

Diagramme de contexte statique

Figure 1-4.

Diagramme de contexte statique du GAB

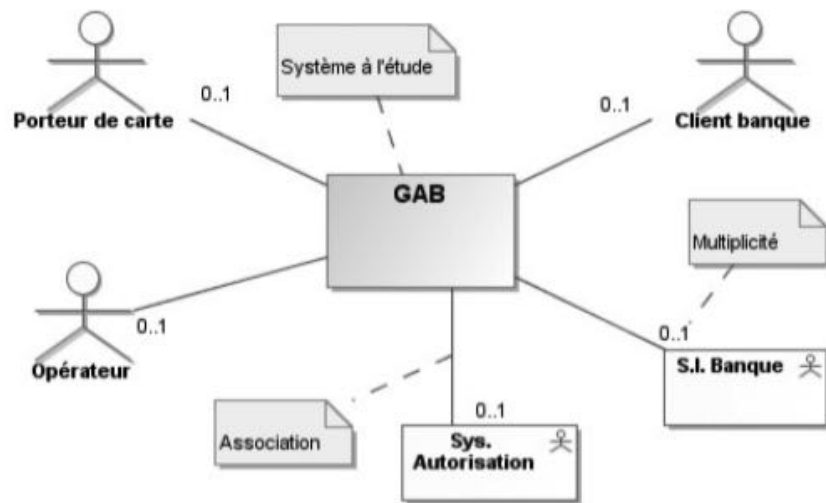


Figure 1-6.

Version plus élaborée du diagramme de contexte statique du GAB

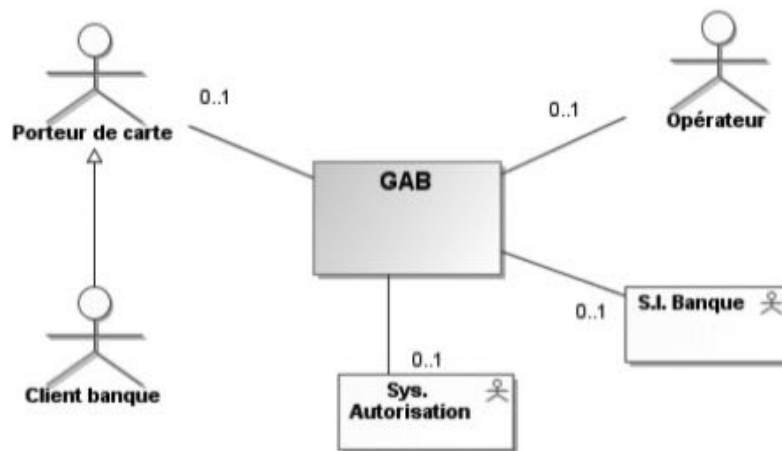


Diagramme de cas d'utilisation

Figure 1-8.

Version plus sophistiquée du diagramme de cas d'utilisation préliminaire

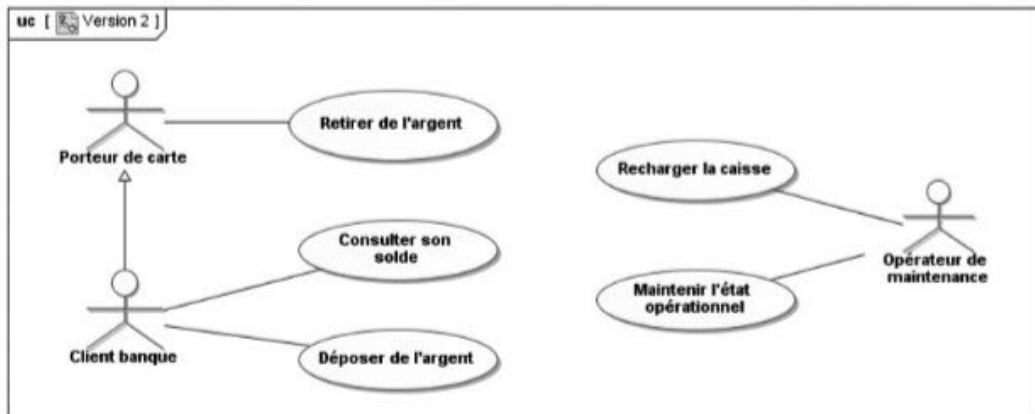


Figure 1-9.

Version simple du diagramme de cas d'utilisation complété

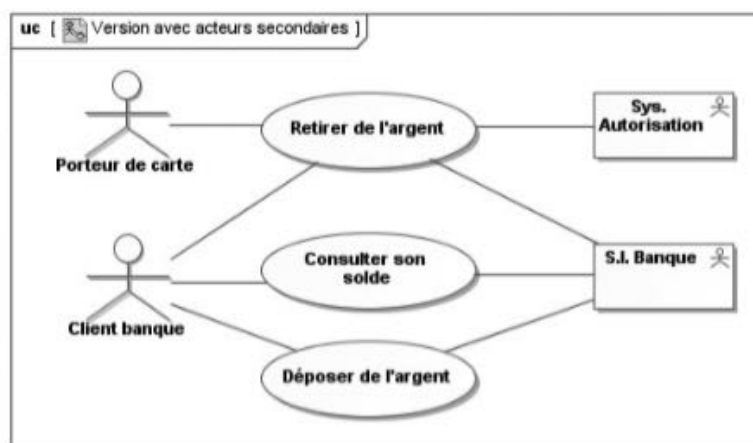
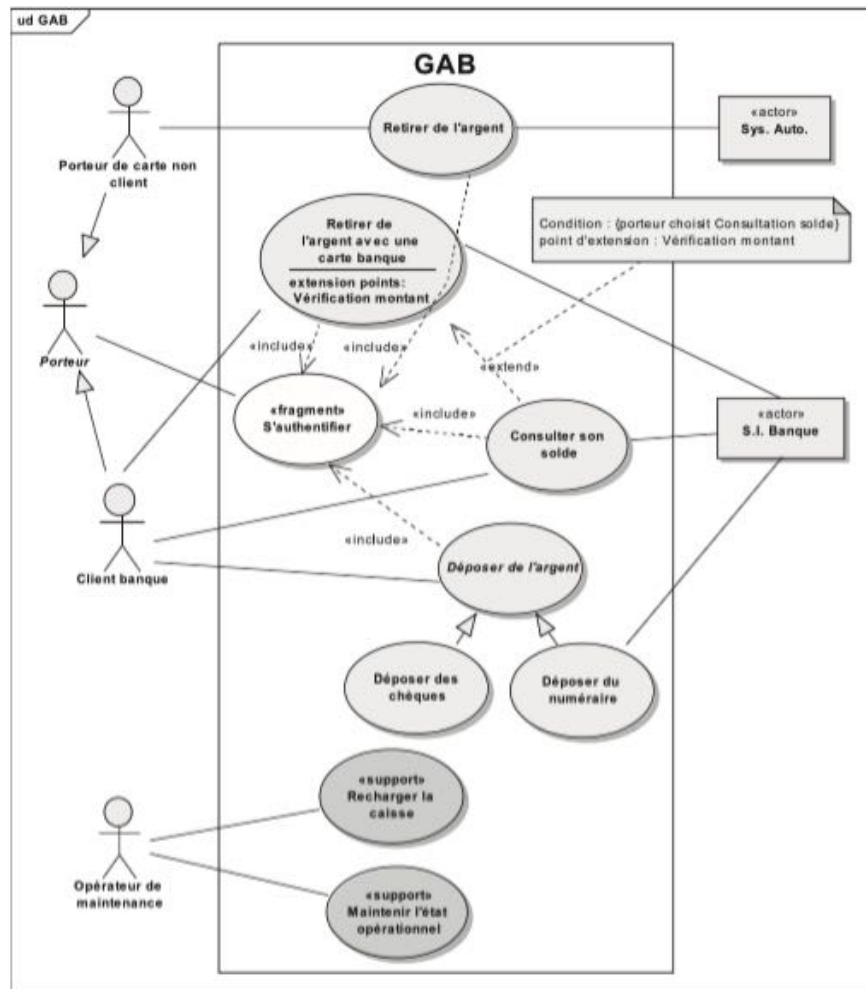


Figure 1-22.
Diagramme
de cas d'utilisation
complet du GAB



Description textuelle des cas d'utilisation

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 : 02/03/09

Date de mise à jour : 05/05/09

Version : 1.7

Responsable : Pascal Roques

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 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 crédit 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 crédit 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*

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 crédit 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 crédit hebdomadaire.

Le scénario nominal reprend au point 8.

Enchaînements d'erreurs

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.

Autre présentation des enchaînements alternatifs et d'erreur : indiquer les différentes alternatives par des lettres collées au chiffre du numéro de l'étape du scénario nominal concernée. Une version alternative de la solution précédente pourrait être alors :

2a. Carte illisible ou non valable :

Le GAB avertit le Porteur et éjecte la carte ;
le cas d'utilisation se termine en échec.

2b. Carte périmée :

Le GAB avertit le Porteur et confisque la carte ;
le cas d'utilisation se termine en échec.

Postconditions

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).

Exigences non fonctionnelles

Contraintes	Descriptif
Temps de réponse	L'interface du GAB doit réagir en l'espace de 2 secondes au maximum. Une transaction nominale de retrait doit durer moins de 2 minutes.
Concurrence	Non applicable (mono-utilisateur).
Disponibilité	Le GAB est accessible 7 jours sur 7, 24 h sur 24 (global). L'absence de papier pour imprimer les tickets ne doit pas empêcher les retraits.
Intégrité	Les interfaces du GAB doivent être très robustes pour prévenir le vandalisme.
Confidentialité	La comparaison du code d'identification saisi sur le clavier du GAB avec celui de la carte doit être fiable à 10^{-6} .

Besoins d'IHM

Les dispositifs d'entrée/sortie à la disposition du Porteur de carte doivent être :

- Un lecteur de carte bancaire.
- Un clavier numérique (pour saisir son code),
avec des touches « validation », « correction » et « annulation ».
- Un écran pour l'affichage des messages du GAB.
- Des touches autour de l'écran

Diagramme d'activité

Figure 1-14.
*Diagramme
d'activité de
Retirer de
l'argent*

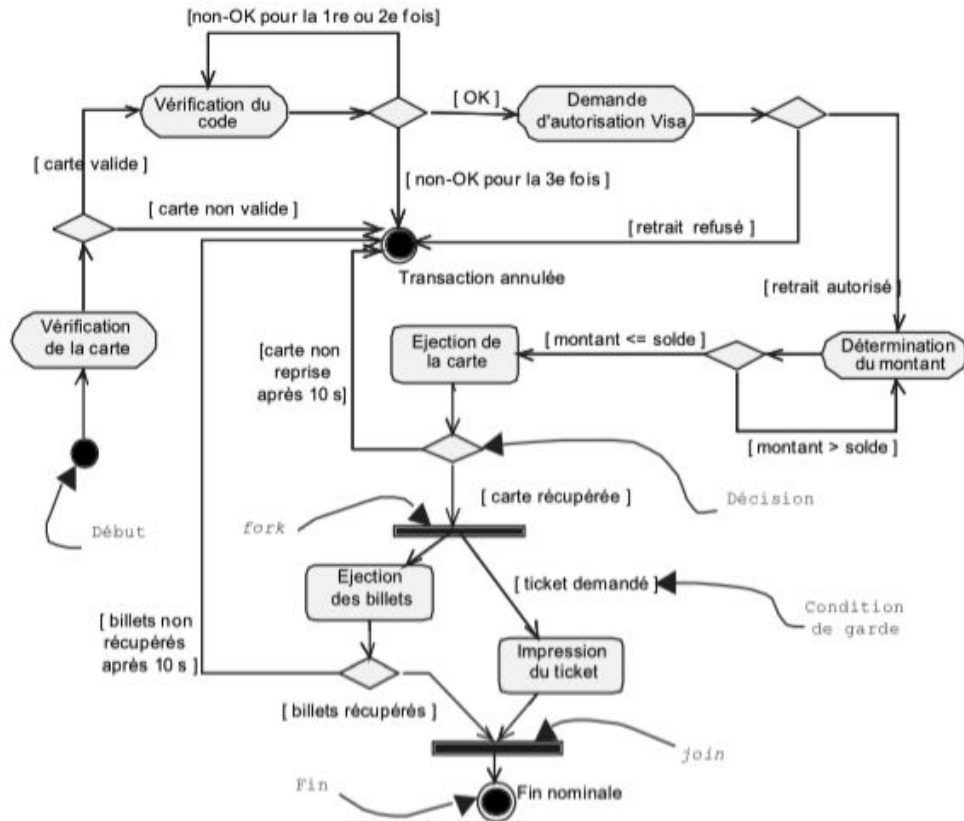


Diagramme de séquence système

Figure 1-15.

Diagramme de séquence système enrichi du scénario nominal de Retirer de l'argent

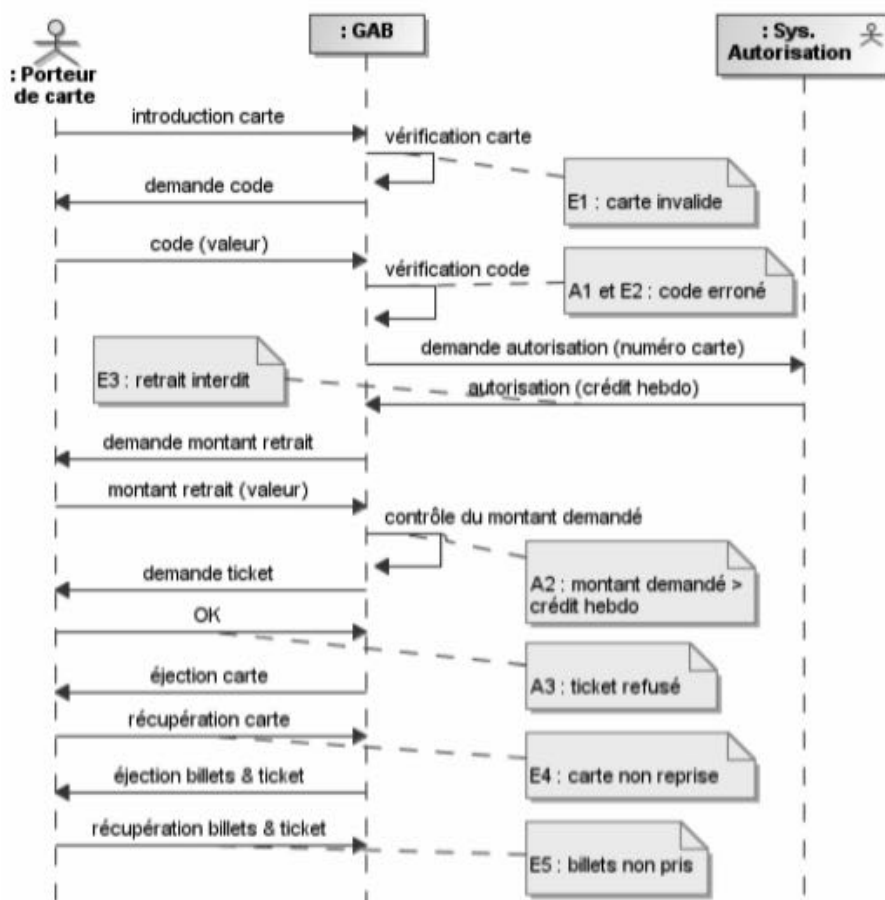
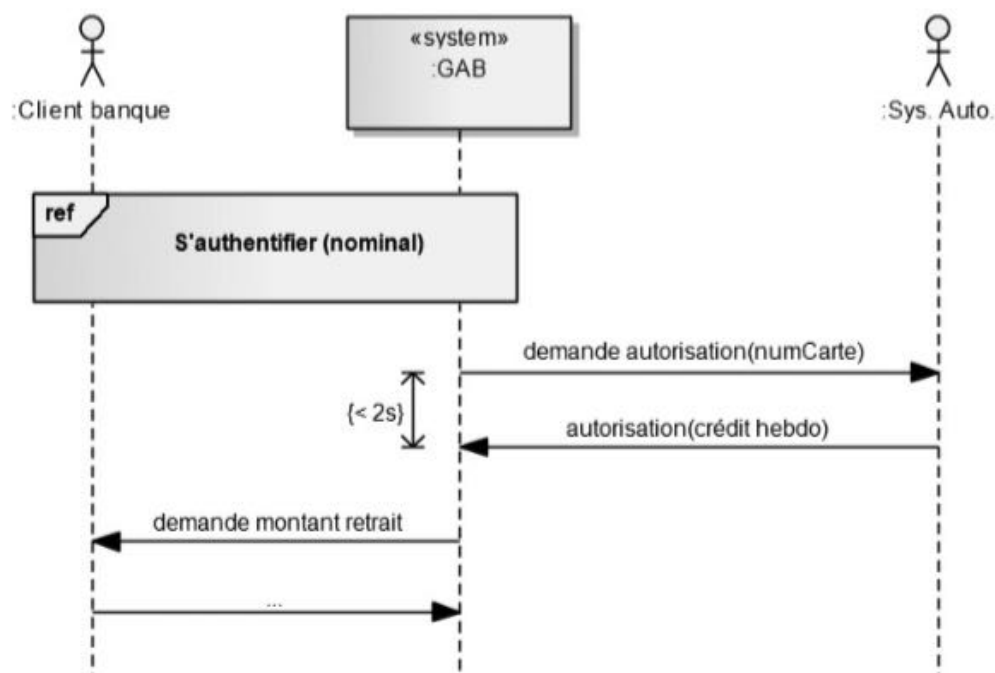


Figure 1-17.

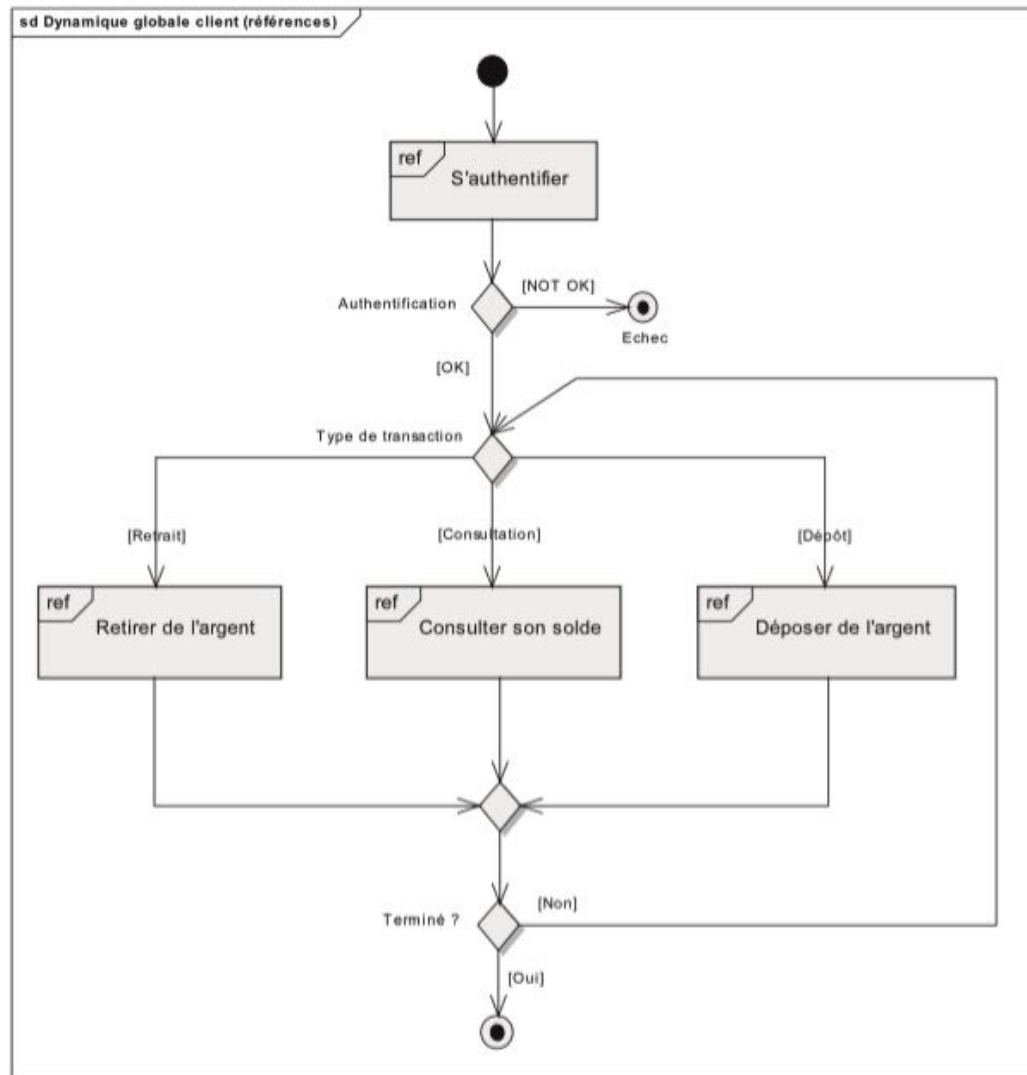
Nouveau diagramme de séquence système avec référence au cas inclus



Interaction Overview Diagram

Figure 1-25.

*Vue d'ensemble
des interactions
du client banque*



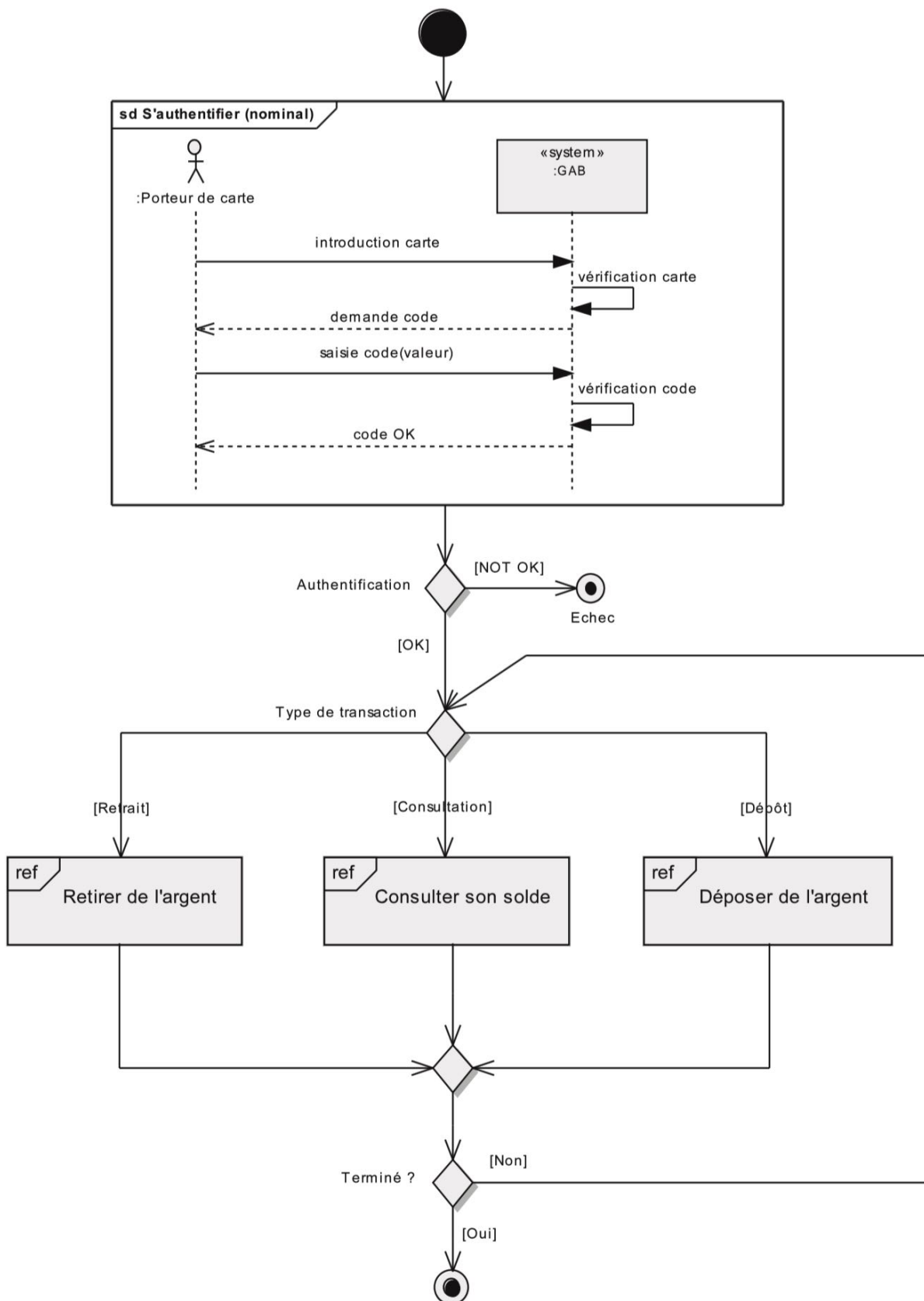
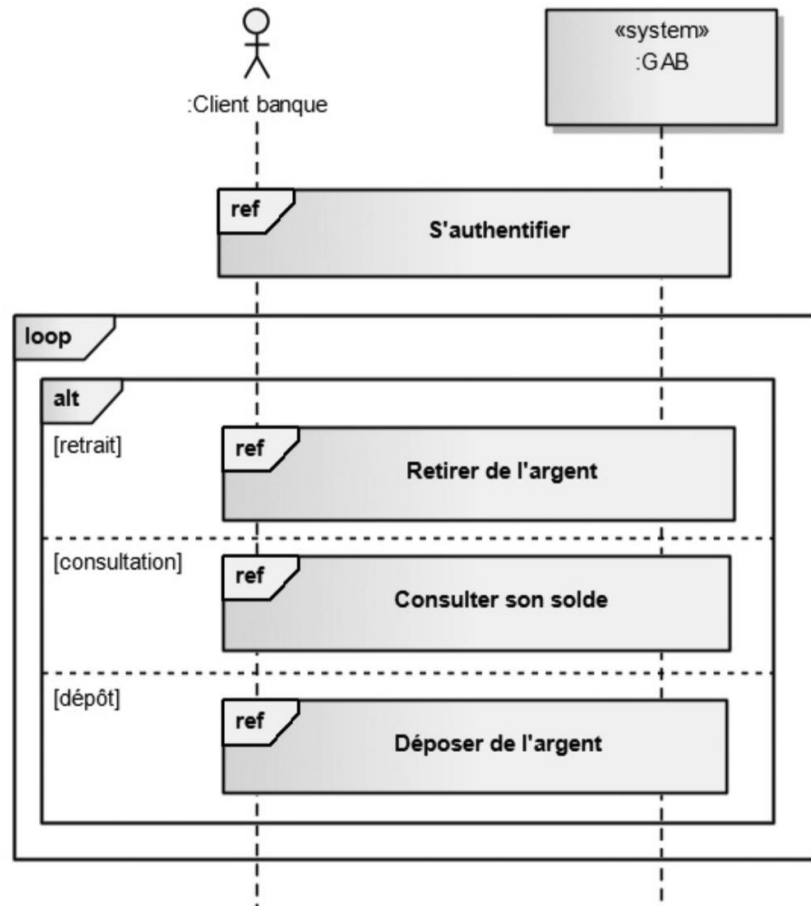


Figure 1-26.

Vue d'ensemble des interactions du client banque (expansé)

Figure 1-27.
*Diagramme de séquence
du comportement du client banque*



User story

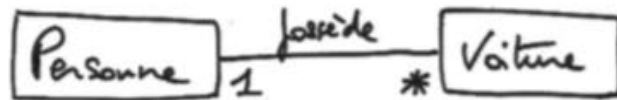
Figure 2-17.
User story et use case

USER STORY	USE CASE
Est une brève description d'une fonctionnalité telle que vue par l'utilisateur (Définition)	Représente une séquences d'actions qu'un système ou toute autre entité peut accomplir en interagissant avec les acteurs du système (Définition)
Format écrit court , laissant la part belle à la discussion orale	Format écrit très riche en informations (pré conditions, Evénement déclencheur, scénario principal, alternatives ...). Peu de place à l'oral
Est une partie d'un Use Case (le scénario principal ou une alternative)	Est la somme d'un scénario principal, et de diverses alternatives (variations, cas d'erreur ...)
Utilisée certes en tant que spécification mais surtout pour l'estimation & la planification	Utilisé seulement en tant que spécification
Emergence rapide au travers d'ateliers de travail collaboratifs	Long travail d'analyse et de formalisation
Grande lisibilité du fait de sa simplicité	Manque de lisibilité même au sein d'un Template
Mode ORAL et COLLABORATIF	Mode ECRIT ET DISTANT
Très facile à maintenir (format fiche, court, indépendant)	Difficile à maintenir (Doc. Word de 150 pages)
Implémentée et testée en une itération obligatoirement	Implémenté et testé en une ou plusieurs itérations
Ecrite facilement par un Utilisateur ou un Client (accompagné dans sa démarche)	Souvent rédigé par des User Proxies (AMOA, Analyste ...), rarement par le client
Contient des tests d'acceptation (au dos de la carte) => implication des testeurs	Ne contient pas les cas de test qui en découlent => pas d'implication des testeurs
Difficile à lier les unes aux autres. Absence de vue globale	Liaison et vision globale facilitée : Sous Use Case, conditions préalables, Diagramme des cas utilisation
Associée historiquement à eXtreme Programming et aux méthodes Agiles	Associé historiquement au Processus Unifié
Un auteur de référence : Mike Cohn	Un auteur de référence : Alistair Cockburn

Association et cardinalité

Figure 3-2.

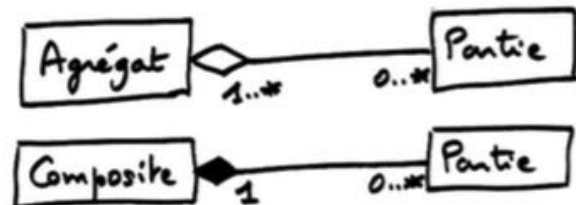
Exemples de multiplicité d'association



Agrégation et composition

Figure 3-3.

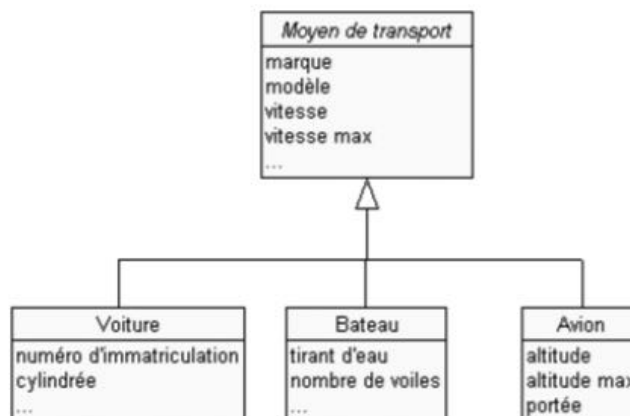
Exemples d'agrégation
et de composition



Généralisation, super-classe, sous-classe, classe abstraite

Figure 3-4.

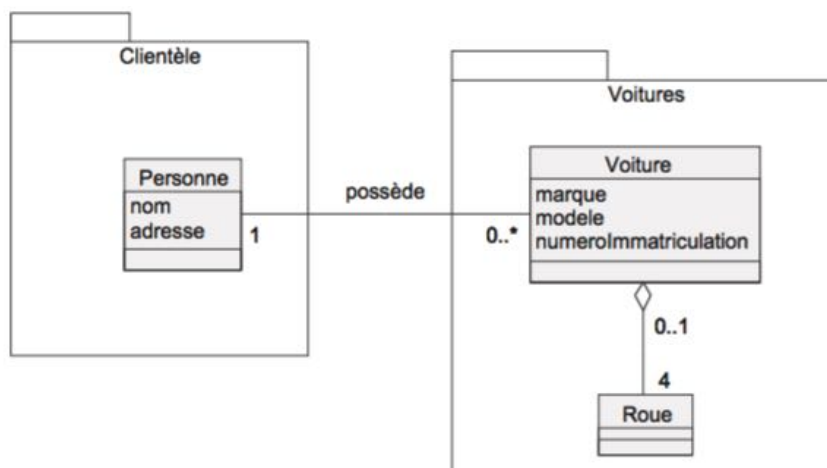
Exemple de super-classe
et sous-classes



Package

Figure 3-5.

Exemples de packages
contenant des classes



Conventions de nommage en UML

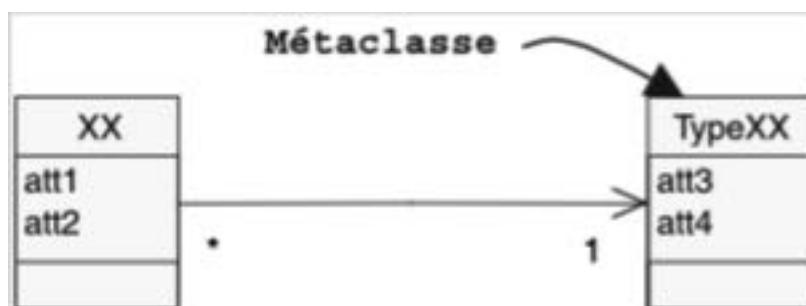
Les noms des attributs commencent toujours par une minuscule (contrairement aux noms des classes qui commencent systématiquement par une majuscule) et peuvent contenir ensuite plusieurs mots concaténés, commençant par une majuscule. Il est préférable de ne pas utiliser d'accents ni de caractères spéciaux. Les mêmes conventions s'appliquent au nommage des rôles des associations, ainsi qu'aux opérations.

Pattern de la métaclasse ou type-exemplaire

La séparation des responsabilités peut se généraliser sous la forme d'un « pattern d'analyse » réutilisable dans d'autres contextes.

On identifie une classe XX qui possède trop de responsabilités, dont certaines ne sont pas propres à chaque instance. On ajoute une classe TypeXX, on répartit les propriétés sur les deux classes et on les relie par une association « * - 1 ». La classe TypeXX est qualifiée de « métaclasse », car elle contient des informations qui décrivent la classe XX.

Ce pattern serait par exemple utile pour modéliser les livres d'une bibliothèque. La classe Livre jouerait le rôle de TypeXX, avec des attributs comme date de parution, ISBN, nombre de pages, etc. La classe ExemplaireLivre jouerait quant à elle le rôle de la classe XX, avec un attribut état (neuf, moyen, abîmé), une association est empruntée par vers Adhérent, etc. On parle d'ailleurs également de pattern « Type – Exemplaires ».



Inversion de dépendance

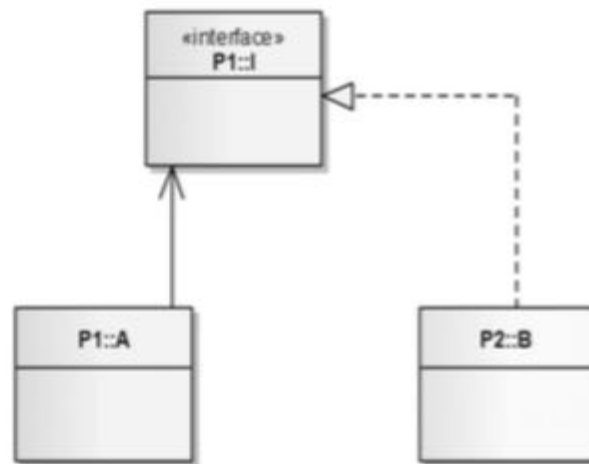
Supposons que nous ayons deux classes A et B reliés par une association unidirectionnelle de A vers B. Ces deux classes appartiennent respectivement aux packages P1 et P2. Il existe donc une dépendance de P1 vers P2.

Figure 3-41.
*Exemple de dépendance
à inverser*



Nous souhaitons inverser les dépendances entre P1 et P2 sans déplacer les classes A et B. Pour résoudre ce problème, il suffit d'extraire une interface I que B implémentera et de relier A à cette interface plutôt qu'à B directement. L'interface I sera positionnée dans le même package P1 que la classe A. C'est maintenant le package P2 qui dépend du package P1 !

Figure 3-42.
*Ajout d'une interface
pour inverser la
dépendance*



Le Pattern Composite

La solution de la figure suivante a été décrite d'une façon plus générale dans l'ouvrage de référence sur les Design Patterns, sous le nom de pattern composite. Ce pattern fournit une solution élégante pour modéliser des structures arborescentes qui représentent des hiérarchies composant/composé. Le client peut ainsi traiter de la même façon les objets individuels (feuilles) et leurs combinaisons (composites).

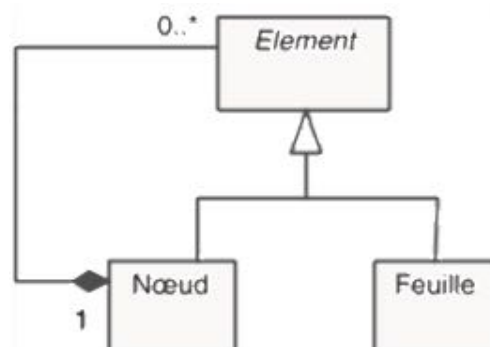


Figure 4-45.
Pattern du composite sous forme simple