



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

DIAGRAMME DE SEQUENCE



Présenté par :

Chokri Salah

Niba Amine

Zahim Jamal

Formateur :

Bouhlali

2022/2023



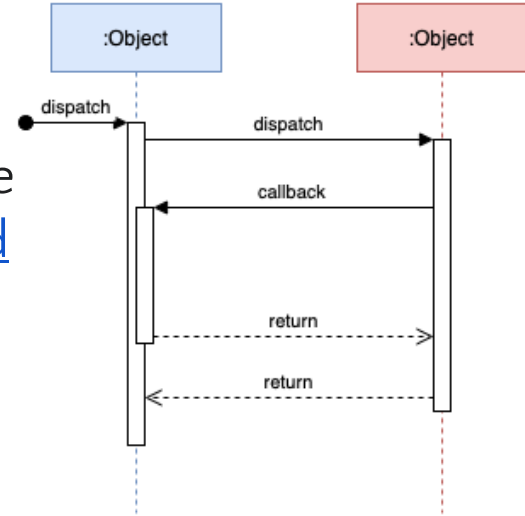
Plan :

- ① 1 . Définition du diagramme de séquences
- ② 2 . Avantages des diagrammes de séquence
- ③ 3 . Composants et symboles élémentaires
- ④ 4 . Les Types de messages
- ⑤ 5. Fragments d'interactions combinés

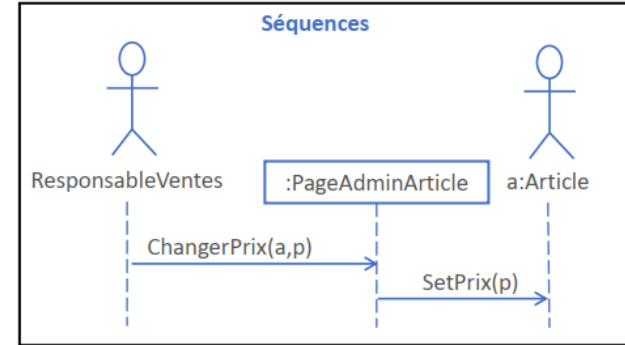
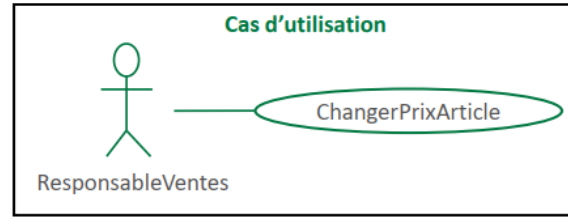


1 – Définition du diagramme de séquences

Les **diagrammes de séquences** sont la représentation graphique des [interactions](#) entre les [acteurs](#) et le système selon un ordre chronologique dans la formulation [Unified Modeling Language](#). Ces diagrammes sont utilisés à la fois par les développeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau système ou documenter un processus existant



❖ Modéliser les interactions



Classes pour spécifier les opérations nécessaires



- Pour être complètement spécifiée, une interaction doit être décrite dans plusieurs diagrammes UML :



2 . Avantages des diagrammes de séquence

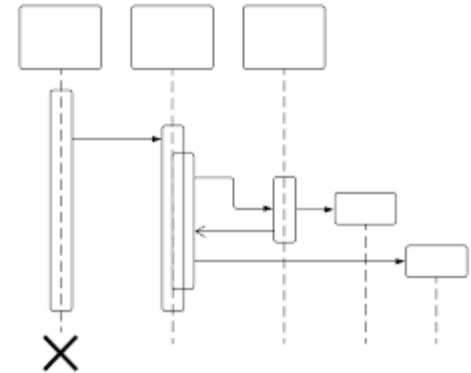
- **Les diagrammes de séquence peuvent constituer des références utiles pour les entreprises et d'autres organisations. Essayez de dessiner un diagramme de séquence pour :**
 - Représenter les détails d'un cas d'utilisation UML
 - Modéliser le déroulement logique d'une procédure, fonction ou opération complexe
 - Voir comment les objets et les composants interagissent entre eux pour effectuer un processus.
 - Schématiser et comprendre le fonctionnement détaillé d'un scénario existant ou à venir


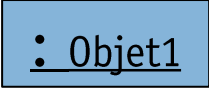








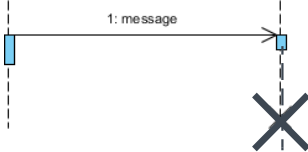
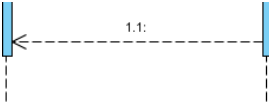

3 . Composants et symboles élémentaires

Pour comprendre ce qu'est un diagramme de séquence, vous devez connaître ses symboles et ses composants. Les diagrammes de séquence sont composés des icônes et des éléments suivants :

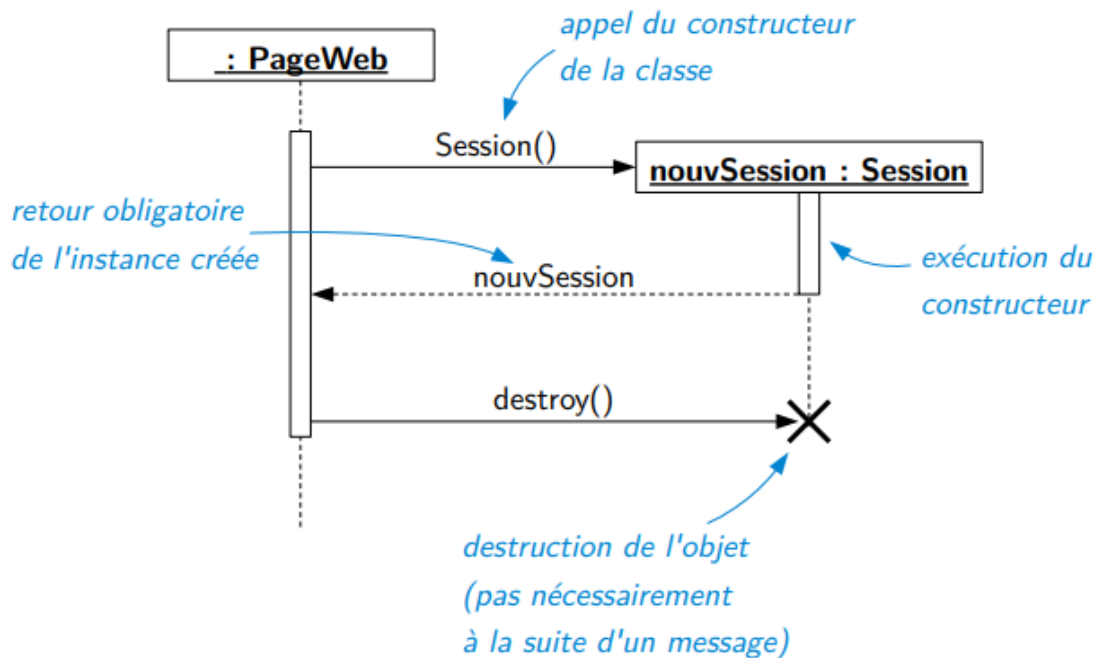


Symbole	Notation Description
	<p>Acteur</p> <p>Montre les entités qui interagissent avec le système ou qui sont extérieures à lui.</p>
	<p>Objet</p> <p>les objets sont des instances des classes, et sont rangés horizontalement. La représentation graphique pour un objet est similaire à une classe (un rectangle) précédée du nom d'objet (facultatif) et d'un point-virgule (:) .</p>
	<p>Noter</p> <p>Cette entité permet d'inscrire un commentaire sur le <i>diagramme de séquence système</i>. Ce commentaire peut être relié à toute entité du diagramme.</p>

Symbole	Notation Description
	<p>Ligne de vie les lignes de vie, <i>LifeLine</i>, identifient l'existence de l'objet par rapport au temps. La notation utilisée pour une ligne de vie est une ligne pointillée verticale partant de l'objet.</p>
	<p>Activation les activations, sont modélisées par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action.</p>
	<p>Message les messages, modélisés par des flèches horizontales entre les activations, indiquent les communications entre les objets.</p>

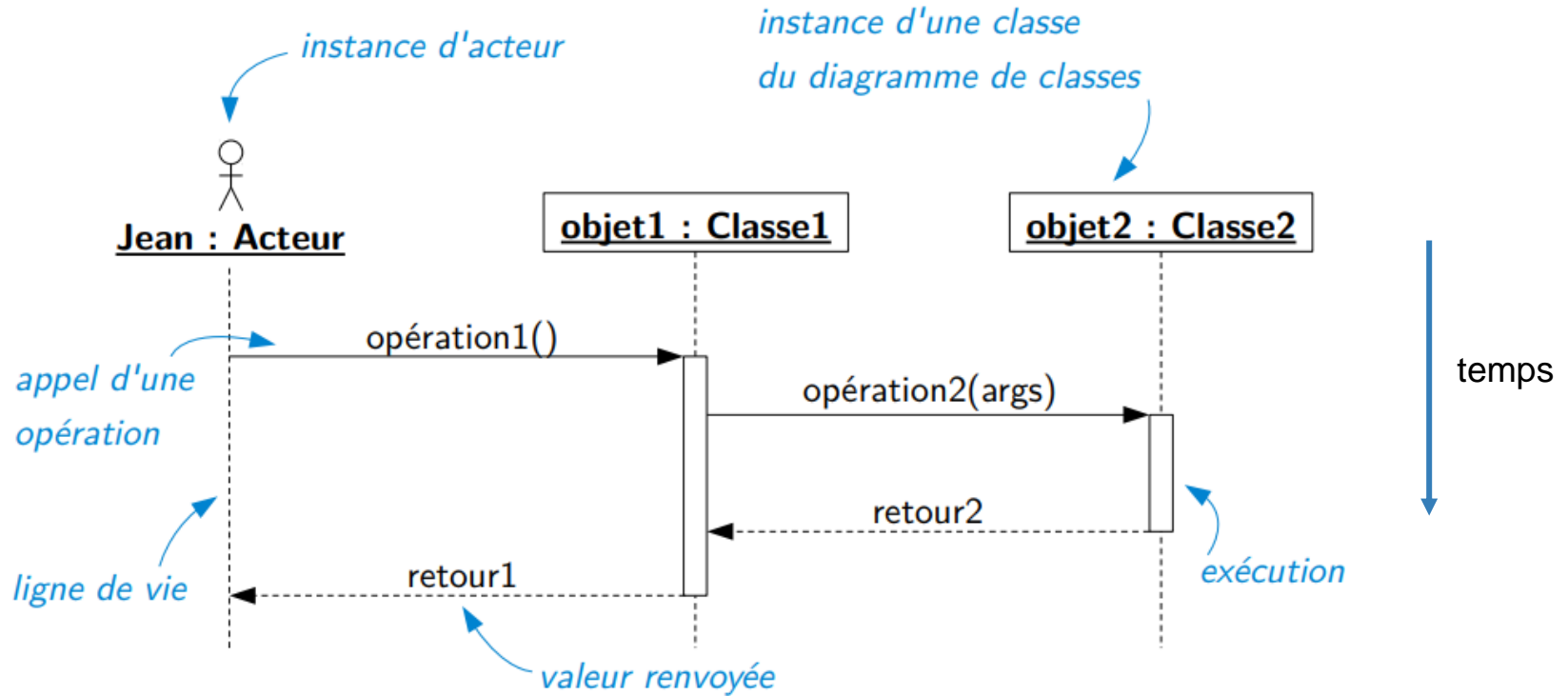
Symbole	Notation Description
 <p>1: message</p>	<p>Symbole de messages de suppression Représentés par une ligne pleine terminée par une pointe de flèche pleine, suivie du symbole X. Ces messages détruisent un objet</p>
 <p>1:1:</p>	<p>Symbole de messages de réponse Représentés par une ligne en pointillés terminée par une pointe de flèche, ces messages sont des réponses aux appels.</p>
 <p><<create>></p>	<p>Symbole de messages de création asynchrones Représentés par une ligne en pointillés terminée par une pointe de flèche. Ces messages créent de nouveaux objets.</p>

Création et destruction d'objet





Éléments de base

Exemple :





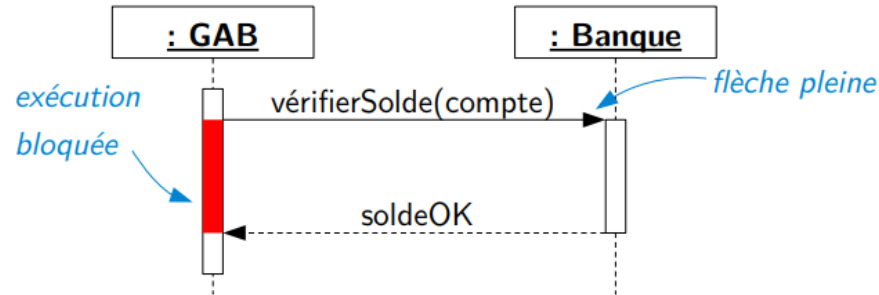
4 . Les Types de messages

Symbole	Notation Description
	<p>Symbole de messages asynchrones</p> <p>Représentés par une ligne pleine terminée par une pointe de flèche. Les messages asynchrones ne nécessitent pas de réponse avant que l'expéditeur ne continue. Seul l'appel doit être inclus dans le diagramme</p>
	<p>Symbole de messages synchrones</p> <p>Représentés par une ligne pleine terminée par une pointe de flèche pleine. On utilise ce symbole lorsqu'un expéditeur doit attendre une réponse à un message avant de continuer. Le diagramme doit montrer à la fois l'appel et la réponse.</p>

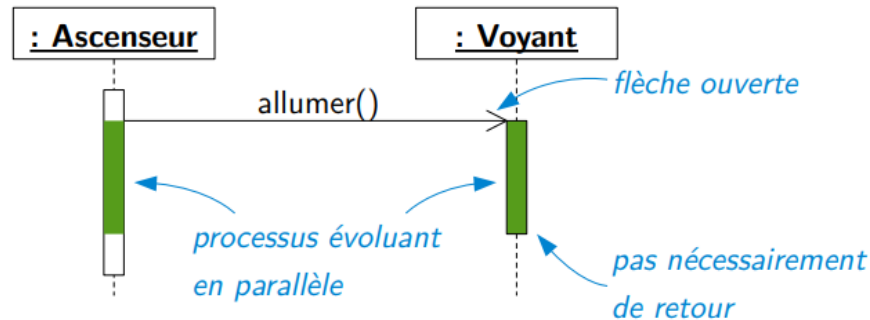
Exemple :

Types de messages


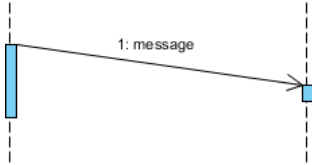
Message synchrone : Émetteur **bloqué** en attente du retour



Message asynchrone : Émetteur **non bloqué**, continue son exécution

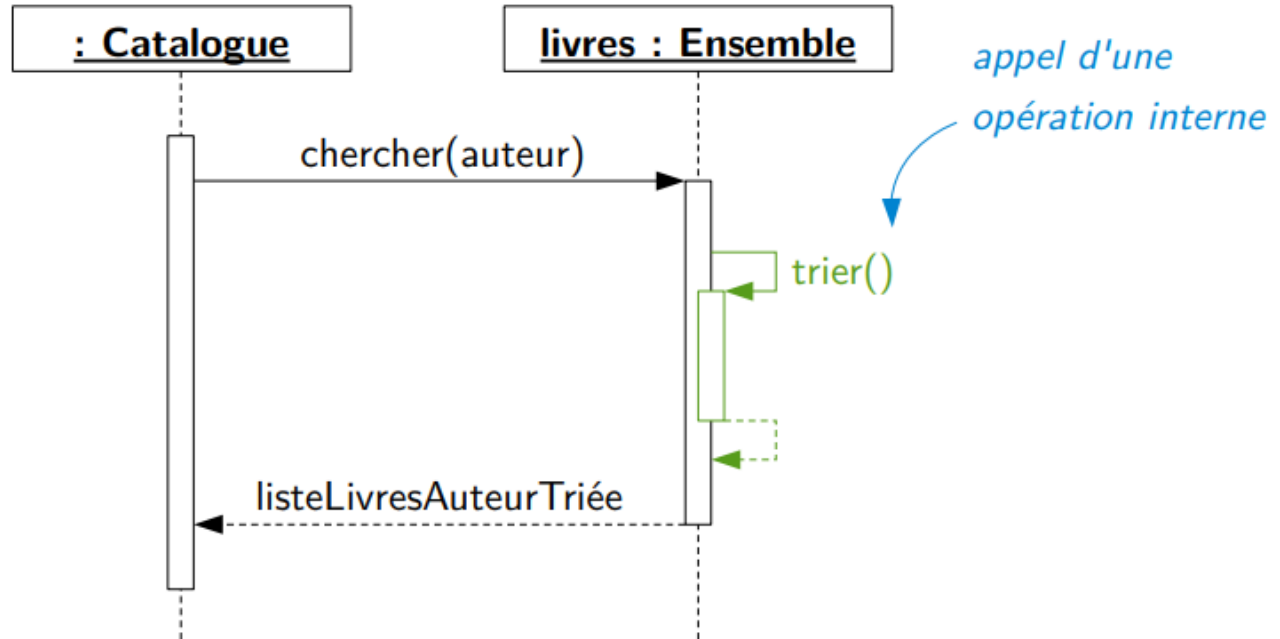




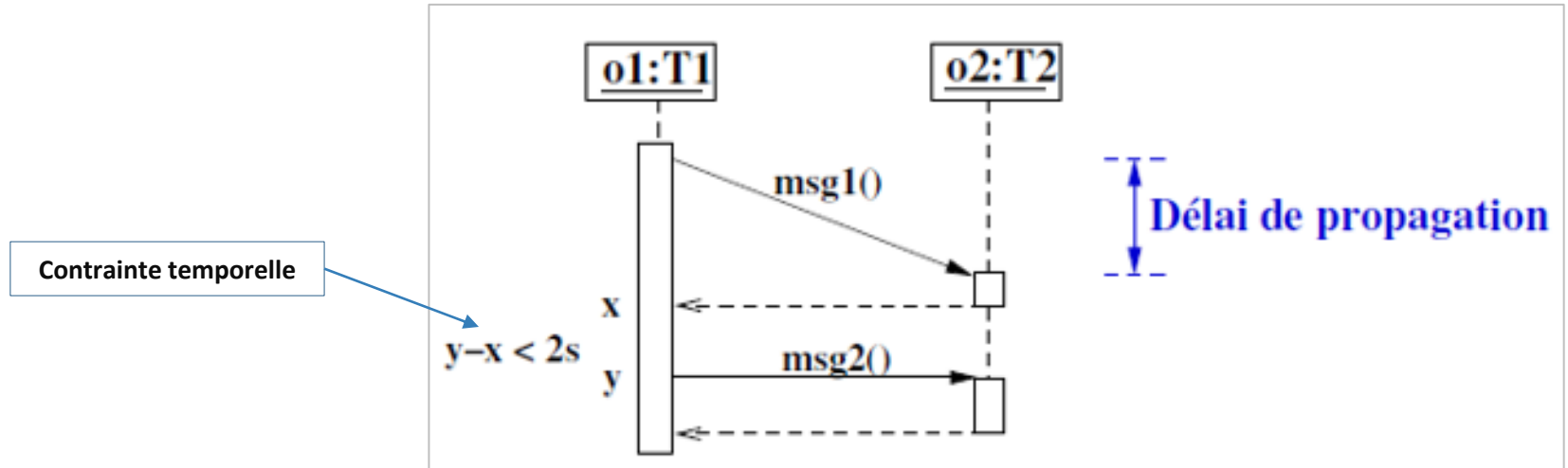
Symbole	Notation Description
	<p>Auto-message (réflexif)</p> <p>L'auto-message est une sorte de message qui représente l'invocation du message de la même ligne de vie.</p>
	<p>Duration message</p> <p>Le message de durée indique la distance entre deux instants pour l'appel d'un message.</p>

Example :

Message réflexif



❑ Messages synchrones – Contraintes temporelles





❖ Autres types – Message perdu et trouvé

- 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.



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



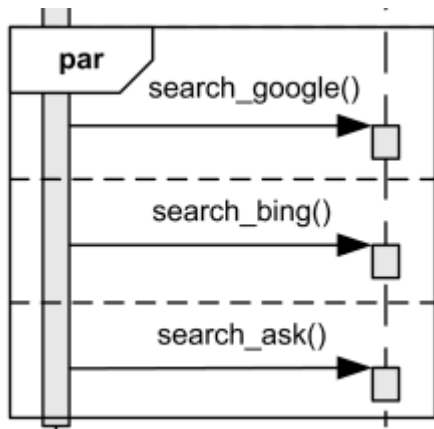


5. Fragments d'interactions combinés

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

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)



❖ Operator Fragment Type

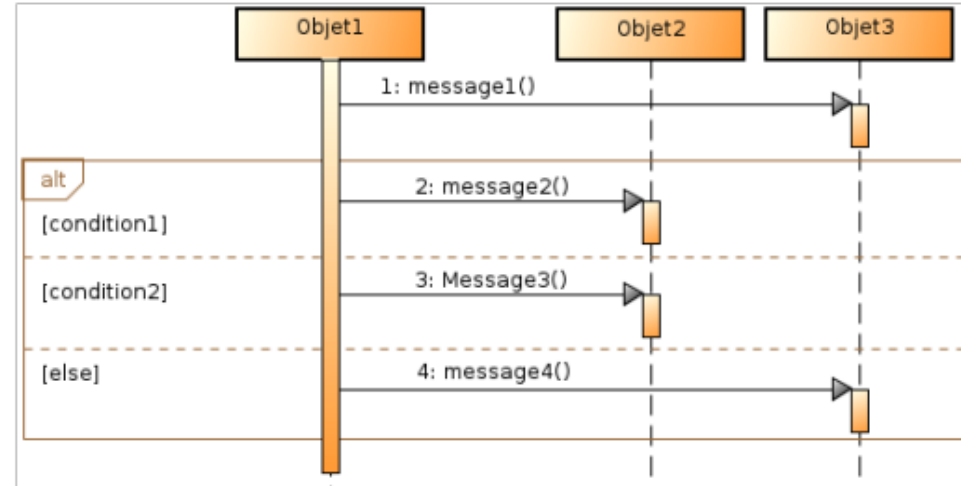
- **alt** : fragments multiple alternatifs (si alors sinon)
- **opt** : fragment optionnel
- **par** : fragment parallèle (traitements concurrents)
- **Loop** : le fragment s'exécute plusieurs fois
- **region** : région critique (un seul thread à la fois)
- **neg** : une interaction non valable
- **break** : représente des scenario d'exception
- **ref** : référence à une interaction dans un autre diagramme
- **sd** : fragment du diagramme de séquence en entier

❖ La liste suivante regroupe les opérateurs d'interaction par fonctions :

- **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 ;

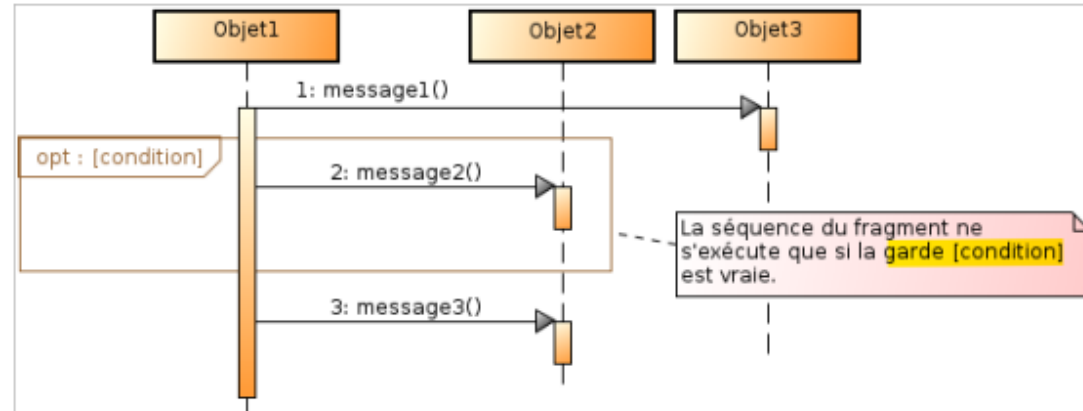
❖ Représentation d'un fragment avec l'opérateur alt

- L'opérateur *alternative*, ou *alt*, est un opérateur conditionnel possédant plusieurs opérandes séparés par des pointillés.



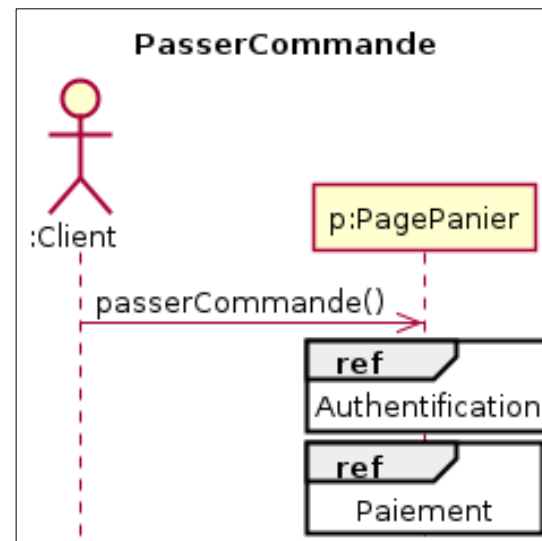
❖ Représentation d'un fragment avec l'opérateur *opt*

- L'opérateur *option*, ou *opt*, comporte une opérande et une condition de garde associée.



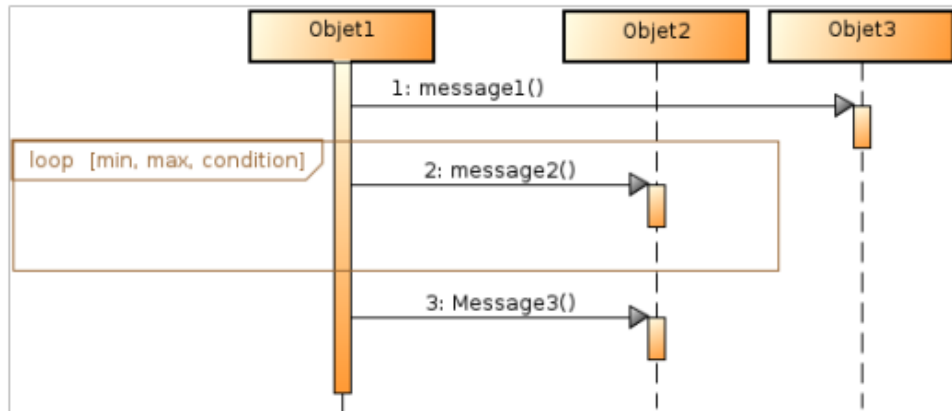
❖ Représentation d'un fragment avec l'opérateur ref

- Un fragment ref permet d'indiquer la réutilisation d'un diagramme de séquences défini par ailleurs. En supposant qu'il existe un diagramme intitulé Authentification et un autre Paiement.



❖ Représentation d'un fragment avec l'opérateur loop

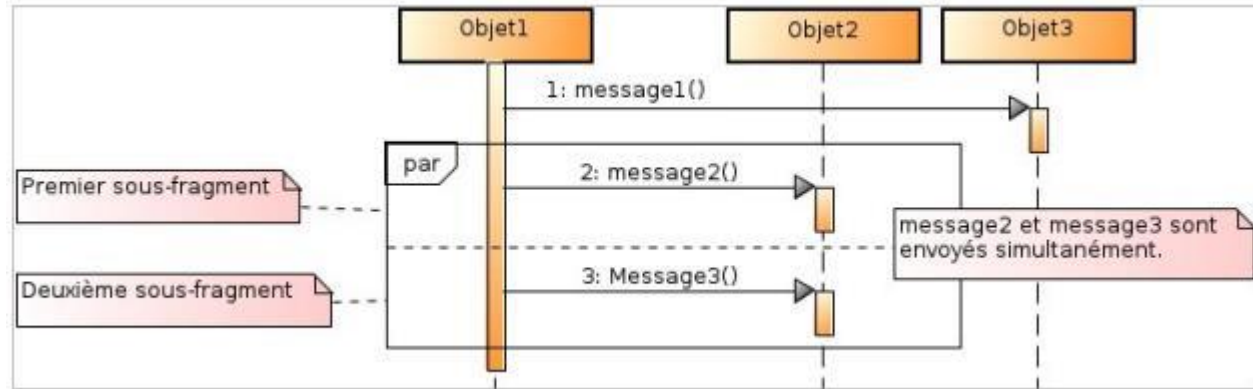
- L'opérateur de boucle (*loop*) exécute la séquence contenue dans le fragment tant que la garde qui lui est associée est vraie.





❖ Représentation d'un fragment avec l'opérateur *par*

- Un fragment d'interaction avec l'opérateur de traitements parallèles (*par*) contient au moins deux sous fragments (opérandes) séparés par des pointillés qui s'exécutent simultanément (traitements concurrents).





❑ Sources

- ◎ <https://creately.com/blog/>
- ◎ <https://laurent-audibert.developpez.com/>
- ◎ <https://lipn.univ-paris13.fr/>
- ◎ <https://fr.wikipedia.org/>
- ◎ <https://www.lucidchart.com/pages/>
- ◎ <https://www.visual-paradigm.com/guide/>
- ◎ <https://www.youtube.com/>



Merci !

Des *questions* ?