



# Chapitre 4:

# Diagramme de séquence



**Dr. JAOUADI Maryem**

jaouadimaryem@gmail.com

*2<sup>ème</sup> année SI*

# Objectifs



- Comprendre l'utilité d'un diagramme de séquence.
- Connaitre les éléments d'un diagramme de séquence.
- Différencier entre les différents composants d'un diagramme de séquence.
- Savoir passer d'un diagramme de séquence vers un diagramme de collaboration.



# Plan



Introduction au diagramme de séquence

Concepts d'un diagramme de séquence

Exemple

Activité(s)



# Introduction au diagramme de séquence



- La modélisation dynamique d'un système consiste à décrire son comportement lors de sa réaction à son environnement.
- Elle se fait à l'aide des diagrammes suivants :
  - Diagramme des cas d'utilisation
  - **Diagramme de séquences**
  - Diagramme de collaboration
  - Diagramme états-transitions
  - Diagramme d'activités

# Introduction au diagramme de séquence



## ■ Présentation

- Ce diagramme représente les interactions entre les objets de point de vue temporel.
- Il permet de détailler un diagramme de cas d'utilisation.
- Un diagramme de séquence décrit un scénario d'un cas d'utilisation.

# Concepts d'un diagramme de séquence (1/20)



## ▪ concepts de base

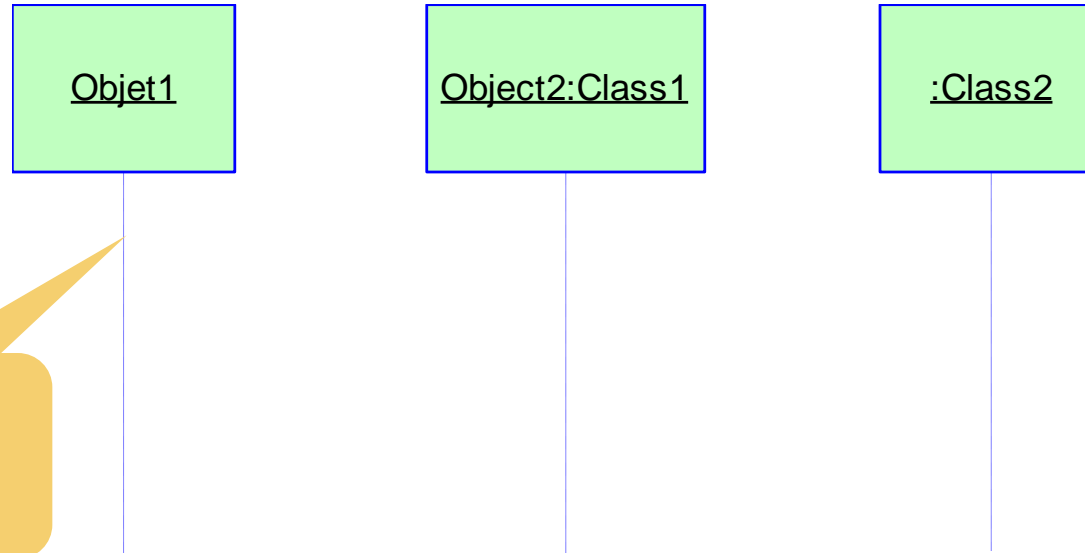
Le diagramme de séquences se base sur les concepts suivants :

- **Objet** : Description d'un objet du monde réel (instance de classe). Il peut être une personne ou une chose.
- **Message** : C'est le véhicule de la communication entre les objets.

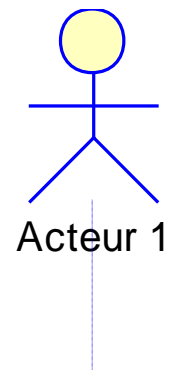


# Concepts d'un diagramme de séquence (2/20)

## ▪ Représentation d'un objet

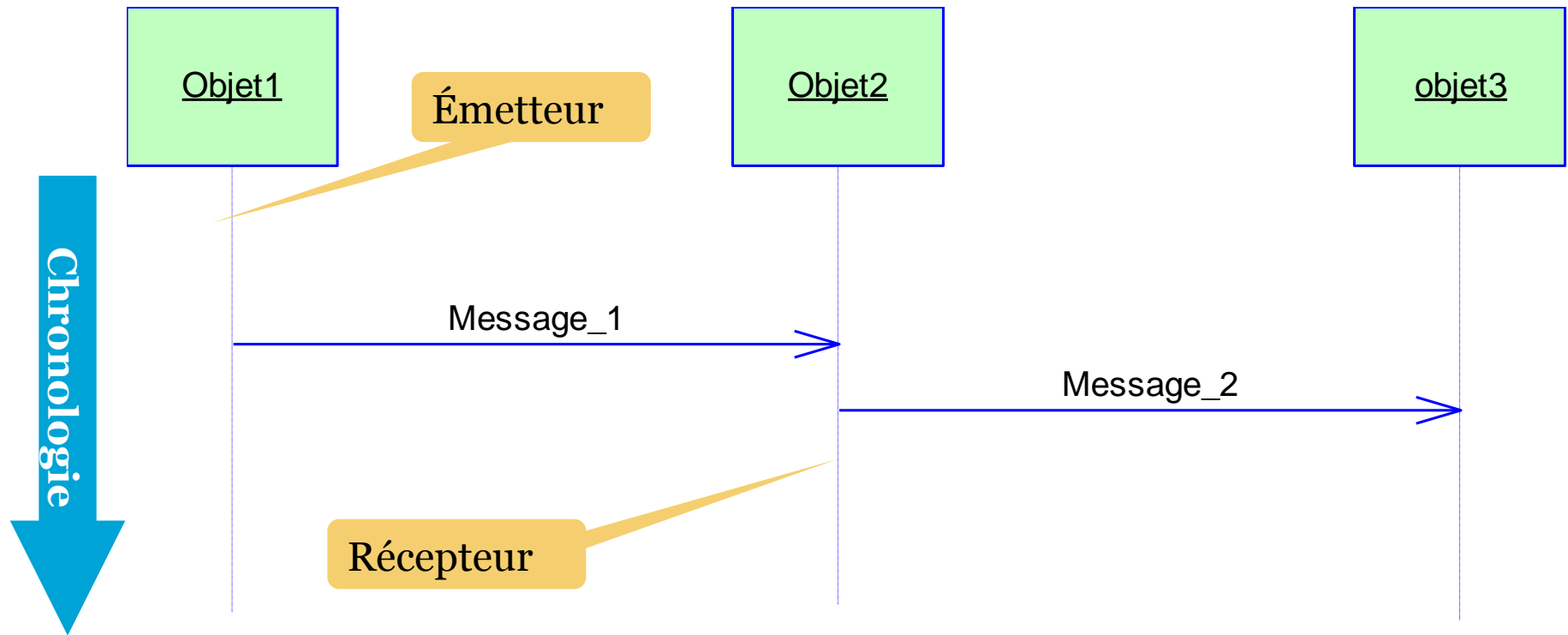


**Le premier objet d'un diagramme de séquence représente généralement un utilisateur du système. Il peut être représenté comme un acteur.**



# Concepts d'un diagramme de séquence (3/20)

## ▪ Représentation d'un message



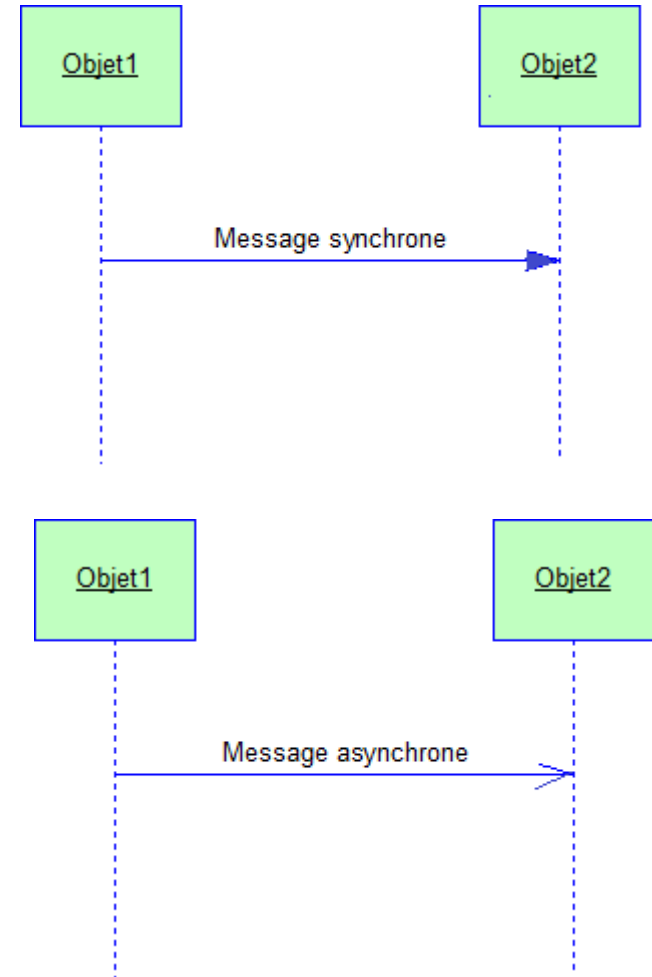


# Concepts d'un diagramme de séquence (4/20)

## ▪ Type d'envoi de message

L'envoi d'un message entre deux acteurs peut être :

- **Synchrone** : l'objet émetteur est bloqué jusqu'à ce que le récepteur termine le traitement du message.
- **Asynchrone** : l'objet émetteur est libre dès que le message est émis.

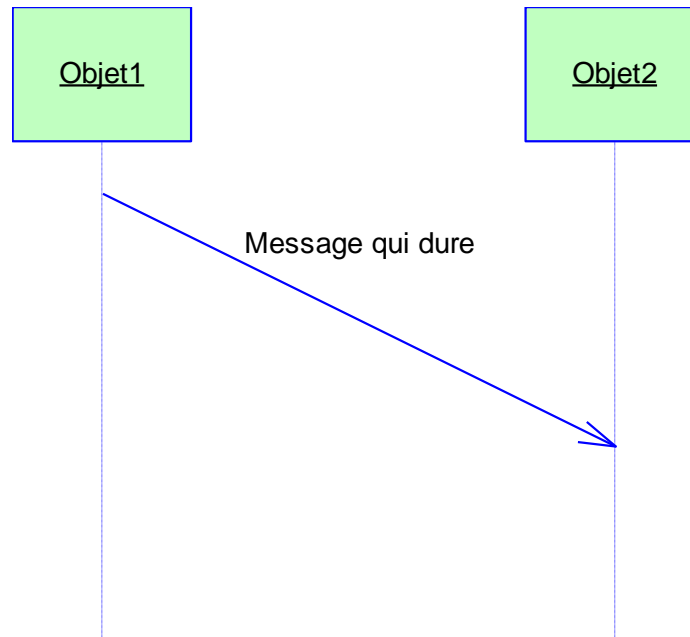


# Concepts d'un diagramme de séquence (5/20)



## ▪ Durée de transmission d'un message

- Lorsque la durée de transmission d'un message est importante, elle peut être représentée graphiquement :

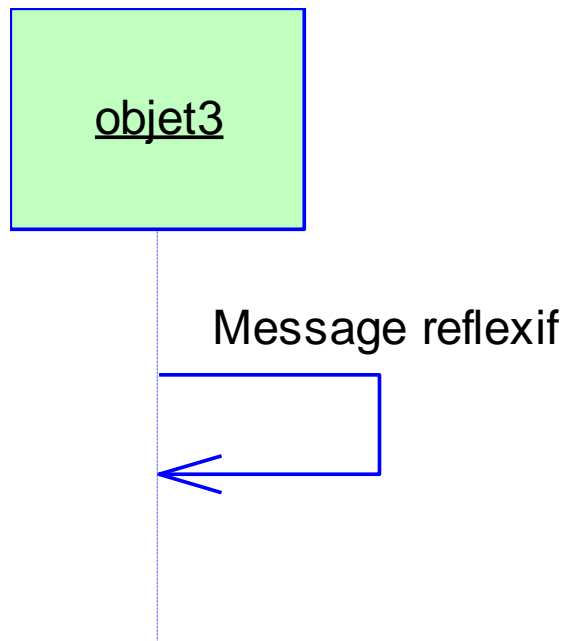


# Concepts d'un diagramme de séquence (6/20)



## ▪ Message réflexif

- Un objet peut envoyer un message à lui même:



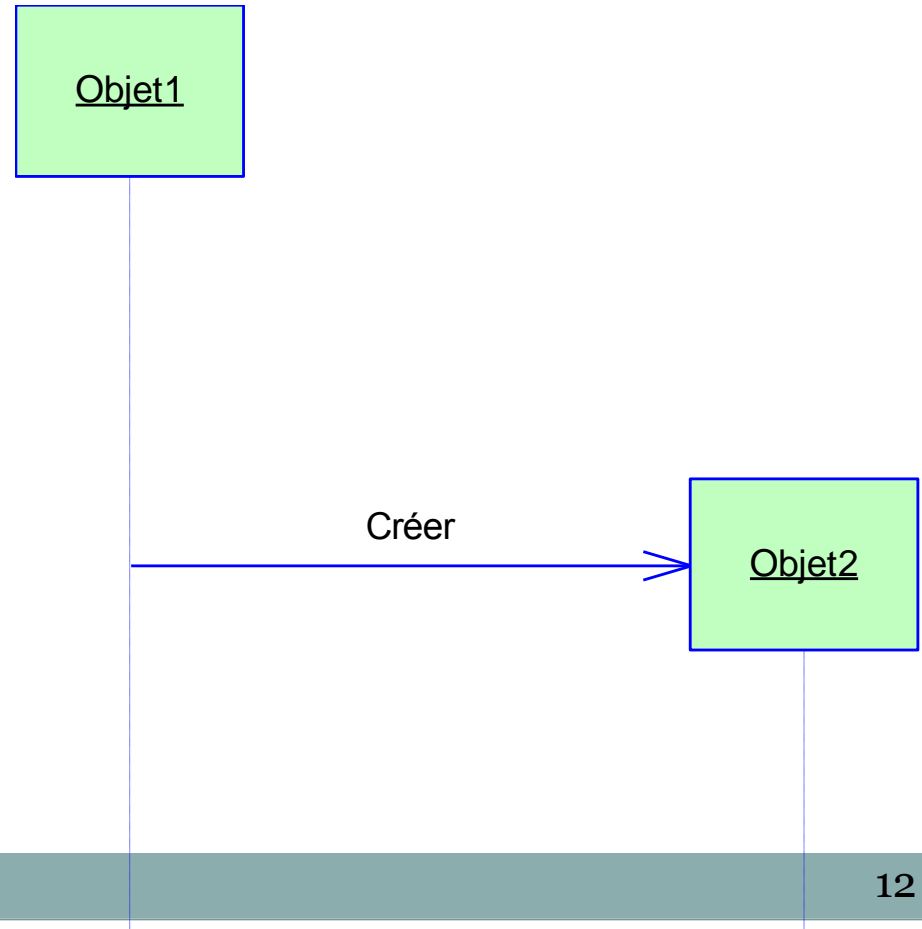
**Remarque :**  
Il s'agit généralement d'un objet composite.

# Concepts d'un diagramme de séquence (7/20)



## ▪ Message de création d'objet

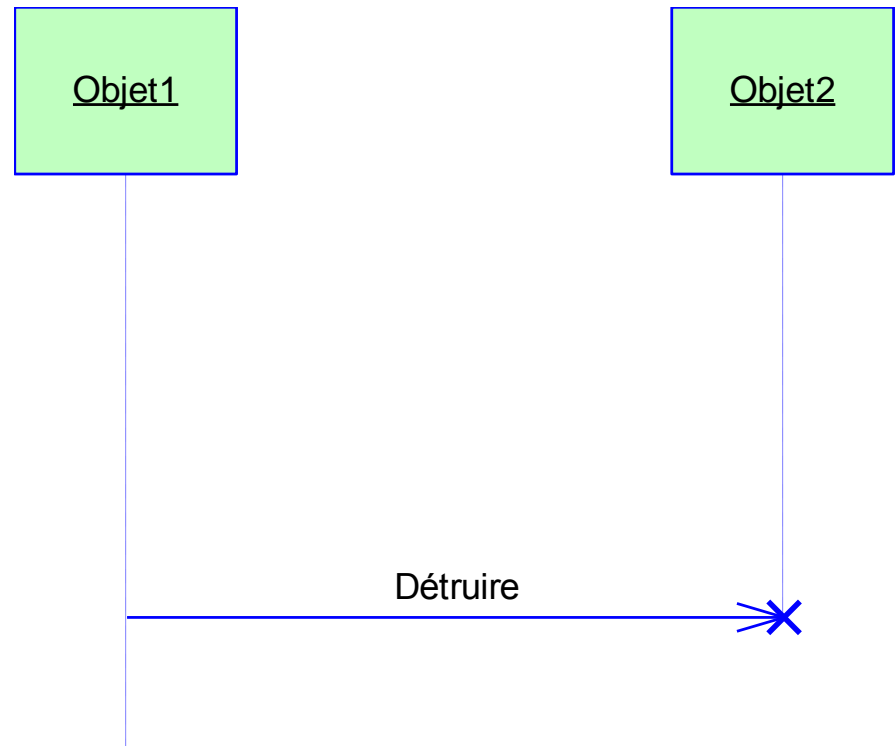
- Dans certains cas, un objet n'existe pas initialement.
- Il est **créé** par un message.



# Concepts d'un diagramme de séquence (8/20)

## ▪ Message de suppression d'objet

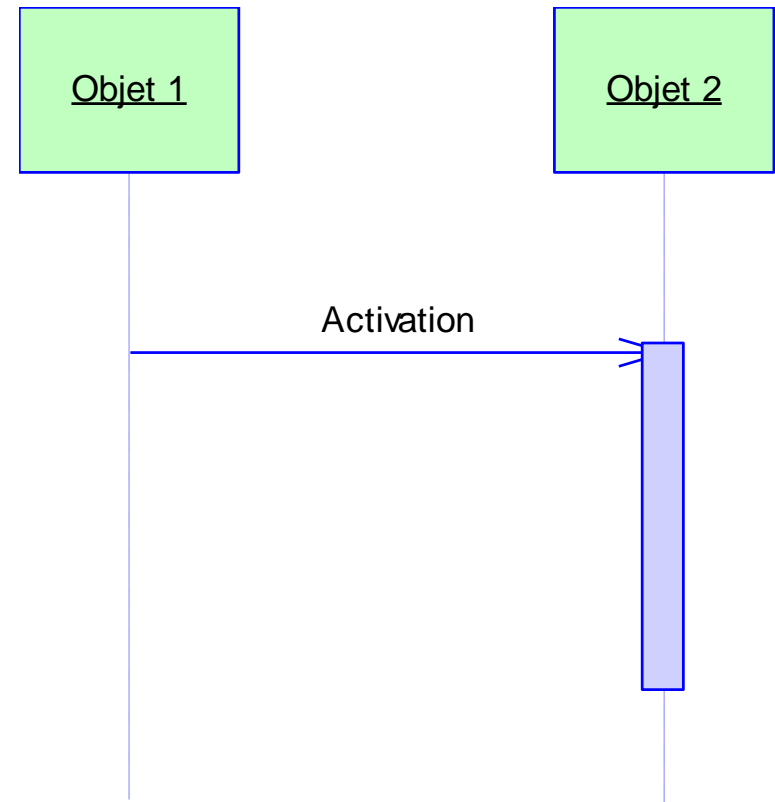
- Dans certains cas, un objet **disparaît** dans un diagramme de séquence.
- Il est **détruit** par un message.



# Concepts d'un diagramme de séquence (9/20)

## ▪ Période d'activité d'un objet

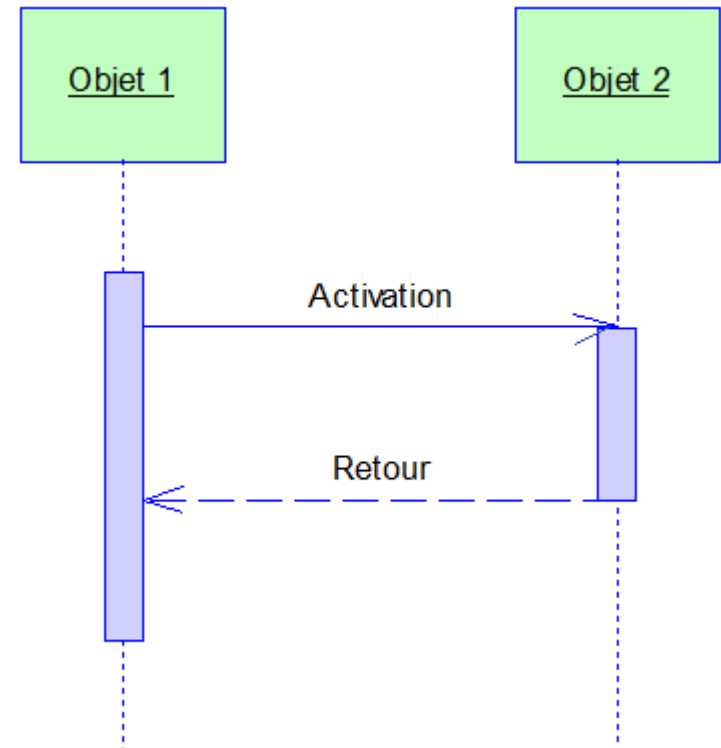
- Dans certains cas, il est utile de représenter la période pendant laquelle un objet est actif : **période d'activité**.



# Concepts d'un diagramme de séquence (10/20)

## ▪ Message de retour

- Dans le cas d'activation d'un objet par un message synchrone, le message de retour est implicite.
- Dans le cas d'activation d'un objet par un message asynchrone, le message de retour doit être représenté explicitement.

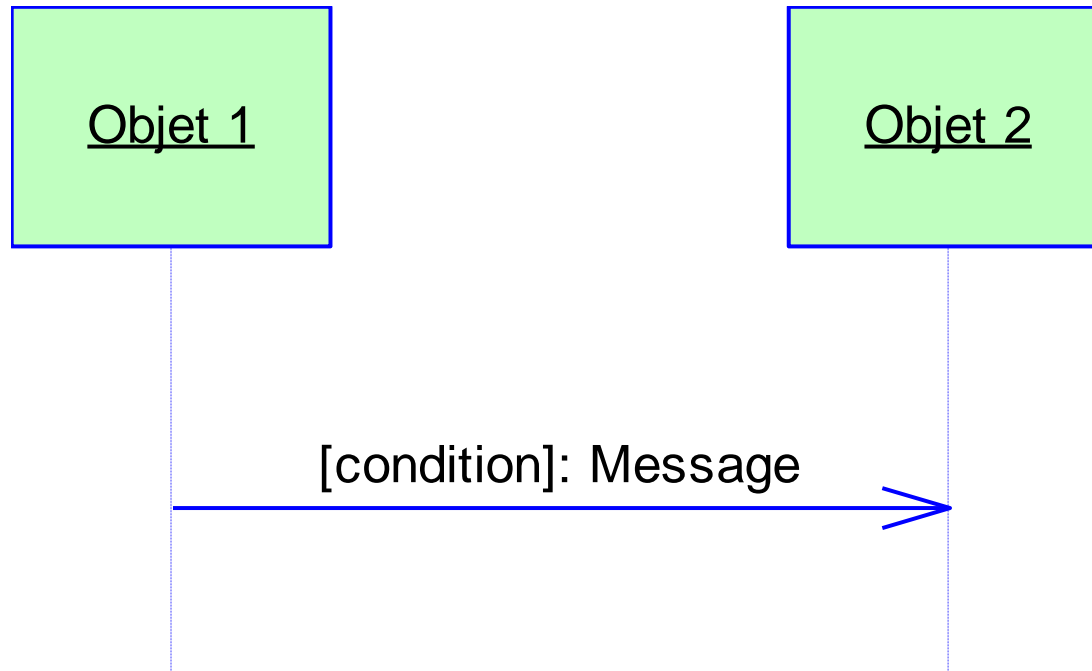


# Concepts d'un diagramme de séquence (11/20)



- **Message conditionnel**

- Dans certains cas, l'envoi d'un message est soumis à une condition.



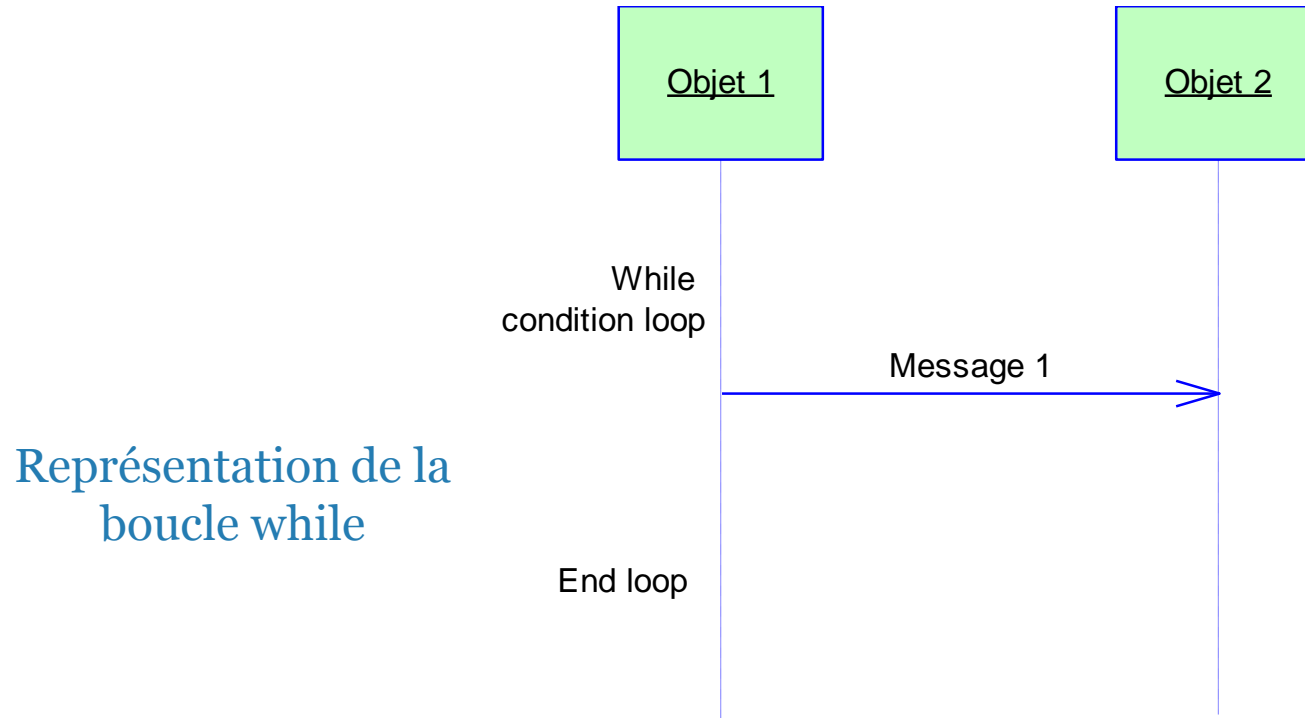


# Concepts d'un diagramme de séquence (12/20)



## ▪ Structures de contrôle

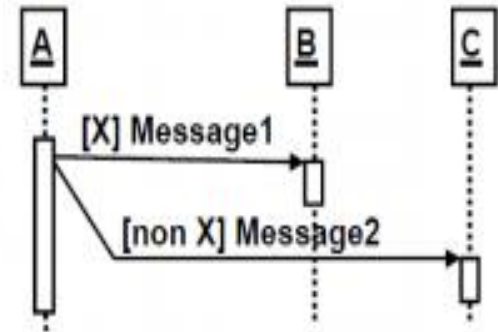
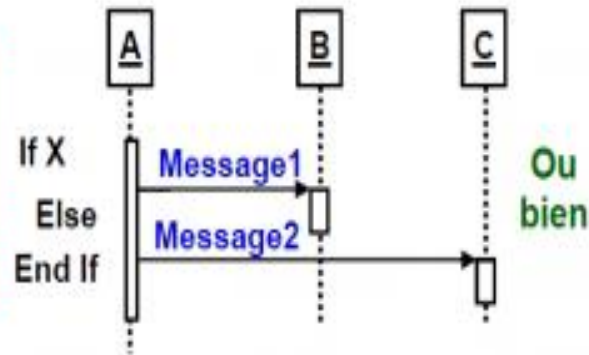
- Les diagrammes de séquences peuvent être complétés par des indications textuelles exprimées sous forme de texte libre ou pseudo-code.



# Concepts d'un diagramme de séquence (13/20)

- Structures de contrôle
- Branchements conditionnels

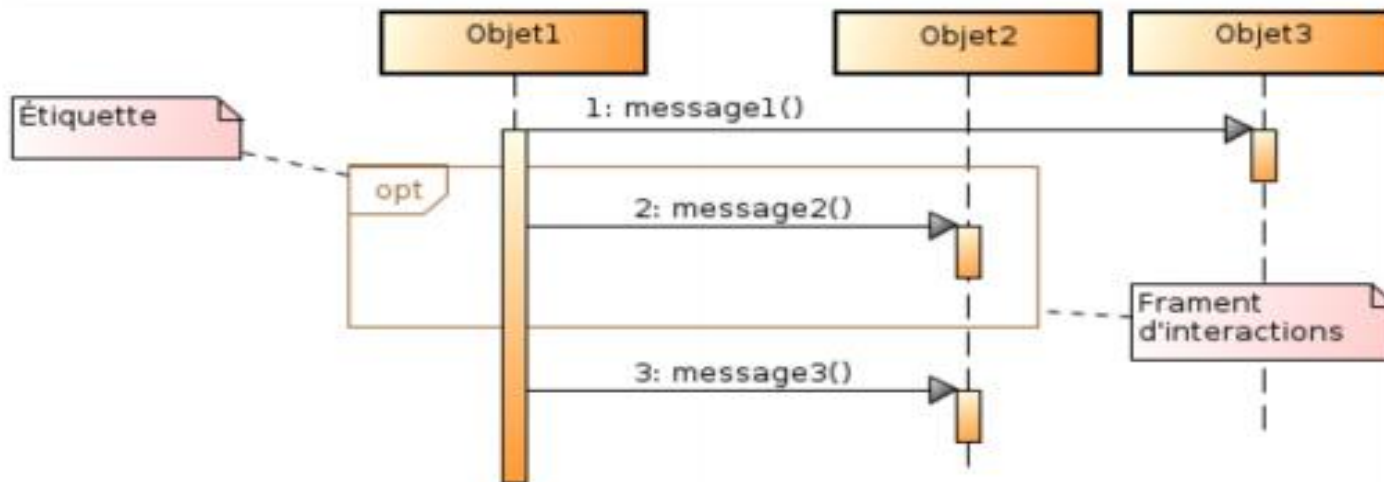
```
Si X = Vrai Alors  
    Envoyer(A, B, Message1)  
Sinon  
    Envoyer(A, C, Message2)  
FINSI
```



# Concepts d'un diagramme de séquence (14/20)

## ▪ Fragments d'interaction combinés

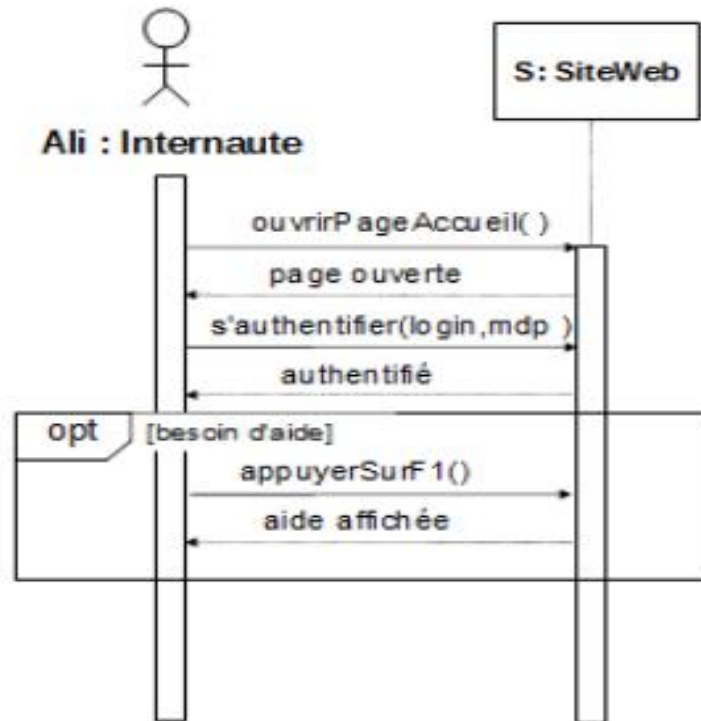
- Un fragment d'interactions est une partie du diagramme de séquence (délimitée par un rectangle) associée à une étiquette (dans le coin supérieur gauche).
- L'étiquette contient un opérateur d'interaction qui permet de décrire des modalités d'exécution des messages à l'intérieur du cadre.



# Concepts d'un diagramme de séquence (15/20)

## ▪ Fragments d'interaction combinés

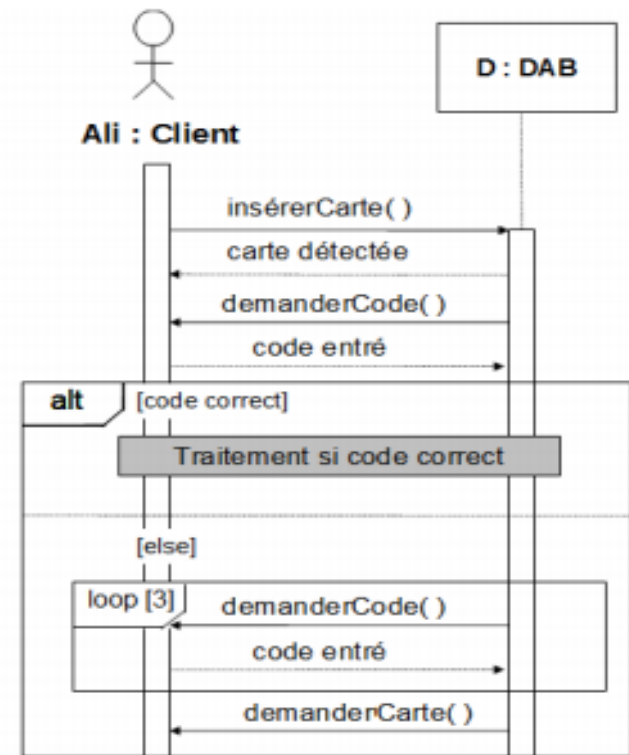
- Option (ou opt) représente un comportement qui peut se produire ou pas.



# Concepts d'un diagramme de séquence (16/20)

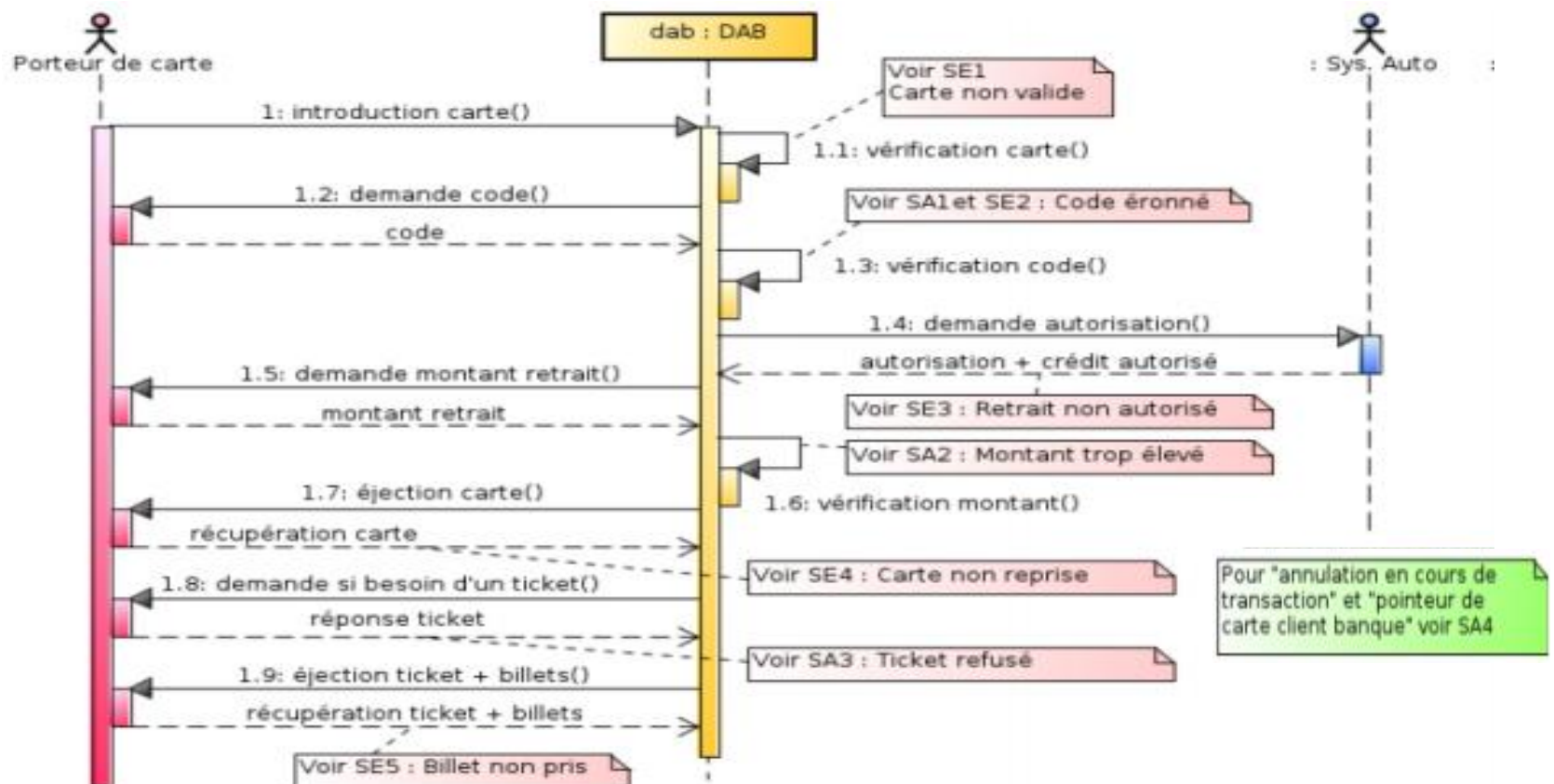
## ■ Fragments d'interaction combinés

- Alternative (ou alt) : opérateur conditionnel
- Equivalent d'une exécution à choix multiples (switch).
- Peut posséder plusieurs opérandes, chacune détient une condition de garde.
- Condition else: vraie si aucune autre condition n'est vraie.
- "Loop" équivalent d'une boucle for et décrit des interactions qui s'exécutent en boucle.



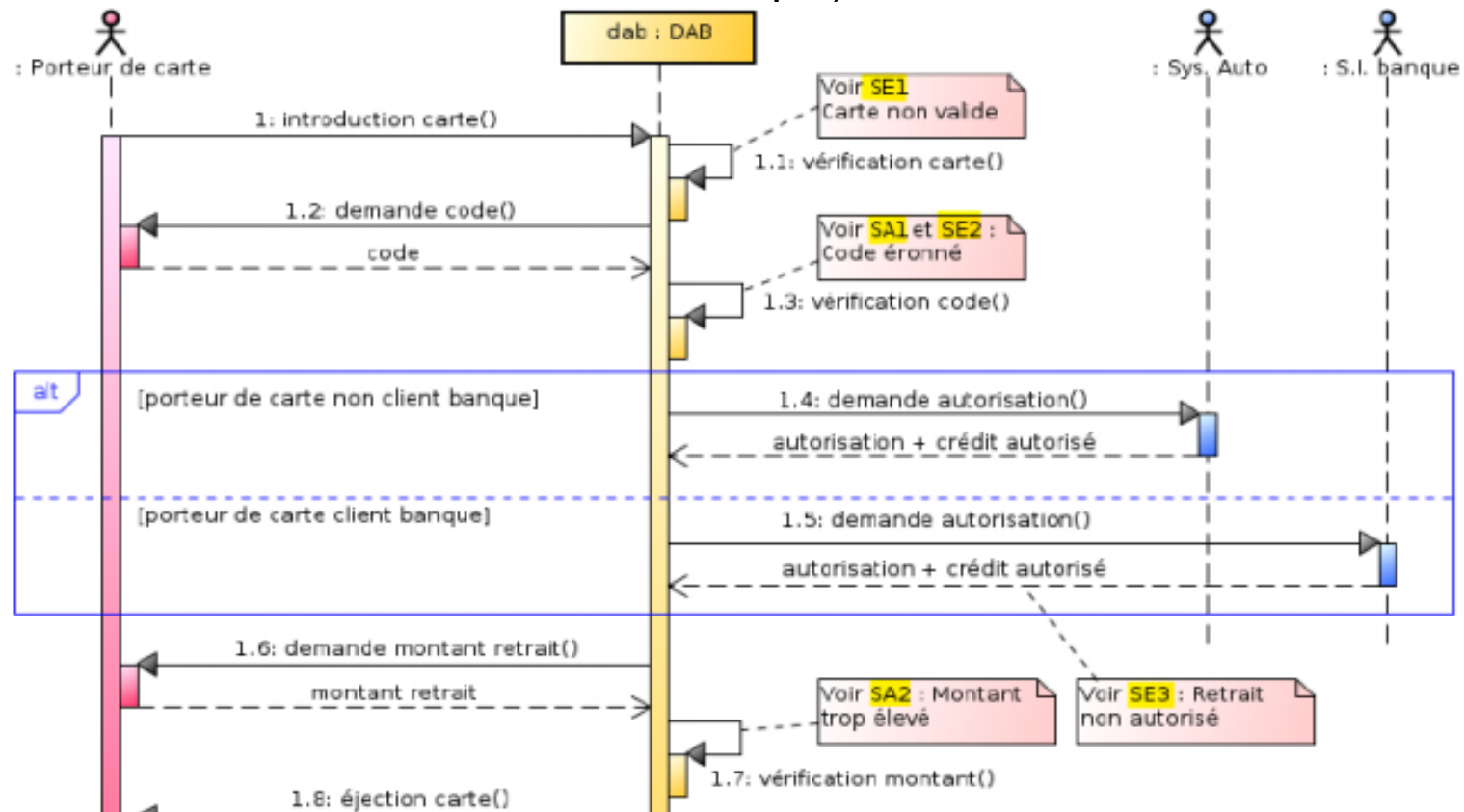
# Concepts d'un diagramme de séquence (17/20)

- **Exemple** Diagramme de séquence du cas d'utilisation "Retirer de l'argent" (scénario nominal)



# Concepts d'un diagramme de séquence (18/20)

- **Exemple** Diagramme de séquence du cas d'utilisation "Retirer de l'argent" (scénario nominal + scénario alternatif SA4) (porteur de carte client banque)



# Concepts d'un diagramme de séquence (19/20)



## ▪ Stéréotypes de Jacobson

- A l'intérieur d'un système, il existe très souvent des classes qui possèdent un rôle bien particulier, c'est le cas notamment :
  - pour les classes qui représentent des composants de l'IHM ;
  - pour les classes de contrôle
  - pour les classes qui implémentent la persistance des attributs (associées à une base de données)



# Concepts d'un diagramme de séquence (20/20)

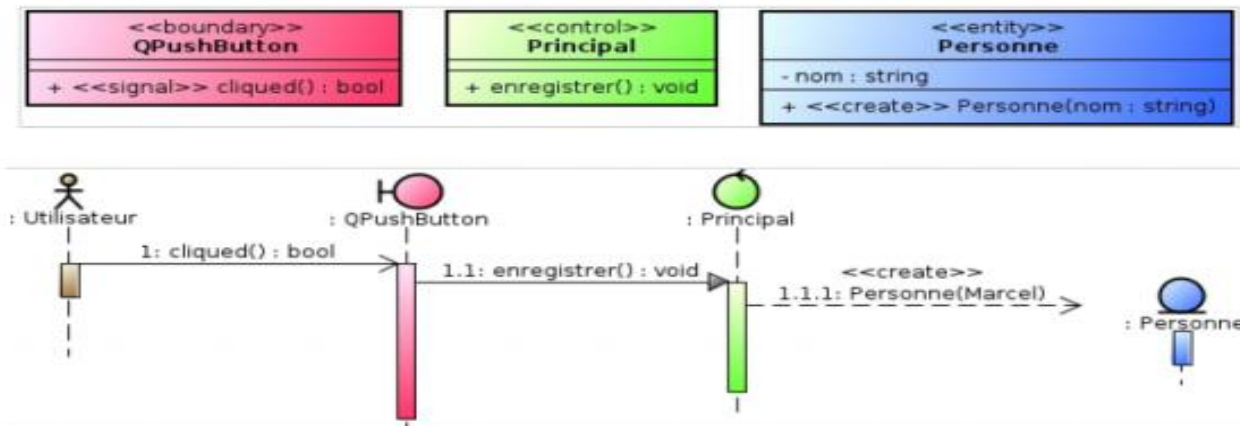
## ■ Stéréotypes de Jacobson

- Trois stéréotypes sont distingués :

- " **boundary** " : classes qui servent à modéliser les interactions entre le système et ses acteurs.

- " **control** " : classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.

- " **entity** " : classes qui servent à modéliser des informations durables et souvent persistantes.



# Activité



## **Système d'organisation d'une conférence internationale**

Considérons les règles de gestion suivantes d'un système d'organisation d'une conférence internationale.

- Le comité d'organisation adresse un appel à communication aux auteurs intéressés par les thèmes de la conférence.
- Un auteur soumet un papier décrivant ses travaux dans l'un des thèmes de la conférence.
- Un comité de programme est responsable de l'affectation des papiers reçus à des examinateurs spécialisés pour évaluation.
- Chaque examinateur évalue les papiers attribués.

Question: Représenter le diagramme de séquence relatif à ces règles.



# Activité

