

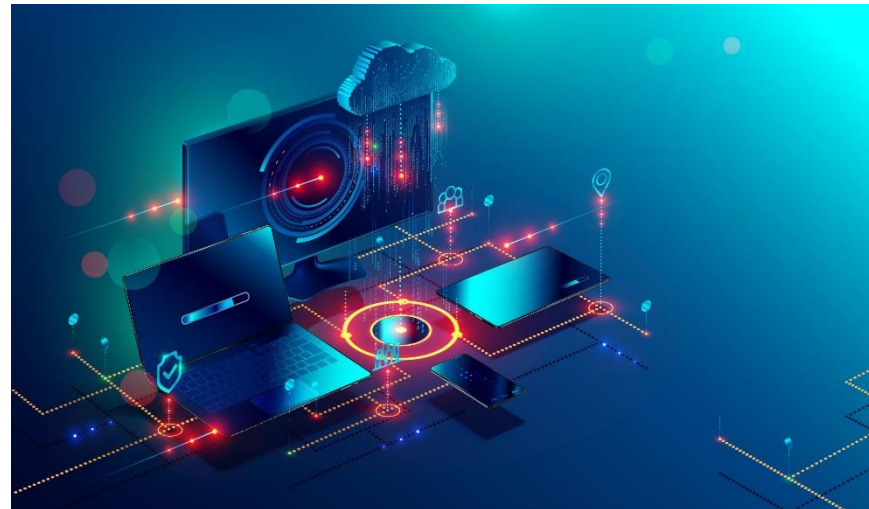
Analyse Orientée Objet

I. Introduction

II. Approche objet et système d'information.

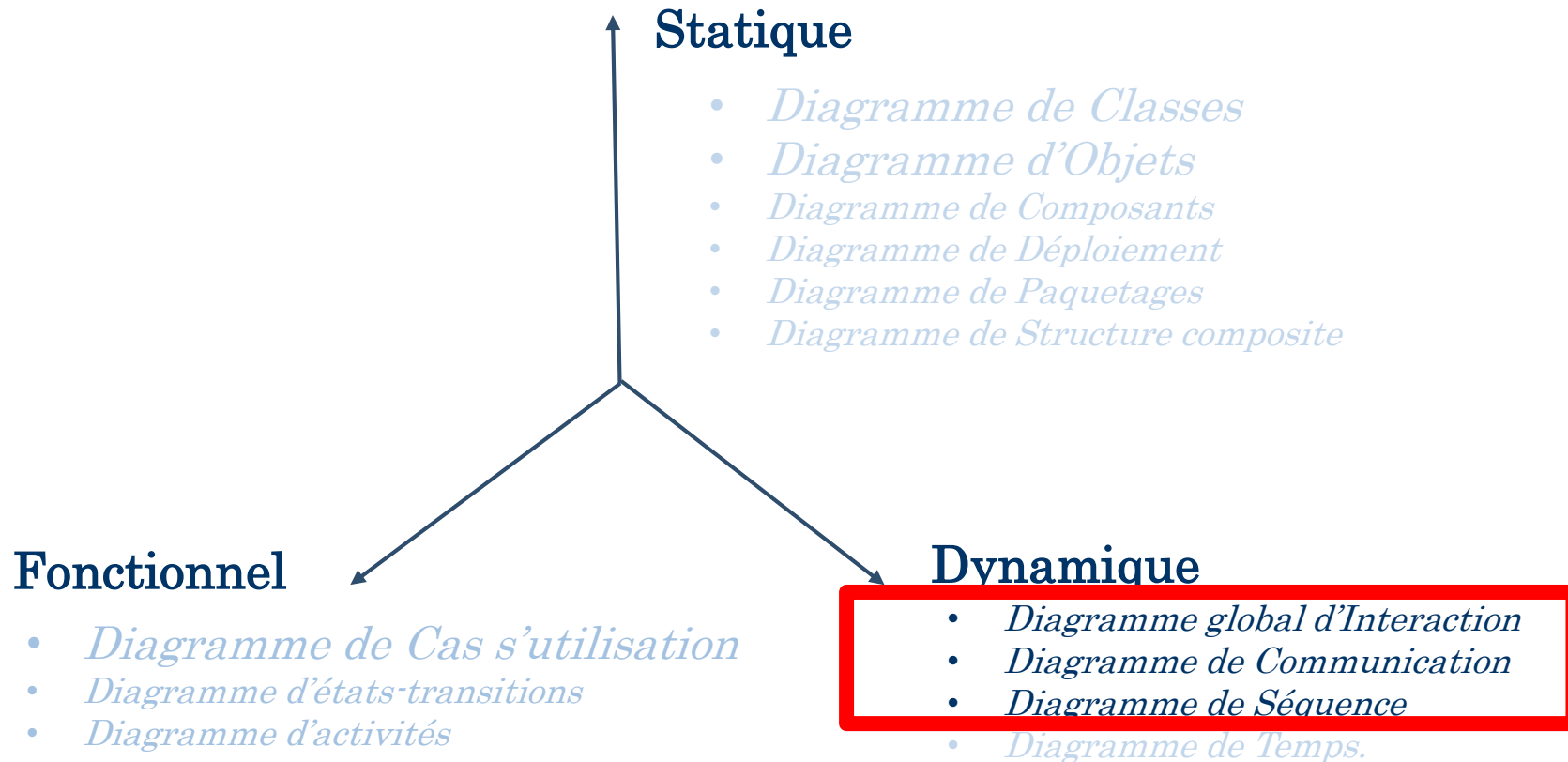
III. Principes Objet

IV. UML



IV. UML

3. Les diagrammes UML.



IV. UML

Diagramme d'interaction

1. Objectif.
2. Interactions et ligne de vie.
3. Diagrammes d'interaction.
 - 3.1 Diagramme de communication.
 - 3.2 Diagramme de séquence.
4. Exercices

IV. UML

Diagramme d'interaction

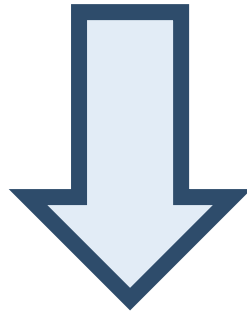
1. **Objectif.**
2. Interactions et ligne de vie.
3. Diagrammes d'interaction.
 - 3.1 Diagramme de communication.
 - 3.2 Diagramme de séquence.
4. Exercices

IV. UML

Diagramme d'interaction

1. Objectif.

- Un objet interagit avec d'autres objets pour implémenter un comportement



- Un diagramme d'interaction permet d'offrir une vue global du comportement d'objets

IV. UML

Diagramme d'interaction

1. Objectif.

On sait que:

- Le diagramme de **cas d'utilisation** montre les **acteurs** qui **interagissent** avec les grandes **fonctions** d'un système (vision **fonctionnelle** et externe d'un système)

ET

- Le diagramme de **classes**, décrit le cœur d'un système et montre des **classes** et la **façon dont elles sont associées**. (vision **statique** et structurelle).

IV. UML

Diagramme d'interaction

1. Objectif.

Définition:

Les diagrammes d'interaction permettent d'établir un pont entre ces deux approches (statique et fonctionnel). Ils montrent comment des instances au cœur du système communiquent pour réaliser une certaine fonctionnalité.

Aspect dynamique de la modélisation.

IV. UML

Diagramme d'interaction

1. Objectif.
2. Interactions et ligne de vie.
3. Diagrammes d'interaction.
 - 3.1 Diagramme de communication.
 - 3.2 Diagramme de séquence.
4. Exercices

IV. UML

Diagramme d'interaction

2. Interactions et ligne de vie.

Une interaction décrit le comportement d'un classeur en se focalisant sur l'échange d'informations (via des messages) entre les éléments du classeur.

Formalisme : L'interaction se représente par une flèche avec un texte décrivant le message.

IV. UML

Diagramme d'interaction

2. Interactions et ligne de vie.

De manière générale, une interaction est vue comme l'envoi d'un message entre un objet émetteur et un objet destinataire.

Représentation :



IV. UML

Diagramme d'interaction

2. Interactions et ligne de vie.

La ligne de vie précise l'existence de l'objet concerné durant un certain laps de temps.

Les lignes de vie sont toutes des participants à une interaction.

Formalisme : La ligne de vie se représente par une ligne verticale en traits pointillés placée sous l'objet concerné.

IV. UML

Diagramme d'interaction

2. Interactions et ligne de vie.

Enfin, une interaction est composée d'un jeu de lignes de vie où chacune des lignes de vie correspond à un participant.

Représentation :



IV. UML

Diagramme d'interaction

2. Interactions et ligne de vie.

Exemple :



IV. UML

Diagramme d'interaction

1. Objectif.
2. Interactions et ligne de vie.
3. **Diagrammes d'interaction.**
 - 3.1 Diagramme de communication.
 - 3.2 Diagramme de séquence.
4. Exercices

IV. UML

Diagramme d'interaction

3. Diagrammes d'interaction.

UML propose principalement deux diagrammes pour illustrer une interaction: le diagramme de communication et le diagramme de séquence.

Une même interaction peut-être représentée aussi bien par l'un que par l'autre.

Remarque importante:

Les diagrammes de **communication** et de **séquence** peuvent représenter la **même interaction**.
MAIS

Diagramme de communication montre une **représentation spatiale** des lignes de vie.

Diagramme de séquence met l'accent sur les **séquencement temporel** de **messages échangés** entre les lignes de vie

IV. UML

Diagramme d'interaction

1. Objectif.
2. Interactions et ligne de vie.
3. **Diagrammes d'interaction.**
 - 3.1 **Diagramme de communication.**
 - 3.2 Diagramme de séquence.
4. Exercices

IV. UML

Diagramme d'interaction

3.1 Diagramme de communication (collaboration).

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de communication.
3. Exemple de diagramme de communication.

IV. UML

Diagramme d'interaction

3.1 Diagramme de communication (collaboration).

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de communication.
3. Exemple de diagramme de communication

IV. UML

Diagramme d'interaction

3.1 Diagramme de communication (collaboration).

Définition:

Les diagrammes **de communication** permettent de mettre en évidence les interactions entre les différents objets du système.

Il sera utilisé pour:

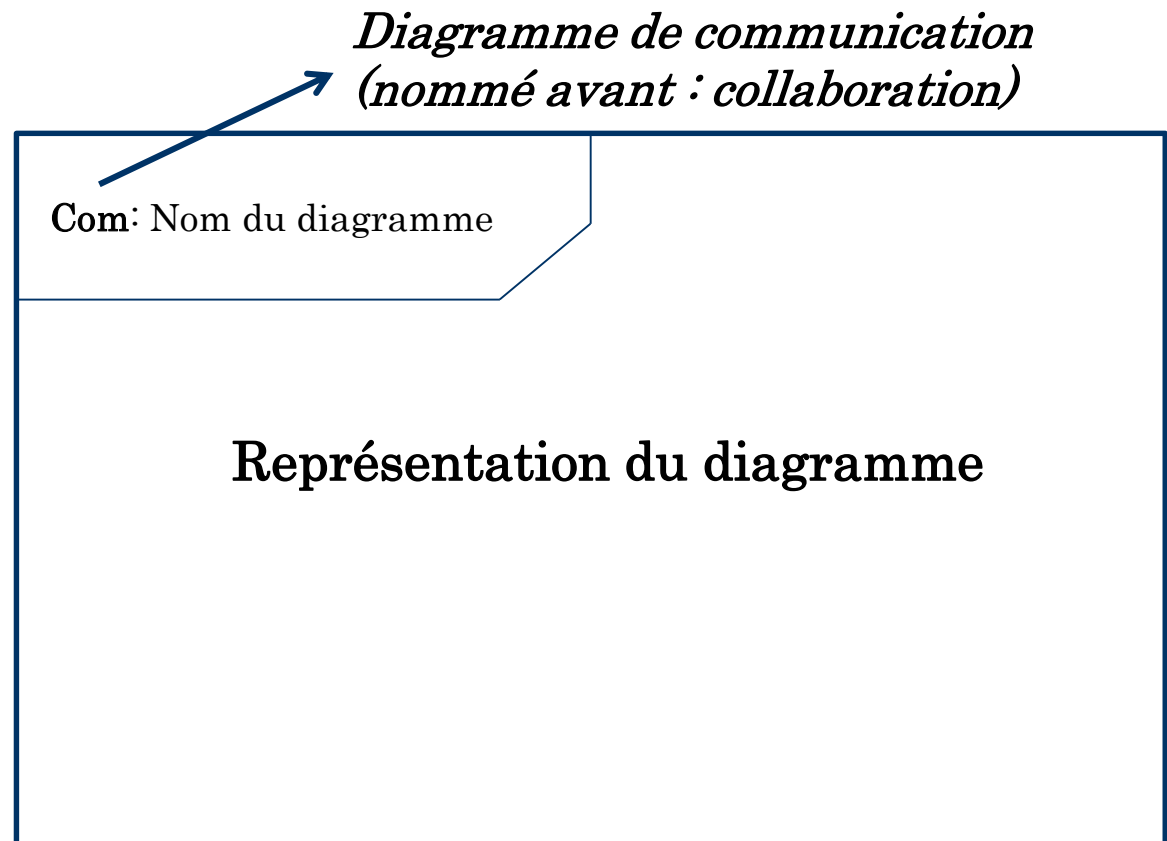
- **Préciser** le contexte dans lequel l'objet évolue,
- **Mettre** en évidence les **dépendances** entre les **différents** objets impliqués dans l'exécution d'un processus ou d'un cas d'utilisation.

IV. UML

Diagramme d'interaction

3.1 Diagramme de communication (collaboration).

Formalisme:



IV. UML

Diagramme d'interaction

3.1 Diagrammes de communication.

Le diagramme de communication est souvent utilisé pour **illustrer un cas d'utilisation** ou pour **décrire une opération**.

Il est aussi une **aide pour valider les associations** du **diagramme de classes** en les utilisant comme support de transmission des messages.

IV. UML

Diagramme d'interaction

3.1 Diagramme de communication (collaboration).

1. Définition.
2. Les éléments constitutifs d'un diagramme de communication.
3. Exemple de diagramme de communication

IV. UML

Diagramme d'interaction

3.1 Diagrammes de communication.

2. Les éléments constitutifs d'un diagramme de communication:
 1. Les lignes de vie.
 2. Les connecteurs.
 3. Les messages.

IV. UML

Diagramme d'interaction

3.1 Diagrammes de communication.

2. Les éléments constitutifs d'un diagramme de communication:
 1. Les lignes de vie.
 2. Les connecteurs.
 3. Les messages.

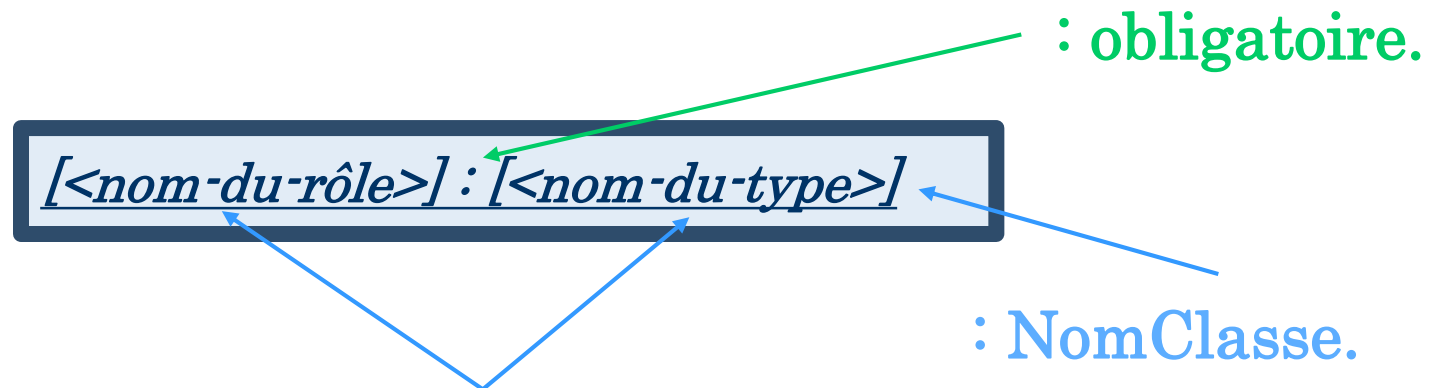
IV. UML

Diagramme d'interaction

2. Les éléments constitutifs d'un diagramme de communication:

1. Les lignes de vie.

Les lignes de vie sont représentées par des rectangles contenant une étiquette dont la syntaxe est :



Au moins un des deux noms doit être spécifié

IV. UML

Diagramme d'interaction

3.1 Diagrammes de communication.

2. Les éléments constitutifs d'un diagramme de communication:
 1. Les lignes de vie.
 2. Les connecteurs.
 3. Les messages.

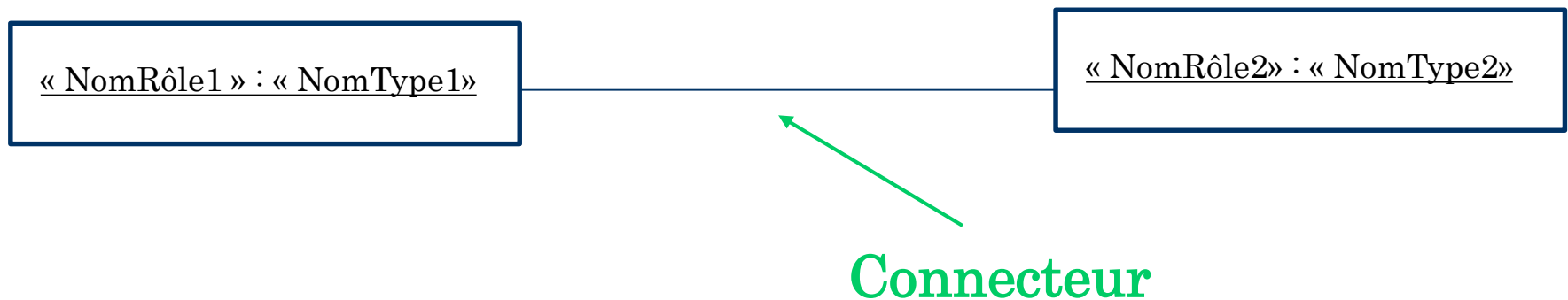
IV. UML

Diagramme d'interaction

2. Les éléments constitutifs d'un diagramme de communication:

2. Les connecteurs.

Les connecteurs sont les relations entre les lignes de vie et se représentent par un trait continu reliant deux lignes de vie et dont les extrémités peuvent être ornées de multiplicités.



IV. UML

Diagramme d'interaction

3.1 Diagrammes de communication.

2. Les éléments constitutifs d'un diagramme de communication:
 1. Les lignes de vie.
 2. Les connecteurs.
 3. Les messages.

IV. UML

Diagramme d'interaction

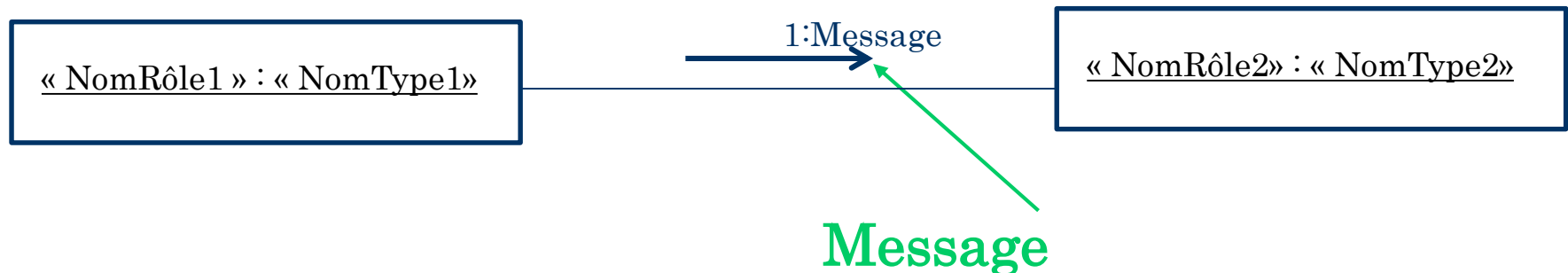
2. Les éléments constitutifs d'un diagramme de communication:

3. Les messages.

Les messages définissent une **communication particulière** entre les lignes de vie.

Ils sont **ordonnés** selon un numéro de séquence croissant.

Le message est représenté par une flèche pointant vers l'objet destinataire.



IV. UML

Diagramme d'interaction

2. Les messages.

```
[ '['<cond>']' [<séq>] [ *['|'] ] ['['<iter>']'] ] : [<var> :=] <msg>(['<par>'])
```

<cond>: est une condition sous forme d'expression booléenne entre crochets.

<séq> : est le numéro de séquence du message. Les messages sont numérotés par envoi et sous-envoi désignés par des chiffres séparés par des points.

<iter> : spécifie entre crochets, l'envoi séquentiel (*['<iter>'] ou parallèle (['|']*['<iter>'])) de plusieurs messages.

La spécification de ['<iter>'] peut être ignorée, la seule présence de * ou de |* permet d'indiquer un message récurrent sans plus de précision.

<var>: est la valeur de retour du message, qui sera, par exemple transmise en paramètre à un autre message.

<msg>: est le nom du message.

<par>: désigne des paramètres (optionnels) du message.

IV. UML

Diagramme d'interaction

2. Les messages.

```
[ '['<cond>' ] [<séq>] [ * [ | ] '['<iter>' ] ] : [ <var> := ] <msg>([<par>])
```



La syntaxe permet de préciser l'ordonnancement et la synchronisation des messages entre objets du diagramme.

IV. UML

Diagramme d'interaction

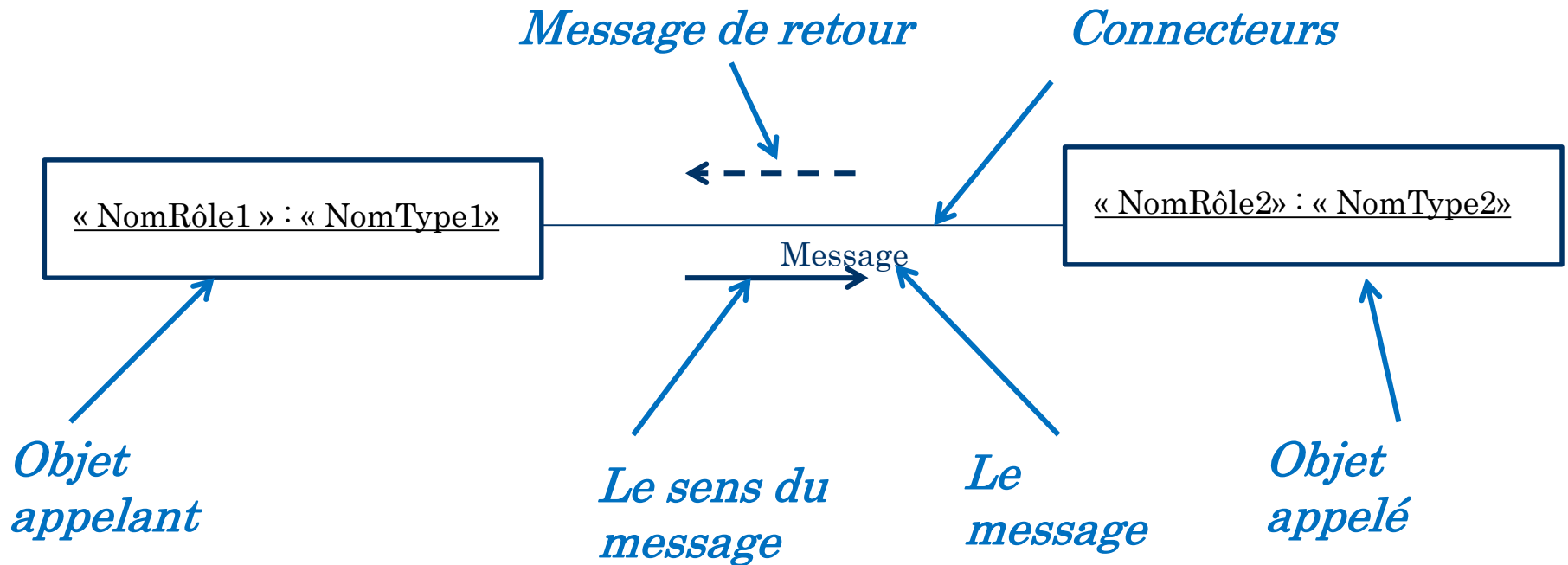
3.1 Diagramme de communication (collaboration).

1. Définition.
2. Les éléments constitutifs d'un diagramme de communication.
3. Exemple de diagramme de communication

IV. UML

Diagramme d'interaction

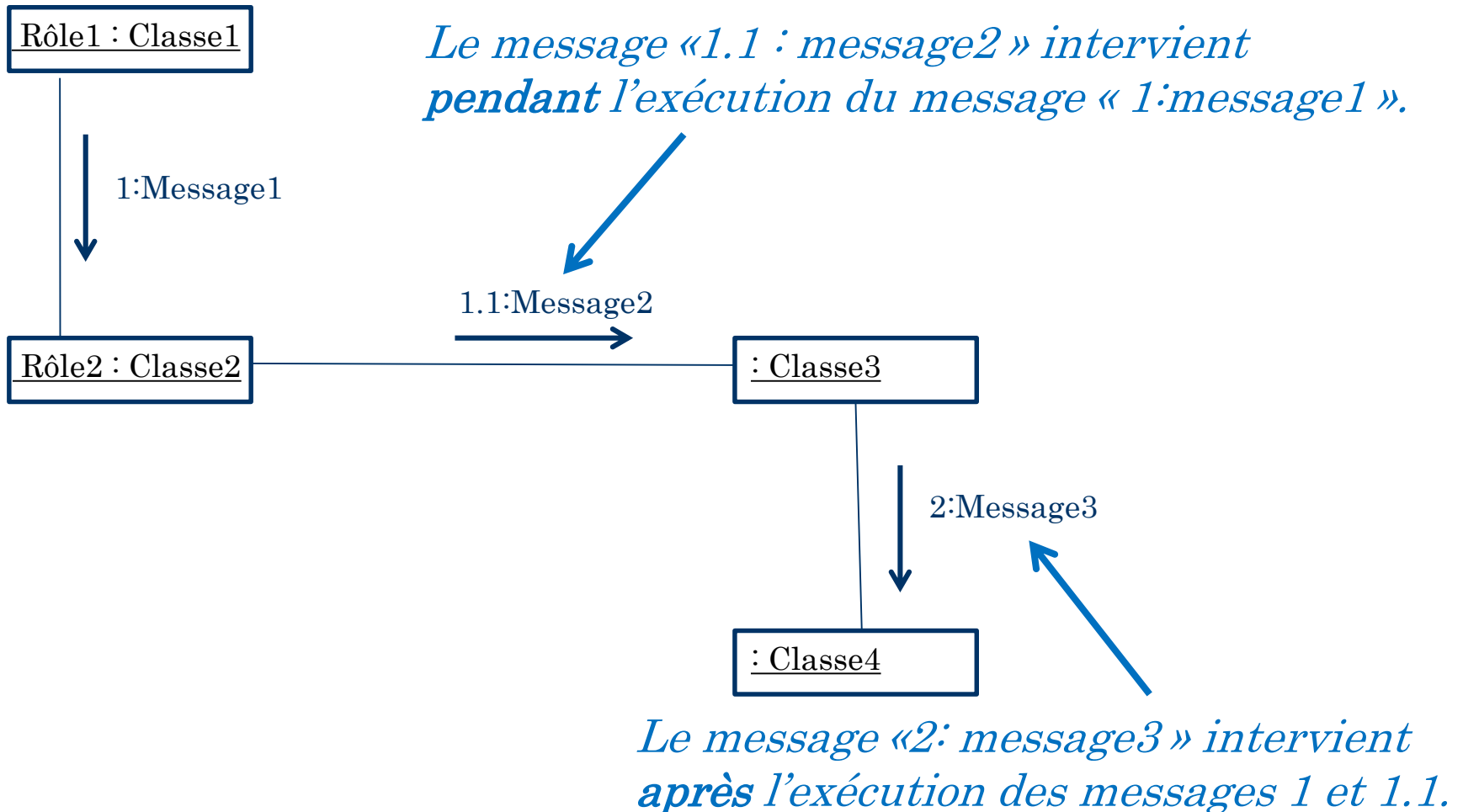
3. Exemple de diagramme de communication



IV. UML

Diagramme d'interaction

3. Exemple de diagramme de communication

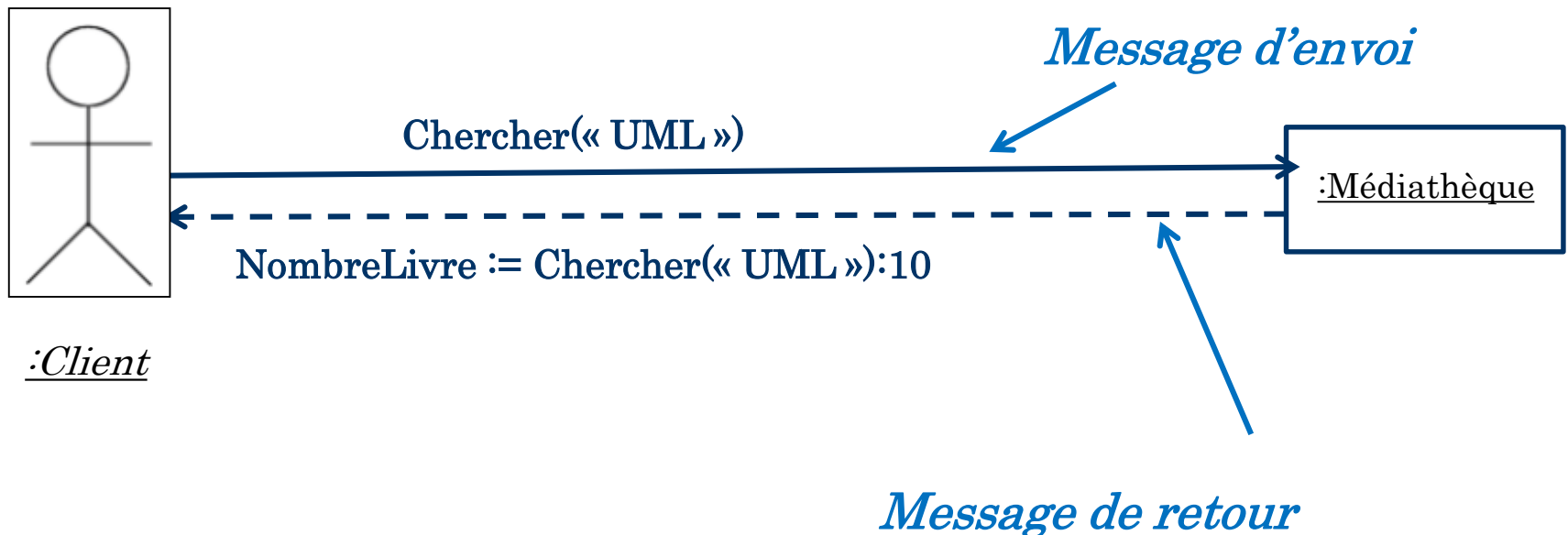


IV. UML

Diagramme d'interaction

3. Exemple de diagramme de communication

Résultat:= message(arguments)[:valeur retour]



IV. UML

Diagramme d'interaction

3. Exemple de diagramme de communication

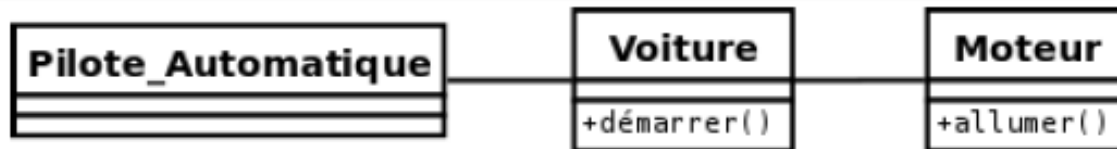


Diagramme de classes

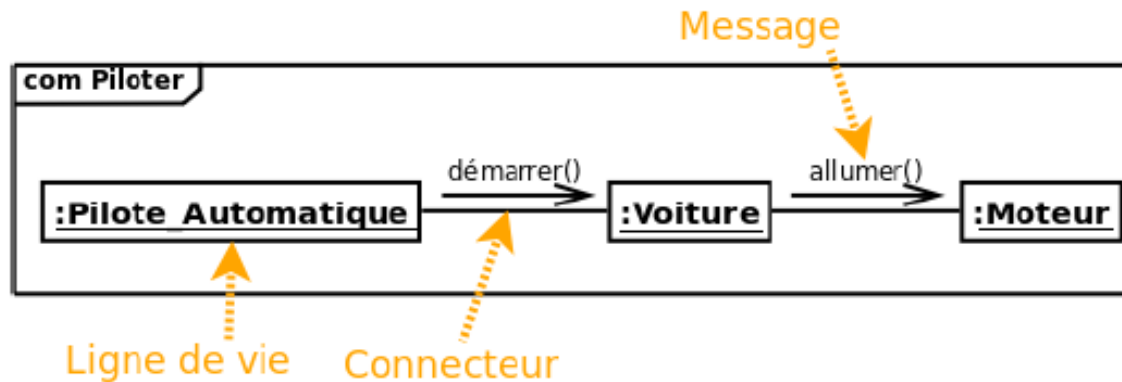


Diagramme de communication : focus sur **organisation structurelle** des objets qui communiquent.

- Une **interaction** se focalise sur l'échange d'informations entre les participants.
- Elle est composée de **lignes de vie**
- Elle spécifie les **messages échangés** entre participants

IV. UML

Diagramme d'interaction

1. Objectif.
2. Interactions et ligne de vie.
3. Diagrammes d'interaction.
 - 3.1 Diagramme de communication.
 - 3.2 Diagramme de séquence.**
4. Exercices

IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence:

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de séquence.
3. Exemple de diagramme de séquence.
4. Fragments d'interaction.

IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence:

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de séquence.
3. Exemple de diagramme de séquence.
4. Fragments d'interaction.

IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence

Définition:

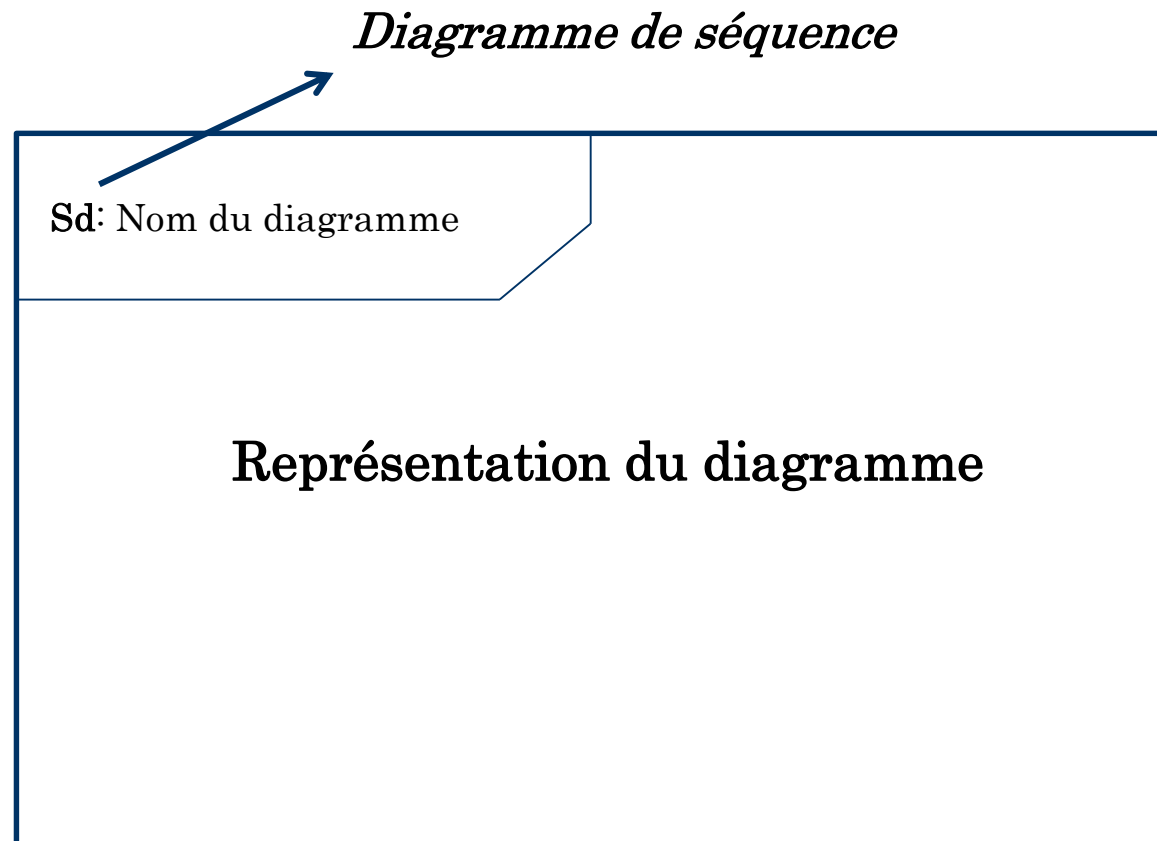
Les diagrammes de séquence décrivent le **déroulement de chaque cas d'utilisation**, en montrant la façon dont les diverses entités mises en œuvre dans les cas interagissent et collaborent dans le **temps** afin de réaliser les **fonctionnalités** attendues.

IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence

Formalisme:



IV. UML

Diagramme d'interaction

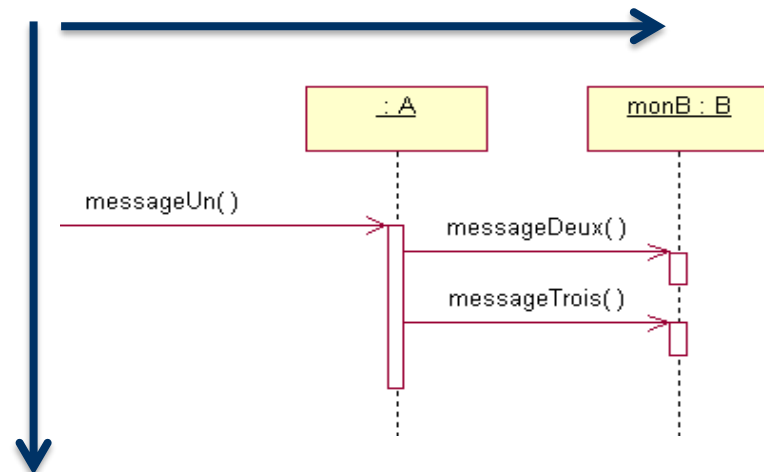
3.2 Diagrammes de séquence.

Le diagramme de **séquence** permet de visualiser les **messages** par une lecture de **haut en bas**.

DEUX DIMENSIONS

1.dimension horizontale :
les OBJETS, les ACTEURS et le SYSTEME
(ordre sans importance)

2.dimension verticale :
LE TEMPS (s'écoule de haut en bas)



IV. UML

Diagramme d'interaction

3.2 Diagrammes de séquence.

Le diagramme de **séquence** est utilisé pour **déterminer** :

- Les **divers objets**, mises en jeu dans la réalisation d'une fonctionnalité.
- Les **interactions** entre ces divers objets.
- Le **déroulement** dans le **temps** de ces interactions.

IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence:

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de séquence.
3. Exemple de diagramme de séquence.
4. Fragments d'interaction.

IV. UML

Diagramme d'interaction

2. Les éléments constitutifs d'un diagramme de séquence.
 1. Les objets et ligne de vie.
 2. Les bandes d'activation.
 3. Les messages.

IV. UML

Diagramme d'interaction

2. Les éléments constitutifs d'un diagramme de séquence.
 1. Les objets et ligne de vie.
 2. Les bandes d'activation.
 3. Les messages.

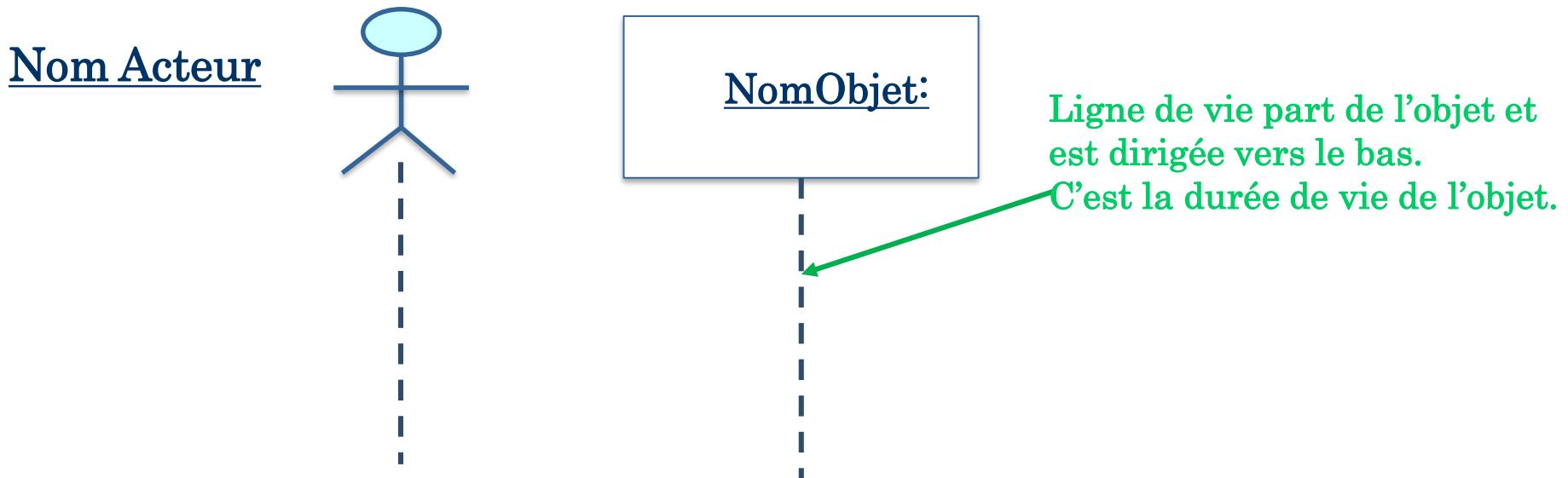
IV. UML

Diagramme d'interaction

1. Les objets et ligne de vie.

Un **objet** mis en œuvre dans un diagramme de séquence peut symboliser :

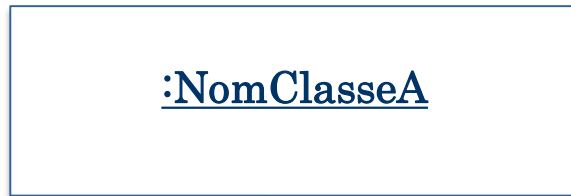
- Un **acteur** humain ou non-humain
- Le **système** ou une partie du système.



IV. UML

Diagramme d'interaction

1. Les objets et ligne de vie.



Ligne de vie d'un objet anonyme de la classe A



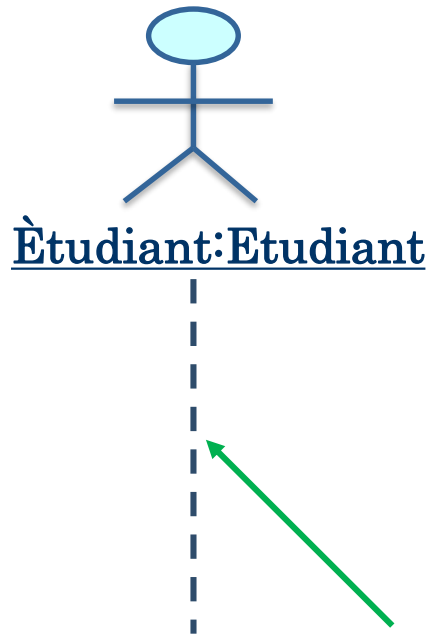
Ligne de vie de l'objet 1 de la classe A

IV. UML

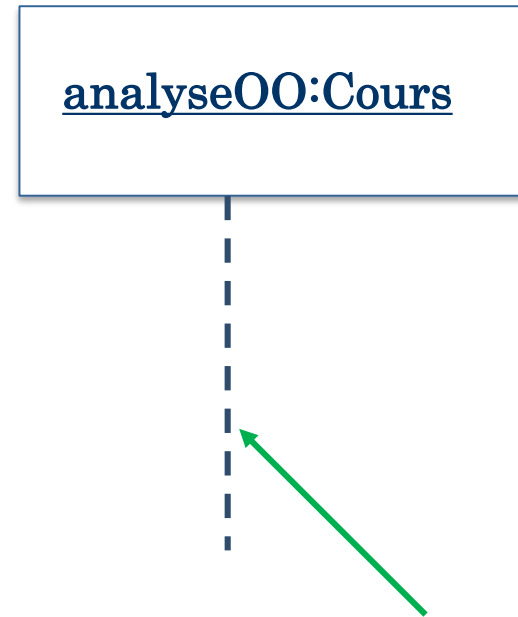
Diagramme d'interaction

1. Les objets et ligne de vie.

Exemple:



Ligne de vie de l'acteur
étudiant.



Ligne de vie de l'objet
analyseOO

IV. UML

Diagramme d'interaction

2. Les éléments constitutifs d'un diagramme de séquence.
 1. Les objets et ligne de vie.
 2. Les bandes d'activation.
 3. Les messages.

IV. UML

Diagramme d'interaction

2. Les bandes d'activation.

Les diagrammes de séquence permettent 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- traitement.

IV. UML

Diagramme d'interaction

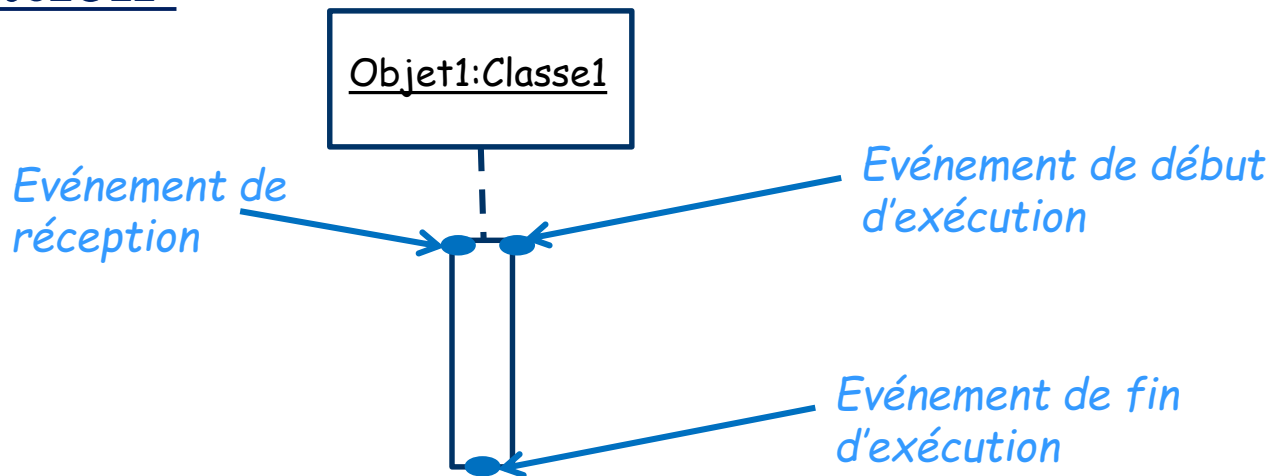
2. Les bandes d'activation.

Formalisme:

Les **bandes d'activation** (périodes d'activités) se représentent par des **bandes rectangulaires** placées sur la ligne de vie des objets.

Le code de l'objet est alors exécuté

Représentation:



IV. UML

Diagramme d'interaction

2. Les éléments constitutifs d'un diagramme de séquence.
 1. Les objets et ligne de vie.
 2. Les bandes d'activation.
 3. Les messages.

IV. UML

Diagramme d'interaction

3. Les messages.

1. Définition.
2. Catégories de message.

IV. UML

Diagramme d'interaction

3. Les messages.

1. Définition.
2. Catégories de message.

IV. UML

Diagramme d'interaction

3. Les messages.

Définition:

Un message définit une **communication particulière** entre les lignes de vie.

Un message en UML correspond à un appel de méthodes en programmation.

En plus, les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie.

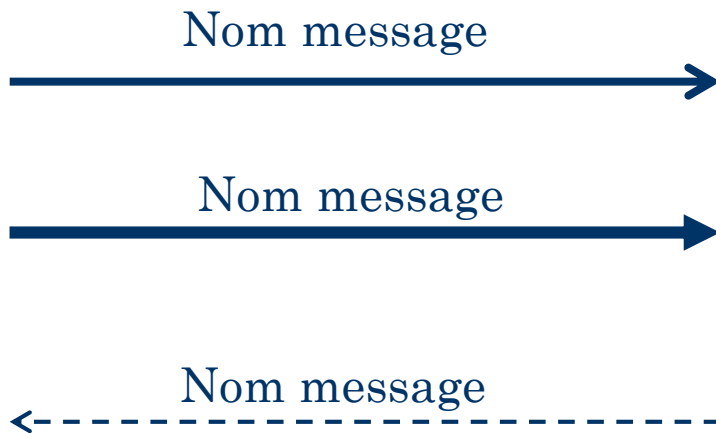
IV. UML

Diagramme d'interaction

3. Les messages.

Graphiquement, le message est représenté par une flèche accompagnée par un nom.

Il existe plusieurs types de message



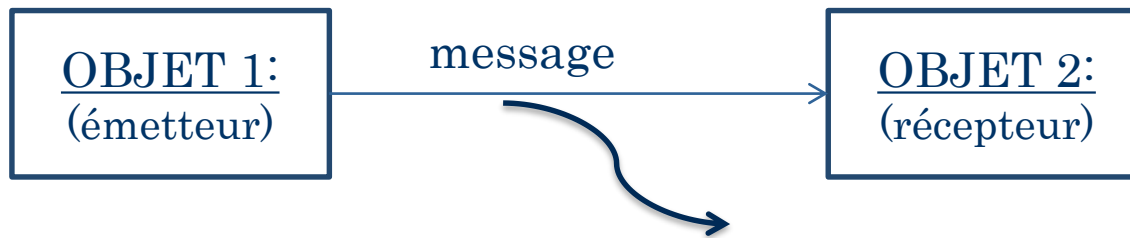
Et plusieurs représentations graphiques.

IV. UML

Diagramme d'interaction

3. Les messages.

Représentation



Flèche pleine et continue

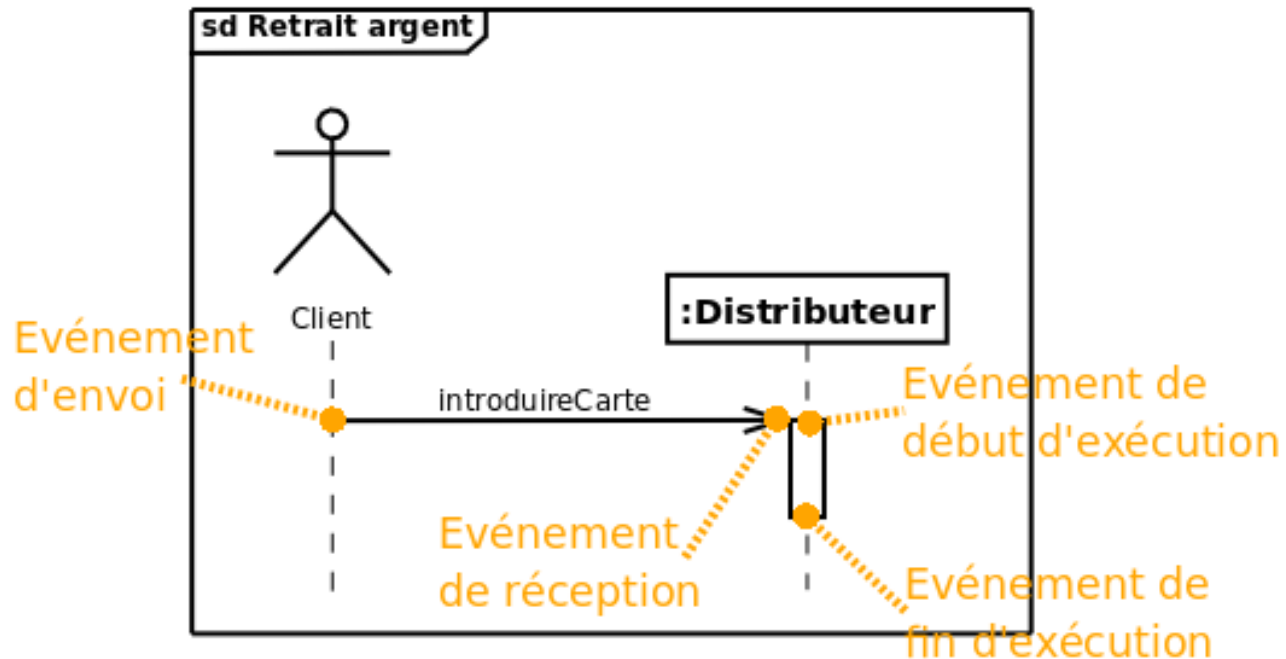
Remarque importante:

Comme vu dans les diagrammes de **communication**, les messages seront numérotés de façon à refléter l'imbrication des envois de messages.

IV. UML

Diagramme d'interaction

3. Les messages.



Comme le montre le schéma ci-dessus, UML permet de **séparer** clairement l'*envoi du message*, sa *réception*, ainsi que le *début* de l'*exécution* de la *réaction* et sa *fin*.

IV. UML

Diagramme d'interaction

3. Les messages.

Syntaxe:

```
[<attribut> = ] message [ : <valeur_de_retour>]
```

Message: représente le message d'envoi.

IV. UML

Diagramme d'interaction

3. Les messages.

1. Définition.
2. Catégories de message.

IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message d'un objet à lui-même.
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message d'un objet à lui-même.
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

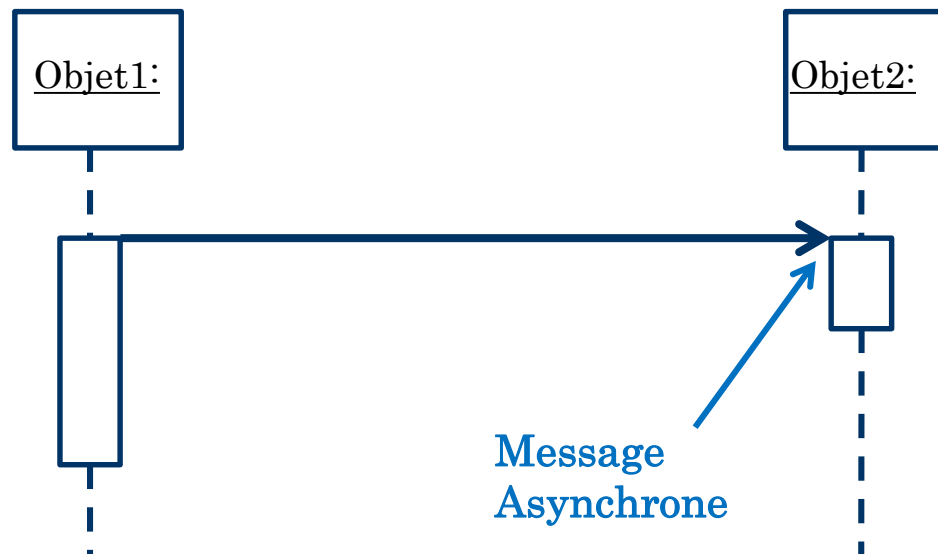
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message asynchrone.

*L'émetteur du message **n'attend pas** de réponse à son message, il poursuit l'exécution de ses opérations.*



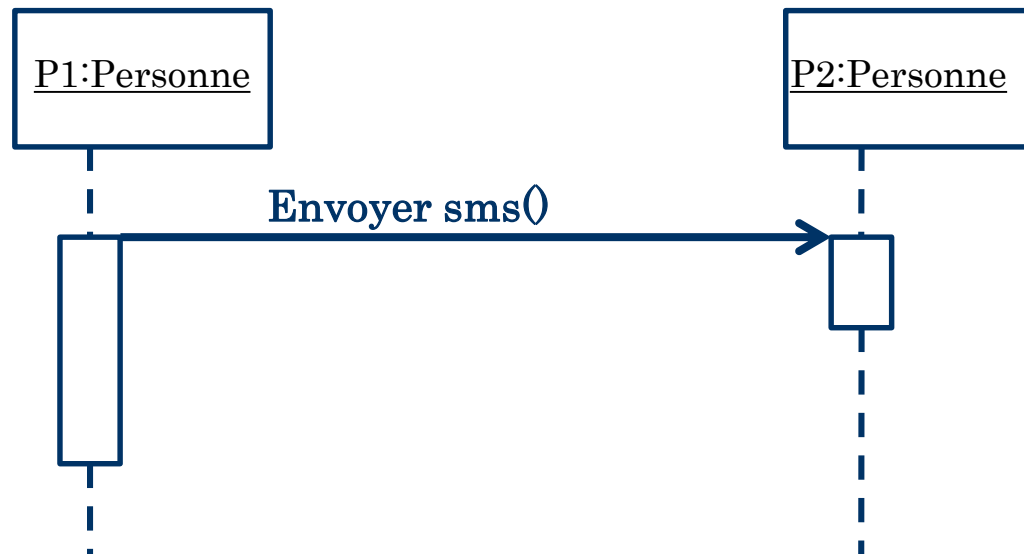
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message asynchrone.

*L'émetteur du message **n'attend pas** de réponse à son message, il poursuit l'exécution de ses opérations.*



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- **Message synchrone.**
- Message retour (avec résultats et arguments).
- Message d'un objet à lui-même.
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

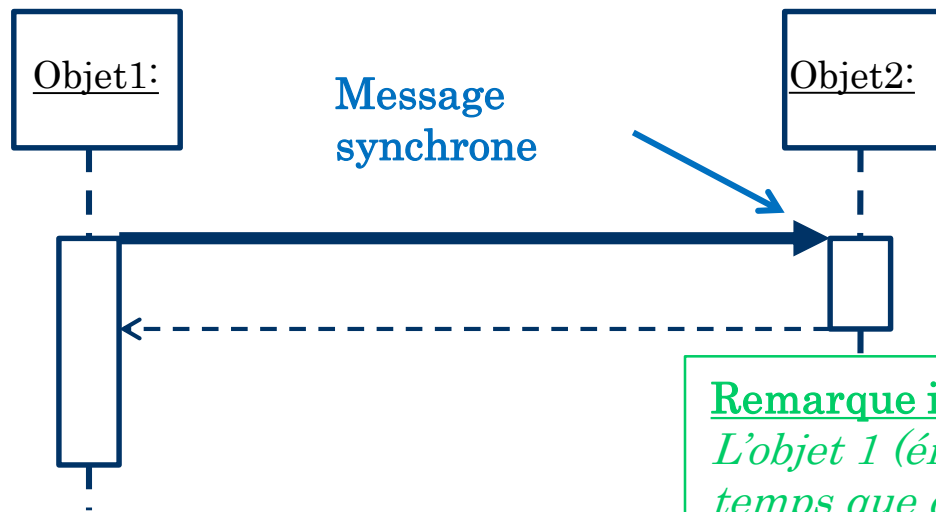
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message synchrone.

*L'émetteur du message **attend** la réponse à son message avant de **poursuivre** ses actions.*



Remarque importante:

*L'objet 1 (émetteur) est **bloqué** le temps que dure l'exécution et jusqu'à **recevoir le résultat** de l'objet 2.*

IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- **Message retour** (avec résultats et arguments).
- Message d'un objet à lui-même.
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

IV. UML

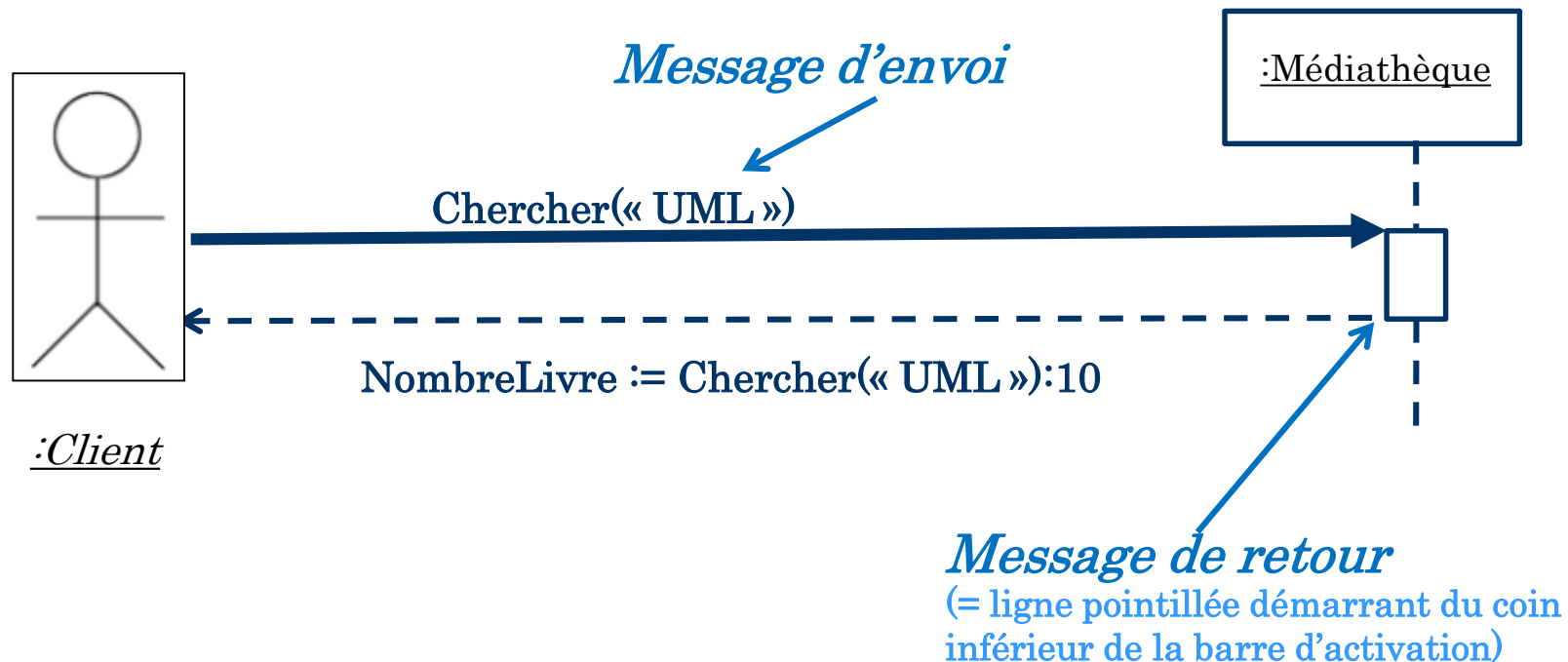
Diagramme d'interaction

2. Catégories de message.

- Message retour (avec résultats et arguments).

Syntaxe:

Résultat := message(arguments)[:valeur retour]



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- **Message d'un objet à lui-même.**
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

IV. UML

Diagramme d'interaction

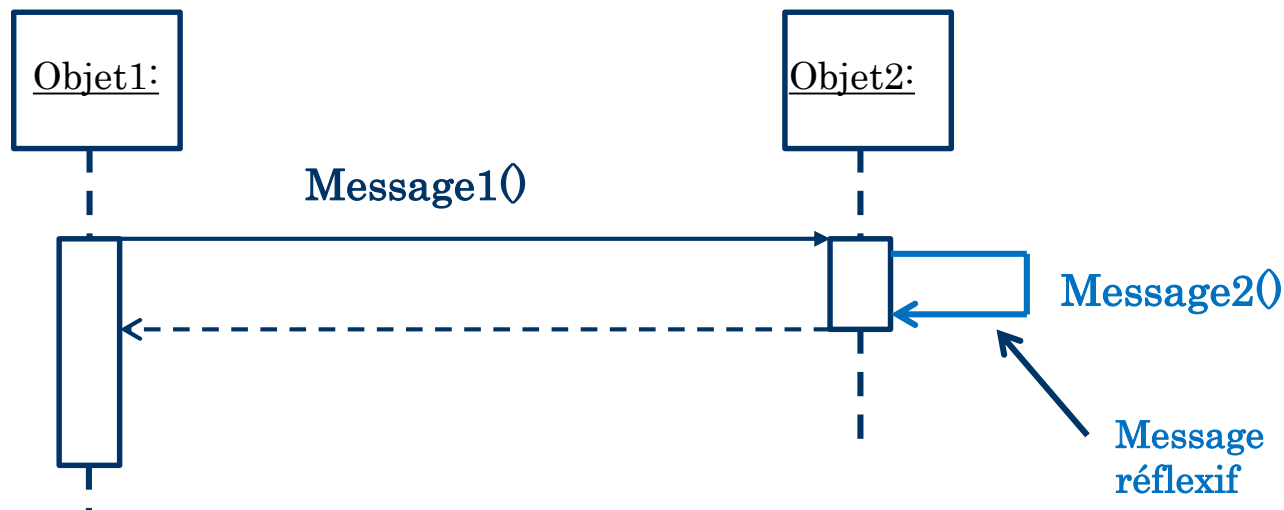
2. Catégories de message.

➤ Message d'un objet à lui même

Un objet peut s'envoyer un message.

Message réflexif peut servir pour indiquer une activité de plus bas niveau qui s'exerce au sein de l'objet.

*Message réflexif représente les **interactions internes** entre objets.*

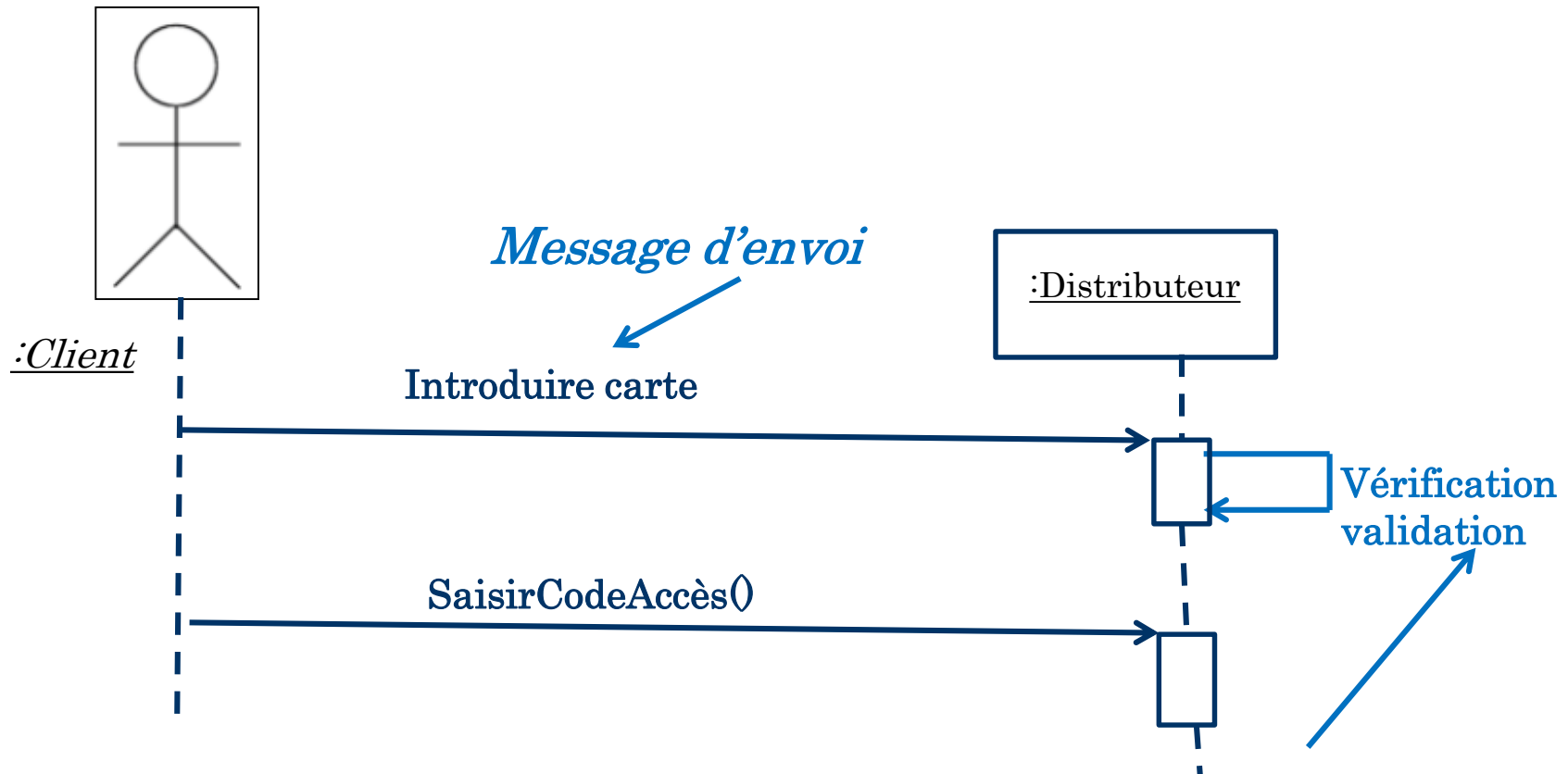


IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message d'un objet à lui même



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message d'un objet à lui-même.
- **Message de création/destruction d'un objet.**
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

IV. UML

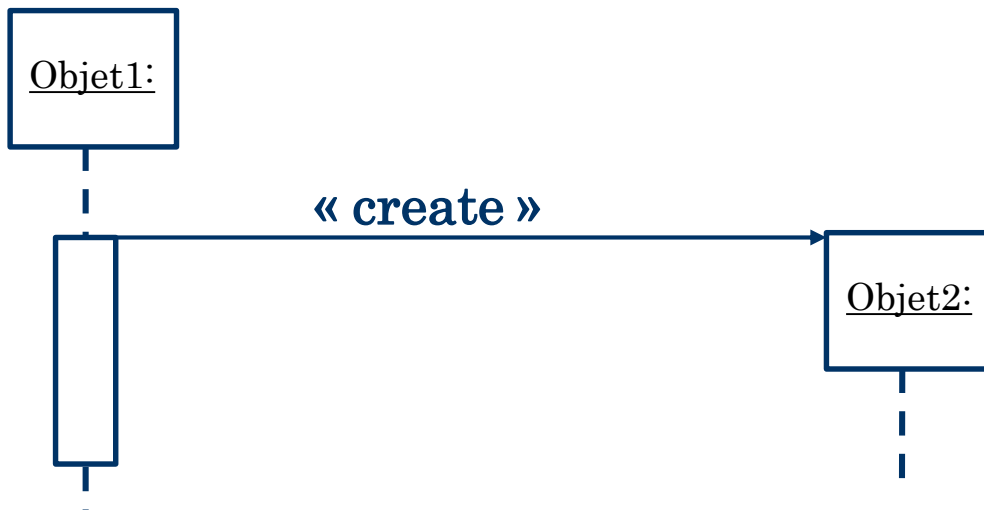
Diagramme d'interaction

2. Catégories de message.

➤ Message de création/destruction d'un objet

Création:

Message asynchrone stéréotype « create » de création pointant sur le rectangle qui symbolise l'objet créé. (:objet2).



L'objet 1 envoie un message de création à l'objet 2

IV. UML

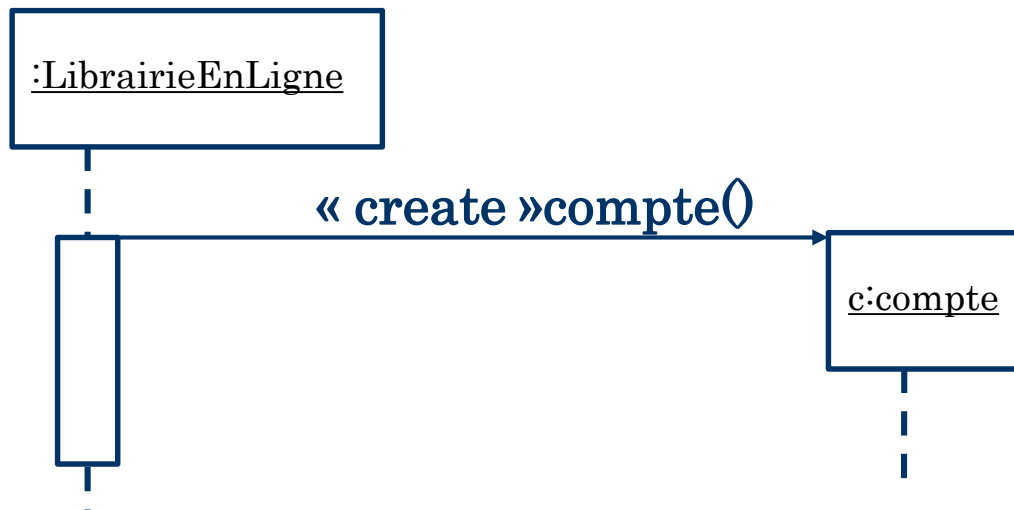
Diagramme d'interaction

2. Catégories de message.

➤ Message de création/destruction d'un objet

Création:

Message asynchrone stéréotype « create » de création pointant sur le rectangle qui symbolise l'objet créé. (:objet2).



L'objet anonyme librairie en ligne crée l'objet c de la classe compte à travers le constructeur compte

IV. UML

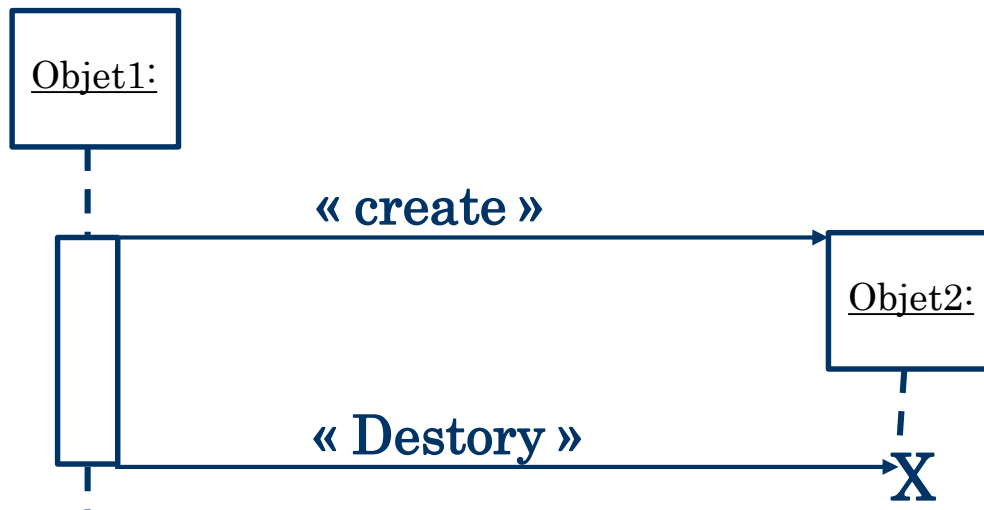
Diagramme d'interaction

2. Catégories de message.

➤ Message de création/destruction d'un objet

Destruction:

Message asynchrone stéréotype « destroy » précédant la croix sur la ligne de vie. (:objet2).



L'objet 1 envoie un message de destruction à l'objet 2

IV. UML

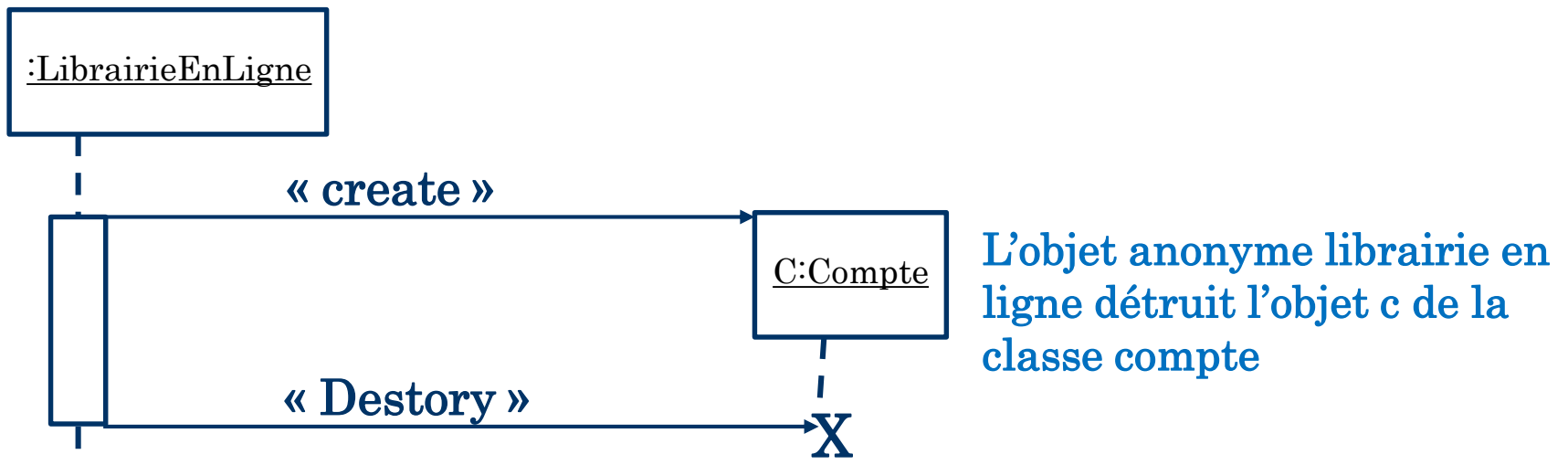
Diagramme d'interaction

2. Catégories de message.

➤ Message de création/destruction d'un objet

Destruction:

Message asynchrone stéréotype « destroy » précédant la croix sur la ligne de vie. (:objet2).



IV. UML

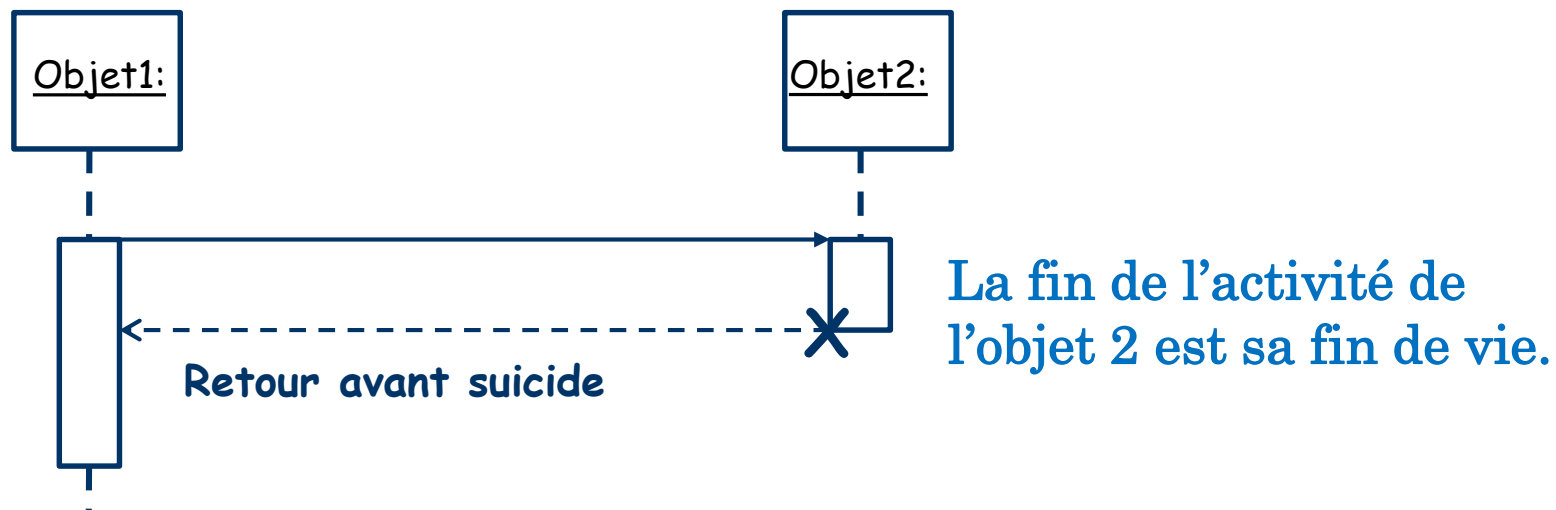
Diagramme d'interaction

2. Catégories de message.

➤ Message de création/destruction d'un objet

Suicide: la fin de la bande d'activation.

L'objet 2 reçoit un message de l'objet 1, l'objet 2 s'active, envoie une réponse à l'objet 1 puis se suicide.



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message de création/destruction d'un objet.
- **Message avec délai de transmission.**
- Message perdu/trouvé.
- Message avec garde.
- Message avec itération.

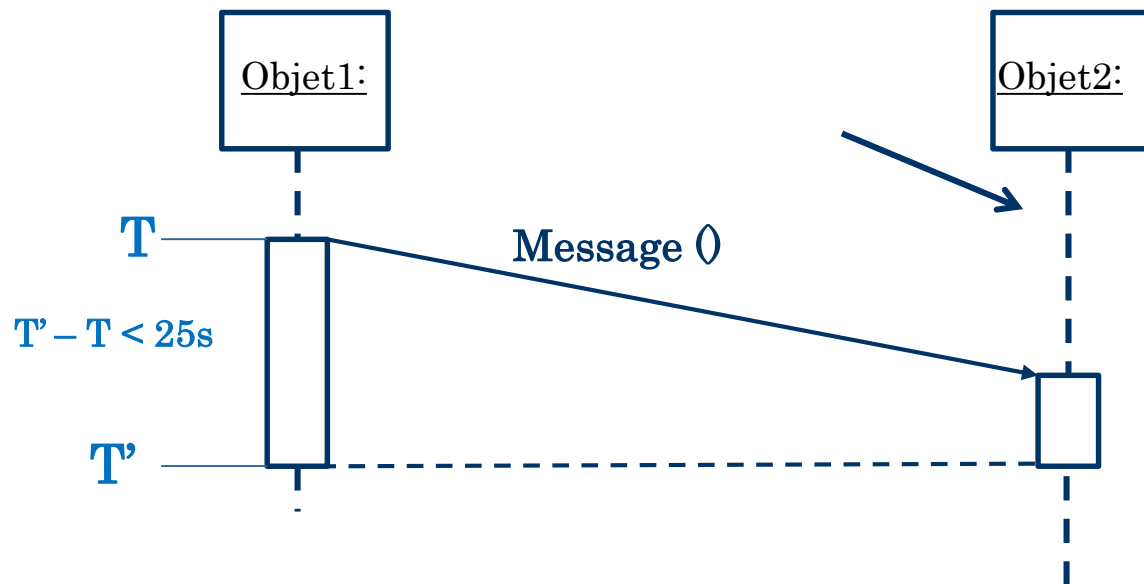
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message avec délai de transmission

*L'envoi d'un message est généralement considéré comme instantané. Pour **marquer une durée significative**, on **incline** la flèche du message vers le bas.*



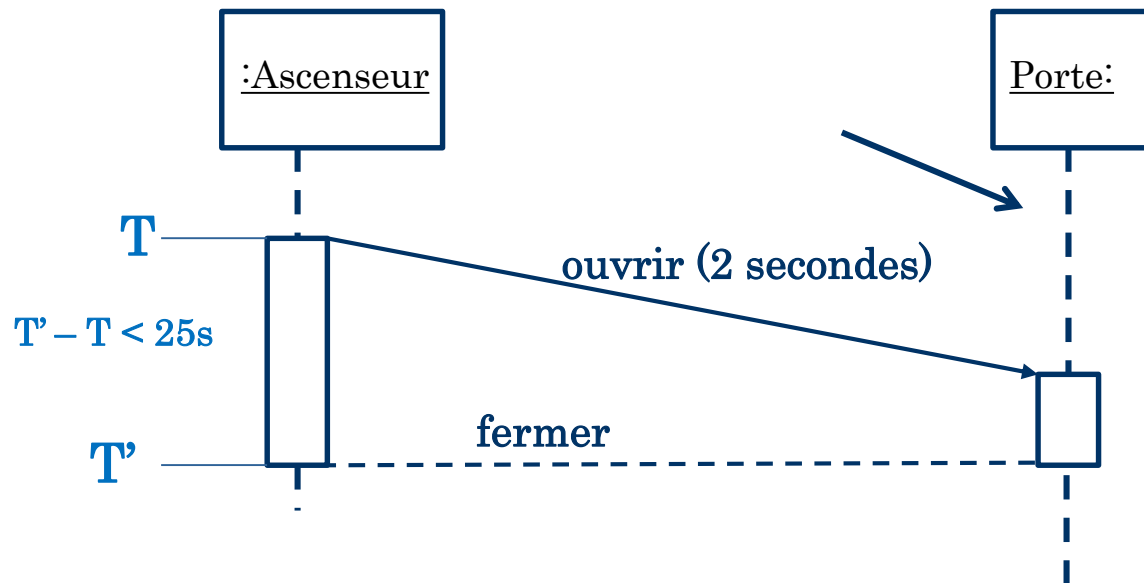
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message avec délai de transmission

*L'envoi d'un message est généralement considéré comme instantané. Pour **marquer une durée significative**, on **incline** la flèche du message vers le bas.*



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- **Message perdu/trouvé.**
- Message avec garde.
- Message avec itération.

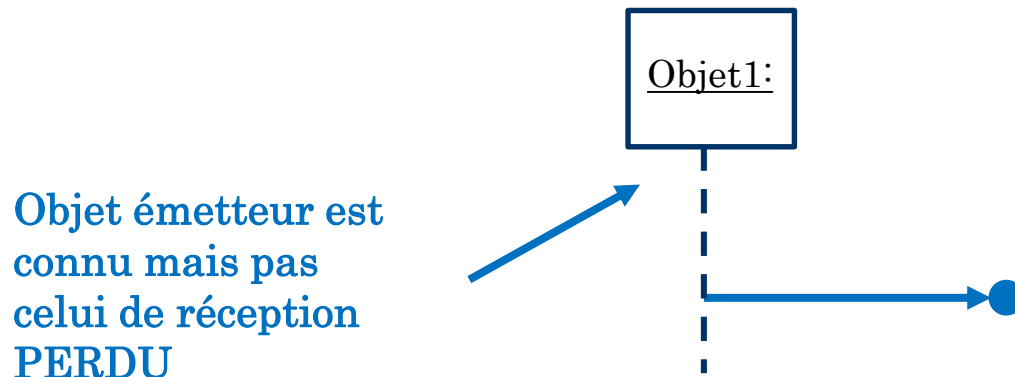
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message perdu/trouvé

*Un **message perdu** est un message dont l'événement de **d'envoi** est connu, mais pas l'événement de réception.*



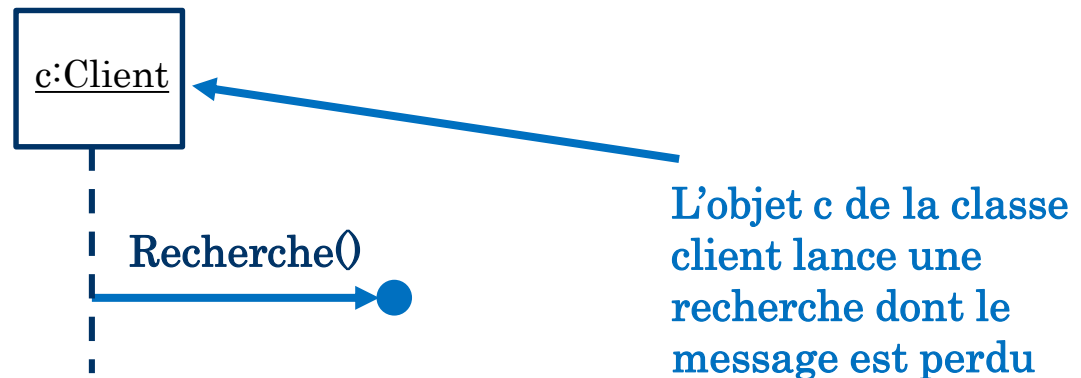
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message perdu/trouvé

*Un message perdu est un message dont l'événement de **d'envoi** est connu, mais pas l'événement de réception.*



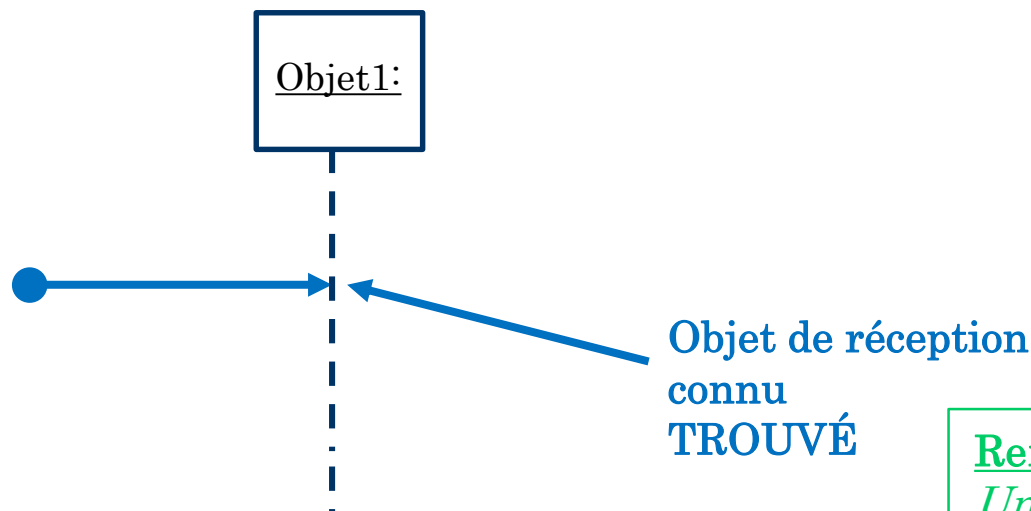
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message perdu/trouvé

Un message trouvé est un message dont l'événement de réception est connu, mais pas l'événement d'émission.



Remarque importante:

Un message complet est un message dont les événements d'envoi et de réception sont connus.

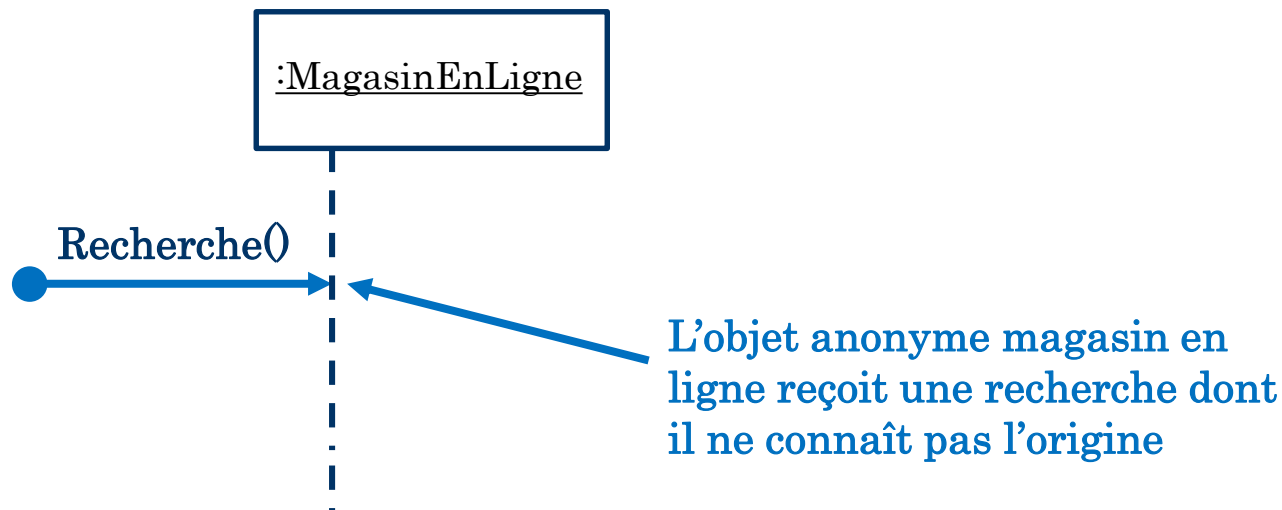
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message perdu/trouvé

Un message trouvé est un message dont l'événement de réception est connu, mais pas l'événement d'émission.



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- **Message avec garde.**
- Message avec itération.

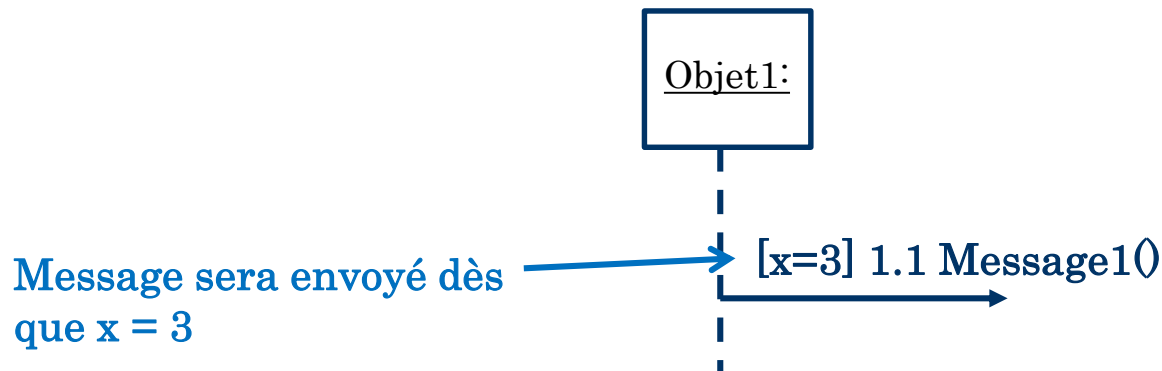
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message avec garde

Message envoyé seulement si la garde est vraie.



IV. UML

Diagramme d'interaction

2. Catégories de message.

- Message asynchrone.
- Message synchrone.
- Message retour (avec résultats et arguments).
- Message de création/destruction d'un objet.
- Message avec délai de transmission.
- Message perdu/trouvé.
- Message avec garde.
- **Message avec itération.**

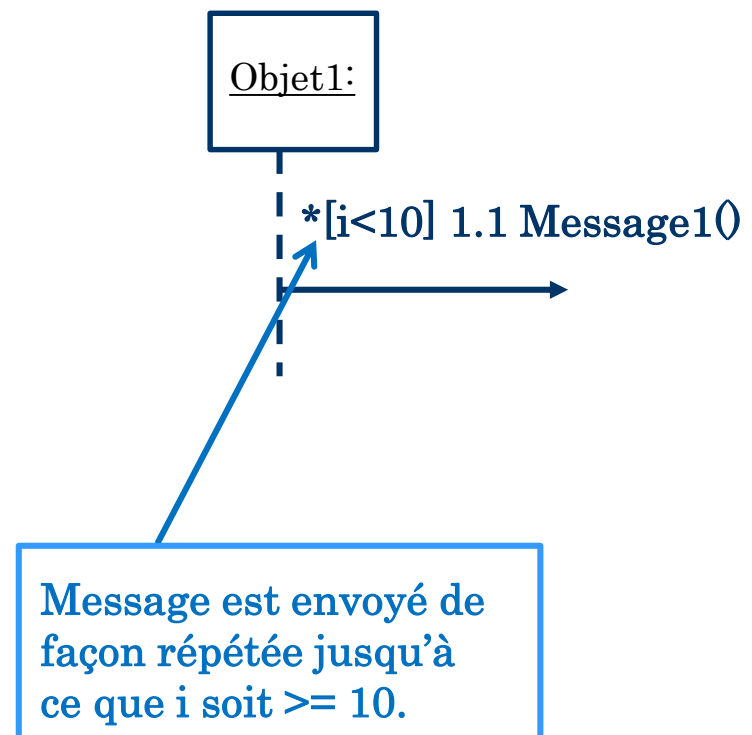
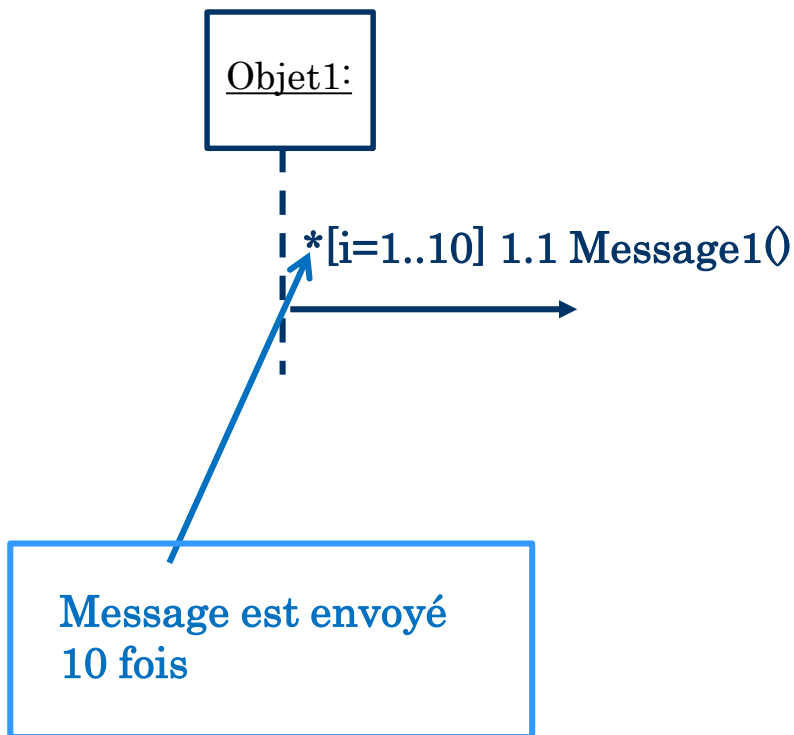
IV. UML

Diagramme d'interaction

2. Catégories de message.

➤ Message avec itération

Message envoyé de façon répétée jusqu'à que la condition soit validée.



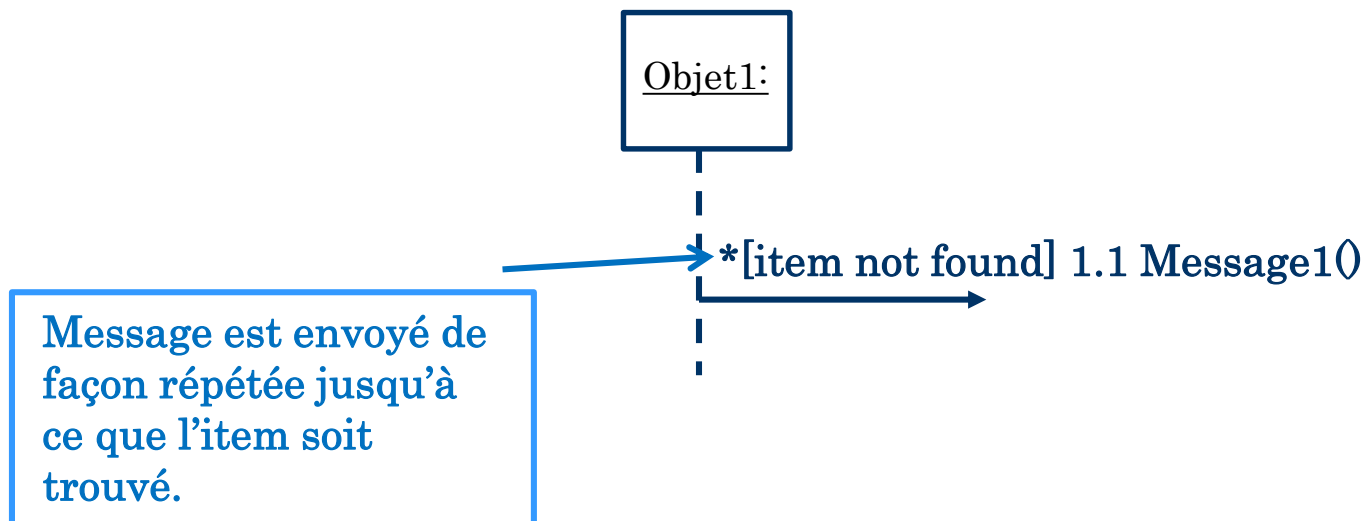
IV. UML

Diagramme d'interaction

2. Catégories de message.

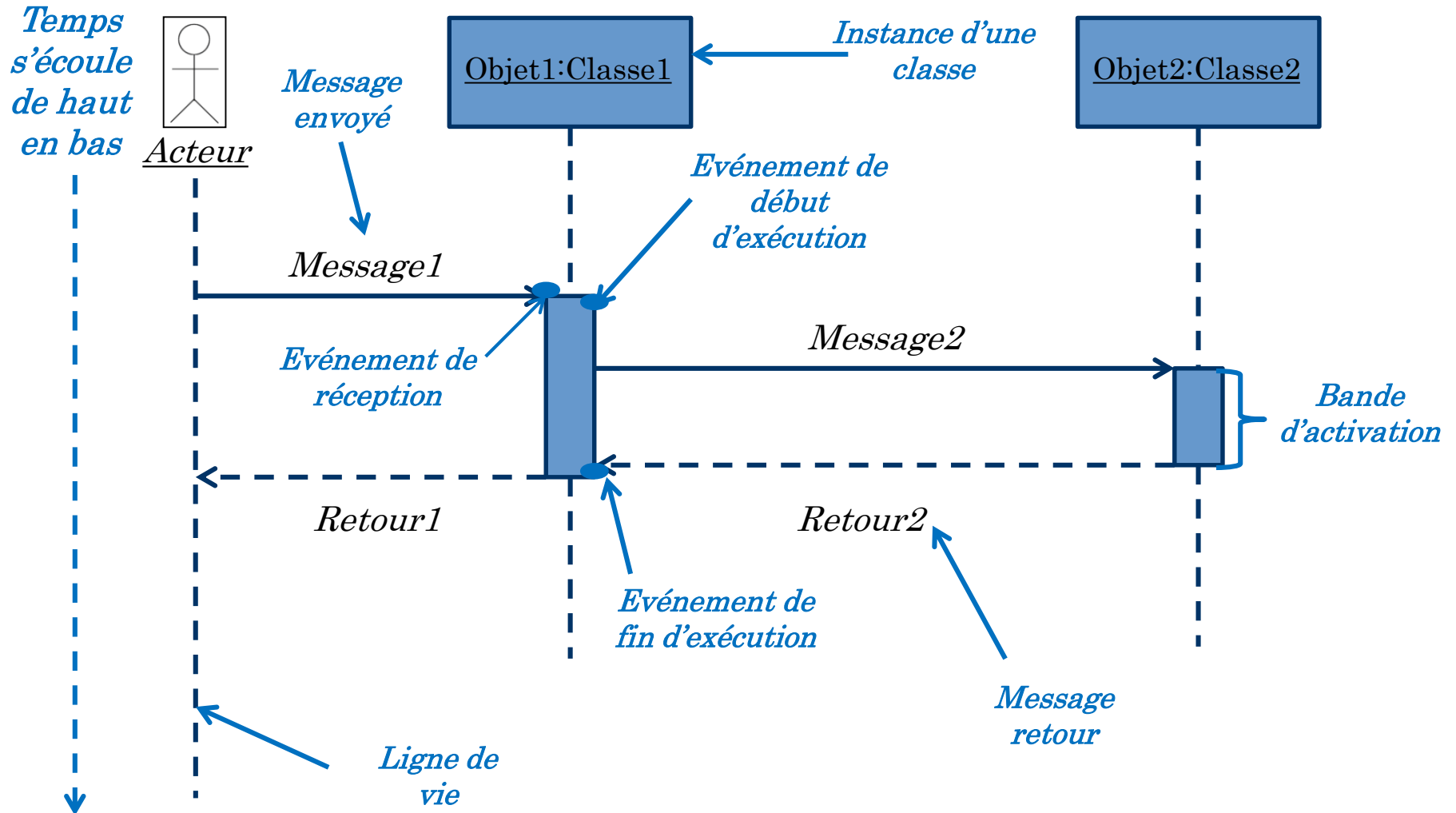
➤ Message avec itération

Message envoyé de façon répétée jusqu'à que la condition soit validée.



IV. UML

Enfin



IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence:

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de séquence.
3. Exemple de diagramme de séquence.
4. Fragments d'interaction.

IV. UML

Diagramme d'interaction

3.2 Diagramme de séquence:

1. Définition et formalisme.
2. Les éléments constitutifs d'un diagramme de séquence.
3. Exemple de diagramme de séquence.
4. Fragments d'interaction.

IV. UML

Diagramme d'interaction

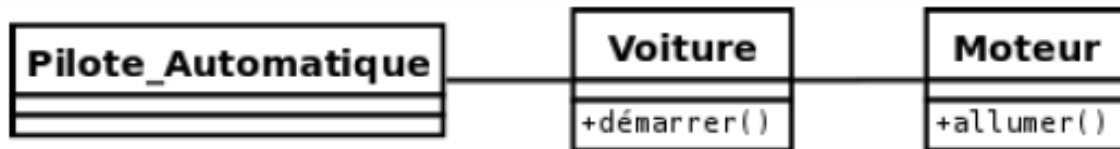


Diagramme de classes

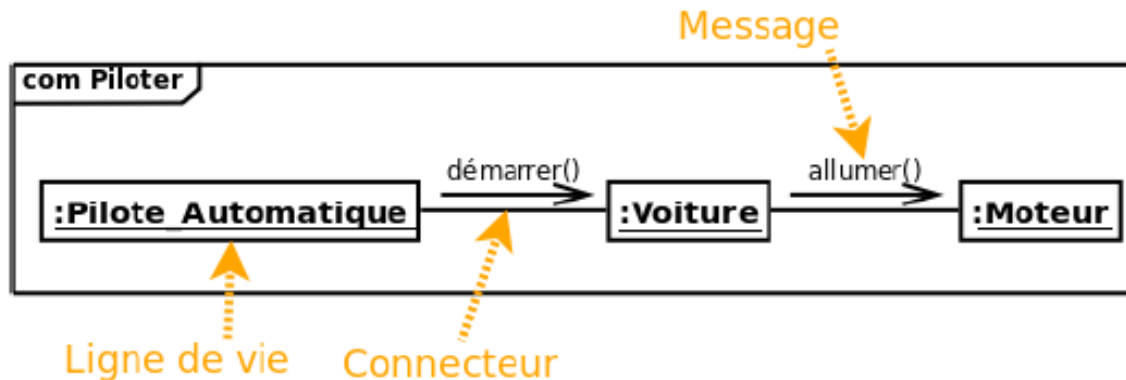
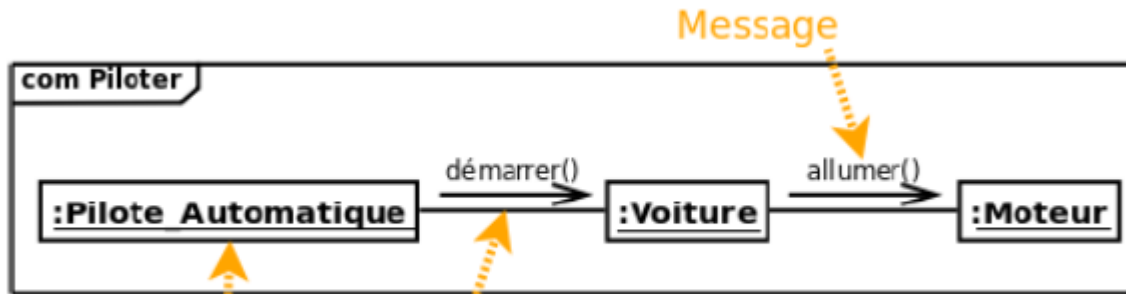


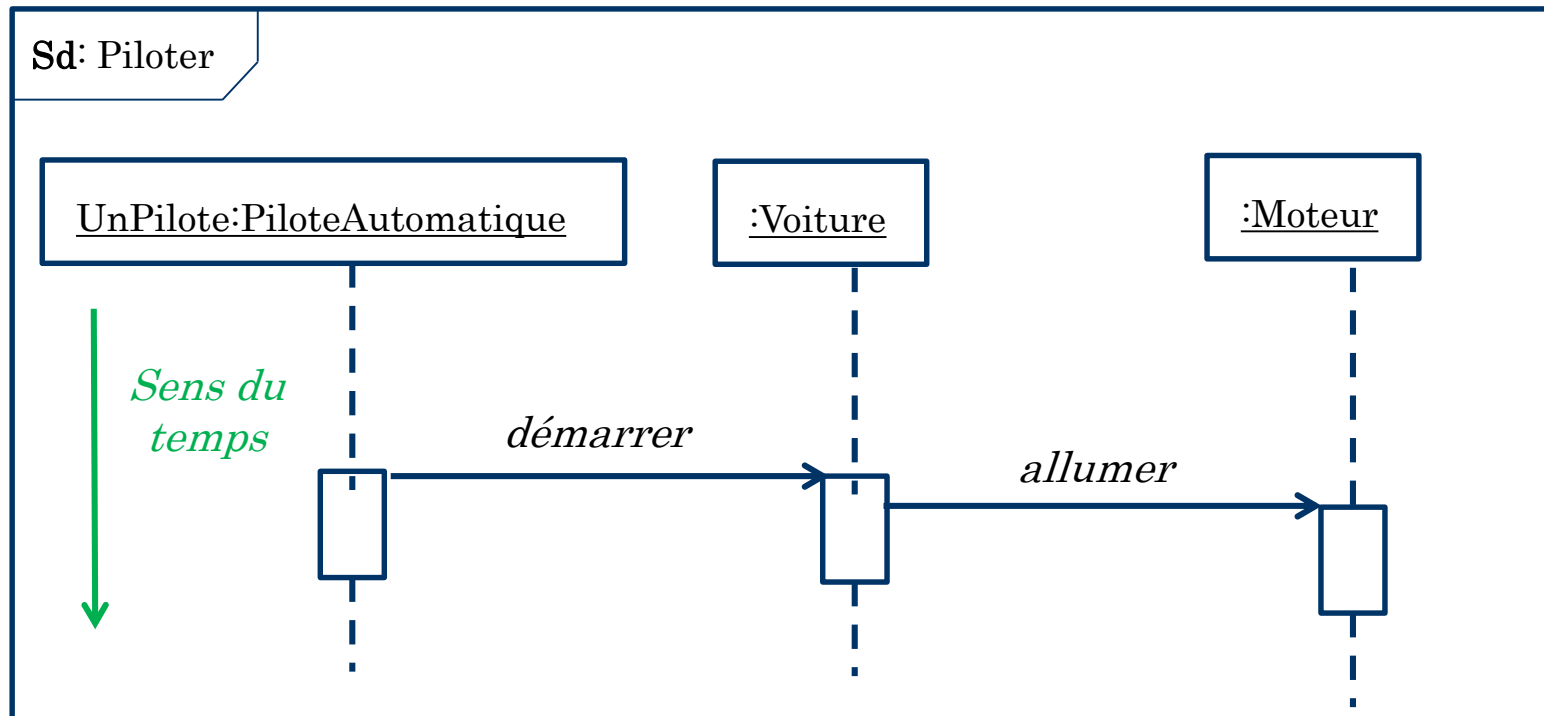
Diagramme de communication : focus sur **organisation structurelle** des objets qui communiquent.

IV. UML

Diagramme d'interaction



Ligne de vie Connecteur

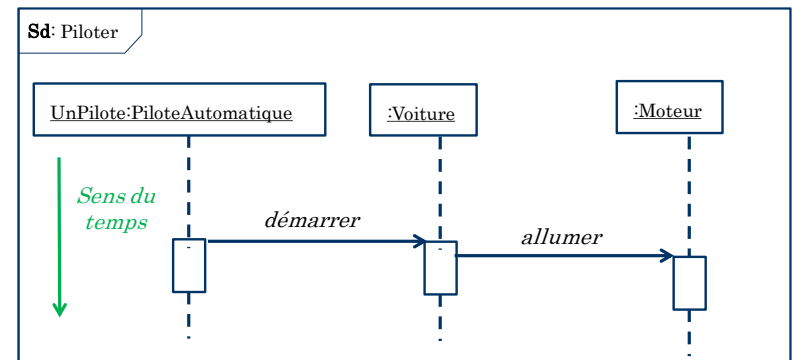
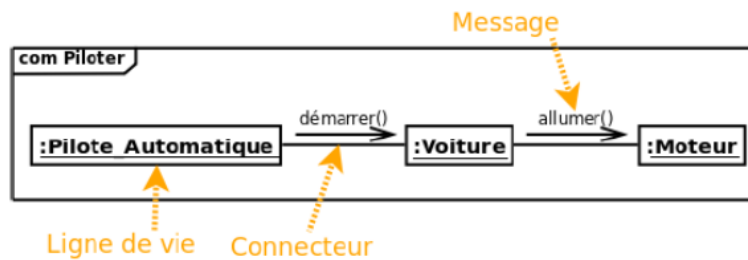


IV. UML

Diagramme d'interaction

Conclusion:

Diagrammes de séquence et de communication, peuvent représenter la même interaction, MAIS de point de vue différent.



Diagrammes de communication met l'accent sur le lien qui unit les lignes de vie, et qui est le vecteur du message

Diagrammes de séquence met l'accent sur le séquençage temporel des messages. (temps s'écoule de haut en bas).

IV. UML

Diagramme d'interaction

**Enoncés dans mon centre de
ressources**

IV. UML

Diagramme d'interaction

4. Fragments d'interaction.

1. Introduction.
2. Formalisme.
3. Les opérateurs dans UML2.0

IV. UML

Diagramme d'interaction

4. Fragments d'interaction.

1. Introduction.
2. Formalisme.
3. Les opérateurs dans UML2.0.

IV. UML

Diagramme d'interaction

1. Introduction.

Les diagrammes de séquence tels que définis en **UML 1** souffraient d'un gros inconvénient:

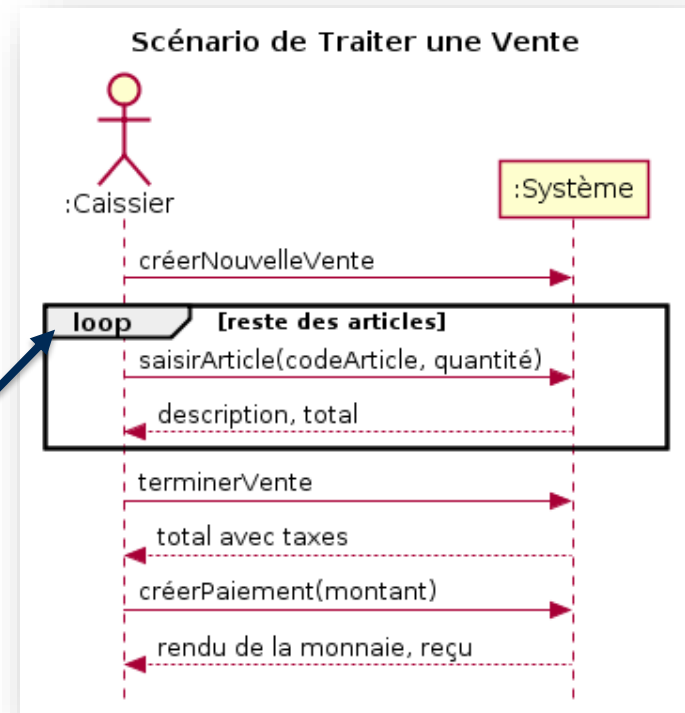
- La **quantité** de diagramme à réaliser pouvait atteindre un **nombre conséquent**.
- La **mise à jour** des diagrammes lors d'un changement des exigences était très **couteuse**.

IV. UML

Diagramme d'interaction

1. Introduction.

UML.2.0 a donné plus de puissance à ses diagrammes de séquences en introduisant un certain nombre d'opérateurs dans des **fragments d'interaction** combinés.



Exemple : représentation d'une boucle

IV. UML

Diagramme d'interaction

1. Introduction.

Fragments d'interaction:

Fragments d'interaction sont des regroupements **logiques** représentés par un rectangle.

Ils contiennent les structures **conditionnelles** qui affectent le flux de messages.

L'opérateur d'interaction identifie le **type d'instruction conditionnelle** ou **logique** qui définit le comportement du fragment combiné.

IV. UML

Diagramme d'interaction

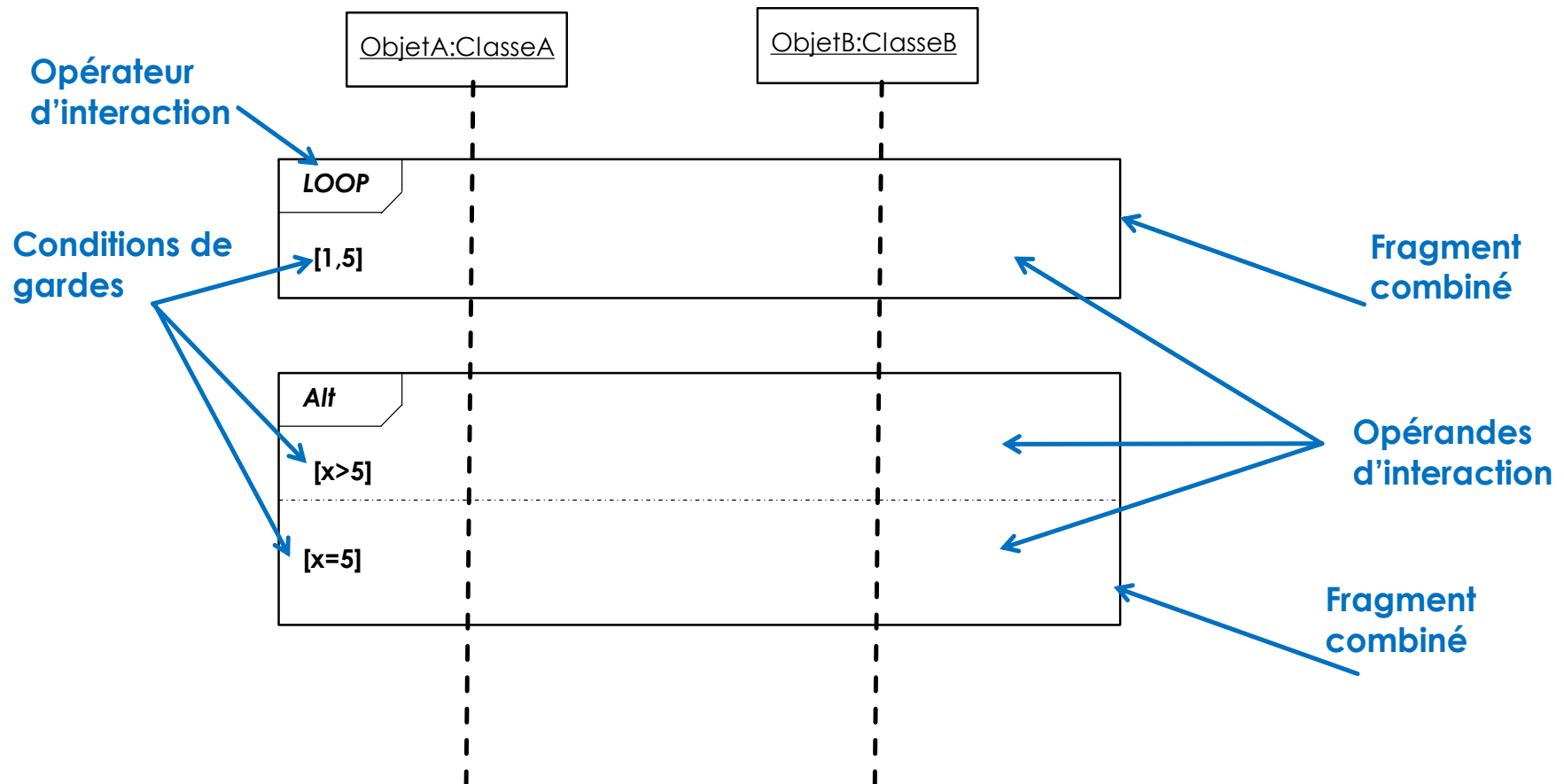
4. Fragments d'interaction.

1. Introduction.
2. Formalisme.
3. Les opérateurs dans UML2.0.

IV. UML

Diagramme d'interaction

2. Formalisme.



IV. UML

Diagramme d'interaction

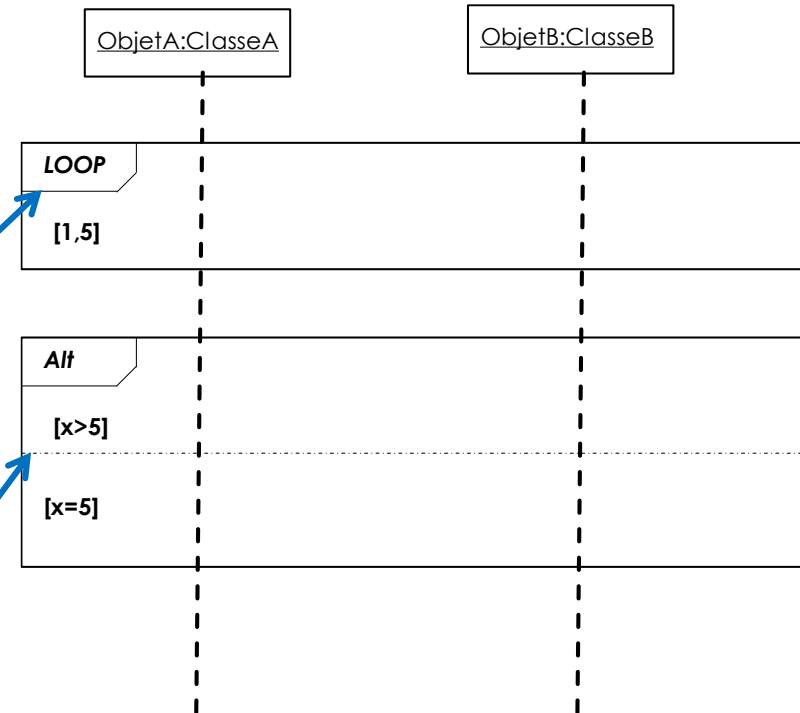
2. Formalisme.

Fragment combiné est défini par un opérateur et des opérandes.

Représenté par un rectangle dont le coin supérieur gauche contient un pentagone

Opérateur conditionne la signification du fragment.

Les opérandes d'un opérateur sont séparés par une ligne pointillée



IV. UML

Diagramme d'interaction

4. Fragments d'interaction.

1. Introduction.
2. Formalisme.
3. Les opérateurs dans UML2.0.

IV. UML

Diagramme d'interaction

2. Les opérateurs dans UML 2.0.

Il existe 12 opérateurs définis dans la notation UML 2.0

La liste suivante regroupe les opérateurs d'interaction par fonction:

- Les opérateurs de choix et de boucle:
alternative, option, break et loop.
- Les opérateurs contrôlant l'envoi en parallèle de messages:
parallel et critical region.
- Les opérateurs contrôlant l'envoi de messages:
ignore, consider, assertion et negative.
- Les opérateurs fixant l'ordre d'envoi des messages:
weak sequency, strict sequency.

Nous n'aborderons que quelques opérateurs (les plus utilisés) dans la suite .

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- *Par* (parallèle)
- *Strict* (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- *Par* (parallèle)
- *Strict* (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

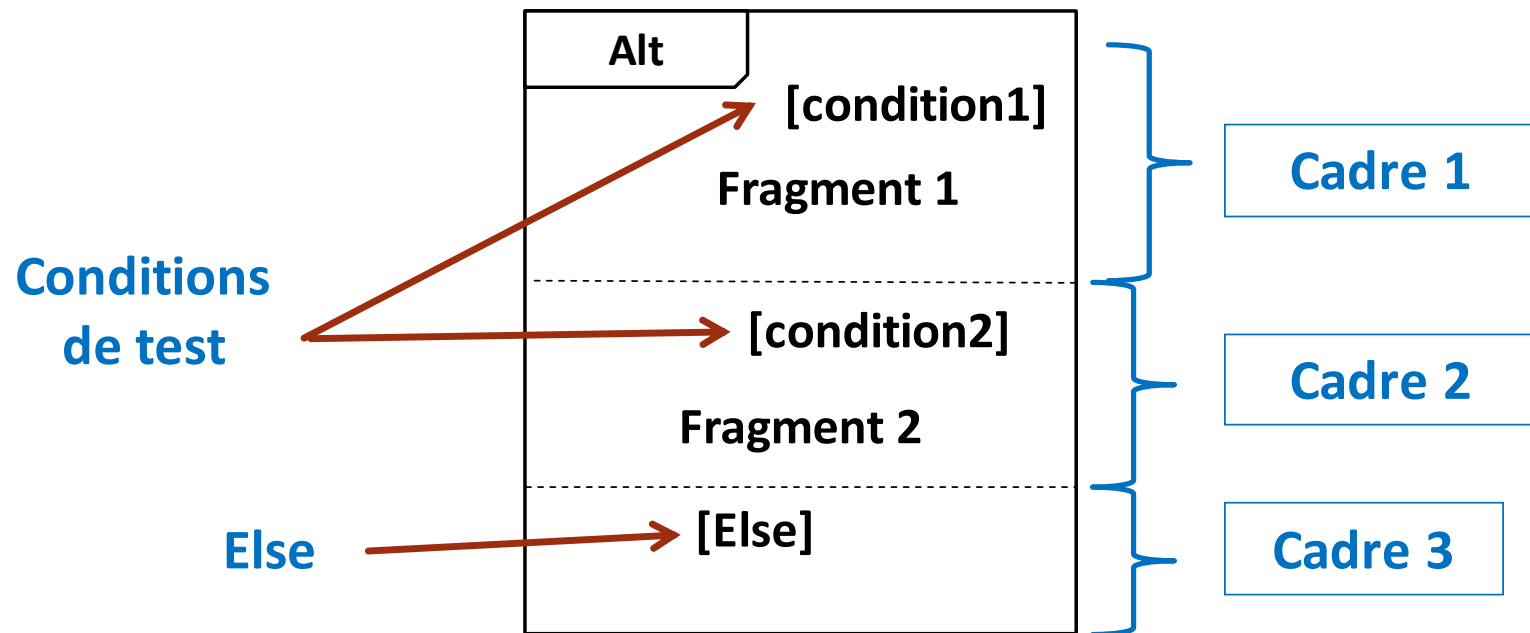
Opérateur Alt (alternative):

- *Equivalent à une instruction de type **if-then-else**.*
- *Seule l'une des alternatives proposées s'exécutera sur une passe à travers l'interaction.*
- *L'opérateur « Alt » correspond à une instruction de test avec une ou plusieurs alternatives possibles.*

IV. UML

Diagramme d'interaction

Opérateur Alt (alternative):



*Le contenu du **cadre 1** (fragment 1) est exécuté si la **condition 1** est remplie.*

*Le contenu du **cadre 2** (fragment 2) est exécuté si la **condition 2** est remplie.*

*Si aucune des conditions n'est vérifiée ALORS le contenu du **cadre 3** (associé au Else) est exécuté.*

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- *Par* (parallèle)
- *Strict* (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

Opérateur Opt (optionnel):

Equivalent à une instruction de type if.

*Pour que le **fragment s'exécute**, il faut que la **condition** soit **respectée**.*

Dans le cas contraire, le comportement est ignoré.

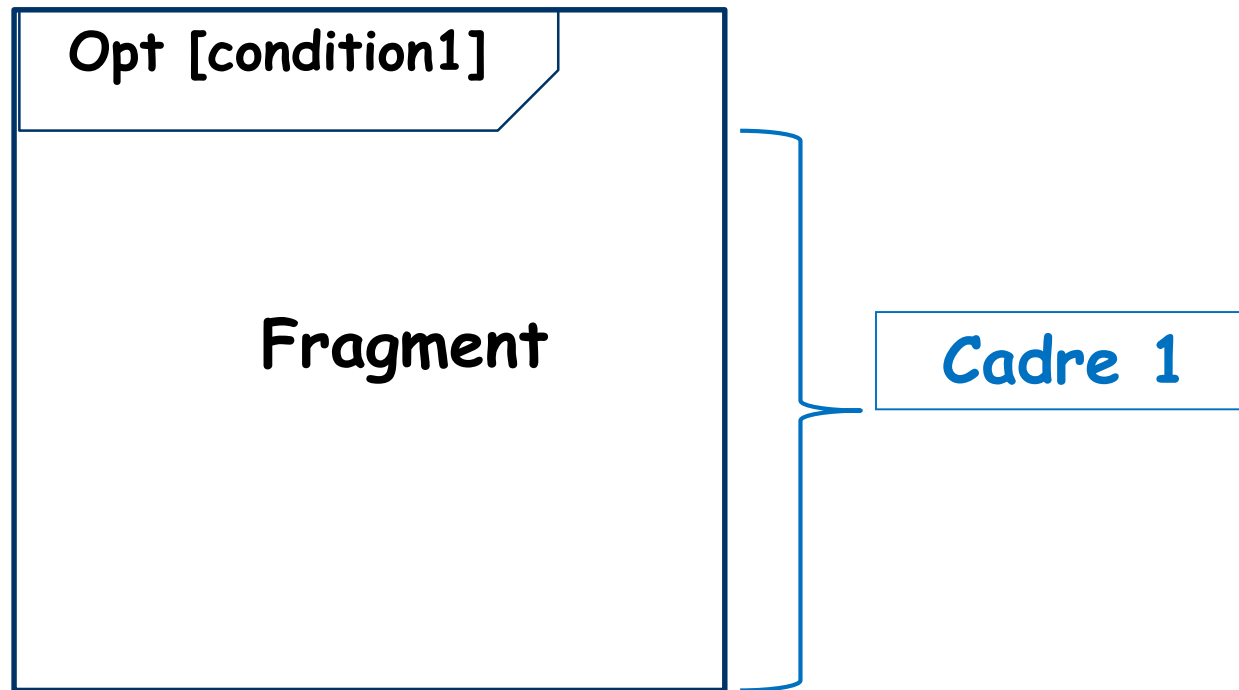


*C'est une instruction de test
sans alternative.*

IV. UML

Diagramme d'interaction

Opérateur Opt (optionnel):



*Si la **condition** de test (« condition1 ») est vérifiée alors le contenu du **cadre** (fragment) est exécuté.*

Si la condition est fausse, le fragment ne s'exécute pas.

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- ***Loop*** (boucle)
- *Par* (parallèle)
- *Strict* (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

Opérateur Loop (boucle)

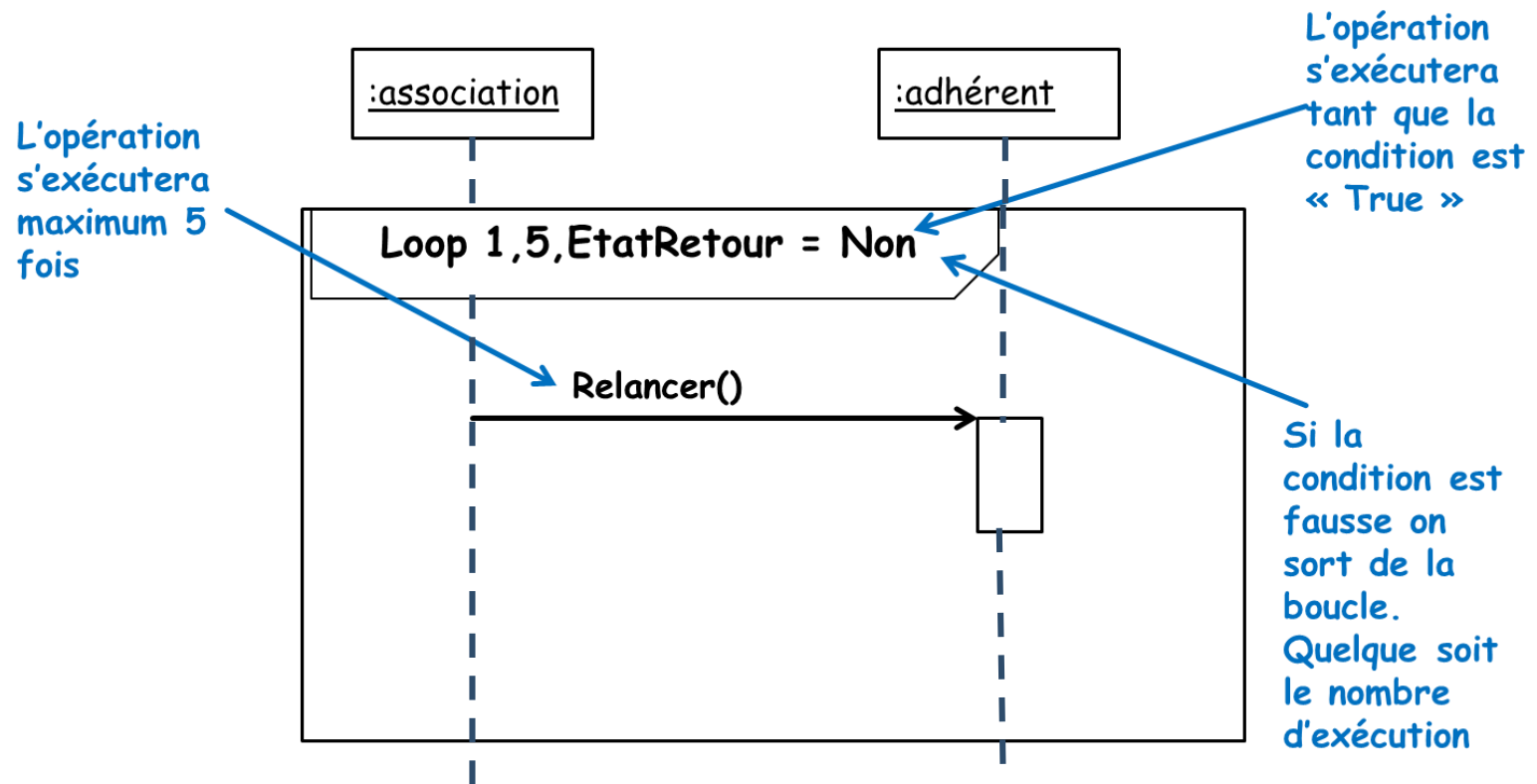
*L'opérateur « **Loop** » correspond à une instruction de **boucle** qui permet d'exécuter une séquence d'interaction tant qu'une **condition** est **satisfaite**.*

*Il est possible de **limiter le nombre** d'exécutions à un nombre minimum et maximum*

IV. UML

Diagramme d'interaction

Opérateur Loop (boucle):



IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

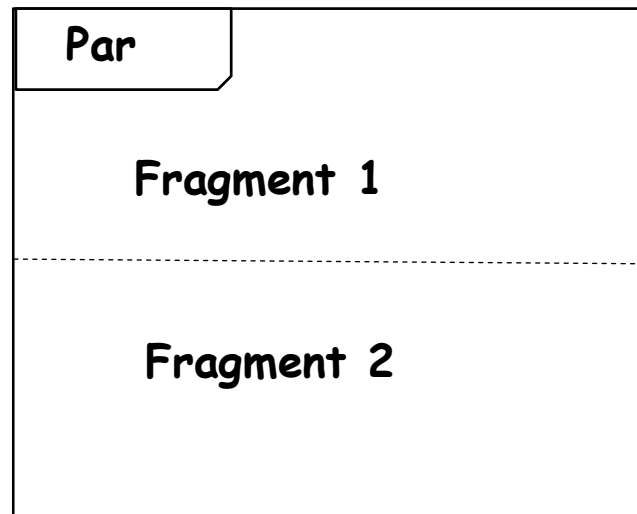
- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- ***Par*** (parallèle)
- *Strict* (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

Opérateur Par (parallèle)

*L'opérateur « Par » permet de représenter deux séries d'instructions qui se déroulent en **parallèles**.*

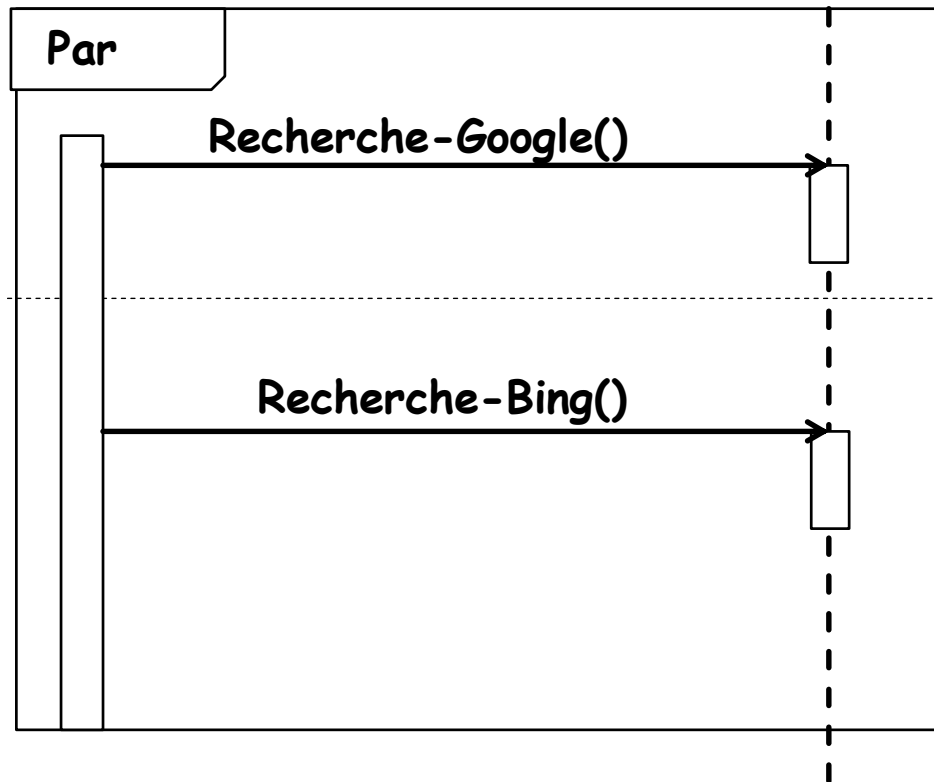


*Le fragment 1 et le fragment 2 sont exécutés **sans préférence d'ordre**.*

IV. UML

Diagramme d'interaction

Opérateur Par (parallèle)



La recherche se fera par Google et par Bing en parallèle, sans préférence.

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- *Par* (parallèle)
- ***Strict*** (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

Opérateur Strict (stricte)

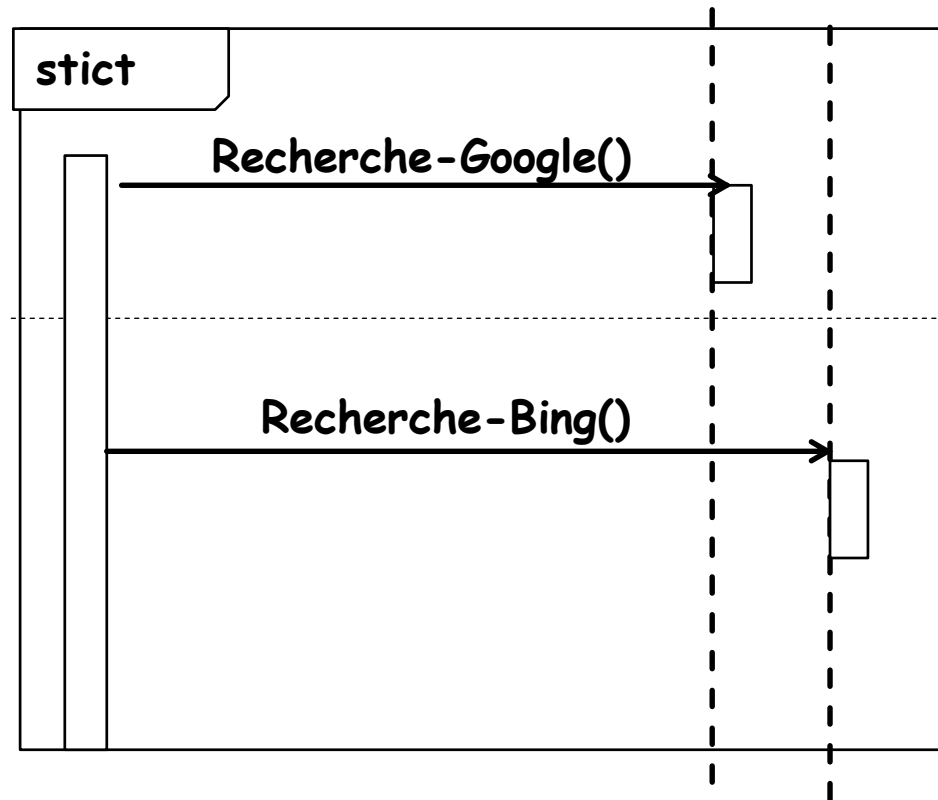
*L'opérateur d'interaction « **Strict** » définit l'ordre d'exécution des fragments d'interaction.*

Il force l'achèvement de l'interaction en cours avant l'exécution d'interactions imbriquées supplémentaires.

IV. UML

Diagramme d'interaction

Opérateur Strict (stricte)



La recherche se fera d'abord par Google et puis par Bing.

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- *Par* (parallèle)
- *Strict* (stricte)
- ***Critical*** (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

Opérateur *Critical* (région critique)

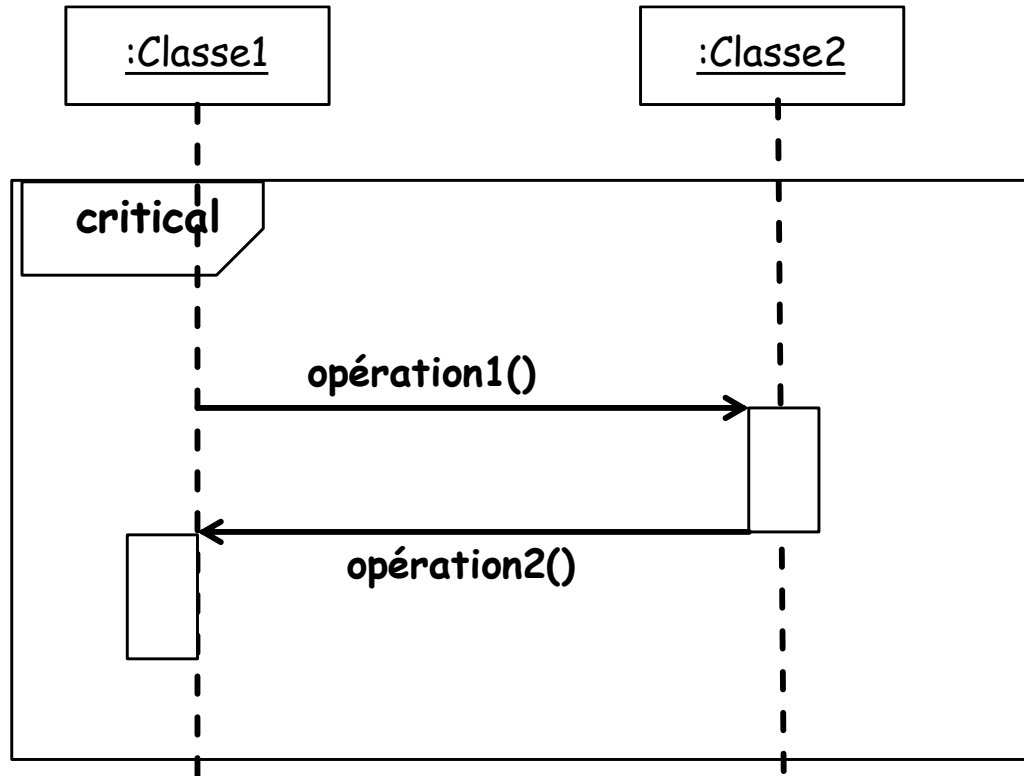
*L'opérateur « **Critical** » indique que la séquence d'interactions **ne peut être interrompue** compte tenue du caractère critique des opérations traitées.*

*On considère que le **traitement des interactions** comprises dans la séquence critique est **atomique**.*

IV. UML

Diagramme d'interaction

Opérateur Critical (région critique)



Les opérations du fragment « Critical » doivent s'exécuter sans interruption.

IV. UML

Diagramme d'interaction

3. Les opérateurs dans UML 2.0.

Opérateurs les plus utilisés:

- *Alt* (alternative)
- *Opt* (optionnel)
- *Loop* (boucle)
- *Par* (parallèle)
- *Strict* (stricte)
- *Critical* (région critique)
- *Break* (interruption)

IV. UML

Diagramme d'interaction

Opérateur Break (interruption)

L'opérateur « Break » permet de représenter une situation exceptionnelle.

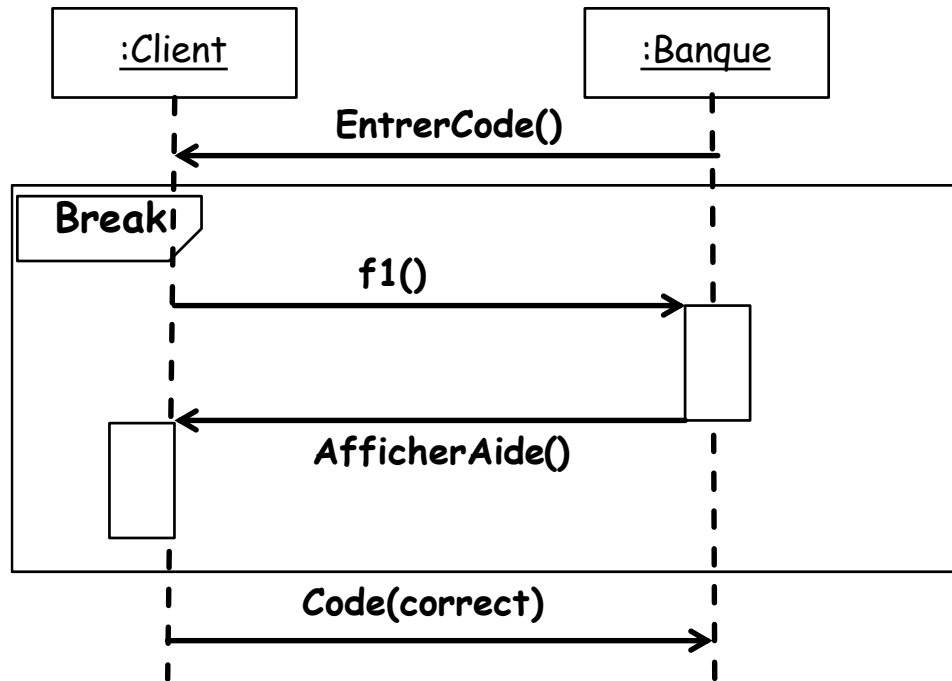
L'opérateur « Break » est utilisé dans les fragments qui représentent des scénarios d'exception.

Lorsque la condition de franchissement est remplie (true), l'exécution de l'interaction en cours est abandonnée.

IV. UML

Diagramme d'interaction

Opérateur Break (interruption)

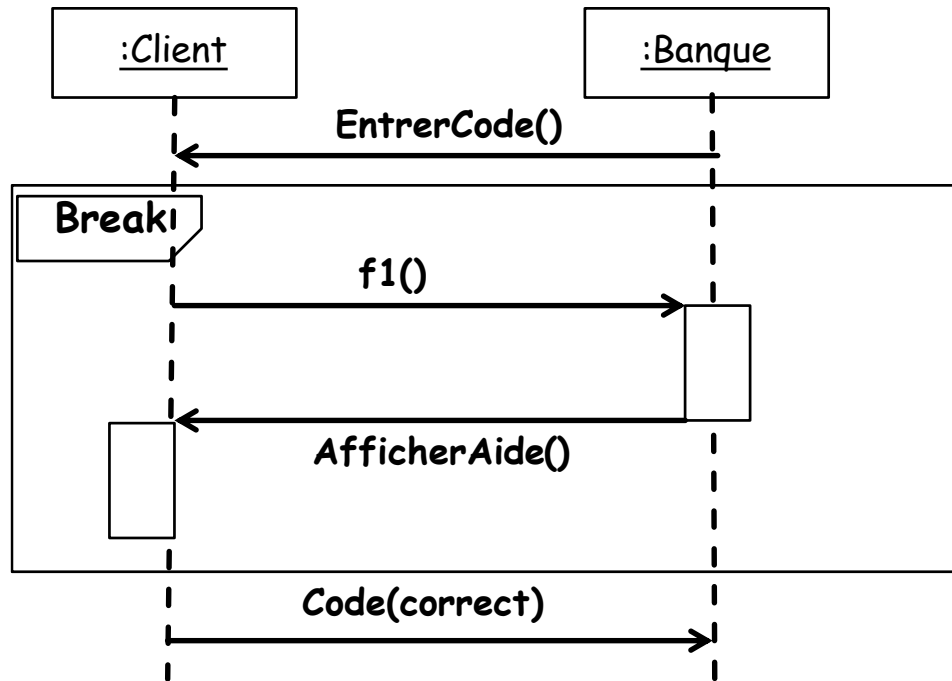


*Le distributeur demande « **EntrerCode** », l'utilisateur peut choisir de rentrer son code (« **Code(correct)** ») ou consulter l'aide.*

IV. UML

Diagramme d'interaction

Opérateur Break (interruption)



*S'il choisit de **consulter l'aide**, l'interaction « Code(correct) » est **interrompue**.
Les interactions de l'opérateur **break** sont **exécutées**.*

IV. UML

Diagramme d'interaction

1. Objectif.
2. Interactions et ligne de vie.
3. Diagrammes d'interaction.
 - 3.1 Diagramme de communication.
 - 3.2 Diagramme de séquence.
4. **Exercices**

IV. UML

Diagramme d'interaction

**Enoncés dans mon centre de
ressources**

Bibliographie

1. Benoît CHARROUX, Aomar OSMANI, Yann THIERRY-MIEG. UML2 Pratique de la modélisation. 3^{ème} édition. PEARSON.
2. Laurent AUDIBERT. UML2 de l'apprentissage à la pratique. 2^{ème} édition. ELLIPSES.
3. Christian SOUTOU. Modélisation des bases de données (UML et les modèles entité-association). 3^{ème} édition. EYROLLES.
4. Chantal MORLEY, Jean HUGUES, Bernard LEBLANC. 4^{ème} édition. UML 2 pour l'analyse d'un système d'information. DUNOD.
5. Hugues BERSINI. L'orienté objet. 3^{ème} édition. EYROLLES.
6. Laurent DEBRAUWER, Fien VAN DER HEYDE. UML 2.5. 4^{ème} édition. ENI Editions.
7. Jean-Luc HAINAUT. Bases de données concepts, utilisation et développement. DUNOD.
8. Gilles ROY. Conception de bases de données avec UML. Presses de l'université du Québec.
9. Craig LARMAN. UML2 et les design patterns. 3^{ème} édition. PEARSON Education.
10. Frank BARBIER. UML 2 et MDE. DUNOD.
11. Laurent DEBRAUWER, Naouel KARAM. UML 2 entraînez-vous à la modélisation. Seconde édition. ENI Editions.
12. Corine COSTA. Cours Projets et bureau d'études.