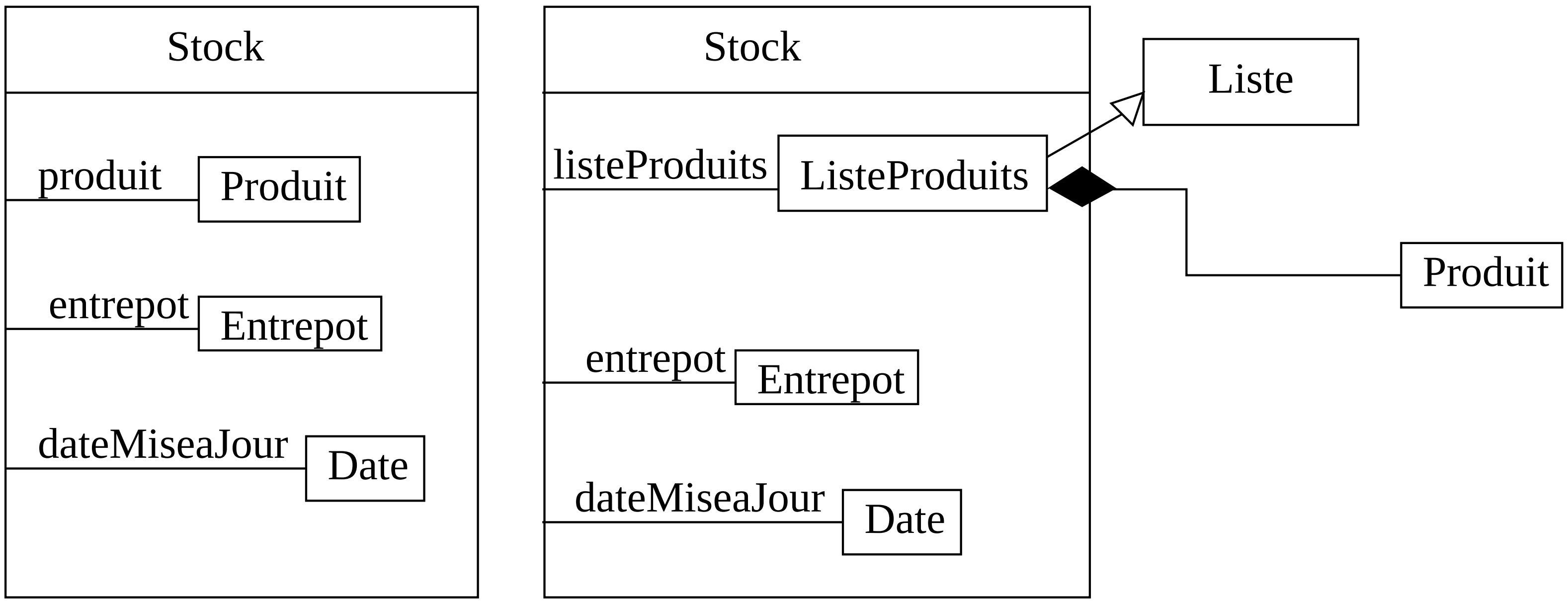
Association de classes

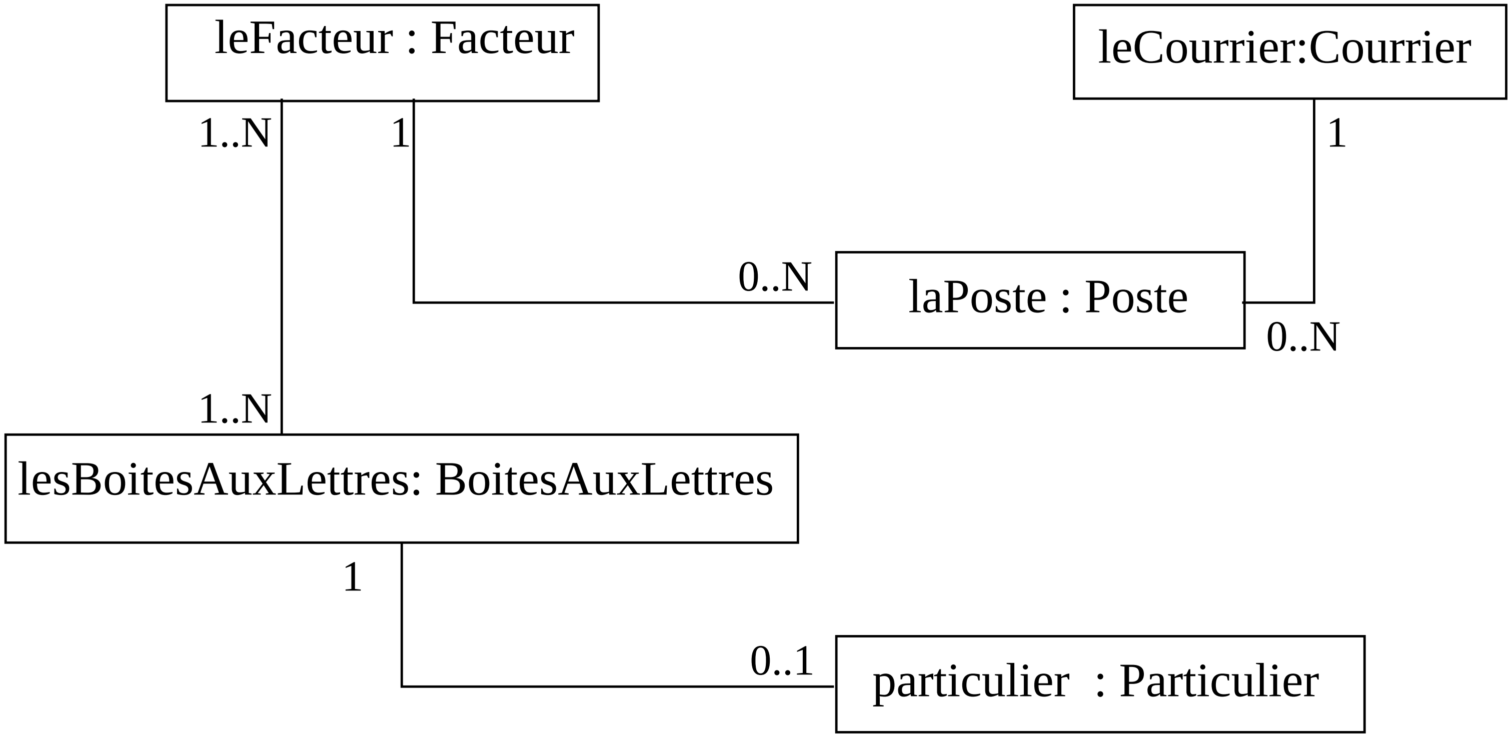
Les liens d'association doivent être portés sur une classe pas sur les champs (instances).



Arité d'une association

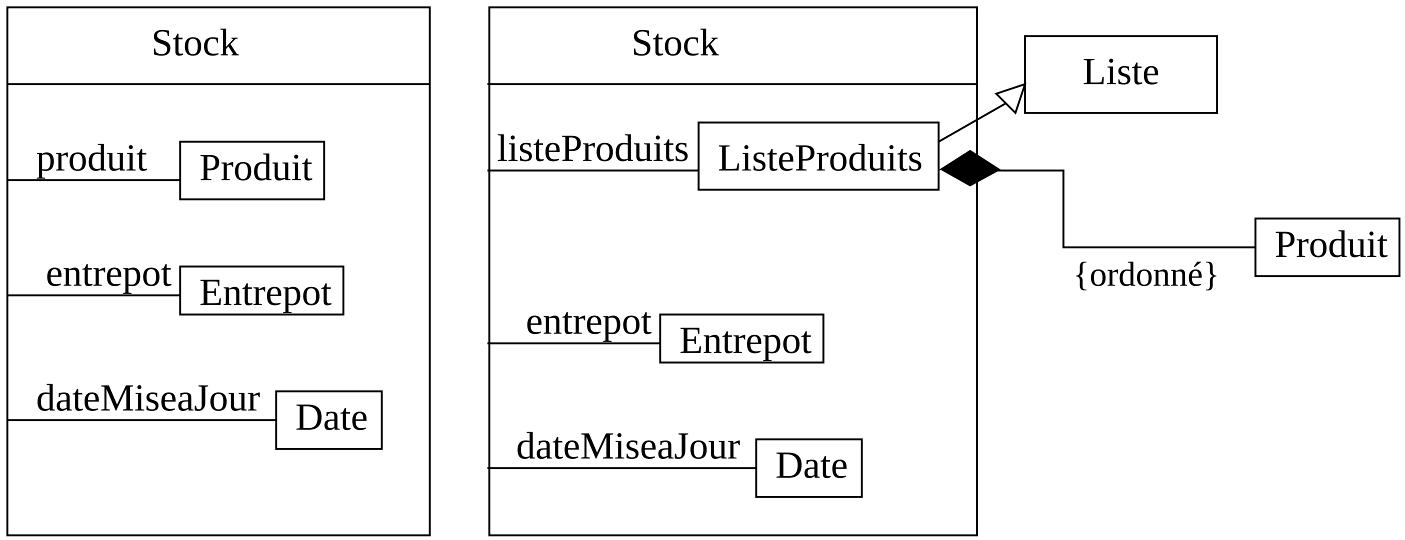
1	1 est une seule
0..1	0 ou 1 association
M..N	de M à N (M et N entiers naturels)
$\star$	de 0 à plusieurs
0..$\star$	de 0 à plusieurs
1..$\star$	de 1 à plusieurs

Marquer les arités dans le diagramme de classe



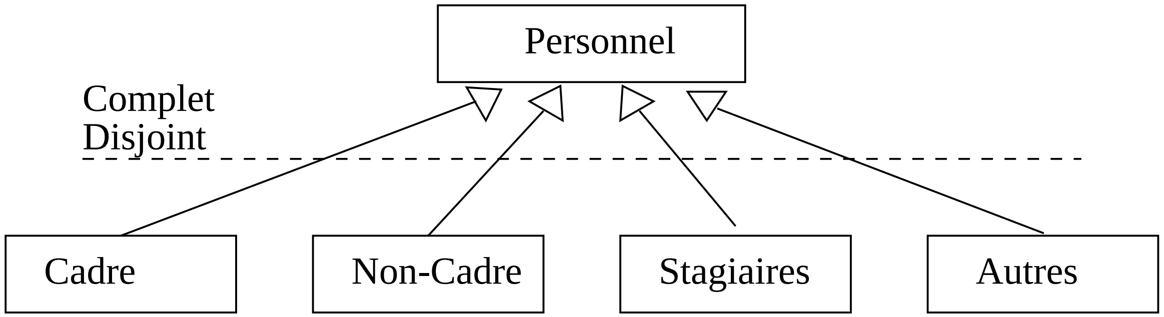
Les contraintes

Il est possible de préciser les contraintes d'une association directement sur le diagramme de classe.



Les contraintes - 2

Les liens d'héritage peuvent aussi être étiquetés par des contraintes.



Les Méta-Classes

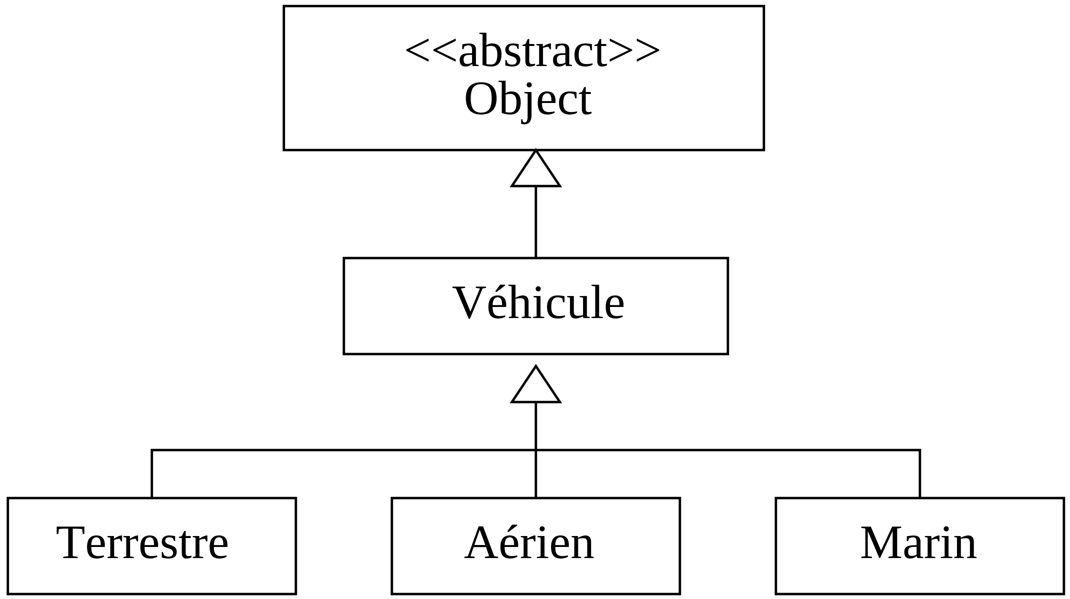
- Définition :
  Une méta-classe définit les caractéristiques d'une classe, c'est un modèle générique de classse (attribut ou comportement).

- Exemple :
  classe abstraite, interface, par extension (abus de langage !) toute classe d'une classe.

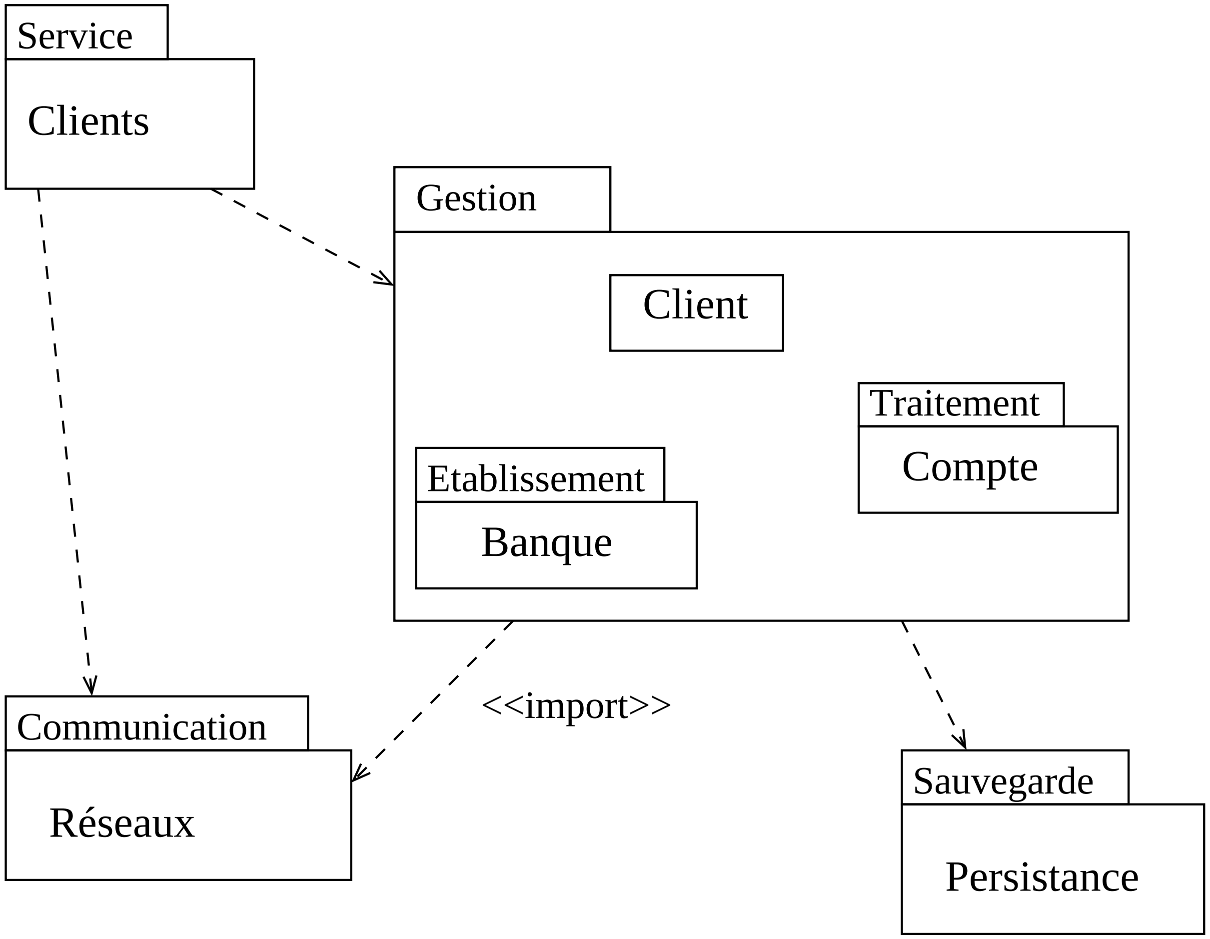
- On notera qu'une méta-classe est également un objet dont la classe est la classe de base de référence à partir de laquelle tous les objets du système sont construits (classe Objet en Java).

- A l'étape d'analyse et de conception, il n'existe pas de différence entre une classe et une méta-classe.

Les Méta-Classes

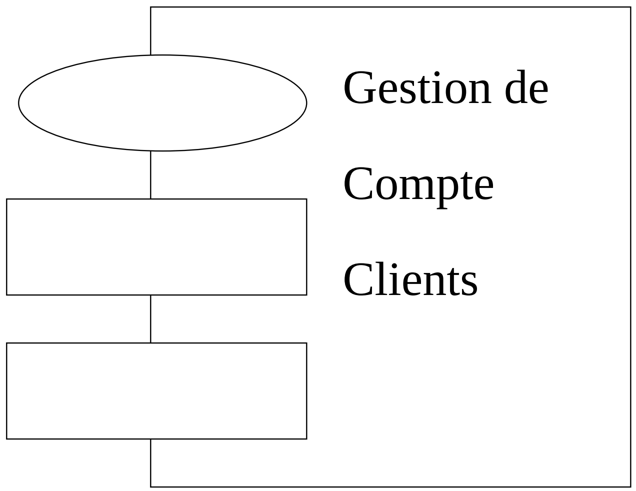


Représentation des Paquetages



Représentation de modules

Les modules sont des unités de compilation. Certains langages de programmation n'ont aucune correspondance avec ce concept.



Le diagramme de classes

- Le diagramme de classe est une **vue statique** du modèle.
- Il décrit la structure interne des classes et leurs relations (dépendances, héritage, composition . . .) .
- Les termes : *static structural diagram* et *class diagram* sont équivalents dans la terminologie UML.
- Il est une collection des éléments du modèle déclaratif. Ces éléments sont classifiés grâce au mécanisme de typage.

Le diagramme d'objets

- C'est le graphe des instances des différentes classes d'objets.
- Il est lui-même une instance du diagramme de classes.
- Il sert uniquement à illustrer des exemples.

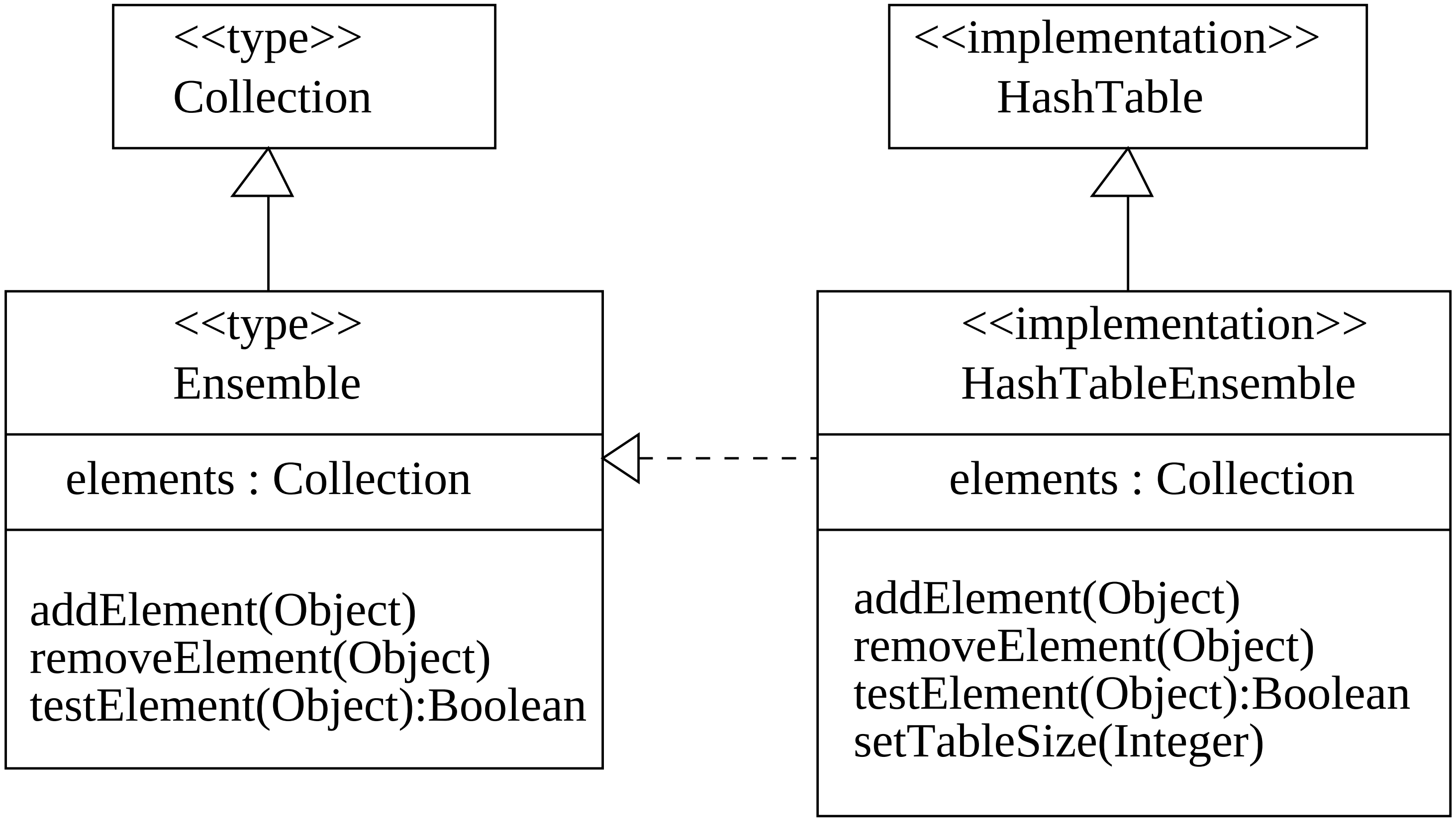
Classifier

➔ Il est également possible de construire un diagramme qui ne contient que les interfaces et les classes abstraites. On parle alors de **Méta-modèle**.

Type vs Implementation

- Une classe peut-être spécialisée par des classes d'implémentation ou de typage.
- Un **type** se caractérise par un rôle (modifiable) qu'un objet peut adopter puis abandonner.
- Une **implémentation** définit la structure de données physique et les procédures qui caractérisent un objet.
- Un objet peut avoir plusieurs types (qui peut être changé dynamiquement) mais une seule implémentation.
- NB : si leur usage est différent, leur structure interne est identique.

Type vs Implementation



Les diagrammes de collaboration

➜ Rôle :
Ils montrent les interactions entre les objets.

Ils expriment le contexte d'un groupe d'objets.
Ils sont des extensions du diagrammes d'objets.

➜ 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 avec des flêches orientées vers le destinataire du message.

Diagramme de collaboration

La numérotation donne l'ordre d'envoi des messages.
Le temps n'est pas représenté.

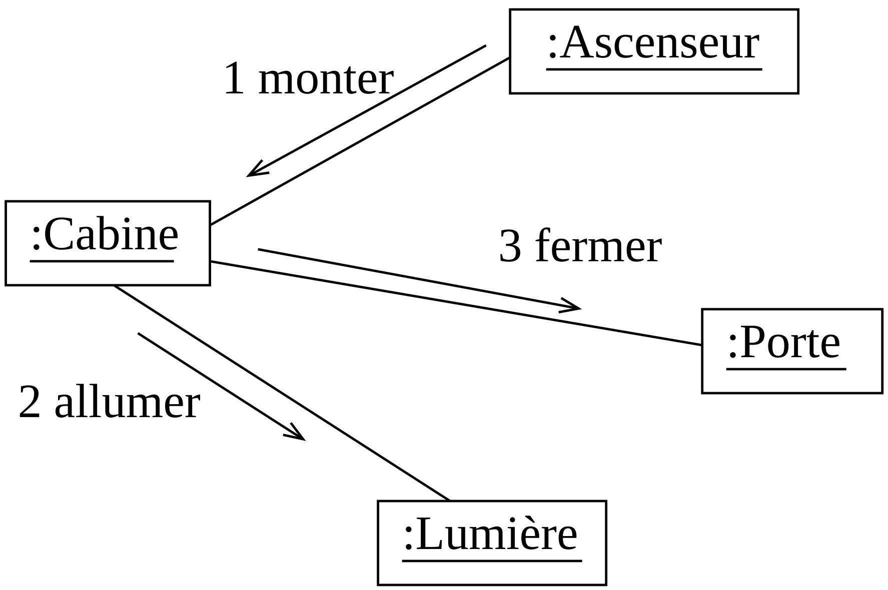


Diagramme de collaboration - 2

- ➔ Le diagramme de collaboration montre simultanément les interactions entre les objets et les relations structurelles qui permettent ces interactions.

- ➔ Ne pas confondre ces liens avec ceux de composition des diagrammes de classes ou d'objets.

Cas d'utilisation[1]

Problème :

➔ Les besoins d'un système (cf cahier des charges) sont souvent exprimés de manière non structurée, sans forte cohérence (imprécision, oublis, contradictions).

Principes de la démarche :

➔ Décrivent le comportement du système du point de vue d'un utilisateur.

➔ Permettent de définir les limites du système et les relations entre le système et l'environnement.

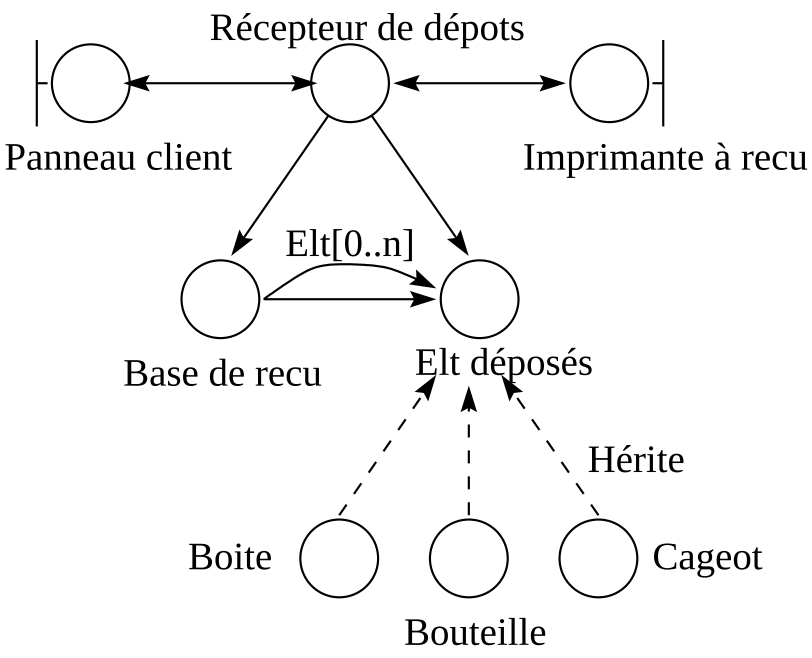
➔ La structuration de la démarche s'effectue par rapport aux interactions d'**une seule catégorie d'utilisateurs** à la fois.

[1]Ont été formalisés par I. Jacobson *Object-Oriented Software Engineering, A Use Case Driven Approach*, Addison-Wesley, 1992

Le modèle des cas d'utilisation

➔ Le modèle des cas d'utilisation représente une partie du modèle des besoins.

➔ Il est issu du modèle d'analyse. Une fois les **objets identifiés** et **décrits**, on peut exprimer comment ces objets participent au cas d'utilisation.



Représentation UML

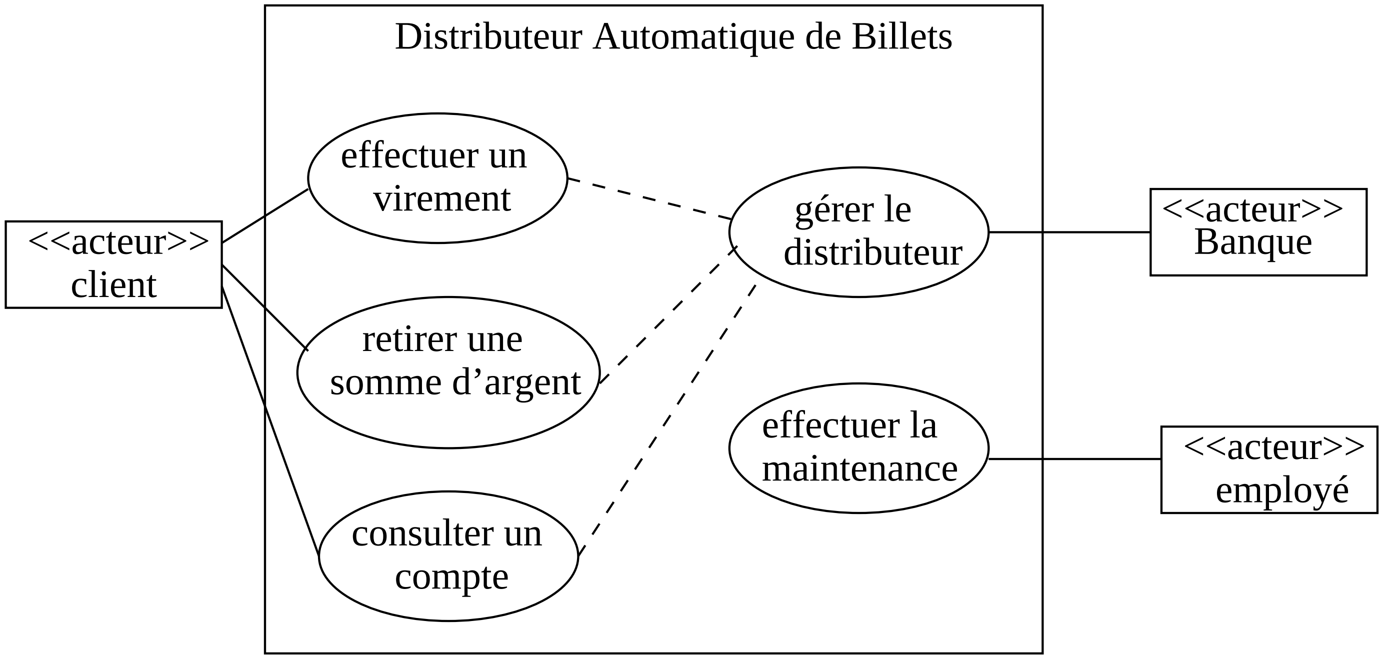
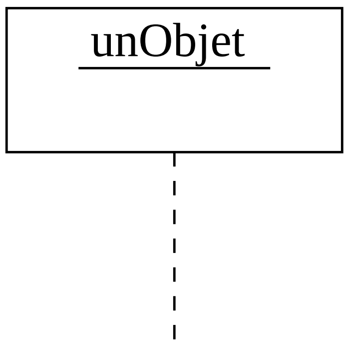


Diagramme de séquence

➔ Montrent des interactions entre objets selon un point de vue temporel.

➔ Pas de représentation explicite du contexte des objets.

➔ Notation[2] :
Un objet est matérialisé par un rectangle est une barre verticale appelée ligne de vie



[2]Object Message Sequence Chart, Siemens Pattern Group. Wiley 1996 *Pattern-riented Softawre*

Diagramme de séquence - 2

➜ L'ordre d'envoi d'un message est donné par la position sur l'axe vertical.

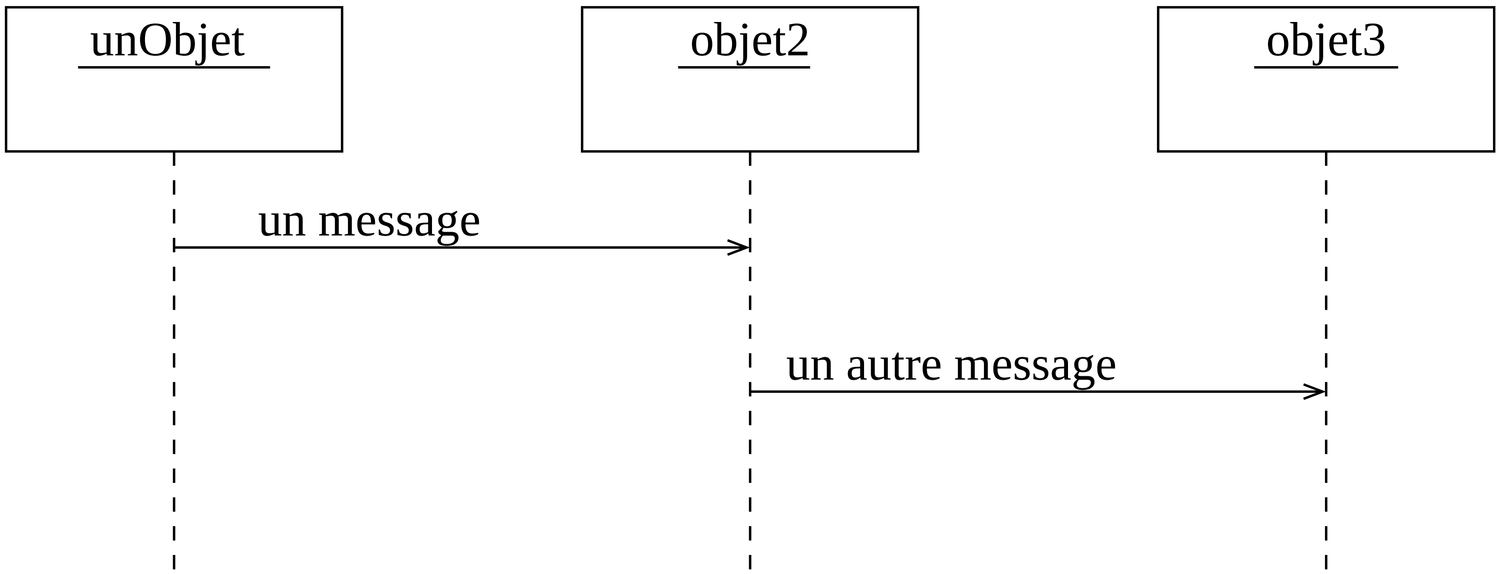


Diagramme de séquence - utilisation

Deux utilisations possibles :

➜ Documentation des cas d'utilisation : description des interactions entre objets sans détails de synchronisation. Les flèches correspondent à des **événements** qui surviennent dans le domaine de l'application. Pas de distinction entre **flots de contrôle** et **flots de données**.

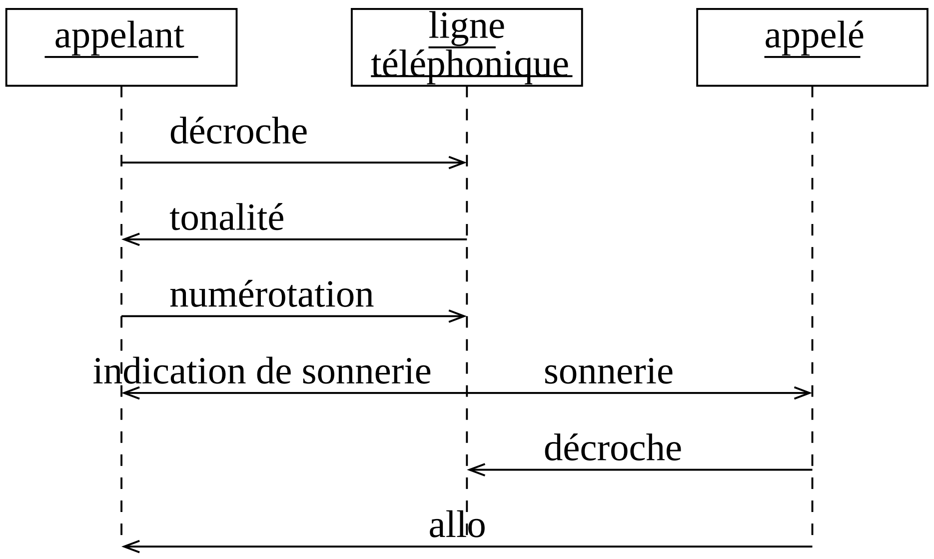


Diagramme de séquence - utilisation - 2

➜ Représentation précise des interactions entre objets.

le concept de message unifie toutes les formes de communication : appel de procédures, événemnt discret, signal de flots, interruption matérielle . . .

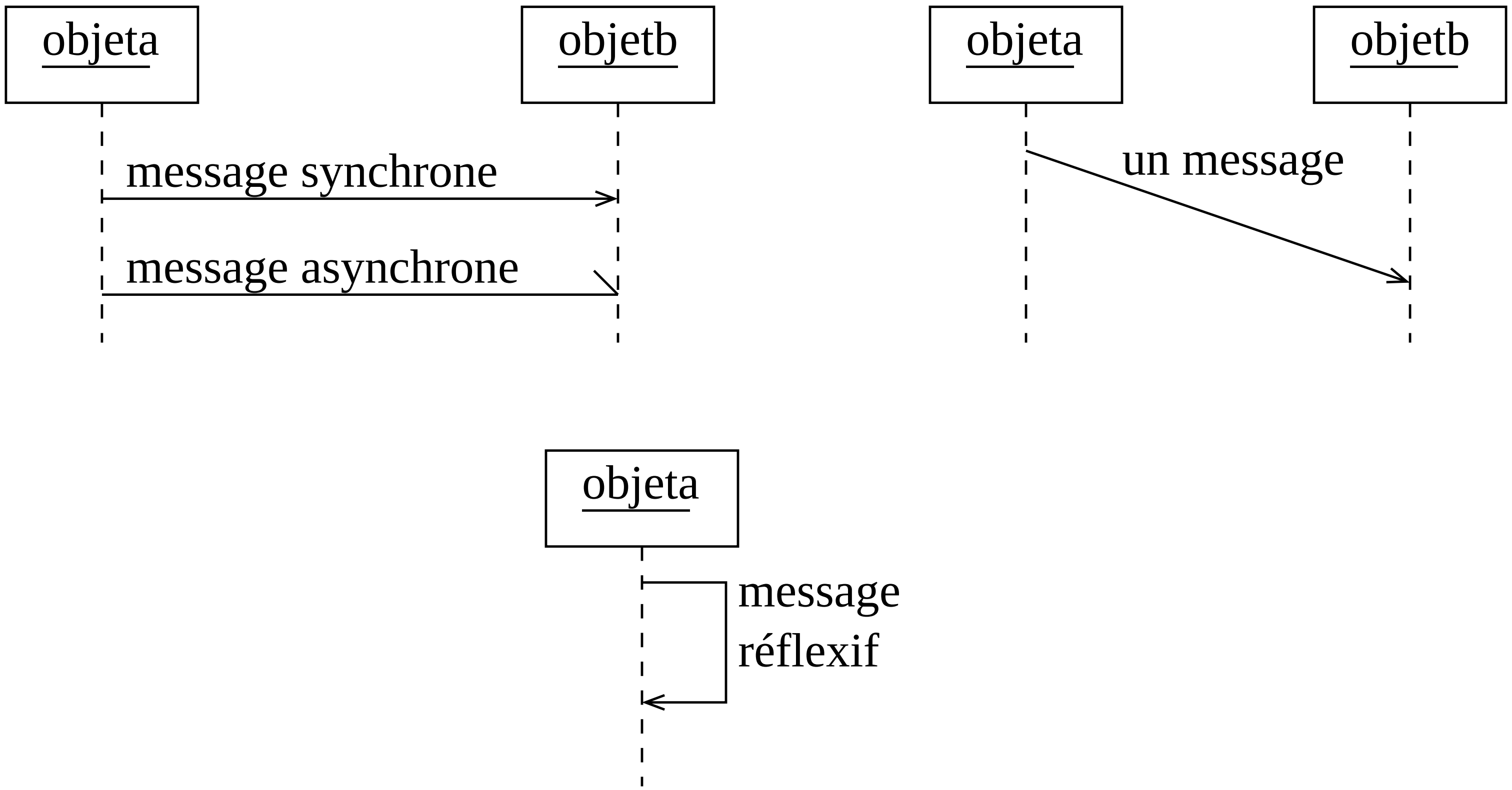


Diagramme de séquence - utilisation - 3

➔ Un message réflexif peut aussi être un point d'entrée dans une activité qui s'exerce au sein d'un objet (par exemple composite).

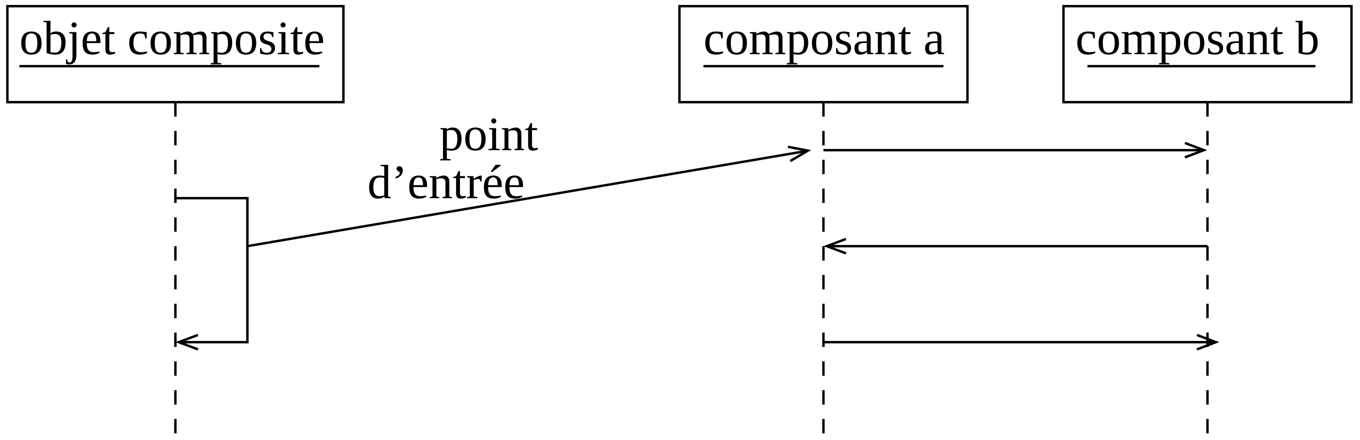


Diagramme de séquence - utilisation - 4

➜ Création et destruction d'un objet.

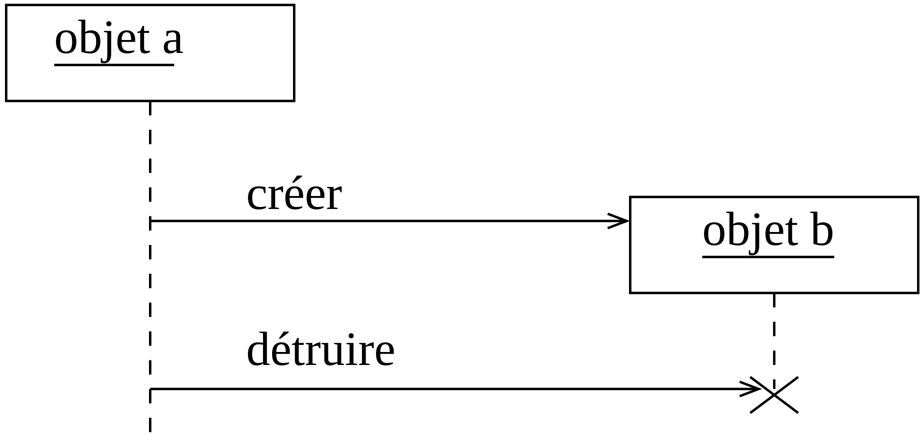


Diagramme de séquence - utilisation - 5

➔ Représentation des périodes d'activité des objets = temps pendant lequel un objet effectue une action.

➔ le début et la fin de la bande rectangulaire correspondent au début et à la fin d'une période d'activité.

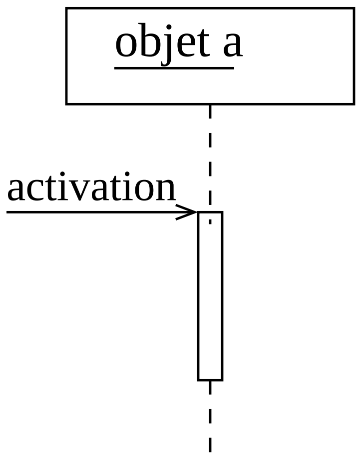


Diagramme de séquence - utilisation - 6

➜ Les digrammes de séquence permettent également de représenter les périodes d'activité des objets.

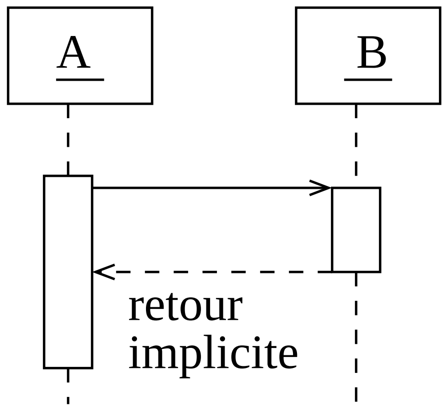


Diagramme de séquence - utilisation - 7

➔ Dans le cas d'envoi asynchrone, le retour doit être signalé.

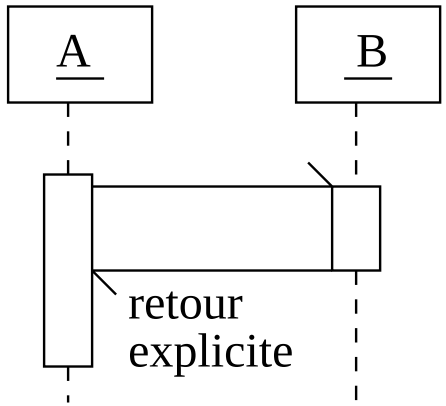
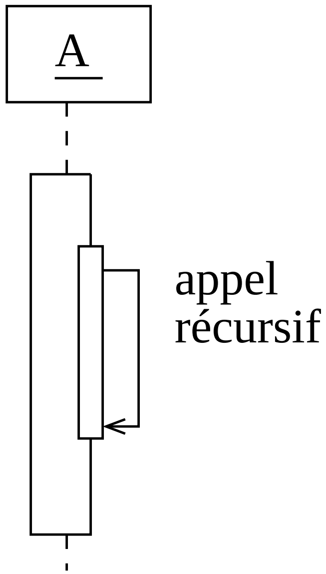


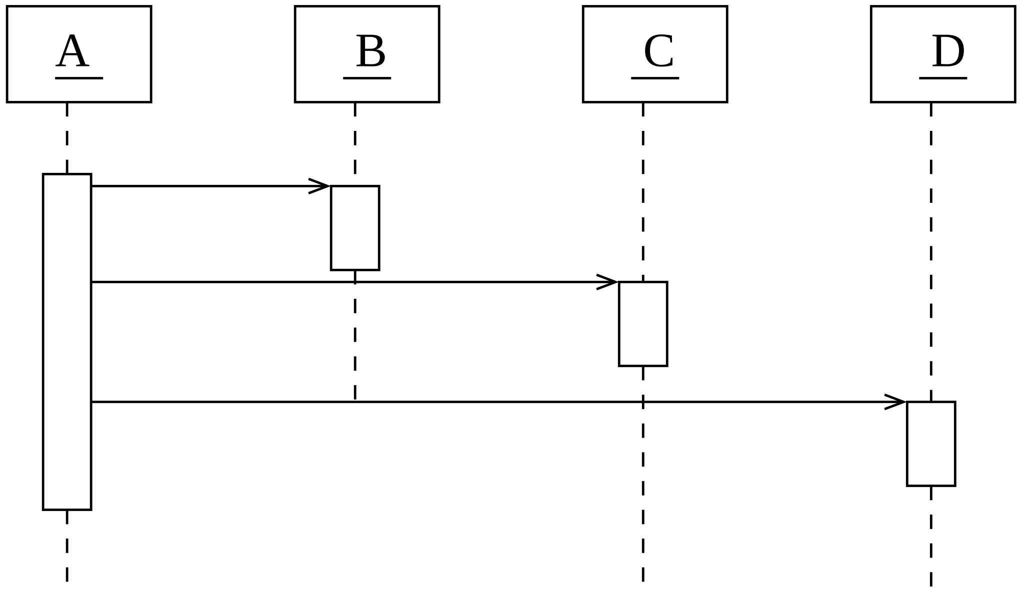
Diagramme de séquence - utilisation - 8

➜ Cas des messages récursifs



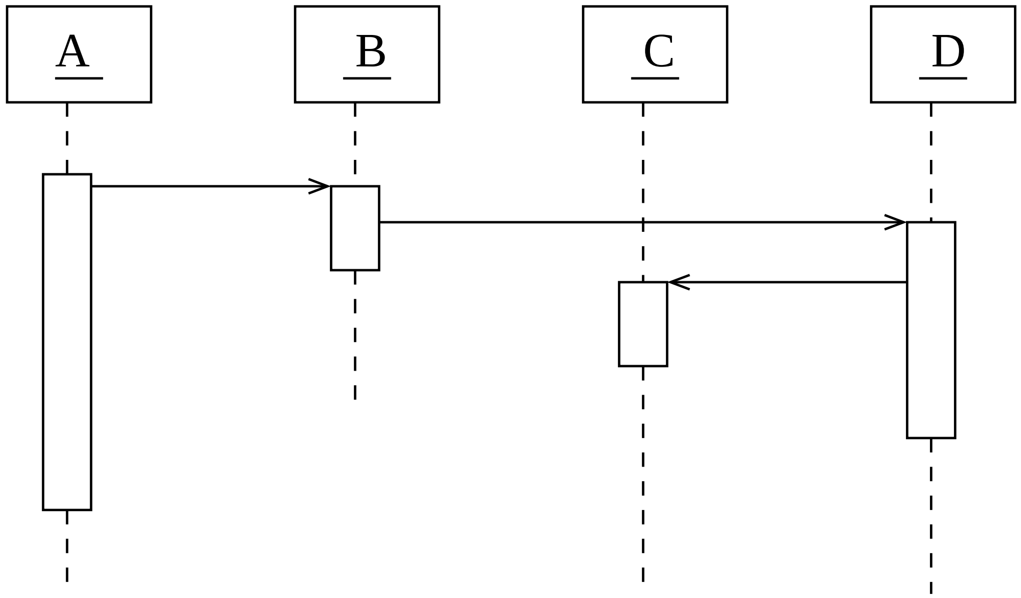
Mode centralisé - Mode décentralisé

➔ Les diagrammes de séquences reflètent le choix des structures de controle.

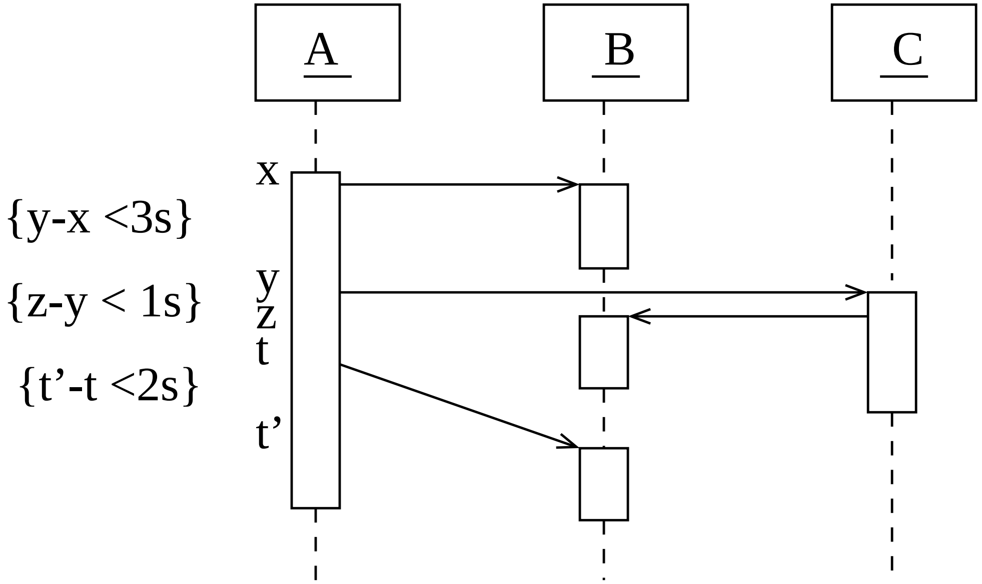


Mode centralisé - Mode décentralisé

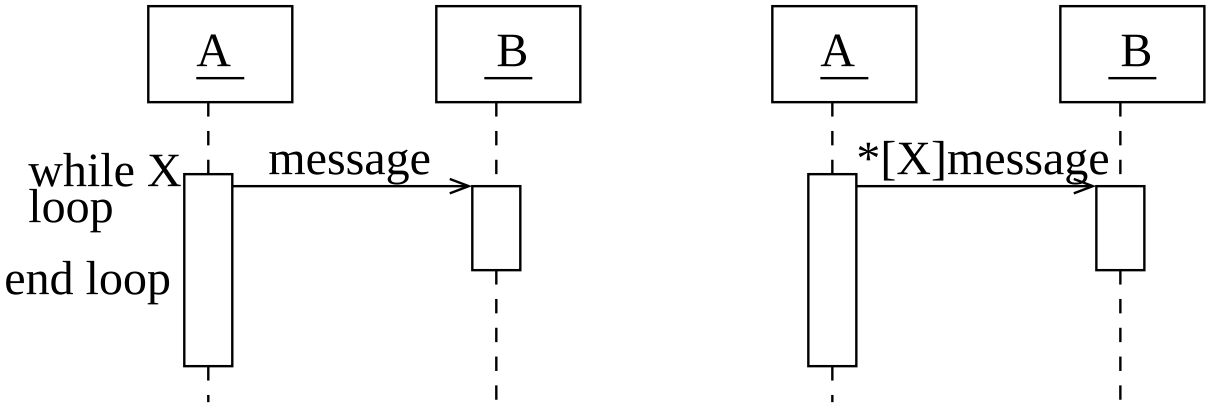
➔ Envoi décentralisé de messages.



Expression des contraintes temporelles



Boucles et branchements



Branchement Conditionnel