



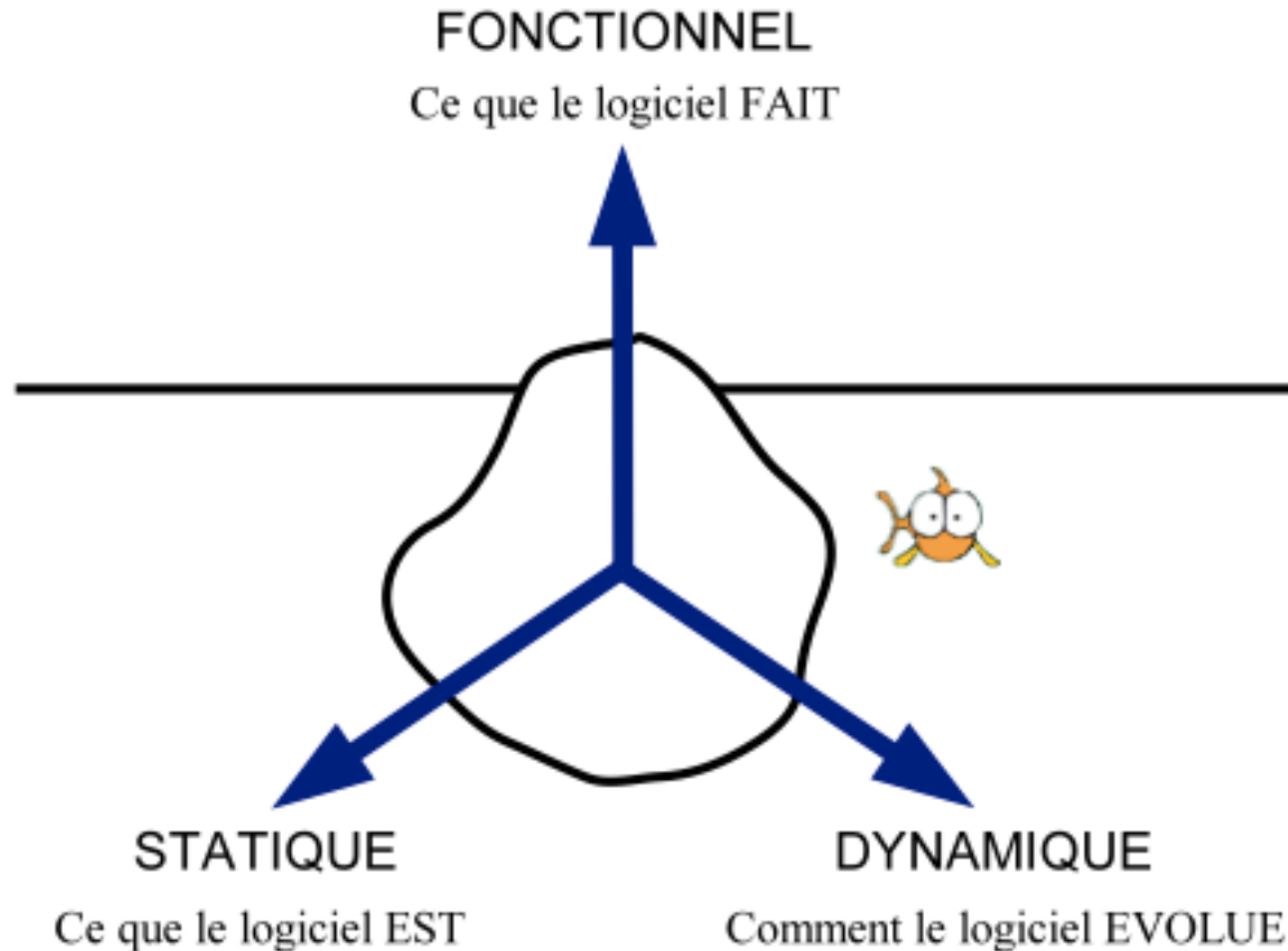
Génie Logiciel et projet de synthèse

Analyse fonctionnelle

Kévin Bailly
Institut des Systèmes Intelligents et de
Robotique

kevin.bailly@upmc.fr

Axe de modélisation



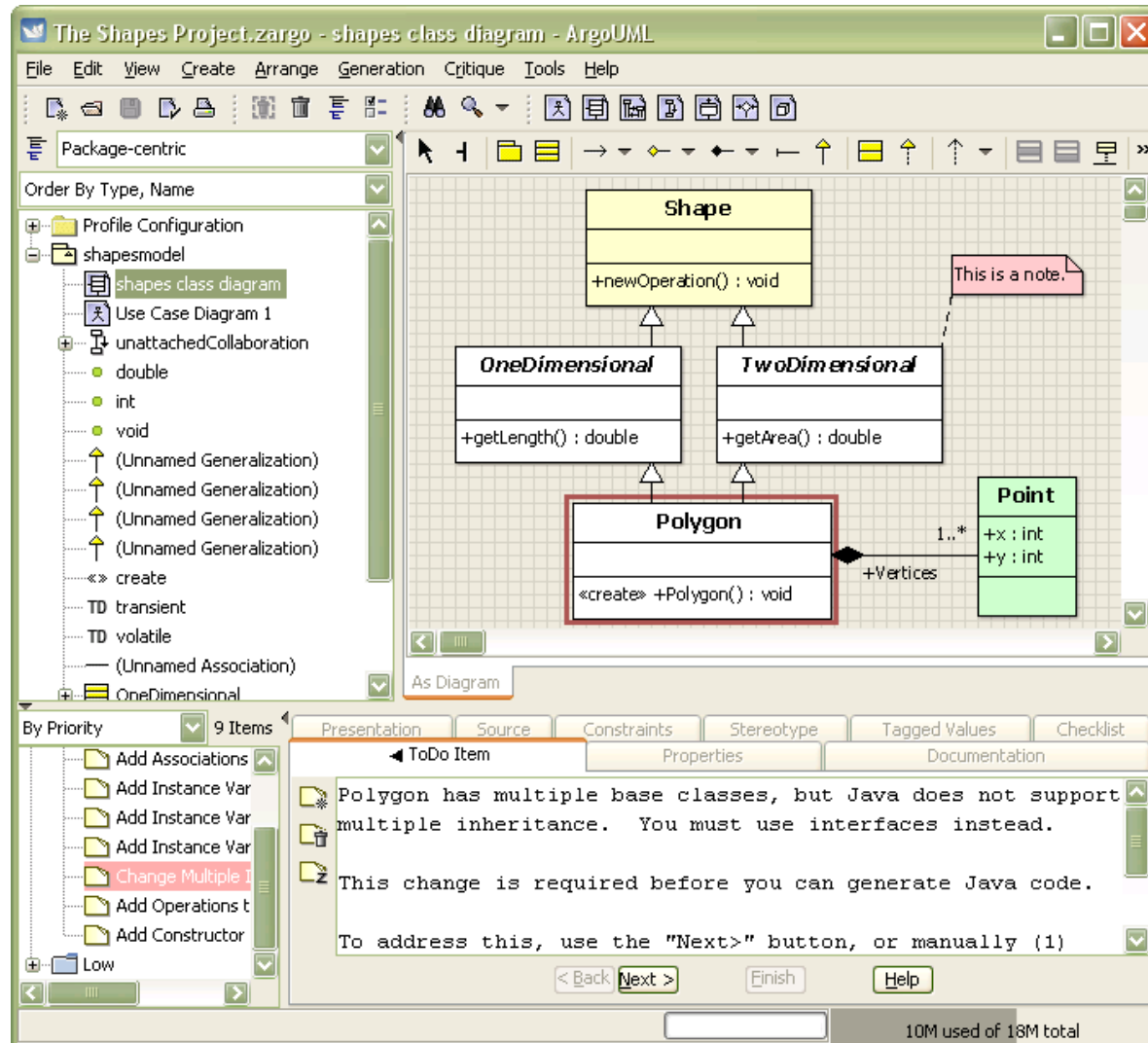
Qu'est ce qu'UML

- UML est un **langage universel de modélisation** objets
- UML est une **notation**, un outil de communication **visuelle** (diagrammes)
- UML est une **norme** :
 - maintenue par l'organisation a but non lucratif OMG (Object Management Group)
 - disponible gratuitement sur www.uml.org
- UML n'est pas un processus de développement
- UML n'est pas un langage de programmation

Logiciels de modélisation



ArgoUML

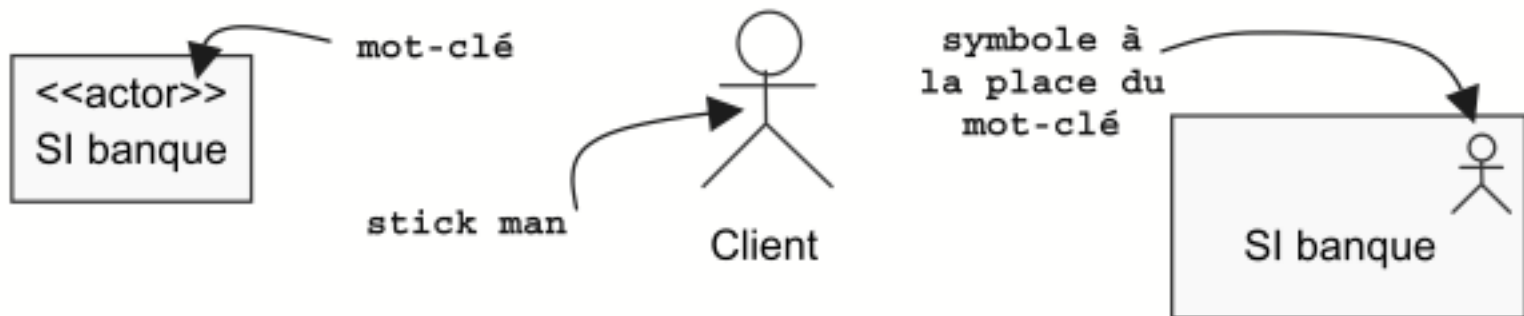


- Fonctionnalités (simplifiées) d'un d'un Guichet Automatique de Banque (GAB) :
 - 1) Distribution de l'argent à tout porteur de carte de crédit, via un lecteur de carte et un distributeur de billets
 - 2) Consultation du solde, dépôt d'argent liquide ou de chèques
 - 3) Les transactions sont sécurisées par le système d'autorisation des cartes bancaires pour les retraits et le système d'information de la banque pour les autres opérations
 - 4) Certaines opérations de maintenances sont nécessaires (rechargement en billets, récupérations des cartes avalées...

- L'objectif de la phase de modélisation fonctionnelle est de :
 - Identifier les **acteurs**
 - Identifier les **cas d'utilisation**
 - Réaliser un **diagramme de cas d'utilisation**
 - Décrire les cas d'utilisation par des **scénarios**
 - **Organiser/hierarchiser** des cas d'utilisation

- L'objectif de la phase de modélisation fonctionnelle est de :
 - Identifier les **acteurs**
 - Identifier les **cas d'utilisation**
 - Réaliser un **diagramme de cas d'utilisation**
 - Décrire les cas d'utilisation par des **scénarios**
 - **Organiser/hierarchiser** des cas d'utilisation

- Rôle spécifique d'une **entité externe** (humain, autres système) qui **interagit directement** avec le système étudié : acteur \neq utilisateur
- Un acteur peut **consulter et/ou modifier** l'état du système en émettant et recevant des **messages**
- Représentation :



Exercice

- Identifier les acteurs du GAB

- Point (1) : **Porteur de carte**
 - Pas le lecteur de carte ni le distributeur de billet, ils font partis du GAB
 - Pas la carte : on privilégie l'acteur logique (celui qui profite de l'opération) à l'acteur « physique »
→ indépendant de la technologie utilisé
- Point (2) : **Client de la banque**
- Point (3) : Systèmes externes
 - **Système d'autorisation CB** (pour les retraits)
 - **Système d'information de la banque** (autres op)
- Point (4) : **Opérateurs de maintenance**

- L'objectif de la phase de modélisation fonctionnelle est de :
 - Identifier les **acteurs**
 - Identifier les **cas d'utilisation**
 - Réaliser un **diagramme de cas d'utilisation**
 - Décrire les cas d'utilisation par des **scénarios**
 - **Organiser/hierarchiser** des cas d'utilisation

- Les CU décrivent des **acteurs** qui interagissent avec le **système** pour atteindre des **objectifs** spécifiques
- Objectifs des CU :
 - Décrire les exigences fonctionnelles du système :
 - Ce concentre sur le **Quoi ?** pas sur le **Comment ?**
 - Point de vue de l'acteur
 - Chaque CU décrit une tâche **élémentaire** du système
 - Tâche effectuée par une personne à un seul endroit
 - En réponse à un événement métier
 - laisse les données dans un état cohérent

- Comment les identifier et les analyser ?
 - Rechercher les différentes intentions métier
 - Déterminer les fonctionnalités attendues du système (à partir du cahier des charges)
 - Préciser les variantes possibles : cas nominal et cas alternatifs (en cas d'erreur)
- Comment les nommer ?
 - Verbe à l'infinitif + complément centré acteur
 - Ex : Consulter solde, démarrer voiture...

Exercice

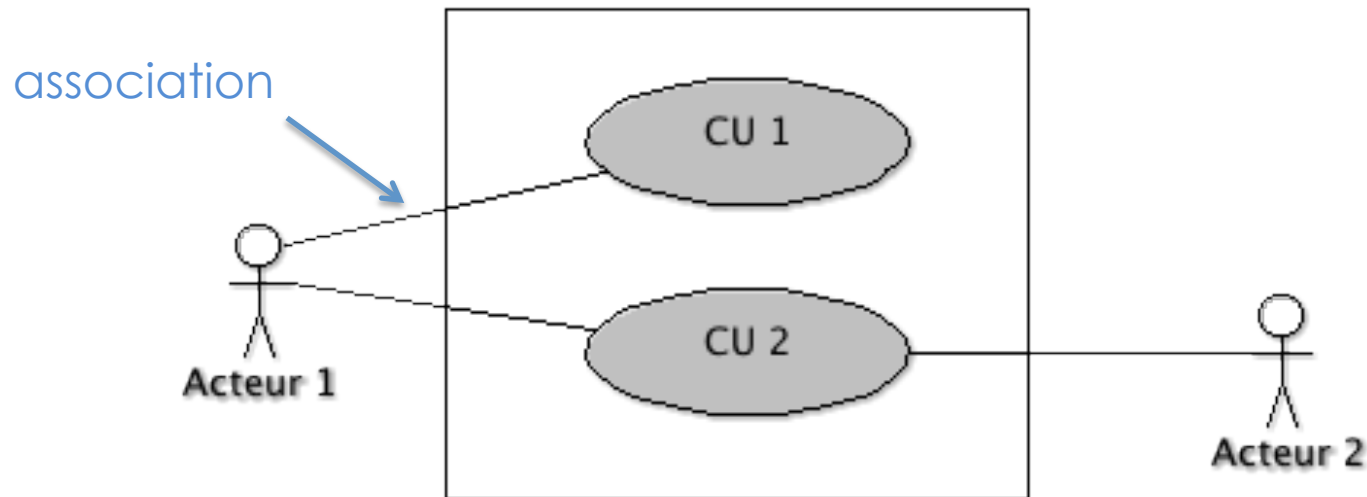
- Identifier les acteurs du GAB
- Identifier les cas d'utilisation

- Porteur de carte :
 - **Retirer** de l'argent
- Client banque :
 - **Retirer** de l'argent
 - **Consulter** le solde
 - **Déposer** du liquide
 - **Déposer** des chèques (séparer liquide et chèque si les scénarios sont très différents, cf. description détaillée)
- Opérateur de maintenance
 - **Recharger** distributeur
 - **Maintenir** état opérationnel (récupérer carte avalées, récupérer chèques déposés...)
- Système d'autorisation
 - **Néant**
- Système d'information banque
 - **Néant**

- L'objectif de la phase de modélisation fonctionnelle est de :
 - Identifier les **acteurs**
 - Identifier les **cas d'utilisation**
 - Réaliser un **diagramme de cas d'utilisation**
 - Décrire les cas d'utilisation par des **scénarios**
 - **Organiser/hierarchiser** des cas d'utilisation

Cas d'utilisation (CU)

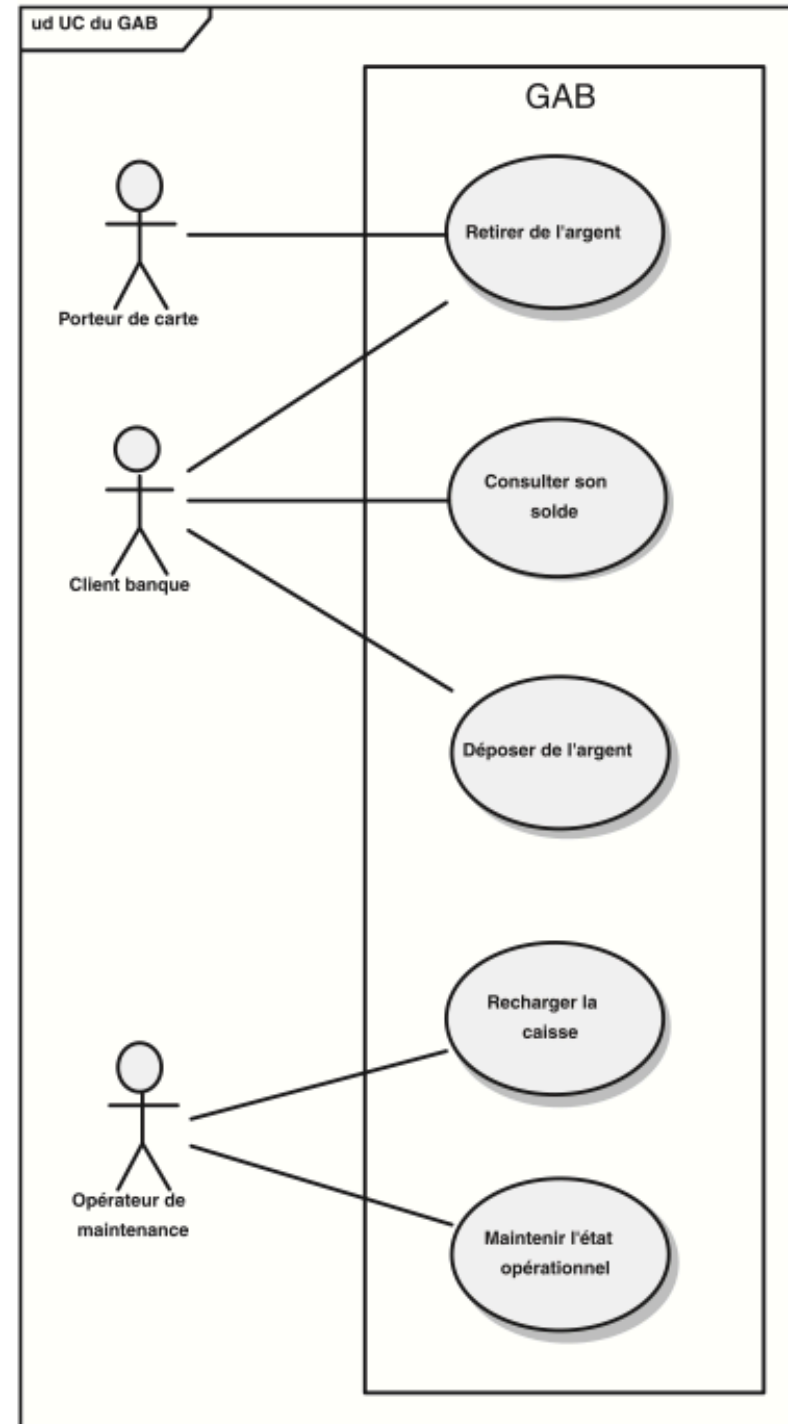
- Comment les représenter ?



Exercice

- Identifier les acteurs du GAB
- Identifier les cas d'utilisation
- Réaliser le digramme des CU

Proposition de diagramme des CU



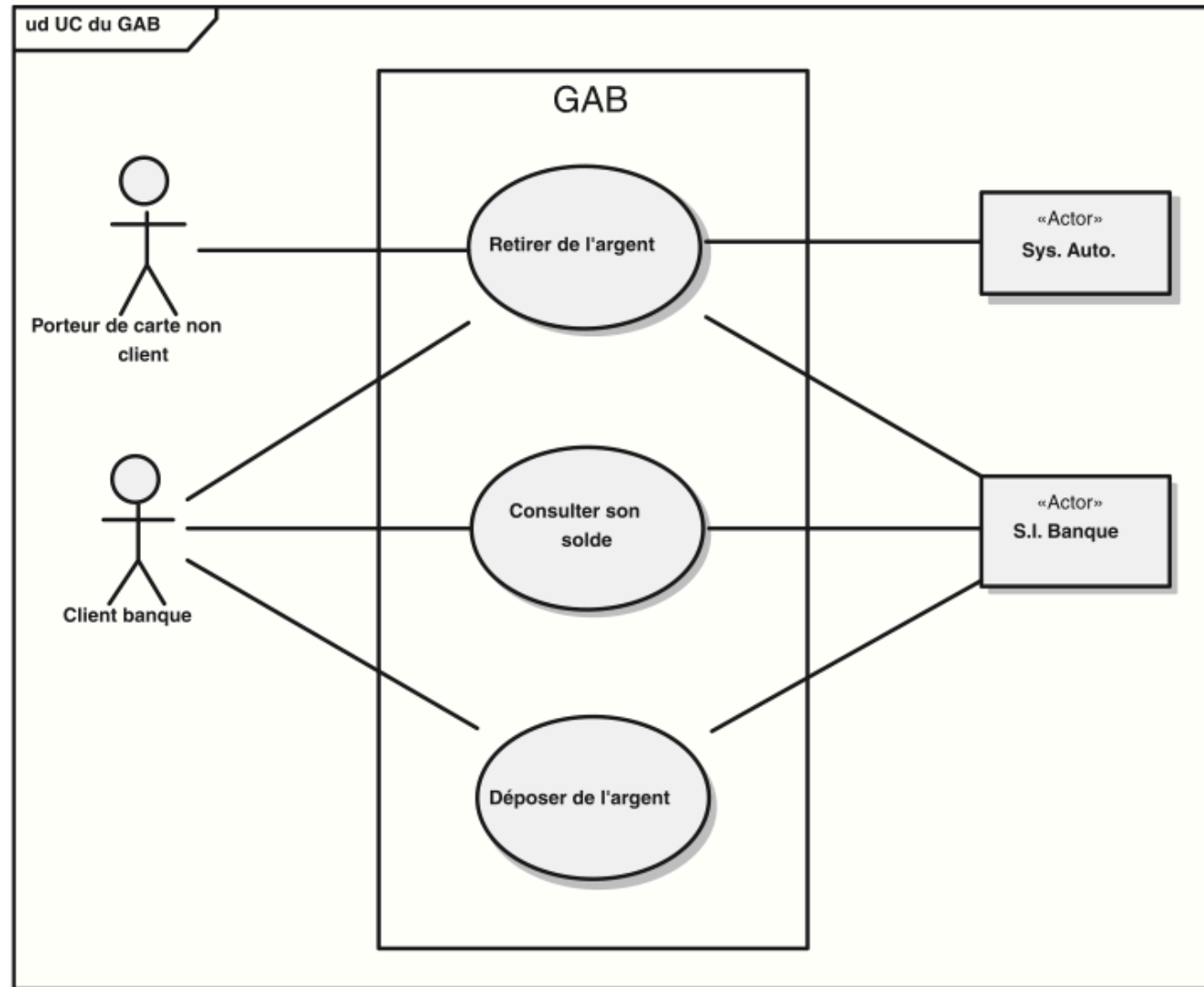
Tous les acteurs n'utilisent pas forcément le système :

- Acteur principal : le cas d'utilisation produit un **résultat observable** pour cet acteur
 - Acteur secondaire : autres participant du cas d'utilisation. **Consultation et/ou information** du système uniquement
-
- Convention : acteur principal à gauche et acteur secondaire à droite

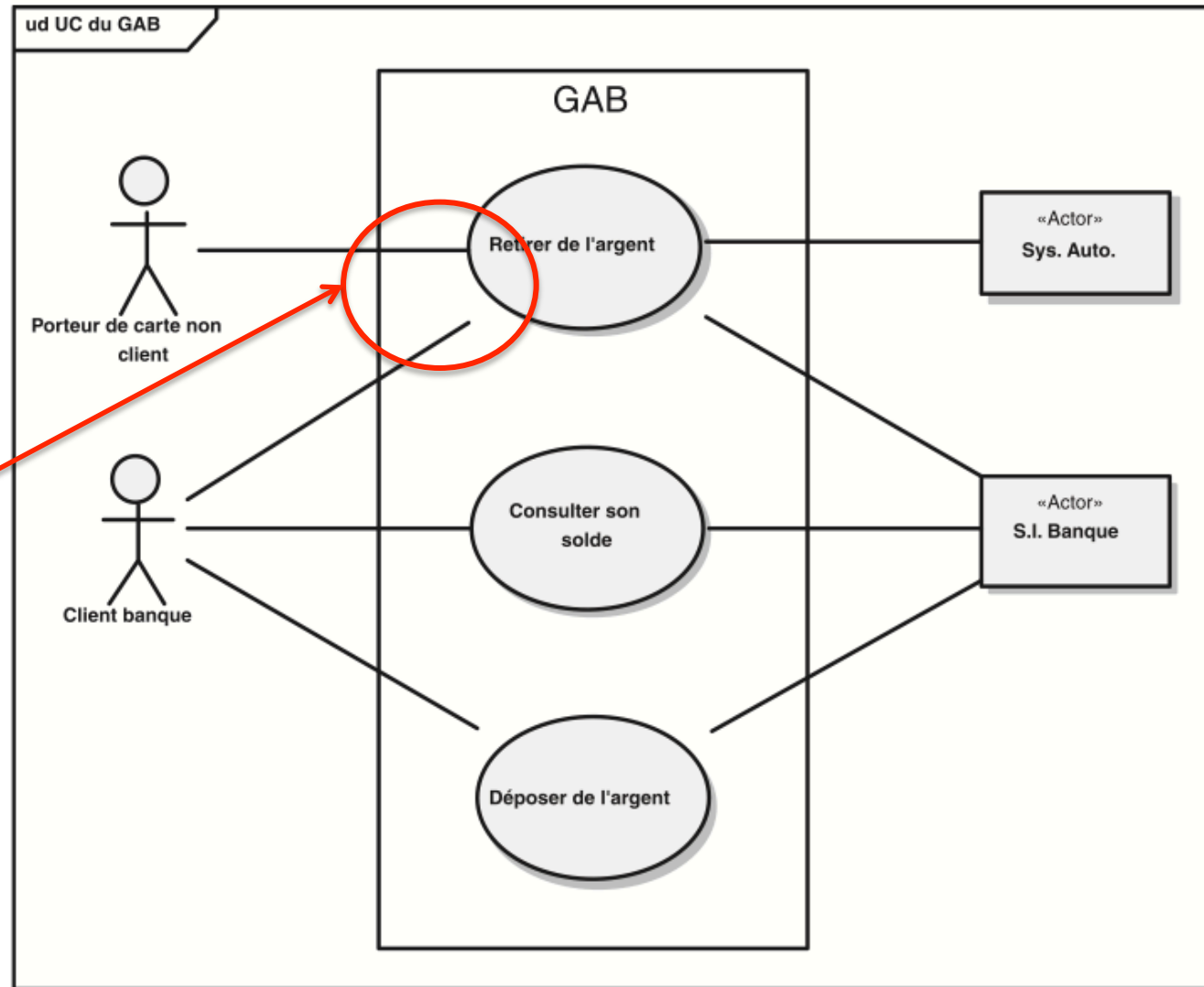
Exercice

- Identifier les acteurs du GAB
- Identifier les cas d'utilisation
- Réaliser le digramme des CU
- Identifier les acteurs principaux et secondaires et compléter le diagramme

Solution simple



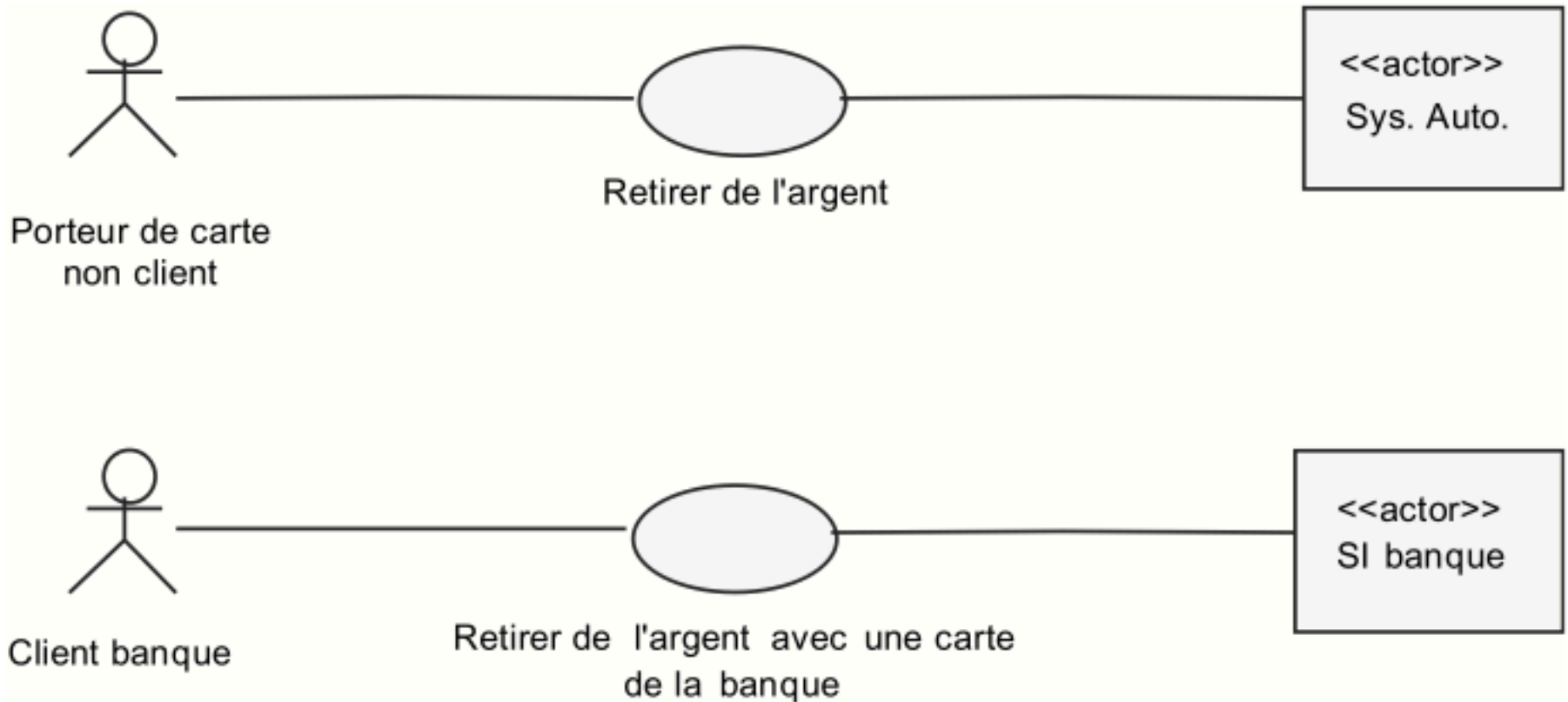
Solution simple



Risque de confusion
sur le rôle de
chaque acteur dans
ce cas d'utilisation

Autre solution

- Séparer en 2 cas d'utilisation différents :



- L'objectif de la phase de modélisation fonctionnelle est de :
 - Identifier les **acteurs**
 - Identifier les **cas d'utilisation**
 - Réaliser un **diagramme de cas d'utilisation**
 - Décrire les cas d'utilisation par des **scénarios**
 - **Organiser/hierarchiser** des cas d'utilisation

- *Scénario* : **séquence spécifique d'actions** et d'interactions entre les acteurs et le système
= **instance** d'un cas d'utilisation
- Un CU contient généralement :
 - Un scénario **principal** : déroulement normal
 - Des scénarios **alternatifs** : reprennent ensuite à une étape du scénario principal
 - Des scénarios **d'erreur** : terminent brutalement (objectifs de l'acteur non atteint)

- Description textuelle des CU :
 - **Pas normalisée** dans UML...
 - ...mais **indispensable** ! (seul moyen de communiquer avec les experts métiers)
- Une fiche de description doit contenir :
 - **Sommaire d'identification** : titre, résumé, acteurs, date de création et modification, version, auteur
 - **Description des scénarios** : pré-conditions, scénarios nominal et alternatifs, post-conditions
 - **Exigences non fonctionnelles** (optionnelles) : performance, fiabilité, précision, contraintes ergonomique, charte graphique...

Exercice

- Identifier les acteurs du GAB
- Identifier les cas d'utilisation
- Réaliser le digramme des CU
- Identifier les acteurs principaux et secondaires et compléter le diagramme
- Faire une description textuelle du cas d'utilisation *retirer de l'argent*

- **Titre** : Retirer de l'argent
- **Résumé** : ce cas d'utilisation permet à un Porteur de carte, qui n'est pas client de la banque, de retirer de l'argent, si son crédit hebdomadaire le permet.
- **Acteurs** :
 - Porteur de carte (principal)
 - Système d'autorisation (secondaire)
- **Date de création** : 02/03/10 Date de mise à jour : 05/05/12
- **Version** : 2.0
- **Responsable** : M Dupont

- Pré-conditions
 - La caisse du GAB est alimentée (il reste au moins un billet)
 - Aucune carte ne se trouve déjà coincée dans le lecteur.
 - La connexion avec le Système d'autorisation est opérationnelle

- Scénario nominal

1. Le Porteur de carte introduit sa carte dans le lecteur de cartes du GAB.
2. Le GAB vérifie que la carte introduite est bien une carte bancaire.
3. Le GAB demande au Porteur de carte de saisir son code d'identification.
4. Le Porteur de carte saisit son code d'identification.
5. Le GAB compare le code d'identification avec celui qui est codé sur la puce de la carte.
6. Le GAB demande une autorisation au Système d'autorisation.
7. Le Système d'autorisation donne son accord et indique le solde hebdomadaire.
8. Le GAB demande au Porteur de carte de saisir le montant désiré du retrait.
9. Le Porteur de carte 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 carte s'il veut un ticket.
12. Le Porteur de carte demande un ticket.
13. Le GAB rend sa carte au Porteur de carte.
14. Le Porteur de carte reprend sa carte.
15. Le GAB délivre les billets et un ticket.
16. Le Porteur de carte prend les billets et le ticket.

- Enchaînements alternatifs
 - A1 : *code d'identification provisoirement erroné*, démarre au point 5 du scénario nominal
 - 6. Le GAB indique au Porteur de carte que le code est erroné, pour la première ou deuxième fois.
 - 7. Le GAB enregistre l'échec sur la carte. Le scénario nominal reprend au point 3
 - Le scénario nominal reprend au point 3.

- Enchaînements alternatifs
 - A2 : *montant demandé supérieur au solde hebdomadaire*, démarre au point 10
 - 11. Le GAB indique au Porteur de carte que le montant demandé est supérieur au solde hebdomadaire.
 - Le scénario nominal reprend au point 8.

- Enchaînements alternatifs
 - A3 : *ticket refusé*, démarre au point 11
 - 12. Le Porteur de carte refuse le ticket
 - 13. Le GAB rend sa carte au Porteur de carte
 - 14. Le Porteur de carte reprend sa carte
 - 15. Le GAB délivre les billets
 - 16. Le Porteur de carte prend les billets

- Enchaînements d'erreur
 - E1 : *carte non-valide, démarre au point 2 du scénario nominal.*
 - Le GAB indique au Porteur que la carte n'est pas valide (illisible, périmée, etc.), la confisque ; le cas d'utilisation se termine en **échec**.
 - E2 : *code d'identification définitivement erroné, démarre au point 5 du scénario nominal.*
 - Le GAB indique au Porteur de carte que le code est erroné, pour la troisième fois
 - Le GAB confisque la carte
 - Le Système d'autorisation est informé ; le cas d'utilisation se termine en échec
 - *Autres enchainement d'erreur : retrait non autorisé, carte non reprise, billets non pris, annulation de la transaction*

Description graphique des CU

- La **description textuelle est indispensable**, mais la représentation graphique est un complément très utile
- Permet de **visualiser** facilement
 - La succession des enchainements
 - Les interactions entre le système et les différents acteurs
- Facilite la **maintenance**
- On utilise généralement 2 diagrammes UML
 - Diagramme de **séquence**
 - Diagramme d'**activité**

Diagramme de séquences

- Pour modéliser les **enchainements d'action d'un scénario**

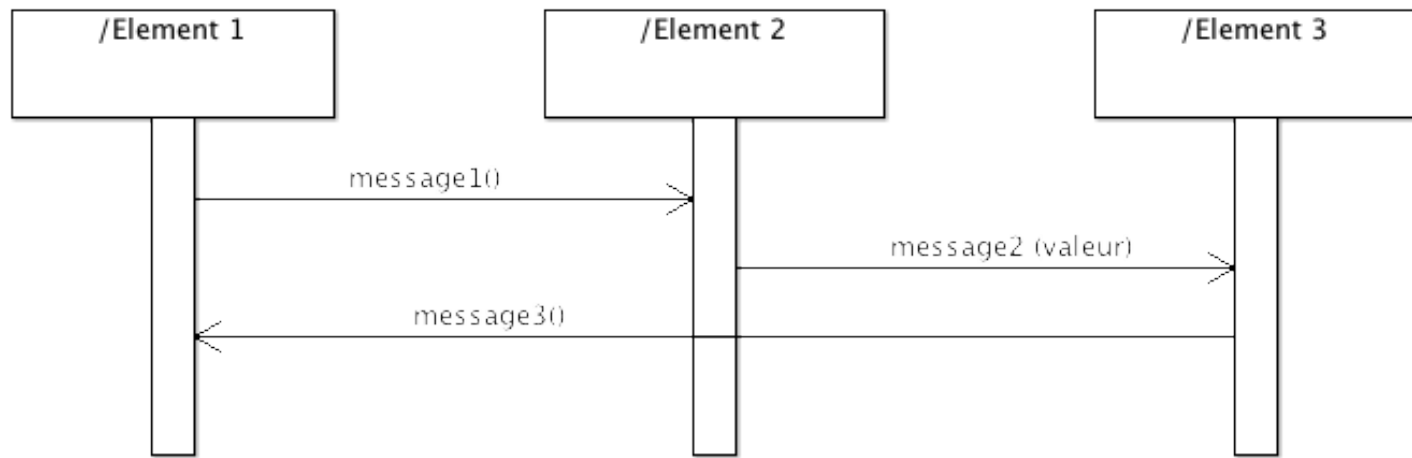


Diagramme de séquences

- Recommandation :

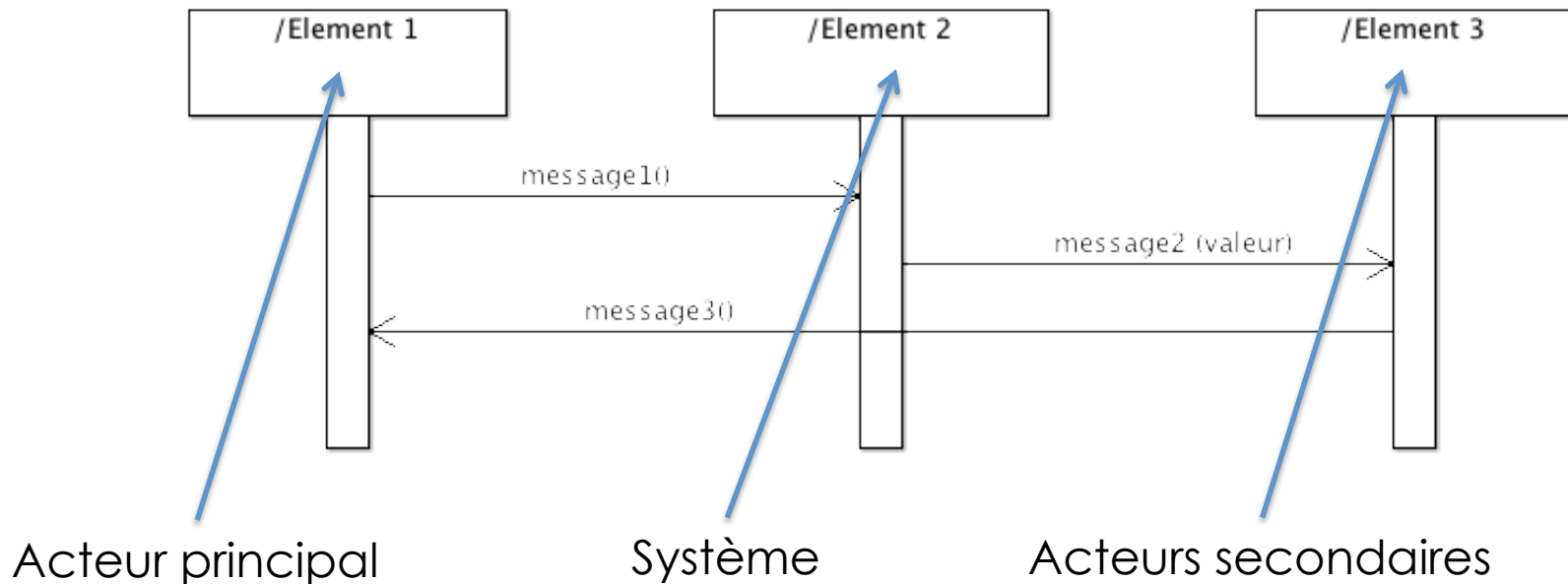
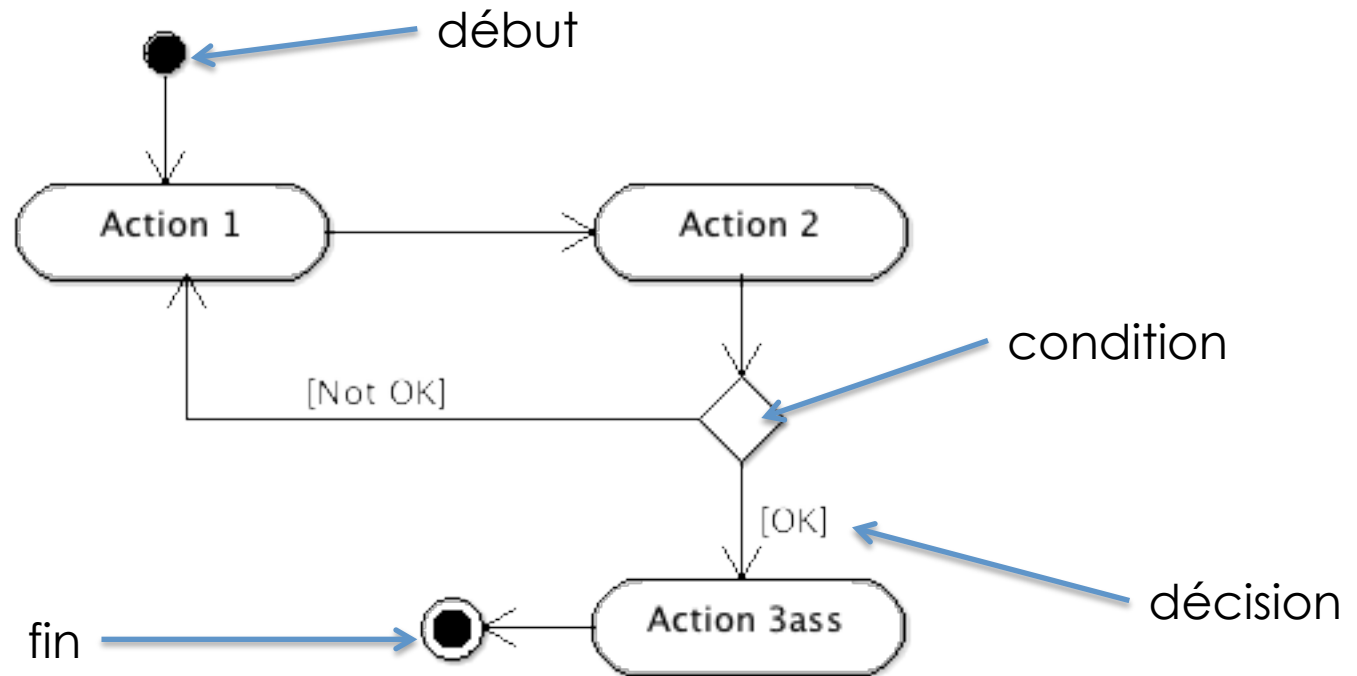


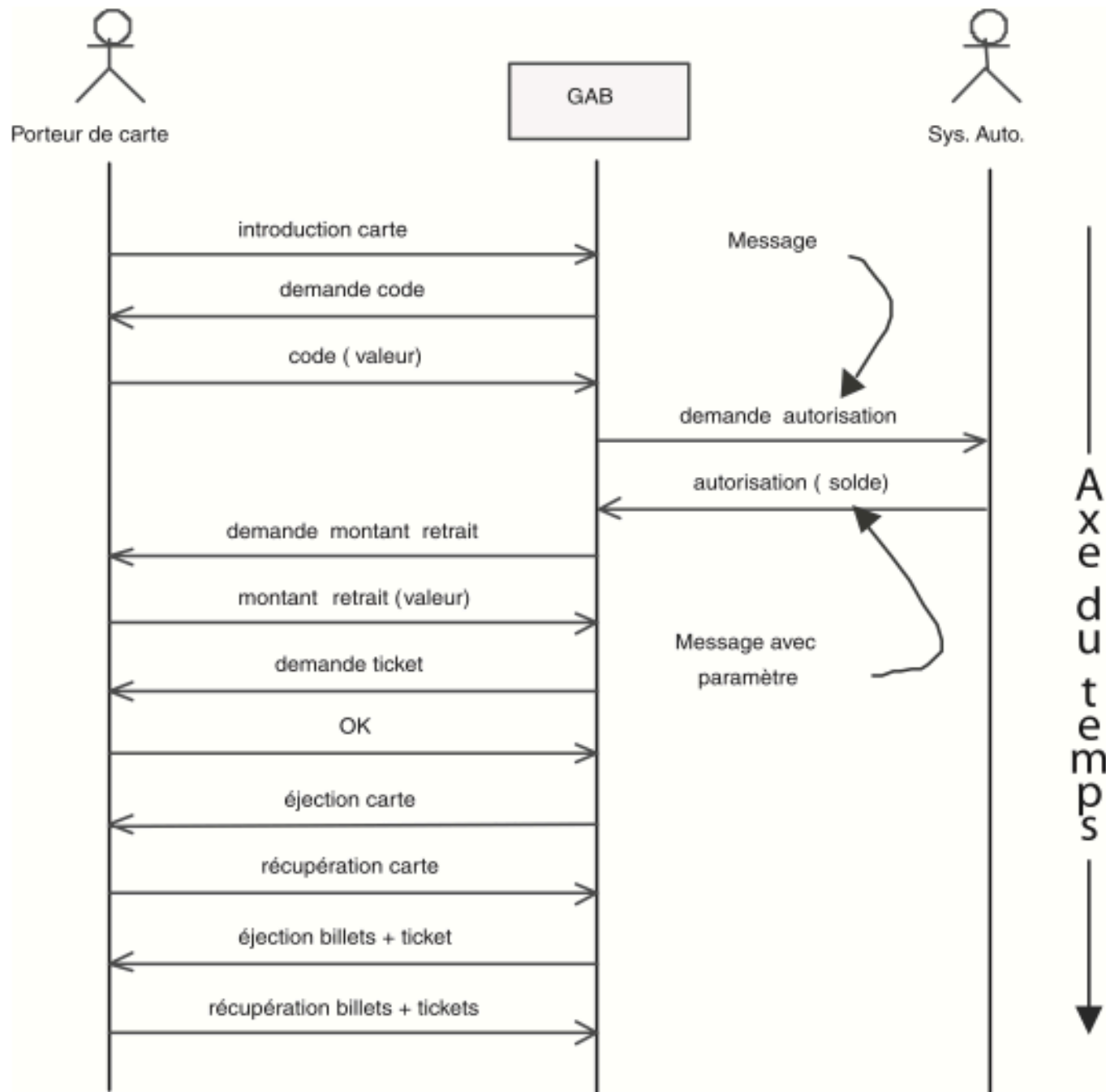
Diagramme d'activité

- Pour modéliser la **dynamique d'un cas d'utilisation**



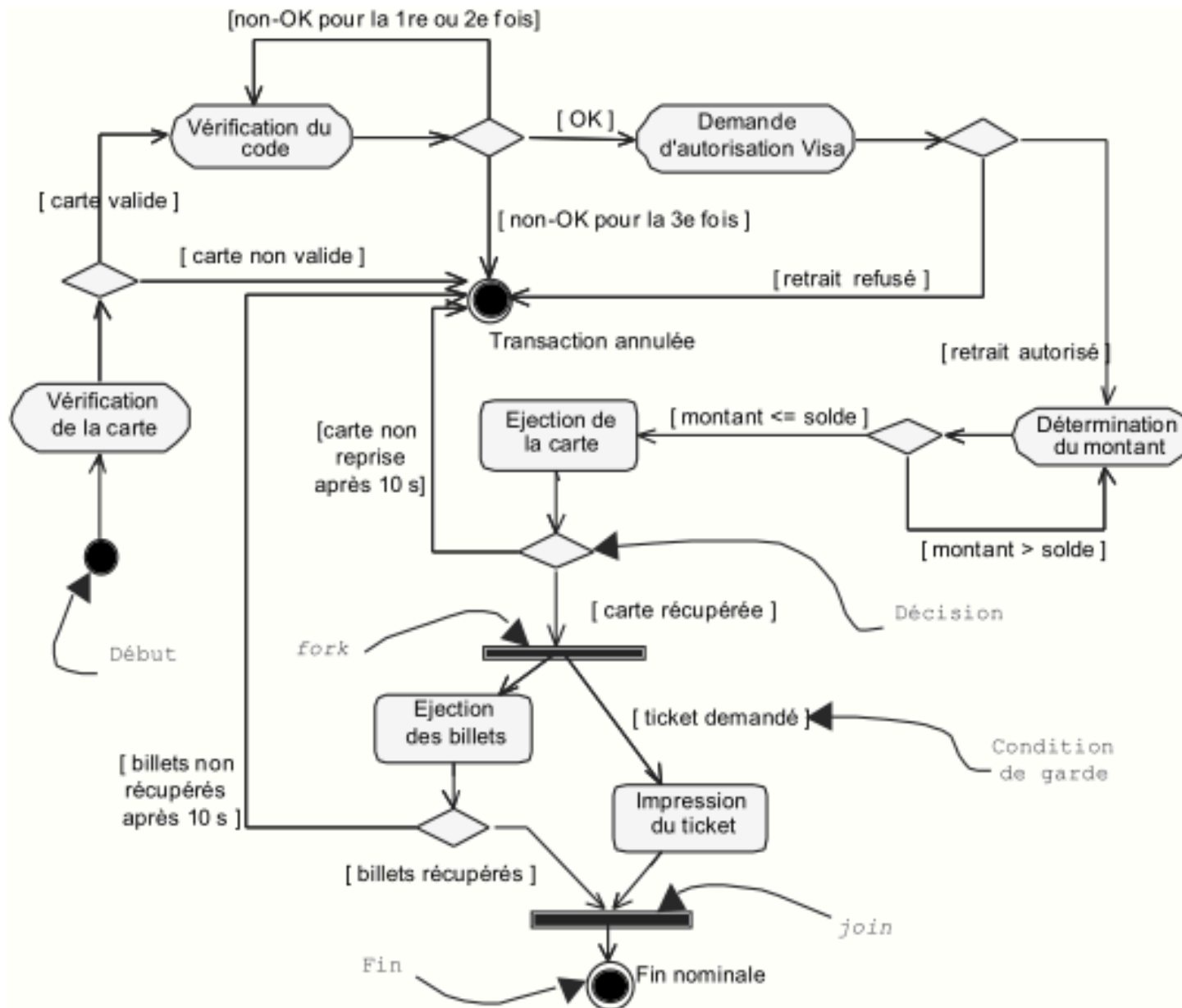
Exercice

- Identifier les acteurs du GAB
- Identifier les cas d'utilisation
- Réaliser le digramme des CU
- Identifier les acteurs principaux et secondaires et compléter le diagramme
- Faire une description textuelle du cas d'utilisation *retirer de l'argent*
- Dessiner le diagramme de séquence système qui décrit le scénario principal du cas d'utilisation *retirer de l'argent*



Exercice

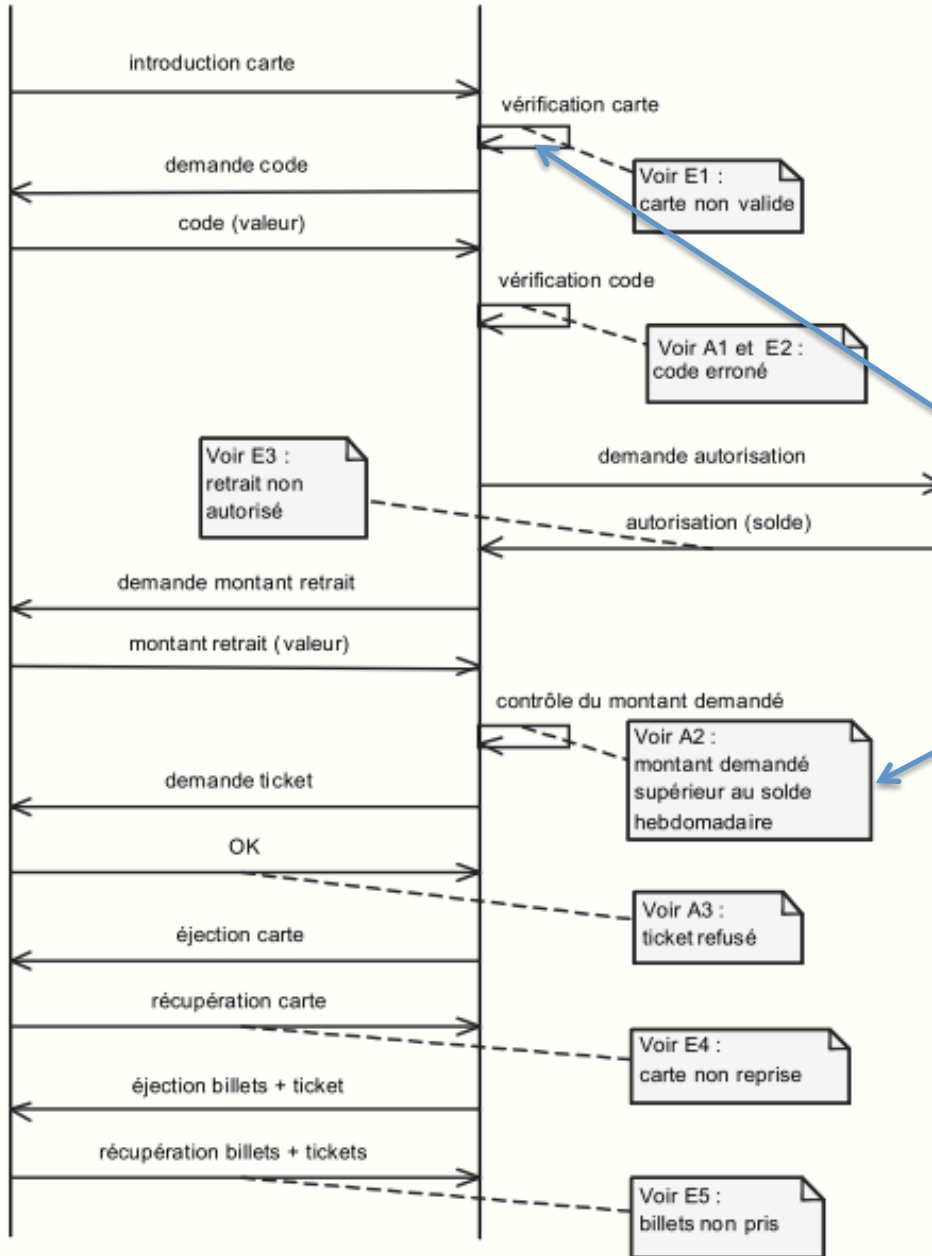
- Identifier les acteurs du GAB
- Identifier les cas d'utilisation
- Réaliser le digramme des CU
- Identifier les acteurs principaux et secondaires et compléter le diagramme
- Faire une description textuelle du cas d'utilisation *retirer de l'argent*
- Dessiner le diagramme de séquence système qui décrit le scénario principal du CU retirer de l'argent
- Dessiner le diagramme d'activité du CU retirer de l'argent



Autre solution : diagramme de séquence système enrichi

Actions internes du système

renvois aux enchainements alternatifs



- L'objectif de la phase de modélisation fonctionnelle est de :
 - Identifier les **acteurs**
 - Identifier les **cas d'utilisation**
 - Réaliser un **diagramme de cas d'utilisation**
 - Décrire les cas d'utilisation par des **scénarios**
 - **Organiser/hierarchiser** des cas d'utilisation

2 manières complémentaires de détailler et d'organiser les CU :

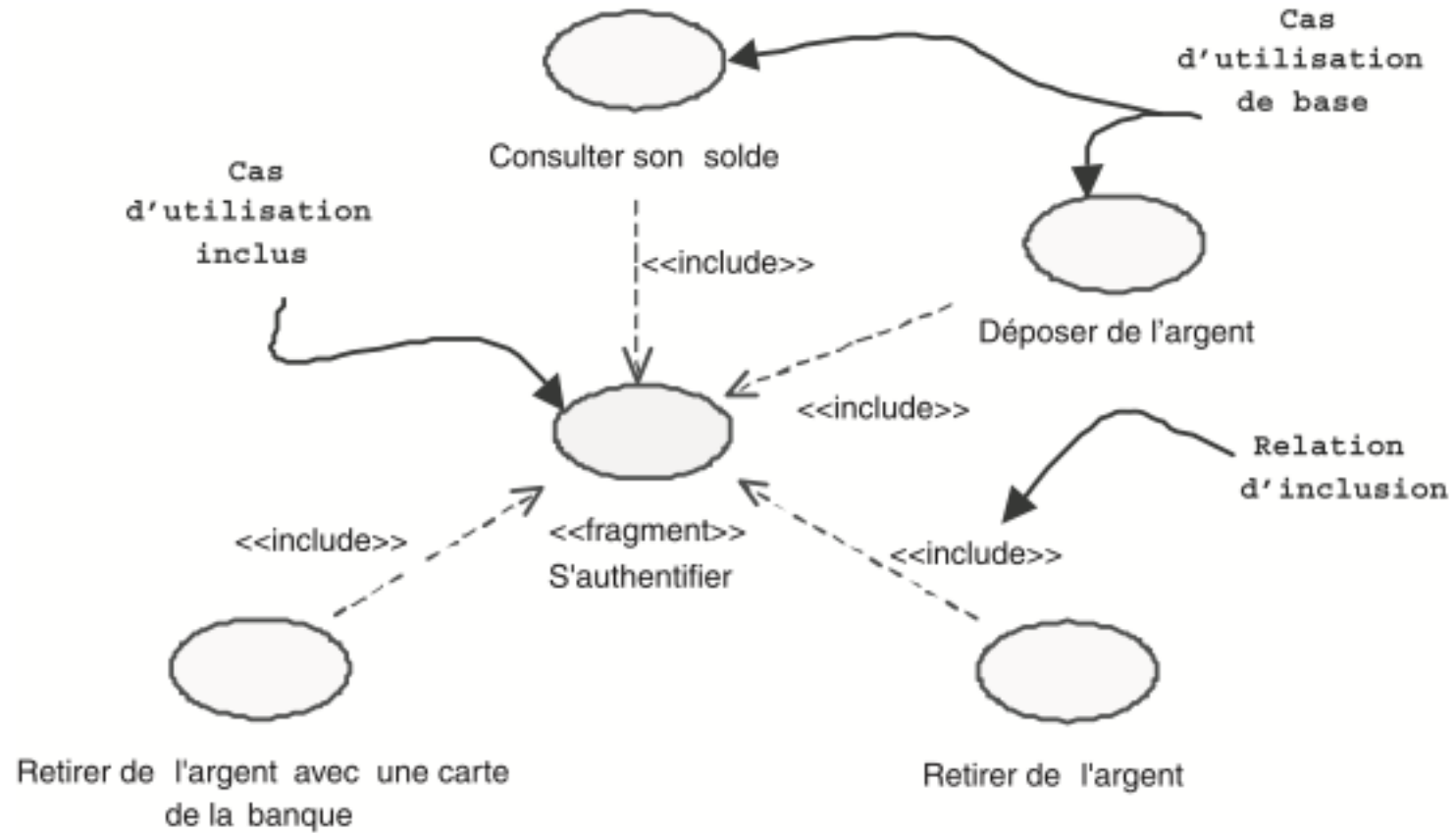
- Définir des relations entre les CU :
 - Inclusion
 - Extension
 - Généralisation
- Regroupe les CU en groupe fonctionnel cohérents (packages)

- Certains **enchainements sont communs** à plusieurs CU et peuvent constituer un nouveau CU (également appelé **fragments**)
- Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend de B
- Notation : flèche de dépendance + mot-clé `<<include>>`

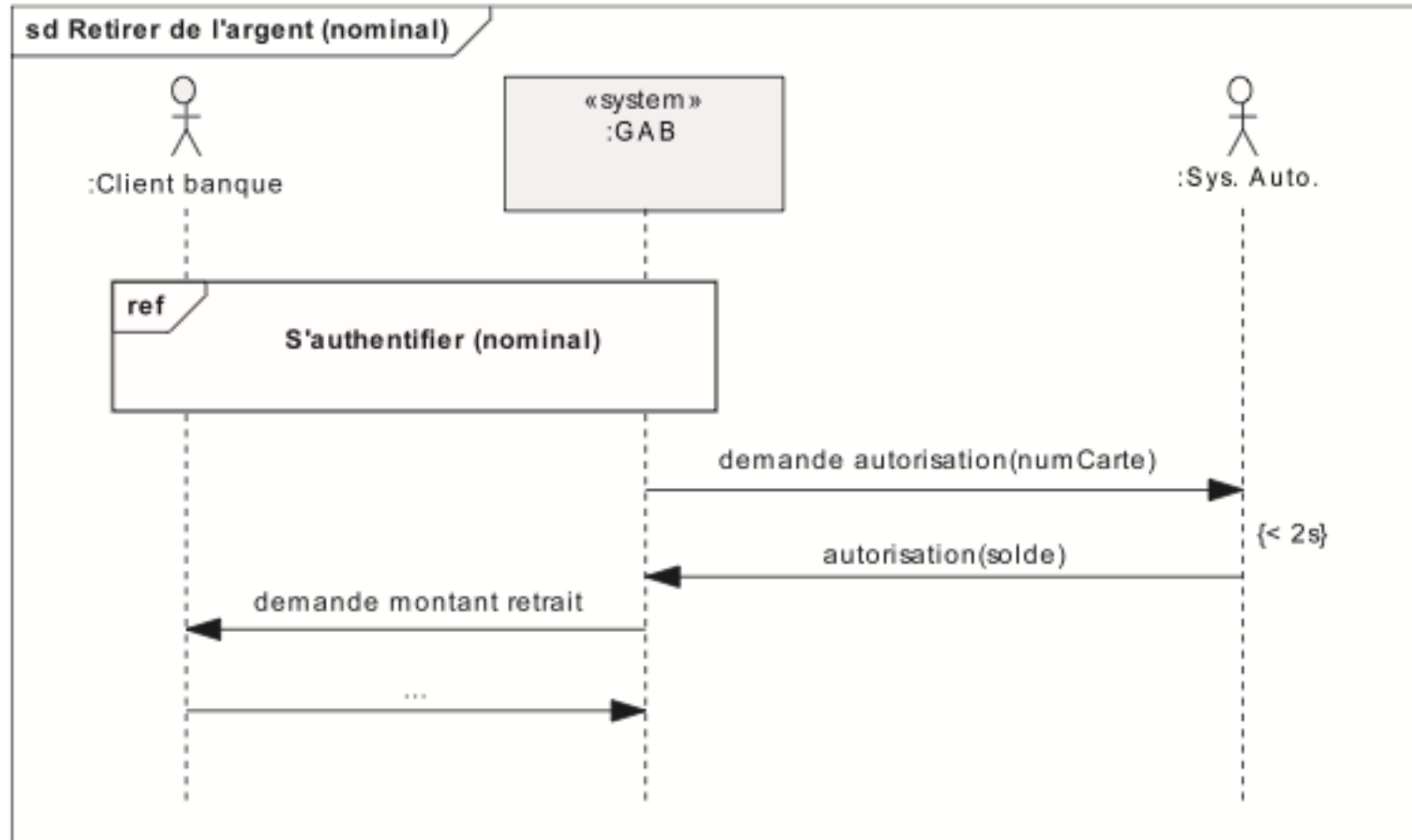
Exercice

- Identifier les acteurs du GAB
- Identifier les cas d'utilisation
- Réaliser le digramme des CU
- Identifier les acteurs principaux et secondaires et compléter le diagramme
- Faire une description textuelle du cas d'utilisation *retirer de l'argent*
- Dessiner le diagramme de séquence système qui décrit le scénario principal du CU retirer de l'argent
- Dessiner le diagramme d'activité du CU retirer de l'argent
- Proposer un diagramme de CU faisant apparaître le fragment **s'authentifier** et ses dépendances

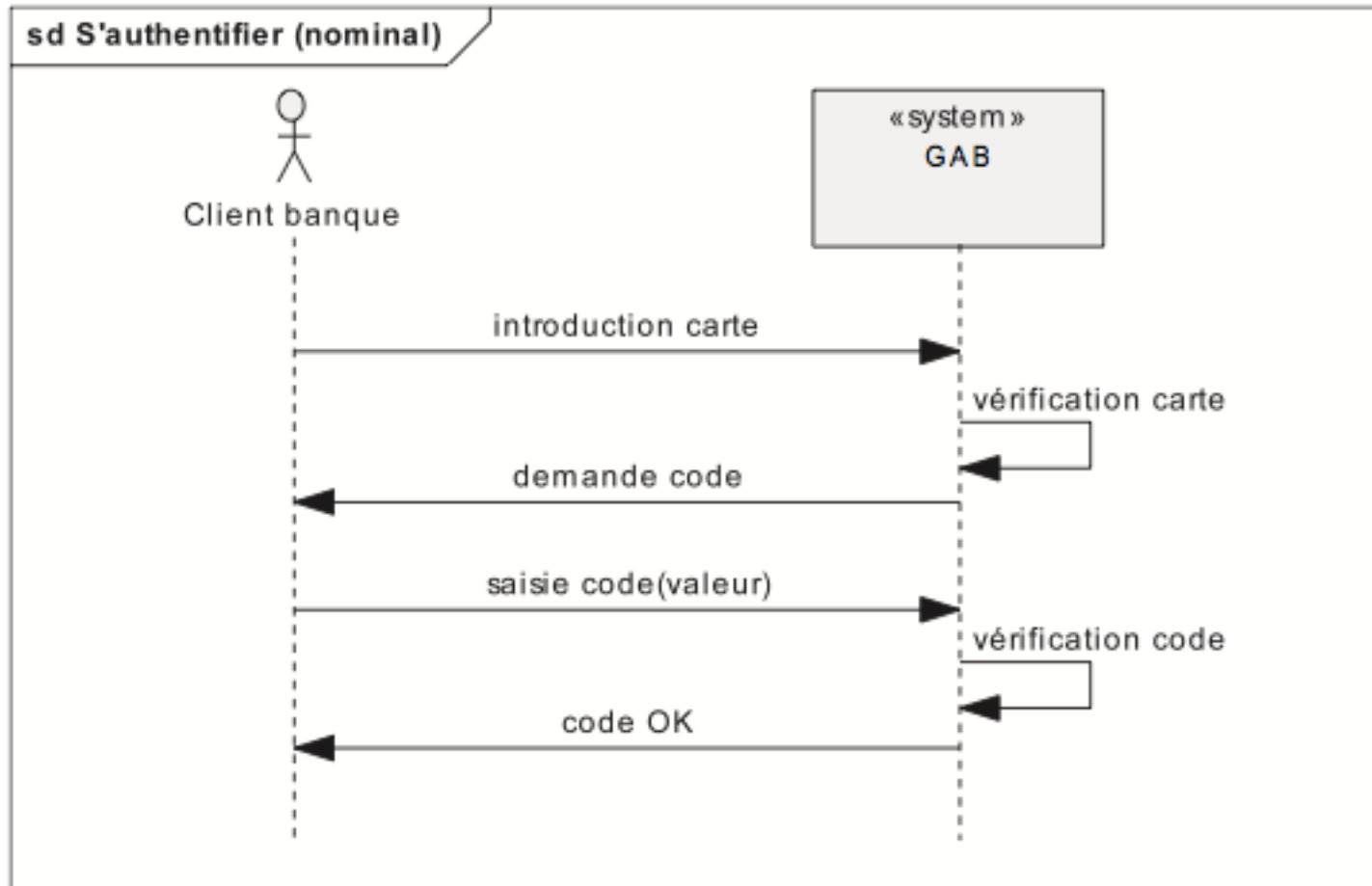
Solution



Impact sur le diagramme de séquence système



Impact sur le diagramme de séquence système

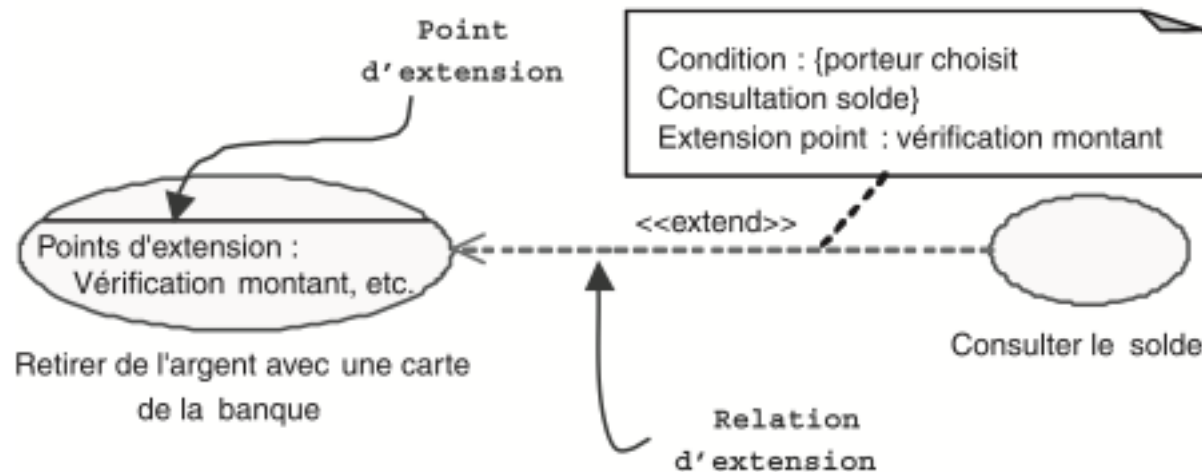


- Un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B.
- Exécuter B peut éventuellement entraîner l'exécution de A : contrairement à l'inclusion, l'extension est **optionnelle**.
- L'extension peut intervenir à un point précis du cas étendu. Ce point s'appelle **le point d'extension**

- **Consulter solde** peut être une **extension** du CU **retirer de l'argent avec carte de la banque**
- Utile si un client de la banque veut connaître son solde avant de retirer de l'argent : cette relation a du sens du point de vue métier

Exemple

- **Consulter solde** peut être une **extension** du CU **retirer de l'argent avec carte de la banque**



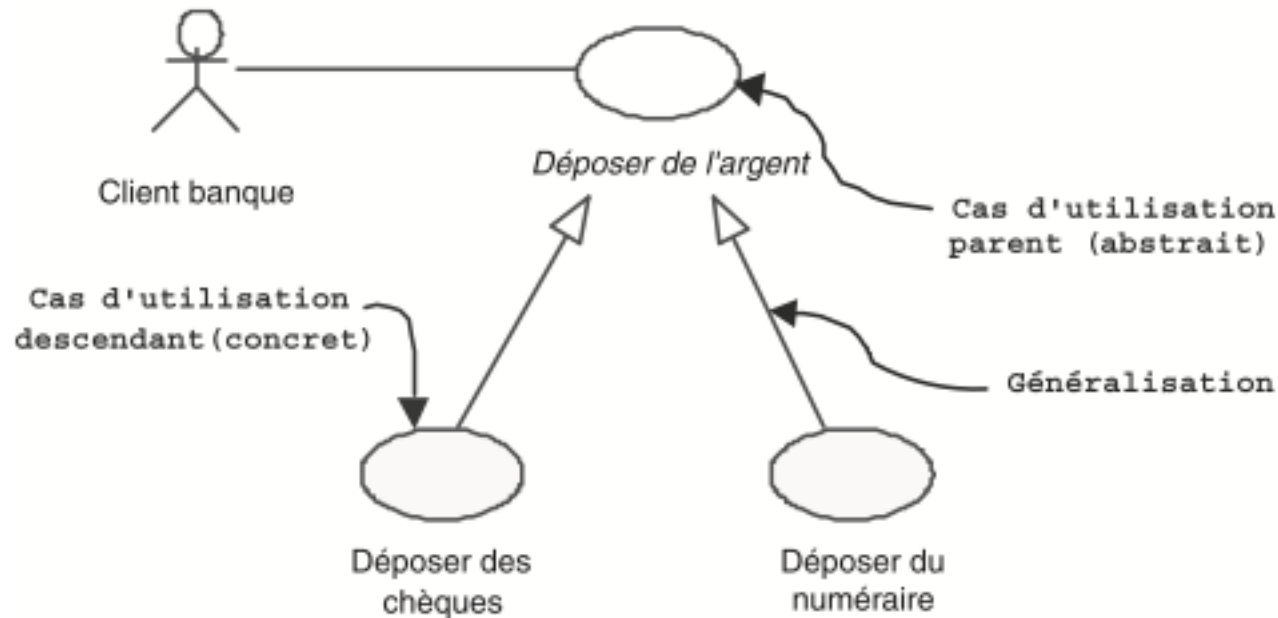
- Un cas A est une **généralisation** d'un cas B si B est un **cas particulier** (= une spécialisation) de A
- Se traduit généralement par la notion **d'héritage** lors de l'implémentation en langage orienté objet
- Pour traduire une **unité fonctionnelle**, pas une unité de traitements

Exemple

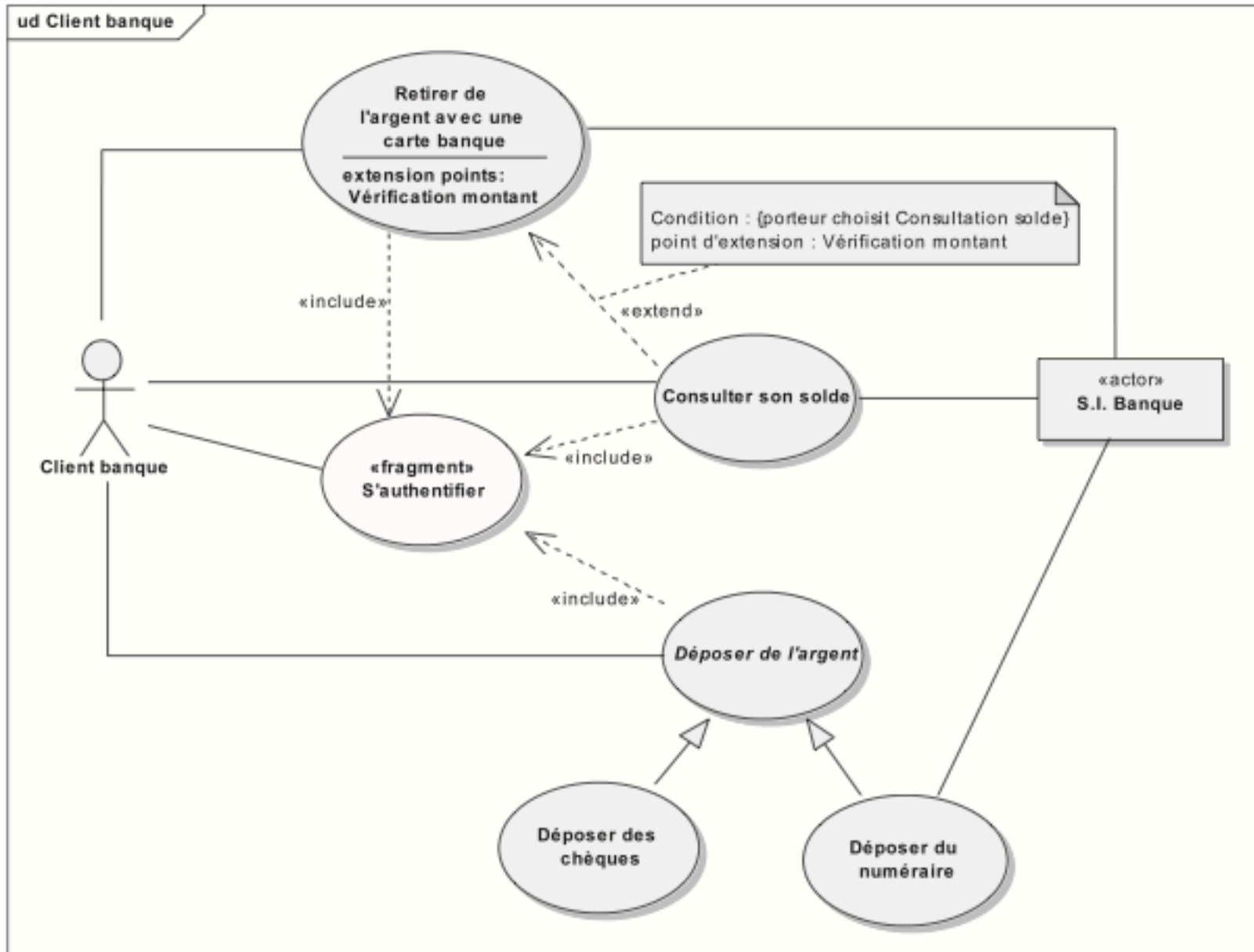
- **Déposer liquide et déposer chèque** peuvent être vues comme une **spécialisation** du CU **déposer de l'argent**

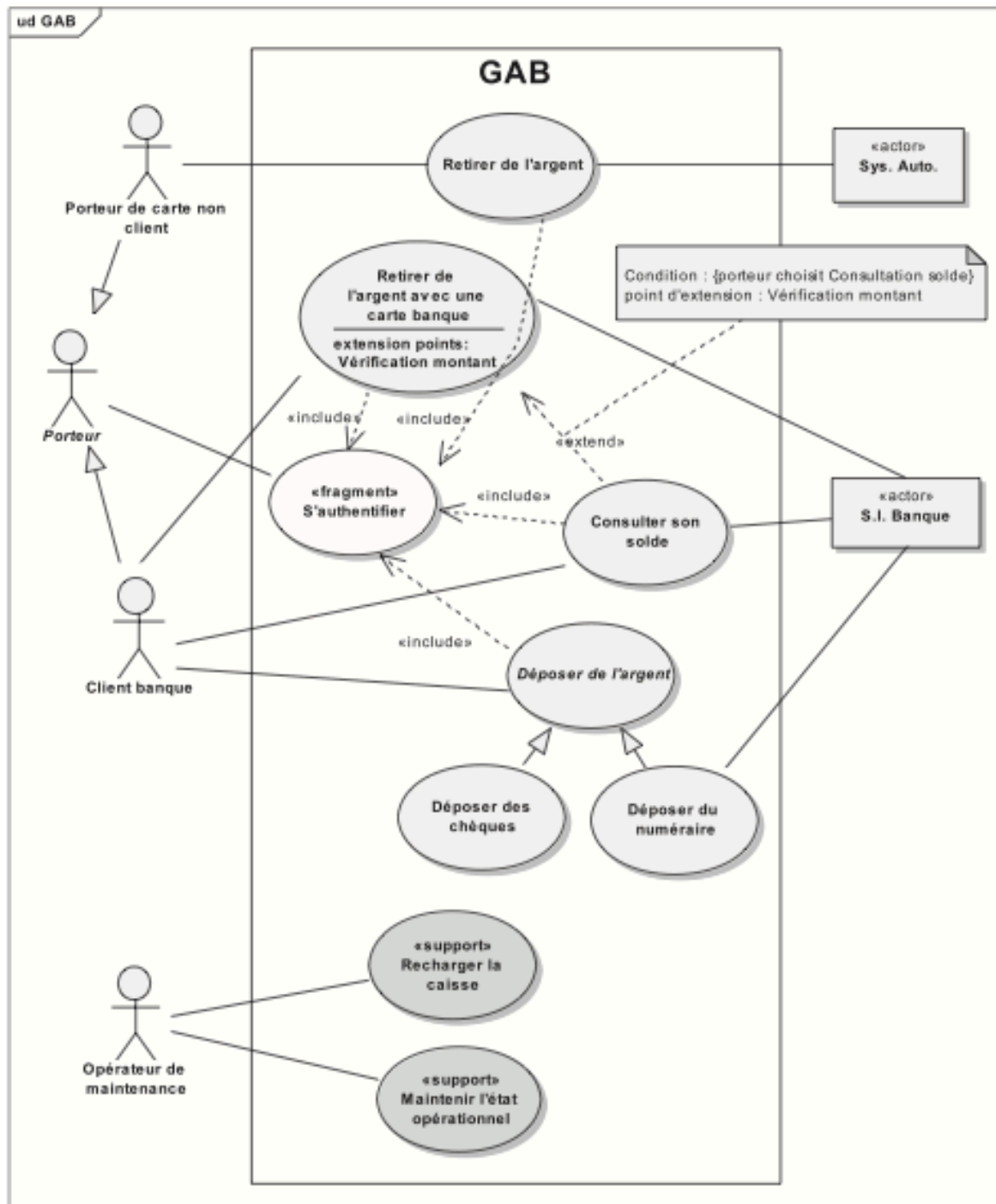
Exemple

- **Déposer liquide** et **déposer chèque** peuvent être vus comme une **spécialisation** du CU **déposer de l'argent**



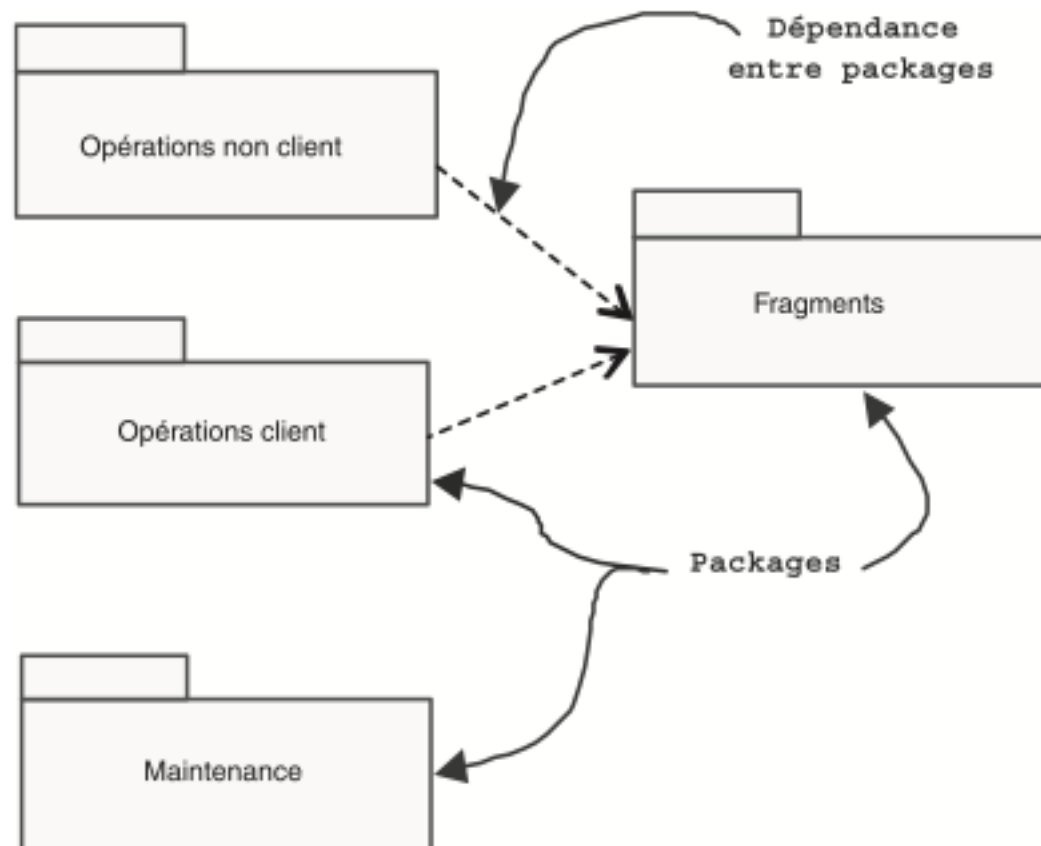
Les CU du Client banque



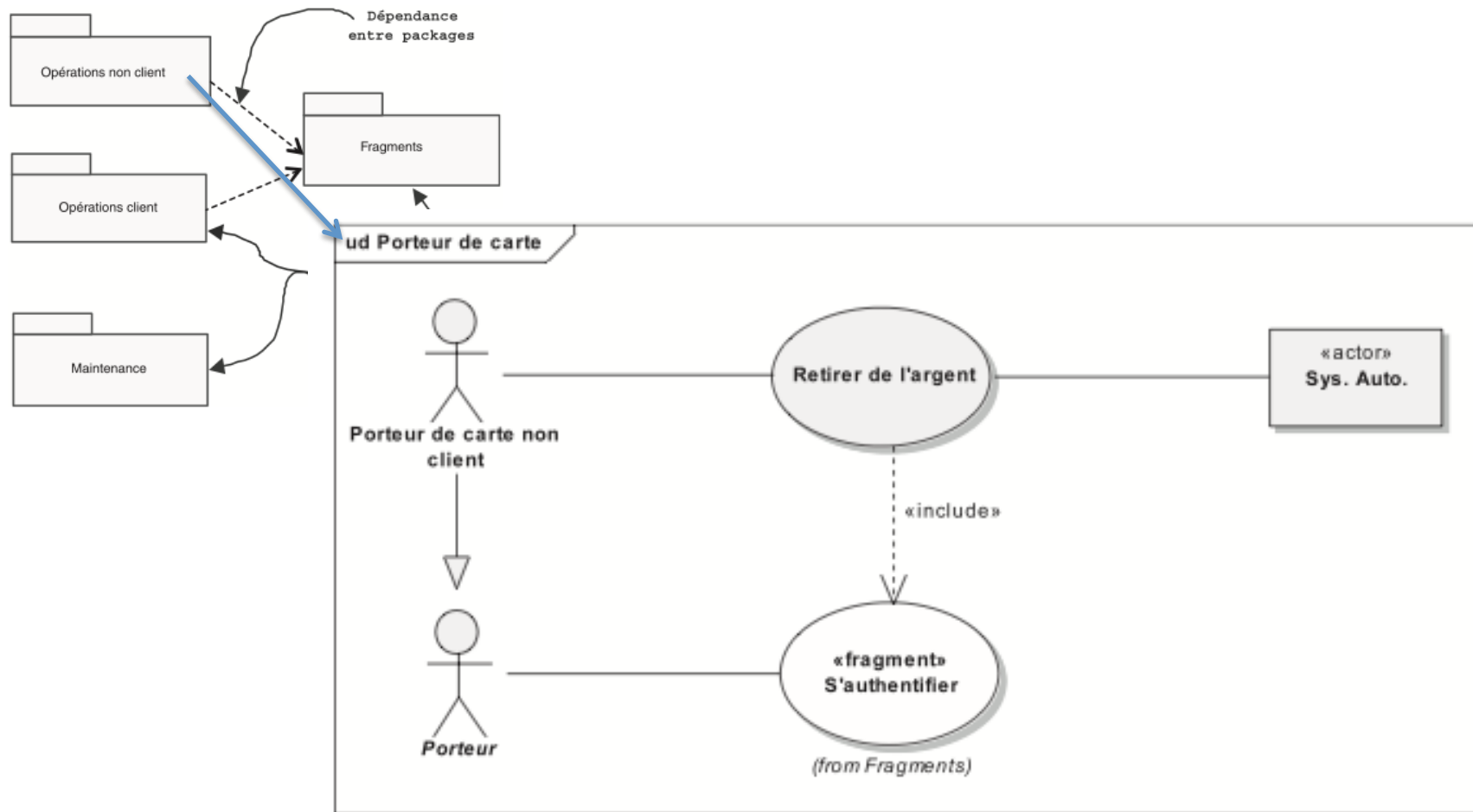


Structuration par packages

- Pour regrouper les CU au sein d'entités cohérentes
 - regroupement par acteur ou par fonctionnalité par exemple



Exemple



- Livres
 - **UML 2 Par la pratique, Etudes de cas et exercices corrigés**, *Pascal Roques*, Eyrolles
 - **Applying UML and Patterns: an introduction to Object-Oriented Analysis and Design and Iterative Development**, *Craig Larman*, Prentice Hall
 - **UML2 de l'apprentissage à la pratique**, *Laurent Audibert*, Ellipses
 - **Writing Effective Use Cases**, *Alistair Cockburn*, Addison-Wesley
- Site internet
 - <http://uml.developpez.com/cours>
 - <http://laurent-audibert.developpez.com/Cours-UML/>
 - COMPLETER