

Diagrammes d'interaction

Objectif des diagrammes d'interaction (1)

- Les **diagrammes de classe** apporte une première réponse à la **modélisation statique** d'un système. Ils permettent de définir les **packages**, les **classes**, les **noms**, les **attributs** et les **signatures des méthodes**, mais pas le corps de celles-ci.
- Ce corps des méthodes est la transcription des interactions entre les objets dans le cadres des scénarios des Cas d'Utilisation.



- **Question** : On souhaite **décrire un scénario d'un cas d'utilisation** de façon à mettre en évidence les interactions entre les objets du système.
- Un scénario d'un cas d'utilisation est **réalisé** par une **collaboration**.
 - **Collaboration** = ensemble d'objets qui interagissent afin d'accomplir une tâche, selon un scénario de Cas d'Utilisation.
 - **Interaction** = échange de messages entre objets du système
 - Plusieurs collaboration peuvent donc être nécessaires pour réaliser un cas d'utilisation.

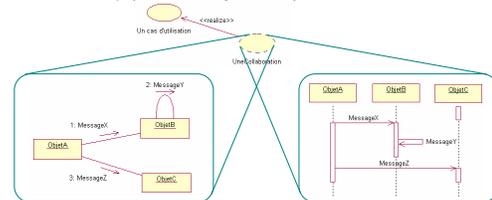
1

Diagrammes d'interaction

Objectif des diagrammes d'interaction (2)

- UML propose deux types de **diagrammes d'interaction** permettant de modéliser cette collaboration :

- **Diagramme de séquence** : interactions projetées sur une ligne de temps
- **Diagramme de collaboration (diagramme de communication en UML2)** : interactions projetées sur un diagramme d'objets



Représentation d'un scénario d'interaction via un diagramme de collaboration

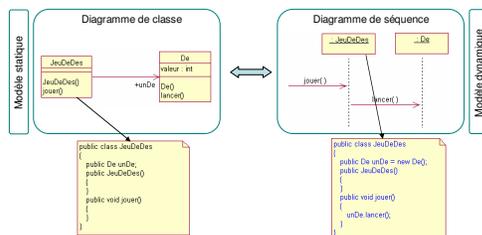
Représentation d'un scénario d'interaction via un diagramme de séquence

2

Diagramme d'interaction

Diagrammes de classes et diagrammes d'interactions (1)

- Les diagrammes de classes modélise la plateforme statique du système sur laquelle les diagrammes d'interaction vont modéliser la collaborations des objets dans le cadre des divers scénarios.



3

Diagrammes d'interactions

Diagrammes de classes et diagrammes d'interactions (2)

- Les diagrammes de classes peuvent éventuellement être complétés voire modifiés lors de l'élaboration des diagrammes d'interactions (ils doivent demeurer cohérents) . Parfois, les diagrammes de classes et les diagrammes d'interaction sont réalisés simultanément.
- Un erreur assez répandue : Penser que les diagrammes de classes sont les plus importants. C'est durant la modélisation dynamique que l'on va relier, par des message, les objets qui ont été préalablement séparés durant l'étape d'analyse.
- Les diagrammes d'interactions répondent à deux questions auxquelles les diagrammes de classes ne répondent pas:
 - Les objets en besoin de quoi pour collaborer ?
 - Comment ils vont faire pour collaborer ?

4

Diagrammes d'interactions

Points forts et points faibles des diagrammes des séquences et de diagramme de collaboration (1)

Chaque diagramme à ses points forts et ses points faibles : il n'y a pas de meilleurs choix. Le choix est laissé au modélisateur lui-même selon ses propres références.

■ Pour les diagrammes de séquences

- Les diagrammes de séquences sont généralement préférés en raison de leur grande richesse notationnelle.
- Les diagrammes de séquence sont plus facile à lire les flots d'appels : on les lit simplement de haut en bas. Dans un diagramme de collaboration, il faut rechercher la numérotation des messages pour comprendre leur enchaînement.
- Les diagrammes de séquence sont très adaptés aux besoins de la documentation, notamment la documentation des Cas d'Utilisation.

■ Pour les diagrammes de collaboration

- Les diagrammes de collaboration permettent de représenter le contexte d'une interaction: DC permettent de préciser les états des objets qui interagissent.

5

Diagrammes d'interactions

Points forts et points faibles des diagrammes des séquences et de diagramme de collaboration (2)

■ Diagrammes de séquences ou diagrammes de collaboration ?

Il n'y a pas de réponse. Mais pour décrire les scénarios des cas d'utilisation on privilégiera:

- Analyse: diagrammes de séquence pour documenter les cas d'utilisation (& d'activités)
- Conception: diagrammes de collaboration (& séquence & d'activités)

6

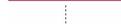
Diagramme de séquence

Corpus d'un diagramme de séquence (1)

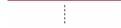
👉 Les objets

- Les objets représentés sont **uniquement** ceux participant à la collaboration
- Un objet est matérialisé par un rectangle et une barre verticale, appelé **ligne de vie** de l'objet (Object lifeline).
- 3 représentations sont possibles :

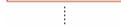
UnObjet Le nom de l'objet uniquement



·UneClasse Le nom de la classe uniquement



UnObjet ·UneClasse Le nom de l'objet et de la classe, séparés par :



7

Diagramme de séquence

Corpus d'un diagramme de séquence (2)

👉 Les messages

- Les objets communiquent en échangeant des messages.
- Un message est représenté par une flèche horizontale, orientée de l'émetteur vers le destinataire.



8

Diagramme de séquence

Corpus d'un diagramme de séquence (3)

L'utilisateur

- Un acteur est représenté afin de traduire les interactions déclenchées par un élément extérieur au système.
- déclenche le premier message de l'interaction.
- a la même représentation graphique qu'un acteur dans les Cas d'Utilisation.

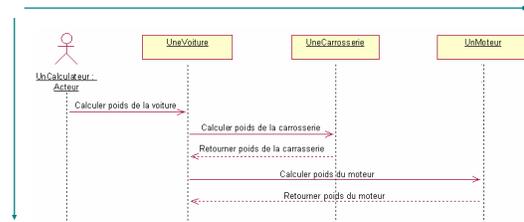


9

Diagramme de séquence

Exemple de diagramme de séquence

Modéliser le calcul du poids d'une voiture (poids moteur + poids carrosserie) par un calculateur.



Commentaire :

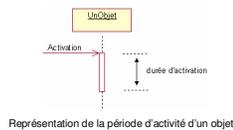
- Dimension verticale : (le temps) ; L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule "de haut en bas" de cet axe ; le numérotation des messages est optionnelle.
- Dimension horizontale : les objets (et les acteurs) ; La disposition des objets sur l'axe horizontal n'a pas de conséquence pour la sémantique du diagramme.

10

Diagramme de séquence

Activations et envois de messages (1)

- Une **période d'activité** (ou **d'activation**) (Focus of control) correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet qui lui sert de sous-traitant.
- Les périodes d'activité se représentent par des bandes rectangulaires placées sur les lignes de vie.
- Le début et la fin d'une bande correspondent respectivement au début et à la fin d'une période d'activité.



Remarque :

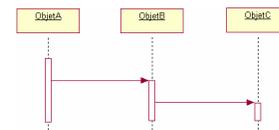
- La fin d'activation d'un objet ne correspond pas à la fin de sa vie.
- Un même objet peut être activé plusieurs fois au cours de son existence.

11

Diagramme de séquence

Activations et envois de messages (2)

- La période d'activité d'un objet inclut la durée d'attente d'une réponse d'un autre objet.
- **Exemple :**
Lors d'un appel d'une procédure de ObjetB de la part de ObjetA, celui-ci est bloqué jusqu'à ce que ObjetB lui réponde. ObjetB, à son tour, fait appel d'une procédure de ObjetC et il est bloqué à son tour jusqu'à ce que ObjetC lui réponde.



La durée d'activation d'un objet inclut la durée d'attente lorsqu'il envoie un message d'appel au procédé.

12

Diagramme de séquence

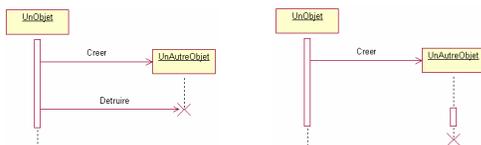
Création et destruction d'objet (1)

Création:

La création d'un objet est représentée en faisant pointer le message de création sur le rectangle qui symbolise l'objet créé.

Destruction:

- La destruction d'un objet est représentée par un **X** à la fin de sa ligne de vie.
- Si l'objet est détruit par un autre objet (non par lui-même), le message issu de l'objet destructeur pointe sur le **X** ou juste à la suite de l'extrémité du message.
- Il est possible d'un objet se détruit lui-même.



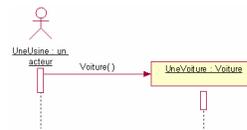
13

Diagramme de séquence

Création et destruction d'objet (2)

Création d'un objet en Java :

le message de création est représenté par le constructeur de la classe de l'objet créé.

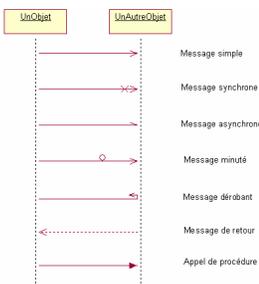


14

Diagramme de séquence

La communication (1)

- 7 types de messages (rappel) :



15

Diagramme de séquence

La communication (2)

- 7 types de messages (rappel) :

UML propose un certain nombre de stéréotypes graphiques pour décrire la nature du message (ces stéréotypes graphiques s'appliquent également aux messages des diagrammes de collaborations) :

- message simple** : Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.
- message synchrone** : Bloque l'expéditeur jusqu'à la prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) 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).
- message minuté (timeout)** : Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), 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 débordant** : N'interrompt pas l'exécution de l'expéditeur. Déclenche immédiatement une opération chez le récepteur qui est supposé être préalablement en attente de ce message.
- message d'appel au procédure** (flot de contrôle emboîté), la séquence emboîtée doit se terminer pour que la séquence englobante reprenne le contrôle.
- message de retour** : C'est le message attendu par l'objet appelant. Dans le cas d'un appel au procédure, il est implicite.

16

Diagramme de séquence

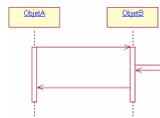
La communication (3)

■ Simplification des messages (UML 1.4+) :

On ne retient que 3 grandes catégories d'envois de messages : les messages simples, les Appels de procédures et les retour explicites d' appels de procédures (les autres en sont des cas particuliers)

□ Les messages simples :

- (ou flot de contrôle à plat) Catégorie de messages utilisée pour indiquer la **progression** vers une prochaine étape d'une séquence.
- Tous les messages de cette catégorie sont **asynchrones**.
- Messages représentés par une flèche simple.



17

Modélisation Orienté Objet

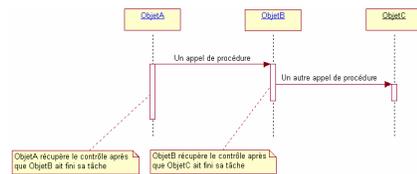
H. El mansouri

Diagramme de séquence

La communication (4)

□ Les appels de procédures :

- Dans un appel de procédure (ou flot de contrôle emboîté), l'expéditeur attend un retour de la part du destinataire.
- Lorsqu'il y a des appels de procédure en cascade, chaque traitement emboîté doit se terminer pour que le traitement englobant reprenne le contrôle.
- Les appels de procédure sont représentés par des flèches à pointe triangulaire pleine.



18

Modélisation Orienté Objet

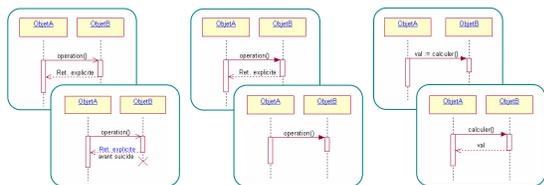
H. El mansouri

Diagramme de séquence

La communication (5)

□ Les messages de retour

- Le retour sur appel procédure est implicite à la fin d'une activation
- En cas d'envois message asynchrones et parallèles, il est utile d'expliquer la fin de l'exécution de sous-procédures et le retour éventuel de paramètres.
- On utilise une flèche pointillée (décoré éventuellement par les valeurs retournées).



Ret. explicite dans le cas des envois asynchrones

Ret. explicite et implicite dans le cas des appels de procédures

Ex. de Ret. Implicite et explicite dans le cas des appels de procédures avec valeur de retour

19

Modélisation Orienté Objet

H. El mansouri

Diagramme de séquence

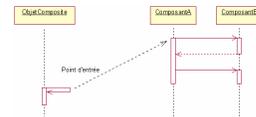
La communication (6)

■ Cas particulier : les messages réflexifs

- Lorsque l'objet s'envoie un message.
- Représentation graphique :



- Un message réflexif peut ne pas représenter un envoi d'un message, mais une activité interne à l'objet dans le cas d'un objet composite. Dans ce cas, il représente les interactions entre objets composant ce dernier, et peut être décrit au moyen d'un autre diagramme de séquence.



20

Modélisation Orienté Objet

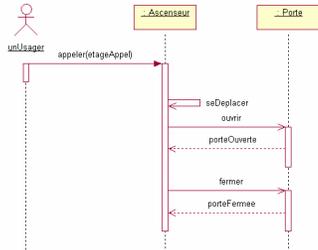
H. El mansouri

Diagramme de séquence

La communication (7)

■ Exemple

Modéliser le déplacement de l'appel d'un ascenseur par un usager



21

Diagramme de séquence

La communication (8)

■ Étiquette des messages

- Une étiquette décrit les messages auxquels elle est attachée.
- Syntaxe générale :

`[garde] *[itération]numero_séquence: variable_retour := nom_opération(liste_arguments)`

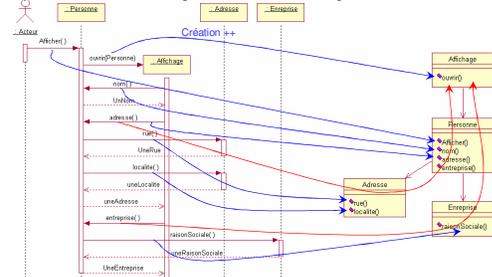
- **nom_opération** : le nom de l'opération à invoquer.
- **liste_arguments** : facultative, liste des arguments transmis à l'opération invoquée. Les éléments sont séparés par des virgules.
- **variable_retour** : facultative, valeur retournée par l'opération invoquée.
- **garde** : facultative, la condition préalable à l'envoi du message.
- **itération** : facultative, le nombre d'envois du message ou conditions d'itération (* et les [] sont omis si pas d'itération).
- **Numéro_séquence** : facultatif, l'ordre du message dans la séquence (les : sont omis si Numéro_séquence n'est pas spécifié).

22

Diagramme de séquence

Exemple :

- Un système de lecture et d'affichage d'informations dont le diagramme de classe est le suivant :



- Remarques :**
- Les instances ne peuvent communiquer que s'elles se connaissent
 - Une instance peut communiquer indirectement avec une autre (ex. affichage/entreprise)

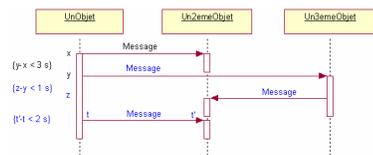
23

Diagramme de séquence

Contraintes temporelles (1)

- Pour modéliser les délais de transmission, on utilise l'une des deux notations suivantes :
 - une flèche oblique (lorsque l'outil de modélisation le permet),
 - ou des notations temporelles dans la marge.
- Les instants d'émission et de réception d'un message peuvent être représentés par le couple (nom, nom prime).

■ Exemple 1 :



Exemple de contraintes temporelles construites à partir des noms de transitions

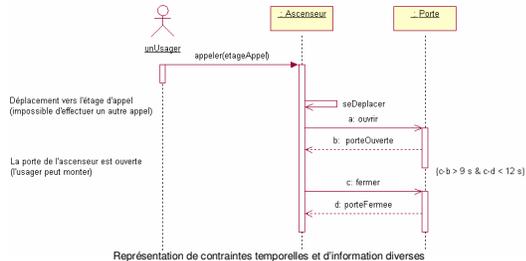
24

Diagramme de séquence

Contraintes temporelles (2)

■ Exemple 2 :

Compléter le modèle de l'ascenseur par des contraintes temporelles



Représentation de contraintes temporelles et d'information diverses
Remarque : Un diagramme de séquence peut être décoré par des indications textuelles de toute sorte ou en pseudo-code.

Diagramme de séquence

Structures de contrôle (1)

■ Les boucles

■ Dans les diagrammes de séquences, il est possible de modéliser les envois de messages répétés (structures de contrôles itératives).

■ Deux techniques de représentation :

□ Par pseudo-code (while X loop end loop) (X est la condition d'itération) ou par un langage quelconque comme java, etc. (UML ne précise aucun format pour les conditions).

□ Par garde (*[X]) placé devant le message (* représente l'itération).

■ **Exemple :** ObjetA envoie un message à ObjetB tant que la condition X est vérifiée.

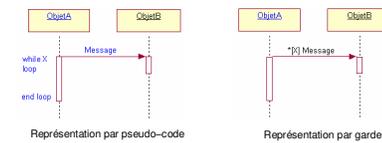


Diagramme de séquence

Structures de contrôle (2)

■ Les boucles

■ Exemples de conditions d'itérations :

*[i:=1..10] message	Le message est envoyé 10 fois
*[x<10] message	Le message est envoyé de façon répétée jusqu'à ce que x soit plus grand ou égal à 10.
*[item not found] message	Le message est envoyé de façon répétée jusqu'à ce que l'item soit trouvé.
*[pour chaque employé] message	Le message est envoyé jusqu'à ce que tous les employés soient traités.
* message	La condition d'itération est omise (non spécifiée).

Diagramme de séquence

Structures de contrôle (3)

■ Les messages conditionnels

■ Dans les diagrammes de séquences, il est possible de modéliser les envois de messages conditionnels (structures de contrôles conditionnelles).

■ Deux techniques de représentation :

□ Par pseudo-code (if X else end if).

□ Par garde ([X]) placée devant le message.

■ **Exemple 1/2 :** Envoie conditionnel de messages impliquant deux destinataires.

■ **Exemple 3 :** Envoie conditionnel de messages impliquant un seul destinataire.

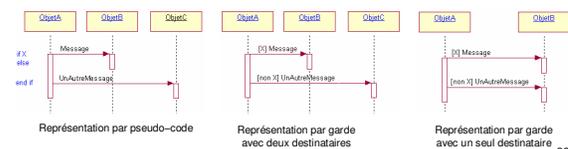
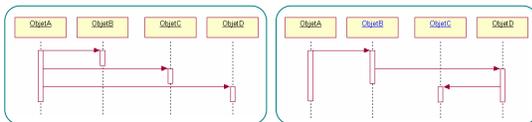


Diagramme de séquence

La notion de contrôle

Deux Types de contrôle :

- Le contrôle centralisé
- Le contrôle décentralisé



Un seul objet détient le contrôle de la séquence : c'est le chef d'orchestre

Le contrôle de la séquence est dispatché entre les objets participant à la coopération : le contrôle se transmet par délégation

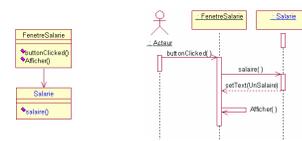
29

Diagramme de séquence

La granularité (1)

- Le niveau de granularité reflète le **niveau de détail** que l'on veut donner au diagramme de séquence.
- Il faut conserver le même niveau de granularité pour les différents messages au sein de même diagramme.
- On peut choisir des niveaux de granularité différents pour deux diagrammes.
- Le niveau de granularité doit être cohérent par rapport au diagramme de classe.

Exemple 1 : Niveau de granularité grossier

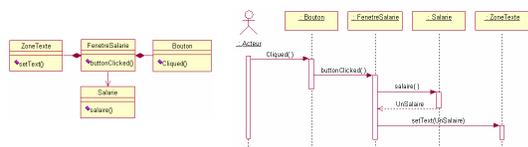


30

Diagramme de séquence

La granularité (2)

Exemple 2 : Niveau de granularité fin



31

Diagramme de séquence

Cas des objets à rôles multiples (1)

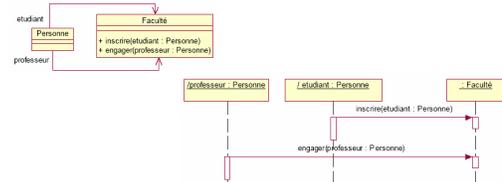
- Les rôles modélisés dans un diagramme de séquence sont des **rôles joués dans le contexte d'une collaboration**.

Syntaxe :

nom_objet '/' nom_rôle ':' nom_classe

- Un objet qui joue plusieurs rôles dans une collaboration doit figurer autant de fois que de rôles joués.

Exemple :



32

Diagramme de séquence

Cas des objets à rôles multiples (2)

- Les rôles modélisés dans un diagramme de séquence peuvent ne correspondre pas aux rôles modélisés dans le diagramme de classe, mais y demeurent liés.
- **Exemple** : Un diagramme de séquence représentant une collaboration dans un contexte de location d'un appartement par un locuteur pour un loyer et un coût donnés.

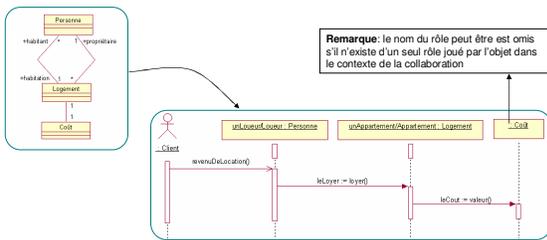


Diagramme de séquence

Autre usage du diagramme de séquence : Documenter les Cas d'Utilisation

- Le diagramme de séquence se concentre sur la description des interactions en termes proches de l'utilisateur et sans entrer dans les détails de la synchronisation : pas représentation interne du système
- C'est un complément de la description textuelle des Cas d'Utilisation. Le Diagramme de séquence ne peut pas représenter certains renseignements, notamment les pré- et les post-conditions, certaines contraintes, etc.
- Le formalisme UML est plus libre.
- Les flèches ne traduisent pas des messages au sens de la programmation mais des événements.

Diagramme de séquence

Exercice 1 :

Établir un diagramme de séquence pour un système de calcul du poids d'une voiture à partir de ses composants. Le diagramme de classe est le suivant :

Diagramme de séquence

Exercice 2 :

- Établissez le Diagramme de Séquence de l'opération d'impression d'un document. Cette opération se déroule de la façon suivante :
 - On sélectionne un document dans le gestionnaire.
 - On demande au gestionnaire d'imprimer le document sélectionné.
 - Le gestionnaire demande alors au document lui-même de s'imprimer.
 - Le document demande alors au gestionnaire de lui retourner l'imprimante courante.
 - Le document communique avec cette imprimante pour lui demander de l'imprimer.
 - Enfin, l'imprimante lui demande successivement son titre, le corps du texte et le pied de page.
- Le diagramme de classe se présente de la façon suivante.



Diagramme de séquence

Exercice 2 (correction) :

37

Diagramme de collaboration

Exercice 3 :

- On vous demande de représenter le diagramme de classe Compatible avec le diagramme de séquence suivant :

38

Diagramme de collaboration

Exercice 3 (corrigé):

39

Diagramme de collaboration

Introduction

- 👉 Un Diagramme de Collaboration représente la même collaboration que le Diagramme de séquence,
- 👉 ne prend pas en compte la dimension temporelle de la collaboration, toutefois la séquentialité des messages est représenté pas la numérotation des messages.
- 👉 met évidence la dimension structurelle (l'organisation) de la collaboration.
- 👉 C'est une extension du diagramme d'objets.

40

