UML :

DIAGRAMME

DES

CAS D'UTILISATION

Centré sur les utilisateurs !

- Ce n'est pas du code !

- Buts : - définir le système du point de vue des utilisateurs

  - définir les limites précises du système (délimiter le système)

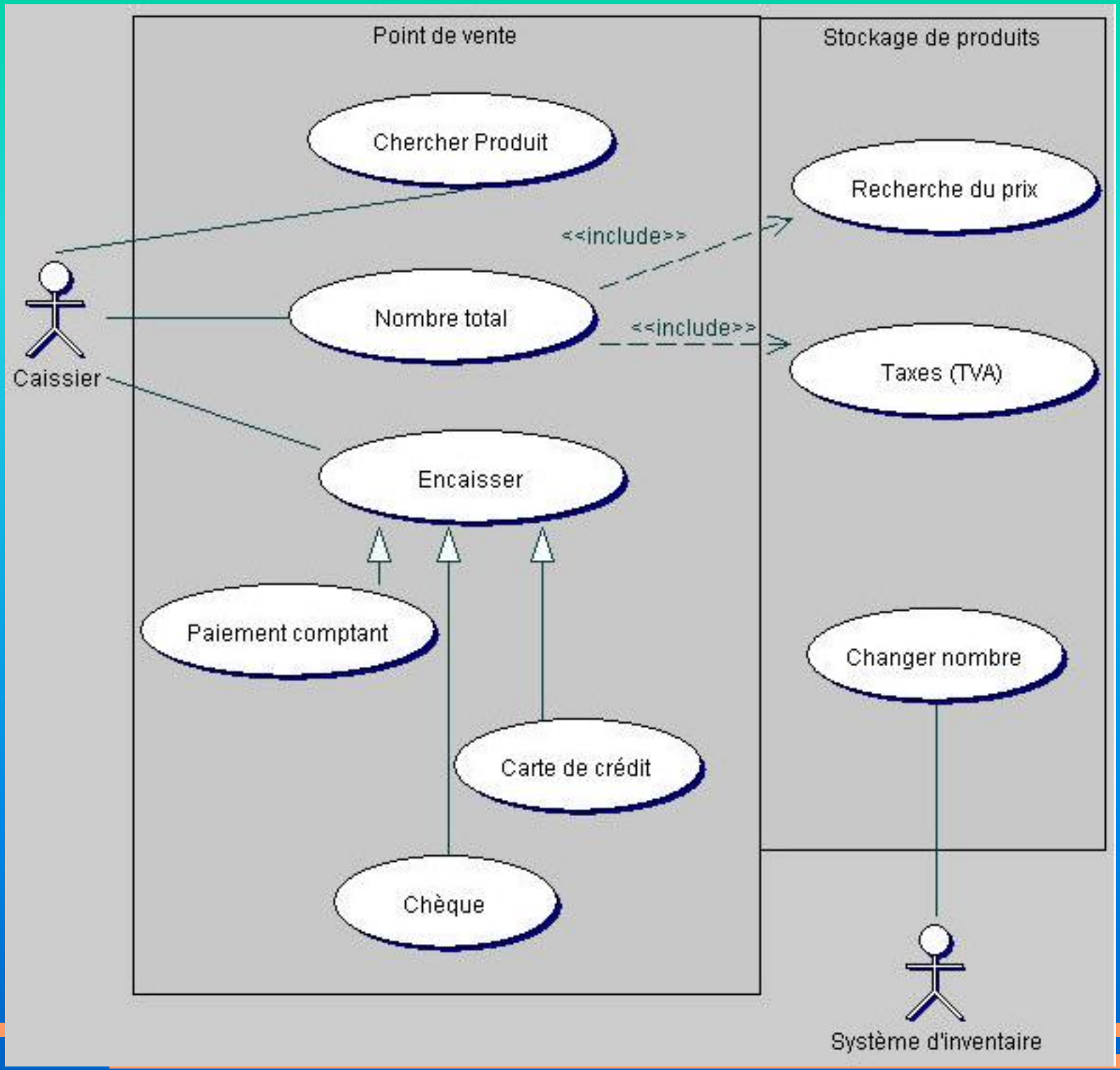
  - donne une vue globale du système (comprendre le système)

  - être sûr que le système développé correspond aux besoins de l'utilisateur final

- Notation très simple, compréhensible par tous **(il ne faut pas être informaticien !!!)**

- Permet de structurer : - les besoins (cahier des charges)

  - le reste du développement


Point de vente
Stockage de produits
Chercher Produit
Recherche du prix
<<include>>
Nombre total
<<include>>
Taxes (TVA)
Caissier
Encaisser
Paiement comptant
Changer nombre
Carte de crédit
Chèque
Système d'inventaire

- Il débute l'analyse d'un système. Considérer l'organisation comme un **système**.
- Comprendre ce que fait l'organisation
- Ce modèle permet de recueillir, d'analyser et d'organiser les **besoins** lors du développement ou l'amélioration d'un système (reconfigurer l'organisation)
- Question que se pose le maître d'œuvre durant le recueil des besoins : Ai-je toutes les connaissances et les informations pour définir ce que doit faire le système ?
- On se place clairement du coté des **utilisateurs** et pas sur le plan technique!
- Long et fastidieux!
- Exhaustif mais rester au niveau des grandes fonctions du système
- Pas de notion temporel

- **Méthodologie : il convient d'identifier pour chaque acteur ses différentes intentions d'utilisation du système**

- **Il est utilisé pour modéliser la vue statique des cas d'utilisation d'un système**

- **Il est particulièrement important pour l'organisation et la modélisation des comportements d'un système (mais pas comment réaliser ces comportements !!!).**

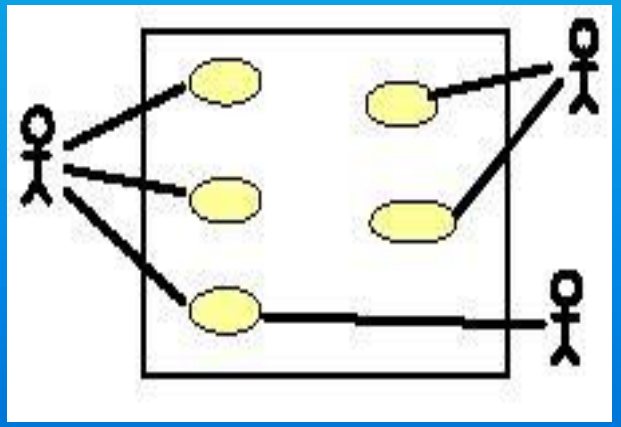
- **Outil pour COMMUNIQUER (utilisateurs/experts du domaines vs. concepteurs/développeurs).**

Le diagramme de cas d'utilisation

contient :

- Des cas d’utilisation 
- Des acteurs 

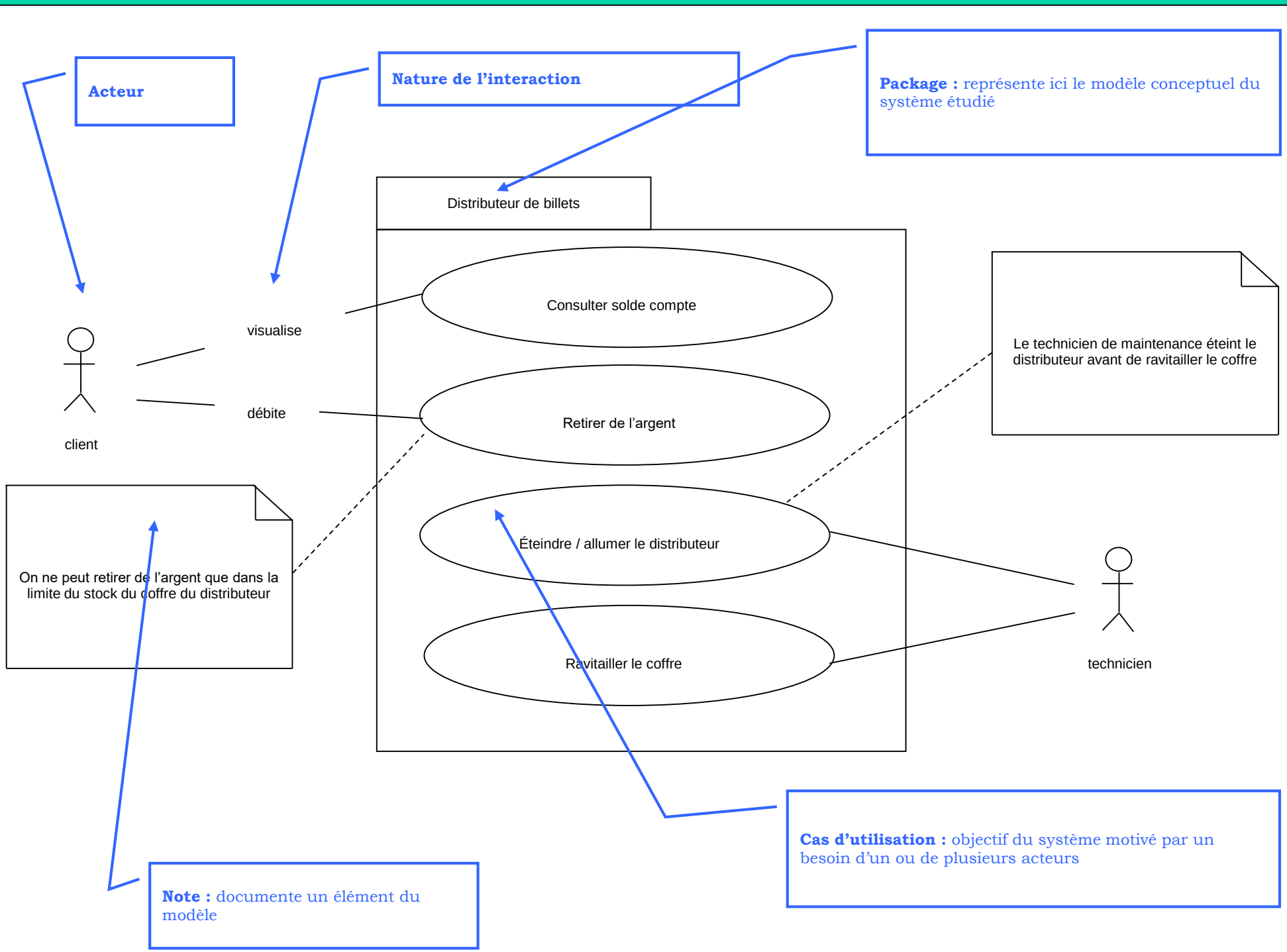
- Le système 
- Des relations de :
  - Dépendances
  - Généralisation
  - Association



- **Il peut aussi contenir, comme tous les autres diagrammes :**
  - **Des notes ou commentaires**
  - **Des contraintes**
  - **Des objets libres**

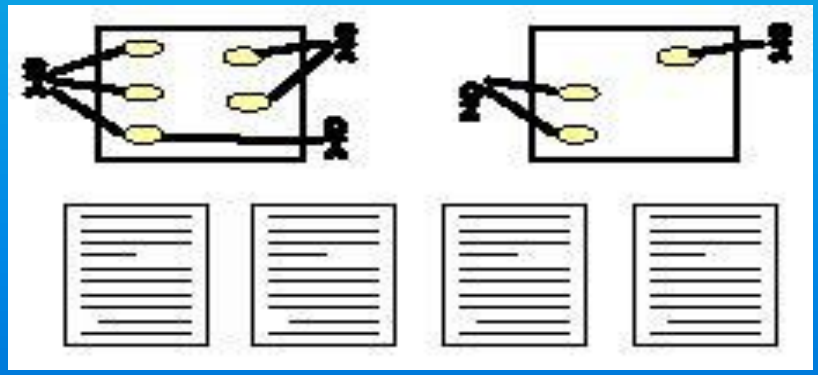
- **Pour modéliser le comportement d'un système on doit :**
  - **Identifier les acteurs**
  - **Identifier les cas d'utilisation**
  - **Construire et décrire les cas d'utilisation**
  - **Organiser et structurer les cas d'utilisation**

- **Cas d'utilisation → Interaction → Suite d'actions. Il faudra décrire cette suite d'actions.**

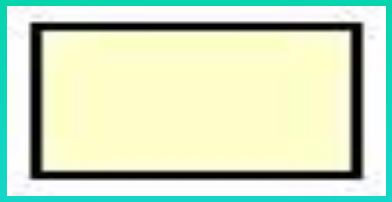

Acteur
Nature de l'interaction
Package : représente ici le modèle conceptuel du système étudié
Distributeur de billets
Consulter solde compte
visualise
client
débite
Retirer de l'argent
Le technicien de maintenance éteint le distributeur avant de ravitailler le coffre
On ne peut retirer de l'argent que dans la limite du stock du coffre du distributeur
Éteindre / allumer le distributeur
technicien
Ravitailler le coffre
Cas d'utilisation : objectif du système motivé par un besoin d'un ou de plusieurs acteurs
Note : documente un élément du modèle

Un modèle de cas d'utilisation

- peut être formé :
  - de plusieurs diagrammes de cas d'utilisation
  - de descriptions textuelles
  - de diagrammes de séquences



Le système



- **Le système est un ensemble de cas d'utilisation qui décrit ce que doit faire le système!**

- **Le système ne contient pas les acteurs (un acteur existe en dehors du système).**

- **Un modèle de cas d'utilisation permet de définir :**
  - **les fonctions essentielles du système**
  - **les limites du système**
  - **le système par rapport à son environnement**

Les acteurs

- **Elément *externe* qui *interagit* avec le système (via un ou plusieurs cas d'utilisation). Correspond à un rôle générique!**
- **Un acteur représente un ensemble cohérent de rôles joués par des entités externes (utilisateur humain, dispositif matériel, …) qui interagissent avec le système étudié**
- **Un acteur prend des décisions, des initiatives, il est « actif »**
- **En règle générale, un acteur représente un rôle joué avec le système, par un homme, une machine ou un autre système : un client, un système informatique externe d'authentification, ...**
- **Une interaction entre l'acteur et le système doit produire un résultat tangible pour l'acteur. Une interaction = une séquence d'actions.**

Différents types d'acteurs

- **Acteurs principaux**. Un cas d'utilisation a obligatoirement un et un seul acteur principal. Ex: client, guichetier
- **Acteurs secondaires**. Facultatifs.

  Ex: contrôleur, directeur, ingénieur système
- **Périphériques externes.** Ex: un capteur, une horloge externe, ...
- **Systèmes externes.** Ex: système interbancaire

Si une authentification fait appel à un système externe, celui-ci est un acteur secondaire.

Si l'authentification fait appel à la B.D. du système, c'est une fonctionnalité du système.

Guichet automatique d'une banque (Mister Cash, Bancontact).

Identifier d'abord les acteurs !

« Le guichet offre différents services :

- **Retrait avec visa ou CB**
- **Consultation solde, dépôt argent et chèque uniquement pour les porteurs CB**
- **Des transactions sécurisées**
- **Recharger le distributeur,… »**

« Actor »
Système autorisation Visa

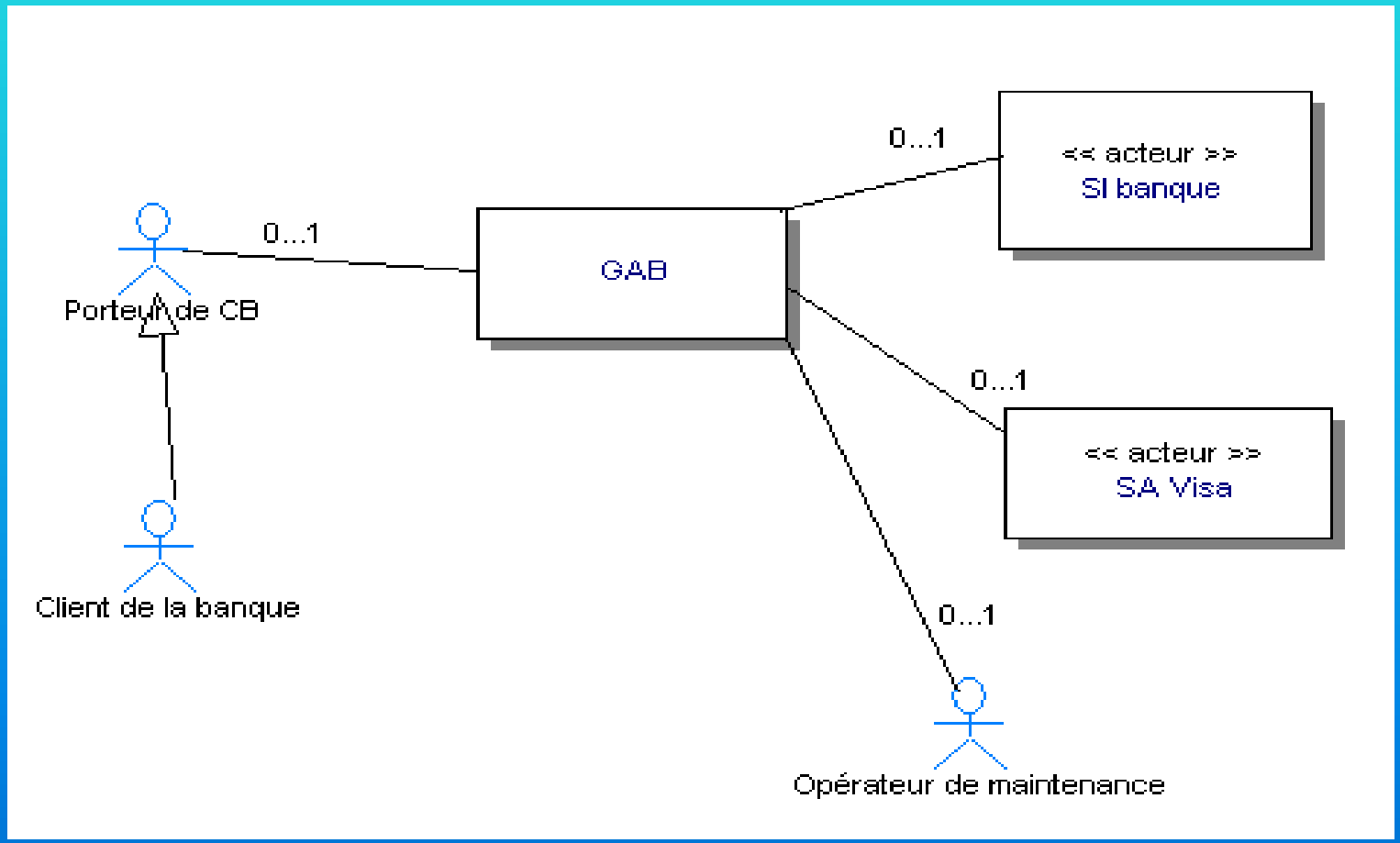
Client

Porteur
CB

Syst. Info. Banque

En général, les acteurs humains sont représentés par le « stick man » et les rectangles pour les systèmes

Diagramme de contexte statique du guichet automatique de banque (le système est au centre dans un rectangle)



- **Identification des acteurs principaux**
  - **Un acteur est dit « principal » pour un cas d'utilisation lorsque le cas d'utilisation rend service à cet acteur. Les autres acteurs sont dits « secondaires »**
  - **Ce sont les utilisateurs du système**
  - **Leur nom est pensé en fonction de leur rôle**
  - **Plusieurs personnes peuvent avoir le même rôle**
  - **Une même personne peut jouer des rôles différents vis-à-vis du systèmes et donc correspondre à plusieurs acteurs**

- **Identification des acteurs secondaires**
  - **Acteurs sollicités pour des informations complémentaires**
  - **Exemples : un système informatique externe au système interagissant avec lui, des logiciels déjà disponibles à intégrer dans le projet,…**
  - **Ils sont généralement disposés à droite sur les diagrammes**

Cas d’utilisation (CU)

- **Est une manière d’utiliser le système.**
- **Il convient donc d’identifier pour chaque acteur ses différentes intentions d’utilisation du système.**
- **Un cas d’utilisation représente un ensemble de séquences d’actions réalisées par la système, qui produisent un résultat observable intéressant pour un acteur particulier.**
- **Correspond à une fonction visible par l’utilisateur.**
- **Permet d’atteindre un but pour un utilisateur. Doit être utile en soi.**

- Se placer du point de vue de chaque acteur et déterminer dans quels cas il utilise le système et à quelles fonctionnalités il doit avoir accès.

- Limiter le nombre de cas en se situant à un bon niveau d'abstraction (ne pas réduire un cas à une action) .

- Chaque cas d'utilisation qui interagit avec l'extérieur possède, à un moment donné, au maximum un acteur principal. Il peut posséder plusieurs acteurs secondaires qui interagissent en même temps sur le même cas.

- On peut donner un sens (direction) à l'interaction entre un acteur et un cas d'utilisation (→ , ←, ←→).

- Différents acteurs peuvent utiliser le même cas de manière indépendante

- Un cas non relié à un acteur est un cas d'utilisation interne !

- **Porteur de CB**
  - **Retirer de l’argent**

- **Client de la banque**
  - **Retirer de l’argent**
  - **Consulter le solde d’un ou plusieurs comptes**
  - **Déposer du numéraire**
  - **Déposer des chèques**

- **Opérateur de maintenance**
  - **Recharger le distributeur**
  - **Récupérer les cartes avalées**
  - **Récupérer les chèques déposés**

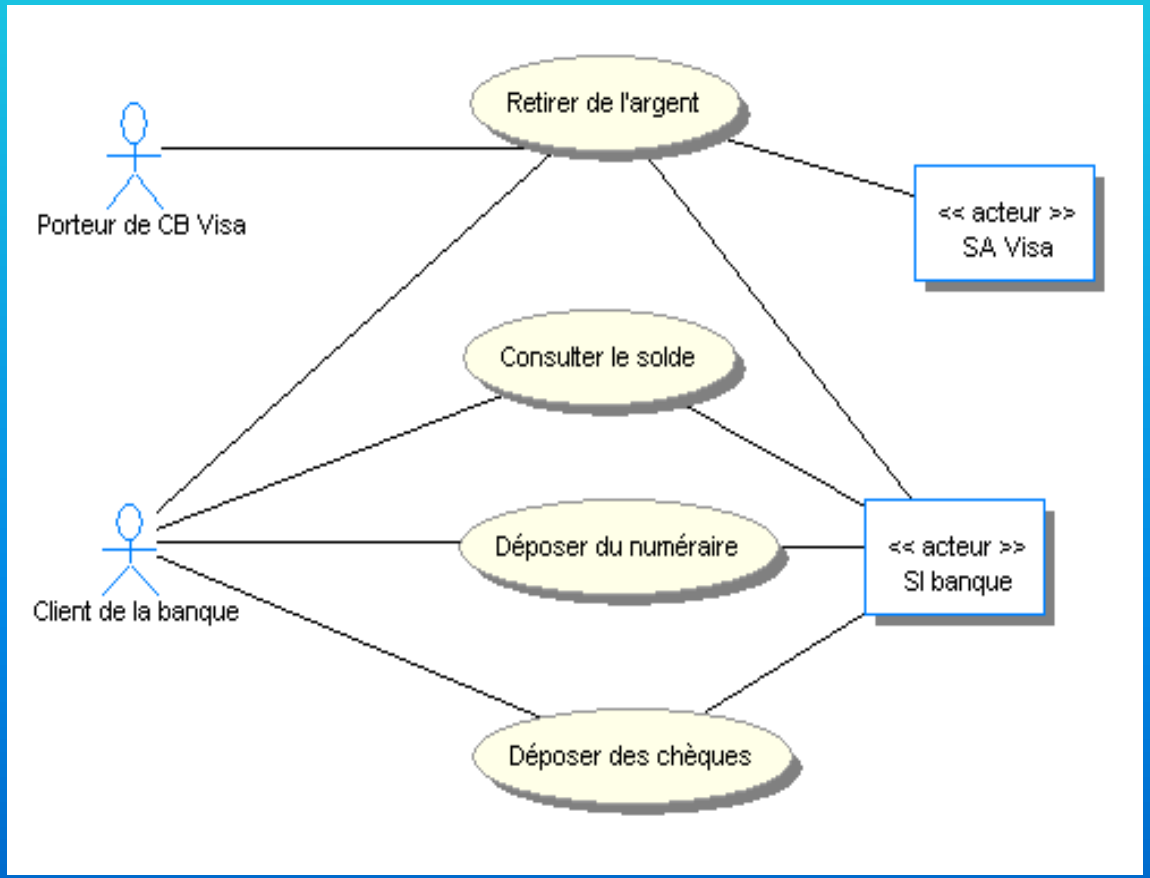
- **Système d’autorisation Visa (acteur secondaire)**
  - **Néant**

- **Système d’information banque (acteur secondaire)**
  - **Néant**

Une fois identifié les cas d'utilisation sont répertoriés dans un diagramme préliminaire (seulement les acteurs principaux) où les cas d'utilisation sont représentés par des ovales à l'intérieur du système (rectangle)

Il reste à ajouter les acteurs secondaires :

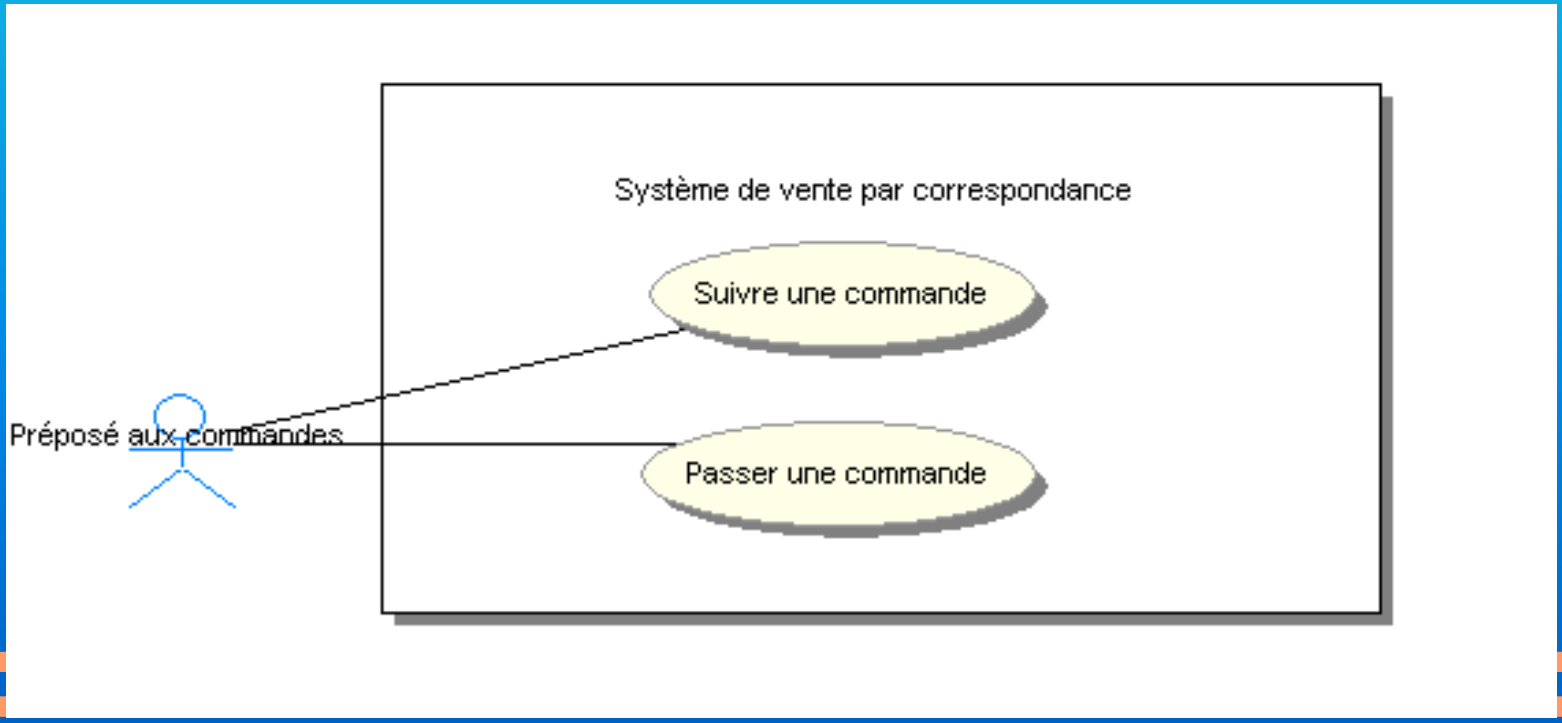
UML propose d'ajouter simplement des associations des acteurs secondaires (les dessiner à droite du diagramme) vers les cas d'utilisation existants



Les acteurs sont connectés aux cas d'utilisation uniquement par une **association** et mettent en évidence toutes les interactions possibles entre le système et le monde extérieur

Représente une communication initiée par l'acteur. Echange de de messages possible dans les 2 sens.

Le système à modéliser apparaît dans un cadre et l'ensemble des cas d'utilisation contenus dans le cadre constitue un sujet



Relations entre cas d'utilisation

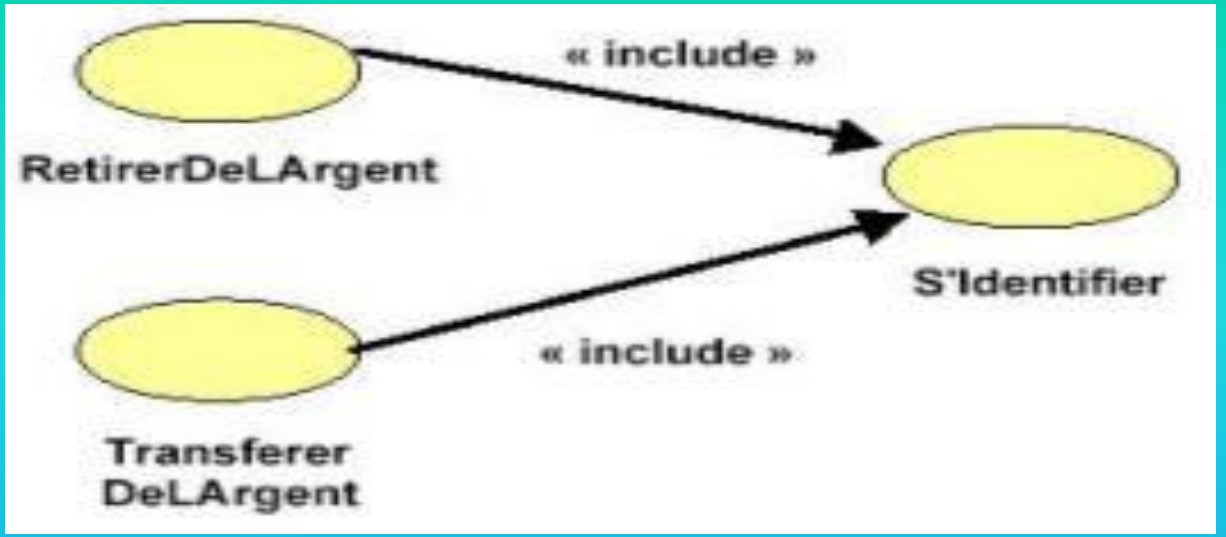
- **Il existe deux types de relation :**
  - **Les dépendances stéréotypées : leur portée est explicitée par la valeur du stéréotype :**
    - **Les stéréotypes les plus utilisés étant :**
      - **L'inclusion**
      - **L'extension**
  - **La généralisation/spécialisation**

Une dépendance est une relation d'utilisation qui indique qu'un changement de spécification d'un élément peut affecter un autre élément qui l'utilise, mais que l'inverse n'est pas nécessairement vrai.

La dépendance a pour représentation graphique une flèche en pointillés, dont l'origine est rattachée à l'élément dépendant. On l'utilise quand il est nécessaire de montrer qu'un élément se sert d'un autre.

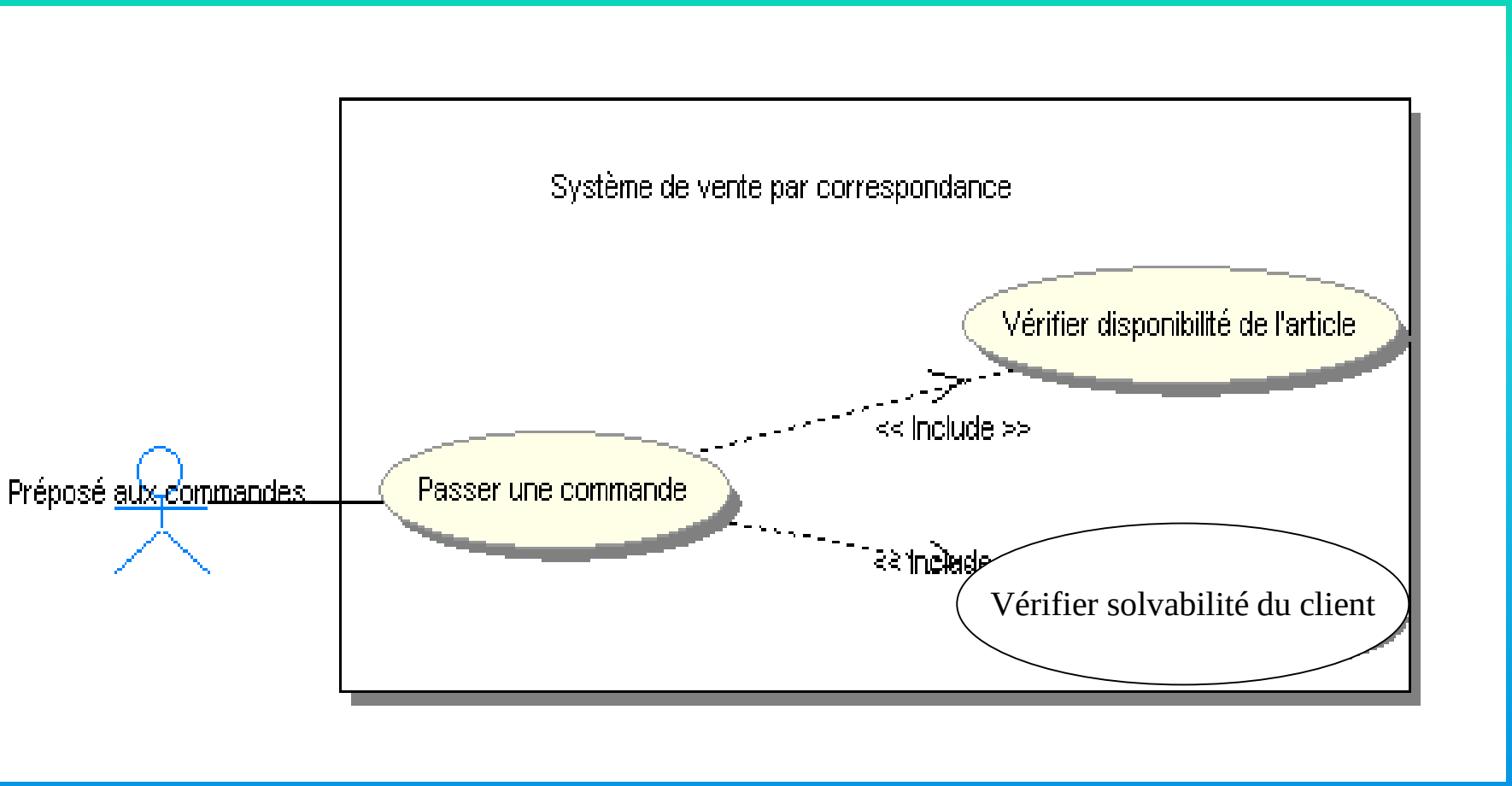
La relation d'inclusion (« Include », « Inclut »)

Un cas A est inclus dans un cas B si le comportement décrit par le cas A est inclus dans le comportement du cas B (B dépend de A)



- **Lorsqu'un CU a besoin de l'aide d'un autre**

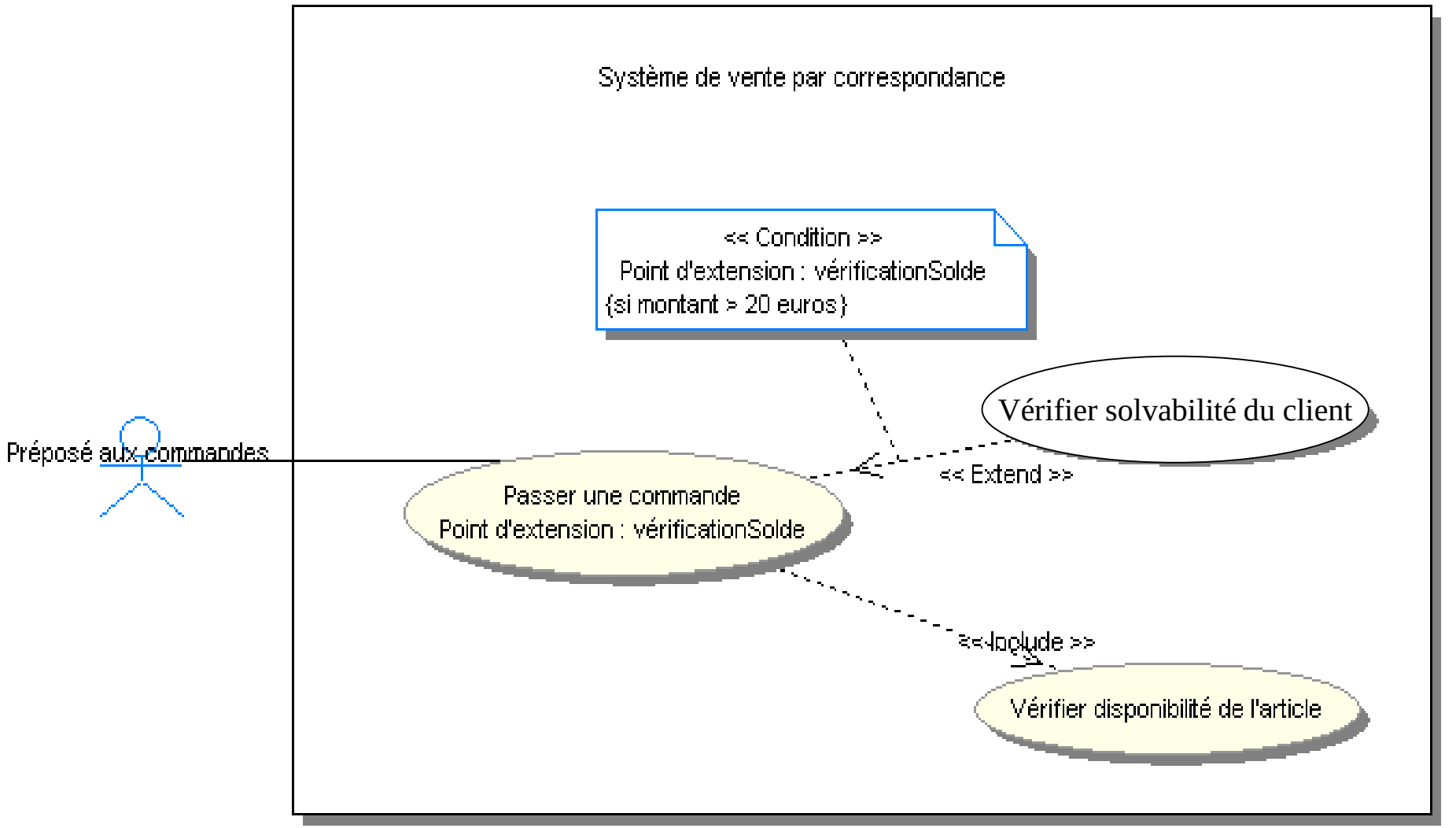
- **Permet d'isoler une tâche afin qu'elle soit réutilisée**


Système de vente par correspondance
Vérifier disponibilité de l'article
<< Include >>
Préposé aux commandes
Passer une commande
Vérifier solvabilité du client

La relation d'extension

- Si le comportement de B peut être étendu par le comportement de A, A étend B **(pas une relation « une sorte de » !!!)**
- Une extension est souvent soumise à condition (contrainte indiquant le moment où l'extension apparaît)
- Graphiquement la condition est exprimée sous la forme d'une note
- Le point d'extension
  - Apparition de l'extension à un point précis du cas étendu
  - Il porte un nom figurant dans un compartiment du cas étendu sous la rubrique « point d'extension »

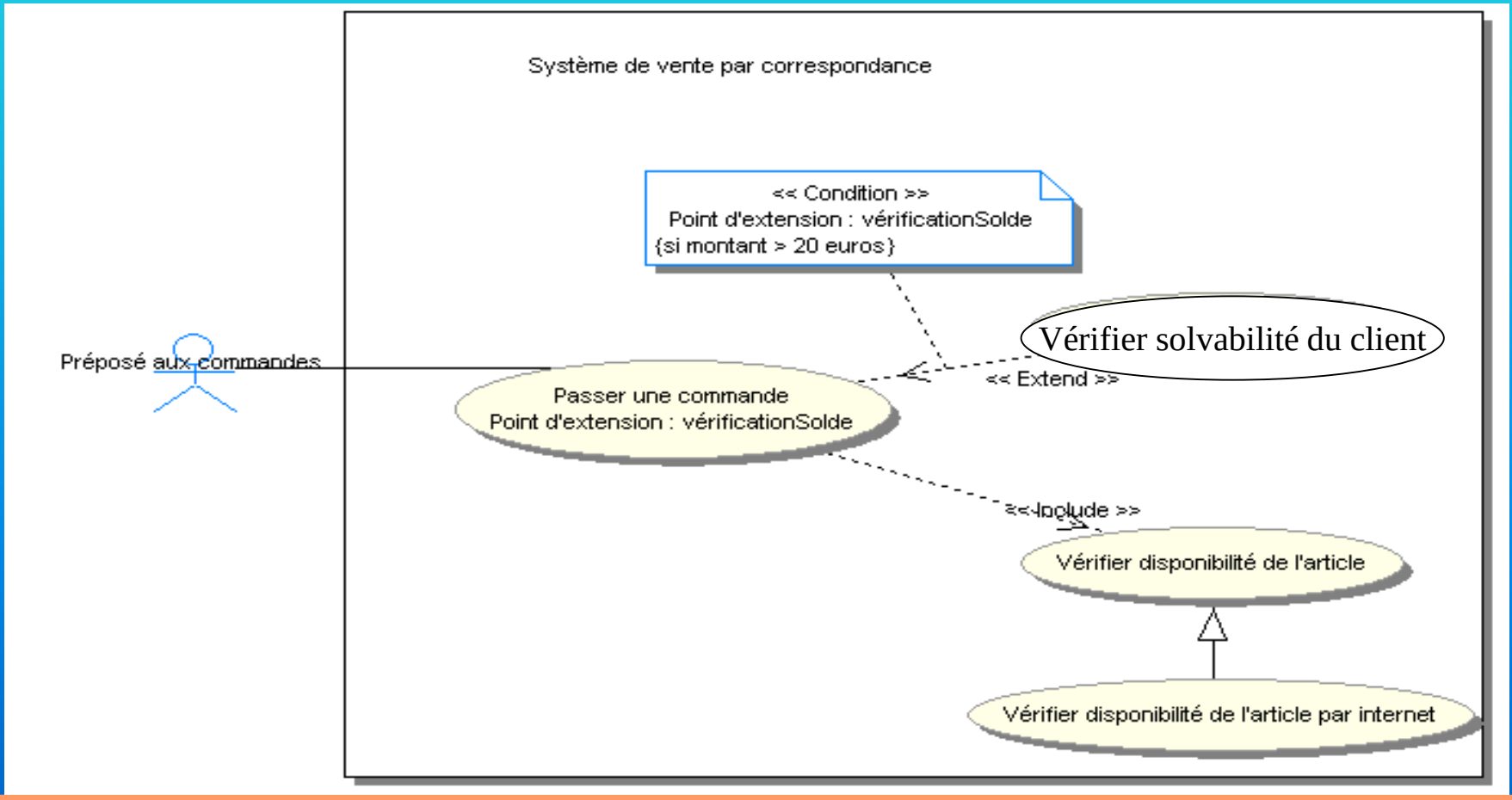
Indique qu'un cas pourrait avoir besoin de l'aide d'un autre



NB: si le cas A inclut B on trace la flèche de A vers B, mais si B étend A, la flèche est dirigée de B vers A (Vérifier étend le comportement Passer)

La relation de généralisation

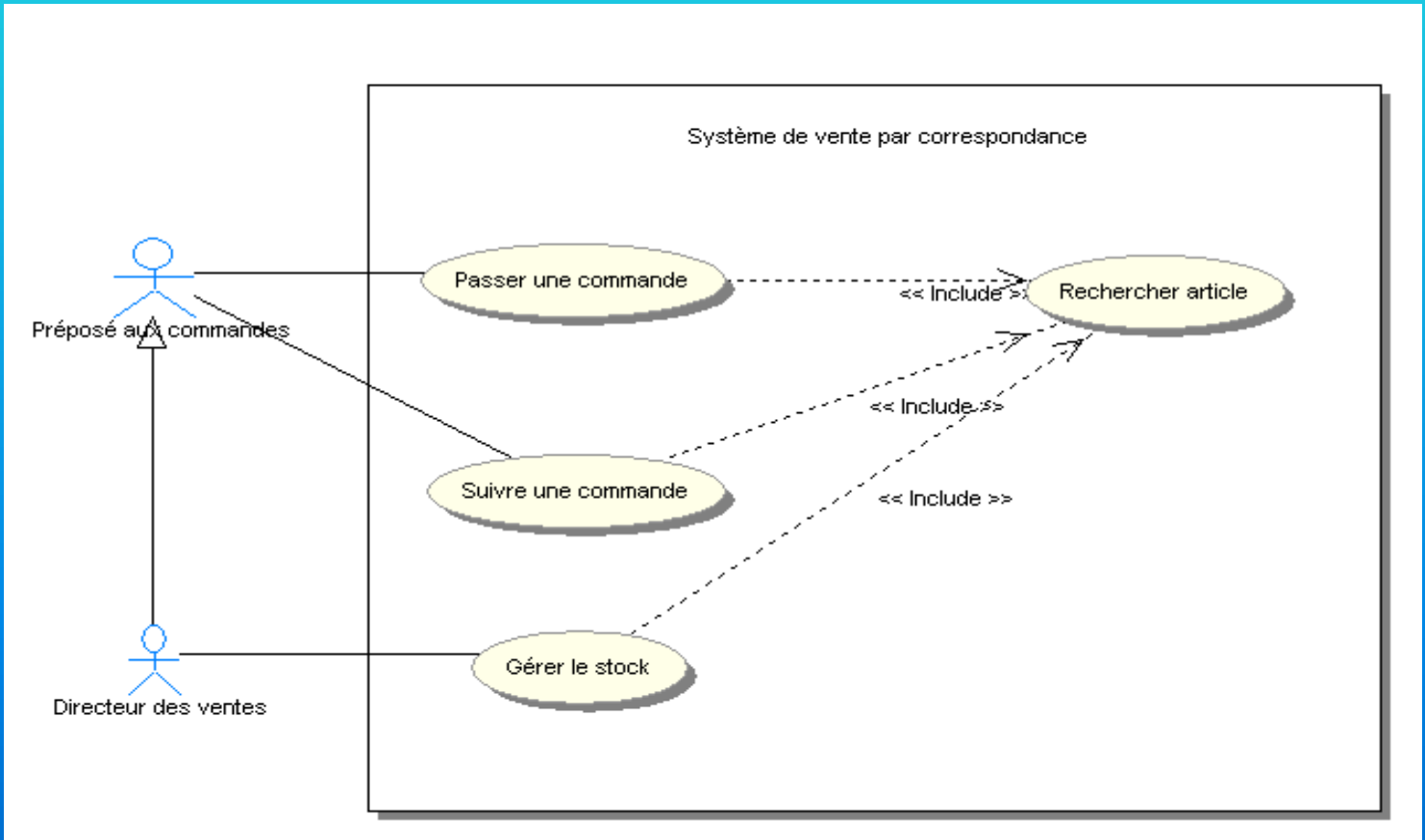
- Un cas A est une généralisation d'un cas B si B est un cas particulier de A (relation une sorte de)
- Concept d'héritage dans les langages orientés objet
- La flèche pointe vers le cas le plus général



Relations entre acteurs

La seule relation possible entre deux acteurs est la généralisation

Un acteur A est une généralisation d'un acteur B si l'acteur A peut être substitué par l'acteur B (une sorte de).



Description des cas d'utilisation

Le diagramme de cas d'utilisation n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation, il décrit seulement les grandes fonctions du système.

On décrit la dynamique du cas d'utilisation par (on parle de spécifications fonctionnelles) :

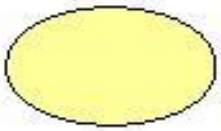
- **Un diagramme de séquence**
- **Une description textuelle des diagrammes d'utilisation**

NB : UML n'impose pas de façon de décrire un cas d'utilisation, le modélisateur peut utiliser le type de diagramme qui paraît le plus adapté pour représenter son problème, par exemple un diagramme d'activité ou tout autre croquis qu'il estime utile !

Pour chaque cas d'utilisation :

- choisir un identificateur représentatif
- donner une description textuelle simple
- la fonction réalisée doit être comprise de tous
- pas trop de détails
- préciser ce que fait le système, ce que fait l'acteur

Les cas d'utilisation ne doivent pas se chevaucher



Retirer
DeLArgent
AuDistributeur

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

- le *client* insère sa carte bancaire dans le distributeur
- le *système* demande le code pour l 'identifier
- le *client* choisi le montant du retrait
- le *système* vérifie qu 'il y a suffisamment d 'argent
- si c 'est le cas, le *système* distribue les billets et retire l 'argent du compte du client
- le *client* prend les billets et retire sa carte

Description textuelle des cas d'utilisation (recommandée mais non obligatoire)

Un cas d’utilisation doit avoir un début et une fin clairement identifiés (délimite le scénario).

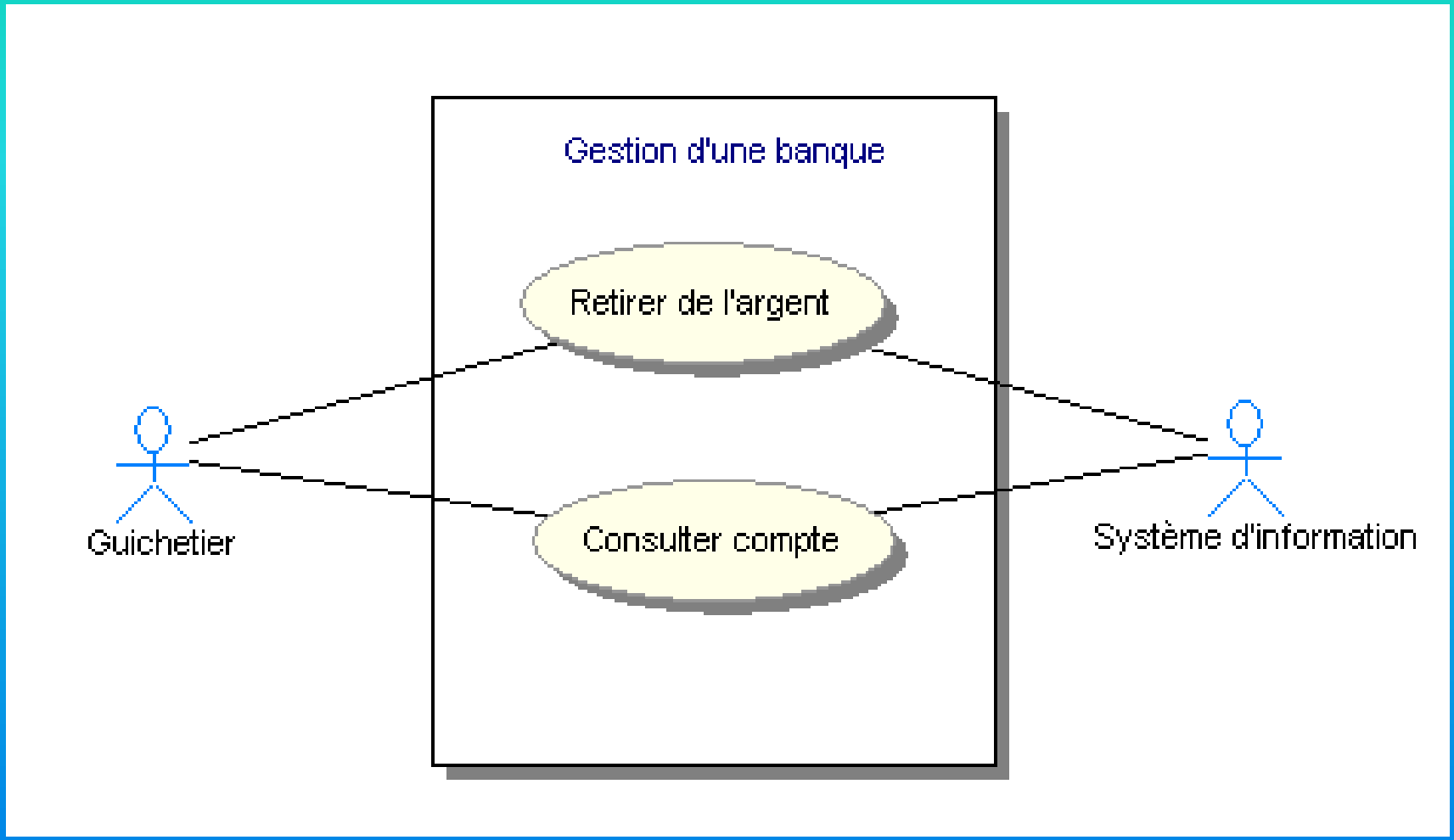
On va recenser de manière informelle toutes les interactions entre acteurs et système. Il faut préciser les variantes : les séquences nominales, alternatives et d’exceptions.

On essaye d’exprimer les enchaînements de messages de manière séquentielle.

Un message est un élément de communication entre objets déclenchant une activité.

- **1 Sommaire d'identification**
  - Titre du cas
  - Résumé de son objectif
  - Acteurs principaux, Acteurs secondaires
  - Date de création et de mise à jour de la description courante
  - Nom des responsables
  - Numéro de version
- **2 Description du fonctionnement du cas** = Séquencement de messages
  - **Séquence nominale** (=scénario) :
    - **Pré-conditions** : état dans lequel se trouve le système avant le déroulement de la séquence
    - **Enchaînements de messages**
    - **Post-conditions** : indication de l'état du système après la description de la séquence nominale
  - **Séquences alternatives** (embranchement d'une séquence nominale avec retour à la séquence nominale)
  - **Séquences d'exception** (interruption du séquencement nominal sans retour)
- **3 Rubriques optionnelles**
  - Contraintes non fonctionnelles (souvent des spécifications techniques)
  - Contraintes liées à l'interface homme-machine

Exemple. Description textuelle du cas « Retirer de l'argent »:



1 Identification

Nom du cas : Retrait d’espèces en euros
But : 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 d’information
Date : le 01/06/2014
Responsable : M. Pierre
Version : 1.0

2 Séquencement

Le cas d'utilisation commence lorsqu'un client demande le retrait d'espèces en euros

Pré-conditions

Le client possède un compte

Séquence nominale

1 Le guichetier saisit le numéro de compte client
2 L'application valide le compte auprès du système d'information
3 L'application demande le type d'opération au guichetier
4 Le guichetier sélectionne un retrait d'espèces de 200 euros
5 L'application demande au système d'information de débiter le compte
6 Le système notifie au guichetier qu'il peut délivrer le montant demandé

Post-conditions

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

3 Rubriques optionnelles

Contraintes non fonctionnelles

Fiabilité : les accès doivent être extrêmement sûrs et sécurisés.

Confidentialité : les informations concernant le client ne doivent pas être divulguées.

Contraintes liées à l'interface homme-machine

Donner la possibilité d'accéder aux autres comptes du client

Toujours demander la validation des opérations de retrait

N.B. Parfois la séquence du cas correspondant a besoin de faire appel à la séquence d'un autre cas : on trouve alors « Appel du cas X »

Séquences alternatives :

Une séquence alternative diverge de la séquence nominale (c'est un embranchement dans une séquence nominale) mais y revient toujours !

A1 : montant demandé supérieur au solde

L'enchaînement A1 démarre au point 5 du scénario nominal

6. **Le guichetier indique au client que le montant demandé est supérieur au solde**
7. **Le guichetier invite le client à préciser un montant inférieur au solde**

La séquence nominale reprend au point 4

Séquences d'exception :

- **Panne du système informatique**
- **Mauvaise identification du client (mauvais numéro de compte)**

Se produisent en cas d'erreur (la séquence nominale s'interrompt sans retour). La survenue des erreurs dans les séquences doit être signalée par « Appel de l'exception Y »

Autre exemple.

Identification

Nom du cas : Retirer de l'argent avec une carte Visa au GBA

But : Ce cas d'utilisation permet à un porteur de carte Visa, qui n'est pas client de la banque, de retirer de l'argent, si son crédit hebdomadaire le permet

Acteur principal : Porteur de CB Visa

Acteur secondaire : SA Visa

Date : le 23/06/2014

Responsable : M. Pierre

Version : 1.0

Séquencement

Pré-conditions

La caisse du GAB est alimentée.

Aucune carte bancaire ne se trouve dans le lecteur.

Enchaînement nominal

1. **Le porteur de CB Visa introduit sa carte Visa dans le lecteur de cartes du GAB**
2. **Le GAB vérifie que la carte introduite est bien une carte Visa**
3. **Le GAB demande au porteur de CB Visa de saisir son code d'identification**
4. **Le porteur de CB Visa saisit son code d'identification**
5. **Le GAB compare le code d'identification avec celui que est codé sur la puce de la carte**
6. **Le GAB demande une autorisation au système d'autorisation VISA**
7. **Le système d'autorisation VISA donne son accord et indique le solde hebdomadaire**
8. **Le GAB demande au porteur de CB Visa de saisir le montant désiré du retrait**
9. **Le porteur de CB Visa saisit le montant désiré du retrait**
10. **Le GAB contrôle le montant demandé par rapport au solde hebdomadaire**

11. Le GAB demande au porteur de CB Visa s’il veut un ticket
12. Le porteur de CB Visa demande un ticket
13. Le GAB rend sa carte au porteur de CB Visa
14. Le porteur de CB Visa reprend sa carte
15. Le GAB délivre les billets et un ticket

Enchaînements « alternatifs

A1 : code d’identification provisoirement erroné

L’enchaînement A1 démarre au point 5 du scénario nominal

6. Le GAB indique au client que le code est erroné, pour la première ou la deuxième fois
7. Le GAB enregistre l’échec sur la carte

Le scénario nominal reprend au point 3

A2 : montant demandé supérieur au solde hebdomadaire

L'enchaînement A2 démarre au point 10 du scénario nominal

11. **Le GAB indique au client que le montant demandé est supérieur au solde hebdomadaire**

Le scénario nominal reprend au point 8.

A3: ticket refusé

L'enchaînement A3 démarre au point 11 du scénario nominal

12. **Le porteur de CB Visa refuse le ticket**

Le scénario nominal reprend au point 13.

Enchaînements d'exception

E1: carte non valide

L'enchaînement E1 démarre au point 2 du scénario nominal

3. Le GAB indique au porteur que la carte n'est pas valide, le cas d'utilisation est terminé

E2 : code d'identification définitivement erroné

L'enchaînement E2 démarre au point 5 du scénario nominal

6. Le GAB indique au client que le code est erroné, pour la troisième fois
7. Le GAB confisque la carte
8. Le système d'autorisation VISA est informé; le cas d'utilisation est terminé

E3 : retrait non autorisé

L'enchaînement E3 démarre au point 6 du scénario nominal

7. Le système d'autorisation VISA interdit tout retrait
8. Le GAB éjecte la carte ; le cas d'utilisation est terminé

E4 : carte non reprise

L’enchaînement E4 démarre au point 13 du scénario nominal

14. Au bout de 15 secondes, le GAB confisque la carte

15. Le système d’autorisation VISA est informé; le cas d’utilisation est terminé

E5 : billets non pris

L’enchaînement E5 démarre au point 15 du scénario nominal

16. Au bout de 30 secondes, le GAB reprend les billets

17. Le système d’autorisation VISA est informé; le cas d’utilisation est terminé

Post-condition

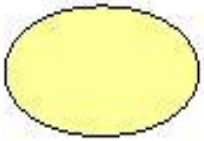
La caisse du GAB contient moins de billets qu’au début du cas d’utilisation

Pour décrire ou valider un CU : les scénarios

- **Un scénario est un exemple :**
  - **une manière particulière d'utiliser le système …**
  - **par une personne particulière …**
  - **dans un contexte particulier**

- **cas d'utilisation = ensemble de scénarios**

- **scénario = une exécution particulière d'un CU**



Retirer DeLArgent AuDistributeur

SCENARIO 4

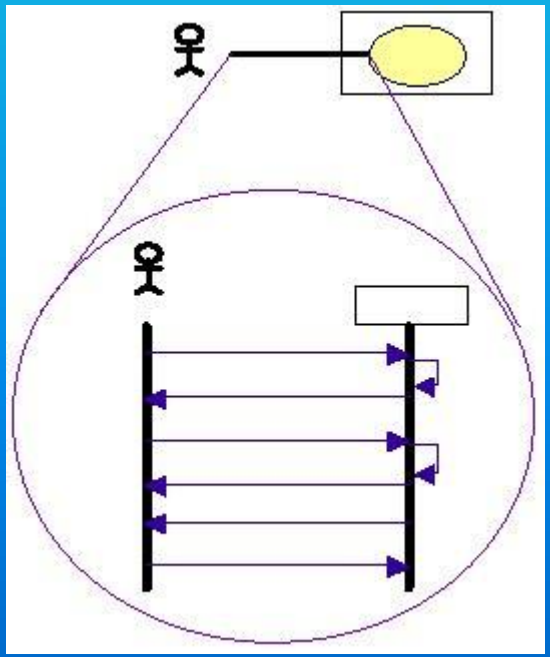
- *Paul* insère sa carte dans le distributeur d2103
- Le *système* accepte la carte et lit le numéro de compte
- Le *système* demande le code
- *Paul* tape ‘ 1234 ’
- Le *système* indique que ce n ’est pas le bon code
- Le *système* affiche un message et propose de recommencer
- *Paul* tape ‘ 6622’
- Le *système* affiche que le code est correct
- Le *système* demande le montant du retrait
- *Paul* tape 50000 fr
- Le *système* vérifie s ’il y a assez d ’argent sur le compte
- ...

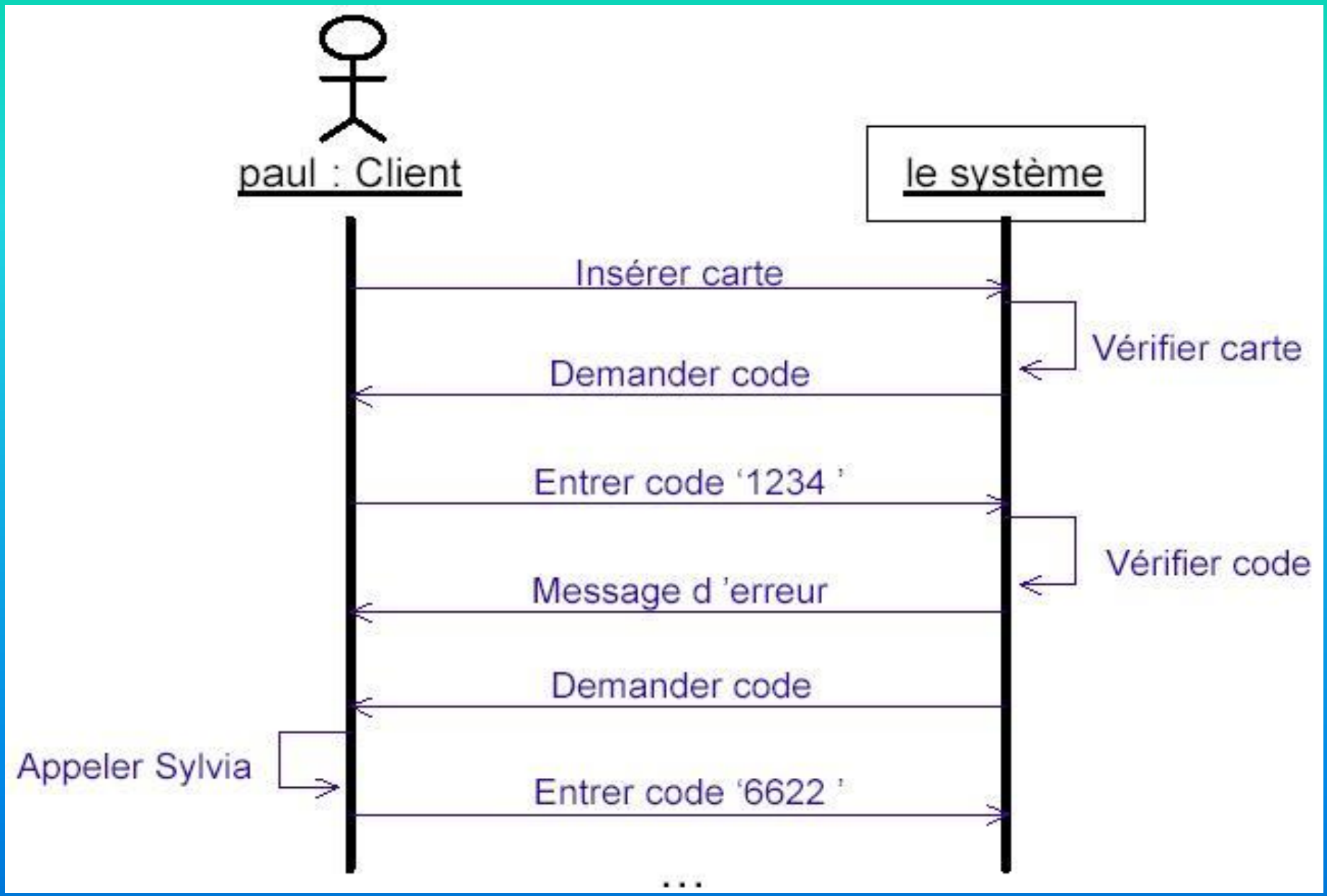
Pour décrire un scénario : un diagramme de séquences

- **Diagramme de séquences :**

  - L'une des notations UML, une notation générale
  - Peut être utilisée dans de nombreux contextes
  - Permet de décrire une séquence des messages échangés entre différents objets
  - Différents niveaux de détails

- **Pour décrire un scénario simple, deux objets : l'acteur et le système**



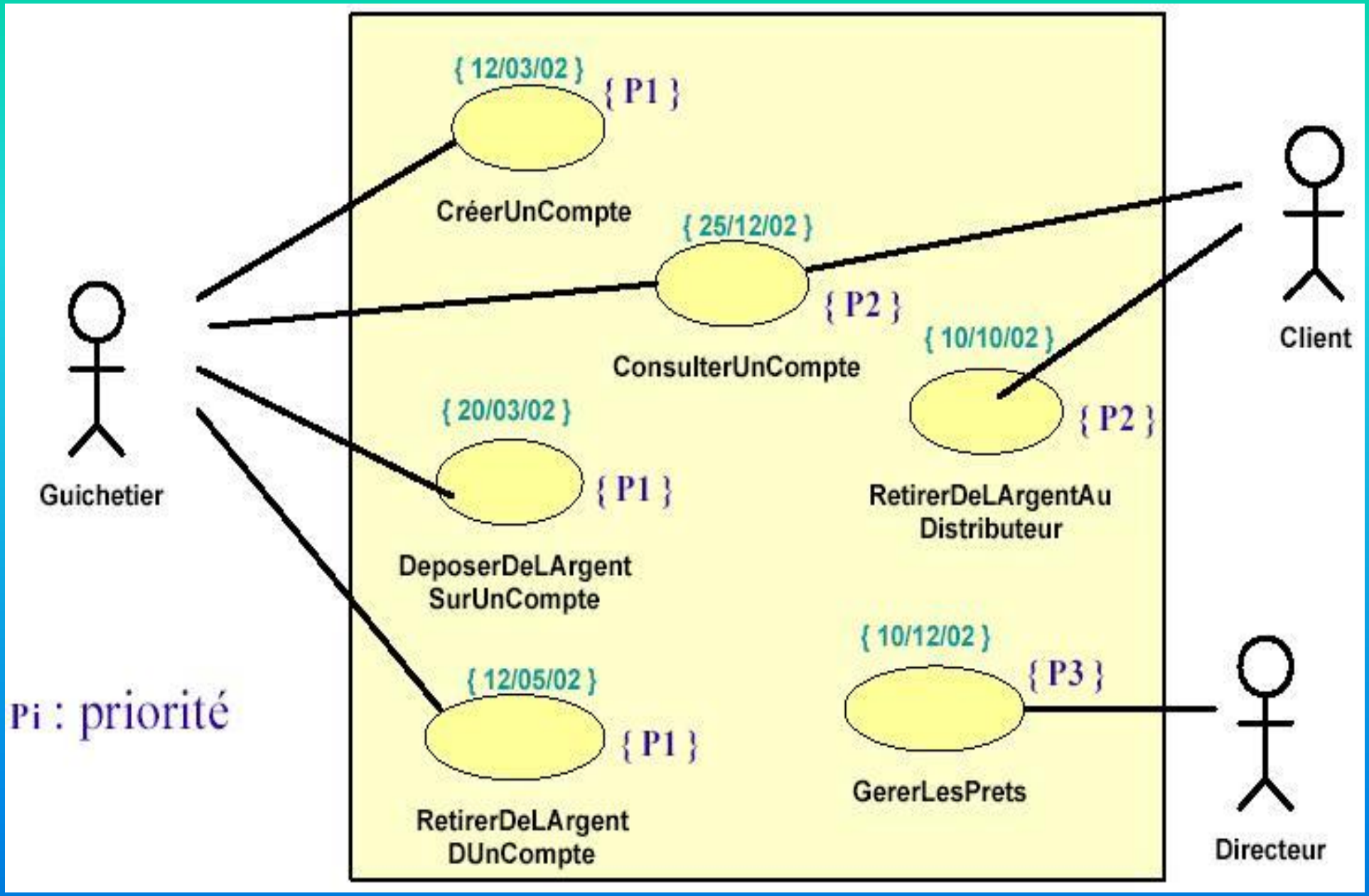

paul : Client
le système
Insérer carte
Vérifier carte
Demander code
Entrer code '1234 '
Vérifier code
Message d 'erreur
Demander code
Appeler Sylvia
Entrer code '6622 '
...

- **PROCESSUS UNIFIE**

- **(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**
  - **(1.5) Décrire le modèle comme un tout**

- **(2) Définir des priorités entre CU**

- **(3) Détailler chaque CU (en tenant compte des priorités)**


{ 12/03/02 }
{ P1 }
CréerUnCompte
{ 25/12/02 }
{ P2 }
ConsulterUnCompte
{ 10/10/02 }
{ P2 }
RetirerDeLArgentAu Distributeur
{ 20/03/02 }
{ P1 }
DeposerDeLArgent SurUnCompte
{ 12/05/02 }
{ P1 }
RetirerDeLArgent DUnCompte
{ 10/12/02 }
{ P3 }
GererLesPrets
Guichetier
Client
Directeur
Pi : priorité