

chapitre 5

Les diagrammes de séquence

Année universitaire 2023-2024

Les aspects dynamiques

Les diagrammes de séquence

Les diagrammes de collaboration

Les diagrammes d'états-transitions

Les diagrammes d'activité

Les diagrammes de séquence

Les diagrammes de séquence permettent de représenter les *interactions* entre objets d'un point de vue chronologique.

Ils se concentrent sur la séquence des interactions d'un point de vue temporel.

Ils font partie des diagrammes dits d'interaction, avec les diagrammes de collaboration.

Les diagrammes de séquence

Les interactions se traduisent par des envois de messages entre les objets.

Les diagrammes de séquences peuvent s'appliquer

- à la documentation des cas d'utilisation, en représentant les interactions entre les acteurs et le système. Les étiquettes des messages sont des événements qui se produisent dans le système.
- à la représentation des interactions informatiques et la répartition des flots de contrôle. Les messages échangés unifient les différentes formes de communication entre les objets (appels de procédure, émission de signaux, etc.).

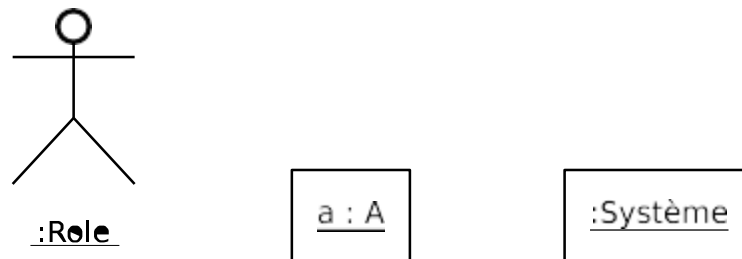
Les objets des diagrammes de séquence

Ce sont des entités appartenant au système (instance d'une classe), ou se trouvant à ses limites (les acteurs), ou le système lui-même.

Ils représentent :

- Soit des concepts abstraits, ou des acteurs (documentation des cas d'utilisations)
- Soit des objets d'implantation (interactions informatiques)

Les objets sont identifiés par l'intermédiaire des diagrammes de cas d'utilisation ou des diagrammes de classes. Ils sont dénotés classiquement.

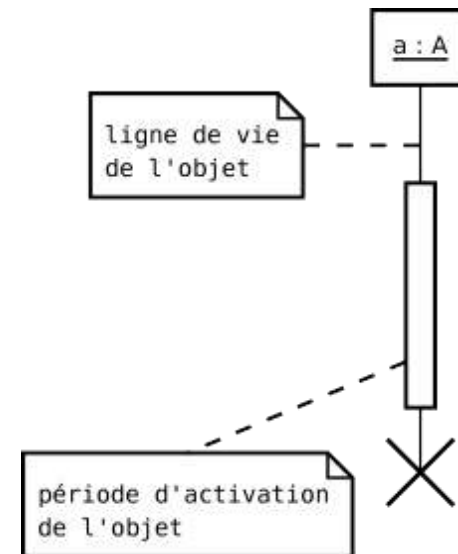


La ligne de vie des objets

Elle représente la période de temps durant laquelle l'objet existe.

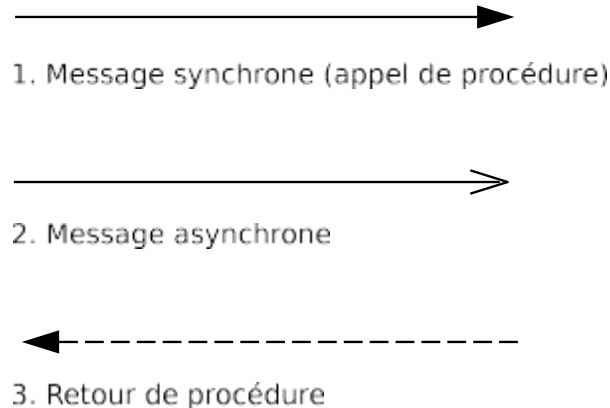
La période d'activation correspond à un intervalle de temps durant lequel l'objet effectue une actions, soit directement, soit par l'intermédiaire d'un autre objet qui lui sert de sous-traitant (appel de procédure).

La vie de l'objet commence par l'objet lui-même et se termine à la fin de la ligne de vie, éventuellement marquée par une croix indiquant la destruction de l'objet.



Les types de messages

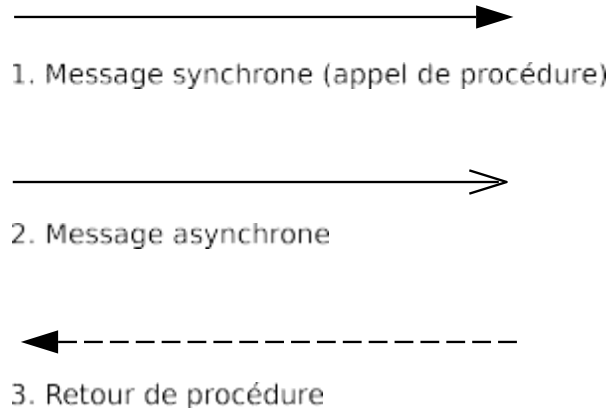
On distingue 3 types de messages.



1. Message synchrone : correspond à un appel de procédure ou flot de contrôle imbriqué. La séquence imbriquée est entièrement faite avant que l'appelant ne continue : l'appelant est bloqué jusqu'à ce que l'appelé termine.

Les types de messages

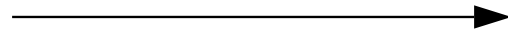
On distingue 3 types de messages.



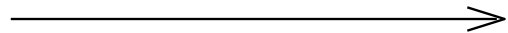
2. Message asynchrone : correspond à un message "à plat", non imbriqué. L'appelant n'est pas bloqué jusqu'à ce que l'appelé termine.

Les types de messages

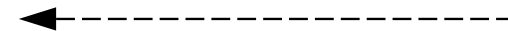
On distingue 3 types de messages.



1. Message synchrone (appel de procédure)



2. Message asynchrone



3. Retour de procédure

3. Retour de procédure : correspond à la terminaison d'une procédure avec le retour éventuel d'une valeur. Ces messages peuvent ne pas être dessinés, ils sont alors implicites.

Un premier exemple

Exemple d'un système de communication téléphonique

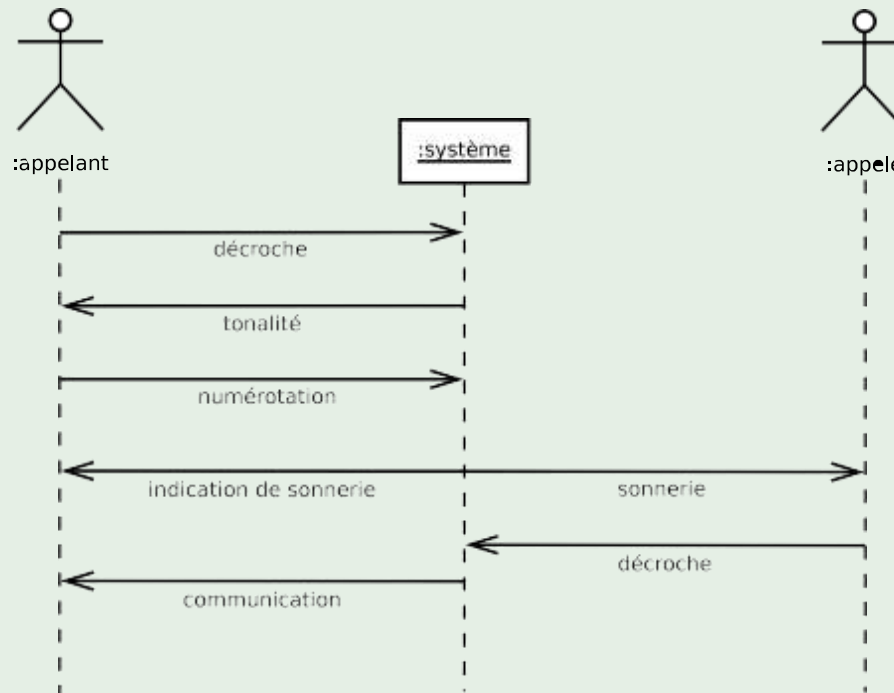
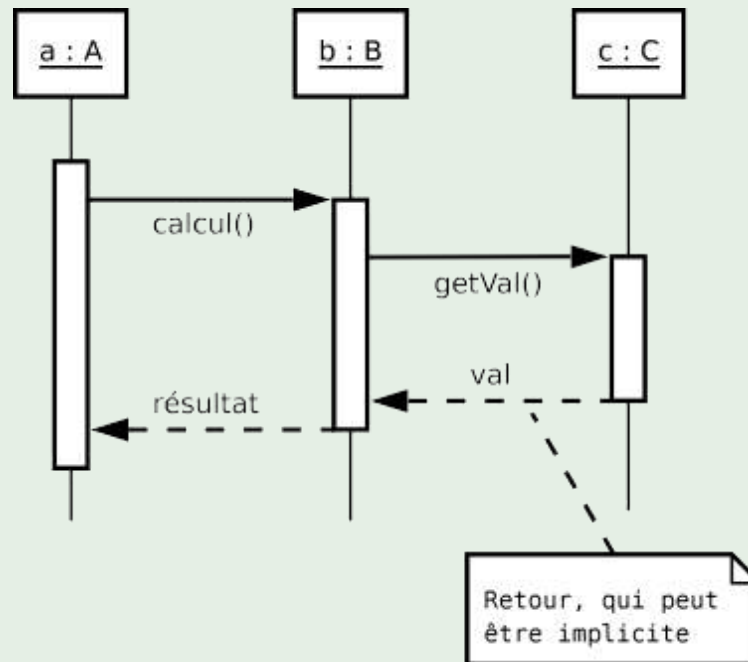


Illustration d'un diagramme de cas d'utilisation, avec messages asynchrones.

Un second exemple

Contrôle de flot d'exécution

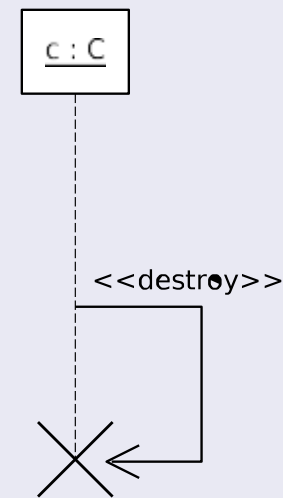
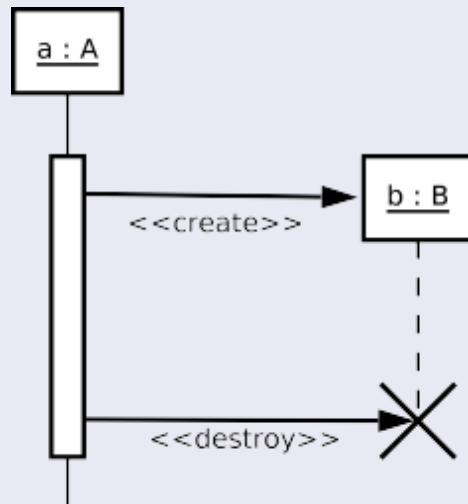


Création et destruction d'un objet

Créer et détruire des objets

Deux types de messages particuliers permettent de créer et de détruire des objets.

- La création se symbolise par un message `<< create >>`
- La destruction se symbolise par un message `<< destroy >>`



Etiquettes des messages

Syntaxe générale

La syntaxe des étiquettes de message est la suivante :

$$[Garde] \quad Iteration \quad resultat := \text{nom_message}(arguments)$$

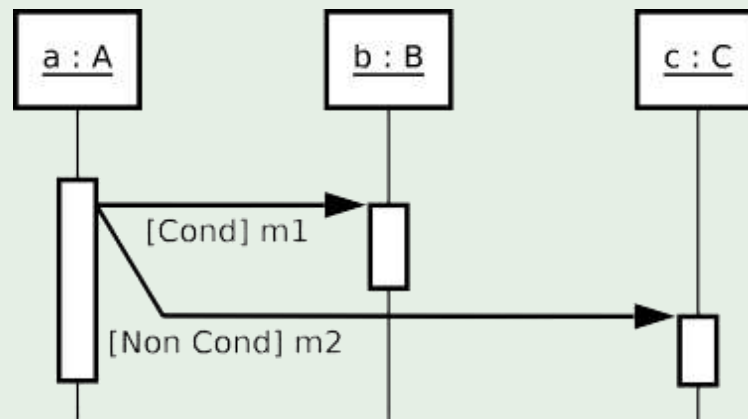
où

- *[Garde]* est la condition d'envoi du message (facultatif)
- *Iteration* décrit une éventuelle itération sur l'envoi du message (facultatif)
- *resultat :=* est l'affectation d'une valeur de retour (facultatif)
- *nom_message* est le nom du message
- *arguments* sont les arguments du message (facultatif)

Etiquettes : messages gardés

Les gardes permettent de spécifier des conditions pour l'envoi des messages. Les messages ne sont envoyés que si la condition est satisfaite.

Messages conditionnels

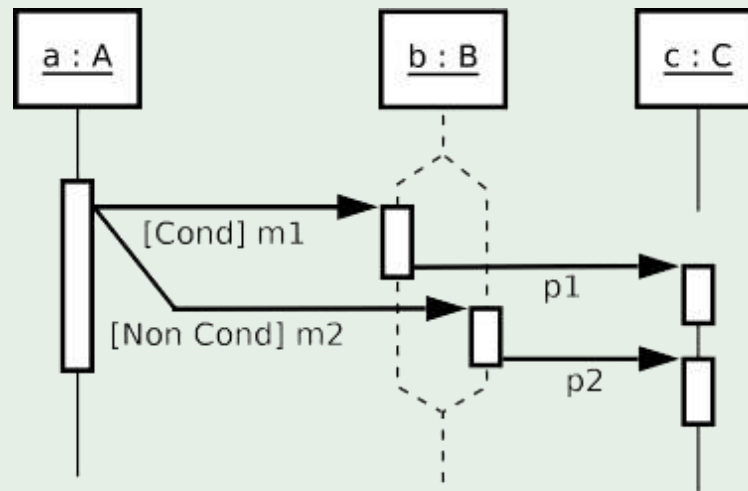


Etiquettes : messages gardés

Les gardes permettent de spécifier des conditions pour l'envoi des messages. Les messages ne sont envoyés que si la condition est satisfaite.

La ligne de vie d'un objet peut être dédoublée pour indiquer des actions qui sont effectuées suite à un message conditionnel (UML 1.5).

Dédoublage de la ligne de vie

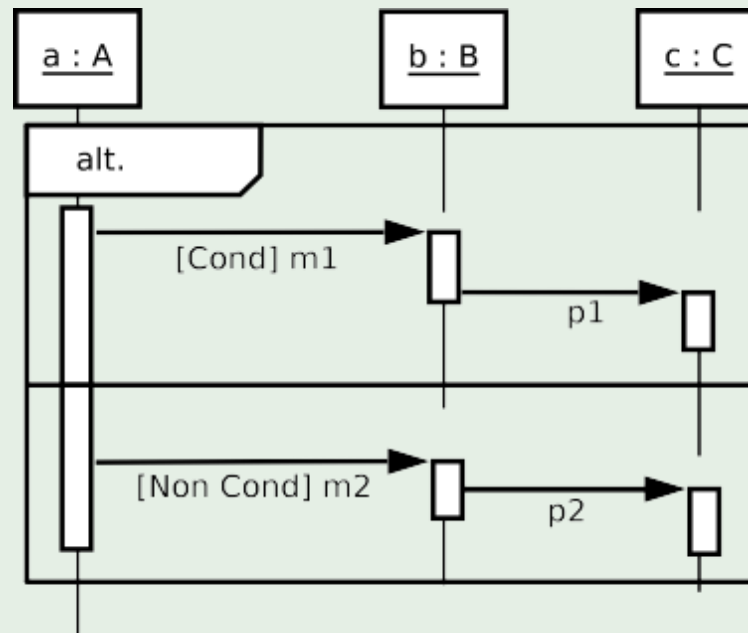


Etiquettes : messages gardés

Les gardes permettent de spécifier des conditions pour l'envoi des messages. Les messages ne sont envoyés que si la condition est satisfaite.

UML 2.0 ajoute la possibilité d'utiliser des fragments, plus lisibles.

Utilisation d'un fragment d'alternative



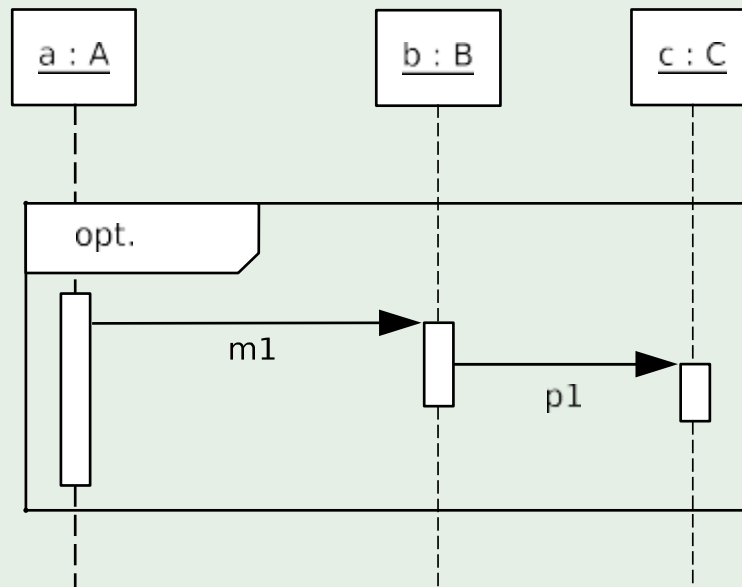
Etiquettes : messages gardés

Les gardes permettent de spécifier des conditions pour l'envoi des messages. Les messages ne sont envoyés que si la condition est satisfaite.

UML 2.0 ajoute la possibilité d'utiliser des fragments, plus lisibles.

Notamment pour les actions optionnelles.

Utilisation d'un fragment optionnel



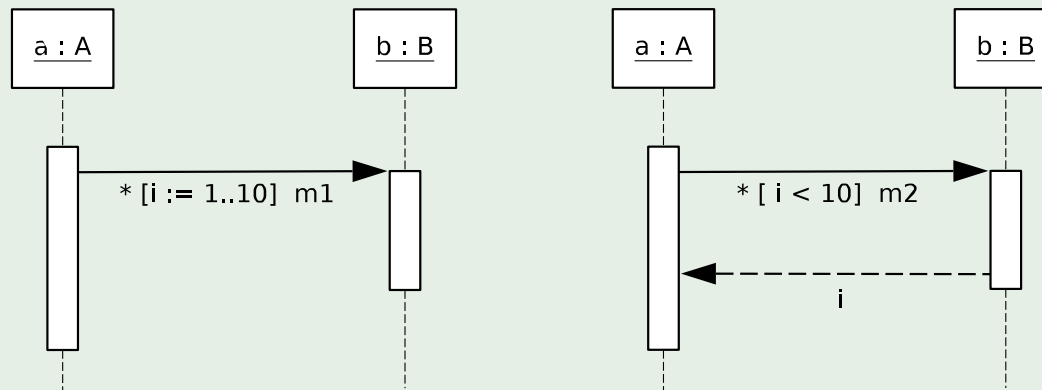
Etiquettes : itérations

Deux types d'itérations

On distingue deux types d'itérations :

- Itération *séquentielle* : envoi séquentiel de plusieurs instances du même message, dénoté par $*[clause\ d'itération]$
- Itération *parallèle* : envoi en parallèle de plusieurs instances du même message, dénoté par $* \parallel [clause\ d'itération]$

Envoi séquentiel



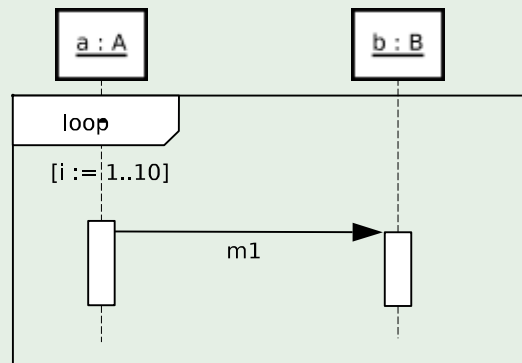
Etiquettes : itérations

Deux types d'itérations

On distingue deux types d'itérations :

- Itération *séquentielle* : envoi séquentiel de plusieurs instances du même message, dénoté par $*[clause\ d'itération]$
- Itération *parallèle* : envoi en parallèle de plusieurs instances du même message, dénoté par $* \parallel [clause\ d'itération]$

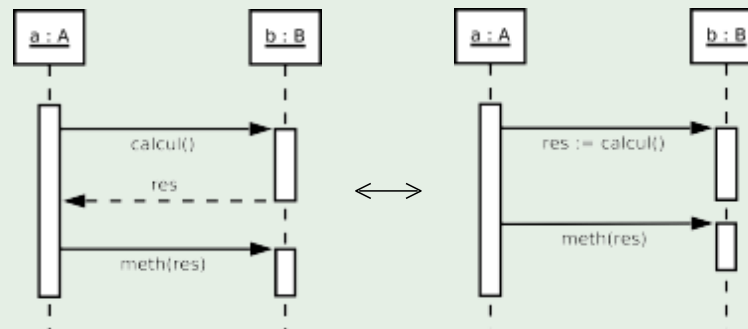
Envoi séquentiel, avec un fragment



Etiquettes : valeur de retour

Le résultat est constitué d'une liste de valeurs retournées par le message. Ces valeurs peuvent être utilisées comme paramètres d'autres messages.

Deux notations possibles



Contraintes temporelles

On peut nommer l'instant d'émission d'un message, ainsi que l'instant de sa réception. Cela permet de poser des contraintes de temps sur l'envoi et sur la réception de messages.

Par convention, si l'instant d'émission est dénoté par x , l'instant de réception de ce même message est dénoté par x' .

Utilisation de contraintes temporelles

