
Cas d'utilisation, une introduction

Olivier Capuozzo

Travaux de relecture: Christine Gaubert-Macon, Valérie Emin

13 Mars 2004

Les cas d'utilisation sont définis par une description textuelle, décrivant les objectifs et interactions entre le système et ses acteurs. Le format de présentation textuelle des cas d'utilisation est libre, mais il existe quelques propositions reconnues dans le domaine, notamment l'ouvrage d'Alistair Cockburn *Rédiger des cas d'utilisation efficaces* aux éditions Eyrolles).

UML propose certains concepts et formalismes dédiés à la représentation graphique des cas d'utilisation. Ainsi le système à concevoir peut-il être modélisé sous l'angle de ses responsabilités vis-à-vis de ses acteurs. Ce type de modèle, appelé Modèle des cas d'utilisation (UP/RUP), regroupe l'ensemble des cas d'utilisation du système.

Les cas d'utilisation donnent une vue d'altitude des interactions visibles d'un système, ils ne fournissent pas d'information sur la structure interne. Ils mettent en évidence les rôles de ses utilisateurs, et contribuent à catégoriser ces derniers, définir leurs attentes (objectifs du système) et obligations (pilotage du système). La recherche des cas d'utilisation permet, en particulier, de formaliser les réponses aux questions : "Pourquoi" (les intentions du système) et "Pour qui" (les acteurs).

Table des matières

1. Introduction	1
1.1. Avertissement	1
1.2. Expression des besoins	2
2. Description textuelle des cas d'utilisation	2
2.1. Format de présentation pour les cas d'utilisation	2
2.2. Niveaux d'objectif	5
2.3. Portée des cas d'utilisation	5
3. Recommandations pour l'écriture des cas d'utilisation	6
3.1. Douze recommandations d'écriture	6
3.2. Exemple	6
4. Exercices	8
4.1. Remplacer sa photo	8
4.2. Transférer de compte à compte	10
5. Représentation graphique des cas d'utilisation avec UML	12
5.1. Introduction	12
5.2. Éléments graphiques UML	12
5.3. Relations entre cas d'utilisation	13
5.4. Relation entre acteurs	14
5.5. Classification des acteurs	15
6. Conclusion	16
7. Bibliographie	16

1. Introduction

1.1. Avertissement

Le terme *use case* (cas d'utilisation) a été introduit par Ivar Jacobson en 1992 et fait partie intégrante d'UML. Toutefois, UML ne fournit pas de consignes d'écriture et ne précise pas de format de présentation textuelle pour les cas d'utilisation. Nous nous référons pour cela aux travaux de Alistair Cockburn (Cockburn, 2001) où dans la préface, il rappelle que l'écriture d'un cas d'utilisation est essentiellement du domaine de « l'écriture d'essais en prose, avec toutes les difficultés d'expression que suppose en général ce type d'exercice ». UML dispose d'une batterie de concepts et de formalismes graphiques (diagrammes de séquence, état-transition...) très utiles pour préciser un cas d'utilisation. Les symboles UML dédiés aux cas d'utilisation sont présentés au paragraphe UML et représentation des cas d'utilisation.

Ce document a pour objectif de sensibiliser le lecteur à une approche d'expression des besoins dans un contexte de développement d'applications piloté par les cas d'utilisation.

Les courtes définitions ci-dessous sont issues des spécifications UML.

Un **cas d'utilisation** spécifie une séquence d'actions, avec variantes éventuelles, réalisée par le système en interaction avec des acteurs du système.

Un **acteur** est une entité externe au système, en interaction avec ce dernier. L'entité est un rôle joué par un utilisateur, par exemple un comptable, ou par un autre système, un capteur par exemple.

Un **scénario** est un *chemin particulier au travers de la description abstraite et générale fournie par le cas d'utilisation* (Muller, 2003).

Les cas d'utilisation décrivent le comportement attendu des utilisateurs et du système. Le système offre des services à ses utilisateurs. La grande difficulté, pour le rédacteur de cas d'utilisation, est de se maintenir à ce niveau d'observation et de description (service), se centrant principalement sur le comportement superficiel du système, le POURQUOI. Il n'est toutefois pas interdit de modéliser l'état interne du système.

Il en résulte un ensemble de documents **compréhensibles** par tout acteur du projet.

Les cas d'utilisation permettent de ne pas perdre de vue les objectifs du système en construction. Ils pilotent le projet tout au long de sa vie. Ils sont le fil conducteur de tous les acteurs du projets (analystes, architectes, qualitatifs, managers, utilisateurs, clients, développeurs, testeurs...)

1.2. Expression des besoins

Les cas d'utilisation sont recueillis, organisés, décomposés dans la phase d'acquisition des besoins du projet. Qu'attendent les utilisateurs du futur système ? Quel est le comportement attendu du système face à ces requêtes?

Etant donné que, par définition, l'utilisateur est le mieux placé pour parler de ses besoins, il est nécessaire d'utiliser un formalisme clair qui permet de faire participer à cette phase des non informaticiens (représentant d'utilisateurs, experts du domaine ou autres managers...).

UML propose un ensemble de symboles graphiques dédiés à la représentation des cas d'utilisation. L'usage excessif de ces symboles, que seuls les informaticiens peuvent décoder, va parfois à l'encontre des objectifs visés.

En revanche, il est courant de préciser des cas d'utilisation par des diagrammes d'activités, de séquence et d'état/transition.

2. Description textuelle des cas d'utilisation

2.1. Format de présentation pour les cas d'utilisation

UML n'impose ni ne préconise aucun format particulier de description textuelle des cas d'utilisation.

Un cas d'utilisation est composé de deux grandes parties : 1/ La description des interactions dans un cas typique de succès (le cas nominal) accompagné d'informations de contexte, 2/ les variations du

cas (cas particuliers, extensions), contraintes diverses, questions ouvertes et 3/ diverses illustrations comme des diagrammes de séquence système par exemple. A ce sujet, Craig Larman (Larman, 2003) détaille également la notion de *contrats d'opération système* .

L'idée est de fournir un format de présentation textuelle à la fois souple et riche. En nous inspirant du livre d'Alistair Cockburn sur la question (CockBurn, 2003) , nous retiendrons le format suivant :

- **Cas d'utilisation** : <ici le nom du cas d'utilisation>

Acteur : <ici le nom du ou des acteurs principaux, déclencheurs du cas>

[Événement déclencheur] : <ici l'événement déclencheur du cas>

Parties prenantes et intérêts : <listes des parties prenantes et leurs intérêts>

Niveau : <ici une valeur parmi { *Stratégique, Objectif utilisateur, Sous-fonctionnalité* }>

Portée : <ici la portée du cas> (voir plus loin)

[Pré-conditions] : <ici les pré-conditions éventuelles>

[Post-conditions] : <ici les post-conditions éventuelles>

Scénario nominal

1. <description de l'action>
2. <description de l'action>
3. ...
4. <description de la dernière action avant la fin du cas>

[Extensions]

- <numéro de l'étape> : <condition> : <action ou sous-cas d'utilisation>
- <numéro de l'étape> : <condition> : <action ou sous-cas d'utilisation>

[Contraintes]

...

[Questions-ouvertes]

...

[Annexes]

...

Les titres entre crochets sont considérés comme optionnels. Il est bien entendu que ce n'est qu'un **simple recommandation** . En phase d'élaboration, lors de la découverte des besoins, il convient de ne pas chercher l'exhaustivité dans la rédaction des cas d'utilisation.

On peut très bien rédiger un cas ne comportant que son nom suivi des différentes étapes. Plusieurs exemples de rédaction sont présentés dans ce document, mais vous gagnerez à aller voir <http://alistair.cockburn.us/> [1].

[1] <http://alistair.cockburn.us/usecases/usecases.html>

Exemple

Un établissement public d'enseignement technique met en place un système de portes ouvertes permanentes sur la toile. Les visiteurs sont invités à communiquer leurs coordonnées et divers renseignements.

- **Cas d'utilisation** : Communiquer des renseignements

Acteur : visiteur

Parties prenantes et intérêts :

- **Visiteur** : il veut communiquer ses coordonnées à l'établissement afin d'être contacté ultérieurement sur les formations qu'il aura sélectionnées.
- **Service Administratif** : il veut pouvoir contacter, en temps voulu, les personnes ayant manifesté un intérêt pour les formations disponibles dans l'établissement.

Niveau : *Objectif utilisateur*

Portée : Système Porte Ouverte

Pré-conditions : aucune

Post-conditions : Les données communiquées par le visiteur sont accessibles par le service administratif.

Scénario nominal

1. Le visiteur communique au système un choix de formation.
2. Le système communique un formulaire d'identification.
3. Le visiteur s'identifie.
4. Le système présente un formulaire correspondant à la formation choisie.
5. Le visiteur renseigne le formulaire et le soumet au système.
6. Le système lui confirme l'enregistrement des informations.

Présentation des items

- Les **acteurs** , ce sont les acteurs déclencheurs du cas.
- **La liste des parties prenantes et intérêts** met l'accent sur les objectifs métier du cas, généralement les acteurs bénéficiaires du cas.
- Le **niveau** nous donne une indication de lecture (les détails que l'on s'attend ou non à trouver).
- La **portée** délimite le périmètre du cas.
- Les **pré-conditions** définissent les conditions de déclenchement du cas.
- Les **post-conditions** déterminent ce qui est vrai après l'exécution du scénario nominal réalisé avec succès.

- Le **scénario nominal** est un scénario typique de succès. Il raconte, étape par étape, l'histoire des interactions entre acteur et système, dans le meilleur des cas, ou plus précisément dans le meilleur des *scénarios*.
- Les **extensions** correspondent aux autres scénarios possibles, avec branchements et incidents de parcours, conduisant aussi bien au succès qu'à l'échec. Ils sont de la forme <condition : étapes numérotées >.
- **contraintes** : ce sont les contraintes non fonctionnelles, telles que temps de réponse, capacité de montée en charge, confidentialité, date de disponibilité, format de stockage des données...
- **questions ouvertes** : tout questionnement susceptible de pointer des zones d'ombre.
- **Annexe** : Illustrations et autres informations concernant le cas.

On peut, ici, présenter une instance du cas d'utilisation (scénario) sous la forme d'un diagramme de séquence système (DSS) par exemple. Un DSS (Larman, 2003) est un diagramme de séquence de haut niveau représentant les interactions entre les acteurs externes et le système, ce dernier étant vu comme une boîte noire (voir exemple de DSS [7]).

2.2. Niveaux d'objectif

Un cas d'utilisation, comme tout diagramme UML, permet de décrire une réalité selon différents niveaux de raffinement. Il convient, entre autres, de signaler le "niveau d'abstraction" de la vue afin de permettre au lecteur une meilleure interprétation de ce qui est et n'est pas montré. Concernant les cas d'utilisation, nous parlons alors de *niveaux d'objectif*.

Cockburn (Cockburn, 2003) définit trois niveaux d'objectif :

- **Niveau stratégique**

Présente le contexte général, les grandes fonctions du système (approche métier), ses objectifs. Un cas d'utilisation de niveau stratégique implique plusieurs objectifs utilisateur et s'étale généralement sur plusieurs jours, semaines, mois ou années.

Exemple : 1/ L'étudiant s'inscrit à une formation. 2/ L'étudiant s'inscrit à des modules 3/ L'étudiant passe un examen partiel 4/ L'étudiant obtient son diplôme.

- **Niveau objectif utilisateur**

C'est l'objectif suivi par un acteur en interaction avec le système. L'objectif utilisateur est celui qui représente le plus d'intérêt. Il correspond au *processus métier élémentaire* en ingénierie des processus métier (Cockburn, 2003). Sa durée, de 2 à 20 minutes, peut être réduite si le déclencheur est un système.

Exemple : Inscription à un module.

- **Niveau sous-fonctionnalité**

Ce sont des cas d'utilisation qui participent au bon déroulement de cas d'utilisation de niveau objectif utilisateur. Un cas d'utilisation de sous-fonctionnalité remplit un objectif partiel d'un cas d'utilisation d'objectif utilisateur, auquel il est lié par une relation d'inclusion (<<include>>).

Exemple : Identifier un utilisateur.

2.3. Portée des cas d'utilisation

Alors que le niveau d'objectif renseigne sur la granularité (verticalité, niveau d'abstraction), la portée renseigne sur l'impact (horizontalité, périmètre).

Trois sortes de portées sont fréquemment utilisées :

- **Entreprise** : Concerne les processus généraux d'une organisation.
- **Système** : Centré sur le logiciel à concevoir.
- **Sous-système** : Restreint à une partie seulement du logiciel, par exemple un *framework* , une couche technique ou un sous-ensemble fonctionnel.

3. Recommandations pour l'écriture des cas d'utilisation

Le coeur de la rédaction d'un cas d'utilisation se situe dans la description des étapes du scénario nominal élu. Le style d'écriture doit être clair, précis et concis, sans ambiguïté.

3.1. Douze recommandations d'écriture

La description des différentes étapes sert à la fois l'utilisateur et le développeur. Le langage naturel doit être un minimum "guidé" afin que tout le monde puisse s'y retrouver. N'oublions pas que le développeur en déduit un système déterministe qui doit satisfaire l'utilisateur final.

Les recommandations essentielles présentées ci après, sont largement discutées dans l'ouvrage de Cockburn (Cockburn, 2003). En voici un résumé :

- R1 : Partir du sommet (les grandes fonctions), et se maintenir le plus possible au niveau *objectif utilisateur*.
- R2 : Centrer son attention sur le cas nominal (un scénario typique de succès).
- R3 : Préciser toujours les parties prenantes et leurs intérêts.
- R4 : Utiliser un verbe au présent de l'indicatif à chaque étape.
- R5 : Utiliser la voie active pour décrire les sous-objectifs en cours de satisfaction.
- R6 : Le sujet doit être clairement localisable (en début de phrase généralement).
- R7 : Rester concis et pertinent (éviter les longs documents).
- R8 : Eviter les *si* , et placer les comportements alternatifs dans les *extensions* .
- R9 : Signaler les sous-cas d'utilisation. Ils sont toujours représentés par la relation d'inclusion `include [13]` d'UML.
- R10 : Identifier le bon objectif.
- R11 : Signaler la portée.
- R12 : Laisser de côté l'interface utilisateur.

3.2. Exemple

Nous allons compléter la description du cas d'utilisation vu précédemment : un établissement public d'enseignement technique met en place un système de portes ouvertes permanentes sur la toile. Les visiteurs sont invités à communiquer leurs coordonnées et divers renseignements.

- **Cas d'utilisation** : Communiquer des renseignements

Acteur : visiteur

Parties prenantes et intérêts :

- **Visiteur** : il veut communiquer ses coordonnées à l'établissement afin d'être contacté ultérieurement sur les formations qu'il aura sélectionnées.
- **Service Administratif** : il veut pouvoir contacter, en temps voulu, les personnes ayant manifesté un intérêt pour les formations disponibles dans l'établissement.

Niveau : *Objectif utilisateur*

Portée : Système Porte Ouverte

Pré-conditions : aucune

Post-conditions : Les données communiquées par le visiteur sont accessibles par le service administratif.

Scénario nominal

1. Le visiteur communique au système un choix de formation.
2. Le système communique un formulaire d'identification.
3. Le visiteur s'identifie.
4. Le système présente un formulaire correspondant à la formation choisie.
5. Le visiteur renseigne le formulaire et le soumet au système.
6. Le système enregistre les informations et signale le succès de l'opération.

Extensions

- *. A tout moment : le visiteur peut abandonner l'opération en cours.
- 1a. Le visiteur est déjà authentifié : aller en 4
- 2a. Le visiteur n'est pas connu du système : sous-cas d'utilisation *Création visiteur*
- 5a. Certains champs obligatoires ne sont pas renseignés : aller en 4.

Contraintes

- **Temps de réponse** : La réception d'une réponse à l'identification (étape 4) ne doit pas dépasser 3 secondes.

Annexe

Illustration du cas par un scénario représenté par un diagramme de séquence système (DSS). Dans ce scénario, le visiteur est connu du système et n'abandonne pas l'opération.

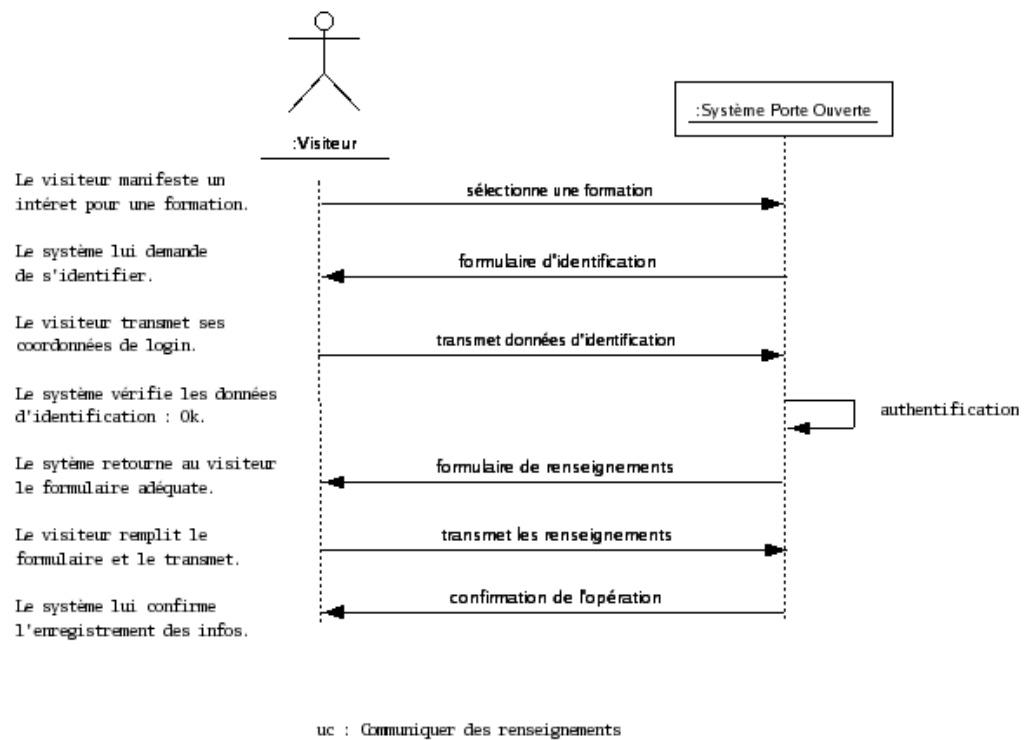


Figure 1. Diagramme de séquence système

4. Exercices

4.1. Remplacer sa photo

4.1.1. Enoncé

Réécrire ce cas d'utilisation, en respectant les recommandations d'écriture.

Un employé souhaite changer sa photo stockée dans l'annuaire LDAP de son organisation.

1. L'utilisateur fournit son login et son mot de passe.
2. Si le système le reconnaît, alors aller en 5.
3. Si le système ne le reconnaît pas, alors aller en 1.
4. Le système transmet la photo à l'utilisateur.
5. L'utilisateur peut sélectionner une nouvelle photo, via un *explorateur* et une *liste déroulante*.
6. La photo est transmise au système.
7. Si la photo est au bon format, le système remplace l'ancienne par la nouvelle.
8. Le système informe l'utilisateur que l'opération s'est bien déroulée.

4.1.2. Une solution

- **Cas d'utilisation** : *Remplacer sa photo*

Acteur : Employé

Parties prenantes et intérêts :

- **Employé** : Souhaite changer sa photo d'identité.
- **Entreprise** : Souhaite disposer d'une collection de photos de ses employés continuellement mise à jour.

Niveau : *Objectif utilisateur*

Portée : Gestion des informations des employés

Scénario nominal

1. Le système communique un formulaire d'identification.
2. L'employé renseigne les champs et valide le tout.
3. Le système transmet la photo de l'employé.
4. L'employé sélectionne une autre photo qu'il transmet au système.
5. Le système remplace l'ancienne photo par la nouvelle.
6. Le système confirme le succès de l'opération.

Extensions

- 1 à 4, L'employé peut annuler l'opération :
 - Le cas s'arrête.
- 3a. l'employé n'est pas connu du système :
 - Retour en 1.
- 3b. l'employé n'a pas encore de photo :
 - Le système transmet l'image d'un pingouin ;-).
- 4, le système ne reconnaît pas le format de l'image transmise par l'employé :
 - Le système en informe l'employé.
 - Retour en 3.

Questions ouvertes

- Faut-il se préoccuper du poids de l'image transmise par l'employé ? (du point de vue système de réception et système de stockage).

Quelques commentaires

- Les traitements conditionnels sont placés dans les **extensions** .
- Nous remarquerons que la référence au moyen de stockage (LDAP) est absente de la solution. Selon le cas, le serveur LDAP pourrait apparaître dans les **contraintes** .
- Les composants IHM (*explorateur, liste déroulante*) ne sont plus mentionnés.
- La voie active remplace la voie passive (étape 3 de la solution).
- La rédaction des extensions favorise la découverte de problèmes à venir, par exemple le cas "absence de photo" et le problème du poids. L'un est réglé par une extension, l'autre fait l'objet d'une question ouverte.

4.2. Transférer de compte à compte

4.2.1. Enoncé

Réécrire ce cas d'utilisation, en respectant les recommandations d'écriture.

Une personne, décide de réaliser des transferts de compte à compte.

1. Le client se connecte au système pour une opération de transfert.
2. Le système le reconnaît et lui retourne la liste de ses comptes.
3. Le client sélectionne un de ses comptes.
4. Le système lui transmet le solde de ce compte.
5. Le montant est saisi par le client.
6. Le client sélectionne un autre compte (compte de destination).
7. Si le montant est supérieur au solde du compte, le système en informe le client qui doit alors modifier le montant.
8. Le système demande confirmation du transfert au client.
9. Le client confirme.
10. Le nouveau solde du compte sélectionné est communiqué au client.
11. Si le client le souhaite, le client retourne en 3.

4.2.2. Une solution

- **Cas d'utilisation** : *Transférer de compte à compte*

Acteur : Client

Parties prenantes et intérêts :

- **Client** : Souhaite gérer ses comptes.
- **Banque** : Souhaite soulager les opérations guichet, en déléguant au titulaire de comptes la gestion de ses derniers pour des opérations courantes.

Post-condition : *Le solde global des comptes reste inchangé.*

Niveau : *Objectif utilisateur*

Portée : Auto-gestion des comptes client

Scénario nominal

1. Le client se connecte (sous cas d'utilisation : "identification client").
2. Le système retourne la liste de ses comptes.
3. Le client sélectionne un de ses comptes (compte source).
4. Le système lui transmet le solde de ce compte.
5. Le client saisit un montant à transférer (du compte sélectionné) et un compte de destination.
6. Le système demande confirmation du transfert au client.
7. Le client confirme.
8. Le système communique le solde des comptes sélectionnés.

--- le client peut retourner au point 2 pour poursuivre les opérations de transferts ---

Extensions

- 1. Le client ne peut se connecter :
 - Le cas s'arrête.
- 6. Le montant du transfert est plus élevé que le solde du compte source :
 - Le système informe le client du dépassement.
 - Retour en 2.
- 7. Le client ne confirme pas :
 - Retour en 2.

Quelques commentaires

- Le scénario alterne action de l'acteur et réponse du système. Un trop grand déséquilibre de cette séquence type (acteur/système), par exemple peu d'actions utilisateur et beaucoup de réponses système (ou inversement), peut être le signe d'une mauvaise rédaction.
- La boucle est notée sous forme de commentaire (et non comme étape) dans la partie descriptive du scénario.

5. Représentation graphique des cas d'utilisation avec UML

5.1. Introduction

Un dessin vaut-il mieux qu'un court discours ? Non certainement pas dans le cas présent où les cas d'utilisation participent activement au cahier des charges d'un projet informatique.

UML propose de représenter les cas d'utilisation sous une forme graphique nommée *diagramme de cas d'utilisation* appartenant au modèle des besoins. Les rôles sont définis pour chaque acteur. Une relation entre acteurs et cas représente une communication entre l'acteur et le cas. Le cas (d'utilisation) est représenté par une ellipse qui porte son nom (à l'intérieur ou en dessous). Des notes peuvent être portées sur le diagramme afin d'y ajouter des informations.

Un diagramme de cas d'utilisation montre acteurs et cas d'utilisation ensemble avec leur relations. La relation entre un acteur et un cas d'utilisation est appelée association et correspond au fait que l'acteur participe à un cas d'utilisation. Les cas d'utilisation représentent les fonctionnalités d'un système, ou d'une entité d'un système, telles qu'elles sont sollicitées en interaction avec des événements extérieurs. Ils donnent une vision "haute" et dynamique du système.

5.2. Éléments graphiques UML

- **Acteur** : un bonhomme en fil de fer (*stick man*) ou une classe stéréotypée <<actor>>. Son rôle est décrit sous ses pieds.
- **Cas d'utilisation** : une ellipse. Le nom du cas d'utilisation est placé soit dans l'ellipse soit en dessous.
- **Association** : (participation d'un acteur à un cas d'utilisation) un trait plein pouvant être orienté (pointe de flèche) et décoré (multiplicité).

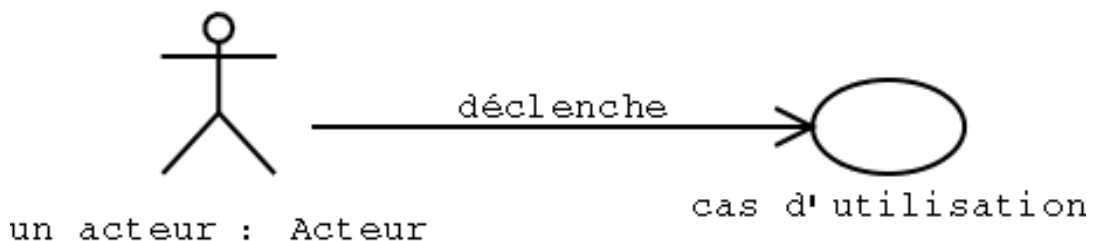


Figure 2. Cas d'utilisation et acteur

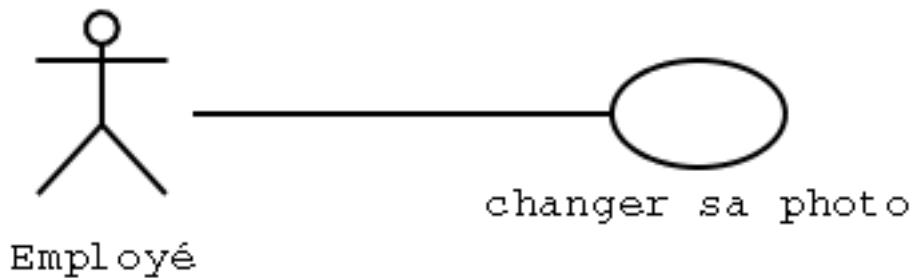


Figure 3. Exemple d'un cas d'utilisation : Changer sa photo

5.3. Relations entre cas d'utilisation

UML propose trois types de relations standard entre cas d'utilisation, `<<include>>`, `<<extend>>` et **généralisation**. Les deux premières sont représentées par un stéréotype de dépendance, l'autre étant la relation de généralisation représentée en UML par une flèche creuse à pointe fermée.

`<<include>>` : Stéréotype représentant le fait qu'un cas d'utilisation inclut un autre cas d'utilisation. On utilise ce stéréotype lorsque que l'on souhaite factoriser un cas d'utilisation partagé par plusieurs autres cas d'utilisation. *Par exemple, une opération de retrait et une opération de transfert nécessitent toutes deux une opération de vérification de l'identité du client.*

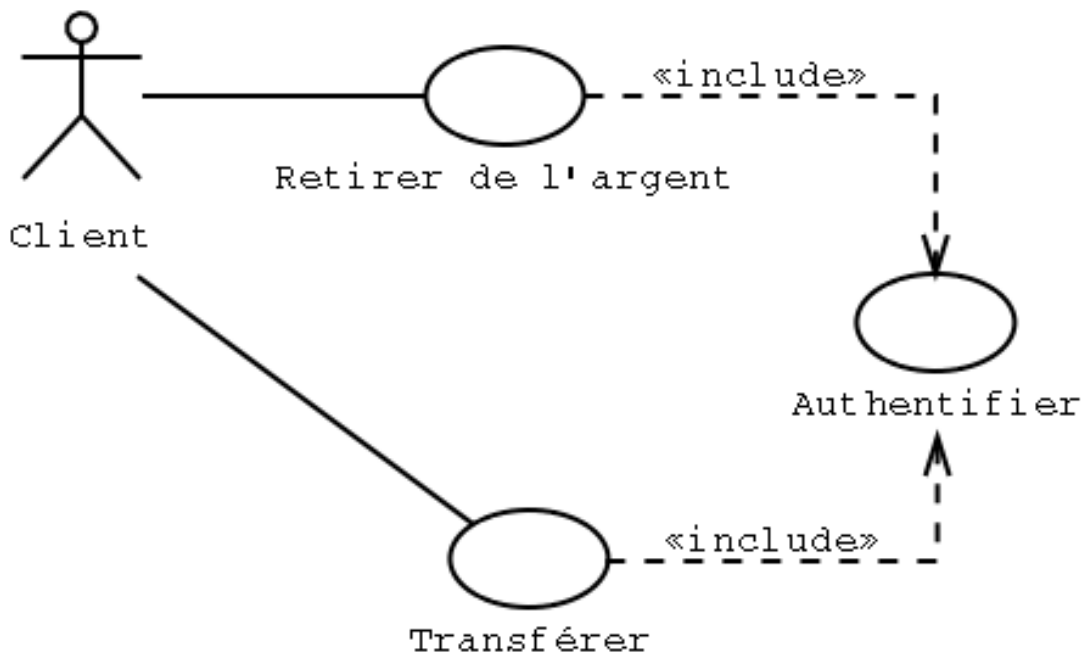


Figure 4. Exemple de relation "include"

`<<extend>>` : Un cas d'utilisation peut déclarer des points d'extension (*extension point*). Un point d'extension localise un endroit (un point) unique dans le cas d'utilisation. C'est dans les limites de ce point que d'autres cas d'utilisation pourront étendre (*extend*) le comportement initial du cas d'utilisation. C'est un moyen pratique de mettre en avant une fonctionnalité optionnelle. *Par exemple, lors de la conception d'un site marchand pour un fabricant de produit de beauté, on souhaite proposer à certains visiteurs de promouvoir la marque dans leur région.*

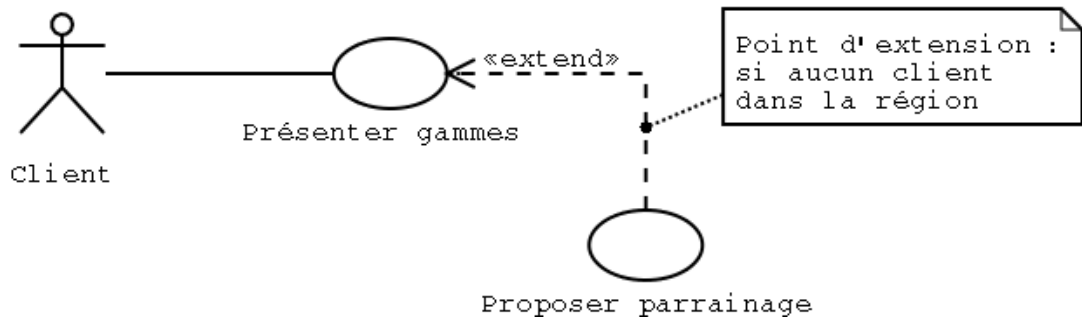


Figure 5. Exemple de relation "extend"

Généralisation : Une relation de généralisation d'un cas d'utilisation B vers un cas d'utilisation A signifie que B est une spécialisation de A. Contrairement aux deux autres relations, la relation de généralisation n'est pas un stéréotype. Elle indique qu'un cas d'utilisation est une variation d'un autre. Cette relation se différencie de <<extend>> par le fait que le cas d'utilisation peut varier *en tout point* de celui hérité. Par exemple dans l'UC "Retirer de l'argent", si il s'agit de retirer de l'argent sur un compte sur livret le comportement de l'UC peut être tout à fait différent.

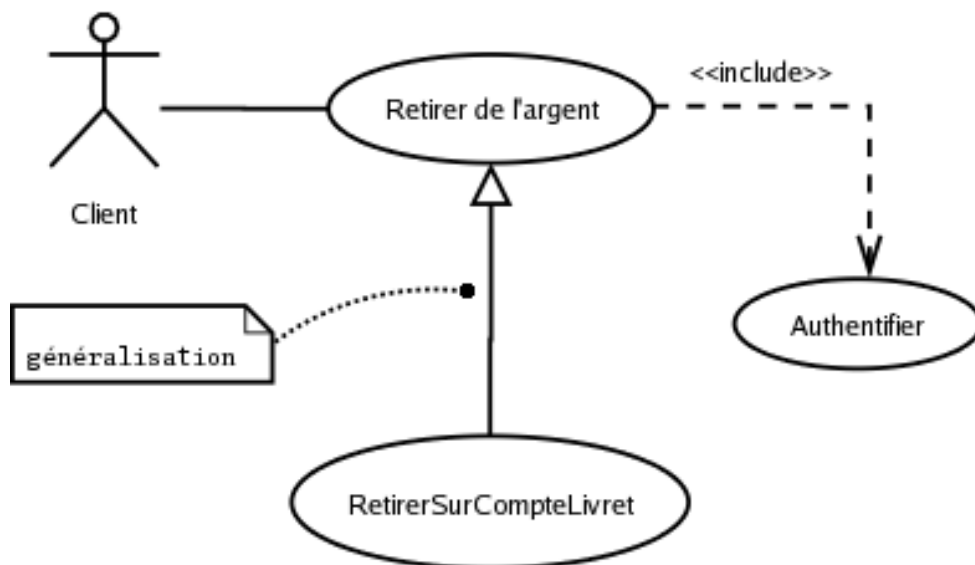


Figure 6. Exemple de relation de généralisation

5.4. Relation entre acteurs

La relation de **généralisation** est applicable dans le cas où un rôle est une spécialisation (**une sorte de**) d'un autre.

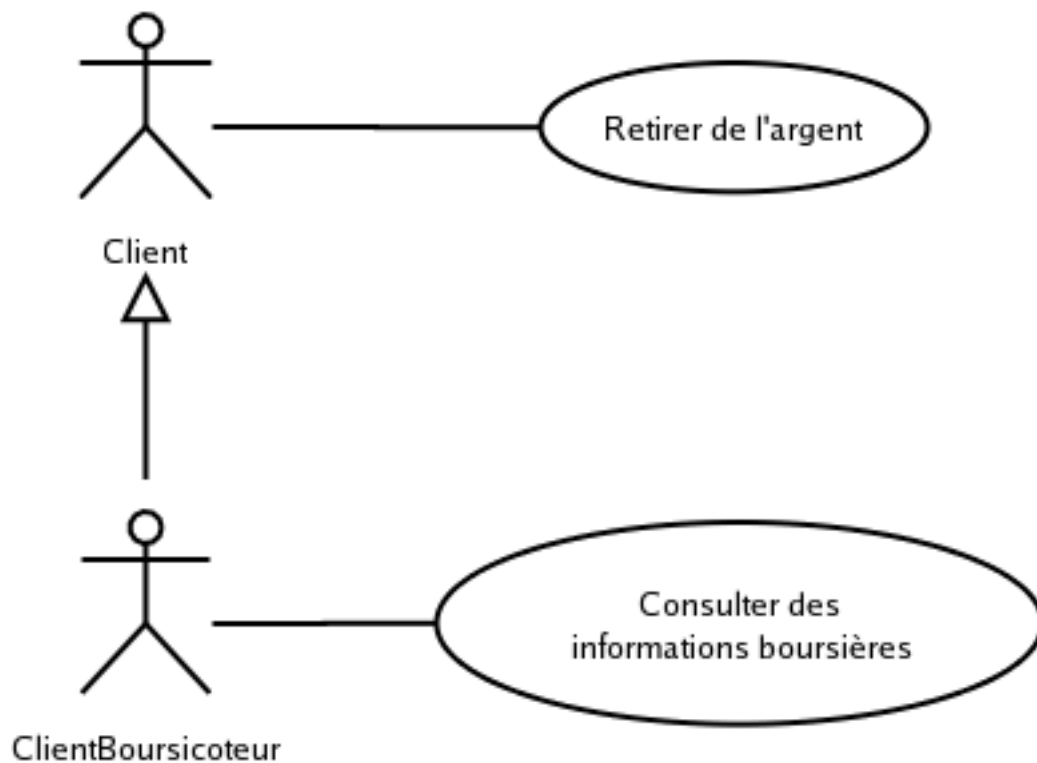


Figure 7. Exemple de la relation entre acteurs

Un client "boursicoteur" est un client comme un autre, mais ayant en plus la possibilité de consulter des informations boursières.

5.5. Classification des acteurs

Les acteurs sont des entités en interaction avec le système. Le niveau de détail de présentation d'un cas d'utilisation correspond à la vision de l'acteur auquel il est relié.

Il est d'usage, mais absent de la norme UML, de distinguer les acteurs principaux des acteurs secondaires. Les fonctionnalités principales du système ont été définies pour les *acteurs principaux*. Afin d'atteindre cet objectif, il est en général nécessaire de réaliser des opérations en amont et en aval de ces fonctions principales. C'est le rôle des *acteurs secondaires*. Cela peut être par exemple la gestion des droits utilisateurs, la sauvegarde de la base de données, etc.

Dans bien des cas, cette classification s'avère suffisante, toute fois certains professionnels proposent de l'affiner :

Un acteur peut être humain ou purement informatique, *hardware* ou *software* (voir les articles de Thierry Cros [2] sur le thème des cas d'utilisation et sur l'approche objet en général).

Un acteur peut avoir de multiples "personnalités", jouer plusieurs rôles dans un ou plusieurs cas d'utilisation; on en identifie quatre (Miller, 2001) : **initiateur**, **serveur**, **receveur** et **facilitateur**.

- **Initiateur** : Rôle joué par un acteur qui déclenche le cas, qui met en mouvement le système.
- **Serveur** : Rôle joué par un acteur lorsqu'il aide le système à assumer ses responsabilités.
- **Receveur** : Rôle joué par un acteur lorsqu'il reçoit des informations du système (par exemple un

[2] <http://tcros.free.fr/articles.htm>

SGBD ou un système de *backup*).

- **Facilitateur** : Rôle joué par un acteur lorsqu'il réalise une action pour le bénéfice d'un autre (un guichetier pour un client par exemple).

Par convention, le lecteur s'attend à trouver, dans un diagramme de cas d'utilisation, les acteurs principaux à gauche du contexte des ellipses et les acteurs secondaires à droite, comme l'illustre l'exemple ci-dessous.

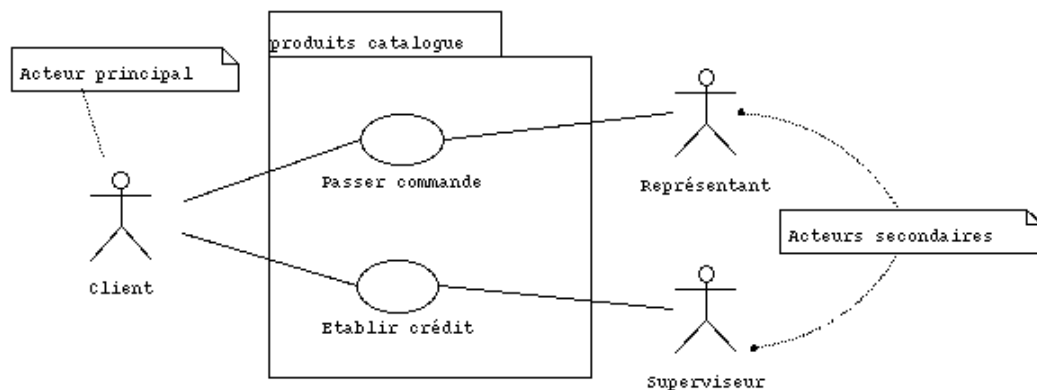


Figure 8. Exemple d'acteurs

6. Conclusion

Les cas d'utilisation jouent un rôle fondamental dans le cycle de vie d'un projet de développement logiciel.

En phase initiale, ils permettent d' **identifier les utilisateurs** et de comprendre leurs **attentes** .

En phase d'élaboration, les développeurs s'appuient sur eux pour découvrir les objets métier et constituer le **modèle de domaine** . Bertrand Meyer met en garde à ce sujet : il ne faut pas tomber dans le piège d'une conception descendante du système qui consisterait à *déduire l'architecture du système directement de l'analyse, ce qui serait en exacte opposition avec une conception orientée objet.* (voir l'article [3]).

En conception et en implémentation, les cas d'utilisation font office de garde-fou auprès des développeurs, leur permettant de garder en **ligne de mire** les objectifs utilisateur.

Les cas d'utilisation sont d'une grande utilité pour la conception des **tests fonctionnels** et de la **documentation** utilisateur.

7. Bibliographie

- Ambler, S.W. (2003) *The Elements of UML Style* , Londres : Cambridge University Press.
 - Cockburn, A. (2001). *Rédiger des cas d'utilisations efficaces* , Paris : Eyrolles. (voir aussi : <http://alistair.cockburn.us/> [4])
 - Meyer, B. (1997). *UML: the positive spin* ,
- [3] <http://www.eiffel.com/doc/manuals/technology/bmarticles/uml/page.html>
 [4] <http://alistair.cockburn.us/usecases/usecases.html>

<http://archive.eiffel.com/doc/manuals/technology/bmarticles/uml/page.html> [5]

- Muller, PA., Gaertner, N. (2003). *Modélisation objet avec UML*, Paris : Eyrolles.
- Muller, PA. (). Didacticiel de modélisation objet avec UML, <http://magda.elibel.tm.fr/refs/UML/didacticiel.pdf> [6]
- Larman, C. (2003). *UML et les design patterns*, Paris : Campus Press.
- Miller, G. (2001). *Introduction to diagramme sequence*, <http://www-106.ibm.com/developerworks/library/j-jmod0508/index.html> [7]
- OMG (2003). *Spécification UML*, <http://www.omg.org/technology/documents/formal/uml.htm> [8]

Les adresses de sites indiquées sont valides en date du 15 Avril 2004.

[5] <http://archive.eiffel.com/doc/manuals/technology/bmarticles/uml/page.html>

[6] <http://magda.elibel.tm.fr/refs/UML/didacticiel.pdf>

[7] <http://www-106.ibm.com/developerworks/library/j-jmod0508/index.html>

[8] <http://www.omg.org/technology/documents/formal/uml.htm>