

Chapitre 5: Vue dynamique d'UML

Introduction :

Dans ce chapitre, nous allons étudier les diagrammes d'interaction (Séquence et collaboration) permettant les interactions entre les objets. Le comportement des objets grâce aux diagrammes d'activités et les diagrammes d'états/ transitions.

1- Diagramme de séquence :

1-1- Définition :

Les diagrammes de séquences permettent de représenter les interactions entre les objets selon un point de vue temporel. L'accent est mis sur la chronologie des envois de messages.

✓ Notations graphiques :

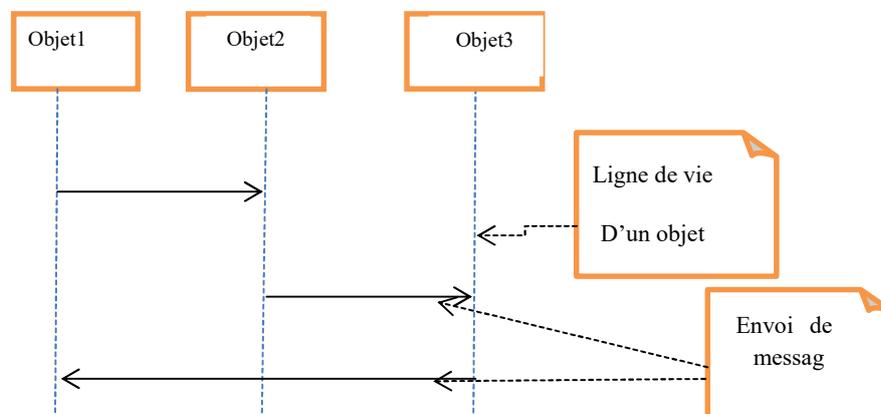


Figure 1 : diagramme de séquence

- L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme : le temps s'écoule de haut en bas de cet axe.
- La disposition des objets sur l'axe horizontal n'a pas de conséquence pour la sémantique du diagramme.
- Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.
- Les diagrammes de séquences peuvent servir à illustrer un [cas d'utilisation](#).

1-2- Barre d'activation :

Sur un diagramme de séquence, il est aussi possible de représenter de manière explicite les différentes périodes d'activité d'un objet au moyen d'une bande rectangulaire superposée à la ligne de vie de l'objet.

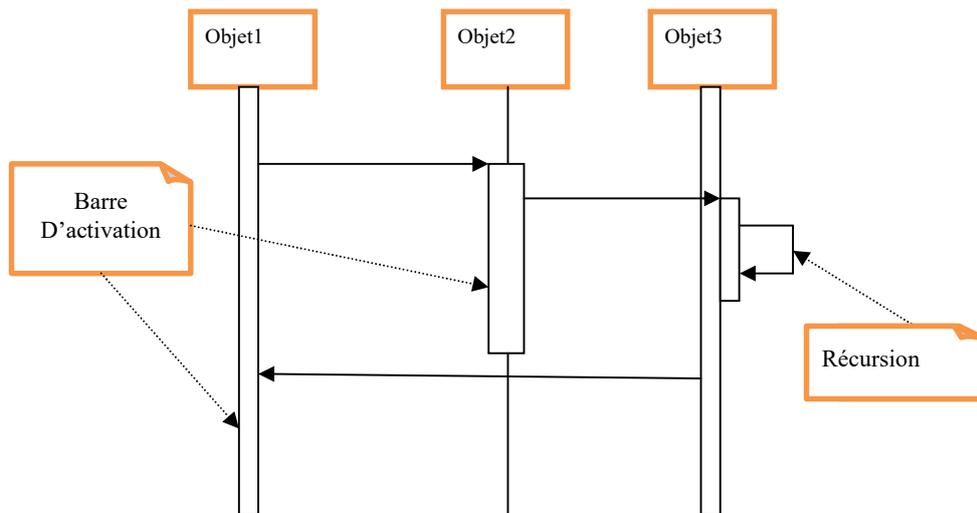


Figure 3 : Barre d'activation

Remarque : Un dédoublement de la barre indique un message d'appel récursif.

1-3- Appels synchrones et appels asynchrones :

Dans un message synchrone, l'appelant attend la réponse pour poursuivre son traitement alors que dans un message asynchrone, l'appelant continue son traitement sans attendre de réponse.

✓ **Notations graphiques :**



Synchrone



Asynchrone

Figure 4 : Notations synchrone et Asynchrone

Exemple :

L'exemple suivant représente un diagramme de séquence décrivant le calcul du Montant d'une commande. Une commande concerne un ou plusieurs produits. Elle comprend un certain nombre de lignes commandes où dans chacune d'elle on décrit un produit, la quantité commandée du même produit. Il y a une tarification pour chaque produit.

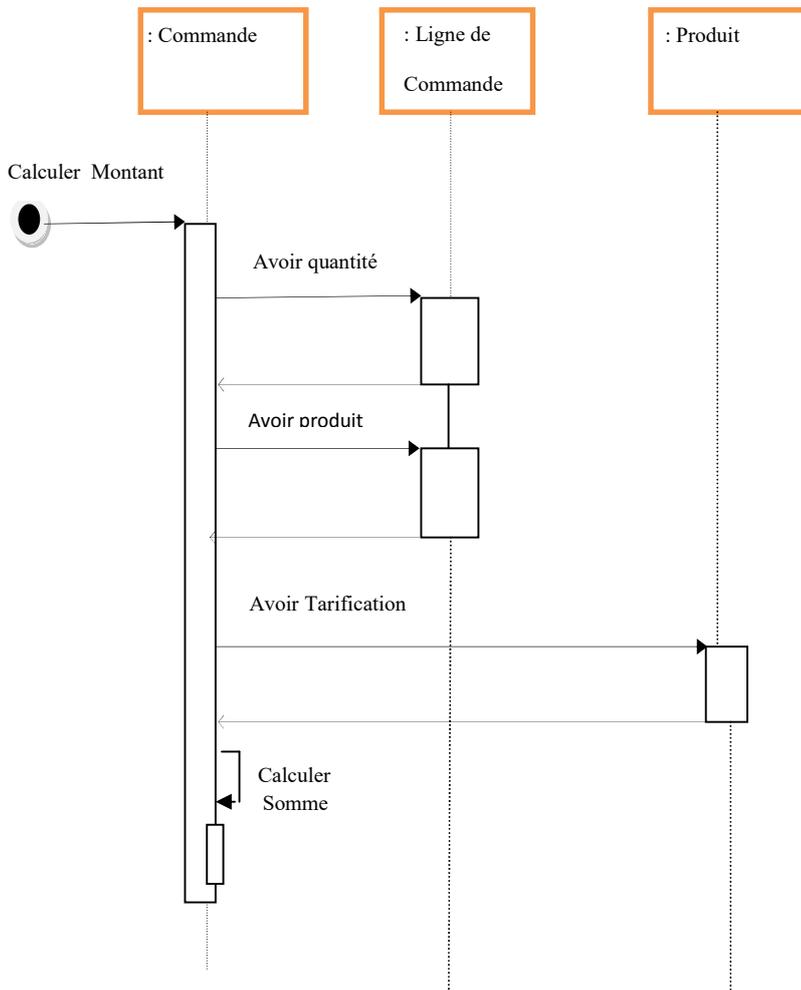


Figure 5 : Exemple de digramme de séquence

1-4- Instructions itératives et conditionnelles :

Pour représenter de manière graphique une exécution conditionnelle, itérative ou parallèle d'un message, on utilise des cadres d'interaction pour délimiter une zone dans un diagramme de séquence, en utilisant la clause « Alt », « loop », « par » pour respectivement alternative, itérative, parallèle.

Exemple : Le calcul décrit dans l'exemple précédent se fait en principe pour chaque ligne. Si on veut montrer l'aspect itératif du traitement dans le diagramme de séquence on utilisera un cadre d'interaction avec la clause « Loop » comme suit :

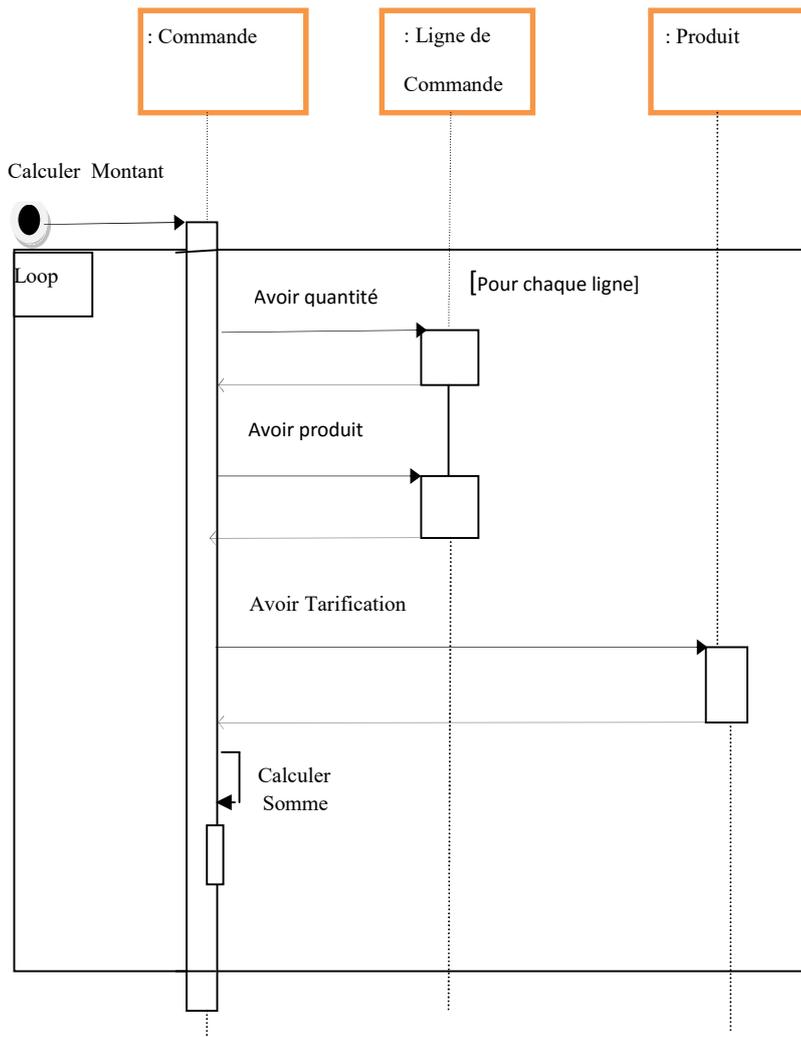


Figure 6 : Exemple de diagramme de séquence avec un cadre d'interaction itératif.

1-5- Avantages et inconvénients des diagrammes de séquences :

Les principaux avantages des diagrammes de séquences sont :

- Ils peuvent servir à illustrer un cas d'utilisation ;
- Ils sont bien adaptés pour visualiser les interactions entre les objets ;
- Ils montrent clairement le traitement effectué par chaque participant ;
- Notations claires.

Les inconvénients des diagrammes de séquences sont :

- Ils sont mal adaptés pour représenter les comportements itératifs et conditionnels ;
- Ils sont mal adaptés pour représenter un algorithme ;
- Ils ne permettent pas d'indiquer que les gardes (conditions booléennes) sont mutuellement exclusives.

2- Les diagrammes d'activité :

2-1- Définition :

Les diagrammes d'activité décrivent le comportement d'une méthode, le déroulement d'un cas d'utilisation, les enchaînements d'activités.

Une activité désigne une suite d'actions.

Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une action et provoquent le début immédiat d'une autre.

Notations graphiques :

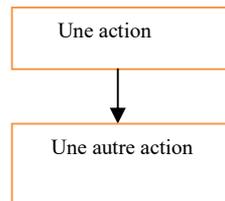


Figure 7 : Transition automatique entre actions

2-2- Décision/fusion :

Le comportement conditionnel est décrit par des décisions/fusions.

Une décision (ou branchement) permet de représenter des transitions conditionnelles en utilisant des gardes (expressions booléennes).

Une fusion marque la fin d'un comportement conditionnel.

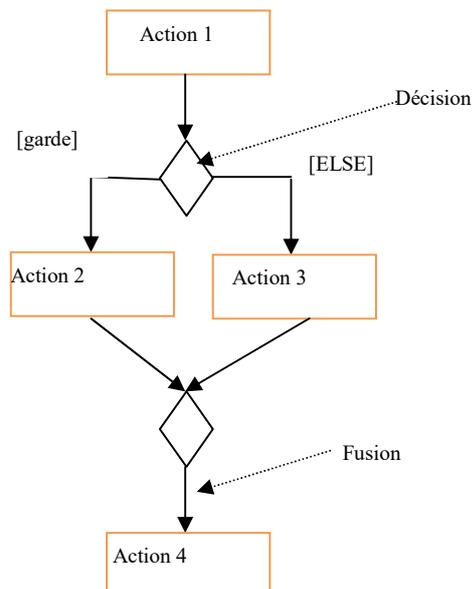


Figure 8 : Transaction conditionnelle

2-5- Débranchement et Jonction :

Il est possible de synchroniser les transitions à l'aide des barres de synchronisation. Ces dernières permettent d'ouvrir (débranchement) ou de fermer (jonction) des branches parallèles au sein d'un flot d'exécution.

- ❖ **Débranchement** : Les transitions qui partent d'un débranchement ont lieu en même temps.

❖ **Jonction** : On ne franchit une jonction qu'après réalisation de toutes les transitions qui s'y rattachent.

Exemple :

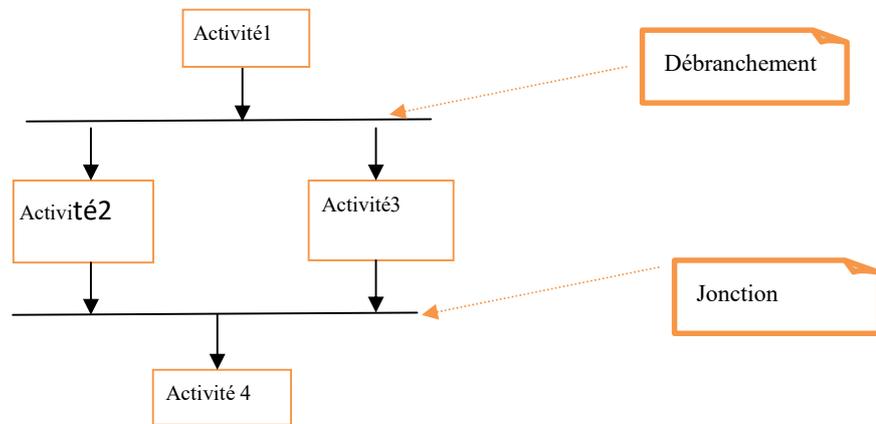


Figure 10 : Débranchement et jonction

Exemple de traitement d'une commande par un diagramme d'activité :

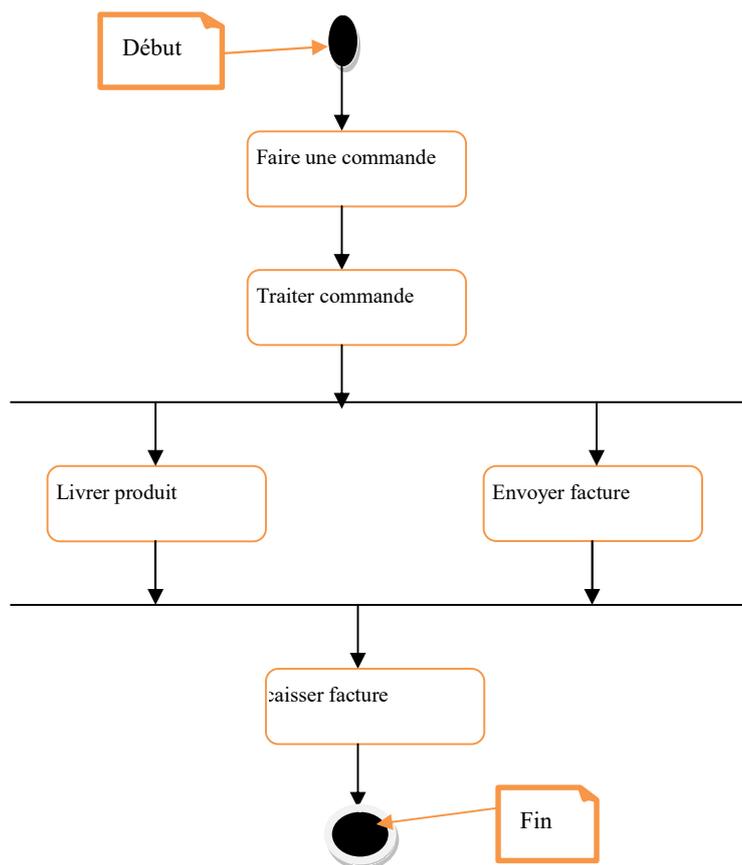


Figure 11 : Exemple de diagramme d'activité

2-6- Avantages et inconvénients des diagrammes d'activité :

Les principaux avantages des diagrammes d'activité sont les suivants :

- Ils sont bien adaptés pour représenter les comportements itératifs et conditionnels ;
- Ils sont bien adaptés pour modéliser la logique de contrôle ;
- Ils peuvent servir à illustrer le comportement dans plusieurs cas d'utilisation ;
- Ils montrent l'organisation séquentielle globale des activités de plusieurs objets et cas d'utilisation.

Les inconvénients des diagrammes d'activités sont :

- Ils sont mal adaptés pour représenter les interactions entre objets ;
- Bien qu'ils encouragent le parallélisme, ils ne règlent pas le problème d'accès concurrent aux données.

3- Diagramme de communication :

3-1- Définition :

Appelés diagramme de collaboration en UML, les diagrammes de collaboration décrivent l'interaction en mettant l'accent sur les liaisons des données entre les différents participants à l'interaction.

Exemple :

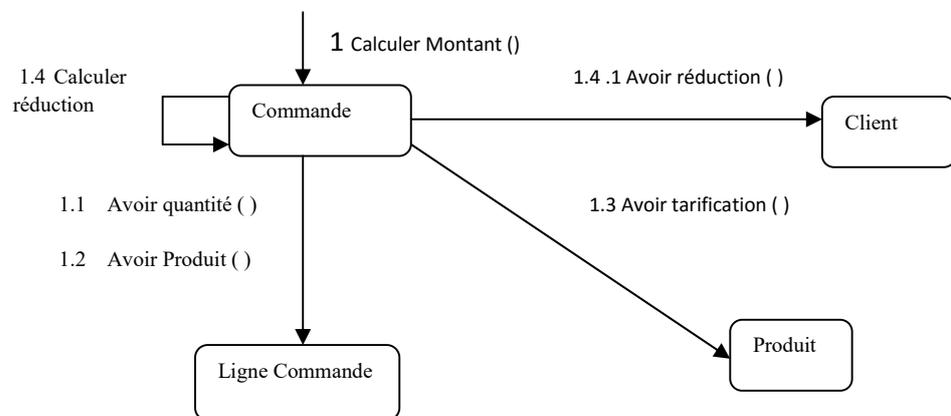


Figure 12 : Un exemple de diagramme de collaboration

Les diagrammes de collaboration montrent des interactions entre objets (instances de classes ou acteurs).

3-2- Synchronisation des messages :

Nous pouvons dans un diagramme de collaboration, préciser l'ordre et les conditions d'envoi des messages.

Pour chaque message, il est possible d'indiquer :

- Les clauses qui conditionnent son envoi ;
- Son rang (son numéro d'ordre par rapport aux autres messages) ;
- Son argument ou paramètre (optionnel) ;
- Nom du message.

Les messages sont numérotés par des chiffres séparés par des points. Ainsi, il est possible de représenter le niveau d'emboîtement des messages et leur précedence.

Exemple : Dans la figure précédente, l'envoi du message 1.2 suit immédiatement celui du message 1.1 et ces deux messages font partie du flot des messages 1.

3-3- Avantages et inconvénients :

Les avantages des diagrammes de collaboration sont les suivants :

- Ils sont bien adaptés comme les diagrammes de séquences pour représenter les interactions entre les objets ;
- Ils permettent de mettre en évidence les liens entre les objets ;
- Ils sont bien adaptés pour explorer les différentes alternatives.

Les inconvénients des diagrammes de collaboration sont comme suit :

- Mêmes inconvénient que ceux des diagrammes de séquences ;
- En plus, malgré la numérotation « hiérarchique » des messages, ils restent mal adaptés pour montrer le séquence ment des messages.