



MODÉLISATION DES INTERACTIONS

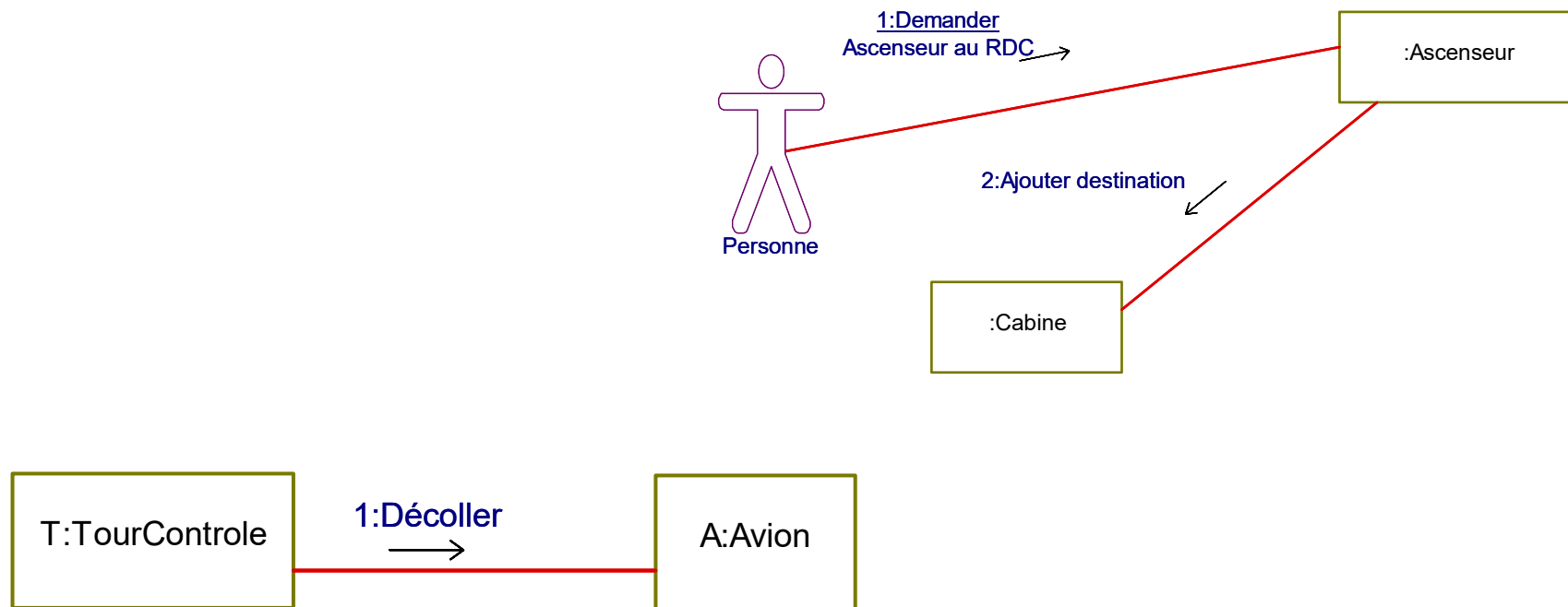
DIAGRAMMES DE SÉQUENCE

DIAGRAMMES DE COMMUNICATION



DÉFINITIONS

- Une interaction est un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.





MESSAGE

- Un message est une communication entre objets. La réception d'un message peut produire un résultat.
- Types de messages:
 - Flux de contrôle
 - Envoi d'un signal: Déclenche une réaction chez le récepteur de façon asynchrone et sans qu'une réponse ne soit nécessaire.
 - appel d'une opération: applique une opération à un objet de façon synchrone ou asynchrone, et peut requérir une réponse de la part de son récepteur.
 - Actions
 - création d'un objet
 - suppression d'un objet existant
- Ces messages correspondent aux stimulus dans le langage UML.
- Un message a un émetteur, un récepteur et une action. L'action est exécutée sur le récepteur. Vous pouvez également créer des messages récursifs, c'est-à-dire des messages dont l'émetteur est en même temps le récepteur.

Message synchrone →

Message asynchrone →

← message de retour

- 
- 
- UML fournit deux types de représentation des interactions :
 - Les **diagrammes de communication**
 - Les **diagrammes de séquence**
 - Les deux types de diagrammes sont équivalents. La différence entre ces deux diagrammes est une différence de centre d'intérêt. Les diagrammes de communication mettent en évidence les liens entre objets et les séquences d'envois de messages sur ces liens. Les diagrammes de séquence mettent en évidence la chronologie des envois de messages.
 - Un sd (sequence diagram) représente les interactions d'un point de vue temporel, alors qu'un com (communication diagram) représente les interactions d'un point de vue spatial.

SPÉCIFICATION D'UN MESSAGE

- Le symbole d'un message est une flèche qui montre son sens, et qui peut également afficher les informations suivantes :

- Un numéro d'ordre qui indique l'ordre dans lequel les messages sont échangés

Exemple: : l'envoi du message 1.2.3 suit immédiatement celui du message 1.2.2 et ces deux messages font partie du flot (de la famille de messages) 1.2.

Le nom du message (ou nom de l'opération associée)

- La condition: [<clause de condition>], condition d'émission du message.
- La valeur de retour:
- Les arguments

[cond] [numéro]: [retour :=] msg([paraètres])

EXEMPLE DE DIAGRAMME DE COMMUNICATION

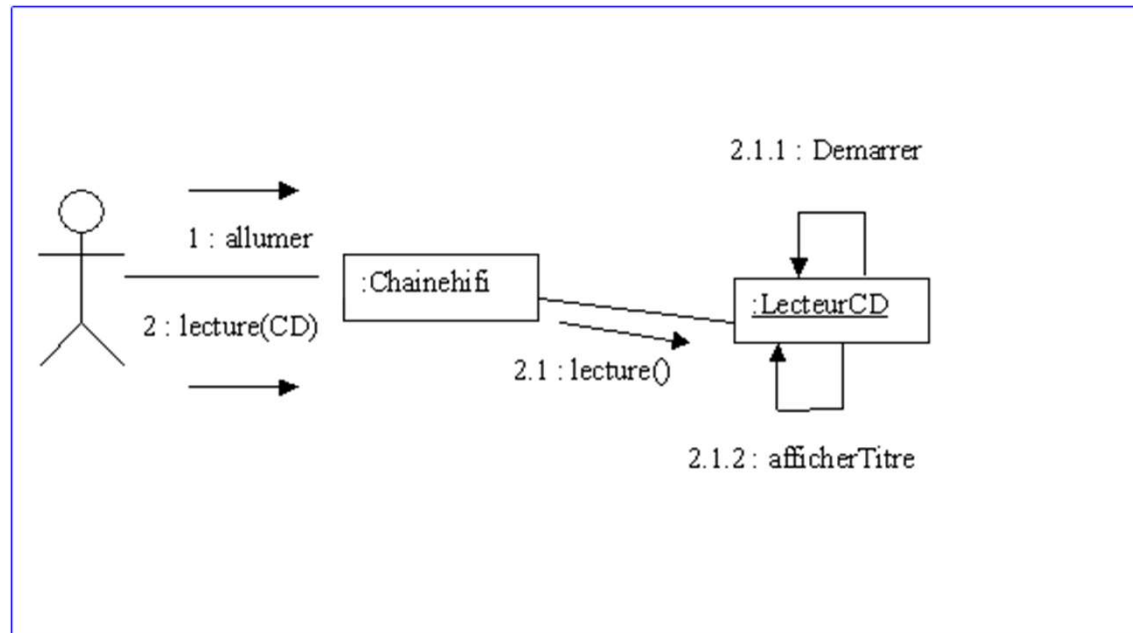


DIAGRAMME DE SÉQUENCE

- Le diagramme de séquence est une description graphique des opérations d'un système sous un angle chronologique. C'est une vue dynamique qui contient les symboles d'objets (instances de classe), d'acteurs et de messages qu'ils échangent.
- La dimension verticale est l'axe temporel : les messages y sont représentés par ordre chronologique. La dimension horizontale montre des objets et des acteurs qui échangent des informations.

INTRODUCTION

- Une interaction est un comportement qui comprend un ensemble de messages échangés par un ensemble d'éléments (lignes de vie) dans un certain contexte pour accomplir une certaine tâche.
- Le diagramme de séquence est une description graphique d'une interaction sous un angle chronologique.
- Une ligne de vie possède un nom et un type

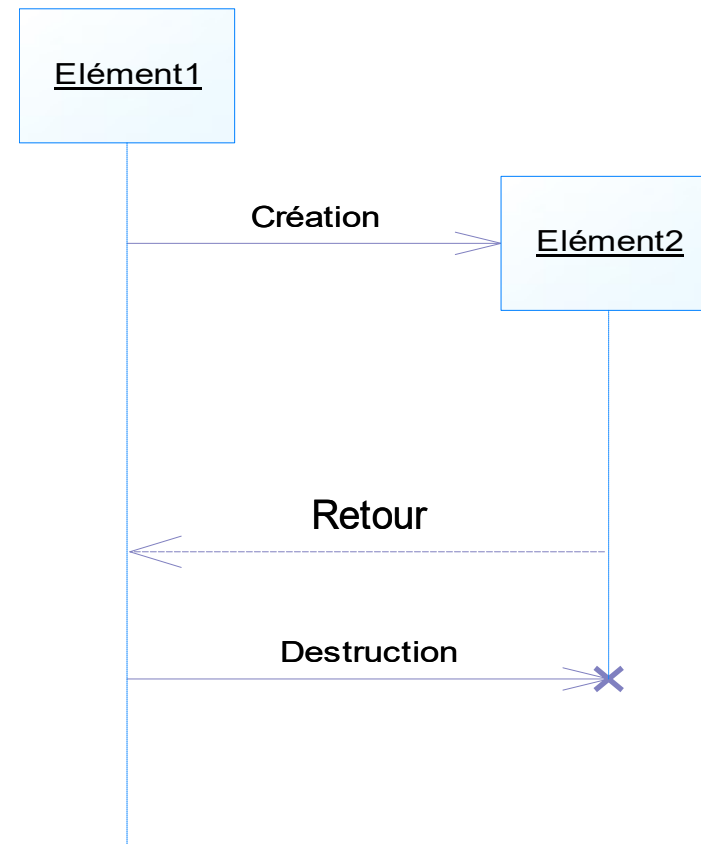
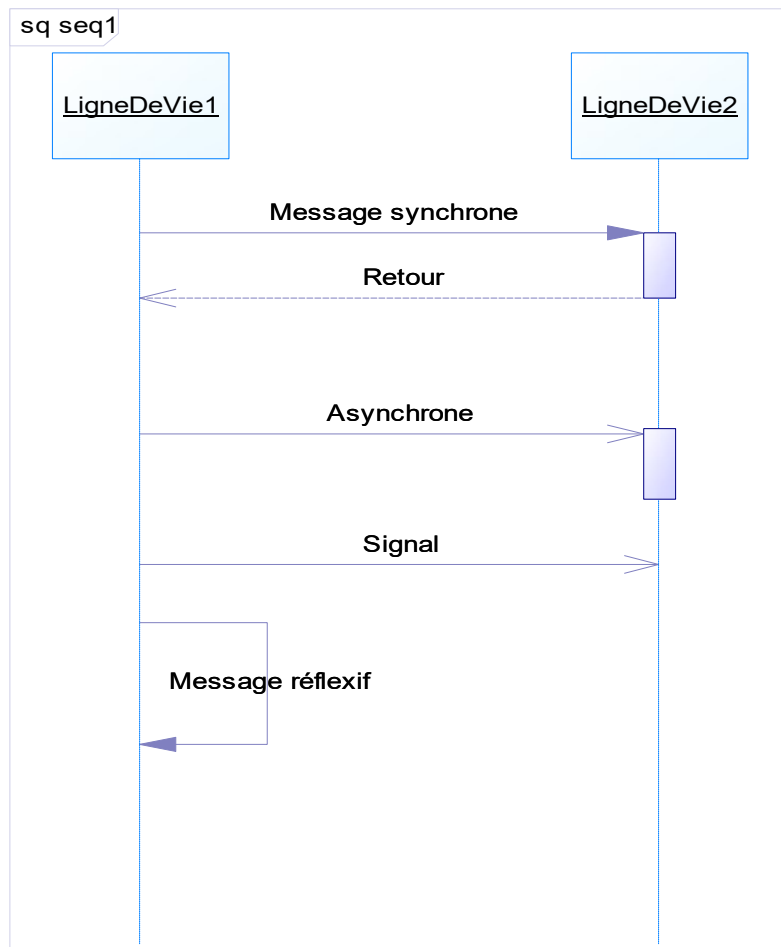


FORMALISME

- Les éléments représentés sur un diagramme de séquence sont:
 - Les **lignes de vie** qui participent à l'interaction,
 - Les **messages** entre les lignes de vie,
 - Une **période d'activité (une activation ou exécution)** correspond au temps pendant lequel une ligne de vie effectue une action. Les périodes d'activité se représentent par des boîtes rectangulaires placées sur les lignes de vie. Le début et la fin d'une boîte correspondent respectivement au début et à la fin d'une période d'activité,
 - **Les fragments d'interaction:**
- **Les contraintes temporelles** : Si l'émission d'un message requiert une certaine durée, il se représente sous la forme d'un trait oblique

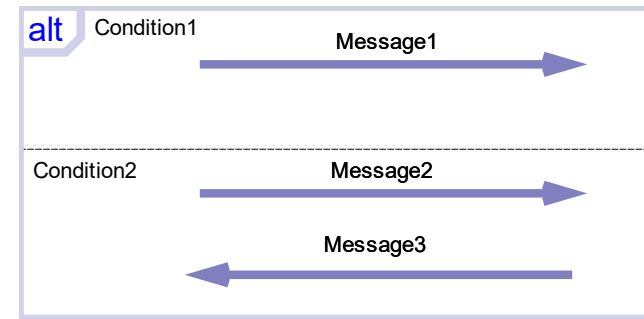
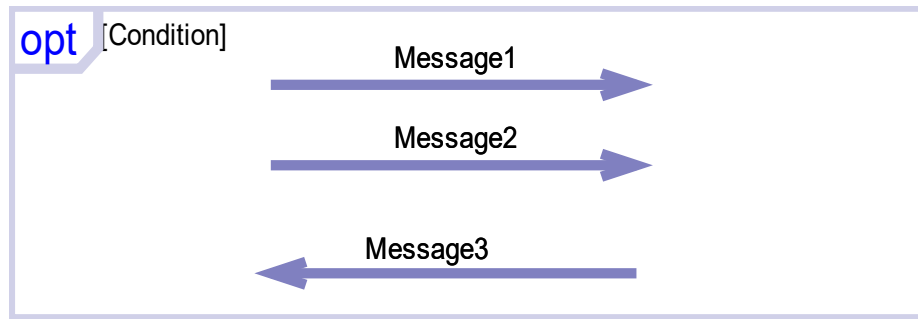


TYPES DE MESSAGES



FRAGMENTS D'INTERACTION

- Un fragment combiné se représente de la même façon qu'une interaction : par un rectangle dont le coin supérieur gauche contient un pentagone.
- Les conditions de choix des opérandes sont données par des expressions booléennes entre crochets.
- Opérateurs
 - alt (alternative): plusieurs fragments
 - opt (optionnel): le fragment s'exécute seulement si la condition est vraie.



LOOP

La syntaxe d'une boucle est la suivante :

```
loop ['( <min> ['> <max> ] )']
```

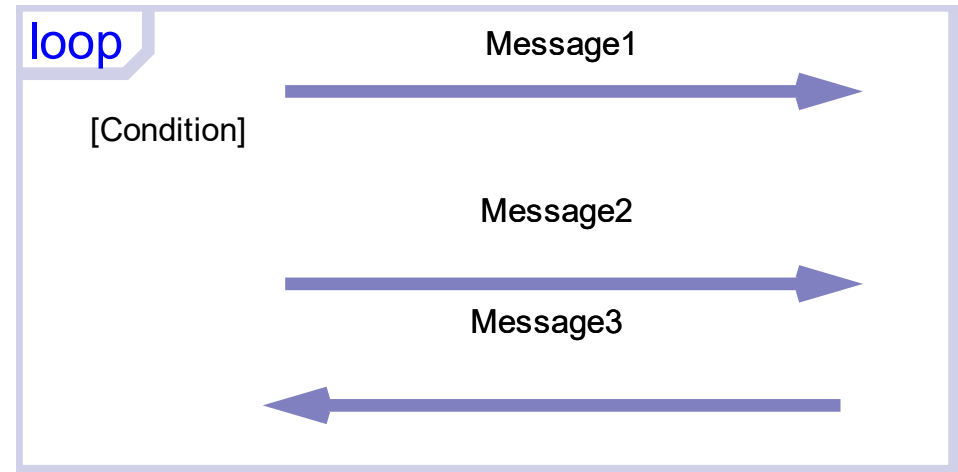
où la boucle est répétée au moins min fois avant qu'une éventuelle condition

booléenne ne soit testée (la condition est placée entre crochets sur la ligne de vie) ;

tant que la condition est vraie, la boucle continue, au plus max fois (min est un entier supérieur ou égal à 0, max est un entier supérieur ou égal à min).

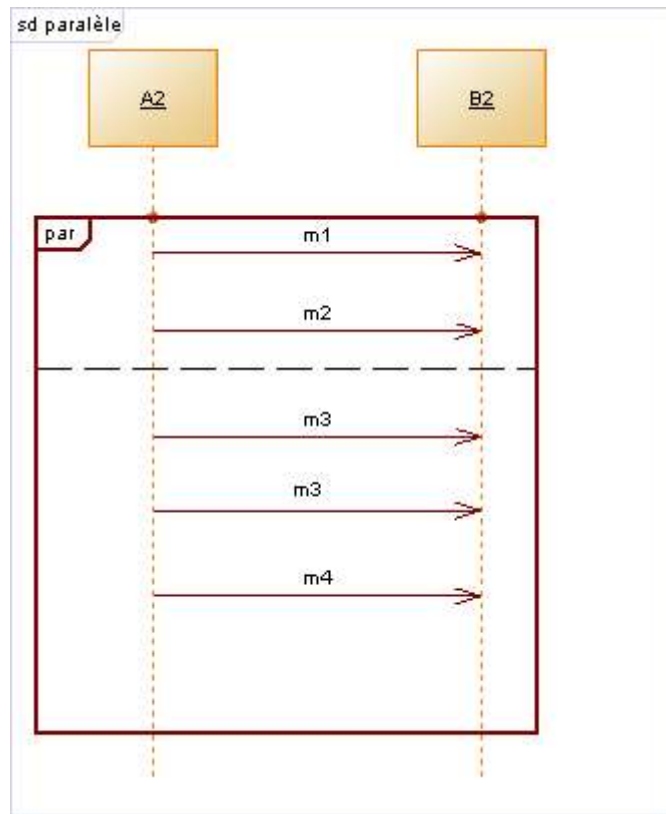
loop(valeur) est équivalent à loop(valeur, valeur).

loop est équivalent à loop(0, *), où * signifie « illimité ».

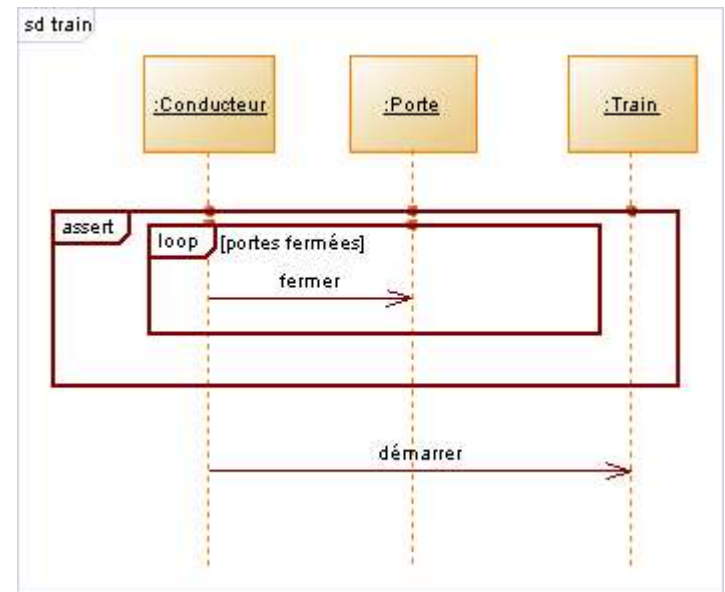


PAR

par: le fragment est scindé en plusieurs régions, toutes seront exécutées en parallèle

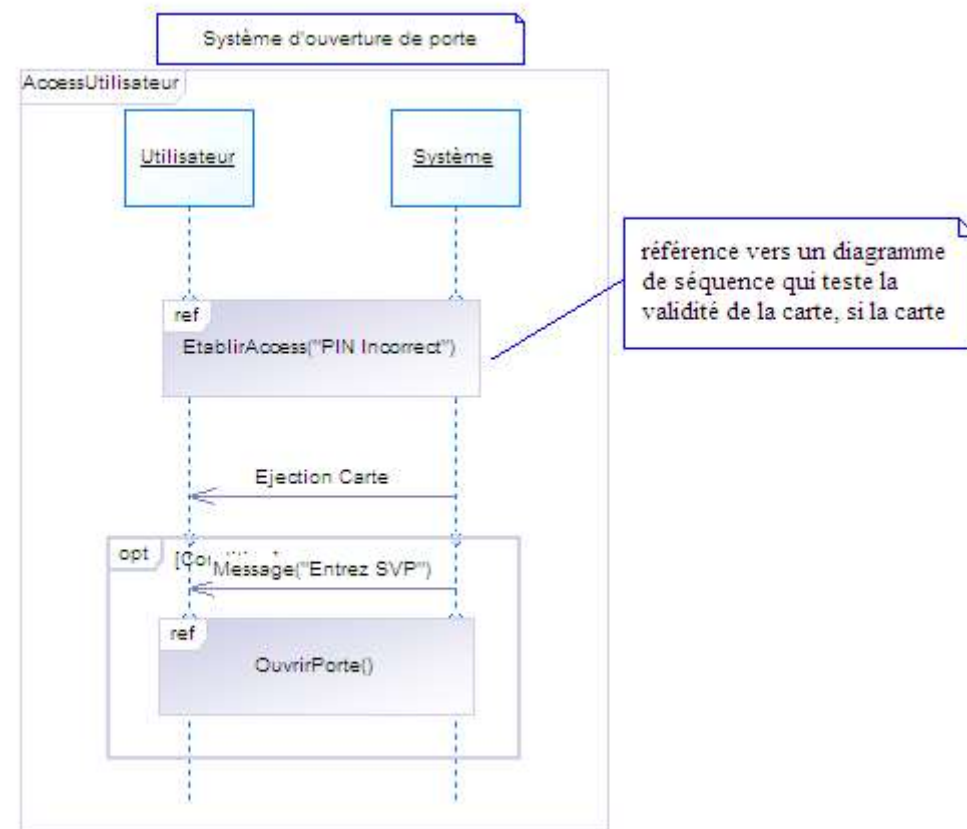


assert: l'interaction doit se produire exactement tel qu'indiqué sous peine d'être invalide



RÉFÉRENCE D'INTERACTION

- Syntaxe:
[<nomAttributPourValeurRetour> '=']
<nom de l'interaction> [' ('
[<argument>'],' ... ')'] [':' <valeur de
retour>]
- La valeur de retour est affectée à un attribut. Les arguments peuvent être en entrée (ils portent alors la marque in), en sortie (out), ou en entrée et en sortie (inout).



EXEMPLES

