

# Diagramme de Use Case

- [Wikipedia - Unified Modelling Language](#)
- [Wikipedia - Use case diagram](#)

Le but d'un diagramme Use Case est de décrire visuellement les besoins d'un organisme (interaction des acteurs avec le système et ses fonctionnalités). Il vient de l'UML (Unified Modelling Language)

On va ainsi définir quelles sont les fonctionnalités (dans le cadre de l'analyse, donc seulement les principales) et qui peut y accéder, pour y faire quoi et comment.

Voici les différents éléments du diagramme :

- Le système est représenté par un rectangle délimitant le périmètre.
- Les acteurs externes (éléments externes qui interagissent avec celui-ci), sont représentés par leur rôle sous forme de bonhomme bâton (stickman). Les acteurs non humains sont définis par un rectangle libellé
  - Les acteurs primaires qui initient les CUs (cas d'utilisation) sont placés à gauche du système
  - Les acteurs secondaires qui aident à la réalisation des CUs mais qui n'en sont pas les bénéficiaires à droite.
  - Une flèche peut être tracée d'un acteur à un autre si le premier acteur peut faire tout ce que le deuxième peut. Exemple: Administrateur → éditeur
- Les cas d'utilisations (CU) sont représentés par des ovales libellés. Le nom contient un verbe d'action à l'infinitif, choisi du point de vue de l'utilisateur, réalise un service du début à la fin. Exemple: « Lire un message ». Ils sont aussi reliés avec les acteurs avec une ligne

Ce diagramme permet d'avoir une vision globale des fonctionnalités du système, compréhensible par tout le monde, même non initié.

Ensuite il y a la « *Spécification textuelle* » qui décrit avec plus de détails les cas d'utilisations. Le texte doit fournir, pour chaque fonctionnalité:

- Le même nom que dans le diagramme (ex: Lire un message)
- L'identification de l'évènement déclencheur (ex: L'utilisateur veut lire ses nouveaux messages ou relire d'anciens messages)
- La description de la fonctionnalité. La description doit être suffisamment générale pour ne pas être dépendante sur l'interface.
- L'identification du/des acteurs impliqué(s) (ex: l'utilisateur).
- Les éventuelles pré/post conditions (ex: L'utilisateur doit être authentifié et une fois lus les messages ne doivent plus apparaître en gras). Les post conditions sont toutes les choses

qui doivent avoir été fait pour que l'UC soit considéré comme terminé.

## En pratique

Pour faire le diagramme :

1. Trouver quel est le système de la situation. Il s'agit bien du système informatique. Donc il faut éviter de représenter un élément comme "Bibliothèque" ou "nom de la société" et être plus précis comme "Système de gestion biliothécaire" ou "Site marchand BeGood". Créer un rectangle libellé représentant le système.
2. Trouver les acteurs primaires et secondaires. Les acteurs sont ceux qui interagissent *directement* avec le système. Les acteurs primaires (qui initient les UC) sont à droite du système, les acteurs secondaires (qui aident à la concrétisation du UC mais qui n'en bénéficie pas) sont mis à gauche. Les acteurs humains sont représentés par des stickman tandis que les non humains peuvent être représenté par un petit rectangle libellé. Le nom doit être un *rôle*, et pas un nom ou personne en particulier.
3. Pour chaque acteur noter les actions possible en excluant toutes celles qui sont en dehors du système. Essayer de le synthétiser un maximum, mais tout en incluant toutes les actions dans le système. Les UC doivent commencer par un *verbe à l'infinitif*. (Relire les consignes en se mettant dans la peau de chaque acteur pour voir ce qu'il pourrait faire). Puis ensuite relire les UC pour vérifier leur portée (mot "Gérer" par exemple)
4. Lier les acteurs avec les UC via une ligne (sans flèche), et lier les acteurs entre eux quand un rôle peut faire tout ce qu'un autre rôle peut faire (administrateur --> utilisateur par exemple)

Pour faire la spécification textuelle (à faire par UC) :

1. Définir quel est l'élément déclencheur, cela peut être un souhait ou un désir d'un acteur, ou encore lié au résultat d'un autre UC.
2. La description doit être complète mais pas trop longue et ne doit pas dépendre de l'interface utilisateur (qui peut changer dans le temps).
3. La liste des acteurs impliqués.
4. Les préconditions (tel que l'authentification par exemple), pour que le UC prenne place.
5. Les postconditions, qui définissent l'état du système après le UC. C'est à dire ce qui doit avoir changé pour que le UC soit défini comme terminé.

💡 Note: Le mot "gérer" fait référence à CRUD (créer, lire, mettre à jour et supprimer) en analyse