

Chapitre 5: Diagrammes UML : vue dynamique

2022-2023

Dr. Kouah Sofia

Plan

- Diagrammes d'interaction (séquence et collaboration)
- Diagramme d'activités
- Diagramme d'états / transitions

Partie 1: Diagrammes d'interaction

- Diagramme de séquences
- Diagramme de communication

Les diagrammes UML

Axe de Modélisation

Statique

(ce que le système EST)

Diagramme de Classes

Diagramme d'Objets

Diagramme de Composants

Diagramme de Déploiement

Fonctionnel

(ce que le système FAIT)

Diagramme de cas d'utilisation

Dynamique

(Comment le système EVOLUE)

Diagramme d'Etats-Transitions

Diagramme d'Activité

Diagramme de Séquence

Diagramme de Collaboration



Diagramme de séquences: Rôle

C'est un diagramme comportemental (dynamique) et plus précisément c'est un diagramme d'interaction.

- permet de représenter des échanges entre les différents **objets** et **acteurs** du système en fonction du temps.
- A moins que le système à modéliser soit extrêmement simple, nous ne pouvons pas modéliser la dynamique globale du système dans un seul diagramme. Nous ferons donc appel à **un ensemble de diagrammes de séquences** chacun correspondant à une sous fonction du système, généralement pour illustrer un cas d'utilisation.

Diagramme de séquences: Composants

- **Délimitation du diagramme de séquence**
- **L'objet**
- **La ligne de vie**
- **Les messages**

Diagramme de séquences: les Composants

- **Délimitation du diagramme de séquence** : Le diagramme de séquence est placé dans un rectangle qui dispose d'une étiquette **sd** en haut à gauche (**sequence diagramm**) suivi du nom du diagramme.
- **L'objet** : Dans un diagramme de séquence, l'objet a la même représentation que dans le diagramme des objets. Le nom de l'objet est généralement souligné et peut prendre l'une des quatre formes suivantes :

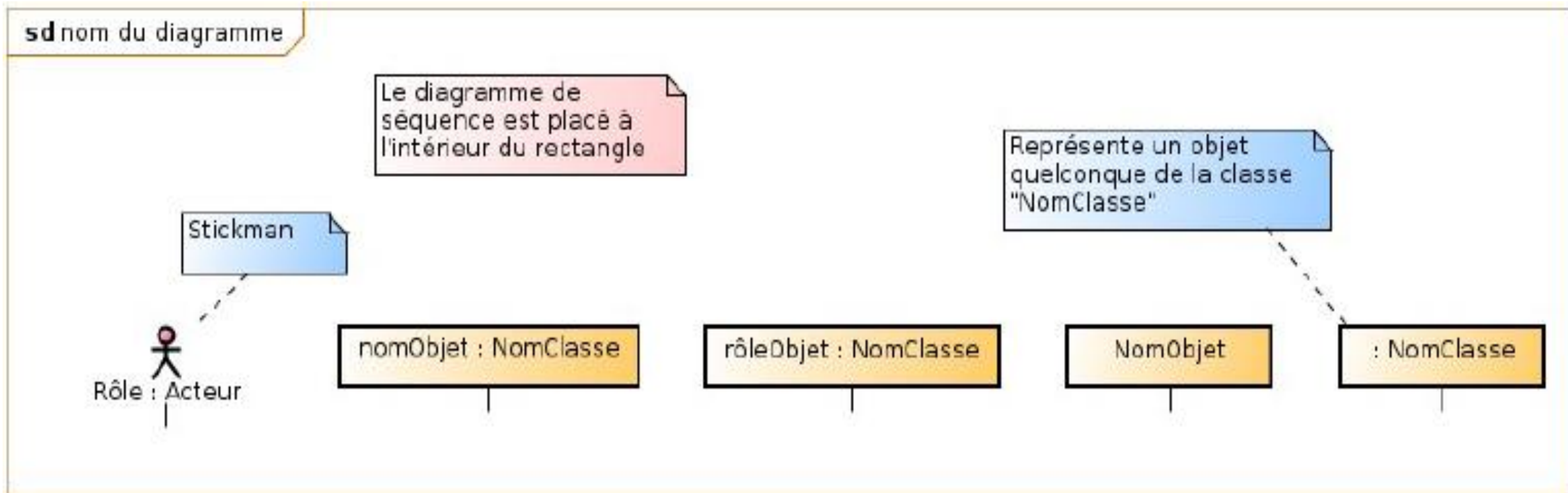


Diagramme de séquences: les Composants

La ligne de vie : Comme il représente la dynamique du système, le diagramme de séquence fait entrer en action les instances de classes intervenant dans la réalisation d'un cas d'utilisation particulier.

Échanges entre les objets mais aussi les échanges avec les acteurs, nous trouverons aussi la représentation du *stickman* (qui peut être considéré comme un objet).

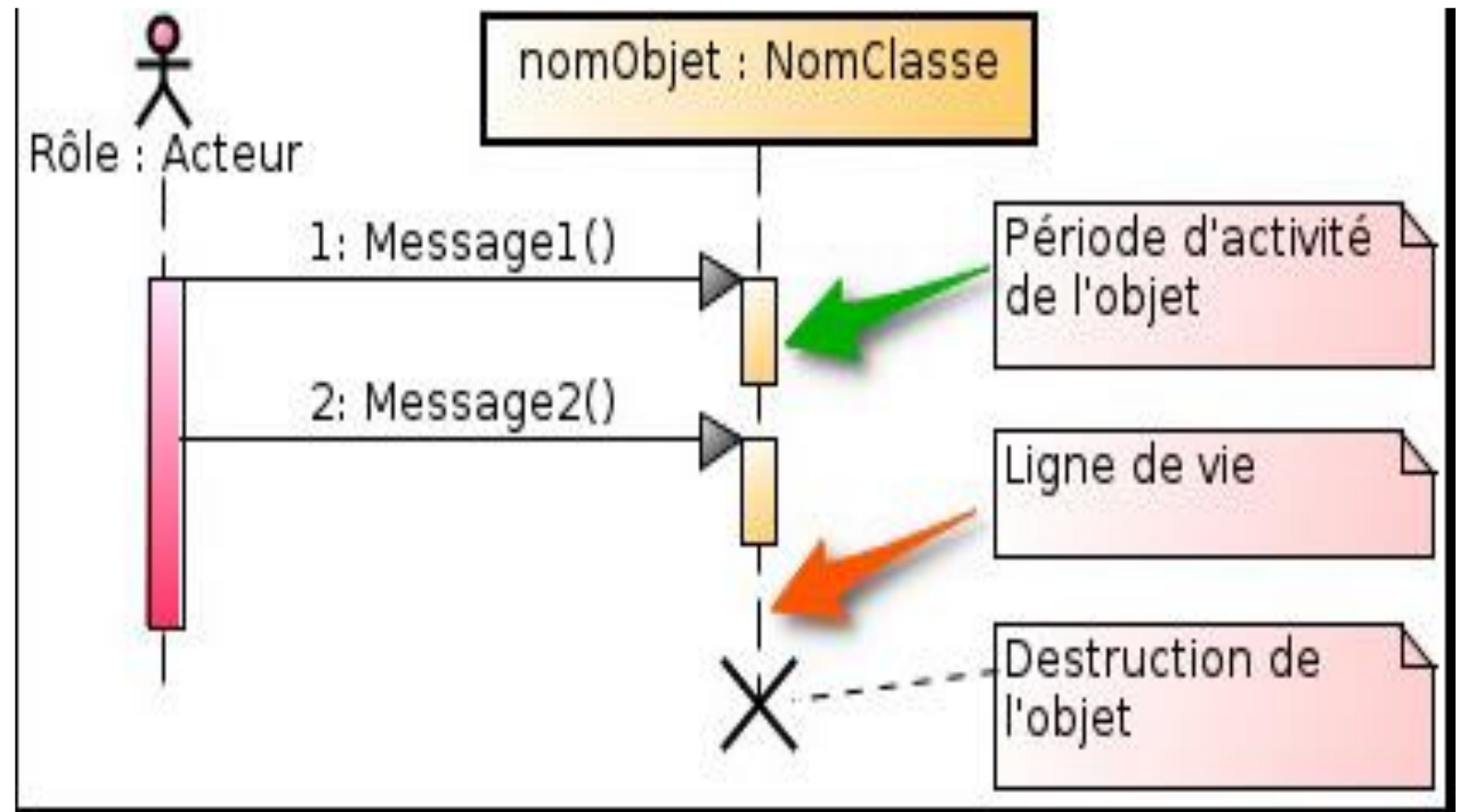


Diagramme de séquences: les Composants

- A chaque objet est associé une **ligne de vie** (en trait pointillés à la verticale de l'objet) qui peut être considéré comme un **axe temporel** (le temps s'écoule du haut vers le bas).
- La ligne de vie indique les **périodes d'activité** de l'objet (généralement, les moments où l'objet exécute une de ces méthodes).
- Lorsque l'objet est détruit, la ligne de vie s'achève par un croix.

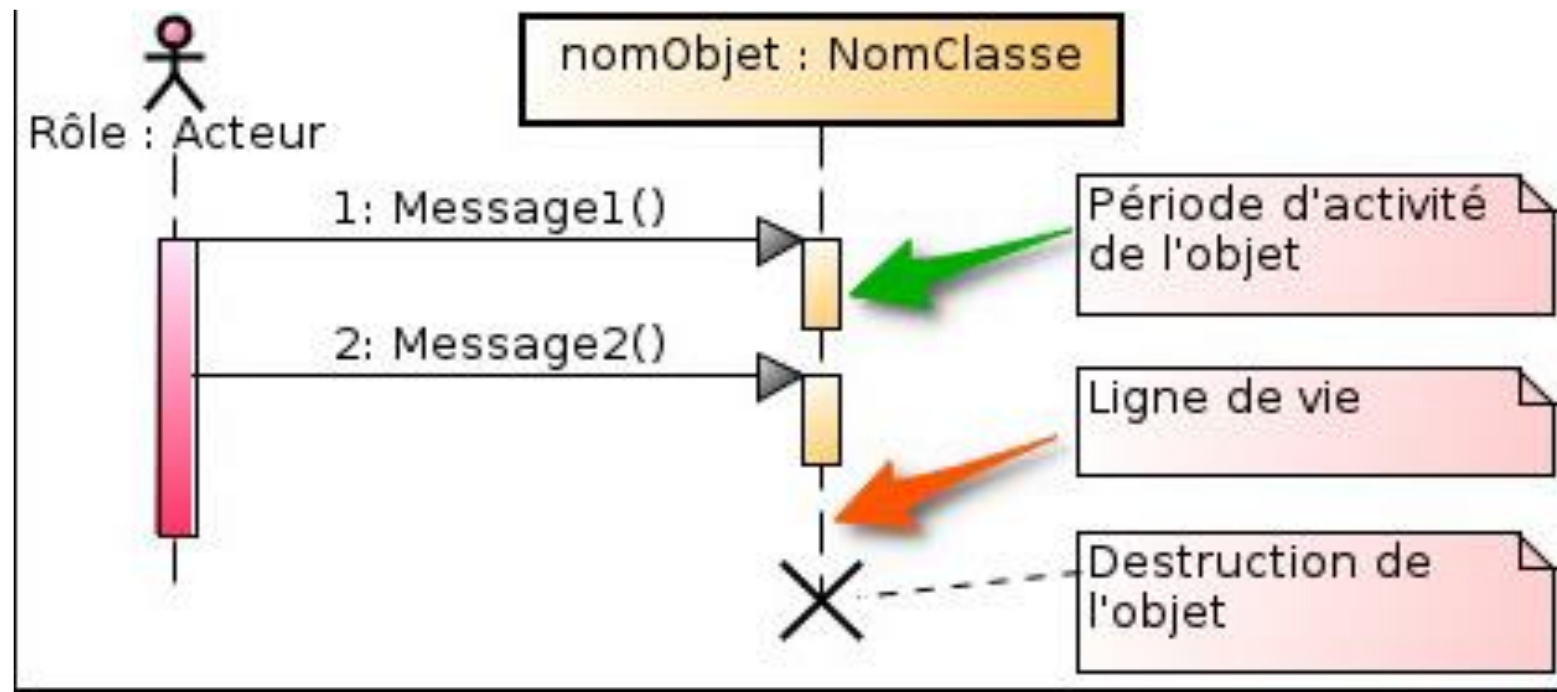


Diagramme de séquences: les Composants

- **Message** : définit une communication particulière entre des lignes de vie.
- 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 :
 - o **L'invocation d'une operation** : *message synchrone* (appel d'une méthode de l'objet cible).
 - o **L'envoi d'un signal** : *message asynchrone* (utilisé pour la gestion événementielle).
 - o La **création** ou la **destruction** d'une instance de classe au cours du cycle principal.

Diagramme de séquences: les Composants

Un message **synchrone** bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.

- Typiquement : appel de méthode
 - Si un objet A invoque une méthode d'un objet B, A reste bloqué tant que B n'a pas terminé.



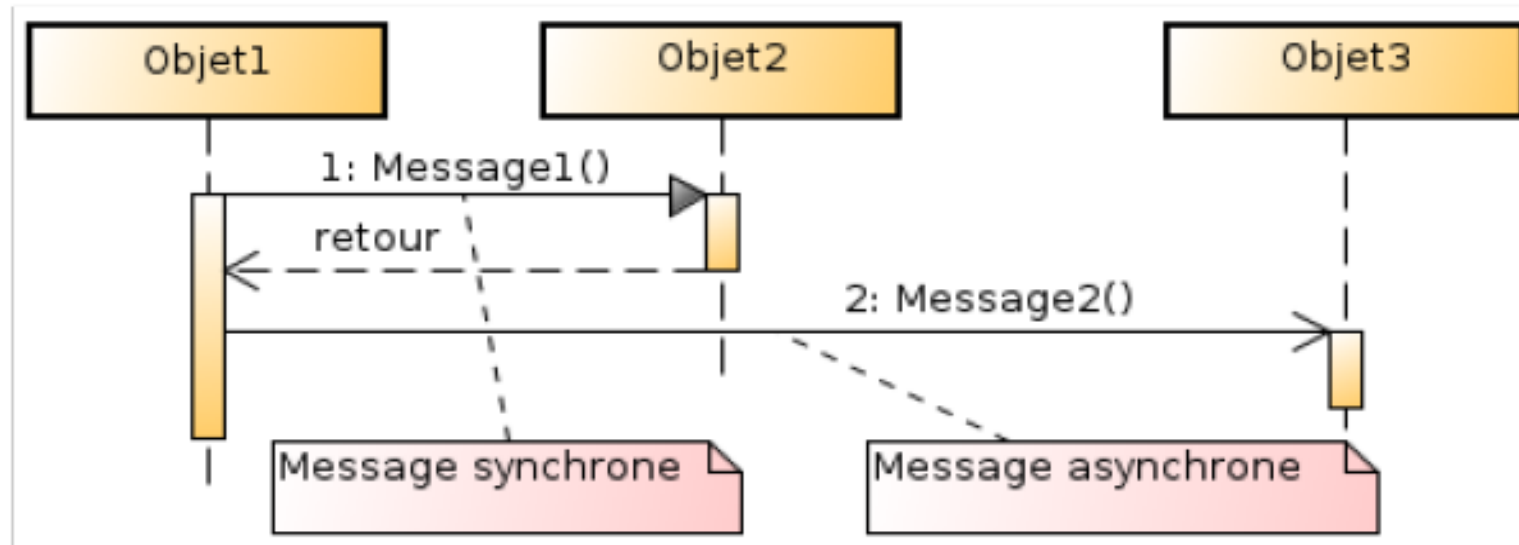
- On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone.

Un message **asynchrone** n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

- Typiquement : envoi de signal (voir stéréotype de classe « signal »).



Diagramme de séquences: les Composants



- Si une **méthode** qui a été activée (par un message) doit retourner des valeurs à la fin de son activation, cela se fait par un **message retour** (**Flèche en Pointillées**).
- Le **message de retour** n'est donc pas un appel de méthode (il ne provoque donc pas l'activation d'un objet).
- Le **message retour** porte souvent le nom de l'élément retourné.

Diagramme de séquences: Exemple

Message asynchrone (ex : le **message5** est lancé alors que l'exécution du **message1** n'est pas terminée)

Message synchrone (ex : on attend que l'exécution du **message2** soit terminée avant de lancer le **message4**)

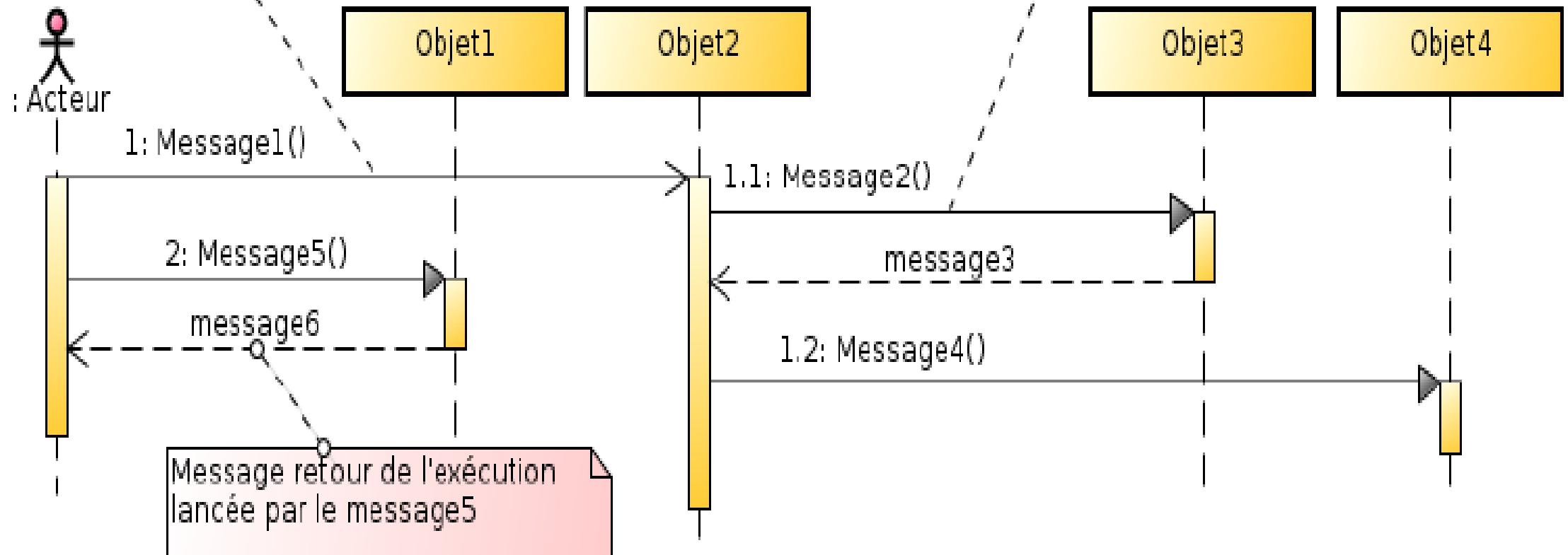


Diagramme de séquences: Création et destruction d'objets

Une séquence peut aussi contenir la **création** ou la **destruction** d'un objet :

- La **création d'un objet** est matérialisée par un **message spécifique**, appel d'un **constructeur**, généralement accompagné du stéréotype **«create»** qui pointe sur le début de la ligne de vie de l'objet créé.
- La **destruction d'un objet** est représentée par une croix à la fin de sa ligne de vie. Souvent l'objet est détruit suite à la réception d'un message mais ce n'est pas obligatoire. Dans ce cas là, il porte le stéréotype **«destroy»**.

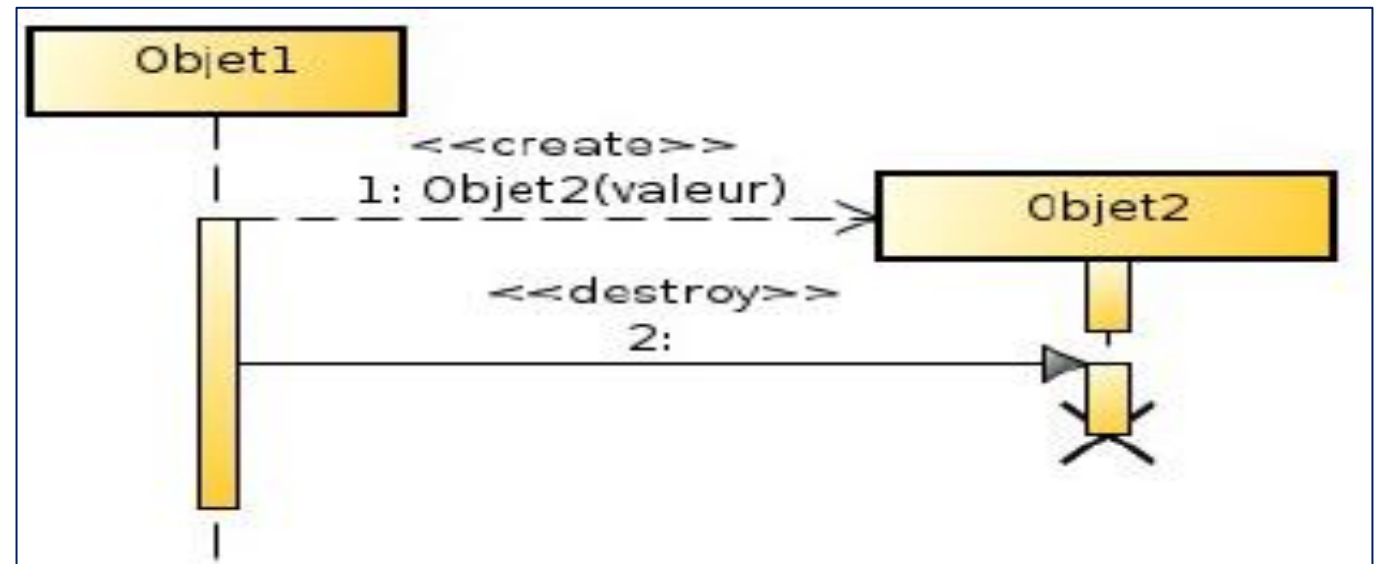


Diagramme de séquences: Message trouvé & Message Perdu

Un **message complet** est tel que les événements d'envoi et de réception sont connus.

- Un message complet est représenté par une flèche partant d'une ligne de vie et arrivant à une autre ligne de vie.

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



- La flèche part d'une ligne de vie mais arrive sur un cercle indépendant marquant la méconnaissance du destinataire.
- Exemple : broadcast.

Un **message trouvé** est tel que l'événement de réception est connu, mais pas l'événement d'émission.



Diagramme de séquences: Fragment combiné

Un **fragment combiné** permet de décomposer une interaction complexe en fragments suffisamment simples pour être compris.

Recombiner les fragments restitue la complexité.

Un fragment combiné se représente de la même façon qu'une interaction. Il est représenté un rectangle dont le coin supérieur gauche contient un **pentagone**.

Dans le pentagone figure le type de la combinaison (appelé opérateur d'interaction).

Diagramme de séquences: Fragment combiné

- Exemple fragment combiné

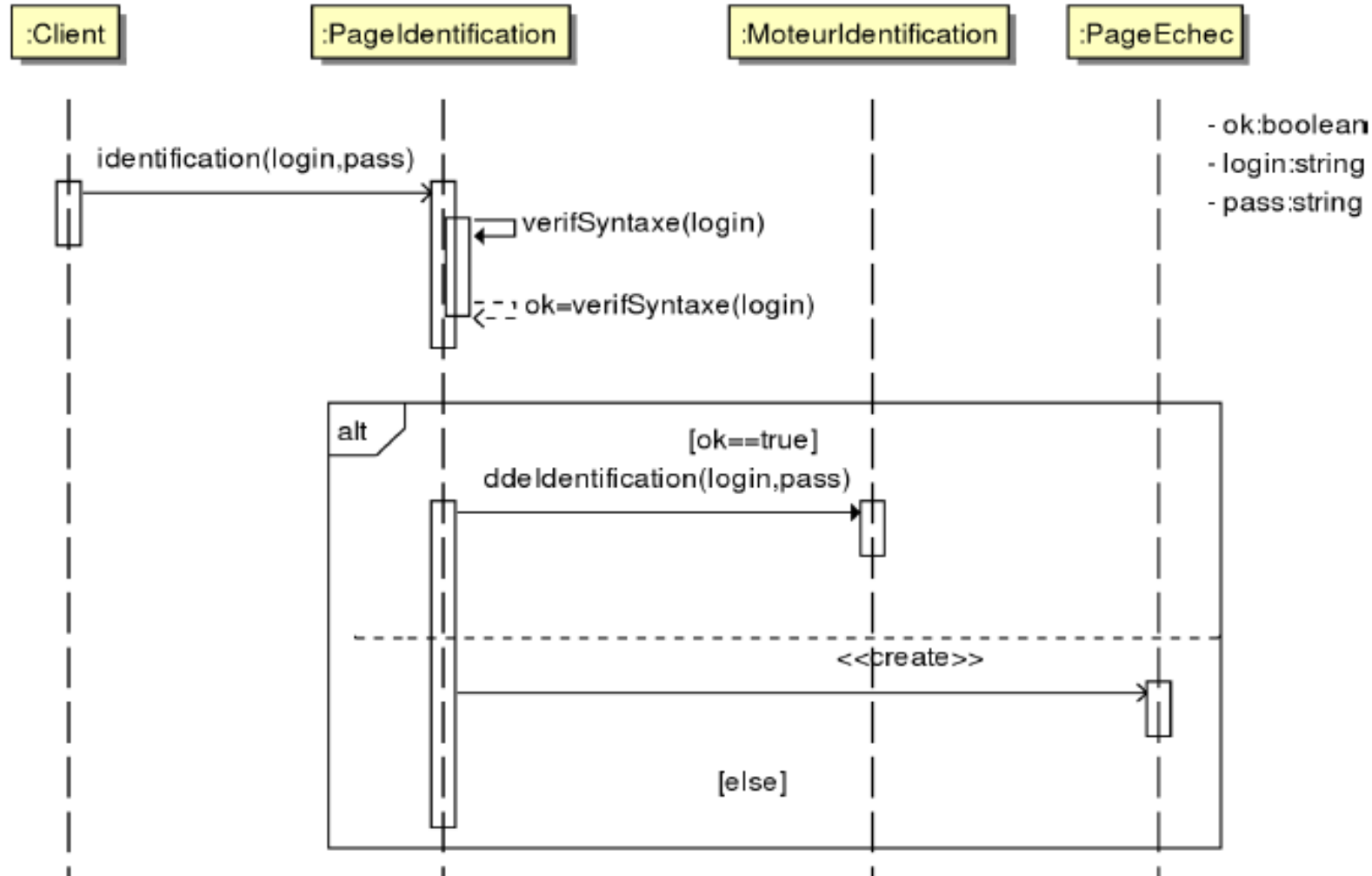


Diagramme de séquences: Fragment combiné

Type d'opérateurs d'interaction:

- Opérateurs de branchement (choix et boucles) : alternative, option, break et loop ;
- Opérateurs contrôlant l'envoi en parallèle de messages : parallel et critical region ;
- Opérateurs contrôlant l'envoi de messages : ignore, consider, assertion et negative ;
- Opérateurs fixant l'ordre d'envoi des messages : weak sequencing et strict sequencing.

Diagramme de séquences: Fragment combiné

- Opérateur parallèle: L'opérateur **par** permet d'envoyer des **messages en parallèle**.
- Ce qui se passe de part et d'autre de la ligne pointillée est indépendant.

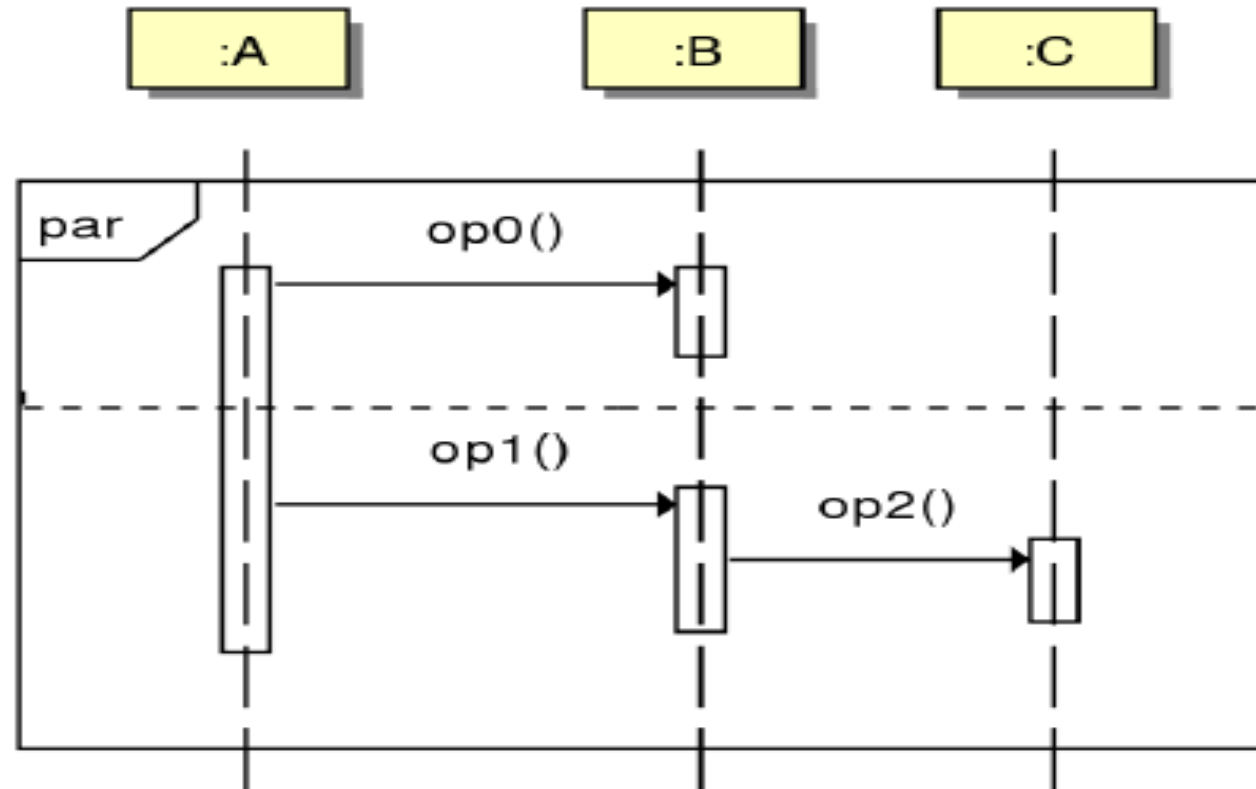


Diagramme de séquences: Fragment combiné

- Réutilisation: Réutiliser une interaction consiste à placer un fragment portant la référence **ref** là où l'interaction est utile.
- On spécifie le nom de l'interaction dans le fragment.

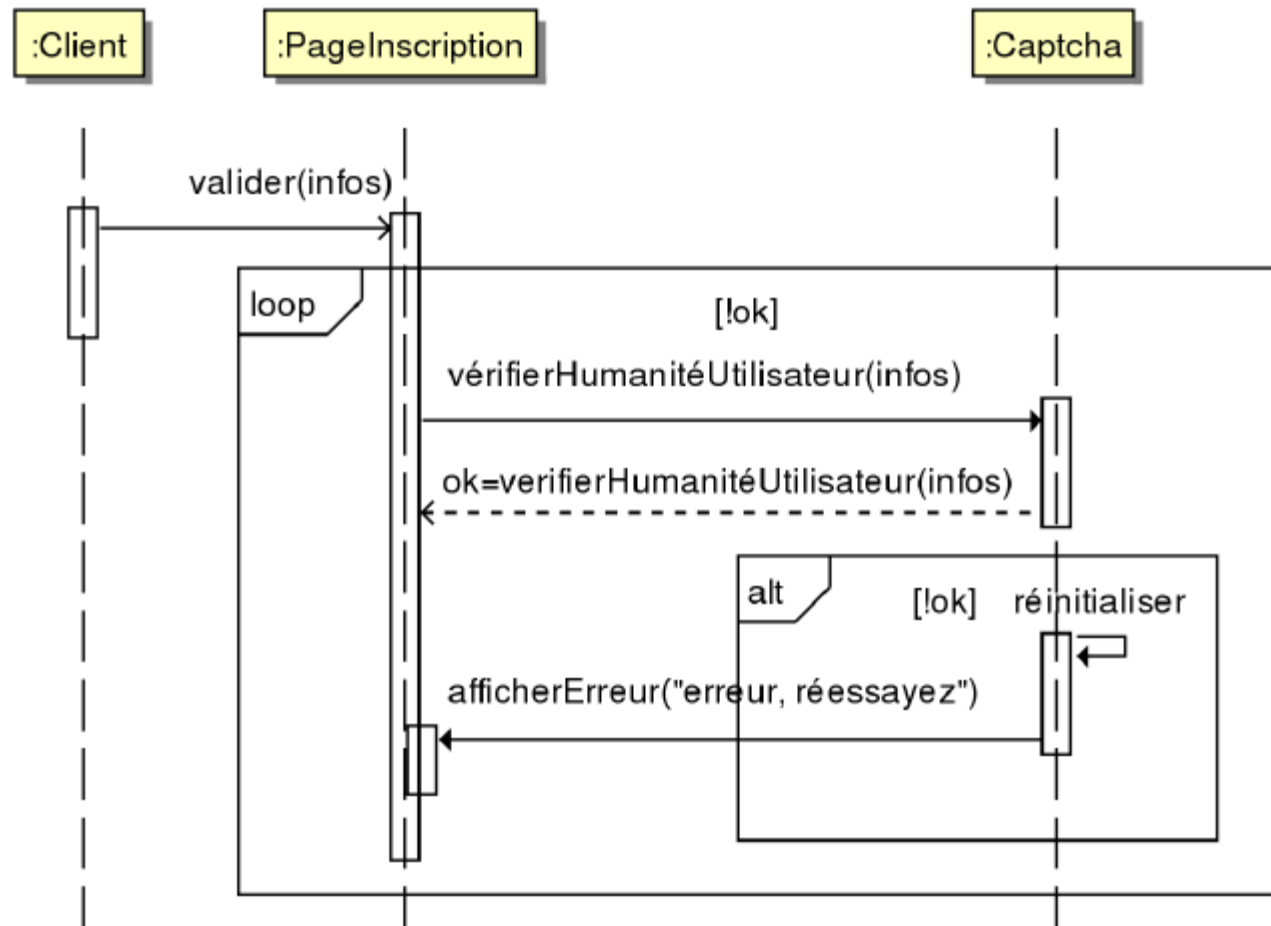


Diagramme de séquences: Fragment combiné

- Réutilisation: Réutiliser une interaction consiste à placer un fragment portant la référence **ref** là où l'interaction est utile.
- On spécifie le nom de l'interaction dans le fragment.

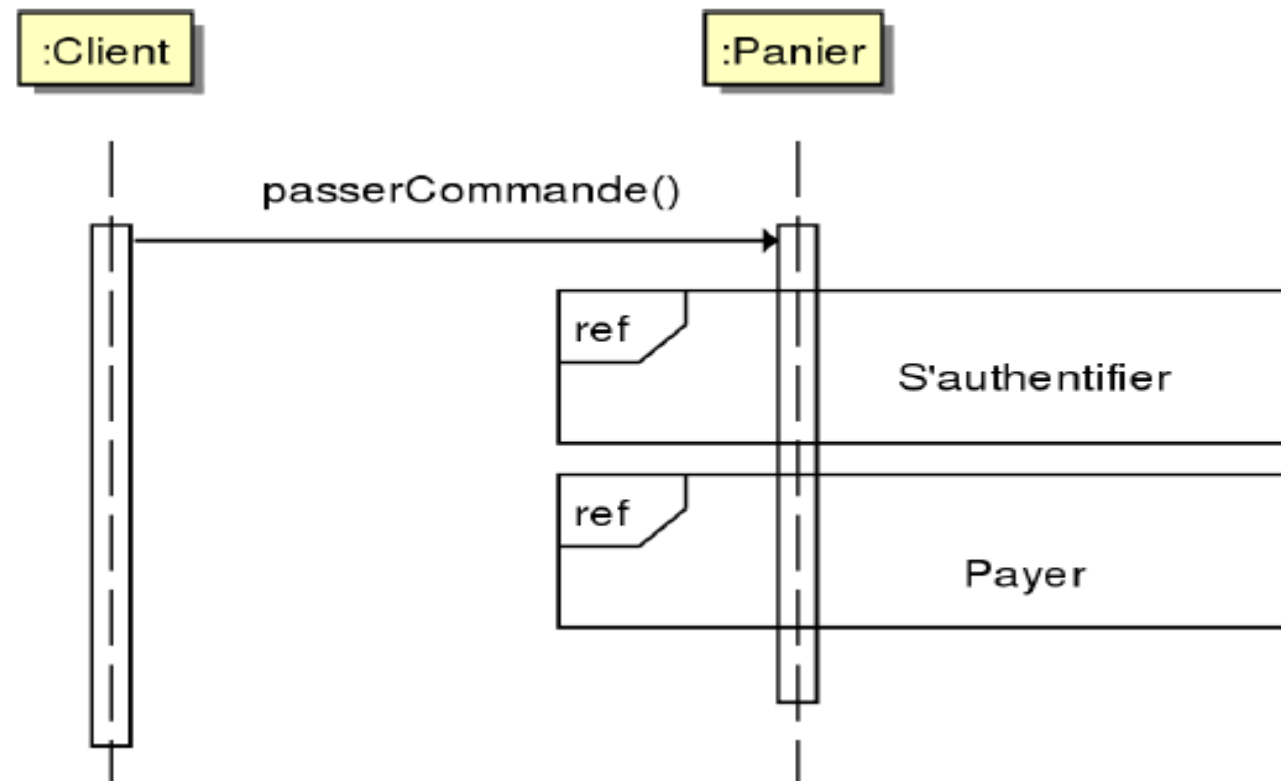


Diagramme de séquences: exemple d'utilisation

Spécifier un cas d'utilisation

Cas d'utilisation :



Diagramme de séquences correspondant :

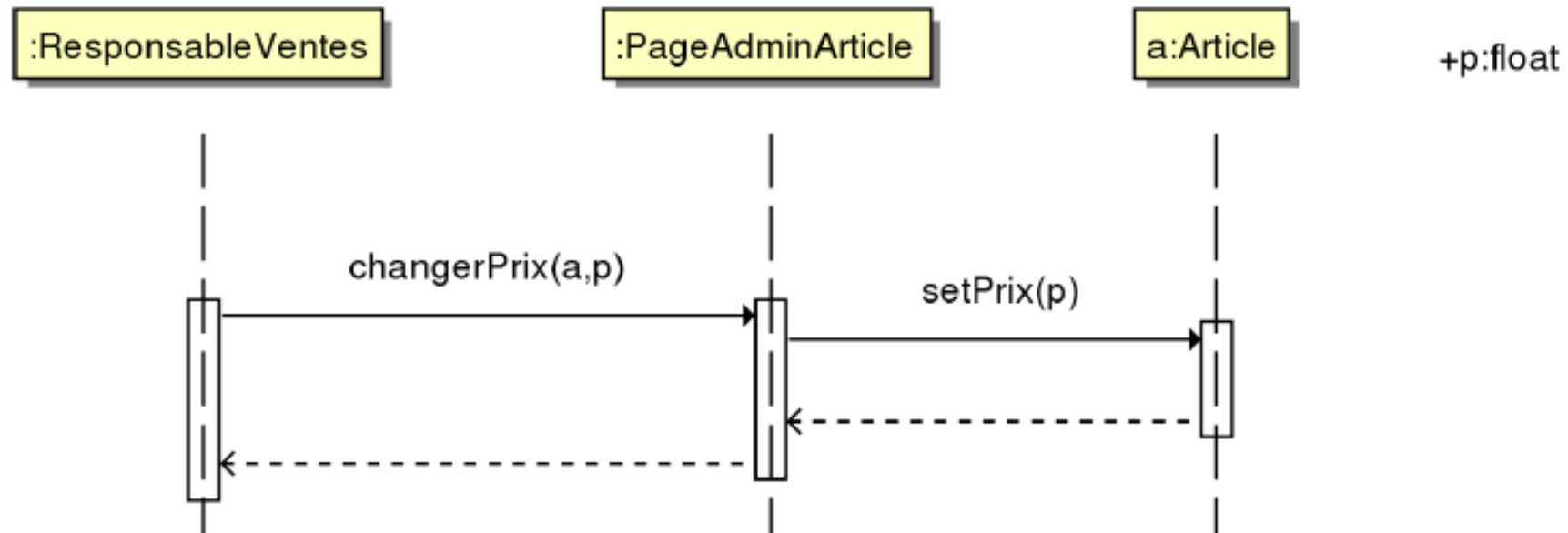


Diagramme de séquences: exemple d'utilisation

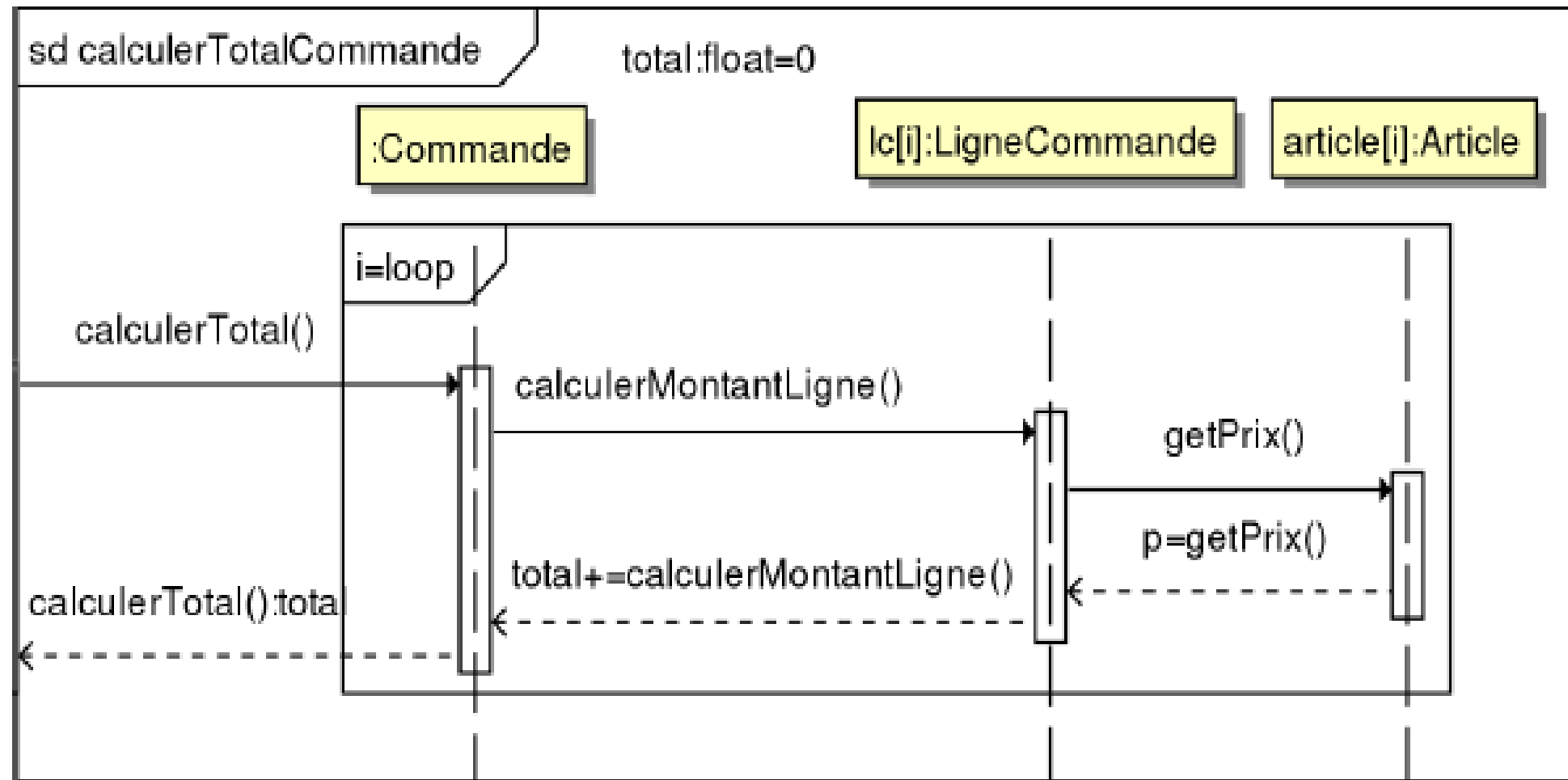


Diagramme de communication

Les diagrammes de communication et les diagrammes de séquences

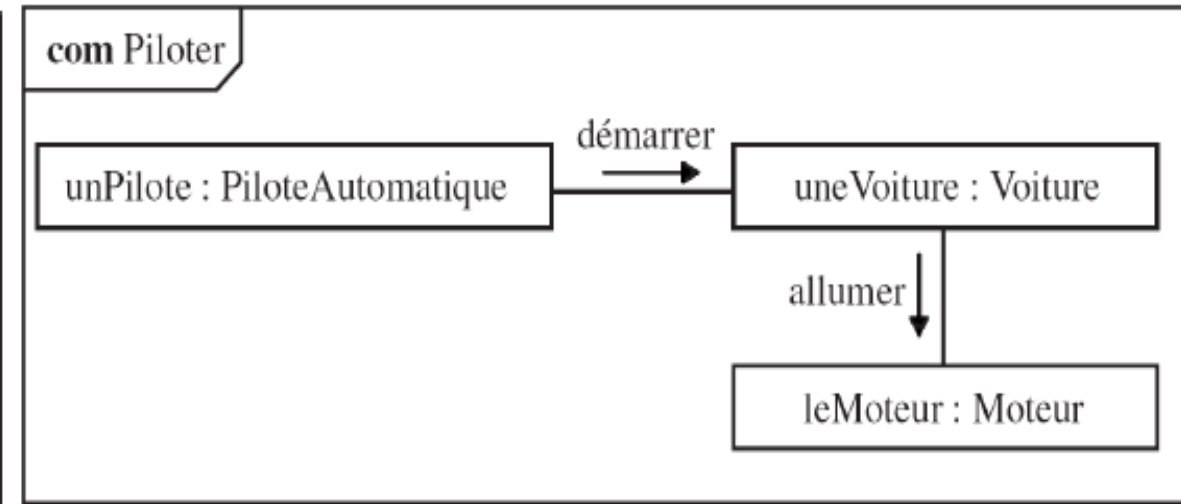
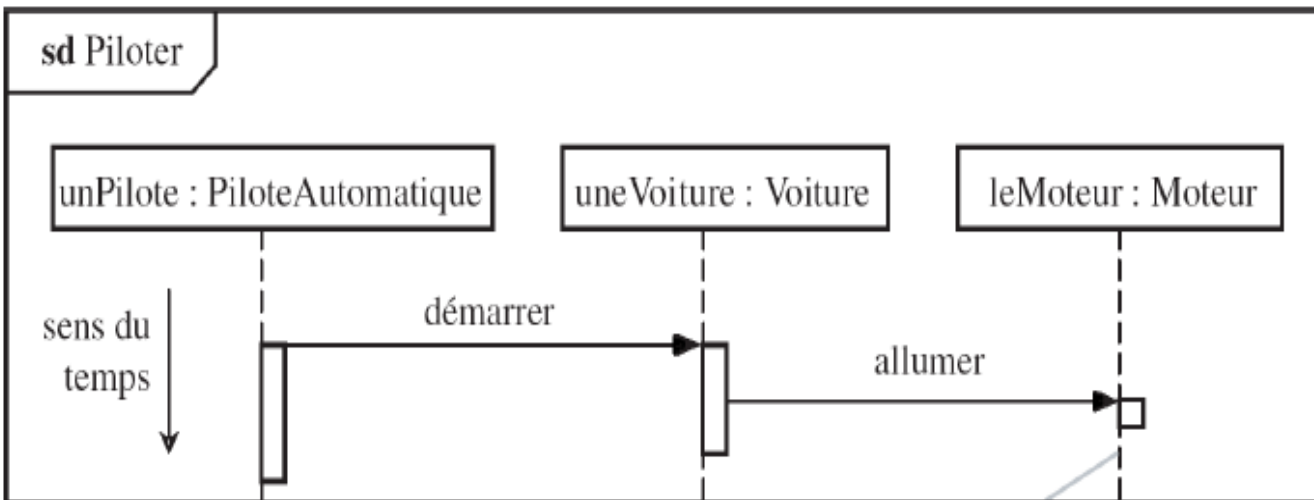
sont deux types de diagramme d'interaction.

- Un **diagramme de séquence** montre des interactions sous un **angle temporel**, en mettant l'emphasis sur le **séquencement temporel de messages échangés entre des lignes de vie**.

Un diagramme de communication montre une **représentation spatiale des lignes de vie**.

Ils représentent la même chose, mais sous des formes différentes.

Diagramme de Séquences & Diagramme de Communication



ligne de vie

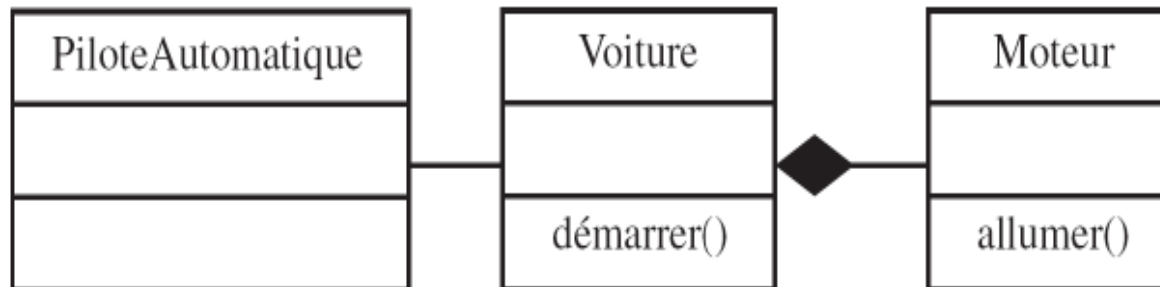


Diagramme de communication: les composants

- **Rôle**
- **Connecteur**
- **Message**

Diagramme de communication: les composants (Rôle & Connecteur)

- Le **rôle** permet de définir le **contexte d'utilisation de l'interaction**.
- Une **rôle dans un diagramme de communication** correspond à **une ligne de vie dans un diagramme de séquences**.
- Les **relations entre les lignes de vie** sont appelées **connecteurs**.
- Un **connecteur** se représente de la même façon qu'une association mais la sémantique est plus large : un connecteur est souvent une association transitoire.

Diagramme de communication: les composants (Message)

Comme dans les diagrammes de séquence, on distingue deux types de messages :

- Un message **synchrone** bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.



- Un message **asynchrone** n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.



Remarque: la distinction entre les messages et les connecteurs : on pourrait avoir un connecteur sans message, mais jamais l'inverse.

Diagramme de communication: les composants (Message)

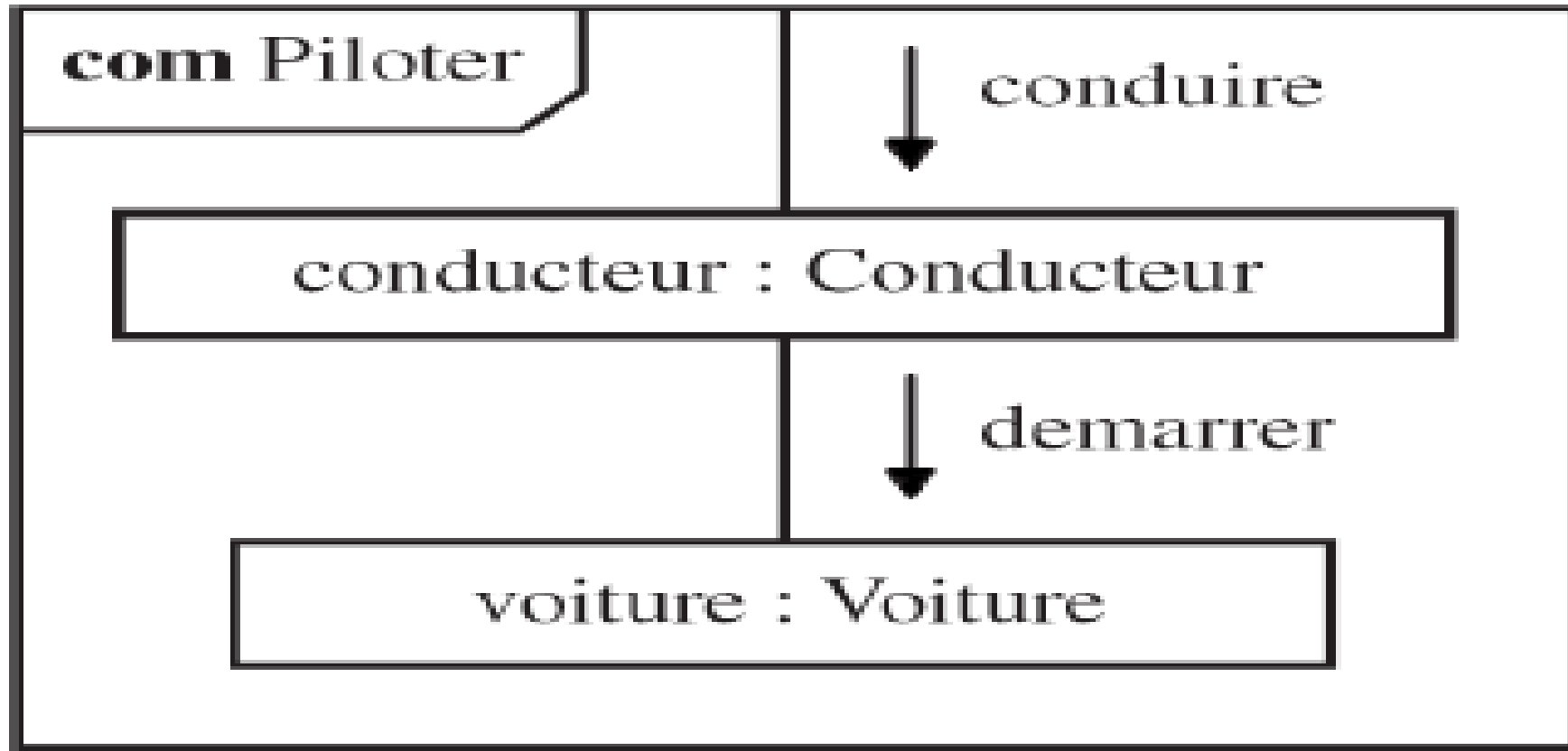


Diagramme de communication: les composants (Message)

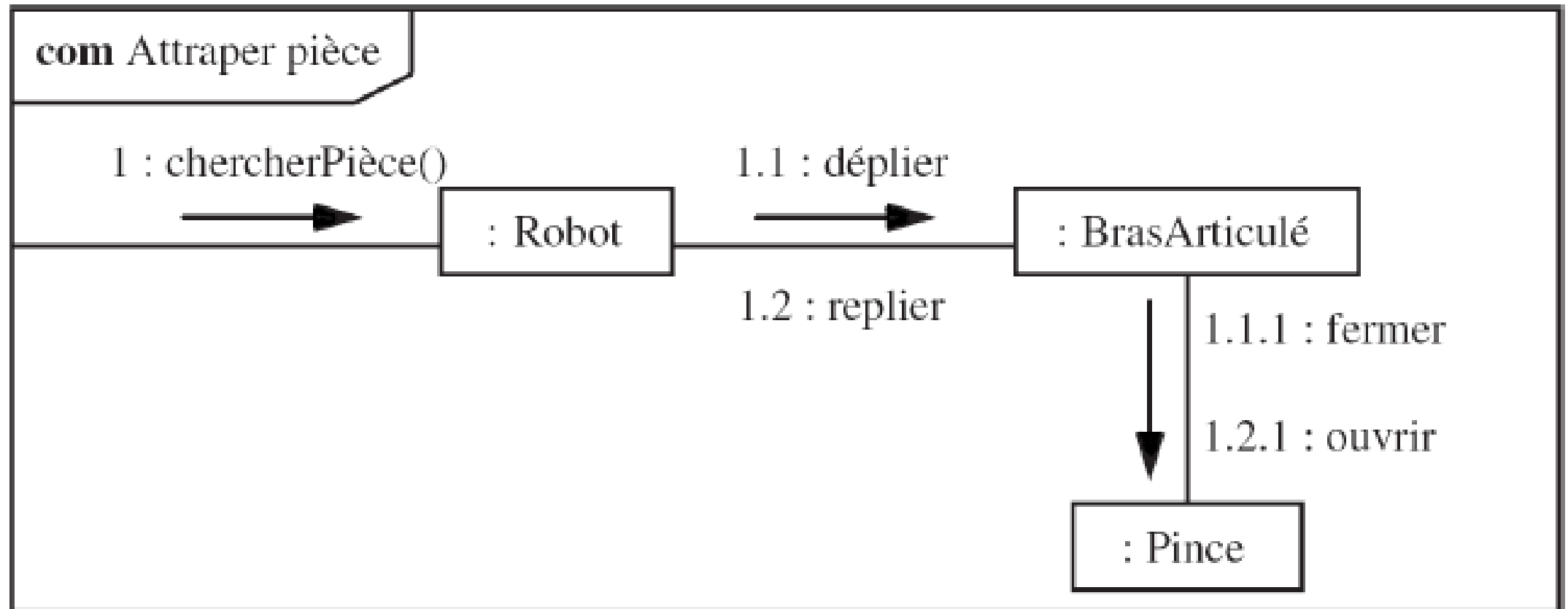
Pour représenter les **aspects temporels**, on adjoint des **numéros de séquence aux messages**.

- Des **messages successifs** sont ordonnés selon un numéro de séquence croissant (1, 2, 3, ... ou encore 3.1, 3.2, 3.3, ...).
- Des messages envoyés en cascade (ex : appel de méthode à l'intérieur d'une méthode) portent un numéro d'emboîtement avec **une notation pointée**:

1.1, 1.2, ... pour des messages appelés par une méthode dont l'appel portait le numéro 1

2.a.1, 2.a.2, ... pour des messages appelés par une méthode dont l'appel portait le numéro 2.a

Diagramme de communication: Exemple



Equivalence entre Diagramme de séquence et diagramme de communication

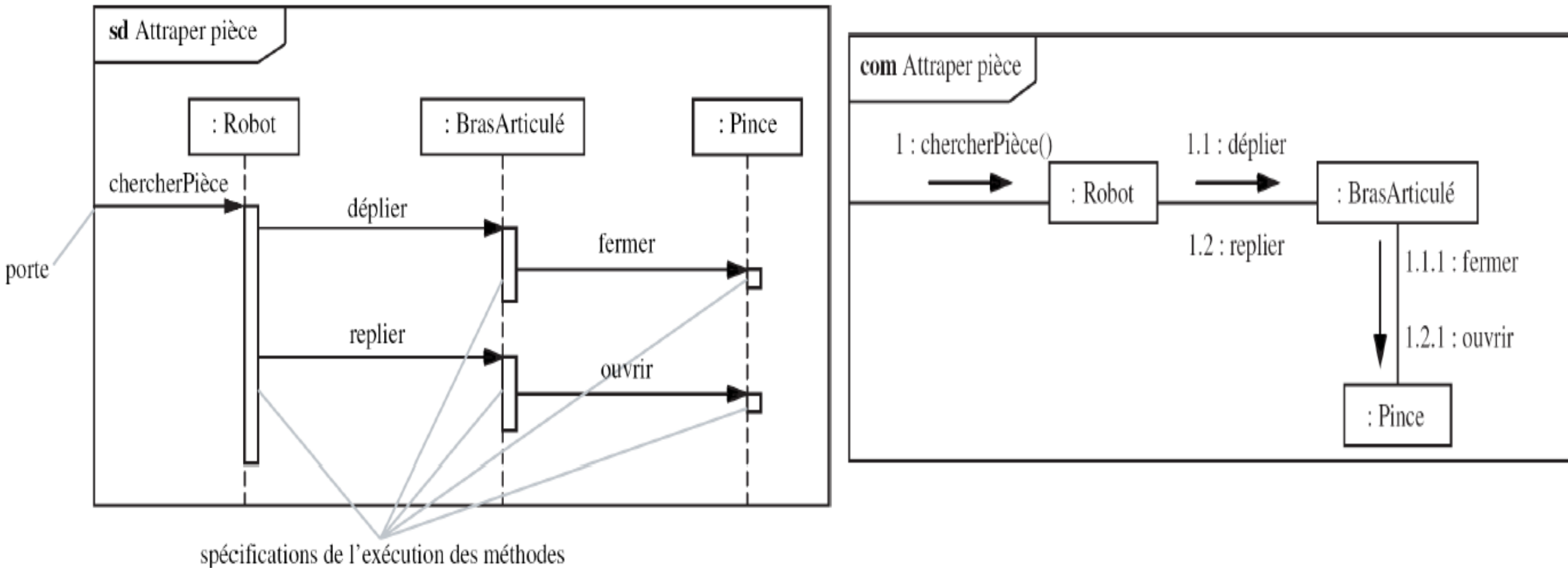


Diagramme de communication: les composants (Message Parallèles)

Lorsque des messages sont envoyés en parallèle, on les numérote avec des lettres

- 1.a, 1.b,... pour des messages simultanés envoyés en réponse à un message dont l'envoi portait le numéro 1

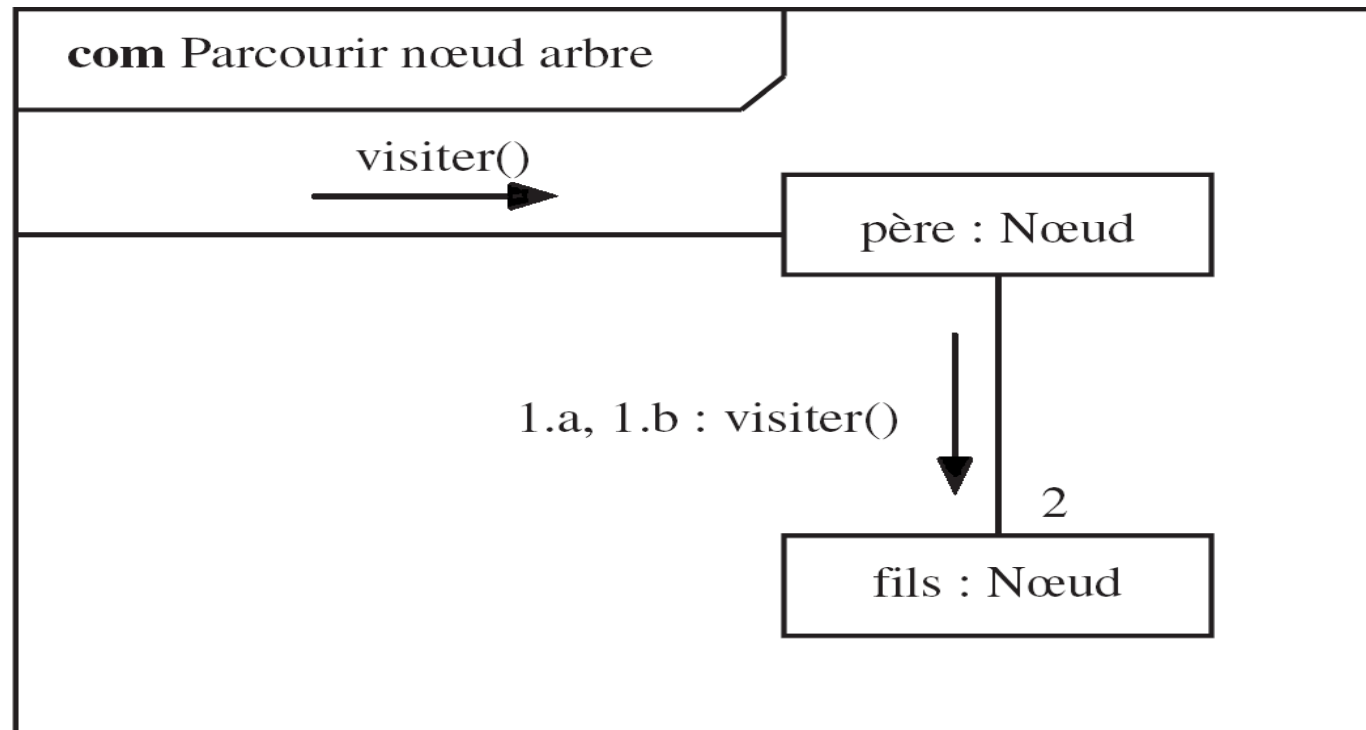


Diagramme de communication: les composants

(Opérateurs de choix et de boucles)

Pas d'opérateurs combinés dans les diagrammes de communication

- `*[<clauseitération>]` représente une itération.

La clause d'itération peut être exprimée dans le format `i:=1..n`

Si c'est une condition booléenne, celle-ci représente la condition d'arrêt

- `*[<clauseitération>]` représente un choix. La clause de condition est une condition booléenne.