

DIAGRAMME DE SÉQUENCE

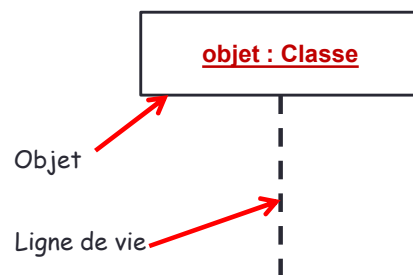
Chapitre 7

Introduction

- ❑ Les diagrammes de séquence font partie des diagrammes d'interaction
- ❑ Ils représentent les interactions entre objets en insistant sur la chronologie des envois de messages.

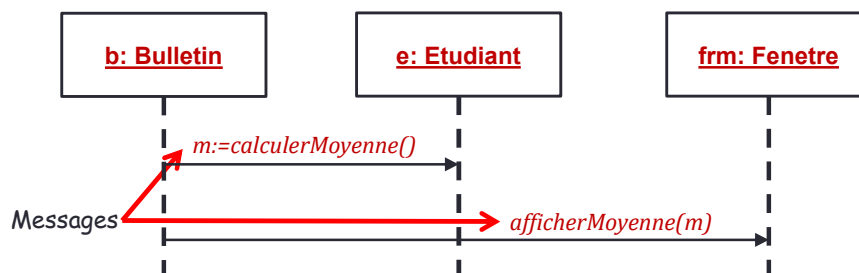
Représentation des objets

- Un objet est matérialisé par un rectangle et une barre verticale appelée ligne de vie des objets.



Représentation des interactions

- Les objets communiquent en échangeant des messages représentés au moyen de flèches horizontales, orientées de l'émetteur du message vers le destinataire.



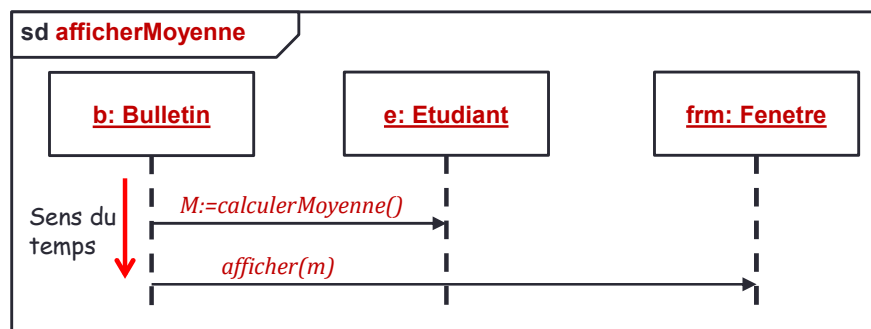
Représentation des interactions

❑ Remarque :

- L'axe vertical peut être gradué afin d'exprimer précisément les contraintes temporelles dans le cas de la modélisation d'un système temps réel par exemple.

Représentation du diagramme

- ❑ Un diagramme de séquence est représenté sous forme d'un cadre avec une étiquette « sd » suivi du nom du diagramme contenant les objets en interactions.

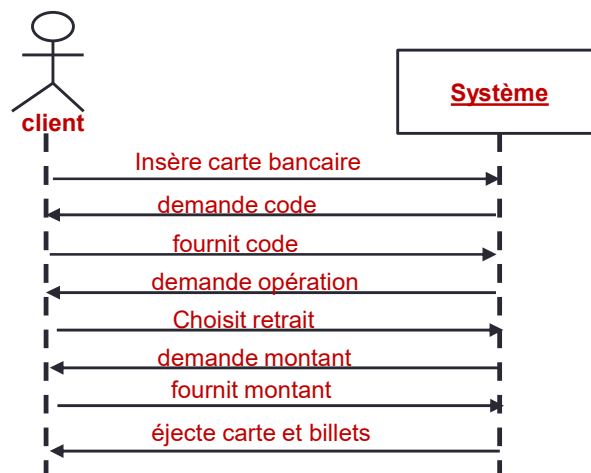


Utilisation des diagrammes de séquence

- ❑ En modélisation objet, les diagrammes de séquence s'utilisent de deux manières:
 1. Pour documenter les cas d'utilisation (boîte noire)
 - Description de l'interaction
 - Utilise les termes propres de l'utilisateur
 - N'entre pas dans les détails de la synchronisation
 - L'indication portée sur les flèches correspond à des événements qui surviennent dans le domaine de l'application.

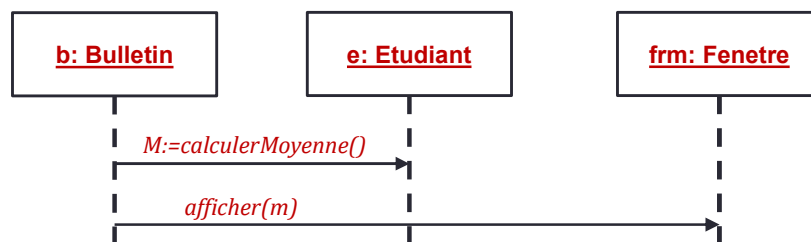
Utilisation des diagrammes de séquence

- ❑ Exemple :



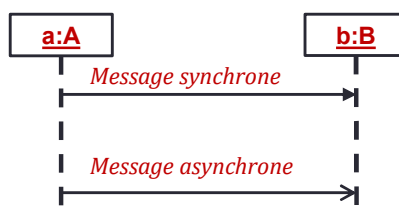
Utilisation des diagrammes de séquence

2. Usage plus informatique (boite blanche) qui permet la représentation précise des interactions entre objets:
 - Le concept message unifie toutes les formes de communication entre objets (appel de méthodes, événements, etc...)



Catégorie d'envoi des messages

- Un message peut être :
- Synchrones (\longrightarrow) : l'émetteur est bloqué jusqu'à réception de la réponse.
 - Asynchrones (\longrightarrow) : l'émetteur continue son exécution en attendant de recevoir la réponse.



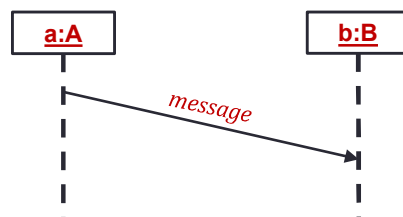
Catégorie d'envoi des messages

❑ Remarque :

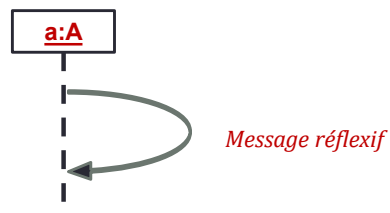
- Dans les message synchrone le retour est implicite (il n'est pas nécessaire de le représenter).
- Dans les messages asynchrones, le retour doit être représenté lorsqu'il arrive.

Représentation graphique de messages

❑ Message avec délai de transmission non négligeable



❑ Message réflexif :



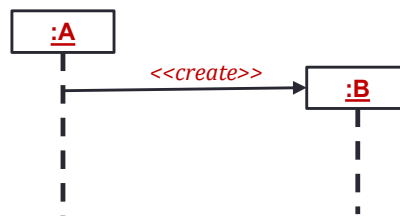
Représentation graphique de messages

❑ Remarque :

- Un message réflexif peut aussi indiquer une interaction interne dans un objet composite.

Création et destruction des objets

- ❑ La création des objets se représente en faisant pointer le message de création sur le rectangle qui symbolise l'objet créé.



Création et destruction des objets

- ❑ La destruction est indiquée par la fin de la ligne de vie et par la lettre X.
 - Soit à la hauteur du message qui cause la destruction
 - Soit après le dernier message envoyé par un objet qui se suicide.

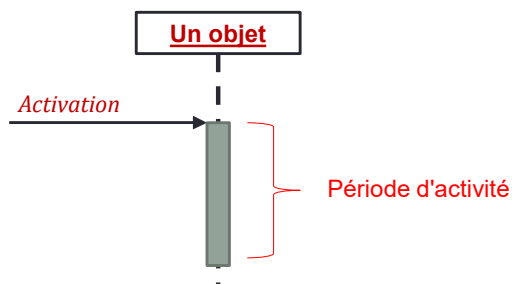


Activité d'un objet

- ❑ Les diagrammes de séquence permettent aussi de représenter les périodes d'activités des objets.
- ❑ Une période d'activité correspond au temps pendant lequel un objet effectue une action.
 - Soit directement
 - Soit par l'intermédiaire d'un autre objet qui lui sert de sous-traitant.

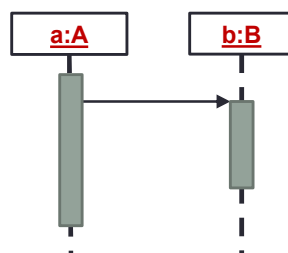
Activité d'un objet

- ❑ L'activité d'un objet est la période pendant laquelle, il effectue un traitement.
- ❑ Elle est représentée par une barre verticale sur la ligne de vie de l'objet.



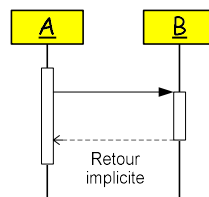
Activité d'un objet

- ❑ Dans l'exemple suivant l'objet A active un objet B (le flot d'exécution est passé de A à).
- ❑ A reste bloqué jusqu'à ce que B lui redonne la main.
- ❑ il reprend son exécution lorsque B termine son activité.



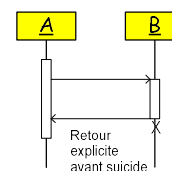
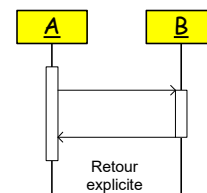
Activité d'un objet

- ❑ Dans le cas des envois synchrones, le retour en fin d'exécution de l'opération est implicite: il n'est pas nécessaire de le représenter dans les diagrammes.
- ❑ L'objet A reprend son exécution lorsque l'action déclenchée dans l'objet B est terminée.



Activité d'un objet

- ❑ Dans le cas des envois asynchrones, le retour doit être matérialisé lorsqu'il existe.
- ❑ Le diagramme suivant montre un objet B initialement activé par un objet A, qui retourne un message à l'objet A avant de cesser son exécution.

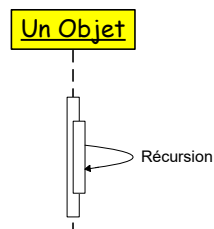


Remarque

- ❑ Il faut noter que la fin de l'activation d'un objet ne correspond pas à la fin de sa vie : un même objet peut être activé de nombreuses fois au cours de son existence.

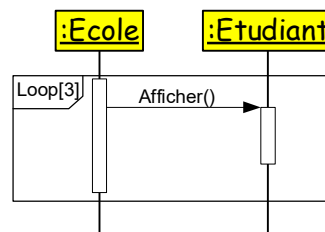
Activité d'un objet

- ❑ Le cas particulier des envois de messages récurrents se présente par un dédoublement de la bande rectangulaire.
- ❑ L'objet apparaît alors comme s'il était actif plusieurs fois.



Les fragments combinés

- ❑ Un fragment combiné représente les articulations d'une interaction.
- ❑ Il est défini par un **opérateur** et des **opérandes**.
- ❑ L'opérateur conditionne la signification du fragment combiné.
- ❑ Les fragments combinés peuvent concerner l'ensemble des entités participant au scénario ou juste un sous-ensemble.
- ❑ Exemple :



Les fragments combinés

- ❑ La liste complète de ces opérateurs est comme suit :
 - Choix et boucles : **alternative**, **option**, **break** et **loop**
 - Envoi en parallèle de messages : **parallel** et **critical region**
 - Envoi de messages : **ignore**, **consider**, **assertion** et **negative**
 - Ordre d'envoi des messages : **weak sequencing** , **strict sequencing**.

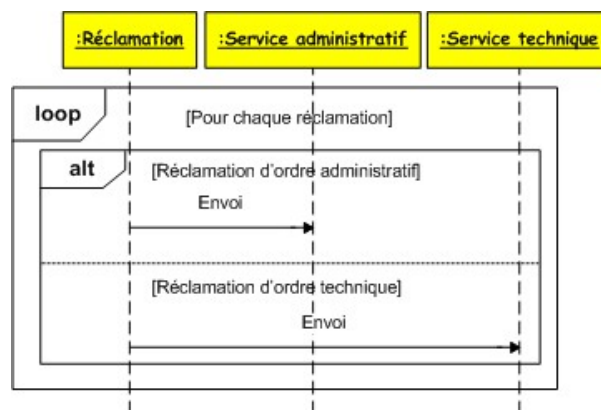
Les fragments combinés

❑ Opérateurs de choix et de boucle

- **Alternative** : noté « **alt** » est un opérateur conditionnel à choix multiple (équivalent à *SI...ALORS SINON...SI...ALORS... SINON ...etc*)
- **Option** : noté « **opt** » désigne un fragment combiné optionnel comme son nom l'indique (équivalent à *SI...ALORS*).
- **Break** : noté « **break** » représente des scénarios d'exception . Les interactions de ce fragment seront exécutées à la place des interactions décrites en dessous.
- **Loop** : noté « **loop** » permet de représenter une boucle.

Les fragments combinés

❑ Exemple :



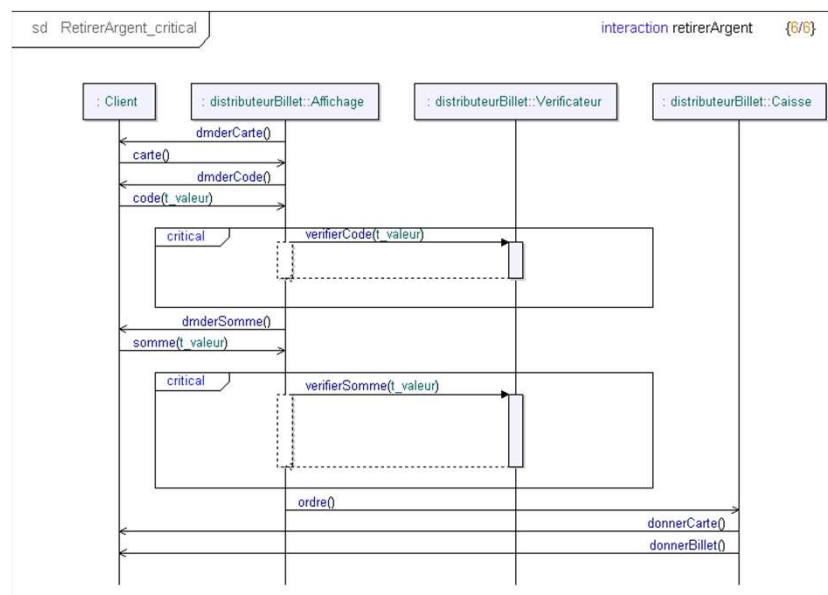
Les fragments combinés

❑ Opérateurs contrôlant l'envoi en parallèle de messages :

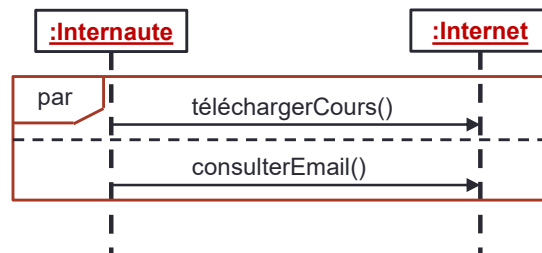
- **Parallel** : noté « **par** » possède au moins deux sous-fragments exécutés simultanément .
- **Critical** : noté « **critical** » désigne des interactions qui ne peuvent pas être interrompues par d'autres interactions décrites dans le diagramme.

❑ Exemple :

- voir diagramme suivant (retrait argent d'un GAB)



Exemple 2



Les fragments combinés

❑ Opérateurs contrôlant l'envoi de messages :

- **Ignore** : noté « **ignore** » désigne des interactions peu significantes.
- **Consider** : noté « **consider** » indique que ses interactions sont à prendre en compte dans la séquence.
- **Negative**: noté "**neg**" désigne un ensemble d'interactions invalides.
- **Assertion** : noté « **assert** » indique que la séquence décrite dans l'opérande désigne l'unique séquence possible. Toutes les autres séquences possibles sont des séquences invalides.

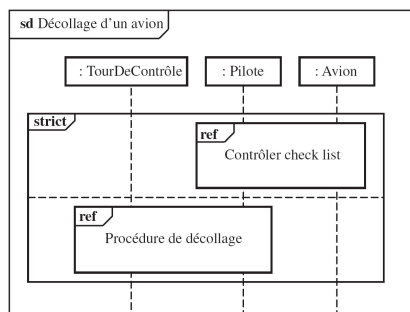
Les fragments combinés

❑ Opérateurs fixant l'ordre d'envoi des messages

- **Weak Sequencing** : noté « **seq** » notifiera que les interactions qui s'opèrent entre des entités indépendantes n'ont pas d'ordre particulier.
- **Strict Sequencing** : noté « **strict** » imposera l'ordre décrit sur le diagramme.

Les fragments combinés

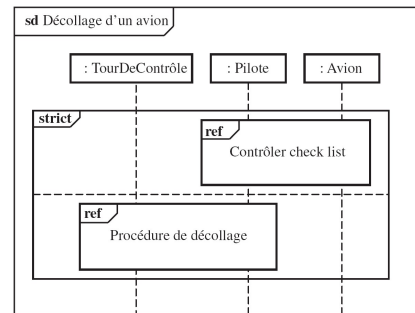
❑ Exemple :



Les fragments combinés

- ❑ Remarque :
 - il est possible de faire référence dans un diagramme de séquence à un autre diagramme de séquence en utilisant l'opérateur « **ref** »

Exemple :



Étude de cas

- ❑ Faire les scénarios « boîte blanche » de notre étude de cas en s'appuyant sur les diagrammes de séquences.