

Expression des besoins

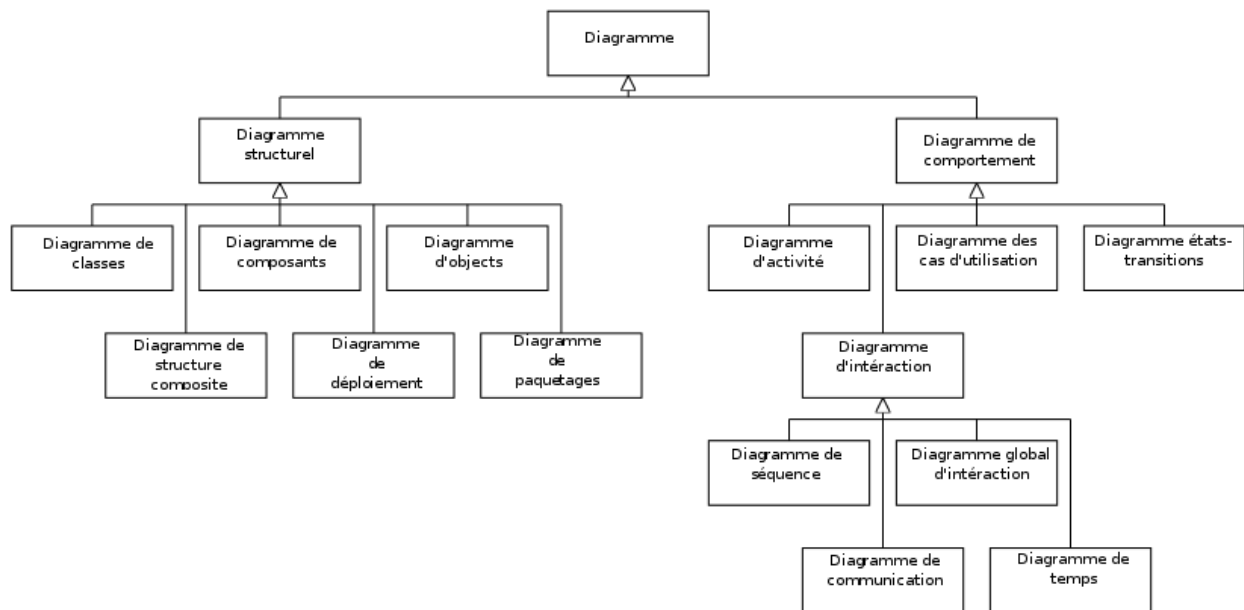
Représentation UML

Introduction

UML est un **langage de modélisation** destiné en particulier à la modélisation objet. UML est devenu une norme OMG en 1997 (OMG est un organisme à but non lucratif créé en 1989 à l'initiative de sociétés comme HP, Sun, Philips, etc.).

UML propose un formalisme indépendant d'un langage de programmation donné. Pour ce faire, UML **normalise** les concepts de l'objet (énumération et définition exhaustive des concepts) ainsi que leur notation graphique. Il peut donc être utilisé comme un moyen de communication entre les étapes de spécification conceptuelle et les étapes de spécifications techniques.

UML est très utile en phase d'analyse, et permet de couvrir un grand éventail de vues (diagrammes):



Par JackPotte — Travail personnel, Domaine public, <https://commons.wikimedia.org/w/index.php?curid=15477550>

1 Représentation graphique des cas d'utilisation avec UML

UML propose de représenter les **cas d'utilisation d'un système** sous une forme graphique nommée *diagramme de cas d'utilisation* (*use-case diagram*) appartenant au modèle des besoins (initialement formalisé par Ivar Jacobson Hjalmar, informaticien Suédois). 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.

Courtes définitions 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).

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.

1.2. 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

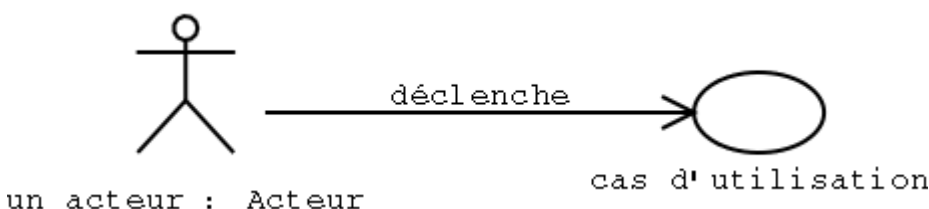
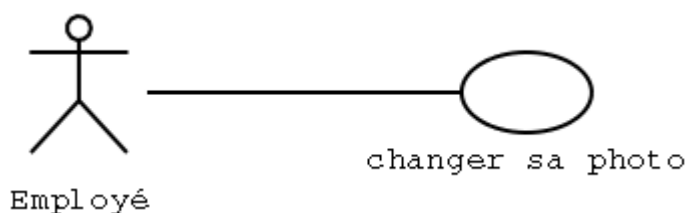


Figure 3. Exemple d'un cas d'utilisation : Changer sa photo

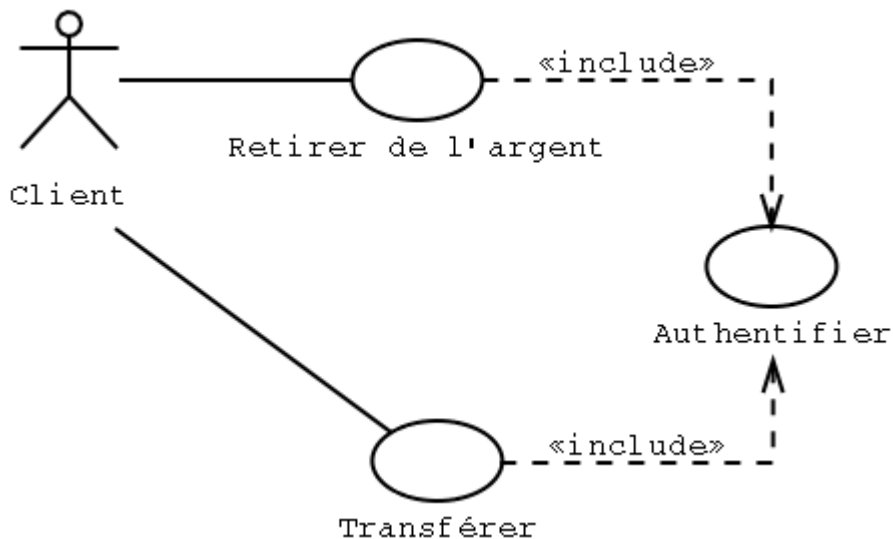


1.3. Relations entre cas d'utilisation

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.

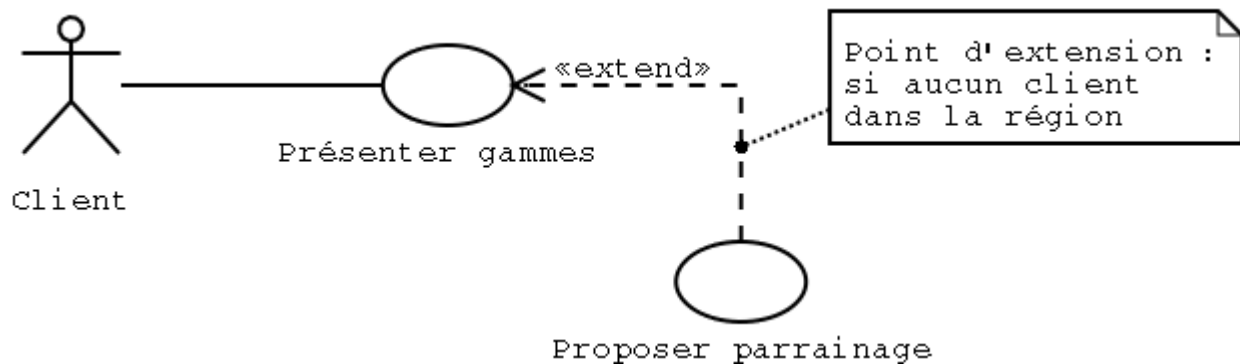
<<include>> : Stéréotype représentant le fait qu'un cas d'utilisation inclut un autre cas d'utilisation. On utilise ce stéréotype lorsque que l'on souhaite factoriser un cas d'utilisation partagé par plusieurs autres cas d'utilisation. *Par exemple, une opération de retrait et une opération de transfert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

Figure 4. Exemple de relation "include"



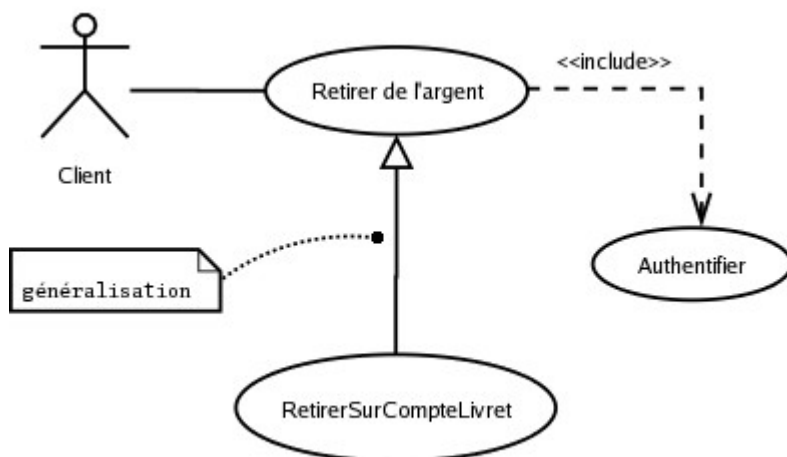
<<extend>> : Un cas d'utilisation peut déclarer des points d'extension (*extension point*). Un point d'extension localise un endroit (un point) unique dans le cas d'utilisation. C'est dans les limites de ce point que d'autres cas d'utilisation pourront étendre (*extend*) le comportement initial du cas 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.*

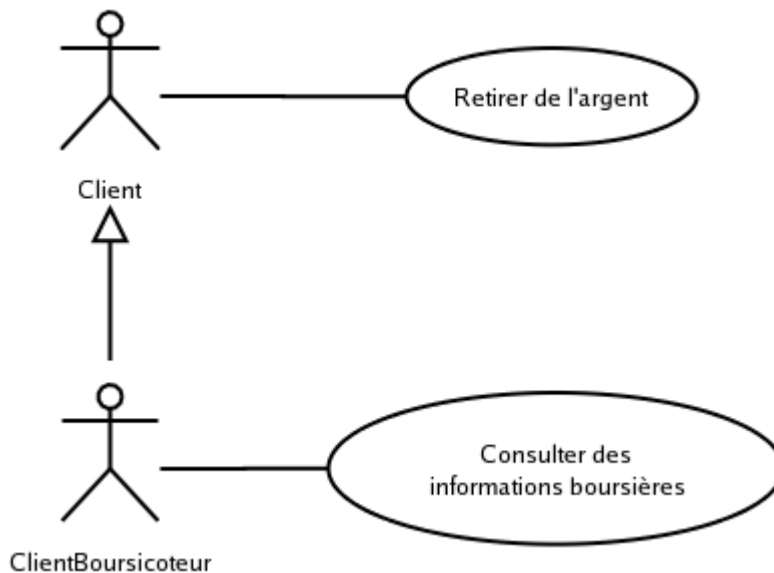
Figure 6. Exemple de relation de généralisation



1.4. 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.

1.5. Classification des acteurs / cas d'utilisation

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. Notons que certains cas d'utilisation sont qualifiés d'« **interne** » s'il ne sont pas reliés directement à un acteur. Les cas d'utilisation de type « **frontal** » sont ceux pour lesquels les acteurs sollicitent le système : ils doivent donc apporter une **plus-value métier à ses utilisateurs**.

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 :

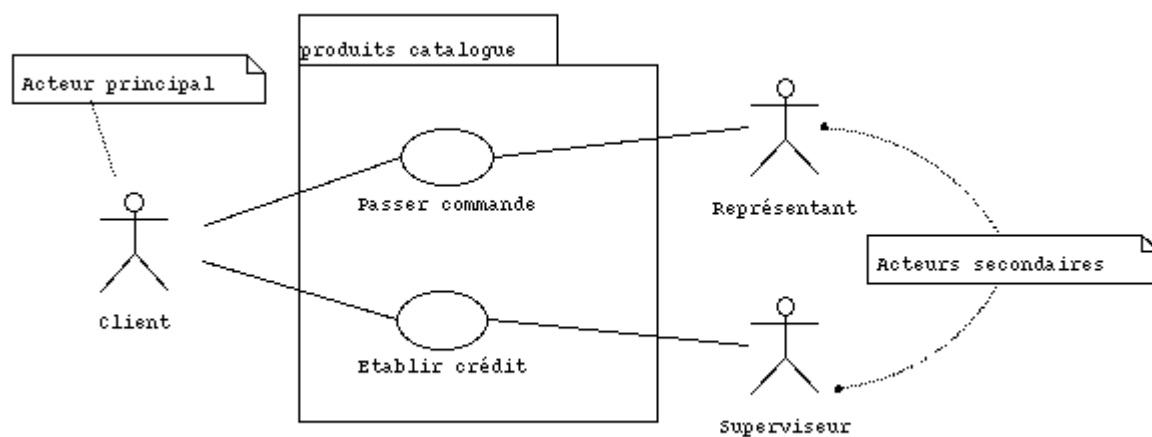
Un acteur peut être humain ou purement informatique, *hardware* ou *software*.

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 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).

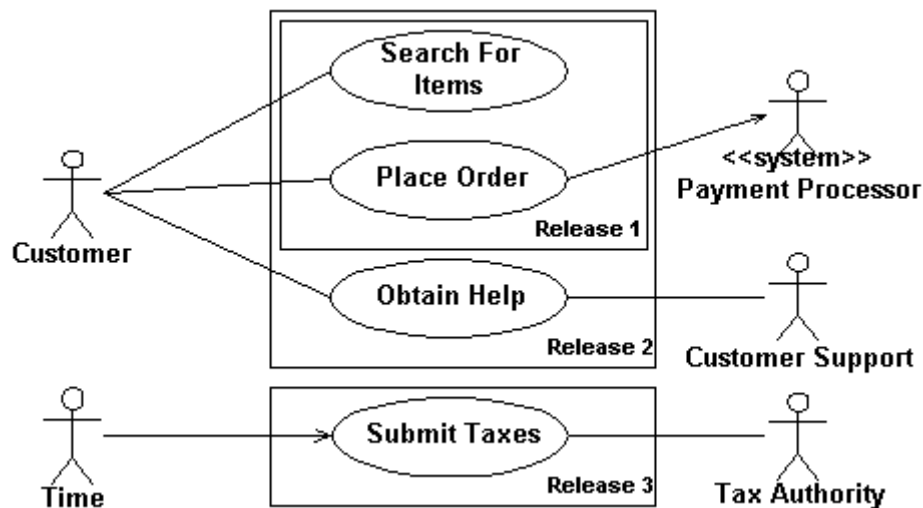
Par convention, le lecteur s'attend à trouver, dans un diagramme de cas d'utilisation, les acteurs principaux à gauche du contexte des ellipses et les acteurs secondaires à droite, comme l'illustre l'exemple ci-dessous.

Figure 8. Exemple d'acteurs



Exercice d'interprétation d'un diagramme de cas d'utilisation

(<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>)



Questions
1. Identifier

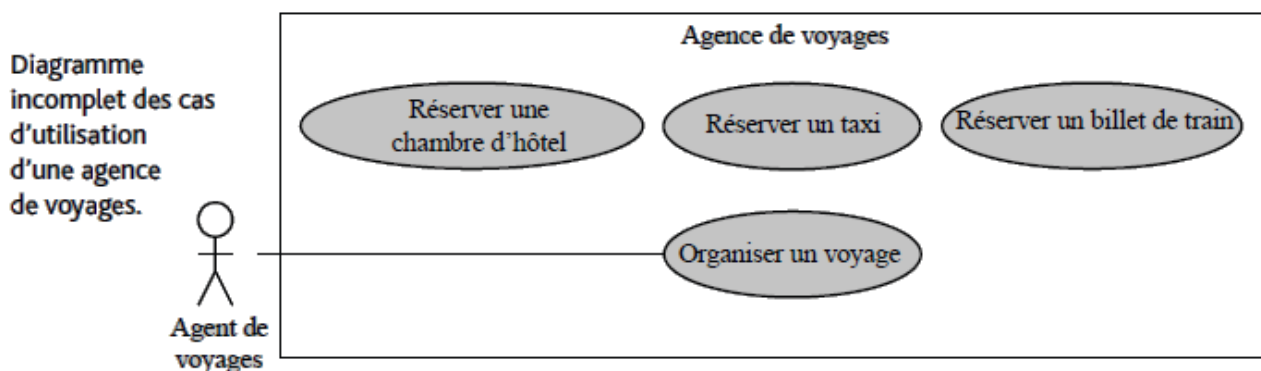
- les acteurs principaux et les secondaires
2. A quoi correspondent les cadres ?
3. Que signifie les annotations Release 1, Release 2 et Release 3 ?

Exercice de conception d'un diagramme de cas d'utilisation

(© 2010 Pearson Education France – UML2, 3e édition)

Choisissez et dessinez les relations entre les cas suivants :

1. Une agence de voyages organise des voyages où l'hébergement se fait en hôtel. Le client doit disposer d'un taxi quand il arrive à la gare pour se rendre à l'hôtel.



2. Certains clients demandent à l'agent de voyages d'établir une facture détaillée. Cela donne lieu à un nouveau cas d'utilisation appelé « Établir une facture détaillée ». Comment mettre ce cas en relation avec les cas existants ?
3. Le voyage se fait soit par avion, soit par train. Comment modéliser la réservation ?

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](#)).


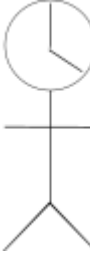
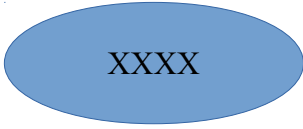
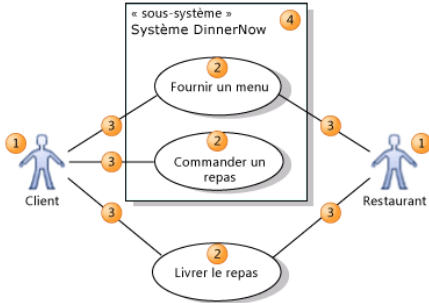
En conception et en implémentation, les cas d'utilisation font office de garde-fou auprès des développeurs, leur permettant de garder en **ligne de mire** les objectifs utilisateur.

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/>)
- Meyer, B. (1997). *UML: the positive spin* , <http://archive.eiffel.com/doc/manuals/technology/bmarticles/uml/page.html>
- 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>
- 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>
- OMG (2003). *Spécification UML* , <http://www.omg.org/technology/documents/formal/uml.htm>
- Cas d'utilisation (wikipédia) https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation
- Cas d'utilisation (sur openclassrooms) <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/etape-3-les-cas-d-utilisation>

Tableau à compléter

Symbole	Définition	Exemple
 <div data-bbox="276 465 466 510" style="border: 1px solid black; padding: 2px; display: inline-block;">actor</div>		
		
		
		

Use Case – Cas d'utilisation