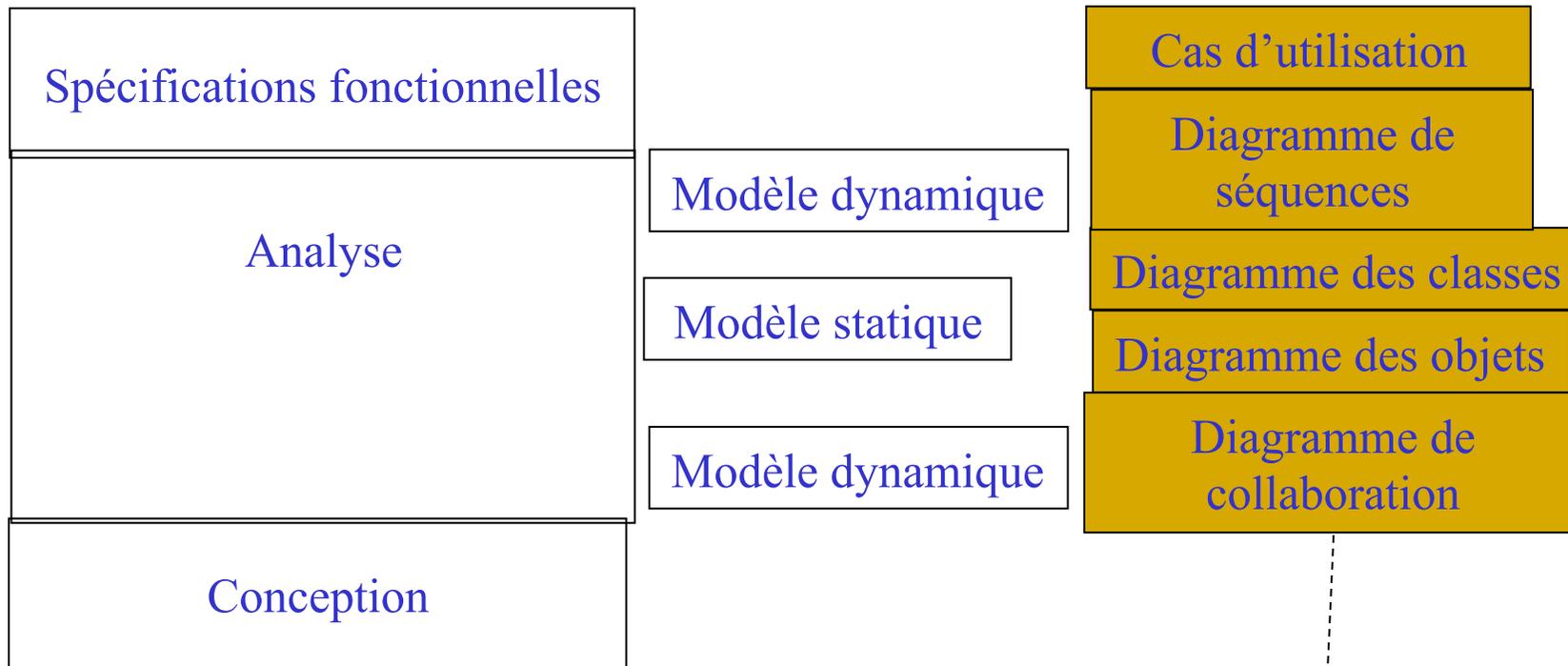

Le diagramme des collaborations

A.Osmani

Site officiel UML : www.uml.org

Le diagramme des collaborations en UML



Sommaire

1. Les diagrammes d'interactions
2. Définition d'une collaboration
3. Diagramme des collaborations
4. Exemple de diagramme des collaborations

Les diagrammes d'interaction

- Le diagramme d'interaction est décrit par un ensemble d'éléments statiques du modèle (objets, interfaces, etc.), les relations les associant ainsi que les messages échangés entre eux.
- Les diagrammes de collaboration et les diagrammes de séquences constituent les diagrammes d'interaction.
- Ils concernent la partie dynamique du système.
- Le diagramme de séquence fait apparaître l'ordre temporel et le diagramme des collaborations l'organisation des objets qui envoient et reçoivent des messages. Le diagramme des collaborations met en évidence l'effet de la structure des objets sur les interactions.
- Les deux diagrammes sont isomorphes.

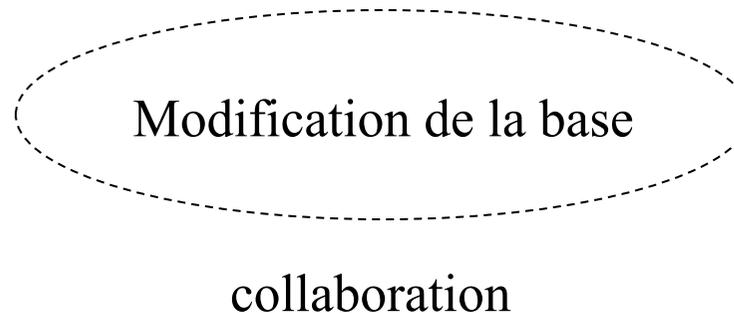
Les collaborations

Une collaboration définit une interaction. Les collaborations ont des dimensions structurelles et comportementales.

Exemple :

Dans une base de données répartie, chaque machine conserve une partie des données : une modification effectuée sur cette base impliquera plusieurs classes qui interagissent pour réaliser la transaction.

Du point de vue de l'utilisation, toutes ces opérations peuvent être représentées sous forme d'une seule transaction. Cet ensemble de classes (partie structurelle) avec leurs interactions (partie comportementale) forme ce que nous appelons en UML une collaboration. Graphiquement, elle est représentée comme suit :



Les collaborations

Une collaboration est représentée par une ellipse en pointillées. Elle comporte un nom et une structure

- ❑ Le nom : il doit être unique. Il est dit nom complet s'il est précédé du nom de package ou il apparaît, sinon il est dit nom simple.
- ❑ La structure : la partie structurelle comporte n'importe quelle combinaison de classifieurs : les classes, les interfaces, les composants, les nœuds, etc. Ces classifieurs peuvent être reliés par l'ensemble des relations habituelles d'UML (associations, généralisations, dépendances, ...)

Important : une collaboration peut traverser plusieurs niveaux d'un système et un élément peut apparaître dans plusieurs collaborations.



Représentation graphique d'une collaboration

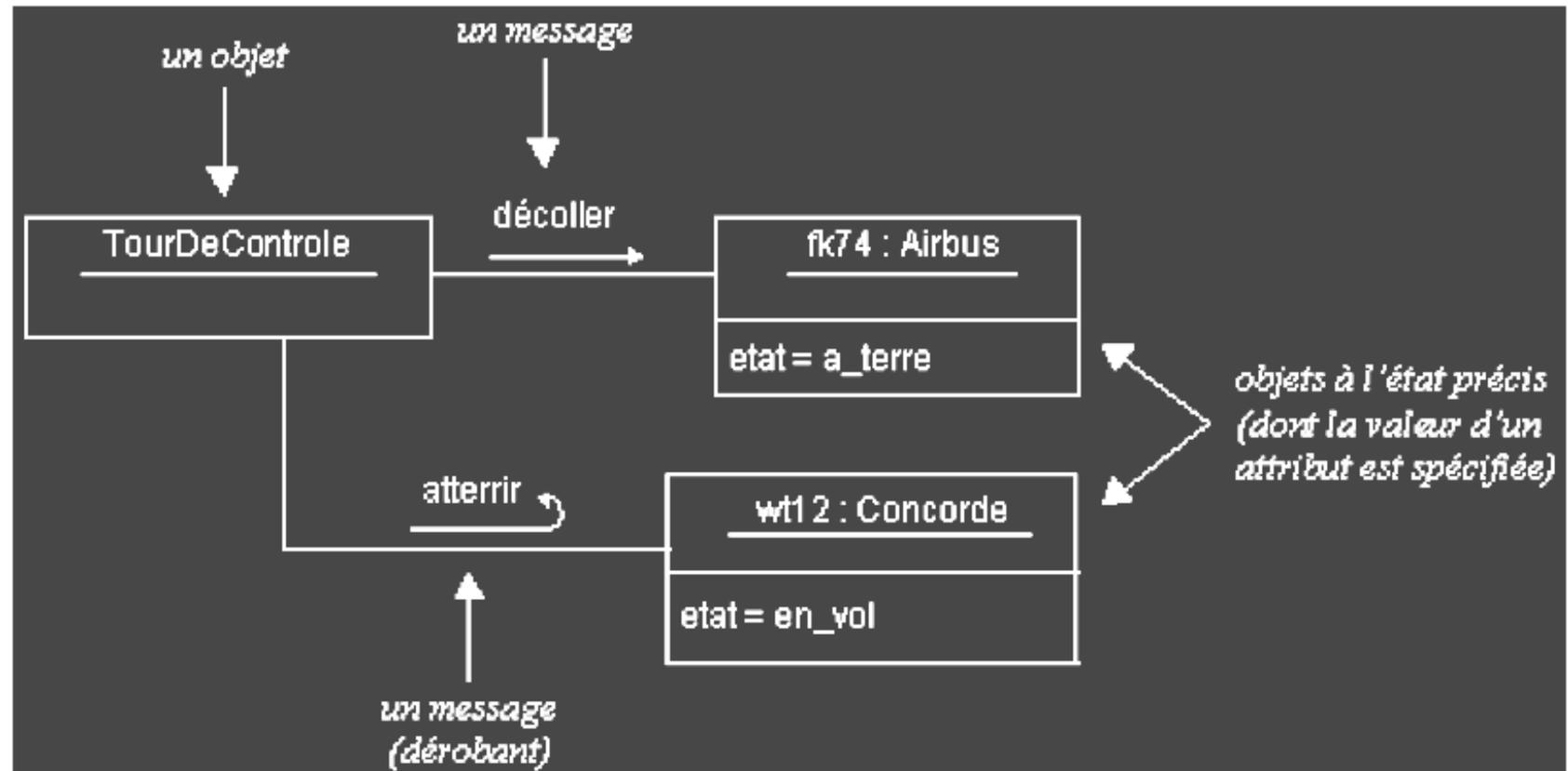
Les collaborations

Les collaborations montrent les interactions entre les objets, les acteurs et d'autres éléments du modèle.

Le but d'une collaboration est de réaliser un objectif du système (ou encore de répondre à un besoin des utilisateurs).

Le contexte d'une interaction peut être représenté dans une collaboration par les états des objets, leurs variables locales, les arguments des messages et les états des liens entre les objets .

Exemple 1

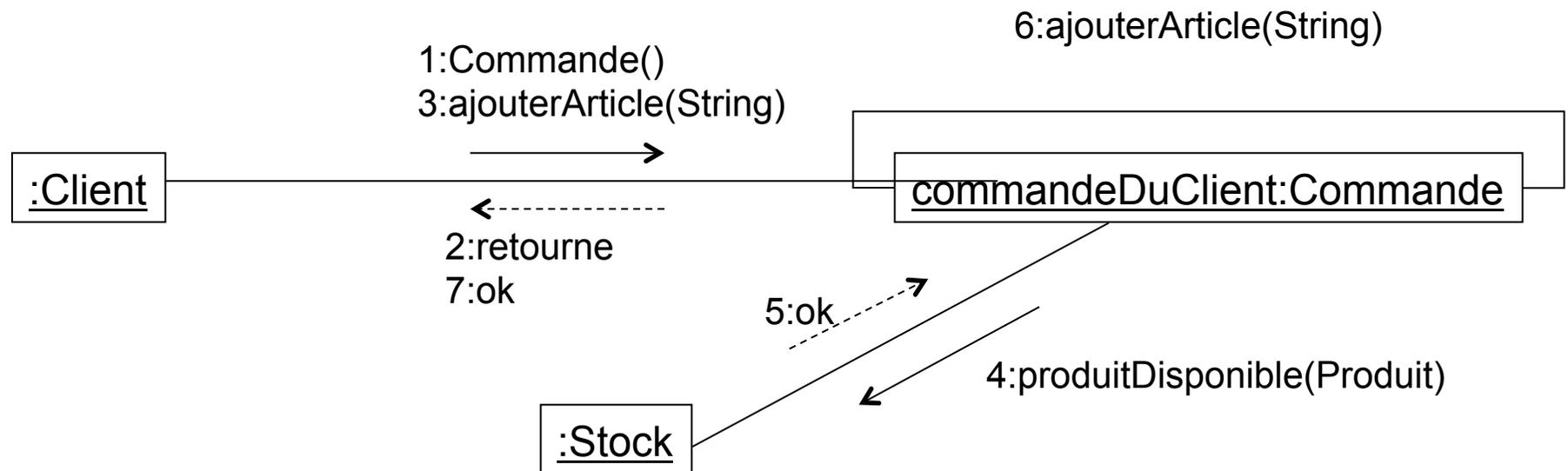


Source de l'exemple uml.fr

Les collaborations

L'analyse des collaborations doit fournir une première définition des responsabilités. Elle permet de définir une partie des méthodes implémentées au niveau des classes (actions exécutées par les objets participant à la collaboration).

Exemple 2 : un client crée une commande et ajoute des articles en vérifiant la disponibilité de chacun.



NB : le suivi des scénarios se fait en suivant les numéros associés à chaque action.

Les messages

Les messages échangés entre les éléments du modèle intervenant dans le diagramme de collaboration peuvent indiquer :

- les clauses qui conditionnent leur envoi
- leurs numéros d'ordre par rapport aux autres messages
- leurs arguments
- Leurs récurrence

Exemples

1 : mange()

[heure = midi] 1 : manger()

3/* || [i :=1..5] : fermer()

le numéro du message est 1

envoyer le message à midi

après l'envoi du message 3, envoi en parallèle de 5 message

Syntaxe d'un message

Les messages apparaissant dans une collaboration ont la syntaxe suivante (les éléments entre deux crochets désignent des options) :

[pré '/'] ['[' cond ']'] [séqu] ['*' ['||'] ['[' iter ']' ':'] [r ':='] msg '(' [par] ')'

Syntaxe d'un message

[pré '/'] : prédécesseurs. Liste de numéros séparés par des virgules. Le message ne sera envoyé qu'après l'envoi de ses prédécesseurs. Exemple : 3,6,9 / 4

[[' cond ']] : conditions de garde. Expression booléenne conditionnant l'envoi du message.

[séqu] : numéro de séquence du message. Les numéros peuvent être représentés par des chiffres séparés par des points. Dans ce cas, l'exécution se fait selon l'ordre lexicographique : 2.3.5 s'exécute avant 2.3.6 et après 2.1
L'envoi de messages simultanés se fait en utilisant des lettres : 2.3.5.a et 2.3.5.c est simultanément.

[* ['||'] [[' iter ']'] :'] : précise la récurrence du message. Permet de préciser en langage naturel l'envoi séquentiel (ou parallèle en utilisant ||) de messages.
*4/1.5 * après le message 4, le message 1.5 sera lancé plusieurs fois*

[r :=] : valeur de retour du message
msg (' [par] ') : message et d'éventuels paramètres

Exemples :

1.3, 2 / [t<10s] 3.1 : age:= demandeAge(identifiant)

2 / [malade] 3.1.a * : prendreMedicament()

2 / [malade] 3.1.b : seCouvrir()

Diagramme des collaborations

L'organisation des collaborations forment le diagramme des collaborations.

Les collaborations ont une influence capitale dans les décisions de conception d'un système. Certains les considèrent comme le cœur de l'architecture d'un système.

Tout système objet bien conçu doit comporter un ensemble de collaboration de taille modeste. D'ou la nécessité de bien organiser les collaborations.

Il y a deux relations à prendre en compte :

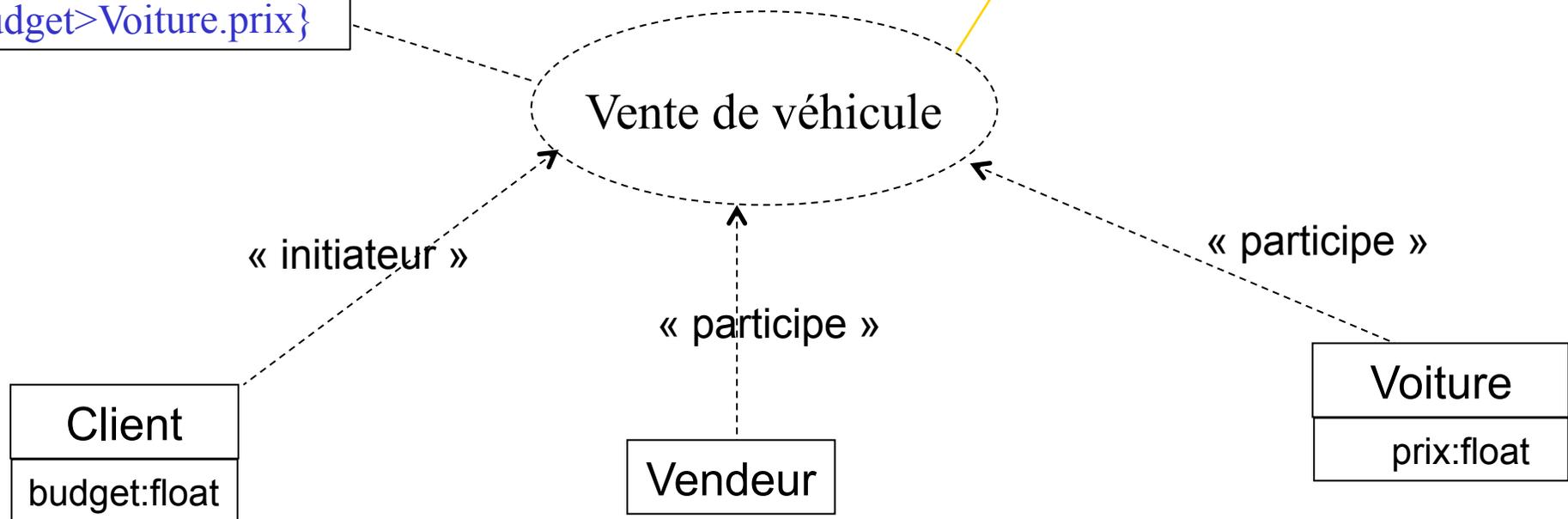
1. La relation entre la collaboration et les éléments qu'elle réalise. Par exemple, une collaboration peut réaliser un cas d'utilisation (voir exemple). Ce cas d'utilisation, ses acteurs et les cas d'utilisation voisins fournissent le contexte à la collaboration.
2. La relation entre les collaborations. Par exemple cette relation peut être une relation de raffinement comme le montre l'exemple suivant :

Exemple de collaboration

Contrainte précisant le rôle ou la portée de
L'élément du modèle

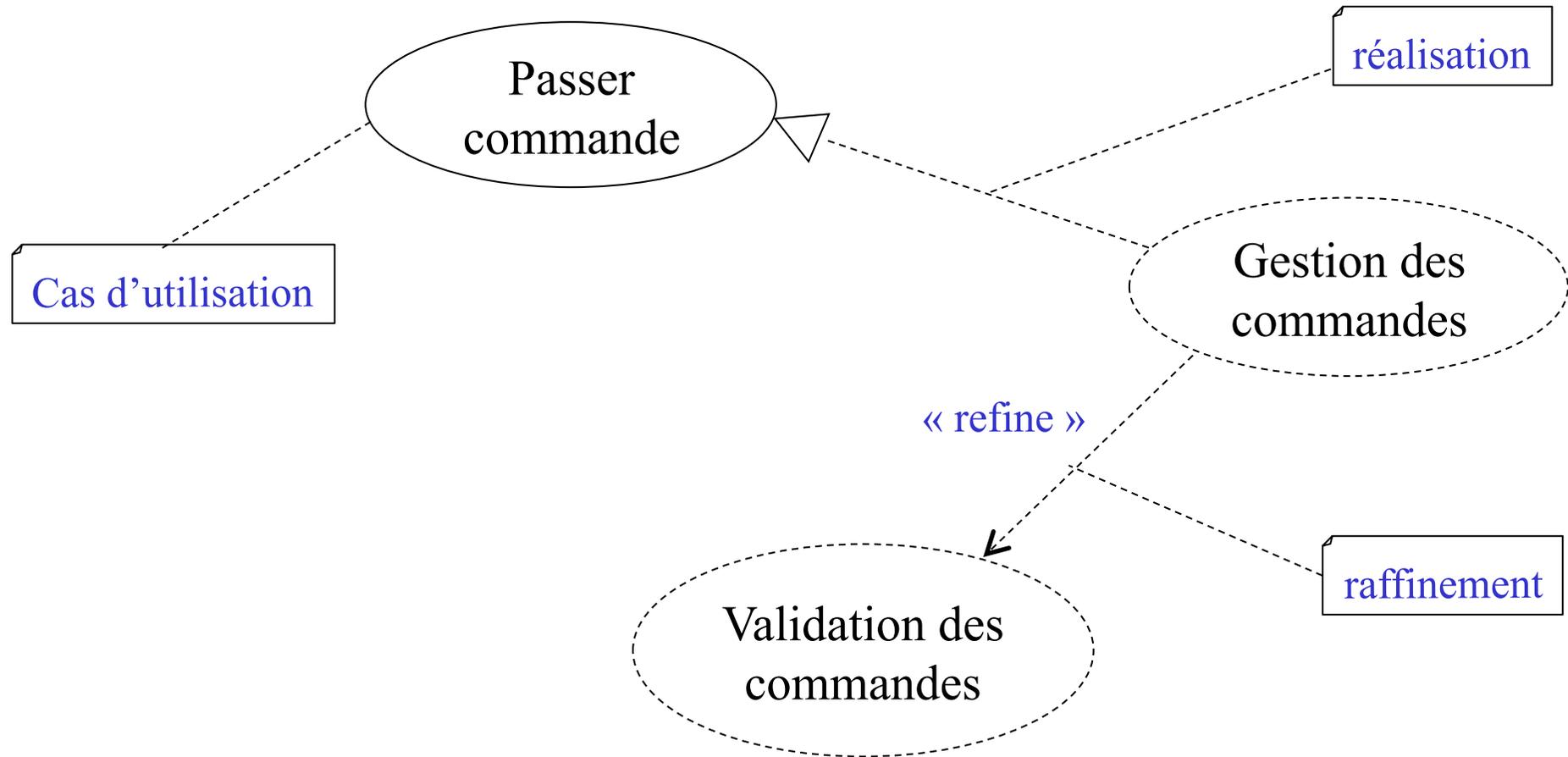
Collaboration : dérive d'un cas d'utilisation
Et correspond à un objectif du système

{client.budget > Voiture.prix}

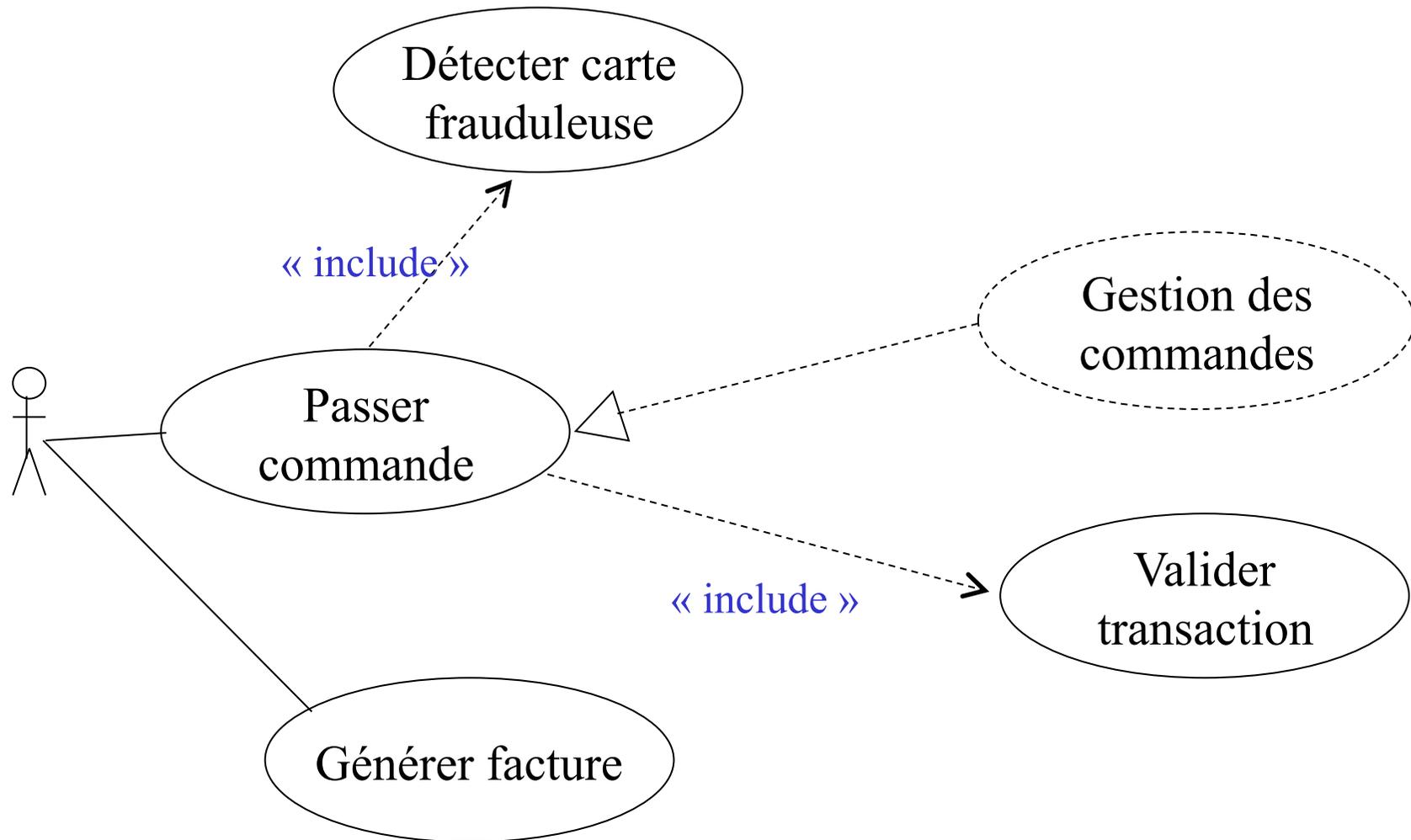


Classes qui participent à la réalisation de la
collaboration

Diagramme de collaboration



Exemple de réalisation d'un use cases



Modélisation de la réalisation d'un cas d'utilisation

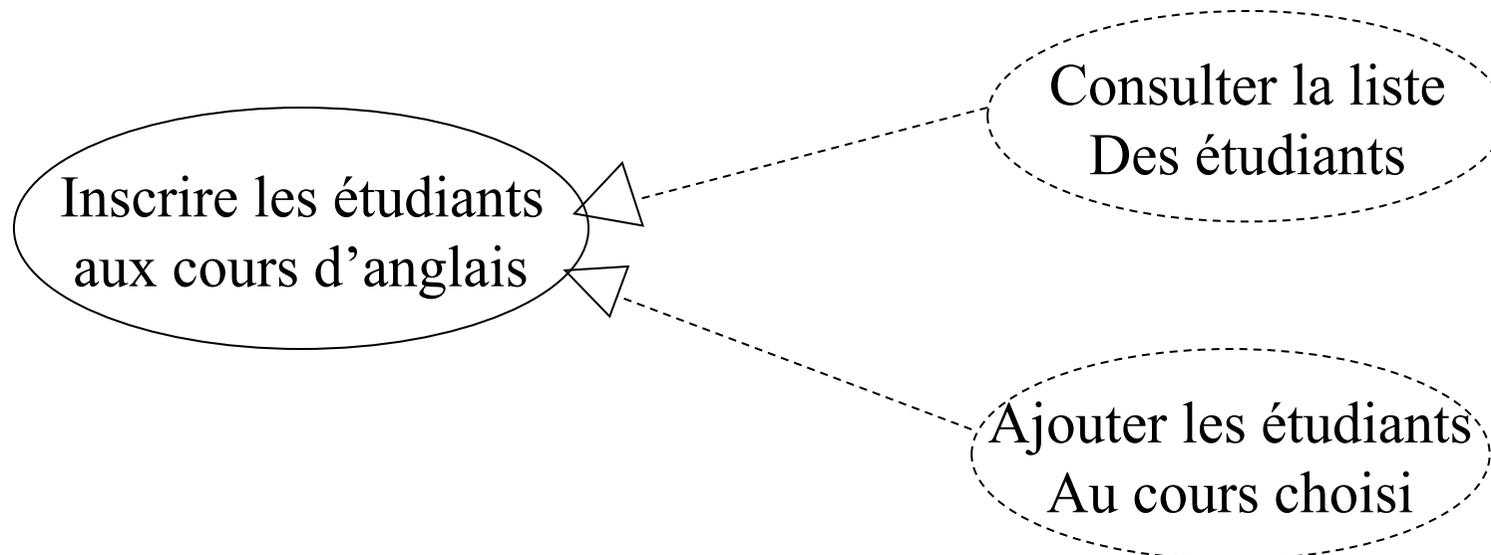
Définition d'une réalisation

Une réalisation est un lien entre un élément décrivant une solution d'implémentation et un élément de spécification.

Un élément de spécification est une description du comportement et/ou de la structure du logiciel (classe abstraite, cas d'utilisation, interface, etc.)

Un élément d'implémentation est l'explication détaillée de la façon dont les éléments de spécifications sont effectivement implantés (classes, composant, collaboration).

Exemple : réalisation d'un cas d'utilisation par deux collaborations



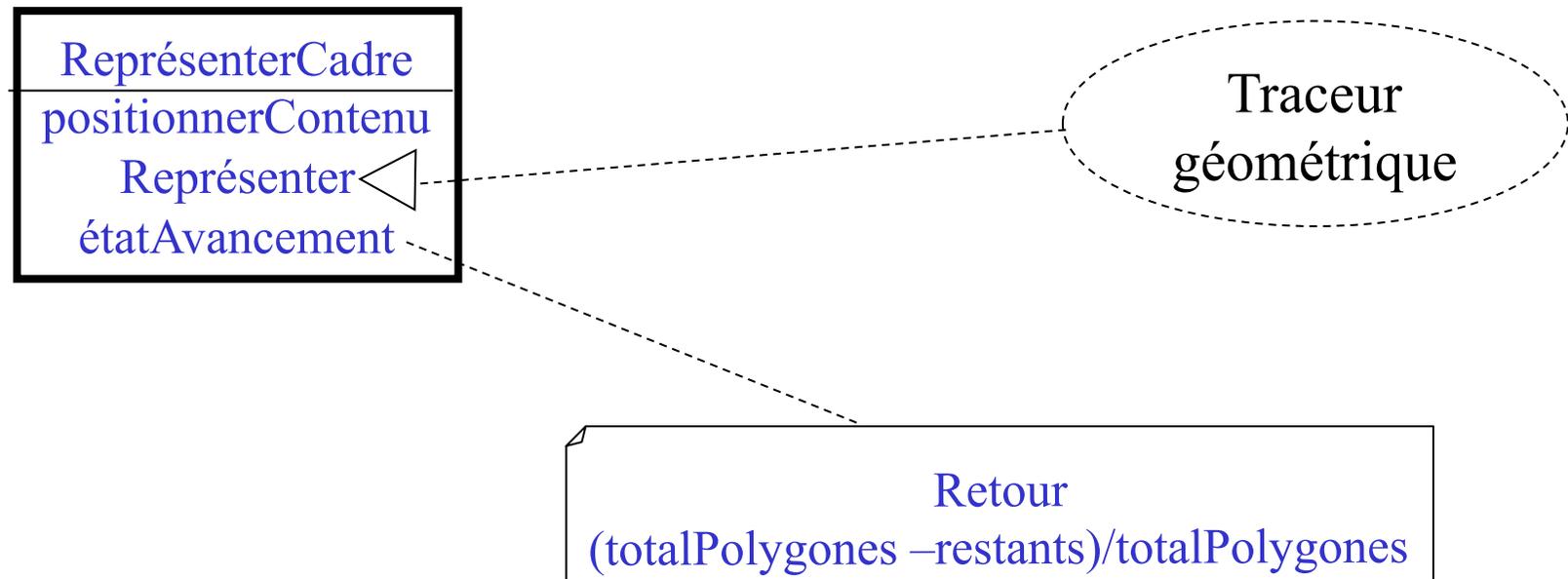
Exemple de réalisation d'une opération

La modélisation de la réalisation d'une opération est un objectif possible des collaborations. Souvent, il est possible de réaliser les opération directement dans le code source. Mais si celles-ci font intervenir un ensemble d'objets, il est préférable de réaliser la collaboration avant la phase de codage.

Pour modéliser l'implémentation d'une opération :

- identifier les paramètres, la valeur de retour, et les objets visibles depuis l'opération
- Si l'opération est triviale, décrire directement son code source
- Si l'opération est à forte composante algorithmique, utiliser le diagramme des activités
- Si l'opération est complexe et nécessite un travail de conception détaillée, utiliser le diagramme des collaboration (voir exemple).

Exemple de réalisation d'une opération

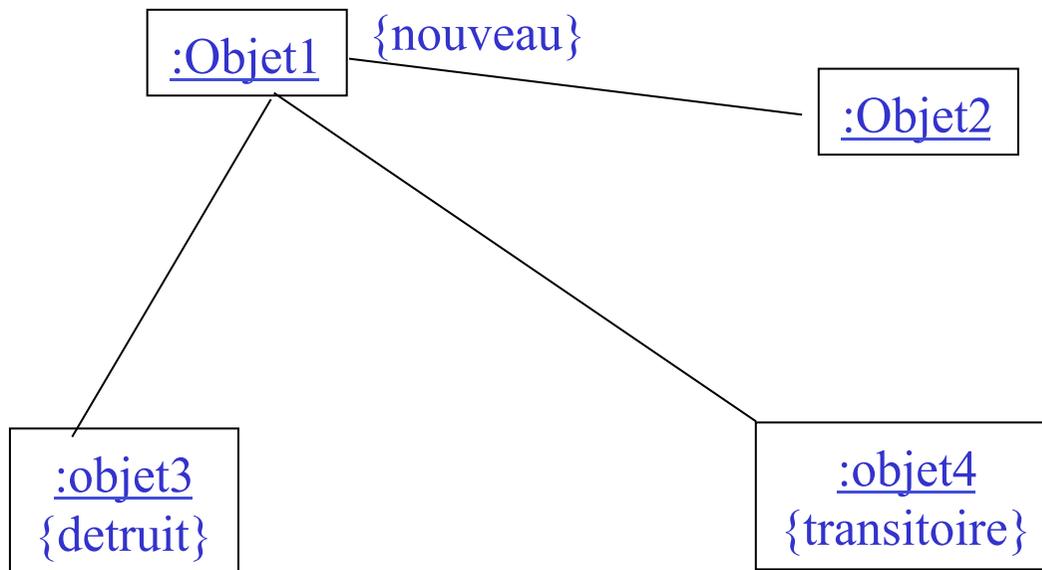


Modélisation de la réalisation d'une opération

Quelques utilisations de collaborations

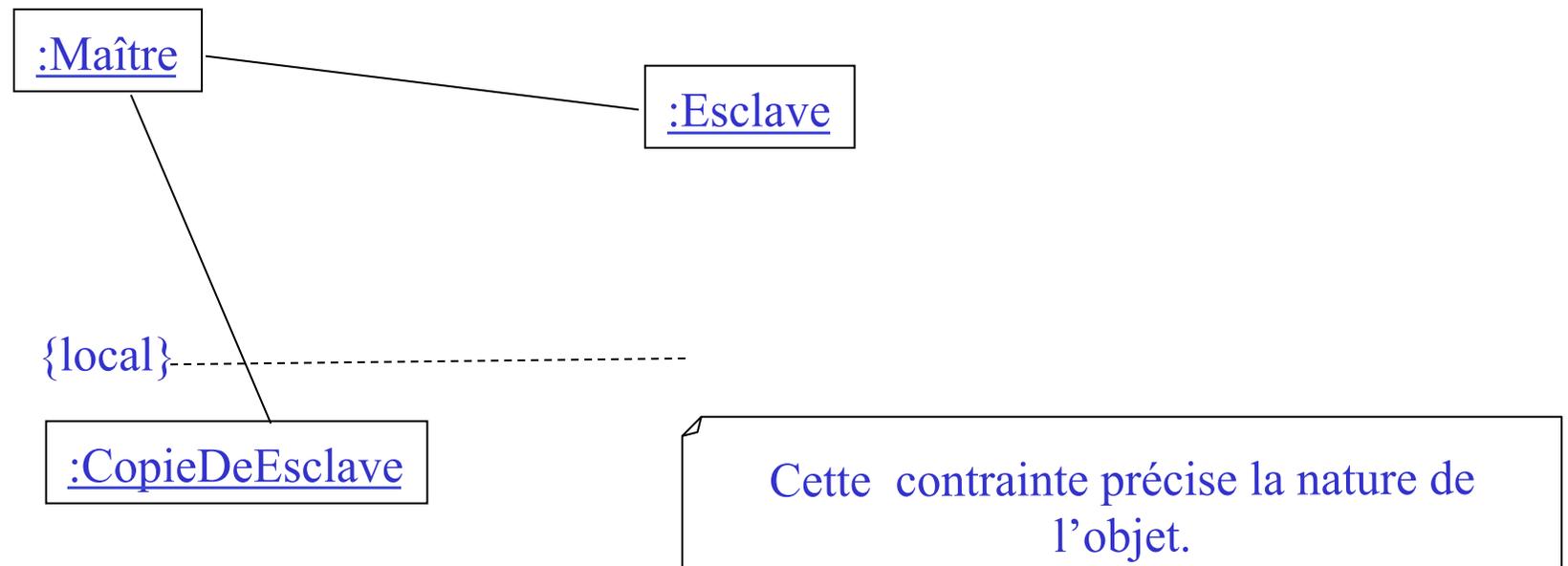
Nous utiliserons dans ce cours les contraintes uniquement les contraintes suivantes :

- {Local} : objet local
- {nouveau} et {détruit} : création et destruction d'un objet
- {transitoire} : objet crée puis détruit dans la même interaction



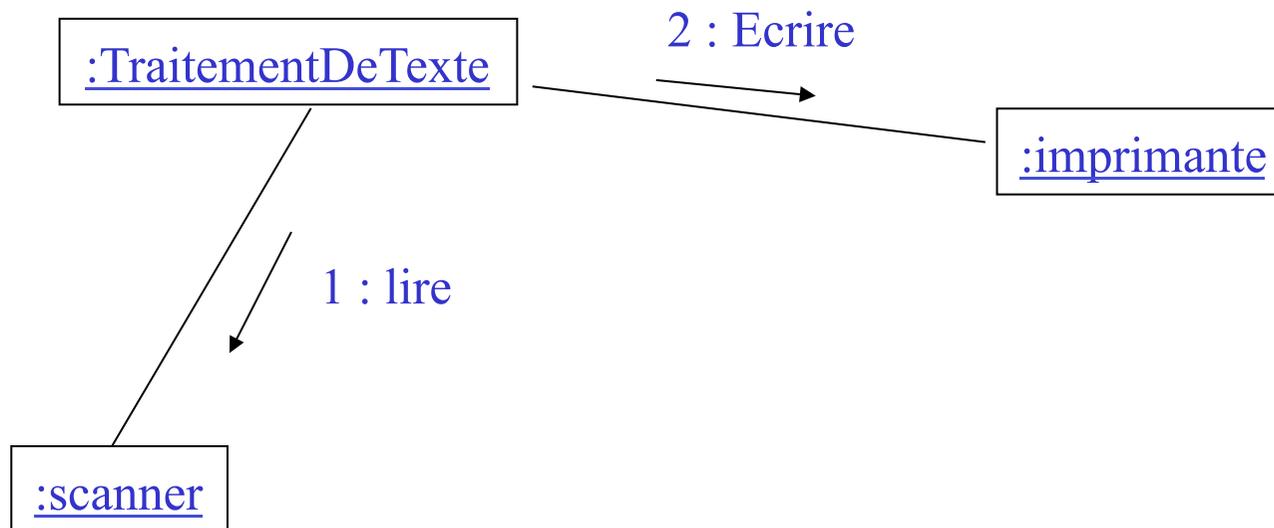
Quelques utilisations de collaborations

Dans cet exemple, l'objet de la classe Maître crée localement une copie de l'objet de la classe esclave avant de la détruire. De sorte que si l'opération venait à être annulé, il est possible de le restituer.



Quelques utilisations de collaborations

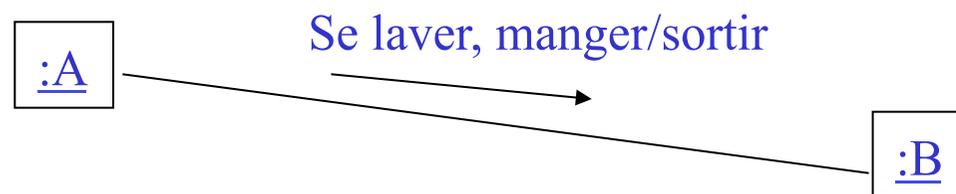
Les objets actifs : Les objets qui possèdent le flot de contrôle sont dits actifs. Un objet actif peut activer un objet passif pour un temps d'une opération, en lui envoyant un message.



Quelques utilisations de collaborations

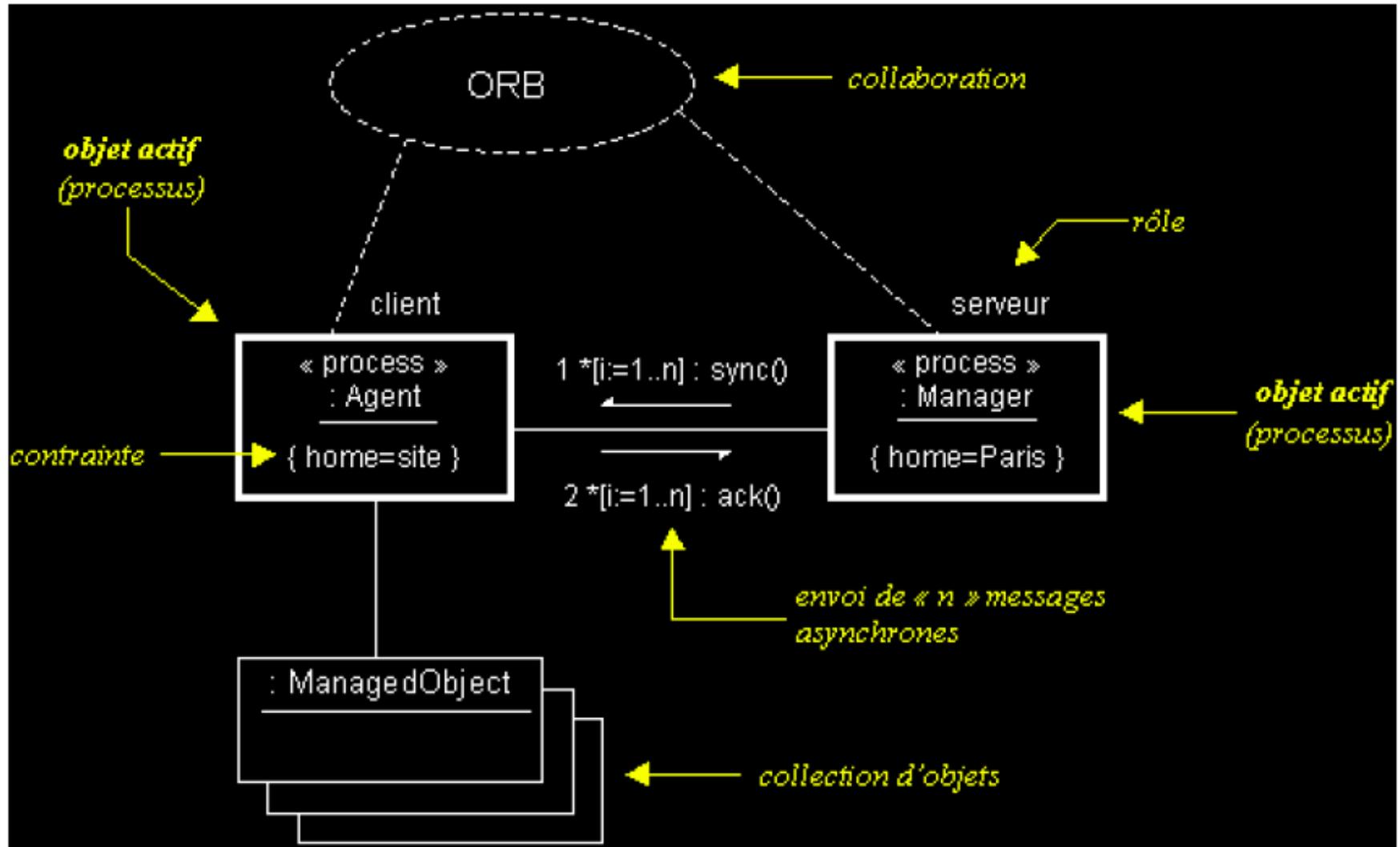
La synchronisation : Le point de synchronisation est exprimé par l'envoi de messages terminé par le caractère /.

Quand les messages ne sont pas numérotés, la synchronisation s'exprime comme suit : tous les messages avant le caractère '/' doivent être envoyés avant de valider le message après le caractère '/'.



Collaboration entre objets actifs

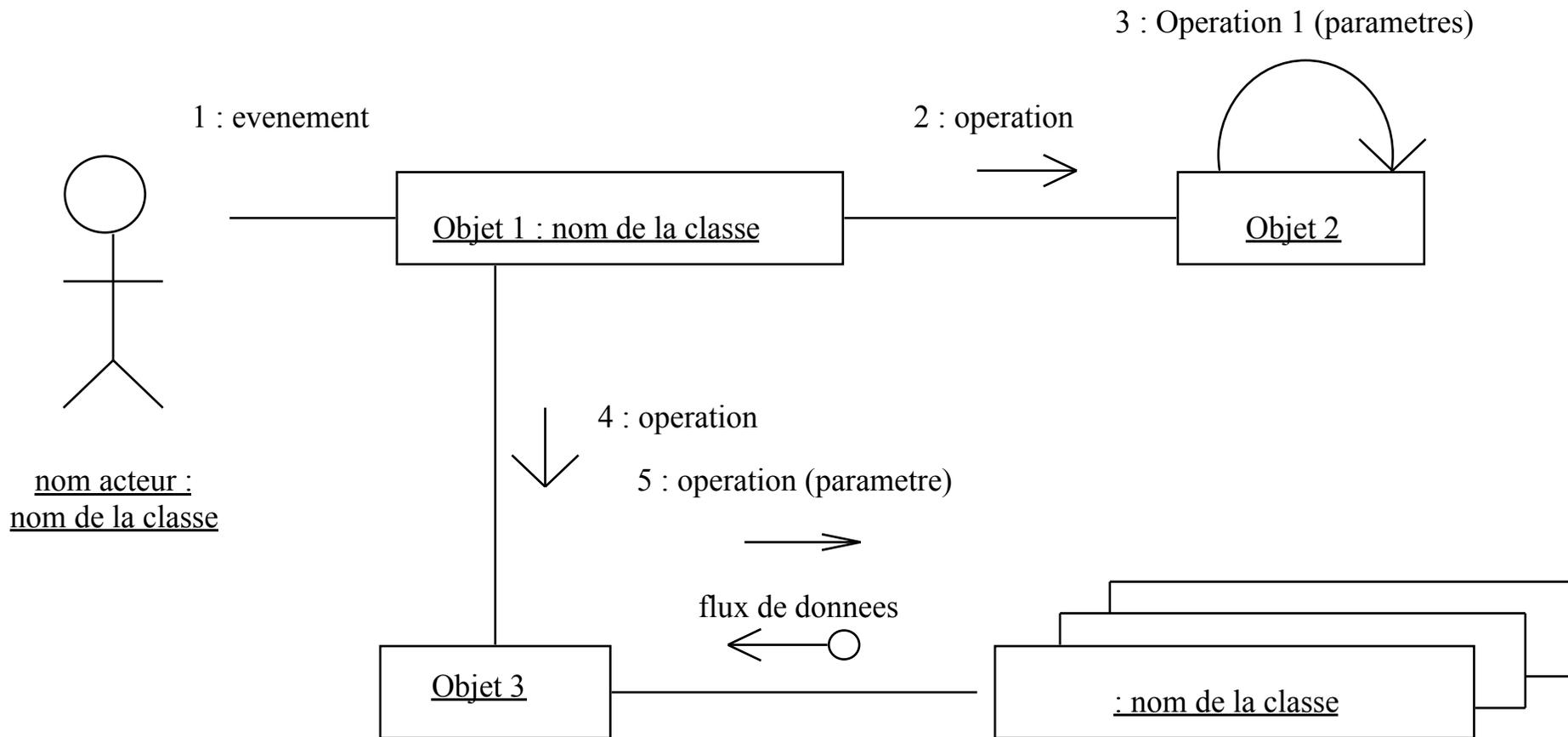
Les collaborations permettent d'exprimer les communications entre les objets actifs (exemple : les threads) de manière concurrente.



Source de l'exemple uml@free.fr

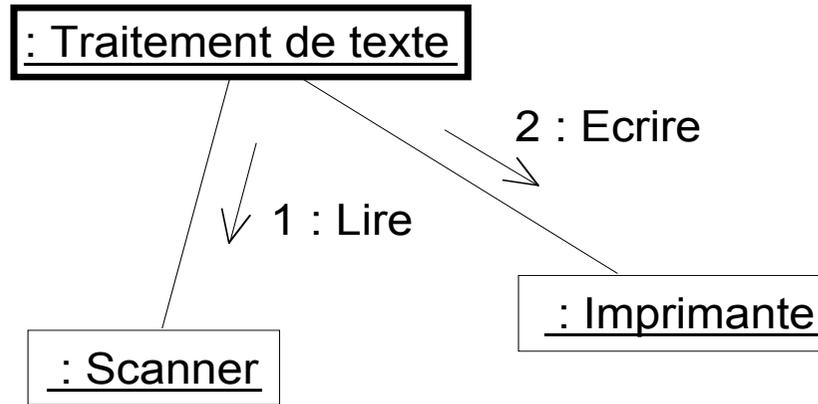
Fin

Diagramme de collaboration

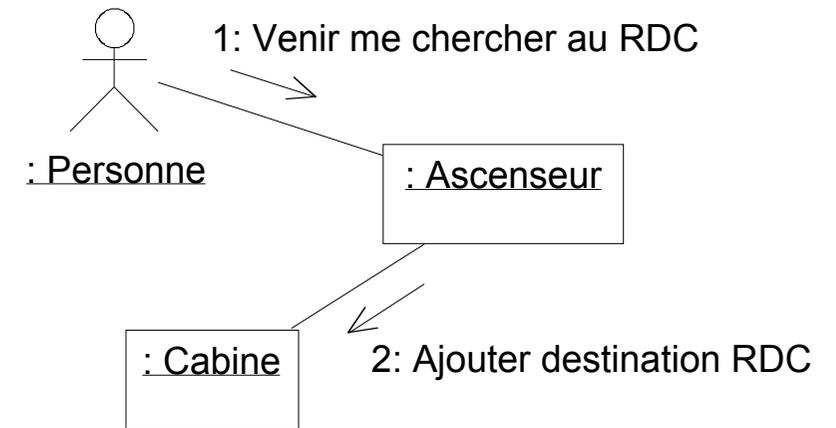


Exemples

■ Object Actif



■ Avec Acteur



■ Multi-objets

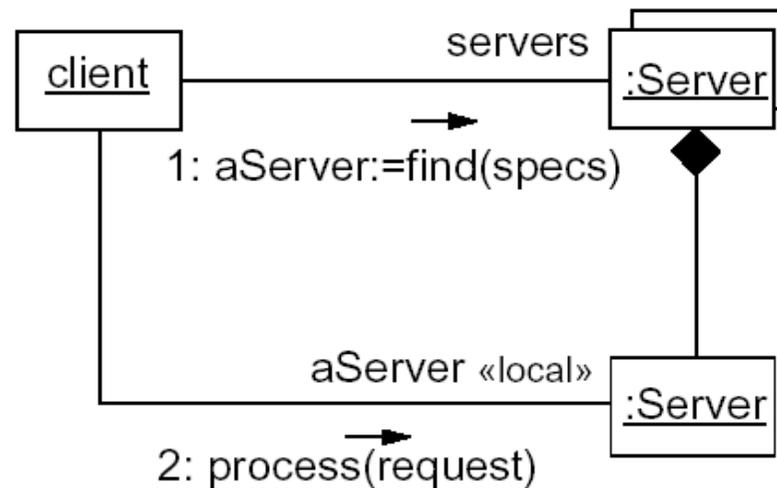
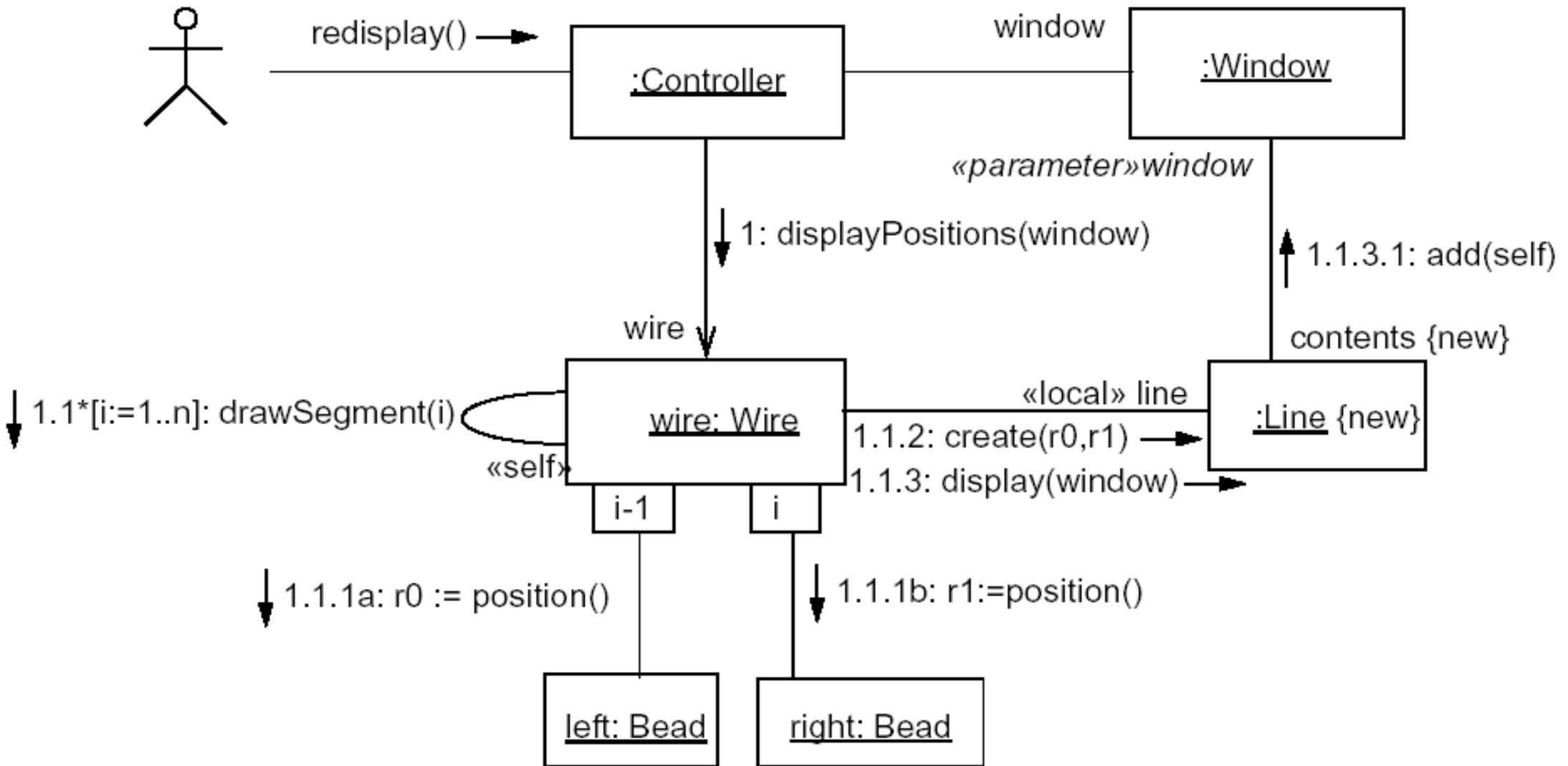


Diagramme de collaboration (niveau instances)



Exemple

- Un objet **A** envoie un stimulus **X** à un objet **B**, puis l'objet **B** envoie un stimulus **Y** à un objet **C**, et enfin **C** s'envoie un stimulus **Z**.

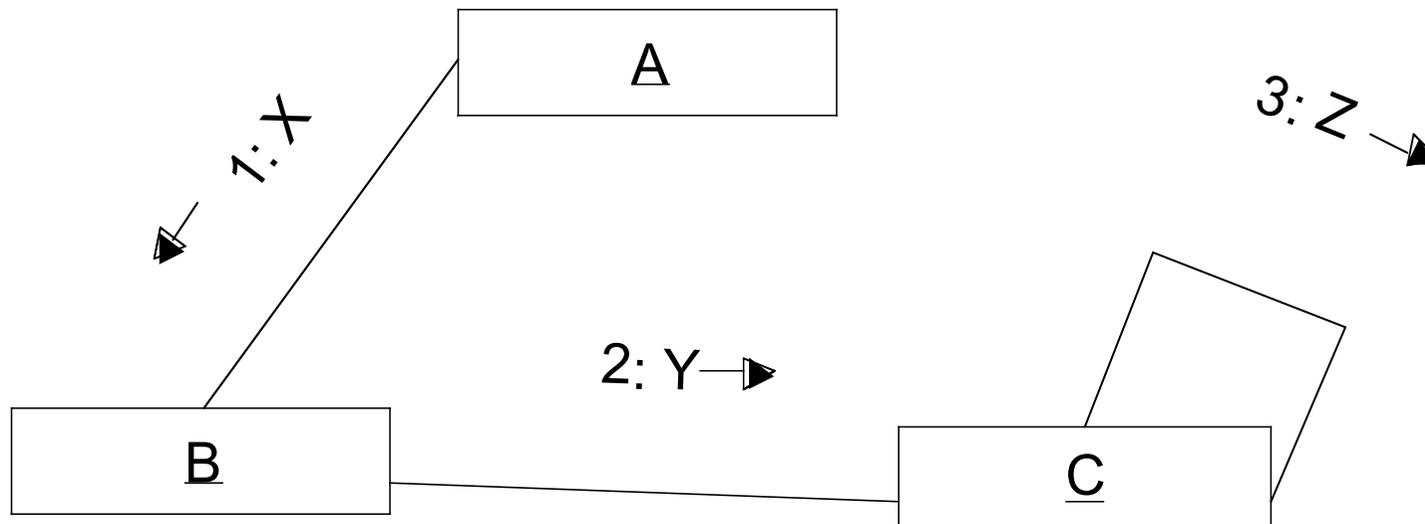


Diagramme de collaboration

■ Avec contraintes et généralisation

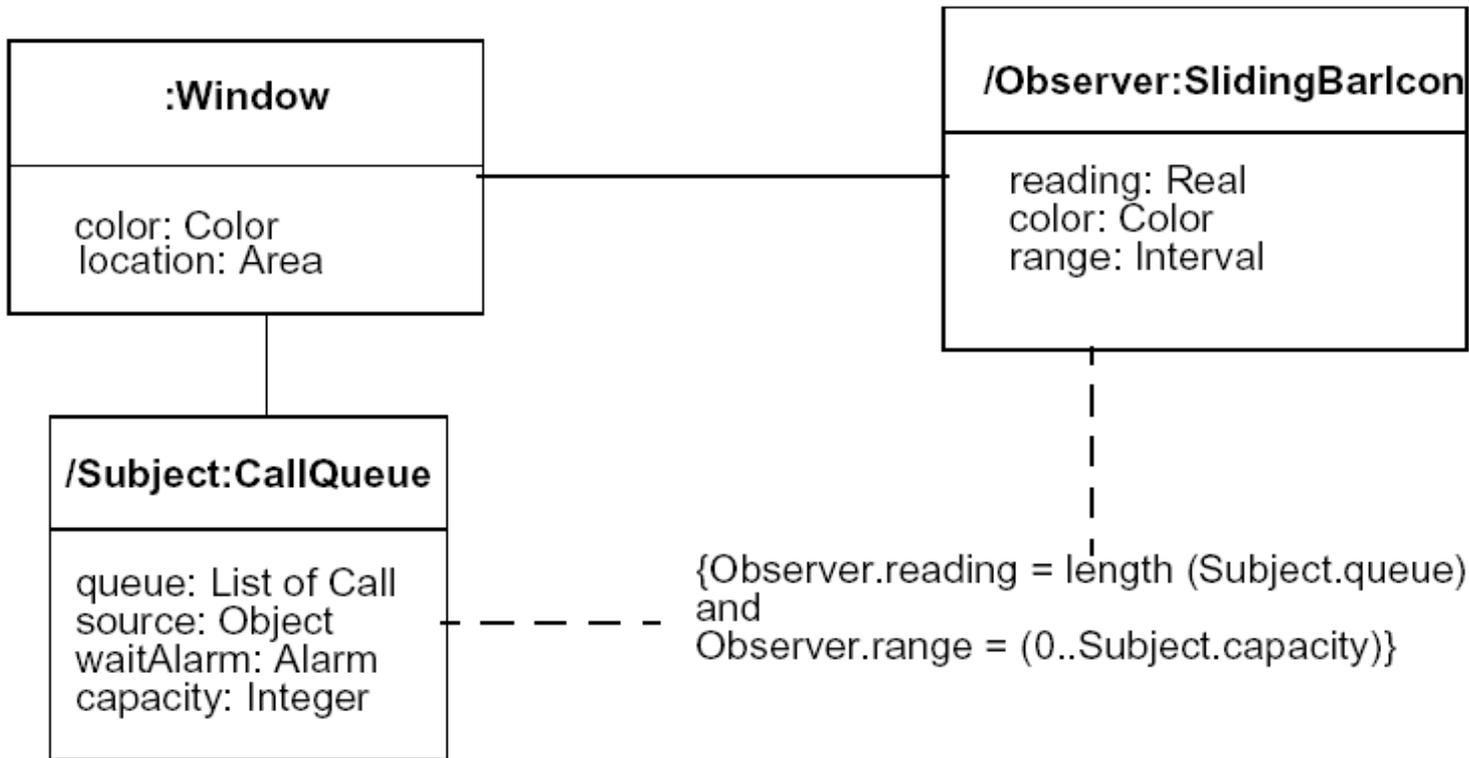
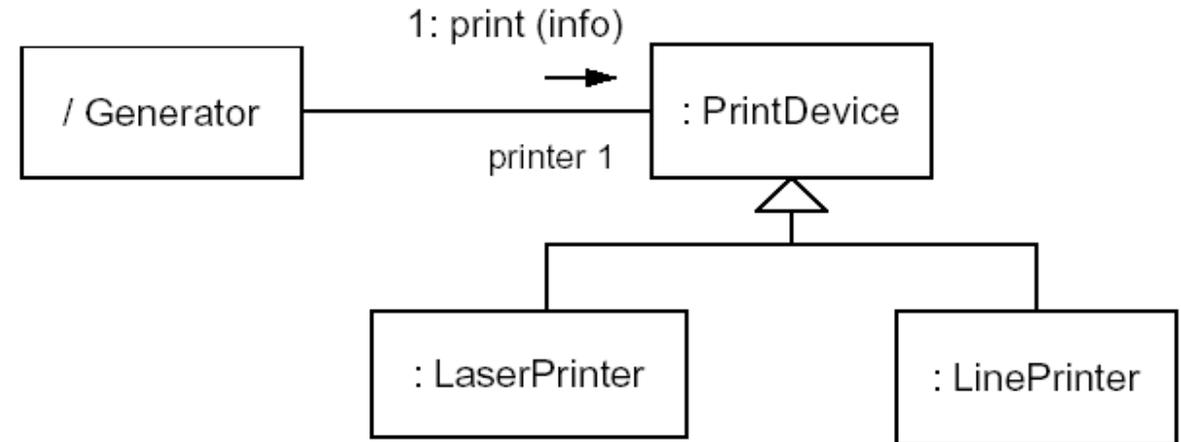
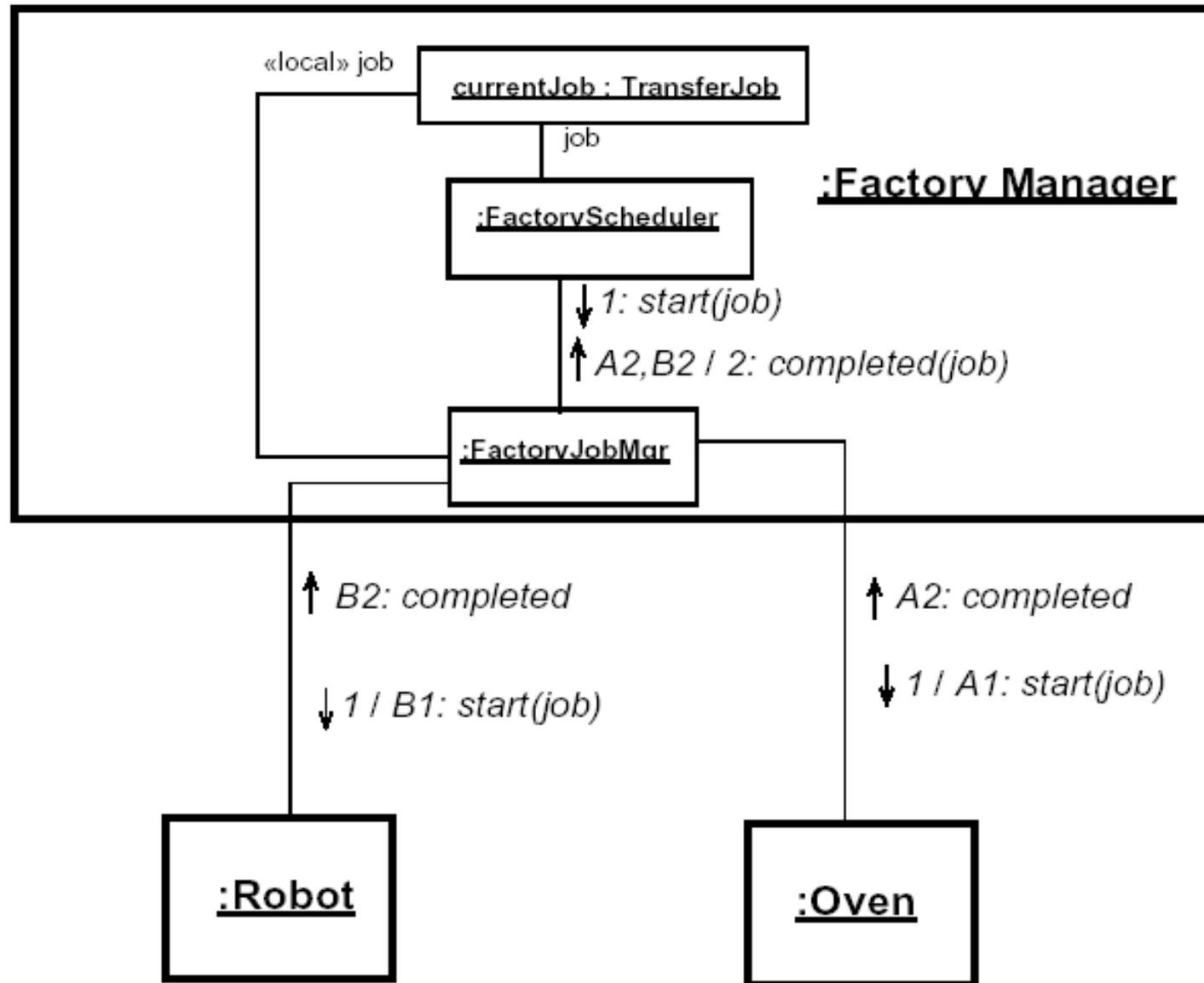


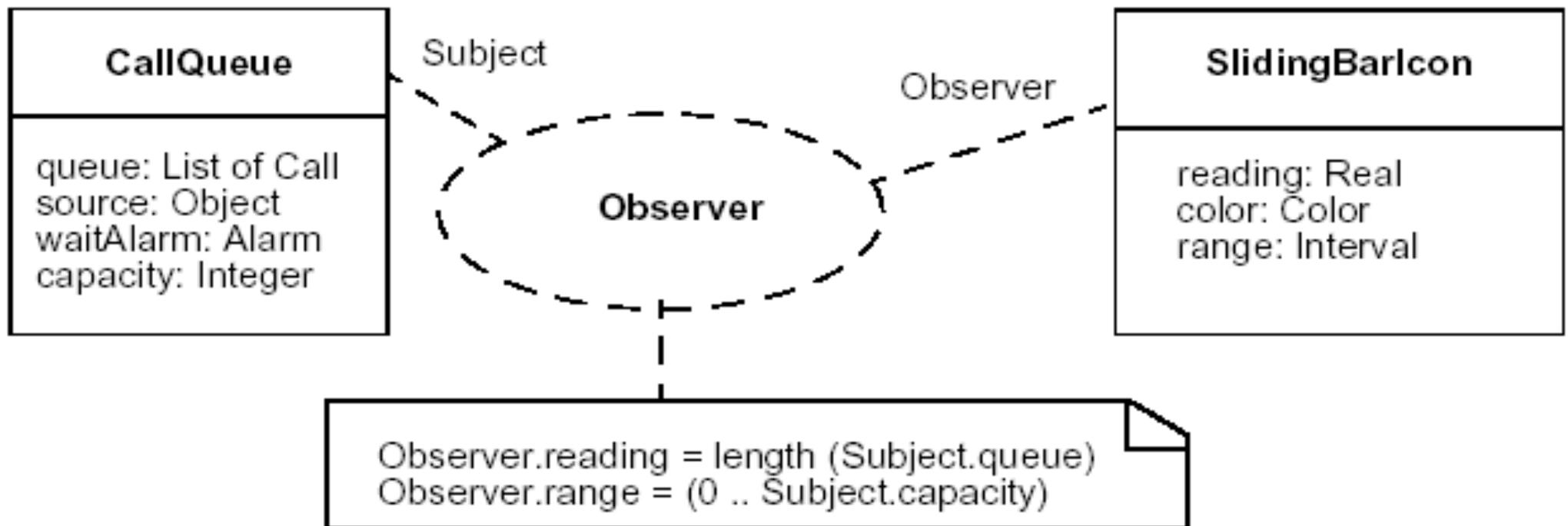
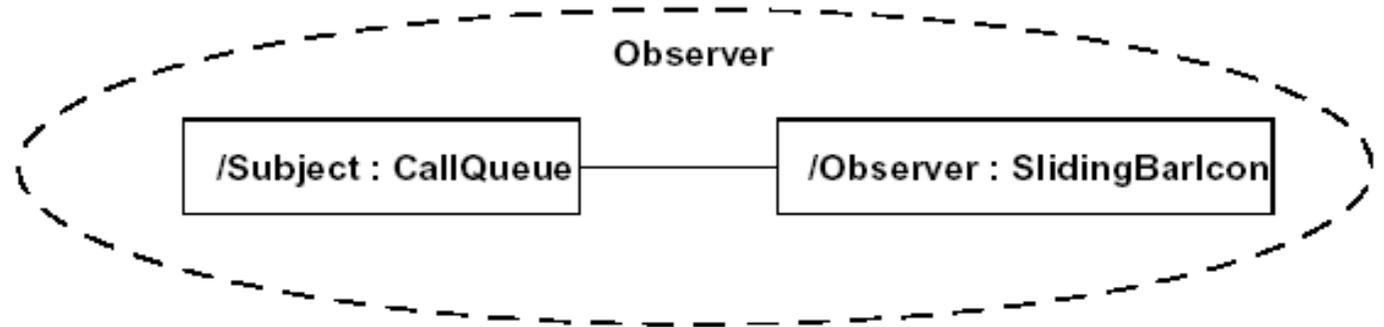
Diagramme de collaboration

■ Avec c



Représentation des patterns par des collaborations

■ Vue interne



Exemple

