



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

Université Abderrahmane Mira de Bejaïa  
Faculté des sciences exactes  
DEPARTEMENT D'INFORMATIQUE

---

# ***Génie Logiciel***

Niveau : 3<sup>ème</sup> année licence (MI)

## **Chapitre 5 : Diagrammes UML (vue dynamique - Partie 1)**

Chargés de cours :  
Dr M. MOHAMMEDI  
& Dr N. YESSAD

Année universitaire 2024/2025

# Plan du chapitre 5 - Partie 1



Diagrammes d'interaction



Diagramme de séquence



Diagramme de communication

# Diagrammes d'interaction

- ❑ Le **diagramme d'interaction** décrit la **coopération dynamique** entre les **objets** et les **acteurs**.
- ❑ Il représente la **séquence d'actions** dans le **processus de coopération**.
- ❑ Les **diagrammes d'interaction** sont **utilisés** pour **décrire le comportement** d'un **cas d'utilisation**.
- ❑ Ils **montrent** les **objets impliqués** dans le **cas d'utilisation** et les **messages échangés** entre **ces objets**.
- ❑ Il existe deux **formes** de **diagrammes d'interaction** : les **diagrammes de séquence** et les **diagrammes de communication**.

# Diagrammes d'interaction

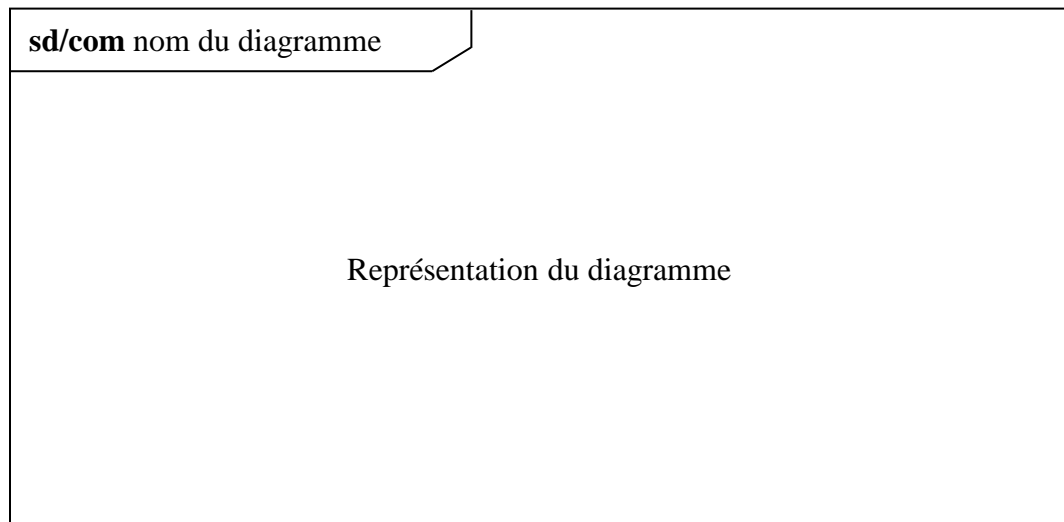
- ❑ Le **diagramme de séquence** présente les **échanges de messages** dans l'**ordre chronologique**.
- ❑ Le **diagramme de communication** (anciennement appelé **diagramme de collaboration**) se concentre sur la façon dont les **composants du système fonctionnent ensemble**.

# Diagrammes d'interaction

- ❑ Le diagramme de **séquence** et le diagramme de **communication** sont utilisés pour **représenter l'interaction** et le **comportement** d'un **système** sous différents angles.
- ❑ Ces **deux types de diagrammes** peuvent être **transformés l'un dans l'autre**, ce qui signifie qu'ils peuvent **représenter les mêmes informations**, mais avec une **présentation graphique différente**.
- ❑ Le **diagramme d'interaction** est **placé** dans un **grand rectangle** avec une **étiquette** en haut à gauche.
- ❑ L'étiquette est **représentée** sous forme d'un **pentagone** et contient un mot-clé (**sd** ou **com**) suivi du **nom de l'interaction**.

# Diagrammes d'interaction

- ❑ L'utilisation du mot-clé "sd" indique qu'il s'agit d'un **diagramme de séquence**, tandis que l'utilisation du mot-clé "com" indique qu'il s'agit d'un **diagramme de communication**.
- ❑ Le **formalisme** général d'un **diagramme d'interaction** suit une **structure spécifique**, qui **peut varier** en **fonction** du **type** de **diagramme utilisé** :



# Diagrammes de séquence

- ❑ Un **diagramme de séquence** est un **outil visuel** qui **permet** de **représenter l'interaction** entre différents **éléments** d'un **système** en mettant l'accent sur l'ordre **chronologique** des **messages échangés**. Il est **largement utilisé** dans le **domaine** de l'**ingénierie logicielle** pour **modéliser** le **comportement d'un système**.
- ❑ Il **existe** deux **types** principaux de **diagrammes de séquence**. Le premier type est le **diagramme de séquence système**, qui met en évidence les **interactions** entre le système **étudié** et son **environnement externe**. Ce type de diagramme **permet** de **visualiser** comment le **système réagit** aux **entrées** et **génère** des **sorties** en **fonction** des **messages reçus**.

# Diagrammes de séquence

- Le deuxième type est le **diagramme de séquence décrivant les interactions entre objets**. Ce type de **diagramme** se **concentre** sur les **échanges de messages** entre les **différents objets d'un système**. Il **permet de modéliser les interactions** entre les **objets**, de **montrer l'ordre** dans lequel les **messages** sont **envoyés** et **reçus**, ainsi que les **conditions de synchronisation** entre les **objets**.



# Diagrammes de séquence : Les diagrammes de séquence système

## 1) Diagramme de séquence système :

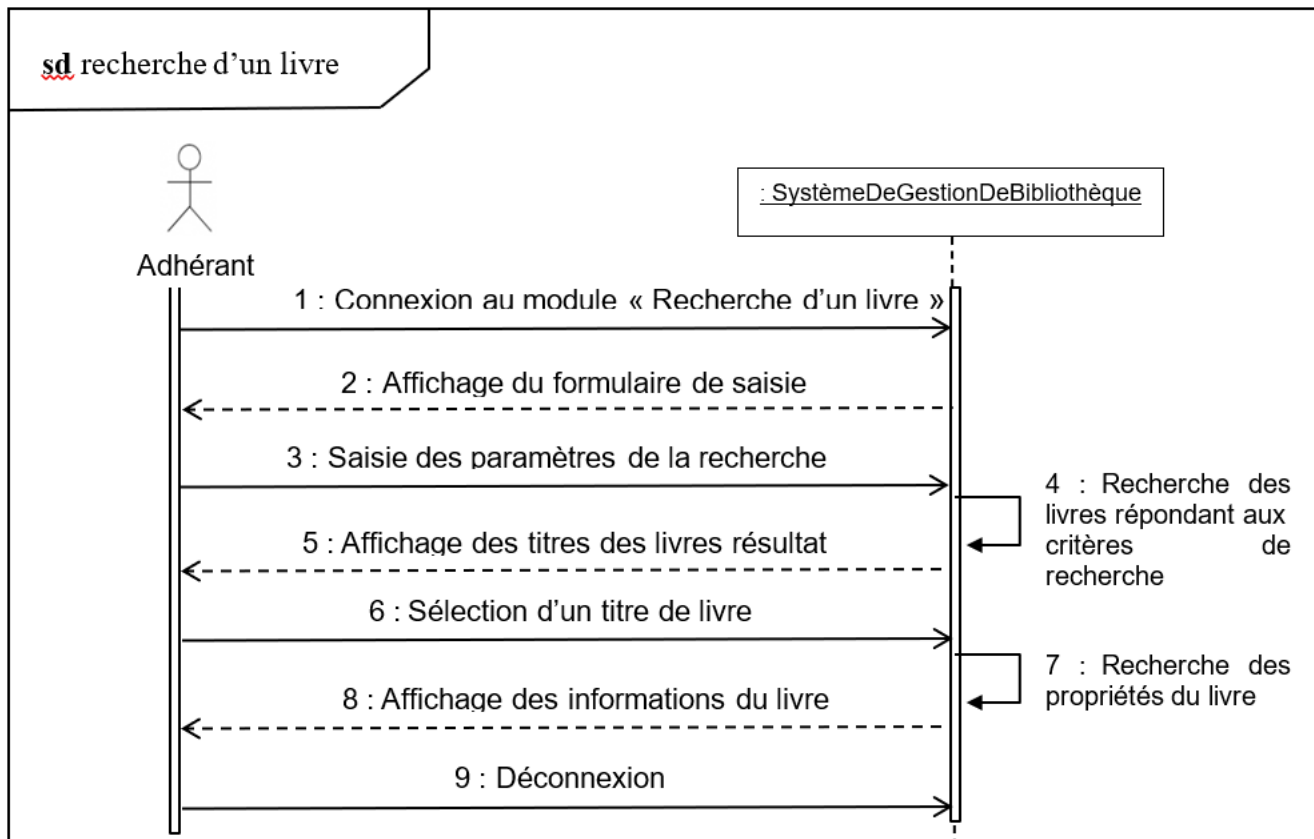
- ❑ Les diagrammes de séquence système sont **utilisés** pour la **documentation d'un cas d'utilisation**.
- ❑ Ils **assurent** la **description** des **interactions** dans des **termes proches** de **l'utilisateur**.
- ❑ Les **scénarios** de la **description textuelle** des **cas d'utilisation** peuvent être **considérés** comme des **instances** de **cas d'utilisation**.
- ❑ Ils sont illustrés par des **diagrammes de séquence système**.
- ❑ Pour **chaque cas d'utilisation**, il est **nécessaire** de **représenter** au moins le **scénario nominal** par un **diagramme de séquence**.
- ❑ Ce **diagramme** doit **rendre compte** de **l'interaction** entre **l'acteur** (ou les acteurs) et le **système**.

# Diagrammes de séquence : Les diagrammes de séquence système

- ❑ Le **système** est **considéré** dans son **ensemble** et est **représenté** par une **ligne de vie**.
- ❑ Chaque **acteur** est **associé** à une **ligne de vie**.
- ❑ Dans le **diagramme de séquence système**, nous avons la **disposition suivante** des **participants** :
  1. L'**acteur principal** est **positionné** à **gauche** ;
  2. Le **système**, **représenté** par une **boîte noire**, est **placé** au **centre** ;
  3. Les **acteurs secondaires**, qui **peuvent** être **sollicités** pendant le **scénario**, sont **positionnés** à **droite** du **système**.

# Diagrammes de séquence : Les diagrammes de séquence système

- L'exemple suivant représente un diagramme de séquence système correspondant au scénario nominal associé au cas d'utilisation «recherche d'un livre».



# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

## 2) Diagramme de séquence décrivant les interactions entre objets :

Ils sont utilisés pour représenter les interactions entre objets dans un système logiciel. Ils fournissent une vision chronologique des échanges entre les objets, mettant en évidence les messages échangés, les paramètres associés et les valeurs de retour. Les éléments clés qui sont impliqués dans un diagramme de séquence représentant les interactions entre objets sont :

**1. Ensemble d'objets** : Un diagramme de séquence implique un ensemble d'objets qui interagissent les uns avec les autres. Ces objets peuvent être des instances de classes dans le système logiciel ou des acteurs externes qui interagissent avec le système.

**2. Chronologie des échanges** : Le diagramme de séquence représente la séquence des échanges entre les objets.

# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

Il montre l'ordre dans lequel les messages sont envoyés et reçus entre les objets. Ainsi, il permet de visualiser le flux d'exécution des interactions.

**3. Messages** : ils sont les moyens par lesquels les objets communiquent entre eux. Ils peuvent être des appels de méthode, des demandes d'informations ou des notifications. Les messages peuvent inclure des paramètres, qui spécifient les données transmises, ainsi que des valeurs de retour qui représentent les résultats renvoyés par l'objet appelé.

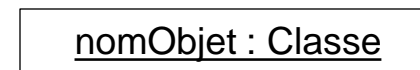
**4. Contraintes de temps** : Les diagrammes de séquence peuvent également prendre en compte les contraintes de temps. Ils peuvent montrer la durée des messages, les délais entre les envois de messages et les synchronisations entre les objets.

# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

En combinant ces éléments, les diagrammes de séquence offrent une représentation visuelle claire des interactions entre objets dans un système logiciel. Ils sont couramment utilisés lors de la conception et de l'analyse de systèmes orientés objet pour comprendre et spécifier le comportement dynamique du système.

## Ligne de vie

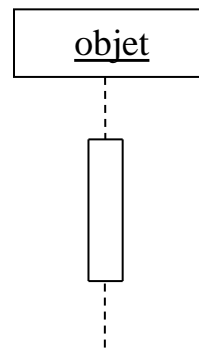
Une ligne de vie est utilisée pour indiquer l'existence d'objets dans le diagramme de séquence sur une période de temps. Elle est représentée graphiquement sous la forme d'une ligne en pointillés s'étendant vers le bas à partir de l'icône de l'objet concerné, indiquant la durée pendant laquelle l'objet existe, comme illustré dans la figure suivante :



# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

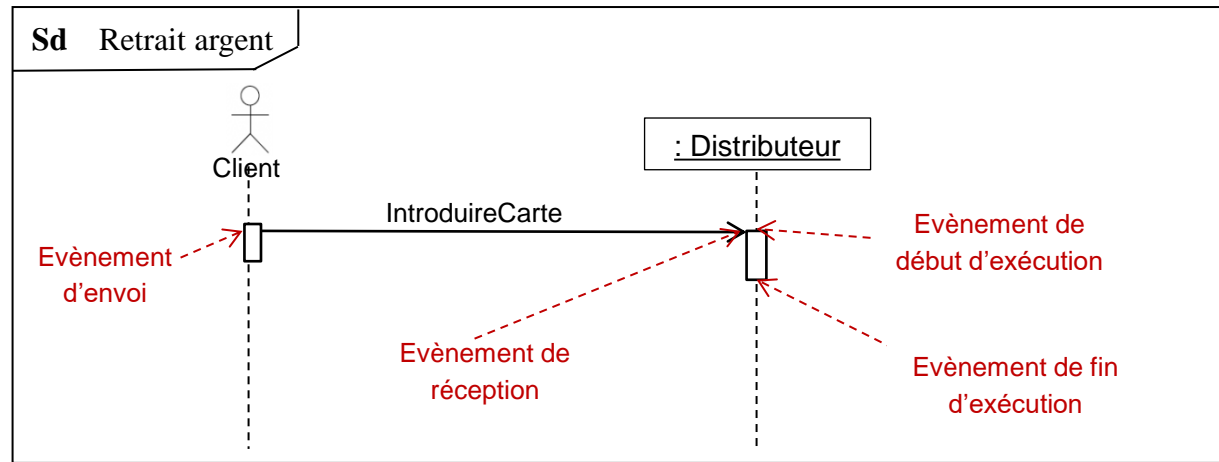
## Barre d'activation

Lorsqu'un **objet** est **déjà activé** il peut quand même **recevoir** d'autres **messages** (appel d'une autre de ses méthodes), cette **période d'activité** se représente par une **bande rectangulaire** sur la **ligne de vie de l'objet**. Le **début** et la **fin** d'une **bande** correspondent **respectivement** au **début** et à la **fin** d'une **période d'activité**.



# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

Exemple :



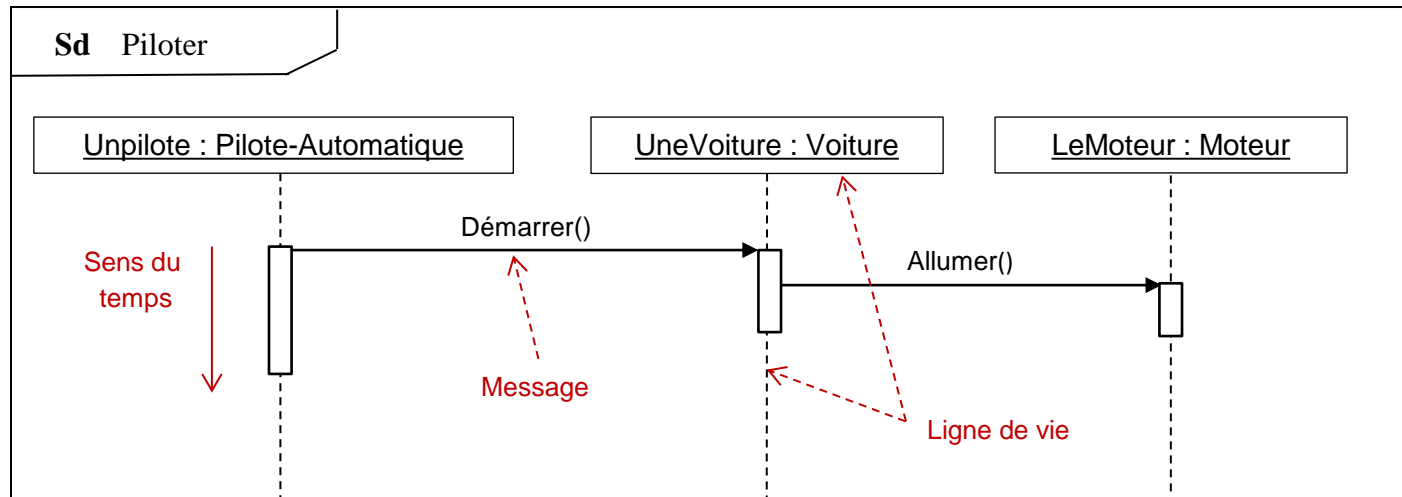
## Représentation des interactions

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.



# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

Exemple :



## Messages

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

Ainsi, un **message** est une **communication d'un objet** vers un **autre objet**. La **réception d'un message** est **considérée** par **l'objet récepteur** comme un **événement qu'il faut traiter (ou pas)**. **Plusieurs types de messages** existent, les

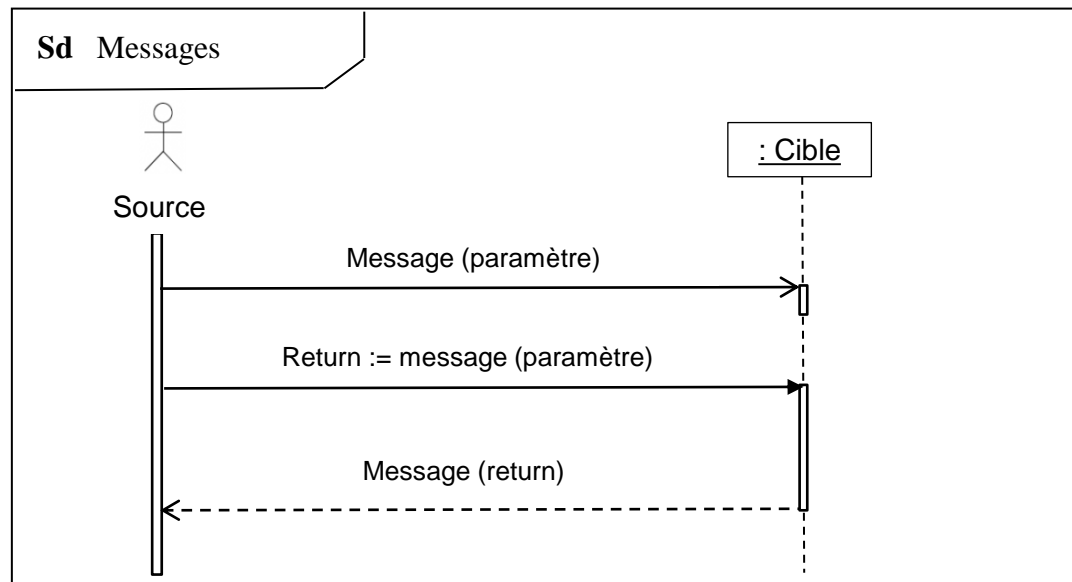
plus communs sont :

# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

1. L'envoi d'un signal ;
2. L'invocation d'une opération (appel de méthode) ;
3. La création ou la destruction d'une instance.

Les messages sont généralement divisés en message synchrone, message asynchrone et message de retour. Comme illustré dans la Figure suivante :

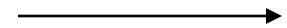
**Exemple** : Type de messages



# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

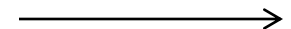
## 1. Message synchrone

L'expéditeur du message **passé le contrôle** au destinataire du message, puis **arrête l'activité**, en attendant que le destinataire du message **abandonne** ou **reprenne le contrôle**. Utilisé pour **exprimer la signification de la synchronisation**. La **flèche avec extrémité pleine** symbolise ce type de message.



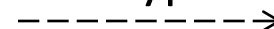
## 2. Message asynchrone

L'expéditeur du message **transmet le signal** au destinataire du message via le message, puis **poursuit ses propres activités sans attendre** que le destinataire **renvoie le message** ou le **contrôle**. Le destinataire et l'expéditeur des messages asynchrones fonctionnent **simultanément**. La **flèche avec une extrémité non pleine** qui symbolise ce type de message.



## 3. Message de retour

Le message de retour **signifie le retour de l'appel de procédure**. Une **flèche en pointillé avec une extrémité non pleine** qui symbolise ce type de message.

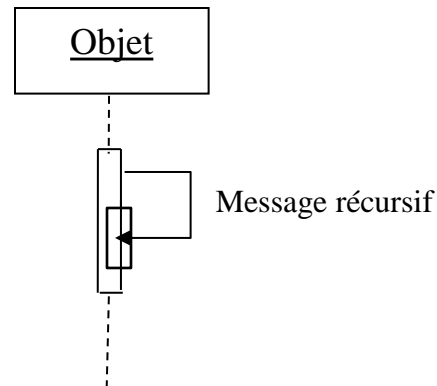


# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

## Message récursif

L'envoi de **messages récursifs** se **représente** par un **dédoubllement** de la **bande d'activation**. Dans ce cas, l'objet **apparaît** alors comme s'il était **actif plusieurs fois**.

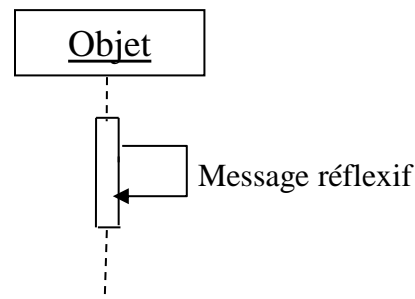
Exemple :



## Message réflexif

Il est **possible** de montrer un **objet** qui **s'envoie un message à lui-même** à l'aide d'une **flèche qui boucle**. Dans ce cas, la flèche **commence et se termine** sur la **ligne de vie du même objet**.

Exemple :

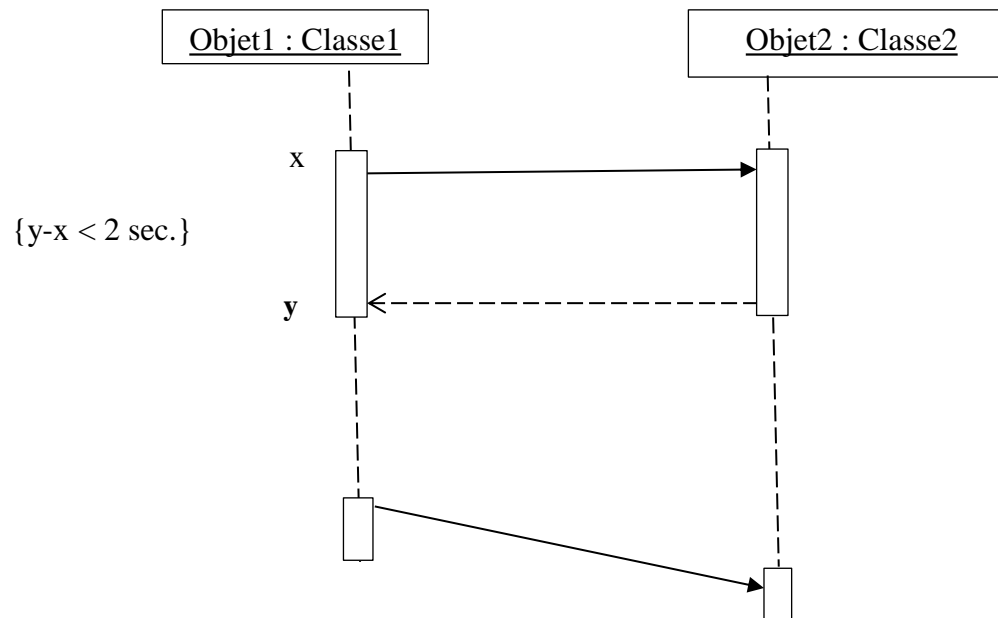


# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

## Contraintes temporelles

Des **repères temporels** avec des **contraintes** peuvent être placés le long de la **ligne de vie**. Un **message** avec un **temps** de propagation **non négligeable** peut être représenté sous la forme d'un **trait oblique** ou en l'**écrivant explicitement**.

**Exemple** : représentation de contrainte temporelle.

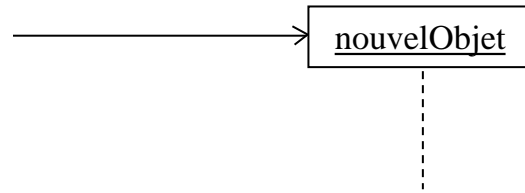


# Diagrammes de séquence : Les diagrammes de séquence décrivant les interactions entre objets

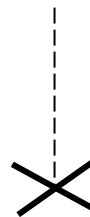
## Création et destruction d'objets

Certains objets vivent pendant tout le diagramme, d'autres sont créés et/ou meurent pendant la séquence.

**A. Création d'objets** : Si un objet est créé par une opération, celui-ci n'apparaît qu'au moment où il est créé. La création d'un objet est concrétisée par une flèche qui pointe sur le sommet d'une ligne de vie.



**B. Destruction d'objets** : Si un objet est détruit par une opération, la destruction se représente par une croix « X » qui marque la fin de la ligne de vie de l'objet.



# Diagrammes de séquence

## Fragment d'interaction

Un **fragment d'interaction** est **constitué** d'un ensemble **d'interactions** sur lequel un **opérateur** est **appliqué**. La **représentation générale** d'un **fragment d'interaction** prend la forme d'un **diagramme de séquence**, avec une **indication** située dans le **coin supérieur** gauche **indiquant** le **nom de l'opérateur**.

Les **opérandes** d'un **opérateur d'interaction** sont **séparés** par une **ligne en pointillés**. Les conditions de **choix** des **opérandes** sont **exprimées** à l'aide d'**expressions booléennes** placées **entre crochets**. La liste suivante classe les **opérateurs d'interaction** en fonction de leurs **rôles** :

- **Choix et boucle** : alternative, option, break, loop
- **Contrôle d'envoi en parallèle de messages** : parallel, critical region
- **Contrôle d'envoi de messages** : ignore, consider, assertion, negative
- **Ordre d'envoi des messages** : weak sequencing, strict sequencing
- **Référencement** : ref

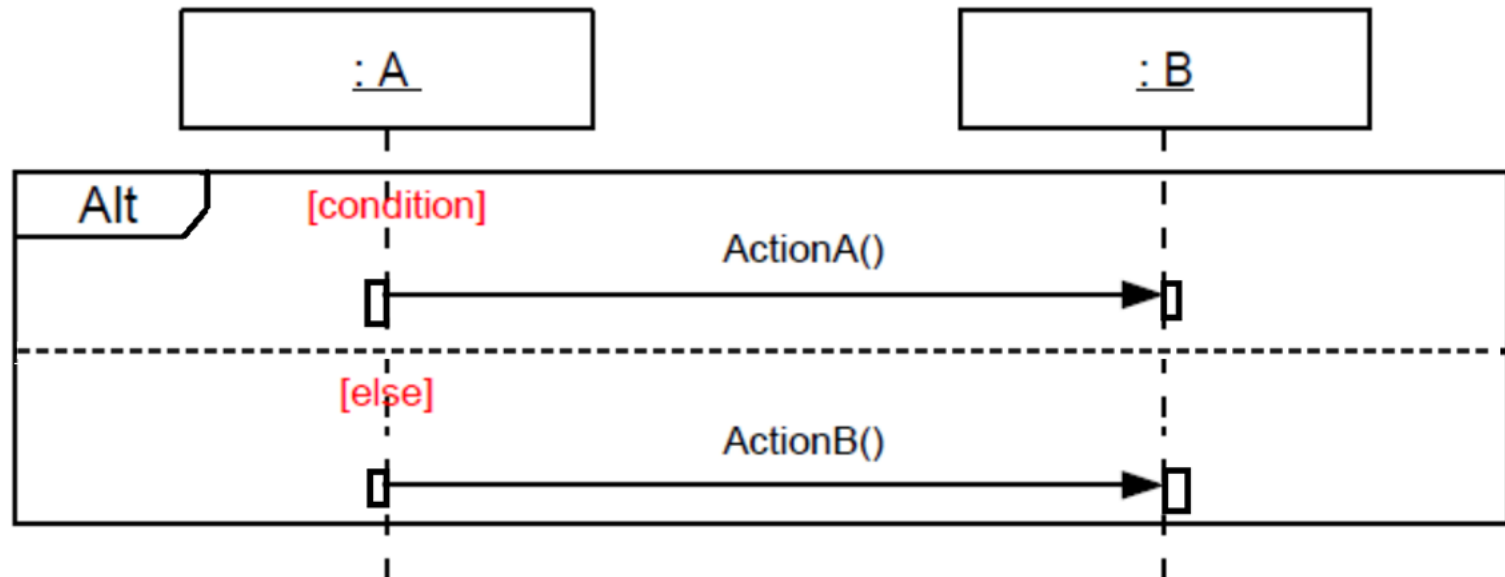
# Diagrammes de séquence

## Fragment d'interaction (suite)

**Opérateur alt** : il correspond à une instruction de test comportant une ou plusieurs alternatives possibles. Il est également possible d'utiliser des clauses de type sinon (else).

### Formalisme :

L'opérateur **alt** se représente dans un fragment possédant au moins deux parties séparées par des pointillés.



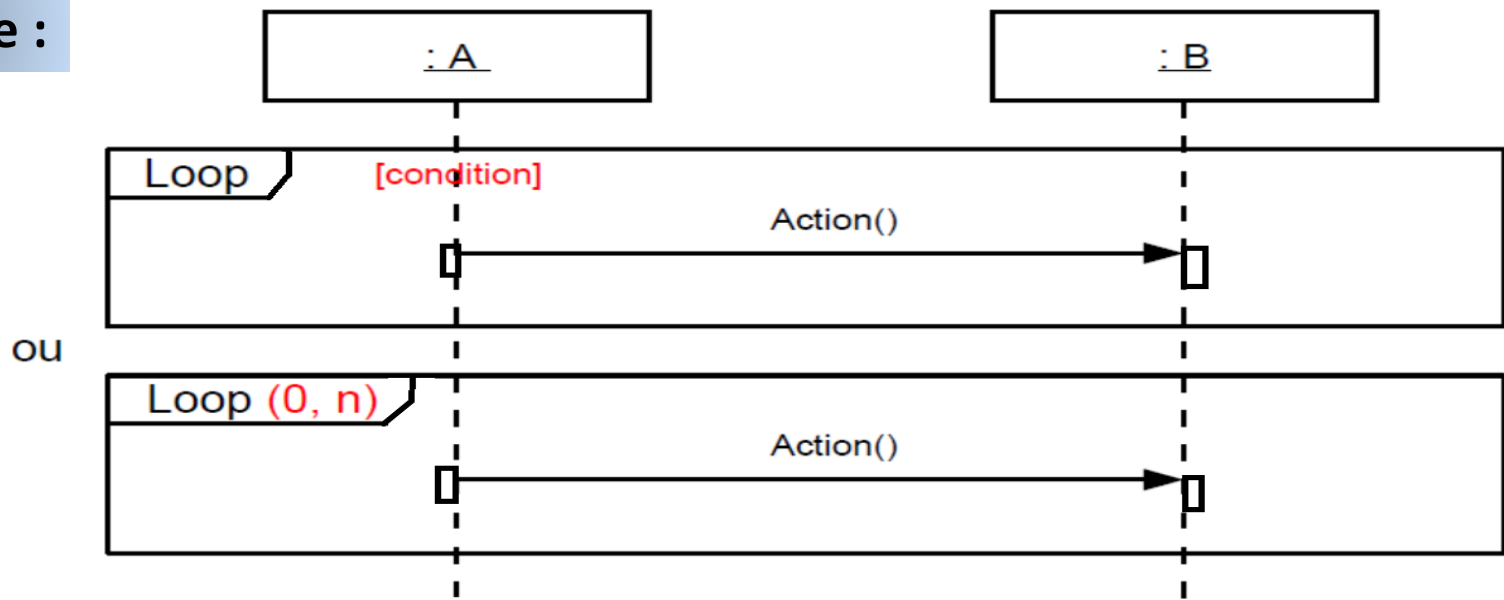


# Diagrammes de séquence

## Fragment d'interaction (suite)

**Opérateur loop** : Cet opérateur correspond à une **instruction de boucle** qui permet **d'exécuter une séquence d'interaction** tant qu'une **condition est satisfaite**. Il est également **envisageable** d'utiliser une **condition** pour **définir un nombre minimum et maximum d'itérations** de la **boucle** en utilisant la **syntaxe** suivante : **boucle min, max**. Ainsi, la **boucle** sera **exécutée au moins min fois** et au **maximum max fois**. On **peut** également **combinaison** cette option **min/max** avec une **condition** associée à la **boucle**.

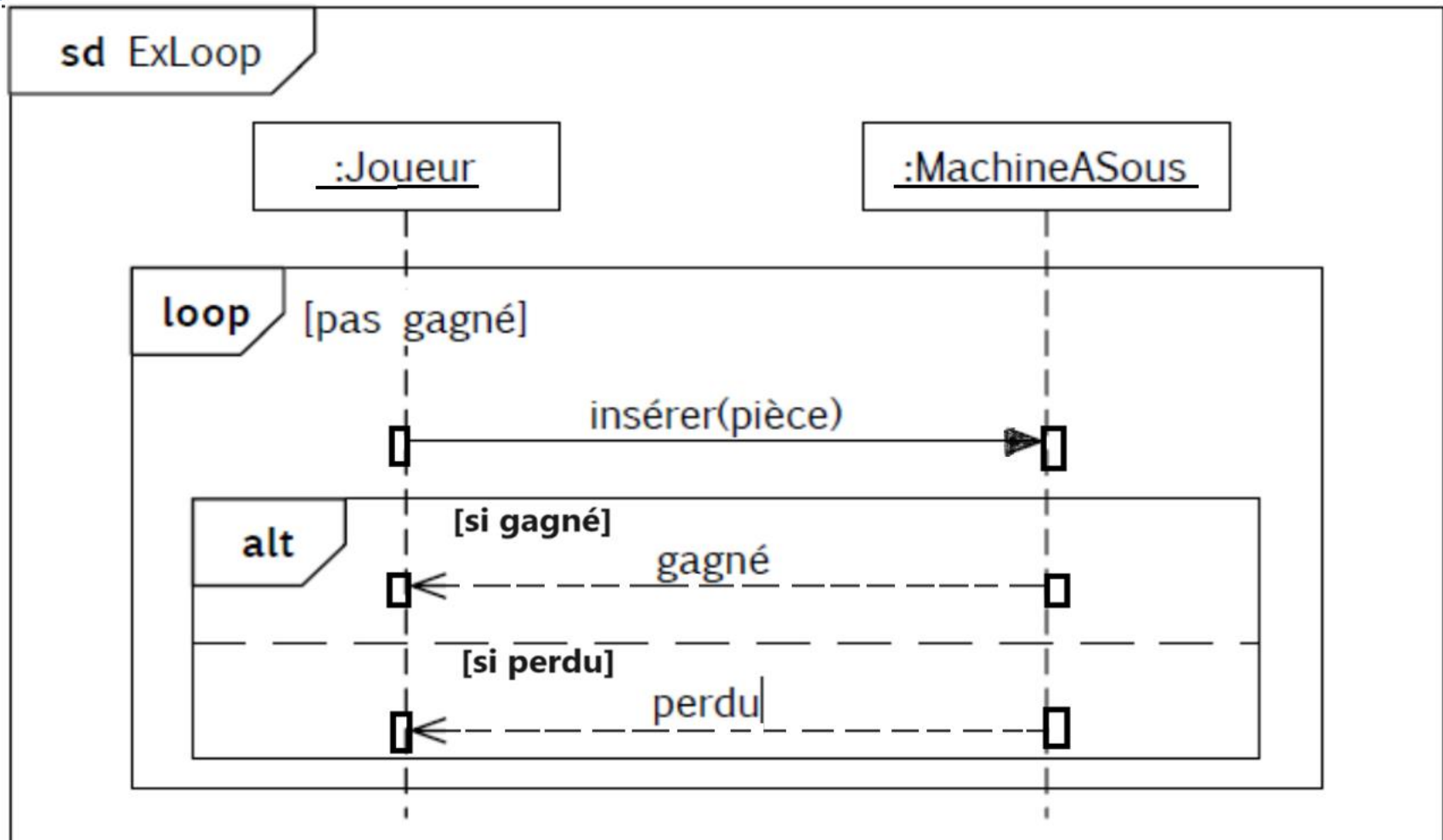
### Formalisme :



# Diagrammes de séquence

## Fragment d'interaction (suite)

Exemple : fragments combinés

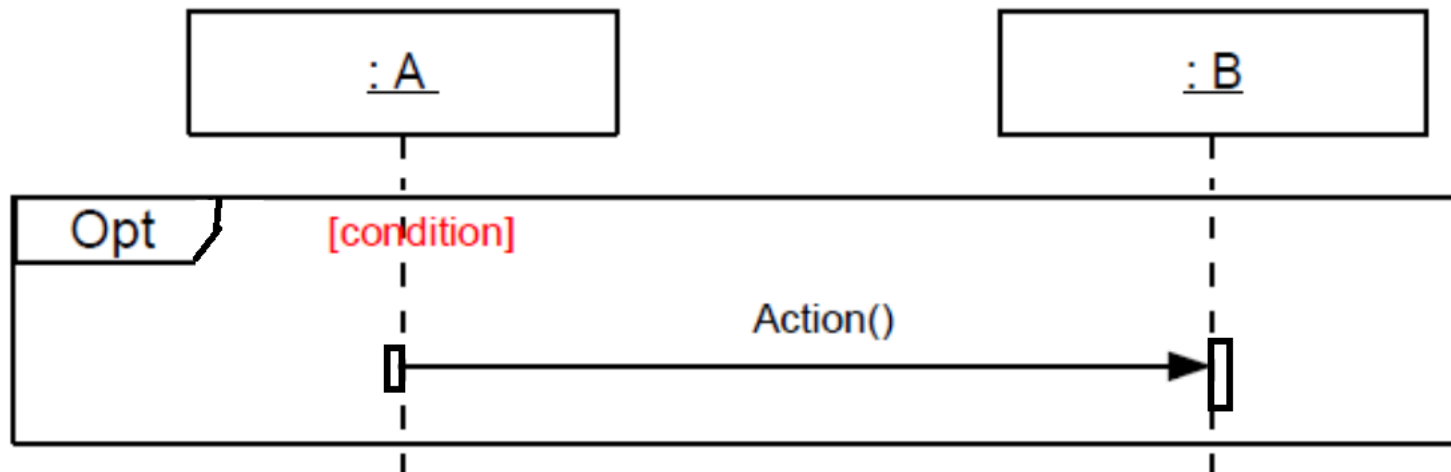


# Diagrammes de séquence

## Fragment d'interaction (suite)

**Opérateur opt** : cet opérateur permet de représenter une instruction de test sans alternative (sinon). L'opérateur **opt** est représenté dans un fragment qui ne comporte qu'une seule partie.

Formalisme :



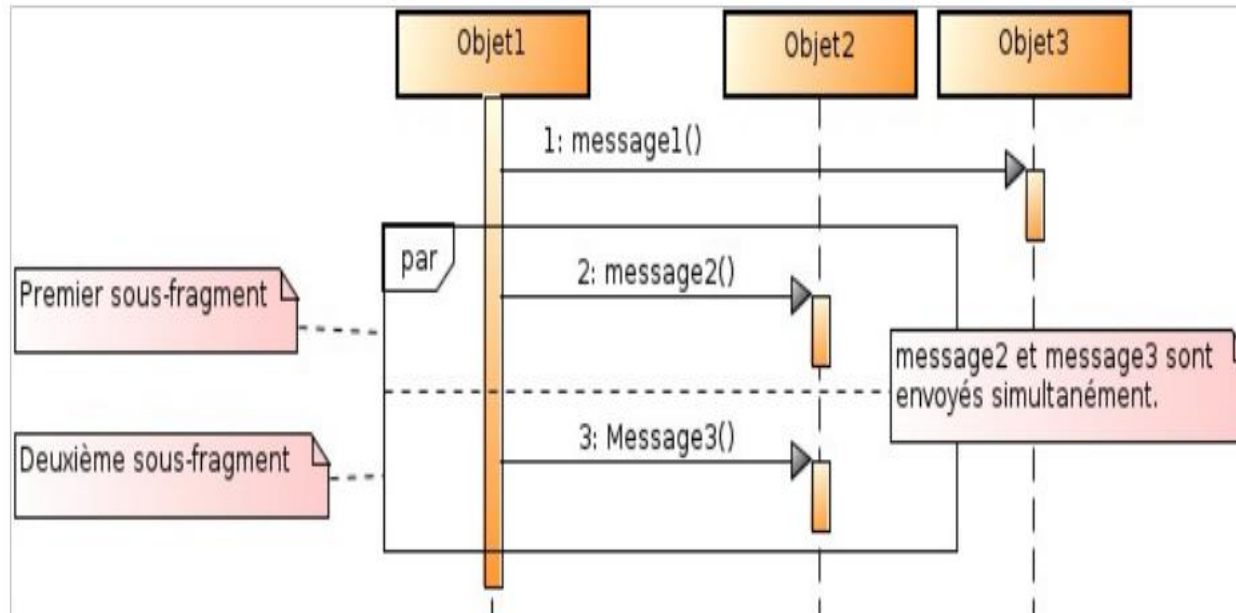
# Diagrammes de séquence

## Fragment d'interaction (suite)

**Opérateur par** : cet opérateur permet de représenter deux séries d'interactions qui se déroulent simultanément ou en parallèle.

L'opérateur **par** se représente dans un fragment possédant deux parties séparées par une ligne en pointillé.

### Exemple :



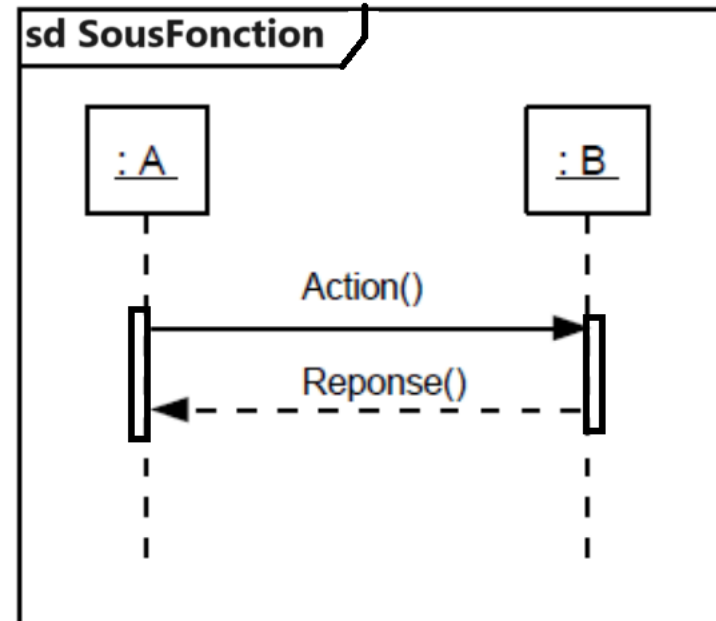
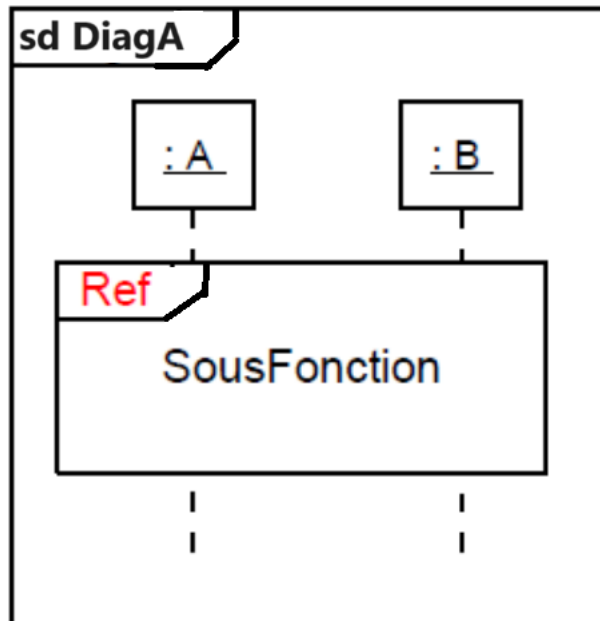
# Diagrammes de séquence

## Fragment d'interaction (suite)

**Opérateur ref** : Cet opérateur permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous diagramme de séquence.

### Formalisme :

L'exemple suivant **montre** que l'on fait **appel** à un fragment « **SousFonction** » qui est **décrit** par ailleurs :



# Diagrammes de communication

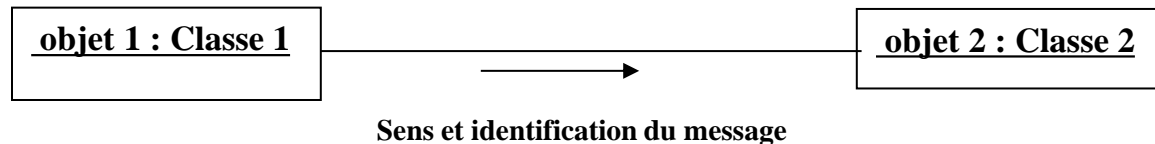
- ❑ Le **diagramme de communication** montre l'**interaction** entre les **objets** ou les **rôles** associés à la **ligne de vie** et les **messages** directement **transmis** par la **ligne de vie**.
- ❑ Dans les **versions antérieures d'UML**, le **diagramme de communication** s'appelait **diagramme de collaboration** et la notation était **différente**.
- ❑ Un **diagramme de communication** est un **diagramme interactif** qui peut être utilisé pour **explorer** le **comportement dynamique** d'un **système** ou d'une **application logicielle**.
- ❑ Il **fournit** les mêmes **informations** que le **diagramme de séquence** via une autre **vue**.
- ❑ Le **diagramme de séquence** est destiné à **décrire** la **séquence des messages** après une **période de temps**, tandis que le **diagramme de communication** est30

# Diagrammes de communication

destiné à décrire la **structure des messages** passés entre les **objets participant** à l'interaction.

- Ces **diagrammes** illustrent le **flux de messages** entre les **objets** et les **relations implicites** entre les **classes**.

Le lien entre les objets est symbolisé par un **trait matérialisant** le support des **messages échangés**.



# Diagrammes de communication

- ❑ Ce type de **diagramme** constitue une **autre représentation** des **interactions** que celle du **diagramme de séquence**. En effet, le **diagramme de communication** met plus **l'accent** sur **l'aspect spatial** des **échanges** que **l'aspect temporel**.
- ❑ Chaque **participant** à un **échange** de message **correspondant** à une **ligne de vie** dans le **diagramme de séquence** se **représente** sous forme d'un **rôle** dans le **diagramme de communication**.



# Diagrammes de communication

L'exemple suivant donne le **formalisme** de base du **diagramme de communication**.

Formalisme et exemple :

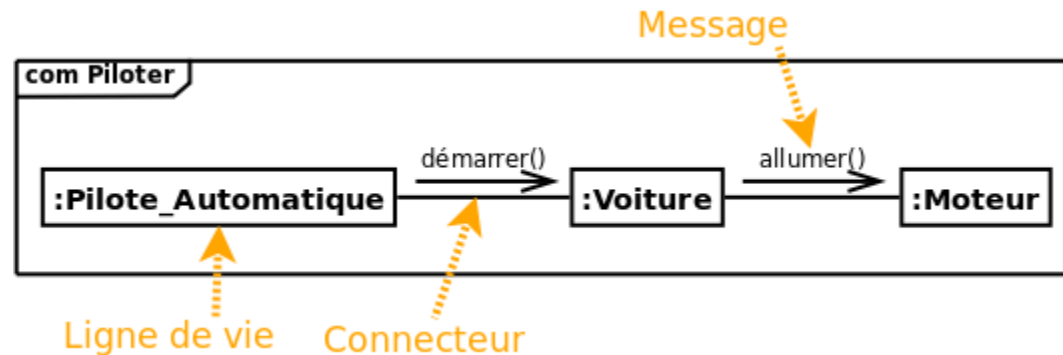


Diagramme de communication d'un système de pilotage.

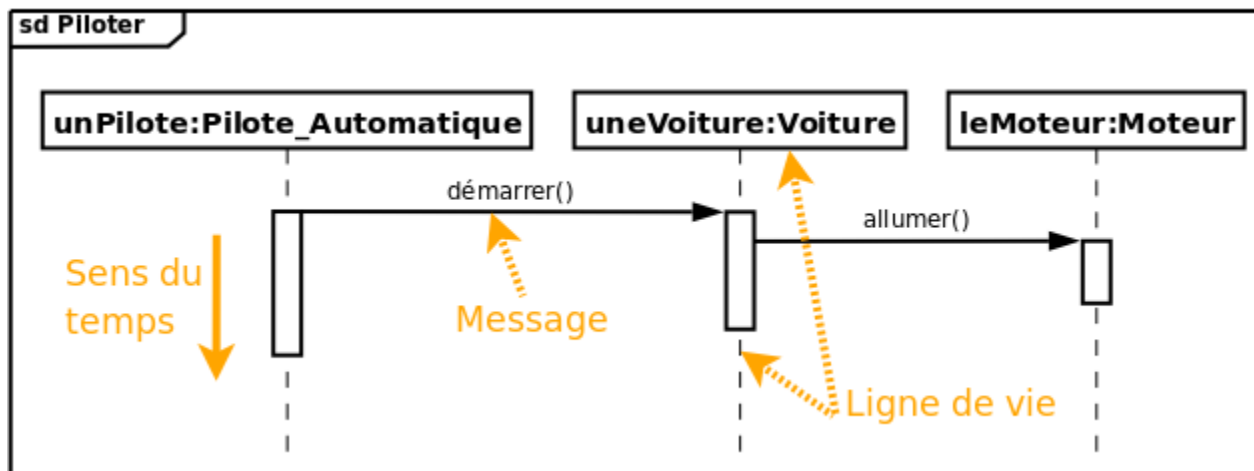


Diagramme de séquence d'un système de pilotage.

# Fin du Chapitre 5. Partie 1

