

UML

Aspects statiques

Compléments pour la conception

Emmanuel Pichon
2013

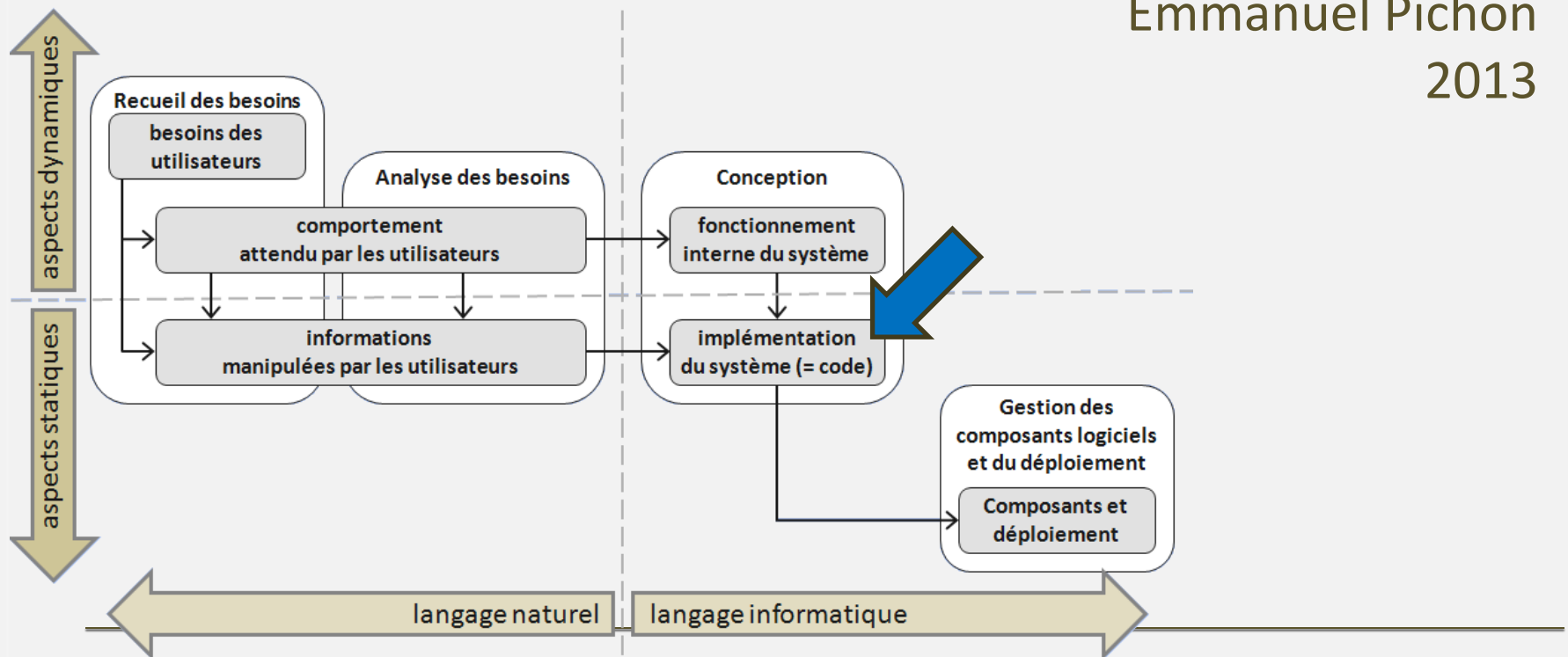


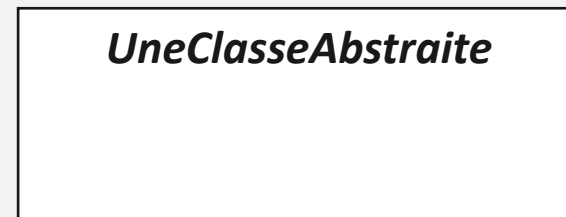
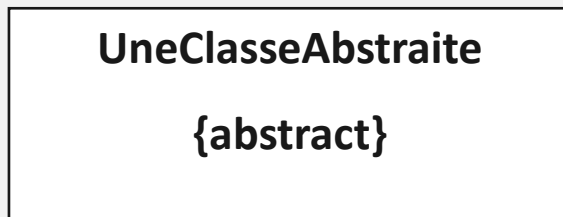
Diagramme de classes (*class diagram*)

Classe abstraite (*abstract class*)

◎ Sens

- Classe ne pouvant pas créer d'objet
- Classe utilisée pour factoriser, via un héritage, les opérations, les attributs et les relations communs à plusieurs classes

◎ Notations UML



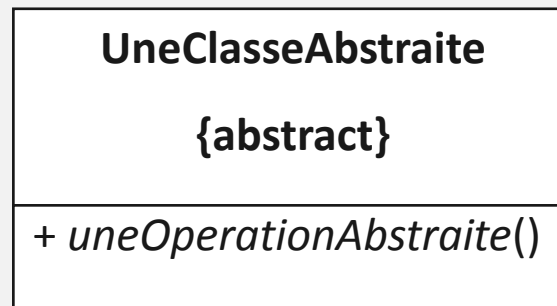
(nom en italique)

Opérations abstraites

◎ Sens

- Déclaration d'opération sans implémentation directe
- L'opération doit être redéfinie par les sous-classes

◎ Notation UML : libellé de l'opération est en italique



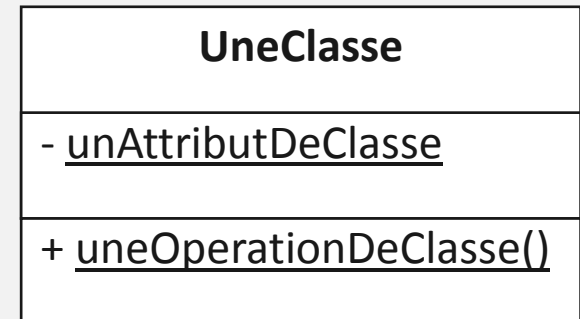
Opération et attribut de classes (*static feature*)

◉ Sens

- Opération dont le résultat est identique pour tous les objets
- Attribut dont la valeur est identique pour tous les objets

◉ Notation UML

- le nom de l'opération ou de l'attribut est souligné



◉ NB : une opération de classe n'a pas accès aux attributs de ses objets

◉ En Java : méthodes et variables de classes (mot-clé *static*)

- <visibilité> static <type> <nom de variable>;
- <visibilité> static <type> <nom de méthode> {...}

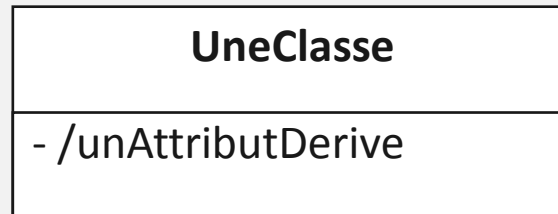
Attributs et associations dérivés (*derived property, derived association*)

◎ Sens

- Attributs pouvant être calculés à partir d'autres attributs
- Associations pouvant être calculés à partir d'autres associations

◎ Notation UML :

- / entre la visibilité et le nom de l'attribut



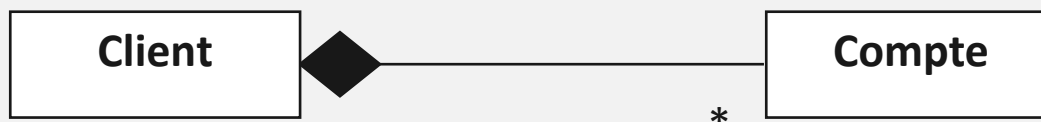
- / entre la visibilité et le nom du rôle pour les associations

Composition (*composite aggregation*)

- ◎ Sens : version forte de la relation d'agrégation impliquant des contraintes entre les cycles de vie des objets
 - L'objet « contenant » doit exister avant l'objet « contenu »
 - L'objet « contenant » est responsable de la création de l'objet « contenu »
 - La destruction de l'objet « contenant » implique la destruction de l'objet « contenu »

- ◎ Notation UML

- Un diamant plein du côté de l'objet contenant



Composition (*composite aggregation*)

Impacts sur la création d'objets en Java

◎ Solution fréquemment citée

- Le constructeur de l'objet contenant intègre la création de l'objet contenu

◎ Solution avec encapsulation

- la classe de l'objet contenant possède un getteur pour l'objet contenu
- Ce getteur crée l'objet contenu au premier accès

Composition (*composite aggregation*)

Impacts sur la destruction d'objets en Java

- ⊙ Dans la classe de l'objet contenant
 - `finalize()` doit prendre en charge la destruction de la référence vers l'objet contenu
 - Exemple : `objetContenu = null;`
- ⊙ Le Garbage Collector détruira
 - L'objet contenant
 - L'objet contenu
 - (s'il ne reste aucune référence à ces objets)

Caractérisation de la cardinalité *

- ⊙ Définition : par défaut * indique la présence d'un ensemble d'objets
 - Groupe non ordonné avec unicité (pas de doublons)
 - Équivaut à {unique, unordered}
- ⊙ Principales caractérisations possibles
 - {nonunique} = doublons autorisés
 - {ordered} = ordonné

nonunique	ordered	Correspondance Java
-	-	Set
-	●	SortedSet
●	-	Collection
●	●	List

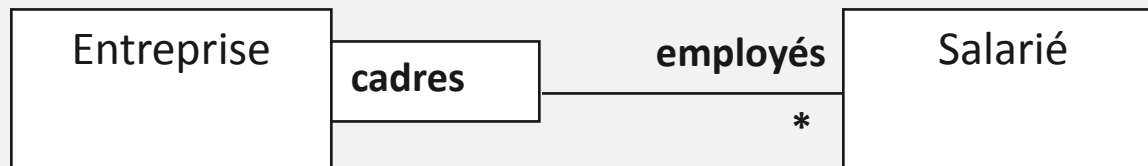
Qualifieur (*qualifier*)

⊙ Définition

- Pour les associations 1-vers-* ou *-vers-*
- Filtre permettant de sélectionner un sous-ensemble coté *

⊙ Notation UML

- Un rectangle du coté de l'objet accédant à une liste

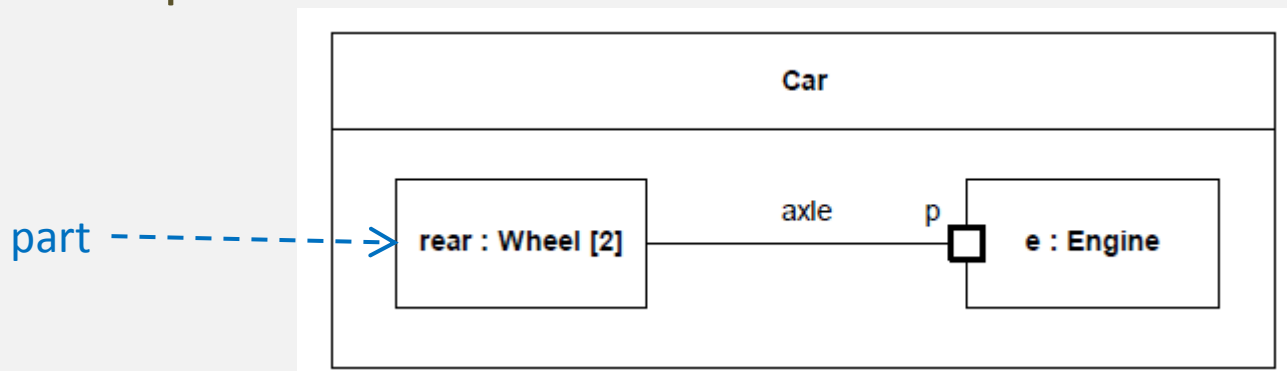


- ⊙ Dans l'exemple, l'association vers les salariés est limitée aux cadres de l'entreprise

Diagramme de structure composite (*composite structure diagram*)

Les structures composites

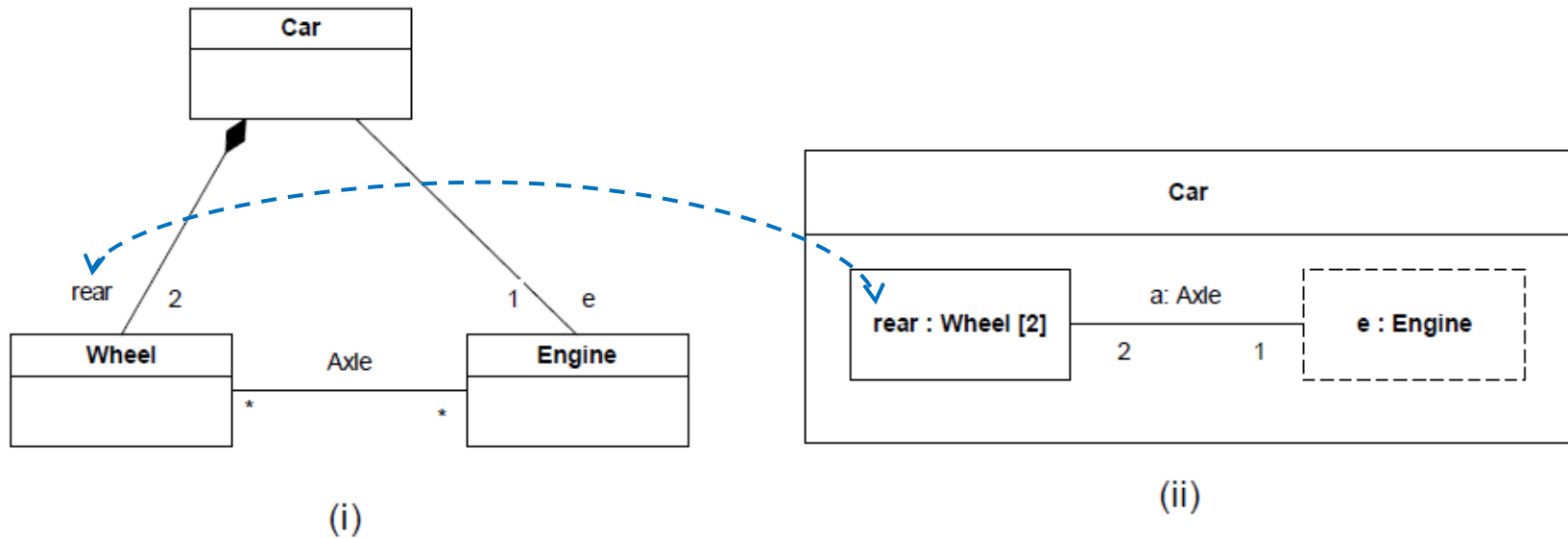
- ◎ UML 2 intègre des notions issues de la méthode ROOM
 - Décomposition d'une structure en « *parts* »
 - Les « *ports* » définissent les points de communication entre la structure interne et son environnement
 - Les « *connectors* » spécifient les liens de communication internes
- ◎ Exemple



Correspondance avec le diagramme de classes

⊙ Diagramme de classes

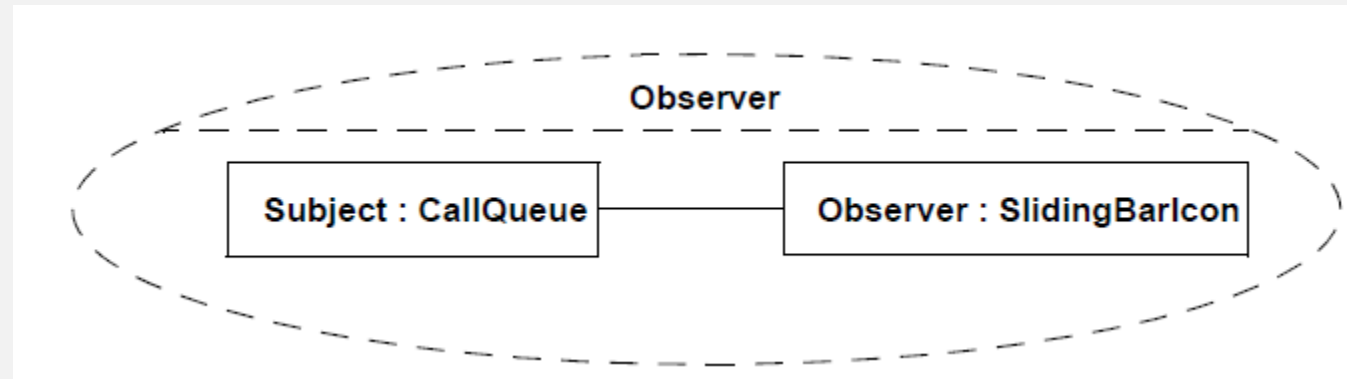
Diagramme de structure composite



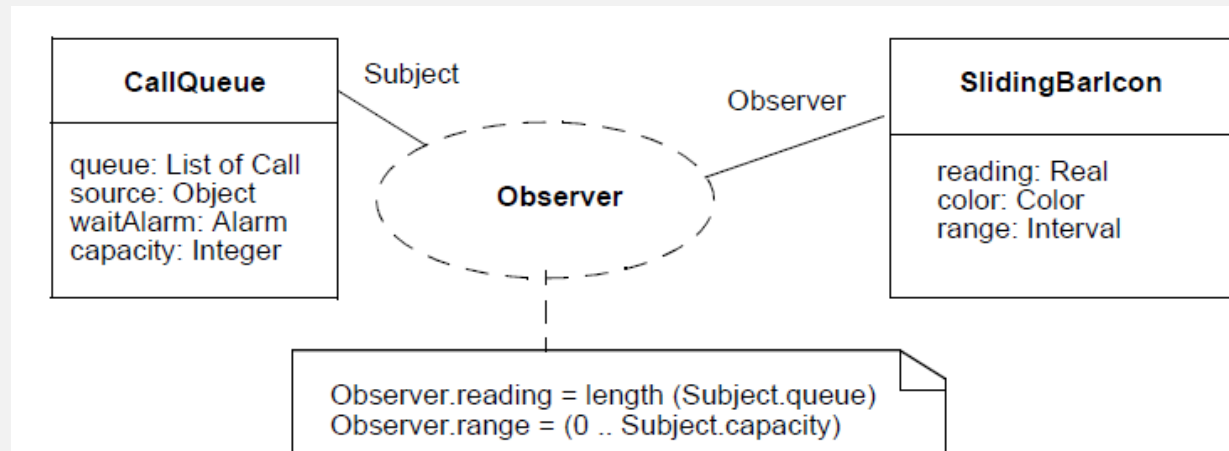
- ⊙ Notations équivalentes mais le diagramme (ii) définit en plus un contexte permettant d'appliquer des règles locales
 - Les « parts » correspondent aux rôles (cf. rear et e)
 - Un « part » est en pointillé si agrégation partagée (cf. e)

Collaboration d'objets

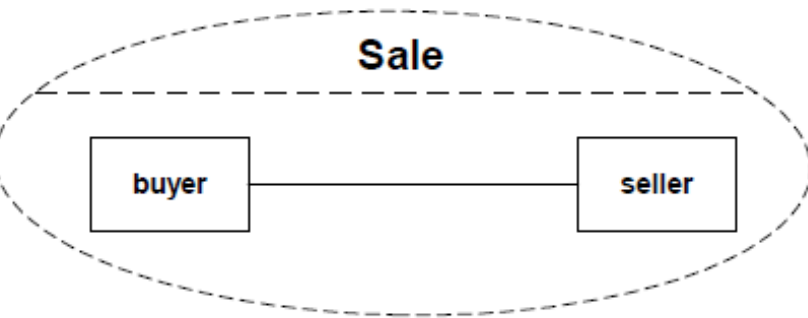
◎ Notation UML



◎ Notation alternative



Utilisation d'une collaboration



<- Collaboration

- Utilisation de cette collaboration

