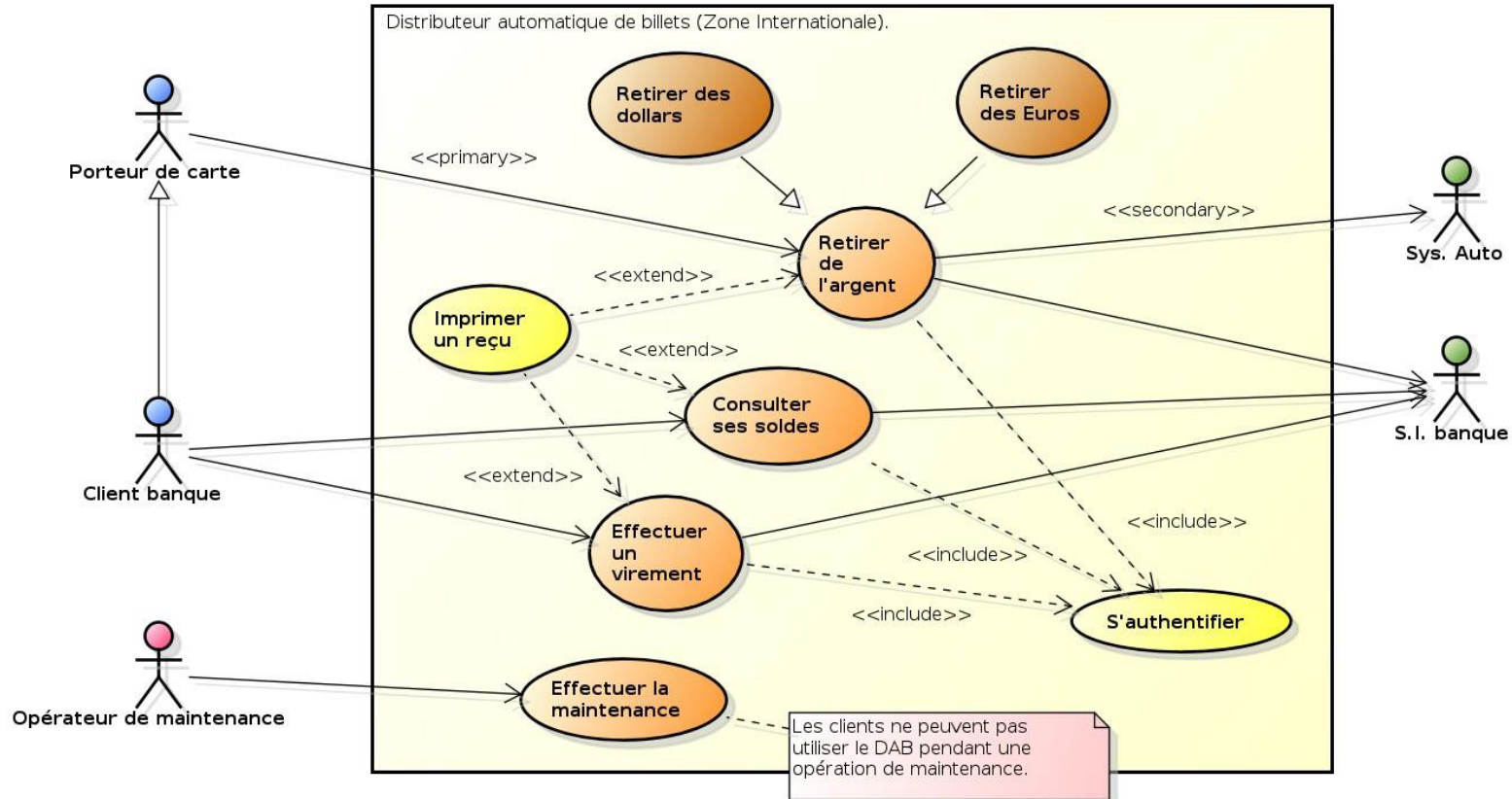


# Relation de **généralisation** entre les acteurs

- La seule relation possible entre 2 acteurs est la généralisation (même comportement et même représentation graphique que la relation de généralisation entre 2 cas d'utilisation).
- **Exemple** : Dans le cas du DAB, l'acteur **Client banque** est une spécialisation de l'acteur **Porteur de carte**.



# Relation de généralisation entre les acteurs





# Résumé

---

- Un logiciel à développer est utilisé par des **acteurs principaux**, et parfois, il peut être lié à des **acteurs secondaires**. Les acteurs secondaires échangent des informations avec le système, mais ne déclenchent aucune des fonctionnalités.
- Les fonctionnalités utiles aux acteurs sont appelées des **cas d'utilisation**. Un diagramme de cas d'utilisation permet d'illustrer **QUI devrait pouvoir faire QUOI**, grâce au système.
- Les liens entre cas d'utilisation peuvent être : d'inclusion (Impérative et systématique) ou d'extension (Optionnel).
- Une relation de généralisation peut être présent entre deux acteurs plusieurs acteurs.

