

Chapitre IV : Diagrammes comportementaux d'UML 2



1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un diagramme de comportement qui représente le comportement attendu d'un système. Les diagrammes des cas d'utilisation constituent un moyen permettant de présenter les principaux acteurs et de recueillir, d'analyser, d'organiser, et de recenser les importantes fonctionnalités d'un système. autrement dit, les diagrammes de cas d'utilisation sont destinés à représenter les besoins des utilisateurs par rapport au système.

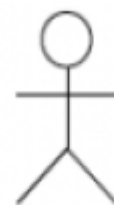
Les différentes étapes de la démarche à adopter afin d'aboutir à un diagramme de cas d'utilisation sont:

- identifier le système à modéliser;
- Identifier les acteurs du système;
- Identifier les cas d'utilisation associés aux différents acteurs;
- Ajouter les relations entre cas d'utilisation et/ou relations entre acteurs.

1.1. Acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) au système étudié et qui interagit avec celui-ci. Une même entité concrète peut jouer différents rôles par rapport au système étudié et être modélisée par plusieurs acteurs. Réciproquement, le même rôle peut être tenu simultanément par plusieurs entités concrètes, qui seront modélisée par le même acteur.

Un acteur est représenté graphiquement par un petit bonhomme portant son nom (son rôle).



Nom de l'acteur

Représentation UML d'un acteur

1.1.1. Acteurs principaux et secondaires

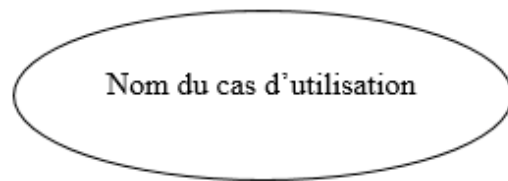
Nous distinguons deux catégories d'acteurs à savoir:

- **Acteurs principaux** : acteurs utilisant le système (ie. les acteurs sont à l'initiative des échanges nécessaires pour réaliser le cas d'utilisation)
- **Acteurs secondaires**: acteurs administrant le système (ie, les acteurs paramètrent le système en lui fournissant les informations nécessaires)

1.2. Cas d'utilisation

Pour chaque acteur identifié, il convient de rechercher comment et pourquoi chaque acteur se sert du système, autrement dit, il s'agit de recenser les différentes fonctionnalités selon lesquelles il utilise le système. En effet, un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Cette fonctionnalité souvent nécessite une série d'actions plus élémentaires.

Un cas d'utilisation se représente graphiquement par un ovale contenant l'intitule du cas (un verbe à l'infinitif).

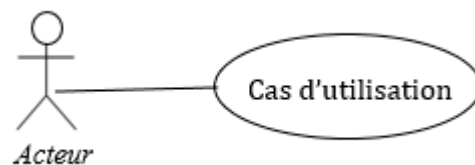


Représentation UML d'un cas d'utilisation

1.3. Relations/ Interactions

Les acteurs sont reliés aux cas d'utilisation qu'ils peuvent déclencher par une relation qui signifie simplement : « participe à » appelée interaction ou relation.

L'interaction est représentée par un trait plein continu entre acteurs et cas d'utilisation.



Représentation UML d'une interaction

1.4. Système

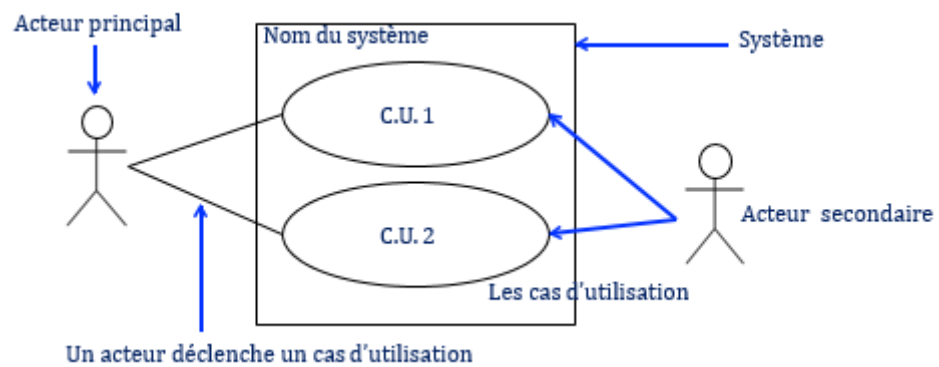
Les cas d'utilisation peuvent être contenus dans un cadre qui représente les limites du système indiquant qu'ils appartiennent à ce système. Les acteurs sont placés à l'extérieur du cadre puisqu'ils ne font pas partie du système. Mais, l'acteur est responsable de piloter directement (ou indirectement) l'exécution du cas d'utilisation qui lui est associé.

Par convention et dans la mesure du possible ; Les acteurs principaux sont situés à gauche du cas d'utilisation. Les acteurs secondaires sont situés à droite du cas d'utilisation.

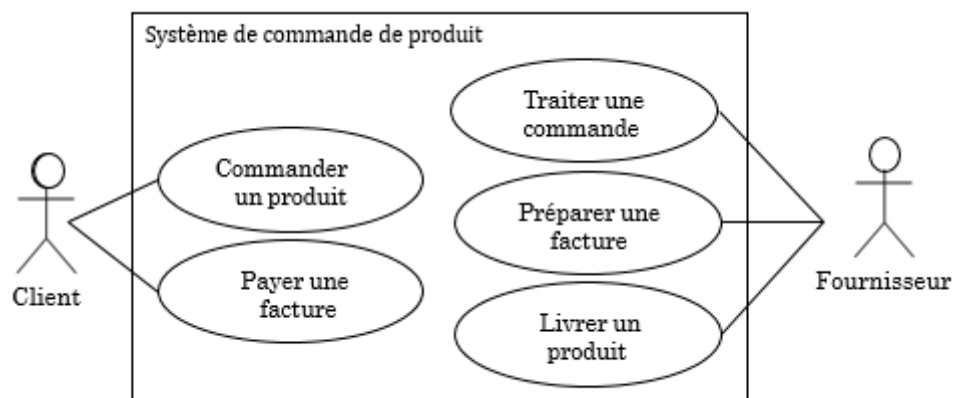
Nom du système

Le nom du système figure à l'intérieur du cadre, en haut

1.4.1. Schéma général d'un diagramme de cas d'utilisation



Exemple



1.5. Relations entre cas d'utilisation

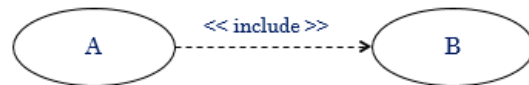
UML définit trois types de relations standardisées entre cas d'utilisation à savoir: relation d'inclusion, relation d'extension et relation de généralisation. Le but de ces relations est de décomposer le système en fonctionnalités plus fines.

1.5.1. Relation d'inclusion

Une relation d'inclusion est une relation dans laquelle un cas d'utilisation A inclut le comportement décrit par le cas d'utilisation B. La

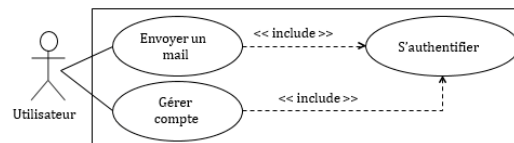
relation d'inclusion prend en charge aussi la réutilisation des fonctionnalités dans un modèle de cas d'utilisation. En d'autres termes, elle permet de décomposer des comportements et de définir des comportements partageables entre plusieurs cas d'utilisation.

Elle est symbolisée avec une dépendance étiquetée par le stéréotype « include » (ou « inclut » en français).



Exemple

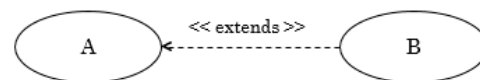
Dans cet exemple, la gestion de compte et l'envoi des email sont des transactions sécurisées et nécessitent par conséquent une authentification.

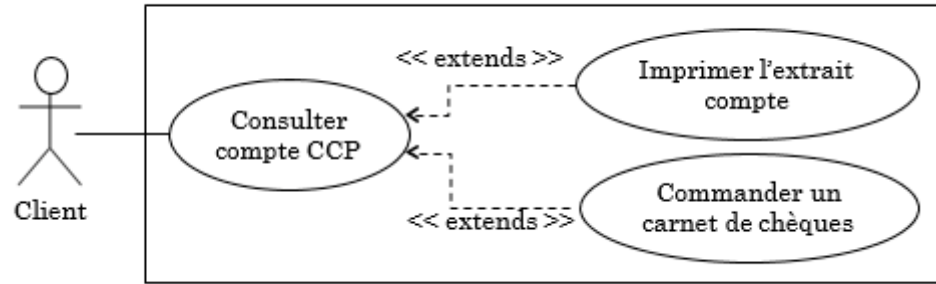


1.5.2. Relation d'extension

Une relation d'extension peut être utilisée pour spécifier qu'un cas d'utilisation d'extension (cas d'utilisation B) étend le comportement d'un autre cas d'utilisation A. Autrement dit, une relation d'extension dénote un comportement optionnel (B est une partie optionnelle de A).

La relation d'extension est symbolisée par une dépendance étiquetée avec le stéréotype « extends » (ou « étend » en français)

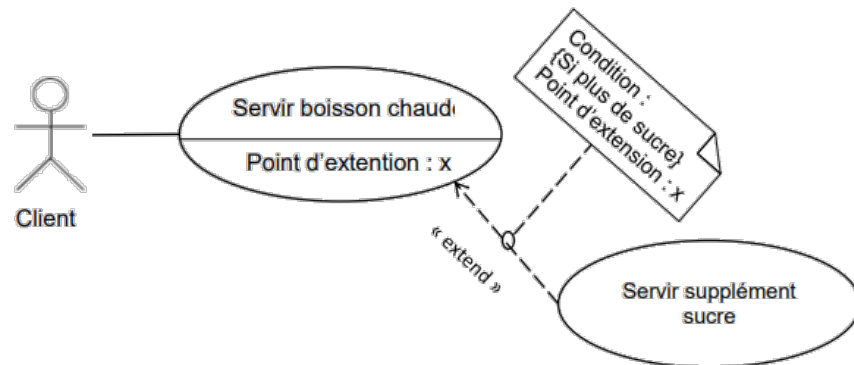




L'extension peut intervenir à un point précis du cas étendu. Ce point s'appelle le point d'extension. Il porte un nom, qui figure dans un compartiment du cas étendu sous la rubrique point d'extension, et est éventuellement associé à une contrainte indiquant le moment où l'extension intervient. L'extension peut être soumise à une condition. Graphiquement, la condition est exprimée sous la forme d'une note.

Une contrainte définit des propositions devant être maintenues à vraies pour garantir la validité du système modélisé. On représente une contrainte sous la forme d'une chaîne de texte placée entre accolades { }.

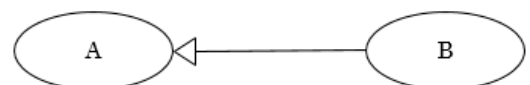
Une note en UML est une annotation qui peut concerner n'importe quel élément UML. Une note contient une information textuelle comme un commentaire, un corps de méthode ou une contrainte. Graphiquement, elle est représentée par un rectangle dont l'angle supérieur droit est plié. Une note n'indique pas explicitement le type d'élément qu'elle contient, toute l'intelligibilité d'une note doit être contenue dans le texte même. On peut relier une note à l'élément qu'elle décrit grâce à une ligne en pointillés



1.5.3. Relation de généralisation

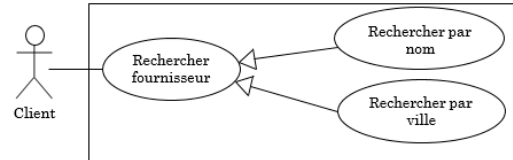
Une relation de généralisation de cas d'utilisation peut être définie conformément au principe de la spécialisation - généralisation déjà présentée pour les classes. Le cas d'utilisation A est une généralisation de B, si B est un cas particulier de A

La relation de généralisation entre cas d'utilisation est représentée symboliquement par une ligne continue avec une flèche triangulaire creuse indiquant le cas d'utilisation le plus général



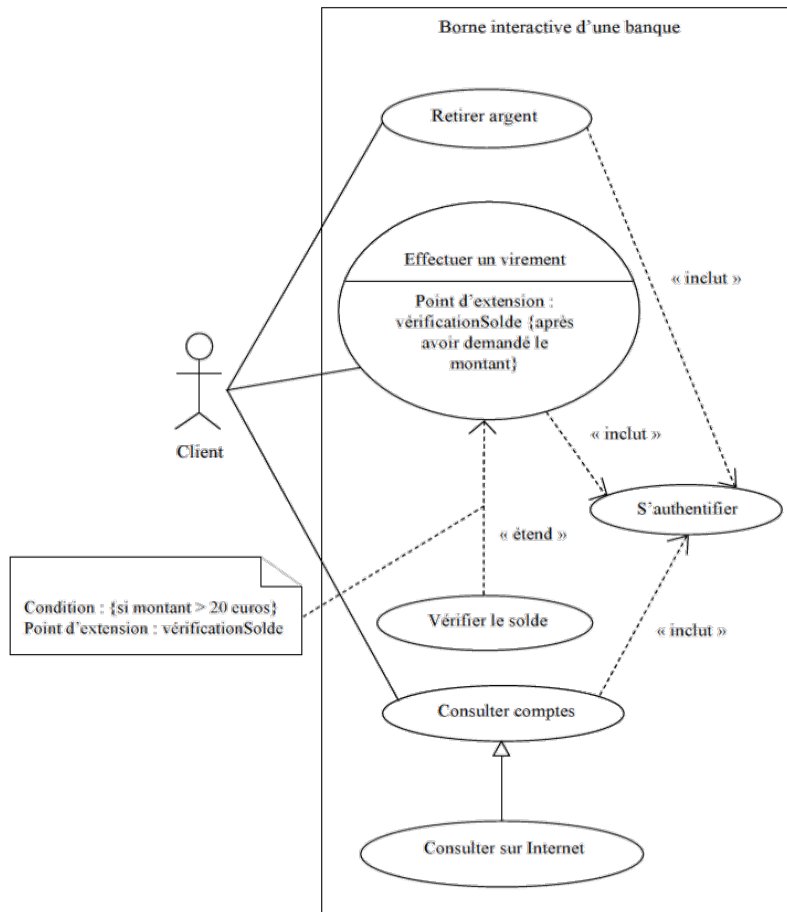
Exemple

Dans cet exemple, on représente le fait qu'un client peut rechercher un fournisseur soit via son nom soit via sa ville.



1.5.4. Exemple d'un diagramme de cas d'utilisation

Exemple



Remarque

Quand un cas n'est pas directement relié à un acteur, il est qualifié de cas d'utilisation interne.

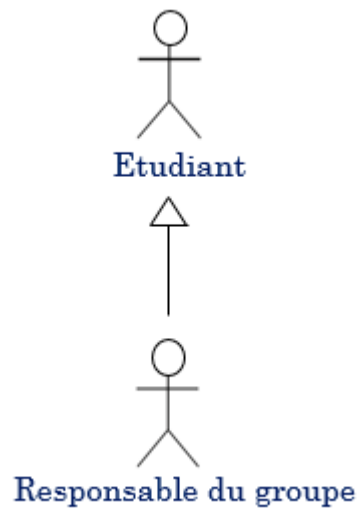
1.6. Relation entre acteurs

La seule relation possible entre deux acteurs est la généralisation. un acteur A est une généralisation d'un acteur B si l'acteur A peut être substitué par l'acteur B. Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai.

La relation de généralisation entre acteurs est symbolisée par un trait plein avec une flèche triangulaire creuse indiquant l'acteur le plus général.

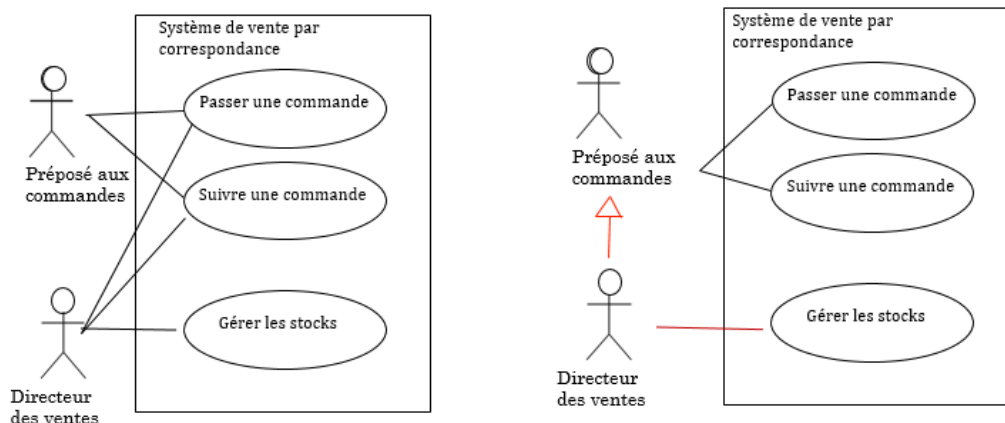


👉 Exemple



👉 Exemple

Le directeur des ventes est un préposé aux commandes avec un pouvoir supplémentaire : en plus de pouvoir passer et suivre une commande, il peut gérer le stock. Par contre, le préposé aux commandes ne peut pas gérer le stock.



1.7. Description textuelle des cas d'utilisation

Un diagramme de cas d'utilisation peut être suivi par une description textuelle et ce dans le but de faire figurer des renseignements supplémentaires. Cette description repose sur la notion de scénario. Un scénario représente une succession particulière d'enchaînements s'exécutant du début à la fin du cas d'utilisation. UML ne propose pas de présentation type de cette description textuelle. Cependant, les travaux menés par Alistair Cockburn sur ce sujet constituent une référence en la matière. Une description textuelle couramment utilisée se compose de trois parties.

- La première partie permet d'identifier le cas
- La deuxième partie contient la description du fonctionnement du cas sous la forme d'une séquence de messages échangés entre les acteurs et le système.
- La troisième partie de la description d'un cas d'utilisation est une rubrique optionnelle.

Sommaire d'identification	
Titre	Nom du cas d'utilisation
Résumé	But du cas d'utilisation
Acteurs	Acteurs participants au cas d'utilisation
Description des scénarios	
Préconditions	Conditions décrivant dans quel état doit être le système avant que le cas d'utilisation puisse être déclenché
Scénario nominal	Séquences d'actions associées au cas d'utilisation se déroulant lorsqu'il n'y a pas d'erreurs
Enchaînement d'erreurs	Séquences d'action conduisant à un <u>échec</u> . En d'autres termes, un scénario d'exception intervient quand une erreur se produit, le séquençement nominal s'interrompt, sans retour à l'enchaînement nominal.
Postconditions	Conditions décrivant l'état du système à l'issue des différents scénarios
Rubrique optionnelle (spécifications non fonctionnelles)	

Présentation type de description textuelle des cas d'utilisation.

Sommaire d'identification	
Titre	Consulter le compte depuis internet
Résumé	Détailler les étapes permettant à un client à consulter son compte bancaire
Acteurs	Client, Banque (secondaire)
Description des scénarios	
Préconditions	Le client existe et son compte a été créé
Scénario nominal	2.1 Le client s'authentifie 2.2 Le système vérifie le login et le mot de passe ont été saisis 2.3 Le système vérifie que le login et le mot de passe sont valides 2.4 Le système affiche la page web du compte pour le client
Enchainement d'erreurs	a En (2.2) : si le login ou le <u>mdp</u> sont incorrectement saisis, le client est averti de l'erreur, et invité à recommencer b En (2.3) : si les informations sont erronées, elles sont <u>re-demandées</u> au client
Postconditions	Le système stocke dans l'historique la date d'accès au compte du client
Rubrique optionnelle	
Contraintes non fonctionnelles : Confidentialité: les informations concernant le client ne doivent pas être divulgués Contraintes liées à l'interface homme-machine : Toujours demander la validation de la part du client	

1.8. Exercice

Un nouveau Système de Gestion de Stock (SGS) doit être mis en place. Celui-ci sera utilisé par le service stock et doit couvrir les activités de gestion des stocks et de réapprovisionnement. Dans le service stock, se côtoient les chefs d'équipes et les magasiniers. Les magasiniers sont responsables par la réception des marchandises et leur expédition aux autres magasins de la société. Les chefs d'équipe gèrent le réapprovisionnement : ils soumettent aux fournisseurs les demandes de devis pour les marchandises et élaborent les commandes. Par contre, tous les employés, qu'ils soient chef d'équipe ou magasinier, sont responsables de la gestion du catalogue (inclusion des nouveaux produits, mise à jour des produits, etc.). Toutes ces tâches seront gérées par le système de gestion SGS. Pendant le réapprovisionnement les magasins transmettent au service de stock leurs besoins. Le chef d'équipe vérifie si les marchandises demandées sont disponibles. Si c'est le cas, un ordre d'expédition est transmis aux magasiniers, qui vont expédier les produits au magasin qui les a demandés.

Travail demandé: Modéliser ce système avec un diagramme de cas d'utilisation.