

# Chapitre 3. Diagramme UML de cas d'utilisation : vue fonctionnelle

## 1.Introduction

Bien souvent, la maîtrise d'ouvrage et les utilisateurs ne sont pas des informaticiens. Il leur faut un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation qui permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

## 2.Éléments des diagrammes de cas d'utilisation

### 1. Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom (i.e. son rôle) inscrit dessous.



Exemple de représentation d'un acteur

Il est également possible de représenter un acteur sous la forme d'un classeur stéréotypé :

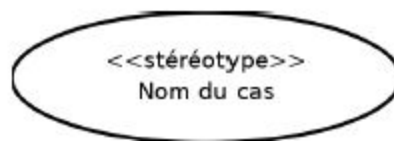


Exemple de représentation d'un acteur sous la forme d'un classeur

## 2. Cas d'utilisation

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

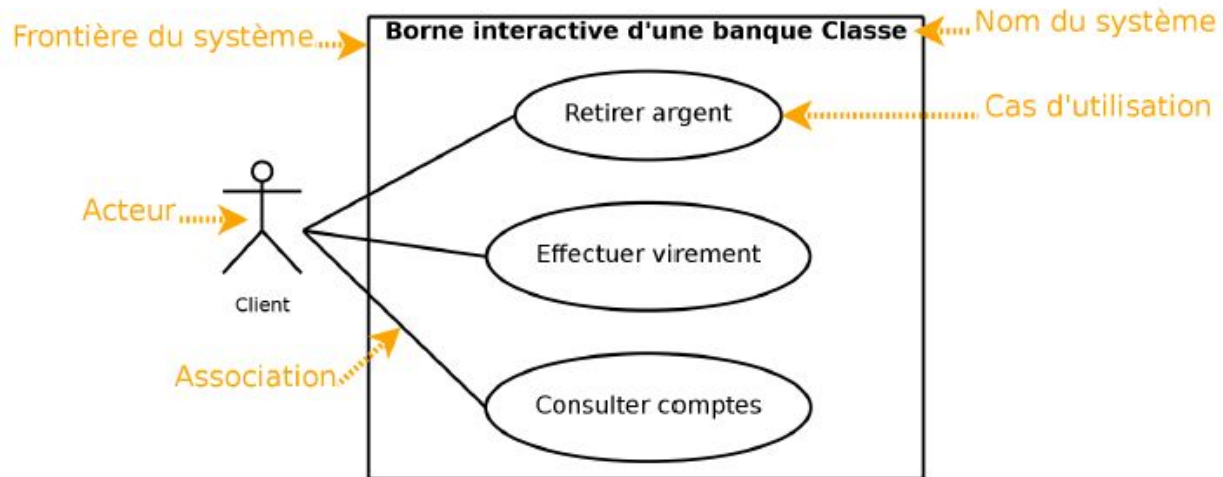
Un cas d'utilisation se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.



Exemple de représentation d'un cas d'utilisation

## 3. Représentation d'un diagramme de cas d'utilisation

Comme le montre la figure en dessous, la frontière du système est représentée par un cadre. Le nom du système figure à l'intérieur du cadre, en haut. Les acteurs sont à l'extérieur et les cas d'utilisation à l'intérieur.



Exemple simplifié de diagramme de cas d'utilisation modélisant une borne d'accès à une banque.

#### 4. Relations dans les diagrammes de cas d'utilisation

##### 1. Relations entre acteurs et cas d'utilisation

###### - Relation d'association

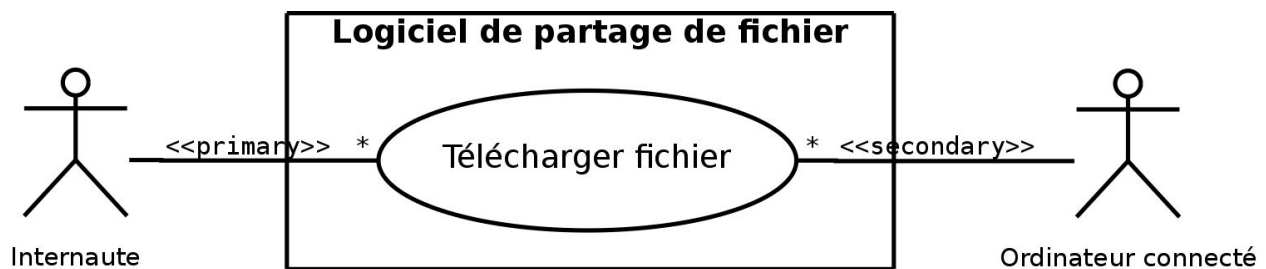


Diagramme de cas d'utilisation représentant un logiciel de partage de fichiers

Une relation d'association est chemin de communication entre un acteur et un cas d'utilisation et est représenté un trait continu.

###### - Cas d'utilisation interne

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

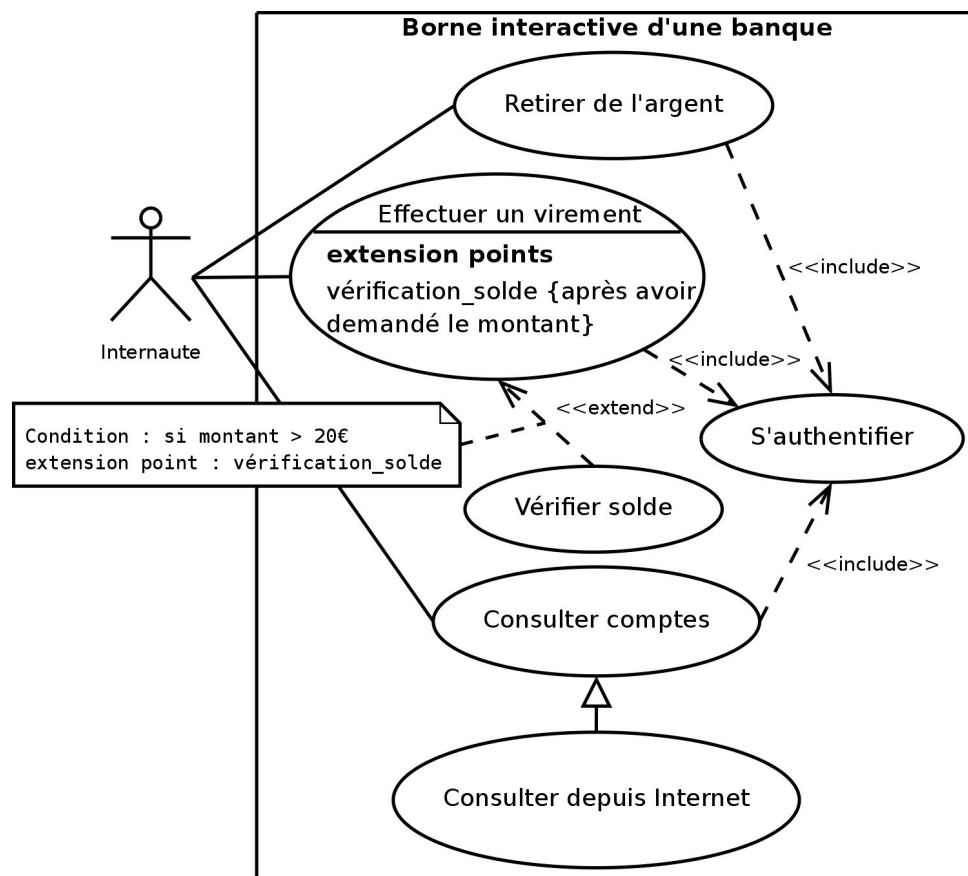
##### 2. Relations entre cas d'utilisation

Il existe principalement deux types de relations :

- les dépendances stéréotypées, qui sont explicitées par un stéréotype (les plus utilisés sont l'inclusion et l'extension),
- et la généralisation/spécialisation.

Une dépendance entre deux cas d'utilisation se représente par une flèche avec un trait pointillé. Si le cas A inclut ou étend le cas B, la flèche est dirigée de A vers B.

Le symbole utilisé pour la généralisation est un flèche avec un trait plein dont la pointe est un triangle fermé désignant le cas le plus général. L'exemple suivante illustre les différents type de relations dans un diagramme de cas d'utilisation :



Exemple de diagramme de cas d'utilisation

## - Relation d'inclusion

Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend de B. Lorsque A est sollicité, B l'est obligatoirement, comme une partie de A. Cette dépendance est symbolisée par le stéréotype « include ». Par

exemple, l'accès aux informations d'un compte bancaire inclut nécessairement une phase d'authentification avec un identifiant et un mot de passe.

### - Relation d'extension

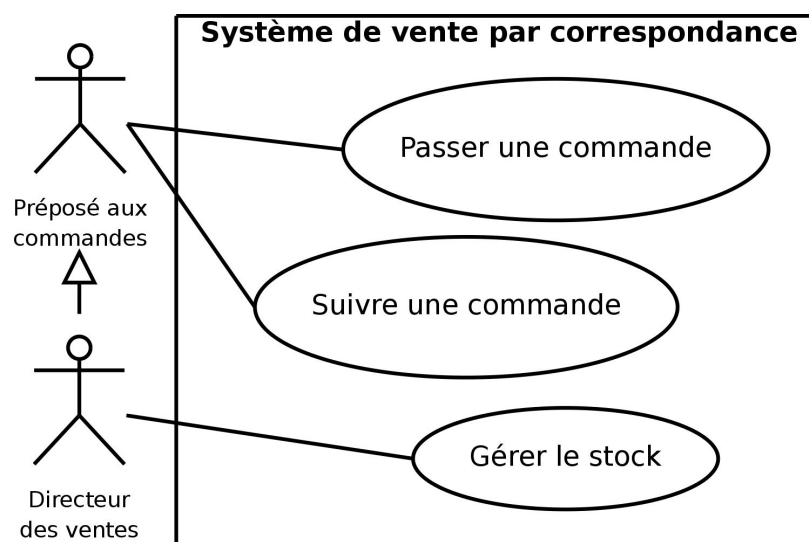
On dit qu'un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Exécuter B peut éventuellement entraîner l'exécution de A : contrairement à l'inclusion, l'extension est optionnelle. Cette dépendance est symbolisée par le stéréotype « extend ».

### - Relation de généralisation

Un cas A est une généralisation d'un cas B si B est un cas particulier de A. Dans la figure 2.7, la consultation d'un compte via Internet est un cas particulier de la consultation. Cette relation de généralisation/spécialisation est présente dans la plupart des diagrammes UML et se traduit par le concept d'héritage dans les langages orientés objet.

## 3. Relations 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.



Relations entre acteurs

## 3. Modélisation des besoins avec UML

### 1. Comment identifier les acteurs ?

Les acteurs d'un système sont les entités externes à ce système qui interagissent (saisie de données, réception d'information, . . .) avec lui. Les acteurs sont donc à l'extérieur du système et dialoguent avec lui.

Chaque acteur doit être nommé. Ce nom doit refléter son rôle car un acteur représente un ensemble cohérent de rôles joués vis-à-vis du système. Pour trouver les acteurs d'un système, il faut identifier quels sont les différents rôles que vont devoir jouer ses utilisateurs (ex : responsable clientèle, responsable d'agence, administrateur, approuvateur, . . .). Il faut également s'intéresser aux autres systèmes avec lesquels le système va devoir communiquer comme :

- les périphériques manipulés par le système (imprimantes, hardware d'un distributeur de billet, . . .) ;
- des logiciels déjà disponibles à intégrer dans le projet ;
- des systèmes informatiques externes au système mais qui interagissent avec lui, etc.

Pour faciliter la recherche des acteurs, on peut imaginer les frontières du système. Tout ce qui est à l'extérieur et qui interagit avec le système est un acteur, tout ce qui est à l'intérieur est une fonctionnalité à réaliser.

Vérifiez que les acteurs communiquent bien directement avec le système par émission ou réception de messages.

### 2. Comment recenser les cas d'utilisation ?

L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un de ses acteurs.

Pour identifier les cas d'utilisation, il faut se placer du point de vue de chaque acteur et déterminer comment et surtout pourquoi il se sert du système. Il faut éviter les redondances et limiter le nombre de cas en se situant à un bon niveau d'abstraction. Trouver le bon niveau de détail pour les cas d'utilisation est un problème difficile qui nécessite de l'expérience.

Nommez les cas d'utilisation avec un verbe à l'infinitif suivi d'un complément. Par exemple, un distributeur de billets aura probablement un cas d'utilisation retirer de l'argent et non pas Distribuer de l'argent.