

Chapitre 3 : La vue statique d'UML : Les diagrammes de structure

1- Introduction : Les diagrammes de structure décrivent l'aspect statique du système. Ils permettent de représenter les éléments d'analyse : classes et les packages ainsi que le diagramme objets.

2- Diagrammes de classes :

2-1- Définition :

Les diagrammes de classes sont les diagrammes les plus utilisés d'UML. Ils décrivent les types des objets qui composent un système et les différents types de relations statiques qui existent entre eux.

Les diagrammes de classes font abstraction du comportement du système.

Exemple : Nous avons trois classes : fabricant, Produit et unité de vente et deux associations entre fabricant et produit nommée « fabriqué par » indiquant qu'un fabricant peut fabriquer un à plusieurs produits et qu'un produit est fabriqué par un à plusieurs fabricants. Et une association nommée « vend » entre produit et unité de vente décrivant le fait qu'une unité de vente vend un ou plusieurs produits et qu'un même produit est vendu par une ou plusieurs unités de vente.

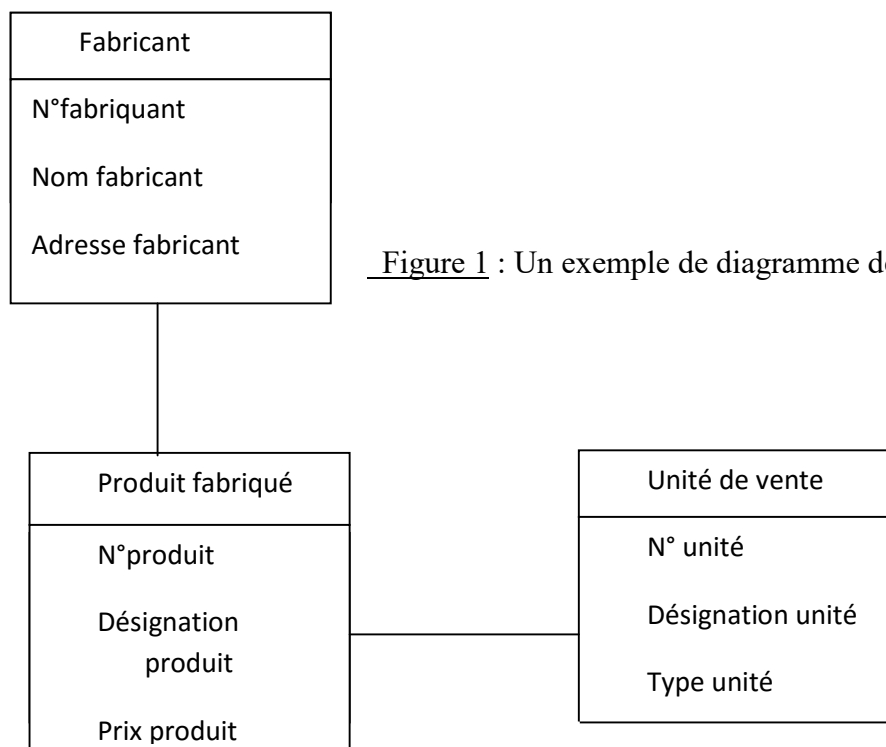


Figure 1 : Un exemple de diagramme de classes

2-2- Attributs de classes :

La syntaxe d'un attribut de classe est la suivante :

[Visibilité] Nom [: type][multiplicité][=valeur par défaut]

Où les crochets indiquent les clauses facultatives : seul nom est obligatoire.

- Visibilité : indique si l'attribut est public(+) ou privé (-) ;
- Nom : Nom donné à l'attribut ;
- Type : type de l'attribut ;
- Multiplicité : indique le nombre de valeurs possibles de l'attribut pour un objet ;
- Valeur par défaut : valeur par défaut attribué à l'attribut d'un objet s'il n'y a pas de valeur spécifiée à sa création.

Exemple :

- Nom : string [1]= « sans nom »

Dans cet exemple, nom est un attribut privé de type chaîne de caractères devant avoir une et une seule valeur. C'est-à-dire qu'un objet ne peut avoir qu'un seul nom. A la création d'objet, si on ne spécifie pas de valeur pour nom, on lui attribut la valeur par défaut « sansnom ».

2-3- Opération d'une classe :

La syntaxe d'une opération est la suivante :

[Visibilité] nom ([liste de paramètres])[: type] Où,

- ✓ Visibilité : indique public(+) ou privé (-)
- ✓ Nom : nom donné à l'opération ;
- ✓ Liste de paramètre : définition d'un ou de plusieurs paramètres ;
- ✓ Type : type de la valeur retournée par l'opération.

La syntaxe de la définition d'un paramètre est comme suit :

[Direction] nom : type [= valeur par défaut] où,

- ✓ Direction : in (en entrée), out (en sortie), inout (en entrée/sortie).

La direction par défaut est in.

- ✓ Nom : nom du paramètre ;

- ✓ Type : type du paramètre ;
- ✓ Valeur par défaut : valeur par défaut associée au paramètre.

Exemple : + DemanderPrix() : Monnaie

Dans cet exemple, DemanderPrix() est une opération publique qui effectue le calcul du prix d'une formation. Le résultat retourné par cette opération est de type monnaie.

Remarquons que formation n'est pas donnée comme paramètre de l'opération. Ceci est dû au fait que DemanderPrix doit être une opération de la classe formation.

+ CalculerNbAbsences(date_A : Date) : Entier.

Dans cet exemple, calculerNbAbsences est une opération publique qui effectue le calcul du nombre d'absences d'un employé à une date donnée. Le résultat retourné par cette opération est de type entier.

+DemanderSalairedeBase(catégorie : entier, section : entier) : Monnaie

+DemanderSalairedeBase() est une opération de la classe grille de Salaire, appelée par l'opération CalculerSalaire (..) de la classe employé. Elle effectue le calcul du salaire de base de l'employé. Le résultat retourné par cette opération est de type Monnaie.

2-4- Association entre classes : Une association est représentée par une ligne pleine entre deux classes.

Exemple :

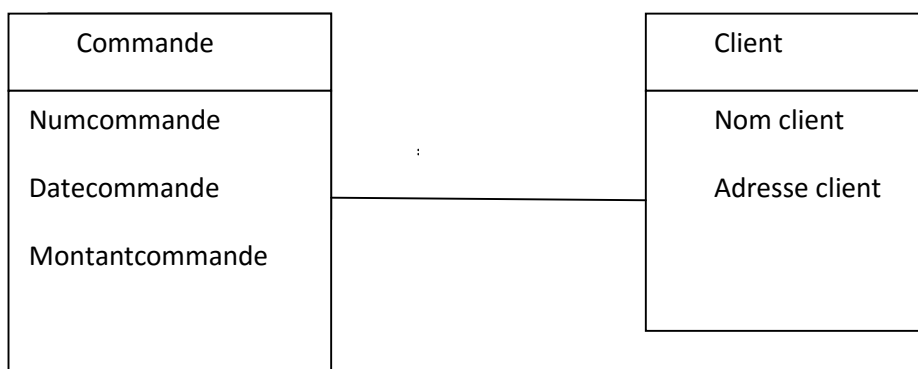


Figure 2 : Description de l'association « Commande – Client »

- a- Cardinalité :** Elle indique le nombre d'objets d'une classe qui peuvent participer à l'association. Elle est généralement définie par une borne inférieure et une borne supérieure. La borne inférieure est un entier positif ou nul et la borne supérieure est un entier strictement positif ou * (pour infini). Si les deux bornes sont égales, on utilisera un seul nombre (1..1 est équivalent à 1) comme * est équivalent à 0..*

Ainsi dans l'exemple de la figure 2, une commande donnée ne peut être associée qu'à un et un seul client : on met 1 à côté du client. Alors qu'un client peut ne pas faire de commande ou peut faire un nombre illimité de commandes on met * au côté commande.

b- Rôle : Il spécifie le rôle ou la fonction d'une classe dans une association donnée.

Exemple1 :

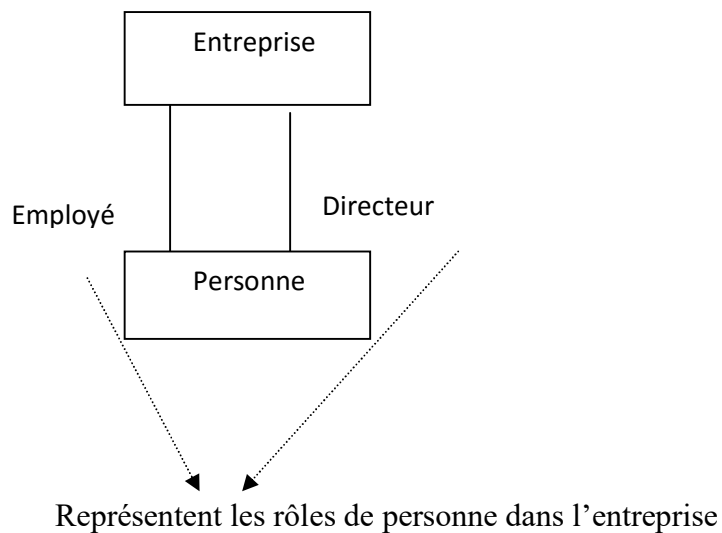


Figure 3 : Un Exemple d'association avec rôle

Exemple2 :

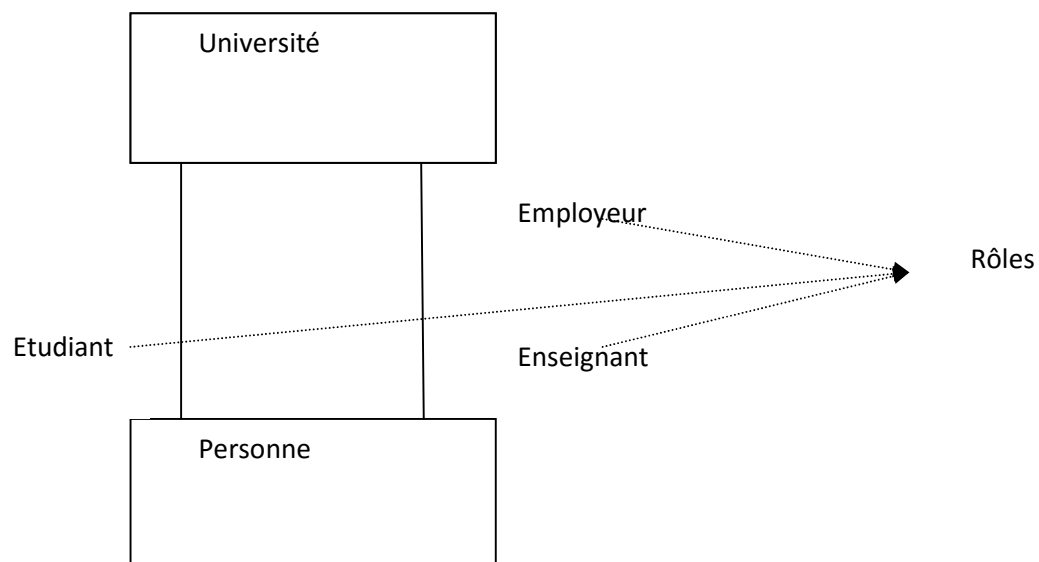


Figure 4 : Un autre exemple d'association avec rôles

C- Association n-aire : Il s'agit d'une association qui relie plus de deux classes

Exemple :

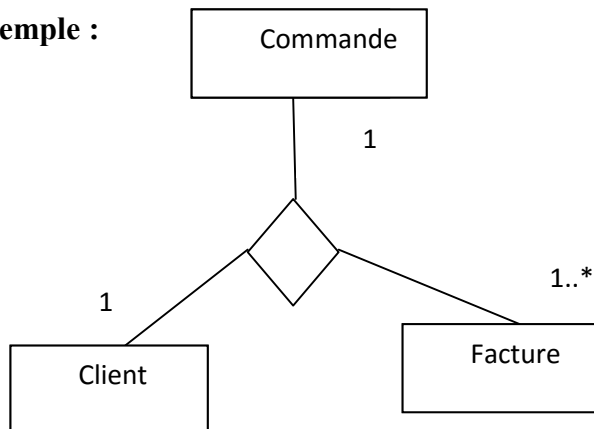


Figure 5 : Une Association n-aire

Expliquons, à travers le même exemple, comment déterminer les cardinalités dans une association n-aire.

- Un client pour une même commande peut avoir une ou plusieurs factures : 1..* au côté facture.
- Un client pour une facture est lié à une et une seule commande : 1 au côté commande.
- Une facture et une commande sont liées à un et un seul client : 1 au côté client.

d- Classe d'associations :

Lorsqu'une association possède des propriétés caractérisant le lien entre les classes associées, celle-ci est remplacée par une classe appelée classe associative et les propriétés du lien deviennent les attributs de cette classe.

Exemple 1 :

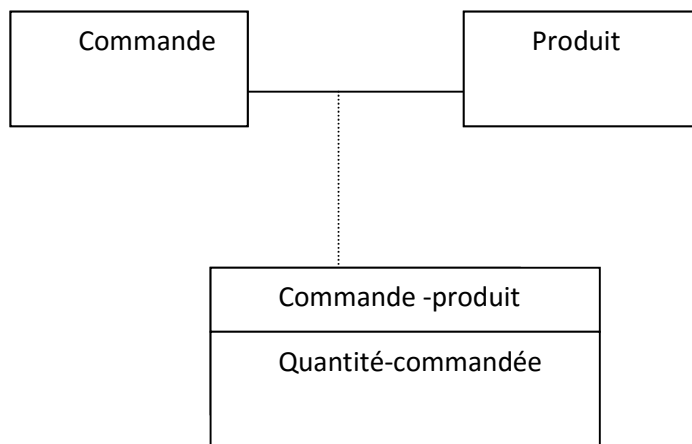


Figure 6 : une classe d'association porteuse de propriétés

Remarque : Pour les liens d'héritage, d'agrégation, de composition, leurs notations ont été données dans le chapitre 2.

e- Les stéréotypes :

UML autorise l'enrichissement ou la personnalisation de la notation au moyen des stéréotypes.

3- Diagramme d'objets ou diagramme d'instance :

Un diagramme d'objets représente un instantané des objets d'un système à un moment donné. Ces diagrammes sont souvent appelés diagrammes d'instances car ils représentent des instances et non pas des classes.

Exemple :

Reprenons l'exemple de la figure 3, où, nous avons représenté un lien entre deux classes Commande et Client. Si nous nous intéressons à un client en particulier pour voir les commandes qu'il a effectué, nous représentons ceci par un diagramme objets. Voir la figure suivante :

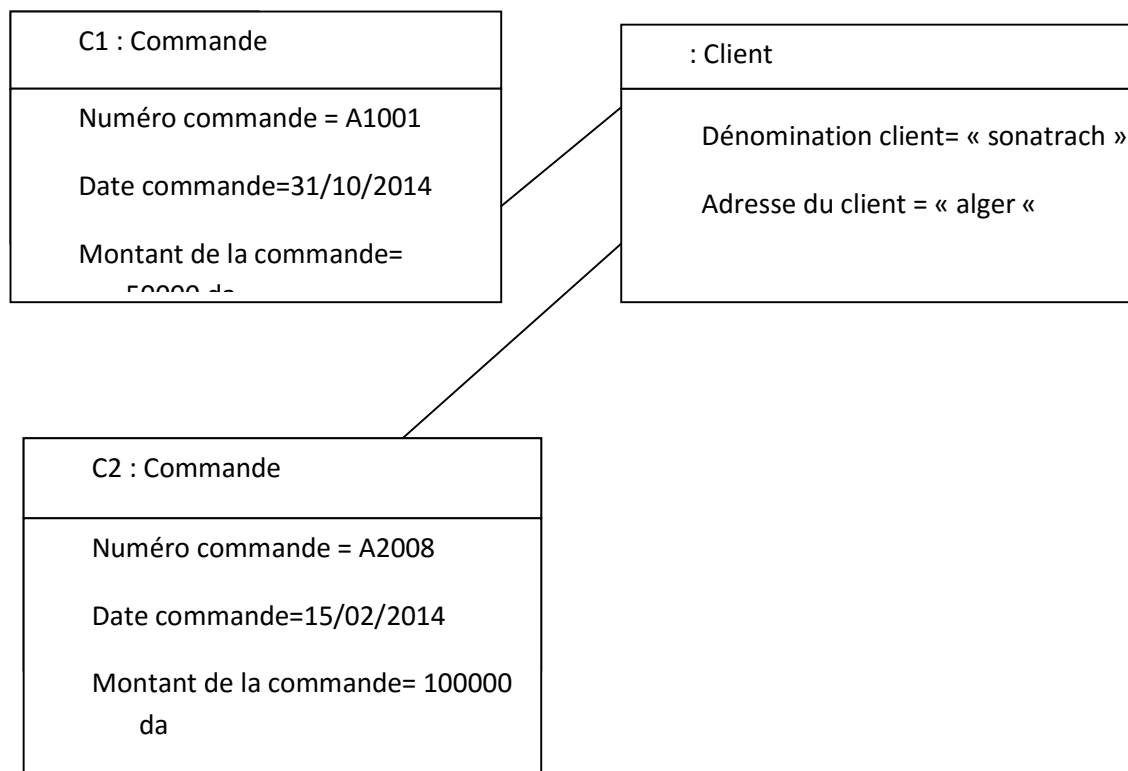


Figure 7 : instances de classes ou diagramme objets

4- Diagramme de package :

Les diagrammes de packages permettent de grouper des éléments UML dans des unités de plus haut niveau. En effet, ils peuvent servir à organiser des éléments UML selon leurs types, leurs fonctionnalités ou leurs architectures.

- **Notation graphique :** Un package (ou paquetage) est représenté graphiquement par un dossier.

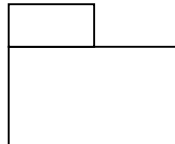


Figure 8 : un diagramme de package

Exemple :

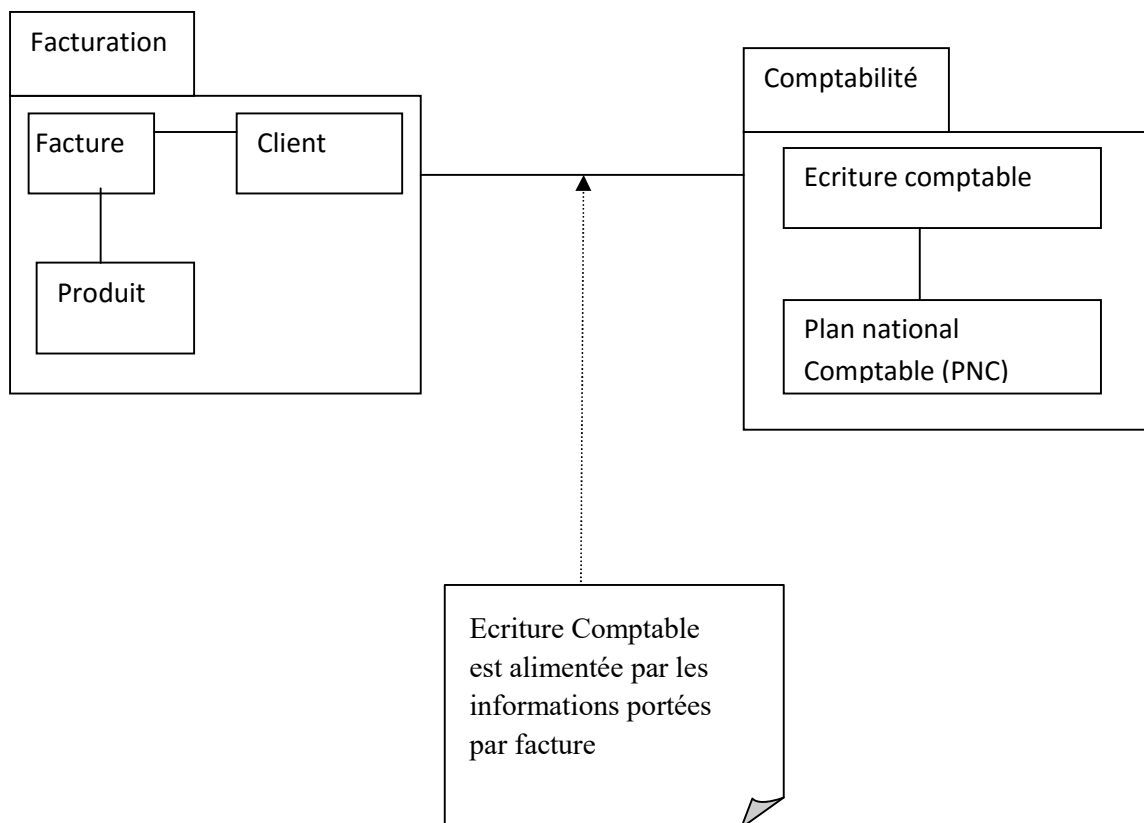


Figure 9 : Exemple de diagramme de package