

École Nationale des Sciences
Appliquées d'Oujda



Unified Modeling Language

Chapitre 4 :

Diagramme de séquence

28 mars 2021

Plan du cours

- 1 **Définition et utilité**
- 2 Les composants du diagramme
- 3 Les fragments d'interactions

Plan du cours

- 1 Définition et utilité
- 2 Les composants du diagramme
- 3 Les fragments d'interactions

Plan du cours

- 1 Définition et utilité
- 2 Les composants du diagramme
- 3 Les fragments d'interactions

Plan du cours

- 1 Définition et utilité
- 2 Les composants du diagramme
- 3 Les fragments d'interactions

Définition et utilité

Comment ?

- Les diagrammes de cas d'utilisation modélisent à **QUOI** sert le système, en organisant les interactions possibles avec les acteurs. Ils décrivent le comportement attendu du système
- Les diagrammes de classes permettent de spécifier la structure et les liens entre les objets dont le système est composé : ils spécifie **QUI** sera à l'oeuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation.
- Les diagrammes de séquences permettent de décrire **COMMENT** les éléments du système interagissent entre eux et avec les acteurs.
 - Les objets au cœur d'un système interagissent en s'échangeant des messages.
 - Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Définition et objectif

Objectif

L'objectif du diagramme de séquence est de représenter les **interactions** entre objets en indiquant la **chronologie** des échanges.

Les diagrammes de séquences peuvent servir à **illustrer un cas d'utilisation**.

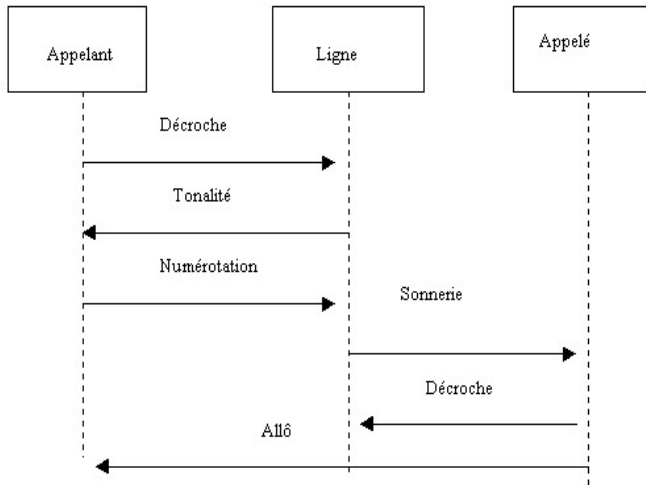
Définition et objectif

Objectif

- Permet de décrire les interactions entre différentes entités et/ou acteurs :
 - par exemple des objets dans un modèle d'un logiciel, des sous-systèmes dans un modèle d'un système complet.
- Des flèches représentent les messages qui transitent d'une entité vers l'autre. Le nom des message apparaît sur chaque flèche.
 - Si l'extrémité de la flèche est pleine, le message est synchrone.
 - Si l'extrémité de la flèche est creuse, le message est asynchrone.
- Se concentre sur l'échange de messages entre un certain nombre de lignes de vie

Définition et utilité

Exemple



Plan du cours

- 1 Définition et utilité
- 2 Les composants du diagramme**
- 3 Les fragments d'interactions

Les composants du diagramme

Les interactions comprennent principalement les éléments suivants :

- Les rôles (acteurs)
- Les instances (classe et objets)
- Les lignes de vie
- Les messages

Les composants du diagramme

Un diagramme de séquence a deux dimensions

- **Dimension verticale** : le temps

L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule « de haut en bas »

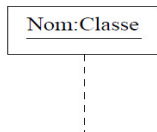
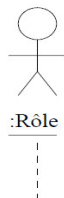
- **Dimension horizontale** : les objets (et les acteurs)

L'ordre de disposition des objets sur l'axe horizontal est sans importance

Les composants du diagramme

Il existe 2 types de participants

- Entités appartenant au système (instance d'une classe)
- Acteurs (utilisateurs du système)

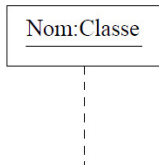


Les composants du diagramme

Objets

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 de deux points (:) .

Les objets qui sont impliqués dans les interactions sont placés sur l'axe des x. Les objets sont les émetteurs et les récepteurs des messages dans le diagramme de séquence.

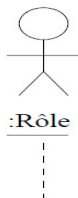


Les composants du diagramme

Acteurs

Les acteurs peuvent également communiquer avec des objets, ainsi ils peuvent eux aussi être énumérés en colonne.

Un acteur est modélisé en utilisant le symbole habituel

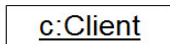


Les composants du diagramme

Ligne de vie

Une ligne de vie représente l'ensemble des opérations exécutées par un objet.

Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour.



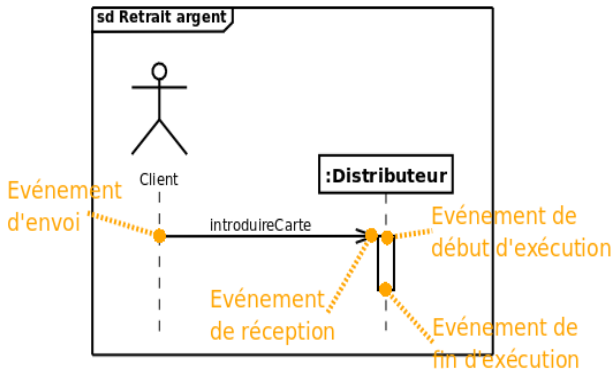
Les composants du diagramme

Message : définition

- Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.
- Un message définit une communication particulière entre des lignes de vie (objets ou acteurs). Plusieurs types de messages existent, dont les plus courants :
 - l'envoi d'un signal ;
 - l'invocation d'une opération (appel de méthode) ;
 - la création ou la destruction d'un objet.
- La réception des messages provoque une période d'activité (rectangle vertical sur la ligne de vie) marquant le traitement du message (spécification d'exécution dans le cas d'un appel de méthode).

Les composants du diagramme

Événement



Les composants du diagramme

Messages

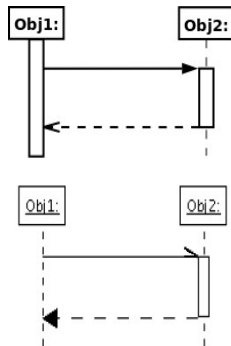
- **Message simple** : Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.
- **Message minuté (timeout)** : Bloque l'expéditeur pendant un temps donné, en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.
- **Message synchrone** : Bloque l'expéditeur jusqu'à prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur à la prise en compte du message.
- **Message asynchrone** : N'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

Les composants du diagramme

Messages

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

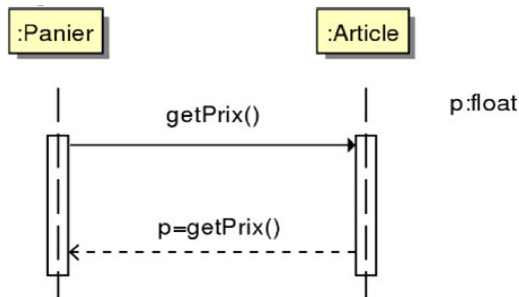
Un message **asynchrone** n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.



Les types de message

Message synchrone

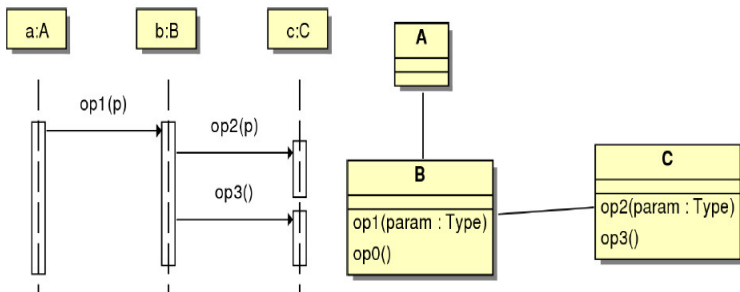
- Le récepteur d'un message synchrone rend la main à l'émetteur du message en lui envoyant un message de retour
- Les messages de retour sont optionnels : la fin de la période d'activité marque également la fin de l'exécution d'une méthode.
- Ils sont utilisés pour spécifier le résultat de la méthode invoquée.



Les types de message

Message synchrone

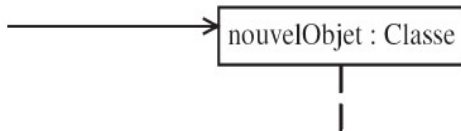
- Les messages synchrones correspondent à des opérations dans le diagramme de classes.
- Envoyer un message et attendre la réponse pour poursuivre son activité revient à invoquer une méthode et attendre le retour pour poursuivre ses traitements.



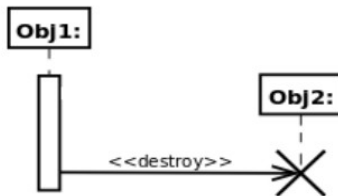
Les types de message

Message de création et message de destruction

- La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie. On peut aussi utiliser un message asynchrone ordinaire portant le nom «create» .



- La destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet.



Les types de message

Message perdu et message trouvé

- 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.

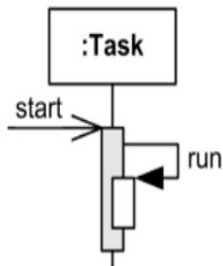


- Un message trouvé est tel que l'événement de réception est connu, mais pas l'événement d'émission.



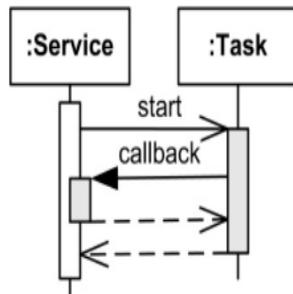
Les types de message

Message auto et message de rappel



Exécution sur la même ligne de vie
auto message

Message de rappel.

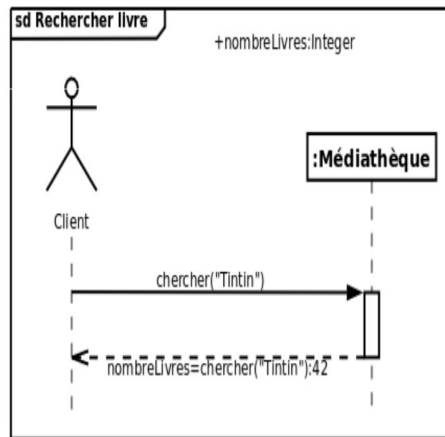


Message : syntaxe

Syntaxe de la déclaration d'un message

[attribut =] message([Arguments]) [: valeurDeRetour]

- La syntaxe des arguments est la suivante : nomParametre = valeurParametre
- Pour un argument modifiable : nomParametre : valeurParametre
- Exemples :
 - appeler("Capitaine Hadock", 54214110)
 - afficher(x,y)
 - initialiser(x=100)
 - f(x :12)



Plan du cours

- 1 Définition et utilité
- 2 Les composants du diagramme
- 3 Les fragments d'interactions**

Les fragments d'interactions

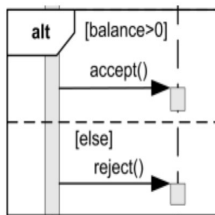
Définition

- 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 par un rectangle dont le coin supérieur gauche contient un pentagone.
- types de fragments
 - 10 types.

Les fragments d'interactions

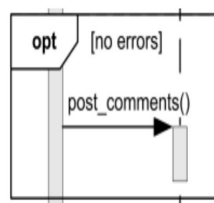
alternative & option

- Alternative : utilisés pour désigner un choix mutuellement exclusifs entre deux ou plusieurs séquences de messages



Appelez accepter () si $balance > 0$, appeler rejeter () sinon.

- Option : utilisé pour modéliser une séquence qui, étant donné une certaine condition, se produira ; sinon, la séquence ne se produit pas. (simple "if then" statement)

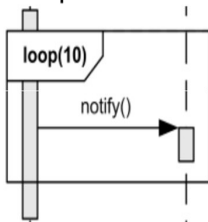


Postez des commentaires s'il n'y avait pas d'erreurs.

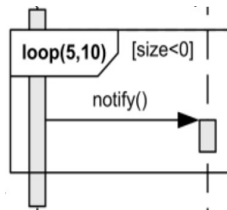
Les fragments d'interactions

boucle

Loop : modéliser une séquence répétitive.



Boucle qui s'exécute exactement 10 fois.

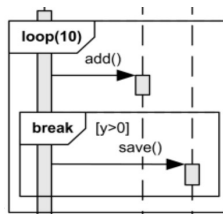


La boucle devrait s'exécuter minimum 5 fois, Et pas plus de 10 fois. Si l'état de la taille [size<0] devient plus petite que 0 la boucle se termine indépendamment du nombre minimum d'itérations spécifié.

Les fragments d'interactions

break

- Break : représente un scénario de rupture ou d'exception qui est exécuté à la place du reste du fragment d'interaction enveloppante.

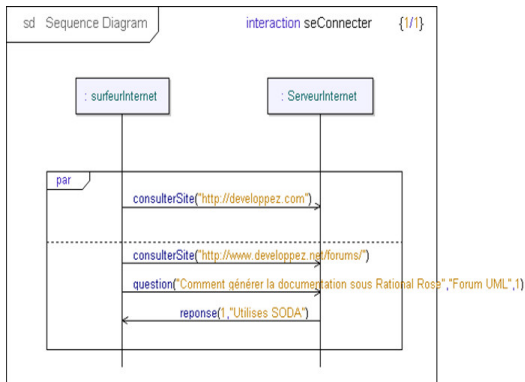


Arrêter la boucle si $y > 0$

Les fragments d'interactions

parallèle

- Parallel : Exécution parallèle

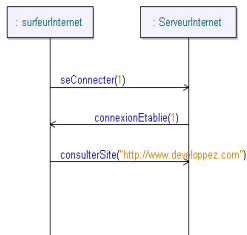


Les fragments d'interactions

Consider & Ignore

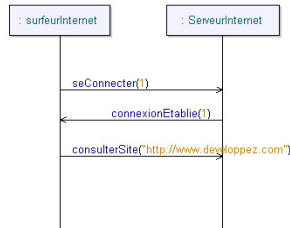
- Consider : les messages à être pris en compte dans un flux
- Ignore : messages à ignorer dans un flux

sd seConnecter consider {seConnecter,consulterSite}



interaction seConnecter

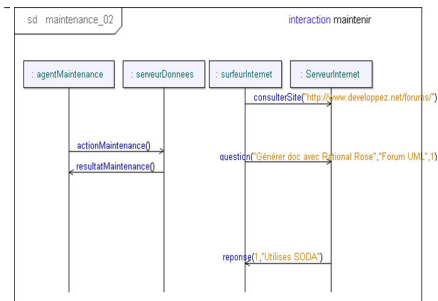
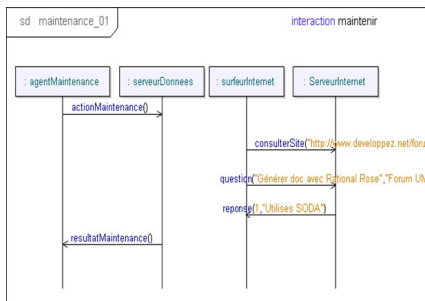
sd seConnecter ignore connexionEtablie



Les fragments d'interactions

Séquence

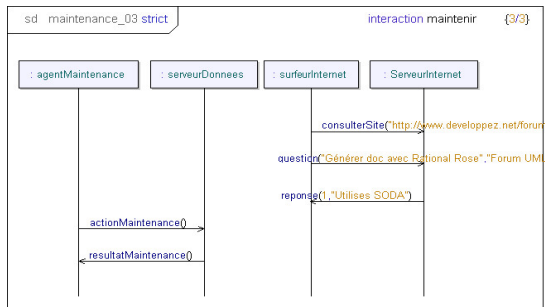
- Weak Sequencing (seq) : Exécution séquentielle d'interaction entre les même lifeline, tandis que les interactions présentes sur des "lignes de vie" (lifeline) différentes dans des opérandes différents peuvent arriver dans n'importe quel ordre ;



Les fragments d'interactions

Séquence

- Strict Sequencing (strict) : Séquencement ordonné pour tout le diagramme
- Assert : Une assertion désigne une proposition que l'on avance et que l'on soutient comme vraie. La séquence décrite dans l'opérande désigne l'unique séquence possible.
- Référence à un autre diagramme de séquence ; combiner (réutilisation) les séquences simples pour produire des séquences plus complexes.



:Web Customer

