

UML Diagramme de Use Case

Objectif

L'objectif d'un cas d'utilisation est de décrire une interaction entre un utilisateur (ou un acteur) et un système. Cela implique de comprendre les besoins de l'utilisateur et de décrire comment le système répondra à ces besoins.

Acteur

Un acteur est un utilisateur externe au système. Cela peut être :

- Une personne.
- Du matériel (capteurs, moteurs, relais...).
- Un autre système.

Notion de Use Case

En langage de modélisation unifié (UML), un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. La création de ce type de diagramme UML requiert un ensemble de symboles et de connecteurs spécifiques.

Les Types de Cas d'Utilisation

1. Il y a trois types principaux de cas d'utilisation:
 - Les cas d'utilisation principaux: Ils décrivent les interactions principales entre l'utilisateur et le système. Ils sont souvent appelés "scénarios de base".
 - Les cas d'utilisation secondaires: Ils décrivent des interactions moins fréquentes et moins importantes entre l'utilisateur et le système. Ils sont souvent appelés

"scénarios alternatifs".

- Les cas d'utilisation exceptionnels: Ils décrivent des interactions exceptionnelles ou des erreurs qui peuvent se produire lors de l'utilisation du système. Ils sont souvent appelés "scénarios d'exception".

Les liens (ou relations) entre cas d'utilisation

Il y a plusieurs types de relations qui peuvent exister entre les cas d'utilisation, notamment:

La relation d'inclusion

La relation d'inclusion sert à enrichir un cas d'utilisation par un autre cas d'utilisation (c'est une sous fonction).

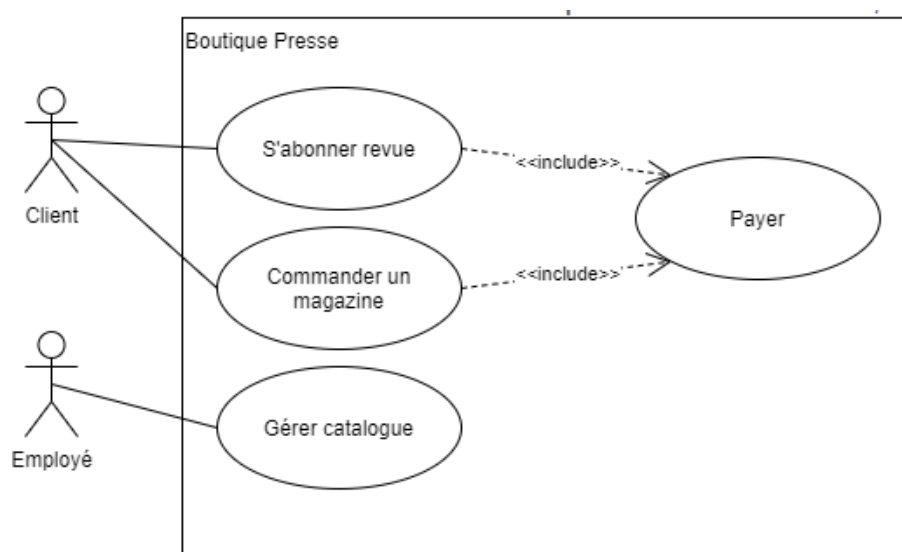
La relation d'inclusion est impérative et donc systématique.

Dans un diagramme des cas d'utilisation, cette relation est représentée par une flèche pointillée

reliant les 2 cas d'utilisation et munie du stéréotype « **include** ».

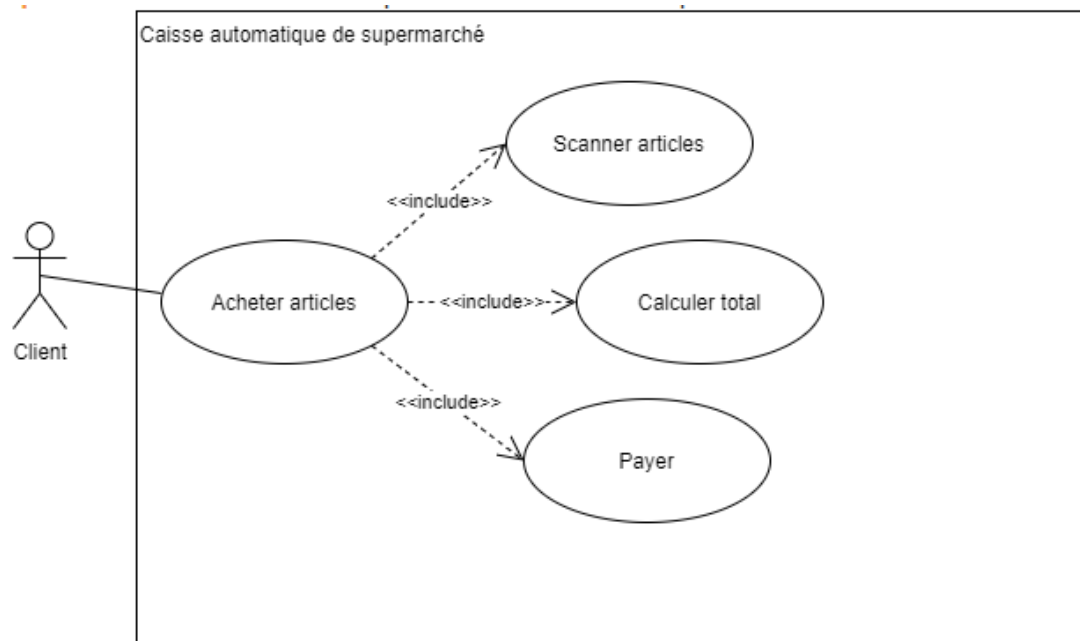
L'inclusion permet de :

- Partager une fonctionnalité commune entre plusieurs cas d'utilisation.



Exemple: « include » pour découper un UC complexe

- Décomposer un cas d'utilisation complexe en décrivant ses sous fonctions.

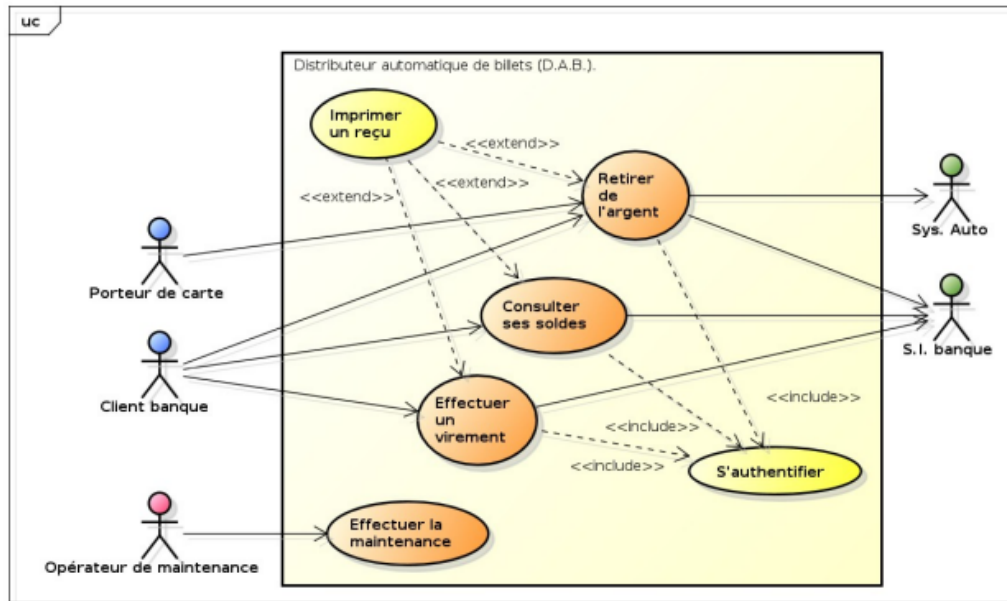


Exemple: «include» pour isoler opérations communes entre plusieurs UC

La relation d'extension

Comme la relation d'inclusion, la relation d'extension enrichit un cas d'utilisation par un autre cas d'utilisation de **sous fonction mais celui-ci est optionnel**.

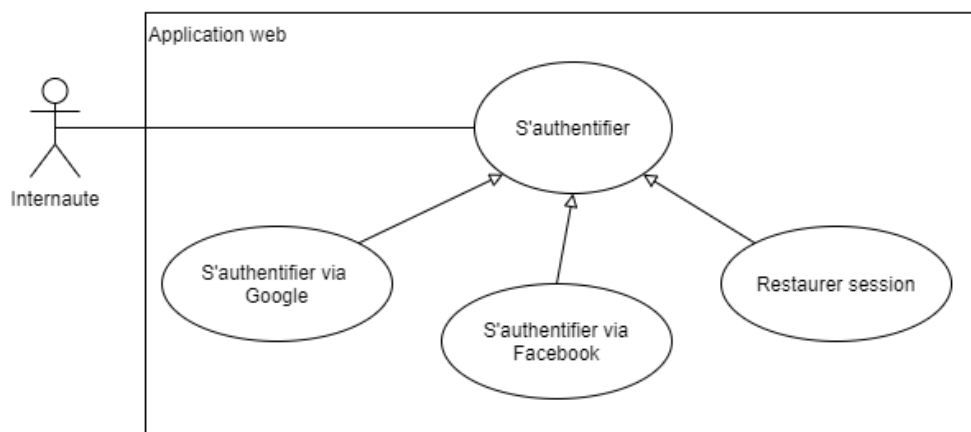
Cette relation est représentée par une flèche en pointillée reliant les 2 cas d'utilisation et munie du stéréotype « **extend** ».



Exemple: <<extend>> pour les opérations facultatives.

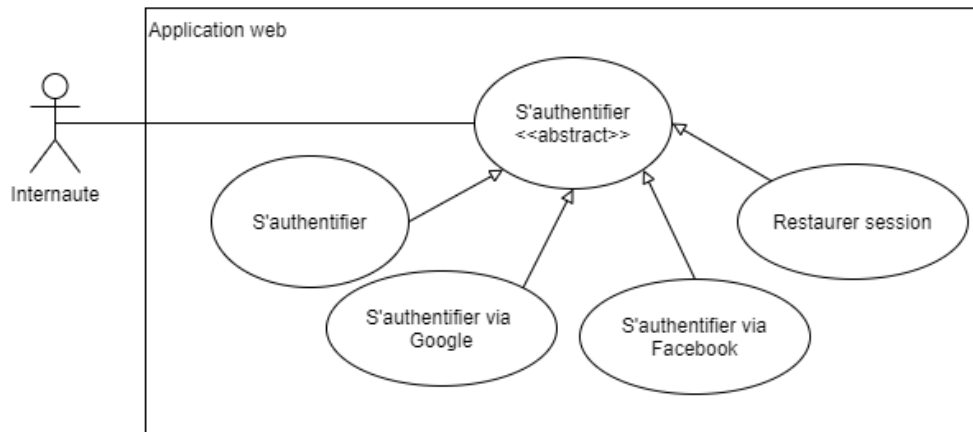
La Relation de généralisation ou de spécialisation

La relation de **spécialisation** (ou de **généralisation** en fonction des ouvrages) spécifie une variante d'opérations (UC "enfant") qui peut être réalisée à la place d'un UC (UC "parent").



Exemple: authentification locale par défaut

Le UC "parent" peut être déclaré **abstrait**; dans ce cas, le UC ne peut être réalisé en lui-même; seules les variantes spécialisées peuvent être réalisées.



Exemple: pas de technique d'authentification par défaut

Résumé des différentes relations entre UC

Inclusion <<include>>

Le UC de base ne peut être réalisé seul;
la réalisation de chaque UC "inclus" est obligatoire.

Extension <<extend>>

Le UC de base peut être réalisé seul;
la réalisation d'une UC d'extension est facultative.

Specialisation

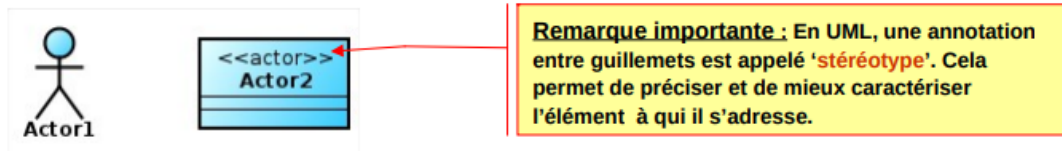
Le UC "parent" peut être **abstrait** ou non;
si le UC "parent" est abstrait, une UC spécialisée au moins est obligatoire.

Formalisme d'un Diagramme de UC

Un diagramme de cas d'utilisation est composé de trois éléments principaux:

Les acteurs

Les acteurs se représentent sous la forme d'un petit personnage (stick man) ou sous la forme d'une case rectangulaire (appelé classeur) avec le mot clé « actor ».
Chaque acteur porte un nom.



Quelquefois, nous utilisons :

- le stick man si l'acteur est humain
- le classeur si l'acteur est du matériel ou un autre système.

Les cas d'utilisation

Un cas d'utilisation se représente par une ellipse contenant le nom du cas d'utilisation (phrase commençant par un verbe à l'infinitif) et optionnellement un stéréotype au dessus du nom.



Les relations

Les relations sont représentées par des traits qui relient les acteurs aux cas d'utilisation et par des flèches qui relient les cas d'utilisation les uns aux autres.

Description textuelle des cas d'utilisation :

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle, car c'est une forme souple qui convient dans bien des situations.

Une description textuelle couramment utilisée se compose de trois parties.

1. La première partie permet d'identifier le cas, elle doit contenir les informations qui suivent.

Nom :

- utiliser une tournure à l'infinitif (ex. : Réceptionner un colis).

Objectif :

◦ une description résumée permettant de comprendre l'intention principale du cas d'utilisation. Cette partie est souvent renseignée au début du projet dans la phase de découverte des cas d'utilisation.

Acteurs principaux :

◦ ceux qui vont réaliser le cas d'utilisation (la relation avec le cas d'utilisation est illustrée par le trait liant le cas d'utilisation et l'acteur dans un diagramme de cas d'utilisation).

Acteurs secondaires :

◦ ceux qui ne font que recevoir des informations à l'issue de la réalisation du cas d'utilisation.

Dates :

- les dates de création et de mise à jour de la description courante.

Responsable :

- le nom des responsables.

Version :

- le numéro de version.

2. 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. Elle contient toujours une séquence nominale qui décrit le déroulement normal du cas. À la séquence nominale s'ajoutent fréquemment des séquences alternatives (des embranchements dans la séquence nominale) et des séquences d'exceptions (qui interviennent quand une erreur se produit).

Les préconditions :

◦ elles décrivent dans quel état doit être le système (l'application) avant que ce cas d'utilisation puisse être déclenché.

Des scénarios :

◦ ces scénarios sont décrits sous la forme d'échanges d'événements entre l'acteur et le système. On distingue le scénario nominal, qui se déroule quand il n'y a pas d'erreur, des scénarios alternatifs qui sont les variantes du scénario nominal et enfin les scénarios d'exception qui décrivent les cas d'erreurs.

Des postconditions :

- elles décrivent l'état du système à l'issue des différents scénarios.

3. La troisième partie de la description d'un cas d'utilisation est une rubrique optionnelle. Elle contient généralement des spécifications non fonctionnelles (spécifications techniques...). Elle peut éventuellement contenir une description des besoins en termes d'interface graphique.

Objectifs

Explique ce que le cas d'utilisation est censé accomplir.

Scénario Nominal

Décrit les étapes principales du scénario de base.

Scénario Alternatif

Décrit les étapes alternatives qui peuvent être prises si certaines conditions sont remplies.

Scénario Exception

Décrit les erreurs ou les problèmes qui peuvent survenir lors de l'exécution du scénario.

Structure d'une fiche descriptive d'UC

Décrit la structure et les éléments de la fiche descriptive d'un cas d'utilisation.