

Modélisation Objet Avancé

3ème année Cyberlog/Cyberdata 2023-2024

Mazen EL-SAYED

mazen.el-sayed@univ-ubs.fr

Organisation du cours

2

- 27h de cours (CM/TD/TP)
- Contrôle continu
- Projet possible

Références

3

- UML 2.0
 - Benoît Charoux, Aomar Osmani, Yann Thierry-Mieg
 - Editions Pearson, Education France (2008)
- Modélisation et Conception Orientée Objet avec UML 2
 - Michael Blaha, James Rumbaugh
 - Editions Pearson, Education France (2005)
- Cours de Flavio Oquendo
- Cours de Pierre Gérard
 - <https://lipn.univ-paris13.fr/~gerard/>
- Cours UML-[ESI4-ITIS-EVRY](https://sites.google.com/site/umlesi4itisevry/home/Cours/)
 - <https://sites.google.com/site/umlesi4itisevry/home/Cours/>

Plan

4

0. Diagramme de classes
1. Diagramme de cas d'utilisation
2. Diagramme de séquences
3. Diagramme d'états-transitions
4. Diagramme d'activités
5. Compléments (si le temps le permet)

Chapitre 1

Diagramme de cas d'utilisation



Modélisation des besoins

6

➤ Expression des besoins

- Discuter avec les clients afin de comprendre de quoi ils ont besoin.
- Déterminer les fonctionnalités.
- Sans parler de comment les réaliser.

➤ Modéliser les besoins permet de :

- Faire l'inventaire des fonctionnalités attendues du système.
- Scinder les fonctionnalités du système en unités cohérentes (les cas d'utilisation) ayant un sens pour les acteurs
- Identifier les différents utilisateurs (acteurs) du système
 - Environnement du système

➤ Choisir le bon diagramme UML qui peut être utilisé

Les éléments du diagramme de cas d'utilisation

7

- En UML, on modélise les besoins au moyen de **diagrammes de cas d'utilisation**.
 - Ils décrivent les fonctionnalités du logiciel du point de vue de l'utilisateur
 - On ne cherche pas à savoir COMMENT les besoins vont être réalisés.
 - Pas d'aspects techniques : Ils ne présentent pas des solutions d'implémentation.
- Le diagramme de cas d'utilisation décrit l'interaction entre les acteurs et le système.
- Pour élaborer un diagramme de cas d'utilisation, on commence par identifier :
 - Le **système** (délimiter le système)
 - Les **acteurs** (externes) qui interagissent avec le système
 - Les actions des acteurs sur le système
 - Ensemble de cas d'utilisation

Exemple de diagramme de cas d'utilisation

8

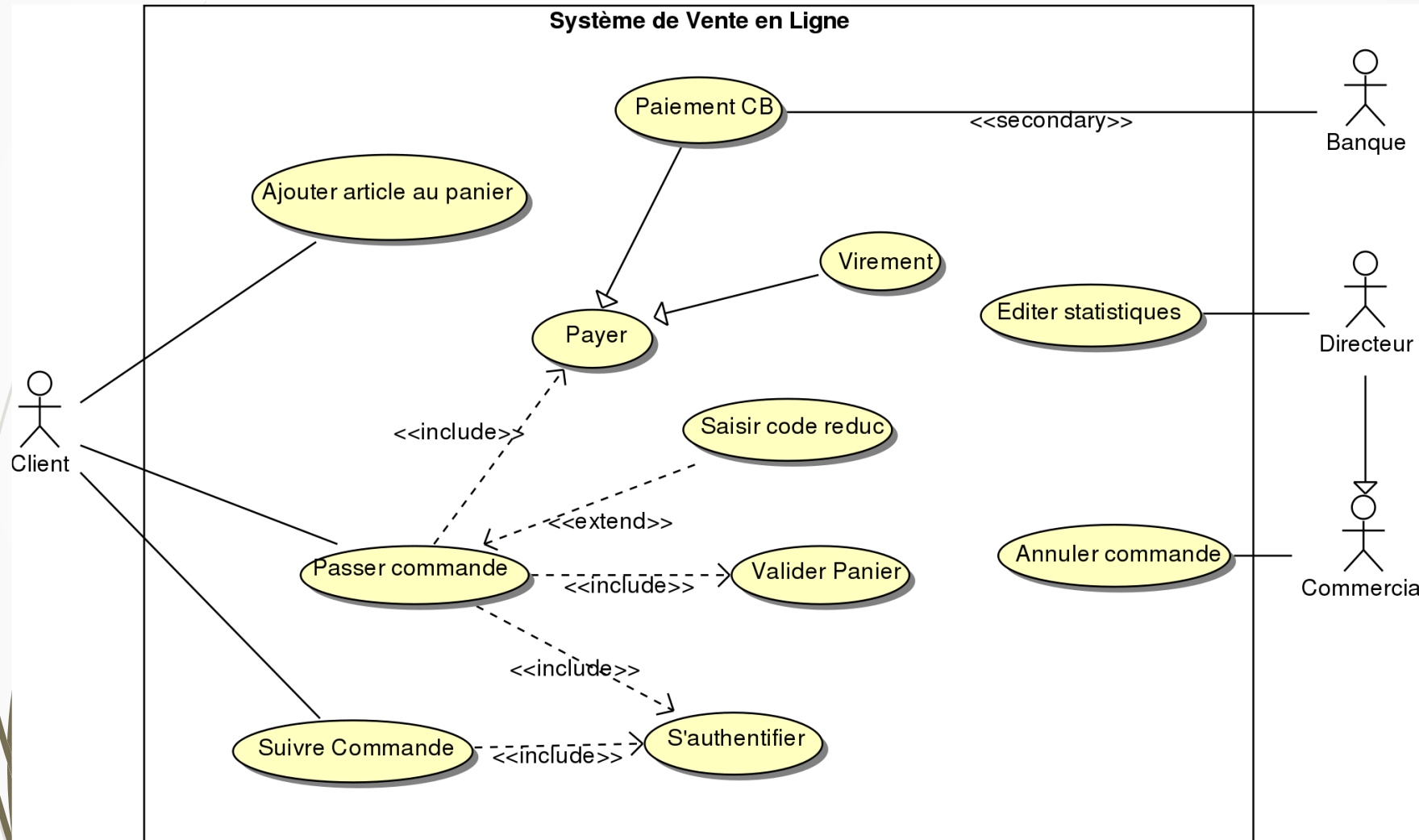


Diagramme de cas d'utilisation

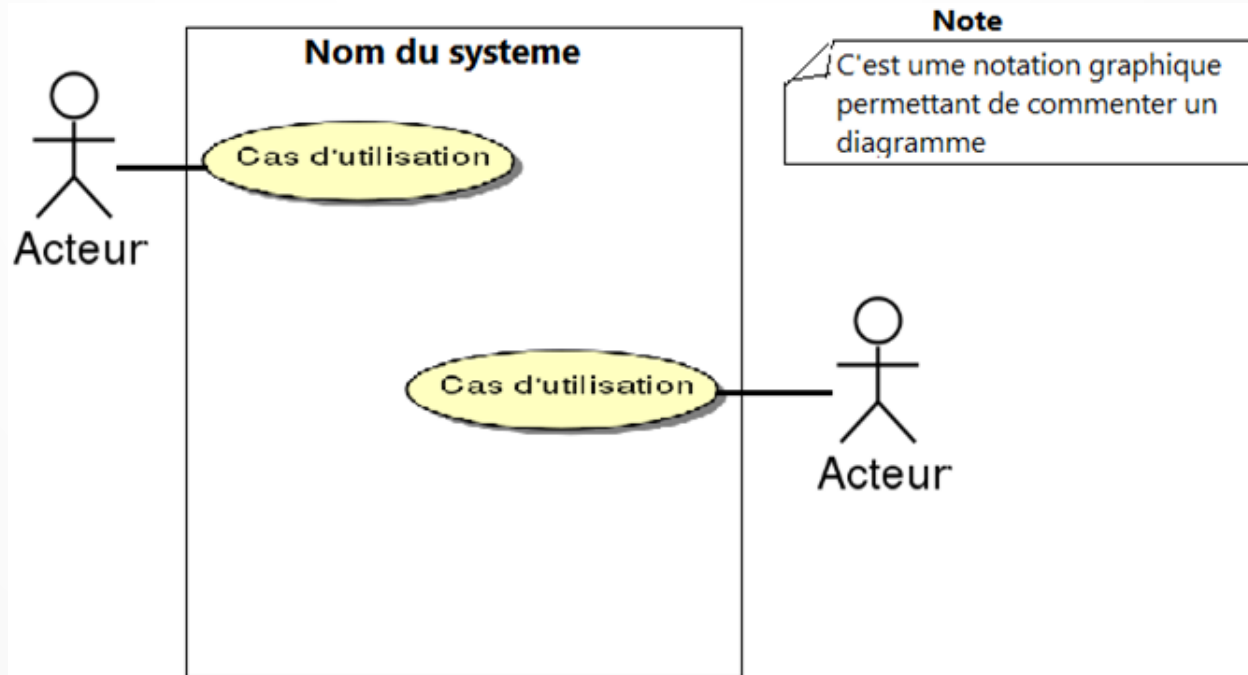
9

- Les principaux éléments du diagramme de cas d'utilisation
 - Le système
 - Les cas d'utilisation
 - Les acteurs
 - Des relations : acteur - acteur, acteur - cas d'utilisation et finalement cas d'utilisation – cas d'utilisation

Un système

10

- Un système représente une application dans le modèle UML.
- Il est identifié par un nom et contient les cas d'utilisation représentant les fonctionnalités offertes par l'application à son environnement
- L'environnement représente les acteurs qui communiquent avec les cas d'utilisation
- Il est représenté par un rectangle contenant le nom



Cas d'utilisation

11

- Un cas d'utilisation représente une unité cohérente représentant une fonctionnalité visible de l'extérieur.
- Un cas d'utilisation est un service rendu à l'utilisateur, il implique des séries d'actions plus élémentaires.
 - Il est conseillé de ne pas réduire le cas d'utilisation à une simple action élémentaire.



- Il exprime les interactions entre les acteurs et le système
- Un cas d'utilisation est représenté par une ovale contenant le nom du cas qui est souvent un **verbe à l'infinitif et un complément**.

Les acteurs

12

- Un acteur est une entité extérieure au système et qui interagit directement avec lui.
- Qui peut être un acteur :
 - Qui est intéressé par un certain besoin?
 - Par qui le système est utilisé dans l'organisation?
 - Qui bénéficiera de l'utilisation du système?
 - Qui fournira les informations au système et qui l'utilisera?
 - Qui va supporter le système et le maintenir?

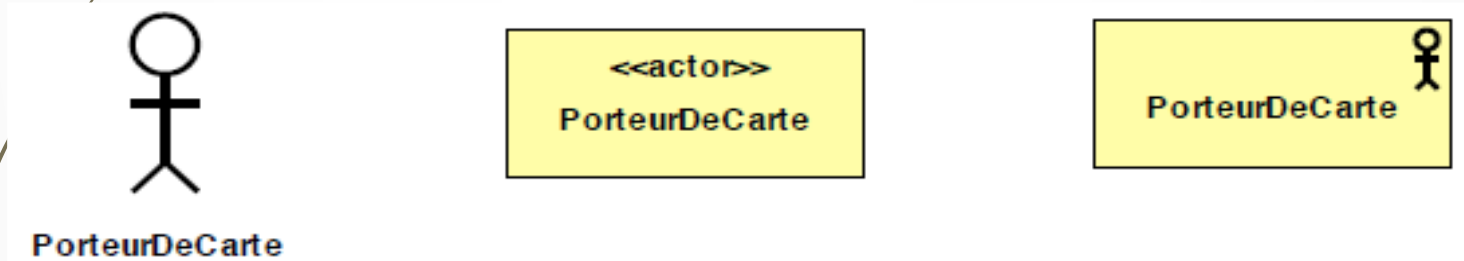
Les acteurs

13

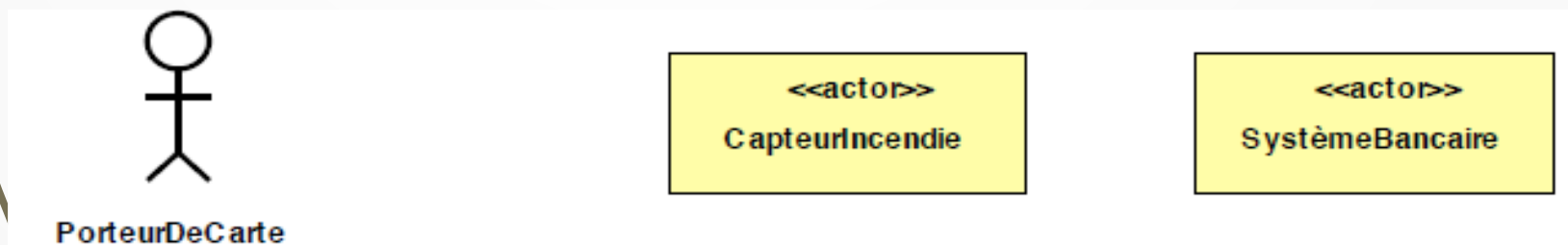
- Un acteur est représenté par un bonhomme avec son nom (qui indique son rôle) inscrit en dessous



- Notations alternatives pour les acteurs



- Utiliser plutôt le stéréotype **<<actor>>** pour les acteurs non humains



Identification des acteurs

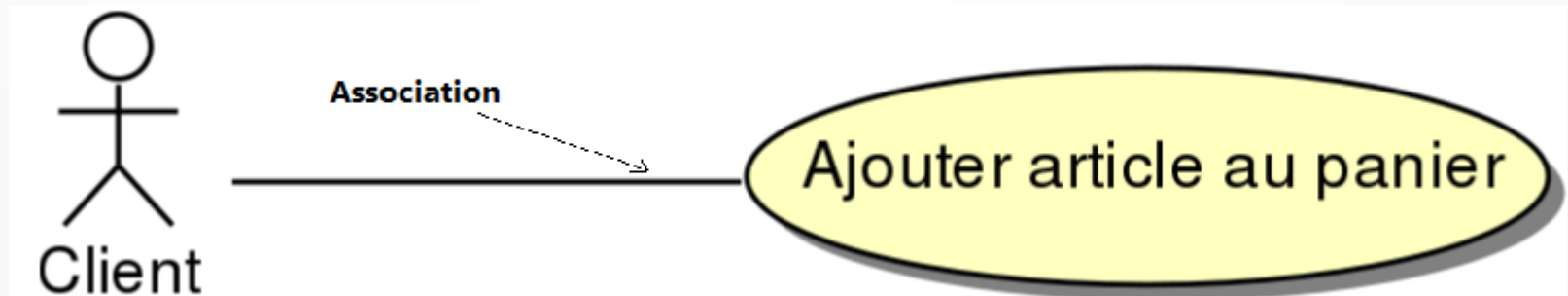
14

- Les principaux acteurs sont les utilisateurs du système.
- Un acteur correspond à **un rôle**, pas à une personne physique.
- Une même personne physique peut être représentée par plusieurs acteurs si elle a plusieurs rôles.
- Si plusieurs personnes jouent le même rôle vis-à-vis du système, elles seront représentées par un seul acteur.
- En plus des utilisateurs, les acteurs peuvent être :
 - Des périphériques manipulés par le système (imprimantes, capteurs, ...) ;
 - Des systèmes informatiques externes au système mais qui interagissent avec lui, etc.

Relations entre cas d'utilisation et acteurs

15

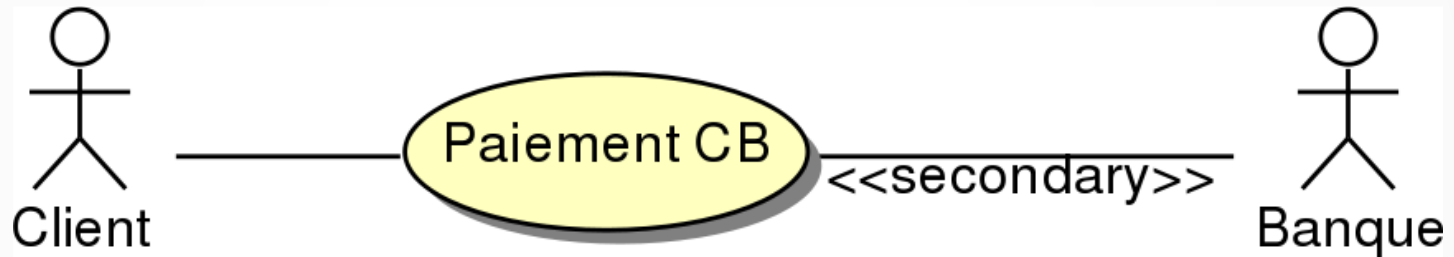
- Les acteurs impliqués dans un cas d'utilisation lui sont liés par une association.
- Cette association peut être traduite par <<communiqué>>
- Une association est représentée par une ligne solide entre l'acteur et le cas d'utilisation.
- L'association est la seule relation possible entre un acteur et les cas d'utilisation.



Acteurs principaux et secondaires

16

- L'acteur est dit **principal** pour un cas d'utilisation lorsque **l'acteur est à l'initiative des échanges** nécessaires pour réaliser le cas d'utilisation. Il est le principal bénéficiaire du résultat de l'exécution du cas d'utilisation.



- Les acteurs **secondaires** sont **sollicités par le système** alors que le plus souvent, les acteurs principaux ont l'initiative des interactions.
 - Le plus souvent, les acteurs secondaires sont d'autres **systèmes informatiques** avec lesquels le système développé est inter-connecté.
- Il est conseillé de mettre les acteurs principaux à gauche du diagramme et les acteurs secondaires à droite.

Description des acteurs

17

- Pour chaque acteur :
 - choisir un identificateur représentatif de son rôle
 - choisir une forme nominale décrivant un rôle
 - identification concise mais précise
 - terme provenant autant que possible du métier
- Importance :
 - référencé tout au long des documents
 - pourra apparaître dans les manuels utilisateurs, dans l'interface homme-système, dans le code ...

Description des acteurs

18

- Pour chaque acteur :
 - choisir un identificateur représentatif de son rôle
 - 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.

Relations cas d'utilisation - acteurs : multiplicité

19

- Un acteur peut utiliser plusieurs fois le même cas d'utilisation.
- Lorsqu'un acteur peut agir plusieurs fois avec un cas d'utilisation, on peut ajouter une multiplicité sur l'association (*, n, n..m). Cette notion n'est pas propre à ce diagramme
- La multiplicité figure dans chaque extrémité de l'association
 - La cardinalité située à l'extrémité du cas d'utilisation indique avec combien d'instances du cas d'utilisation, chaque instance de l'acteur situé à l'autre extrémité est liée.
 - La cardinalité située à l'extrémité de l'acteur indique avec combien d'instance de l'acteur, chaque instance du cas d'utilisation situé à l'autre extrémité est liée.



mult

mult

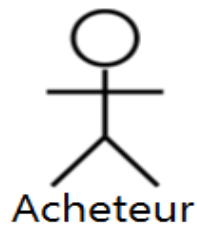
Ajouter article au panier

Relations cas d'utilisation – acteurs : multiplicité

20

- Les valeurs de **mult** sont indiquées dans le tableau suivant :

Valeur :	signification :
1	Un et un seul
0..1	Zéro ou un
M .. N	De M à N (entiers naturels)
* (ou 0 .. *)	De zéro à plusieurs
N	Exactement N
1 .. *	D'un à plusieurs



1..*

1..*

Acheter cheval



1

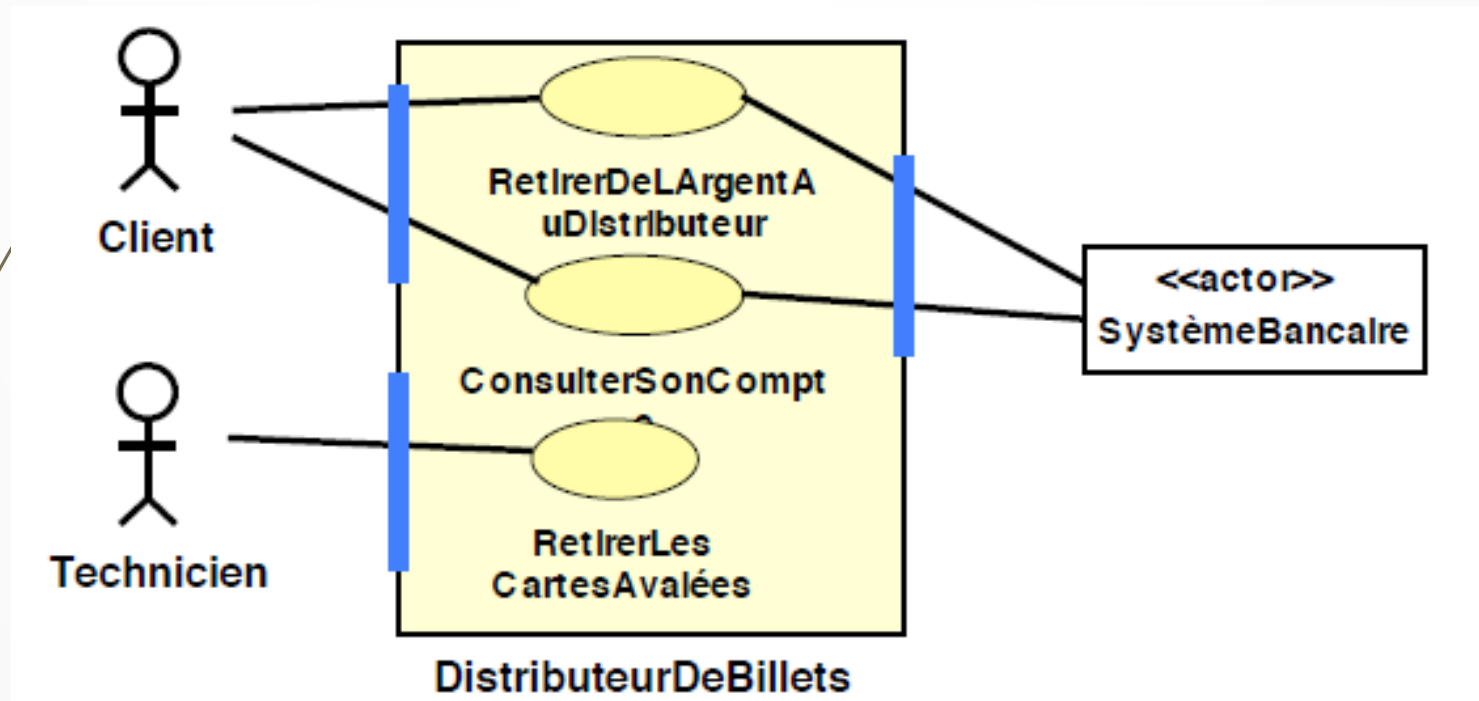
*

Ajouter article au panier

Limites du système et interfaces

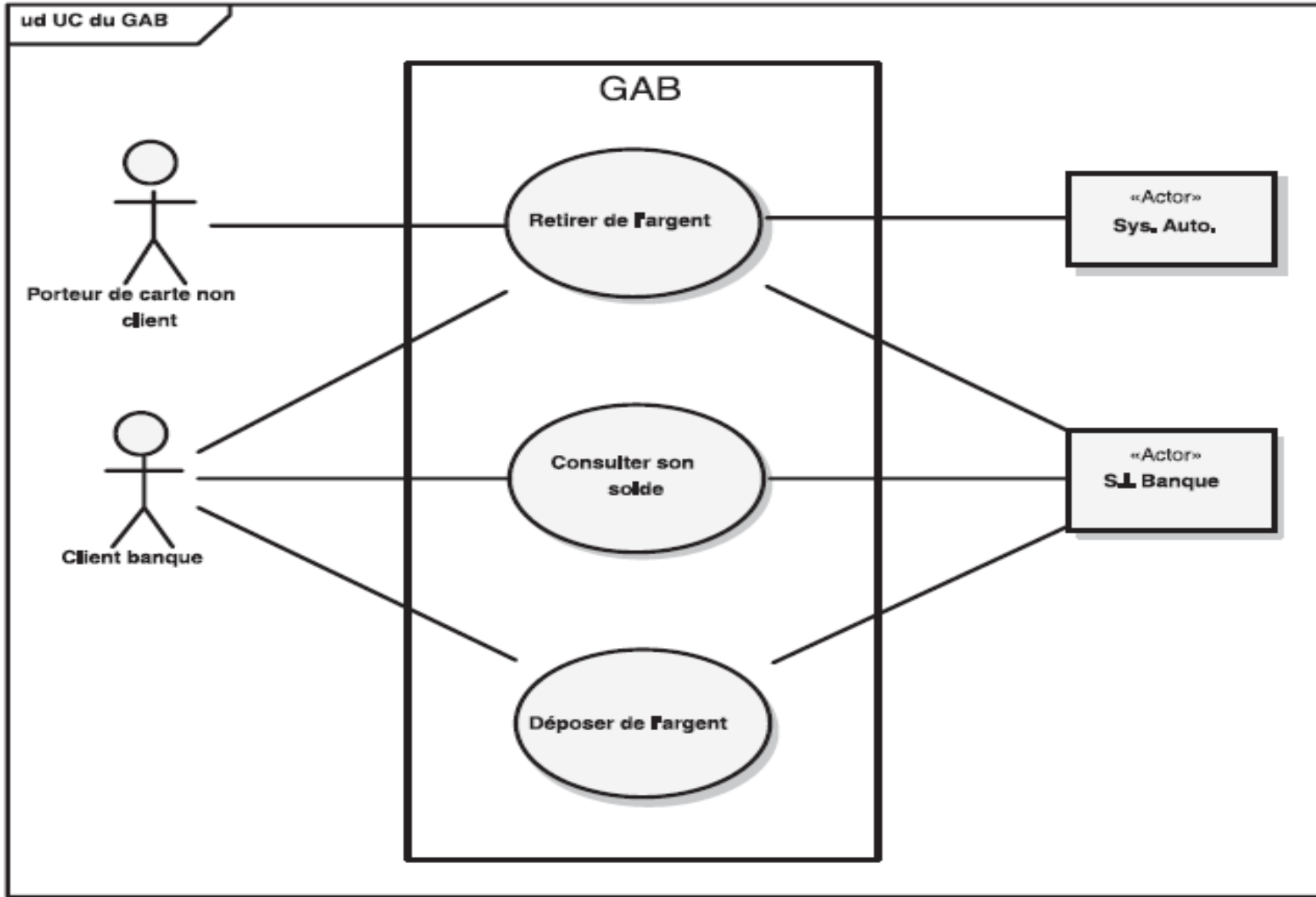
21

- ▀ Limites du système ⇒ interface de communication
 - ▀ acteur humain ⇒ interface homme – système (IHM)
 - ▀ acteur logiciel ⇒ interface logicielle (e.g. API)



Exemple : diagramme préliminaire

22



Raffinement du diagramme

23

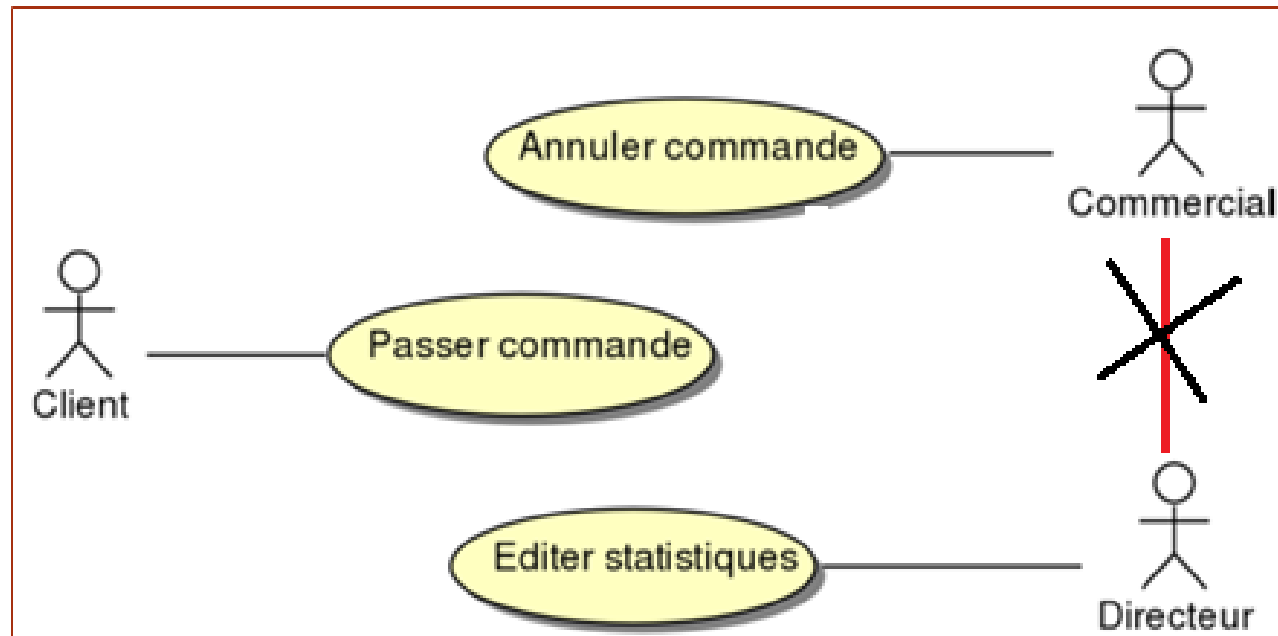
- Le modèle peut être raffiné en introduisant des relations
 - Entre acteurs :
 - Généralisation (héritage)
 - Entre cas d'utilisation
 - Inclusion
 - Extension
 - Généralisation (héritage)
- **Attention** : Ce sont les seules relations utilisées à l'intérieur du système.
 - Pas d'associations entre cas d'utilisation



Relations entre acteurs

24

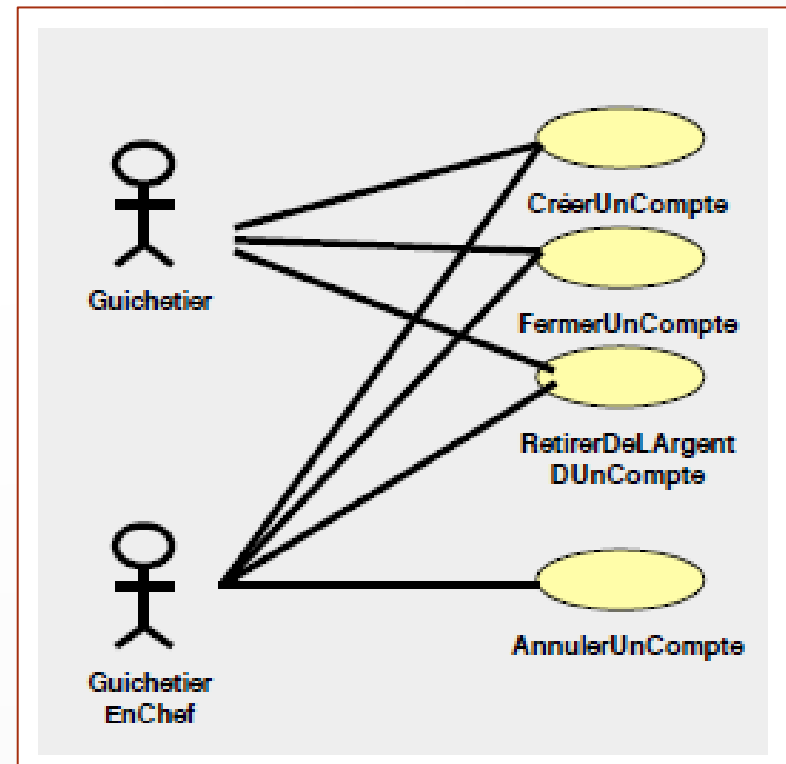
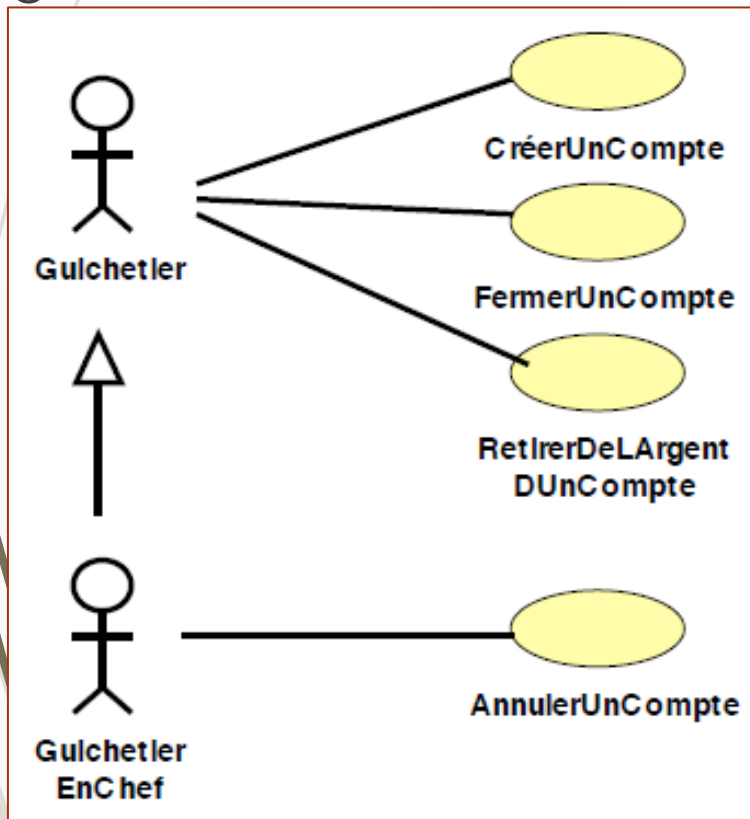
- Communications entre entités externes non modélisées en UML
- UML se concentre sur la description du système logiciel et de ses interactions avec l'extérieur



Relations entre acteurs : généralisation

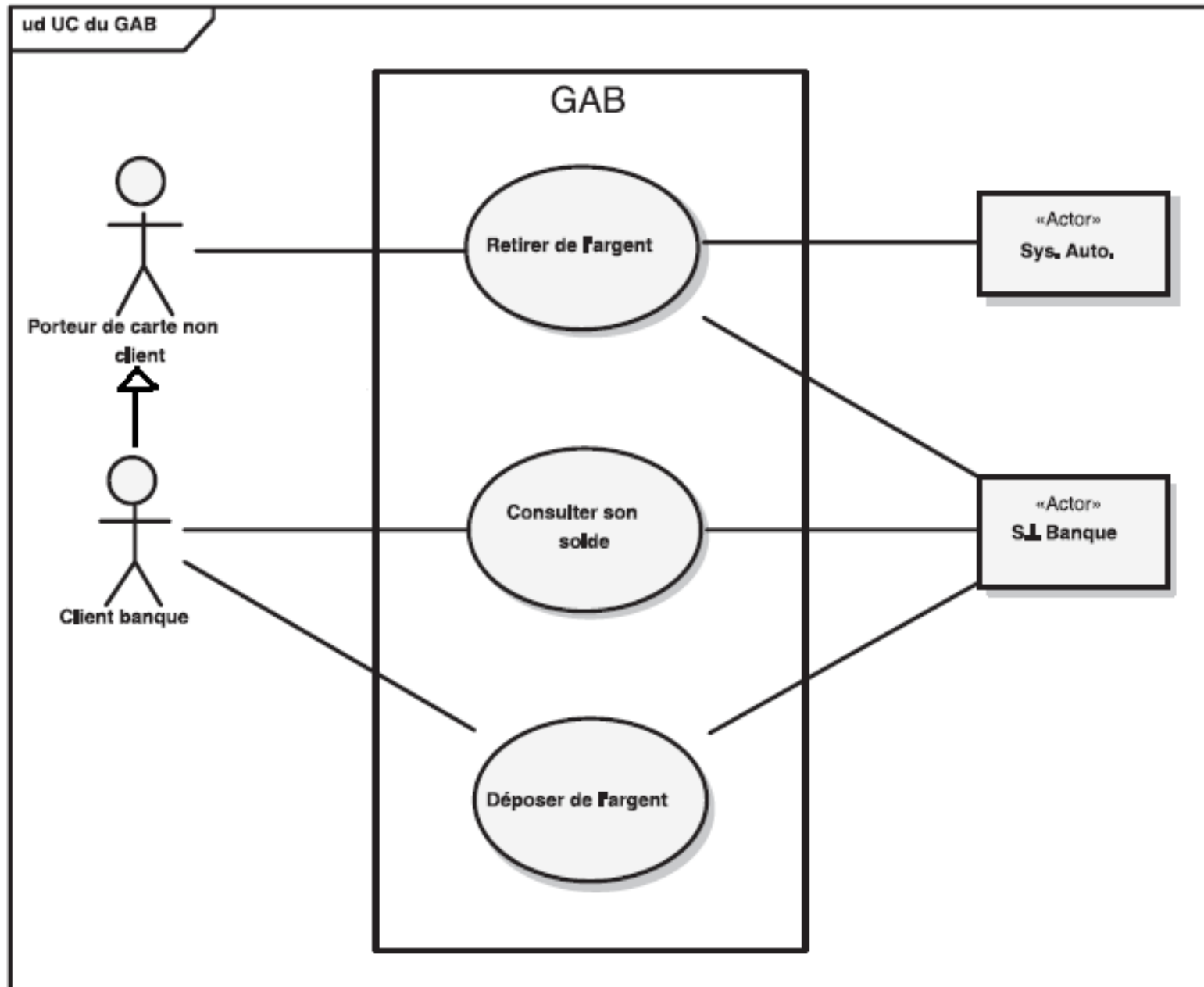
25

- La seule relation entre acteurs est la relation de généralisation (ou héritage)
- La flèche pointe vers l'acteur général.
- L'acteur spécifique (dérivé) hérite des associations de l'acteur général



Relations entre acteurs : généralisation

26



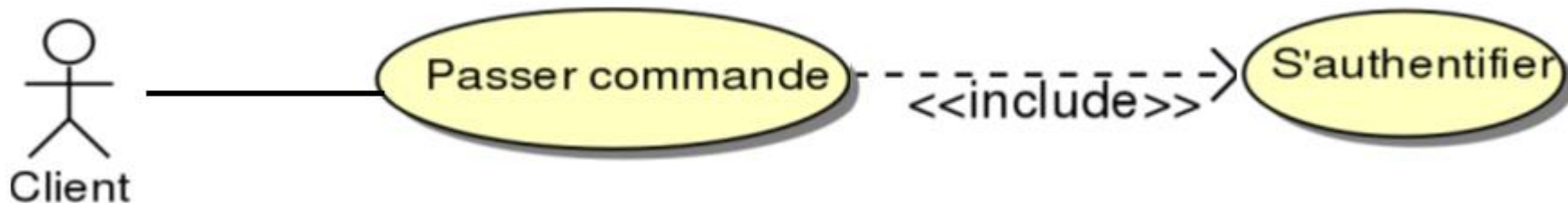
Relations entre cas : Inclusion <<include>>

27

- **Inclusion** : le cas A inclut le cas B (B est une partie obligatoire de A).



- **<<include>>** permet d'incorporer le comportement d'un autre cas d'utilisation.
- Le cas de base incorpore explicitement un autre cas, à un endroit spécifié dans sa description.



Décomposition grâce à include

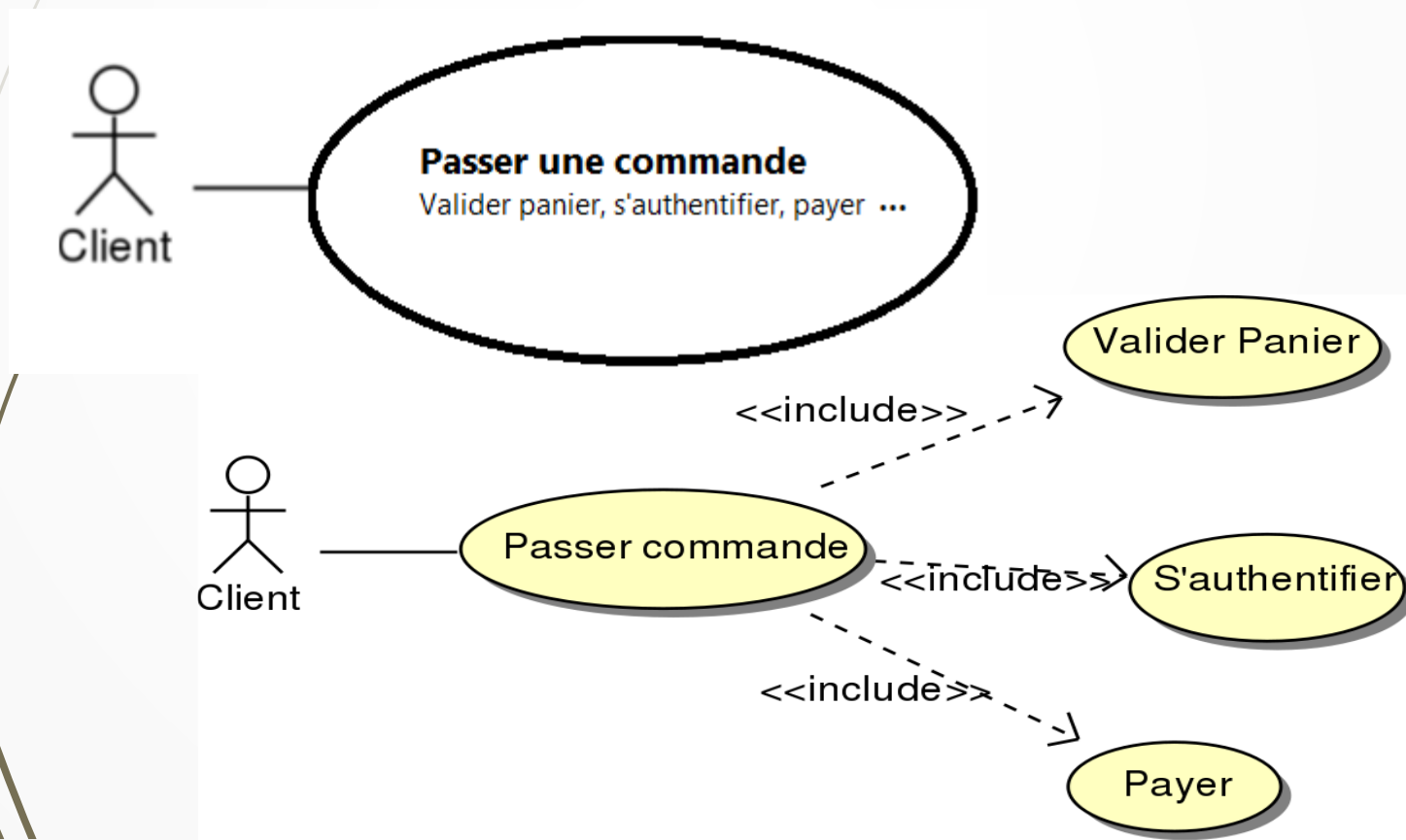
28

- Quand un cas est trop complexe (faisant intervenir un trop grand nombre d'actions), on peut procéder à sa décomposition en cas plus simples.
 - La décomposition n'est pas unique
 - Le diagramme devient plus précis et plus détaillé
 - Peut réduire le nombre de modules lors de l'implantation

Décomposition des cas : exemple

29

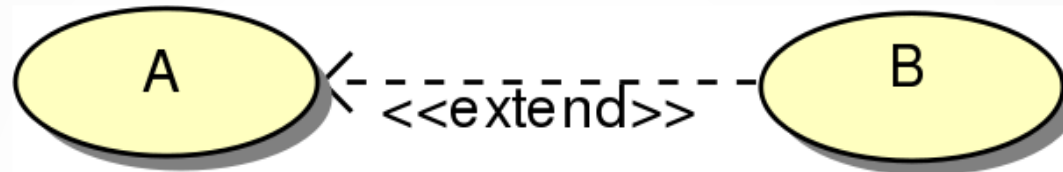
- Par exemple, on peut décomposer le cas d'utilisation "**Passer commande**" qui comporte des actions élémentaires "**Valider au panier**", "**s'authentifier**" et "**Payer**" en 4 cas d'utilisation et en utilisant la relation `<<include>>`.



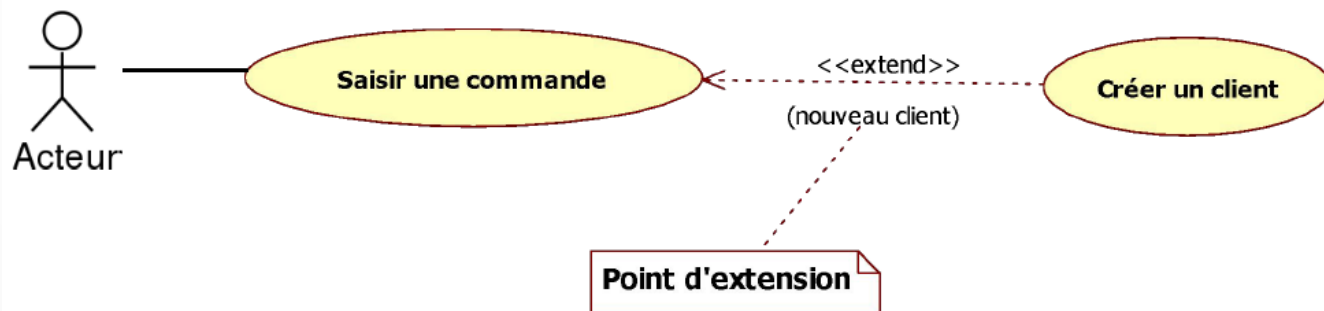
Relations entre cas : Extension <<extend>>

30

- **Extension** : le cas B étend le cas A (B est une partie optionnelle de A).



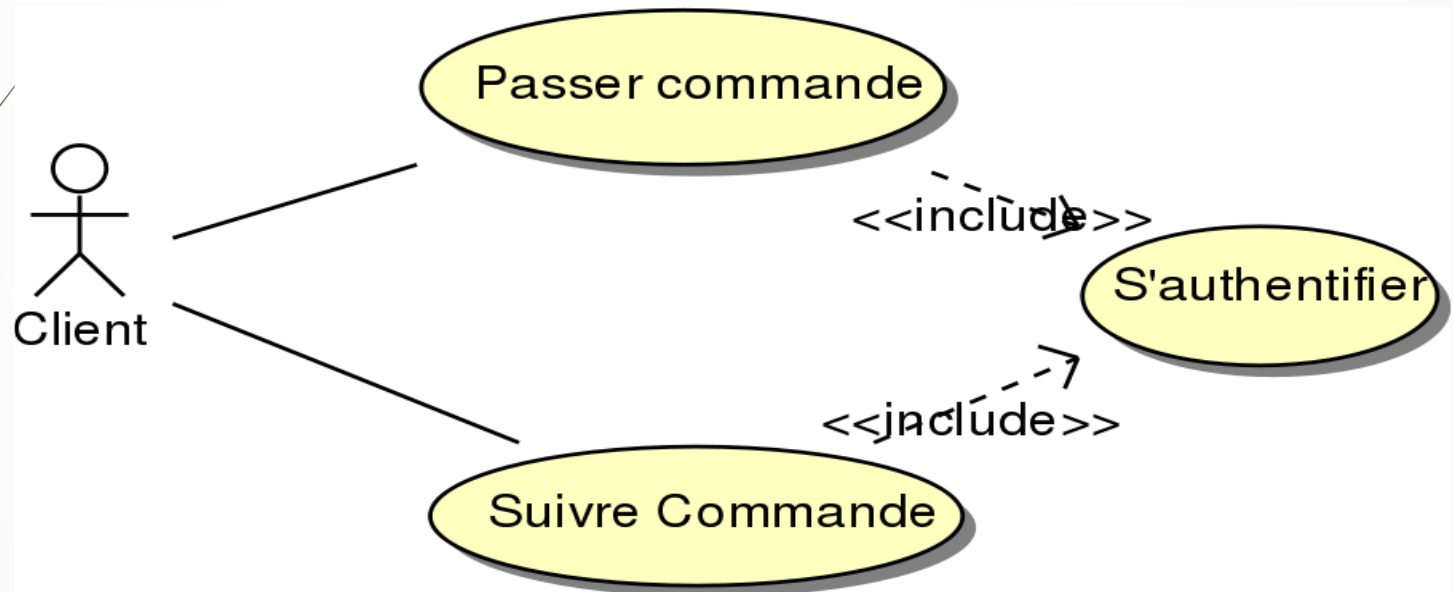
- <<extend>> permet de modéliser la partie d'un cas d'utilisation considérée comme facultative (B).
- Le cas de base peut fonctionner seul (A), mais il peut aussi être complété par un autre, sous certaines conditions, et uniquement à certains points particuliers de son flot d'événements appelés **points d'extension**.



Réutilisabilité avec include et extend

31

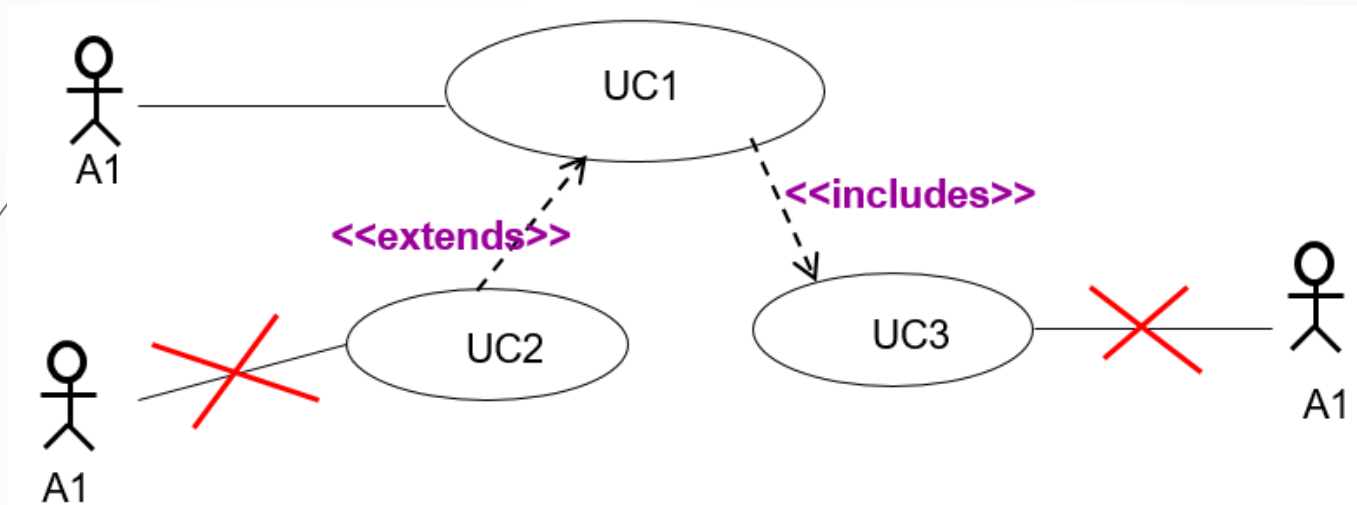
- Les relations entre cas permettent la **réutilisabilité** du cas s'authentifier : il sera inutile de développer plusieurs fois un module d'authentification.



Remarques

32

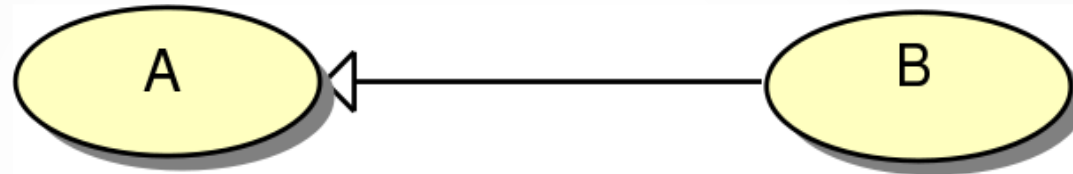
- Si un acteur est lié à un cas d'utilisation, il est inutile de le lier aux cas d'utilisation qui étendent ou sont inclus dans le cas d'utilisation courant



Relations entre cas : Généralisation

33

- **Généralisation** : le cas A est une généralisation du cas B (B est une sorte de A, un cas spécifique de A).

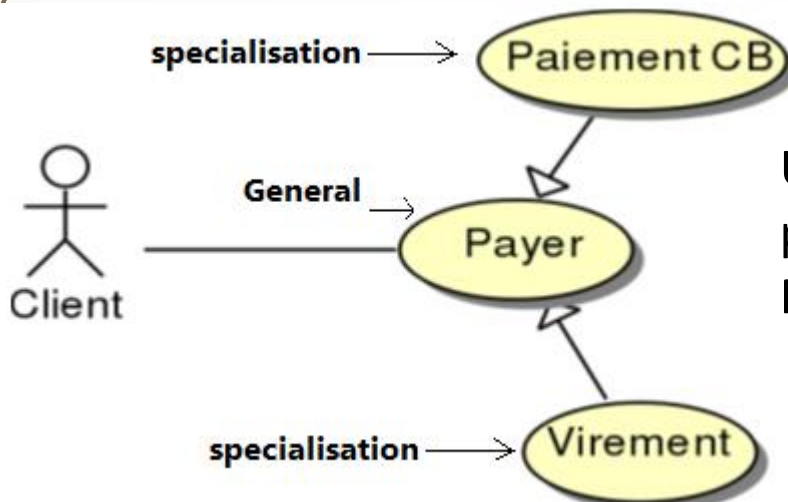


- Cette relation de généralisation/spécialisation est présente dans la plupart des diagrammes UML et se traduit par le concept **d'héritage** dans les langages orientés objet.
- Un cas peut hériter un autre cas.
- La flèche pointe vers l'élément général.

Généralisation

34

- Factoriser les cas, mettre les actions communes des plusieurs cas d'utilisation dans un cas d'utilisation virtuelle
- On applique cette règle que lorsque les relations ont le sens suivant « A est une sorte de B »
- Par exemple, on peut factoriser les deux cas d'utilisation « Paiement par carte bancaire » et « Virement » et en extraire les actions communes pour construire un nouveau cas d'utilisation « Payer »
 - Le nouveau cas est virtuel puisqu'il n'y a pas de paiement que par « carte bancaire » ou « par virement ».

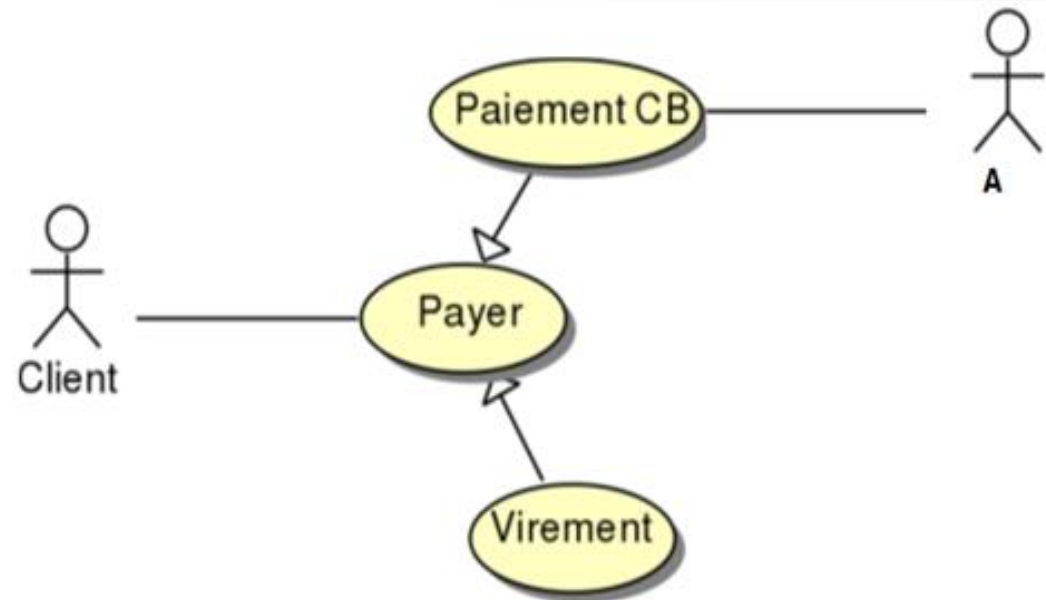


Un virement est un cas particulier de paiement (ou une sorte de Paiement)

Généralisation

35

- La généralisation des cas permet la **réutilisabilité** du cas général.
- La généralisation réduit le nombre de module de l'implantation
- Les cas d'utilisation descendants (Paiement carte, Virement) héritent des interactions de leur parent. Ils peuvent comprendre des interactions supplémentaires.



Recenser les cas d'utilisation

36

- Il n'y a pas une manière mécanique et totalement objective de repérer les cas d'utilisation.
 - Il faut se placer du point de vue de chaque acteur et déterminer comment il se sert du système, dans quels cas il l'utilise, et à quelles fonctionnalités il doit avoir accès.
- Il faut éviter les redondances et limiter le nombre de cas en se situant au bon niveau d'abstraction (par exemple, ne pas réduire un cas à une seule action).
- Il ne faut pas faire apparaître les détails des cas d'utilisation, mais il faut rester au niveau des grandes fonctions du système.

Construction du diagramme de cas d'utilisation

37

- Analyser bien le cahier de charge du système à développer
- Délimiter le système
- Identifier les acteurs
- Recenser les cas d'utilisation du point de vue de chaque acteur
- Améliorer le diagramme en appliquant les relations suivantes tant que possible et jusqu'à l'obtention du diagramme final:
 - Inclusion, Extension, Généralisation
- Documenter et valider le diagramme par :
 - Des descriptions textuelles pour tous les cas d'utilisation du système.
 - D'autres diagramme UML tels que les diagrammes de séquence et d'activité.

Attention: on peut obtenir plusieurs diagrammes possibles

Exemple : Gestion de bibliothèque

38

- Un gérant de bibliothèque désire automatiser la gestion des prêts
- Il commande un logiciel permettant aux utilisateurs de connaître les livres présents et d'en réserver jusqu'à 2.
- L'adhérent possède un mot de passe et une carte magnétique qui lui sont donnés à son inscription et ceci pour s'identifier.
- L'emprunt est toujours réalisé par les employés qui travaillent à la bibliothèque. Après avoir identifié l'emprunteur, ils savent si le prêt est possible (nombre max de prêts = 5), et s'il a la priorité (il est celui qui a réservé le livre).
- Ce sont les employés qui mettent en bibliothèque les livres rendus et les nouveaux livres. Il leur est possible de connaître l'ensemble des prêts réalisés dans la bibliothèque

Exemple : Gestion de bibliothèque

39

- Un gérant de bibliothèque désire automatiser la gestion des prêts
- Il commande un logiciel permettant aux **utilisateurs** de connaître les livres présents et d'en réserver jusqu'à 2.
- **L'adhérent** possède un mot de passe et une carte magnétique qui lui sont donnés à son inscription et ceci pour s'identifier.
- L'emprunt est toujours réalisé par les **employés** qui travaillent à la bibliothèque. Après avoir identifié l'emprunteur, ils savent si le prêt est possible (nombre max de prêts = 5), et s'il a la priorité (il est celui qui a réservé le livre).
- Ce sont les **employés** qui mettent en bibliothèque les livres rendus et les nouveaux livres. Il leur est possible de connaître l'ensemble des prêts réalisés dans la bibliothèque

Exemple : Gestion de bibliothèque

40

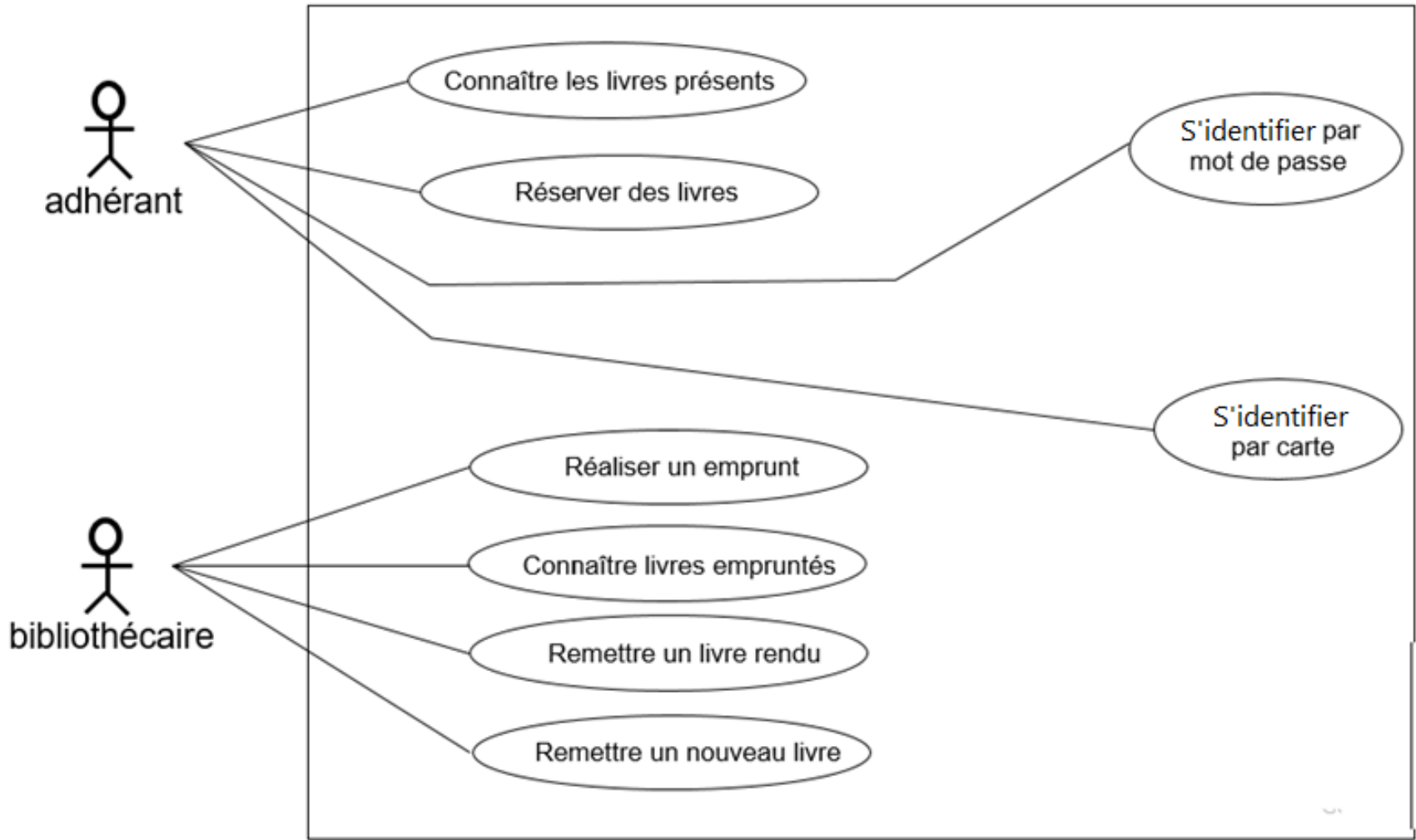
- Un gérant de bibliothèque désire automatiser la gestion des prêts
- Il commande un logiciel permettant aux **utilisateurs** de **connaître les livres présents** et **d'en réserver** jusqu'à 2.
- **L'adhérent** possède un mot de passe et une **carte magnétique** qui lui sont donnés à son inscription et ceci pour s'identifier.
- **L'emprunt est toujours réalisé** par les **employés** qui travaillent à la bibliothèque. Après **avoir identifié l'emprunteur**, ils savent si le prêt est possible (nombre max de prêts = 5), et s'il a la priorité (il est celui qui a **réservé le livre**).
- Ce sont les **employés** qui **mettent en bibliothèque les livres rendus et les nouveaux livres**. Il leur est possible de **connaître l'ensemble des prêts réalisés** dans la bibliothèque

Exemple : Gestion de bibliothèque

41

➤ Le diagramme préliminaire

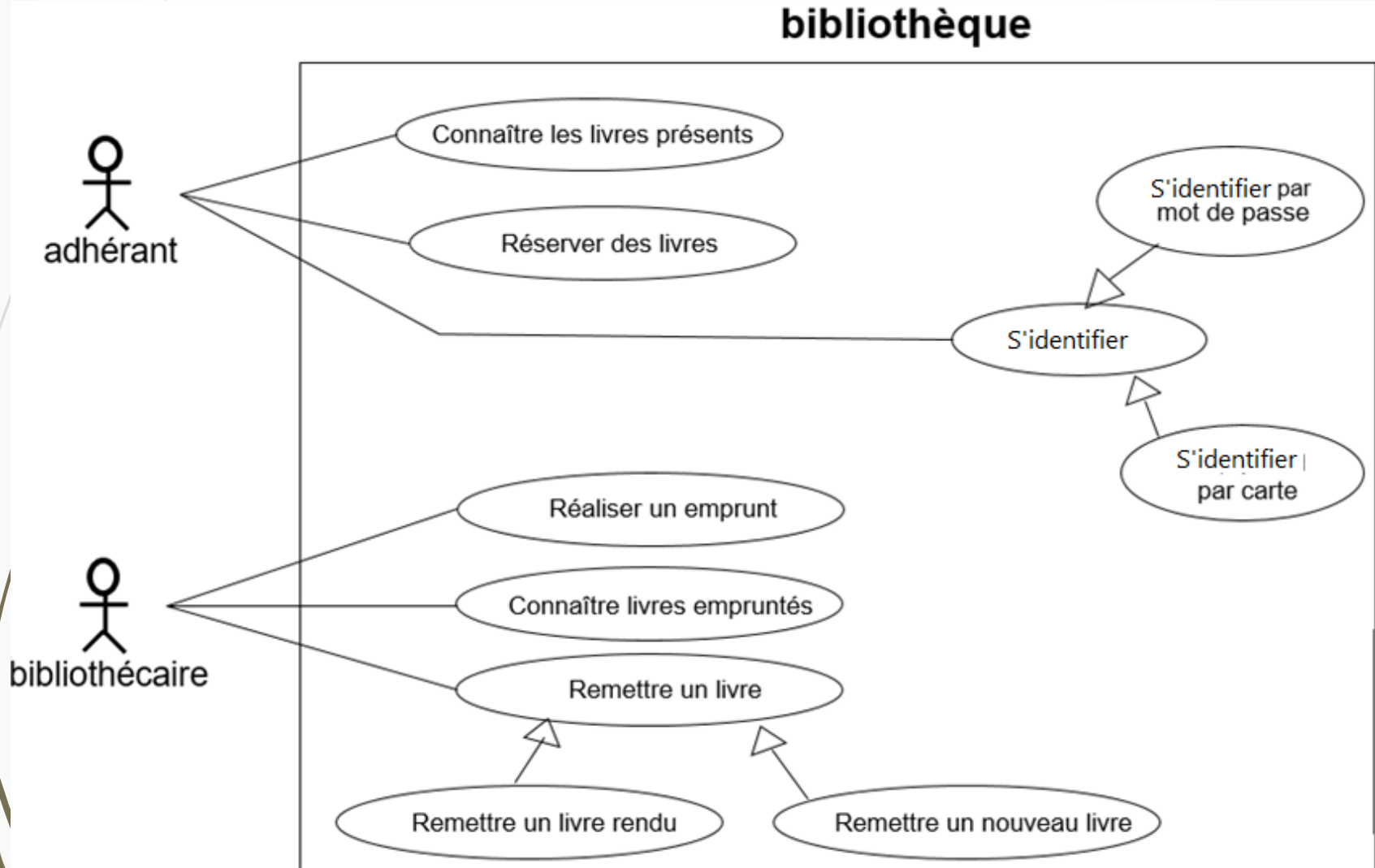
bibliothèque



Exemple : Gestion de bibliothèque

42

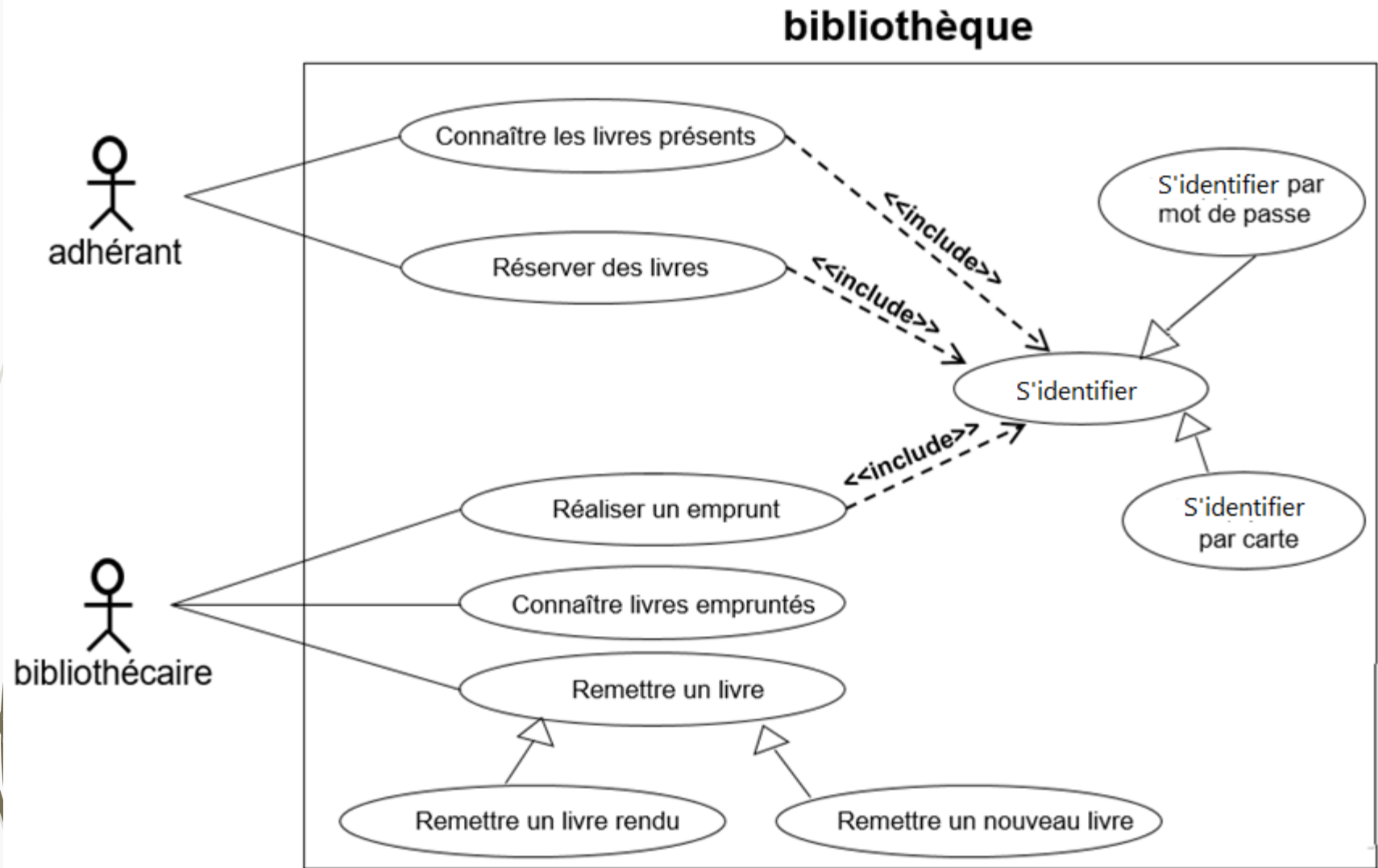
- Utilisation de généralisations



Exemple : Gestion de bibliothèque

43

Utilisation d'inclusion

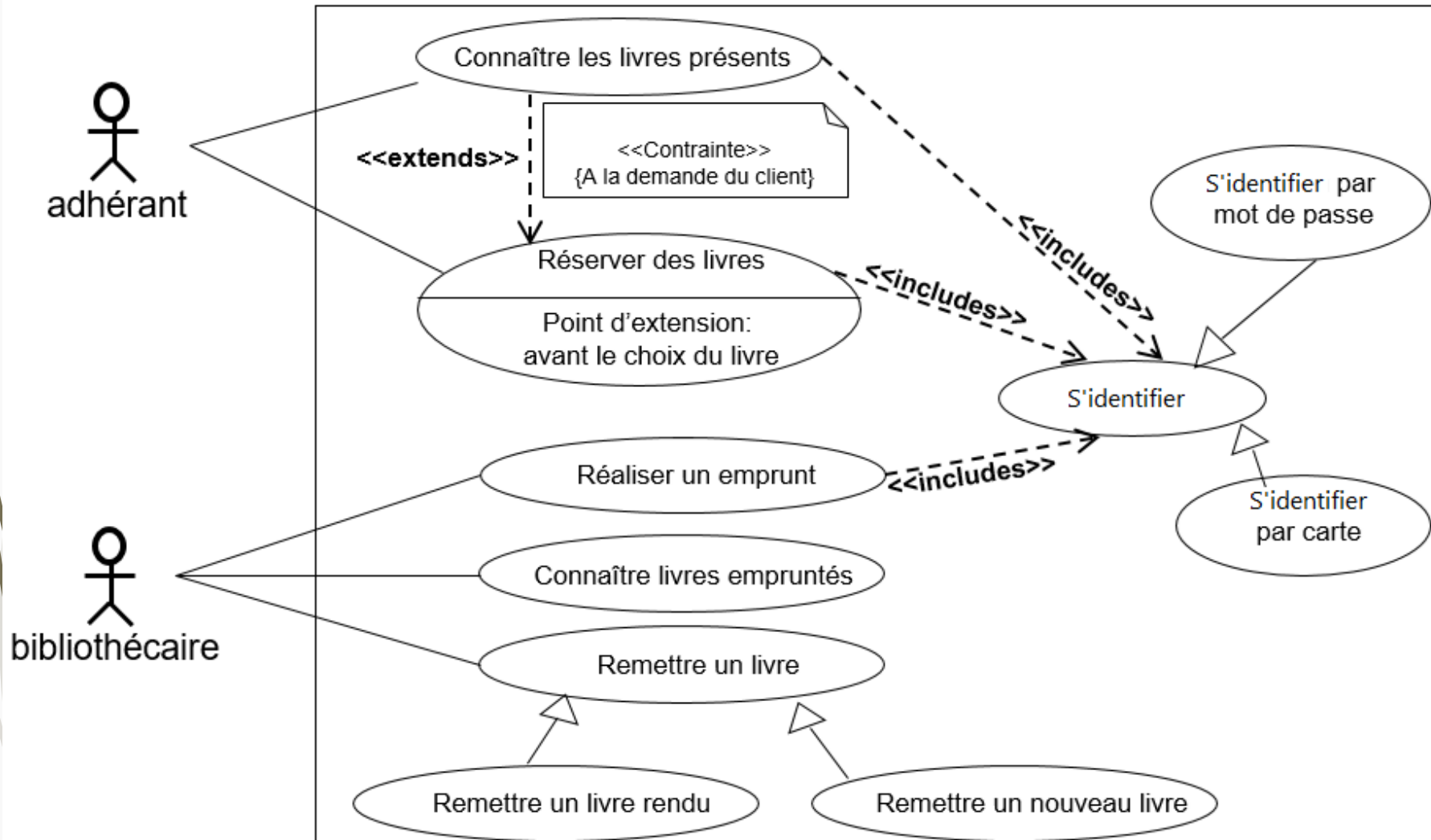


Exemple : Gestion de bibliothèque

44

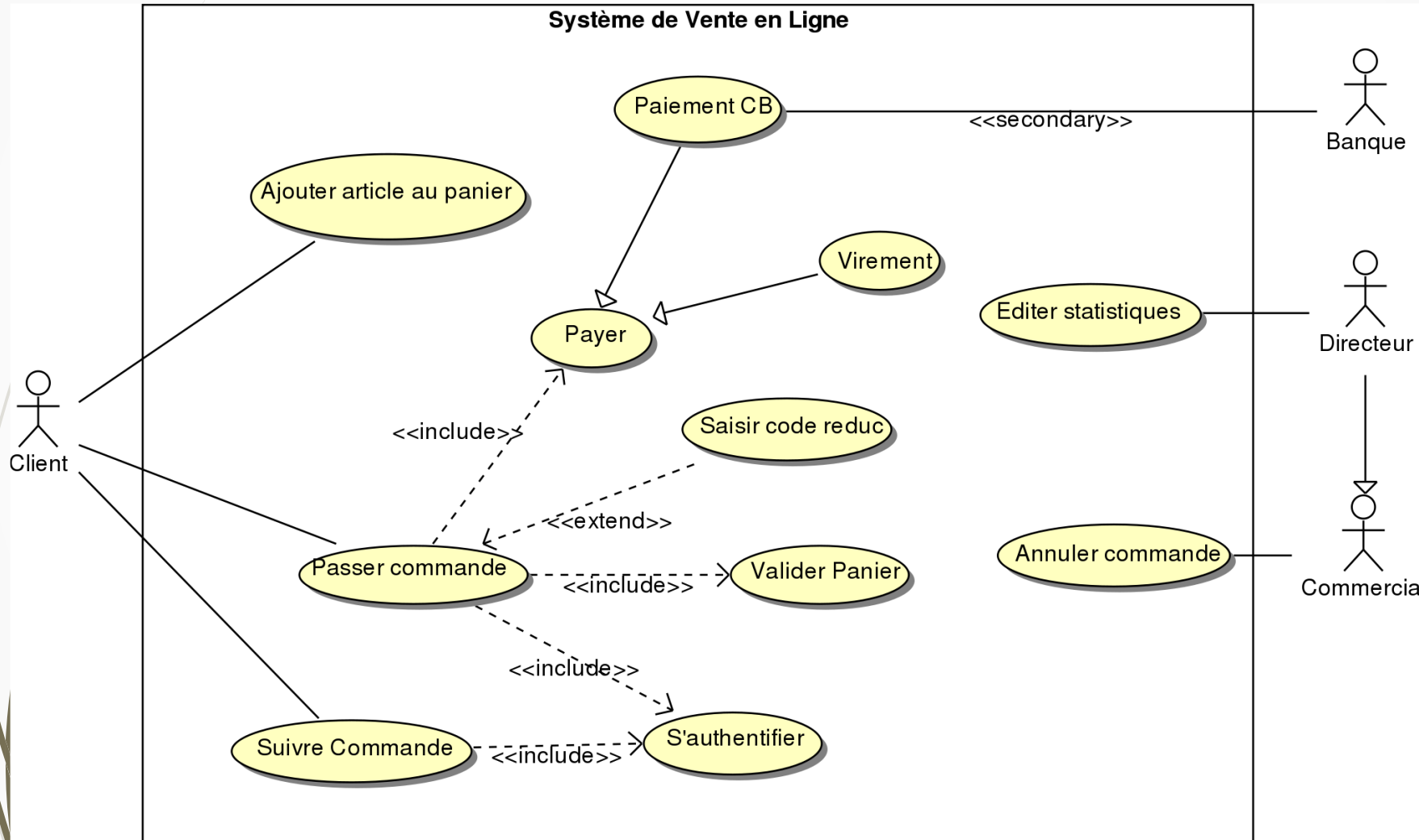
Utilisation des extensions

bibliothèque



Exemple de diagramme de cas d'utilisation

45



Cas d'utilisation vs scénario

46

- **Cas d'utilisation** : représente un cas en général, une représentation générale et synthétique d'un ensemble de scénarios similaires décrits sous la forme d'enchaînements.
- **Scénario** : Il décrit un cas d'utilisation *spécifique*
 - Cette description se fait de manière simple, par un texte compréhensible par les personnes du domaine de l'application.

Cas d'utilisation vs scénario

47

Pour décrire ou valider un Cas d'Utilisation : les scénarios

- Un scénario est un exemple :
 - une manière particulière d'utiliser le système ...
 - ... par un acteur particulier ...
 - ... dans un contexte particulier.
- cas d'utilisation = ensemble de scénarios
- scénario = une exécution particulière d'un Cas d'Utilisation

Exemple de scenario : nominal

48



Paul demande le paiement par carte bancaire

1. Paul saisit les informations de sa carte bancaire
2. Le système vérifie que le numéro de CB est correct
3. Le système vérifie la carte auprès du système bancaire
4. Le système demande au système bancaire de débiter le client
5. Le système notifie Paul du bon déroulement de la transaction

Un autre Scenario : un autre client Pierre, il saisit les informations de sa carte et le système trouve que le numéro est incorrect, alors le système lui demande de recommencer à partir de l'étape 1.

Exemple de scenario : avec erreurs

49

Païement CB

Pierre demande le paiement par carte bancaire

1. Pierre saisit les informations de sa carte bancaire
2. Le système vérifie le numéro de CB, il le trouve incorrect
3. Le système lui demande de ressaisir le numéro de CB
4. Pierre saisit les informations de sa carte bancaire
5. Le système vérifie le numéro de CB, il le trouve incorrect
6. Le système lui demande de ressaisir le numéro de CB
7. Pierre saisit les informations de sa carte bancaire
8. Le système vérifie le numéro de CB, il le trouve correct
9. Le système vérifie la carte auprès du système bancaire
10. Le système demande au système bancaire de débiter le client
11. Le système notifie Pierre du bon déroulement de la transaction

Description des cas d'utilisation

50

- Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation.
- Un simple nom est tout à fait insuffisant pour décrire un cas d'utilisation.
- Chaque cas d'utilisation doit être documenté pour qu'il n'y ait aucune ambiguïté concernant son déroulement et ce qu'il recouvre précisément.
- Description détaillée plus ou moins formelle
 - langue naturelle mais structurée, vocabulaire précis
 - Scénarios décrits plus formellement avec des diagrammes UML (Diagramme de sequence, ...)
- Il est possible de décrire un cas d'utilisation par un diagramme d'activité

Description textuelle d'un cas d'utilisation

51

Identification :

- Nom du cas : **Nom**
- Objectif : **Resumés sur le cas**
- Acteurs : **Client (principal), Banque (secondaire)**
- Date : **1/12/2018** date de mise a jour : **5/5/2019**
- Responsables : **Dupond**
- Version : **1.0**
- Début du cas d'utilisation : événement qui le déclenche
 - « Le cas d'utilisation débute quand X se produit »
- Fin du cas d'utilisation : événement qui cause son arrêt
 - « Quand Y se produit, le cas d'utilisation est terminé »

Description textuelle

52

➤ Pré-conditions : Pré-conditions

➤ Description des scenarios

➤ Scenario nominal :

1. Etape 1

2. Etape 2, ...



Scenario sans erreurs :
tout va bien

➤ Enchaînements alternatifs

1. Etape 1

2. Etape 2, ...



Scenarios avec des fautes mais ne
font pas sortir de l'exécution du cas

➤ Enchaînement d'erreurs

1. Etape 1

2. Etape 2,



Scenarios avec des erreurs qui
arrêtent l'exécution du cas

➤ PostConditions : Postconditions

➤ Contraintes non fonctionnelles :

➤ Contraintes liées à l'interface homme-machine

} Rubriques
optionnelles

Description textuelle

53

- Identification :
- **Nom du cas** : Payer CB
- **Objectif** : Détailler les étapes permettant à un client de payer par carte bancaire
- **Acteurs** : Client (principal), Banque (secondaire)
- **Date** : 1/12/2018 **date de mise à jour** : 5/5/2019
- **Responsables** : Dupond
- **Version** : 1.0

Description textuelle

54

- Le cas d'utilisation commence lorsqu'un client demande le paiement par carte bancaire
- **Pré-conditions**
 - Le client a validé sa commande
- **Scenario nominal :**
 1. Le client saisit les informations de sa carte bancaire
 2. Le système vérifie que le numéro de CB est correct
 3. Le système vérifie la carte auprès du système bancaire
 4. Le système demande au système bancaire de débiter le client
 5. Le système notifie le client du bon déroulement de la transaction

Description textuelle

55

➤ Enchaînements alternatifs

1. En (2) : si le numéro est incorrect, le client est averti de l'erreur, et invité à recommencer, on continue à partir de 1
2. ...

➤ Post-conditions

- La commande est validée
- Le compte de l'entreprise est crédité

Description textuelle

56

➤ Rubriques optionnelles

➤ Contraintes non fonctionnelles :

- Fiabilité : les accès doivent être sécurisés
- Confidentialité : les informations concernant le client ne doivent pas être divulgués

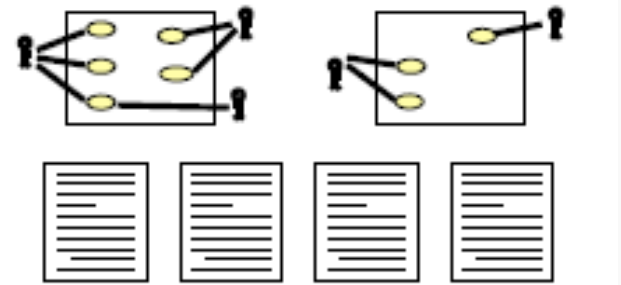
➤ Contraintes liées à l'interface homme-machine :

- Toujours demander la validation des opérations bancaires

Modèle des cas d'utilisation

57

- Un **modèle de cas d'utilisation** d'un système ou d'une partie d'un système (grande fonction) peut être formé
 - de plusieurs diagrammes de cas d'utilisation,
 - de descriptions textuelles ou de diagrammes précisant chacun des cas d'utilisation
 - Diagrammes de sequences
 - Diagrammes d'activité
 - ...



Exemple : description d'un cas d'utilisation

58

Réserver d'un livre

<u>Préconditions:</u>	<ul style="list-style-type: none">-le client doit être inscrit à la bibliothèque-le client ne doit pas avoir atteint le nombre maximum de réservation-un exemplaire du livre doit être enregistré	
<u>Postconditions:</u>	<ul style="list-style-type: none">-le client a une réservation supplémentaire-le nombre d'exemplaires disponibles du livre est décrémenté de 1	
<u>Action de l'acteur</u>	<u>Action du système</u>	
2) Le client choisit l'opération réservation parmi les différentes opérations proposées	1) Le système affiche un message d'accueil	
4) Le client donne son identification	3) Le système lui demande de s'authentifier	
6) Le client précise le livre qu'il désire	5) Le système demande le titre du livre en donnant la possibilité de choisir dans une liste	
	7) Le système lui précise qu'un exemplaire lui est réservé	

Exemple : description d'un cas d'utilisation

59

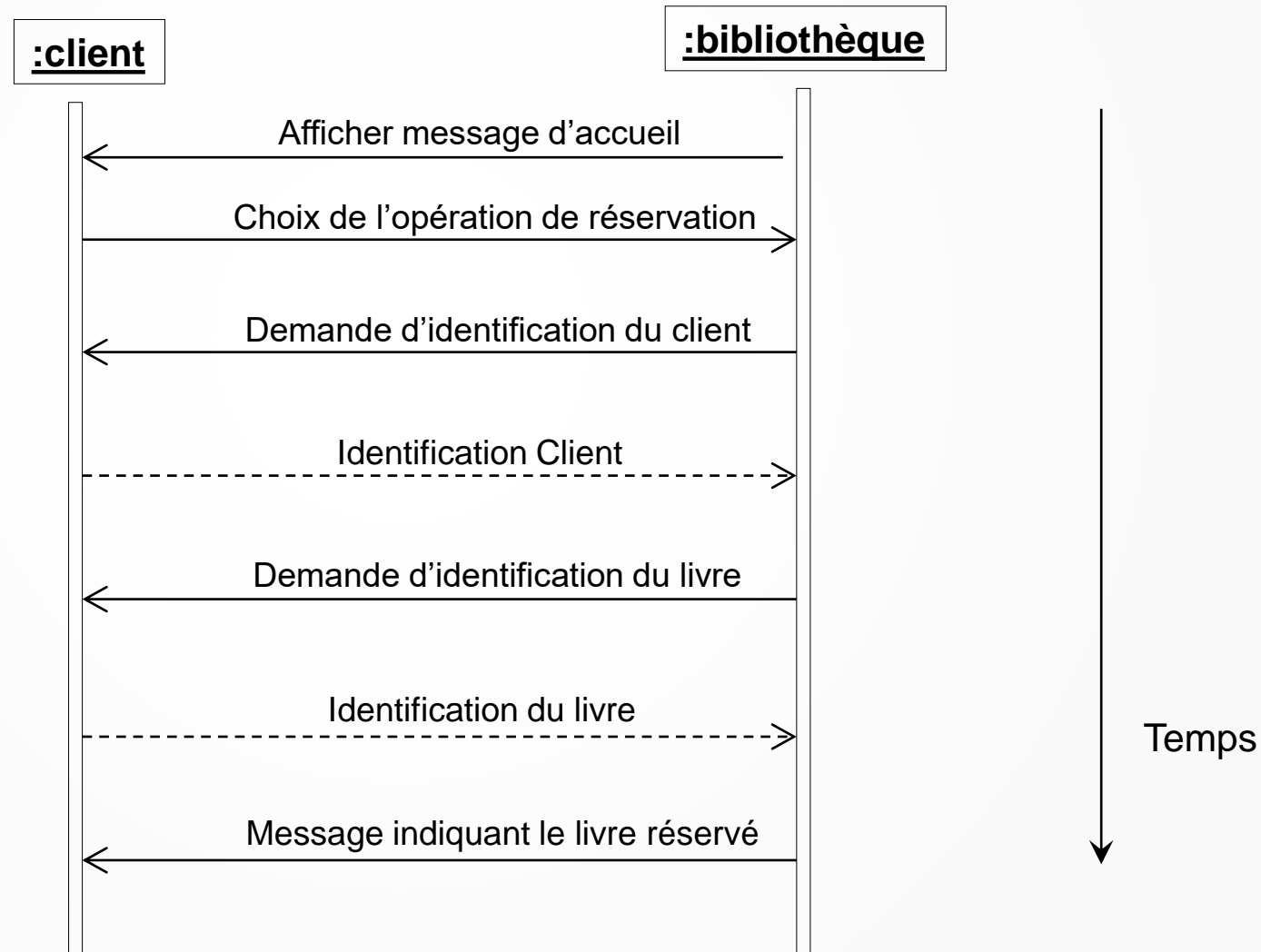
➤ Enchaînements alternatifs

1. En (6), le client demande à connaître les livres présents
2. En (4), le client n'est pas reconnu. Dans ce cas, tant qu'il n'est pas reconnu, on lui redemande de s'authentifier
3. En (4), le client est reconnu mais le mot de passe est incorrect. Il peut avoir 5 essais, par la suite le client sera interdit pendant 1 heure.
4. En (4), le client n'a plus le droit de réserver
5. En (7), le livre n'est pas libre

Exemple : description d'un cas d'utilisation

60

❑ Scénario par diagramme de séquence



Exercice

61

- Soient les cas d'utilisation suivants :
 - Passer une commande
 - Passer une commande urgente
 - Suivre une commande
 - Valider l'utilisateur
 - Expédier commande totale ou partielle
- Le suivi de la commande désigne le processus complet, du passage à l'expédition. Il peut toutefois arriver qu'une commande passée ne soit pas envoyée. Passer une commande urgente est un cas particulier de passer une commande. Pour passer une commande, il faut nécessairement valider l'utilisateur.
- **Question** : Donner le diagramme de cas d'utilisation sans représenter les acteurs

Exercise

62

