**Diagramme de cas d’utilisation**

1. **Le Diagramme de cas d’utilisation (Use Case Diagram) :**

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. Il scinde la fonctionnalité du système en unités cohérentes, les cas d’utilisation, ayant un sens pour les acteurs. 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.

Les cas d’utilisation décrivent sous la forme d’actions et de réactions, le comportement du système étudié du point de vue des utilisateurs. Entre un utilisateur et le système, un cas d’utilisation décrit les interactions liées à un objectif fonctionnel de l’utilisateur.

La description de l’interaction est réalisée suivant le point de vue de l’utilisateur. La représentation d’un cas d’utilisation met en jeu trois concepts :

***l’acteur, le cas d’utilisation et l’interaction entre l’acteur et le cas d’utilisation*.**

1. **Construction du diagramme de cas d’utilisation :**
2. **Les éléments du diagramme :**

* **Acteur :**

Un **acteur** est un utilisateur type qui a toujours le même comportement vis-à-vis d’un cas d’utilisation. Ainsi les utilisateurs d’un système appartiennent à une ou plusieurs classes d’acteurs selon les rôles qu’ils tiennent par rapport au système.

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 inscrit dessous.



* **Cas d’utilisation :**

Un **cas d’utilisation** correspond à un certain nombre d’actions que le système devra exécuter en réponse à un besoin d’un acteur. Un cas d’utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système. Une **interaction** permet de décrire les échanges entre un acteur et un cas d’utilisation.



1. **La représentation du diagramme :**

Tout système peut être décrit par un certain nombre de cas d’utilisation correspondant aux besoins exprimés par l’ensemble des utilisateurs. À chaque utilisateur, vu comme acteur, correspondra un certain nombre de cas d’utilisation du système. L’ensemble de ces cas d’utilisation se représente sous forme d’un diagramme.

La figure ci-dessous modélise une borne interactive qui permet d’accéder à une banque. Le système à modéliser apparaît dans un cadre (cela permet de séparer le système à modéliser du

monde extérieur). Les utilisateurs sont représentés par des petits bonshommes, et les grandes

fonctionnalités (les cas d’utilisation) par des ellipses.

L’ensemble des cas d’utilisation contenus dans le cadre constitue « un sujet ». Les petits

bonshommes sont appelés « acteurs ». Ils sont connectés par de simples traits (appelés

« associations ») aux cas d’utilisation et mettent en évidence les interactions possibles entre

le système et le monde extérieur. Chaque cas modélise une façon particulière et cohérente

d’utiliser un système pour un acteur donné.

****

1. **Les relations dans le diagramme : (inclusion, extension et généralisation)**
2. **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 (tous les cas d’utilisation accessibles à A le sont aussi à B, mais l’inverse n’est pas vrai).

La figure ci-dessous montre que le directeur des ventes est un préposé aux commandes avec un pouvoir supplémentaire (en plus de pouvoir passer et suivre une commande, il peut gérer le

stock). Le préposé aux commandes ne peut pas gérer le stock.



1. **Relations entre cas d’utilisation :**

Pour clarifier un diagramme, UML permet d’établir des relations entre les cas d’utilisation.

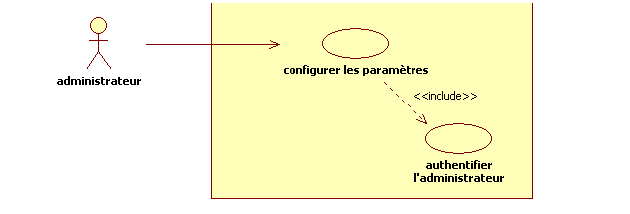
Il existe principalement deux types de relations : les dépendances stéréotypées et la généralisation/spécialisation. Les dépendances stéréotypées sont des dépendances dont la portée est explicitée par le nom du stéréotype. Les stéréotypes les plus utilisés sont l’inclusion et l’extension.

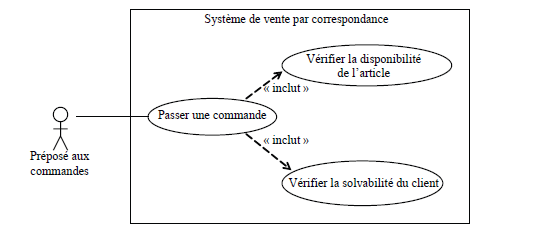
* ***La relation d’inclusion :***

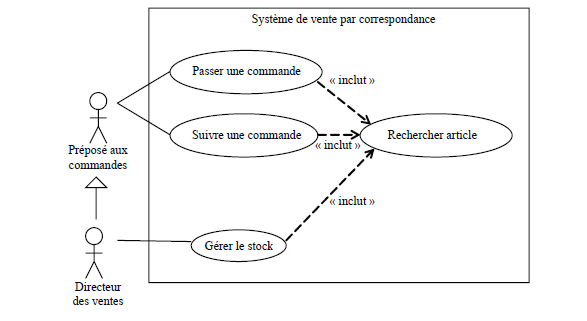
Un cas A est inclus dans un cas B si le comportement décrit par le cas A est inclus dans le comportement du cas B : on dit alors que le cas B dépend de A. Cette dépendance est symbolisée par le stéréotype *inclut*. Par exemple, l’accès aux informations d’un compte bancaire inclut nécessairement une phase d’authentification avec un mot de passe.

Une **relation d’inclusion** d’un cas d’utilisation A par rapport à un cas d’utilisation B signifie qu’une instance de A contient le comportement décrit dans B.

***Inclusion* :** B est une partie obligatoire de A et on lit *A inclut B* (dans le sens de la flèche).





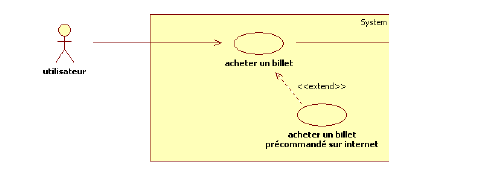


* ***La relation d’extension.*** Si le comportement de B peut être étendu par le comportement de A, on dit alors que A étend B. Une extension est souvent soumise à condition. Graphiquement, la condition est exprimée sous la forme d’une note.

Une **relation d’extension** d’un cas d’utilisation A par un cas d’utilisation B signifie qu’une instance de A peut être étendue par le comportement décrit dans B.

***Extension* :** B est une partie optionnelle de A et on lit *B étend A* (dans le sens de la flèche).





* ***La relation de généralisation.*** Un cas A est une généralisation d’un cas B si B est un cas particulier de A. La consultation d’un compte bancaire *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. Les inclusions permettent aussi de décomposer un cas complexe en sous-cas plus simples.



