



UML - UCs

UE15

Informatique appliquée

Frédéric Pluquet

pluquetf@helha.be



Diagramme de cas d'utilisation (Use case diagram)

- Diagrammes de classes, objets et packages
 - Décrivent les données du système analysé
- Diagramme de cas d'utilisation
 - Décrivent les interactions avec le système analysé
 - Définissent les limites en terme de fonctionnalités

Sommaire

- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - Communication entre acteurs et système
- Problèmes récurrents
 - Cas d'utilisation orienté-solutions / cas d'utilisation essentiels
 - Problèmes des intermédiaires
 - Problèmes liés à la relation acteur - cas d'utilisation
 - Problème de granularité

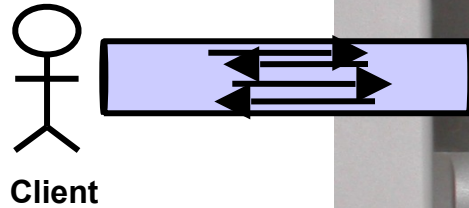
Idée



Client



Idée : interactions (canal de communication)



Idée : différents acteurs



Diagramme de cas d'utilisation (exemple 1)

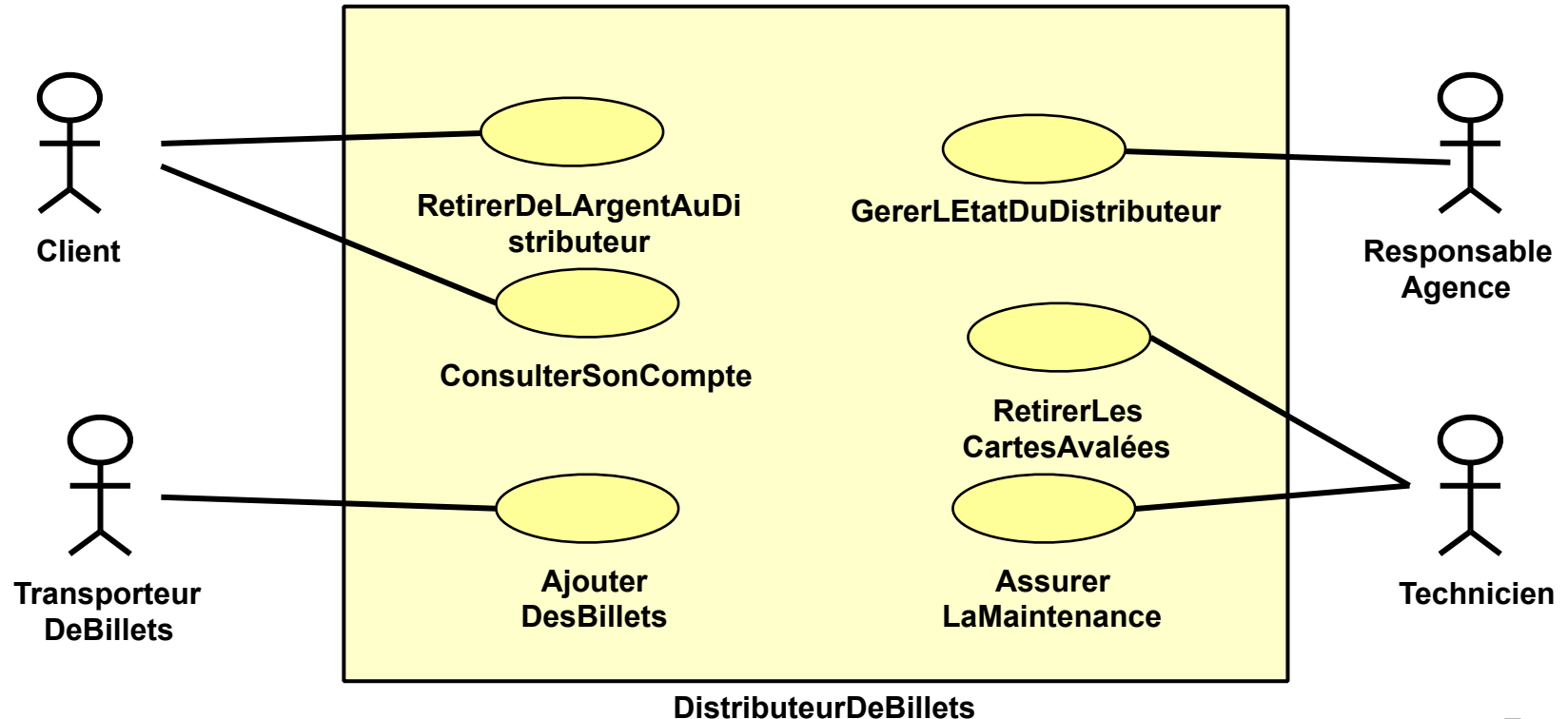


Diagramme de cas d'utilisation (exemple 2)

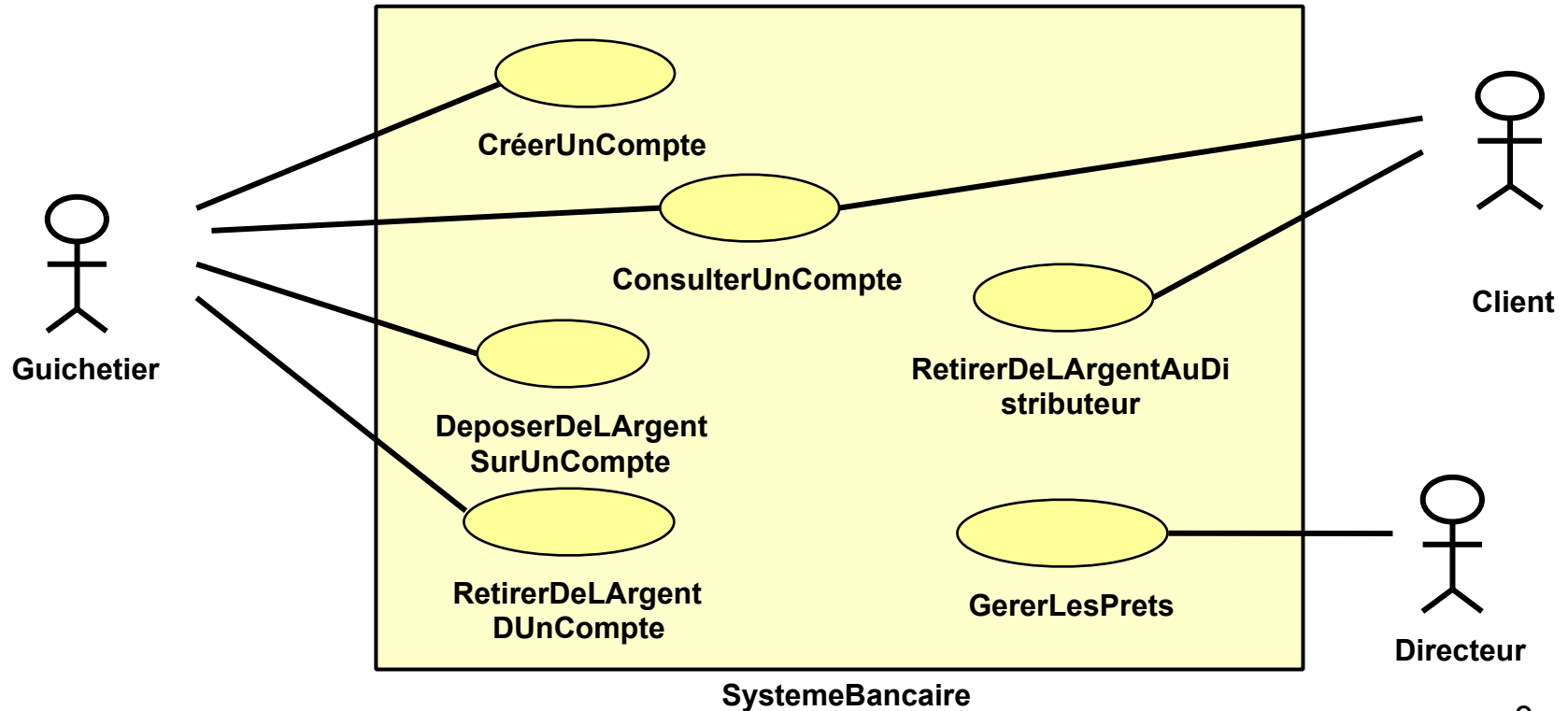
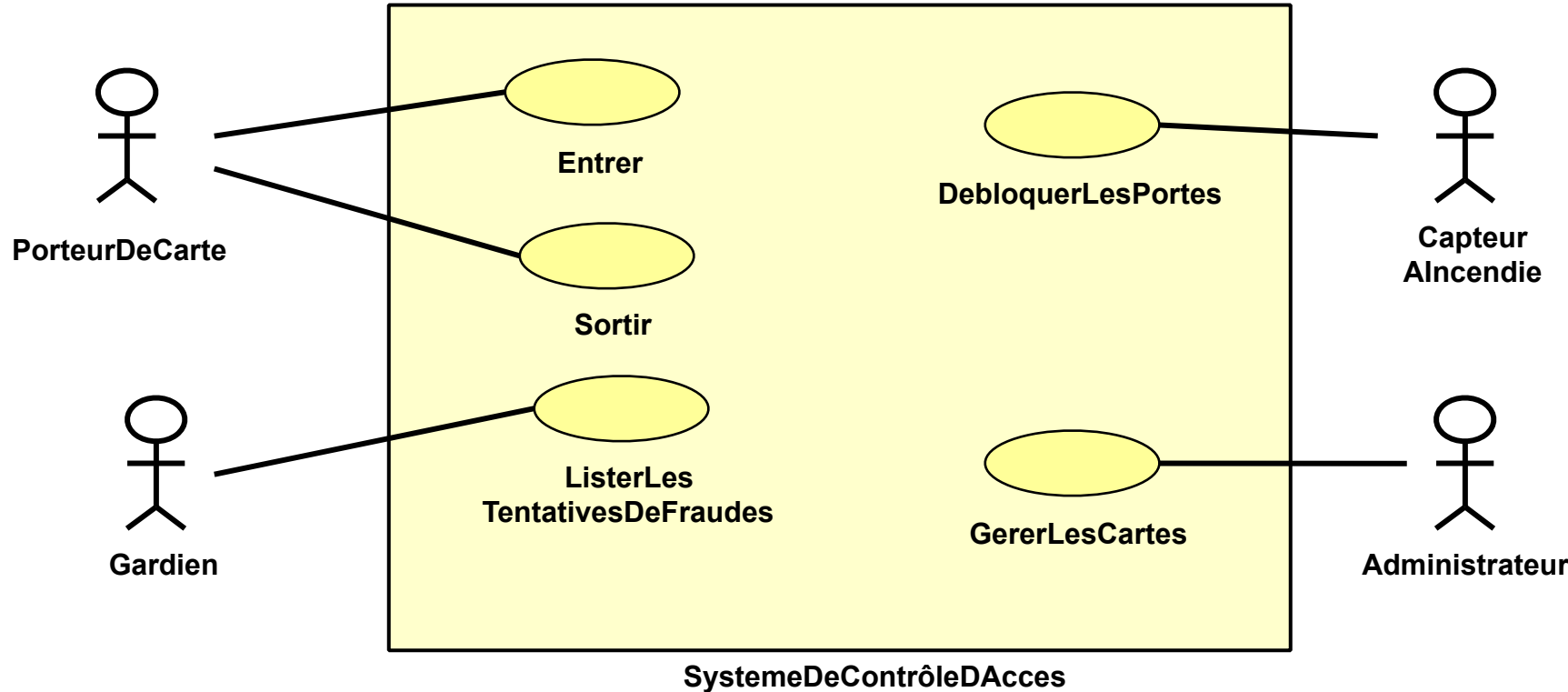


Diagramme de cas d'utilisation (exemple 3)



Éléments de base



SystèmeN



CasDUtilisationN



ActeurN

Les acteurs



Un Acteur est

- élément **externe** qui *interagit* avec le système (prend des décisions, des initiatives. Il est "actif")
- **rôle** qu'un "utilisateur" joue par rapport au système



PorteurDeCarte



Gardien



Administrateur



Capteur
Alcendie

Acteurs vs. utilisateurs



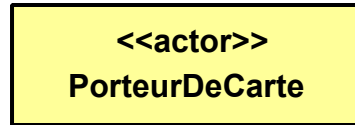
- Une même personne peut jouer plusieurs rôles
ex: Maurice est directeur mais peut jouer le rôle de guichetier
- Plusieurs personnes peuvent jouer un même rôle
ex: Paul et Pierre sont deux clients
- Un rôle par rapport au système plutôt que position dans l'organisation
ex: PorteurDeCarte plutôt qu'Enseignant
- Un acteur n'est **pas forcément** un être **humain**
ex: un distributeur de billet peut être vu comme un acteur

Différentes notations

- Ces trois notations indiquent le même acteur



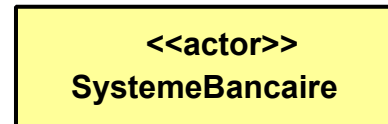
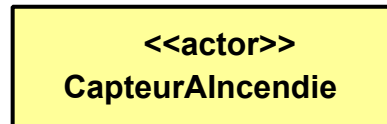
PorteurDeCarte



- Nous allons utiliser le stéréotype **actor** uniquement lorsque l'acteur n'est pas joué par un humain



PorteurDeCarte



Différents types d'acteurs



- Utilisateurs principaux
ex: client, guichetier
- Utilisateurs secondaires
ex: contrôleur, directeur, ingénieur système, administrateur...
- Périphériques externes
ex: un capteur, une horloge externe, ...
- Systèmes externes
ex: système bancaires

Acteur : règles de nommage



- Forme nominale
- Vocabulaire métier
- Style CamelCase

Cas d'utilisation (CU ou UC)

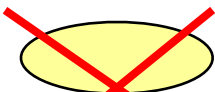


CasDUtilisation

- Représente
 - une manière d'utiliser le système
 - une suite d'interactions entre un acteur et le système
- Correspond à une fonction du système **visible par l'acteur**
- Permet à un acteur d'atteindre **un but**
- Doit être **utile en soi**
- Regroupe un ensemble de scenarii correspondant à un même **but**
- Possède une **unité de temps**



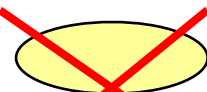
Entrer



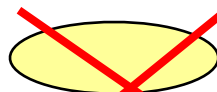
EnregistrerEntrée



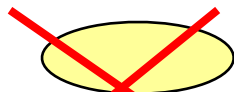
RetirerDeLArgentAuDi
stributeur



Sidentifier



EntrerPendant
LesHeuresDOuverture



TaperSonCode

Cas d'utilisation : règles de nommage



- Vocabulaire métier
 - Forme verbale décrivant une action
 - L'acteur est généralement le sujet
 - Éviter les connecteurs (et, ou, puis, ...)
 - Style CamelCase ou littéral
 - Terme générique comme "Gérer" en cas de besoin seulement
 - Gérer = Créer, Lire, Ajouter/Modifier & Supprimer (CRUD)
- Exemple: GérerLesDroits

Unicités des cas d'utilisation

Chaque CU doit respecter les 4 unicités suivantes :

- **Unicité d'objectif** Le CU doit fournir un résultat appréciable à l'acteur avec lequel il interagit.
- **Unicité de responsabilité** Pour un CU interactif, un seul acteur direct
- **Unicité de temps**
 - **Unicité d'exécution** Un enchaînement d'actions qui se suivent sans interruption
 - **Unicité de périodicité** Exécution à la même fréquence
- **Unicité de mode** manuel, interactif ou automatisé, et unitaire ou par lot

Unicités des cas d'utilisation : Mode

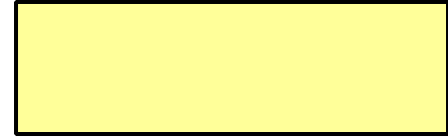
	Unitaire	Par lot
Manuel	Remplir un formulaire	Vérifier manuellement tous les formulaires reçus
Interactif	Enregistrer un passager sur un vol	Encoder les 1000 livres achetés par la bibliothèque
Automatisé	Traitement d'un signal d'alarme	Envoyer au siège central les opérations bancaires d'un distributeur de billets toutes les heures

Le système

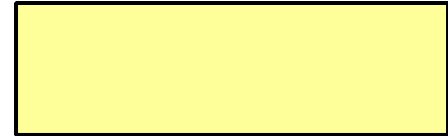
- Le système est
 - modélisé par un ensemble de cas d'utilisation
 - vu comme une **boîte noire**
- Le système contient les cas d'utilisation
 - mais pas les acteurs
- Permet de définir
 - les limites du système



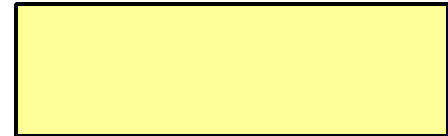
Système



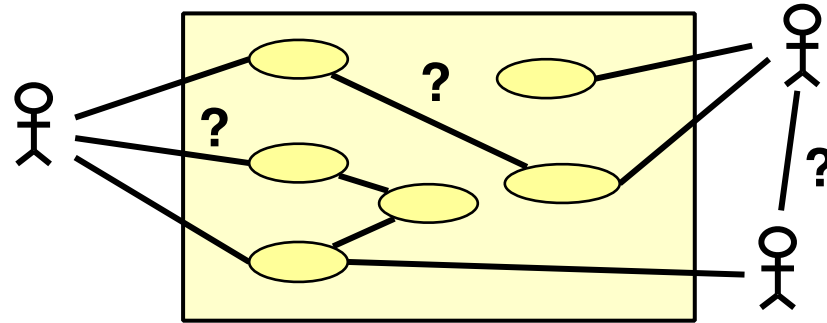
DistributeurDeBillets



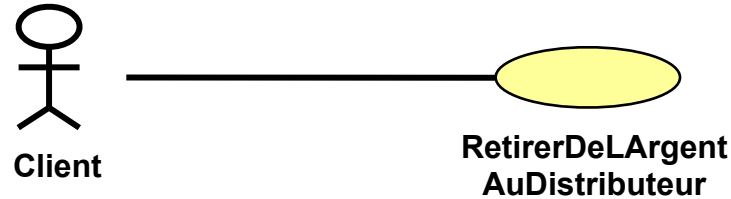
SystemeDeControleDAcces



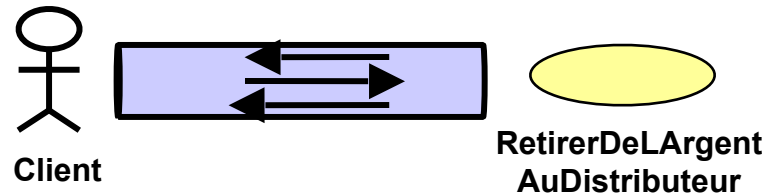
SystemeDeSécuritéIncendie



Relation acteur - cas d'utilisation

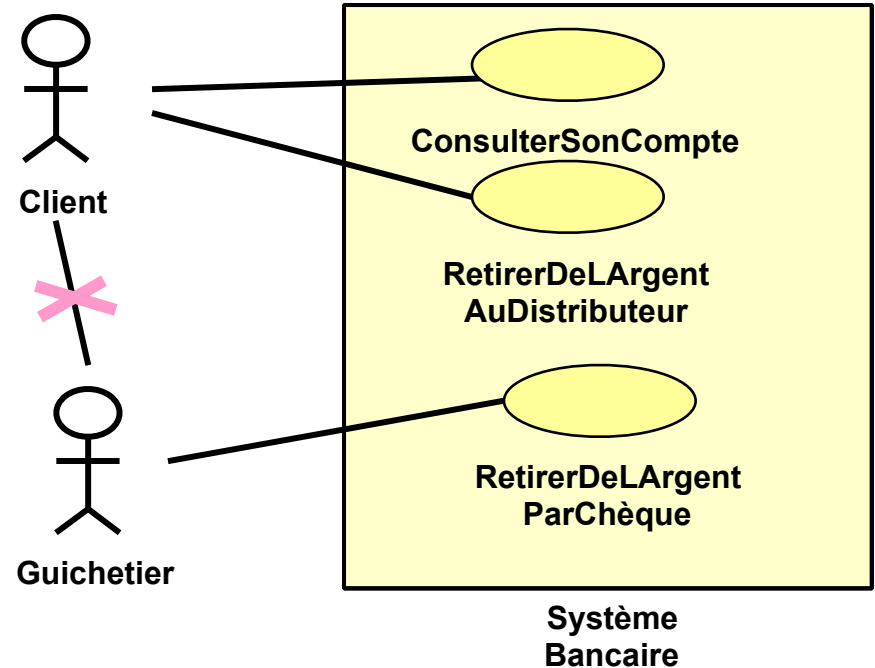


- vision objectif : possibilités d'atteindre un but
- vision technique : canal/protocole de communication



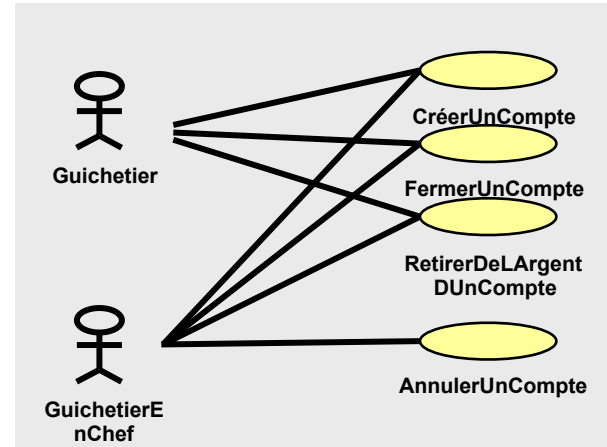
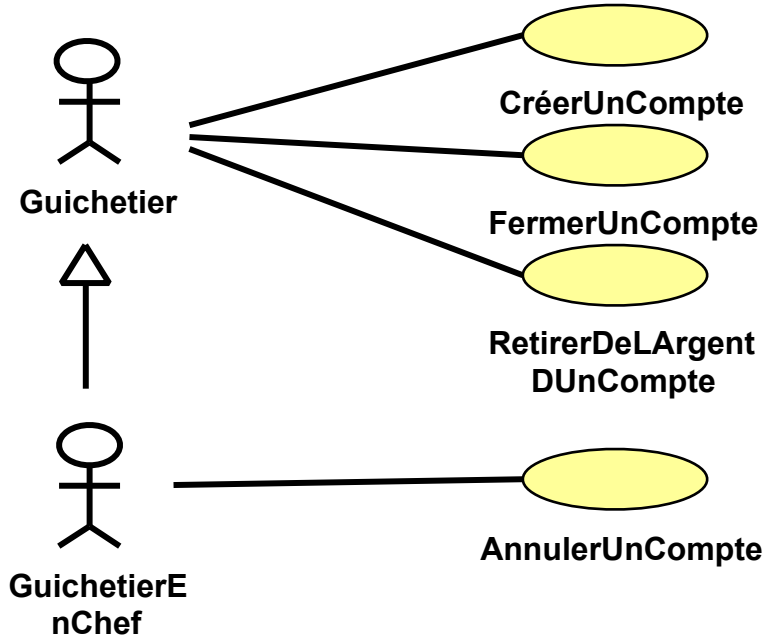
Relation acteur - acteur : communication

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



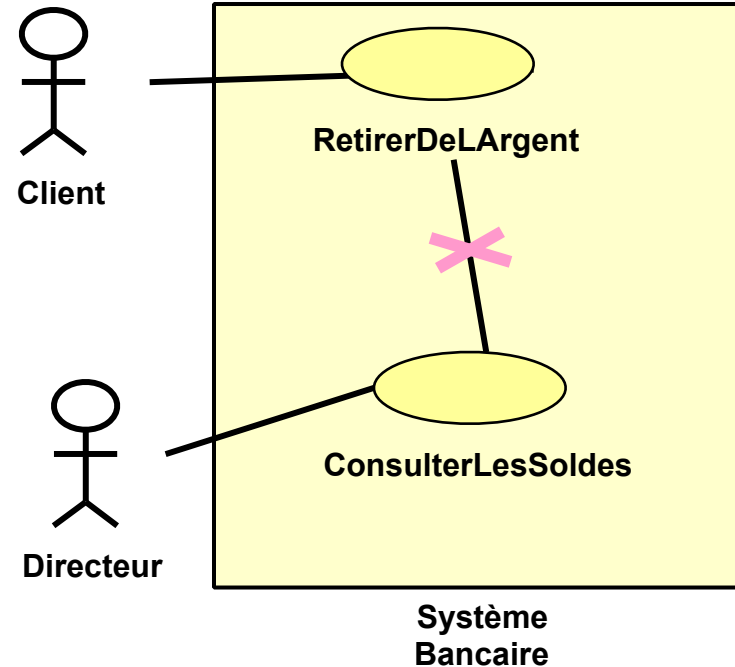
Relation acteur - acteur : généralisation

La seule relation entre acteurs est la relation de généralisation



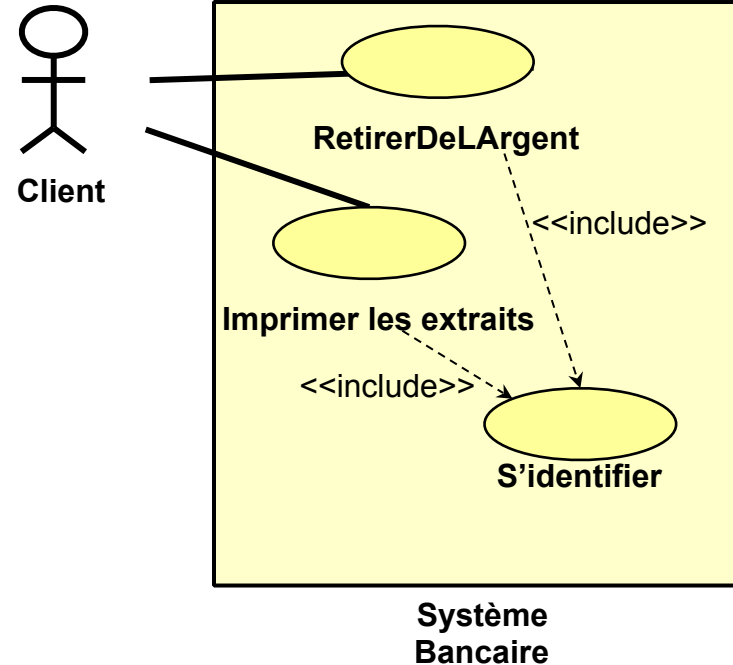
Relation acteur - cas d'utilisation

- Communications internes non modélisées
- Interactions système ↔ extérieur
- Formalisme bien trop pauvre pour décrire l'intérieur du système. Utiliser les autres modèles UML pour cela.
- Autres relations possibles entre CU
 - Héritage, inclusion et extension



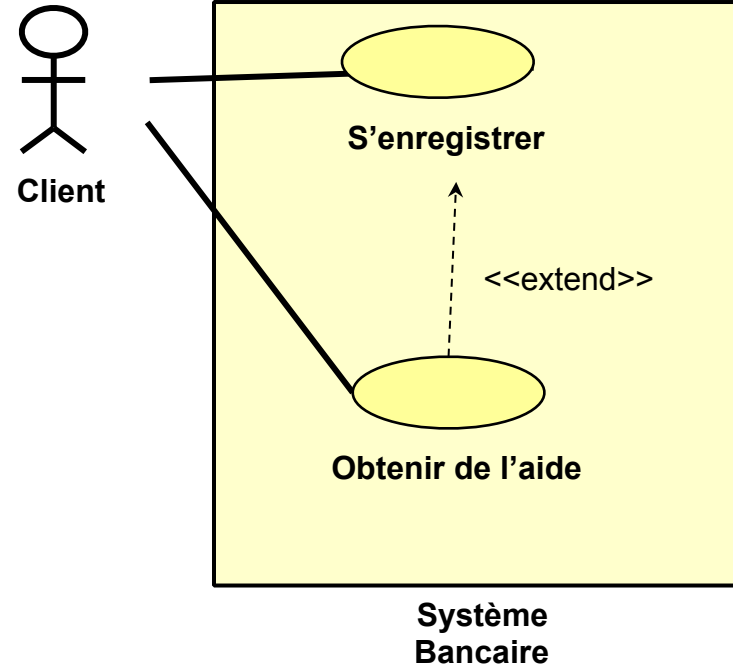
Dépendance entre CU : include

- Si plusieurs CU ont des sous-ensembles communs d'actions
- Permet la factorisation des CU
- Une instance de RetirerDeLArgent va engendrer une instance de S'identifier
- RetirerDeLArgent dépend de S'identifier
- S'identifier n'existe pas seul
- La relation d'inclusion suppose une **obligation** d'exécution des interactions



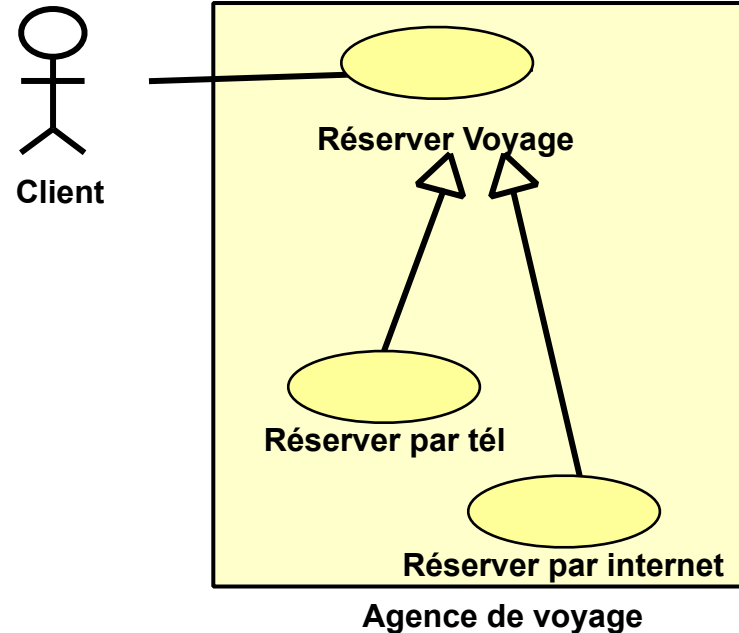
Dépendance entre CU : extend

- Permet d'étendre les fonctionnalités d'un CU
- Ce CU peut fonctionner seul mais peut également être complété par un autre, sous certaines conditions et à certains moments précis
- La relation d'extension suppose une **option** d'exécution des interactions (pas d'obligation)



Généralisation/Spécification des CU

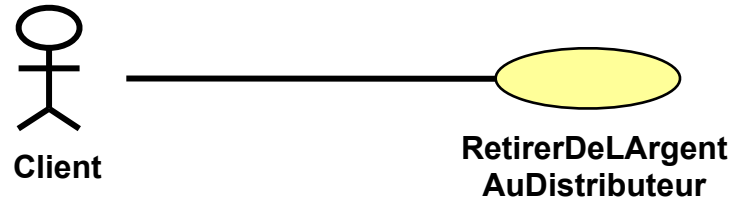
- Un CU peut hériter d'un autre CU
- À voir comme un *polymorphisme* de cas
- Le but est le même mais les interactions pour y arriver ne sont pas les mêmes



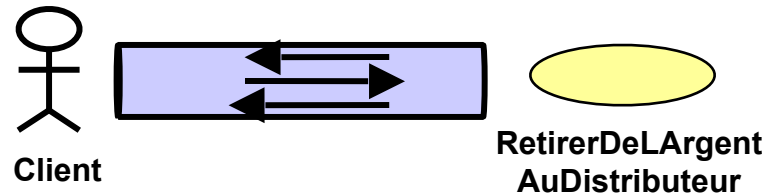
Sommaire

- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - **Communication entre acteurs et système**
- Problèmes récurrents
 - Cas d'utilisation orienté-solutions / cas d'utilisation essentiels
 - Problèmes des intermédiaires
 - Problèmes liés à la relation acteur - cas d'utilisation
 - Problème de granularité

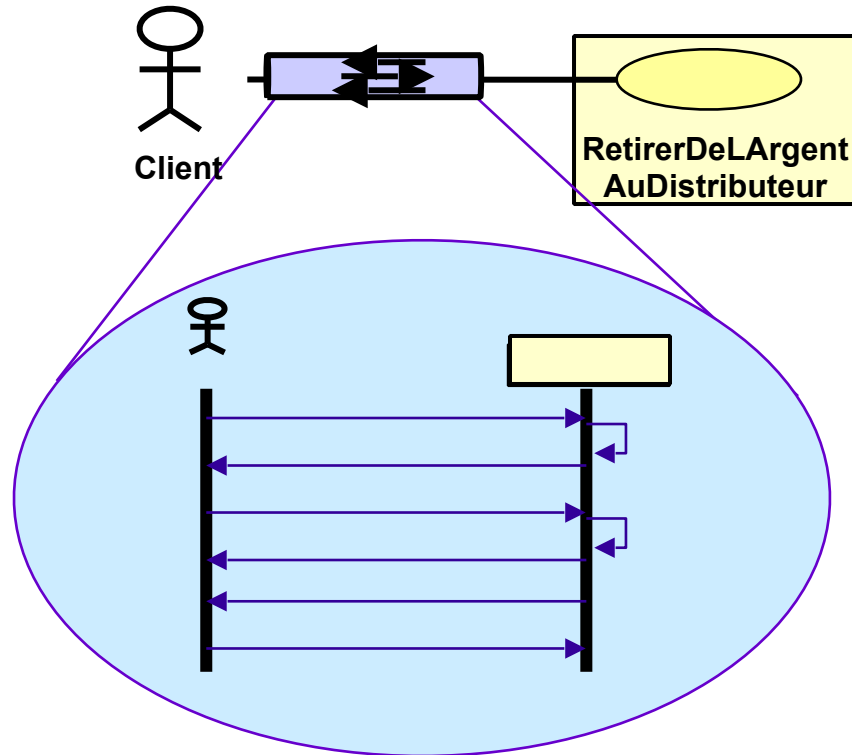
Retour sur la relation acteur - cas d'utilisation



- Canal de communication
- Décrit le comportement du système vu de l'extérieur
- Echange de messages



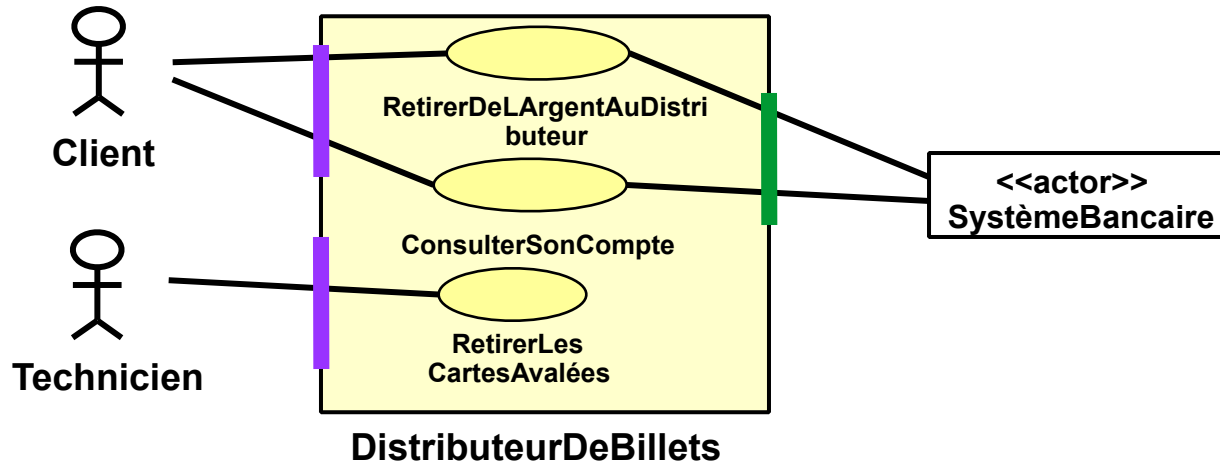
Description de l'interaction



Diagrammes de
séquences
"système"
(cf. ANA3)

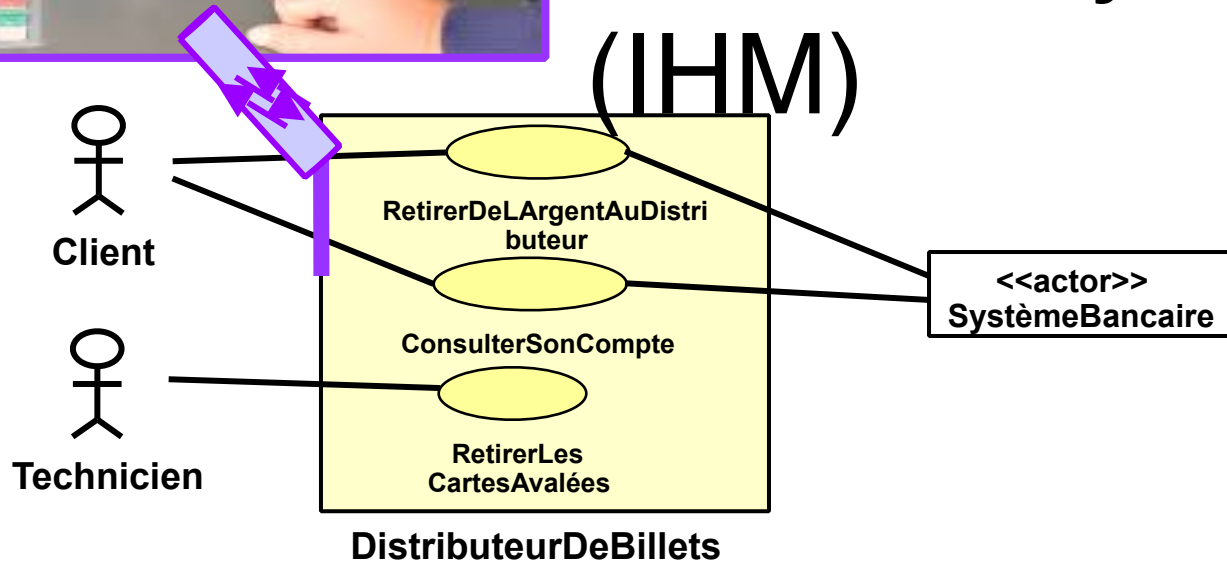
Limites du système et interface

- Humain → IHM : Interface homme - système (machine)
- Logiciel → API : Interface système - système



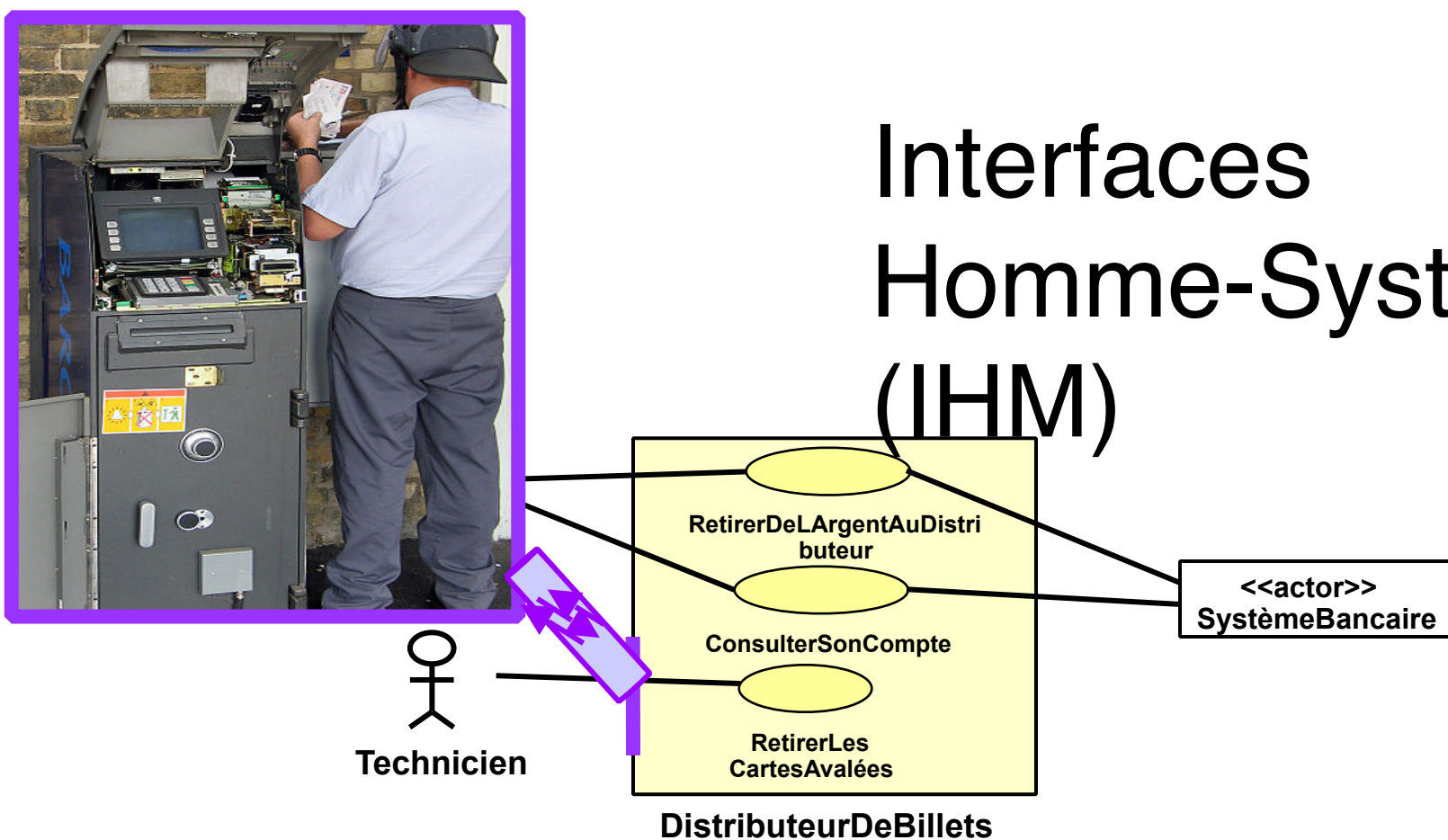


Interfaces Homme-Système (IHM)



Description (par la suite) dans les documents de spécification externes

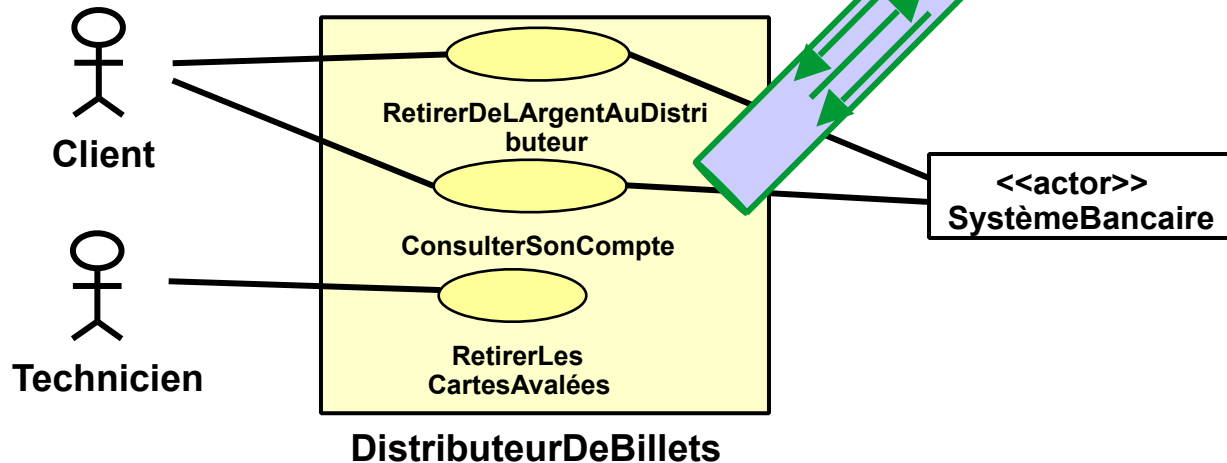
Interfaces Homme-Système (IHM)



Description (par la suite) dans les documents de spécification externes



Interfaces Système- Système



```
...  
<xs:element name="note">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="to" type="xs:string"/>  
      <xs:element name="from" type="xs:string"/>  
      <xs:element name="heading" ... >  
      <xs:element name="body" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
...
```

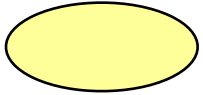
Description (par la suite) dans les documents de spécification externes

Description des cas d'utilisation



CasDUtilisationX

- Les cas d'utilisation sont alors décrits de manière textuelle
- Exemple :



**Retirer
DeLArgent
AuDistributeur**

Lorsqu'un *client* a besoin de d'argent liquide il peut en utilisant un distributeur retirer de l'argent de son compte. Pour cela :

- le *client* insère sa carte bancaire
- le *système* demande le code
- le *client* choisit le montant du retrait
- le *système* vérifie qu'il y a suffisamment d'argent sur le compte
- si c'est le cas, le *système* distribue les billets et débite le compte du client
- le *client* prend les billets et retire sa carte

Processus de conception

(1) Définir le modèle de cas d'utilisation

- (1.1) Trouver les acteurs
- (1.2) Décrire brièvement chaque acteur
- (1.3) Trouver les cas d'utilisation
- (1.4) Décrire brièvement chaque cas d'utilisation

(2) Détailler chaque CU (en ANA3)

Sommaire

- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - Communication entre acteurs et système
- **Problèmes récurrents**
 - Cas d'utilisation orienté-solutions / cas d'utilisation essentiels
 - Problèmes des intermédiaires
 - Problèmes liés à la relation acteur - cas d'utilisation
 - Problème de granularité

Problèmes récurrents

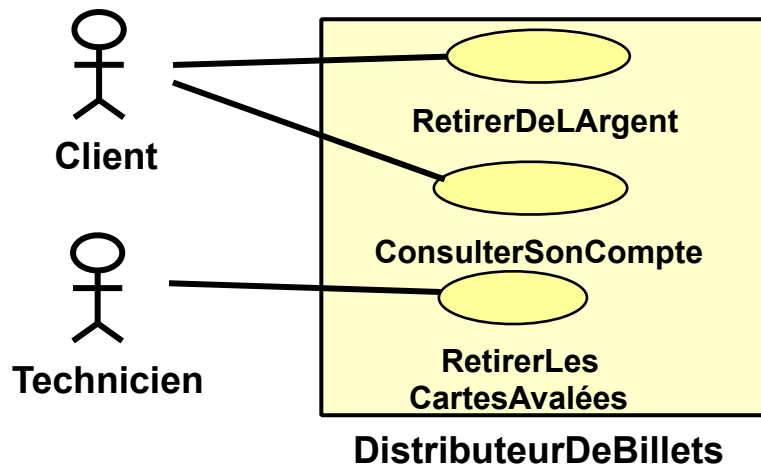
- Les problèmes soulevés dans cette partie correspondent à des questions récurrentes en pratique
 - Problèmes éventuellement sans réponse dans la norme
 - Interprétations et solutions parfois différentes dans les livres
 - Problèmes récurrents souvent implicites
- => Chercher quelles conventions existent dans le contexte de travail ou se mettre d'accord sur des conventions lorsque le problème se pose

Sommaire

- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - Communication entre acteurs et système
- Problèmes récurrents
 - **Cas d'utilisation orienté-solutions / cas d'utilisation essentiels**
 - Problèmes des intermédiaires
 - Problèmes liés à la relation acteur - cas d'utilisation
 - Problème de granularité

Problème : cas d'utilisation orientés-solution

- Il faut décrire les buts et les besoins des acteurs et les interactions
- mais **pas l'interface concrète**
 - Ne pas parler de boutons, click, ...
- Se concentrer sur le POURQUOI, POUR QUI et pas le COMMENT

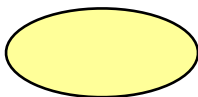


Se concentrer
sur l'essentiel

=> cas d'utilisation "essentiels"

Cas d'utilisation **essentiel**

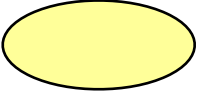
- **Ne pas décrire l'interface concrète**
- Décrire
 - Les objectifs et intentions de l'acteur
 - Décrire les responsabilités du système
 - Les “interactions” abstraites



**Retirer
DeLArgent
AuDistributeur**

- le *client* insère sa carte bancaire dans le distributeur
- le *système* demande le code pour l'identifier
- le *client* tape le montant du retrait sur le clavier
- le *système* vérifie qu'il y a suffisamment d'argent
- le système affiche un message de confirmation

Se concentrer sur l'essentiel

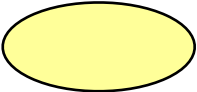


**Retirer
DeLArgent
AuDistributeur**

- le *client* insère sa carte bancaire dans le distributeur
- le *système* demande le code pour l'identifier
- le *client* tape le montant du retrait sur le clavier
- le *système* vérifie qu'il y a suffisamment d'argent
- le *système* affiche un message de confirmation
- ...



Extraction de l'essentiel



**Retirer
DeLArgent
AuDistributeur**

- le *client* s'identifie
- le *système* vérifie l'identification
- le *client* détermine le montant du retrait
- le *système* vérifie qu'il y a suffisamment d'argent

Se concentrer sur l'essentiel

- Idée : éviter les décisions trop rapides

analyse des besoins



modèle de cas d'utilisation (UML)

conception des
interfaces
personne-système



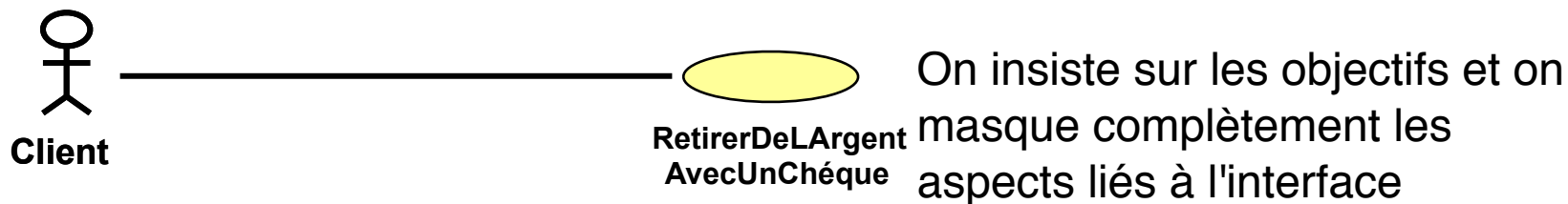
modèle de tâches ou autre
(pas dans UML)

Sommaire

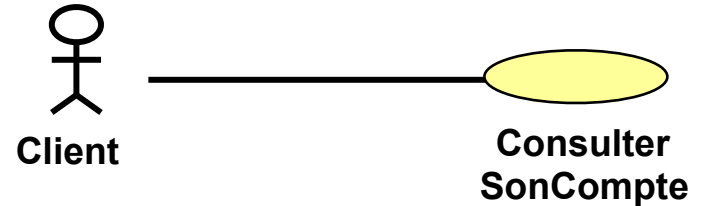
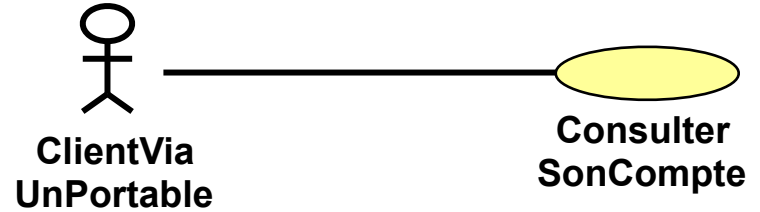
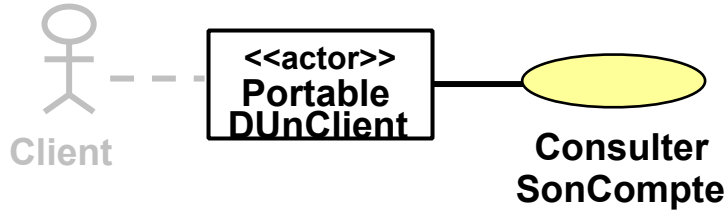
- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - Communication entre acteurs et système
- Problèmes récurrents
 - Cas d'utilisation orienté-solutions / cas d'utilisation essentiels
 - **Problèmes des intermédiaires**
 - Problèmes liés à la relation acteur - cas d'utilisation
 - Problème de granularité

Problème des intermédiaires

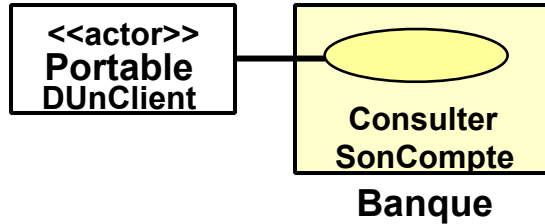
- Représentation des intermédiaires entre le système et l'intéressé ?
- **Différents points de vue**



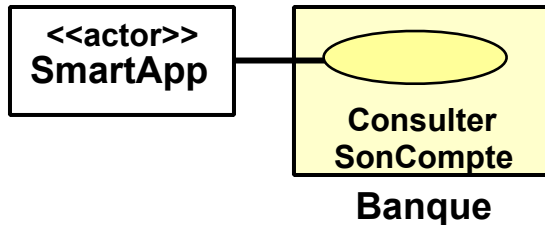
Problème des intermédiaires



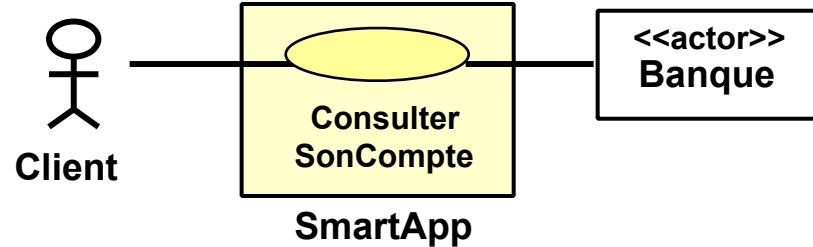
Ne pas confondre les systèmes



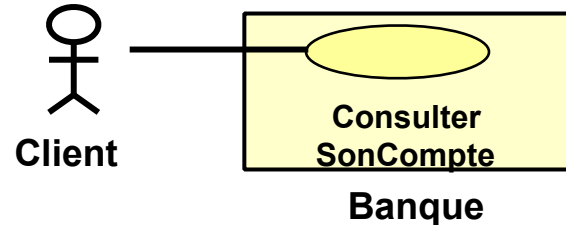
Projet: développer le système centralisé accessible à partir d'un portable



Projet: développer le système centralisé accessible à partir du système embarqué du téléphone



Projet: développer le système embarqué dans un portable pour accéder au système centralisé

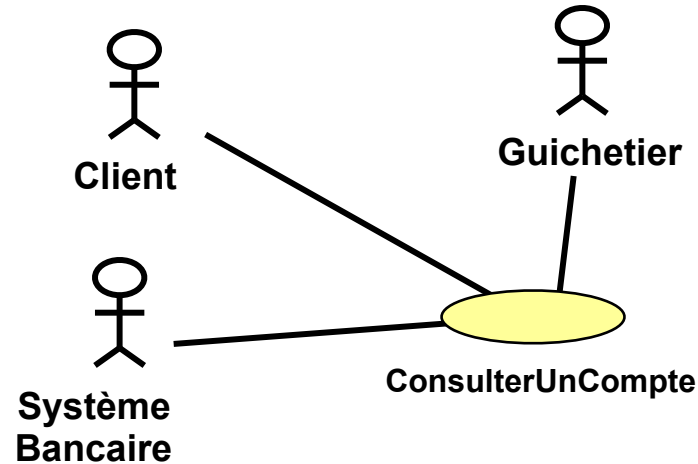


Projet: développer le système global

Sommaire

- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - Communication entre acteurs et système
- Problèmes récurrents
 - Cas d'utilisation orienté-solutions / cas d'utilisation essentiels
 - Problèmes des intermédiaires
 - **Problèmes liés à la relation acteur - cas d'utilisation**
 - Problème de granularité

Relation acteur - cas d'utilisation peu informative



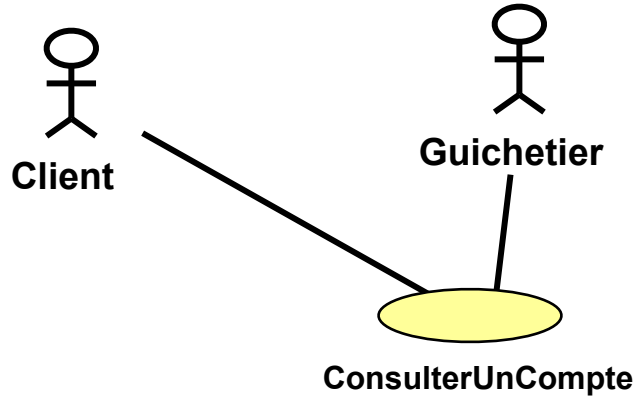
La relation acteur - cas d'utilisation est peu informative :

qui réalise le cas d'utilisation ? qui collabore à son déroulement ?

quels acteurs peuvent participer à un même scénario simultanément ?

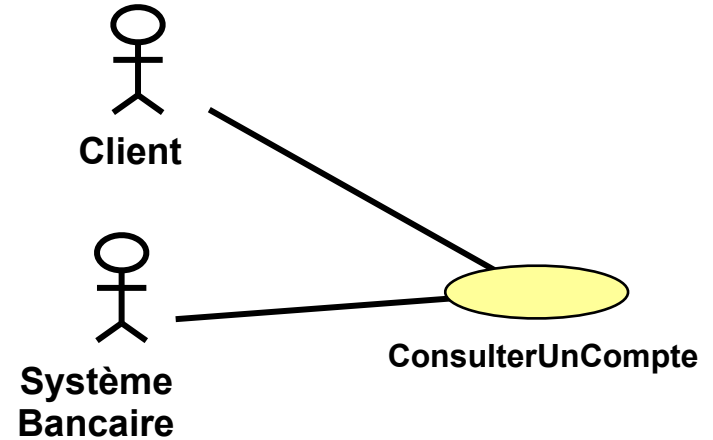
Pas de notation standard pour exprimer les réponses...

Deux interprétations possibles



(1) CAS D'UTILISATION "PARTAGE"

Deux acteurs peuvent réaliser le cas d'utilisation mais pour répondre à des objectifs qui leur sont propres



(2) CAS D'UTILISATION "COLLABORATIF"

Deux acteurs collaborent à la réalisation d'un objectif. Le système interagit avec les deux acteurs.

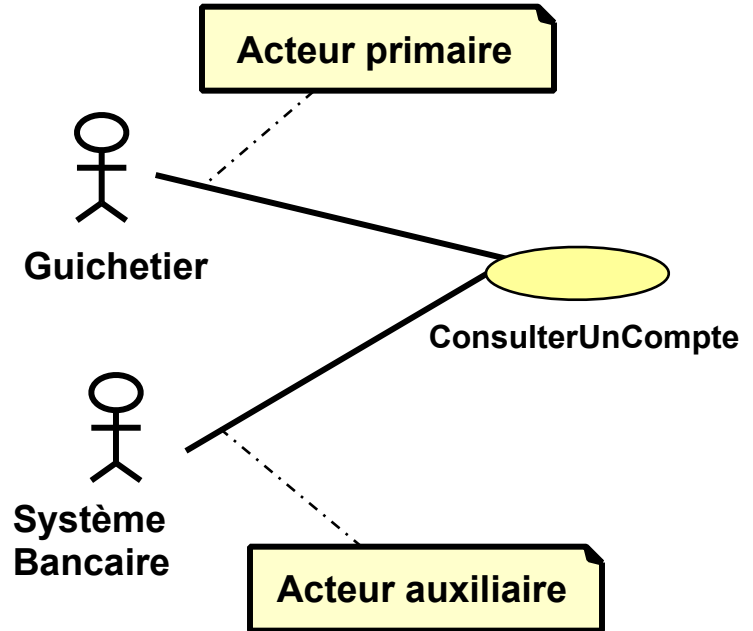
Problème des CU collaboratifs

Acteur "primaire"

- utilise le système comme outil pour réaliser son but
- initie généralement la communication

Acteur(s) "auxiliaire(s)"

- interviennent suite à l'intervention de l'acteur primaire
- offrent généralement leurs services au système

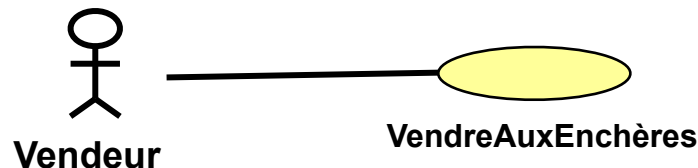
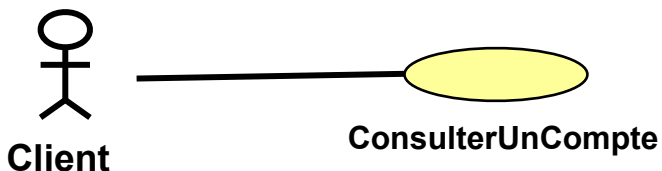


Différents styles dans la pratique

- **STYLE "primaire":**
Ne représenter que les acteurs primaires dans les diagrammes
- **STYLE "décoré":**
Utiliser une décoration particulière (e.g. auxiliaire ou initiator)
- **STYLE "gauche/droite":**
Positionner les acteurs primaires à gauche, secondaires à droite
- **STYLE "fléché":**
Utiliser une flèche pour indiquer l'acteur primaire (à éviter)

Style “primaire”

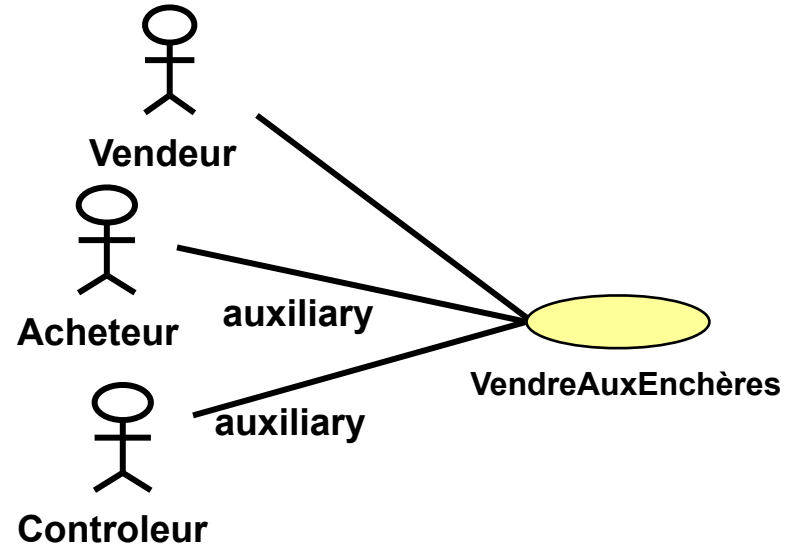
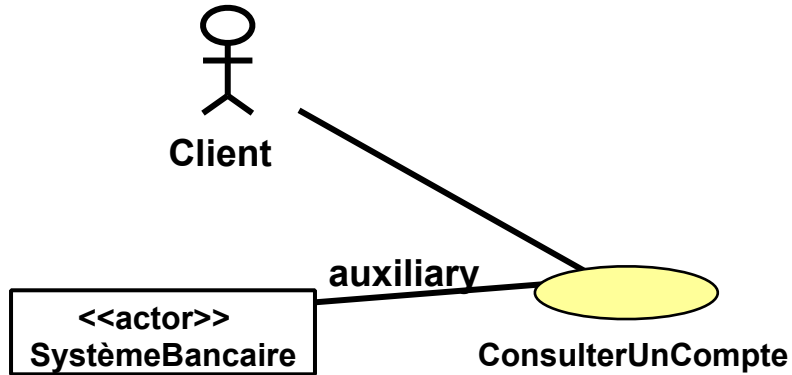
- Ne représenter que l'acteur primaire



- A le mérite d'être simple et d'obtenir des diagrammes lisibles
- Sans doute la meilleure solution lors des premières itérations

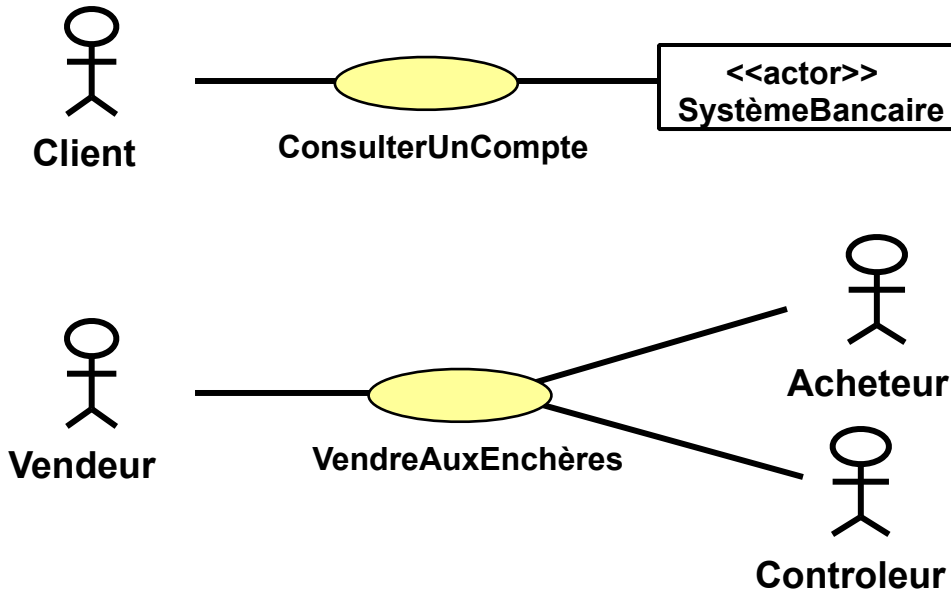
Style “décoré”

- Utiliser une décoration particulière (e.g. auxiliaire ou initiator)



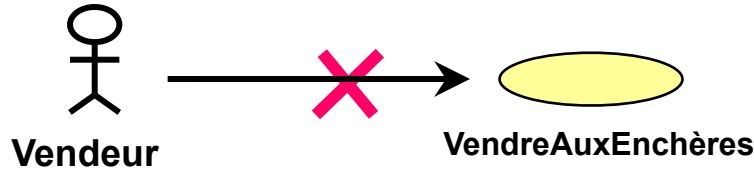
Style “droite/gauche”

- primaire à gauche, secondaire à droite



- convention "invisible" sans indication

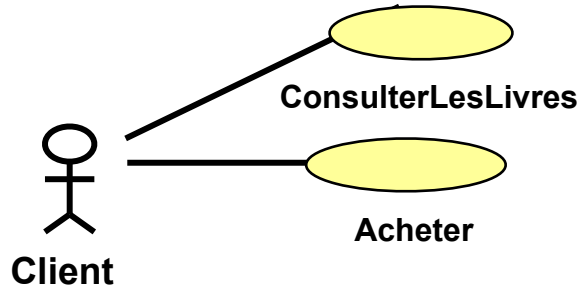
Ne pas mettre de flèche



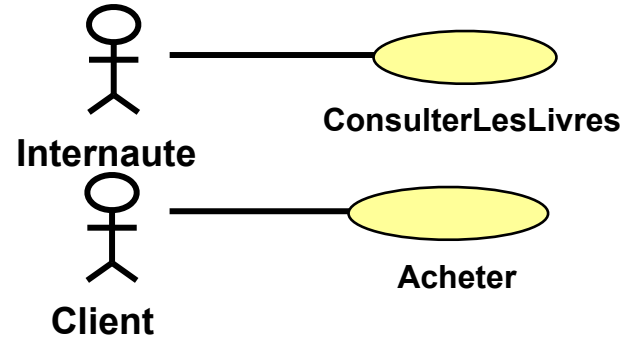
- Interprétation diverses et variées :
 - "l'acteur est initiateur"
 - "la communication se fait que dans un seul sens"
 - "je ne savais pas comment enlever la flèche avec cet outil UML..."

Problème des CU partagés

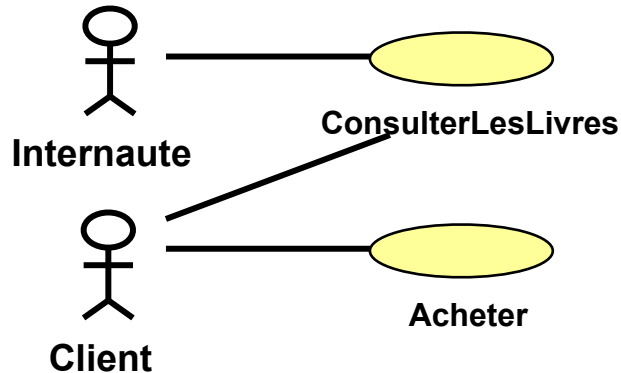
A



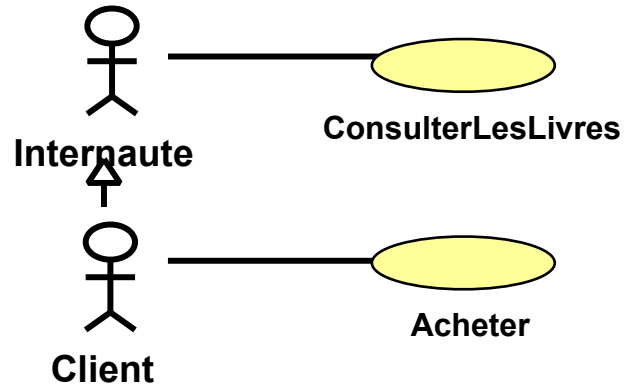
B



C

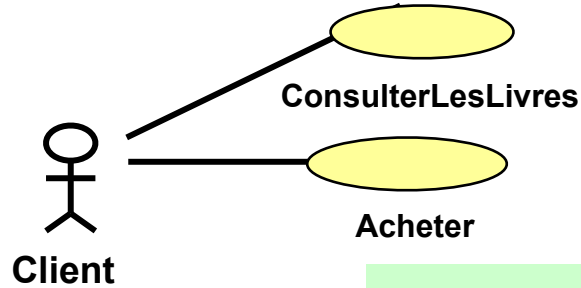


D

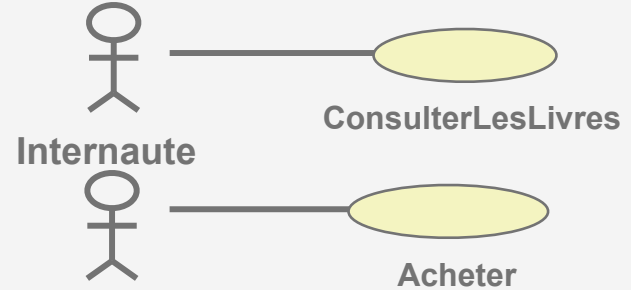


Problème des CU partagés

A



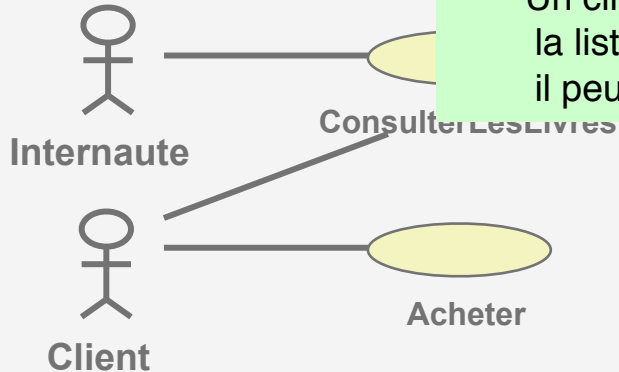
B



SYSTEME DE VENTE EN LIGNE

Un client peut consulter
la liste des livres et
il peut en acheter

C



Internaute

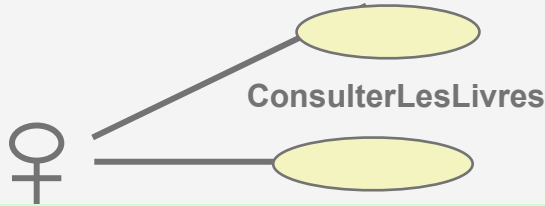
Client

ConsulterLesLivres

Acheter

Problème des CU partagés

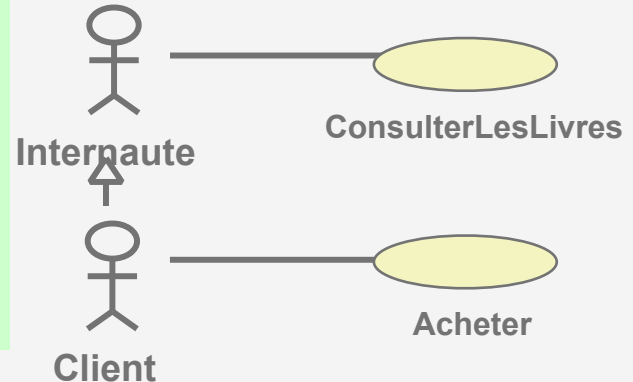
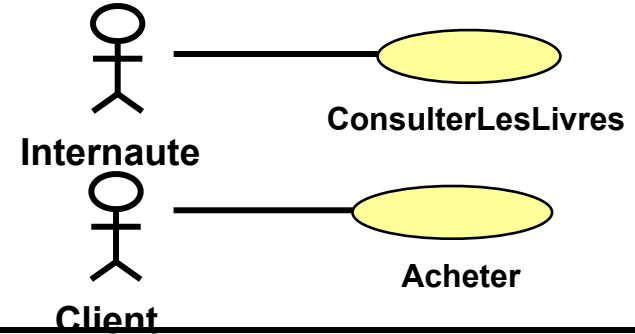
A



- On insiste sur le fait que l'une des fonctions importantes est d'accueillir des internautes quelconques et de leur permettre de consulter la liste des livres sans que leur objectif soit d'acheter
- La différence est faite entre un internaute et un client (potentiellement habitué)
- Une personne peut changer de rôle dynamiquement en jouant le rôle internaute puis de client.
- Ce changement de rôle est une caractéristique extérieure au système

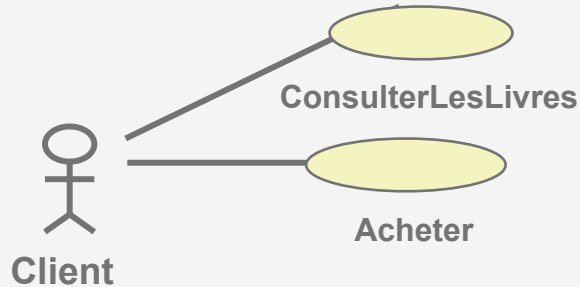
Client

B

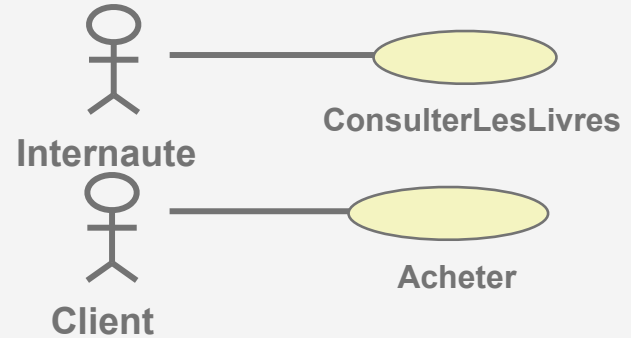


Problème des CU partagés

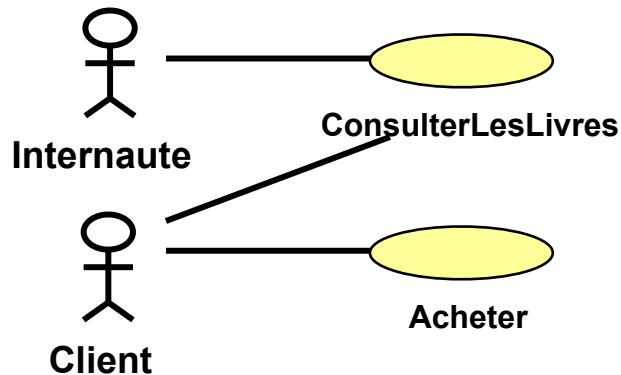
A



B



C

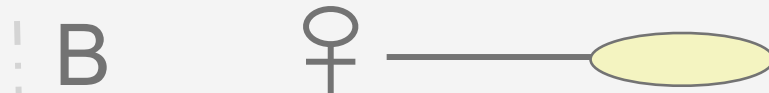
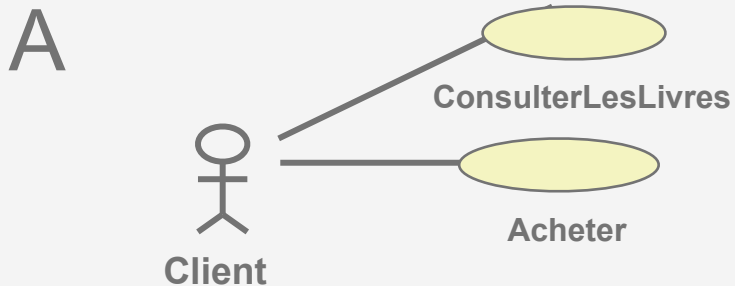


- Il est considéré comme important de séparer les clients des internautes
- ConsulterLesLivres est un cas d'utilisation normal pour un client
- Acheter aussi

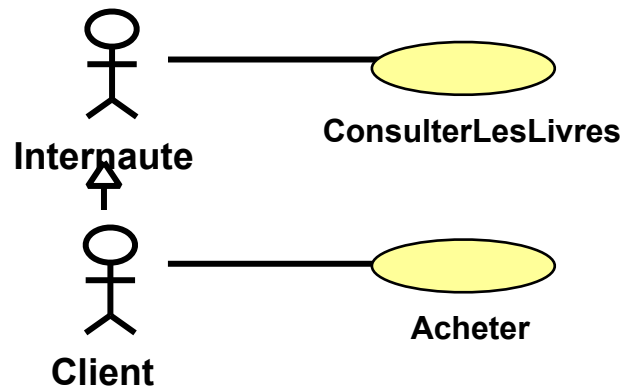
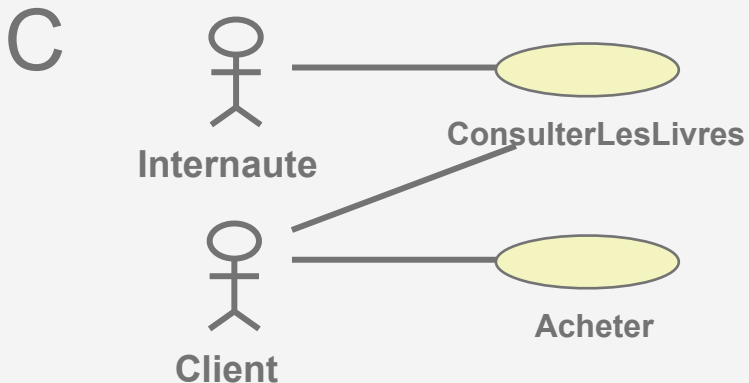
Client

Acheter

Problème des CU partagés



- Un client peut faire tout ce que peut faire un internaute (héritage des cas d'utilisation)
- Un client est un cas particulier d'internaute (spécialisation)



Sommaire

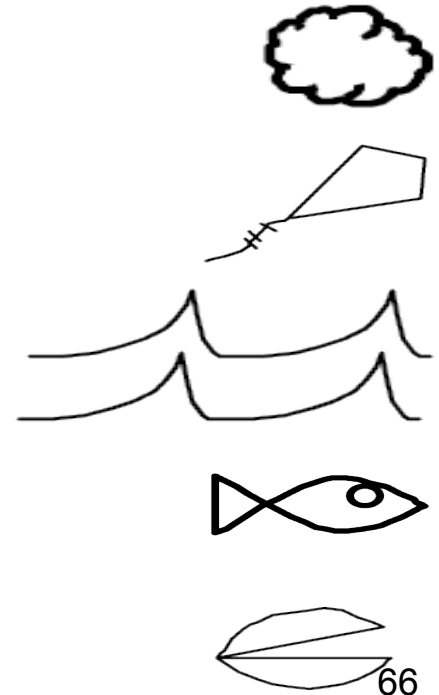
- Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Éléments de base
 - Acteurs, cas d'utilisation et système
 - Relations entre éléments de base
 - Communication entre acteurs et système
- Problèmes récurrents
 - Cas d'utilisation orienté-solutions / cas d'utilisation essentiels
 - Problèmes des intermédiaires
 - Problèmes liés à la relation acteur - cas d'utilisation
 - **Problème de granularité**

Problème de granularité

- A quel niveau décrire les cas d'utilisation ?
- Bonne question ... mais pas de réponse
- Trop haut
 - trop loin du système
 - trop abstrait et "flou"
 - trop complexe à décrire
- Trop bas
 - trop de cas d'utilisation
 - trop près de l'interface
 - trop loin des besoins métiers
- Conclusion: choisir le "bon" niveau ...

Granularité des cas d'utilisation

- Tous les cas d'utilisations ne sont pas tous au même niveau
 - Différents niveaux de détails
 - POURQUOI vs. COMMENT
 - Décoration du niveau selon *Cockburn*
 - Non standardisé mais intuitif et utile
-
- Niveau normal des cas d'utilisation : la mer



Niveau de granularité

Clouds
Level



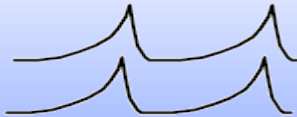
- Trop haut

Kite
Level



- Niveau résumé : décrit un regroupement correspondant à un objectif plus global

Sea
Level



- **Niveau normal** : décrit un but de l'acteur qu'il peut atteindre via une interaction avec système

Fish
Level



- Niveau détaillé : décrit une interaction le système, pas un but en soi

Clam
Level



- Trop bas

Exemple

