

## Chap3 Diagramme de cas d'utilisation (vue fonctionnelle)

### 1. Introduction

Les cas d'utilisation ('use cases') servent à exprimer le comportement du système en termes d'actions et de réactions, *selon le point de vue de chaque utilisateur*. Les cas d'utilisation délimitent le système, ses fonctions (ses cas), et ses relations avec son environnement. Ils constituent un moyen de *déterminer les besoins* du système. Pour élaborer les cas d'utilisation, il faut se fonder sur des entretiens avec les utilisateurs.

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

Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un de ses acteurs. Aussi, 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. Nommez les cas d'utilisation avec un verbe à l'infinitif suivi d'un complément en vous plaçant du point de vue de l'acteur et non pas de celui du système. Par exemple, un distributeur de billets aura probablement un cas d'utilisation Retirer de l'argent et non pas Distribuer de l'argent.

**Dans tous les cas, il faut bien garder à l'esprit qu'il n'y a pas de notion temporelle dans un diagramme de cas d'utilisation.**

### 3. 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. 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 un verbe à l'infinitif suivi d'un complément.

Objectif :

- une description résumée du cas d'utilisation.
- Acteurs principaux : ceux qui vont réaliser le 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 contient toujours une séquence nominale qui décrit de 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 pré conditions :

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

Des scénario :

- 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 scénario alternatifs qui sont les variantes du scénario nominal et enfin les scénarios d'exception qui décrivent les cas d'erreurs.

Des post conditions :

- 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.

#### **4. Démarche d'utilisation des cas d'utilisation :**

- 1) Identifier les acteurs et les cas.
- 2) Décrire les cas en écrivant des scénarios sous forme textuelle.
- 3) Dessiner les diagrammes de cas.
- 4) Récapituler les cas, les structurer et s'assurer de la cohérence d'ensemble

#### **5. Éléments des diagrammes de cas d'utilisation**

##### **5.1. Acteur**

Un acteur est une *personne* ou un *système* qui interagit avec le système étudié, en échangeant de l'information (en entrée et en sortie). On trouve les acteurs en observant les utilisateurs directs du système, les responsables de sa maintenance, ainsi que les autres systèmes qui interagissent avec lui. Un acteur représente un *rôle* joué par un utilisateur qui interagit avec le système. La même personne physique peut jouer le rôle de plusieurs acteurs. D'autre part, plusieurs personnes peuvent jouer le même rôle, et donc agir comme un même acteur.

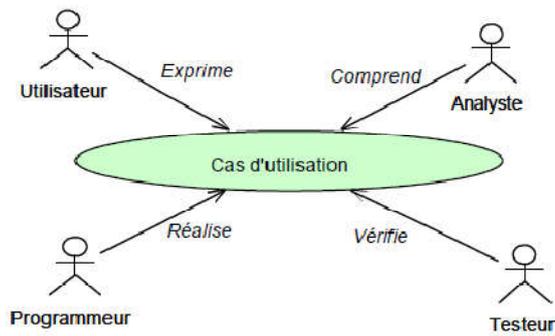


Figure 1 : les cas d'utilisation, fil conducteur du projet.

Un acteur se représente par un petit bonhomme avec son nom (i.e. son rôle) inscrit dessous.



Figure 2 : Exemple de représentation d'un acteur.

Il est également possible de représenter un acteur sous la forme d'un classeur stéréotypé << actor >>.



Figure 3 : Exemple de représentation d'un acteur sous la forme d'un classeur.

### 5.2. Cas d'utilisation

Les cas d'utilisations représentent le dialogue entre l'acteur et le système de manière abstraite. 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 se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.

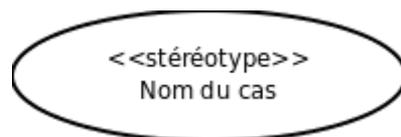


Figure 4 : Exemple de représentation d'un cas d'utilisation.

Dans le cas où l'on désire présenter les attributs ou les opérations du cas d'utilisation, il est préférable de le représenter sous la forme d'un classeur stéréotypé << use case >>.

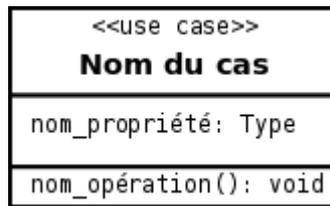


Figure 5 : Exemple de représentation d'un cas d'utilisation sous la forme d'un classeur.

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

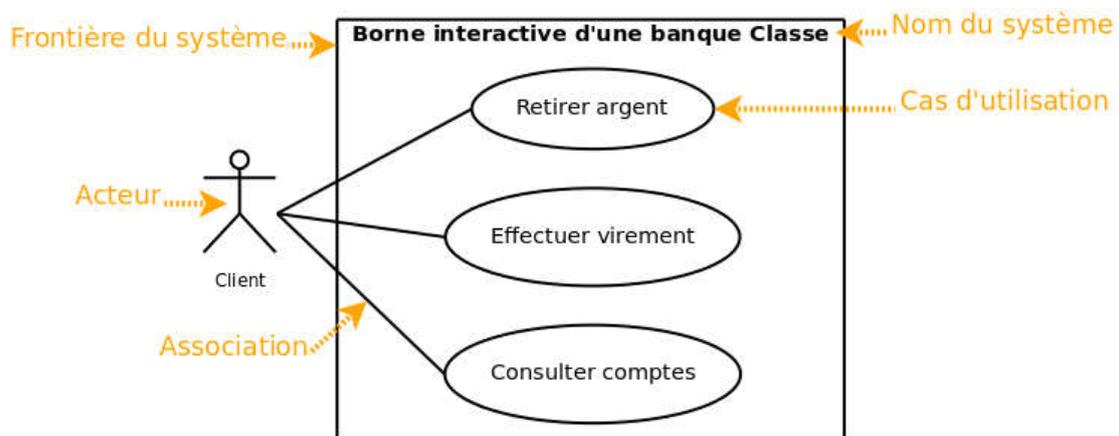


Figure 6 : Exemple simplifié de diagramme de cas d'utilisation modélisant une borne d'accès à une banque. Les acteurs sont à l'extérieur et les cas d'utilisation à l'intérieur.

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

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

##### a. Relation d'association



Figure 7 : Diagramme de cas d'utilisation représentant un logiciel de partage de fichiers.

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

Lorsqu'un acteur peut interagir plusieurs fois avec un cas d'utilisation, il est possible d'ajouter une multiplicité sur l'association du côté du cas d'utilisation.

### -6.1.2. Acteurs principaux et secondaires

Un acteur est qualifié de principal pour un cas d'utilisation lorsque ce cas rend service à cet acteur. Les autres acteurs sont alors qualifiés de secondaires. Un acteur principal obtient un résultat observable du système tandis qu'un acteur secondaire est sollicité pour des informations complémentaires. Le stéréotype << primary >> est utilisé pour relier l'acteur principal au cas d'utilisation, le stéréotype << secondary >> est utilisé pour les acteurs secondaires.

### -6.1.3. Cas d'utilisation interne

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

## -6.2. Relations entre cas d'utilisation

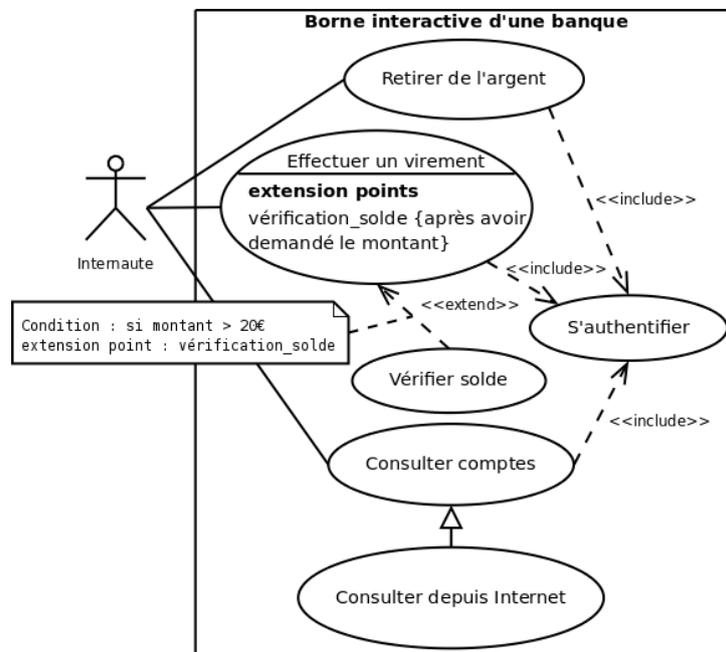


Figure 8 : Exemple de diagramme de cas d'utilisation.

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.

### a. 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.

Les inclusions permettent également de décomposer un cas complexe en sous-cas plus simples.

Figure 8.

**Attention !!! Les cas d'utilisation ne s'enchaînent pas, puisqu'il n'y a aucune Représentation temporelle dans un diagramme de cas d'utilisation.**

#### b. 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 >>. Une extension est souvent soumise à condition.

Graphiquement, la condition est exprimée sous la forme d'une note. Le cas est expliqué en figure 8.

#### c. Relation de généralisation

Un cas A est une généralisation d'un cas B si B est un cas particulier de A. 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.

#### -6.3. 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.

Le symbole utilisé pour la généralisation entre acteurs est une flèche avec un trait plein dont la pointe est un triangle fermé désignant l'acteur le plus général (comme nous l'avons déjà vu pour la relation de généralisation entre cas d'utilisation).

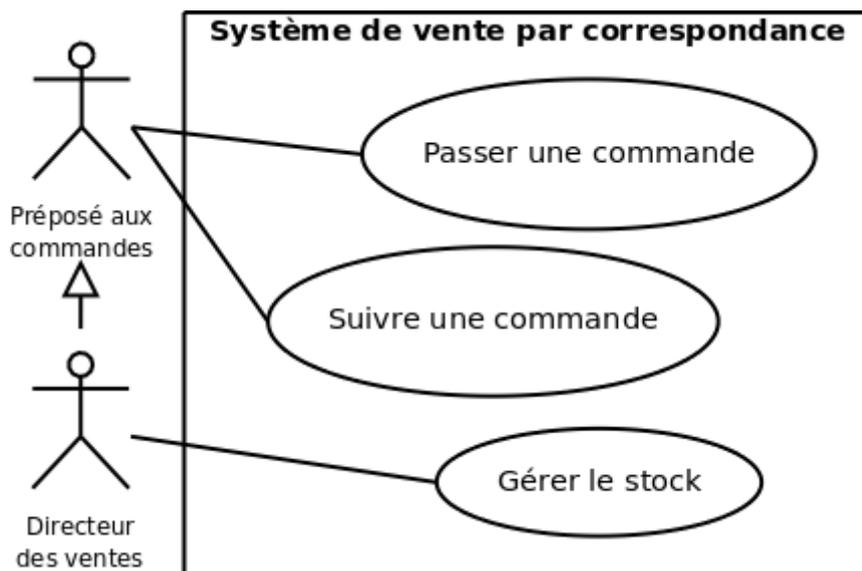


Figure 9 : Relations entre acteurs.

**Référence :**

1. G. Booch, J. Rumbaugh, I. Jacobson, « The Unified Modeling Language (UML) user guide », Addison-Wesley, 1999.
2. Benoit Charroux, Omar Osmani, Yann Therry-Mieg, « UML2 synthèse et exercices », Pearson édition France, ISBN 2-7440-7124-2, 2005.
3. G. Booch et al., « Object-Oriented Analysis and Design, with applications », Addison-Wesley, 2007.
4. Cours UML 2.0 de Laurent Audibert, site <http://www.developpez.com>