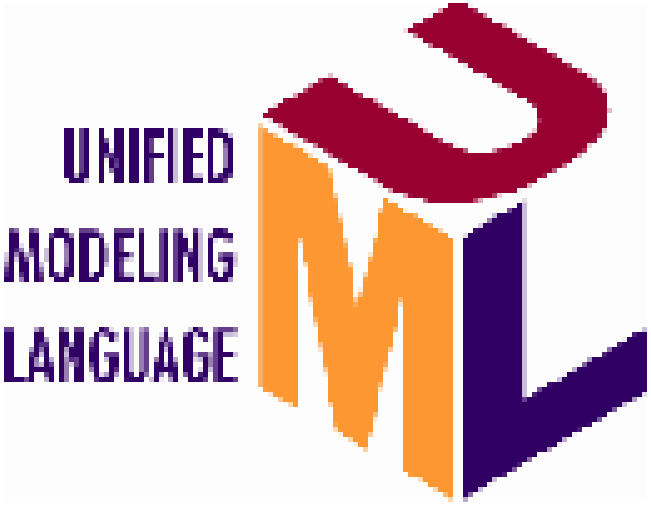


UML – Langage unifié pour la modélisation objet

Les diagrammes d’objets

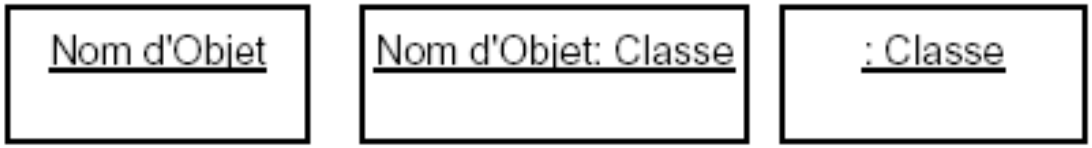
Les diagrammes d'objets

Les diagrammes d'objets, ou diagrammes d'instances, montrent des objets et des liens. Comme les diagrammes de classes, les diagrammes d'objets représentent la structure statique. La notation retenue pour les diagrammes d'objets est dérivée de celle des diagrammes de classes ; les éléments qui sont des instances sont soulignés.

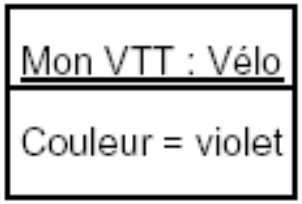
- Les diagrammes d'objets s'utilisent principalement pour montrer un contexte, par exemple avant ou après une interaction, mais également pour faciliter la compréhension des structures de données complexes, comme les structures récursives.

Représentation des objets

Chaque objet est une instance de classe. Les objets sont représentés par un rectangle dont le nom et la classe sont soulignés. Quand le nom de l'objet est omis l'objet est dit **anonyme**.



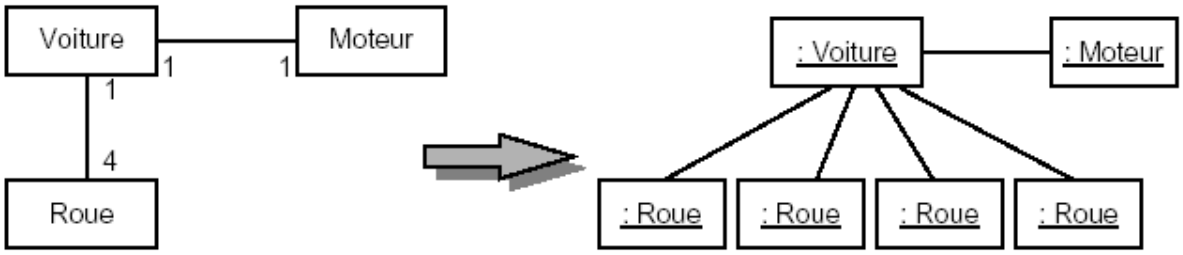
On peut rajouter un deuxième compartiment à l'objet pour indiquer les valeurs des attributs.



Représentation des liens

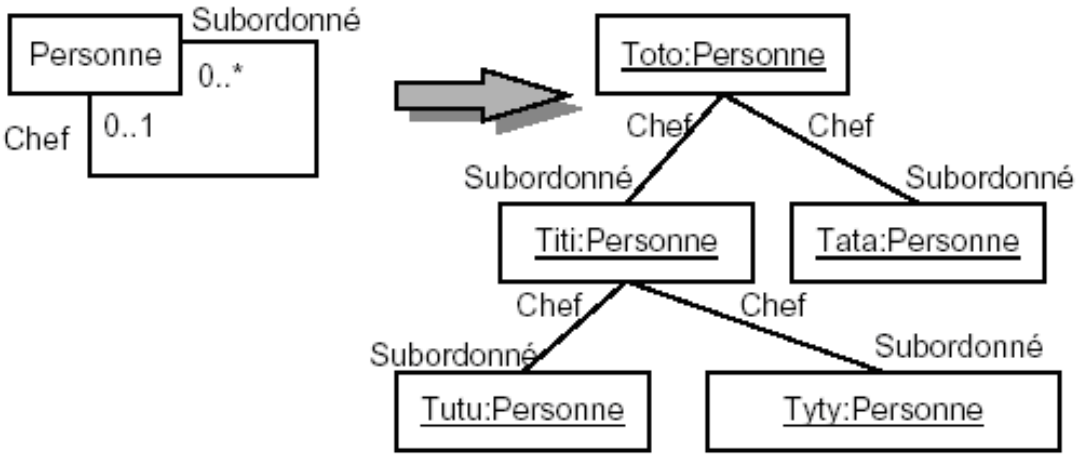
Les objets sont reliés par des liens qui sont instances des relations entre les classes des objets considérés. La représentation concrète d'une structure par des objets est souvent plus parlante que celle abstraite par des classes, surtout dans le cas de structures récursives.

Le diagramme d'objets suivant montre une partie de la structure générale des voitures. Chaque voiture possède un moteur et quatre roues (roue de secours exclue).



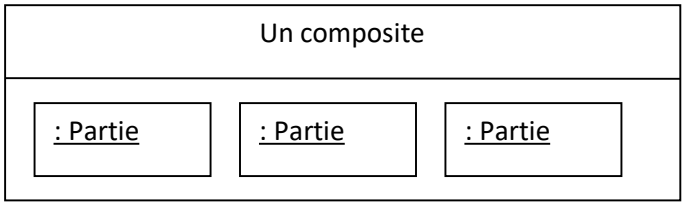
Relations réflexives

Les liens instances des associations réflexives peuvent relier un objet à lui-même. Dans ce cas, le lien est représenté par une boucle attachée à un seul objet. Le diagramme suivant représente deux liens, instances de la même association réflexive.

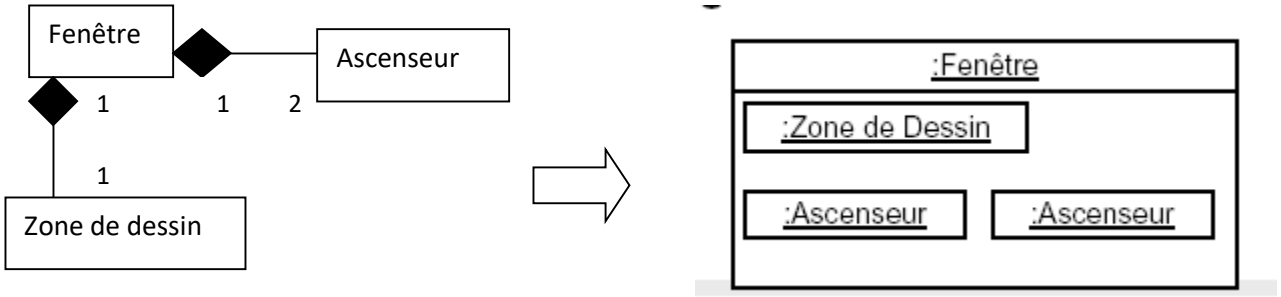


Les objets composites

Les objets composés de sous-objets peuvent être représentés au moyen d'un objet composite, afin de réduire la complexité des diagrammes. Le diagramme suivant reprend la forme graphique des objets composites.



Une composition se fait par valeur c'est-à-dire copie du ou des contextes des enfants dans celui des parents.

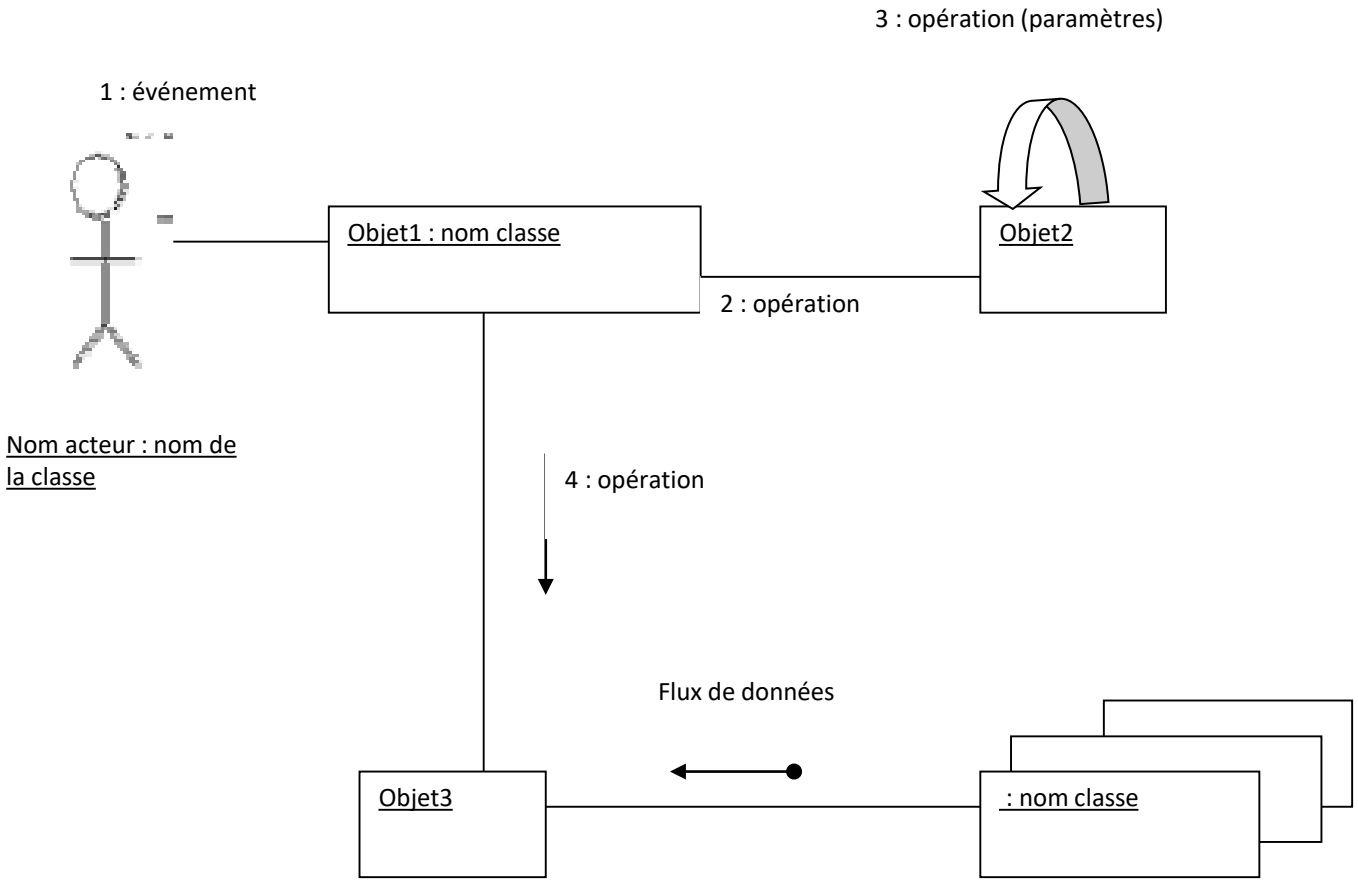


Les diagrammes de collaboration

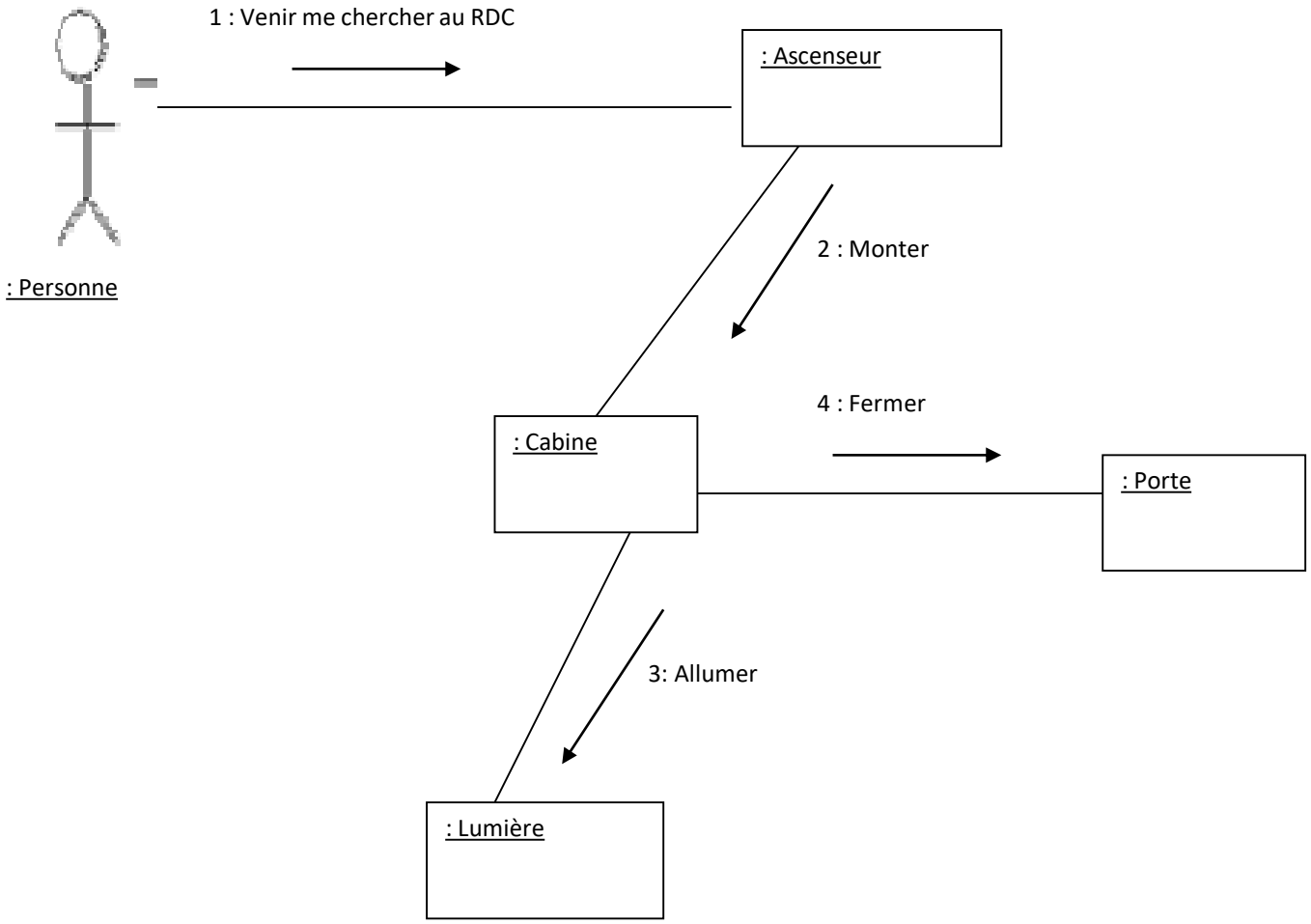
Les diagrammes de collaboration

- Les diagrammes de collaboration montrent des interactions entre objets, en insistant plus particulièrement sur la structure spatiale statique qui permet la mise en collaboration d'un groupe d'objets. Les diagrammes de collaboration expriment à la fois le contexte d'un groupe d'objets (au travers des objets et des liens) et l'interaction entre ces objets (par la représentation des envois de messages). Les diagrammes de collaboration sont une extension des diagrammes d'objets.
- Le contexte d'une interaction comprend les arguments, les variables locales créées pendant l'exécution, ainsi que les liens entre les objets qui participent à l'interaction.
- Une interaction est réalisée par un groupe d'objets qui collaborent en échangeant des messages. Ces messages sont représentés le long des liens qui relient les objets, au moyen de flèches orientées vers le destinataire du message. L'ordre des messages peut être indiqué en mettant une numérotation devant chaque message.

Les diagrammes de collaboration

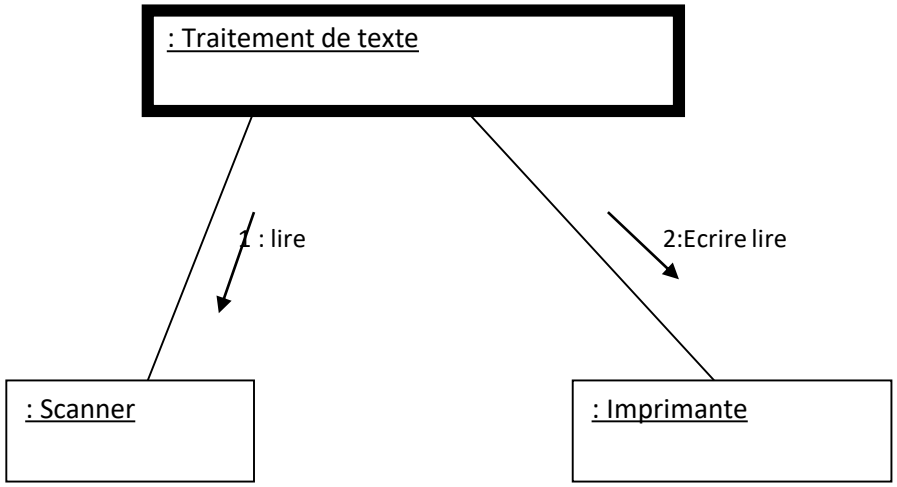


Exemple :


1 : Venir me chercher au RDC
: Ascenseur
2 : Monter
: Personne
4 : Fermer
: Cabine
: Porte
3: Allumer
: Lumière

Les diagrammes de collaboration

On peut indiquer dans le schéma quel est l'objet actif (qui possède un flot de contrôle) en donnant au symbole de l'objet concerné une bordure plus épaisse. Un objet actif peut activer un objet passif, en lui envoyant un message. Une fois le message traité, le flot de contrôle est restitué à l'objet actif. Dans un environnement multitâche, plusieurs objets peuvent être actifs simultanément.

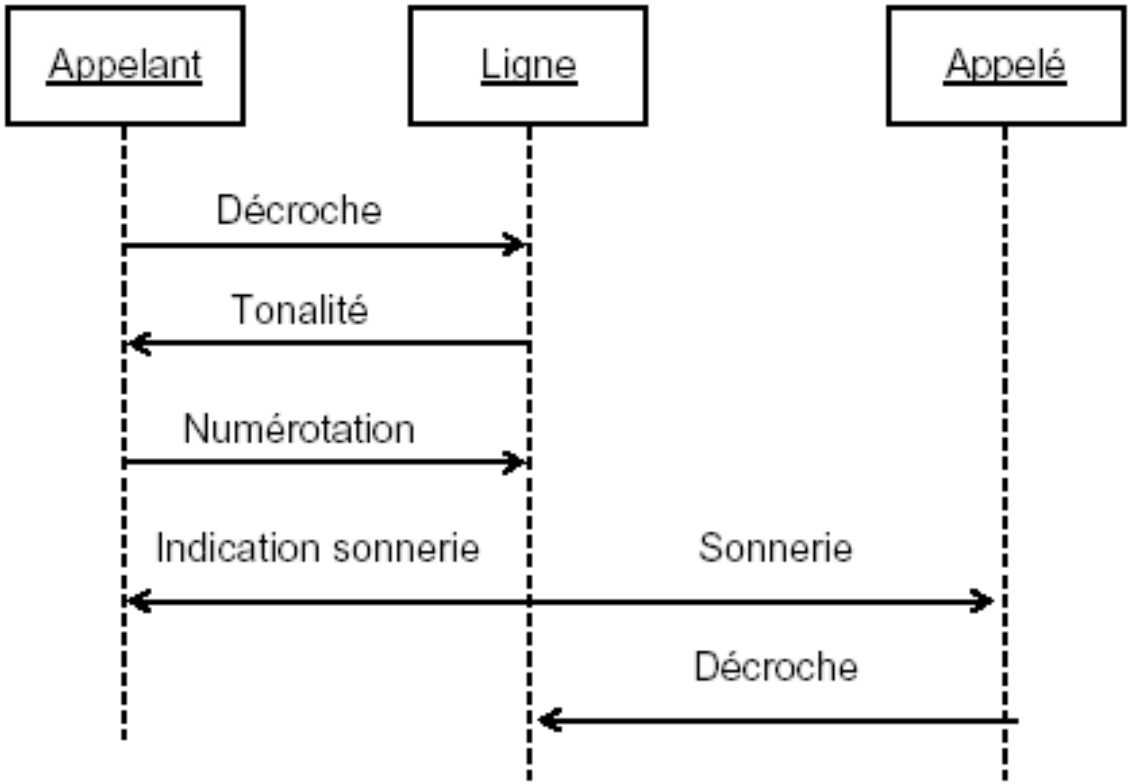


Les diagrammes de séquences

Les diagrammes de séquences

- Les diagrammes de séquences montrent des interactions entre objets selon un point de vue temporel. Le contexte des objets n'est pas représenté de manière explicite comme dans les diagrammes de collaboration. La représentation se concentre sur l'expression des interactions.
- Un diagramme de séquences représente une interaction entre objets en insistant sur la chronologie des envois de messages. Un objet est matérialisé par un rectangle et une barre verticale appelée ligne de vie des objets.
- Les objets communiquent en échangeant des messages représentés au moyen de flèches horizontales, orientées de l'émetteur du message vers le destinataire.
- L'ordre d'envoi des messages est donné par la position sur l'axe vertical. L'axe vertical peut être gradué afin d'exprimer précisément les contraintes temporelles.

Exemple de diagramme de séquence

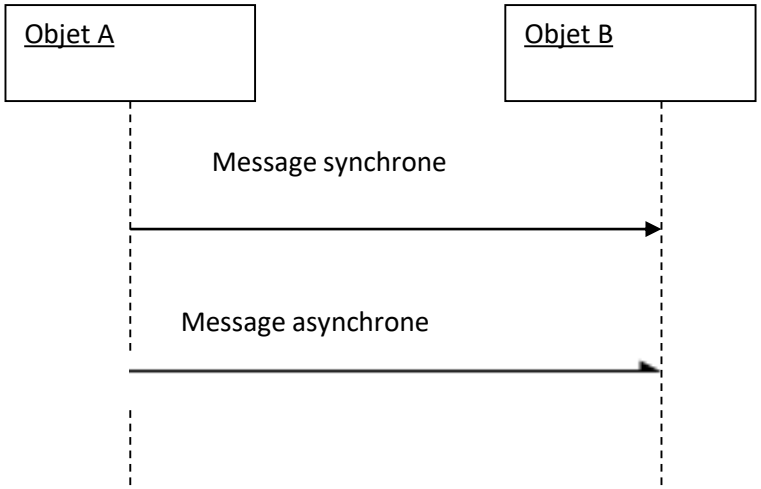


Les diagrammes de séquences

Un message reçu par un objet déclenche l'exécution d'une opération et renvoie un message qui correspond au résultat de l'opération.

Les diagrammes de séquences distinguent deux grandes catégories d'envois de messages :

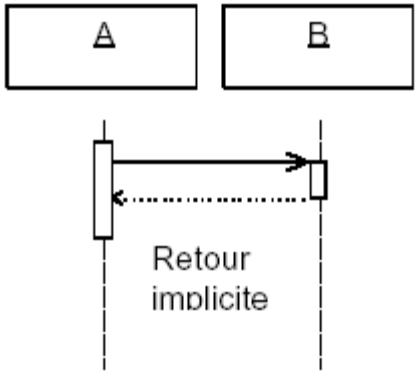
- Les messages synchrones pour lesquels l'émetteur est bloqué et attend que l'appelé ait fini de traiter le message,
- Les messages asynchrones pour lesquels l'émetteur n'est pas bloqué et peut continuer son exécution.



Les diagrammes de séquences

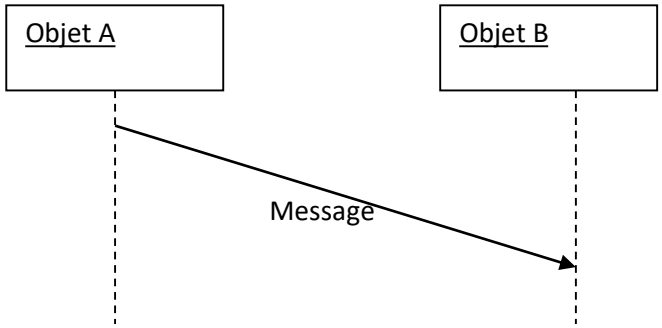
Dans le cas des envois de messages synchrones, le retour en fin d'exécution de l'opération est implicite : il n'est pas nécessaire de le représenter dans les diagrammes. L'objet **A** reprend son exécution lorsque l'action déclenchée dans l'objet **B** est terminée.

En revanche, dans le cas des envois de messages asynchrones, le retour doit être matérialisé lorsqu'il existe. Le diagramme suivant montre un objet **B** initialement activé par un objet **A**, qui retourne un message à l'objet **A** avant de cesser son exécution. Il faut noter que la fin de l'activation d'un objet ne correspond pas à la fin de sa vie. Un même objet peut être activé de nombreuses fois au cours de son existence.



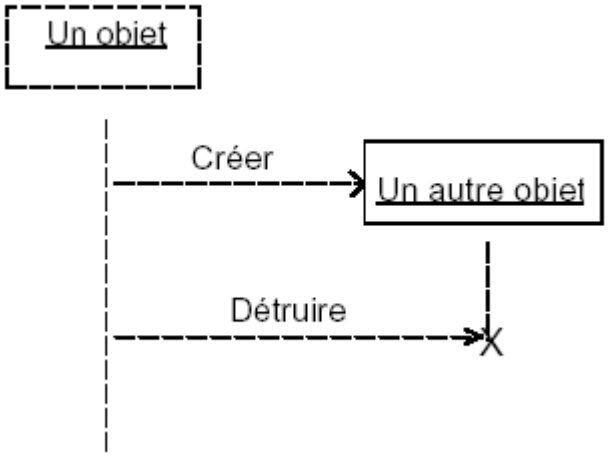
Les diagrammes de séquences

La flèche qui symbolise un message peut être représentée en oblique pour matérialiser les délais de transmission non négligeables par rapport à la dynamique générale de l'application.



Les diagrammes de séquences

La création des objets se représente en faisant pointer le message de création sur le rectangle qui symbolise l'objet créé. La destruction est indiquée par la fin de la ligne de vie et par une lettre **X**, soit à la hauteur du message qui cause la destruction, soit après le dernier message envoyé par un objet qui se suicide.



Les diagrammes de séquences

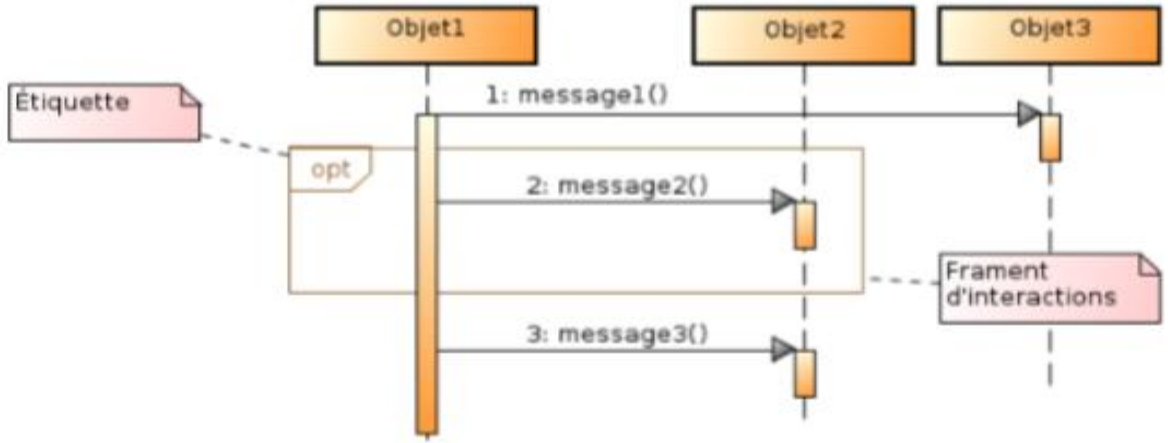
La représentation des périodes d'activité des objets est possible à l'aide de bande rectangulaire le long des lignes de vie des objets et dont les extrémités représentent le début et la fin de l'activité.

Dans le diagramme, on peut indiquer les branchements conditionnels par du pseudo code placé le long de la ligne de vie ou alors entre crochets sur le message à conditionner.

De même, on peut placer en pseudo code les boucles d'itérations (while, for).

Fragments d'interactions combinés

Un fragment d'interactions est une partie du diagramme de séquence (délimitée par un rectangle) associée à une étiquette (dans le coin supérieur gauche). L'étiquette contient un opérateur d'interaction qui permet de décrire des modalités d'exécution des messages à l'intérieur du cadre.

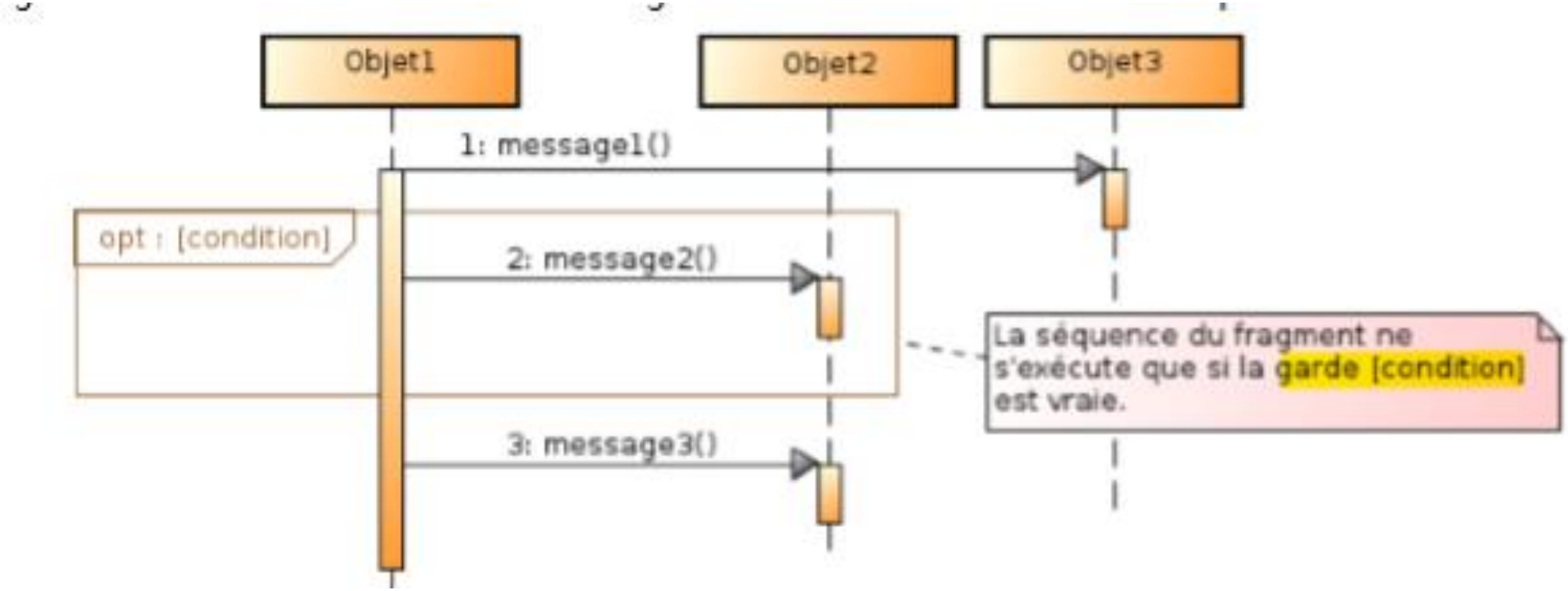


Les diagrammes de séquences

Les opérandes d'un opérateur d'interaction sont séparés par une ligne pointillée. Les conditions de choix des opérandes (éventuels) sont données par des expressions booléennes entre crochets ([ ]). Les principales modalités sont les boucles, les branchements conditionnels, les alternatives, les envois simultanés, etc.

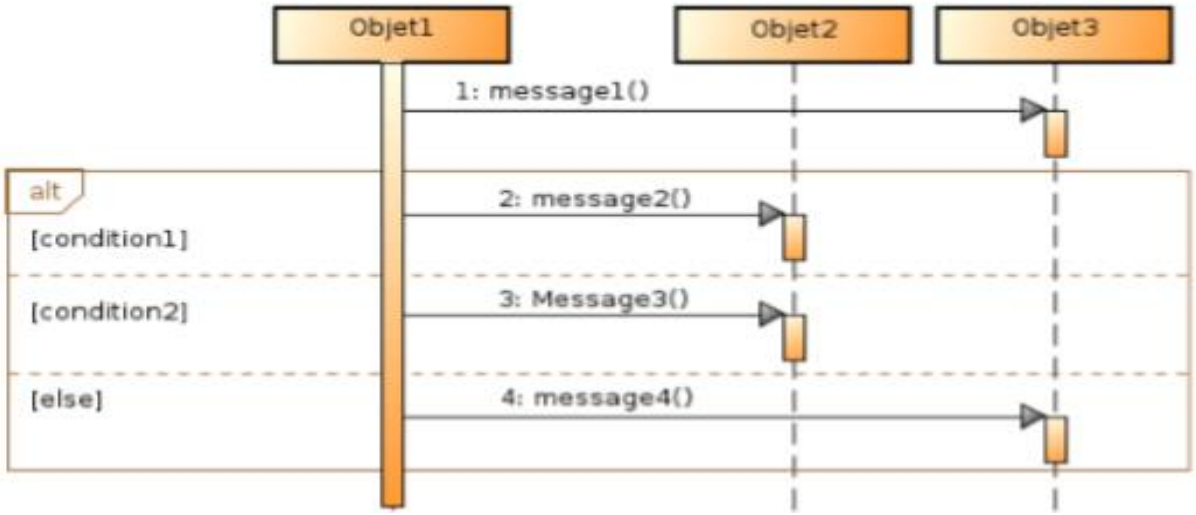
Fragment d'interaction avec opérateur « opt » :

L'opérateur option (opt) comporte un opérande et une condition de garde associée. Le sous fragment s'exécute si la condition de garde est vraie et ne s'exécute pas dans le cas contraire.



Fragment d'interaction avec opérateur « alt » :

L'opérateur alternatives (alt) est un opérateur conditionnel possédant plusieurs opérandes séparés par des pointillés. C'est l'équivalent d'une exécution à choix multiples. Chaque opérande détient une condition de garde. Seul le sous-fragment dont la condition est vraie est exécuté. La condition else est exécutée que si aucune autre condition n'est valide.



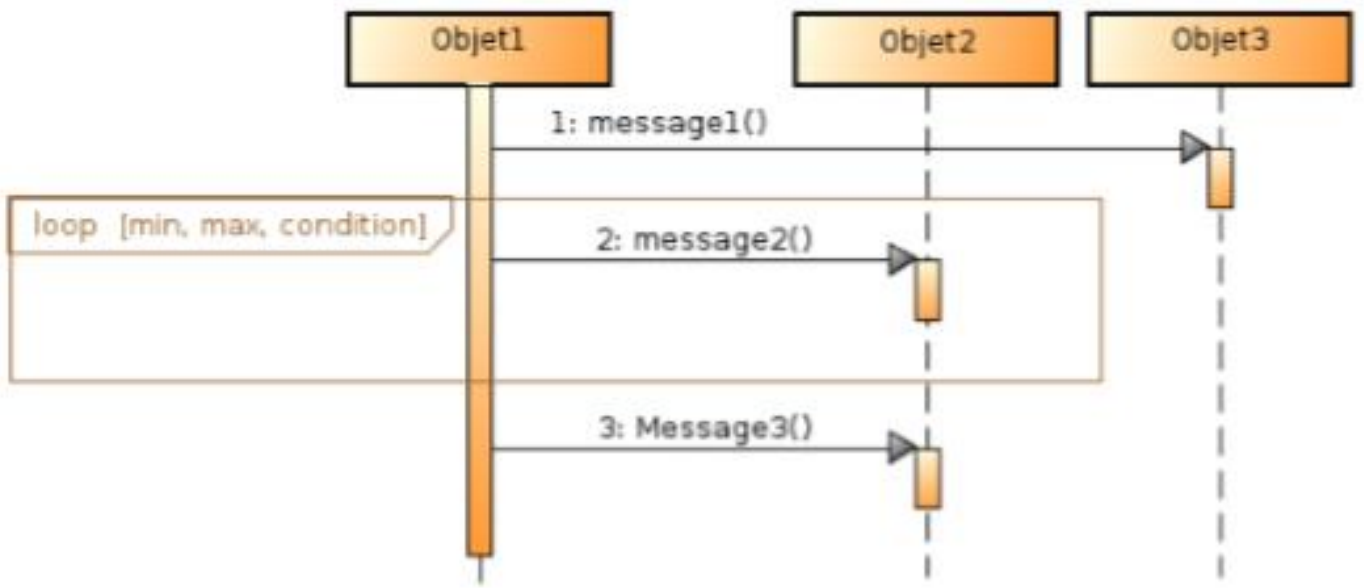
Fragment d'interaction avec opérateur «loop » :

L'opérateur de boucle (loop) exécute une itérative dont la séquence qu'elle contient est exécutée tant que la garde qui lui est associée est vraie.

La garde s'écrit de la façon suivante : loop [min, max, condition] : Chaque paramètre (min, max et condition) est optionnel. Le contenu du cadre est exécuté min fois, puis continue à s'exécuter tant que la condition et que le nombre d'exécution de la boucle ne dépasse pas max fois.

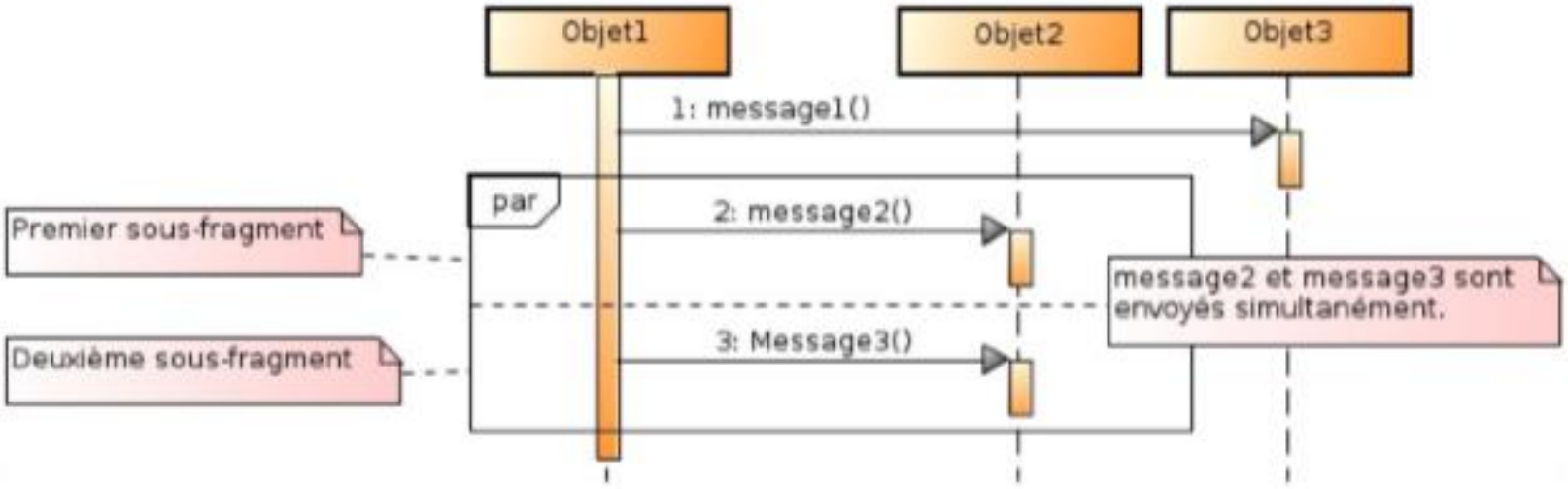
- Exemple de gardes : loop[3] : La séquence s'exécute 3 fois.
- Loop[1, 3, code=faux] La séquence s'exécute 1 fois puis un maximum de 2 autres fois si code=faux.

Les diagrammes de séquences



Fragment d'interaction avec 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).



Autres fragments d'interactions:

Nous venons de voir les 4 fragments d'interactions les plus utilisés (opt, alt, loop et par). Il en existe en réalité 13.

ignore et considere : pour les fragments facultatifs ou obligatoires.

critical : pour les fragments qui doivent se dérouler sans être interrompus.

break : pour les fragments représentants des scenarii exceptionnels ou de ruptures (ex appui sur la touche « Esc »). Le scénario de rupture est exécuté si une condition de garde est satisfaite.

assert : Pour les fragments dont on connaît à l'avance les paramètres du message (exemple : après la saisie des 4 chiffres d'un code, la saisie suivante sera obligatoirement la touche « Entrée »).

seq : indique que le fragment est composé de plusieurs sous fragments qui peuvent s'exécuter dans n'importe quel ordre (mais pas en même temps).

strict : pour les fragments dont les messages doivent se dérouler dans un ordre bien précis.

neg : pour indiquer que la séquence à l'intérieur du fragment n'est pas valide.

ref : permet de faire appel à un autre diagramme de séquence.

Stéréotypes de Jacobson :

A l'intérieur d'un système, il existe très souvent des classes qui possèdent un rôle bien particulier qui serait intéressant de visualiser d'une façon non équivoque dans votre diagramme de séquence. C'est le cas notamment :

- Pour les classes qui représentent des composants de l'IHM.
- Pour la classe qui contrôle globalement le système avec la prise en compte de la gestion événementielle.
- Pour les classes qui implémentent la persistance des attributs (assosiées à une base de données).

Les diagrammes de séquences

Jackobson distinguent les trois stéréotypes suivants :

- « ***boundary*** » : classes qui servent à modéliser les interactions entre le système et ses acteurs.
- « ***control*** » : classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.
- « ***entity*** » : classes qui servent à modéliser des informations durables et souvent persistantes.