

**Master mention Informatique
Spécialité ISI**

**Génie Logiciel et modélisation
M1 / 177UD01**

**C4 – Les diagrammes d'objets et les
diagrammes d'interaction**

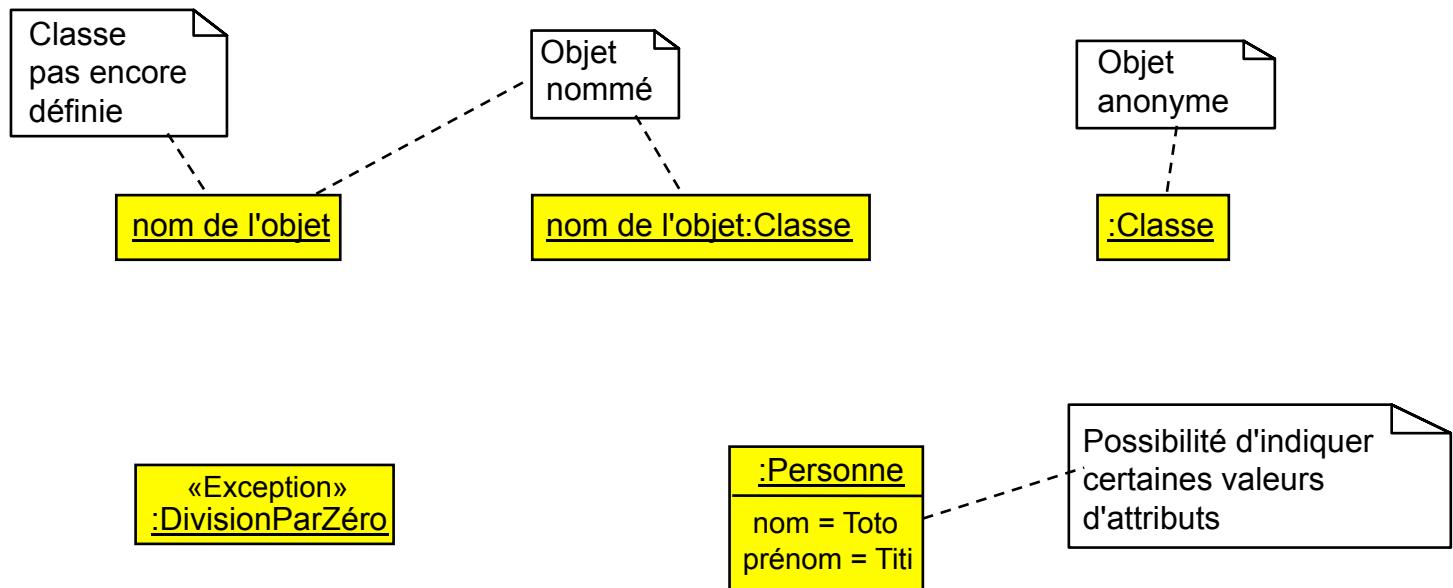
Claudine Piau-Toffolon

Plan du cours

- Les diagrammes d'objets
- Les diagrammes d'interaction
 - Définition
 - Notion de message
 - Notion de signal
 - Les diagrammes de communication
 - Les diagrammes de séquence
- Exemples

Diagrammes d'objets - Définition

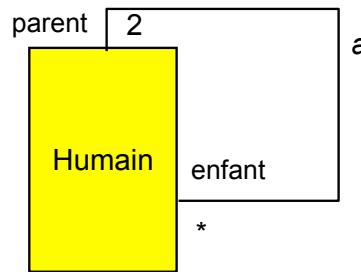
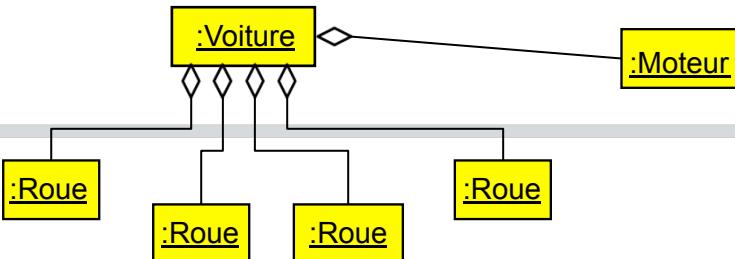
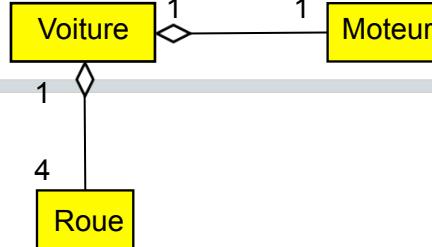
- Ce sont des diagrammes statiques
- Ils sont composés d'objets et des liens entre ces objets
- Ils montrent l'état du système à un instant donné
- Ils s'utilisent pour expliciter et pour la compréhension des structures complexes
- Représentation UML des objets :



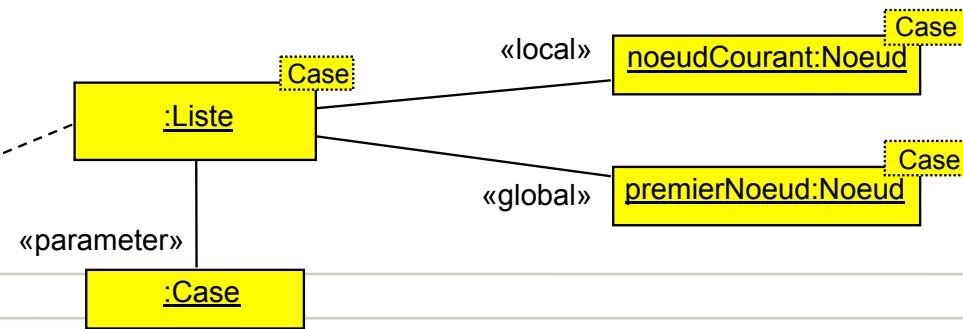
Diagrammes d'objets – Représentation (1/2)

- Représentation des liens :
 - Les liens sont les instances des associations entre les classes des objets considérés à un instant donné
 - Il est possible de reporter les décorations figurant sur les associations : nom, rôles des extrémités, l'agrégation, la composition, la qualification et la navigation, **mais pas la multiplicité**
 - Les stéréotypes des extrémités des liens définis dans UML :
 - <<local>> : l'instance de l'extrémité est une variable locale d'une opération
 - <<global>> : l'instance de l'extrémité est visible par tous les objets du diagramme
 - <<parameter>> : l'instance de l'extrémité est un paramètre d'une opération
 - <<self>> : lien vers l'objet lui-même

Diagrammes d'objets – Représentation (2/2)



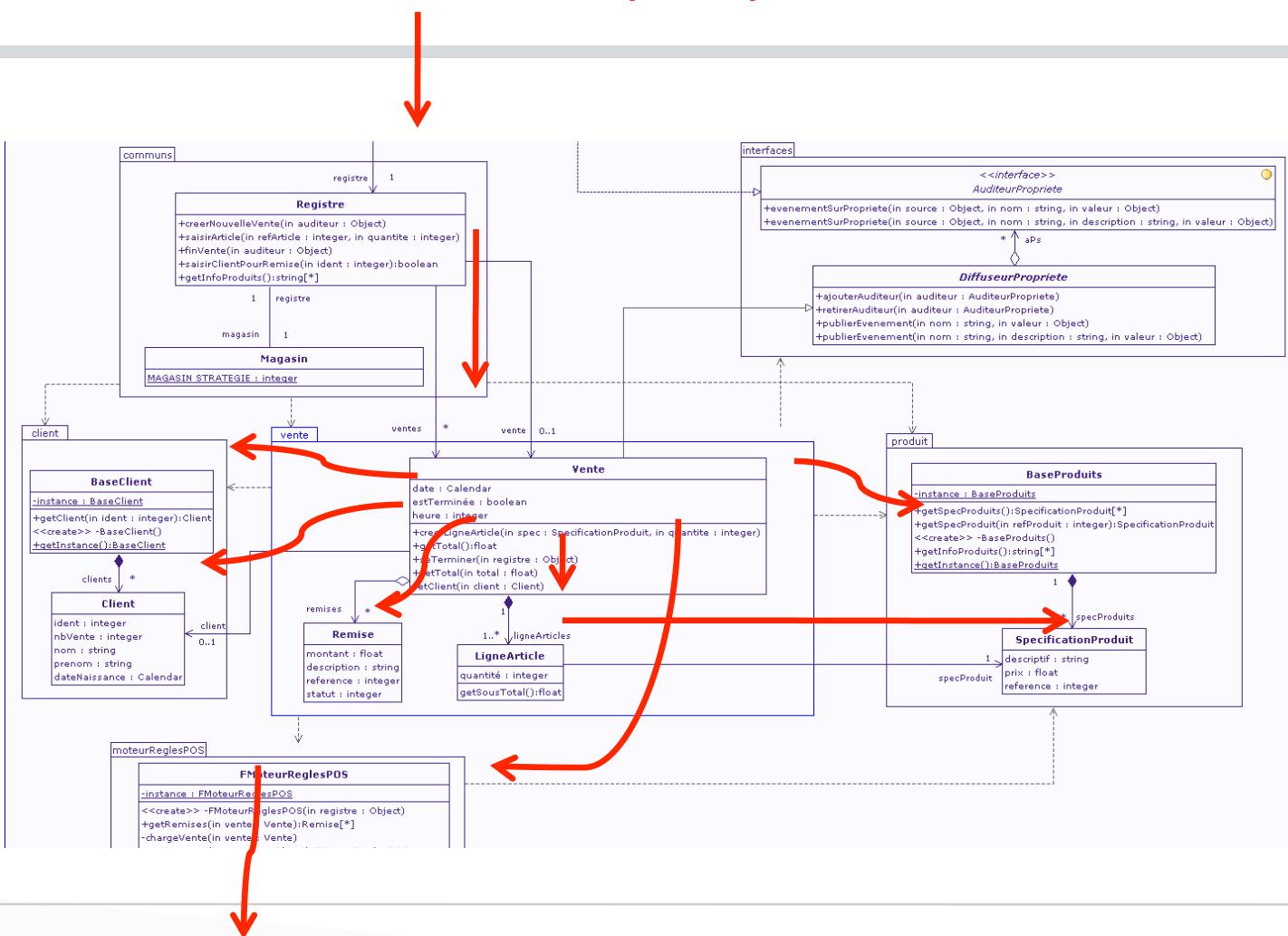
On est dans la opération
`chercherElement(element : Type)`
de la classe `Liste<Case>`



Les diagrammes d'interactions – Définition (1/2)

- Les diagrammes d'interactions sont des diagrammes dynamiques
- Ils permettent de modéliser les **collaborations** d'objets :
 - Collaboration : échangent de messages entre objets de classes de packages différents permettant de fournir un service ou une fonctionnalité.
 - En conception, une collaboration peut traverser différentes couches de l'architecture du logiciel.
 - En analyse, une collaboration correspond aux interactions entre les acteurs et le système
- Deux types de diagrammes d'interactions :
 - Ils sont équivalents du point de vue sémantique mais ne montrent pas le même aspect de la collaboration.
 - Les *diagrammes de communication* permettent de mettre l'accent sur l'organisation structurelle des objets.
 - Les *diagrammes de séquence* permettent de mettre l'accent sur l'ordre chronologique des messages.

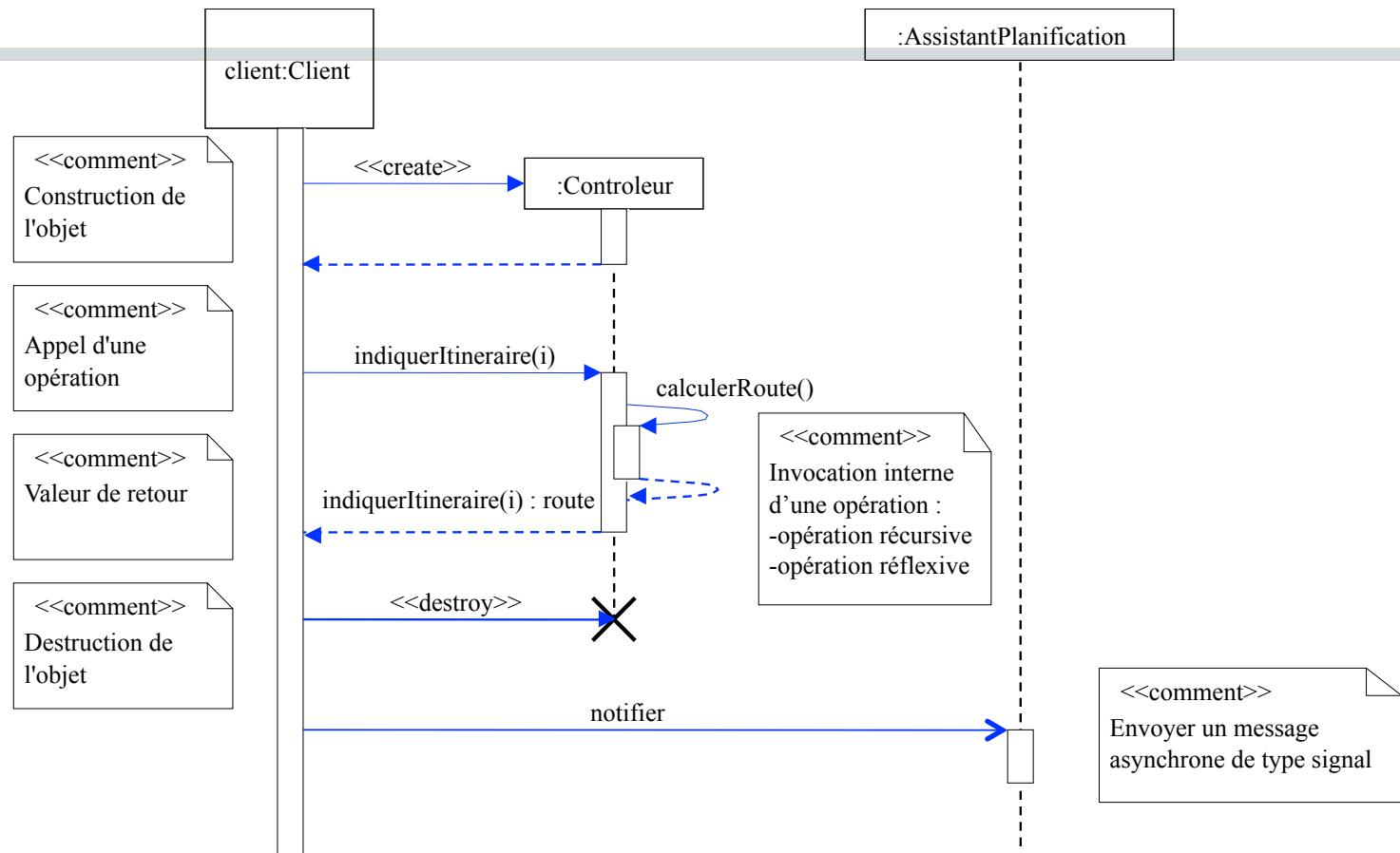
Diagrammes d'interactions – Définition (2/2)



Diagrammes d'interactions – Notion de messages (1/2)

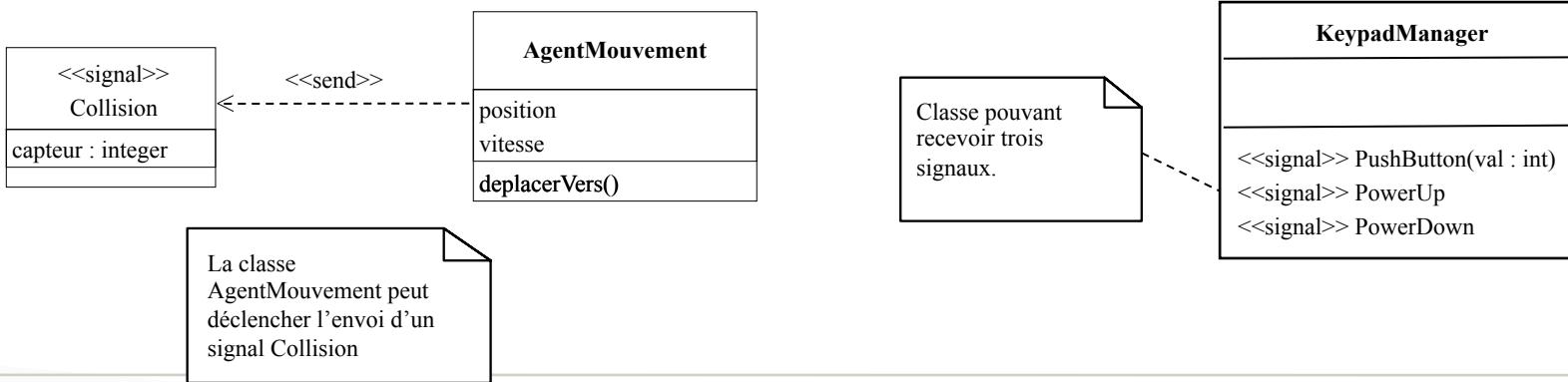
- Notion de messages
 - Un message est la représentation d'une communication entre objets.
 - Il transporte des informations et a pour but de déclencher une activité.
 - Un message est soit synchrone soit asynchrone
 - Les types de message modélisables en UML :
 - call : invoque une opération sur un objet récepteur
 - return : renvoie une valeur à l'objet émetteur
 - send : envoie un signal à un objet
 - create : crée un objet
 - destroy : détruit un objet

Diagrammes d'interactions – Notion de messages (2/2)



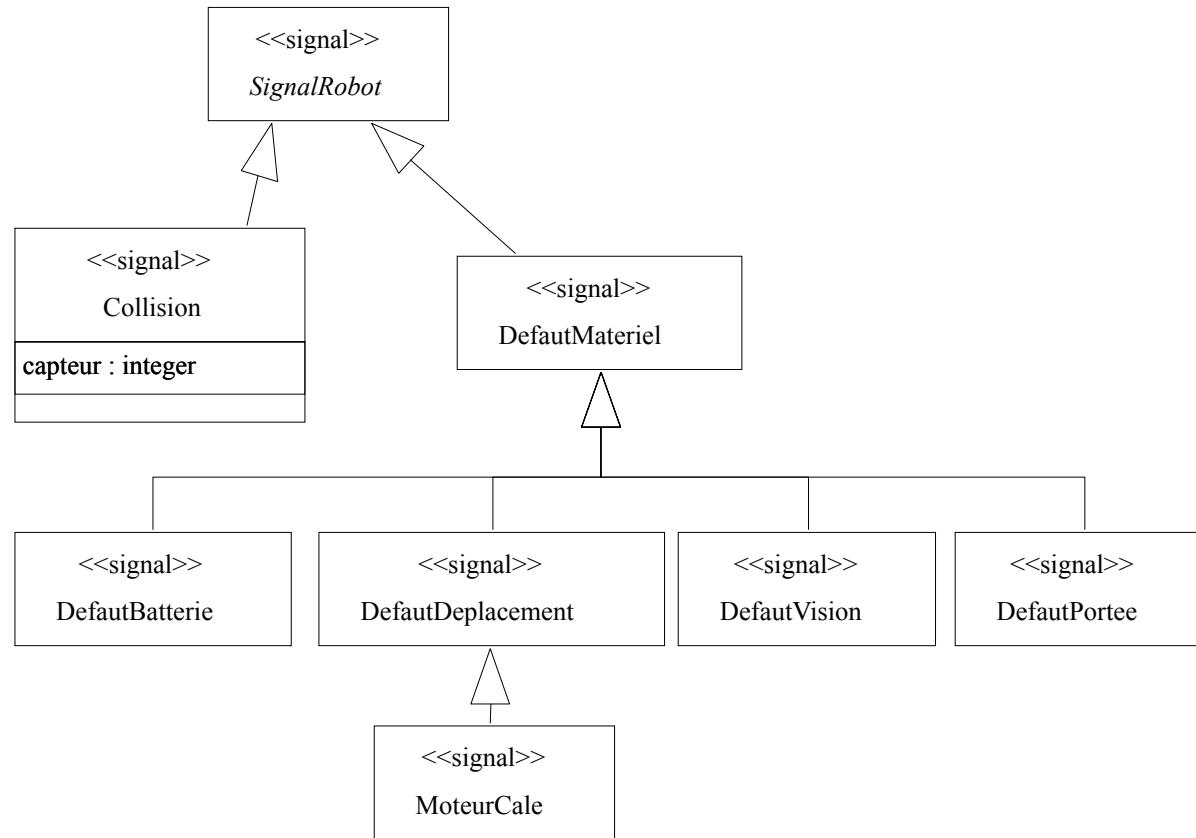
Notion de signal (1/2)

- Définition : un signal représente un objet communiqué de manière asynchrone à un autre objet.
- L'envoie d'un signal provient généralement de l'exécution d'une opération mais peut provenir de l'extérieur du système.
- Un signal est représenté par une classe (*avec des attributs correspondant aux arguments et des opérations d'accès aux attributs*) et peut faire partie d'un graphe d'héritage.
- Une classe peut définir les signaux que ses opérations peuvent envoyer et/ou recevoir.
- Représentation UML :



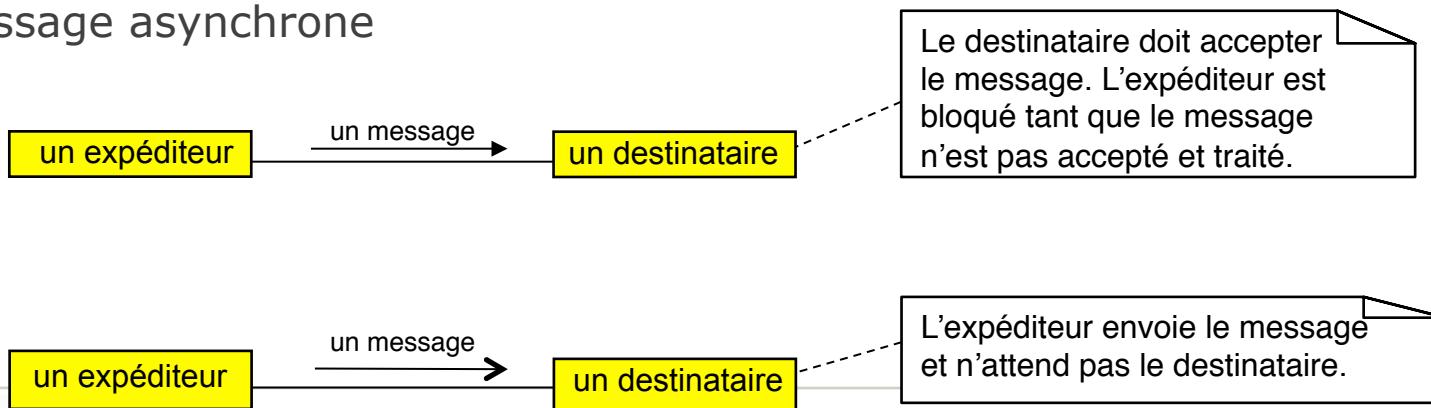
Notion de signal (2/2)

- Les signaux sont généralement hiérarchisés : cela permet de spécifier des objets polymorphes qu'un objet actif peut recevoir.



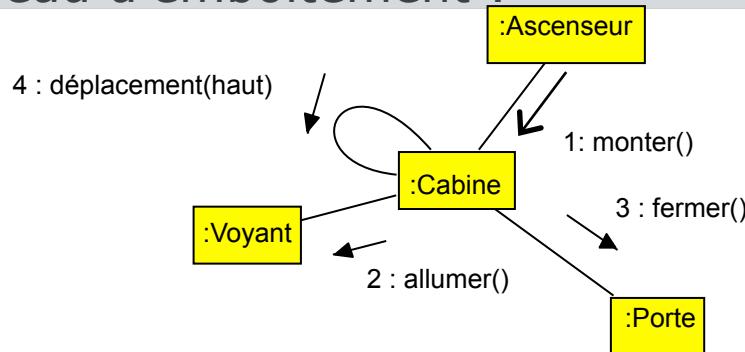
Diagr. de communication–Messages

- Les diagrammes de communication sont des diagrammes d'interactions
- Ils donnent en plus une vue structurelle du système
- Lorsqu'un objet envoie un message à un objet destinataire, il faut que ce message déclenche une activité définie par l'objet destinataire.
- Un message est figuré par une flèche vers le récepteur et un nom de message + des paramètres éventuels + une valeur de retour éventuelle.
- Il peut y avoir plusieurs messages échangés.
- Catégories d'envois de message :
 - Message synchrone
 - Message asynchrone

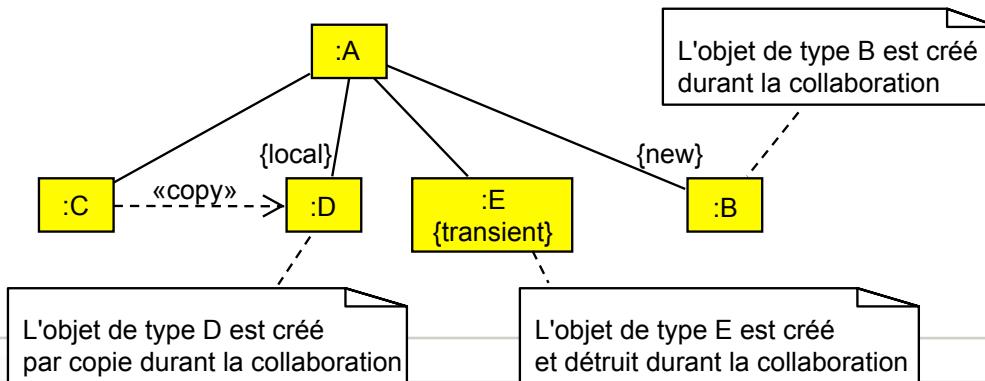


Diagr. de communication—Les interactions

- Les messages sont numérotés pour indiquer l'ordre des envois et/ou le niveau d'emboîtement :



- Il est possible d'indiquer la création et la destruction des objets et des liens durant une interaction par des contraintes sur le lien ou dans l'objet :

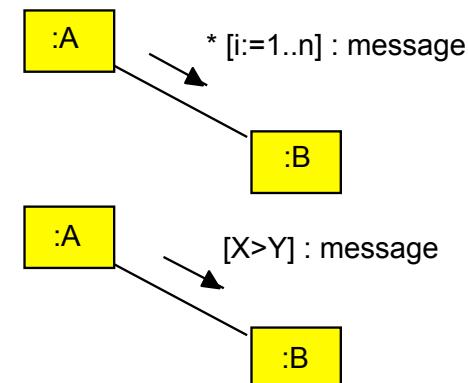
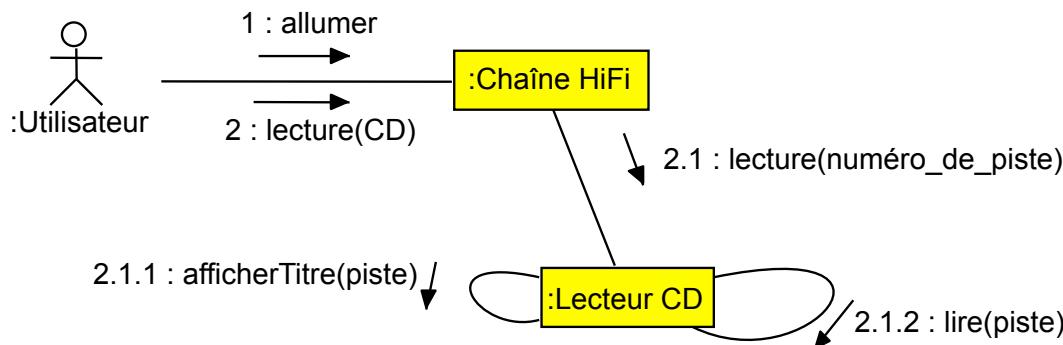


Diagr. de com. – Syntaxe détaillée (1/2)

- Un message suit la syntaxe suivante :

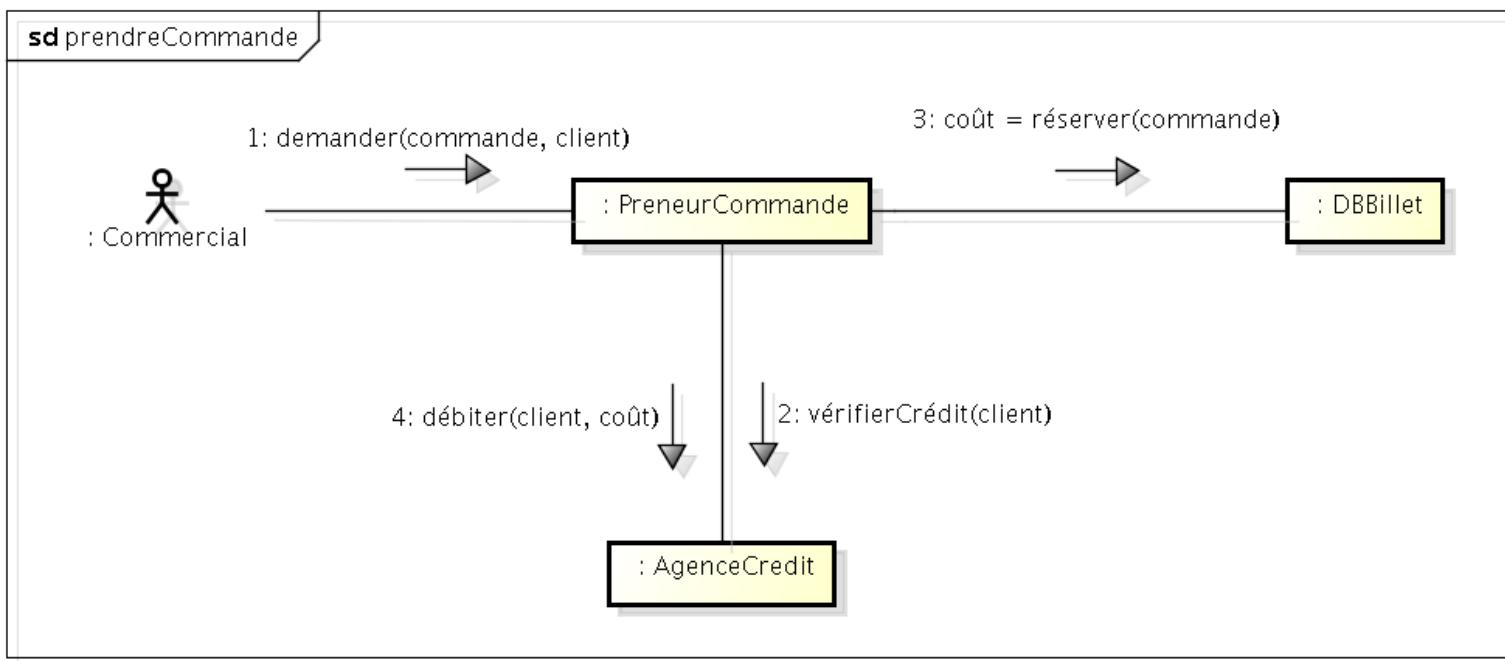
[séquence ':'] [attribut '='] nom_message ['(' liste_arguments ')'] [':' valeur_retour]

- La séquence :
 - Indique le niveau d'emboîtement de l'envoi du message
 - **Syntaxe** : suite de termes séparés par des points
 - séquence ::= rang [récurrence]
 - rang ::= (entier | nom_de_flot_d'exécution) ['' rang]
 - récurrence ::= '*' [' clause_d'itération ']
- L'envoi en parallèle s'écrit *||



Diagr. de com. – Syntaxe détaillée (2/2)

- L'attribut : nom d'un attribut pour stocker la valeur renvoyée par le message qui est utilisable comme paramètre dans les autres messages (variable locale au diagramme)
- Les arguments : liste des paramètres du message séparés par des virgules
- La valeur de retour du message



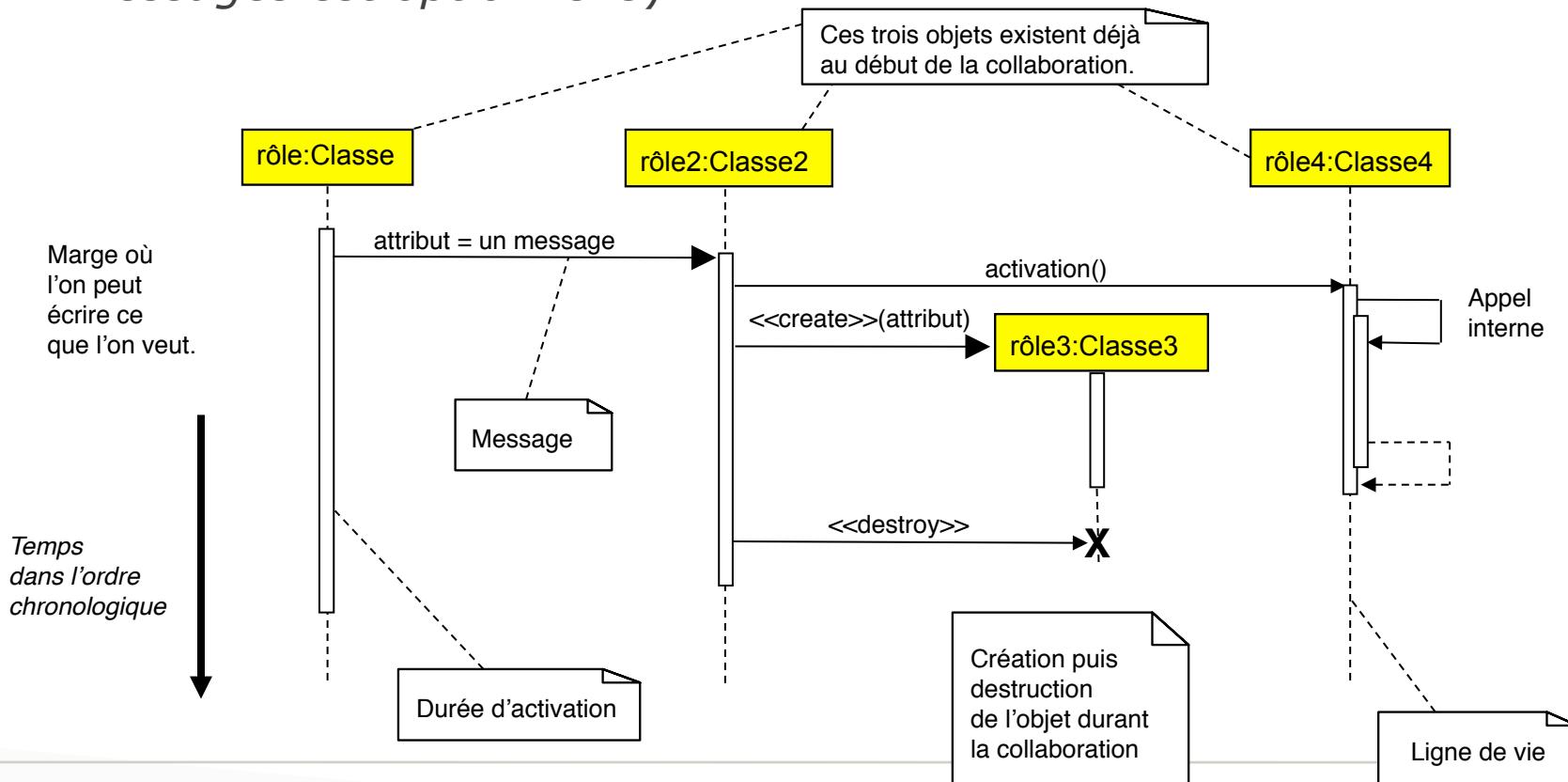
Diagrammes de séquence -

Définition

- Les diagrammes de séquence mettent l'accent sur la modélisation de la temporalité des interactions.
- Lorsqu'il y a peu d'objets et beaucoup de messages échangés, on préférera utiliser ce type de diagramme à la place des diagrammes de communication.
- Utilisations des diagrammes de séquences :
 - **En analyse** (documentation des cas d'utilisation) : ils modélisent les interactions qui surviennent dans les scénarios d'utilisation entre les acteurs et le système (et aussi des concepts métiers).
 - En conception : ils modélisent les interactions entre objets et la répartition du flot d'exécution (pour l'exécution d'une opération par exemple et la délégation aux autres objets).

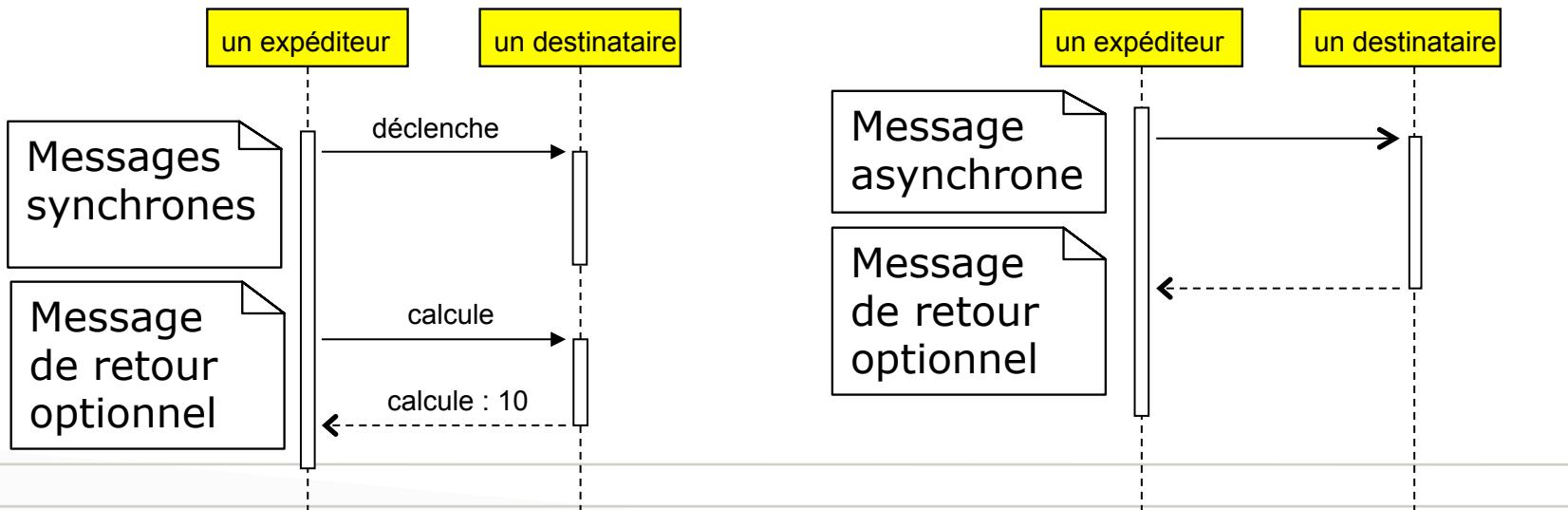
Diagr. de séquence – Représentation (1/2)

- Un objet est représenté avec une ligne de vie.
- Syntaxe des messages échangés : idem que celle des diagrammes de communication (*la numérotation des messages est optionnelle*).



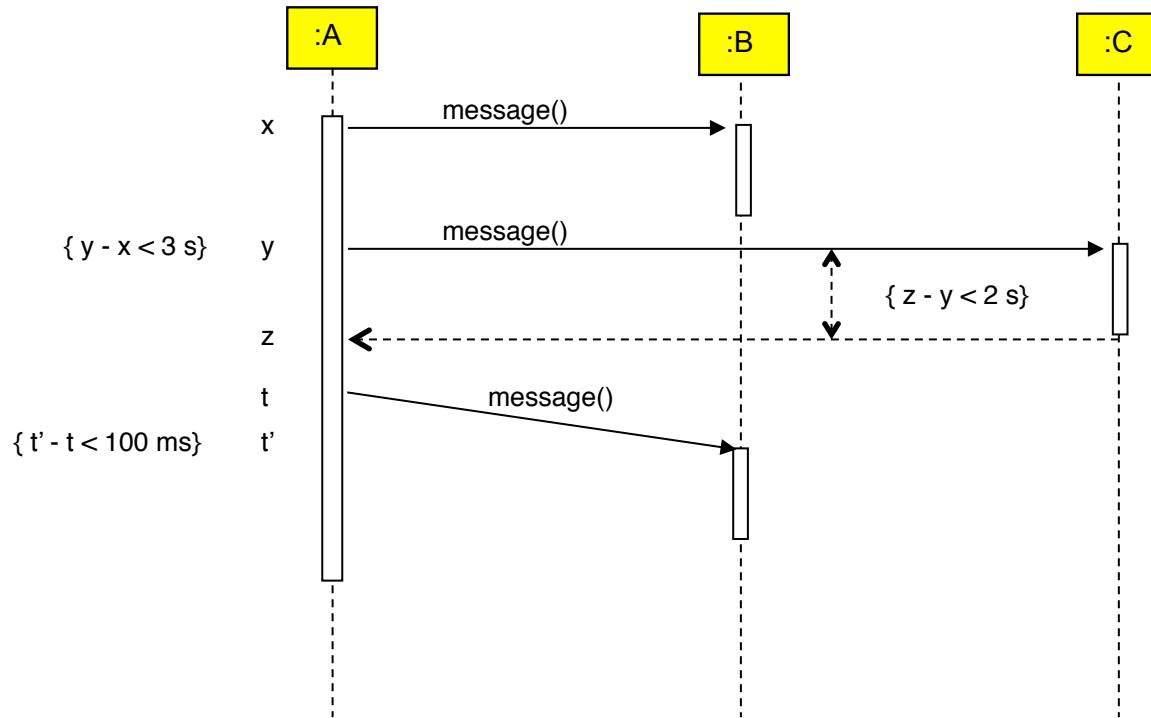
Diagr. de séquence – Représentation (2/2)

- Représentation des types de messages :
 - Quelque soit le type de message, le retour à l'objet appelant est implicite à la fin de l'activation de l'objet appelé mais on peut ajouter une flèche symbolisant la fin de l'activation.
 - Message synchrone :
 - Flèche pleine : la valeur de retour éventuelle *peut être* indiquée sur une flèche vide pointillée de fin d'activation.
 - Message asynchrone :
 - Flèche vide : la fin d'activation *peut être* indiquée par une flèche vide pointillée.

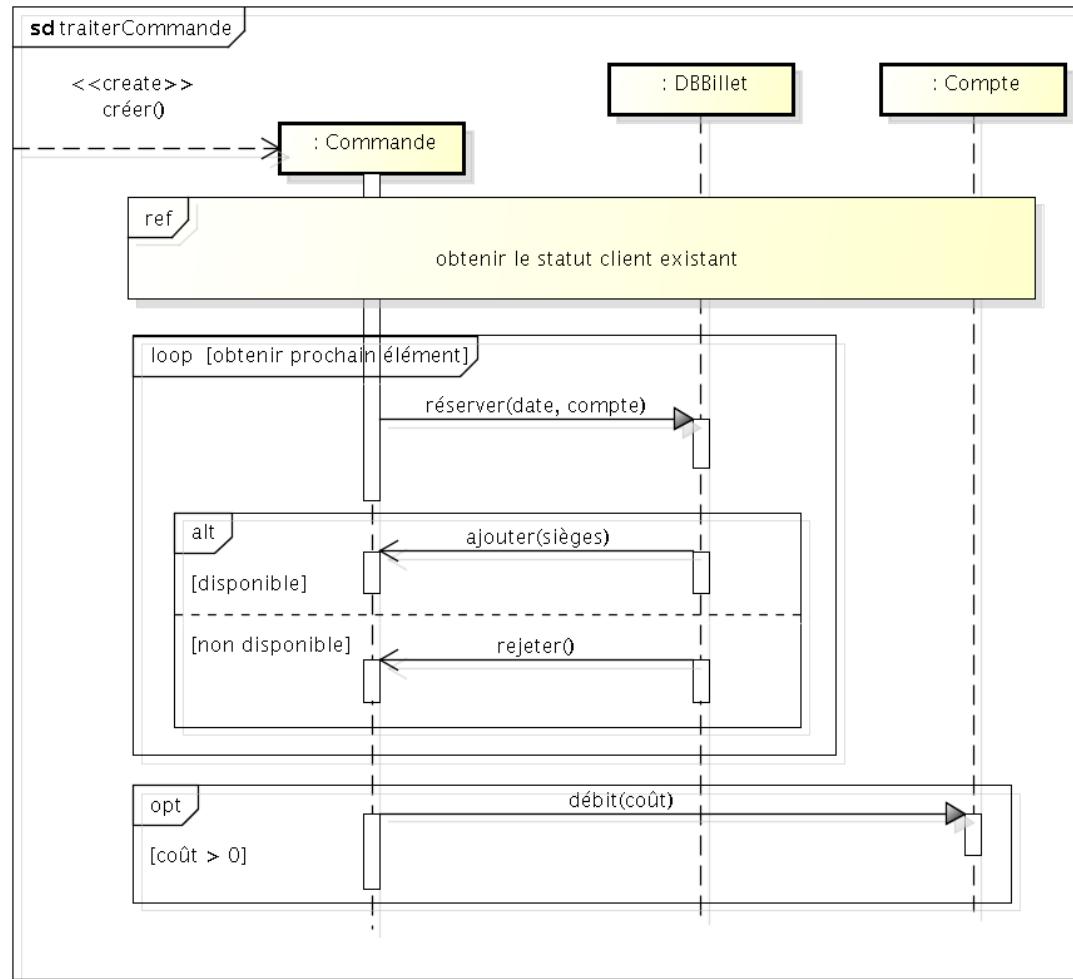


Diagr. de séquence – Contraintes temporelles

- Les lignes de vie peuvent être graduées temporellement.
- Les contraintes temporelles peuvent être exprimées en nommant l'instant d'émission des messages.
- La flèche d'un message peut être oblique afin d'exprimer le temps de propagation d'un message.

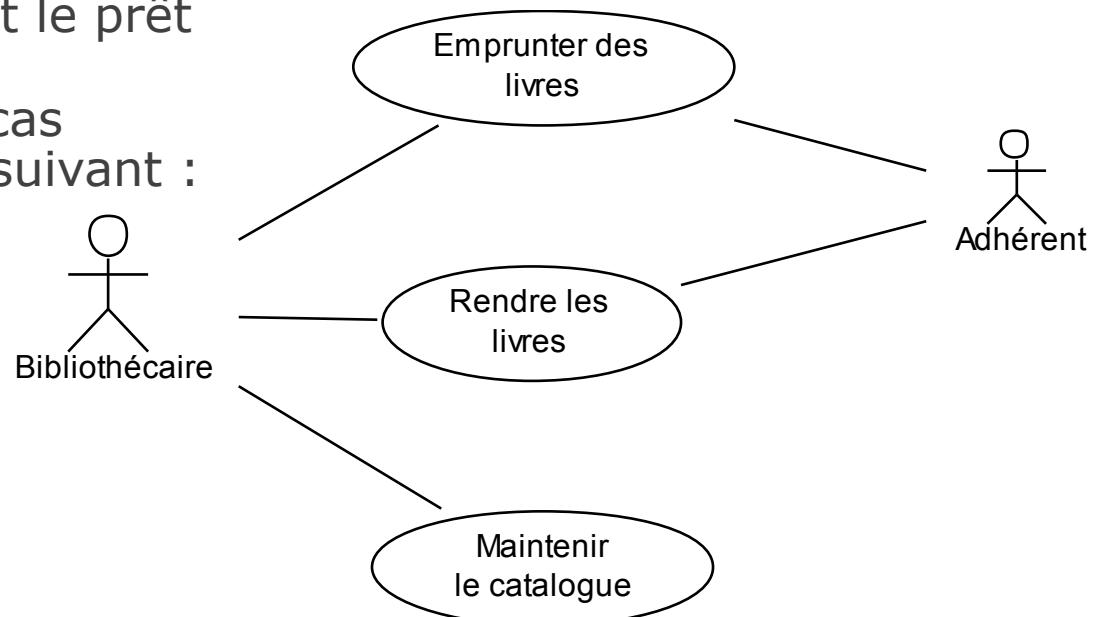


Diagr. de séquence – Structures de contrôle



Exemple de diagr. de collaboration (1/5)

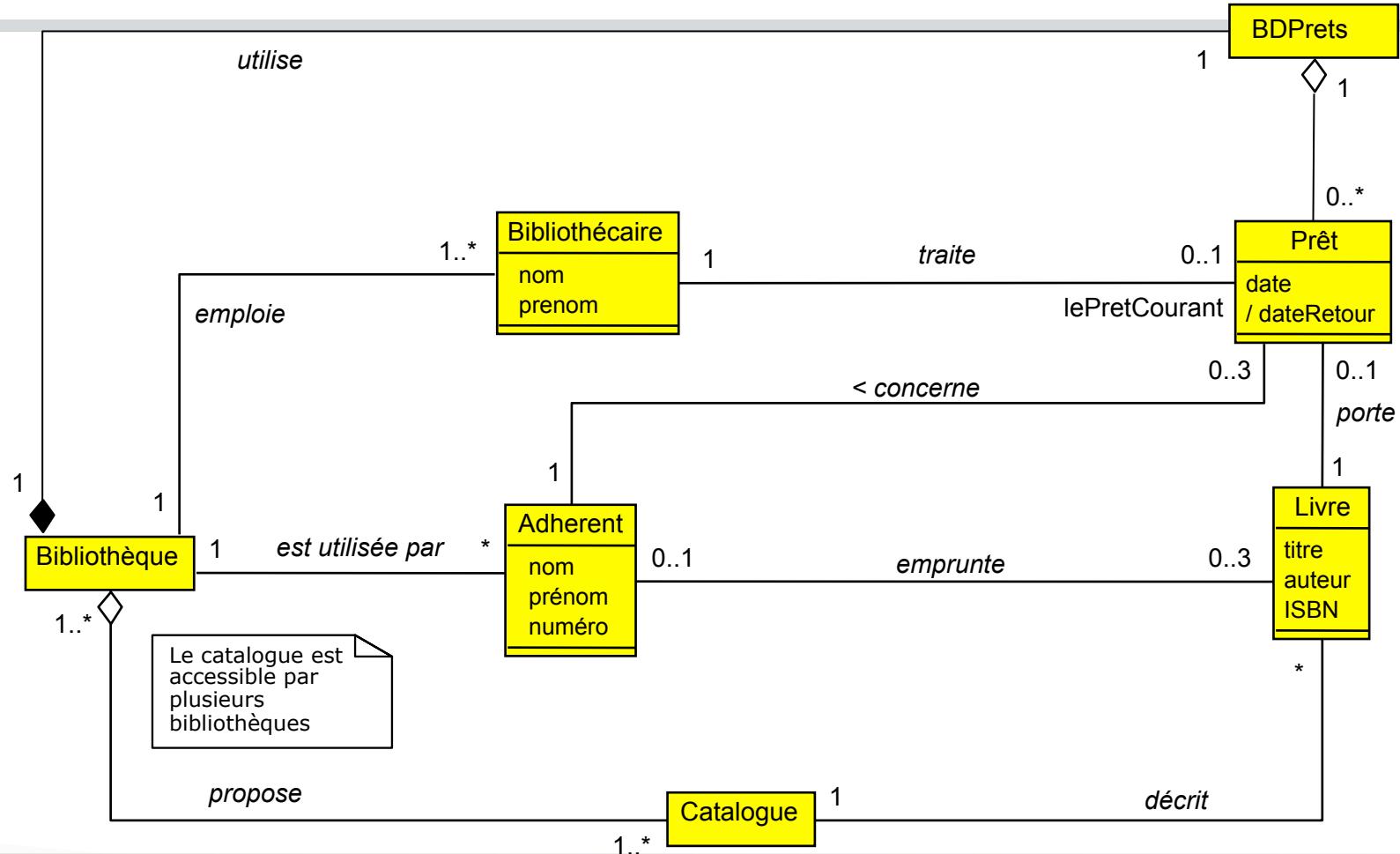
- Imaginez que vous êtes dans une équipe de développement.
- Vous travaillez sur un système de gestion d'une bibliothèque qui fait uniquement le prêt
- Le diagramme de cas d'utilisation est le suivant :



- Proposez un modèle du domaine (niveau analyse)

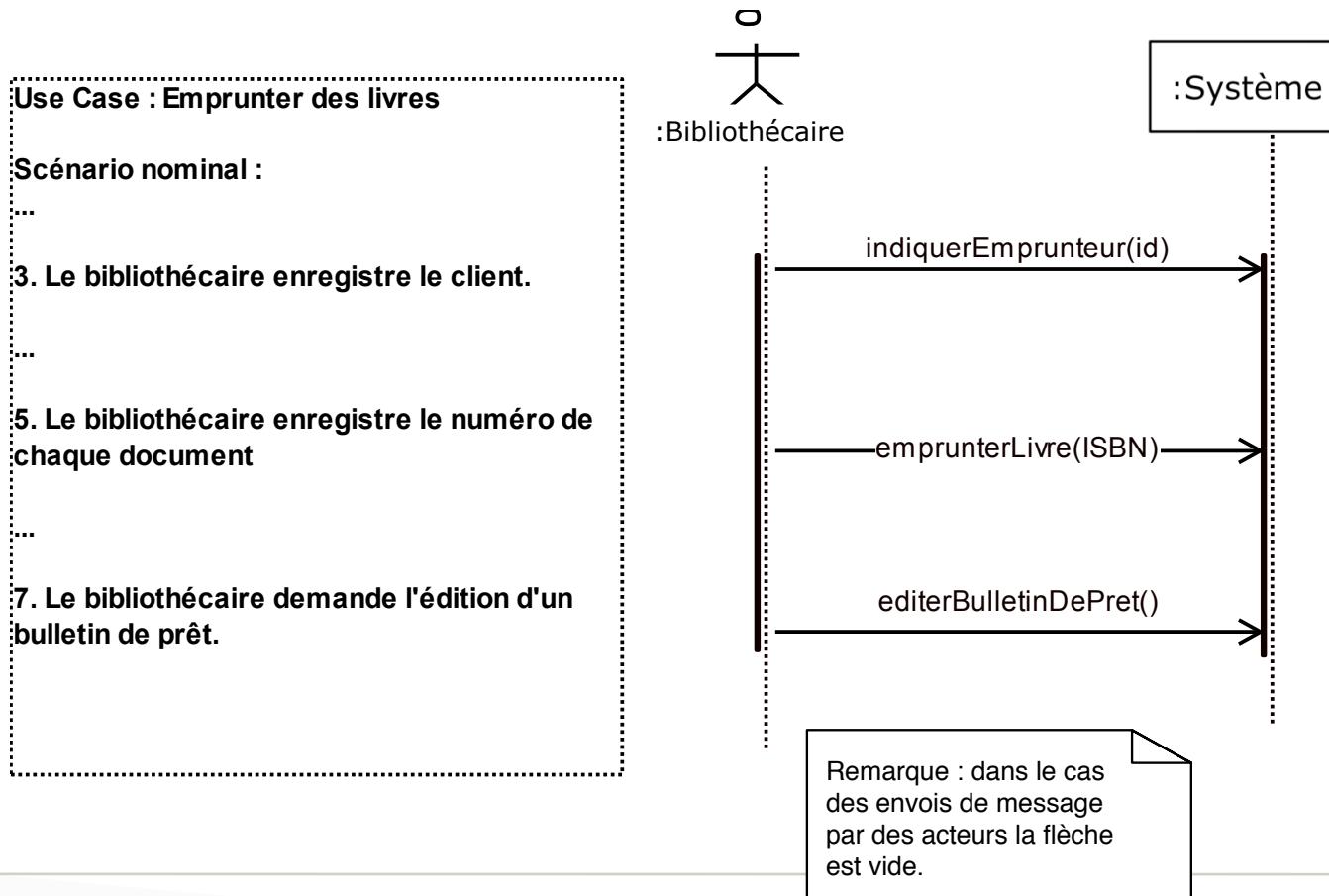
Exemple de diagr. de collaboration (2/5)

- Le modèle du domaine (niveau analyse) est le suivant :



Exemple de diagr. de collaboration (3/5)

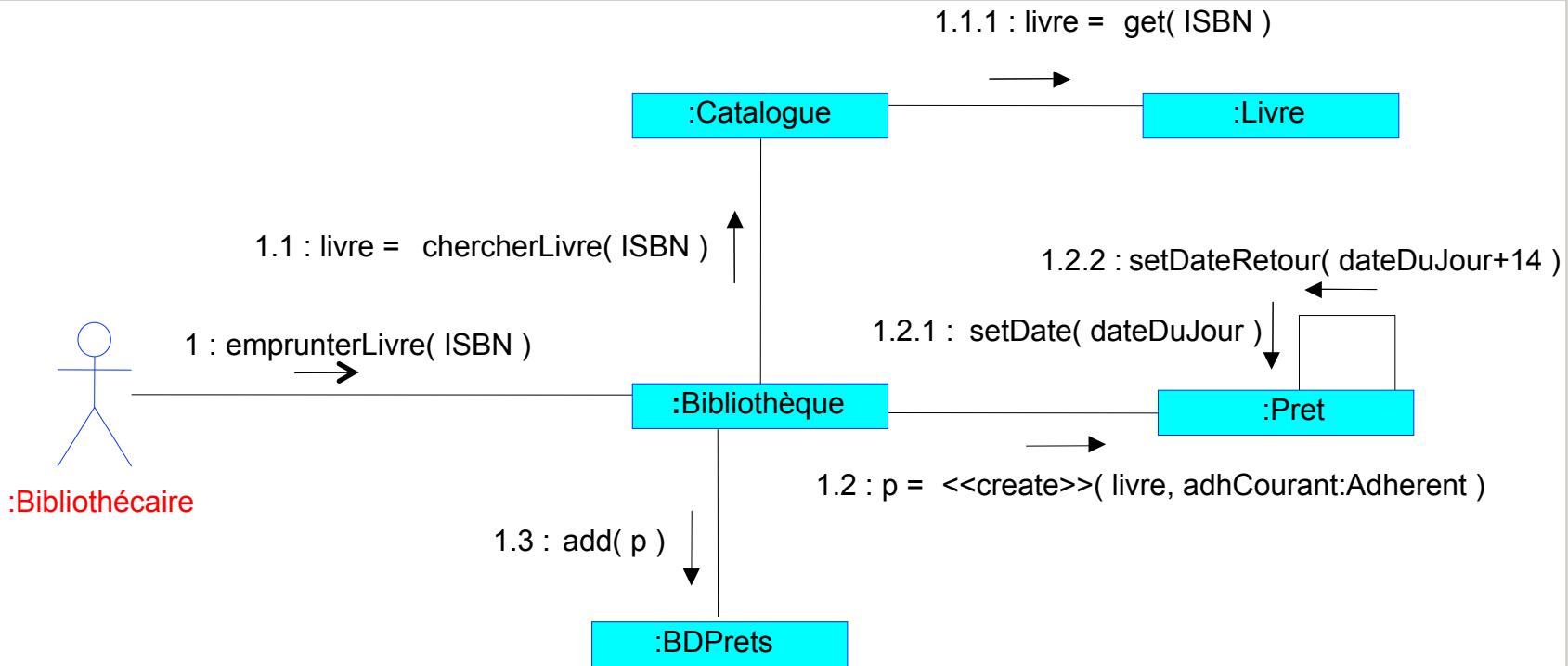
- Extrait du diagramme de séquence du cas d'utilisation
Emprunter des livres :



Exemple de diagr. de collaboration (4/5)

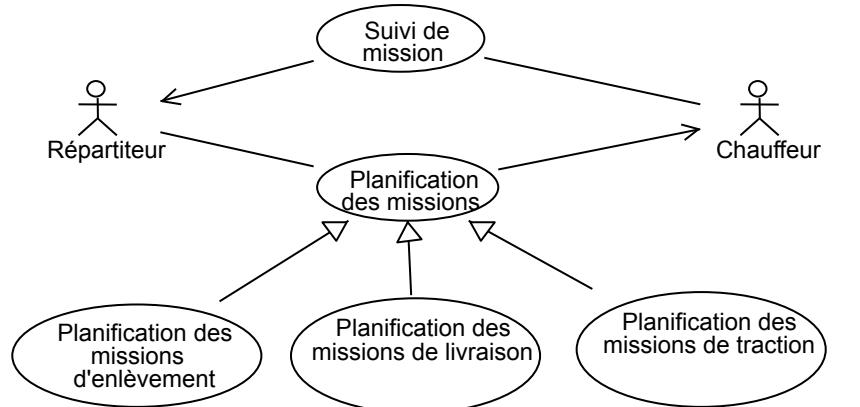
- Nous allons nous intéresser à l'opération *emprunterLivre*.
- Description textuelle :
 - Nom : *emprunterLivre(ISBN)*
 - Responsabilités : Enregistrer l'emprunt d'un livre identifié par son numéro ISBN
 - Références : Cas d'utilisation **Emprunter des livres**
 - Pré-conditions :
 - Le catalogue de livres existe et n'est pas vide.
 - L'adhérent a été reconnu par le système.
 - Post-conditions :
 - Un prêt *p* a été créé.
 - L'attribut *date* de *p* a été positionné à la date du jour.
 - L'attribut *dateRetour* de *p* a été positionné à (la date du jour + deux semaines).
 - *p* a été lié au livre *I* dont l'attribut *ISBN* vaut l'ISBN passé en paramètre.
 - *p* a été lié à l'adhérent concerné et à la bibliothèque.
- Développez un diagramme de collaboration pour l'opération *emprunterLivre*.

Exemple de diagr. de collaboration (5/5)



Correspondance entre diagr. d'interactions (1/3)

- Logiciel SIVex pour une entreprise de messagerie (transport de colis)
- Diagramme de cas d'utilisation du package *Gestion missions* :



- Description textuelle
 - Sommaire d'identification

Titre : Planification des missions

But : Planification des missions d'une agence à partir de la connaissance du plan de transport, des ressources disponibles et des commandes à assurer quotidiennement

Résumé : Création d'une nouvelle mission d'enlèvement, de livraison ou de traction à partir des commandes confirmées. Modification et annulation de mission

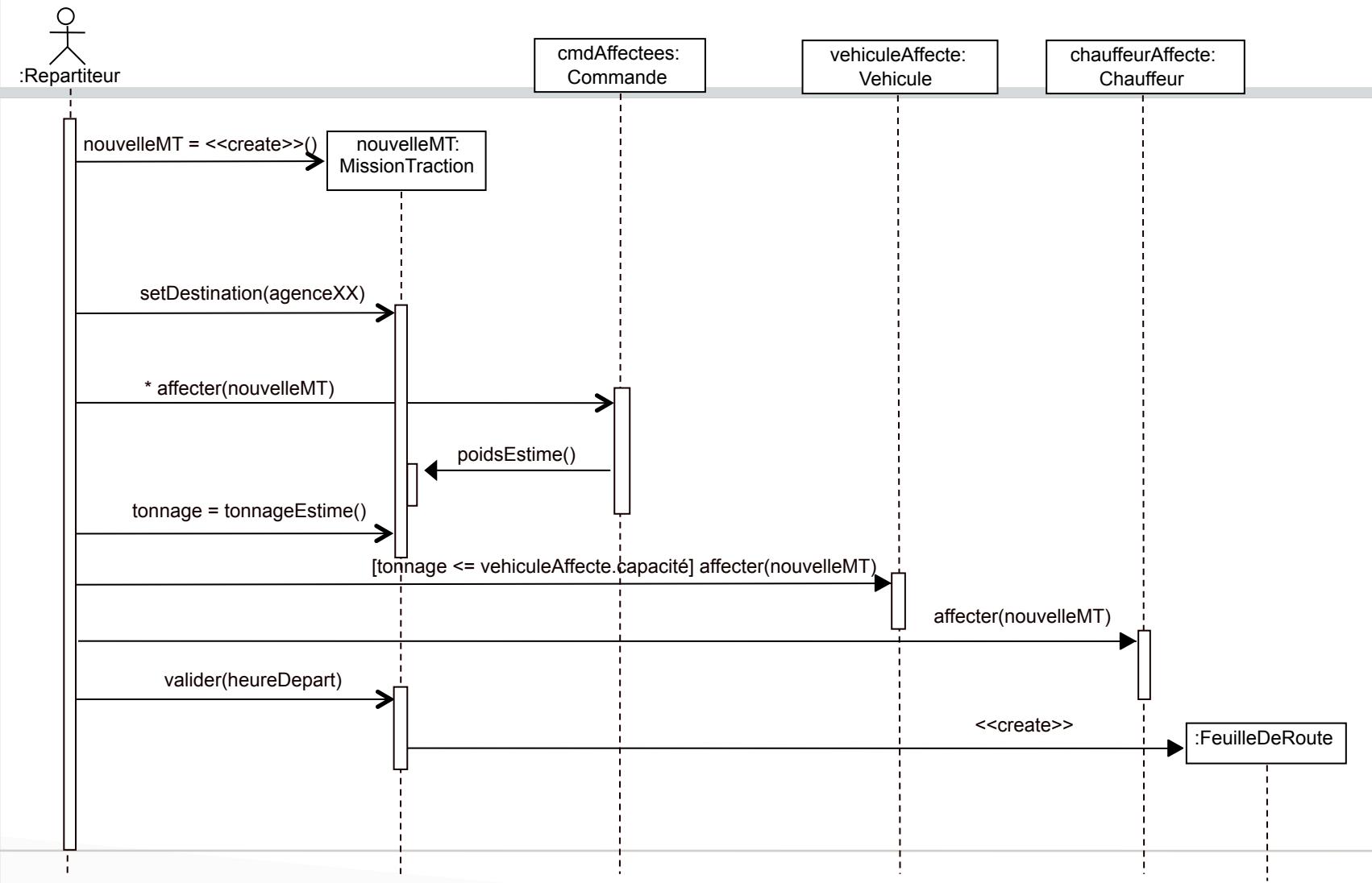
Acteurs : répartiteur (principal), chauffeur (secondaire)

Date de création : 02/02/99 **Date de mise à jour** : 24/07/99

Version : 2.1

Responsable : A. Legion

Correspondance entre diagr. d'interactions (2/3)



Correspondance entre diagr. d'interactions (3/3)

→ Développez un diagramme de collaboration correspondant au diagramme de séquence