

# CHAPITRE 6

## Diagrammes d'interaction

# Interactions

- Description du comportement, incluant un ensemble de messages échangés entre un ensemble d'objets à l'intérieur d'un contexte particulier afin d'accomplir un but spécifique,
- Modéliser **l'aspect dynamique** des collaborations,
- Modéliser le flux de contrôle dans une opération, une classe, une composante, un cas d'utilisation ou un système dans son ensemble,
- Représenter les messages échangés entre des objets, acteurs, etc.
  - Invocation (appel) d'une opération
  - Envoi d'un signal
  - Création et destruction d'objets

# Objets et rôles

Les objets qui participent à une interaction sont :

- Des objets concrets

ex.: `p` est une `personne` en particulier

- Des objets prototypes

ex.: `p` est une `personne` en général

- Des acteurs

ex.: `p` est un `acteur` interagissant avec le système


- Un système ou un sous-système

ex.: `s` est un `sous-système` avec lequel des acteurs interagissent

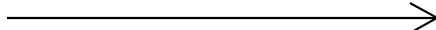

# Liens

- Connexion **sémantique** entre objets.
- Instance d'une association.
- Lorsqu'il y a un **lien** entre deux objets, des messages peuvent être envoyés et reçus entre ces objets.


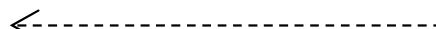
# Types de messages échangés

**Appeler** 

Envoyer un *message synchrone*, correspondant généralement à l'*invocation d'une opération* d'un objet. (En outre, un objet peut s'auto-envoyer un message.)

**Envoyer**  (anciennement )

Envoyer un *message asynchrone*, soit un *signal* ou encore l'*invocation asynchrone d'une opération* d'un objet.

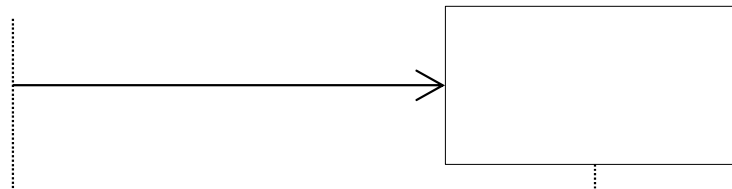
**Retourner**  ou 

Retourner une valeur à l'appelant.

**Créer**

Créer un objet

(ou parfois un message avec un stéréotype <<create>>)



# Deux types de diagrammes d'interaction

Les deux types de diagrammes sont strictement équivalents l'un à l'autre, ils contiennent la même information. Toutefois, ils ne présentent pas l'information de la même façon.

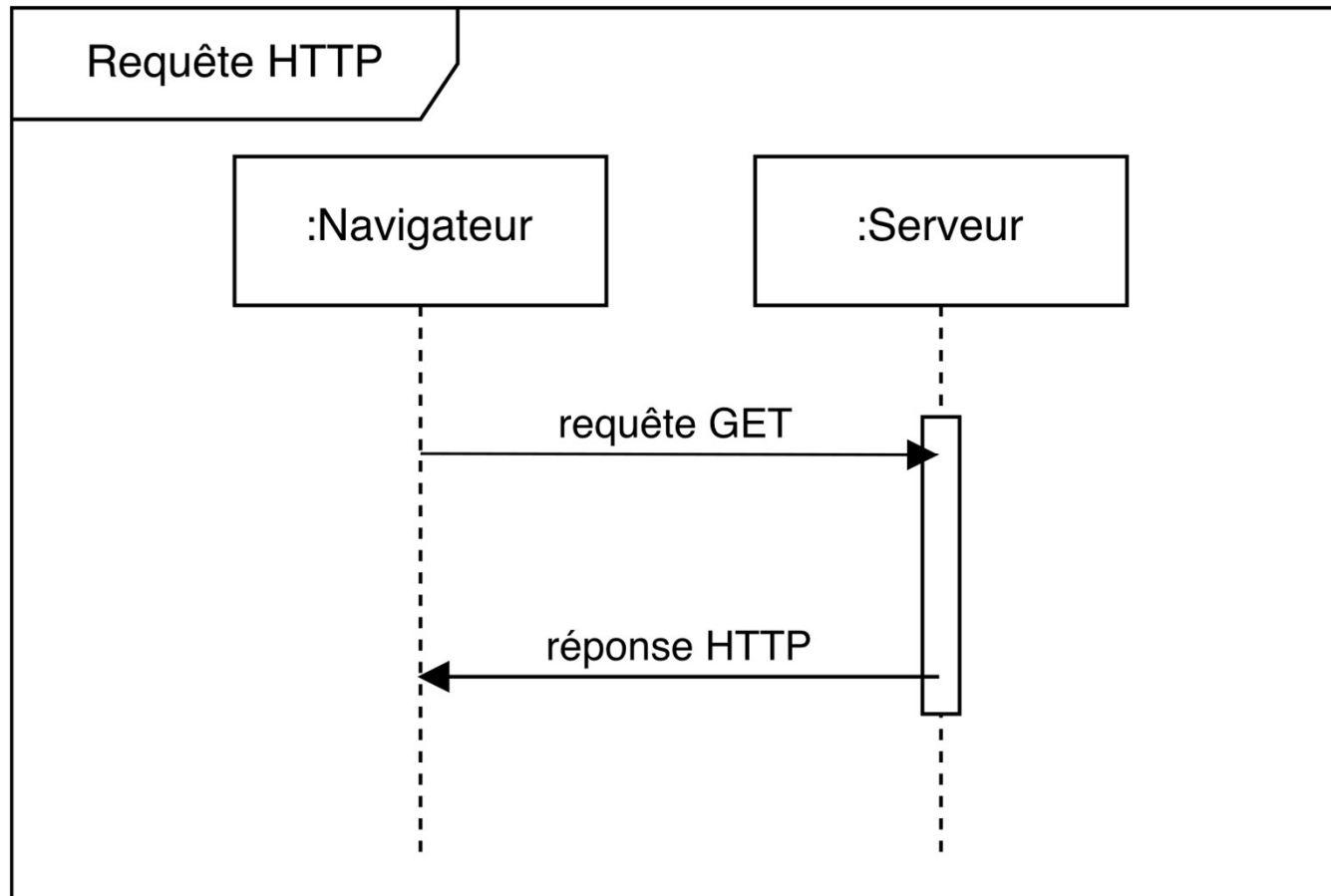
### ➤ Diagramme de séquence:

Se concentre sur **l'ordre séquentiel** des messages.

### ➤ Diagramme de communication:

Se concentre sur **l'organisation structurelle** dans laquelle les objets s'envoient et reçoivent des messages.

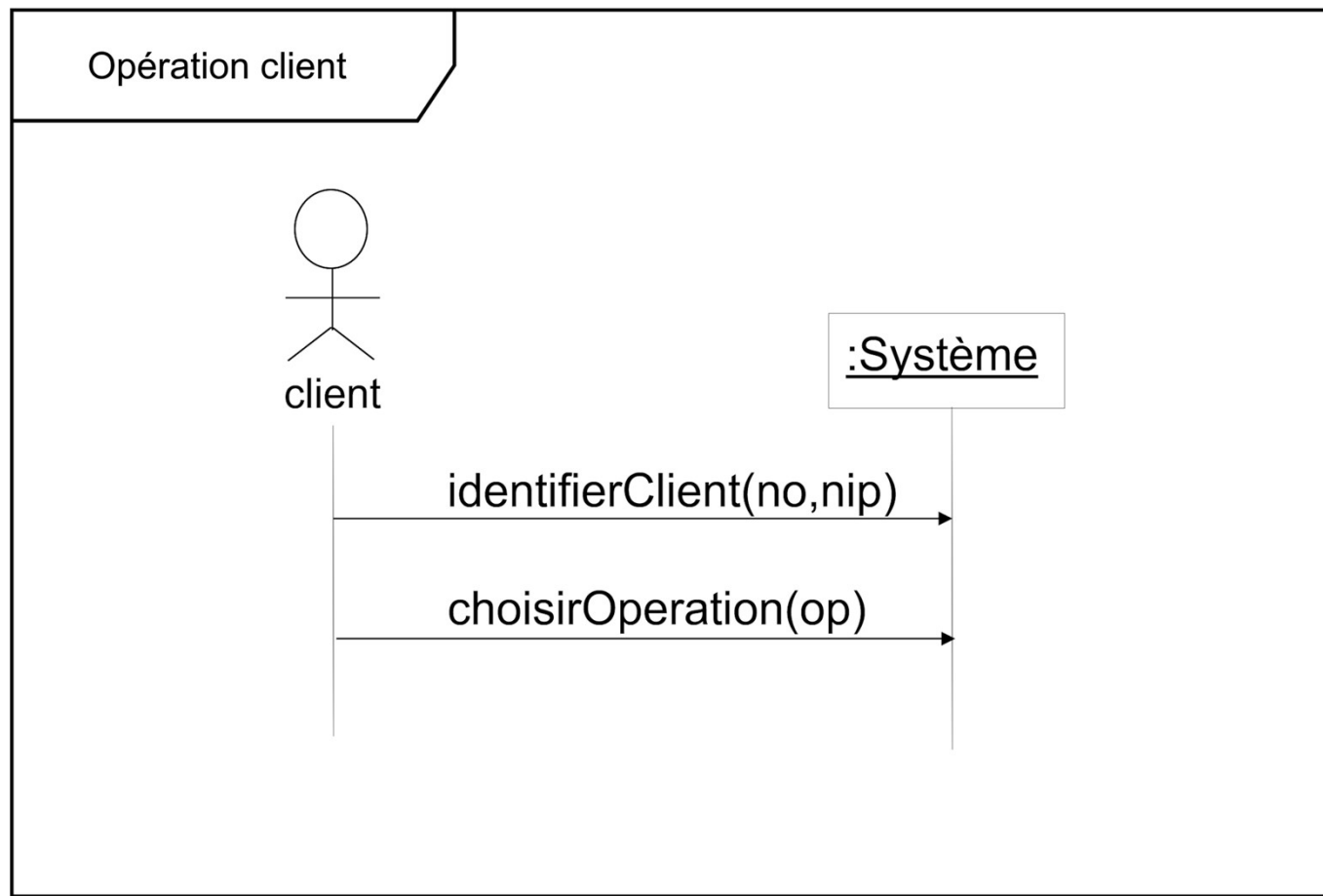
# Diagramme de séquence



Source: *Guide UML*, section « Diagrammes d'interaction »  
[https://gigl-uml.github.io/Guide\\_uml\\_polymtl/diagrammes-d-interaction/](https://gigl-uml.github.io/Guide_uml_polymtl/diagrammes-d-interaction/)

### Cas particulier: le diagramme de séquence du système

Diagramme de séquence entre un acteur et le système





# Diagramme de séquence

## Ligne de vie

Représente **l'existence** d'un objet pendant une période de temps.

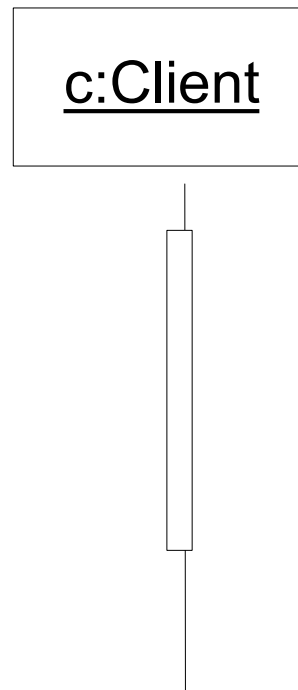


c:Client

# Diagramme de séquence

## Occurrence d'exécution (ou activation)

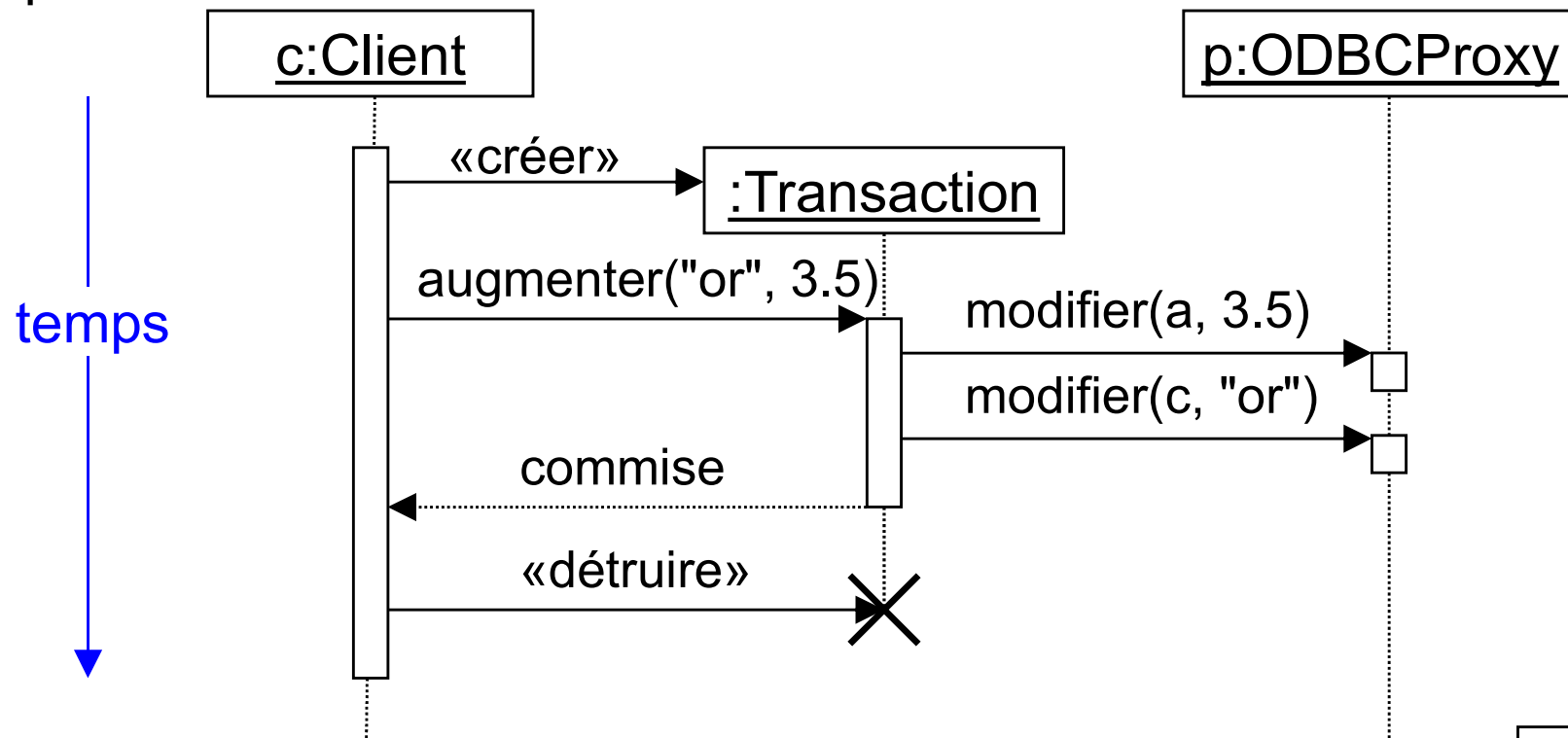
Représente le temps durant lequel un objet est actif, c'est-à-dire en train **d'exécuter une opération**.



## Diagramme de séquence

La séquence des messages est spécifiée par la position verticale des messages échangés.

Exemple:

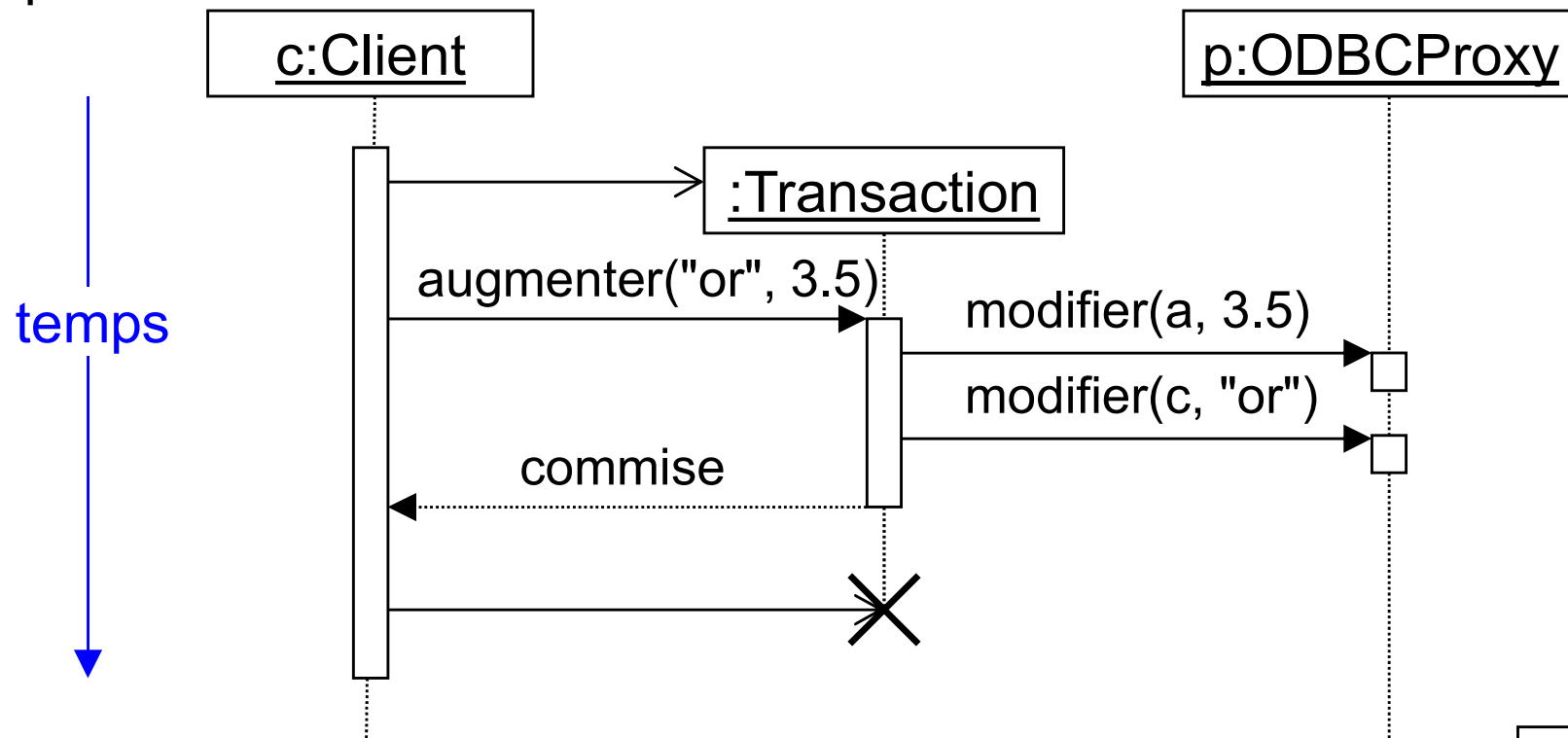


## Diagramme de séquence

Des variations de notation peuvent exister. La norme UML a été mise à jour au fil des ans.

La séquence des messages est spécifiée par la position verticale des messages échangés.

Exemple:



# Diagramme de séquence

## Synchronisation

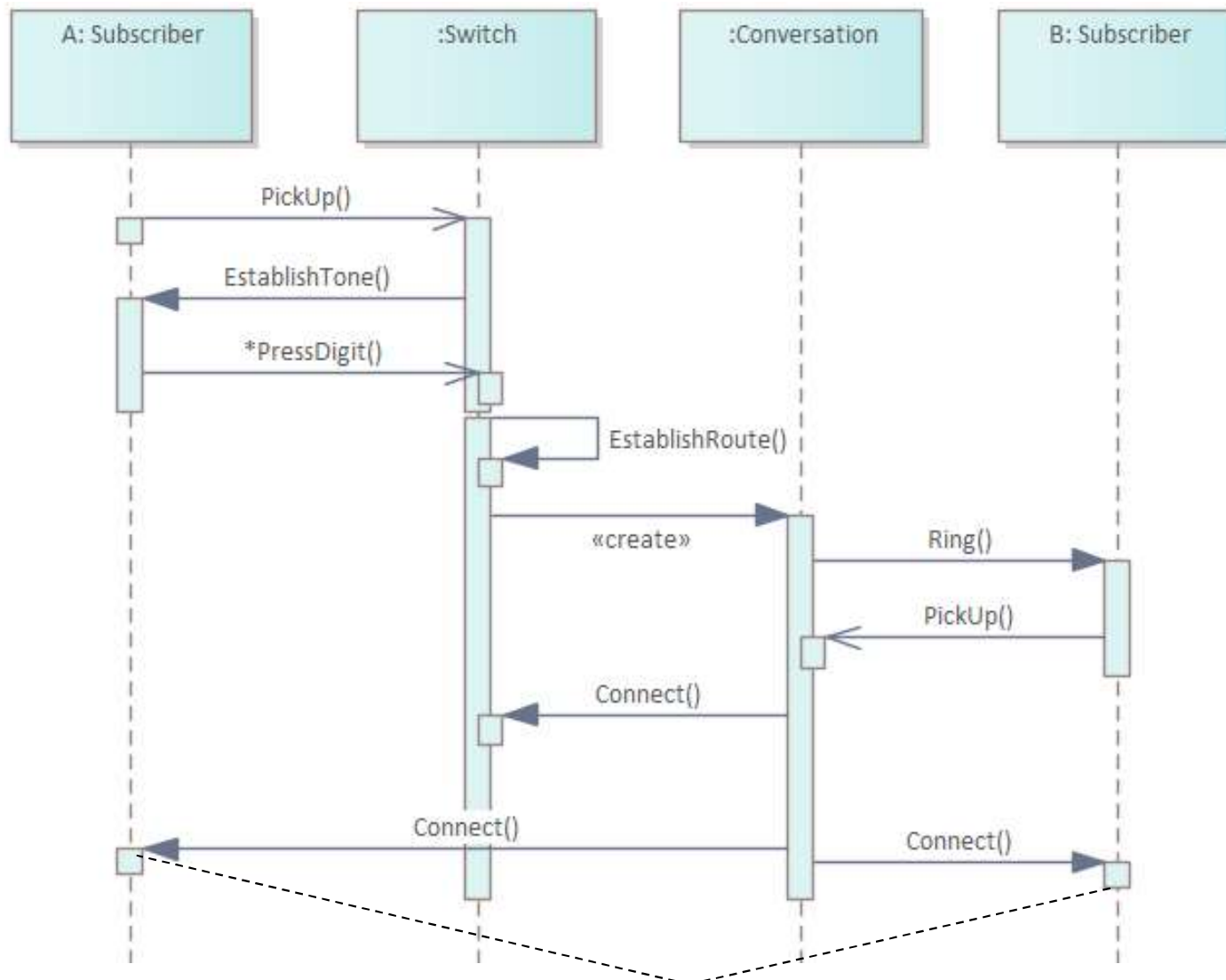
- Mode par défaut des messages.
- Le contrôle est passé de l'appelant à l'appelé; l'opération de l'appelant est suspendue (*wait*).
- Le contrôle est rendu à l'appelant à la fin de l'opération appelée; l'appelant reprend l'exécution de l'opération en cours.

# Diagramme de séquence

## Concurrence

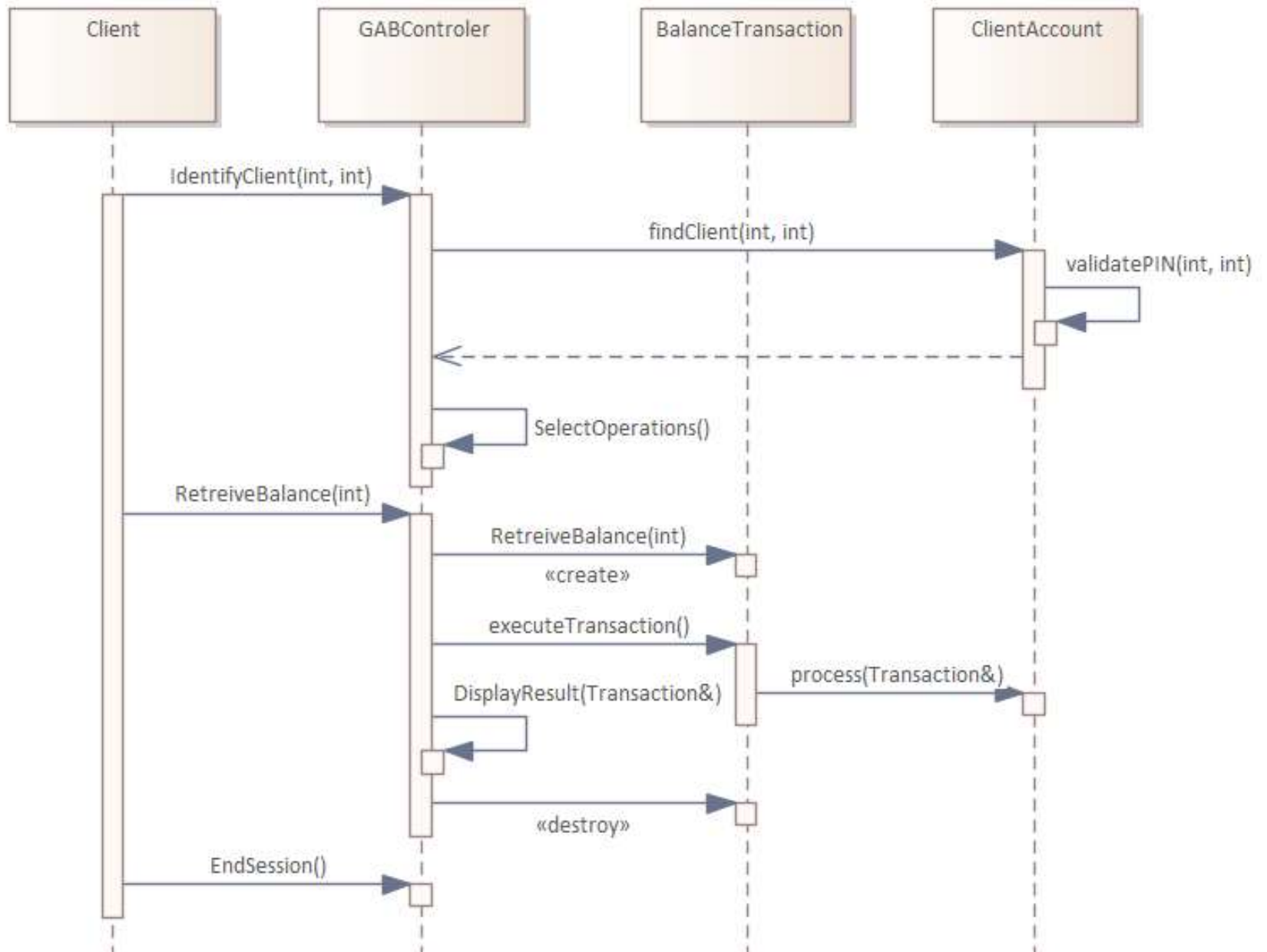
- Spécifiée par l'utilisation de messages d'envoi.
- Un signal est envoyé à un objet; l'objet envoyeur continue son opération.

## Diagramme de séquence



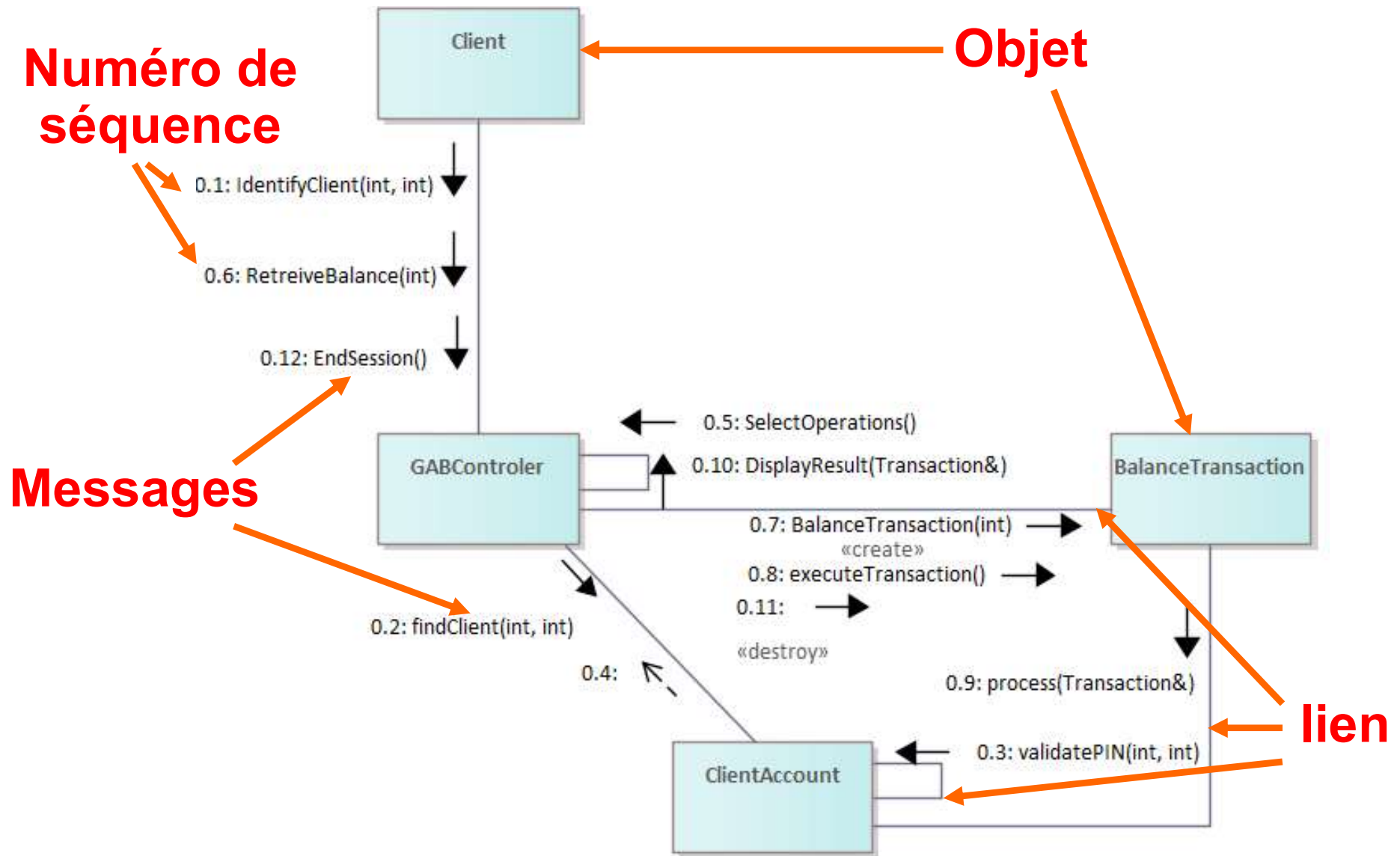
Les abonnés a et b peuvent échanger de l'information une fois connectés

## Diagramme de séquence





## Diagramme de communication



# Diagramme de séquence

## Structures de contrôle

Le diagramme de séquence peut inclure un certain nombre de structures:

- **Branchements**

- Tests

- **Répétitions**

- Boucles

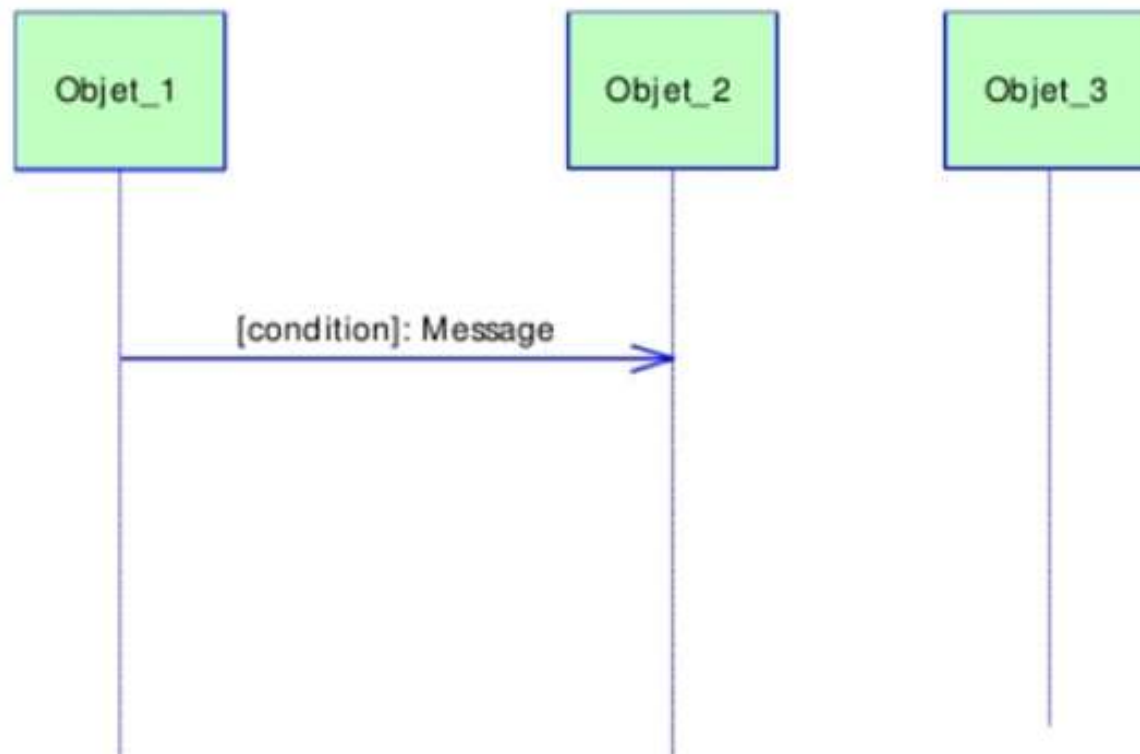
- Itérations

# Diagramme de séquence

Cette notation était valide  
sous la norme UML 1.x.

## Les tests (branchements)

- La condition précède le message et elle est délimitée par des crochets

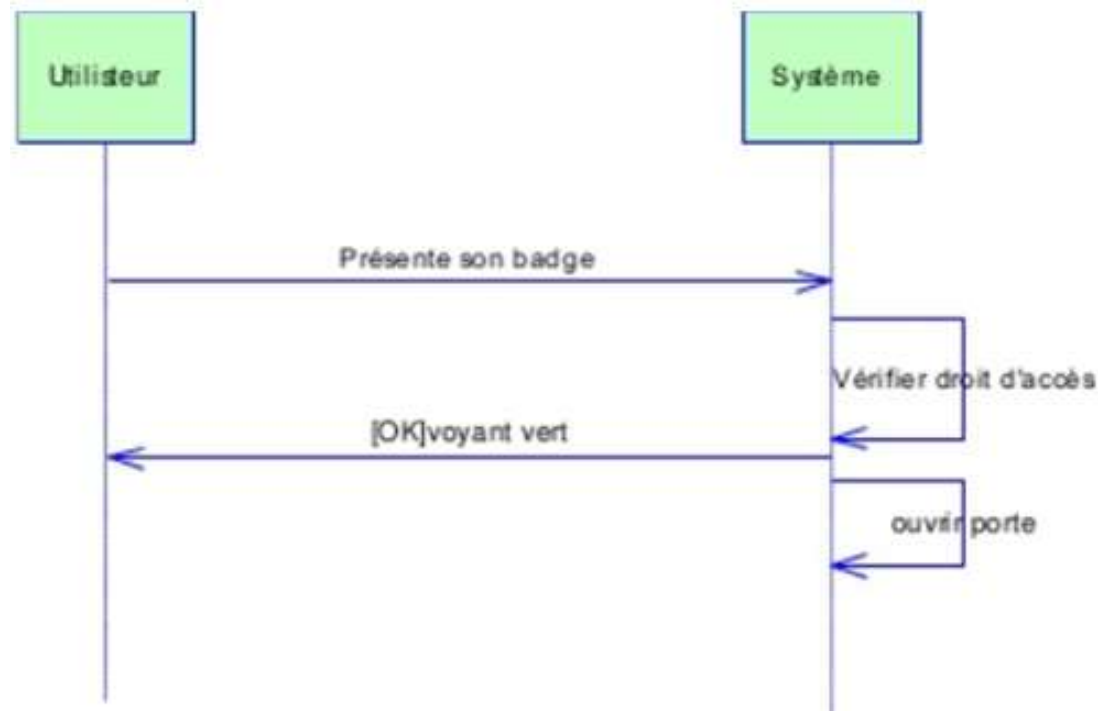


# Diagramme de séquence

Cette notation était valide  
sous la norme UML 1.x.

## Les tests (branchements): exemple

- Pour accéder au laboratoire, l'utilisateur doit présenter sa carte d'identité. S'il a droit d'accès, un voyant vert s'allume et la porte s'ouvre

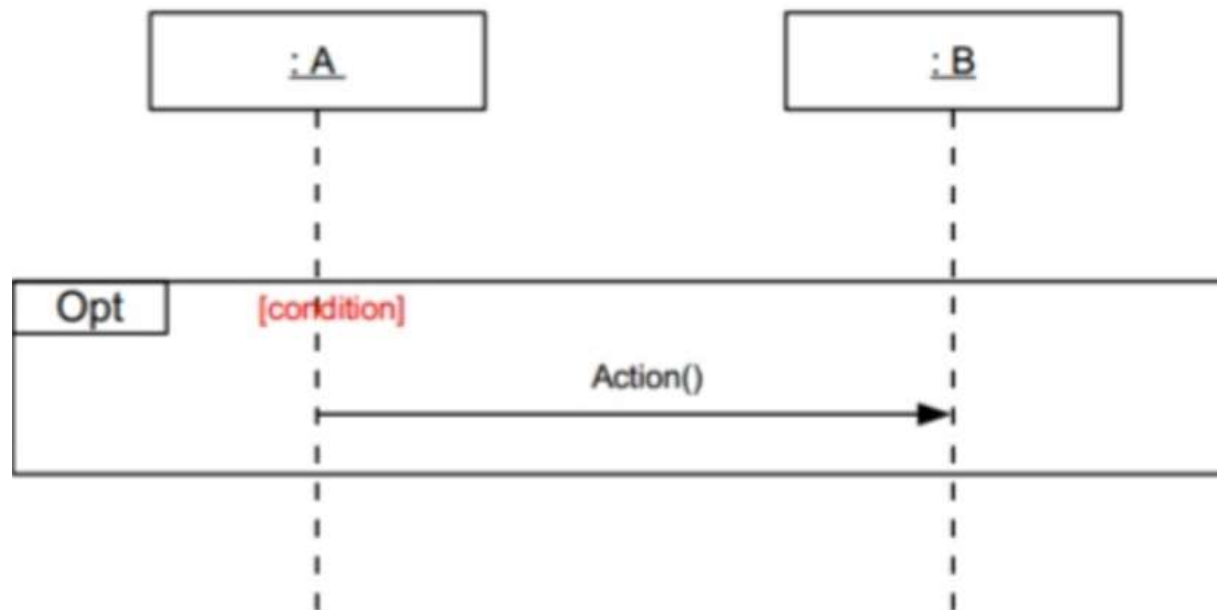


# Diagramme de séquence

## OPT

- Le fragment est parcouru si une condition est vérifiée

Équivalente à  
*Si Condition alors Action*



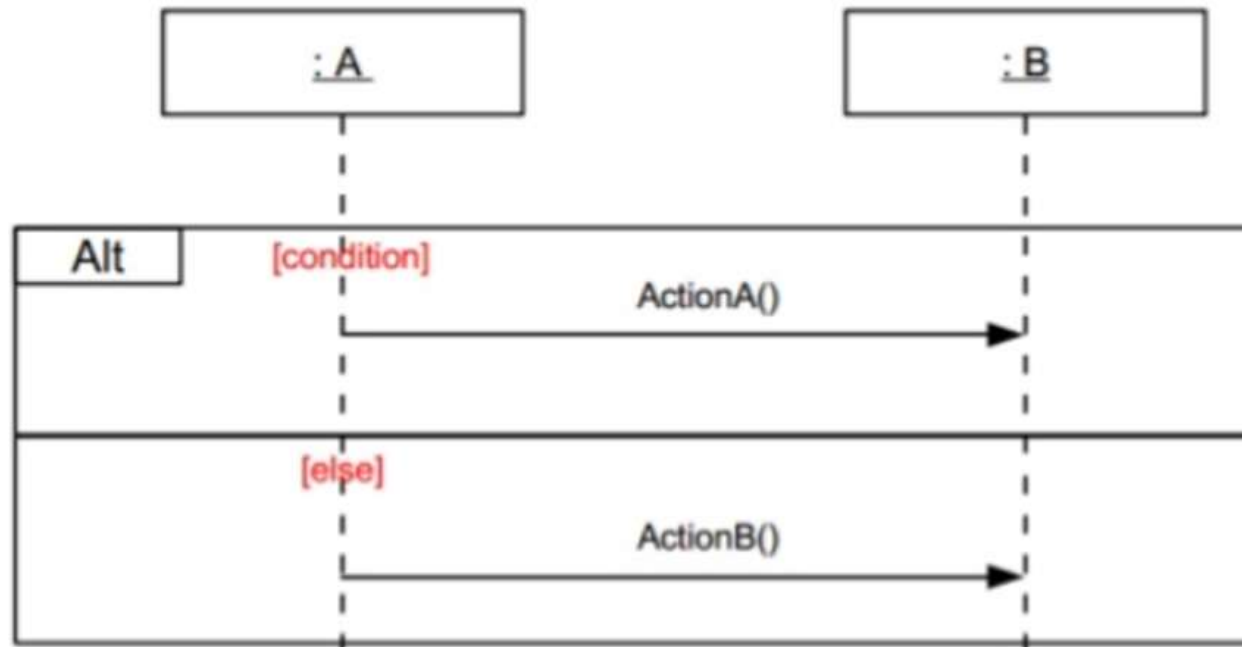
# Diagramme de séquence

## ALT

- Deux fragments à parcourir selon la valeur de la condition

Équivalente à

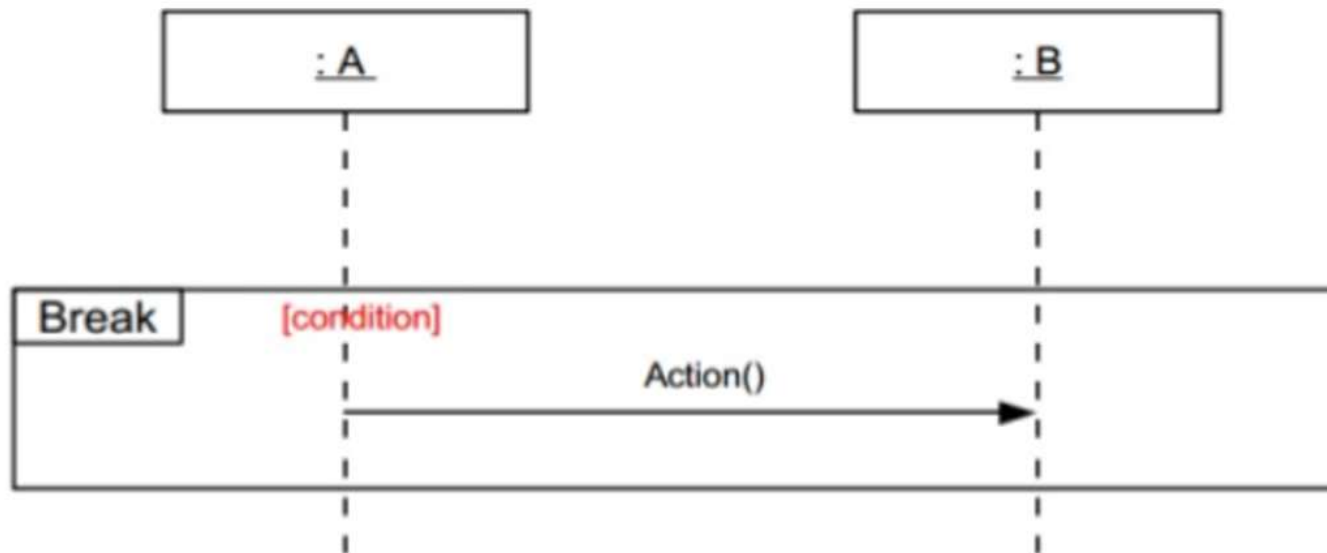
*Si Condition alors ActionA si non ActionB*



# Diagramme de séquence

## BREAK

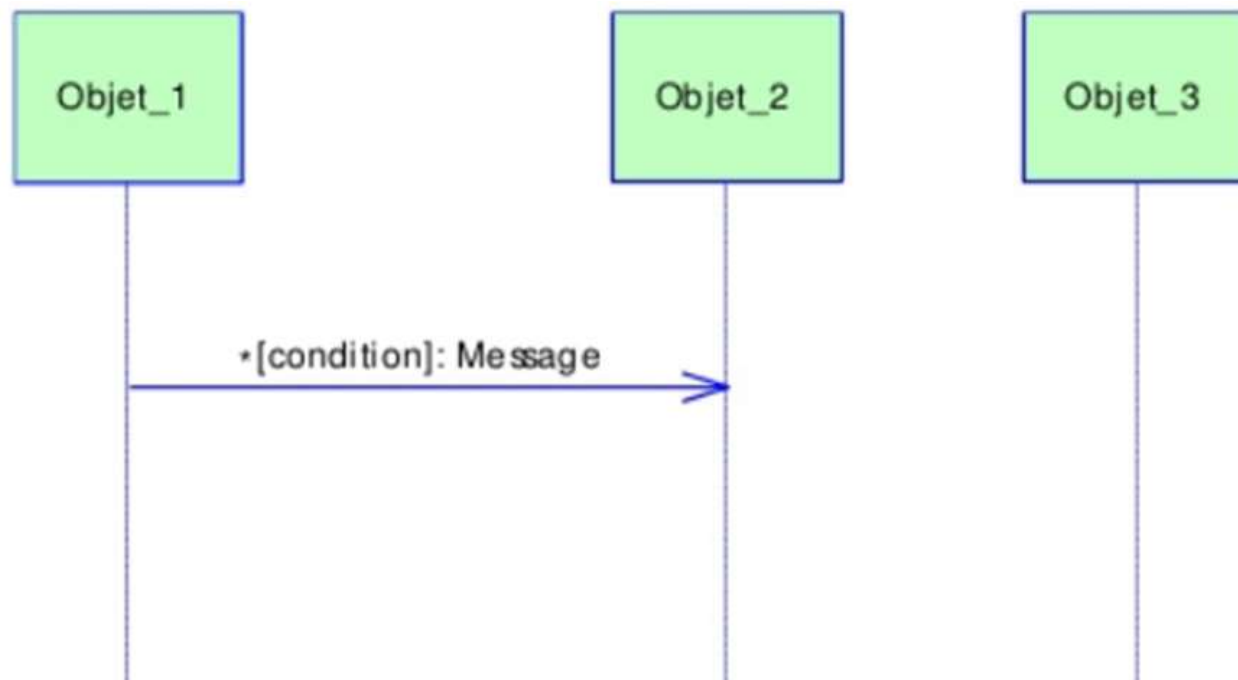
- Le fragment est exécuté et met fin au fragment englobant si une condition est vérifiée



# Diagramme de séquence

## Les boucles (répétitions)

- La boucle se note comme le test, mais la condition est précédée d'un astérisque.





# Diagramme de séquence

## LOOP

- Le fragment est répété tant qu'une condition est vérifiée

Équivalente à

Tant que *Condition* faire *Actions*

