

# Technologie Objet

---

## Diagramme de séquence

Xavier Crégut  
<Prénom.Nom@enseeiht.fr>

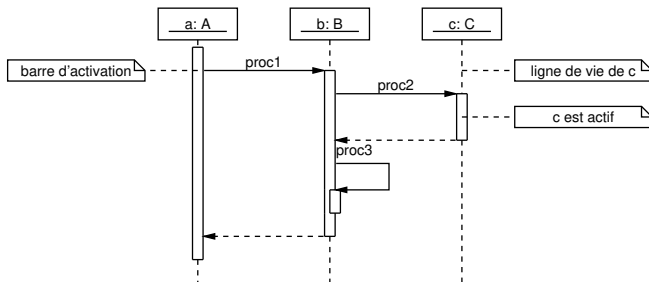
ENSEEIHT  
Sciences du Numérique

## 1 Diagramme de séquence

## Diagramme de séquence

**Objectif :** Décrire les interactions entre objets en privilégiant l'aspect temporel (chronologie des envois de messages, de haut en bas).

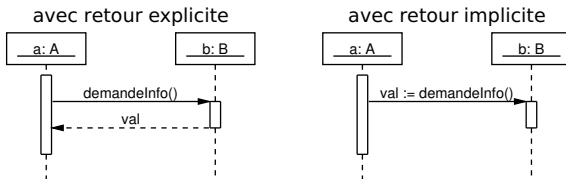
**Principaux constituants :**



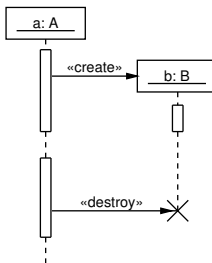
- Le temps s'écoule du haut vers le bas
- La barre d'activation montre que du code de l'objet est en train de s'exécuter
- On se limite aux messages synchrones (il existe aussi des messages asynchrones)

## Diagramme de séquence (2)

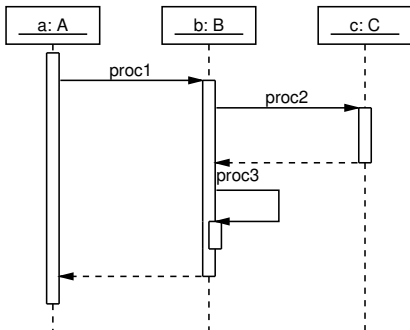
### Retour de méthode



### Création et destruction d'objet



## Diagramme de séquence et code Java



```
...
public class B {
    ...
    ... proc1() {
        ...
        c.proc2()
        ...
        this.proc3()
        ...
    }
    ... proc3() {
        ...
    }
}

public class C {
    ...
    ... proc2() {
        ...
    }
}
```

- Souvent, pas de correspondance parfaite entre diagramme de séquence et code Java.
- Un diagramme de séquence est un exemple, une simplification.

## Des raffinages aux diagrammes de séquence

**Comment** « Arbitrer un jeu entre un devin et un maître » ?

Demander au maître de choisir un nombre

Indiquer au devin les limites du jeu

**Répéter**

Demander au devin de faire une proposition

Demander au maître une indication

Donner au devin l'indication

**Jusqu'À** nombre trouvé

Féliciter le devin

- Les participants sont explicités : devin, maître... et arbitre
- La règle « une étape est un verbe à l'infinitif » est souvent remplacée par « une étape est de la forme : sujet verbe complément »

L'arbitre demande au maître de choisir un nombre

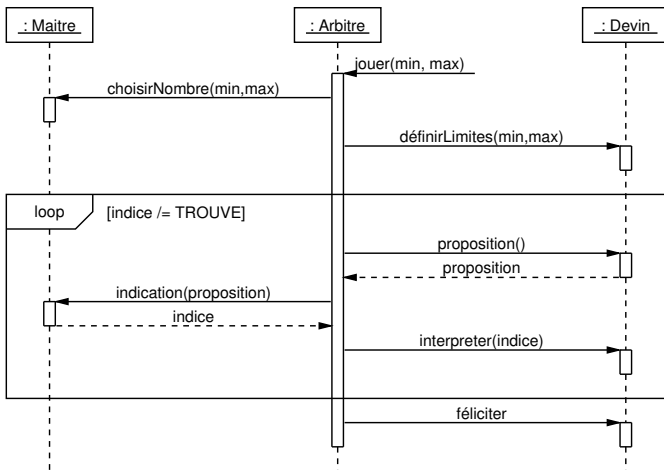
L'arbitre indique au devin les limites du jeu

...

Comment représenter les structures de contrôle sur un diagramme de séquence ?

## Les cadres d'interaction (frame), depuis UML 2

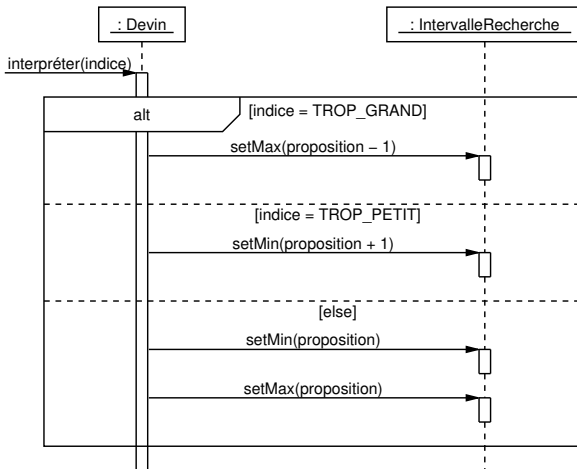
Vers l'expression d'un algorithme



L'étiquette **loop** correspond à un **TantQue** : le contenu du cadre est répété.

## Les cadres d'interaction (frame), depuis UML 2 (2)

Vers l'expression d'un algorithme



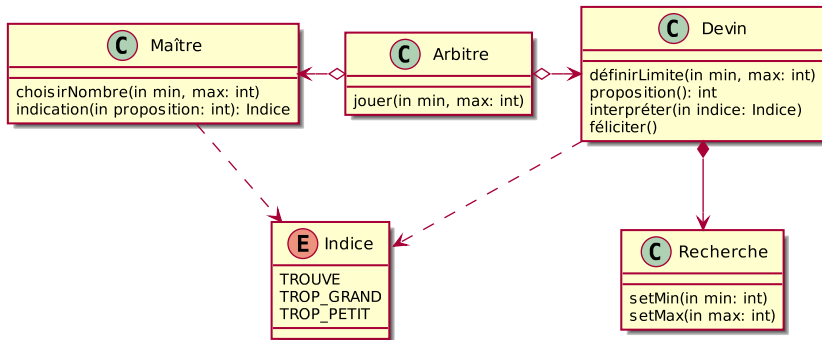
**alt** correspond à une conditionnelle : chaque région est un cas avec sa garde entre crochets.



## Diagramme de classe correspondant

Du diagramme de séquence, on peut déduire des éléments du diagramme de classe :

- les classes : type des objets du diagramme de séquence
- les méthodes : message sur une interaction (dans la classe de l'objet cible)
- les dépendances entre classes (mais pas le type de relation) : chemin de données entre la classe de l'objet origine de l'interaction et la classe de l'objet cible.



## Bilan sur les diagrammes de séquence

- Cadres d'interaction : ajoutés dans la dernière version majeure (UML 2)
- Étiquettes possibles pour les cadres d'interaction :
  - alt : alternatives
  - loop : répétition (while ou foreach)
  - opt : optionnel (if sans else)
  - par : exécution parallèle
  - negative : une interaction invalide
  - ...
- Ils s'**éloignent de l'objectif initial** des diagrammes de séquence :
  - on ne montre pas simplement une exécution particulière
  - mais plusieurs exécutions sur un même diagramme
  - ceci répond à un souci de factorisation... parfois au détriment de la lisibilité
  - voir les diagrammes d'activité d'UML
- Les diagrammes de séquence sont **longs à dessiner** (même avec un outil dédié)
- Un diagramme de séquence explicite de **nombreux choix** : objets, méthodes...
  - souvent trop tôt quand on en est à comprendre le problème
  - un raffinement (décomposition informelle peut alors être préférable)
  - à utiliser surtout pour expliquer un algorithme dans des conditions particulières