

Modélisation fonctionnelle

étude de cas

Mots-clés

■ Acteur ■ Contexte statique ■ Cas d'utilisation ■ Acteur principal, acteur secondaire ■ Diagramme de cas d'utilisation ■ Scénario, enchaînement ■ Fiche de description d'un cas d'utilisation ■ Diagramme de séquence système ■ Diagramme d'activité ■ Inclusion, extension et généralisation de cas d'utilisation ■ Généralisation/spécialisation d'acteurs ■ Package de cas d'utilisation ■ *Interaction Overview Diagram*.

Ce chapitre va nous permettre d'illustrer pas à pas, sur une première étude de cas, les principales difficultés liées à la mise en œuvre de la technique des cas d'utilisation.

Après avoir identifié les acteurs qui interagissent avec le système, nous y développons un premier modèle UML de haut niveau, pour pouvoir établir précisément les frontières du système. Dans cette optique, nous apprenons à identifier les cas d'utilisation et à construire un diagramme reliant les acteurs et les cas d'utilisation. Ensuite, nous précisons le point de vue fonctionnel en détaillant les différentes façons dont les acteurs peuvent utiliser le système. À cet effet, nous apprenons à rédiger des descriptions textuelles de cas d'utilisation, ainsi qu'à dessiner des diagrammes UML complémentaires (comme les diagrammes de séquence ou d'activité).

PRINCIPES ET DÉFINITIONS DE BASE

ACTEUR

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Comment les identifier ?

Les acteurs candidats sont systématiquement :

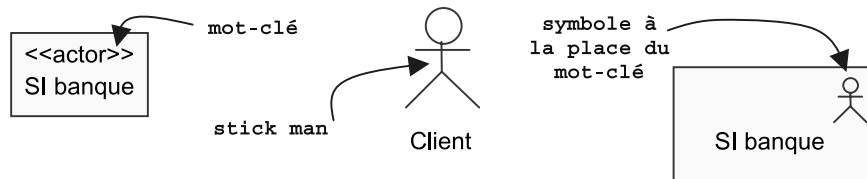
- les utilisateurs humains directs : faites donc en sorte d'identifier tous les profils possibles, sans oublier l'administrateur, l'opérateur de maintenance, etc. ;
- les autres systèmes connexes qui interagissent aussi directement avec le système étudié, souvent par le biais de protocoles bidirectionnels.

Comment les représenter ?

La représentation graphique standard de l'acteur en UML est l'icône appelée *stick man*, avec le nom de l'acteur sous le dessin. On peut également figurer un acteur sous la forme rectangulaire d'une classe, avec le mot-clé `<<actor>>`. Une troisième représentation (intermédiaire entre les deux premières) est également possible avec certains outils, comme cela est indiqué ci-après.

Figure 1-1.

Représentations
graphiques
possibles
d'un acteur



Une bonne recommandation consiste à faire prévaloir l'utilisation de la forme graphique du *stick man* pour les acteurs humains et une représentation rectangulaire pour les systèmes connectés.

CAS D'UTILISATION

Un *cas d'utilisation* (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire *ce que* le futur système devra faire, sans spécifier *comment* il le fera.

Comment les identifier ?

L'objectif est le suivant : l'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un de ses acteurs.

Pour chaque acteur, il convient de :

- rechercher les différentes intentions métier avec lesquelles il utilise le système,
- déterminer dans le cahier des charges les services fonctionnels attendus du système.

Nommez les cas d'utilisation par un verbe à l'infinitif suivi d'un complément, du point de vue de l'acteur (et non pas du point de vue du système).

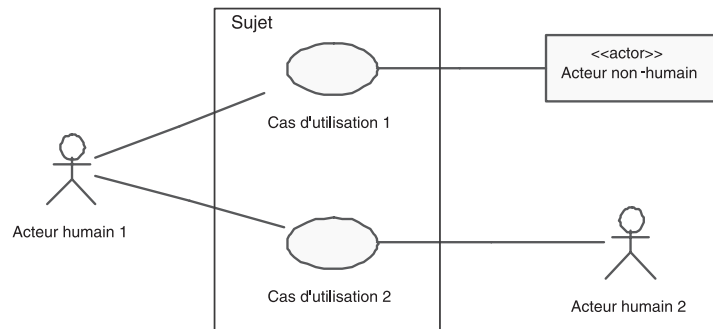
Comment les analyser ?

Pour détailler la dynamique du cas d'utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système. Le cas d'utilisation doit avoir un début et une fin clairement identifiés. Il faut également préciser les variantes possibles, telles que le cas nominal, les différents cas alternatifs et d'erreur, tout en essayant d'ordonner séquentiellement les descriptions, afin d'améliorer leur lisibilité.

Comment les représenter ?

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs (icône du « stick man », ou représentation graphique équivalente). Chaque association signifie simplement « participe à ». Un cas d'utilisation doit être relié à au moins un acteur.

Figure 1-2.
Diagramme de cas
d'utilisation



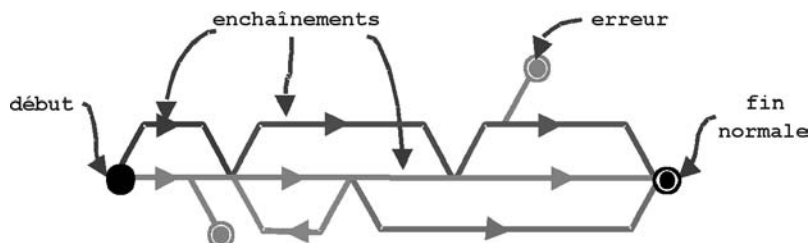
Une fois les cas d'utilisation identifiés, il faut encore les décrire !

SCÉNARIO

Un *scénario* représente une succession particulière d'enchaînements, s'exécutant du début à la fin du cas d'utilisation, un *enchaînement* étant l'unité de description de séquences d'actions. Un cas d'utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs (qui se terminent de façon normale) ou d'erreur (qui se terminent en échec).

On peut d'ailleurs proposer une définition différente pour un cas d'utilisation : « ensemble de scénarios d'utilisation d'un système reliés par un but commun du point de vue d'un acteur ».

Figure 1-3.
*Représentation
des scénarios
d'un cas
d'utilisation*



La fiche de description textuelle d'un cas d'utilisation n'est pas normalisée par UML. Nous préconisons pour notre part la structuration suivante :

Sommaire d'identification (obligatoire)	Inclut titre, résumé, dates de création et de modification, version, responsable, acteurs...
Description des scénarios (obligatoire)	Décrit le scénario nominal, les scénarios (ou enchaînements) alternatifs, les scénarios (ou enchaînements) d'erreur, mais aussi les préconditions et les postconditions.
Exigences non-fonctionnelles (optionnel)	Ajoute, si c'est pertinent, les informations suivantes : fréquence, volumétrie, disponibilité, fiabilité, intégrité, confidentialité, performances, concurrence, etc. Précise également les contraintes d'interface homme-machine comme des règles d'ergonomie, une charte graphique, etc.

ÉTUDE D'UN GUICHET AUTOMATIQUE DE BANQUE

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 et dépôt de chèques pour les clients porteurs d'une carte de crédit de la banque adossée au GAB.

N'oubliez pas non plus que :

3. Toutes les transactions sont sécurisées.
4. Il est parfois nécessaire de recharger le distributeur, etc.

À partir de ces quatre phrases, nous allons progressivement :

- identifier les acteurs ;
- identifier les cas d'utilisation ;
- construire un diagramme de cas d'utilisation ;
- décrire textuellement les cas d'utilisation ;
- compléter les descriptions par des diagrammes dynamiques ;
- organiser et structurer les cas d'utilisation.



L'énoncé précédent est volontairement incomplet et imprécis, comme il en est dans les projets réels !!

Notez également que le problème et sa solution sont basés sur l'utilisation de cartes à puce dans le contexte des systèmes bancaires français. Si le GAB que vous avez l'habitude d'utiliser ne fonctionne pas exactement comme le nôtre, ce n'est pas très important. C'est surtout un prétexte pour vous montrer comment raisonner fonctionnellement avec les cas d'utilisation UML.

Étape 1 – Identification des acteurs du GAB



EXERCICE 1-1. Les acteurs

Identifiez les acteurs du GAB.

solution

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

Considérons linéairement les phrases de l'énoncé.

La phrase 1 nous permet d'identifier immédiatement un premier acteur évident : tout « Porteur de carte ». Il pourra uniquement utiliser le GAB pour retirer de l'argent avec sa carte.

En revanche, attention : le lecteur de carte et le distributeur de billets font partie du GAB. Ils ne peuvent donc pas être considérés comme des acteurs ! Vous pouvez noter ici que l'identification des acteurs oblige à fixer précisément la frontière entre le système à l'étude et son environnement. Si nous restreignions l'étude au système de contrôle-commande des éléments physiques du GAB, le lecteur de carte et le distributeur de billets deviendraient alors des acteurs.

Autre piège : la carte bancaire elle-même est-elle un acteur ? La carte est bien externe au GAB, et elle interagit avec lui... Pourtant, nous ne recommandons pas de la répertorier en tant qu'acteur, car nous appliquons le principe suivant : éliminer autant que possible les acteurs « physiques » au profit des acteurs « logiques ». L'acteur est celui qui bénéficie de l'utilisation du système. C'est bien le Porteur de carte qui retire de l'argent pour le dépenser ensuite, pas la carte !

La phrase 2 identifie des services supplémentaires qui ne sont proposés qu'aux clients de la banque porteurs d'une carte de crédit de cette dernière. Il s'agit donc d'un profil différent du précédent, que nous matérialisons par un deuxième acteur, appelé *Client banque*¹.

La phrase 3 nous incite à prendre en compte le fait que toutes les transactions sont sécurisées. Mais sécurisées par qui ? Pas par le GAB. Il existe donc d'autres entités externes qui jouent le rôle de Système d'autorisation et avec lesquelles le GAB communique directement. Une interview de l'expert métier est nécessaire, pour nous permettre d'identifier deux acteurs différents :

- le Système d'autorisation global Carte Bancaire, pour les transactions de retrait ;
- le Système d'information de la banque, pour autoriser toutes les transactions effectuées par un client avec sa carte de la banque, mais également pour accéder au solde des comptes.

Enfin, la phrase 4 nous rappelle qu'un GAB nécessite également des actions de maintenance, telles que le rechargement en billets du distributeur, la récupération des cartes avalées, etc. Ces actions de maintenance sont effectuées par un nouvel acteur, que nous appellerons pour simplifier : *Opérateur de maintenance*².

1. Il faudrait parler en toute rigueur de « Porteur de carte client de la banque », mais c'est un peu long, d'où la nécessité absolue de documenter tout élément UML, y compris les acteurs.

2. Nous pourrions par exemple identifier un acteur supplémentaire appelé « *Convoyeur* » ou « *Gabiste* », chargé spécifiquement de remplir la caisse du GAB.

Plutôt que de répertorier simplement les acteurs textuellement, on peut réaliser un premier diagramme que nous appelons *diagramme de contexte statique*. Il suffit pour cela d'utiliser un diagramme de classes dans lequel chaque acteur est relié par une association à une classe centrale unique représentant le système, ce qui permet en outre de spécifier le nombre d'instances d'acteurs connectées au système à un moment donné.

Bien que ce diagramme ne fasse pas partie des diagrammes UML « officiels », nous l'avons très souvent trouvé utile dans notre expérience des projets réels.



EXERCICE 1-2.

Diagramme de contexte statique

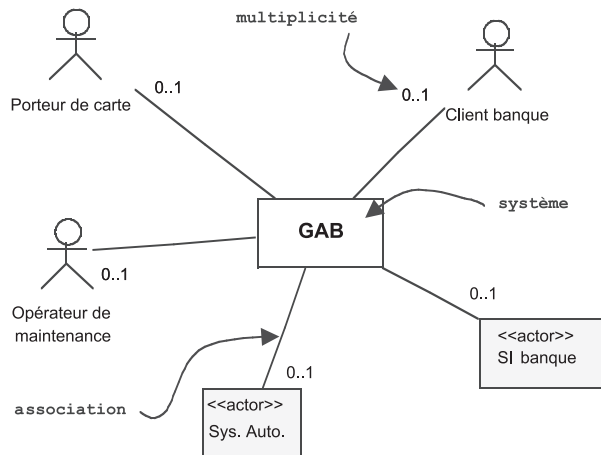
Élaborez le diagramme de contexte statique du GAB.

solution

Le GAB est un système fondamentalement mono-utilisateur : à tout instant, il n'y a qu'une instance de chaque acteur (au maximum) connectée au système.

Figure 1-4.

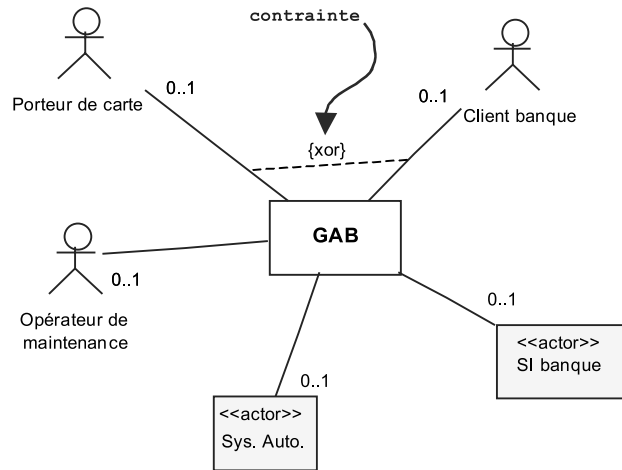
Diagramme de contexte statique du GAB



Il faudrait en toute rigueur ajouter une note graphique pour indiquer qu'en outre les acteurs humains *Client banque* et *Porteur de carte* sont mutuellement exclusifs, ce qui n'est pas implicite d'après les multiplicités des associations.

Figure 1-5.

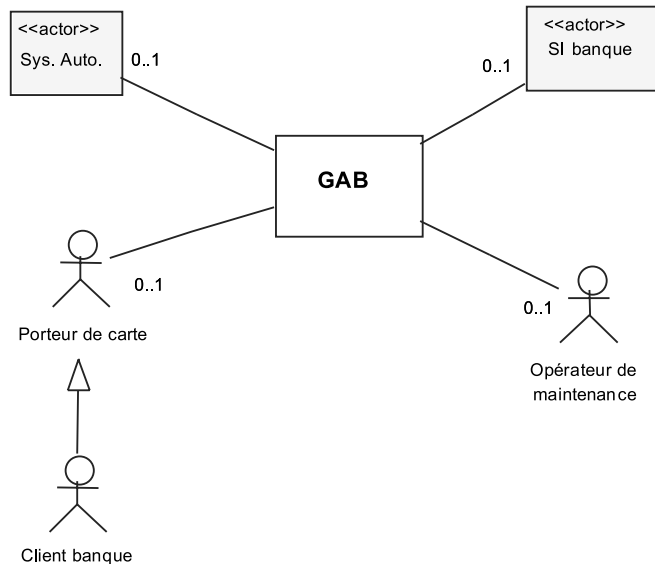
Diagramme de contexte statique du GAB complété



Une autre solution, un peu plus élaborée, consiste à considérer que *Client banque* est une spécialisation de *Porteur de carte*, comme cela est illustré sur la figure suivante. Le problème précité d'exclusivité est ainsi résolu par construction, grâce à l'héritage entre acteurs.

Figure 1-6.

Version plus élaborée du diagramme de contexte statique du GAB



Étape 2 – Identification des cas d'utilisation



EXERCICE 1-3.

Cas d'utilisation

Préparez une liste préliminaire des cas d'utilisation du GAB, par acteur.

solution

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 (à ne pas oublier !).
- Consulter le solde de son compte courant.
- Déposer du numéraire.
- Déposer de l'argent (du numéraire ou des chèques)³.

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.

3. Il n'est pas nécessaire de répertorier deux cas d'utilisation distincts appelés *Déposer du numéraire* et *Déposer des chèques*. En effet, ils auraient le même acteur principal et le même objectif global. Cependant, cette décision pourra être remise en cause lors de la description détaillée des cas d'utilisation si l'on s'aperçoit alors que les scénarios sont trop différents. N'oubliez pas qu'une décision de modélisation ne doit jamais être irréversible !

C'est exactement le cas des deux acteurs « non humains » de notre exemple : le *Sys. Auto.* et le *SI banque* sont uniquement sollicités par le GAB dans le cadre de la réalisation de certains cas d'utilisation. Mais ils n'ont pas eux-mêmes de façon propre d'utiliser le GAB, d'objectif à part entière.

Étape 3 – Réalisation de diagrammes de cas d'utilisation

On obtient sans difficulté un diagramme préliminaire en transcrivant la réponse précédente sur un schéma qui montre les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs principaux (icône du « stick man »).

À retenir

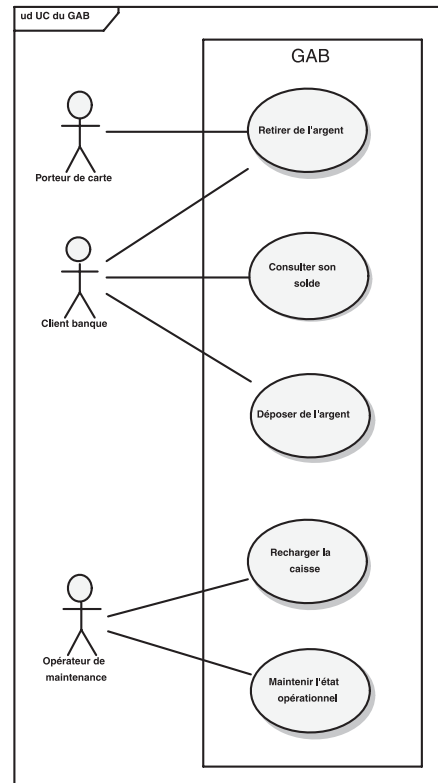
CADRE DE DIAGRAMME : TAG ET NOM

Notez que depuis UML 2.0, un diagramme peut être inclus dans un cadre accueillant tout le contenu graphique. Le cadre a pour intitulé le nom du diagramme et établit sa portée. C'est un rectangle avec un petit pentagone (appelé tag de nom) placé dans l'angle supérieur gauche, qui contient le type du diagramme et son nom. Le cadre n'est cependant pas obligatoire lorsque le contexte est clair. La spécification UML définit les tags de chaque type de diagramme, mais cela n'a pas de caractère obligatoire et chaque outil a fait ses propres choix. Celui que nous avons utilisé majoritairement pour les nouveaux diagrammes UML 2 du livre, Enterprise Architect de la société Sparx Systems^a, a pris le parti de définir tous les tags avec deux lettres : ud, cd, sd, td, ad, sm, id, dd. Vous les découvrirez au fur et à mesure des chapitres. Dans le diagramme suivant, ud signifie use case diagram.

- a. Le lecteur peut visiter le site web suivant <http://www.sparxsystems.com.au/> où il trouvera une version d'évaluation du produit Enterprise Architect. Nous recommandons également l'introduction à UML 2 disponible sur le même site.

Figure 1-7.

Diagramme de cas d'utilisation
préliminaire du GAB

**EXERCICE 1-4.****Généralisation entre acteurs**

Proposez une *autre* version, plus sophistiquée, de ce diagramme de cas d'utilisation préliminaire.

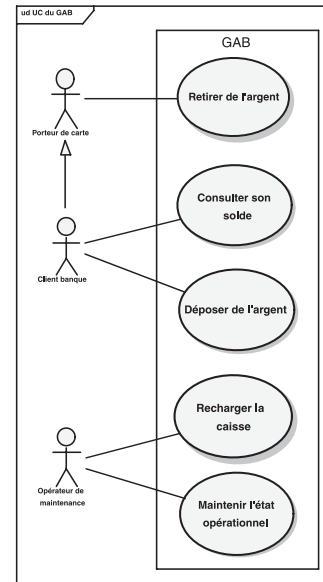
solution

Le cas d'utilisation *Retirer de l'argent* a deux acteurs principaux possibles (mais exclusifs du point de vue de la simultanéité). Une autre façon de l'exprimer consiste à considérer l'acteur *Client de la banque* comme une spécialisation (au sens de la relation d'héritage) de l'acteur plus général *Porteur de carte*, comme nous l'avons déjà indiqué sur la figure 1-6. Un client de la banque est en effet un Porteur de carte particulier qui a toutes les prérogatives de ce dernier, ainsi que d'autres qui lui sont propres en tant que client.

UML permet de décrire une relation de généralisation/spécialisation entre acteurs, comme cela est indiqué sur le diagramme de cas d'utilisation suivant.

Figure 1-8.

Version plus
sophistiquée
du diagramme
de cas d'utilisation
préliminaire



Cependant, dans notre exemple, l'intérêt de cette relation de généralisation n'est pas évident. Elle permet certes de supprimer l'association entre l'acteur *Client banque* et le cas d'utilisation *Retirer de l'argent*, qui est maintenant héritée de l'acteur *Porteur de carte*, mais ajoute en revanche le symbole de généralisation entre les deux acteurs... De plus, nous verrons au paragraphe suivant que les acteurs secondaires sollicités ne sont pas les mêmes dans le cas du Porteur de carte non client et dans celui du client de la banque.

Nous ne retiendrons donc pas cette solution et nous considérerons dans la suite du chapitre que les dénominations *Porteur de carte non client* et *Porteur de carte* sont synonymes.

Il nous reste maintenant à ajouter les acteurs secondaires pour compléter le diagramme de cas d'utilisation. Pour cela, UML propose simplement en standard de faire apparaître ces acteurs avec des associations supplémentaires vers les cas d'utilisation existants.

À retenir

PRÉCISIONS GRAPHIQUES AU DIAGRAMME DE CAS D'UTILISATION

Pour notre part, afin d'améliorer le contenu informatif de ces diagrammes, nous recommandons d'adopter les conventions suivantes :

- par défaut, le rôle d'un acteur est « principal » ; si ce n'est pas le cas, indiquez explicitement que le rôle est « secondaire » sur l'association, du côté de l'acteur ;
- dans la mesure du possible, disposez les acteurs principaux à gauche des cas d'utilisation et les acteurs secondaires à droite.



EXERCICE 1-5.

Acteurs secondaires

Complétez le diagramme de cas d'utilisation préliminaire en ajoutant les acteurs secondaires. Pour simplifier, ne tenez plus compte pour l'instant de l'opérateur de maintenance.

solution

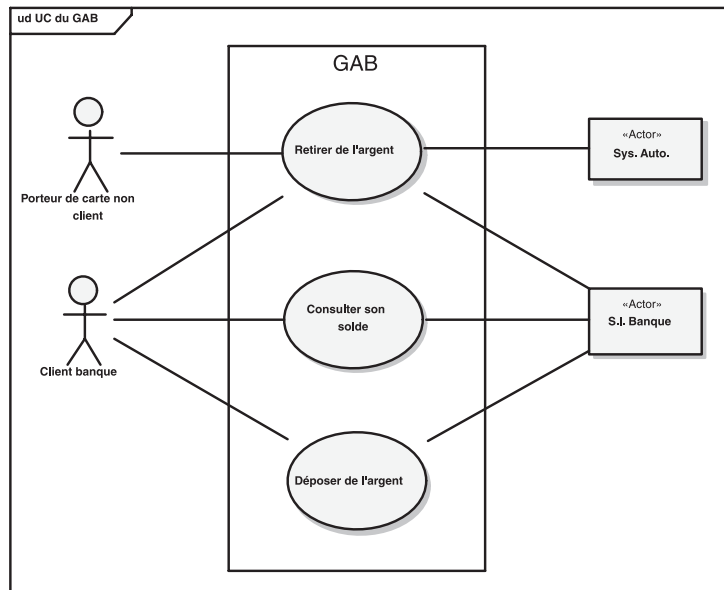
Pour tous les cas d'utilisation propres au client de la banque, il faut clairement faire intervenir comme acteur secondaire *SI banque*.

Mais un problème se pose pour le cas d'utilisation partagé *Retirer de l'argent*. En effet, si l'acteur principal est un Porteur de carte non client, il faudra faire appel au *Sys. Auto.* (qui se chargera ensuite de contacter le SI de la banque du porteur), alors que, s'il s'agit d'un client de la banque, le GAB contactera directement le *SI banque*.

Une première solution consiste à ajouter une association avec chacun des deux acteurs non-humains. Cette modélisation simpliste ne permet pas au lecteur du diagramme de comprendre que les acteurs participent au cas d'utilisation *Retirer de l'argent* de l'argent sélectivement deux par deux et non pas tous ensemble (figure 1-9).

Figure 1-9.

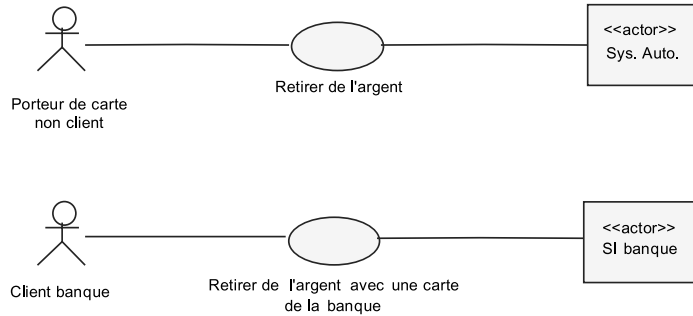
Version simple du diagramme de cas d'utilisation complété



Une autre solution consiste à distinguer deux cas d'utilisation pour le retrait d'argent : *Retirer de l'argent* et *Retirer de l'argent avec une carte de la banque*. Cette modélisation plus précise, mais plus lourde, est plus parlante pour l'expert métier. Elle milite d'ailleurs clairement contre l'utilisation de la généralisation

entre acteurs évoquée précédemment. En effet, la distinction entre les deux cas d'utilisation est contradictoire avec la tentative d'héritage par l'acteur *Client banque* du cas unique *Retirer de l'argent*, qui avait été envisagée plus haut, alors que les acteurs secondaires n'avaient pas encore été ajoutés. Nous garderons cette seconde solution pour la suite de l'exercice⁴ (figure 1-10).

Figure 1-10.
Fragment
de la version
plus précise
du diagramme
de cas
d'utilisation
complété



On notera que le *SI banque* n'est pas un acteur direct du cas d'utilisation *Retirer de l'argent*, car nous considérons que le *Sys. Auto.* se charge de le contacter, en dehors de la portée du GAB.

Étape 4 – Description textuelle des cas d'utilisation



EXERCICE 1-6. Partie obligatoire du cas d'utilisation

Décrivez la partie obligatoire du cas d'utilisation RETIRER DE L'ARGENT (pour l'acteur non client de la banque).

solution

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 : 02/03/02

Date de mise à jour : 05/05/06

Version : 5.0

Responsable : Pascal Roques

4. Il s'agit ici d'un choix de modélisation arbitraire ! Nous ne disons pas que toute autre solution serait mauvaise, mais nous expliquons avec des arguments concrets pourquoi nous préférons la nôtre.

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 carte⁵ 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.

Une autre présentation intéressante⁶ consiste à séparer les actions des acteurs et du système en deux colonnes comme suit :

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'indentification.
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.

5. Nous préconisons de mettre systématiquement une majuscule devant le nom des acteurs pour améliorer la lisibilité du scénario nominal.

6. Cette présentation a été recommandée par C. Larman dans la première version de *Applying UML and Patterns* [Larman 97].

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 alternatifs⁷

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.

7. Nous distinguons les enchaînements alternatifs (Ax) qui reprennent ensuite à une étape du scénario nominal des enchaînements d'erreur (Ey) qui terminent brutalement le cas d'utilisation en échec. L'objectif de l'acteur principal est donc atteint par les scénarios nominaux et alternatifs mais pas par ceux d'erreur.

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.

Une autre présentation intéressante des enchaînements alternatifs et d'erreur consiste à utiliser les conventions préconisées par A. Cockburn⁸. Celui-ci

8. Le lecteur se référera avec profit à l'excellent ouvrage d'Alistair Cockburn : *Rédiger des cas d'utilisation efficaces*, Eyrolles, 2001 [Cockburn 01].

propose d'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.
- 4a. Délai de saisie du code expiré :
Le GAB avertit le porteur et éjecte la carte ; le cas d'utilisation se termine en échec.
- 4-12a.⁹ Le Porteur annule la transaction :
Le GAB éjecte la carte ; le cas d'utilisation se termine en échec.
- 5a. Code d'identification erroné pour la première ou deuxième fois :
 - 5a1. Le GAB enregistre l'échec sur la carte.
 - 5a2. Le GAB avertit le Porteur et le scénario nominal reprend à l'étape 3.
- 5b. Code d'identification erroné pour la troisième fois :
Le GAB avertit le Porteur et confisque la carte ; le cas d'utilisation se termine en échec.
- 7a. Transaction refusée par le Système d'autorisation :
Le GAB avertit le Porteur et éjecte la carte ; le cas d'utilisation se termine en échec.
- 7b. Délai de réponse du Système d'autorisation expiré :
Le GAB avertit le Porteur et éjecte la carte ; le cas d'utilisation se termine en échec.
- 9a. Délai de saisie du montant expiré :
Le GAB avertit le Porteur et éjecte la carte ; le cas d'utilisation se termine en échec.
- 10a. Montant demandé supérieur au solde hebdomadaire :
 - 10a1. Le GAB avertit le Porteur et le scénario nominal reprend à l'étape 8.
- 10b. Solde hebdomadaire insuffisant :
Le GAB avertit le Porteur et éjecte la carte ; le cas d'utilisation se termine en échec.
- 12a. Le Porteur ne demande pas de ticket :
Le cas d'utilisation continue à l'identique, sauf l'impression du ticket.

9. La notation 4-12 signifie : « de l'étape 4 à l'étape 12 ».

- 14a. Délai de retrait de la carte expiré :
 - 14a1. Le GAB confisque la carte et annule la transaction ;
 - 14a2. Le GAB avertit le Système d'autorisation et le cas d'utilisation se termine en échec.
- 16a. Délai de retrait des billets expiré :
 - Le GAB confisque les billets et annule la transaction ; le cas d'utilisation se termine en échec.
- 1-7a. Coupure réseau avec le Système d'autorisation :
 - Le GAB avertit le Porteur et éjecte 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).

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.



EXERCICE 1-7.

Paragraphe optionnels du cas d'utilisation

Complétez la description du cas d'utilisation RETIRER DE L'ARGENT avec les paragraphes optionnels. Détaillez les besoins en interface homme-machine.

solution

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 ¹⁰). L'absence de papier pour imprimer les tickets ne doit pas empêcher les retraits.

10. Cette exigence non fonctionnelle n'est pas réellement spécifique au cas d'utilisation et devra donc se retrouver au final dans un document plus global. Elle a uniquement été citée (ainsi que les suivantes) pour étoffer le paragraphe.

Contraintes	Descriptif
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 pour sélectionner un montant de retrait parmi ceux qui sont proposés.
- Un distributeur de billets.
- Un distributeur de tickets.

Étape 5 – Description graphique des cas d'utilisation

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.

À retenir

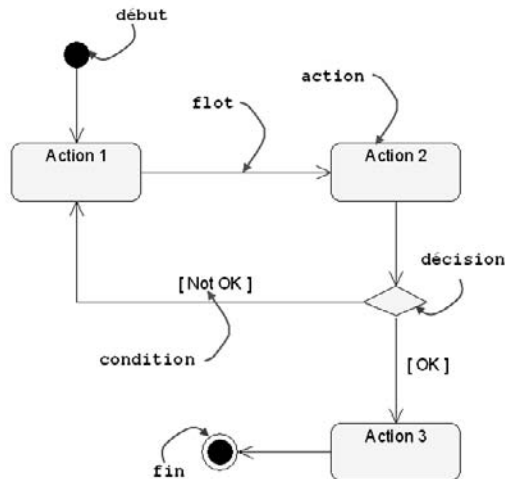
DESCRIPTIONS DYNAMIQUES D'UN CAS D'UTILISATION

Pour les cas d'utilisation, on peut utiliser le *diagramme d'activité* car les utilisateurs le comprennent d'autant plus facilement qu'il paraît ressembler à un organigramme traditionnel^b.

- b. UML 2 propose un nouveau type de diagramme, appelé « Interaction Overview Diagram », qui fusionne les diagrammes d'activité et de séquence. L'utilisation de ce nouveau diagramme paraît prometteuse au niveau des cas d'utilisation, voire du système global. Nous en donnerons un exemple dans la suite du chapitre.

Figure 1-11.

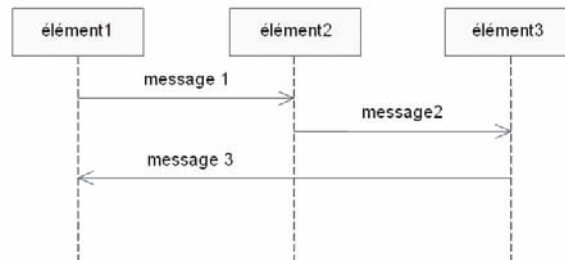
Bases du diagramme d'activité UML 2



Pour des scénarios particuliers, le *diagramme de séquence* est une bonne solution.

Figure 1-12.

Bases du diagramme de séquence UML 2



Nous recommandons de le présenter en montrant l'acteur principal à gauche, puis un objet unique représentant le système en boîte noire, et, enfin, les éventuels acteurs secondaires sollicités durant le scénario à droite du système. Nous utiliserons l'intitulé *diagramme de séquence système* comme cela est proposé dans [Larman 97].

Avec les intéressants ajouts au diagramme de séquence apportés par UML 2, en particulier les cadres d'interactions (avec les opérateurs `loop`, `opt` et `alt` par exemple), ainsi que la possibilité de référencer une interaction décrite par ailleurs, le diagramme de séquence système nous semble constituer une excellente solution.



EXERCICE 1-8. Diagramme de séquence système

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

solution

Il suffit de transcrire sous forme de diagramme de séquence les interactions citées dans le scénario textuel de la réponse 1-6, en utilisant les conventions graphiques citées plus haut :

- l'acteur principal *Porteur de carte* à gauche ;
- un participant représentant le GAB au milieu ;
- l'acteur secondaire *Sys. Auto.* à droite du GAB.

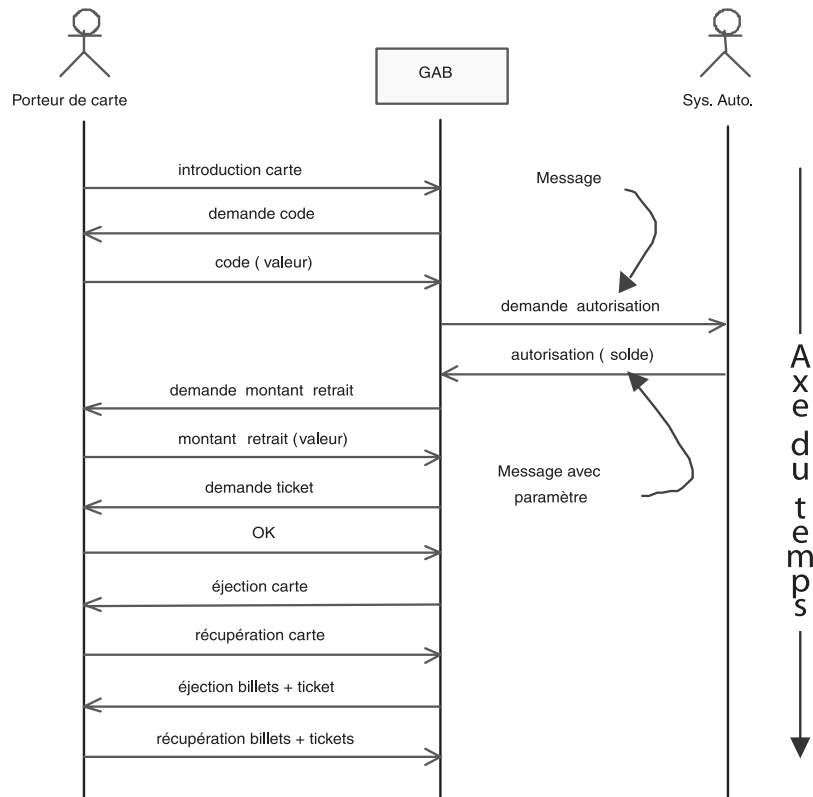


Figure 1-13.
Diagramme
de séquence système
du scénario nominal
de Retirer de l'argent

Contrairement au diagramme de séquence précédent qui ne décrit que le scénario nominal, le diagramme d'activité doit représenter l'ensemble des actions réalisées par le système, avec tous les branchements conditionnels et toutes les boucles possibles.

C'est un graphe orienté d'actions et de transitions. Les transitions sont franchies lors de la fin des actions ; des étapes peuvent être réalisées en parallèle ou en séquence.



EXERCICE 1-9.

Diagramme d'activité

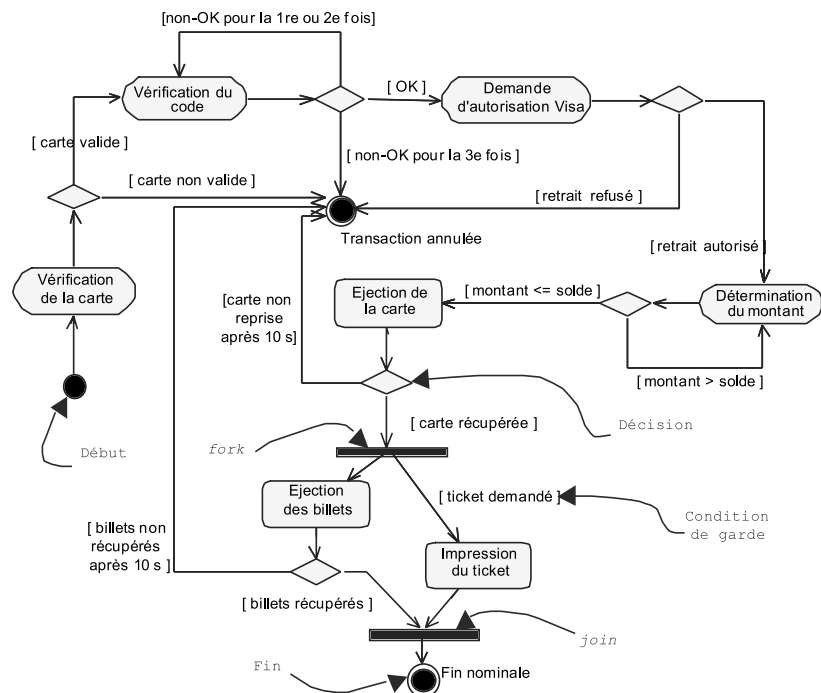
Réalisez un diagramme d'activité qui décrit la dynamique du cas d'utilisation RETIRER DE L'ARGENT.

solution

Le diagramme d'activité est décrit sur la figure suivante avec un repérage des principaux symboles graphiques.

Notez que le diagramme diffère légèrement du texte : il omet l'étape de demande de ticket dans un souci de simplification. Néanmoins, le résultat de cette demande est pris en compte par la condition de garde avec le label « ticket demandé ».

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





EXERCICE 1-10.

Diagramme de séquence système enrichi

À retenir

AJOUTS AU DIAGRAMME DE SÉQUENCE SYSTÈME

Une possibilité intéressante consiste à enrichir le diagramme de séquence système du scénario nominal pour faire apparaître également :

- les principales actions internes du système (au moyen de messages qu'il s'envoie à lui-même) ;
- les renvois aux enchaînements alternatifs et d'erreur (au moyen de notes).

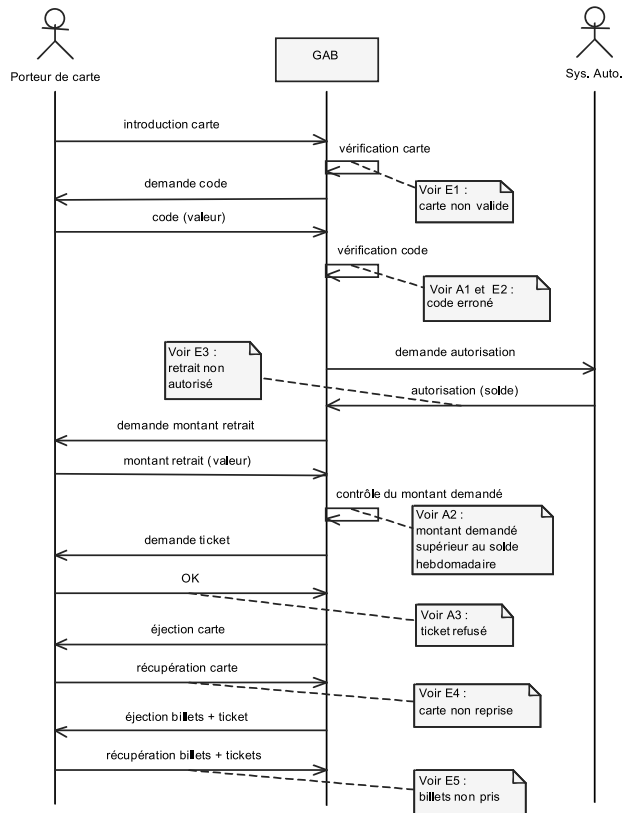
Cela donne souvent un diagramme moins complexe à lire que ne l'est un diagramme d'activité, car moins riche en symboles, mais au contenu informatif appréciable pour l'expert métier.

Enrichissez le diagramme de séquence système qui décrit le scénario nominal du cas d'utilisation RETIRER DE L'ARGENT.

solution

Figure 1-15.

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



Étape 6 – Organisation des cas d'utilisation

Dans cette avant-dernière étape, 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.

Abordons tout d'abord la relation *d'inclusion* : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements. On utilise cette relation pour éviter de décrire plusieurs fois le même enchaînement, en factorisant le comportement commun dans un cas d'utilisation à part.



EXERCICE 1-11.

Inclusion entre cas d'utilisation

Identifiez une partie commune aux différents cas d'utilisation et factorisez-la dans un nouveau cas inclus dans ces derniers.

solution

Si l'on examine en détail la description textuelle du cas d'utilisation *Retirer de l'argent*, on s'aperçoit que le début du scénario nominal va également être applicable à tous les cas d'utilisation du client de la banque, en remplaçant « Porteur de carte » par « Client de la banque » :

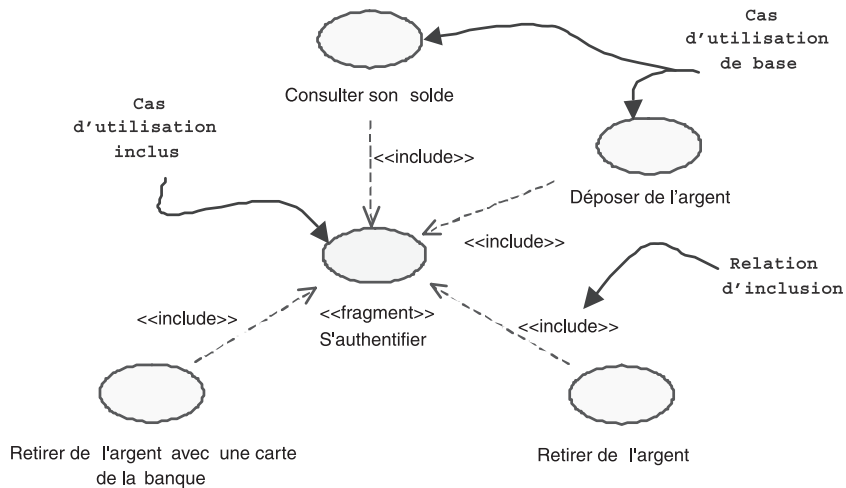
1. Le Porteur de carte introduit sa carte dans le lecteur de cartes du GAB.
2. Le GAB vérifie que la carte introduite est bien une carte bancaire.
3. Le GAB demande au Porteur de carte de saisir son code d'identification.
4. Le Porteur de carte saisit son code d'identification.
5. Le GAB compare le code d'identification avec celui qui est codé sur la puce de la carte.

Cet enchaînement nominal est en outre complété par les enchaînements alternatifs ou d'erreur A1 (code d'identification provisoirement erroné), E1 (carte non valide) et E2 (code d'identification définitivement erroné).

On peut donc légitimement identifier un nouveau cas d'utilisation¹¹ inclus dans les précédents, que nous appellerons *S'authentifier*, et qui contient les enchaînements cités plus haut. Cela nous permettra d'enlever des autres cas d'utilisation toutes ces descriptions textuelles redondantes, en se concentrant mieux sur leurs spécificités fonctionnelles.

En UML, cette relation d'inclusion obligatoire est formalisée par une flèche de dépendance entre le cas d'utilisation de base et le cas inclus, nommée avec le mot-clé `<<include>>`, comme cela est indiqué sur le schéma suivant.

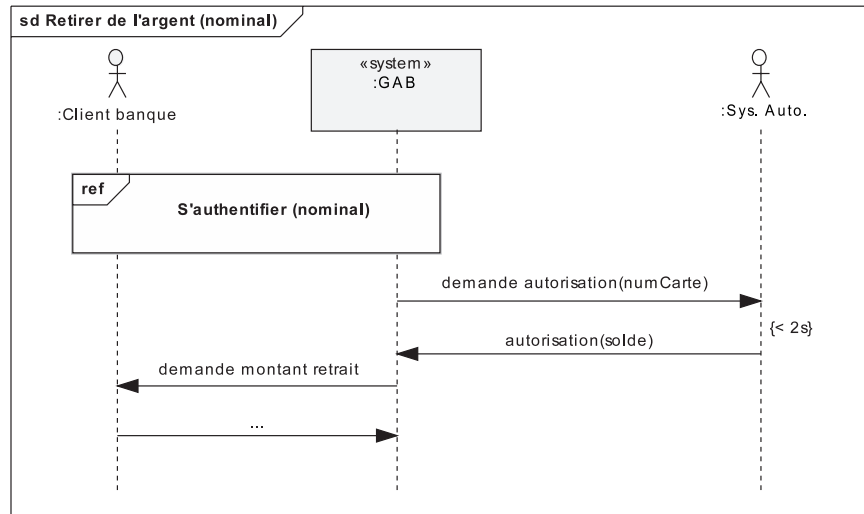
Figure 1-16.
*Relation d'inclusion
entre cas d'utilisation*



L'identification du fragment *S'authentifier* permet d'alléger le diagramme de séquence système en utilisant le cadre *ref* proposé par UML 2. Le début du diagramme remanié est donné sur la figure 1-17. Notez la contrainte temporelle sur la réponse du système d'autorisation.

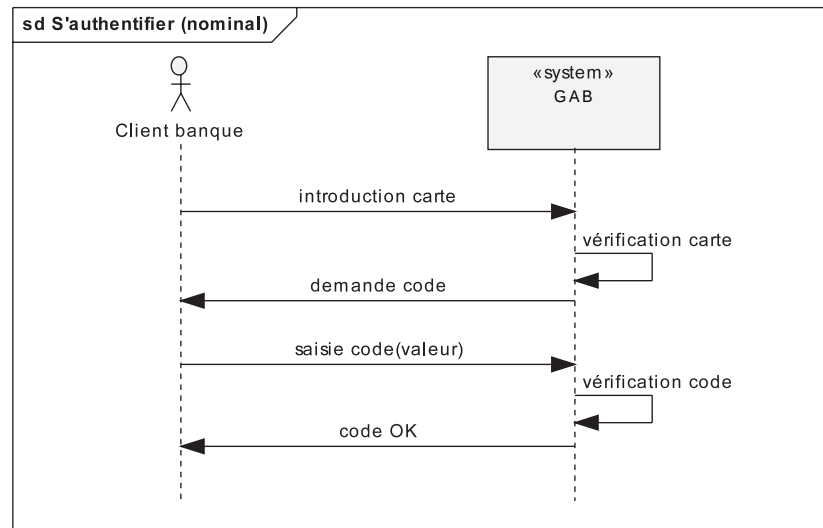
11. Jacobson a proposé dans un article récent de parler de fragment plutôt que de cas d'utilisation à part entière, car s'authentifier n'est pas un objectif à part entière de l'utilisateur du GAB ! Pour renforcer ce concept, nous préconisons d'utiliser un mot-clé (ou stéréotype) « fragment » pour le différencier des autres cas d'utilisation.

Figure 1-17.
Nouveau
diagramme
de séquence
système
avec référence
au cas inclus



Le cadre **ref** fait référence à un autre diagramme de séquence donné figure 1-18.

Figure 1-18.
Diagramme
de séquence système
du fragment référencé



Poursuivons notre analyse par la relation *d'extension* : cette fois-ci, le cas de base en incorpore implicitement un autre, mais de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension. On utilise cette relation pour séparer un comportement optionnel ou rare du comportement obligatoire.



EXERCICE 1-12.

Extension entre cas d'utilisation

En extrapolant sur les besoins initiaux, identifiez une relation d'extension entre deux cas d'utilisation du client de la banque.

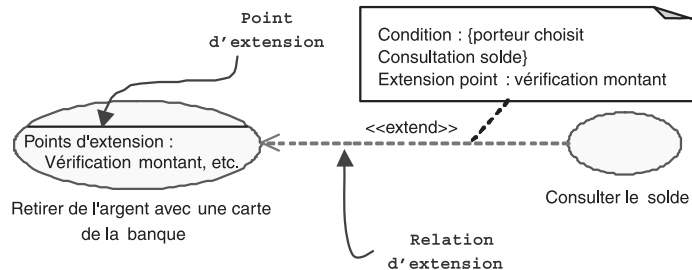
solution

En réexaminant la question du retrait d'argent, on a tôt fait de s'apercevoir que le client de la banque applique quasiment le même enchaînement nominal que le Porteur de carte. Mais, en tant que client, il a également accès aux autres cas d'utilisation : pourquoi ne pas lui permettre de consulter son solde juste avant qu'il ne choisisse le montant de son retrait ? Il pourrait ainsi ajuster le montant demandé avec ce qu'il lui reste à ce moment sur son compte.

Si l'on retient ce nouveau besoin fonctionnel, pour le modéliser en UML il suffit d'ajouter une relation d'extension optionnelle comme cela est indiqué sur la figure suivante.

Figure 1-19.

Relation d'extension entre cas d'utilisation



Les deux cas d'utilisation peuvent bien sûr s'exécuter indépendamment, mais *Consulter le solde* peut également venir s'insérer 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.

...

Poursuivons enfin par la relation de *généralisation / spécialisation* : les cas d'utilisation descendants héritent de la description de leur parent commun. Ils peuvent néanmoins comprendre chacune des interactions spécifiques supplémentaires, ou

modifier les interactions dont ils ont hérités. On utilise cette relation pour formaliser des variations importantes¹² sur le même cas d'utilisation.



EXERCICE 1-13.

Généralisation / spécialisation entre cas d'utilisation

Identifiez une relation de généralisation / spécialisation qui implique un cas d'utilisation du client de la banque.

solution

Considérons le cas d'utilisation *Déposer de l'argent*. Il possède deux scénarios principaux : *Déposer du numéraire* et *Déposer des chèques*. Reprenons la discussion entamée en début de chapitre : est-il souhaitable de distinguer ces scénarios en tant que cas d'utilisation à part entière ? Essayons de trouver les arguments pour et contre.

Ils mettent en jeu les mêmes acteurs : le *Client banque* comme acteur principal et le *SI banque* comme acteur secondaire. Mais surtout, ils parlent de la même chose : la possibilité offerte à un client de la banque d'effectuer un dépôt d'argent grâce au GAB. Le fait que cette transaction consiste à glisser des billets dans un lecteur de billets, ou à simplement déposer une enveloppe contenant un ou plusieurs chèques, n'est pas fondamental. Le résultat sera similaire, à savoir une ligne de crédit sur le compte du client.

Pourtant, le détail des enchaînements va varier notablement : le dépôt de numéraire implique par exemple un dispositif de reconnaissance de billets, avec des interactions liées à chaque introduction de billets, aux erreurs possibles (billet non reconnu, etc.) et à la fin de la transaction. Il est aussi probable que le système de tenue des comptes (qui fait partie du *SI banque*) soit informé en temps réel du dépôt afin de créditer le compte, alors que le dépôt de chèques donnera lieu pour sa part à une vérification manuelle par un guichetier bien après que la transaction fut terminée. En distinguant deux cas d'utilisation, nous ajoutons la possibilité de leur associer des acteurs secondaires différents.

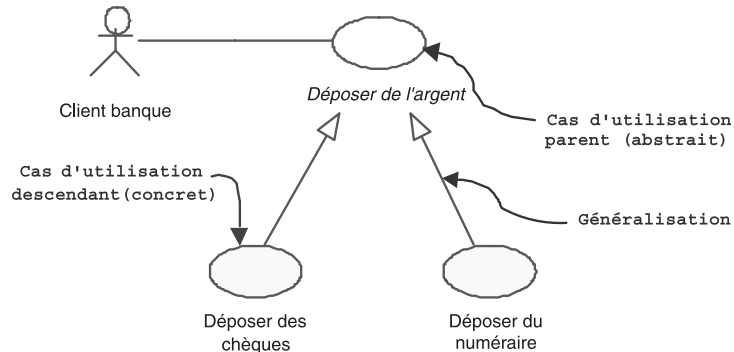
Pour formaliser cette unité fonctionnelle, tout en se gardant la possibilité de décrire les différences au niveau des enchaînements, nous pouvons utiliser la

12. Nous insistons sur le mot « important », car cette relation de généralisation entre cas d'utilisation se révèle souvent inutile dans la pratique, voire dangereuse. Elle ajoute de la complexité au diagramme, ce qui peut rebuter l'expert métier qui doit le valider, surtout si le modélisateur utilise également l'inclusion et l'extension !

relation de généralisation / spécialisation. Il suffit de considérer que *Déposer de l'argent* est un cas d'utilisation généralisé. Ce cas a maintenant la particularité d'être abstrait (il apparaît alors en italiques), car il ne s'instancie pas directement, mais uniquement par le biais de l'un de ses deux cas spécialisés.

Figure 1-20.

Relation de généralisation
entre cas d'utilisation



Le diagramme des cas d'utilisation du client banque fait apparaître également la factorisation rendue possible de la relation d'inclusion avec le fragment de cas d'utilisation *S'authentifier*. Il permet surtout d'associer l'acteur secondaire S.I. Banque avec le seul cas d'utilisation spécialisé *Déposer du numéraire*.

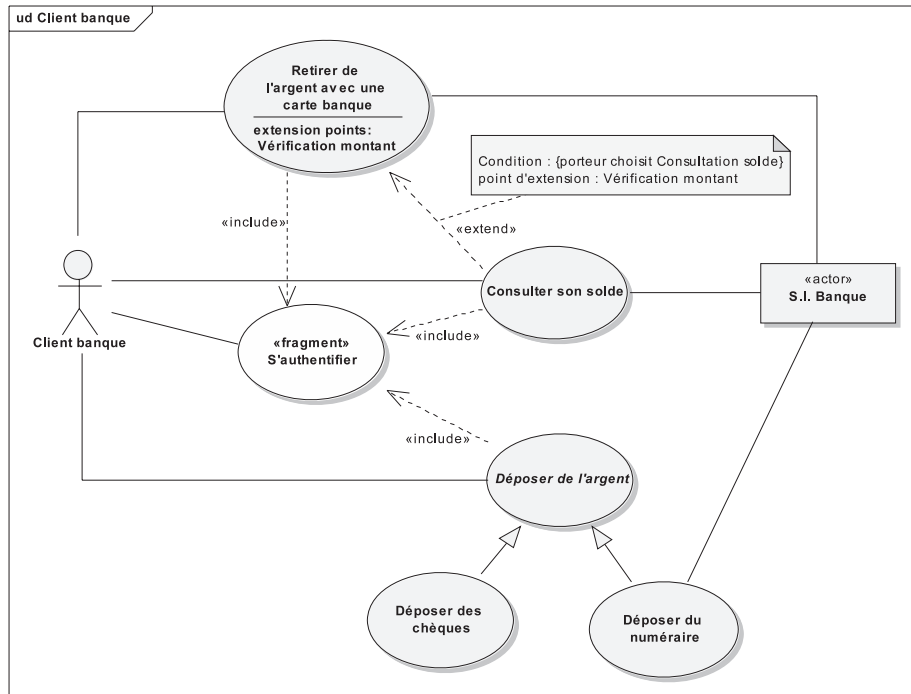


Figure 1-21.

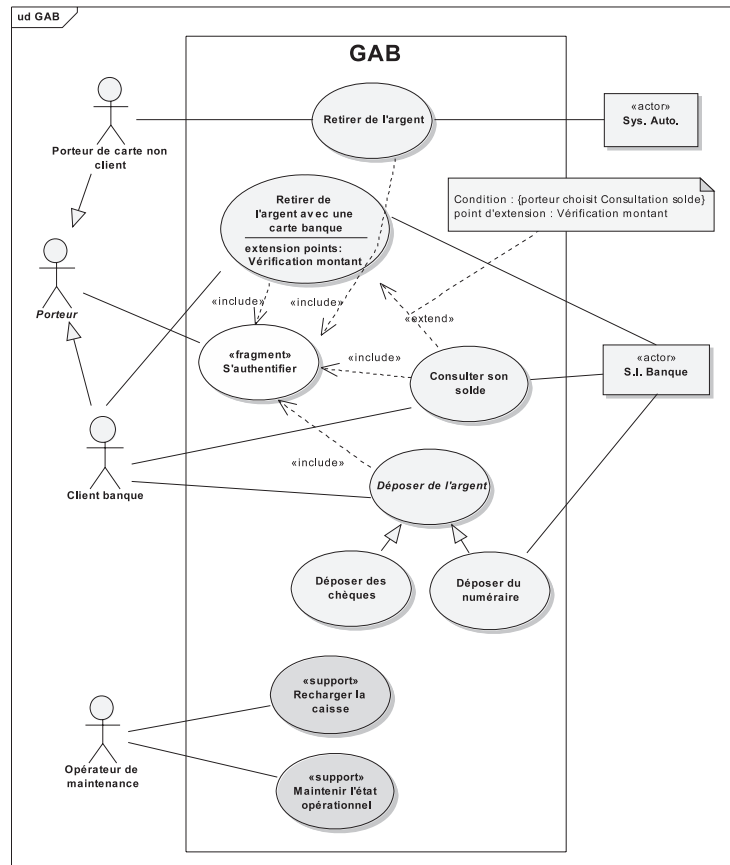
Cas d'utilisation du client banque

Avec tous ces ajouts, que devient donc notre diagramme de cas d'utilisation ? Il est maintenant si complexe (comparé à la figure 1-7) qu'il serait illusoire de penser qu'il puisse être lisible en une seule page, comme le montre le schéma suivant.

Notez que nous avons introduit un acteur généralisé abstrait « *Porteur* » en tant qu'acteur principal du fragment de cas d'utilisation s'authentifier. Cela nous permet ainsi d'inclure ce fragment dans les cas d'utilisation du client mais aussi du porteur non client.

Notez également que nous avons distingué les cas d'utilisation de l'opérateur de maintenance par le mot-clé « *support* », pour mettre l'accent sur la différence de niveau avec les cas principaux que sont ceux des porteurs de carte. Ils sont en général moins importants fonctionnellement, mais ne doivent cependant pas être oubliés. Imaginez un GAB que l'on ne pourrait pas recharger en billet !

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



Pour améliorer notre modèle, nous allons donc organiser les cas d'utilisation et les regrouper en ensembles cohérents. Pour ce faire, nous utilisons le mécanisme général de regroupement d'éléments en UML, qui s'appelle le *package*. Le package est une sorte

de dossier permettant de structurer un modèle en unités cohérentes. Les outils de modélisation du marché se servent pour la plupart de ce concept comme unité de gestion de version, de stockage, et de partage du modèle pour le travail en équipe. Nous y reviendrons plus en détail dans la partie II (modélisation statique).



EXERCICE 1-14.

Structuration des cas d'utilisation en packages

Proposez une structuration des cas d'utilisation du GAB en packages.

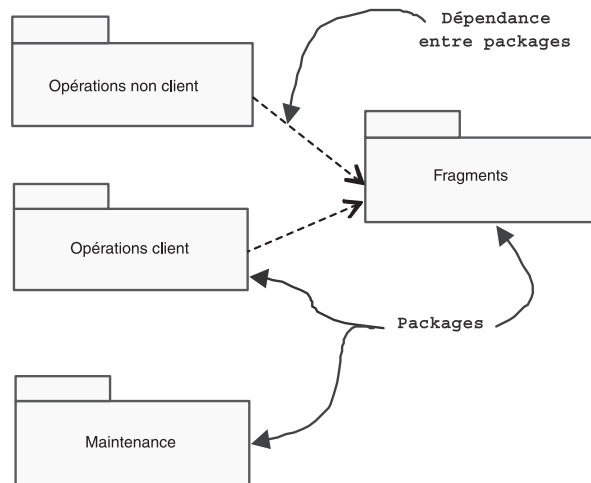
solution

Plusieurs stratégies sont possibles : procéder au regroupement par acteur, par domaine fonctionnel, etc. Dans notre exemple, un regroupement des cas d'utilisation par acteur principal s'impose, car cela permet également de répartir les acteurs secondaires.

Le cas d'utilisation inclus *S'authentifier* est mis dans un package à part, en tant que fragment commun, pour bien le distinguer des vrais cas fonctionnels qui l'incluent. Les flèches de dépendance entre packages de cas d'utilisation synthétisent les éventuelles relations entre les cas, c'est-à-dire ici les inclusions. Le schéma suivant présente la structuration proposée des cas d'utilisation. Il s'agit d'un diagramme de packages, officialisé par UML 2.

Figure 1-23.

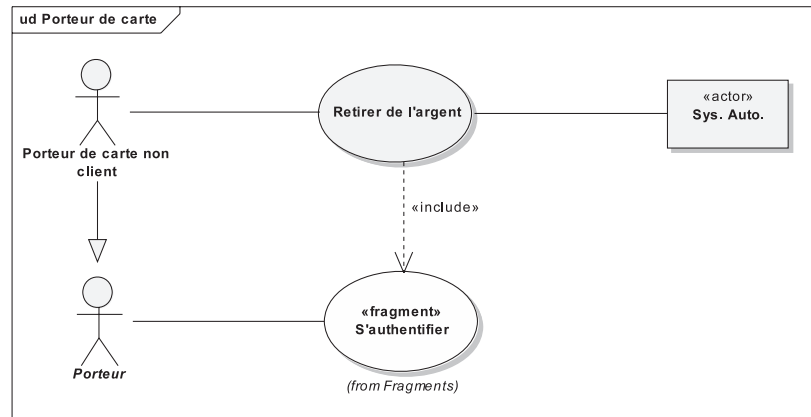
Diagramme de packages des cas d'utilisation du GAB



Il est maintenant possible de dessiner un diagramme de cas d'utilisation par packages. Cela ne présente aucune difficulté et nous donnerons uniquement le diagramme du premier package, pour montrer comment apparaît le fragment

appartenant à un package différent. La notation (*from Fragments*) n'est pas standard UML, mais est utilisée par plusieurs outils du marché.

Figure 1-24.
Diagramme de cas d'utilisation du package Opérations non client



Étape 7 – Dynamique globale : *Interaction Overview Diagram*

Dans cette dernière étape, nous allons explorer l'utilisation d'un nouveau type de diagramme UML 2 : *Interaction Overview Diagram*¹³.

Ce diagramme est une fusion intéressante du diagramme d'activité et du diagramme de séquence ! Il permet d'organiser des interactions, représentées par exemple par des diagrammes de séquence, au moyen des nœuds de contrôle du diagramme d'activité : décision, parallélisme, etc. C'est donc une sorte de diagramme d'activité¹⁴ dans lequel les actions sont remplacées par des interactions.



EXERCICE 1-15. *Interaction Overview Diagram*

Représentez la dynamique globale du GAB dans le point de vue de l'acteur Client banque, en représentant les interactions de ses cas d'utilisation dans un *Interaction Overview Diagram*. Modélisez en particulier le fait que le client peut enchaîner plusieurs transactions (retrait, dépôt, etc.) sans avoir à s'authentifier de nouveau.

13. Nous préférons conserver le nom d'origine des nouveaux types de diagrammes UML 2 en attendant qu'une traduction française s'impose (diagramme de vue globale d'interaction ? diagramme de vue d'ensemble des interactions ? etc.).

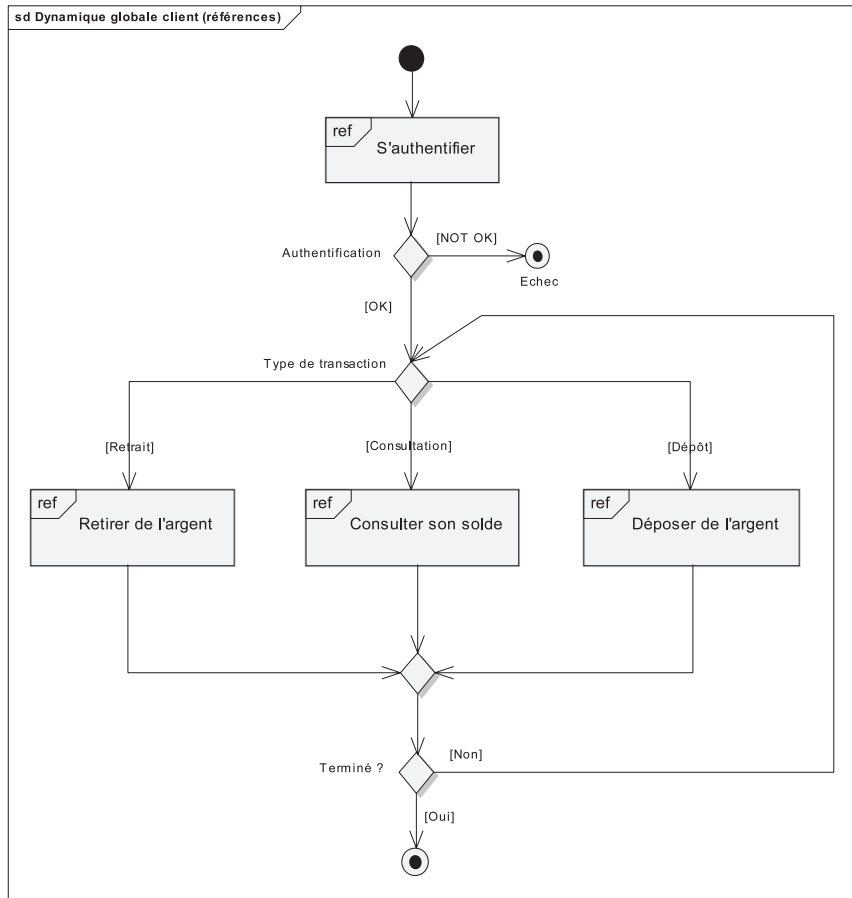
14. Pourtant, l'outil utilise le tag `Sd`, car dans les spécifications UML 2.0, ce diagramme est classé dans la catégorie des diagrammes d'interaction, et ceux-ci (séquence, communication, timing, interaction overview) sont tous représentés avec un tag `Sd`.

solution

Représentons d'abord le processus d'authentification, suivi du choix du type de transaction : retrait, consultation ou dépôt.. Le diagramme global est dessiné sur la figure 1-25.

Figure 1-25.

*Vue d'ensemble
des interactions
du client banque*



Il est également possible de remplacer chaque référence par un diagramme de séquence « inline ». Nous l'avons fait pour l'interaction *S'authentifier*. Il est clair que le souci de lisibilité du diagramme empêche de remplacer chaque référence par un diagramme de séquence entier.

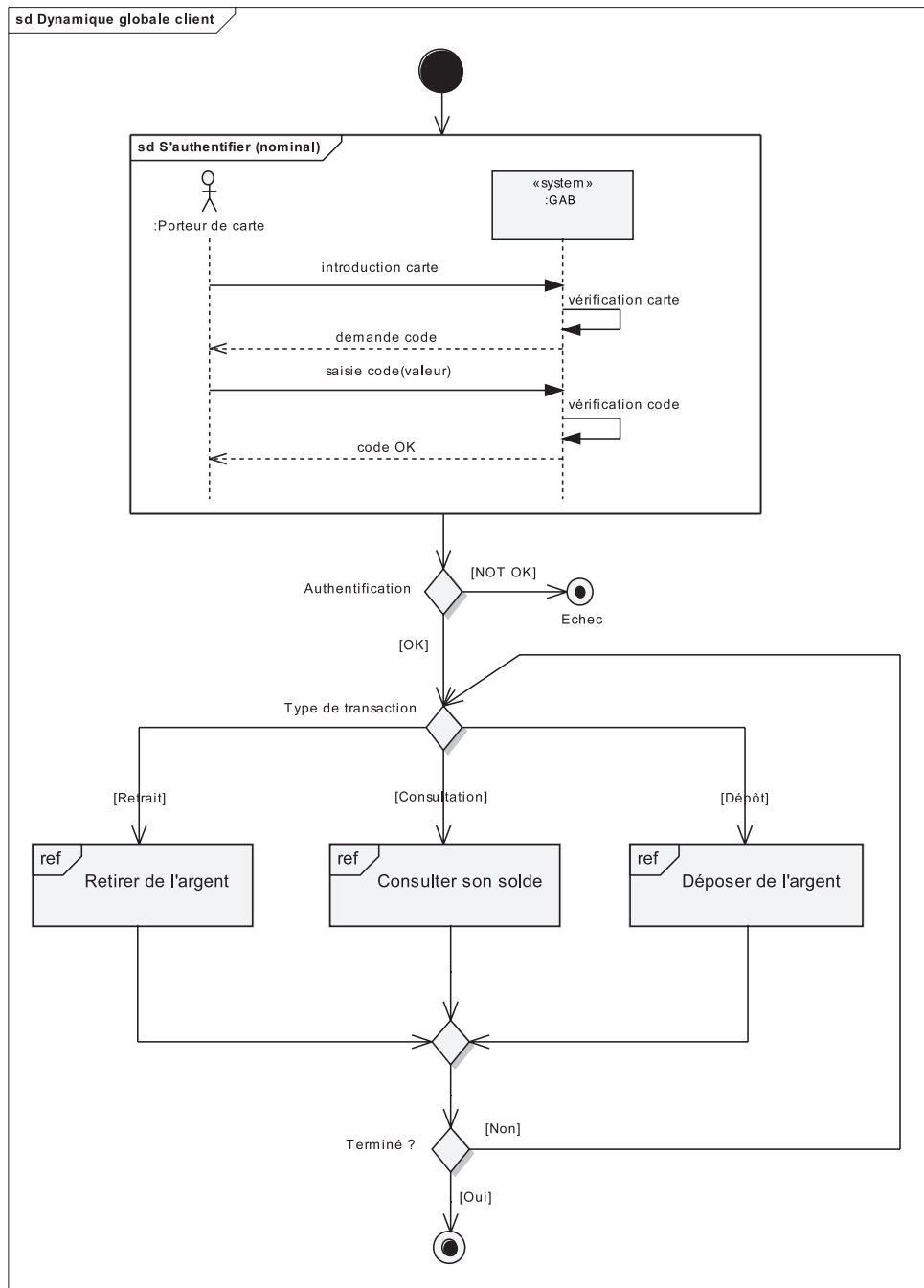


Figure 1-26.

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

