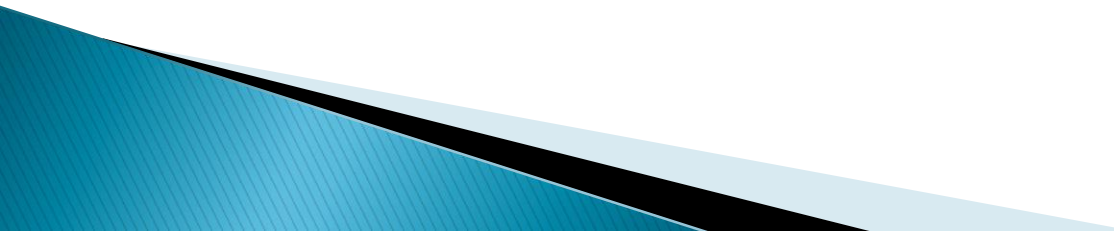


UML-Diagramme de cas d'utilisation

UML = Unified Modeling Language

- ▶ **Unified** : issu de l'unification des démarches de plusieurs chercheurs
 - ▶ **Modeling** : permet de décrire le modèle, l'abstraction d'un système
 - ▶ **Language** : langage normé, partageable, sans ambiguïté
- 

Historique

- Plus de **50 méthodes pour programmer de projets orientés objet** sont apparues durant la période 90-95: Booch, Classe-Relation, OMT, OOA, OOD, OOM, OOSE...
- UML est issu de l'unification de:
 - **OOAD** (Object-oriented Analysis and Design), par **Booch** (1991)
 - **OOSE** (Object-oriented Software Engineering), par **Jacobson** (1992)
 - **OMT** (Object Modeling Technique), par **Rumbaugh** (1991),
- La fusion des démarches créa le **U** de UML

Maîtres d'ouvrage et d'œuvre

► Maître d'ouvrage (client)

- **Ex: Une banque a besoin d'un système informatique pour ses GAB (distributeurs d'argent). Le maître d'ouvrage (notre client) doit:**
 - définir et exprimer en détail les besoins du système
 - valider les solutions proposées par le maître d'œuvre
 - valider le produit livré

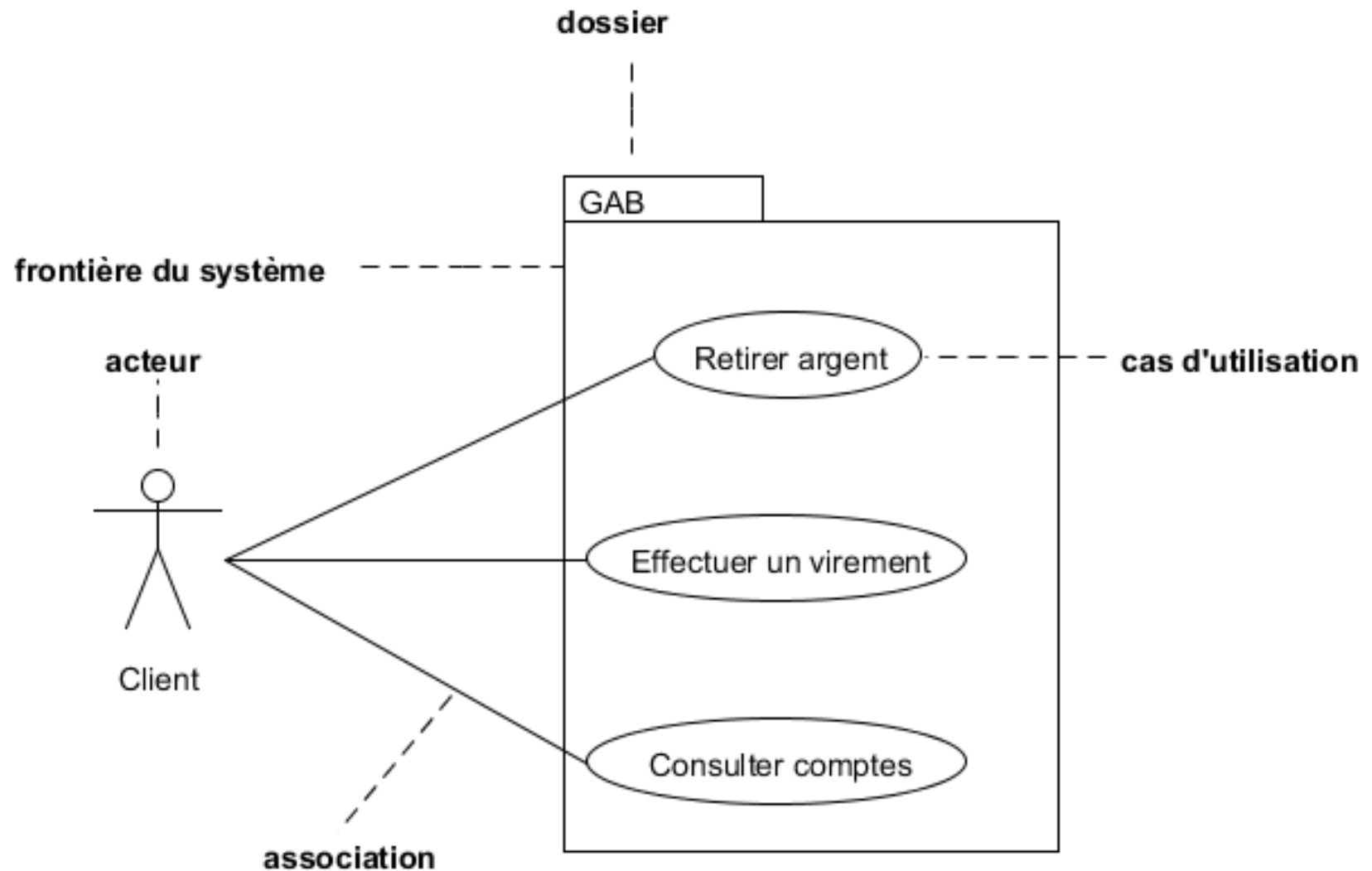
► Maître d'œuvre (développeurs)

- **Ex: une société de services informatiques (notre société) va créer leur système. Le maître d'œuvre (nous!) doit:**
 - connaître le domaine (ex: le domaine de la banque)
 - recueillir toutes les informations pour pouvoir développer le système auprès du maître d'ouvrage

Diagramme de cas d'utilisation

- ▶ C'est un diagramme **comportemental**, car il **décrit les fonctionnalités (cas d'utilisations)** offertes par le système (qu'est-ce que le système fait?)
- ▶ Le diagramme indique les **manières spécifiques d'utiliser un système**. Les **acteurs** utilisent le système et se trouvent à l'extérieur du système
- ▶ Un cas d'utilisation a un **déclenchement, un déroulement et une fin pour l'acteur** qui l'a initié
- ▶ Les diagrammes de cas d'utilisation **permettent d'exprimer les besoins des utilisateurs (client)** dès les premiers stades du développement

Exemple de diagramme de cas d'utilisation



Acteurs et cas d'utilisation

- ▶ **Acteur** : personne ou système qui interagit avec le système en échangeant de l'information.
 - les **utilisateurs directs** du système (ex: le client, le guichetier d'une banque)
 - les **responsables de son fonctionnement** (ex: l'administrateur du système)
 - **d'autres systèmes qui interagissent** avec le système (ex: le système central de la banque, contacté quand le guichetier veut consulter le solde d'un client)
- ▶ Un **acteur représente un rôle** dans le système. La même personne physique peut jouer le rôle de plusieurs acteurs

Ex: Lola est l'administratrice d'un GAB. Elle peut être aussi un client qui utilise le distributeur

et plusieurs personnes physiques peuvent jouer le même rôle et donc agir comme un même acteur

Ex: Claude et Jean sont deux clients qui utilisent le distributeur

Identification des acteurs

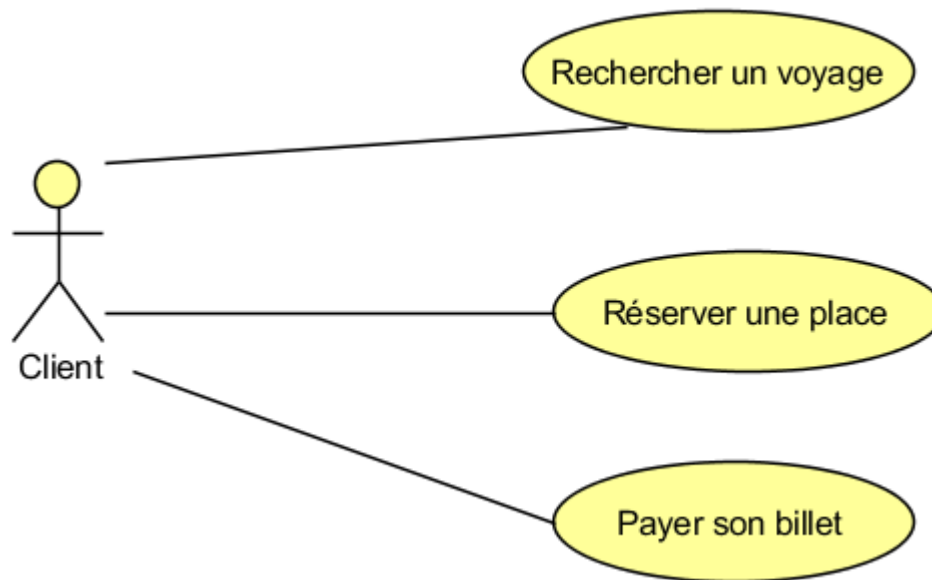
- ▶ Trouver les **utilisateurs** du système
- ▶ Penser à son **nom** et son **rôle**
- ▶ Plusieurs personnes peuvent avoir le même rôle
- ▶ Un seul acteur peut jouer plusieurs rôles
- ▶ Ne **pas oublier les administrateurs** du système
- ▶ Ne **pas oublier d'autres possibles acteurs du système**
 - d'autres logiciels
 - des systèmes informatiques externes
- ▶ Acteurs **principales** et **secondaires**
 - Ex: la personne qui retire, la banque qui vérifie le solde

Identification des cas d'utilisation (I)

- ▶ Il faut se placer du point de vue de chaque acteur
- ▶ Identifier les fonctionnalités auxquelles chaque acteur doit avoir accès
- ▶ **Éviter le séquençage temporel** et la redondance
- ▶ Ne **pas entrer trop dans les détails**: trouver un bon niveau d'abstraction

Identification des cas d'utilisation (II)

- ▶ Ex: réservation de billets de train via des bornes dans des gares
 - Considérez comme acteur à une personne qui souhaite obtenir un billet de train et interagit avec l'application web
 - **Cas d'utilisation:** rechercher un trajet entre deux villes, réserver une place, payer son billet



Relations entre les cas d'utilisation

Dans un diagramme de cas d'utilisation il y a plein de cas d'utilisation (chaque fonctionnalité est exprimée par un cas).

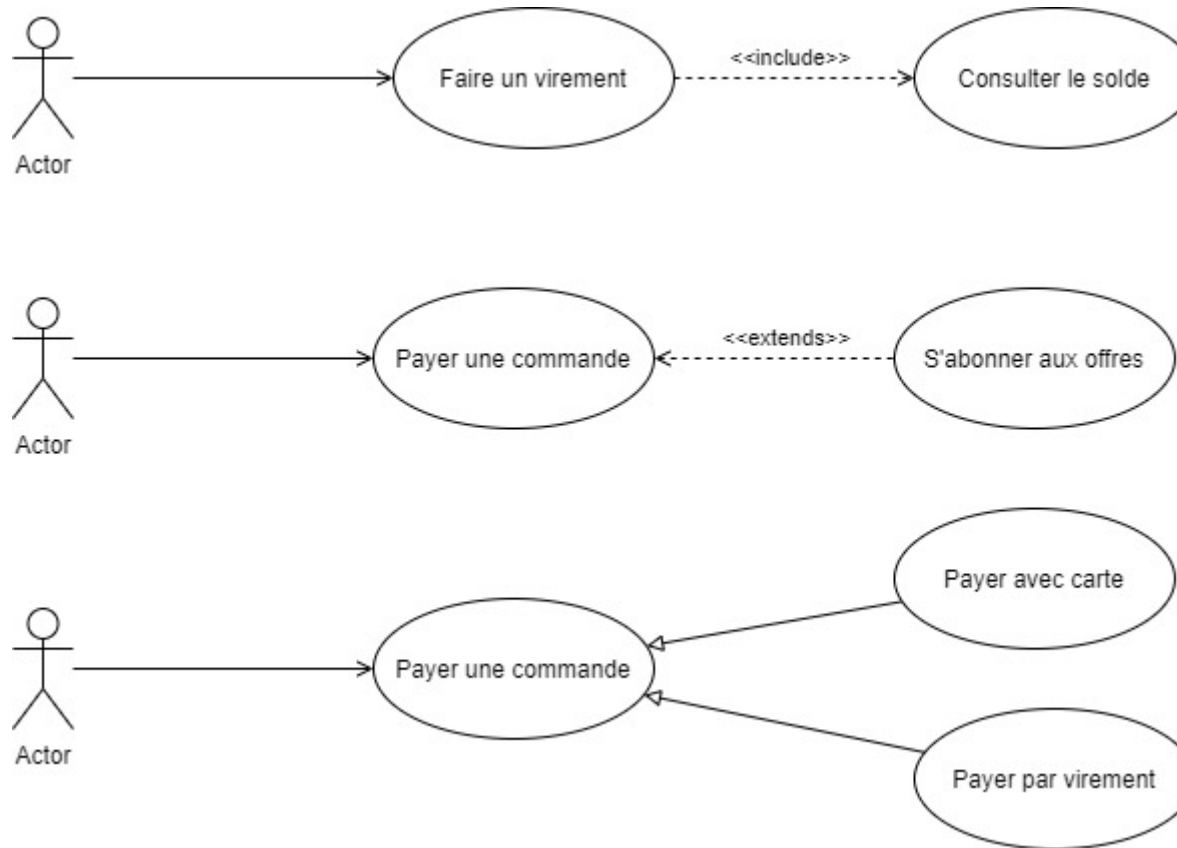
Les cas sont liés avec les acteurs, mais aussi entre eux! Il y a trois type de relations entre les cas d'utilisation

1) Inclusion: Pour qu'un cas soit complété, un autre doit être complété aussi (ex: pour faire un virement – cas 1 - il faut vérifier le solde du compte – cas 2)

2) Extension: Un cas peut être étendu avec un autre cas (ex: dans un site d'e-commerce on paie une commande – cas 1 – et on choisi de recevoir de offres sur les nouveaux produits – cas 2)

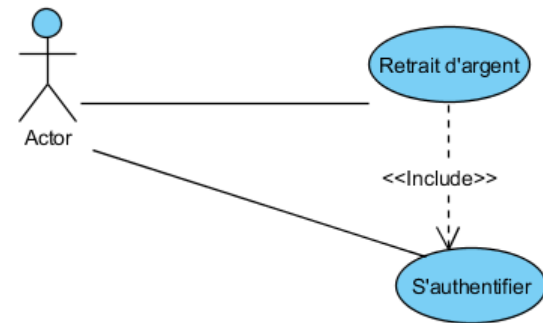
3) Héritage: Un cas 1 peut être réalisé de deux façons – cas 2 et cas 3. Le cas 1 contient la fonctionnalité commune aux deux possibilités (ex: un paiement sur un site peut être réalisé par un virement ou par carte de crédit. Dans les deux cas on doit introduire les coordonnées du client – cas 1)

Inclusion, extension et héritage

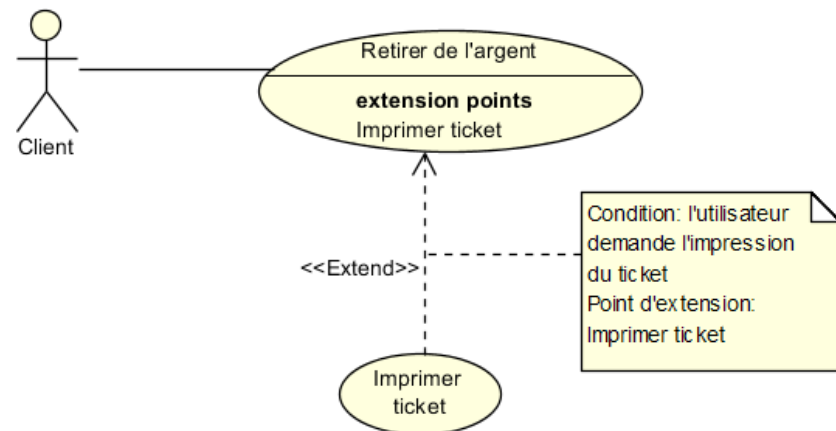


Relations entre les cas d'utilisation

1) **Inclusion:** A **<<include>>** B : pour que le cas A soit complété, les cas B doit être complété aussi. A inclut **obligatoirement** le B. Cette division en deux cas nous permet de décomposer et de factoriser un seul cas

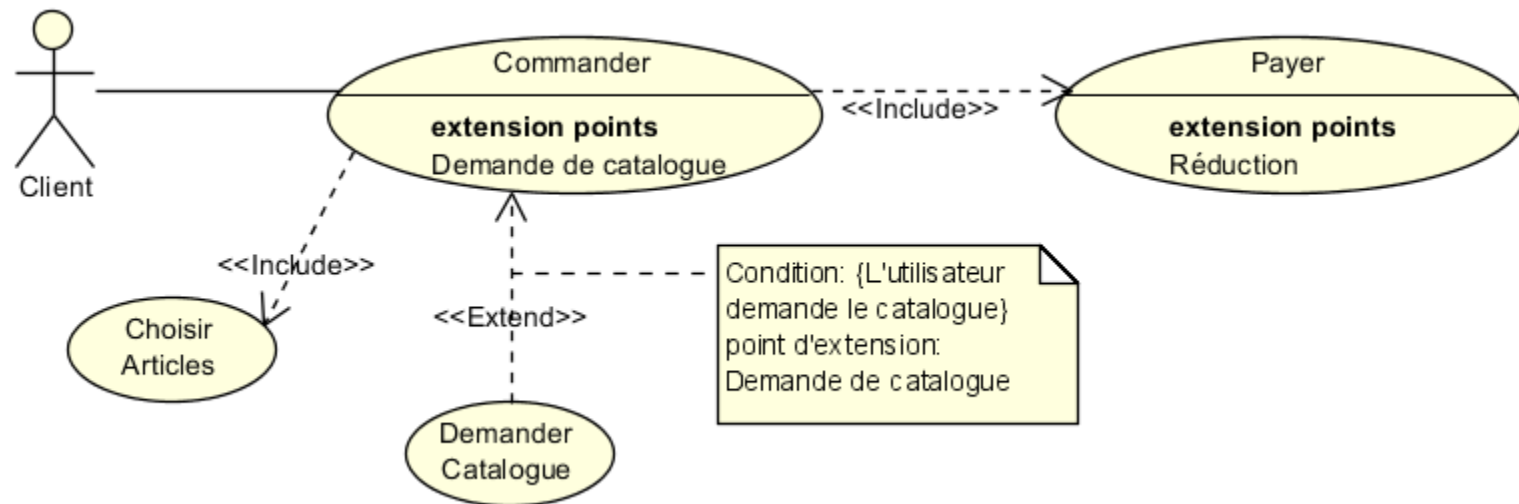


2) **Extension:** B **<<extend>>** A : le cas B est une extension **optionnelle** du cas A à un certain point de son exécution. B a lieu si une certaine condition est vraie

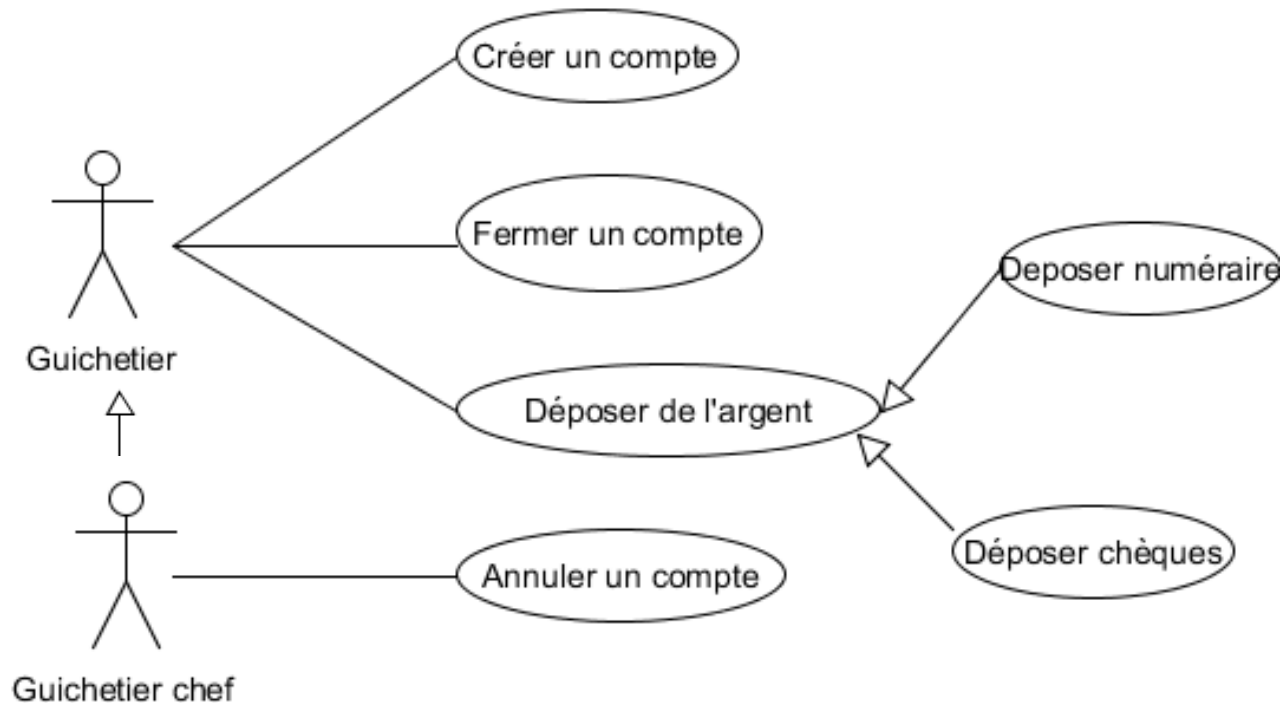


<<xxx>> est un **stéréotype** UML: un moyen de caractériser et classer des éléments des modèles UML; certains sont prédéfinis, mais les utilisateurs peuvent en définir d'autres.

Exemple: commande online



3) **Héritage** entre les acteurs et entre les cas. Appelée aussi **généralisation/spécialisation**.

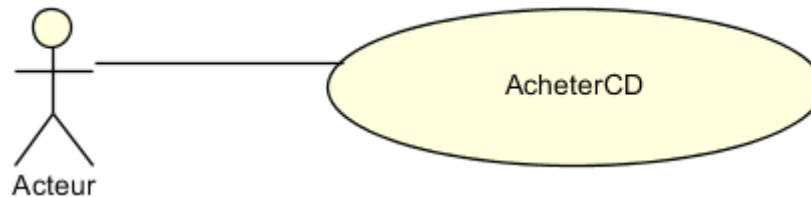


Un 'Guichetier Chef' est un 'Guichetier' spécialisé qui peut exécuter toutes les actions d'un Guichetier et, en plus, il peut annuler un compte. L'héritage simplifie le dessin (moins d'interactions à dessiner). 'Déposer chèques' et 'Déposer numéraire' sont 2 spécialisations de 'Déposer de l'argent sur un compte' (2 manières de faire).

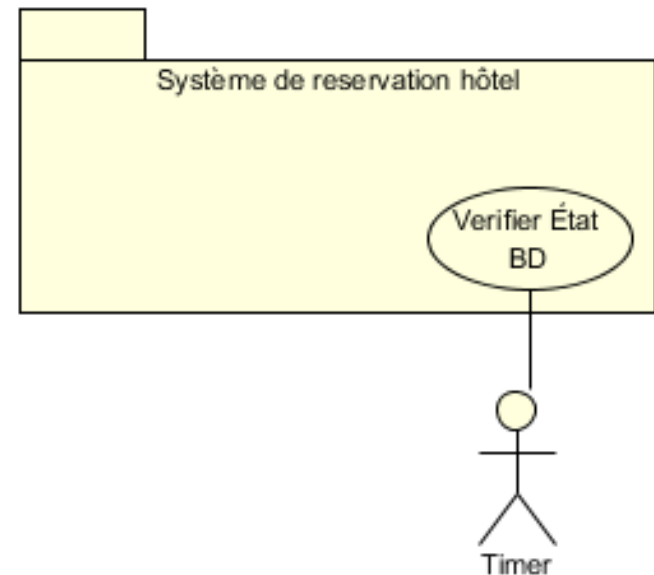
Evènements

- Un cas d'utilisation est déclenché par un évènement
- Il y a trois types d'évènement

1) **Externes:** c'est un acteur qui le déclenche



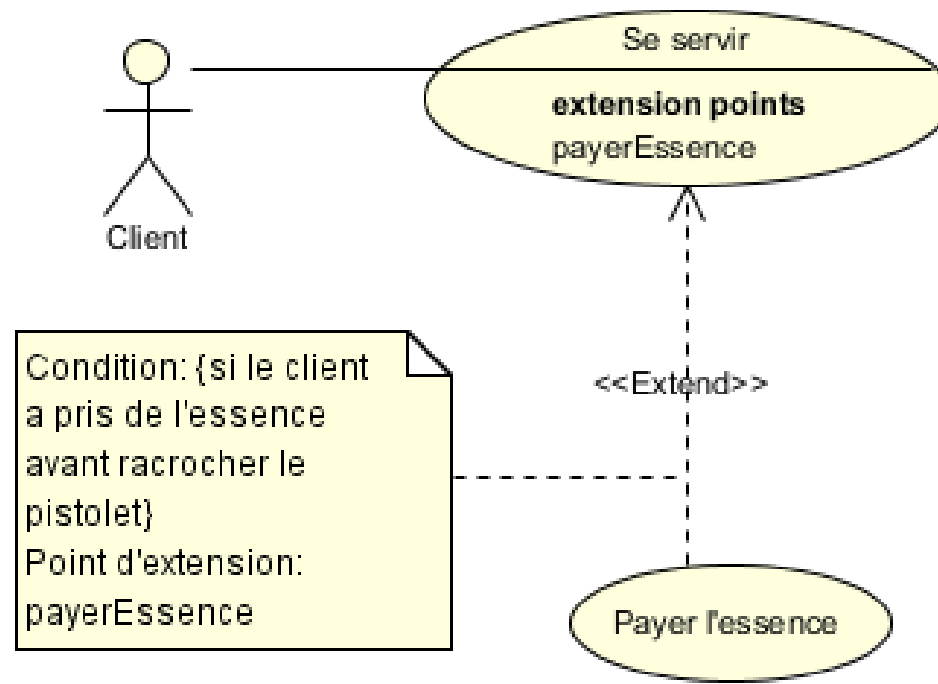
2) **Temporels:** on utilise l'acteur "Timer"



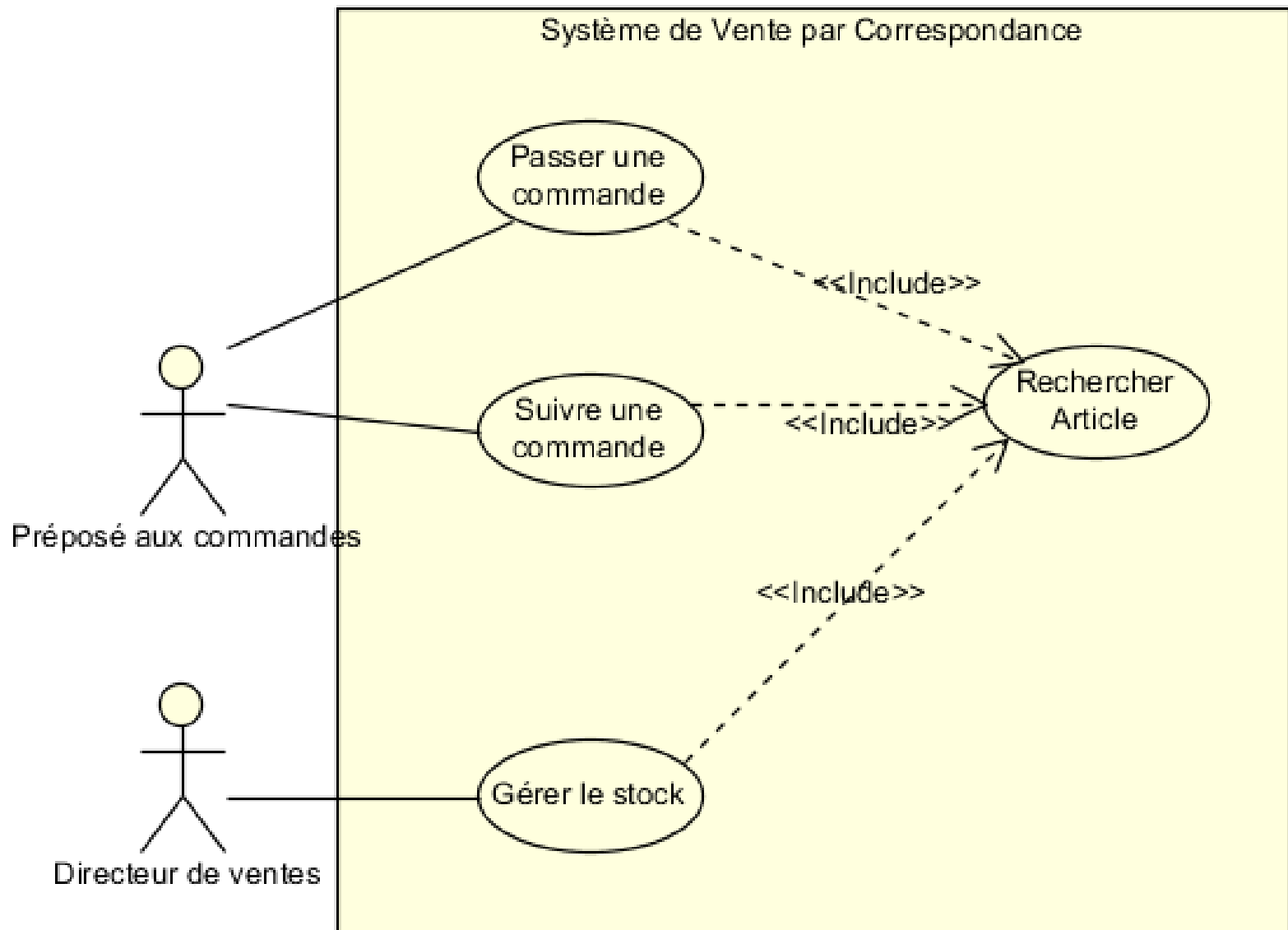
Evènements

3) **D'état**: se produisent à l'intérieur du système. On peut les modeler en utilisant des extensions

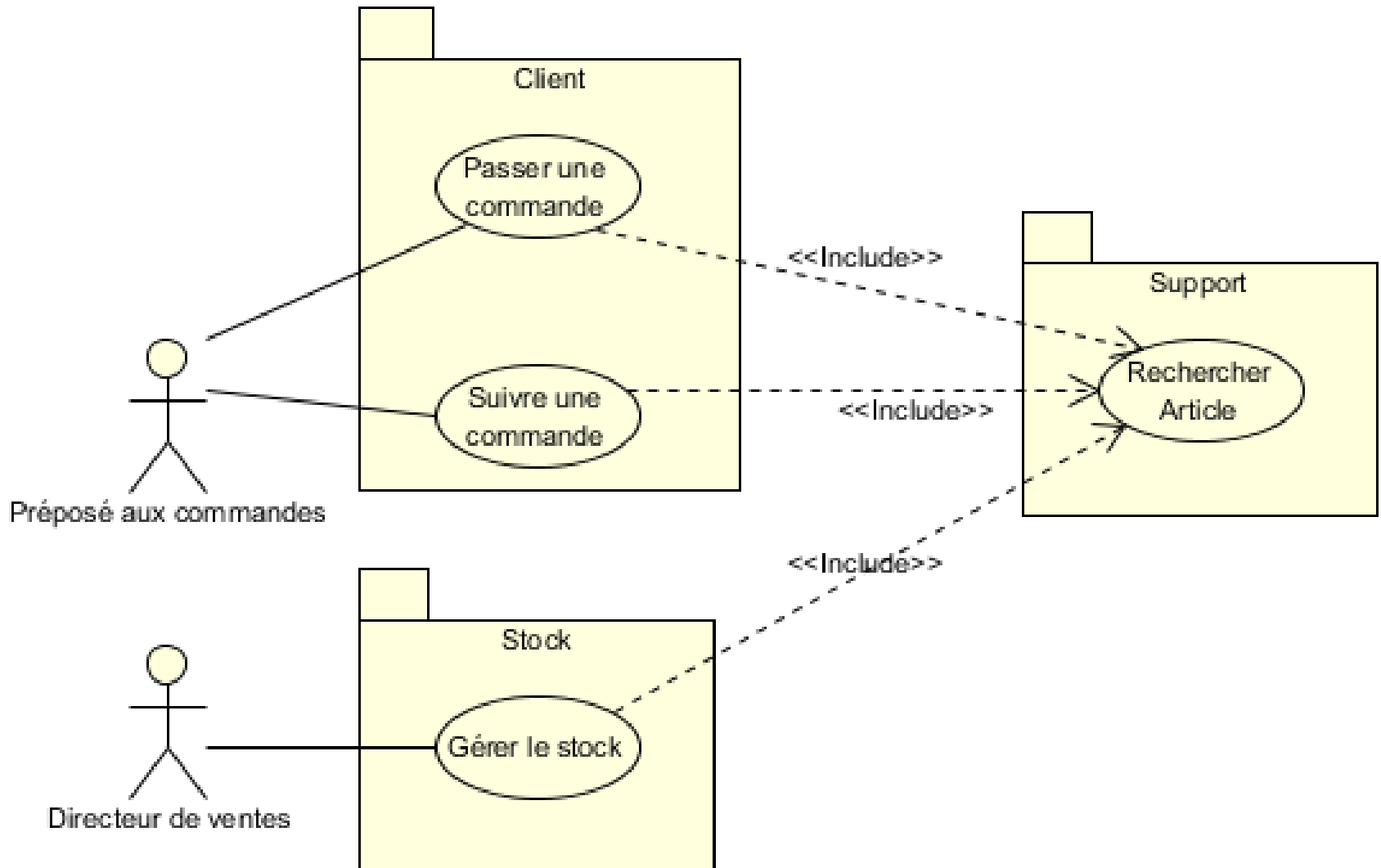
- on établit une **condition** de l'extension (ex: raccrocher le pistolet)



Exemple de système sans paquetage



Exemple de système avec paquetage



Spécification textuelle des cas d'utilisation

Chaque cas d'utilisation **doit être précisé** par une description **textuelle** qui peut être structurée en plusieurs sections :

- ▶ nom, description, acteurs, dates création/mise à jour, numéro de version
- ▶ conditions au démarrage (pré-conditions),
- ▶ conditions à la terminaison (post-conditions),
- ▶ étapes du déroulement normal ("nominal"),
- ▶ variantes possibles et les cas d'erreurs (alternatives et exceptions),
- ▶ informations échangées entre acteur et système,
- ▶ contraintes non fonctionnelles (performance, sécurité, disponibilité, confidentialité...).
- ▶ contraintes interface homme-machine

Nom: UC01- retirer argent (guichetier)

Description: détaille les étapes permettant à un **guichetier** d'effectuer l'opération de retrait d'euros demandé par un client

Acteur principal: guichetier

Acteur secondaire: système central

Date de création: 20/8/2020

Date de mise à jour: 28/8/2020

Num. version: 2

Précondition: Le client possède un compte

Postcondition Le guichetier ferme le compte. Le client récupère l'argent

Enchainement nominal

- (1) le guichetier saisit le numéro de compte du client.
- (2) l'application valide le compte auprès du système central (qui est aussi un acteur!!).
- (3) l'application demande le type d'opération au guichetier
- (4) le guichetier sélectionne un retrait en espèces d'une certaine quantité d'euros
- (5) l'application demande au système central de débiter le compte
- (6) l'application reçoit une réponse positive du système central
- (7) l'application notifie au guichetier qu'il peut délivrer le montant demandé

Enchaînement alternatif

A1: le client retire en dollars

L'enchaînement démarre après le point 3)

4.1) le guichetier sélectionne un retrait en espèces d'une certaine quantité **de dollars**

4.2) le système fait la conversion de dollars en euros

La séquence nominale reprend au point 5)

Enchaînement d'exception

E1: le compte est bloquée

L'enchaînement démarre après le point 2)

3.1) Le système central indique que le compte est bloqué

3.2) Le système central envoie un mail au personnel de la banque

Fin du cas d'utilisation

E2: le client décide d'annuler le retrait d'argent

L'enchaînement démarre après le point 1),2) ou 3)

4.1) Le guichetier annule l'opération (à la demande du client)

Fin du cas d'utilisation

Contraintes non fonctionnelles

Fiabilité: les accès doivent être extrêmement sûrs et sécurisés

Confidentialité: les informations du client ne seront pas divulguées

Contrainte liée à l'interface homme-machine

Avant de débiter le compte, l'application demande une confirmation au guichetier

Description textuelle des relations: <<include>>

- ▶ Une relation d'inclusion peut être décrite comme un appel à un autre cas d'utilisation
 - Ex: **Cas Retirer de l'argent** (contient un appel au cas **authentifier**)

Enchaînement nominal:

- 1) Le client introduit sa carte
- 2) **Appel au cas d'utilisation authentifier**
- 3) Le client est authentifié
- 4) Le client choisit le montant
- 5)



Description textuelle des relations: <<include>>

- ▶ Si le cas d'utilisation appelé échoue, on devra créer un cas d'exception dans le cas principal

Ex: **Cas Retirer de l'argent** (le cas principal)

Enchaînement nominal:

- 1) Le client introduit sa carte
- 2) Appel au cas d'utilisation **authentifier**
- 3) Le client est authentifié
- 4) Le client choisit le montant
- 5)

Enchaînement d'exception:

E1: l'authentification échoue

L'enchaînement démarre après le point 2

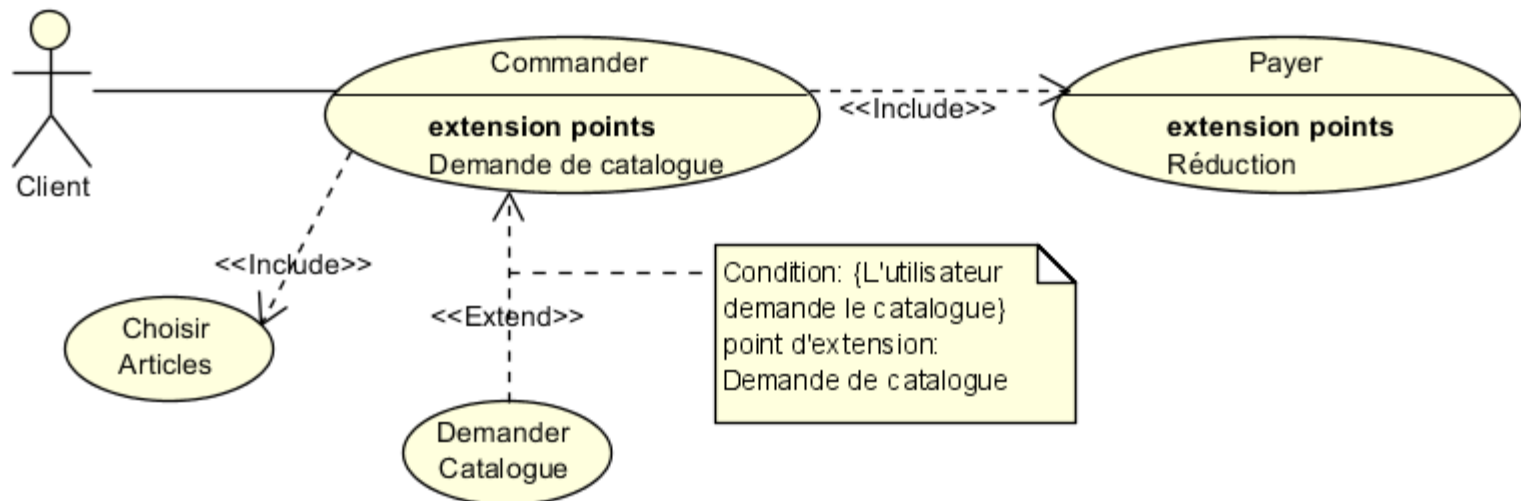
- 3) Le système indique au client que le code est incorrecte

Fin du cas d'utilisation

Description textuelle des relations: <<extends>>

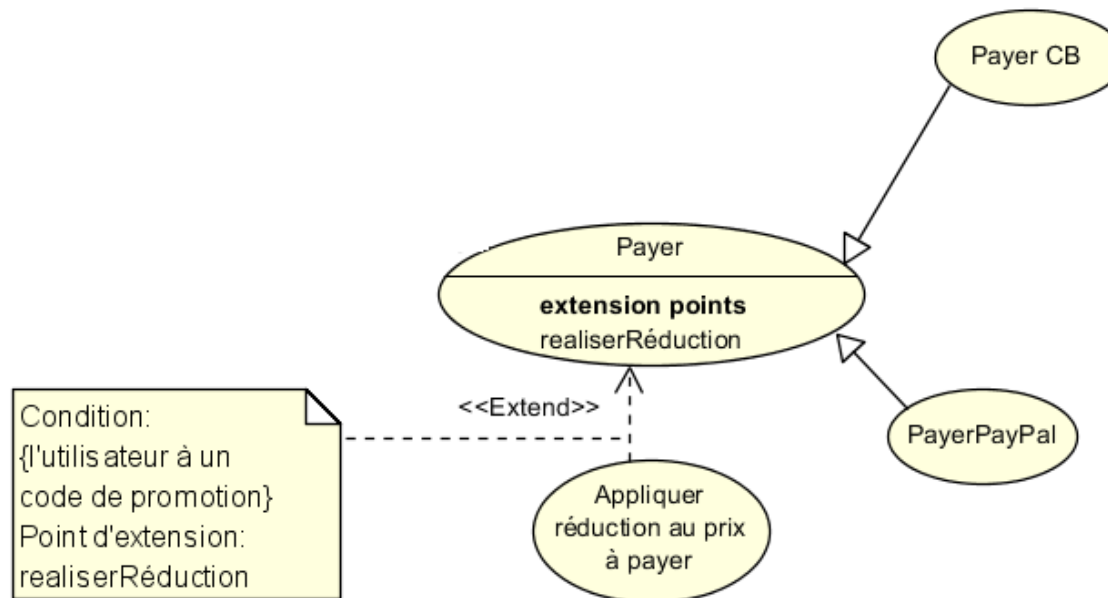
- ▶ Une relation d'extension est déclarée dans la description textuelle en modifiant l'enchainement nominal (en rajoutant le point d'extension)
 - Ex: Cas **Commander** contient un appel à l'extension **Demande du catalogue**
Enchainement nominal
 - 1) Appel au cas d'utilisation **Choisir articles**
 - 2) Les articles sont choisis**Point d'extension: Demande du catalogue**
 - 3) Appel au cas d'utilisation Payer

(après 2 on appelle au cas d'utilisation lié à la demande du catalogue: **Demander Catalogue**, si la condition est satisfaite)



Description textuelle des relations: héritage

- Nous établissons plusieurs appels possibles aux différents cas d'utilisation
 - Ex: Payer avec Paypal ou avec une carte de Crédit
1. Le Client choisit le type de paiement:
 - a) En cas de paiement par carte, appel au cas d'utilisation "Payer par carte".
 - b) En cas de paiement par Paypal, appel au cas d'utilisation "Payer par Paypal"



Les cas d'utilisations peuvent être vus comme des **classes de scénarios**. Chaque **scénario** correspond à une utilisation particulière, par un acteur donné, dans des circonstances données. On peut décrire les principaux (préparation des tests finaux de l'application).

SCENARIO 1

- 1) Le guichetier Smith saisit le numéro de compte 445-455355-355.
- 2) Le système central valide le compte.
- 3) L'application demande le type d'opération à Smith.
- 4) Smith tape 'retrait' et 600 euros.
- 5) L'application demande au système central de débiter cette quantité.
- 6) Le système centrale vérifie le solde et indique à l'application qu'il accepte l'opération.
- 7) L'application indique à Smith qu'il peut délivrer 600 euros au client.

Mode d'emploi des diagrammes de cas d'utilisation

- ▶ **Décrire** le *système* (très haut niveau).

Exemple : site de commerce électronique (très simplifié)

- Le client parcourt le catalogue.
- Il ajoute les objets qui l'intéressent dans son panier.
- Pour payer, il donne ses informations de carte bancaire et son adresse
- Il confirme l'achat.
- Le système contrôle la validité du paiement et confirme la commande.

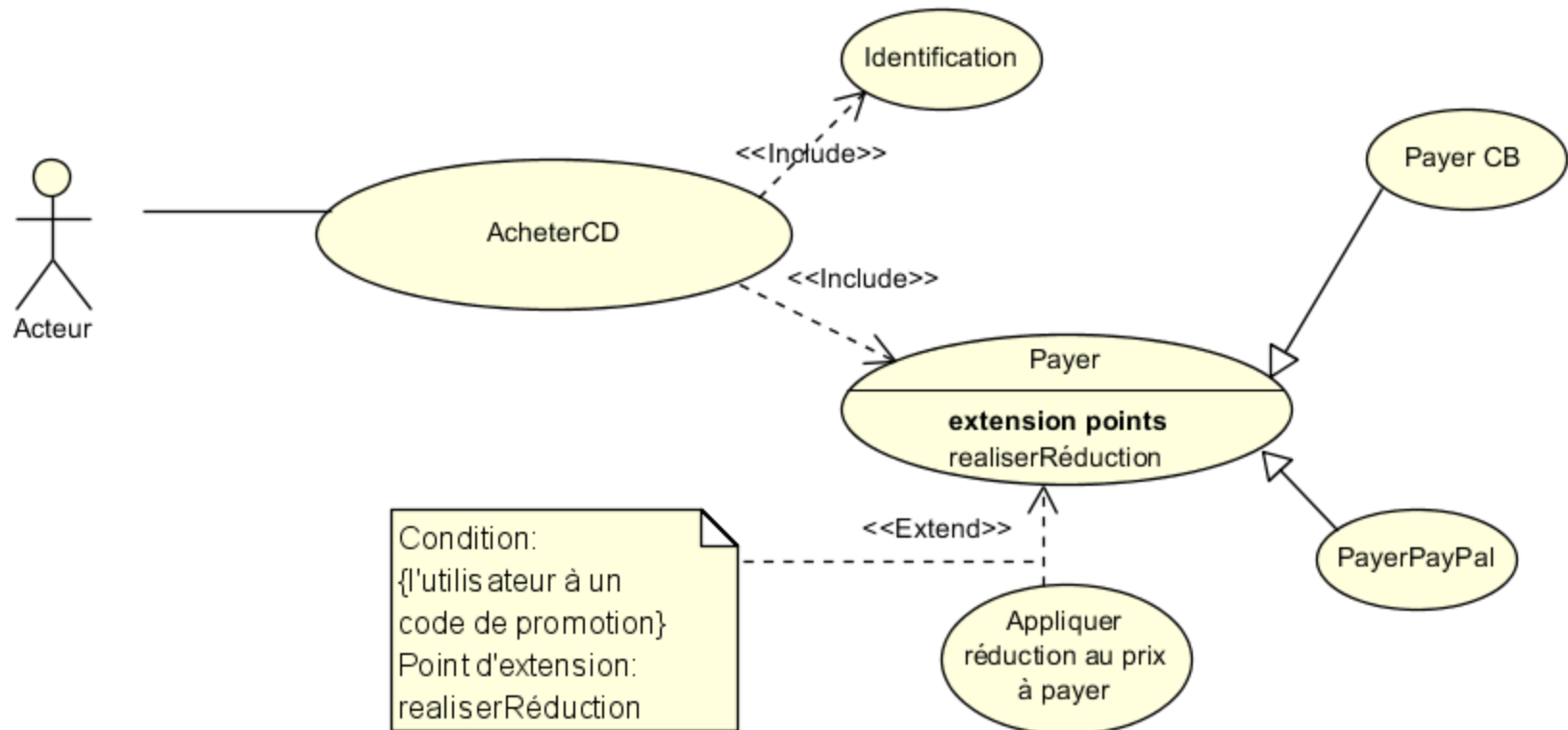
- ▶ En **déduire** une liste d'**acteurs** et de **cas**.
- ▶ Construire un **premier diagramme**. Le structurer pour le rendre plus clair (relations <<include>>, <<extend>>, généralisation/spécialisation)
- ▶ **Détailler chaque cas** (description textuelle structurée)
- ▶ Eventuellement **définir les scénarios** les plus importants qui pourront servir au moment des tests finaux.

Conclusion

- ▶ Permet de décrire les principales fonctionnalités attendues. Permet de les organiser grâce aux relations d'héritage, d'inclusion et d'extension.
- ▶ Avec les descriptions textuelles et les scénarios, l'analyste dispose de moyens simples pour exprimer de manière semi-formelle les **besoins fonctionnels et non fonctionnels du système étudié** (son « cahier des charges »).

Exemple de diagramme: Achat d'un cd sur un magasin online

(extrait, recherche pas incluse)



GAB

