

Réalisation des Cas d'Utilisation

« Diagrammes d'interactions »

- Le cas d'utilisation présente une vue externe du système
- Les **diagrammes d'interaction** montrent comment des sociétés d'objets peuvent collaborer pour réaliser les cas d'utilisation
- **Interaction**: a collection of communications between instances, including all ways to affect instances, like operation invocation, as well as creation and destruction of instances
- The communications are partially ordered (in time)

Les diagrammes d'interactions

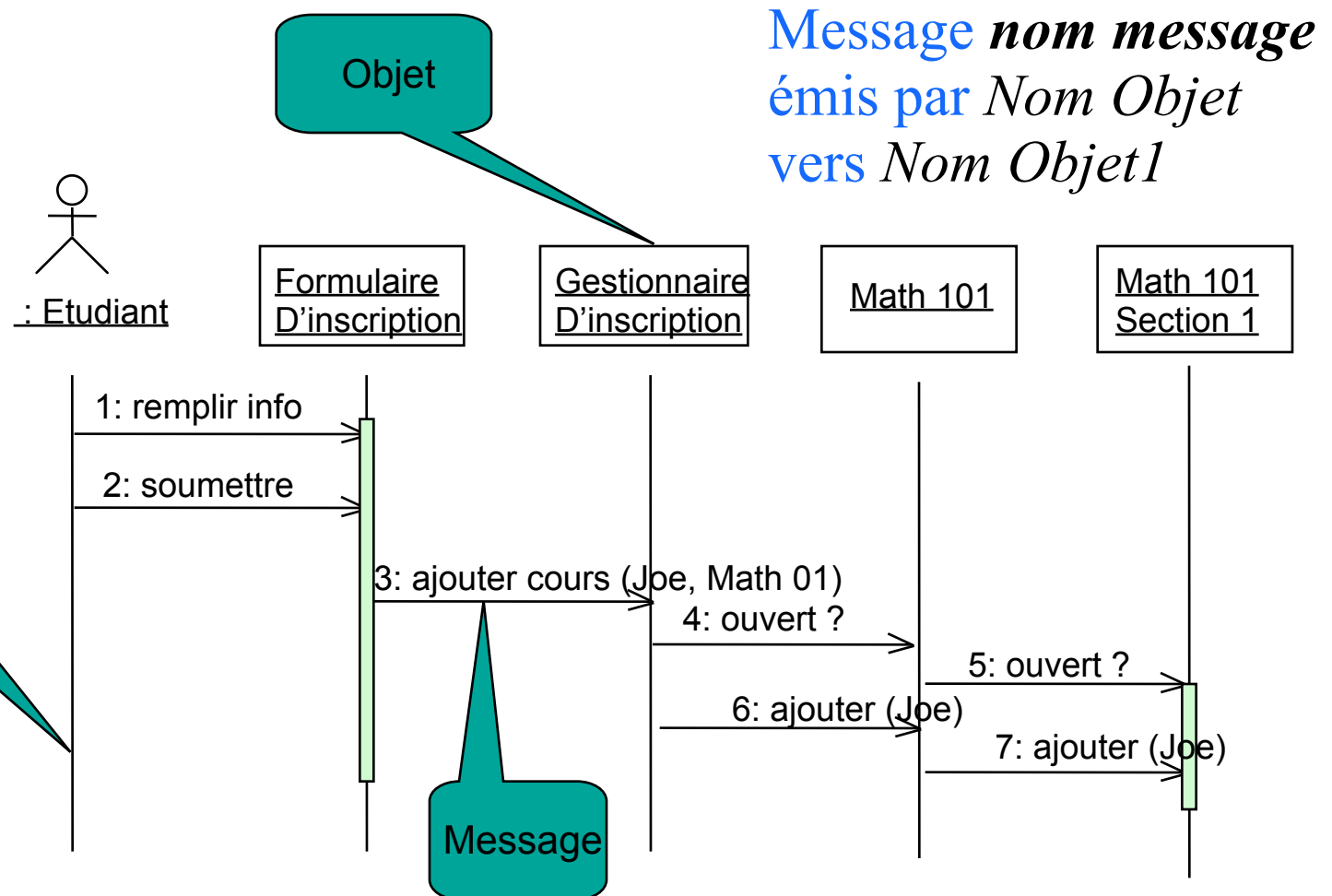
- 2 notations :
 - les diagrammes de séquences
 - les diagrammes de collaboration
- **Diagrammes de séquences :**
 - mise en évidence des objets et des messages échangés
 - correspondent à la structure temporelle
- **Diagrammes de collaboration :**
 - établissement des responsabilités - qui coopère avec qui pour l'obtention d'un but
 - correspondent à la structure spatiale

Nota : les 2 notations sont duales, et souvent on se contente de l'une ou de l'autre !

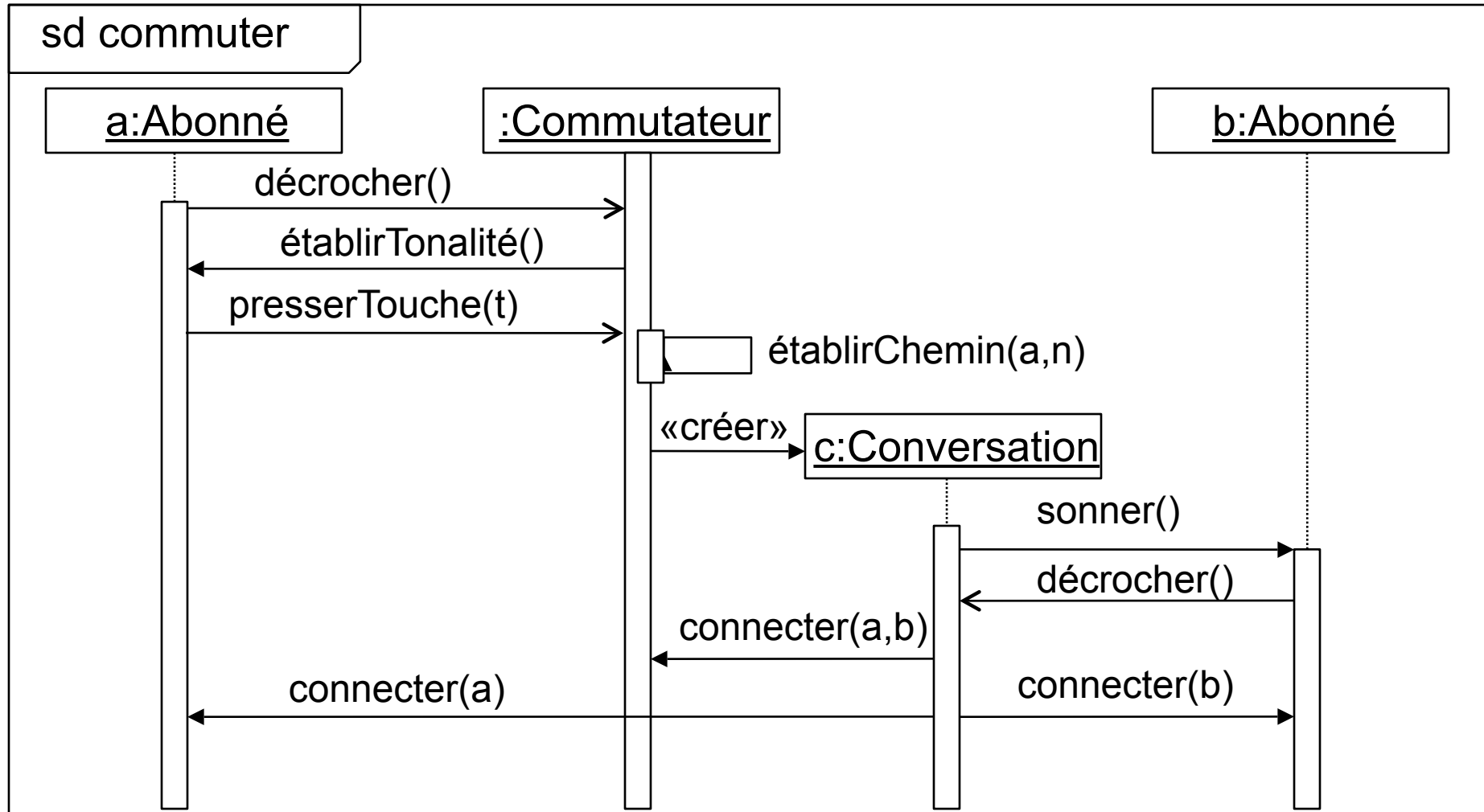
Exemples

Diagramme de Séquence

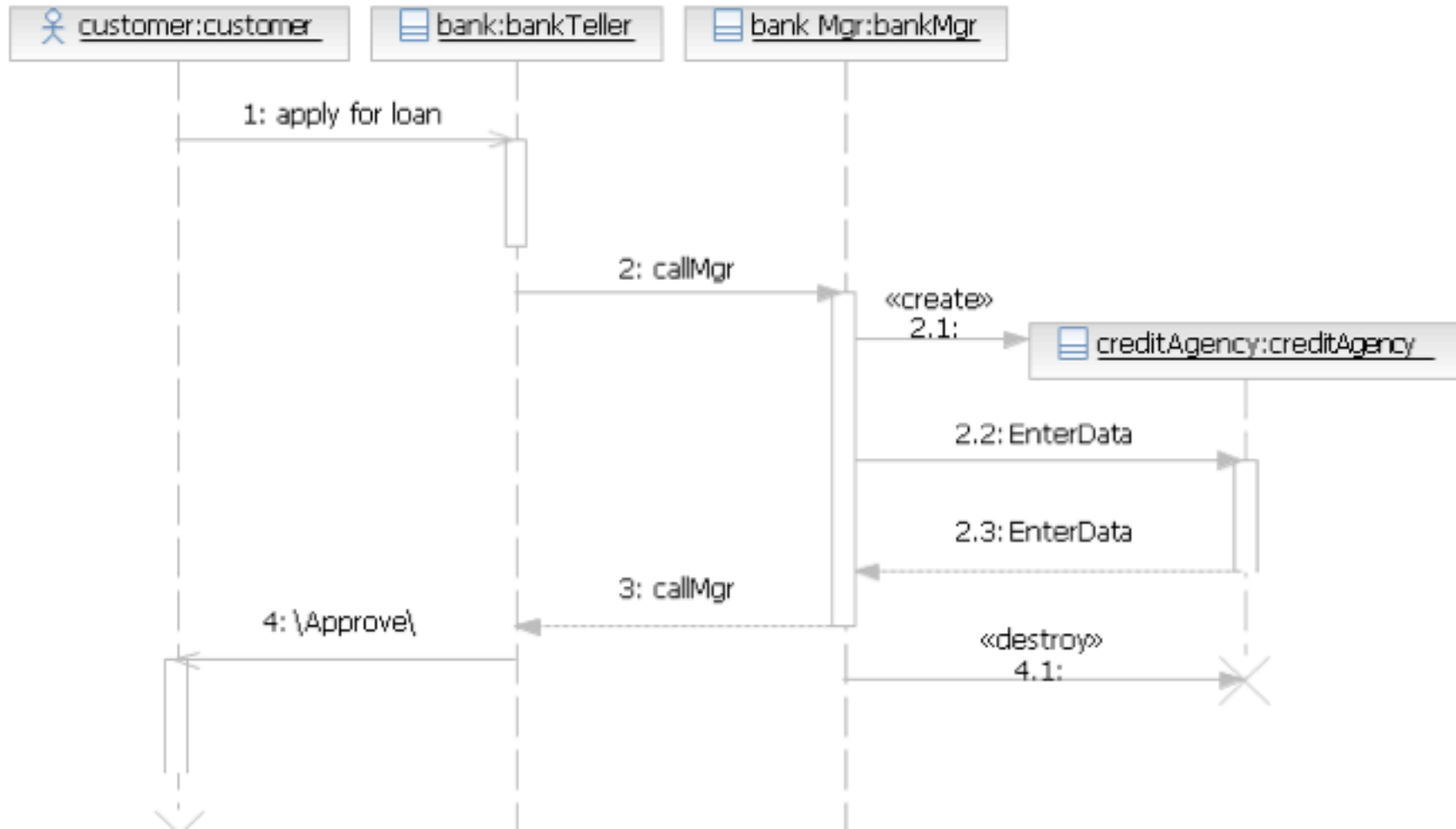
- Vue temporelle de l'interaction entre objets



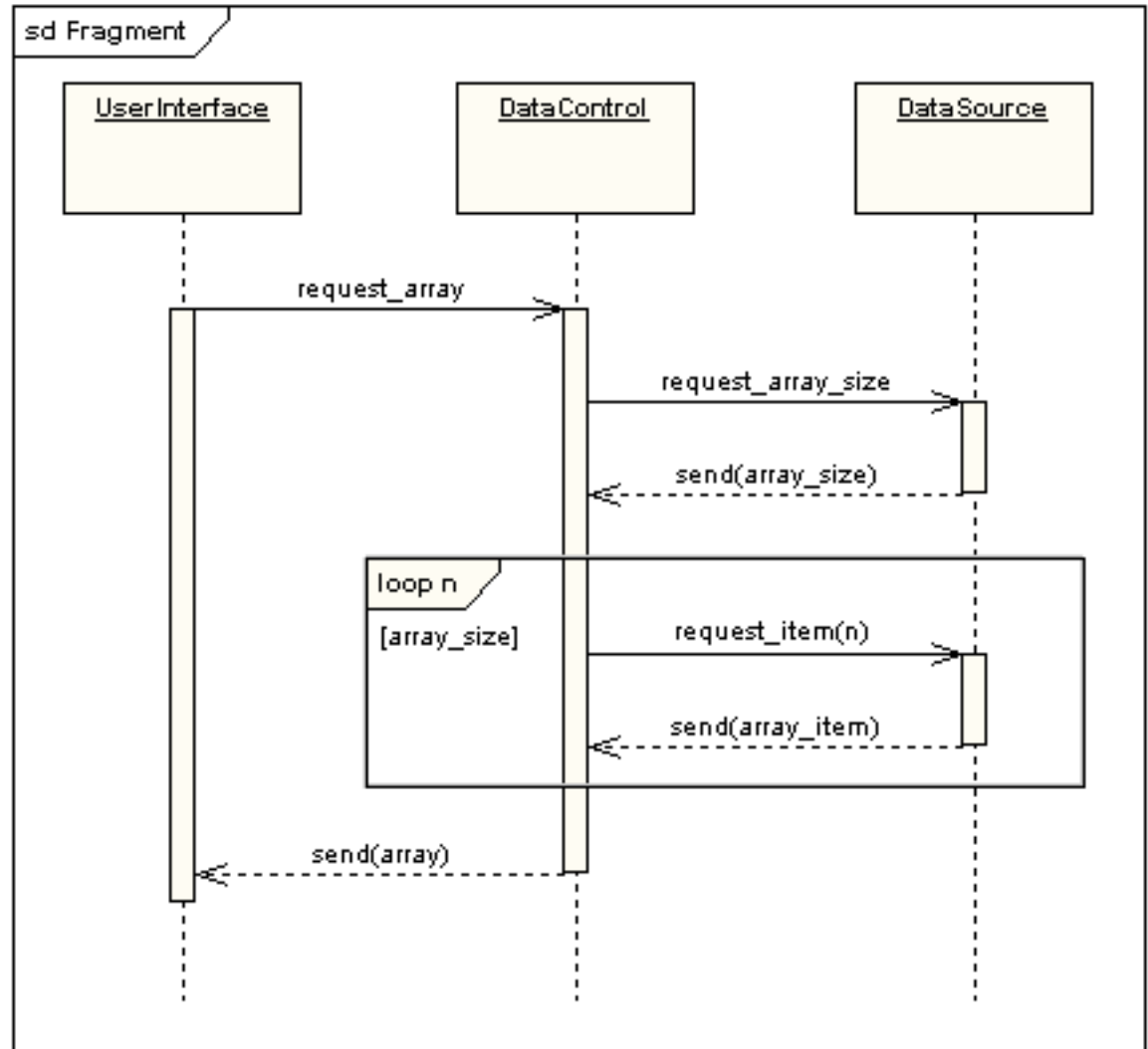
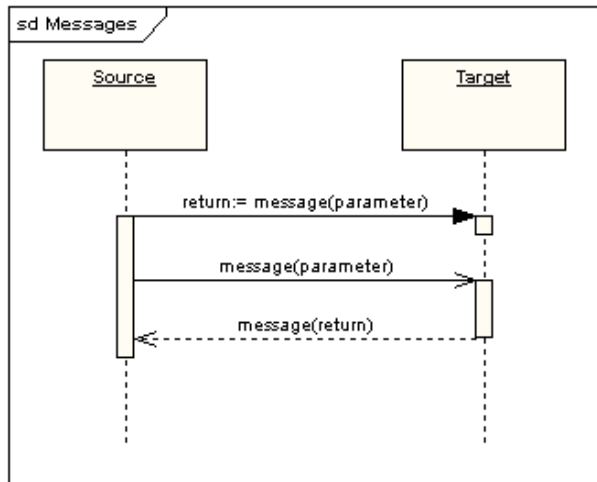
Objets Anonymes, récursion, création



Destruction,



Boucle, et Messages synchrones



Partie 1 : *Diagramme d'interaction*

Olivier Gerbé

Types de messages

- Appel
- Retour
- Envoi
- Création
- Destruction

Définition

- *Un diagramme* d'interaction présente une interaction, c'est à dire, un ensemble d'objets et les messages qu'ils échangent.
- Deux types de diagramme d'interaction :
 - diagramme de séquence
 - diagramme de collaboration

Message

- Un *message* est la spécification d'une communication entre objets avec les informations nécessaires pour qu'une activité s'ensuive.
- Exemple
 - dépiler()
 - presserTouche(2)

Message de retour

- Un message de retour retourne une valeur à l'appelant.
- Représentation graphique

valeurPile
←-----

Message d'envoi

- Un message d'envoi envoie un signal à un objet.
- Un message d'envoi permet d'invoquer une opération d'une manière asynchrone.
- Représentation graphique



Message de création

- Un message «créer» invoque l'opération de création d'un objet.
- Représentation graphique



Message de destruction

- Un message «détruire» invoque l'opération de destruction d'un objet.
- Un objet peut se suicider en s'envoyant un message détruire.
- Représentation graphique



Partie 2 : *Diagramme de séquence*

Olivier Gerbé

Ligne de vie

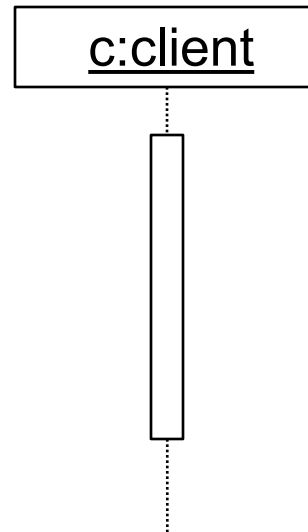
- Une ligne de vie représente l'existence d'un objet pendant une période de temps.
- Représentation graphique

c:Client



Activation

- Une *activation* représente le temps durant lequel un objet est actif, c'est à dire en train d'exécuter une opération.
- Représentation graphique



Synchronisation

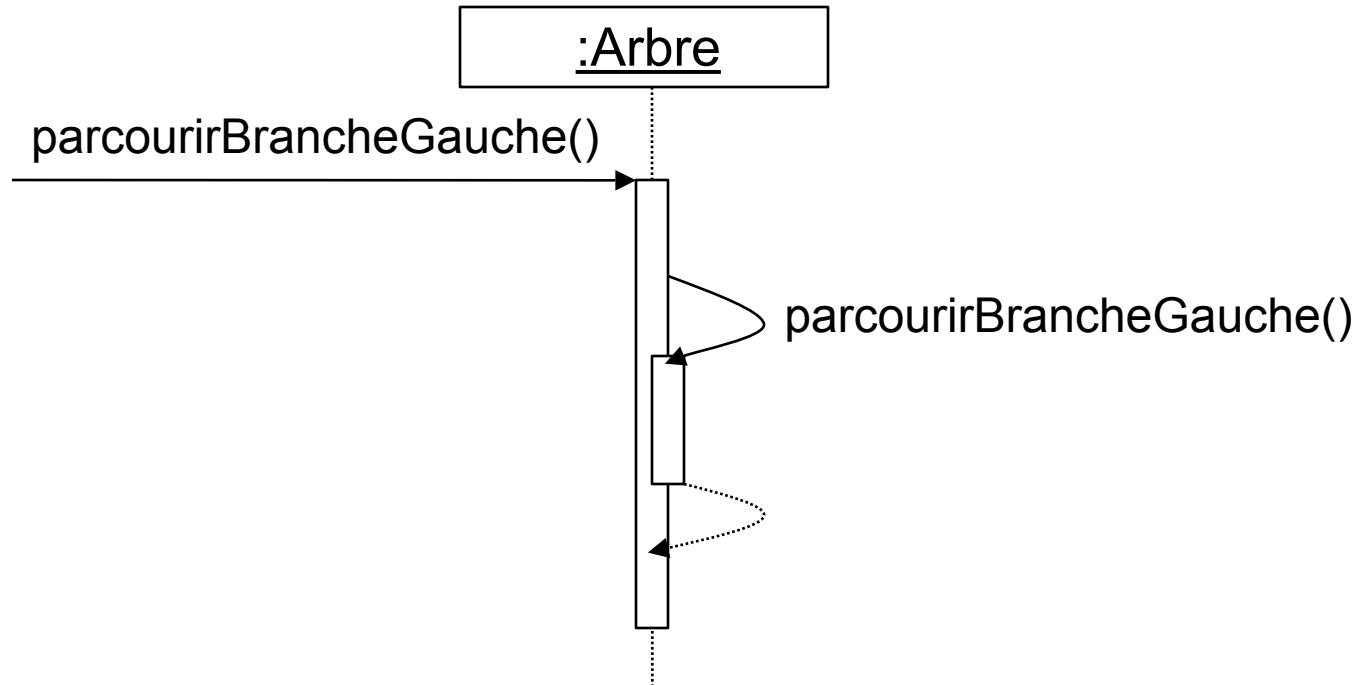
- La *synchronisation* est le mode par défaut des messages :
 - message d'appel
 - message de retour
 - message de création
 - message de destruction
- Le contrôle est passé de l'appelant à l'appelé; l'opération de l'appelant est suspendue.
- Le contrôle est rendu à l'appelant à la fin de l'opération appelée; l'appelant reprend l'exécution de l'opération en cours.

Concurrence

- La *concurrency* est spécifiée par l'utilisation de message d'envoi.
- Un signal est envoyé à un objet; l'objet expéditeur continue son opération.

Récursion

- Un message peut être récursif.
- Représentation graphique



Opérateurs

- Alternatives
- Option
- Boucles
- Etc.

Diagrammes de séquences (scénarios)

- Les éléments constitutifs d'un diagramme de séquence sont :
 - Un ensemble d'objets (et/ou d'acteurs)
 - Un message initiateur du scénario
 - La chronologie des messages échangés subséquemment
 - Les contraintes de temps (aspects temps réel)
- Un scénario a comme principale qualité son exemplarité.

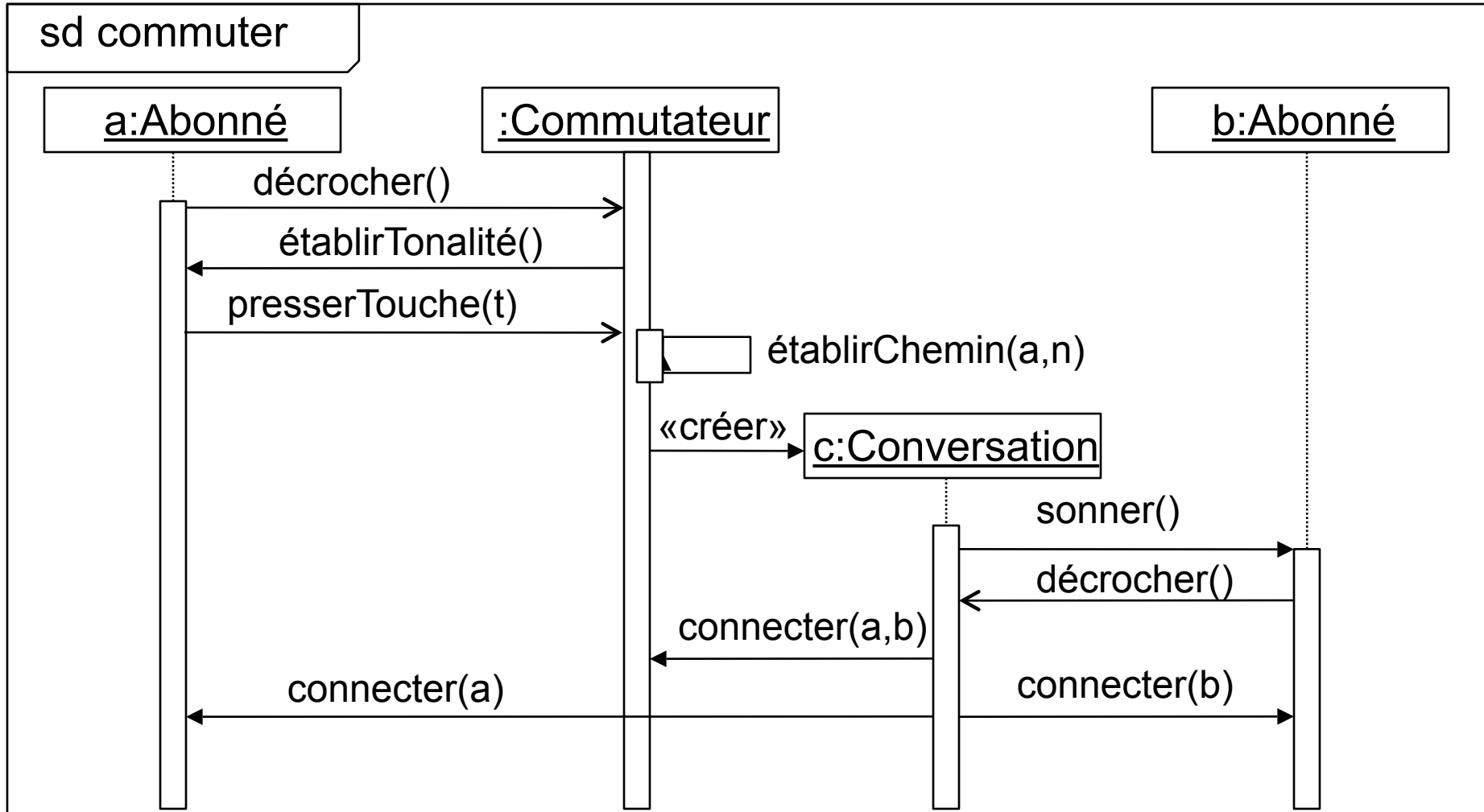
Concepts fournis par les diagrammes de séquence

- **Objet :**
 - objet dédié : une instance particulière d'une classe
 - objet anonyme : n'importe quelle instance d'une classe
- **Stimulus :**
 - une instance de message i.e. représentation de l'échange d'information entre objets
- **Supporte des flots de données et divers types de synchronisation**
- **Déroulement temporelle :**
 - **vertical** : représente la ligne de vie des objets et les périodes d'activité des objets
 - **horizontal** : représente l'enchaînement des stimuli entre 1 objet émetteur et 1 objet récepteur i.e. les flots de contrôle (séquence, répétition, alternative)

Ligne de vie et activation

- La «ligne de vie» représente l'existence de l'objet à un instant particulier
 - Commence avec la création de l'objet
 - Se termine avec la destruction de l'objet
- L'activation est la période durant laquelle l'objet exécute une action lui-même ou via une autre procédure

Notations

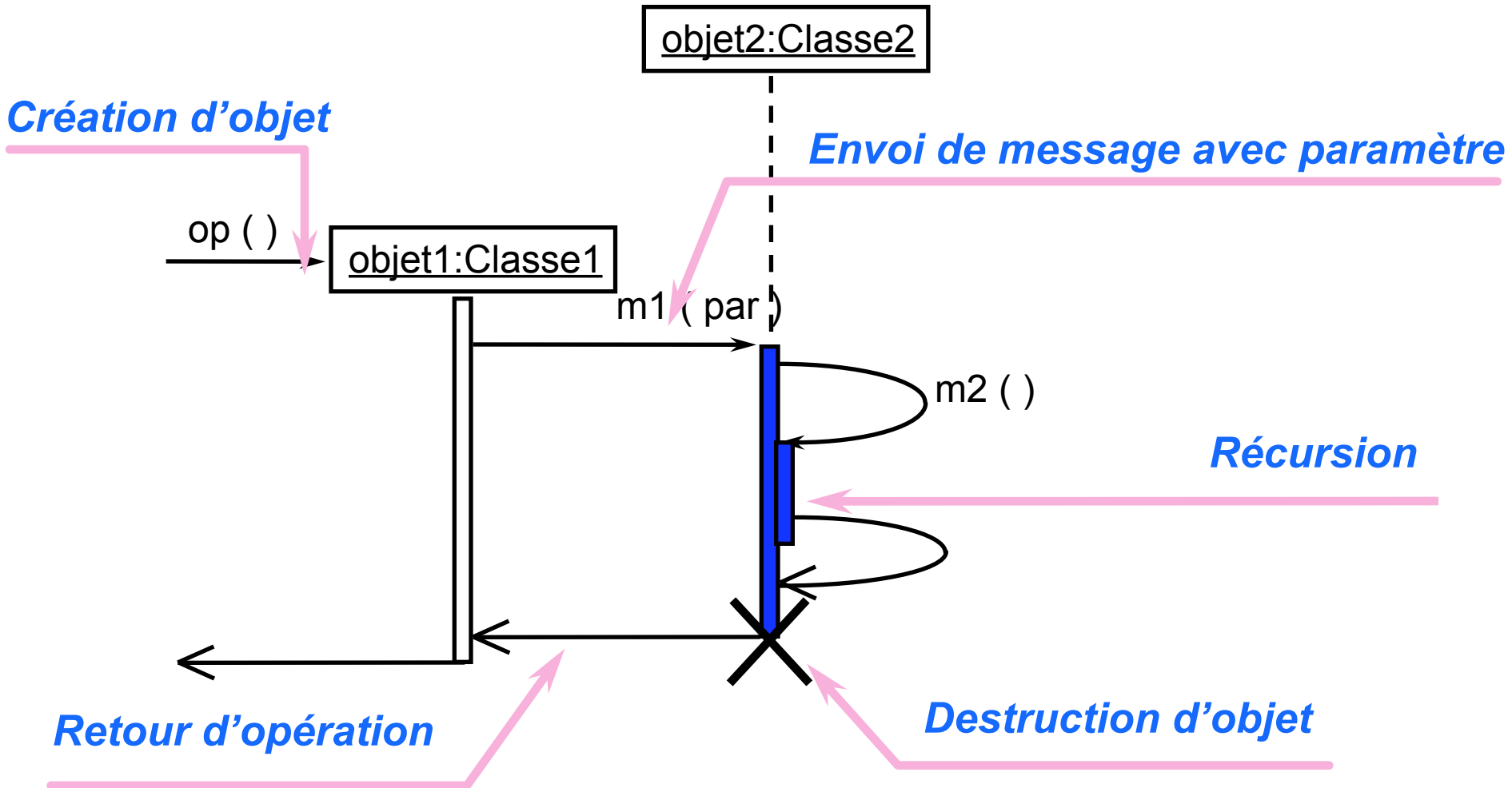


Messages

- Communication entre objets
 - Des paramètres
 - Un retour

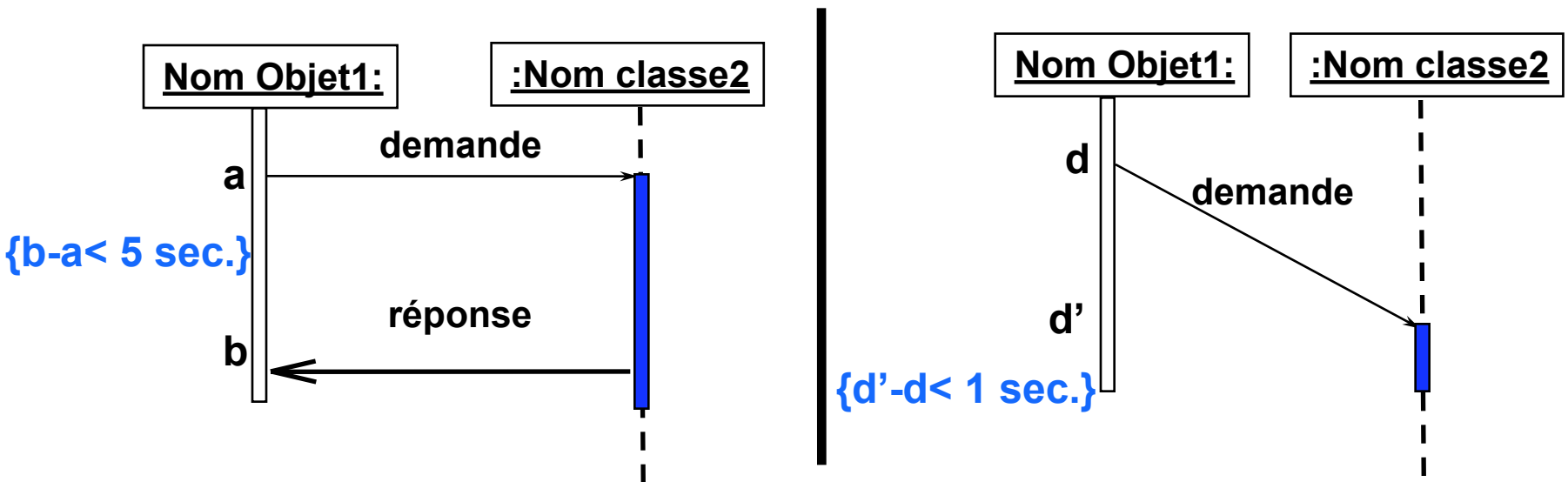
- Cas particuliers
 - Les messages entraînant la construction d'un objet
 - La récursion
 - Les destructions d'objets

Notations

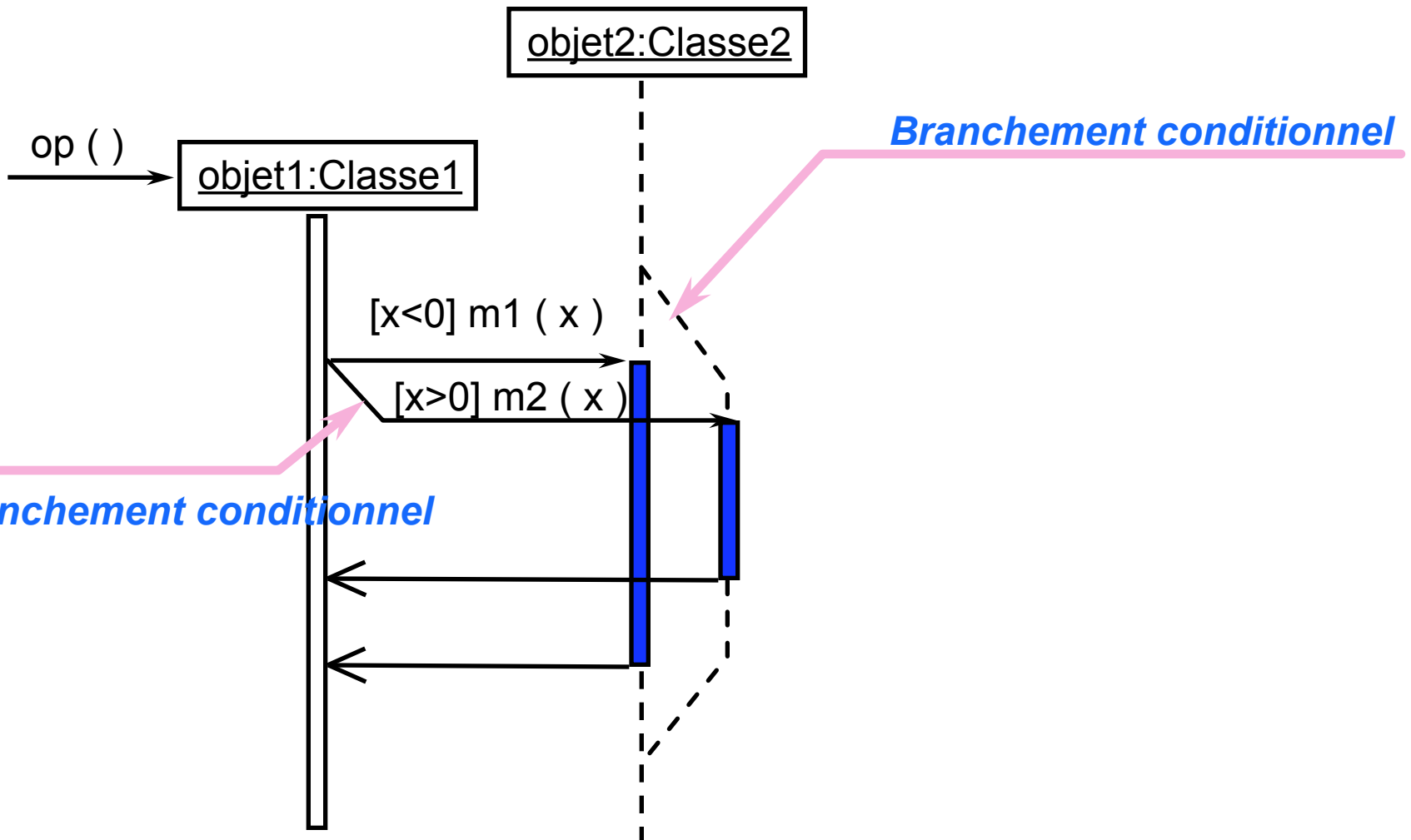


Aspects asynchrones et temps réel

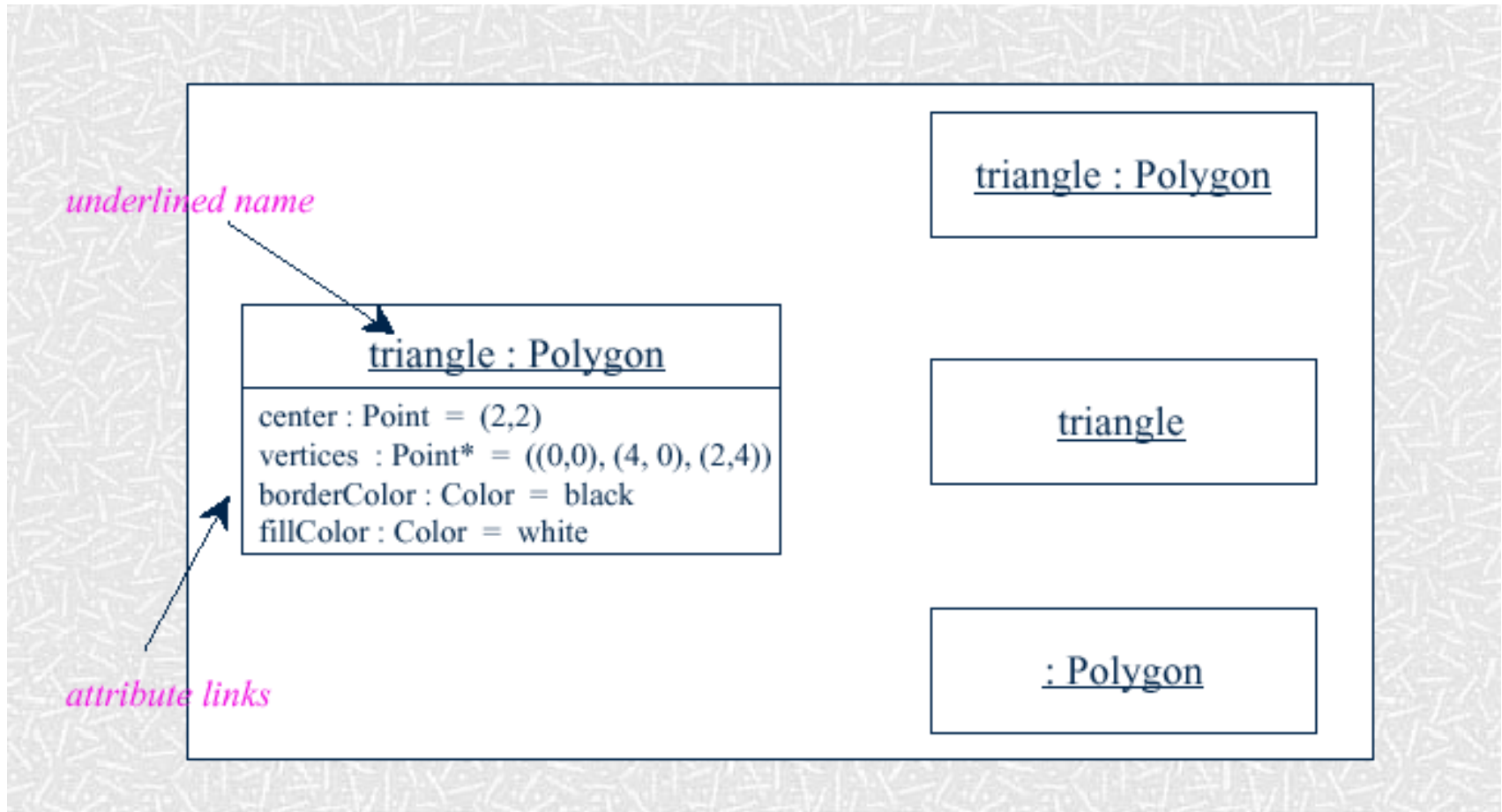
- Lecture du scénario et chronologie
 - Un scénario se lit de haut en bas dans le sens chronologique d'échange des messages.
 - Des contraintes temporelles peuvent être ajoutées au scénario



Représentation de conditionnelles



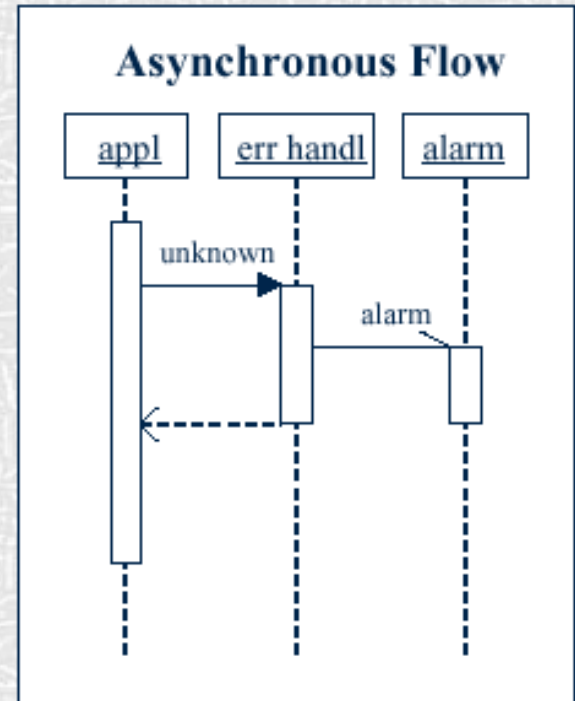
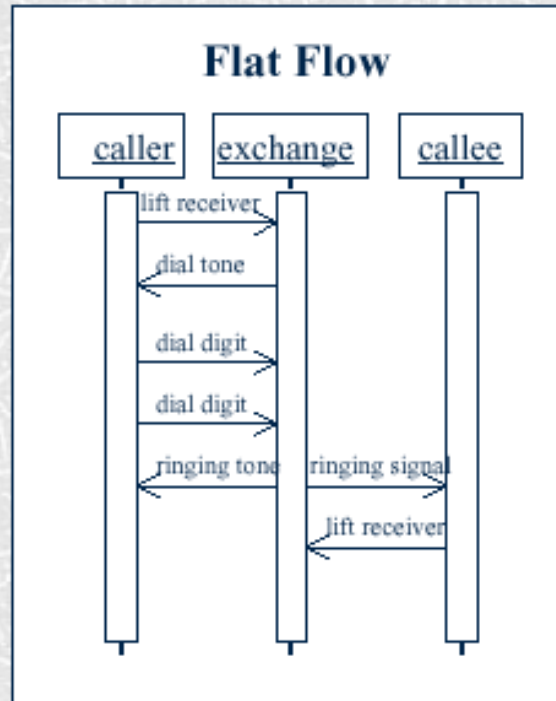
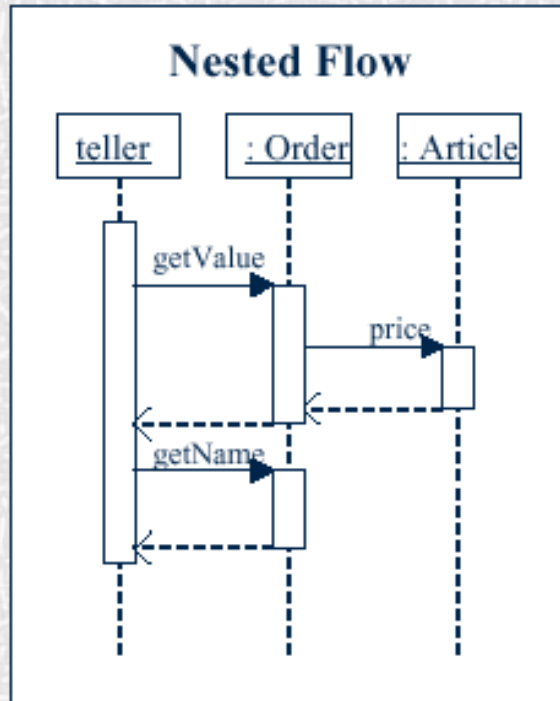
Exemple d'instances



Activations et envois de messages

- Chaque réception de message donne lieu à une durée d'activation : le temps de traitement du message
- La durée d'activation de l'émetteur recouvre celle du récepteur
- Type de messages :
 - flot de contrôle à plat :
 - message synchrone
 - message asynchrone
 - flot de contrôle emboîté ou appel de procédure (avec attente implicite du retour)
 - retour d'un appel de procédure, avec ou sans paramètre de retour


Des communications différentes




Interactions : Résumé (1)

Construct	Description	Syntax		
Instance (object, data value, component instance etc.)	An entity with a unique identity and to which a set of operations can be applied (signals be sent) and which has a state that stores the effects of the operations (the signals).	<table border="1"><tr><td data-bbox="1251 408 1435 468"><u>name</u></td></tr><tr><td data-bbox="1251 468 1435 528">attr values</td></tr></table>	<u>name</u>	attr values
<u>name</u>				
attr values				
Action	A specification of an executable statement. A few different kinds of actions are predefined, e.g. createAction, CallAction, DestroyAction, and UninterpretedAction.	textual		

Interactions : Résumé (2)

Construct	Description	Syntax		
Stimulus	A communication between two instances.			
Operation	A declaration of a service that can be requested from an instance to effect behavior.	textual		
Signal	A specification of an asynchronous stimulus communicated between instances.	<table border="1" data-bbox="1342 678 1526 821"> <tr> <td>«Signal» Name</td> </tr> <tr> <td>parameters</td> </tr> </table>	«Signal» Name	parameters
«Signal» Name				
parameters				

Construct	Description	Syntax
Link	A connection between instances.	
Attribute Link	A named slot in an instance, which holds the value of an attribute.	textual

Interactions : Résumé (3)

