GÉNIE LOGICIEL ORIENTÉ OBJET (GLO-2004)
ANALYSE ET CONCEPTION DES SYSTÈMES ORIENTÉS OBJETS (IFT-2007)

Automne 2016

Module 09 - Diagrammes d'interaction (diagrammes de séquence; diagrammes de communication)

Martin.Savoie@ift.ulaval.ca

B. ing, Chargé de cours, département d'informatique et de génie logiciel

UNIVERSITÉ LAVAL

Questions sur le projet?

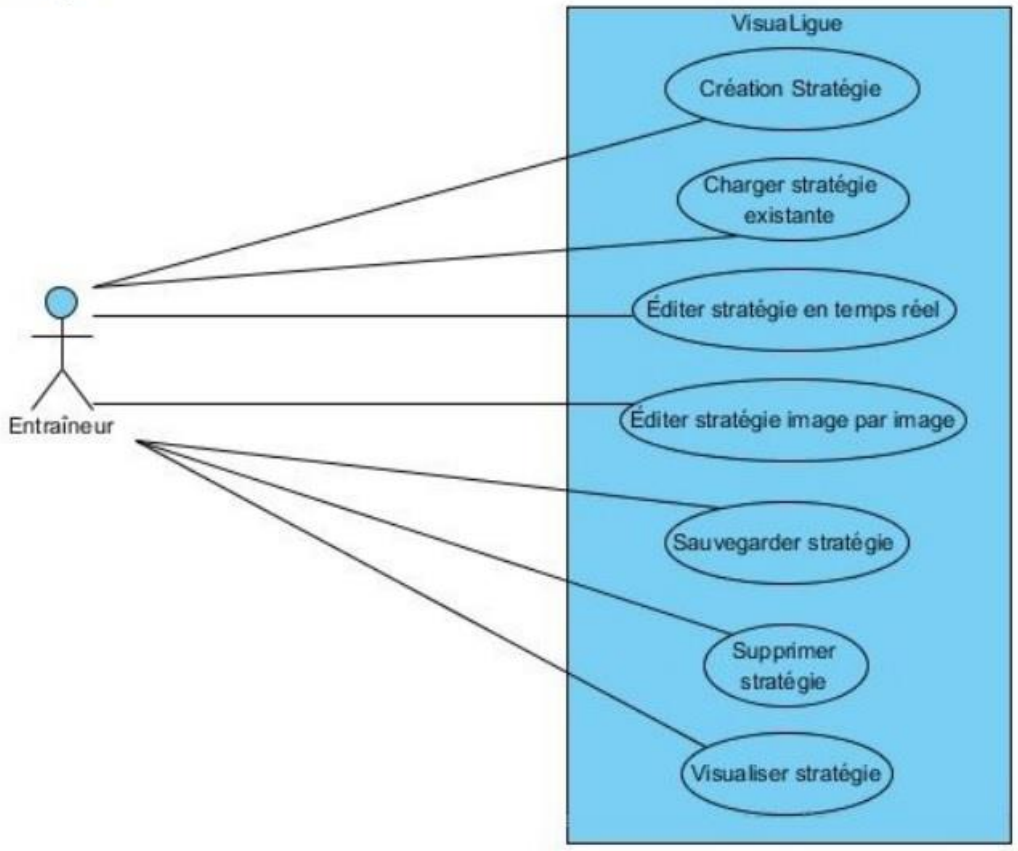
Retour sur livrable 1

- Passage a git?
- Ce qui s'en vient:
  - Rencontre de mi-session
  - Il faut commencer à coder si ce n'est pas déjà fait
  - Se préparer pour le livrable 2
  - présentation de diagramme de certaine équipe

Diagramme de cas d'utilisation

Diagramme de cas d'utilisation

Voici un diagramme représentant tous les cas d'utilisation possibles avec l'application VisuaLigue.



Texte des cas d'utilisation

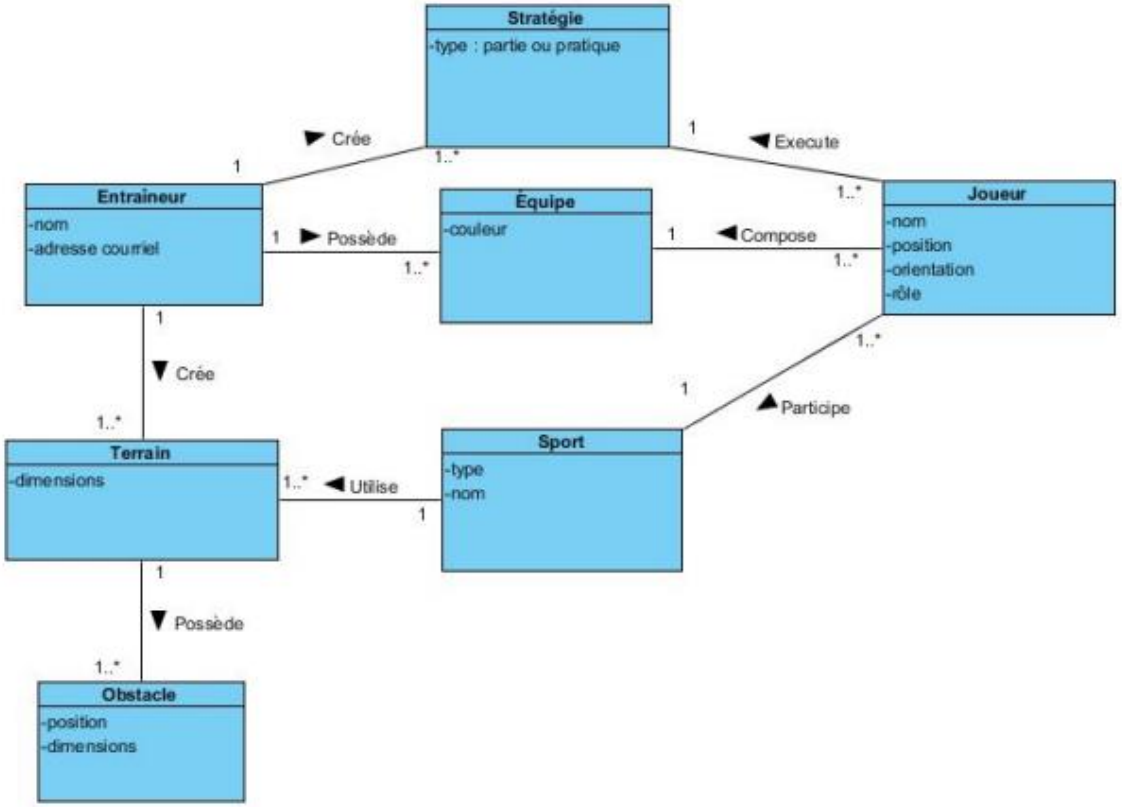
CU3

Cas d'utilisation	Charger une stratégie déjà existante
Système	VisuaLigue
Acteur	Entraîneur
Parties prenantes et intérêts	• L'entraîneur veut être capable de réutiliser ou de modifier des stratégies déjà créées et ce de façon efficace.
Préconditions	• L'application est ouverte. • Au moins une stratégie a été sauvegardée.
Garanties en cas de succès	La stratégie sauvegardée apparaîtra dans l'application. Elle pourra être modifiée selon le style d'édition sélectionné lors de sa création (image par image ou temps réel) ou pourra être visionnée si c'est le cas d'une stratégie chargée.
Scénario principal	
Utilisateur	**VisuaLigue Application**
1. L'entraîneur, sur l'interface de la création d'une stratégie, aura l'option de choisir une stratégie sauvegardée. Il clique sur le bouton à cet effet.	
	2. Le bouton « charger stratégie» de l'interface apparaîtra comme un bouton enfoncé. 3. Une fenêtre présentant le gestionnaire de fichiers de l'ordinateur apparaîtra après le clic du bouton « charger une ancienne stratégie ».
4. L'entraîneur doit choisir la stratégie à charger en double-cliquant sur le fichier voulu ou en sélectionnant le fichier voulu et en cliquant sur le bouton ouvrir.	
	5. Après le double-clic sur le fichier de la stratégie ou la sélection du fichier et l'appui du bouton ouvrir, la fenêtre du gestionnaire de fichiers se referme et l'application passe à l'interface d'édition de la stratégie (image par image ou temps réel).
Scénario alternatif	
Utilisateur	**VisuaLigue Application**
Ligne 4: L'entraîneur ne trouve aucune partie sauvegardée ou ne trouve pas la stratégie désirée. Il clique sur le bouton annuler.	
	Ligne 5: Dans la fenêtre du gestionnaire de fichiers, le bouton annuler apparaîtra comme enfoncé.
Variantes de données	Les stratégies auront tous le même format de fichier.

Modèle du domaine

Modèle du domaine

Voici notre modèle du domaine, qui représente bien les liens et les interactions entre les différentes entités représentant notre application.

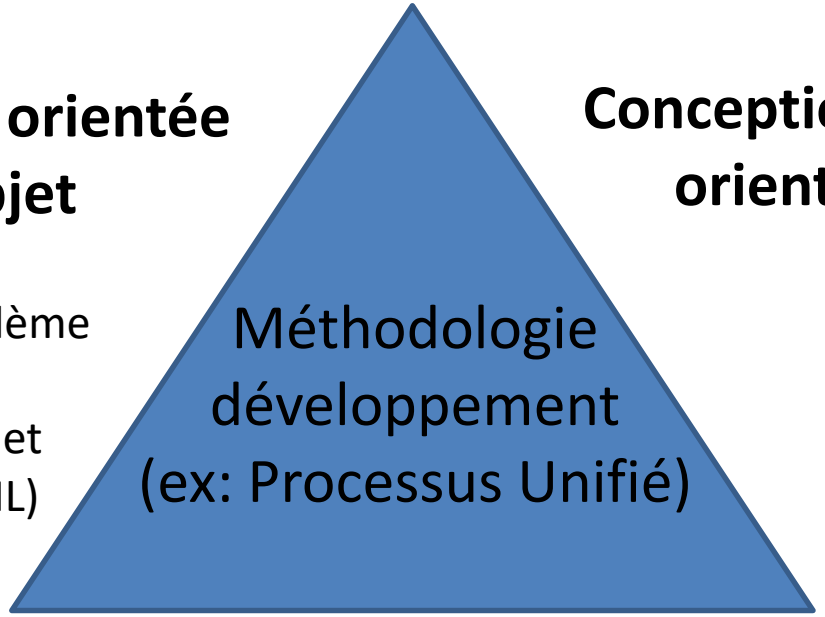


Livrable 2

Génie logiciel orienté objet

Analyse orientée objet

- Comprendre le problème
- Décrire la situation à l'aide de documents et diagrammes (ex: UML)



Conception (design) orienté objet

- Concevoir une solution informatique
- Tracer des plans (plus ou moins détaillés) sous la forme de documents et diagrammes (ex: UML)

Programmation orientée objet

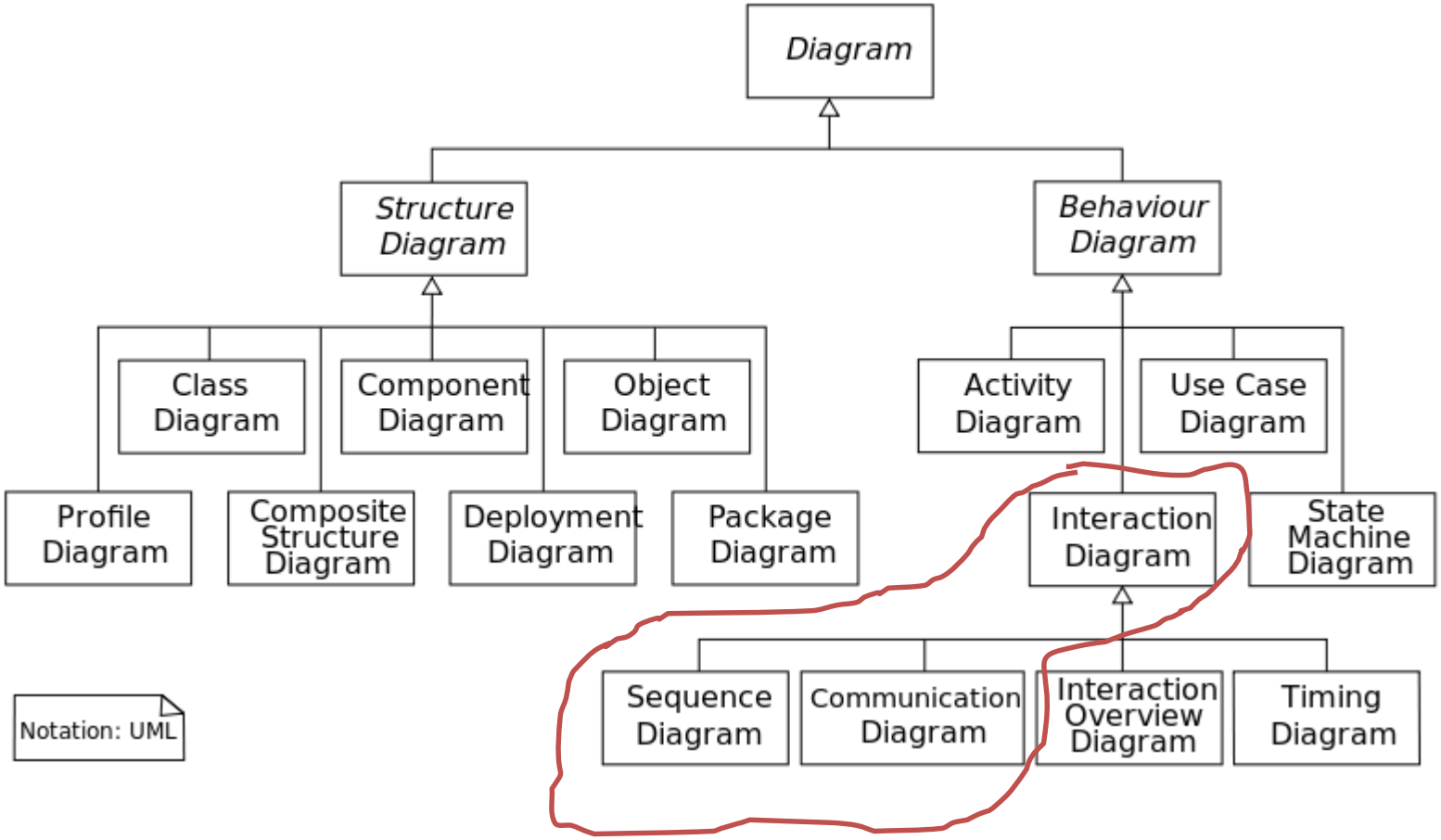
Mettre en œuvre la solution à l'aide d'un langage (ex: Java)

UNIVERSITÉ LAVAL

Introduction

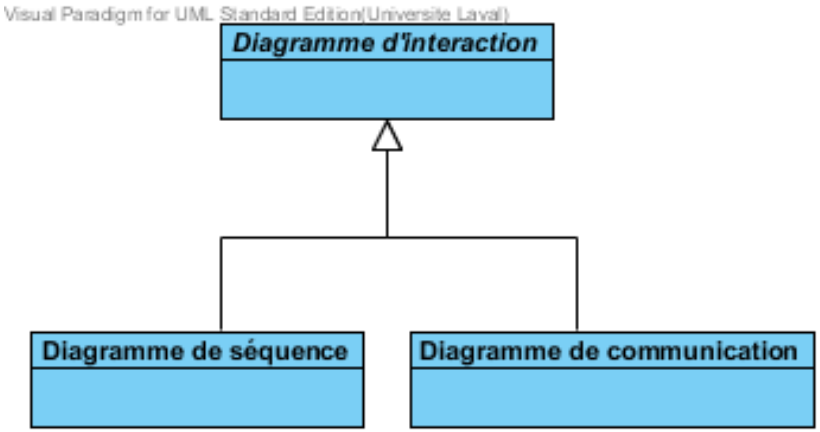
- Le diagramme de classes montre la structure **statique** du système en terme de classes/objets et de relations entre elles.
- Cependant, cette description ne montre pas comment ces classes/objets interagissent ensemble pour réaliser des tâches et fournir les fonctionnalités du système
- Une étude **dynamique** montre comment les objets interagissent dynamiquement à différents moments durant l'exécution du système

UML : 14 types de diagrammes (ouf!)



Diagrammes d'interaction

- Servent à montrer comment les **objets** d 'un système **collaborent** pour réaliser une **fonctionnalité** du système.
- Montre comment les **objets** s'échangent des **messages** (i.e. comment un objet, **le client**, invoquent une **opération** sur un autre objet, **le serveur**). Ces échanges sont appelés **interactions**.



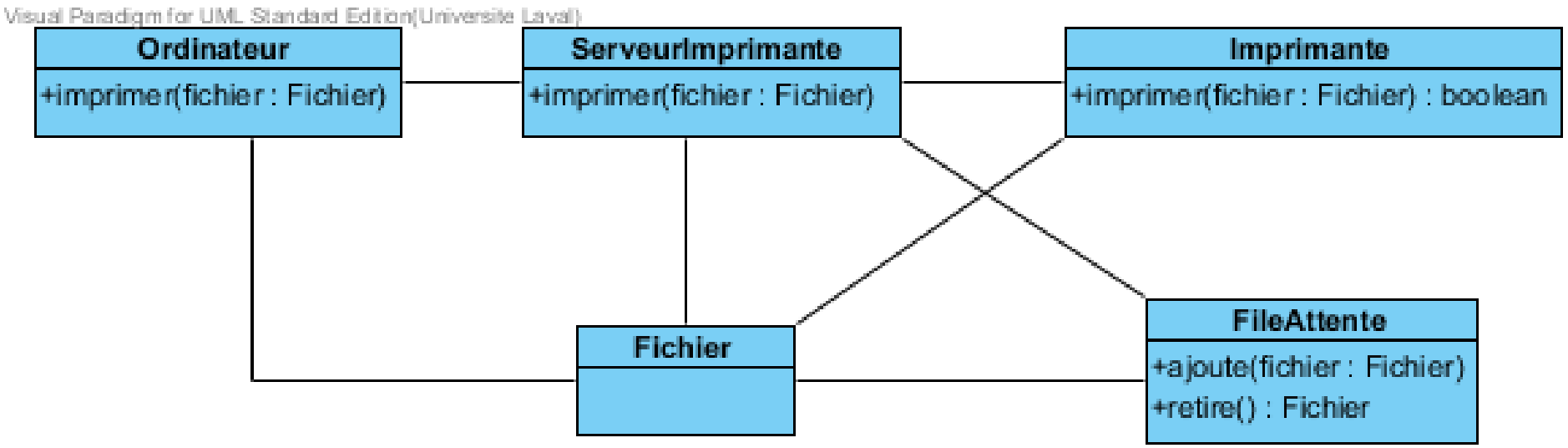
Quelqu'un peut nous dire pourquoi le mot «diagramme d'interaction» est en italique, à droite?

UNIVERSITÉ LAVAL

Les deux types de diagrammes d'interaction

- Le **diagramme de séquences** met l'accent sur l'aspect temporel
- Le **diagramme de communication** met l'accent sur les relations entre les objets
- Ils présentent sensiblement la même information

Diagramme de classes de l'exemple



Des questions subsistent. Comment fonctionnent ces classes? Comment accomplissent-elles quelque chose?

Diagramme de communication

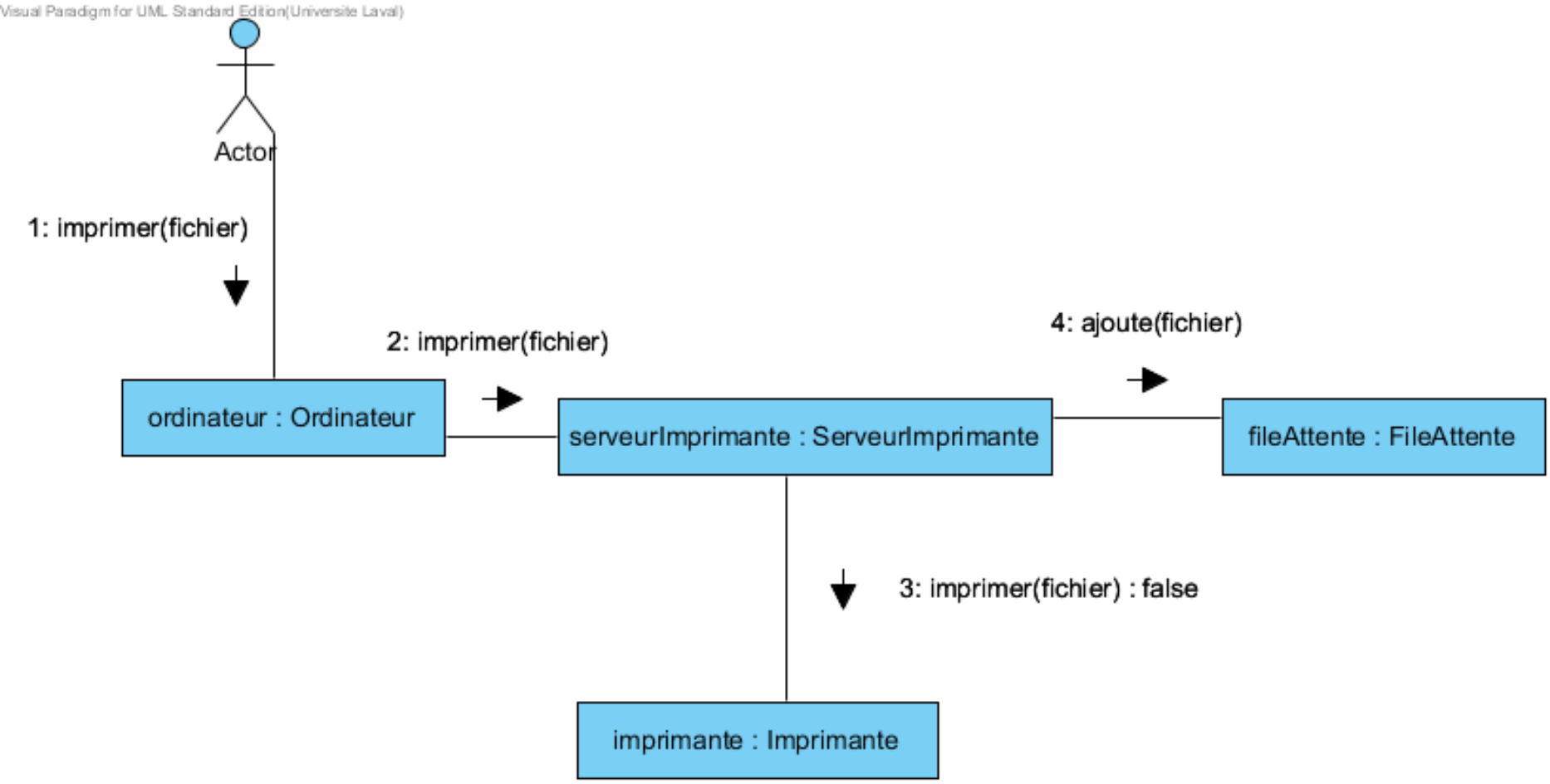


Diagramme de séquence

- La même information, mais représentée sous une forme différente à l'aide d'un diagramme de séquence:

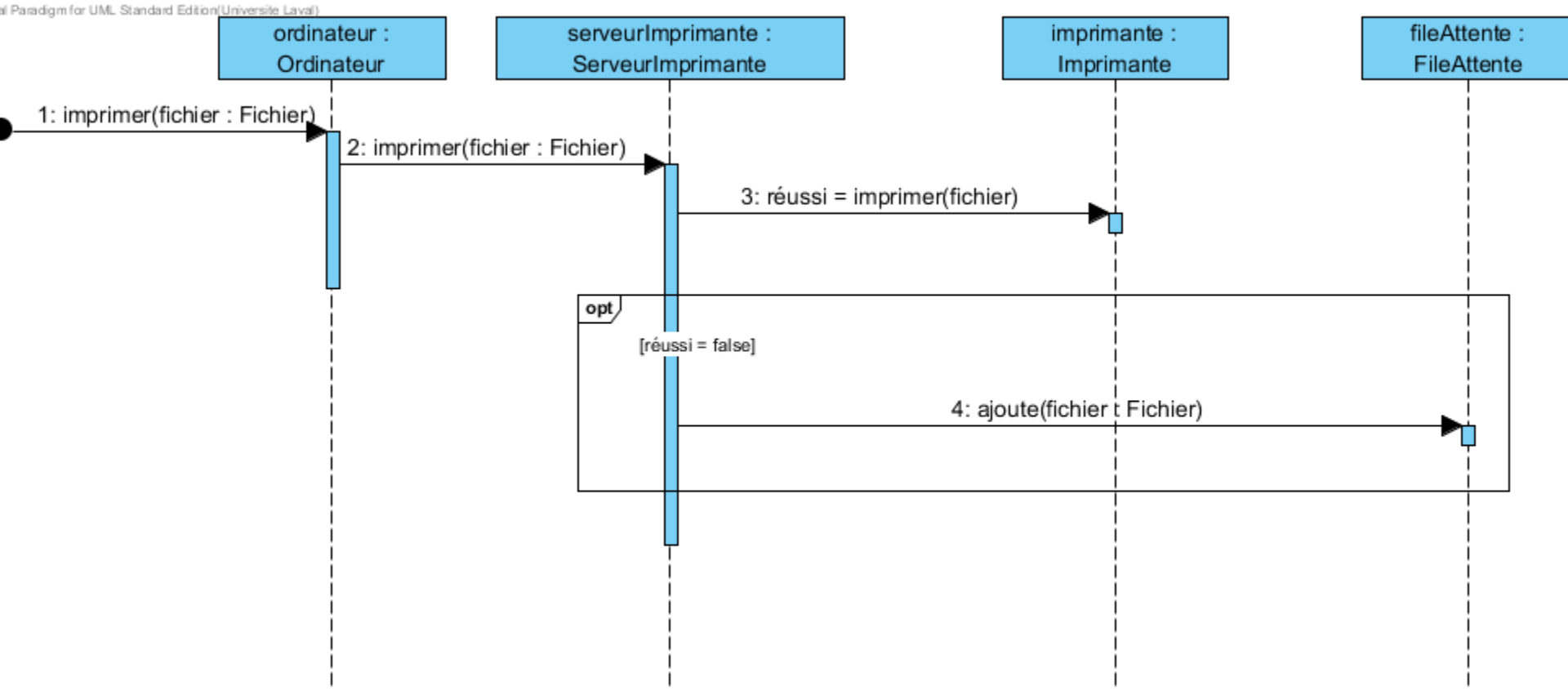


Diagramme de séquence

Les diagrammes de *séquence**

- Ils se concentrent sur la ***séquence des messages envoyés entre les objets*** (c'est-à-dire ***comment*** et ***quand*** les messages sont envoyés et reçus entre les objets)
- Ils possèdent ***deux axes***: l'***axe vertical*** montre le ***temps*** tandis que l'***axe horizontal*** montre l 'ensemble des ***objets en interaction*** dans un ***scénario*** ou une ***scène*** spécifique d'un scénario.

Diagramme de séquence VS Diagramme de séquence système*

- Tel que proposé par Larman, les diagrammes de séquence sont utilisés de deux manières différentes :

1. Diagramme de séquence système (DSS)

- Documentation d'un scénario d'un use-case
- Se concentre sur la description de l'interaction, souvent dans des termes proches de l'utilisateur et sans entrer dans les détails de la synchronisation
- L'inscription portée sur une flèche correspond à un événement (ou une action) qui survient dans le domaine de l'application
- Les flèches ne traduisent pas à ce niveau des envois de messages au sens des langages de programmation.

3.10.1. DÉTAILS DU CAS D'UTILISATION : SUPPRIMER UN COMPOSANT

Cas d'utilisation	Supprimer un composant
Système	VisualElectrique
Acteurs	Utilisateur
Parties prenantes et intérêts	Utilisateur : il veut pouvoir corriger son plan en retirant des composants déjà le schéma
Préconditions	Avoir placé au moins un composant sur le plan en cours
Garanties en cas de succès	Le composant sélectionné est retiré du plan
Scénario principal	1. L'utilisateur sélectionne le composant à supprimer; 2. L'utilisateur demande la suppression du composant; 3. Le système retire le composant sélectionné du plan.
Scénarios alternatifs	3a. Le composant retiré faisait partie d'un circuit : 1. Le système revérifie la validité du circuit modifié et affiche le code de correspondant. 3b. Le composant retiré était une boîte électrique : 1. Le système active la possibilité de sélectionner une boîte électrique p ajouts de composants futurs.

3.10.2. DIAGRAMME DE SEQUENCE SYSTEME : SUPPRIMER UN COMPOSANT

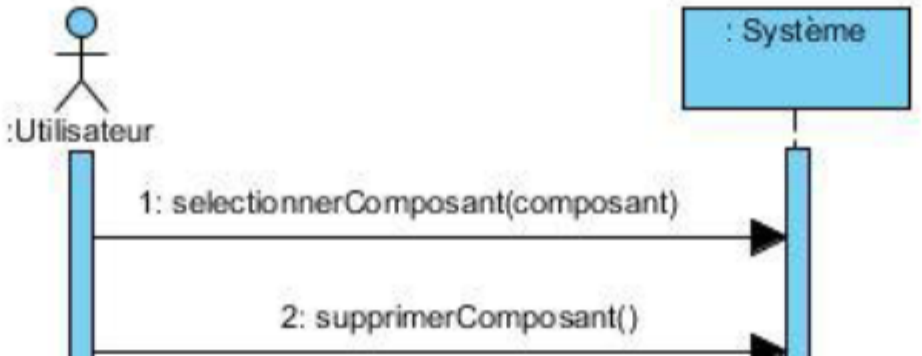


Diagramme de séquence VS Diagramme de séquence système

- Tel que proposé par Larman, les diagrammes de séquence sont utilisés de deux manières différentes (suite …) :

2. Diagramme de séquence (de conception)

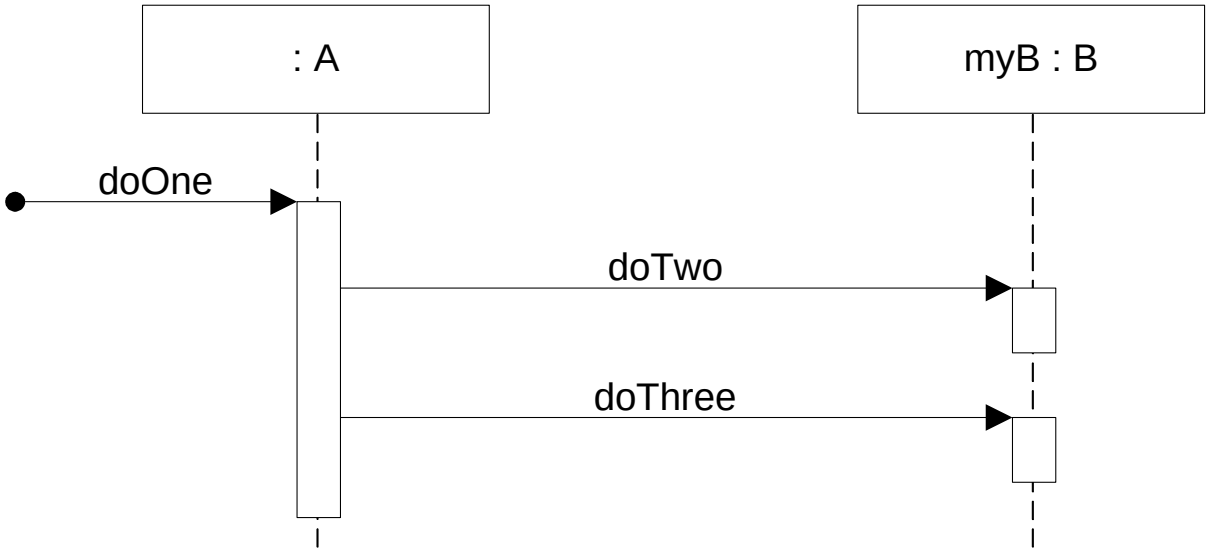
- Correspond à un usage plus « informatique »
- Permet la représentation précise des interactions entre objets, autrement dit le séquencement des flots de contrôle
- Exemple : Représentation des structures de contrôle (IF THEN ELSE) et de la structure de la boucle WHILE

Discipline	Artifact	Incep.	Elab.	Const.	Trans.
	Iteration→	I1	E1..En	C1..Cn	T1..T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	
Implementation	Implementation Model (code, html, ...)		s	r	r

S: start
R: refine

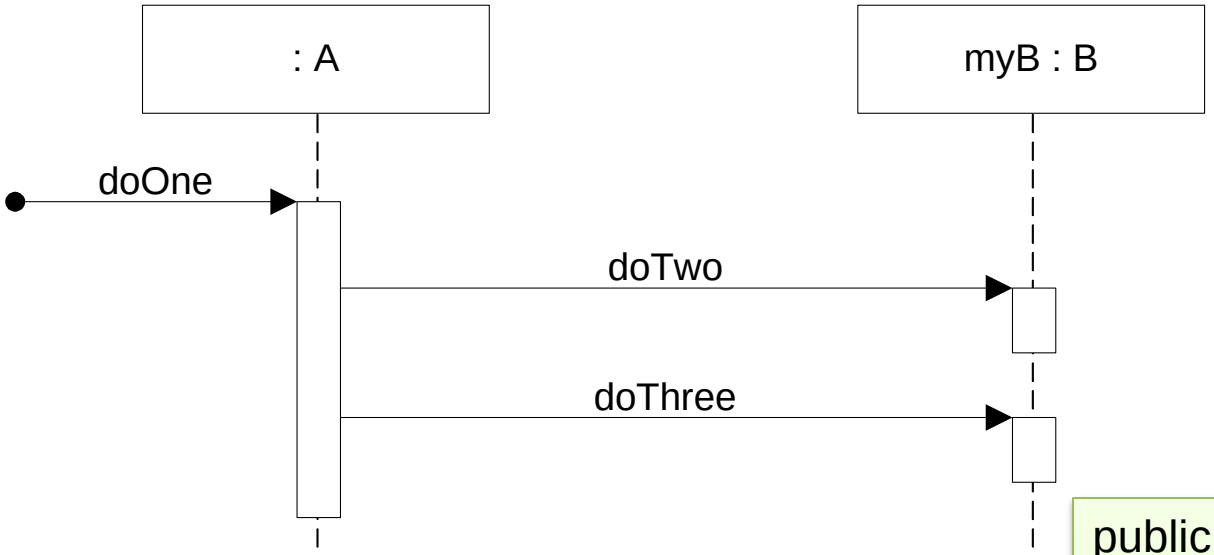
	Activités (appelées *disciplines* dans le Processus Unifié)	**Modèles** et artefacts générés
Analyse	Modélisation domaine d'affaires / Business modeling / Modélisation métier	**Modèle du domaine**: (1) diagramme de classe « conceptuel », (2) parfois un diagramme d'activités
	Analyse des besoins / Exigences / Requirements	(3) Énoncé de vision
		Modèle de cas d'utilisation / Use-case model : (4) diagramme des cas d'utilisation, (5) texte des cas d'utilisation, (6) diagramme de séquence système
		(7) Spécifications supplémentaires
		(8) Glossaire
	Design / Conception	**Modèle de conception / Design model** : (9) diagrammes de classes, (10) diagrammes d'interaction, (11) tout autre diagramme UML pertinent selon le contexte
	Implémentation	(12) Code
	…	

Correspondance diagramme-code



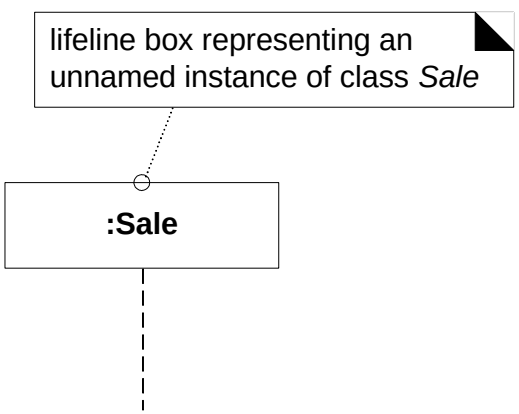
Êtes-vous en mesure d'imaginer le code (Java, C++ ou autre) correspondant à ce diagramme?

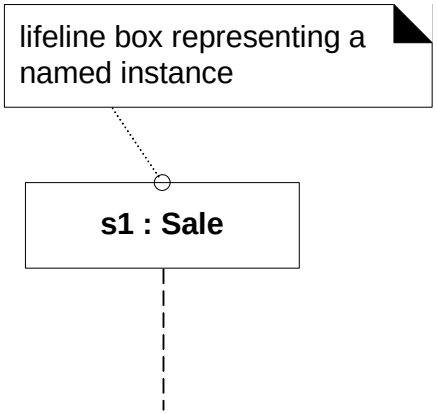
Correspondance diagramme-code



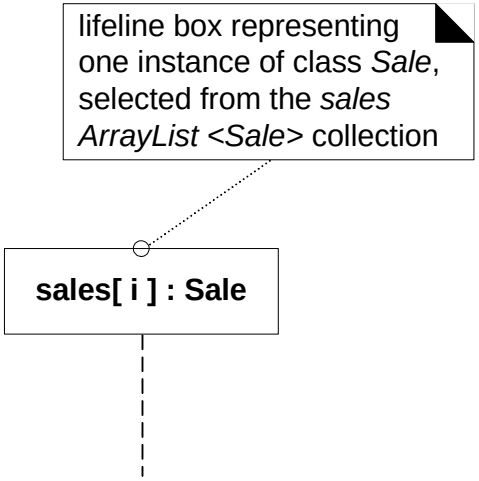

```
public class A
{
        Public void doOne()
        {
                myB.doTwo();
                myB.doThree();
        }
}
```

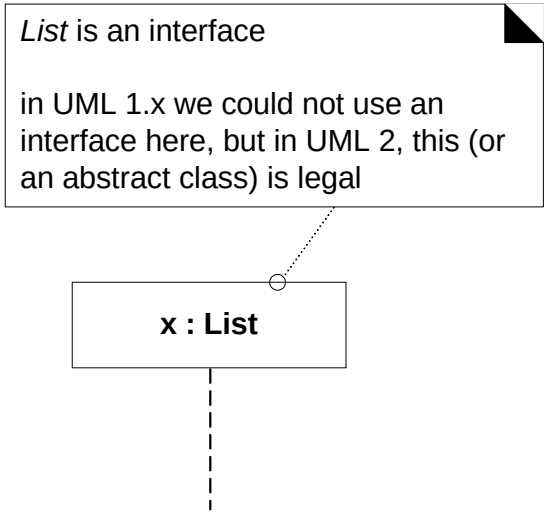
Représentation des objets



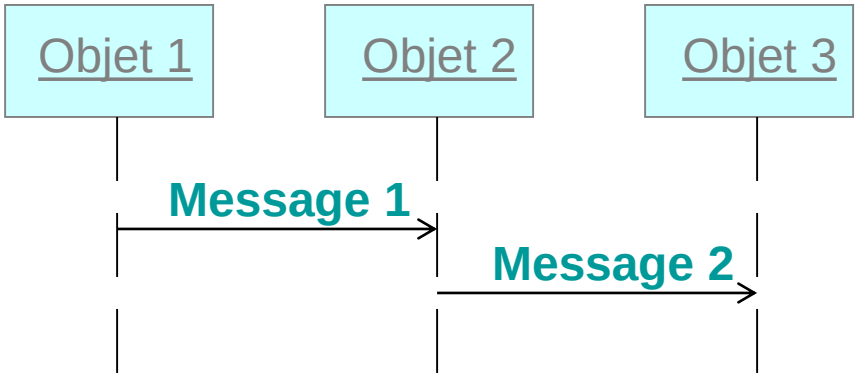


Sachez distinguer le nom de l'objet et le nom de la classe



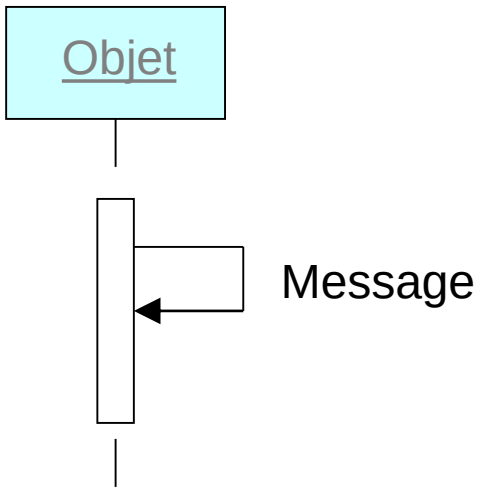


Représentation des interactions*



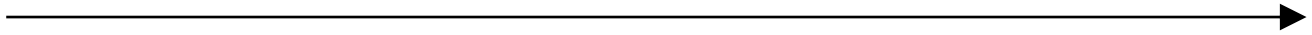
Message 1 précède dans le temps Message 2

Un objet peut appeler ses propres méthodes



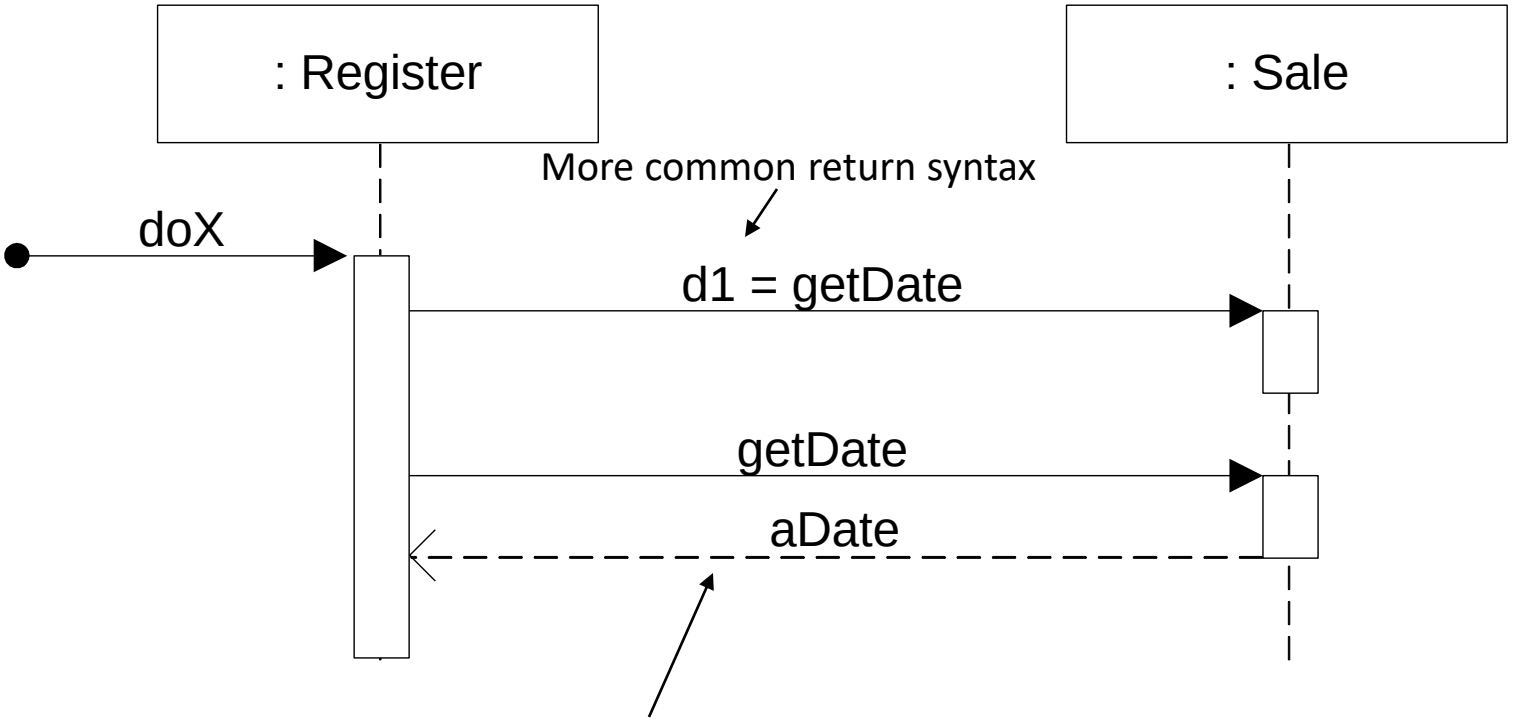
Spécification d'un appel/message

return = msgName(param:paramType1, …) : returnType



Heureusement, nous ne sommes pas obligés de toujours utiliser la syntaxe complète (!)

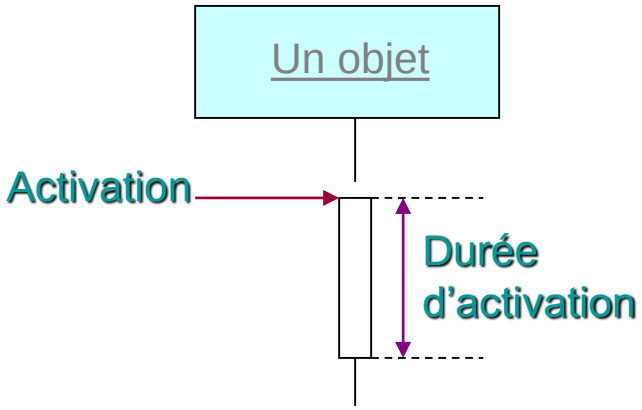
Notation simplifiée



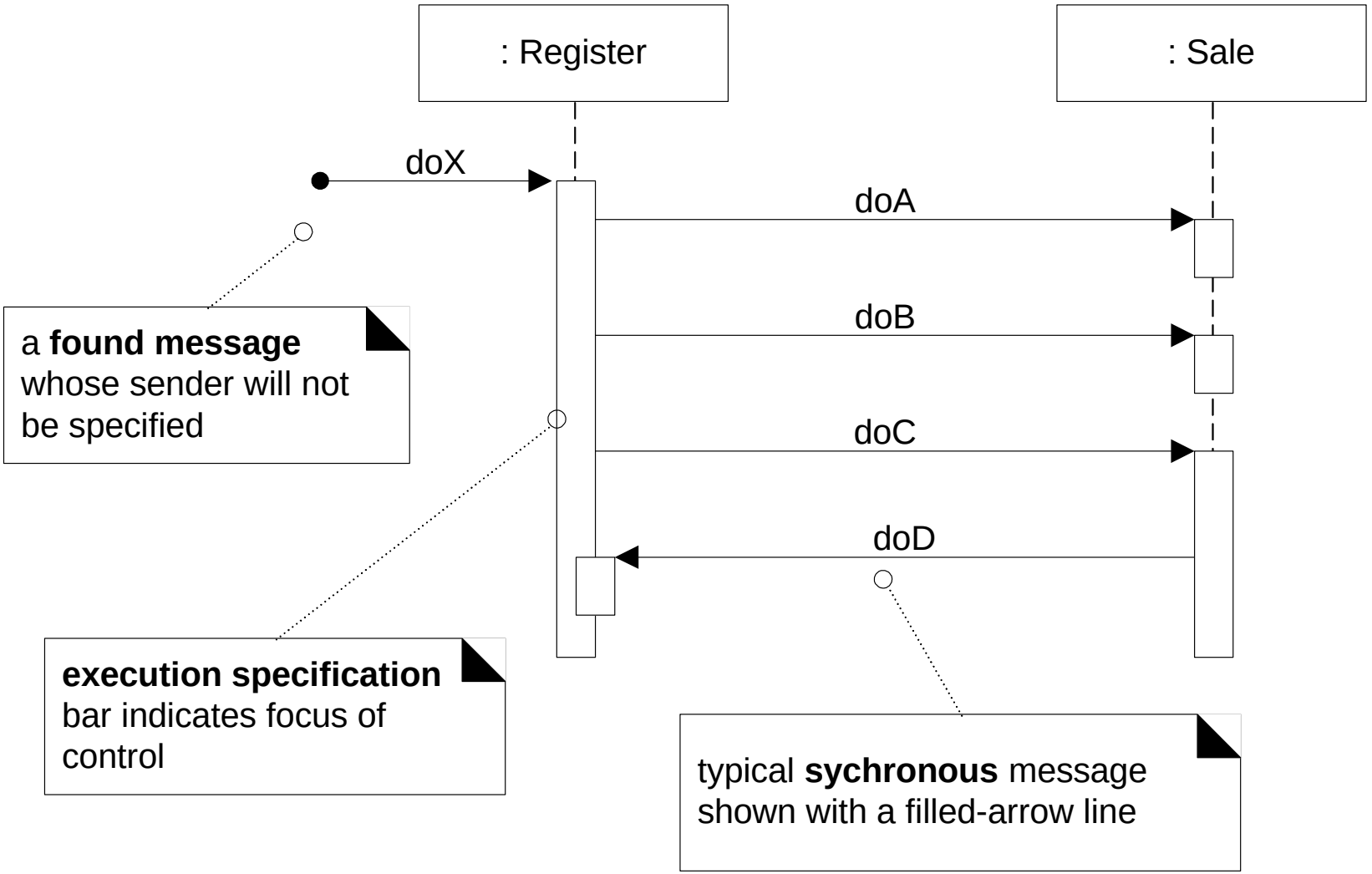
Barre de spécification d'exécution / d'activation

- Quand un objet reçoit un message, on dit qu'il entre dans sa phase d'activation. Un objet dans cette phase peut:
  - exécuter son propre code
  - attendre les résultats retournés par un autre objet à qui il a envoyé un message

- L 'activation d'un objet est représentée par un rectangle étroit sur la ligne de vie de l 'objet. Ce rectangle est appelée «barre de spécification d'exécution / d'activation»

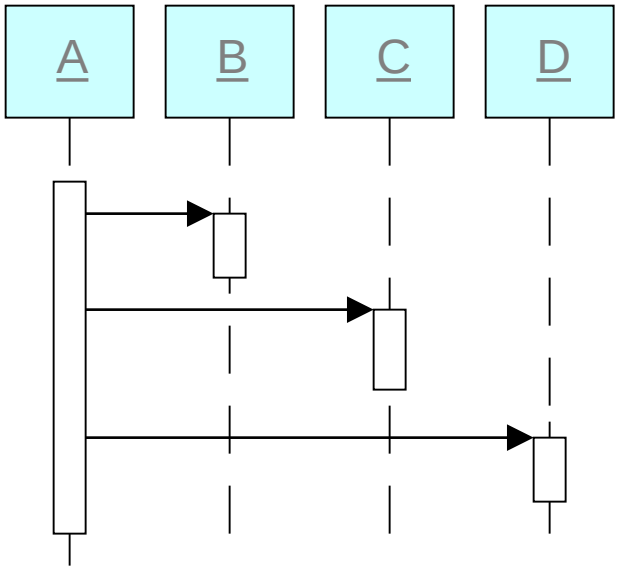


Barre de spécification d'exécution / activation

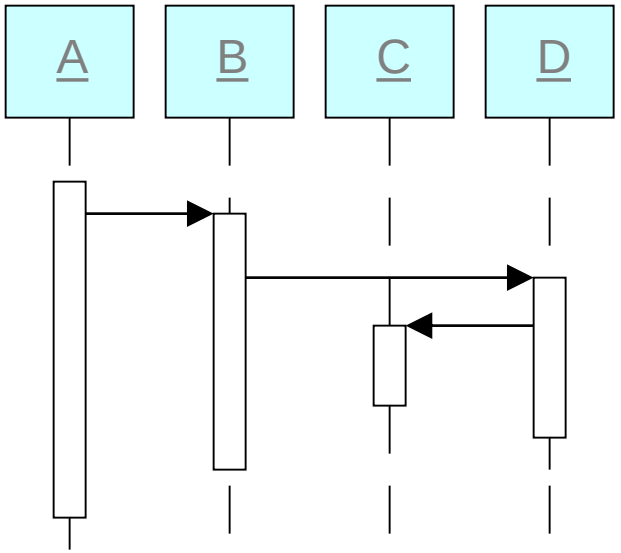


Structures de contrôle

- Les diagrammes de séquences reflètent la structure de contrôle:

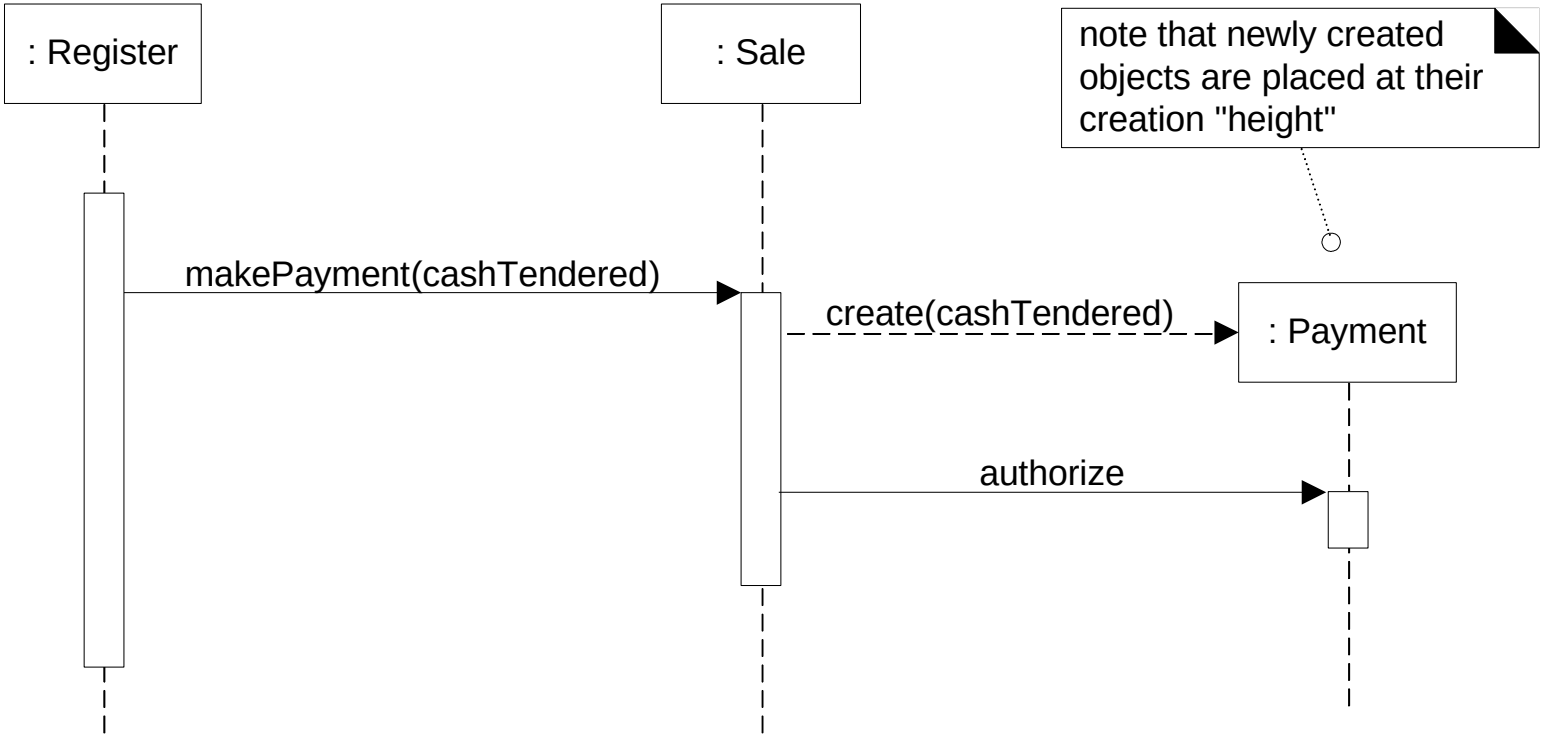


Contrôle centralisé



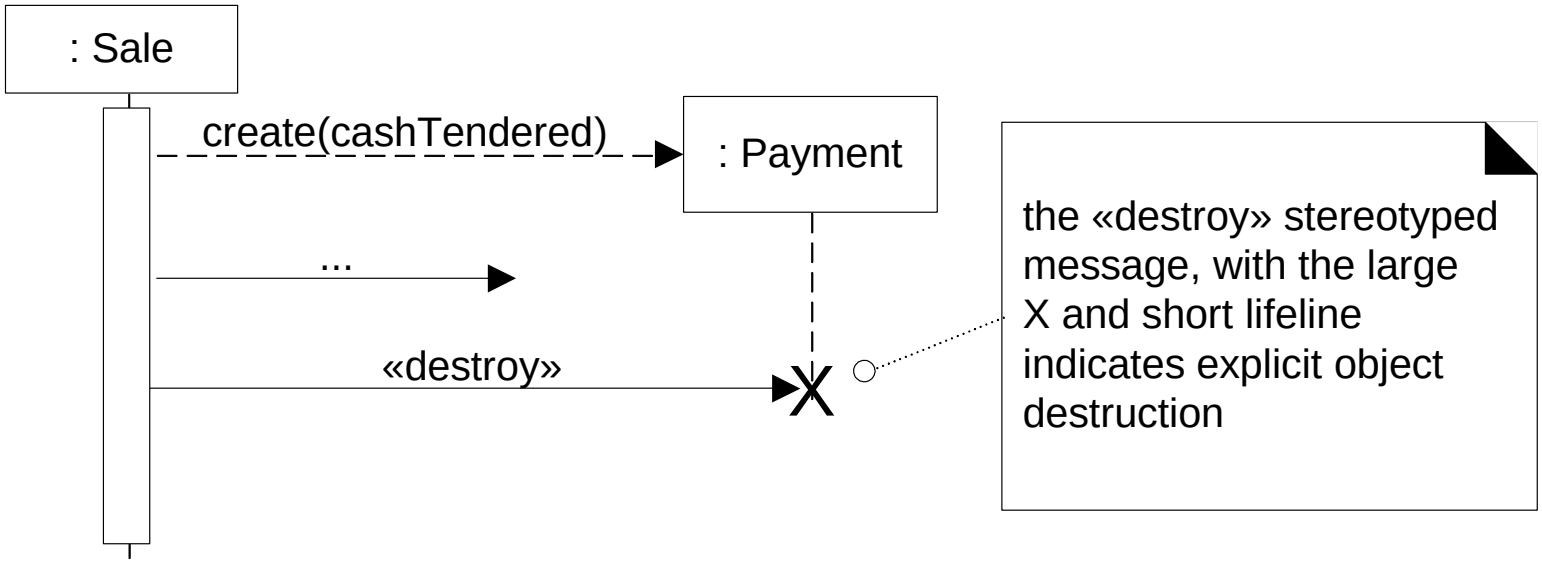
Contrôle décentralisé

Création d'un objet



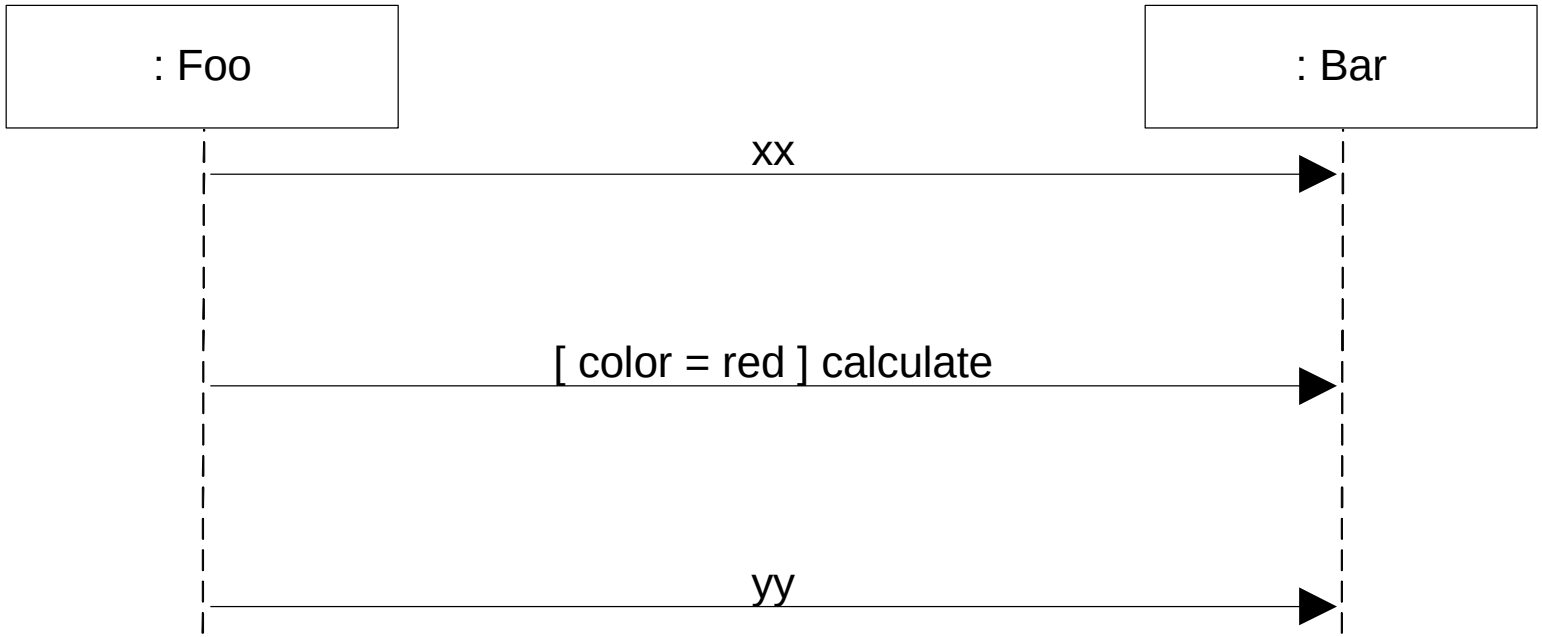
Le mot clé 'create' fait partie du standard UML
En pratique les gens utilisent également le nom du constructeur
L'utilisation du pointillé est préconisée par UML mais en pratique c'est peu utilisé

Destruction d'un objet

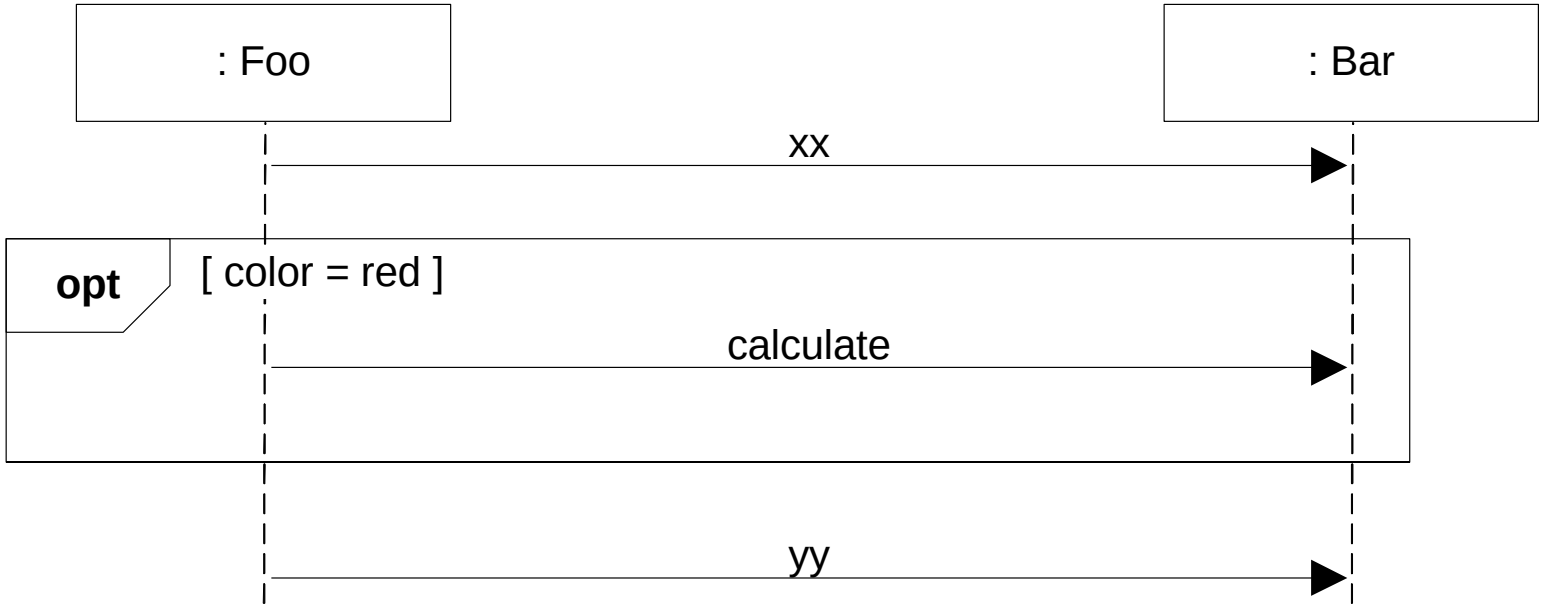


Note: la destruction d'un objet n'est pas nécessaire avec un langage de programmation disposant d'un ramasse-miette (ex: Java)

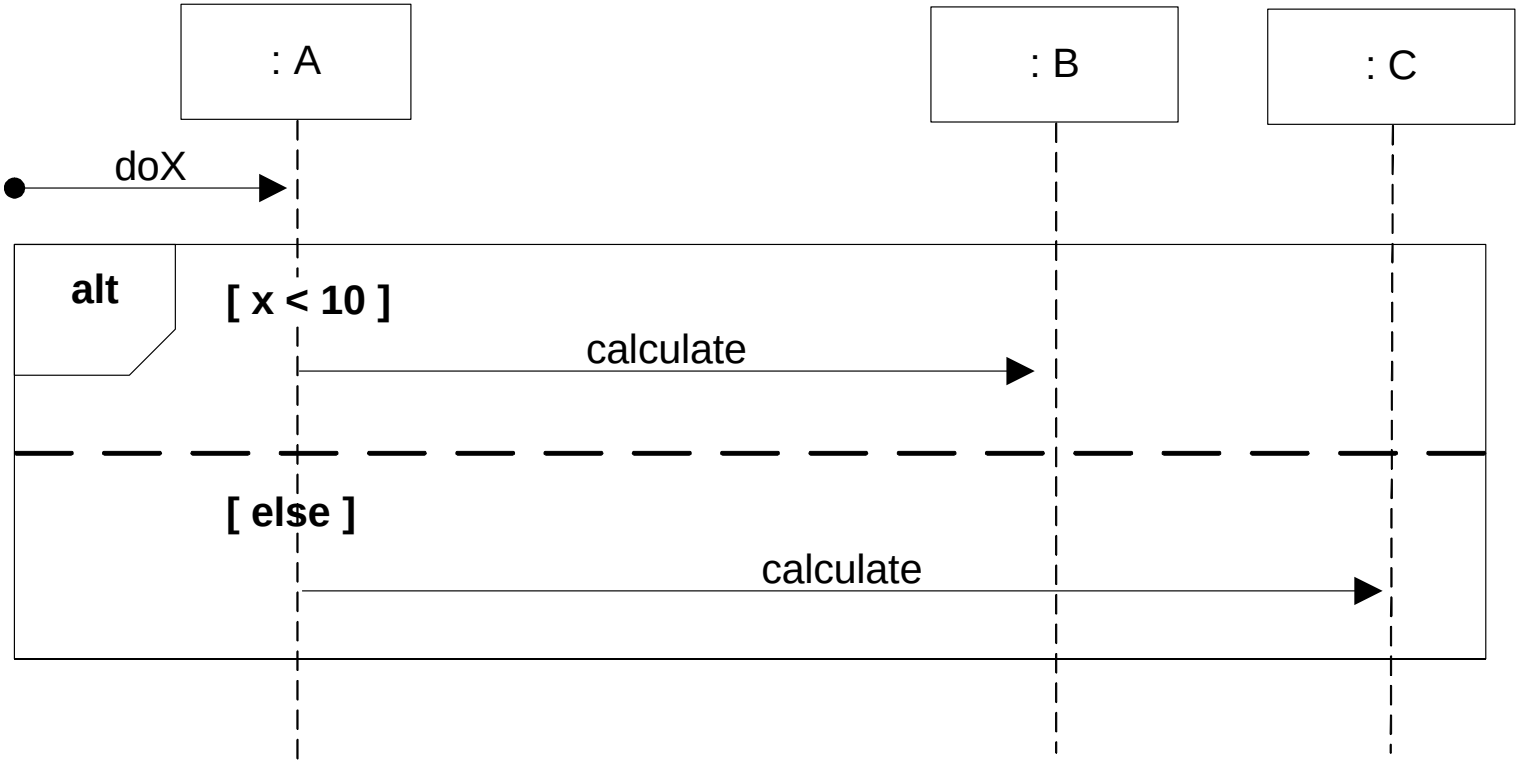
Énoncé conditionnel (if)



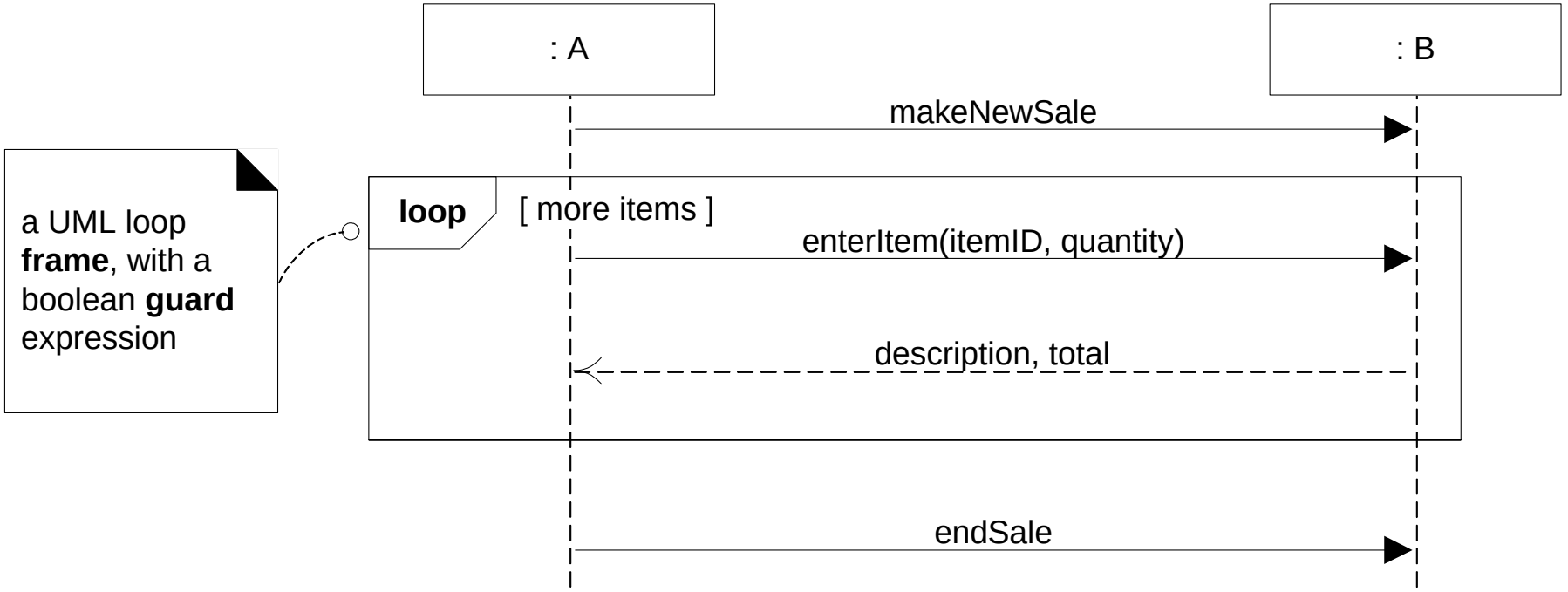
Groupe d’énoncés conditionnels (if)



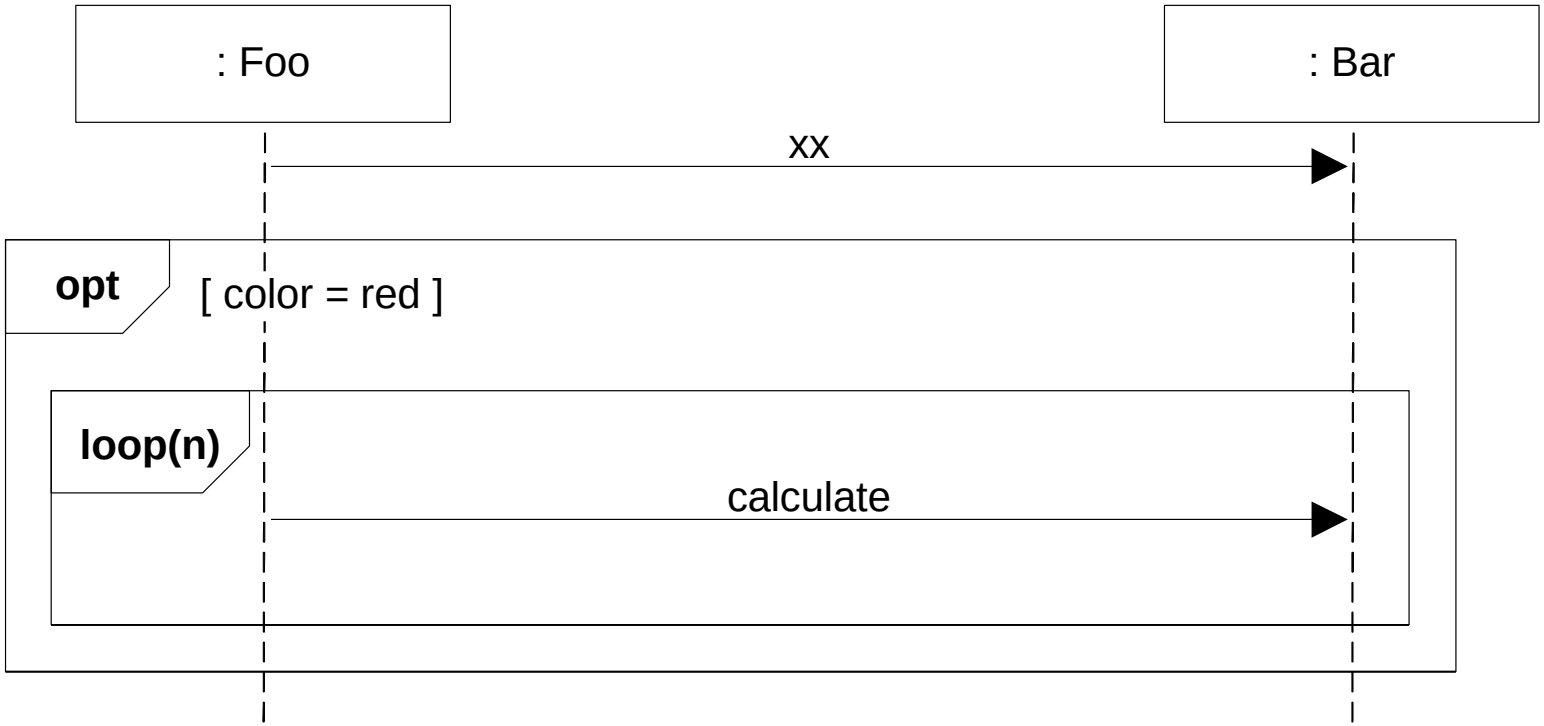
Alternative (if then else)



Boucle

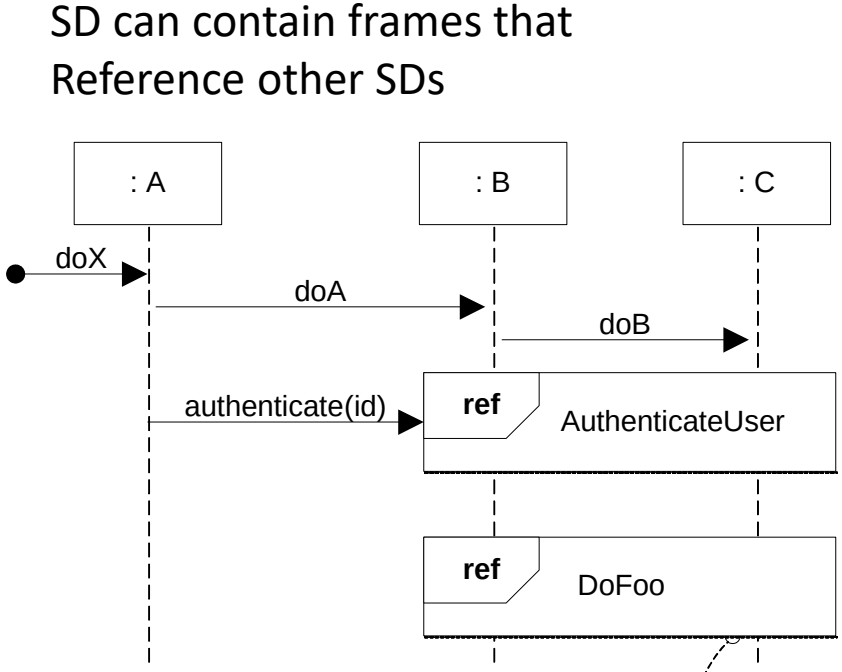

: A
: B
makeNewSale
loop
[more items]
enterItem(itemID, quantity)
description, total
endSale
a UML loop **frame**, with a boolean **guard** expression

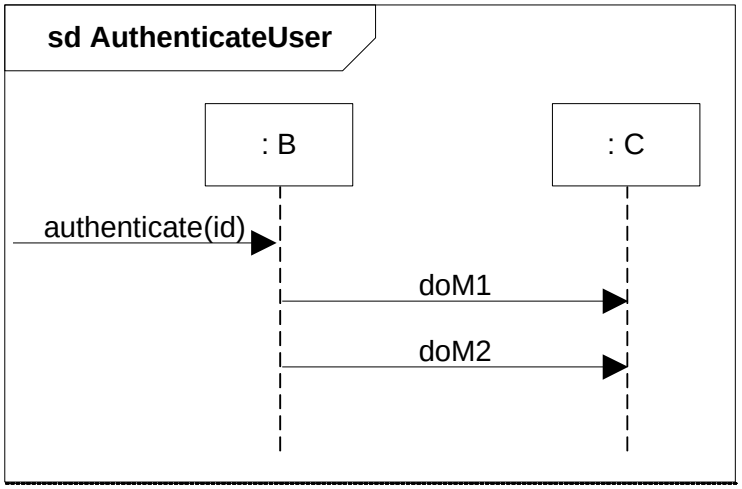
Boucle imbriquée dans un énoncé optionnel

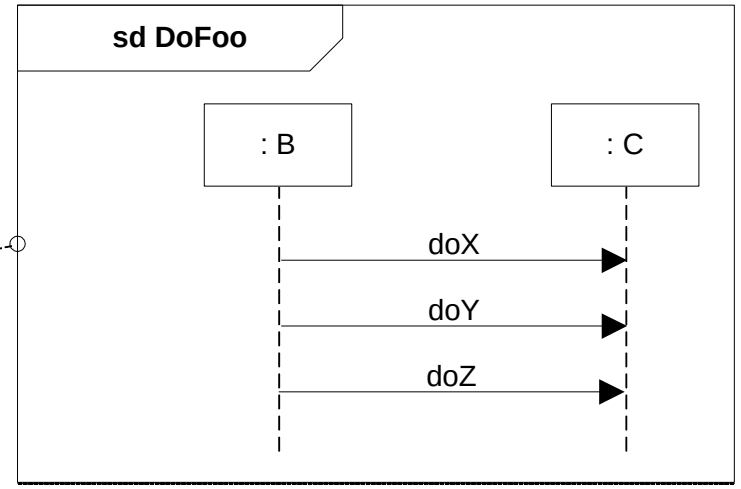


Imbrication des diagrammes de séquence

SD can contain frames that
Reference other SDs







interaction occurrence

note it covers a set of lifelines

note that the sd frame it relates to has the same lifelines: B and C

Les diagrammes de ***séquence*** peuvent être utilisés sous deux ***formes***:

- ***forme générique***: elle décrit ***tous*** les déroulements possibles et peut contenir des ***branchements***, des ***conditions***, et des ***boucles***.

- ***forme d'instanciation***: elle décrit le comportement du système pour une ***situation spécifique*** et ne peut donc contenir ***aucun branchement*** et aucune ***condition*** ou ***boucle***.

Diagrammes de communication

- Attention! En UML 1 ce type de diagramme était nommé diagramme de collaboration

Diagramme de communication

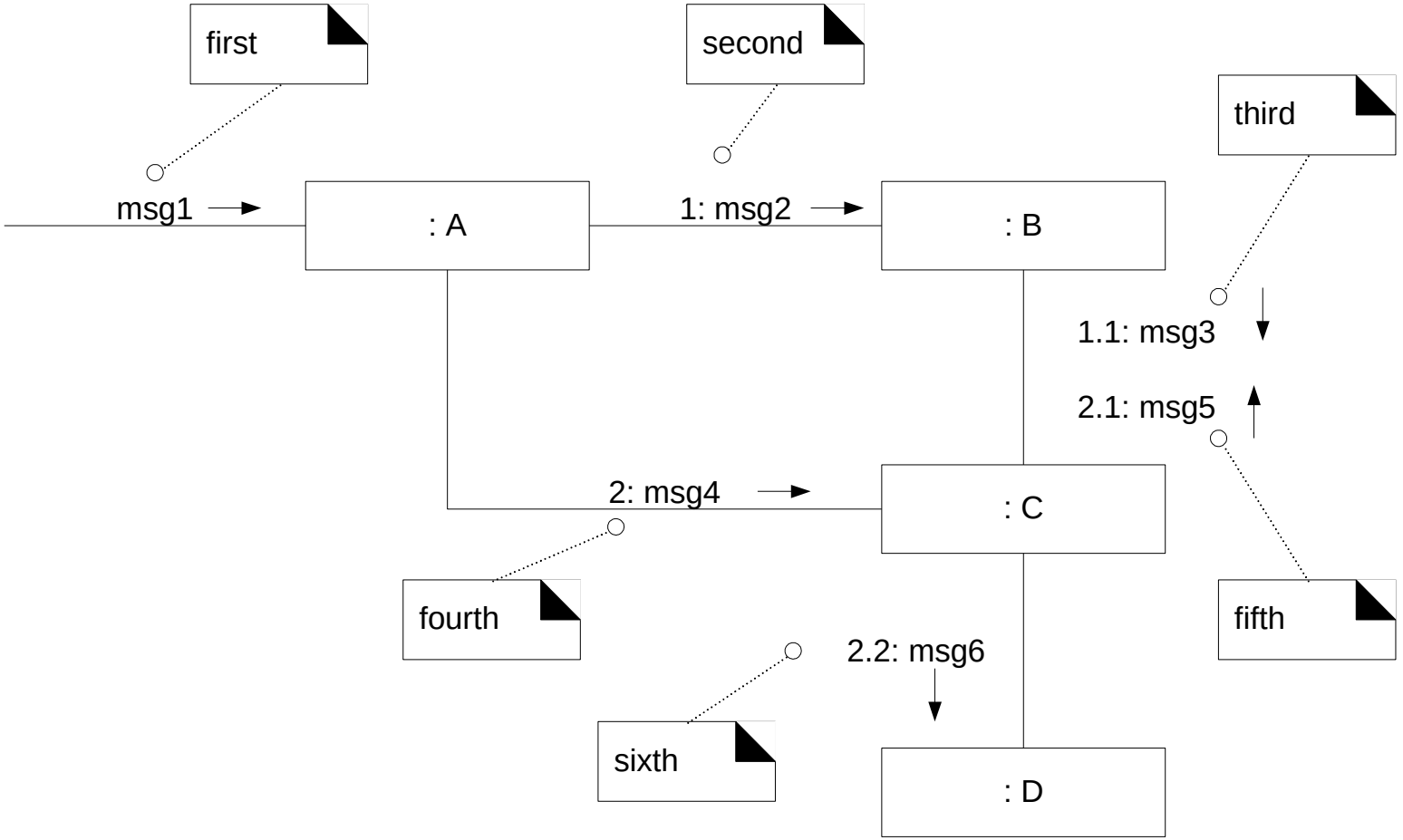
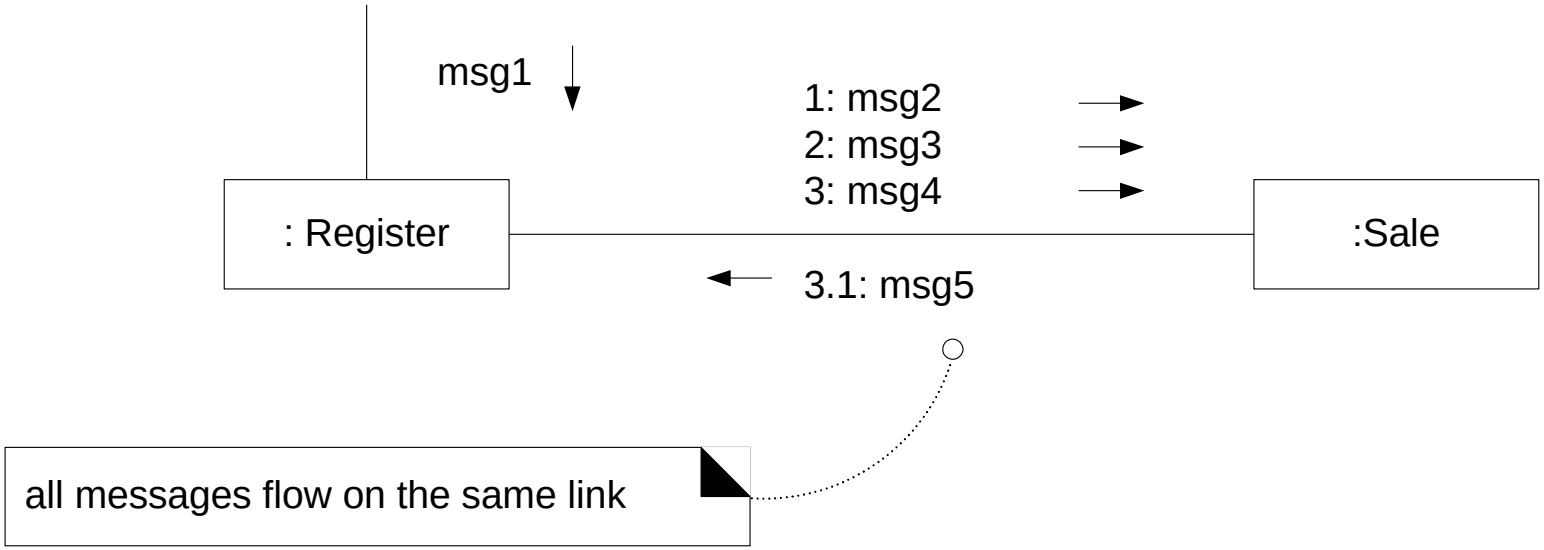
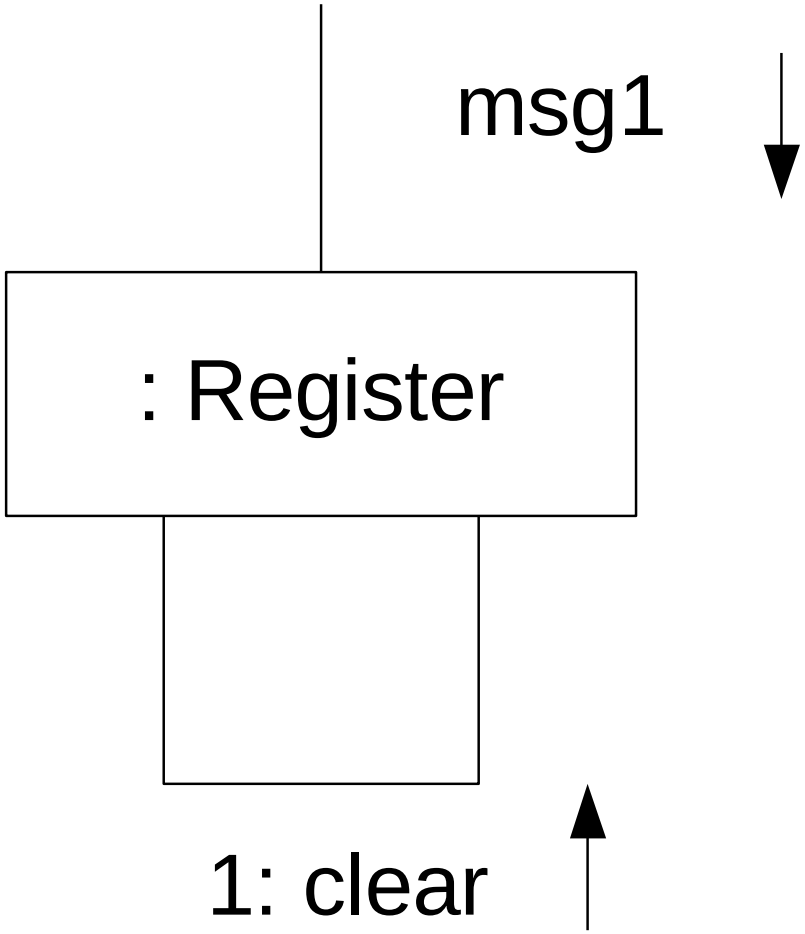


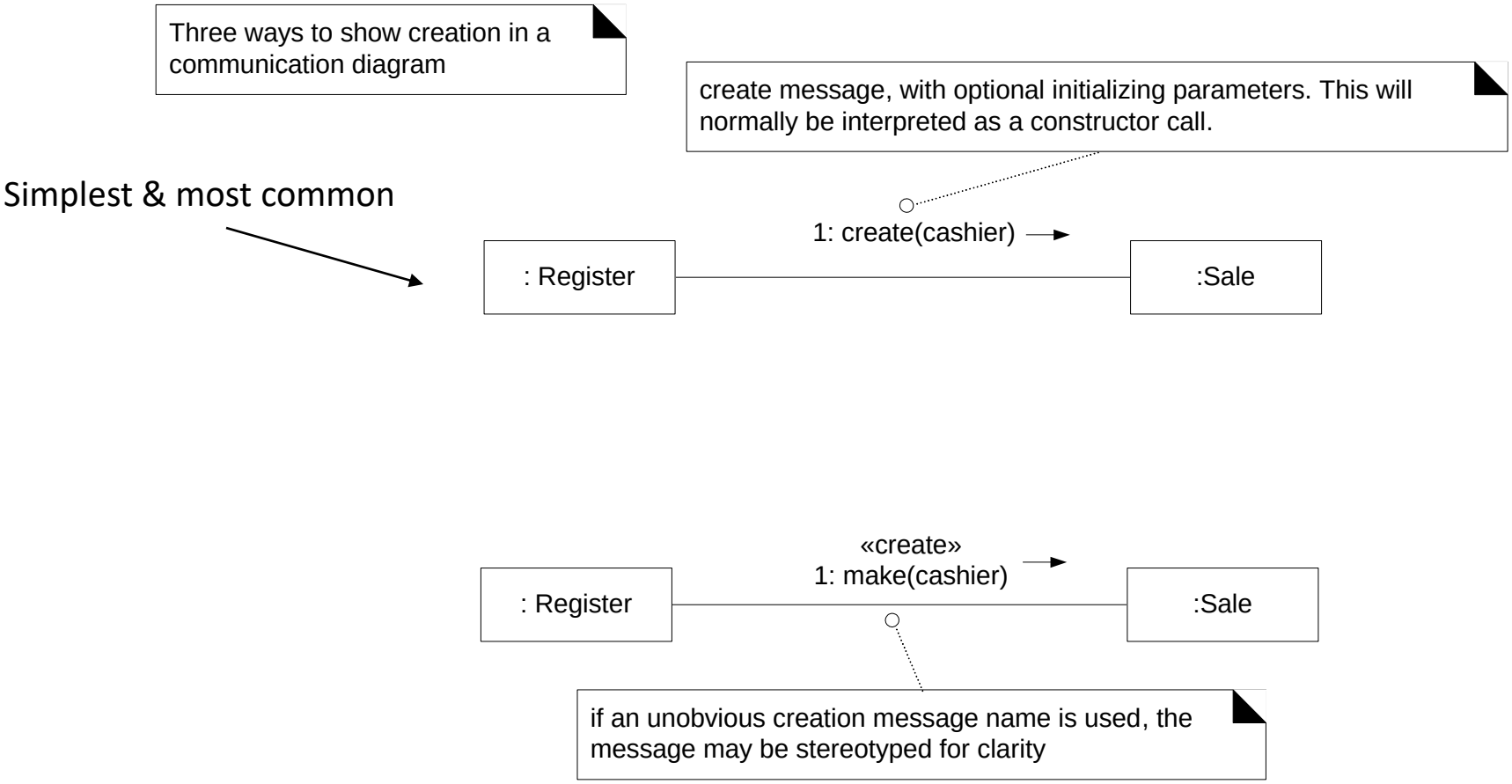
Diagramme de communication



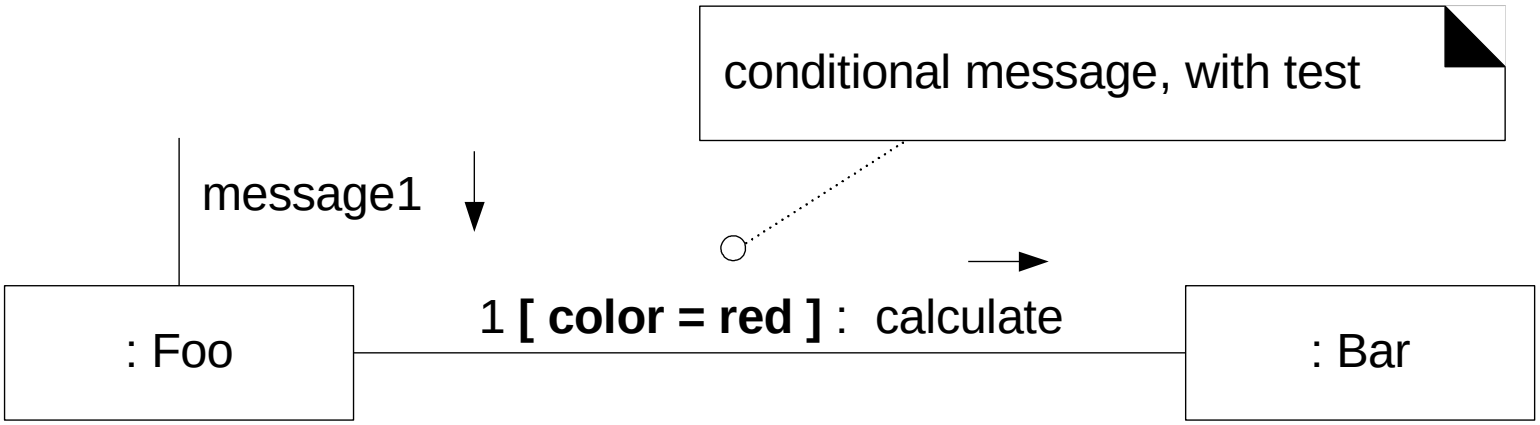
L’objet appelle lui-même une de ses méthodes



Création d'un objet



Condition



Conclusion

- Diagramme de séquence ou de communication?
- Quand créer un diagramme d'interaction?

Diagramme de séquence ou de communication?

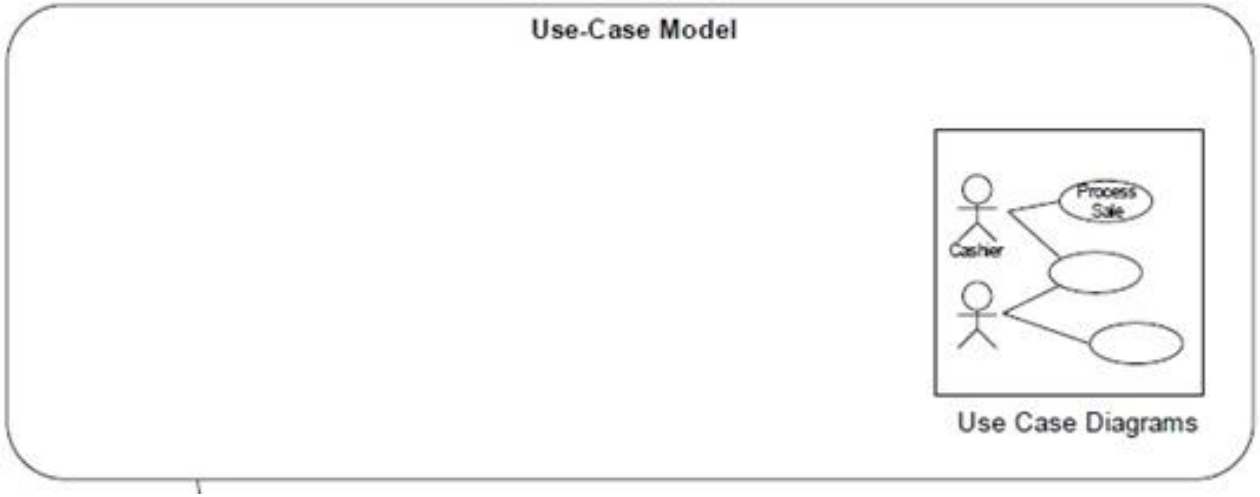
- Diagramme de séquence
  - Plus facile de voir la séquence des appels dans le temps
  - Notation UML plus expressive
  - Bien supporté dans la plupart des outils UML
- Diagramme de communication
  - Permet édition dans un espace limité (ex: tableau blanc)
  - Plus facile à modifier si on travaille à la main
  - Focus sur l'aspect spatial des relations

Quand créer un diagramme d'interaction?

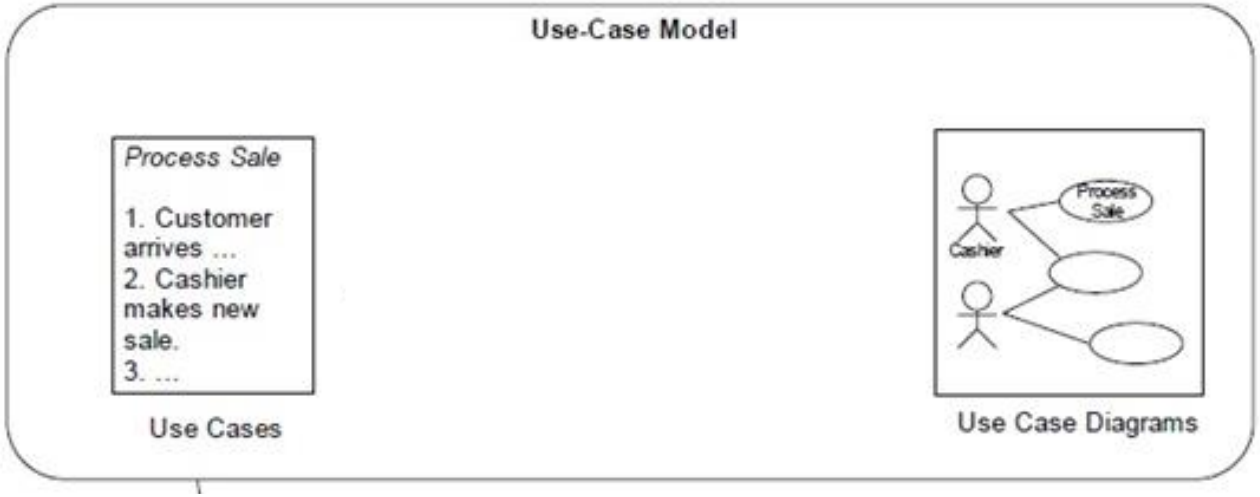
- Pour spécifier ce qui se passe à l'interne du système lors de la réalisation d'un scénario

  (ou d'une partie d'un scénario)
- En pratique, on crée ces diagrammes pour réfléchir aux parties les plus complexes du système
- Facilite la collaboration/communication au sein de l'équipe (rappelez-vous toujours: l'équipe cherche à s'entendre sur un design)

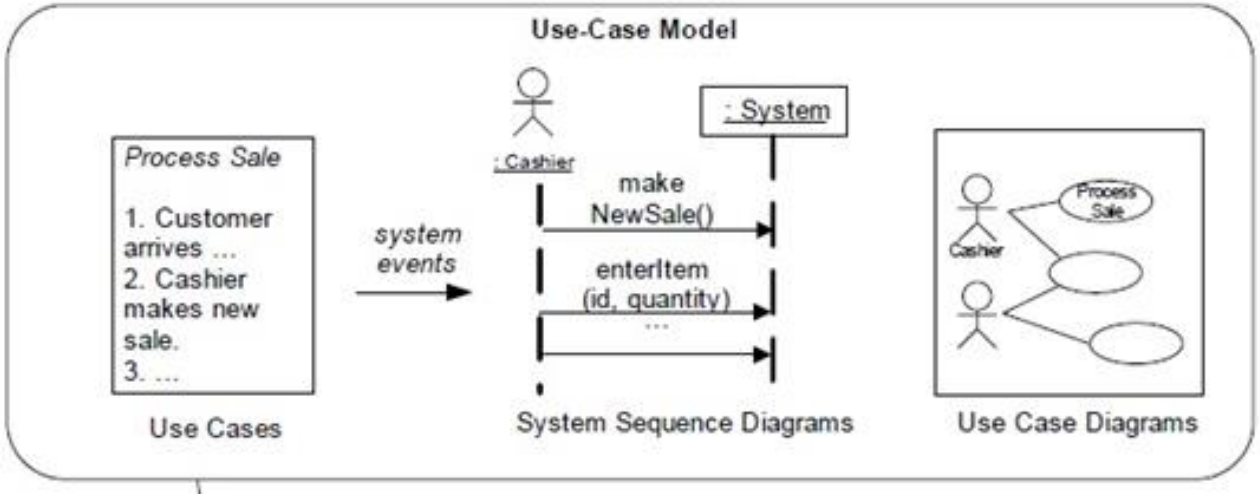
Spécifier ce qui se passe à l'interne lors de la réalisation d'un scénario



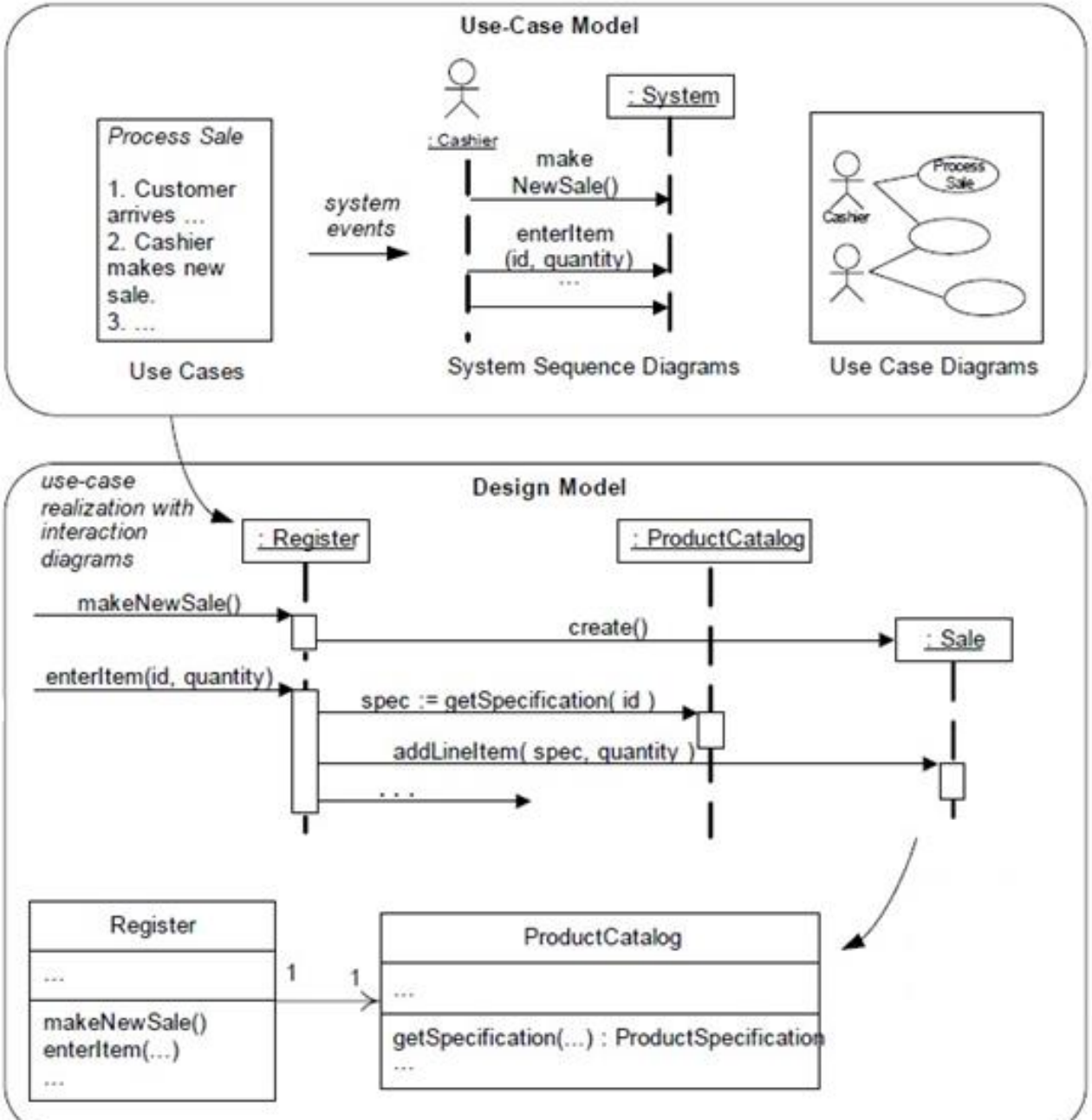
Spécifier ce qui se passe à l'interne lors de la réalisation d'un scénario



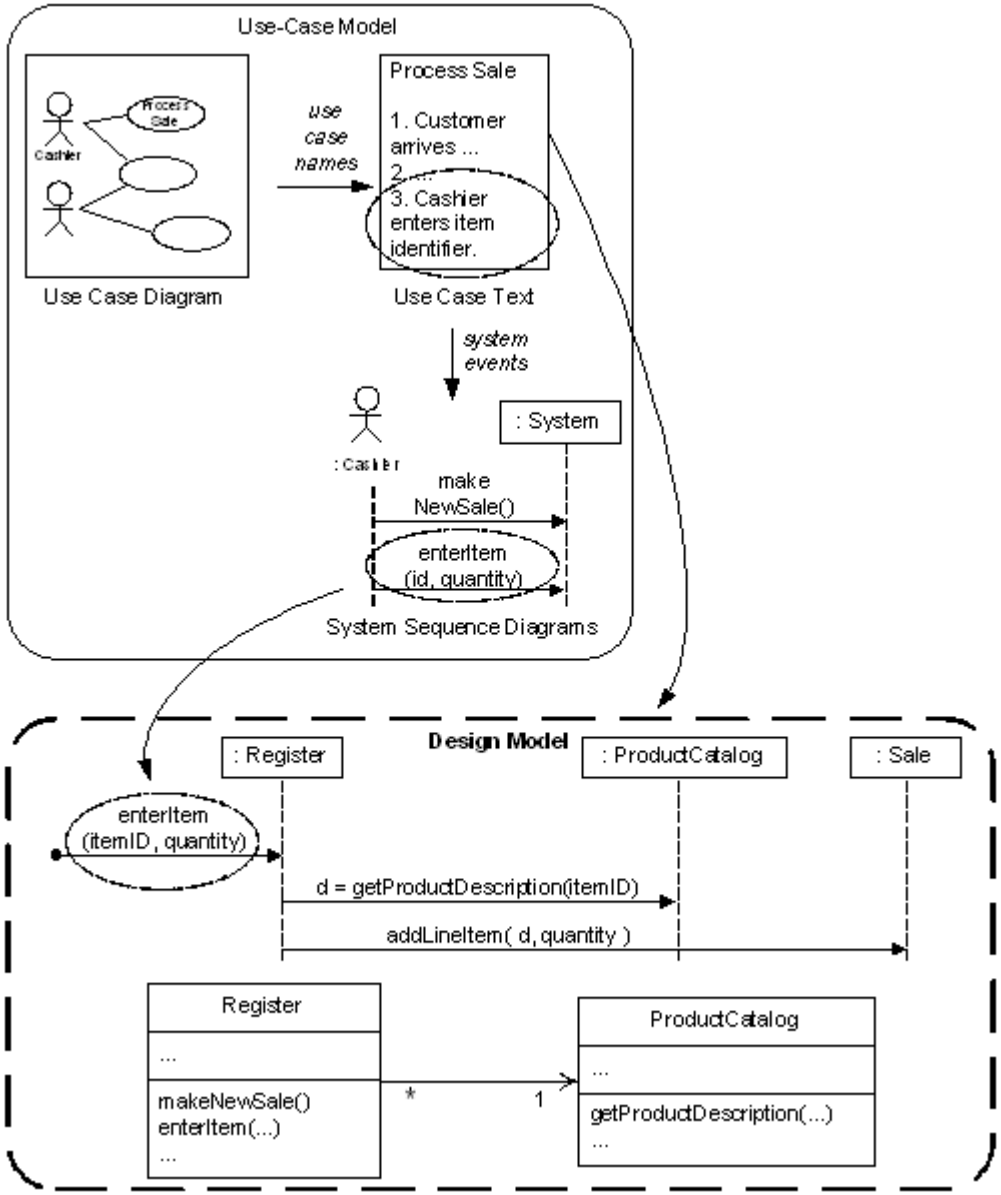
Spécifier ce qui se passe à l'interne lors de la réalisation d'un scénario



Spécifier ce qui se passe à l'interne lors de la réalisation d'un scénario



… ou d'une partie d'un scénario



L’œuf ou la poule ?

- On commence par le diagramme d’interaction ou bien diagramme de classe?