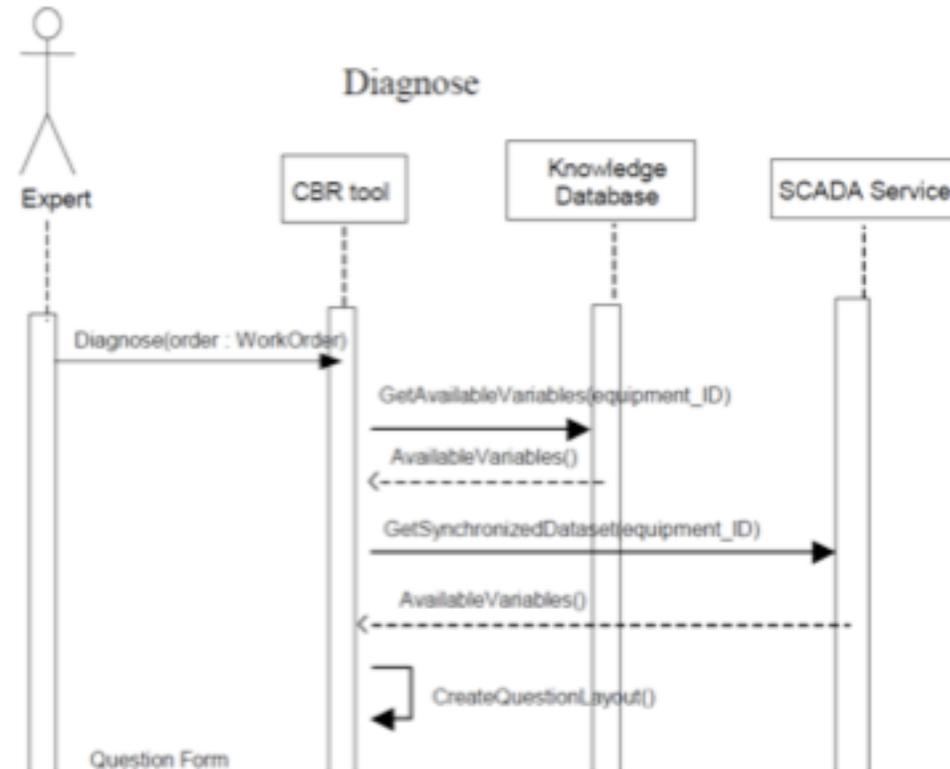

Annexe - cours 8-2

UML : lien analyse / conception

Modélisation dynamique : liens avec la modélisation statique, Contrat d'opération

© **V. Deslandres, Master 1 Miage - Module MOOCSI**
Université Lyon1

- Lien avec la modélisation statique (s.4)
- Les contrats d'opération (s.11)





Activités vs. Séquences

- Rappel : les 2 modélisent des **processus**
- **Diagramme d'ACTIVITÉS (DIT)**
 - Description **fonctionnelle**, peu de liens avec le monde Objet (à part les *nœuds d'objets* impactés par les activités)
 - Souvent utilisé en début d'analyse, pour expliciter un cas d'utilisation
- **Diagramme de SEQUENCES Objets (DSQ)**
 - Utilisé surtout pour l'analyse **détaillée**
 - Et **en conception**



Modélisation dynamique :

LIEN AVEC LA MODÉLISATION STATIQUE

Modélisation dynamique : apport

- La modélisation dynamique est particulièrement **enrichissante** pour les analystes et les utilisateurs.
- Le **diagramme de classes** s'en trouve grandement **complété**
- C'est pourquoi on développe généralement les deux modèles **de front**

Enrichissement du diagramme de classes

La modélisation dynamique *enrichit* le diagramme de classes :

- **état → valeur d'un ou plusieurs attributs**

- État d'une commande : créée, enCours, enLitige, retournée, archivée
- Etat médical d'un patient : les valeurs de ses mesures biologiques (pression, température, etc.)

- Les **messages** se traduisent en :

- **Opération** de classe
 - Lancer (un moteur) → méthode du Moteur
 - (en général la classe qui *reçoit* le message)
- **Changement de valeur** d'attributs

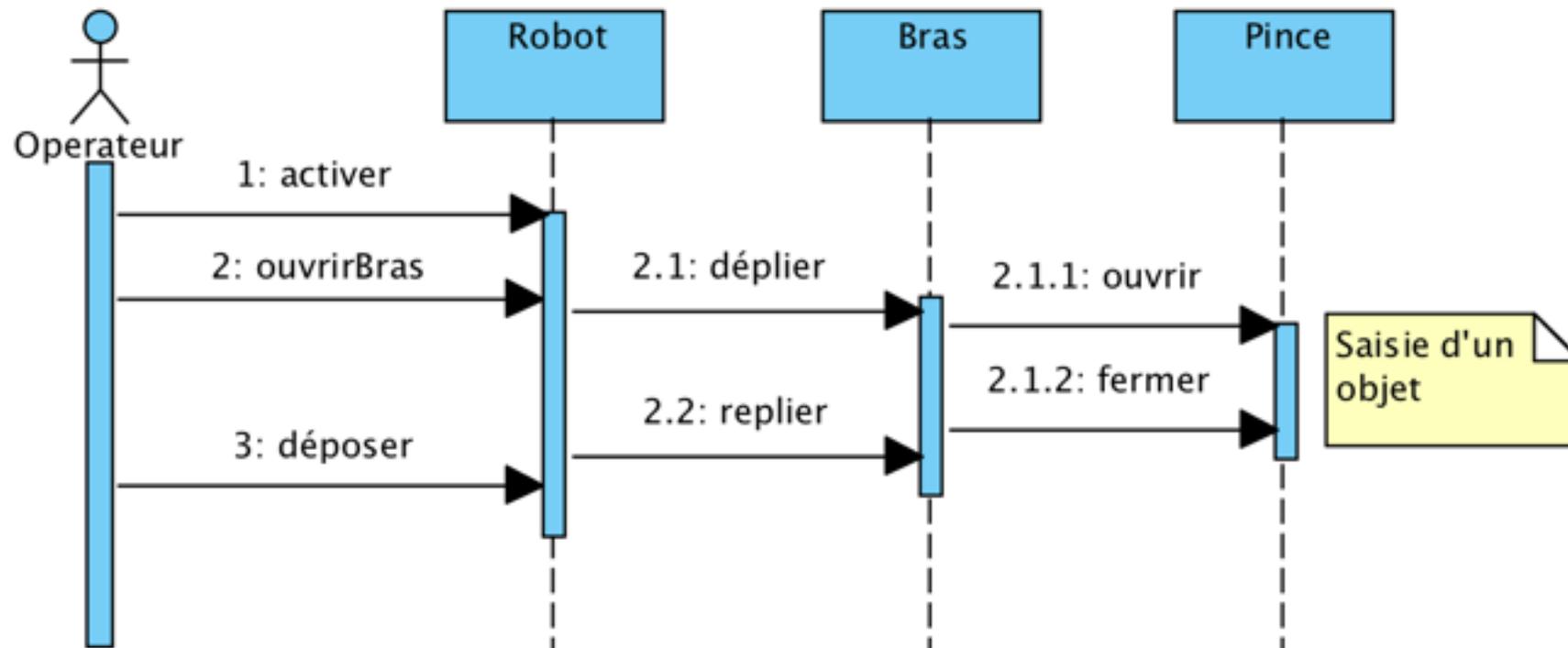


- Les **conditions**

- → **attributs** ou **arguments** des méthodes

Ex. DSQ → DCL

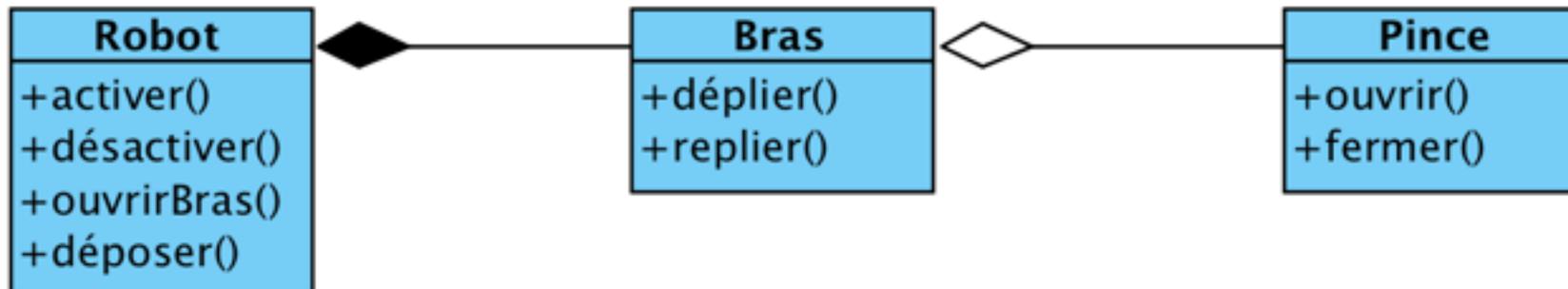
- Un robot une fois activé, va déplier son bras articulé et ouvrir une pince pour saisir un objet



DCL initial



DCL enrichi des opérations du DSQ

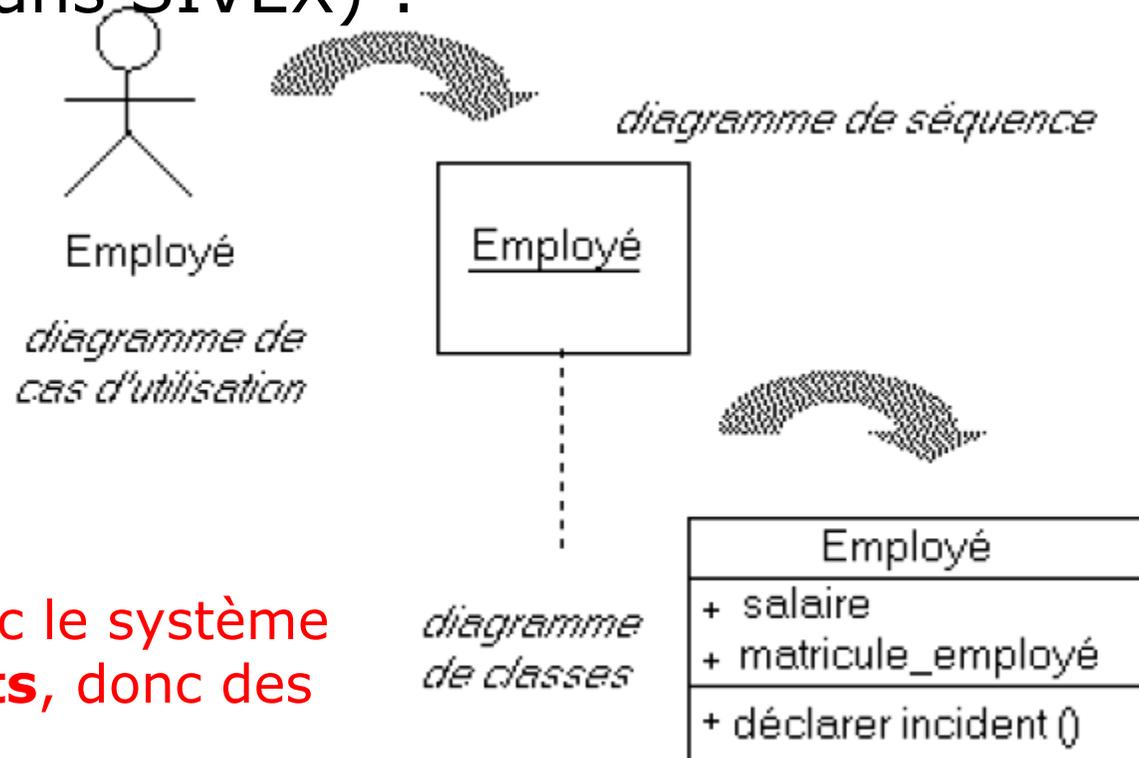


Lien acteur / objet / classe

- Parfois un **acteur externe** du diagramme des cas d'utilisation, peut se retrouver aussi comme composant de l'application (par ex. un employé qui déclare un incident dans SIVEX) :

Seuls **certains** acteurs externes du DCU sont aussi des classes : ceux pour lesquels on conserve des données

Ils interagissent avec le système par des **événements**, donc des appels de méthodes.



Privé/public en analyse – Opération/méthode

- Dans UML sauf indication contraire :
 - les attributs sont privés
 - les méthodes sont publiques

VOC : on parle d'**Opération** en Analyse (→ abstraction)

- Opération SaisirArticle() pour une vente

de **Méthode** en Conception / Implémentation

- Appel aux méthodes modeSaisie() pour gérer l'affichage de la vue, puis saisirArticle(Article unArticle)

« Contrats » d'opération

- Une fois la modélisation stabilisée
 - on commence la rédaction des « contrats » pour les opérations complexes
- Un contrat = spécifications d'une opération par ses pré et post-conditions :
 - 3 catégories de Post-conditions =
 - Création et suppression d'instances
 - Modification d'attributs
 - Formation ou rupture d'associations

Programmation « par contrats »

- Introduite par Bertrand MEYER fin des années 80 avec le langage Eiffel
- **Objectif : qualité logicielle + réutilisation**
- Eiffel : langage de description assez formel (doit être implémenté ensuite)
 - Mécanismes d'assertions (pré/post, invariants)
 - Typage fort
 - Moins formel que des langages de spécification ou de preuve formelle
 - Comme les langages Z (ou B)

Utilisation pour UML

- OMG : les contrats sont cohérents avec le « **Processus Unifié** » de modélisation UML
 - = démarche de modélisation pour les gros projets
 - démarche itérative et incrémentale, avec analyse de risque à chaque itérations, etc.
- On les utilise surtout au niveau des opérations :
contrats d'opération

Ex. de contrat d'opération (en analyse)

- Contrat **SaisirArticle**(*code* : CodeA, *quantité* : int)
- Ref. cas d'utilisation « **Traiter une Vente** »
- Pré-condition :
 - il existe une vente en cours (*Vente*)
- Post-conditions :
 - ✓ Une instance de *LigneArticle* a été créée : *LigA*
 - ✓ *LigA* a été associée à la vente en cours
 - ✓ *LigA.quantité* est devenue *quantité*
 - ✓ Des attributs de *Vente* ont été initialisés

Création d'instance

Formation d'association

Modification d'attributs

(Vente précédente : classes)

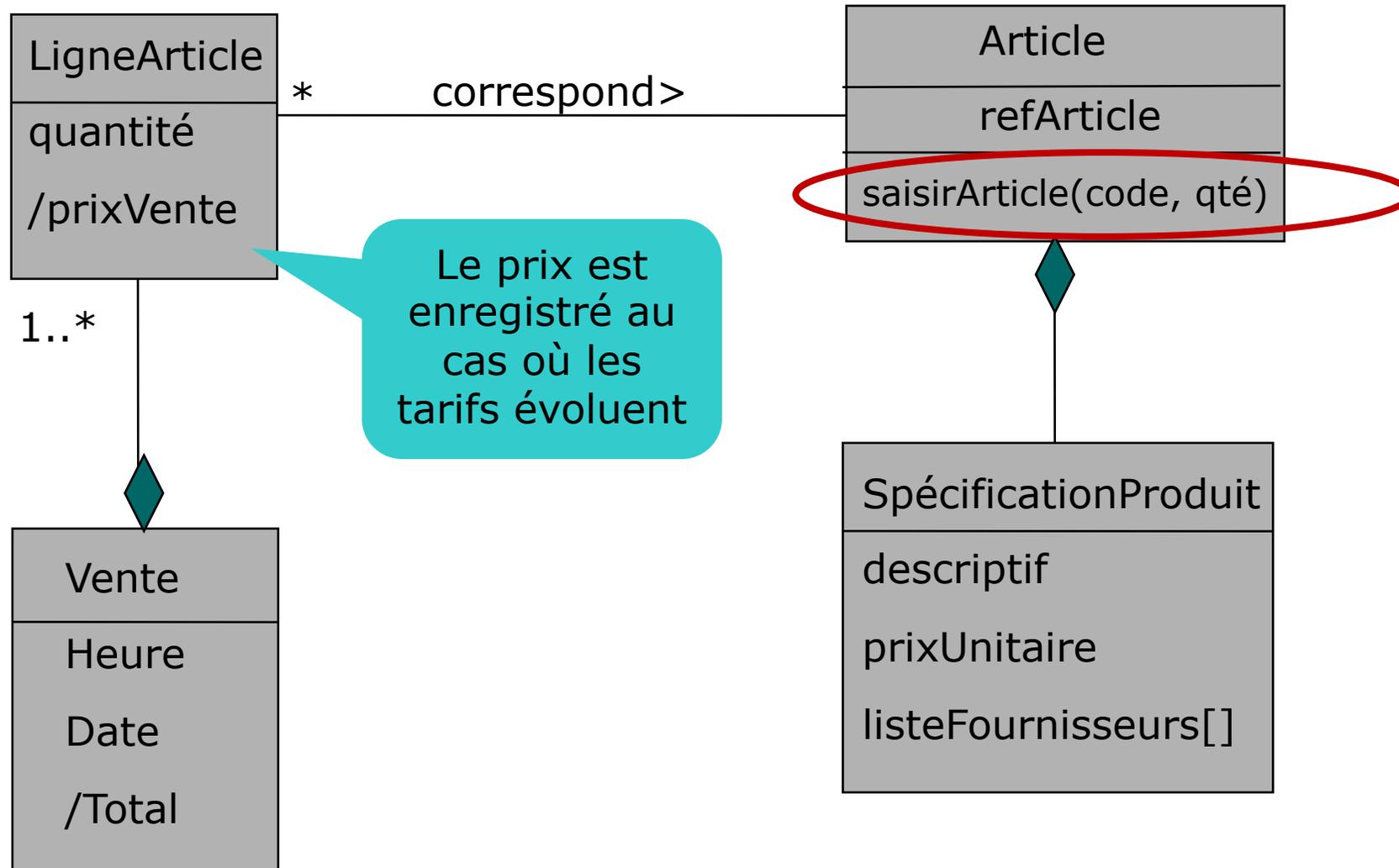


Illustration Conception / Contrats d'opération

- Etude de Cas Caisse de Magasin