

Chapitre 4 : Vue fonctionnelle d'UML

Introduction :

Par comportement, on sous entend le fonctionnement et la dynamique d'un système. Dans ce chapitre, nous allons étudier la vue fonctionnelle d'UML.

En effet, les diagrammes fonctionnels permettent de représenter :

- Le fonctionnement d'un système et ses interactions avec ses utilisateurs grâce aux diagrammes de cas d'utilisation ;

Avant de passer à ces différents diagrammes, il est important de cerner le domaine ainsi que les utilisateurs du système futur ceci en introduisant le diagramme de contexte.

1- Le diagramme de contexte :

Ce diagramme n'est pas officiellement désigné comme diagramme UML. Il ne fait donc pas partie des diagrammes « officiels », mais il est utile pour la définition des acteurs, avant de commencer à s'intéresser à d'autres aspects, tels que les packages et les cas d'utilisation.

Le diagramme de contexte met évidence le champ d'application et les acteurs intervenant.

La modélisation graphique est comme suit :

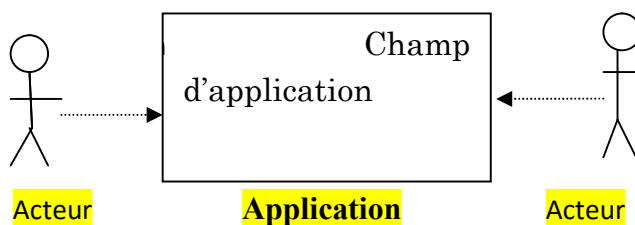


Figure1 : diagramme de contexte

2- Le diagramme de packages

Le diagramme de package permet de décomposer le système (champ d'application) en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d'indiquer les acteurs qui interviennent dans chacun des packages.

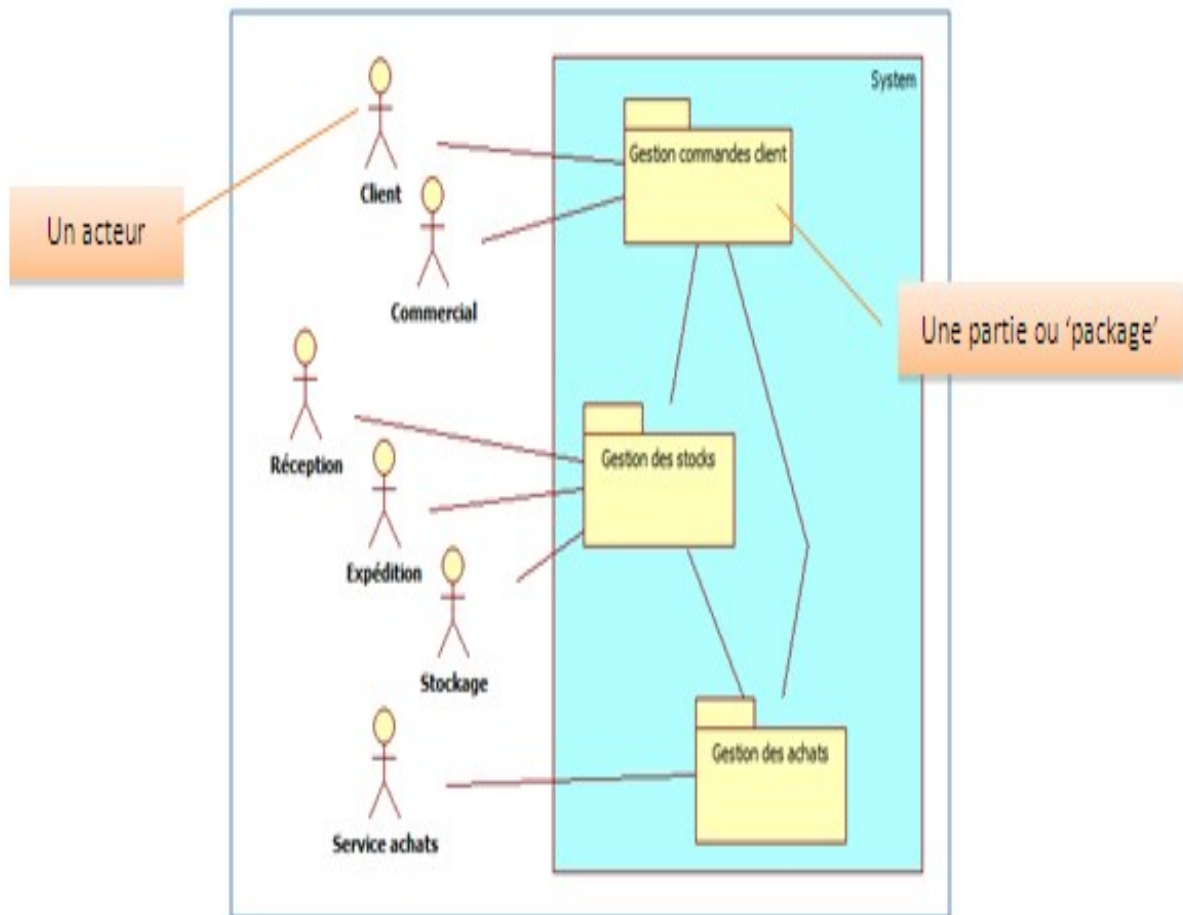


Figure 2 : Un exemple de diagramme de package

Dans l'exemple précédent, nous voyons que le logiciel que nous concevons peut être divisé en trois parties (ou packages) observables séparément :

1. La gestion des commandes client
2. La gestion des stocks
3. La gestion des achats

La boîte qui entoure les packages (la boîte bleue) correspond au système (c'est-à-dire le logiciel) qui est analysé.

3- Diagrammes de cas d'utilisation :

Le diagramme de cas d'utilisation représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d'utilisation pour le logiciel entier ou pour chaque package.

3-1- Les cas d'utilisation :

Les cas d'utilisation expriment le comportement du système (ses fonctionnalités) selon le point de vue des utilisateurs.

✓ **Définition :**

Les cas d'utilisation (en anglais use cases) ont été introduits par IVAR Jacobson dans sa méthode OOSE (Oriented Object Software Engineering en français génie logiciel orienté objet). Ils constituent une technique qui permet de déterminer les besoins des utilisateurs et de capturer les exigences fonctionnelles d'un système. En d'autres termes, ils décrivent le comportement d'un système du point de vue de ses utilisateurs. Ils décrivent les interactions entre les utilisateurs d'un système et le système lui-même.

✓ **Constituants des cas d'utilisation :**

Un modèle de cas d'utilisation comprend : les acteurs, le système et les cas d'utilisations eux-mêmes.

▪ **Acteur :**

Un acteur est un rôle joué par une personne physique ou un autre système. Il représente « un type d'utilisateur ».

• **Système :**

L'ensemble des cas d'utilisations décrit le but du système à modéliser.

• **Cas d'utilisation :**

Un cas d'utilisation est un ensemble d'actions réalisées par le système en réponse à une action d'un acteur.

3-2- Les diagrammes de cas d'utilisation :

Ils délimitent le système, ses fonctions (ses cas d'utilisation), et ses relations avec son environnement. Ils modélisent à la fois des activités (fonctionnalités) et des communications (interactions). Ils constituent un moyen de déterminer les besoins du système.

Les diagrammes de cas d'utilisation comprennent les acteurs, le système et les cas d'utilisation. Ils décrivent le dialogue entre les acteurs et le système représenté comme un ensemble de cas d'utilisation.

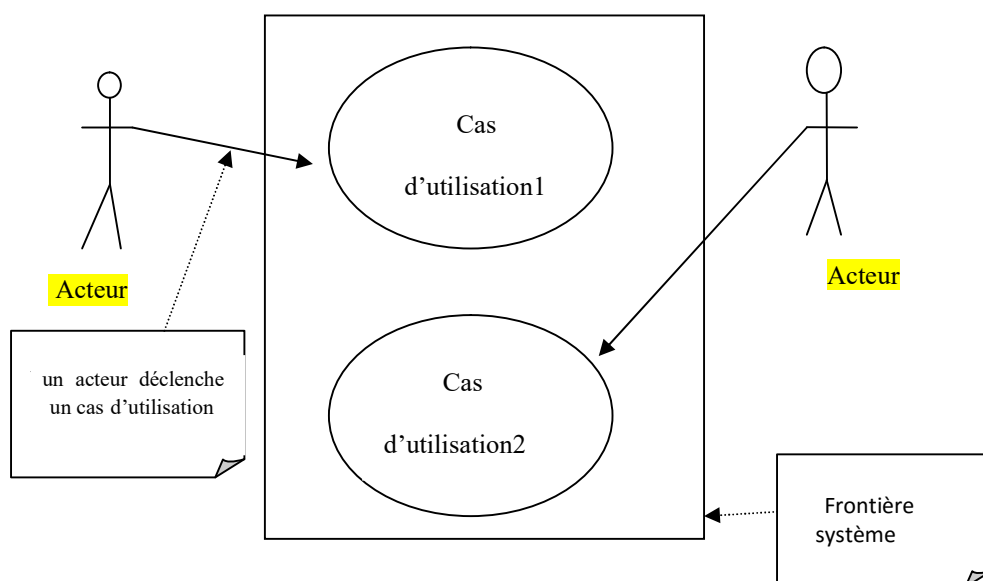


Figure 3 : Diagramme de cas d'utilisation

✓ **Exemple de diagramme de cas d'utilisation :**

Soit le scénario (ou la description textuelle) d'un cas d'utilisation :

- Un client fait une commande d'un produit.
- Le fournisseur traite la commande.
- Le fournisseur livre le produit et prépare la facture.
- Après réception de la facture et du produit*, le client paye la facture.

La représentation de ce scénario sous forme de diagramme de cas d'utilisation sera comme suit :

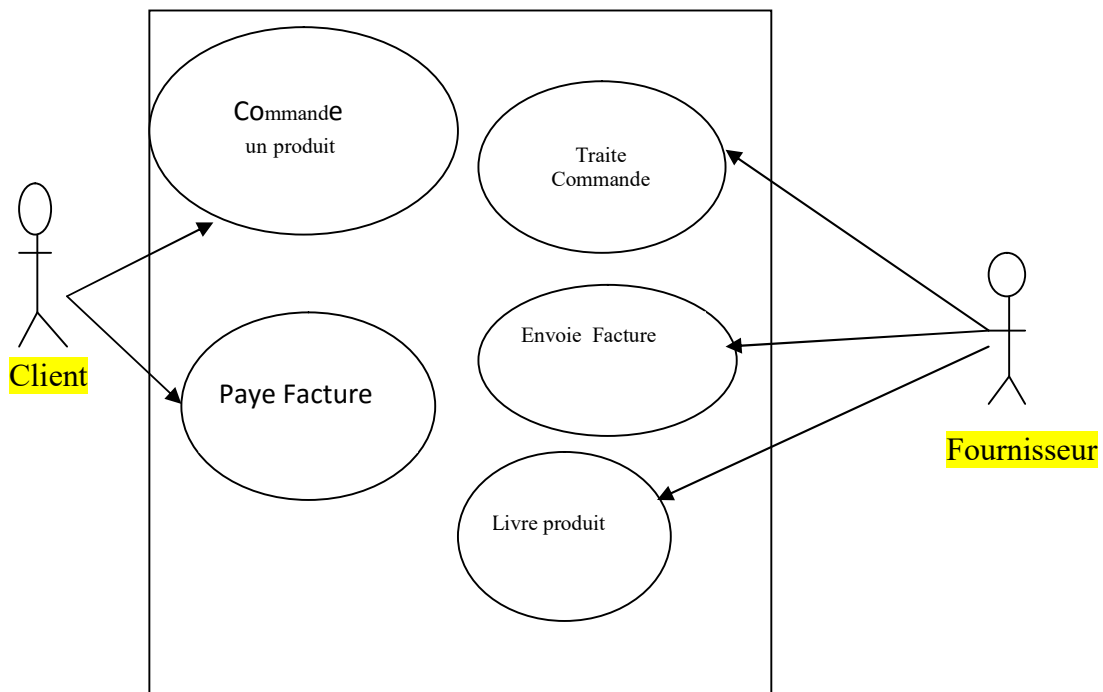


Figure : Un exemple de diagramme de cas d'utilisation

3-3- Relation entre les cas d'utilisation :

Les cas d'utilisation peuvent être structurés. En plus de la relation de communication, qui consiste au déclenchement d'un cas d'utilisation par un acteur, nous pouvons citer deux types de liens ou relations qui sont les plus utilisés : le lien d'utilisation (**include**) et le lien d'extension (**extend**)

✓ **Include :**

Le cas d'utilisation source incorpore explicitement et de manière obligatoire le comportement décrit dans le cas d'utilisation destination.

Notation graphique :

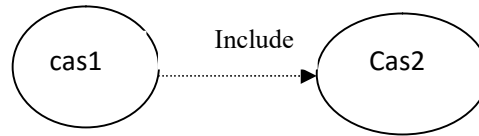


Figure 5 : Lien d'utilisation entre cas d'utilisation

Exemple : Pour imprimer le solde d'un compte, il faut d'abord consulter ce compte.

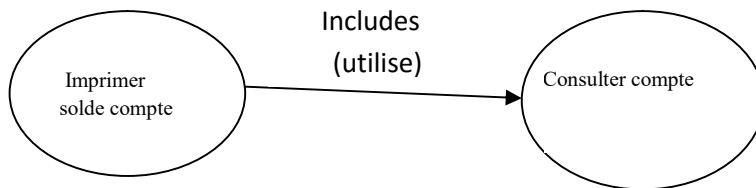


Figure 6 : Un exemple de lien d'utilisation (includes) entre cas d'utilisation

✓ **Extend :**

Le cas d'utilisation incorpore implicitement et de manière facultative un autre cas d'utilisation à l'endroit spécifié. Ceci veut dire que le cas d'utilisation source « étend » (en anglais « extend ») ou précise le cas d'utilisation destination.

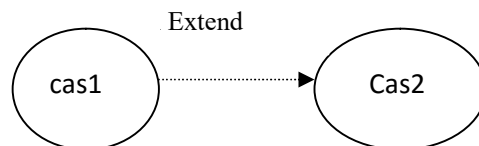


Figure 7 : Lien d'extension entre cas d'utilisation

Exemple :

Si on veut retirer de l'argent, on doit préciser si on veut retirer des dinars ou des devises. Donc, retirer dinars et retirer devises sont des cas particuliers de retirer argent.

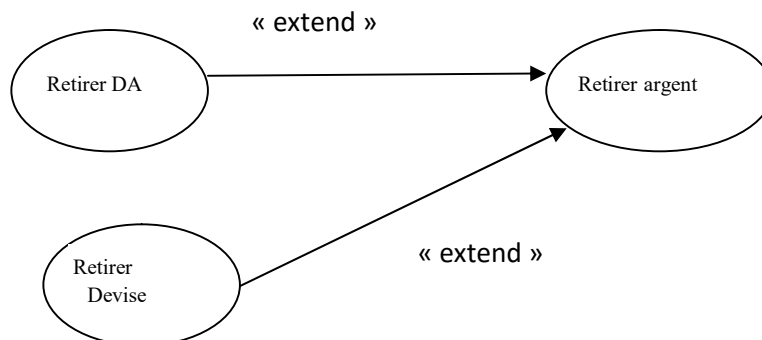


Figure 8 : un exemple de lien d'extension entre cas d'utilisation décrivant le retrait d'argent dans une banque.

Voici un exemple de diagramme de cas d'utilisation utilisant les packages, et les liens extend et includ.

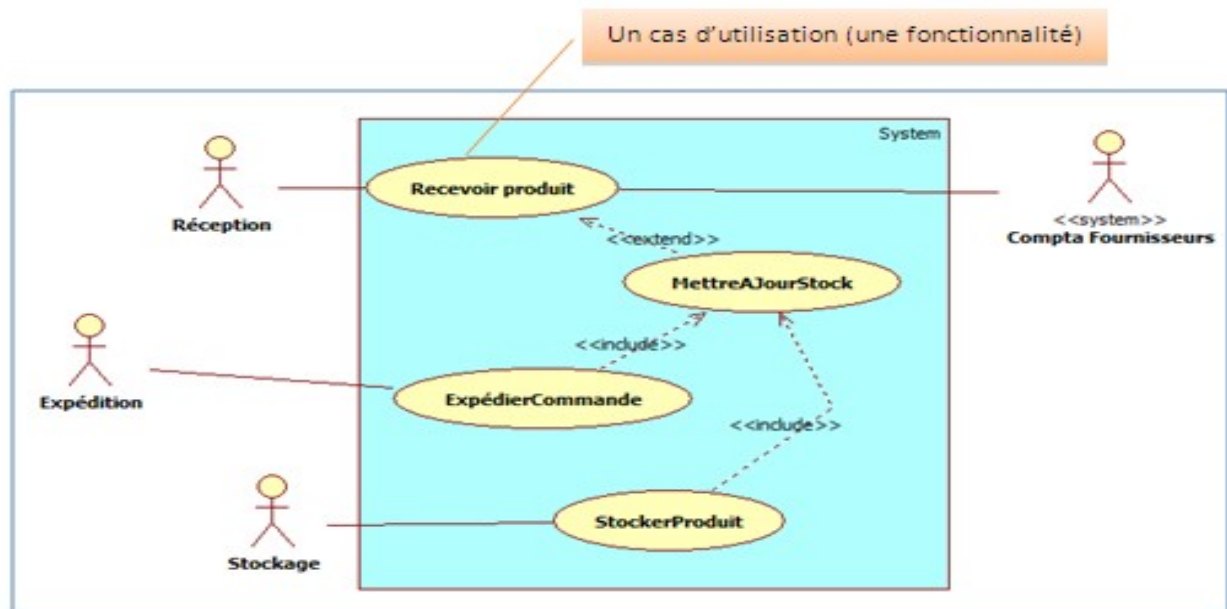


Figure 9 : Un exemple de diagramme de cas d'utilisation pour un package (Gestion des stocks)

Étant donné que le diagramme de cas d'utilisation détaille le contenu d'un package, ici la boîte bleue correspond au package qui est détaillé.