

## Chapitre 3 : UML (Unified Modeling Language)

### Partie 5 : Diagramme de séquence

#### 1. Introduction

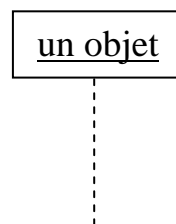
L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges.

Le diagramme de séquence **peut** être aussi utilisé pour documenter un cas d'utilisation. Les interactions entre objets représentent, dans ce cas, des flux d'informations échangés et non pas de véritables messages entre les opérations des objets. Sachant que, le diagramme de cas d'utilisation (voir partie 4) décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. L'acteur est dit principal pour un cas d'utilisation lorsque le cas d'utilisation rend service à cet acteur. Un acteur secondaire est sollicité pour des informations complémentaires.

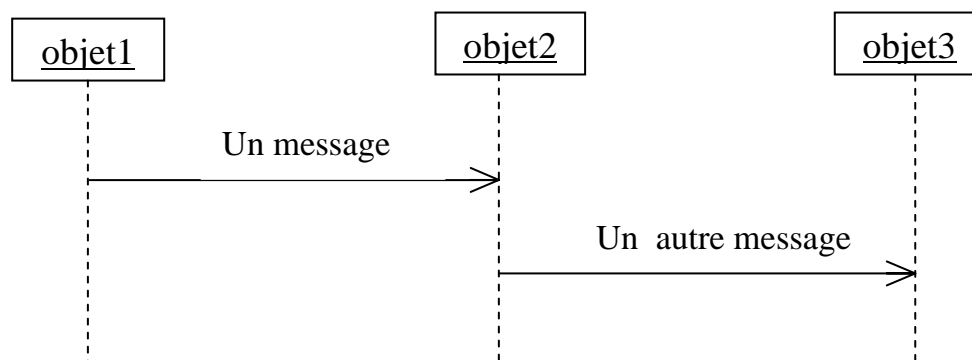
#### 1.1. La ligne de vie des objets

La ligne de vie des objets est représentée par une ligne verticale en traits pointillés, placée sous le symbole de l'objet concerné. Cette ligne de vie précise l'existence de l'objet concerné durant un certain laps du temps.

Notation graphique :



#### 1.2. Exemple de diagramme de séquence

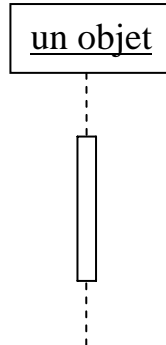


- L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule « de haut en bas » de cet axe.
- La disposition des objets sur l'axe horizontal n'a pas de conséquence pour la sémantique du diagramme.
- Les messages sont étiquetés par le nom de l'opération ou du signal invoqué.

### 1.3.Barre d'activation

Les diagrammes de séquence permettent de représenter les périodes d'activité des objets. Les périodes d'activité se représentent par des bandes rectangulaires placées sur les lignes de vie. Le début et la fin d'une bande correspondent respectivement au début et à la fin d'une période d'activité.




Notation graphique :



### 1.4.Message synchrone et asynchrone

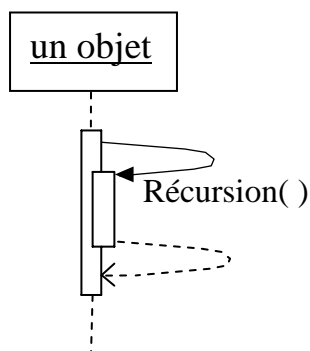
**Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. Le message de retour **peut** ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.

**Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations.

Messages	Notation graphique
Synchrone	
Asynchrone	
Retour	

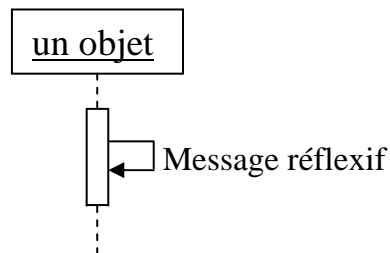
Remarque : Le cas particulier des envois de messages récursifs se représente par un dédoublement de la bande rectangulaire. L'objet apparaît alors comme s'il était actif plusieurs fois.

Exemple :



*Remarque :* Un objet peut également s'envoyer un message. Cette situation se représente par une flèche qui revient en boucle sur la ligne de vie de l'objet.

Exemple :



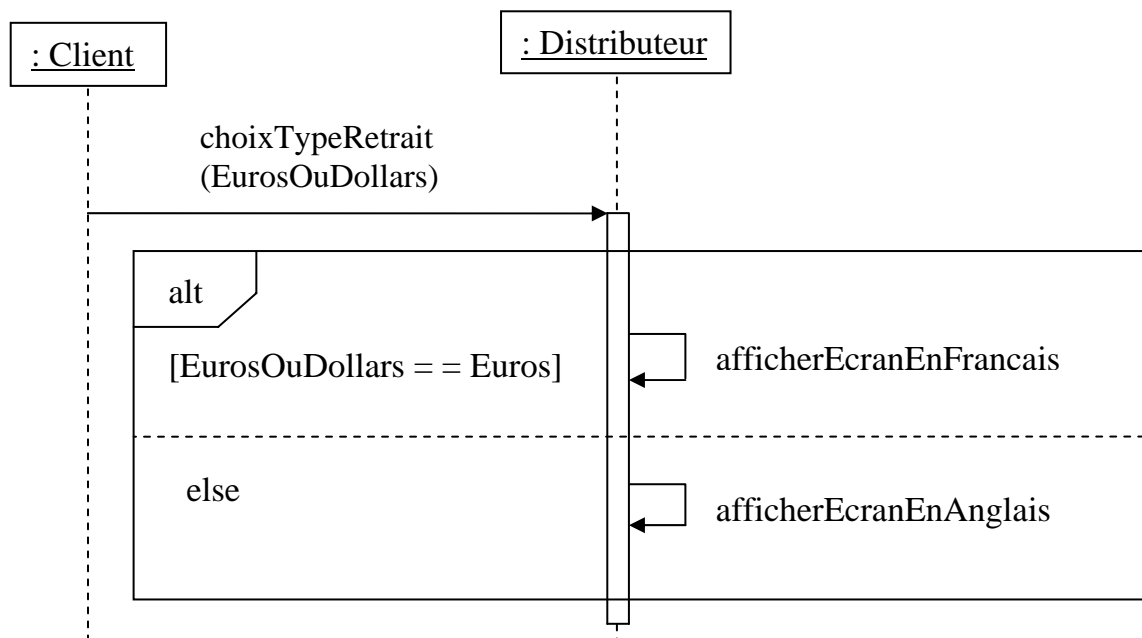
### 1.5.Fragment d'interaction

Un fragment d'interaction dit combiné correspond à un ensemble d'interaction auquel on applique un opérateur. Plusieurs opérateurs ont été définis dans UML : alt, opt, loop, par, etc.

- **Opérateur alt**

L'opérateur alt correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon (else). L'opérateur alt se représente dans un fragment possédant au moins deux parties séparées par des pointillés.

Exemple :

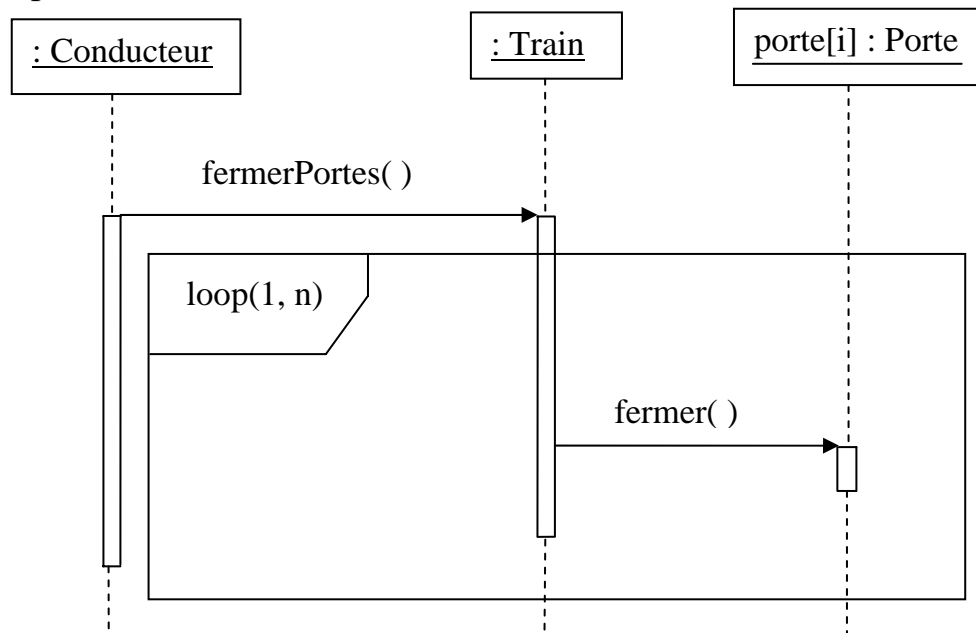


*Remarque :* L'opérateur **opt** correspond à une instruction de test sans alternative (sinon). L'opérateur **opt** se représente dans un fragment possédant une seule partie.

- **Opérateur loop**

L'opérateur loop correspond à une instruction de boucle.

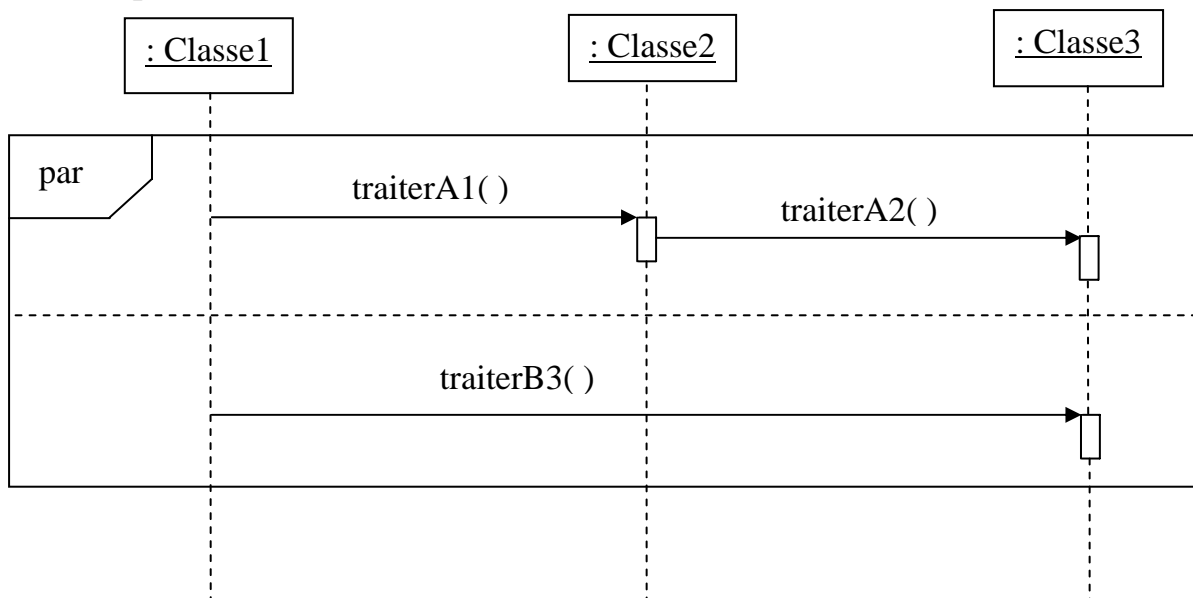
Exemple :



- **Opérateur par**

L'opérateur par (parallel) permet de représenter deux séries d'interactions qui se déroulent en parallèle.

Exemple :



Cet exemple montre que les traitements A1 et A2 sont menés en parallèle au traitement B3.