### Fiche uml

# Use case :

* Relation d’utilisation : <<**include**>>
  + Le cas d’utilisation **contient** un autre cas d’utilisation
* Relation d’extension : <<**extend**>>
  + Le cas d’utilisation **étend** (précise) les objectifs (le comportement) d’un autre cas d’utilisation

#### -Acteurs principaux et secondaires

Un acteur est qualifié de *principal* pour un cas d’utilisation lorsque ce cas rend service à cet acteur. Les autres acteurs sont alors qualifiés de secondaires. Un cas d’utilisation a au plus un acteur principal. Un acteur principal obtient un résultat observable du système tandis qu’un acteur secondaire est sollicité pour des informations complémentaires. En général, l’acteur principal initie le cas d’utilisation par ses sollicitations. Le stéréotype << primary >> vient orner l’association reliant un cas d’utilisation à son acteur principal, le stéréotype << secondary >> est utilisé pour les acteurs secondaires (figure [2.6](http://laurent-audibert.developpez.com/Cours-UML/html/Cours-UML011.html#fig_ex_association)).

# Diagrammes de classe :

Relation de dépendance : (flèche tirée)

* relation d'utilisation unidirectionnelle
* une modification de l'élément dont on dépend, peut nécessiter une mise à jour de l'élément dépendant

Agrégation :

Représente une relation de type "**ensemble / élément**"

* Une agrégation peut notamment (mais pas nécessairement) exprimer :
  + qu'une classe (un "élément") fait partie d'une autre ("l‘ensemble"),
  + qu'un changement d'état d'une classe, entraîne un changement d'état d'une autre,
  + qu'une action sur une classe, entraîne une action sur une autre

Composition :

* **contenance physique**
* Représente une relation de type "**composé / composant**"
* Les **cycles de vies** des composants et du composé sont **liés** : si le composé est détruit (ou copié), ses composants le sont aussi
* **ne peut être liée qu'à un seul composé**

Contrainte :

* Les contraintes sont des expressions qui précisent le rôle ou la portée d'un élément de modélisation (elles permettent d'étendre ou préciser sa sémantique)
* Sur une association, elles peuvent restreindre le nombre d'instances visées ("expressions de navigation")
* Les contraintes peuvent s'exprimer en langage naturel. Graphiquement, il s'agit d'un texte encadré d'accolades

Exemple :

1. Initialement, toutes les personnes doivent être des femmes.

context Person

init : self.gender = Gender::female

2. Initialement le salaire d’un employé est de 1000 euros.

context Job

init : self.salary = 1000

3. Le directeur d’une société doit avoir plus de 40 ans.

context Company::manager : Person -- la notation ‘::’ sert à sélectionner comme contexte un attribut ou une méthode de la classe Company

inv : self.age >= 40

4. L’âge d’un individu est fonction du moment présent et de sa date de naissance.

context Person

derive : self.age = Date.now() – self.birthdate

5. Si le chiffre d’affaires (turnover) d’une société est supérieur à 1 million d’euros, elle doit avoir plus que 10 employés.

context Company

inv : NOT self.turnOver() > 1.000.000 OR self.numberOfEmployees > 10

6. Tous les employés d’une société doivent avoir plus de 18 ans.

context Job

inv : self.employee.age >= 18

7. On ne peut pas commencer à travailler avant sa date de naissance.

context Job

inv : self.startDate > self.employee.birthDate

8. Un homme est marié avec une femme et une femme avec un homme.

context Person

inv : not self.isMarried OR

(self.gender = Gender::male AND self.wife.gender = Gender::female) OR

(self.gender = Gender::female AND self.husband.gender = Gender::male)

Mais aussi (peut être mieux)

context Marriage

inv : self.husband.gender=Gender::male AND self.wife.gender=Gender::female

9. Pour être mariés, il faut avoir plus que 18 ans.

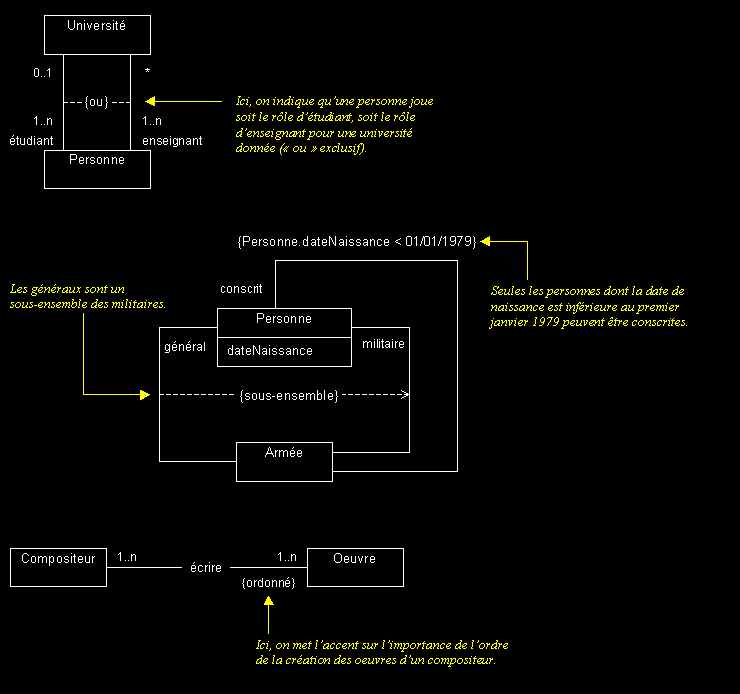
context Person

inv : NOT self.isMarried OR self.age >= 18

10. On ne peut pas commencer à travailler le jour de son mariage.

context Job

inv : NOT self.startDate = self.employee.marriage.date

Exemple : 

# Diagramme séquence :

* Permettent de représenter des interactions
  + Entre objets (et acteurs)
  + Selon un point de vue temporel (chronologie des envois de messages)