# UML2 (Unified Modeling Language)

UML est une notation graphique standardisée (par Object Management Group en 1997) conçue pour représenter, spécifier, construire et documenter un système. UML permet de construire plusieurs modèles d’un système : certains montrent le système du point de vue des utilisateurs, d’autres montrent sa structure interne, d’autres encore en donnent une vision globale ou détaillée. Les modèles se complètent et peuvent être assemblés. Ils sont élaborés tout au long du cycle de vie du développement d’un système (depuis le recueil des besoins jusqu’à la phase de conception).

UML est aujourd’hui un outil d’analyse et de communication incontournable utilisé sur des centaines de projets de par le monde.

**Pourquoi modéliser ?**

Un modèle est une représentation simplifiée d’une réalité. Il permet de capturer des aspects pertinents pour répondre à un objectif défini *a priori.*

**Les modèles ont différents usages :**

* Ils servent à circonscrire des systèmes complexes pour les dominer. Par exemple, il est inimaginable de construire une fusée sans passer par une modélisation permettant de tester les réacteurs, les procédures de sécurité, l’étanchéité de l’ensemble, etc.
* Ils optimisent l’organisation des systèmes. La modélisation da la structure d’une entreprise en divisions, départements, services, etc. permet d’avoir une vision simplifiée du système et par là même d’en assurer une meilleure gestion
* Ils permettent de se focaliser sur des aspects spécifiques d’un système sans s’embarrasser des données non pertinentes. Si l’on s’intéresse à la structure d’un système afin de factoriser ses composants, il est inutile de s’encombrer de ses aspects dynamiques. En utilisant, par exemple, le langage SYSML, on s’intéressera à la description statique (*via* le diagramme de blocs) sans se soucier des autres vues.
* Ils permettent de décrire avec précision et complétude les besoins sans forcément connaître les détails du système.
* Ils facilitent la conception d’un système, avec notamment la réalisation de maquette approximative, à échelle réduite, etc.
* Ils permettent de tester une multitude de solutions à moindre coût et dans des délais réduits et de sélectionner celle qui résout les problèmes posés.

# Les diagrammes UML

Que représenter ?





**Un projet comprend divers intervenants :**

Il est développé pour répondre à des besoins. Celui qui commande le logiciel est le maître d’ouvrage. Celui qui réalise le logiciel est le maître d’œuvre.

Le maître d’ouvrage intervient constamment au cours du projet, notamment pour :

• définir et exprimer les besoins ;

• valider les solutions proposées par le maître d’œuvre ;

• valider le produit livré.

Le diagramme des cas d’utilisation est suffisamment accessible, même pour un nom informaticien, pour servir de support de communication entre les différents acteurs du projet.

## Étude de cas

Cette étude de cas concerne un système simplifié de Guichet Automatique de Banque (GAB). Le GAB offre les services suivants :

1. Distribution d’argent à tout Porteur de carte de crédit, via un lecteur de carte et un distributeur de billets.

2. Consultation de solde de compte, dépôt en numéraire (espèce) et dépôt de chèques pour les clients porteurs d’une carte de crédit de la banque adossée au GAB.

De plus:

3. Toutes les transactions sont sécurisées.

4. Il est parfois nécessaire de recharger le distributeur, etc.

## Démarche

À partir de cet énoncé, nous allons :

1. Identifier les acteurs
2. Identifier les cas d’utilisation
3. Établir le diagramme des cas d’utilisation
4. Décrire textuellement les cas d’utilisation (établir des scénarios)
5. Compléter les descriptions par des diagrammes dynamiques
6. Organiser et structurer les cas d’utilisation (les regrouper dans des packages)

### Identifier les acteurs

Quelles sont les entités externes qui interagissent directement avec le GAB ?

Analysons les phrases de l’énoncé :

* Phrase 1 : Porteur de carte
* Phrase 2 : Client banque
* Phrase 3 : S.I banque, Système automatique
* Phrase 4 : opérateur de maintenance

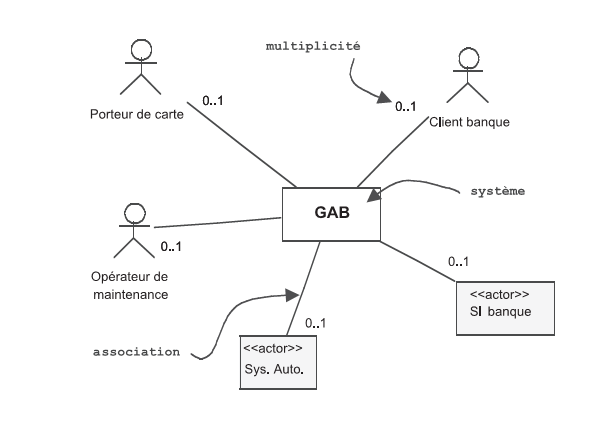


Figure 1:Diagramme de contexte statique du GAB (ne fait pas parti d’UML mais dans la pratique se montre très utile)

### Identifier les cas d’utilisation

Reprenons un à un les cinq acteurs et listons les différentes façons qu’ils ont d’utiliser le GAB :

Porteur de carte :

• Retirer de l’argent

Client banque :

• Retirer de l’argent

• Consulter le solde de son compte courant

• Déposer de l’argent (du numéraire ou des chèques)

Opérateur de maintenance :

• Recharger le distributeur

• Maintenir l’état opérationnel (récupérer les cartes avalées, récupérer les chèques déposés, remplacer le ruban de papier, etc.)

Système d’autorisation (Sys. Auto.) :

• Néant

Système d’information (SI) banque :

• Néant

**À retenir :** ACTEUR PRINCIPAL OU SECONDAIRE

Contrairement à ce que l’on pourrait croire, tous les acteurs n’utilisent pas forcément le système ! Nous appelons acteur principal celui pour qui le cas d’utilisation produit un résultat observable. Par opposition, nous qualifions d’acteurs secondaires les autres participants du cas d’utilisation. Les acteurs secondaires sont souvent sollicités pour des informations complémentaires ; ils peuvent uniquement consulter ou informer le système lors de l’exécution du cas d’utilisation.

### Établir le diagramme des cas d’utilisation

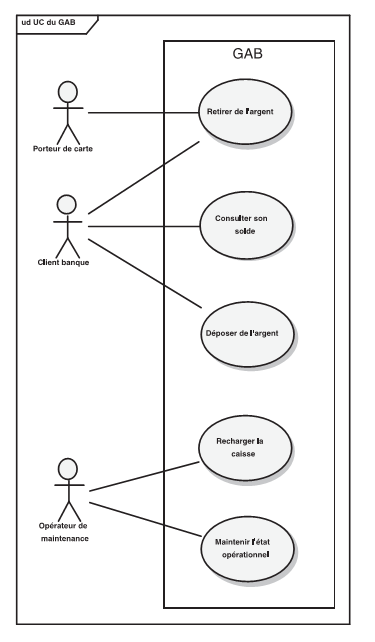


Figure 2: Diagramme préliminaire de cas d’utilisation

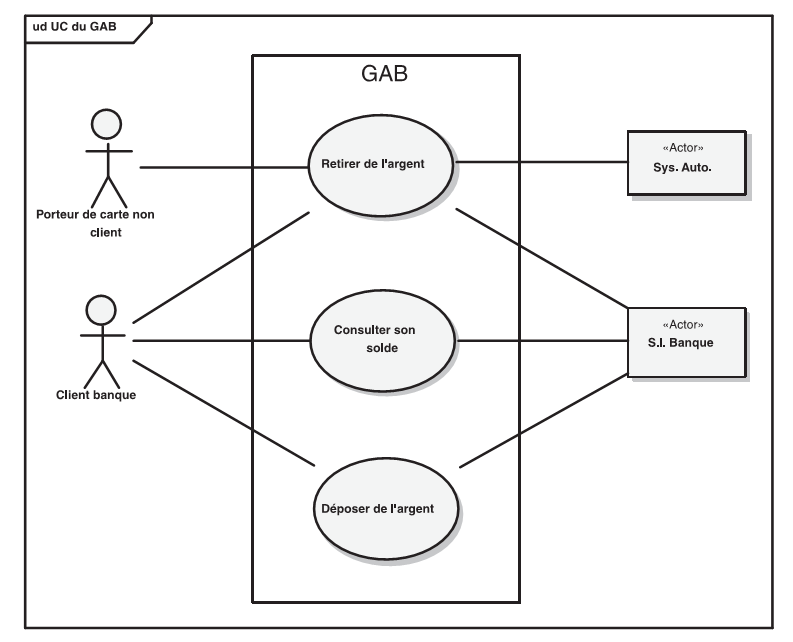


Figure 3 : Diagramme de cas d’utilisation du GAB

### Décrire textuellement les cas d’utilisation (décrire les scénarios)

La description textuelle est obligatoire pour chaque cas d’utilisation.

Exemple : Décrire la partie obligatoire du cas d’utilisation RETIRER DE L’ARGENT (pour l’acteur non client de la banque).

**Sommaire** d’identification

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 : 02/03/2013 Date de mise à jour : 06/11/2013

Version : 5.0 Responsable : M. SALL

**Description des scénarios**

**Préconditions**

• 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.

**Post conditions**

La caisse du GAB contient moins de billets qu’au début du cas d’utilisation (le nombre de billets manquants est fonction du montant du retrait).

Une transaction de retrait a été enregistrée par le GAB avec toutes les informations pertinentes (montant, numéro de carte, date, etc.). Les détails de la transaction doivent être enregistrés aussi bien en cas de succès que d’échec.

**Enchaînements alternatifs**

A1 : code d’identification provisoirement erroné

L’enchaînement A1 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.

A2 : montant demandé supérieur au solde hebdomadaire

L’enchaînement A2 démarre au point 10 du scénario nominal.

11. Le GAB indique au Porteur de carte 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 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

L’enchaînement E1 démarre au point 2 du scénario nominal.

3. 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

L’enchaînement E2 démarre au point 5 du scénario nominal.

6. Le GAB indique au Porteur de carte que le code est erroné, pour la troisième fois.

7. Le GAB confisque la carte.

8. Le Système d’autorisation est informé ; le cas d’utilisation se termine en échec.

E3 : retrait non autorisé

L’enchaînement E3 démarre au point 6 du scénario nominal.

7. Le Système d’autorisation interdit tout retrait.

8. Le GAB éjecte la carte ; le cas d’utilisation se termine en échec.

E4 : carte non reprise

L’enchaînement E4 démarre au point 13 du scénario nominal.

14. Au bout de 10 secondes, le GAB confisque la carte.

15. Le Système d’autorisation est informé ; le cas d’utilisation se termine en échec.

E5 : billets non pris

L’enchaînement E5 démarre au point 15 du scénario nominal.

16. Au bout de 10 secondes, le GAB reprend les billets.

17. Le cas d’utilisation se termine en échec.

E6 : annulation de la transaction

L’enchaînement E6 peut démarrer entre les points 4 et 12 du scénario nominal.

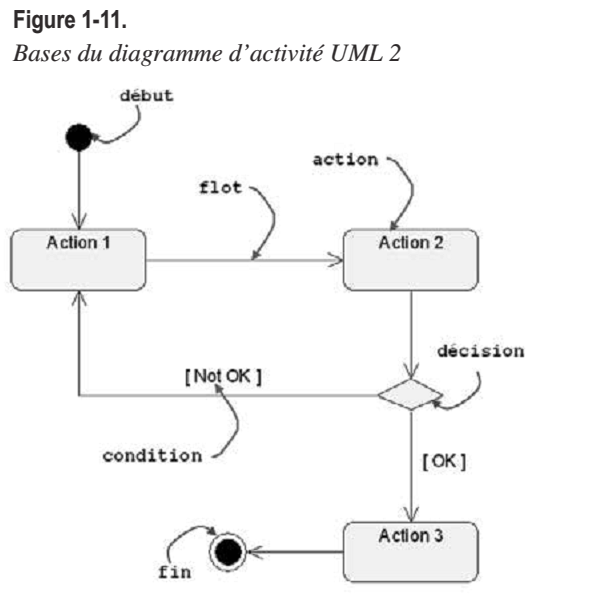
4 à 12. Le Porteur de carte demande l’annulation de la transaction en cours.

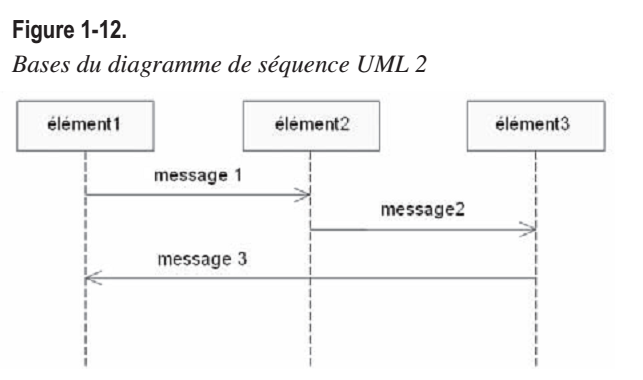
Le GAB éjecte la carte ; le cas d’utilisation se termine en échec.

### Compléter les descriptions par des diagrammes dynamiques

Pour documenter les cas d’utilisation, la description textuelle est indispensable, car elle seule permet de communiquer facilement avec les utilisateurs et de s’entendre sur la terminologie métier employée.

En revanche, le texte présente des désavantages puisqu’il est difficile de montrer comment les enchaînements se succèdent, ou à quel moment les acteurs secondaires sont sollicités. En outre, la maintenance des évolutions s’avère souvent fastidieuse. Il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques UML.





*Exercice : Réalisez un diagramme de séquence système qui décrit le scénario nominal du cas d’utilisation RETIRER DE L’ARGENT.*

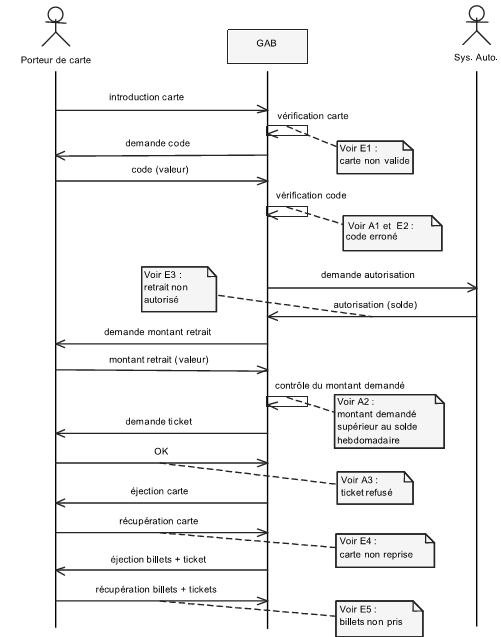


Figure 3:Diagramme de séquence système du scénario nominal de Retirer de l’argent

Exercice : Réalisez un diagramme d’activité qui décrit la dynamique du cas d’utilisation RETIRER DE L’ARGENT

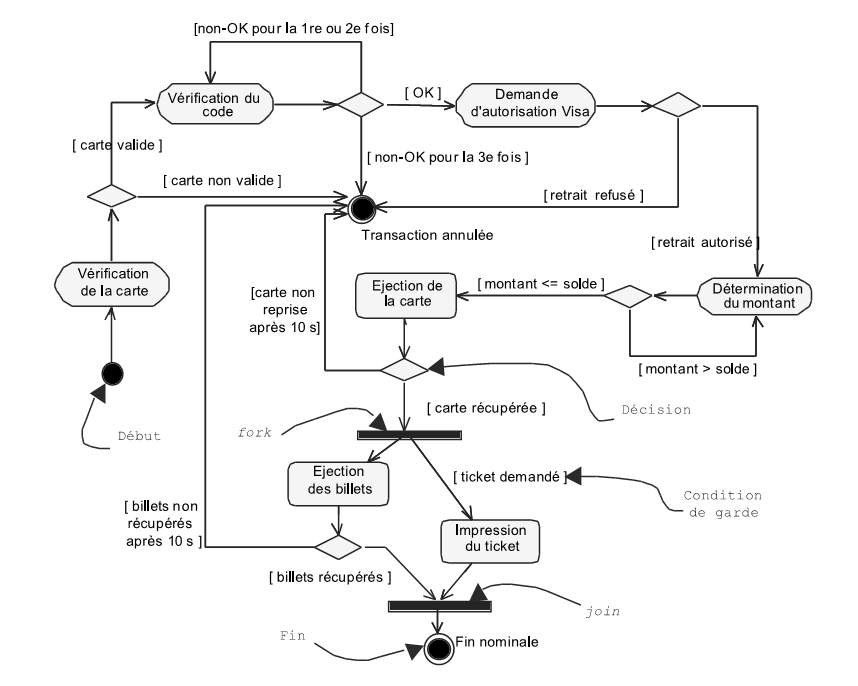


Figure 4: Diagramme d’activité de Retirer de l’argent

### Organiser et structurer les cas d’utilisation (lien entre cas d’utilisation, regroupement dans des packages)

Nous allons affiner nos diagrammes et nos descriptions.

Avec UML, il est en effet possible de détailler et d’organiser les cas d’utilisation de deux façons différentes et complémentaires :

• en ajoutant des relations d’inclusion, d’extension et de généralisation entre cas d’utilisation ;

• en les regroupant en packages, afin de définir des blocs fonctionnels de plus haut niveau.

#### Relation d’inclusion (obligatoire)

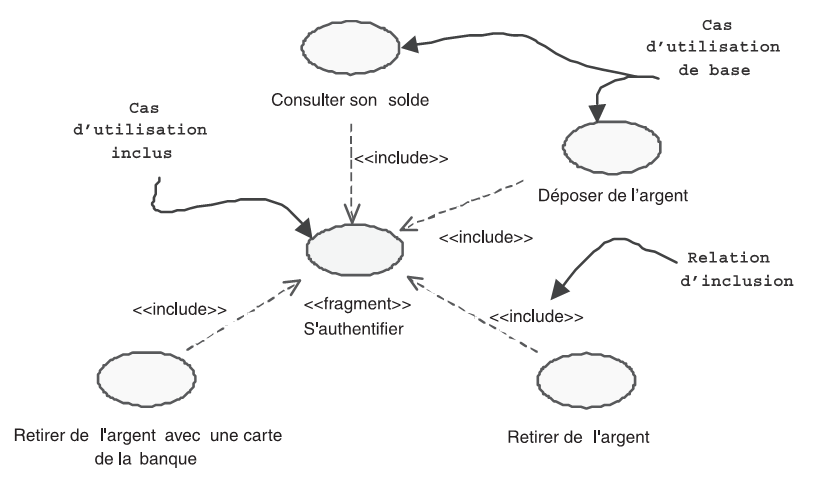


Figure 5: Relation d’inclusion entre cas d’utilisation

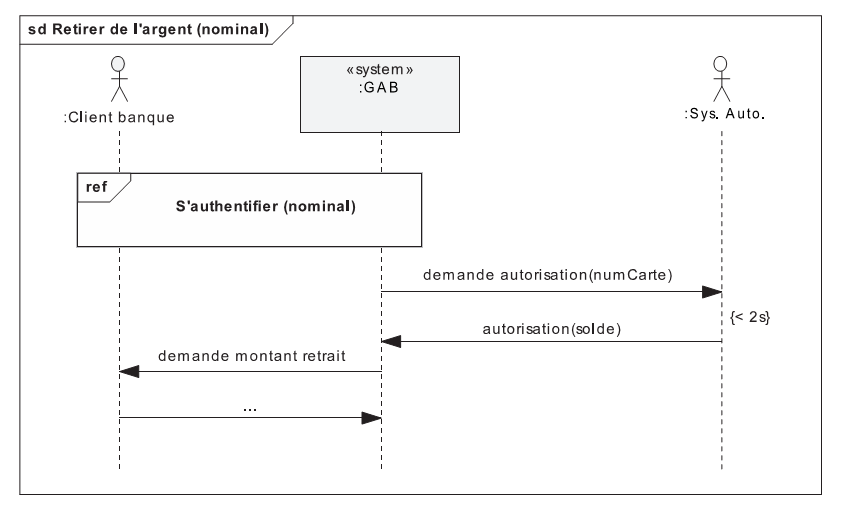


Figure 6: Nouveau diagramme de séquence système avec référence au cas inclus

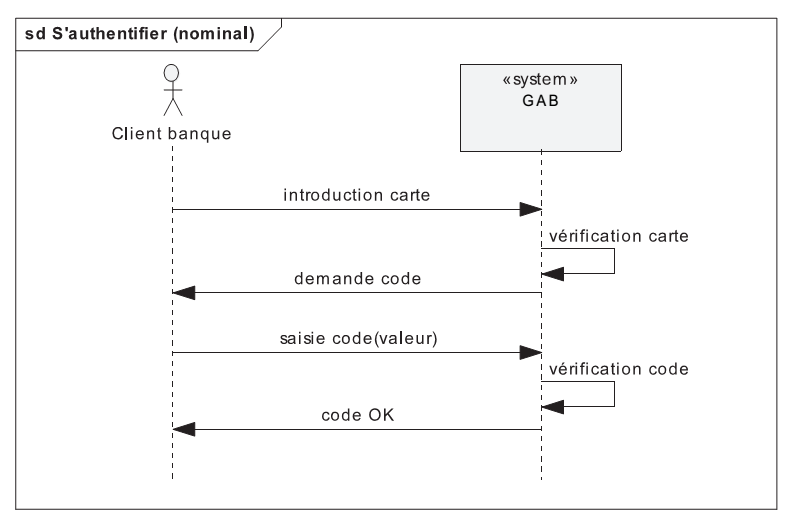


Figure 7 : Diagramme de séquence système du fragment référencé:

#### Relation d’extension (optionnel)

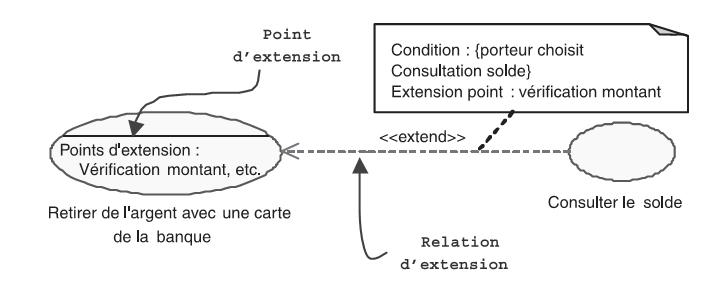


Figure 8: Relation d’extension entre cas d’utilisation

*« Consulter le solde »* peut venir s’intercaler à l’intérieur de « Retirer de l’argent » avec une carte de la banque, au point d’extension Vérification montant. Ce point d’extension doit être déclaré dans la description textuelle, par exemple en modifiant comme ceci l’enchaînement nominal :

...

7. Le SI banque donne son accord et indique le solde hebdomadaire.

8. Le GAB demande au Client banque de saisir le montant désiré du retrait.

Point d’extension : Vérification montant

9. Le Client banque saisit le montant désiré du retrait.

10. Le GAB contrôle le montant demandé par rapport au solde hebdomadaire.

...

#### Généralisation / spécialisation entre cas d’utilisation

Considérons le cas d’utilisation « Déposer de l’argent ». Il possède deux scénarios principaux : Déposer du numéraire et Déposer des chèques.

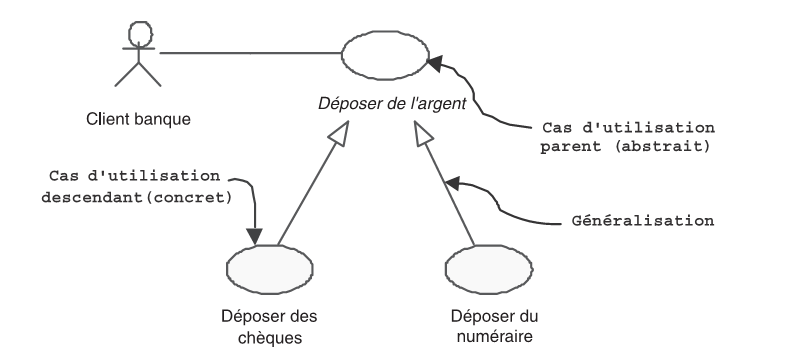


Figure 9: Relation de généralisation entre cas d’utilisation

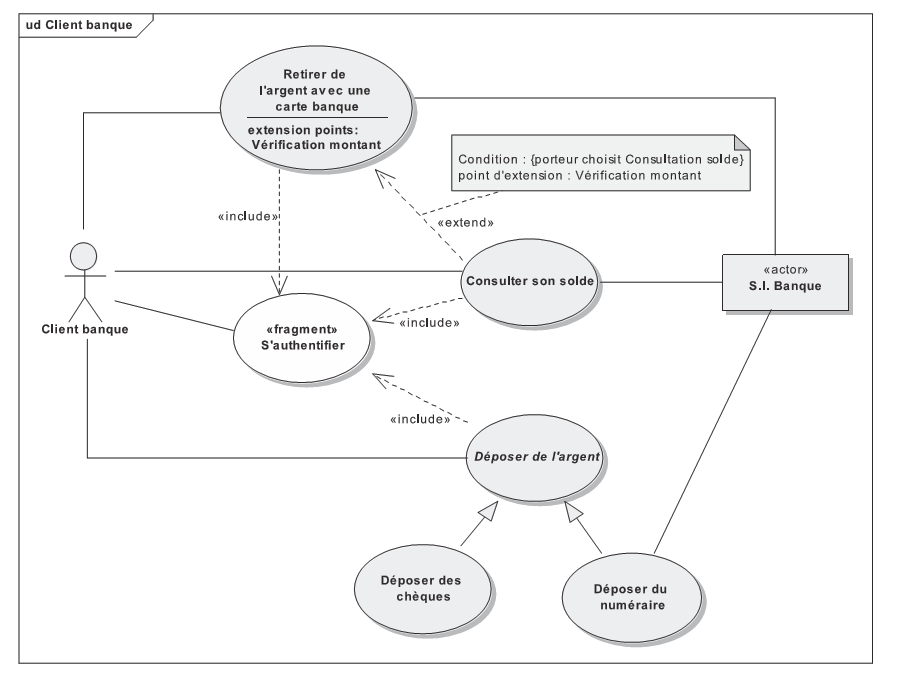


Figure 10: Cas d’utilisation du client banque

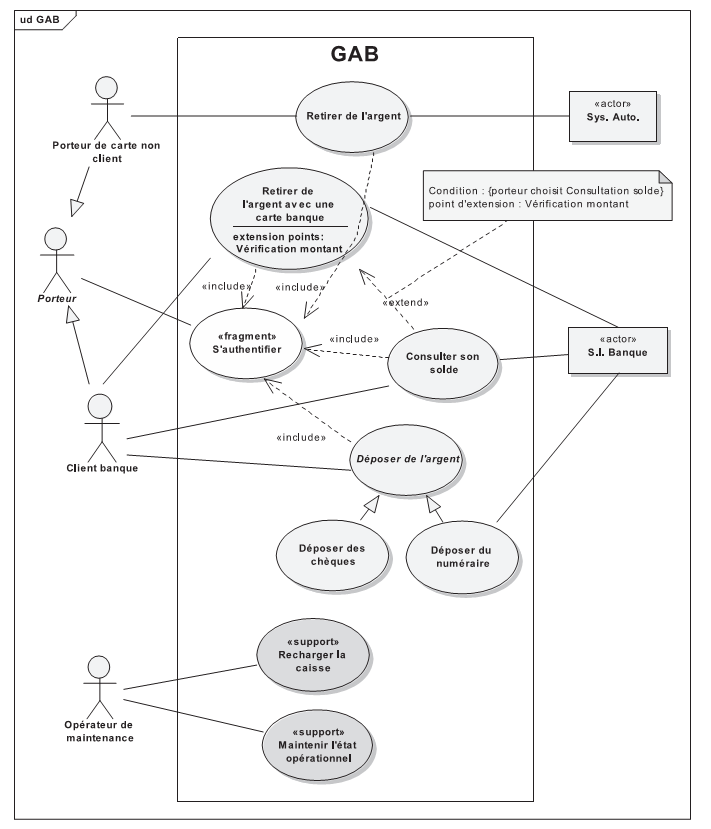
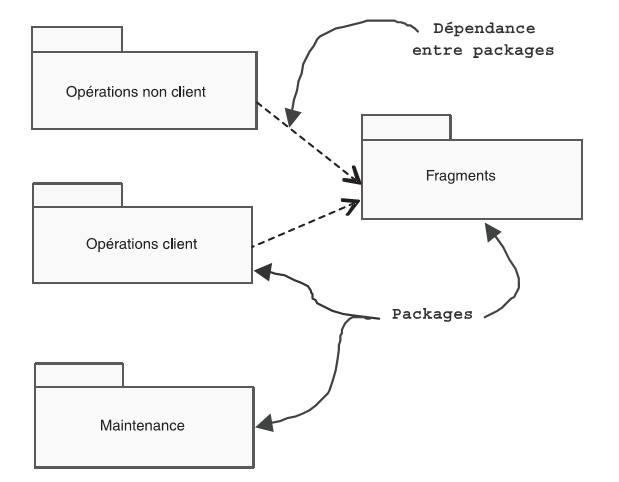


Figure 11:Diagramme de cas d’utilisation complet du GAB

#### Structuration des cas d’utilisation en packages(



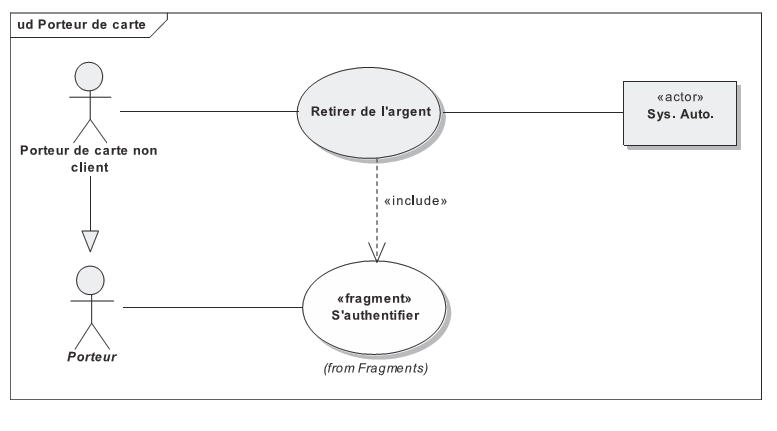


Figure 12: Diagramme de cas d’utilisation du package Opérations non client

Figure 13: Diagramme de packages des cas d’utilisation du GAB

### Dynamique globale : Interaction Overview Diagram (apport UML2)

On peut utiliser les avantages des diagrammes de séquences et d’activité dans un diagramme hybride : Interaction Overview Diagram.

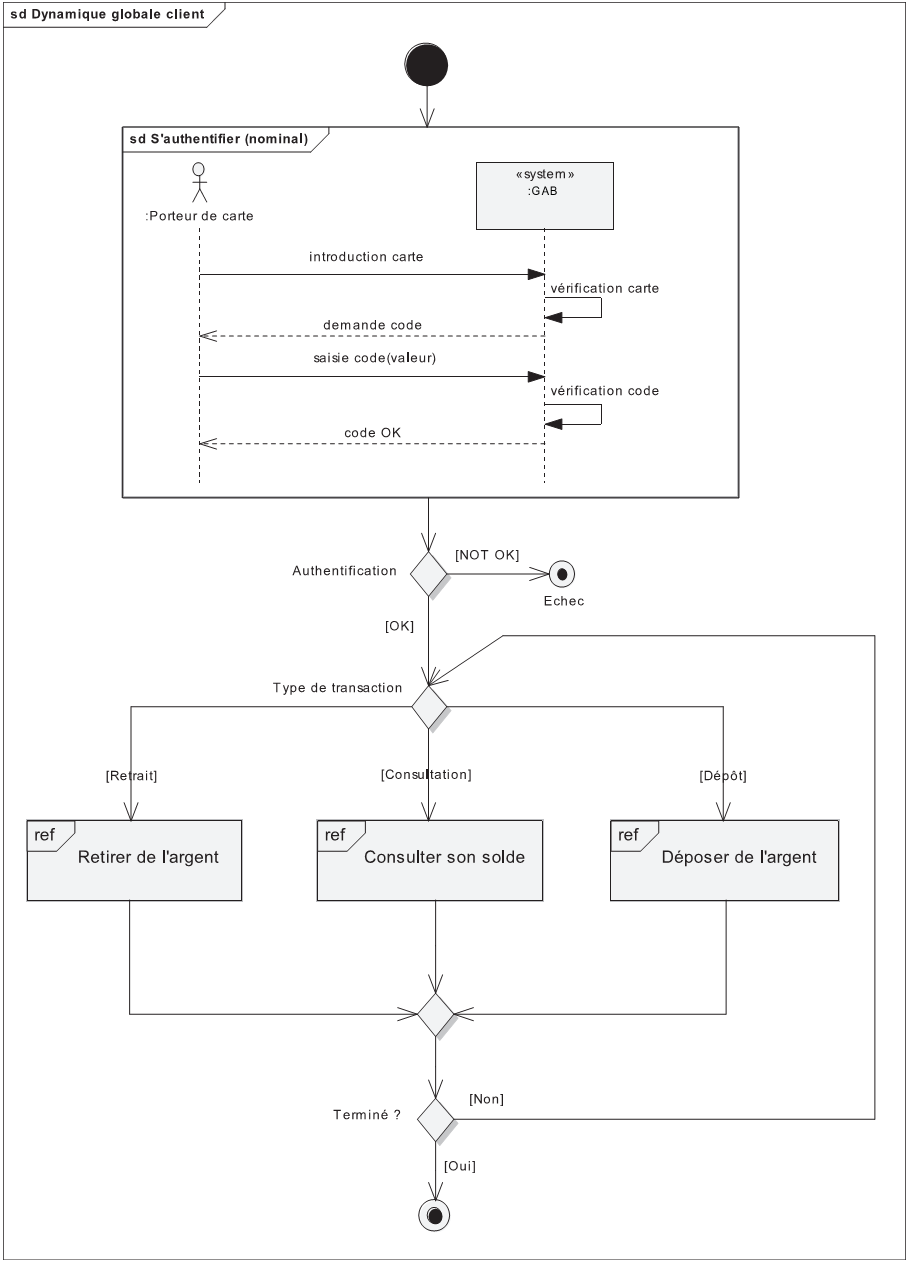


Figure 14: Vue d’ensemble des interactions du client banque