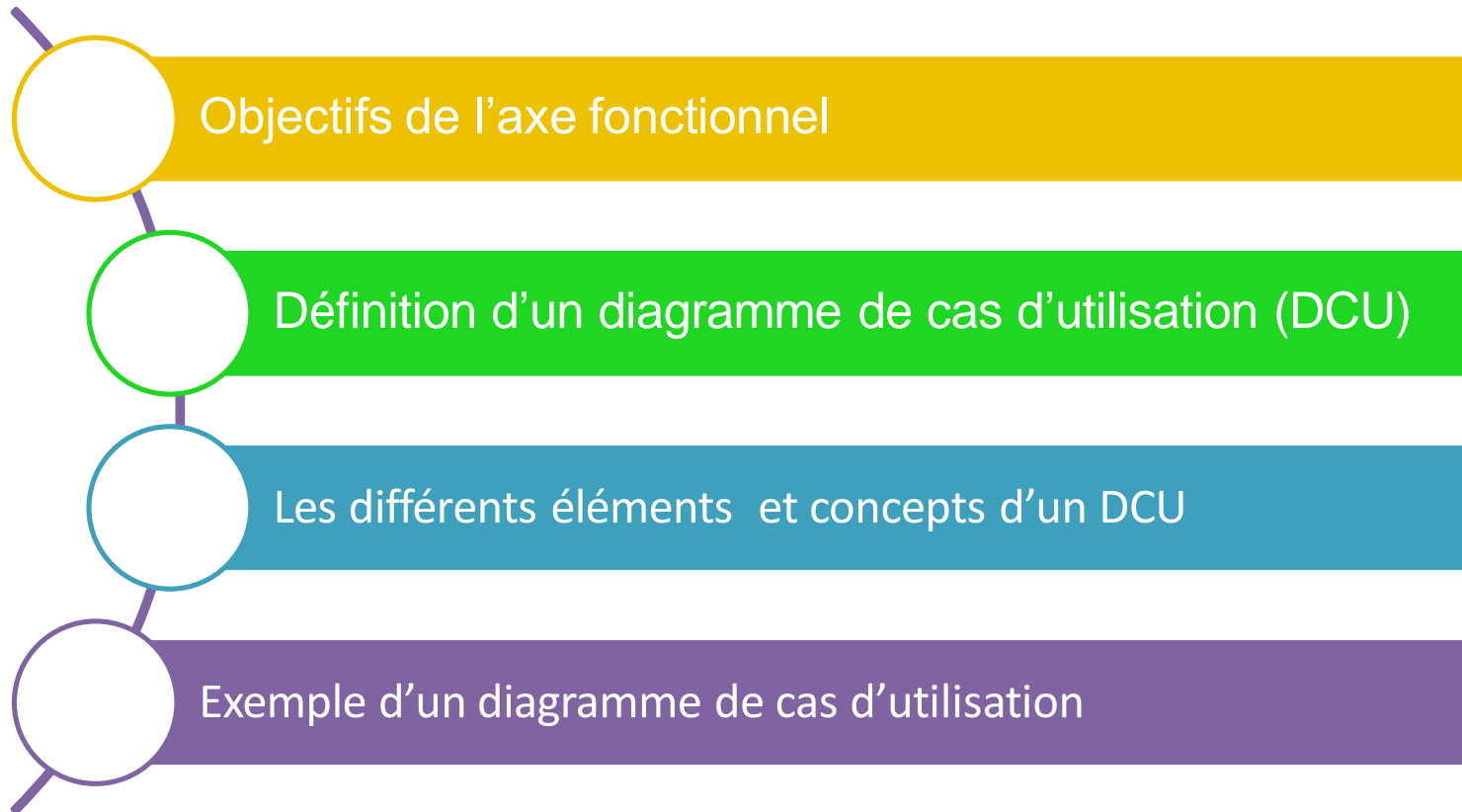




Chapitre I: Analyse fonctionnelle

Module Langage de modélisation UML

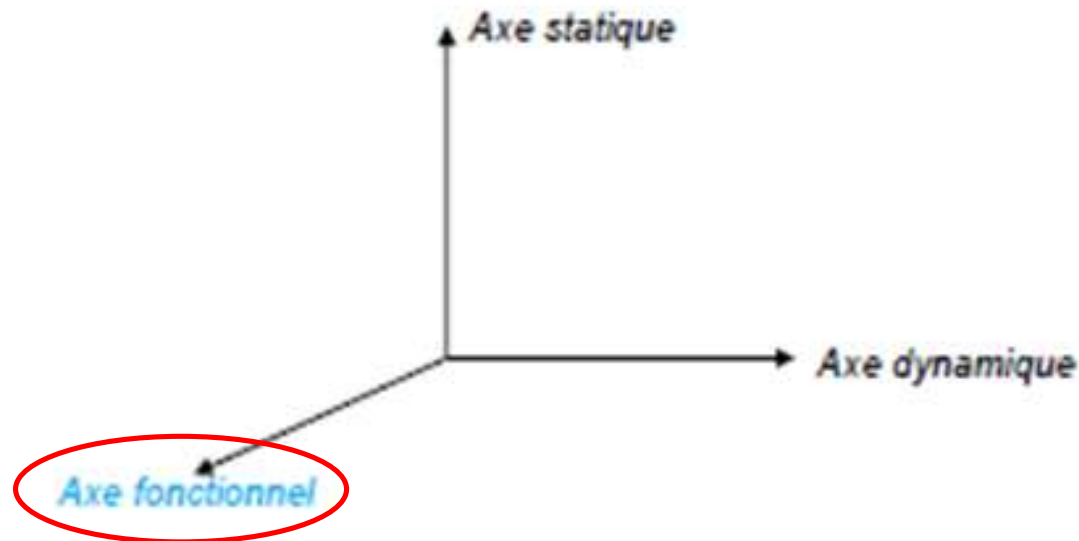
PLAN



Objectifs de l'axe fonctionnel


Axe fonctionnel = Vue utilisateur

- Description du fonctionnement du système.
- Détermination des besoins attendus par chaque acteur?
le QUOI?
- Concevoir une application logicielle de qualité qui répond aux besoins des utilisateurs/clients en respectant les diverses contraintes.



Qu'est-ce que le diagramme des cas d'utilisation?

- ❖ Modélise les besoins fonctionnels des utilisateurs.
- ❖ Identifie les grandes fonctionnalités et les limites du système.
- ❖ Représente les interactions entre le système et ses utilisateurs
- ❖ Elabore l'inventaire des fonctionnalités d'un système d'un point de vue utilisateur.

 Le diagramme des cas d'utilisation apporte une vision utilisateur et absolument pas une vision informatique

Définition d'un Diagramme de cas d'utilisation

Importance d'un diagramme de cas d'utilisation



Les différents éléments et concepts d'un DCU

Pour découvrir et comprendre les différents éléments d'un diagrammes de cas d'utilisation nous allons suivre le plan suivant:

1. Acteur

1.1 Définition d'un acteur

1.2 Types d'acteurs

1.3 Relation entre les acteurs: Héritage

2. Cas d'utilisation.

2.1 Définition d'un cas d'utilisation

2.2 Relation entre les cas d'utilisation

2.2.1: Inclusion

2.2.2: Extension

2.2.3 : Généralisation

1. Les acteurs

1.1 Définition d'un acteur

❖ Un acteur est un utilisateur **EXTERNE** et en relation **DIRECTE** avec le système.

❖ Cela peut être :

A. Une personne

➔ Modélisation: Le stick man si l'acteur est humain



B. Du matériel (capteurs, moteurs, relais...) ou un autre système (autre que le système en question).

➔ Modélisation :Le classeur si l'acteur est du matériel ou un autre système



Remarque importante : En UML, une annotation entre guillemets est appelé '**stéréotype**'. Cela permet de préciser et de mieux caractériser l'élément à qui il s'adresse.

1. Les acteurs

1.2 Types d'acteurs : principal et secondaire

A. Acteur principal

- ❖ **Déclenche** une des fonctionnalités du système
- ❖ Obtient un résultat observable

B. Acteur secondaire

- ❖ **Contribue** à l'achèvement d'un cas d'utilisation (fonctionnalité du système)

➔ l'acteur principal sollicite le cas d'utilisation alors que l'acteur secondaire est sollicité par le cas d'utilisation

➔ Un même acteur peut être principal pour un cas d'utilisation et secondaire pour un autre cas.

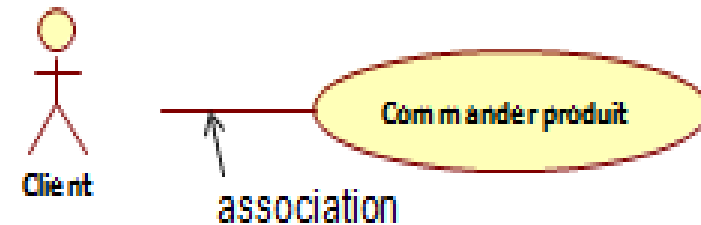
1. Les acteurs

1.2 Types d'acteurs : principal et secondaire

■ Modélisation Acteur Principal

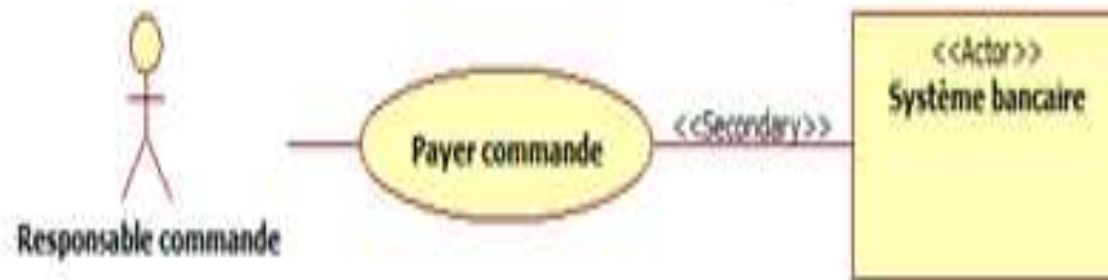
Chemin de communication entre un acteur et un cas

Est représentée par un trait continu



■ Modélisation Acteur secondaire

Ajout du stéréotype « secondary » sur la relation



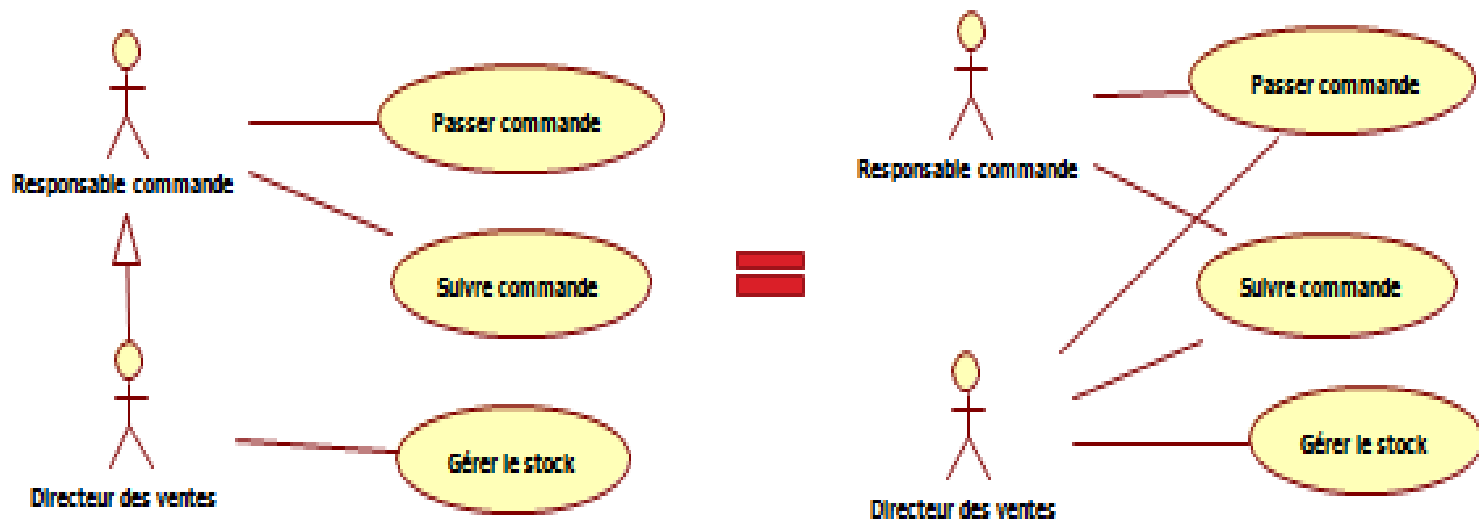
1. Les acteurs

1.3 Relation entre les acteurs: Héritage (Généralisation)

La seule relation possible entre deux acteurs est la **L'héritage (dite aussi 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 de la flèche pointe vers l'acteur le plus général.

Exemple :

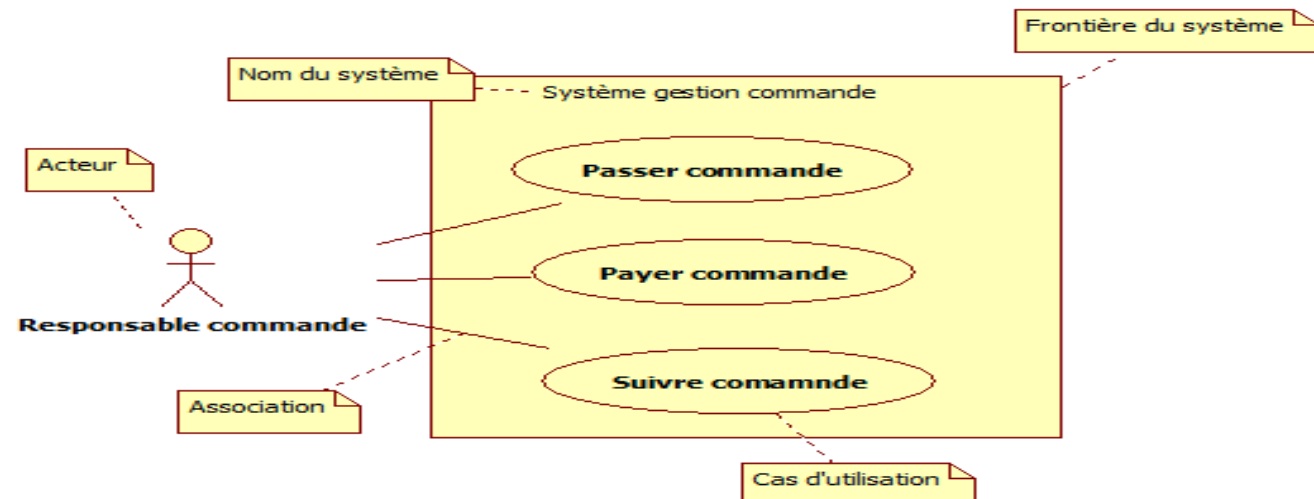


2. cas d'utilisation

1. Définition d'un Cas d'utilisation

- ❖ Fonctionnalité visible de l'extérieur du système dont on désire décrire le fonctionnement.
- ❖ Réponse à un besoin service rendu à l'utilisateur.
- ❖ Implique des séries d'actions plus élémentaires.
- ❖ Exprime un service réalisé de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie.

Modélisation:



2. cas d'utilisation

2.2.1 Relations entre les cas d'utilisations: Inclusion

- ❖ La relation d'inclusion sert à enrichir un cas d'utilisation par un autre cas d'utilisation (c'est une sous fonction).
- ➔ La relation d'inclusion est impérative (obligatoire) et donc systématique

Exemples :

C1 utilise C2 \Rightarrow toute activation de C1 entraîne une activation de C2



2. cas d'utilisation

2.2.2 Relations entre les cas d'utilisations: Extension

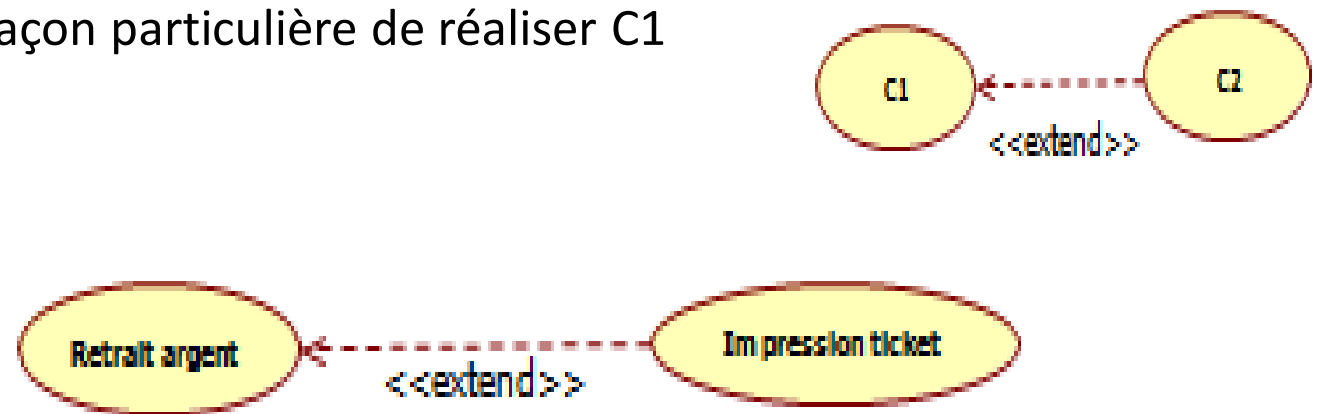
❖ Comme la relation d'inclusion, la relation d'extension enrichit un cas d'utilisation par un autre cas d'utilisation de sous fonction mais celui-ci est optionnel.

➔ La relation d'exclusion est optionnelle (non systématique).

Exemples :

Cette relation est représentée par une flèche en pointillée reliant les 2 cas d'utilisation et munie du stéréotype « extend »

❖ C2 étend C1 \Rightarrow C2 est une façon particulière de réaliser C1



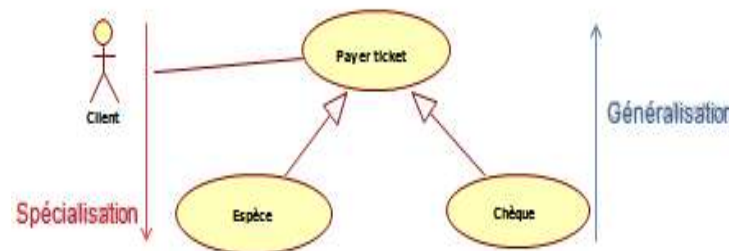
2. cas d'utilisation

2.2.3 Relations entre les cas d'utilisations: Généralisation

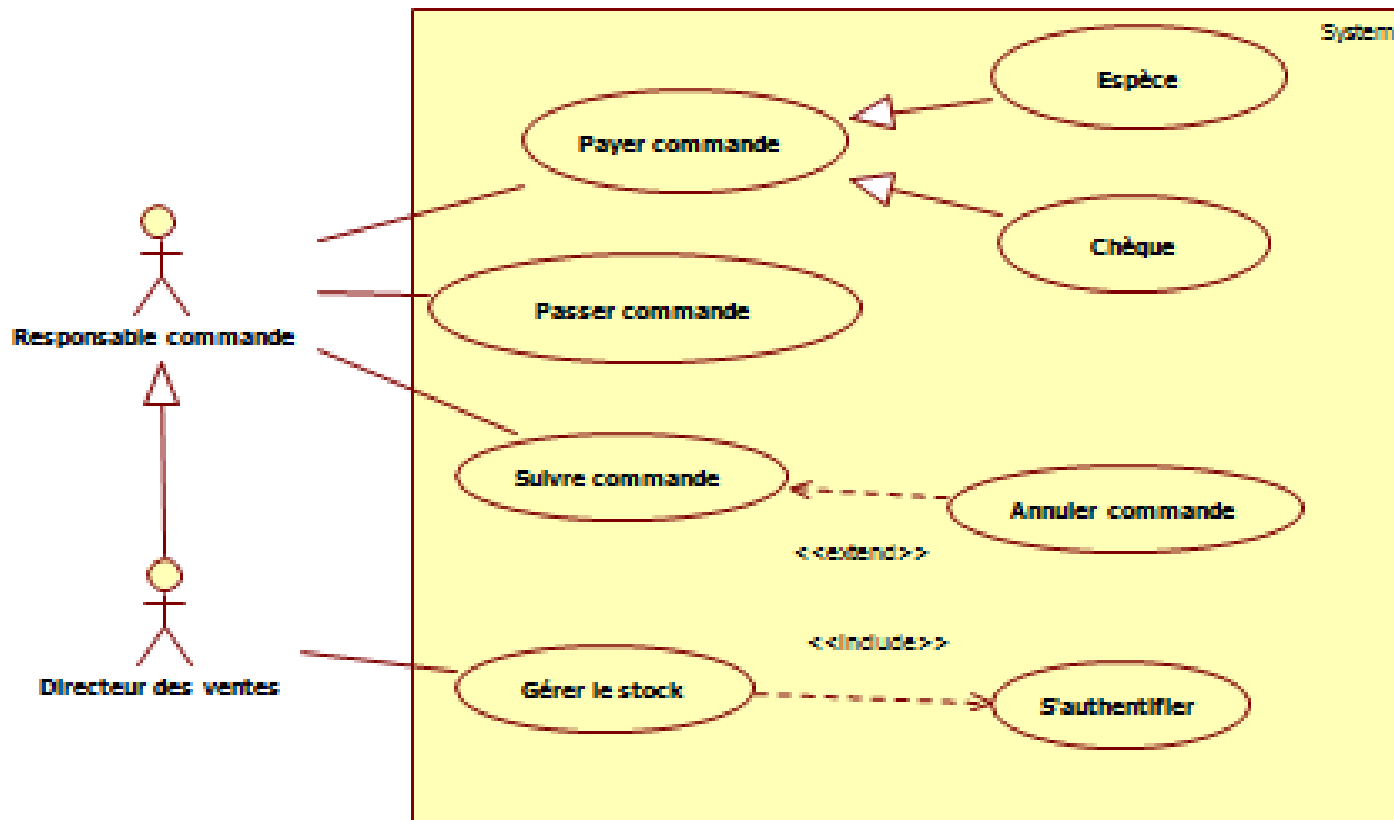
- ❖ La généralisation est un concept fondamental en programmation, en analyse et en conception orientée objet.
- ❖ Cette idée appliquée aux acteurs et aux cas d'utilisation est appelée ***généralisation/spécialisation***
- ❖ Le cas général est considéré comme un cas **abstrait**

Exemples:

- ❖ La relation de généralisation est représentée par une flèche avec une extrémité triangulaire.



Exemple d'un diagramme de cas d'utilisation



Etude de cas

Une société désire développer un site de vente de livres. La principale fonction offerte par le site est la recherche d'ouvrages. Le site doit offrir plusieurs méthodes de recherche : par titre, par N°ISBN ou par auteur. Le client doit pouvoir aussi accéder aux classements des meilleures ventes de livres. Il peut s'il le souhaite imprimer le classement.

Le client a la possibilité de passer la commande en ligne. Le client effectue son paiement sur le web via sa carte bancaire.

Le client doit pouvoir ensuite suivre ses commandes récentes et les détails de livraisons.

Evidemment, le client doit s'authentifier pour pouvoir accéder aux différentes fonctionnalités.

Etude de cas

