

CHAPITRE 7: DIAGRAMMES DE COMMUNICATION



Dr. Ilhem ABDELHEDI ABDELMOULA

Email: ilhemabdelmoula13@gmail.com

Université de Carthage

Enicarthage - École Nationale des Ingénieurs à Carthage

Département informatique

Niveau : 2^{ème} année INFO ING / 1^{ère} Année INFO ING

Semestre : 2

Année universitaire: 2019 - 2020

**• Diagramme de collaboration
(UML 1.X)**

**• Diagramme de
communication (UML 2.X)**

ANALYSE TECHNIQUE	Point de vue	Diagramme UML
	Statique – objet	Classes
		Objets
	Dynamique - objet	Séquence
		Communication (Ex collaboration)
	Dynamique Plutôt non objet	Etats-transitions
		Activités

DIAGRAMME DE COLLABORATION

- Les diagrammes de collaboration font partie des diagrammes d'interaction constituant la vue dynamique du système.
- Un diagramme de communication ou de collaboration, est un schéma d'interaction. Il se focalise sur une représentation spatiale des objets.
- Ils constituent **une extension des diagrammes d'objet en mettant en valeur les interactions entre eux** = un recoupement des diagrammes d'objets et des diagrammes de séquences.
- Ils 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 de l'envoi de messages).

DÉFINITION D'UNE COLLABORATION

- Montre des objets qui collaborent dans un contexte donné pour mettre en œuvre une fonctionnalité donnée ou un CU du système.
- Une collaboration est un comportement collectif d'objets et d'acteurs liés entre eux et communiquant via l'échange de messages dans un contexte donné en vue de réaliser une opération
- Représente le contexte d'une interaction : on peut y préciser les états des objets qui interagissent.

DÉFINITION DE DIAGRAMME DE COMMUNICATION

- *Représente un ensemble de rôles joués par les objets dans un contexte particulier*, ainsi que les liens entre ces objets.
- Met l'accent sur les interactions entre les objets du système et les messages qu'ils échangent et les dépendances entre les objets impliqués dans l'exécution d'un processus ou d'un cas d'utilisation.

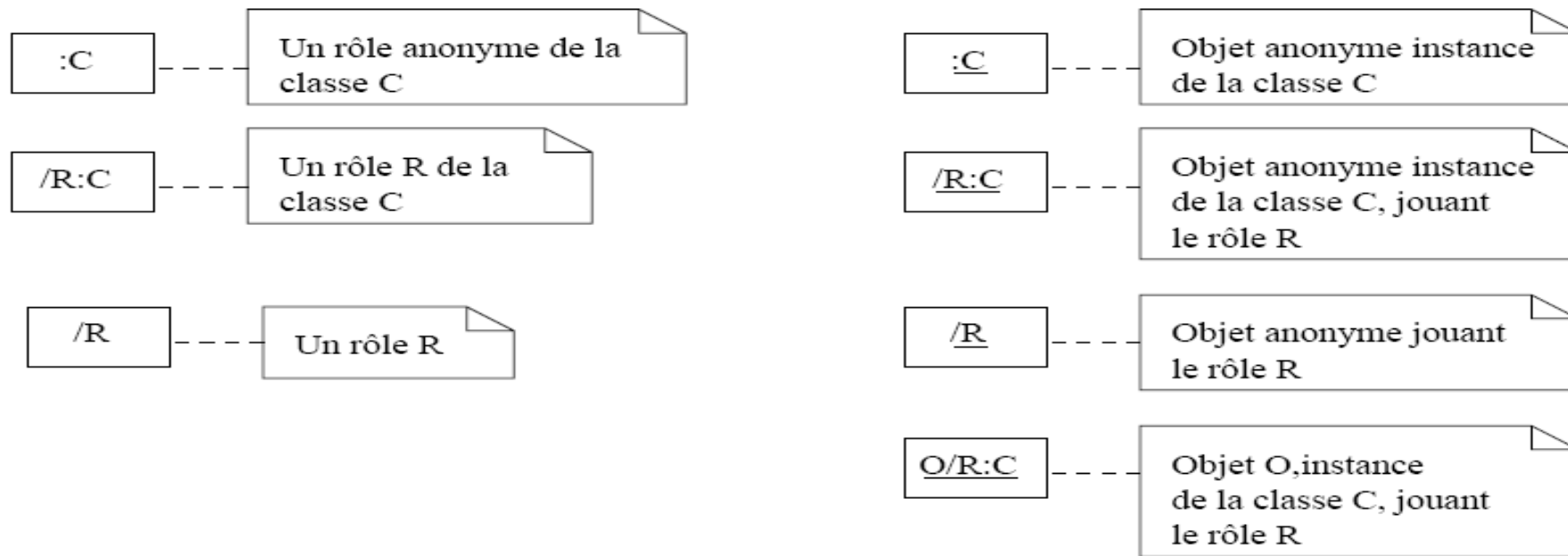
COLLABORATION

Une collaboration est composée de deux descriptions.

- 1. Description générale au niveau de la spécification :** Celle-ci représente les rôles des cas d'utilisation, des classes, des méthodes et des associations. Une interaction sera alors une séquence de messages échangés entre les rôles.
- 2. Description spécifique au niveau instance :** Celle-ci représente une instance particulière d'une interaction composée d'objets et de liens respectant les rôles, et de stimulus qui sont des instances de messages échangés entre ces objets.

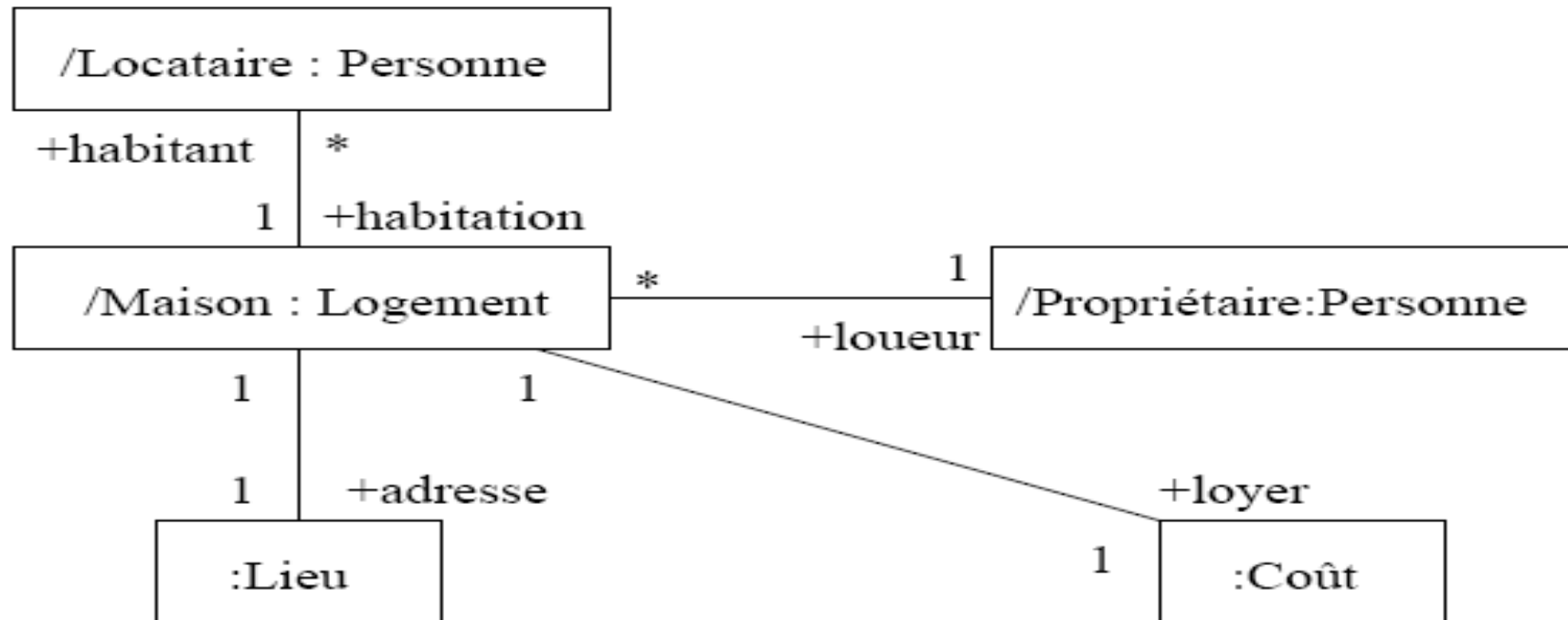
LES RÔLES

- *Les rôles des classificateurs (classes, cas d'utilisation, ...) sont représentés par un symbole de classe :*



1. NIVEAU SPÉCIFICATION

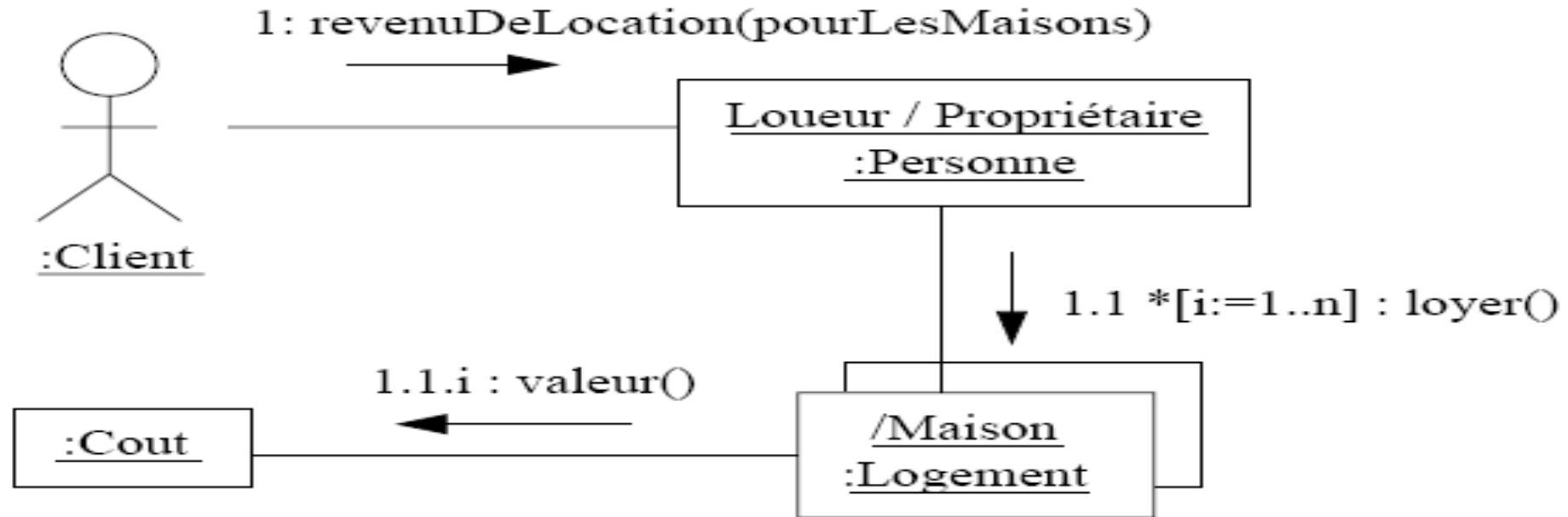
- Une collaboration au niveau spécification représente **un contexte**.
- La collaboration forme un graphe de rôles liés par des rôles d'associations



2. NIVEAU INSTANCE

Diagramme représentant une instance du diagramme au niveau spécification avec des stimulus.

- *Stimulus* : instance d'un message envoyé d'un objet vers un autre.



DIAG.COMMUNICATION VS. DIAG.SEQUENCES

Version simplifié du diagramme de séquence

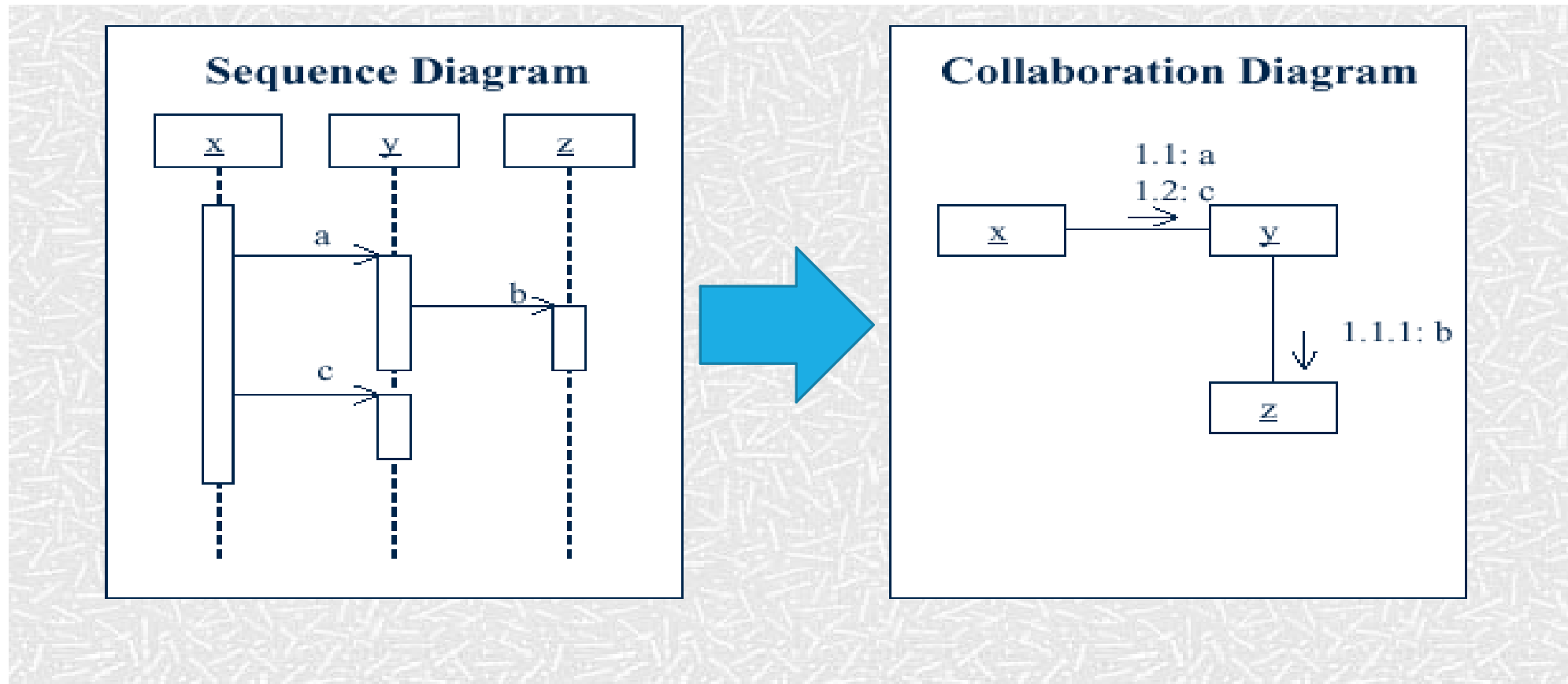
2 vues différentes mais logiquement identiques : **isomorphe**

- Diagramme de communication : Adapté pour montrer les liens
- Diagramme de séquence: Importance des messages

Différences :

- Libre de placer les participants (objets).
- On fait des liens entre eux, et on les numérote (interactions)
- Pas de ligne de vie

DIAG.SEQUENCES VS. DIAG.COMMUNICATION



FORMALISME

- Les concepts importants : **objets, acteurs, liens et messages**
- Les objets sont présentés avec des liens ou des connecteurs d'association entre eux.
- Les messages sont ajoutés à des associations et des flèches montrent la direction du flux de messages.
- La séquence des messages est présentée à travers un système de numérotation.

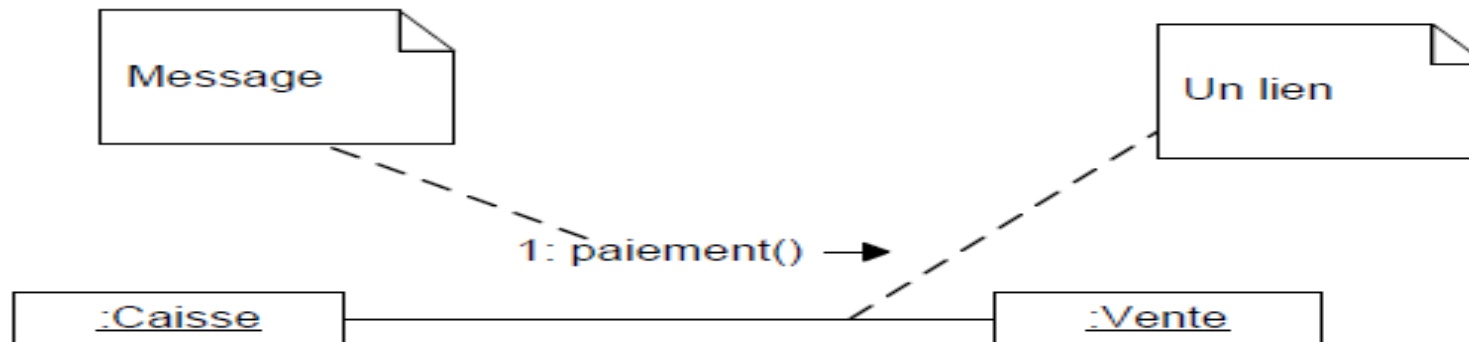
FORMALISME

□ Liens

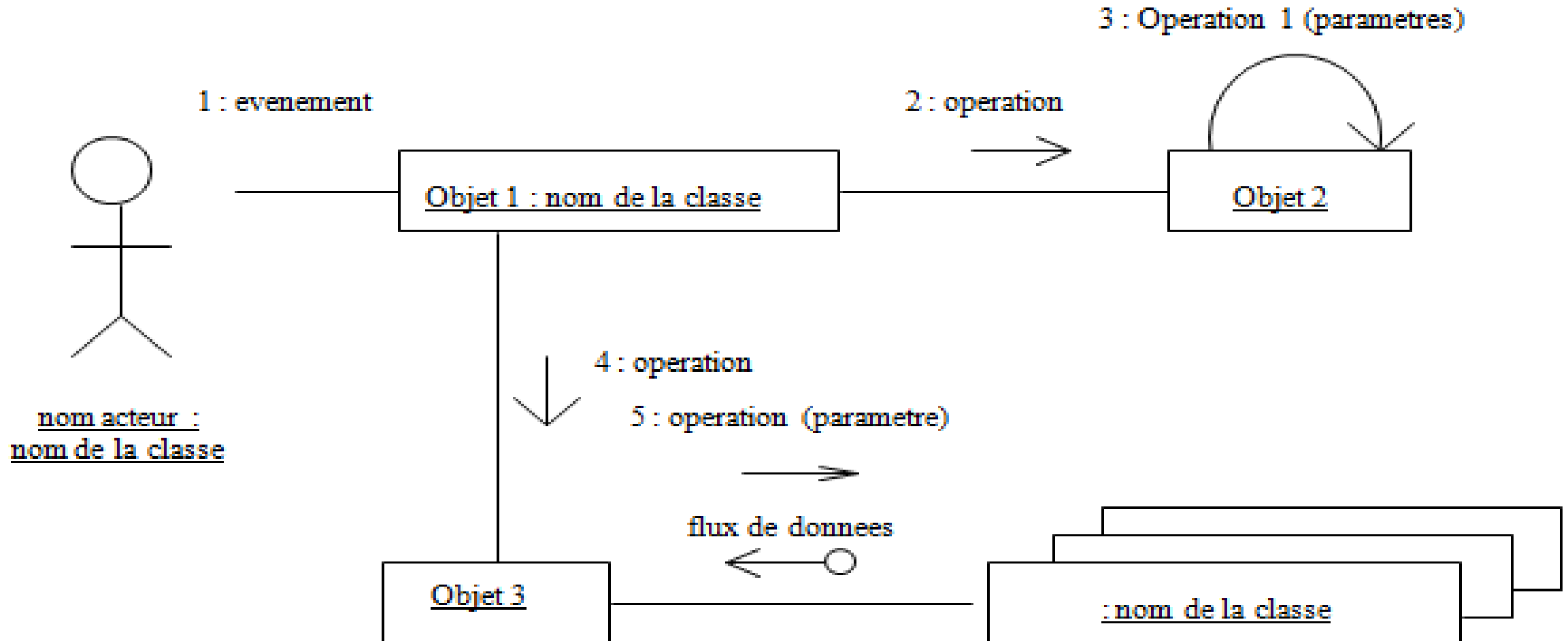
- Un lien montre l'envoi d'un message d'un objet à un autre objet
- Formellement, un lien est une instance d'une association

□ Messages

- Chaque message entre les objets est représenté par une expression de message et une flèche montrant la direction du message

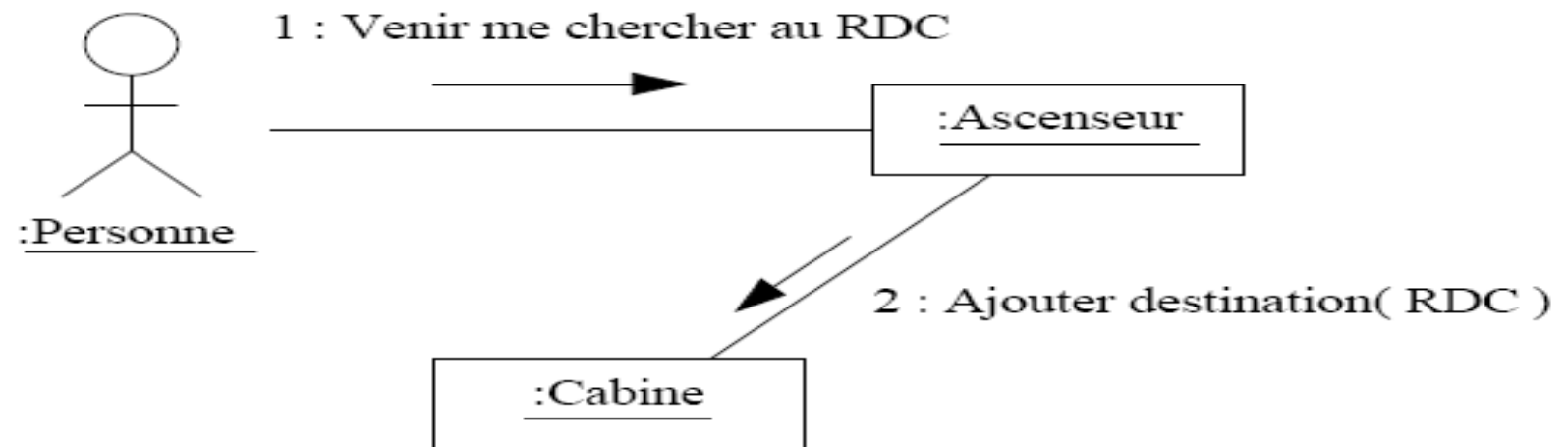


FORMALISME



ACTEUR

- Il spécifie un utilisateur externe, ou un ensemble d'utilisateurs liés qui interagissent avec un système.
- *On représente les interactions déclenchées par un élément extérieur au système, ex : **un acteur***



OBJET

- Un objet est une instance d'une classe. Il peut-être persistant ou transitoire :
 - un objet persistant est un objet qui continue d'exister une fois que le processus qui l'a créé est terminé.
 - Un objet transitoire est un objet qui cesse d'exister une fois que le processus qui l'a créé est terminé.
- Les objets sont obtenus à partir des classes. Chaque objet est décrit par :
 - *Nom-objet :classificateur* ou *nom-objet :nom-classe*.

LES OBJETS

:Objet

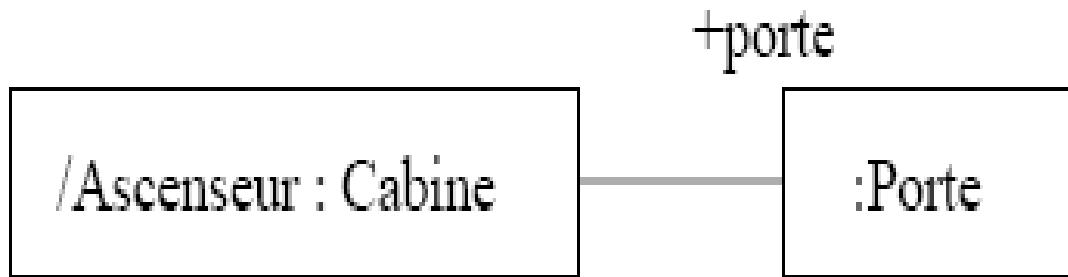
LenovoR500

LenovoR500:Ordinateur

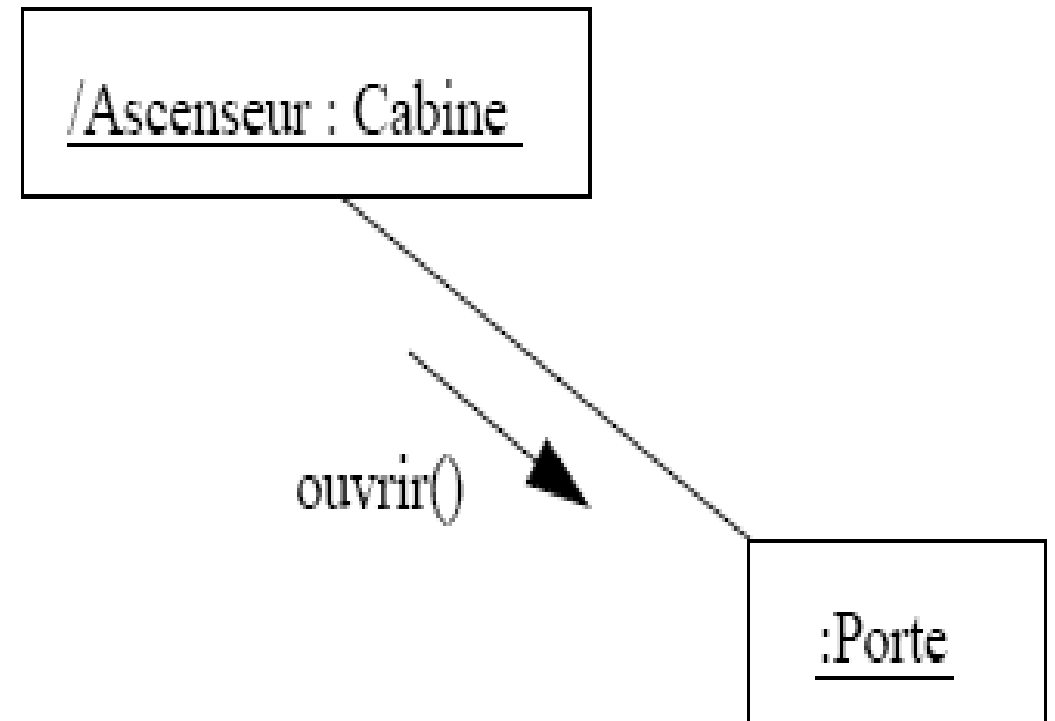
:Ordinateur

- Représenté par un rectangle
- Nommage :
 - Nom de l'objet instancié
 - Nom de l'objet et nom de la classe
 - Nom de la classe
- UML 2.X : plus de soulignement

EXEMPLE



Niveau spécification



Niveau instance

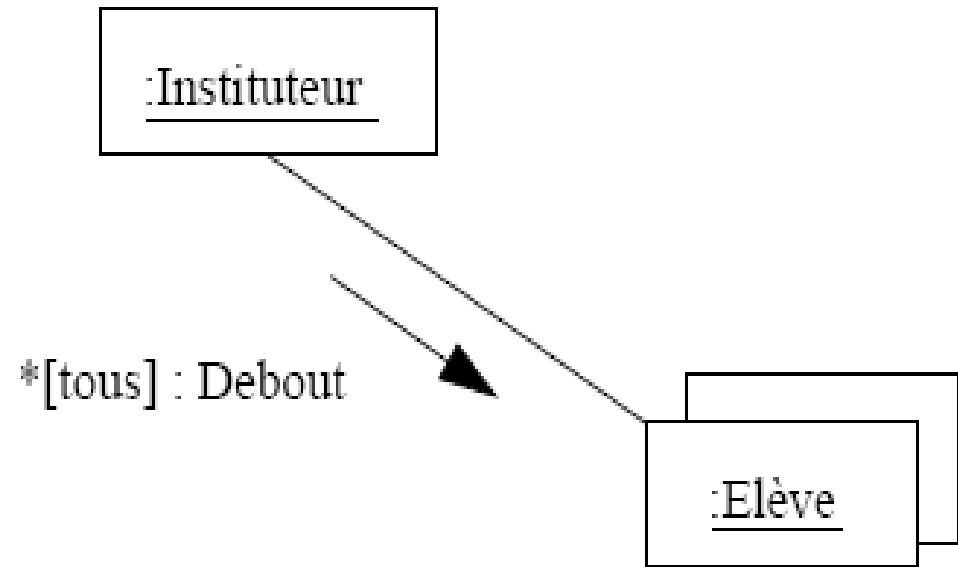
OBJET – UN MULTIPLE

- Un *multiple* définit un jeu d'instances. Il s'agit d'une représentation graphique d'un objet qui présente plusieurs instances.
- Il ne peut en revanche avoir qu'un seul ensemble d'attributs et ce, même s'il représente plusieurs instances.
- Un objet peut communiquer avec un autre objet qui est un multiple. Cette fonctionnalité est principalement utilisée dans le diagramme de collaboration mais peut également être utilisée dans le diagramme d'objets.
- *Exemple* : Un employé gère une liste de documents => c'est la liste qui représente l'objet multiple.

EXEMPLE : REPRÉSENTATION CONDENSÉE



Niveau spécification



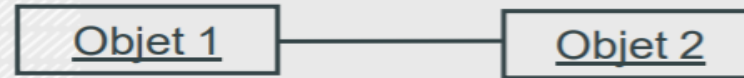
Niveau instance

UN LIEN

- Une connexion entre deux objets, sous forme de navigation permettant de mettre en exergue la collaboration entre objets
- Il est représenté sous forme d'un trait plein entre : Deux objets ou entre un objet et un acteur (ou vice-versa)
- Un lien entre objets peut être une instance d'association entre classes.
- Il permet d'acheminer des messages dans un sens ou dans l'autre.
- Plusieurs messages, et ce dans les deux sens, peuvent circuler sur le même lien.
- **Il n'y a pas un lien par message : tous les messages empruntent le même lien.**

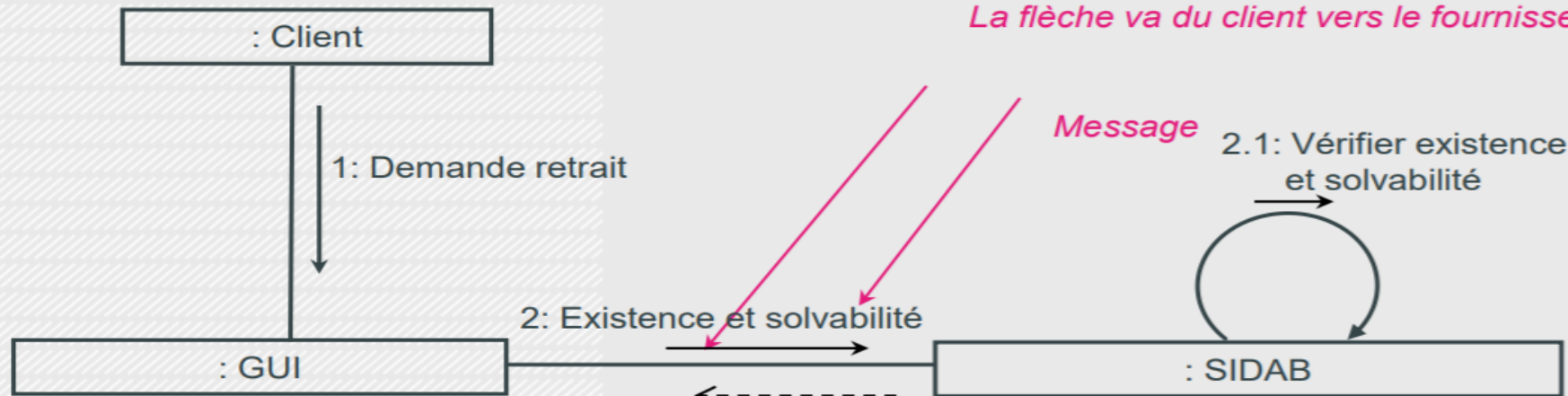
LES LIENS

- Indique un chemin de communication entre 2 objets, sur lequel passent les messages



lien d'interaction

- Exemples :



La flèche va du client vers le fournisseur

Message

2.1: Vérifier existence et solvabilité

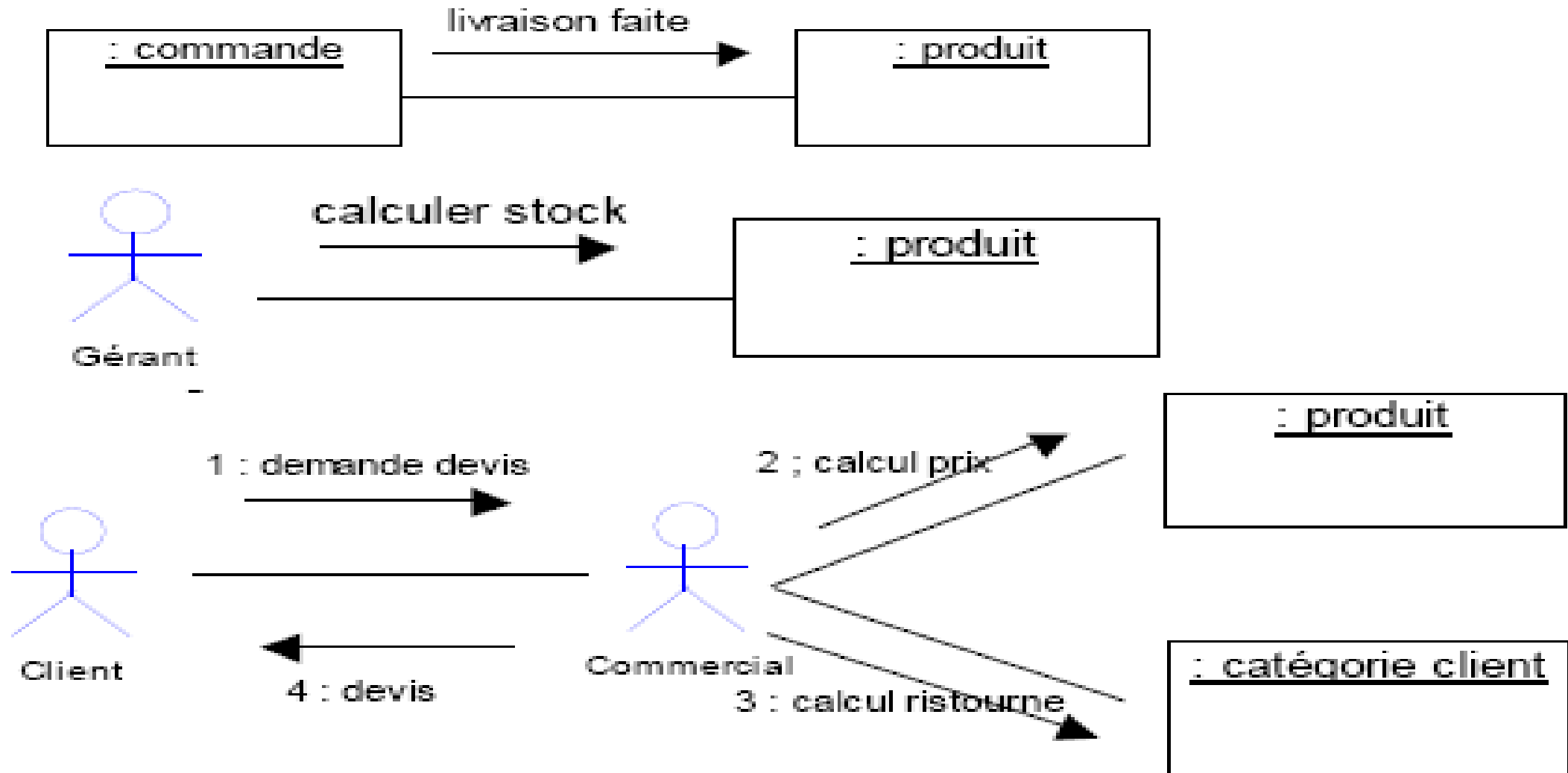
Représentation des Actions internes à l'objet au moyen de message que l'objet s'envoie à lui même

Objet client/appelant

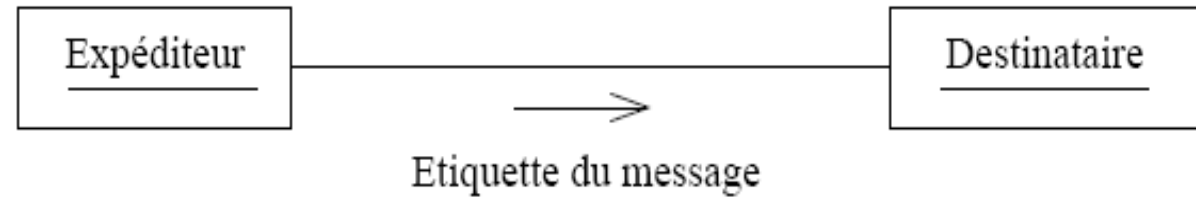
Donnée de retour

Objet fournisseur/appelé

EXEMPLES D'INTERACTION



LES MESSA



Les objets communiquent en échangeant des messages représentés sous forme de flèches.

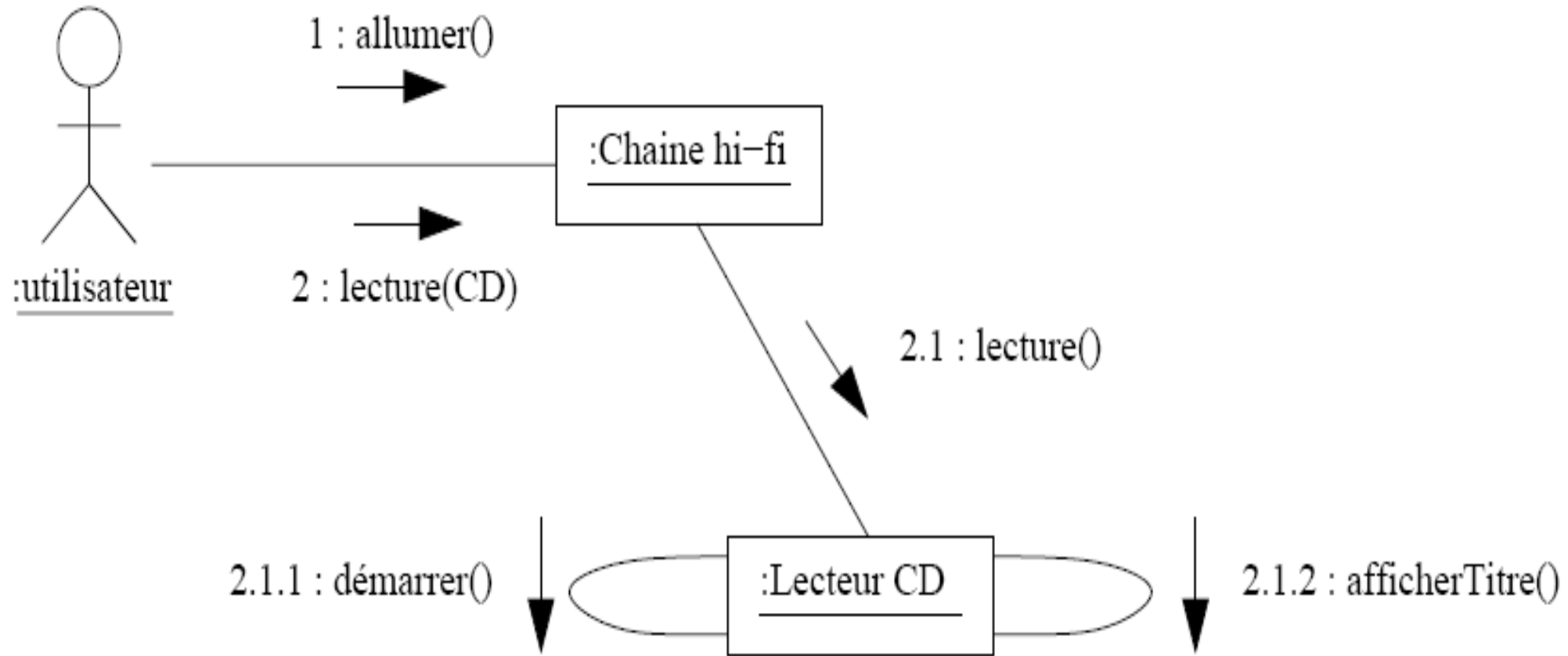
- *Les messages sont étiquetés par le **nom de l'opération ou du signal invoqué**.*
- *L'envoi d'un message nécessite que le récepteur puisse réaliser l'opération.*

[pred] ['[cond']'] [séqu] [iter] [r :=] msg ['('arguments')']

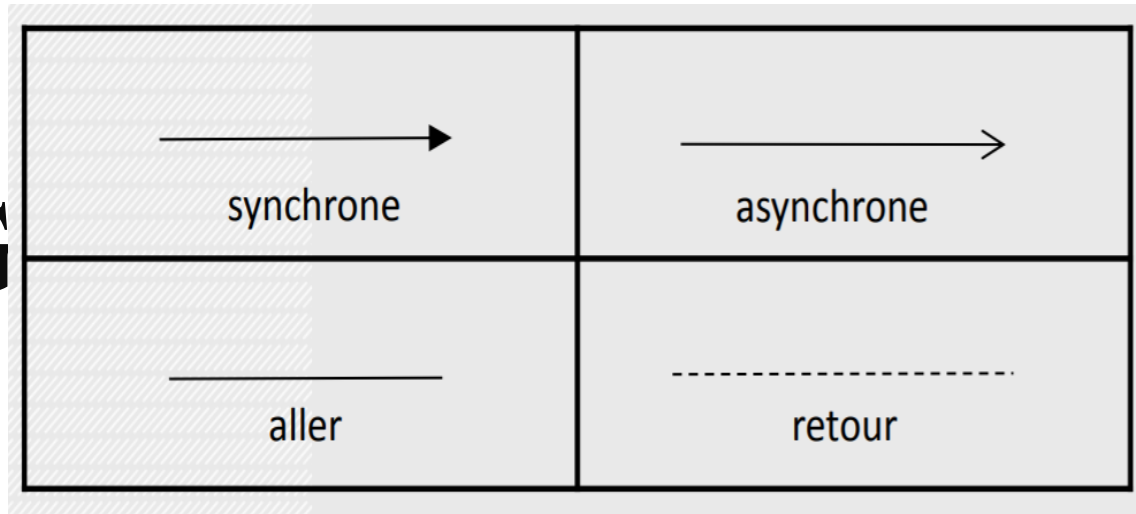
Les étiquettes décrivent les messages auxquels elles sont attachées.

- **msg** : *nom de l'opération ou du signal invoqué* par l'intermédiaire de ce signal
- **cond** : *expression booléenne et optionnelle* autorisant ou non l'envoi d'un message.
- **Seq** : *numéro d'ordre du message par rapport aux autres messages envoyés*

MESSAGE - EXAMPLE



LES MESSAGES

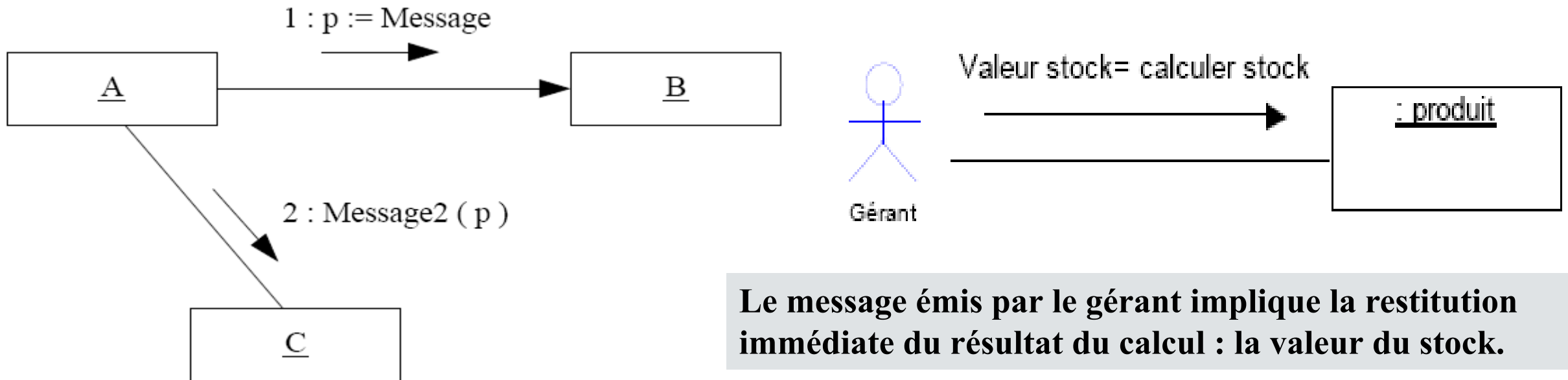


Le message est un envoi de signal ou une invocation d'une opération. Il peut être :

1. **Synchrone** : Un message est envoyé par à un objet à un autre, et le premier objet attends jusqu'à ce que l'action ai finie.
2. **Asynchrone** : Un message est envoyé par à un objet à un autre, mais le premier objet n'attends pas la fin de l'action.
3. **Aller (plat)** : La plupart asynchrone, Chaque flèche représente une progression d'une étape à une autre dans la séquence.
4. **Retour** : Le retour explicite d'un objet à qui le message était envoyé.

MESSAGE - RÉSULTAT

- Le résultat est constitué d'une liste de valeurs retournées par le message.
- *Ces valeurs peuvent être utilisées comme paramètres des autres messages.*

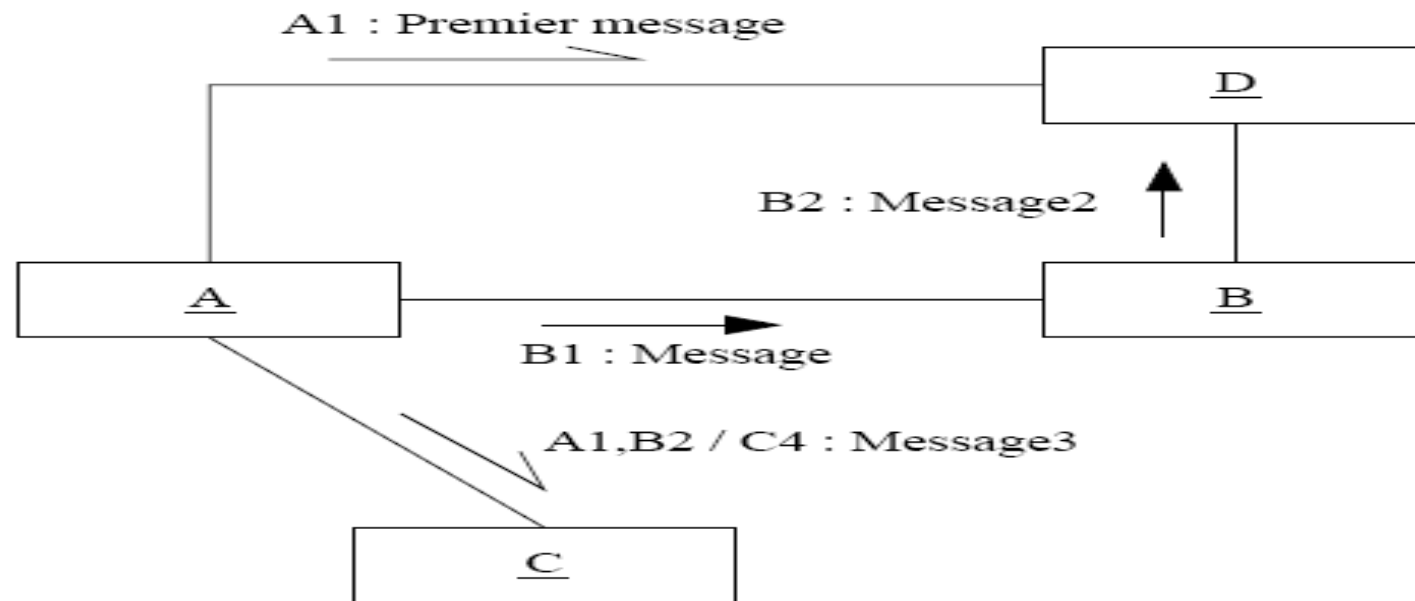


Le message émis par le gérant implique la restitution immédiate du résultat du calcul : la valeur du stock.

MESSAGE : SYNCHRONISATION

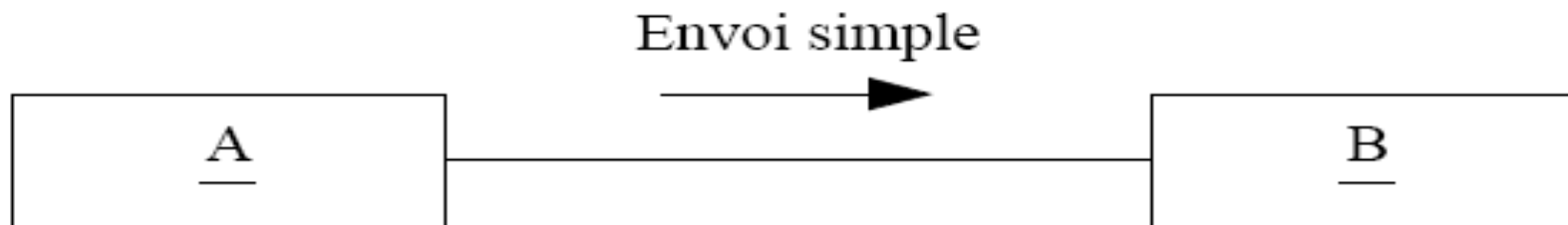
Préd : liste des messages qui précèdent le message courant .
L'envoi de ce message ssi d'autres messages sont déjà envoyés.

- *Syntaxe : **message1, message2 ... /***



MESSAGE - APPEL PROCÉDURE

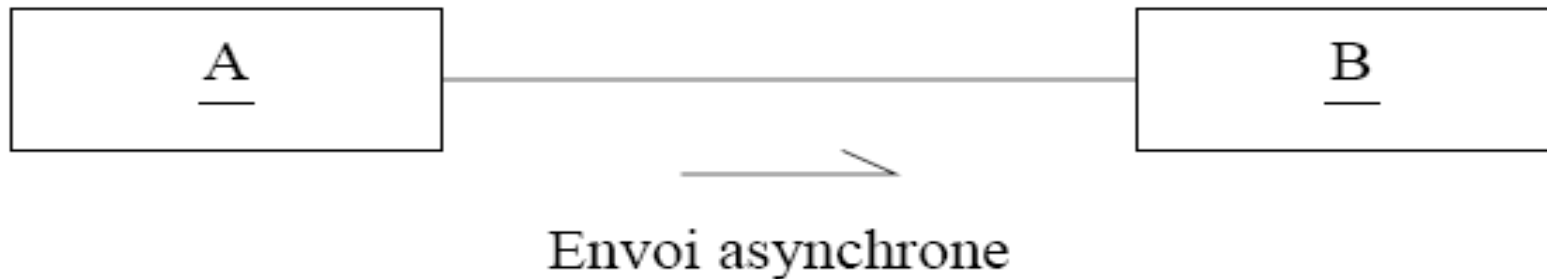
- Dans un appel de procédure (flot de contrôle emboîté), la séquence emboîtée doit se terminer pour que la séquence englobante reprenne le contrôle.
- *Les appels de procédure sont représentés par des flèches à pointe triangulaire*



MESSAGE - FLOT DE CONTRÔLE SYNCHRONE

Dans un flot de contrôle asynchrone, il n'y pas de message englobé ou englobant *pas d'obligation de terminaison d'autre messages pour continuer.*

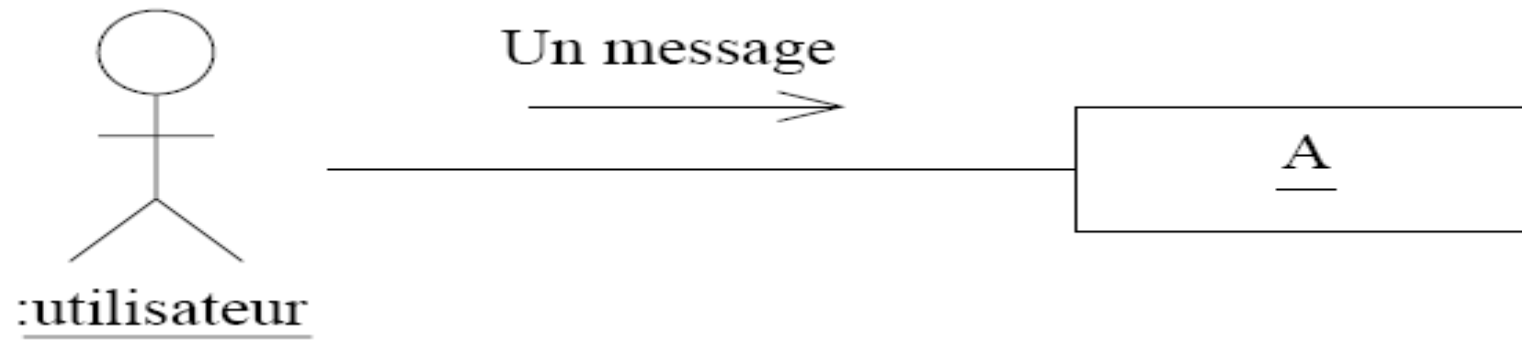
- *Les messages asynchrones sont représentés par des demi-flèches.*



MESSAGE - FLOT DE CONTRÔLE À PLAT

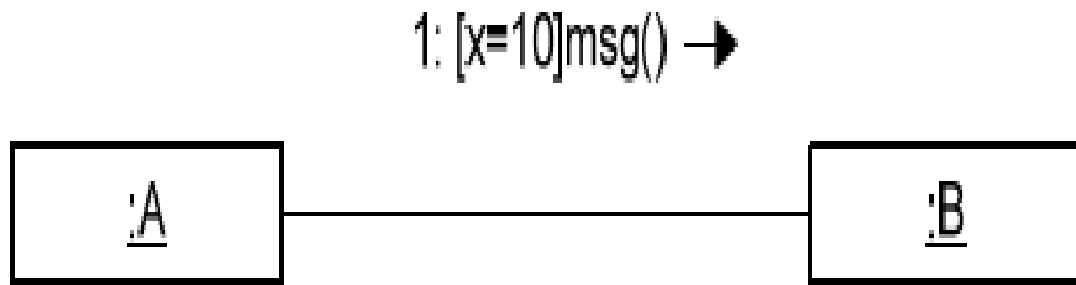
Cas particulier de messages asynchrones.

- *Modélisation d'une progression non procédurale souvent utilisée pour les messages entre un acteur et le système.*
- *Les messages sont représentés par des flèches simples*

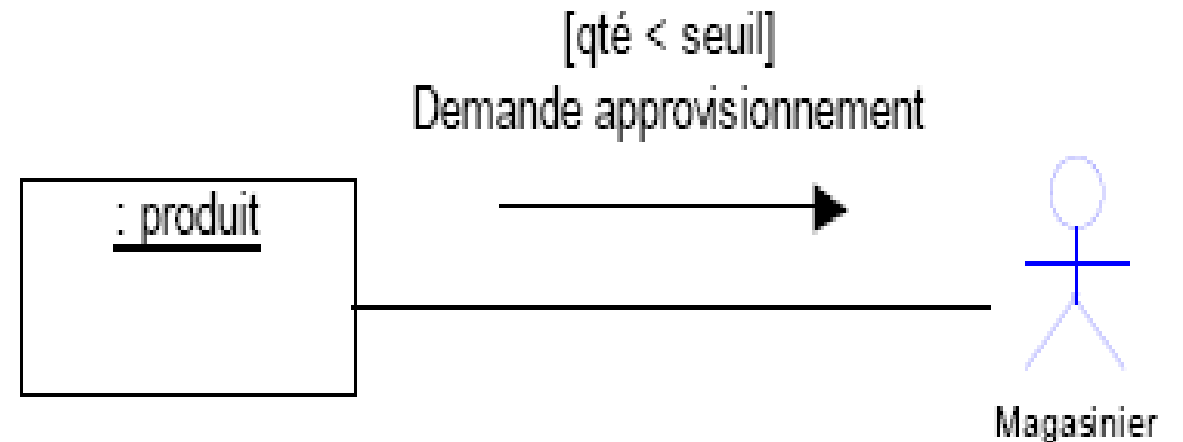


MESSAGE - CONDITION

- Réalisé par une condition spécifiée entre crochets après le numéro du message:
Numéro[condition] : message
- L'émission de message peut également être soumis à **une condition**, qui s'exprime sur le texte du message



La demande de réapprovisionnement n'est envoyée au magasinier que lorsque la quantité en stock est inférieure au seuil de réapprovisionnement

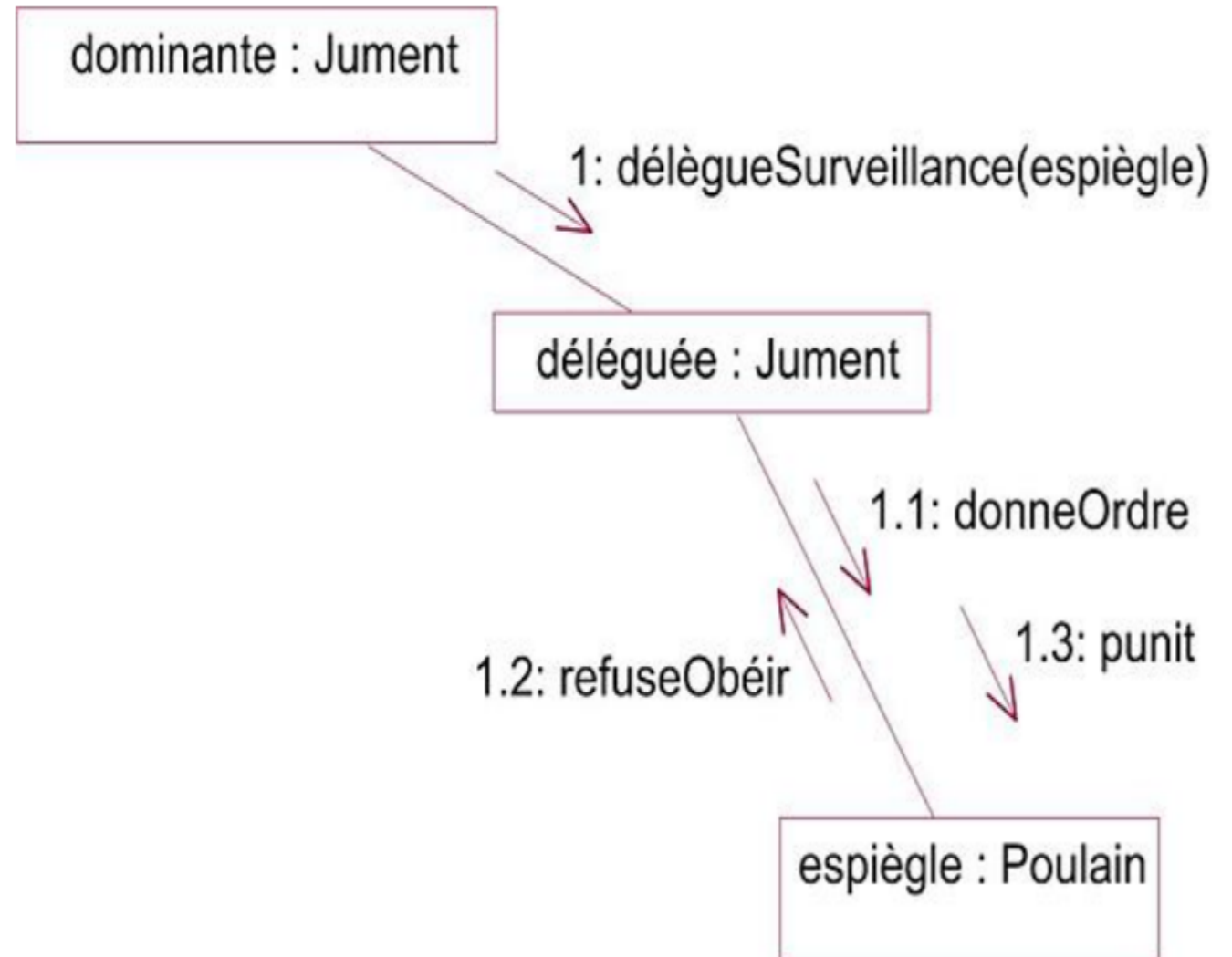


EXEMPLE

Dans un troupeau de chevaux, il existe une jument dominante qui est responsable de l'éducation de tous les poulains. Elle peut passer le relais à une autre jument pour la surveillance d'un poulain particulier.

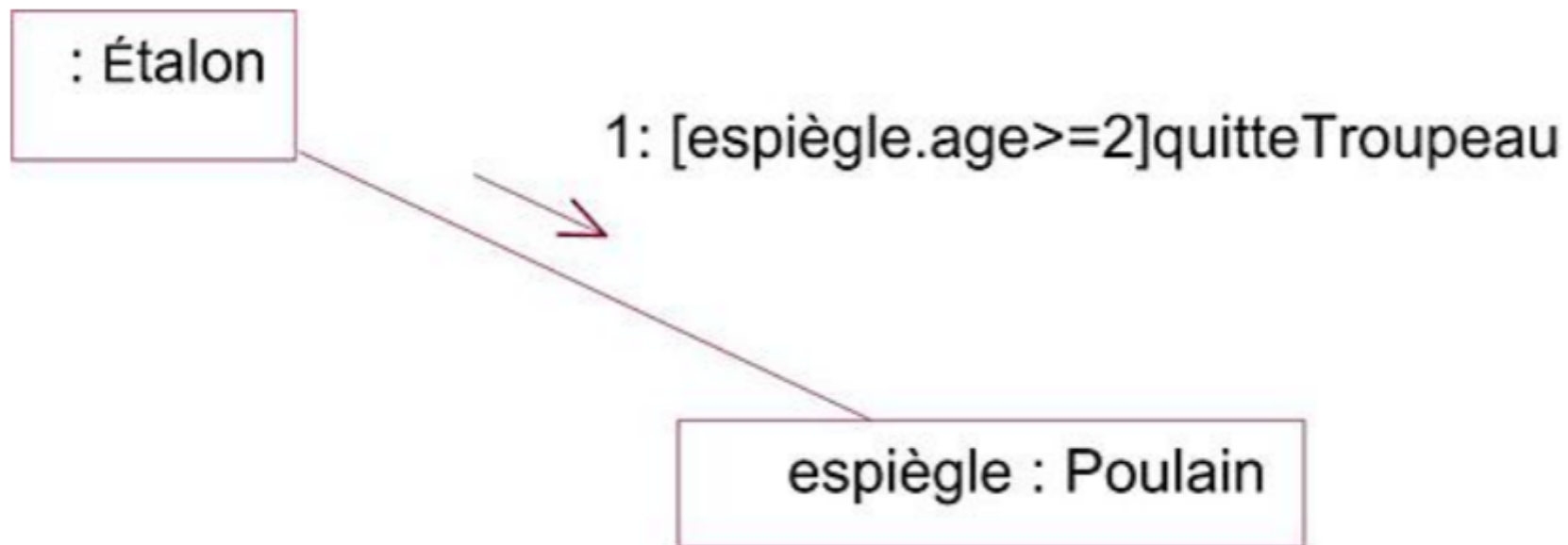
Ex:

La jument dominante délègue la surveillance du poulain Espiègle à une autre jument, qui lui donne un ordre auquel il refuse d'obéir. Il est donc puni.



EXEMPLE – MESSAGE CONDITIONNEL

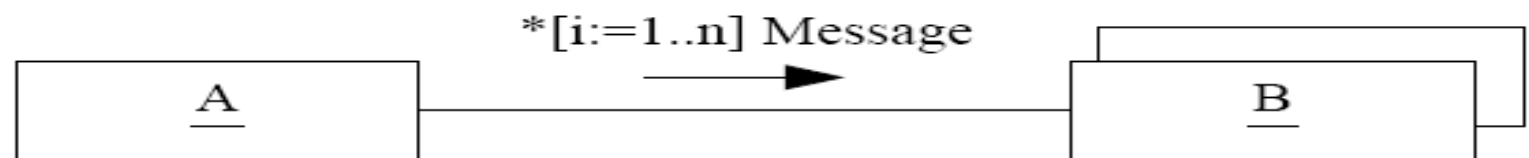
Quand un poulain atteint l'âge de deux ans, il est chassé du troupeau par l'étalon.
Le poulain Espiègle doit quitter le troupeau à l'âge de deux ans



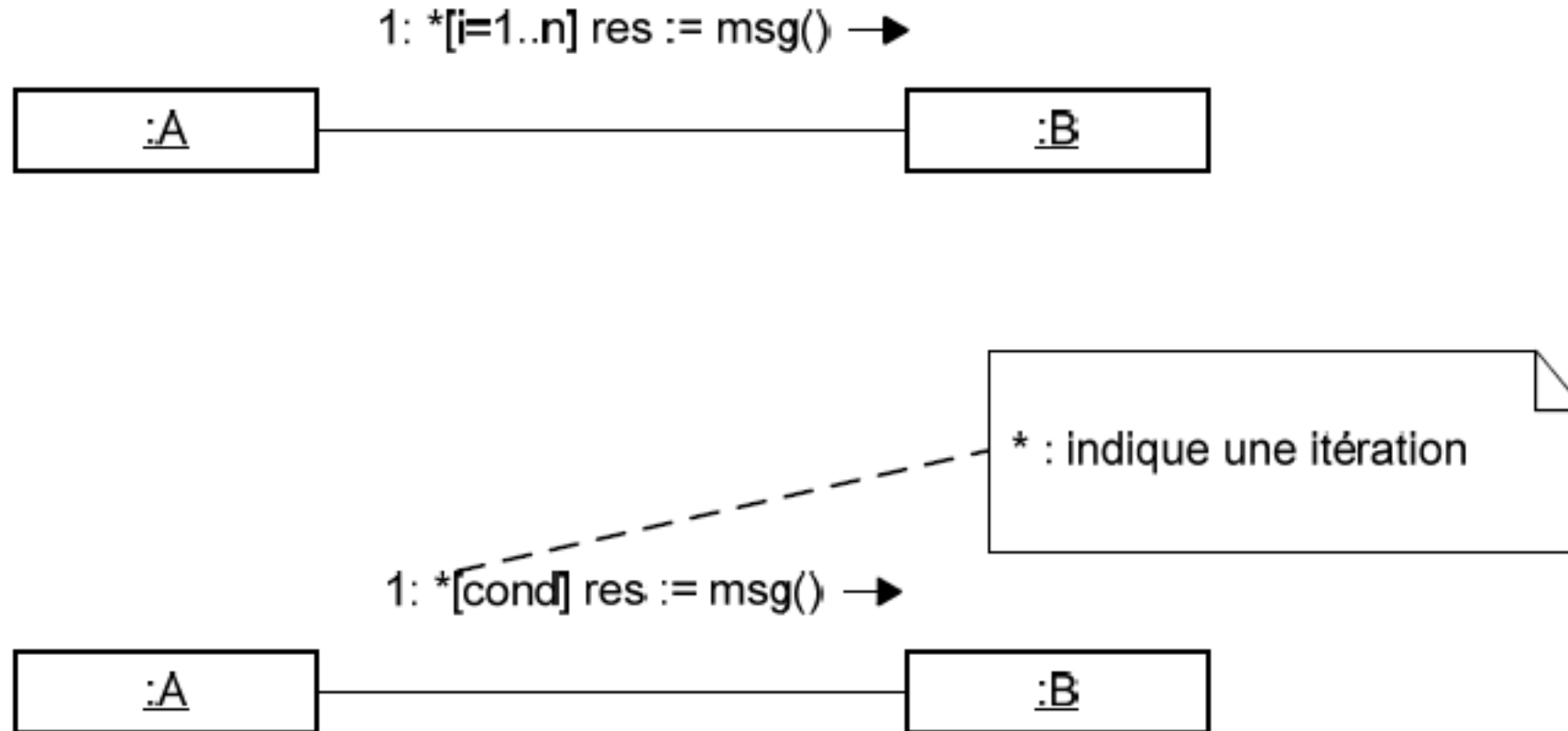
MESSAGES - ITÉRATION

Il existe deux mécanismes de boucle :

1. Itération **séquentielle** : envoi séquentiel de *n instances du même message*.
 - **Numéro*[condition]: message** basé sur une condition. La boucle s'exécute tant que la condition est vraie.
 - **Numéro*[variableBoucle=limiteInf..limiteSup]: message** basé sur une variable de boucle dont les limites inférieures et supérieures sont spécifiées.
2. Itération **parallèle** : envoi parallèle de *n instances du même message*.
 - Syntaxe : ***[[clause d'itération]]**

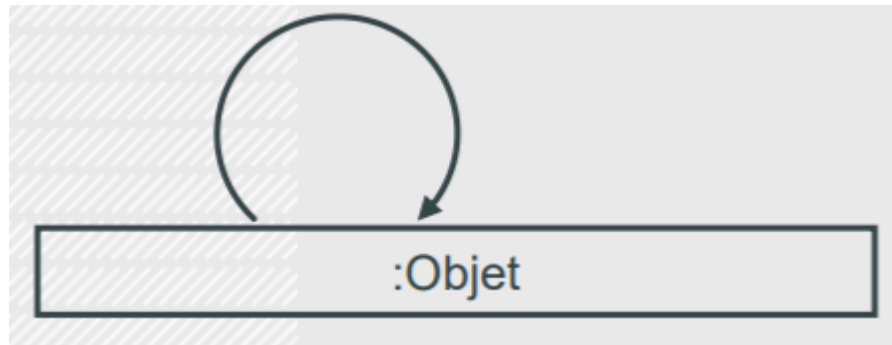


EXEMPLE DE MESSAGE ITÉRATIVE SEQUENTIEL



MESSAGE REFLEXIF

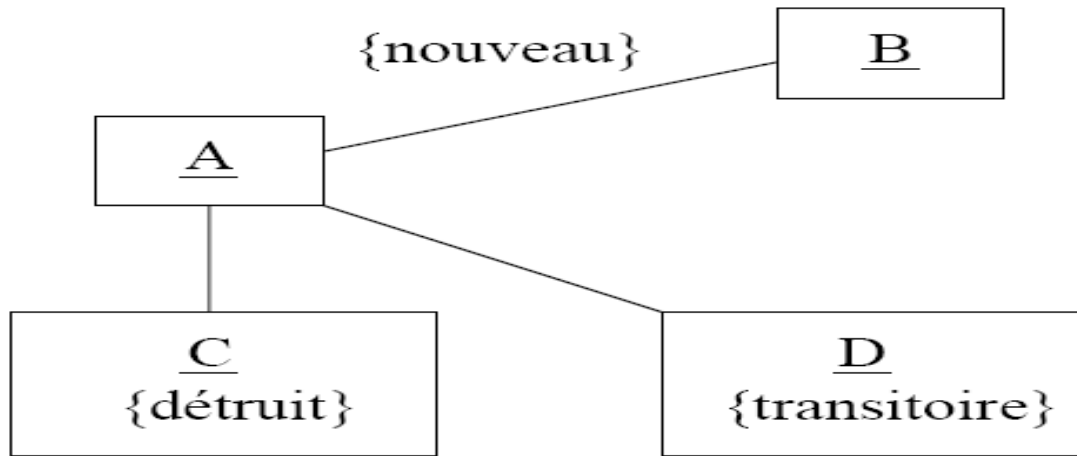
Messages envoyés à lui même (réflexif)



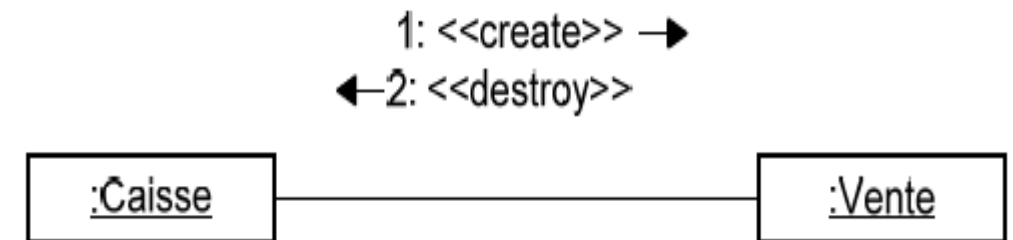
OBJETS CRÉÉS ET DÉTRUITS

Les objets créés ou détruits au cours d'une interaction peuvent **respectivement** porter les contraintes : **{Nouveau}** ou **{Détruit}**

- *Les objets créés, puis détruits au sein de la même interaction, portent la contrainte **{transitoire}**.*

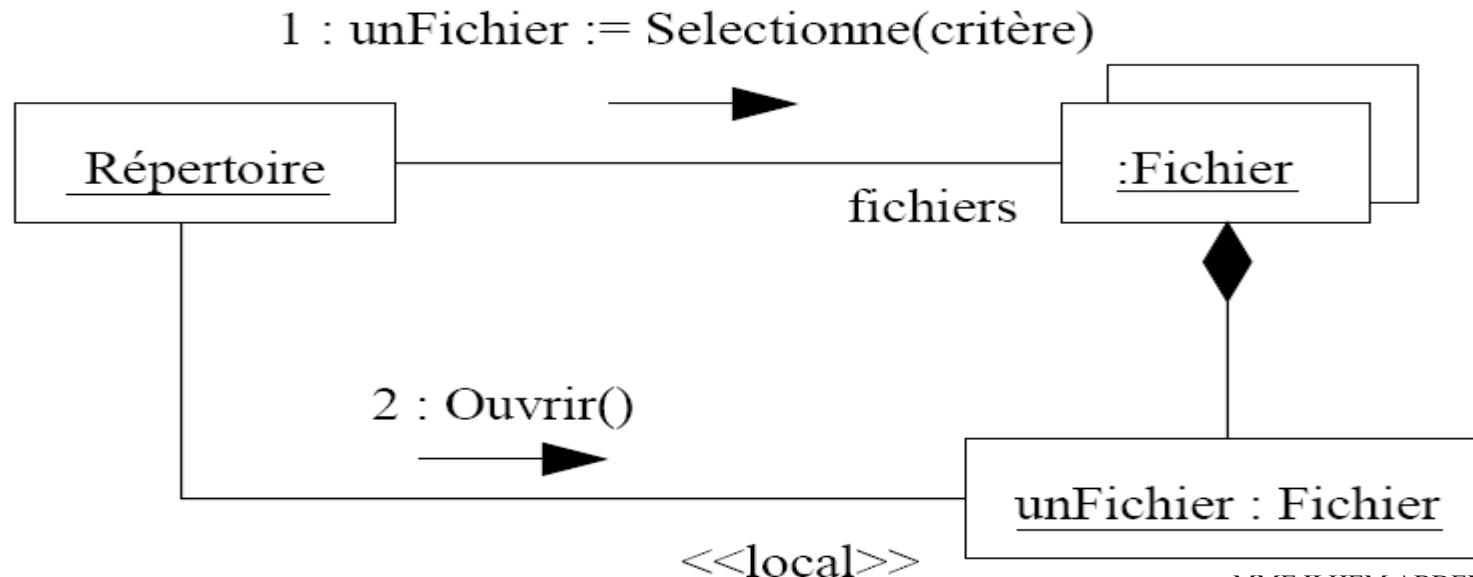


Message de création et message de destruction



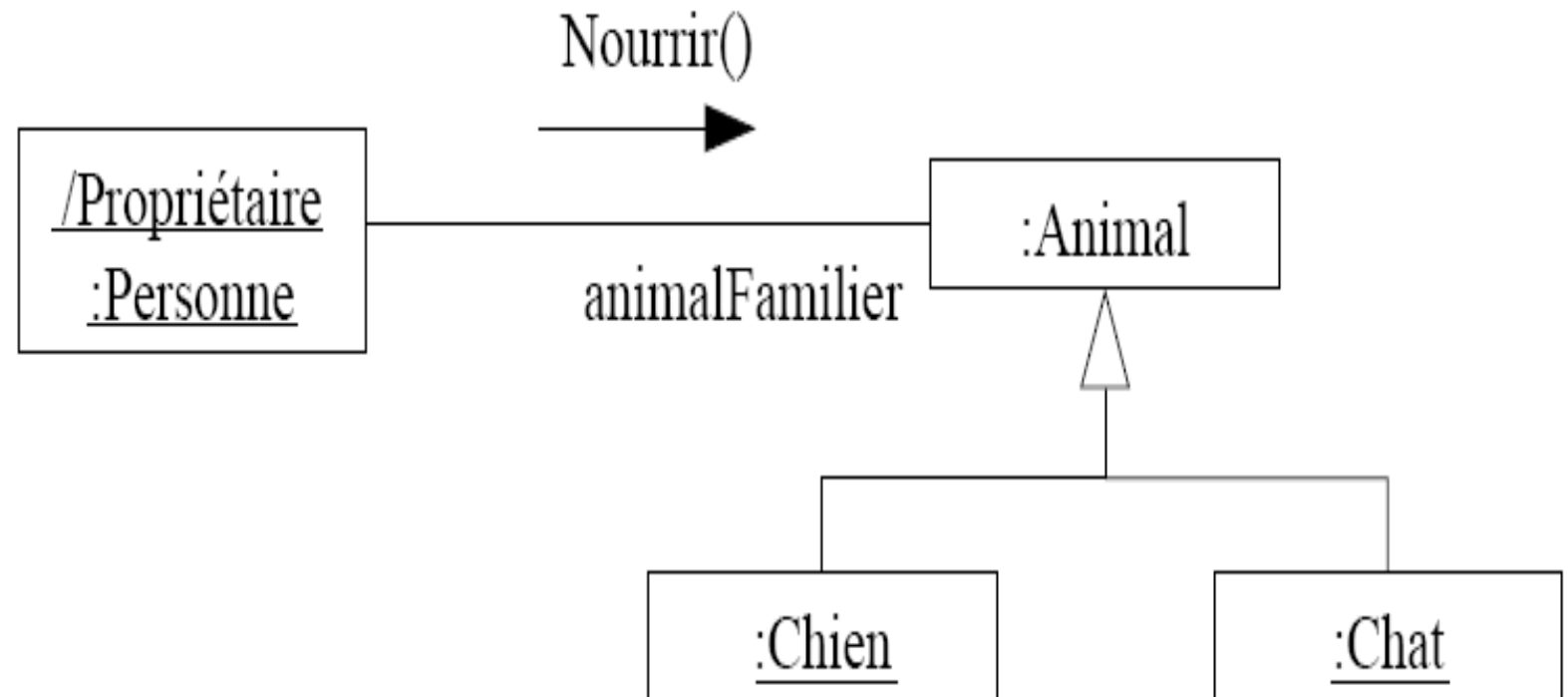
DISTINCTION AU NIVEAU TRAITEMENT

- Possibilité de représenter **un objet particulier appartenant à un groupe** afin de lui appliquer un message particulier
- *Représentation à l'aide d'une composition indiquant que l'objet fait partie de l'ensemble d'objets.*



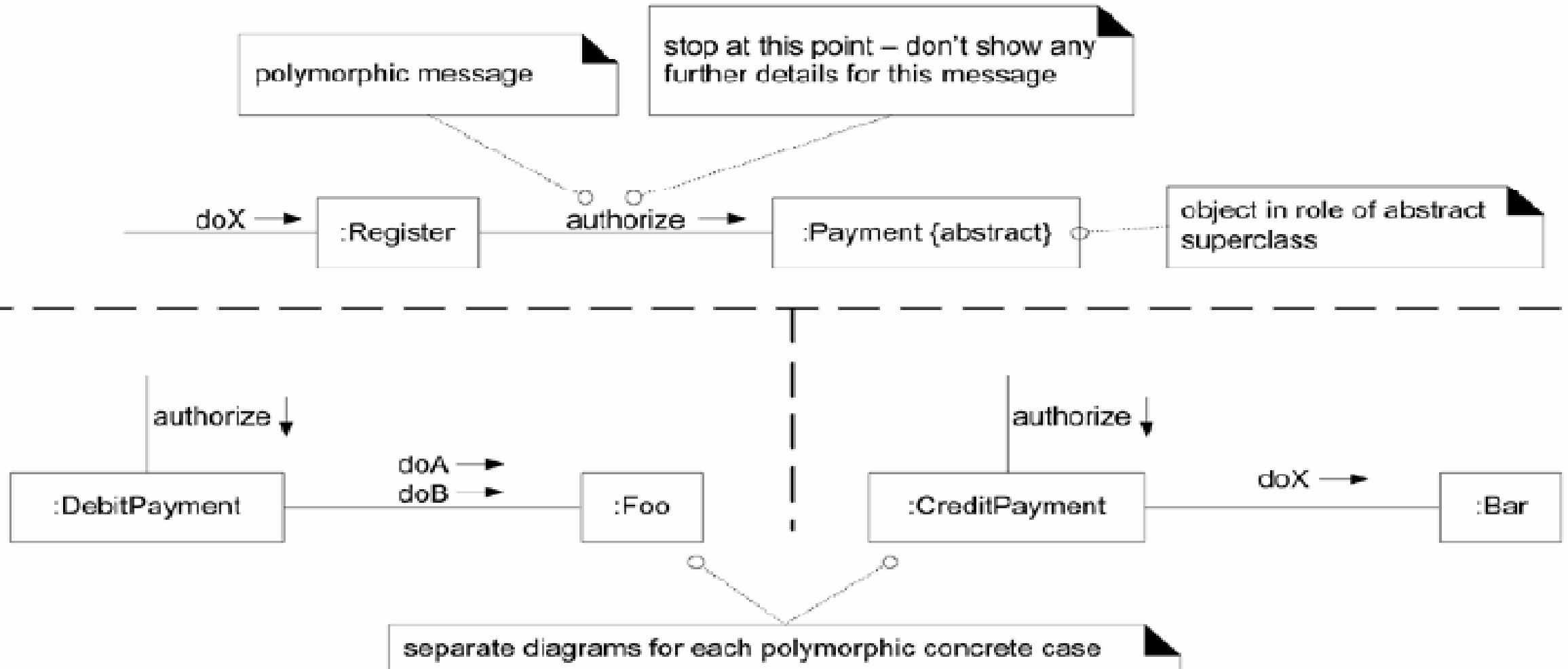
CLASSES ABSTRAITES ET INTERFACE

- Figurent dans les diagrammes de collaboration.
- *Elles représentent des informations complémentaires : liens polymorphes, réalisation d'interfaces, ..*

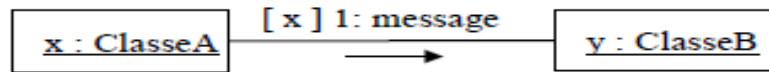


MODÉLISATION MESSAGE

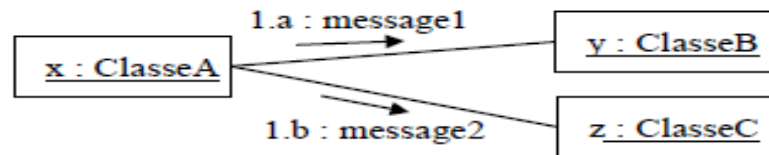
DO NOT MODEL IT



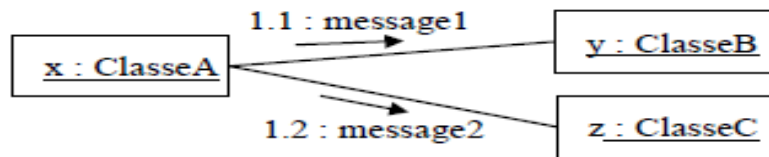
RÉSUMÉ – TYPES DE MESSAGES



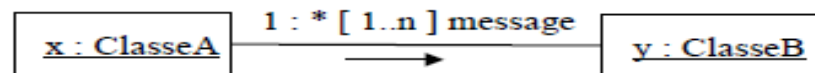
message soumis à la condition x



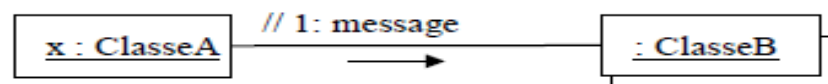
message1 et message 2 en parallèle



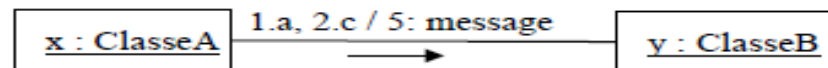
message1.1 et message1.2 envoyés successivement vers y et z



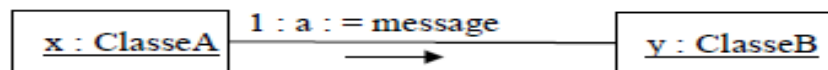
message envoyé n fois



message envoyé en parallèle à plusieurs instances de la classe B



message envoyé à y lorsque 1.a et 2.c sont satisfaits (synchronisation) : Voir diag. d'activités



a récupère la valeur renvoyée par l'exécution du message (récupération d'un résultat)

QUAND UTILISER LE DIAGRAMME DE COMMUNICATION ?

Phase de cadrage

Début de projet pour clarifier le domaine d'étude.

1. Décrire l'interaction des objets entre eux
2. Illustrer les scénarios des use cases
3. Valider les choix d'analyse et de conception (prototypage)
4. Aider au raffinement des diagrammes de classes de conception

Pour représenter une collaboration entre le domaine d'étude et les partenaires car il permet de déterminer : Les flux entrants et sortants, Les acteurs externes, et les domaines connexes

UTILITÉ DES DIAGRAMMES DE COLLABORATION

Documentation conceptuelle :

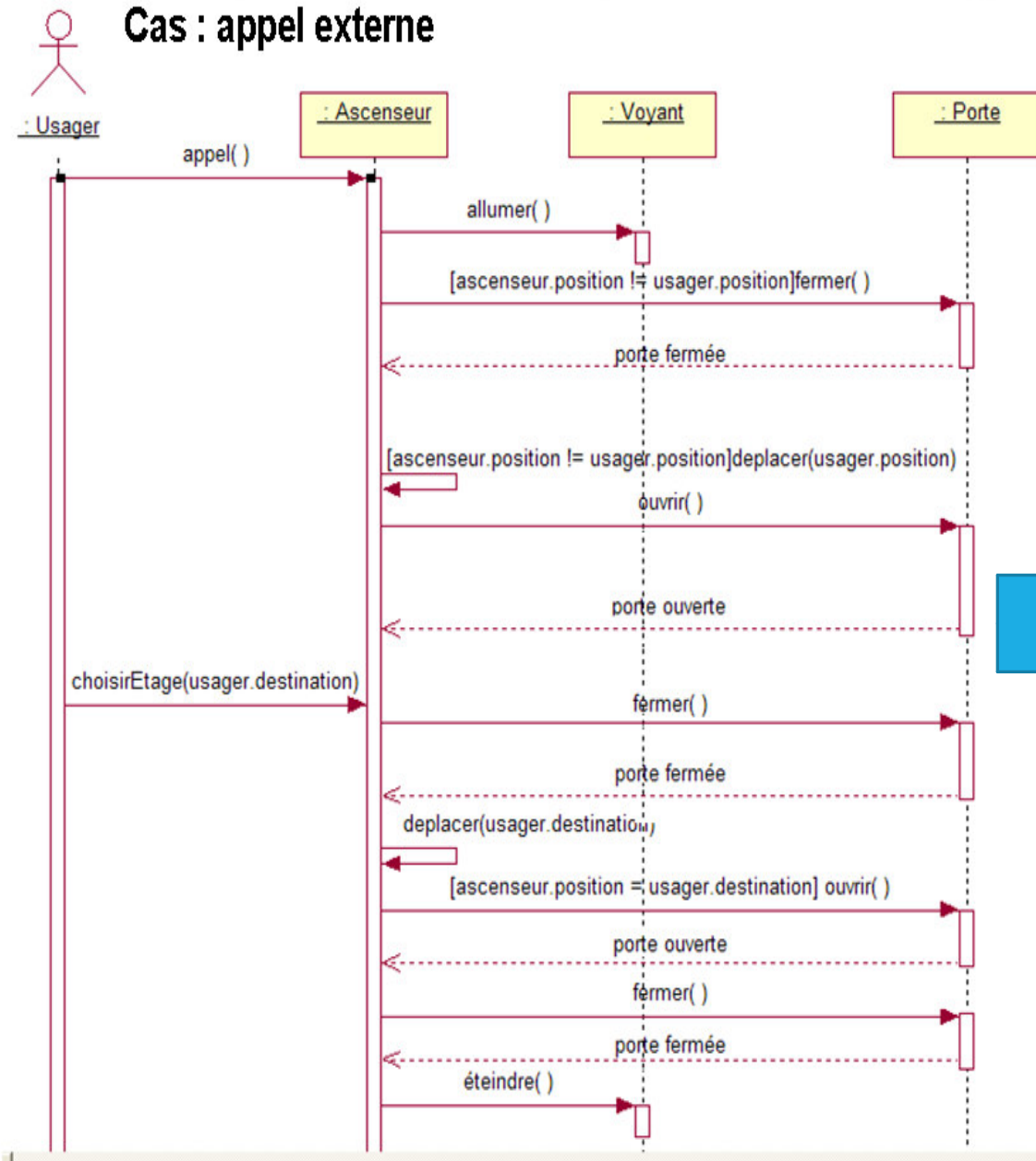
- décrit le **comportement** de classes et d'opérations.
- le **plus adapté** quand une opération fait interagir de **nombreux** objets.

Documentation des cas d'utilisation :

- description permettant de réaliser les cas d'utilisation.
- décrit le comportement du système pour chacun des scénarios accompagnant les cas d'utilisation.
- Facilite la rédaction des diagrammes des classes, des diagrammes état-transition, ...

Exemple de l'ascenseur (niveau cahier charges)

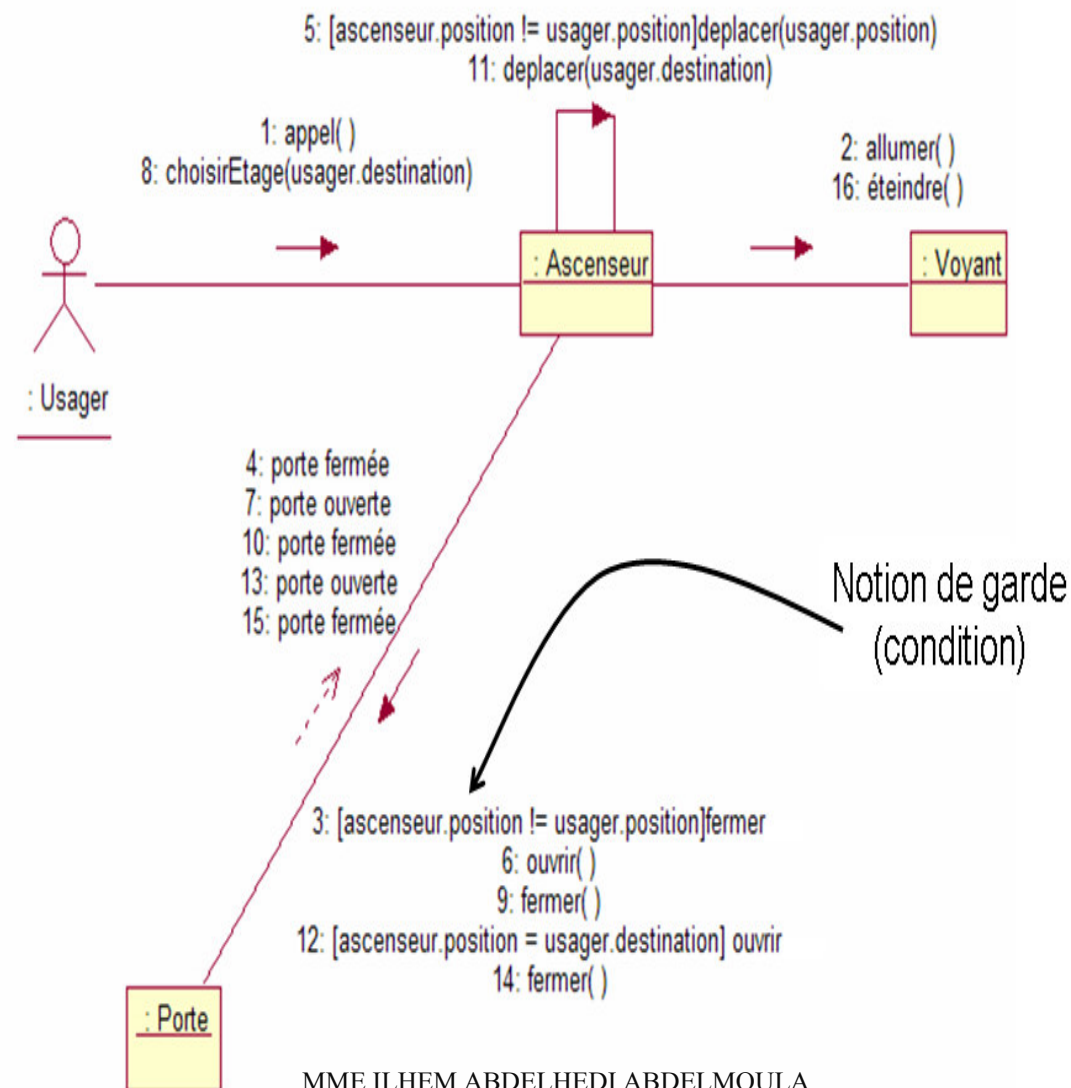
Cas : appel externe



6/28/2020

Exemple : ascenseur (niveau cahier des charges : utilisateur + classes de base)

Cas : appel externe de l'ascenseur



MME ILHEM ABDELHEDI ABDELMOULA

EXEMPLE

Diagramme de classes

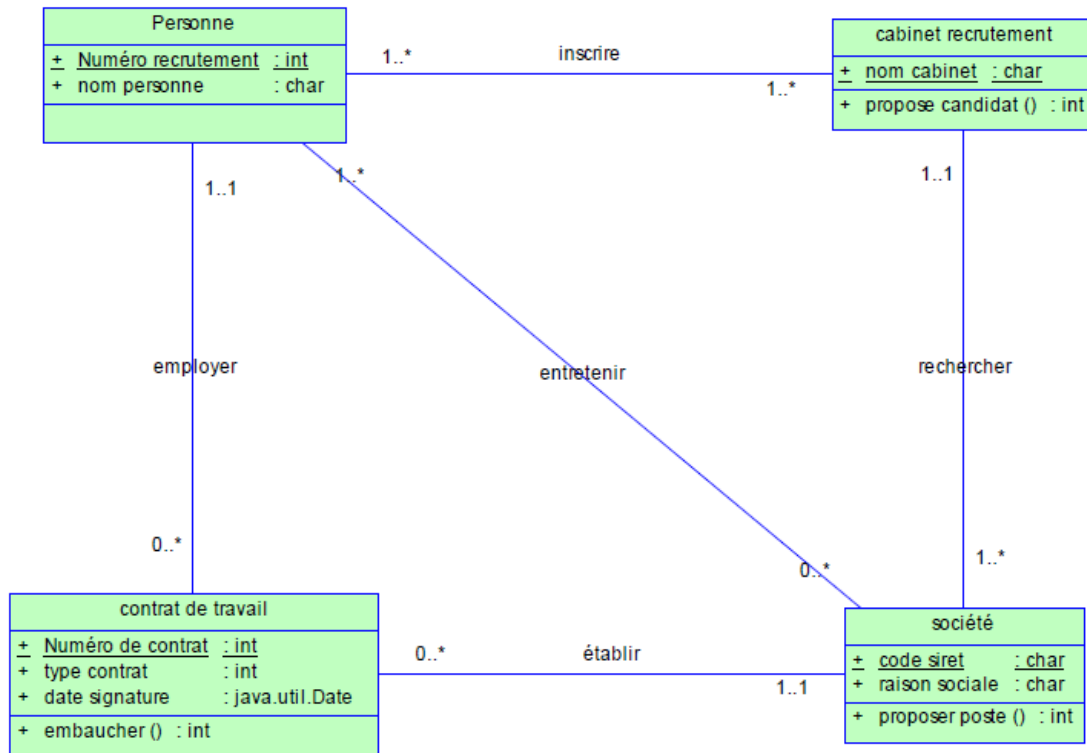
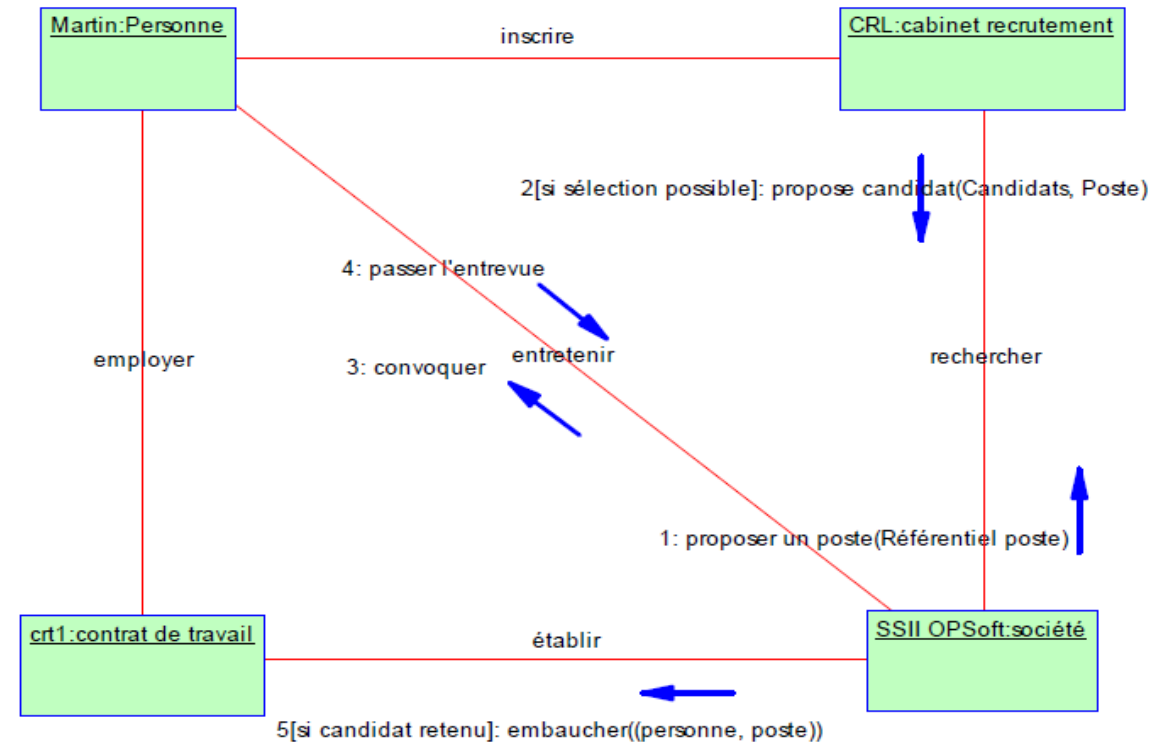


Diagramme de collaboration



EXERCICE

Le client demande au vendeur des renseignements sur les compositions florales.

Le vendeur lui fournit toutes les informations nécessaires. Le client commande alors la composition de son choix et le vendeur émet le bon de fabrication qu'il transmet à son ouvrier fleuriste.

Le vendeur édite ensuite la facture correspondante.

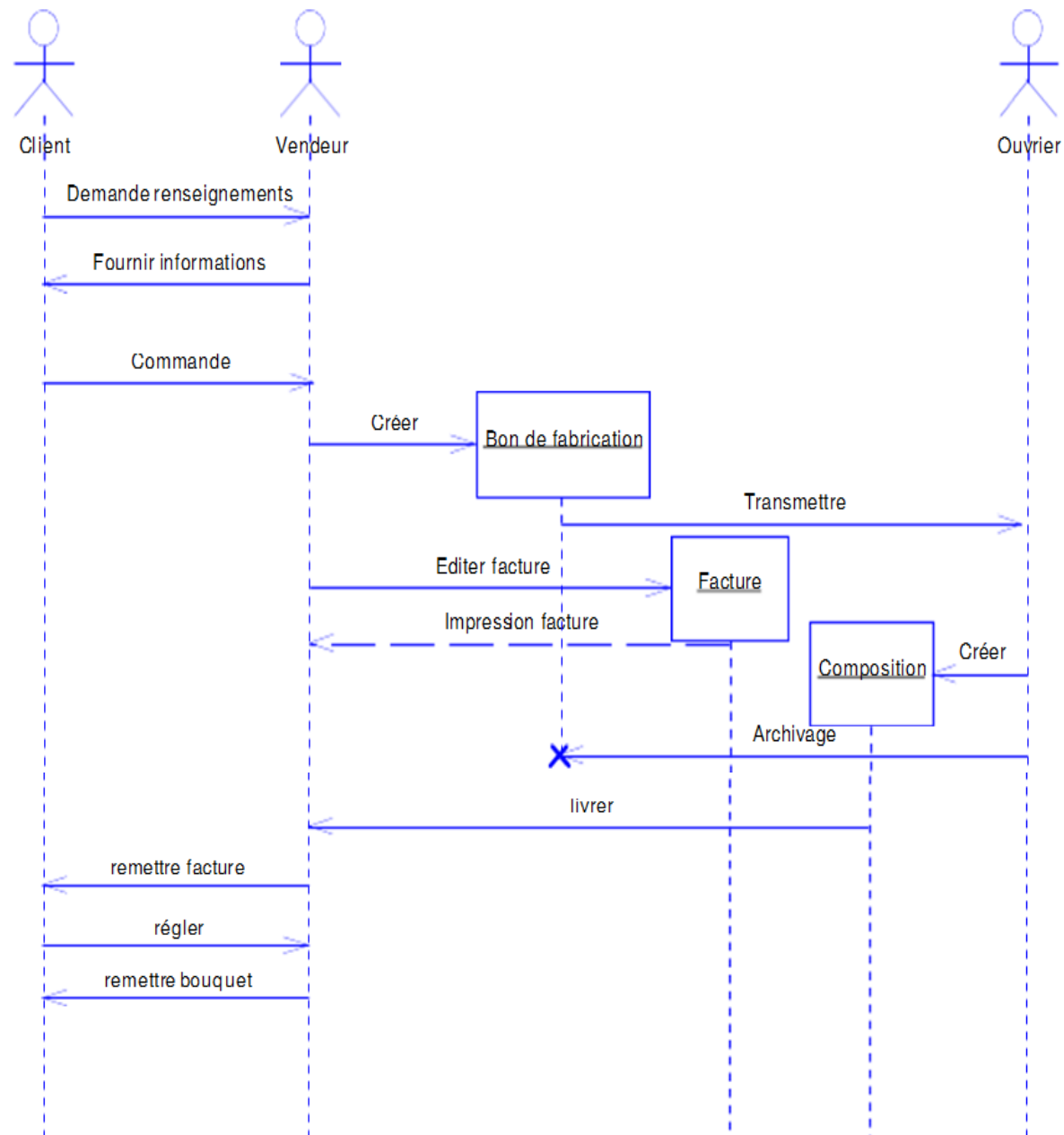
L'ouvrier fleuriste crée la composition puis archive le bon de fabrication

Il remet alors la composition au vendeur

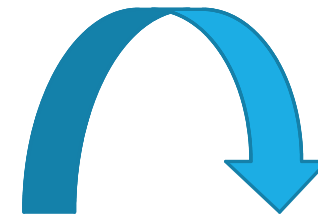
La facture est remise au client pour règlement une fois le bouquet réalisé

Une fois la facture réglée, le client récupère sa composition et quitte le magasin.

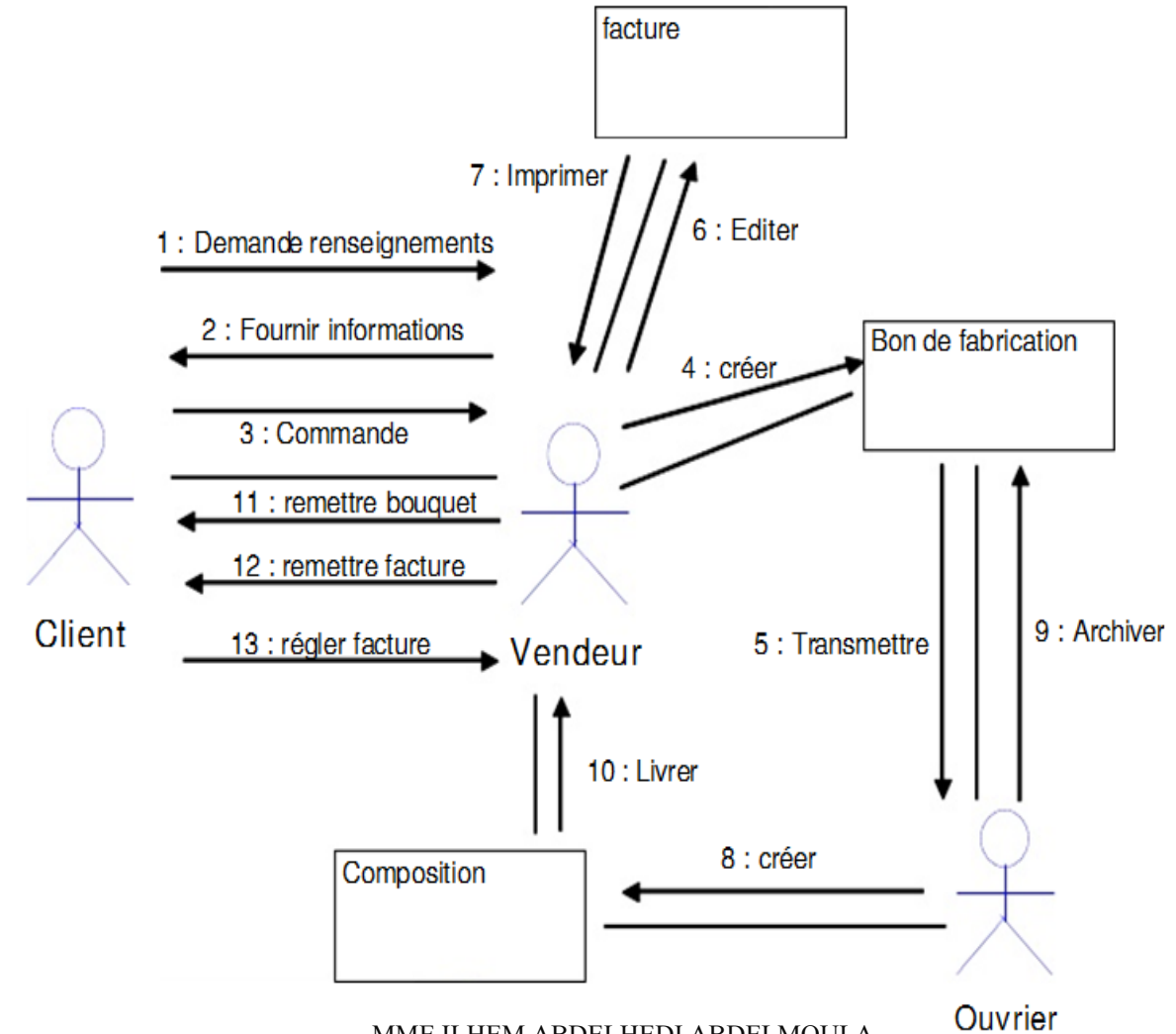
Modéliser cette situation à l'aide d'un diagramme de séquence



6/28/2020

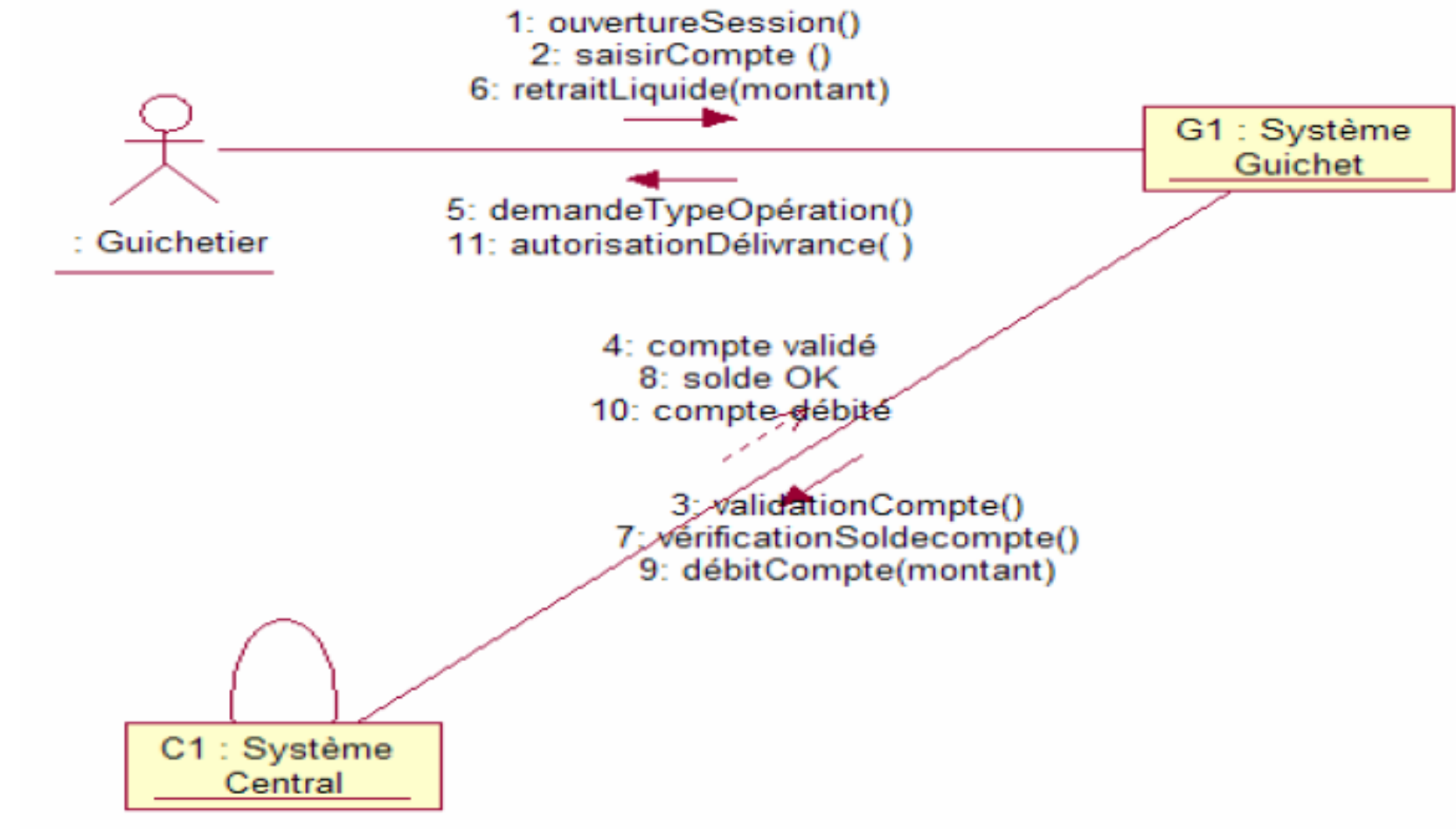


SOLUTION



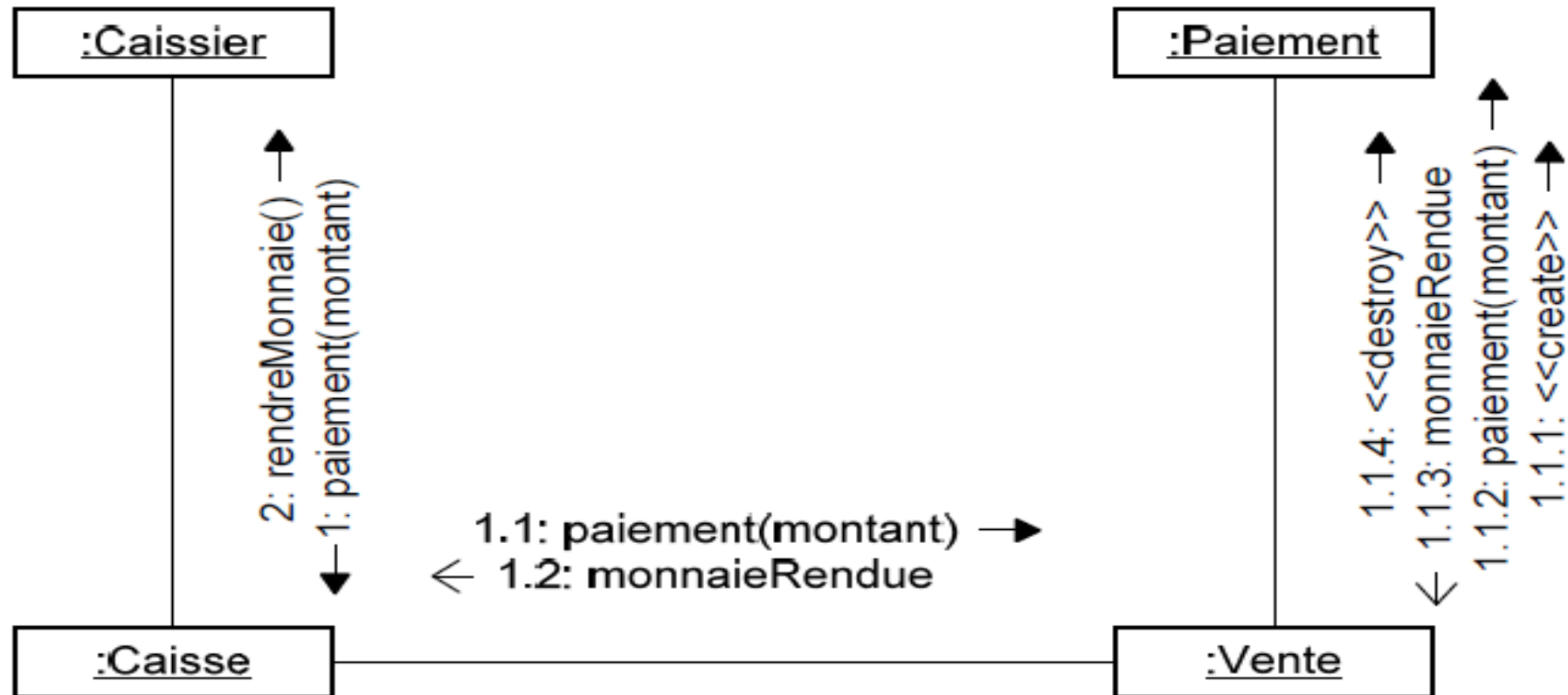
MME ILHEM ABDELHEDI ABDELMOULA

EXEMPLE ■ Retrait en espèce à la banque



EXEMPLE

■ Cas d'utilisation « paiement en espèce »



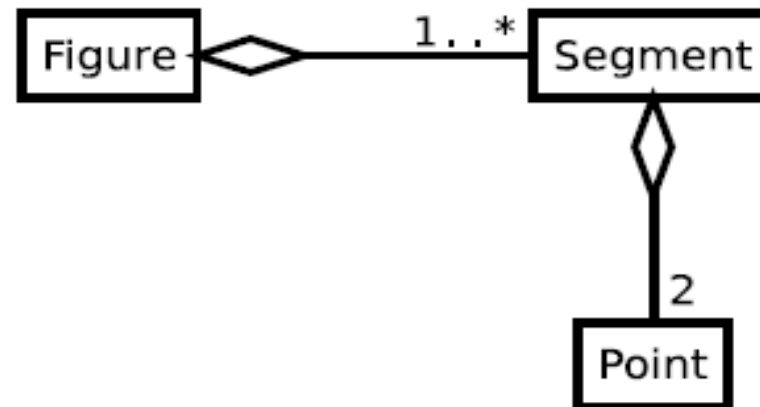
DIFFÉRENCES

Diagrammes de séquences vs. Diagramme de collaborations

- Diagrammes de séquences
 - 👉 Montrer clairement l'ordonnancement temporel des messages
 - 👉 Consommer des espaces (vers la droite)
- Diagrammes de collaboration
 - 👉 Préférable lorsque l'interaction est déduite à partir du diagramme de classes
 - 👉 Consommer moins d'espaces
 - 👉 Difficile de voir la séquence des messages

EXERCICE

Soit le diagramme de classes suivant :



Donner les diagrammes de collaboration représentant l'affichage d'une figure demandé par un utilisateur.

SOLUTION

