

# SLAM2

POO/UML

# UML

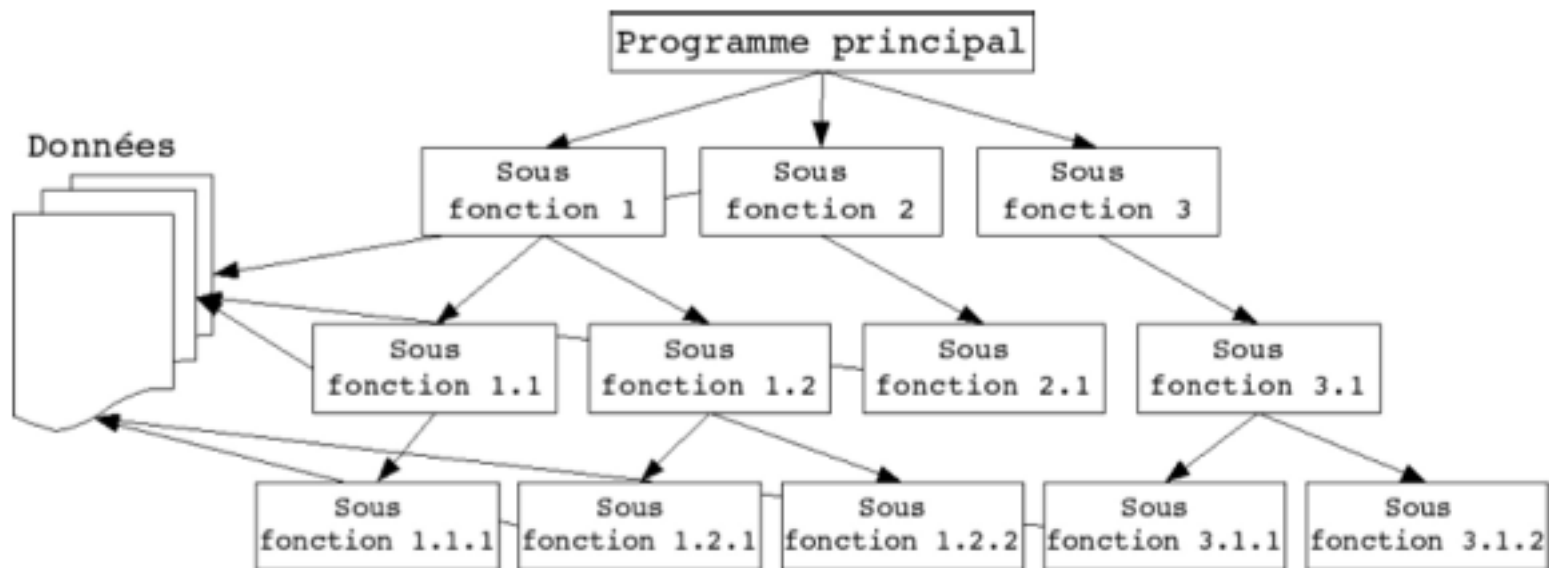
- \* Unified Modeling Language
- \* Langage de description (pas de norme de description des étapes de modélisation)
- \* Agile
- \* Normalisé
- \* International
- \* Métalangage (langage du projet à développer)
- \* Comporte 13 types de diagrammes

# UML

- \* Ne gère pas les problèmes de redondance
- \* Ne modélise pas les BD : pas de gestion de la persistance
- \* POO + SGBDR = SGBDOO
- \* UML adapte la BD aux traitements
- \* Plusieurs approches pour un même problème
- \* Attributs multivalués : non atomique
- \* Agrégation et composition

# Approche UML

\* Non fonctionnelle



# Les diagrammes UML

## \* Diagrammes statiques :

- diagramme de classes (*Class diagram*)
- diagramme d'objets (*Object diagram*)
- diagramme de composants (*Component diagram*)
- diagramme de déploiement (*Deployment diagram*)
- diagramme de paquetages (*Package diagram*)
- diagramme de structures composites (*Composite structure diagram*)

# Les diagrammes UML

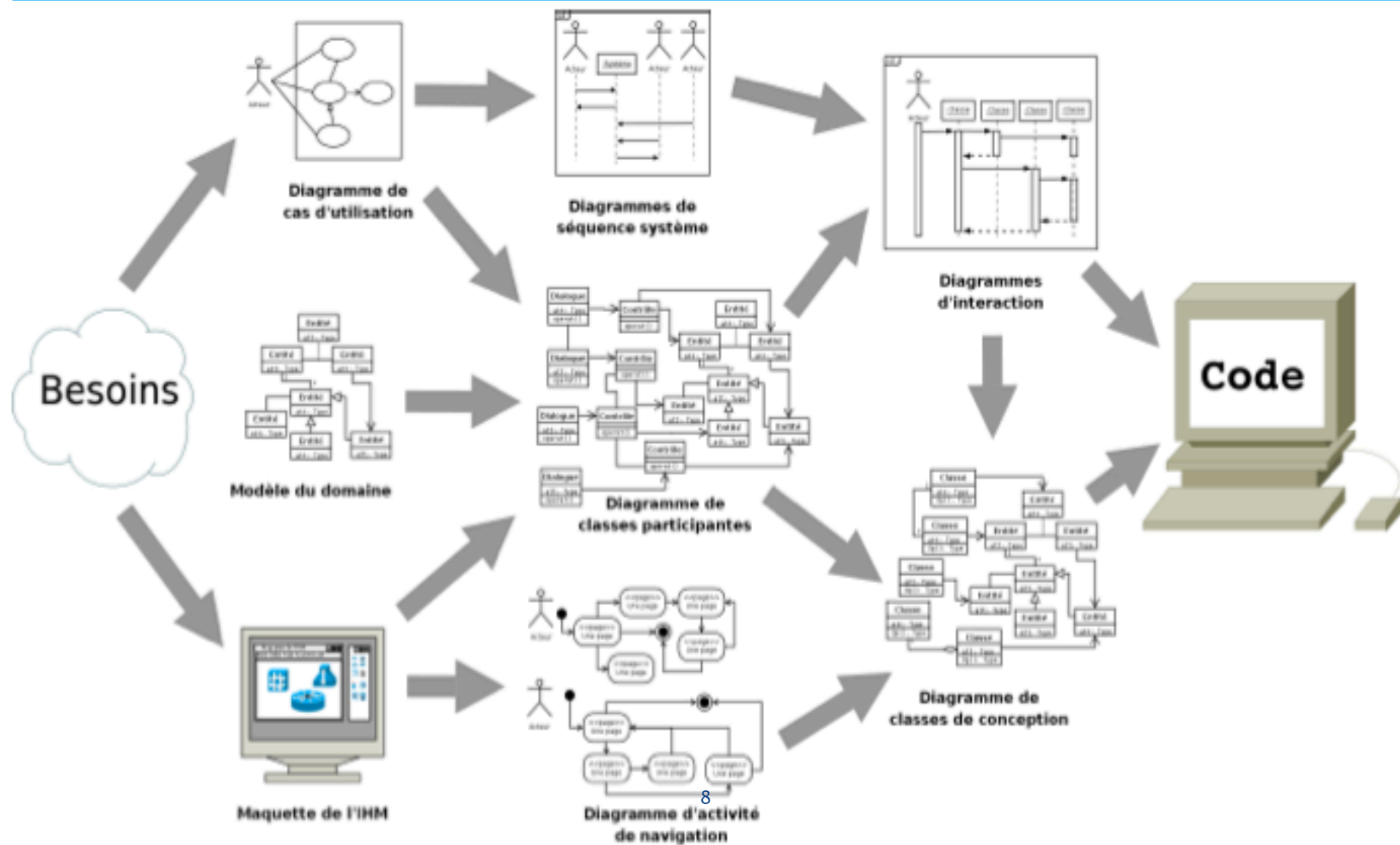
## \* Diagrammes dynamiques :

- diagramme de cas d'utilisation (*Use case diagram*)
- diagramme d'activités (*Activity diagram*)
- diagramme d'états-transitions (*State machine diagram*)
- **Diagrammes d'interaction (*Interaction diagram*)**
  - diagramme de séquence (*Sequence diagram*)
  - diagramme de communication (*Communication diagram*)
  - diagramme global d'interaction (*Interaction overview diagram*)
  - diagramme de temps (*Timing diagram*)

# Diagrammes UML


- \* Diagrammes importants :
- \* Diagramme de classes
- \* Diagramme d'objets
- \* Diagramme de USE CASE
- \* Diagramme d'état transition
- \* Diagramme de séquence

# Chaîne de modélisation





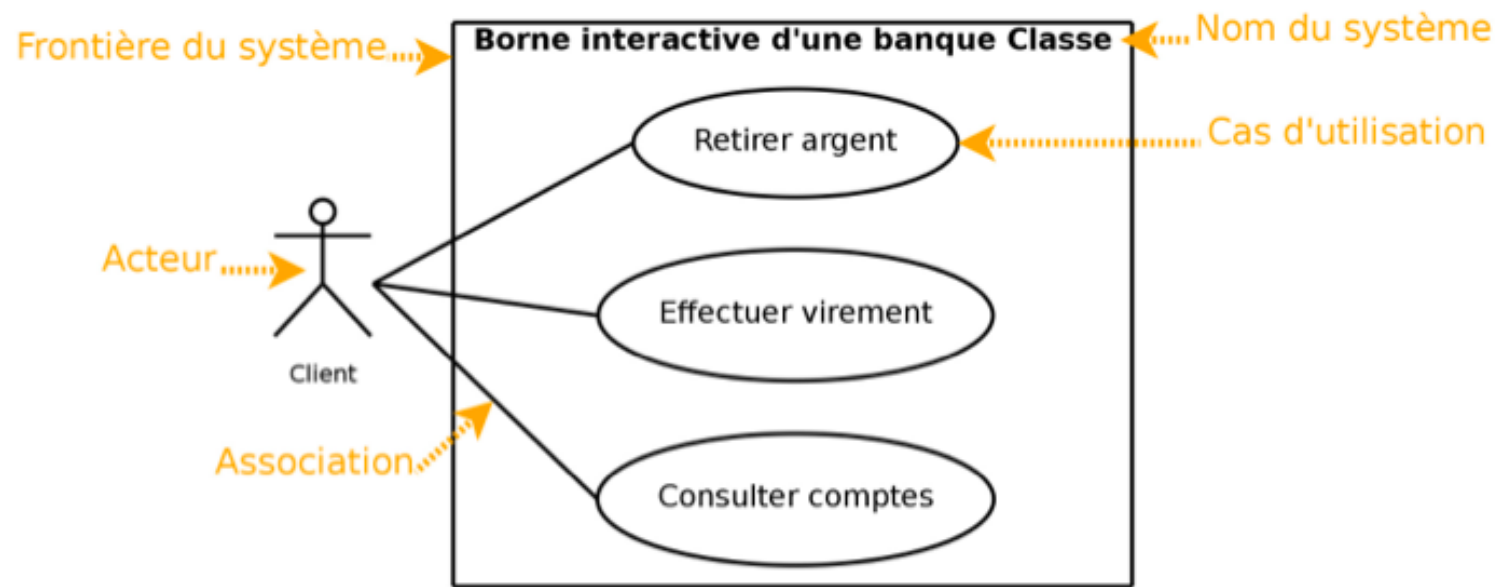
# Démarche modélisation

- 
- \* Diagramme de contexte du système à étudier (MCF)
  - \* Diagramme USE CASE pour chaque cas
  - \* Traduction des USE CASE en scénario
  - \* Traduction des scénarios en Diagramme de séquence et diagramme de collaboration
  - \* Synthèse des DSE et DCO : liste des classes et objets
  - \* DCL
  - \* DET pour chaque classe importante
  - \* GOTO étape 3 jusqu'à : niveau détail suffisant

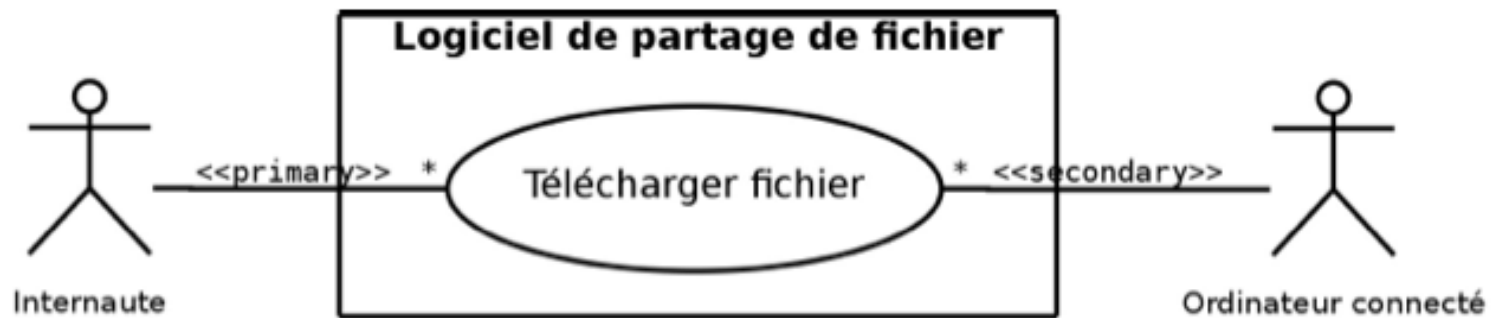
# 5 niveaux de réflexion

- \* USE case view : besoin utilisateur (comportement du système)
- \* Deployment view : contraintes physiques (AMSI, ALSI, topologie)
- \* Implementation view : configuration du système (ALSI, DAIGL)
- \* Design view : interfaces, classes de la solution
- \* Process view : méthodes, procédures, fonctions, processus (performances du système)

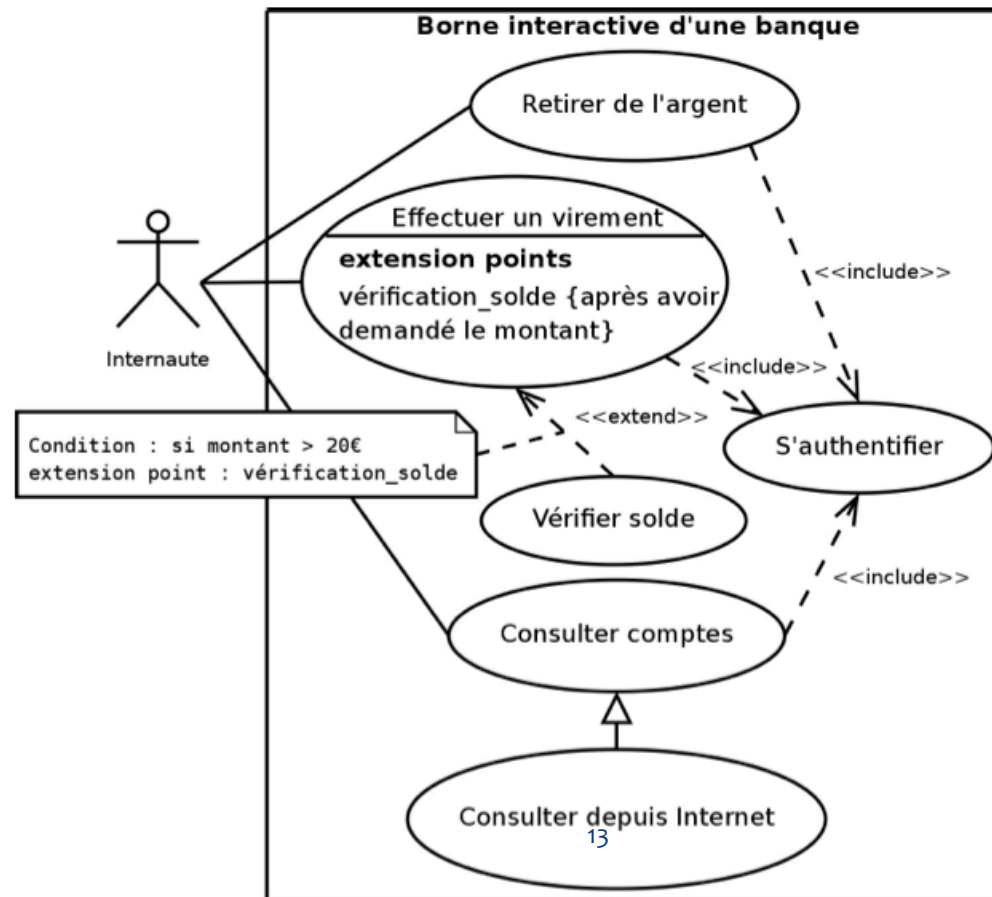
# Diagrammes USE CASE



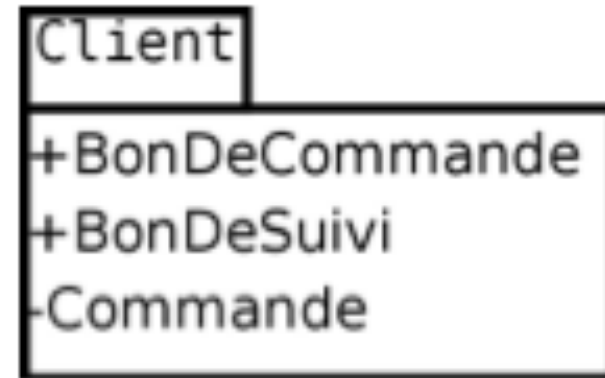
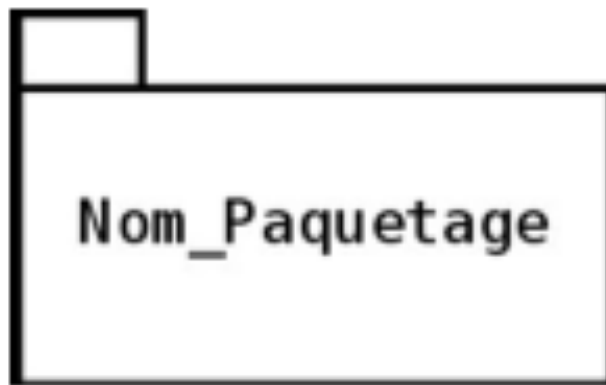
# Diagramme USE CASE



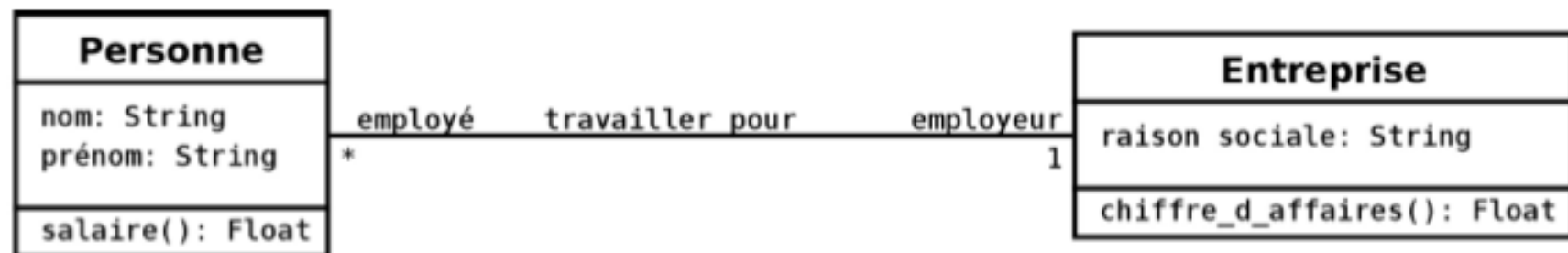
# Diagramme USE CASE



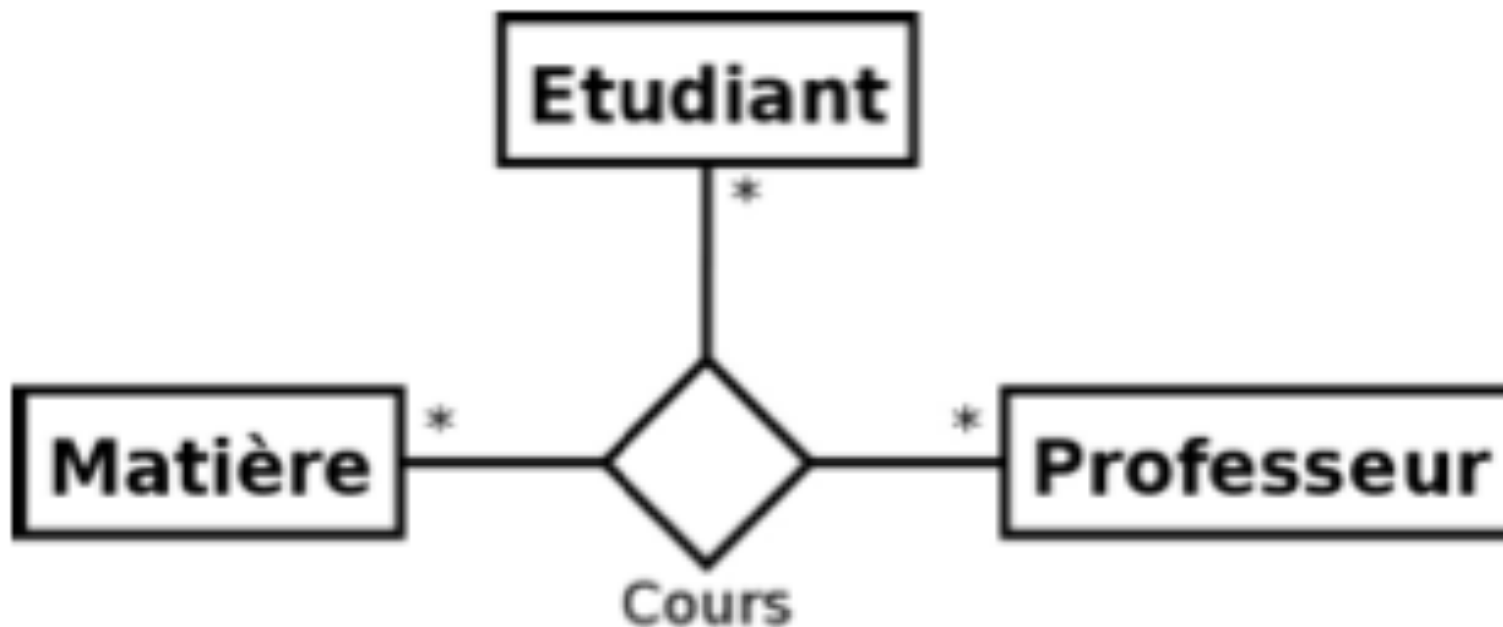
# Diagramme de paquetage



# Diagramme de classe



# Diagramme de classe





# Multiplicités

- exactement un : 1 ou 1..1
- plusieurs : \* ou 0..\*
- au moins un : 1..\*
- de un à six : 1..6

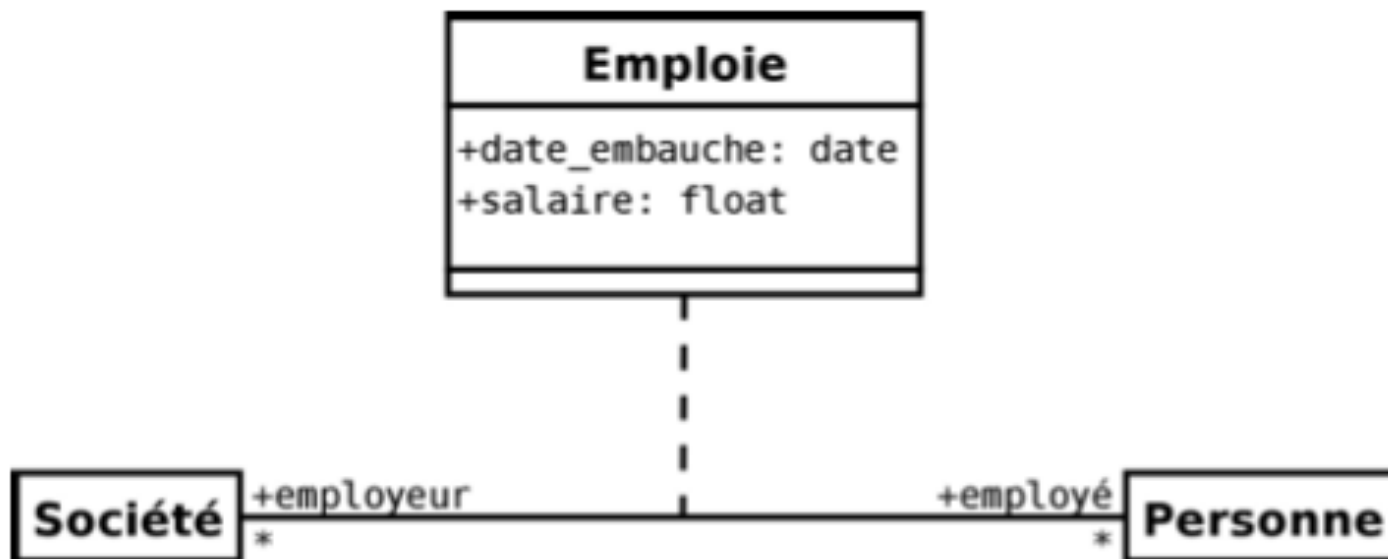
# Multiplicités

- \* Exactement : 'n' : 1, 7, ...
- \* De 'n' à 'm' : 0..1, 3..n, 1..31
- \* Plusieurs : 0..n, 0..\*
- \* n ou plus : 0..\*, 5..\*
- \* Intervalle fermé : 2, 3, 15
- \* Indéterminé : \*

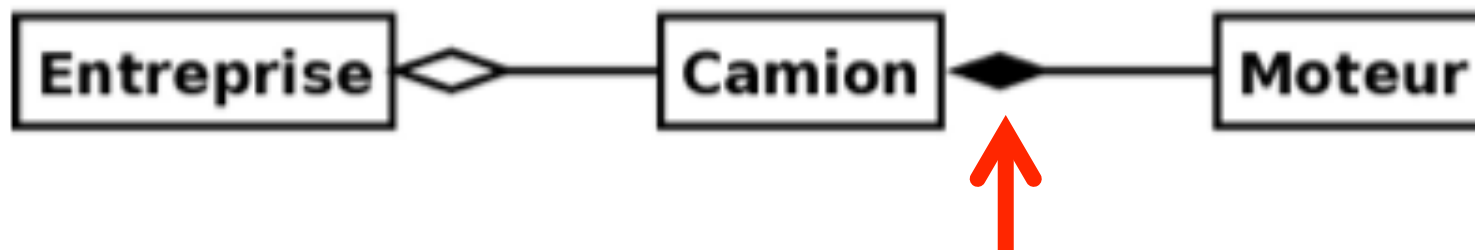
# Navigabilité



# Classe-association

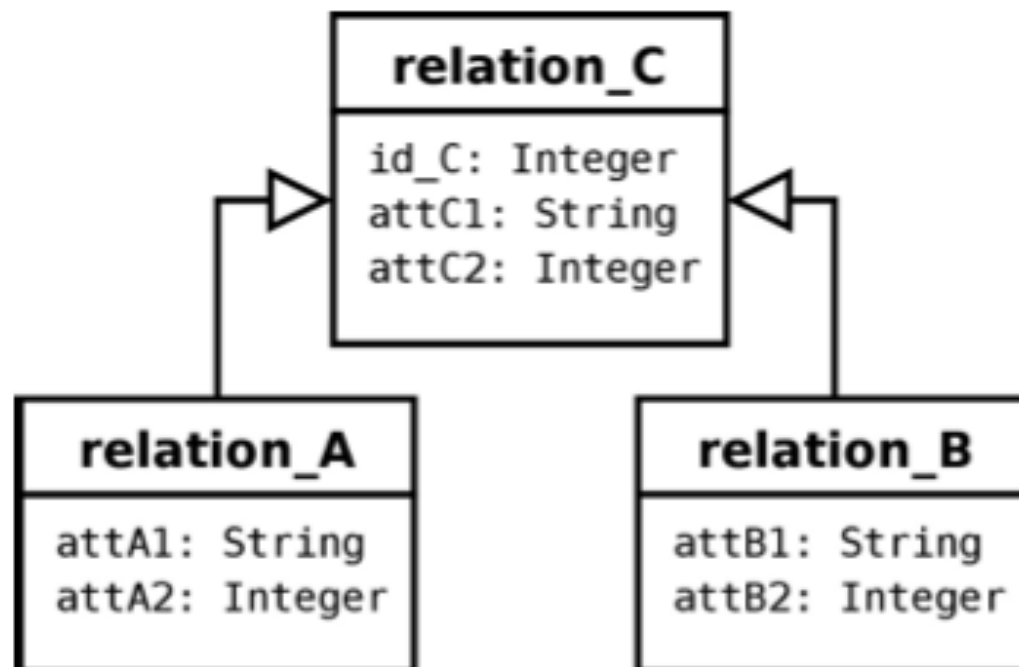


# Agrégation - Composition

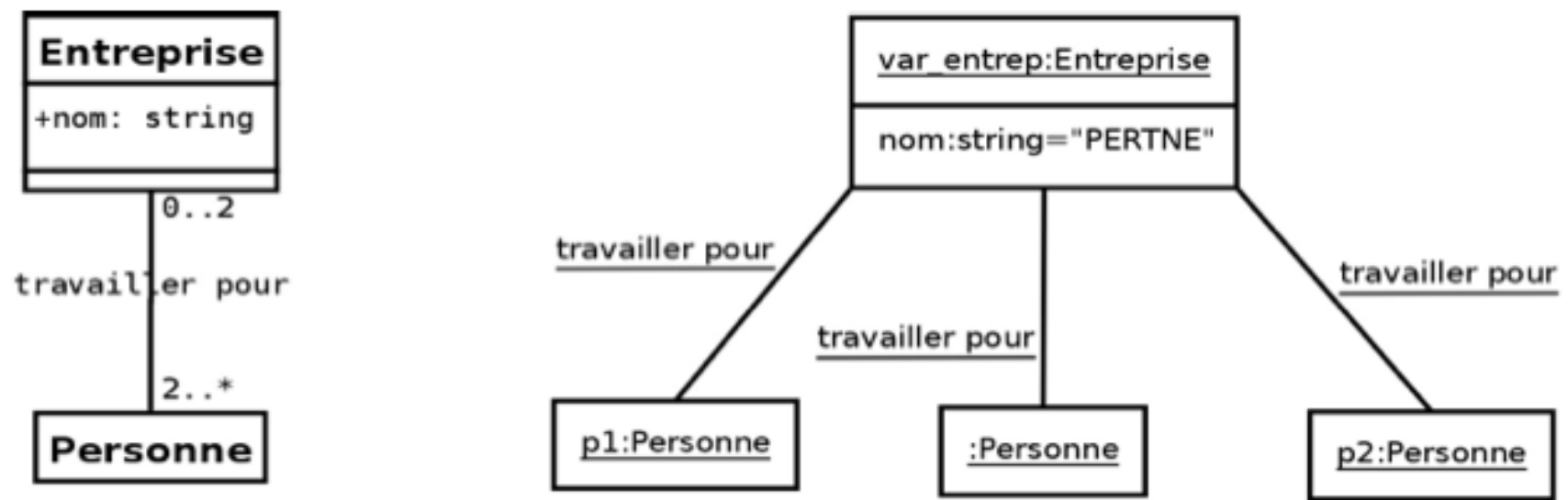


Composition = identifiant relatif MERISE

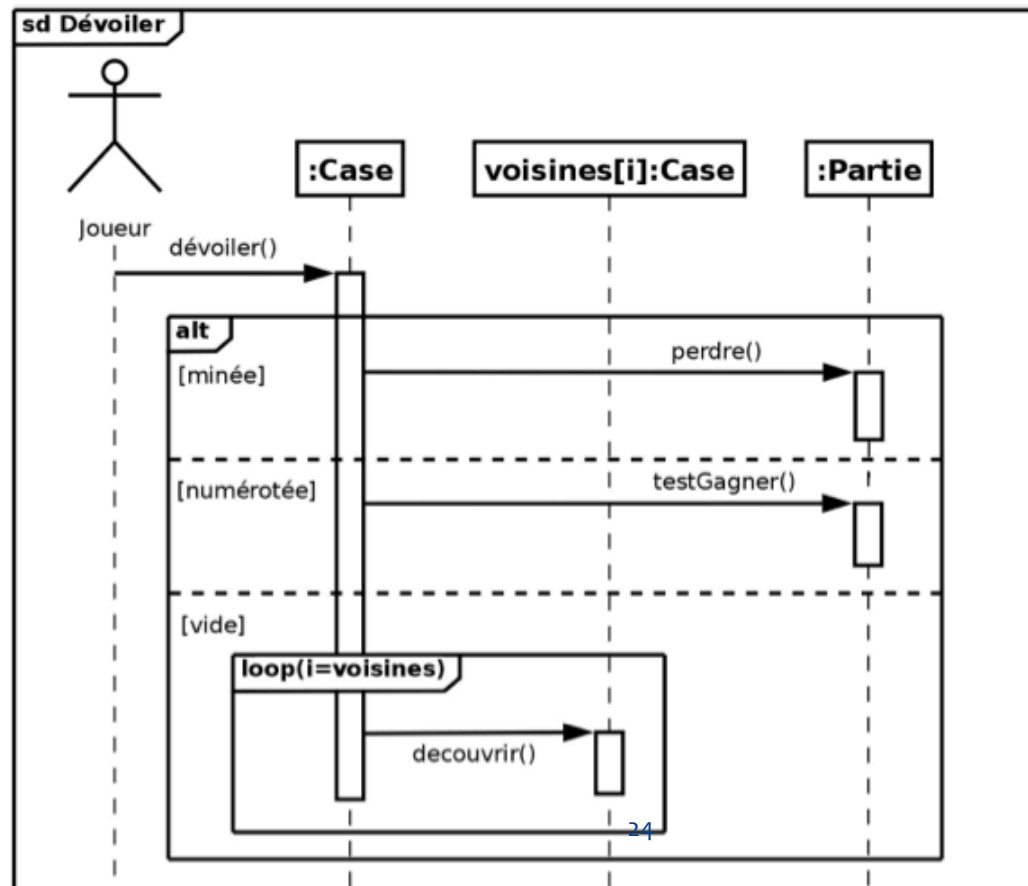
# Héritage



# Diagramme d'objet

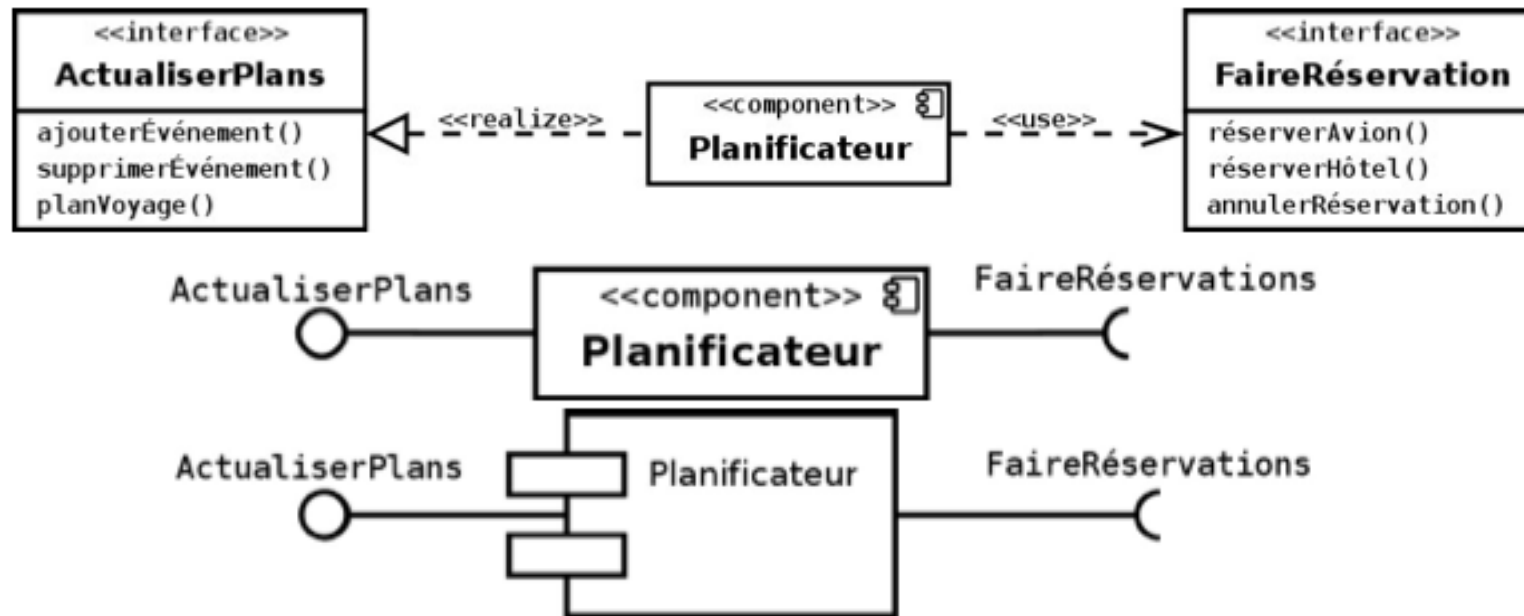


# Diagramme de séquence

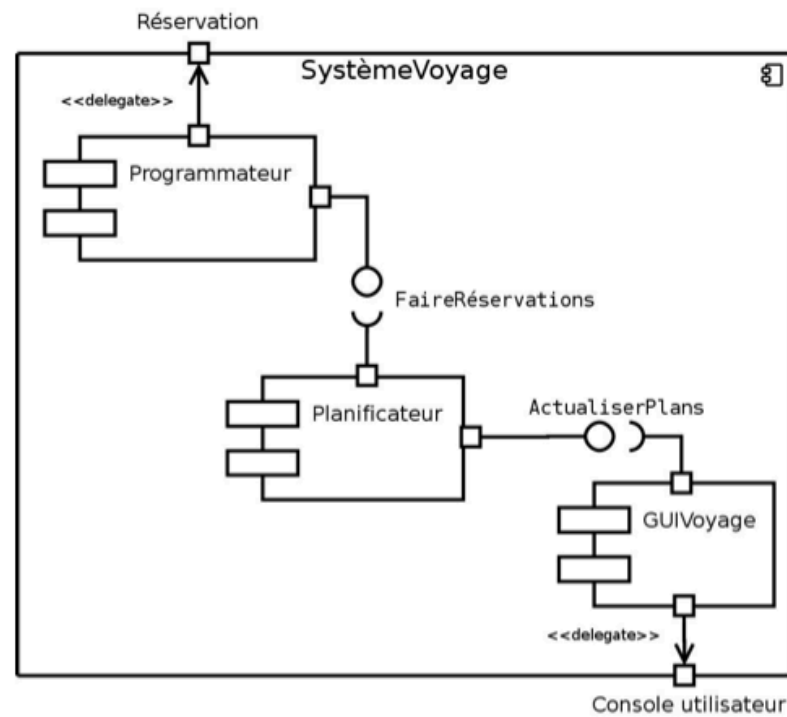




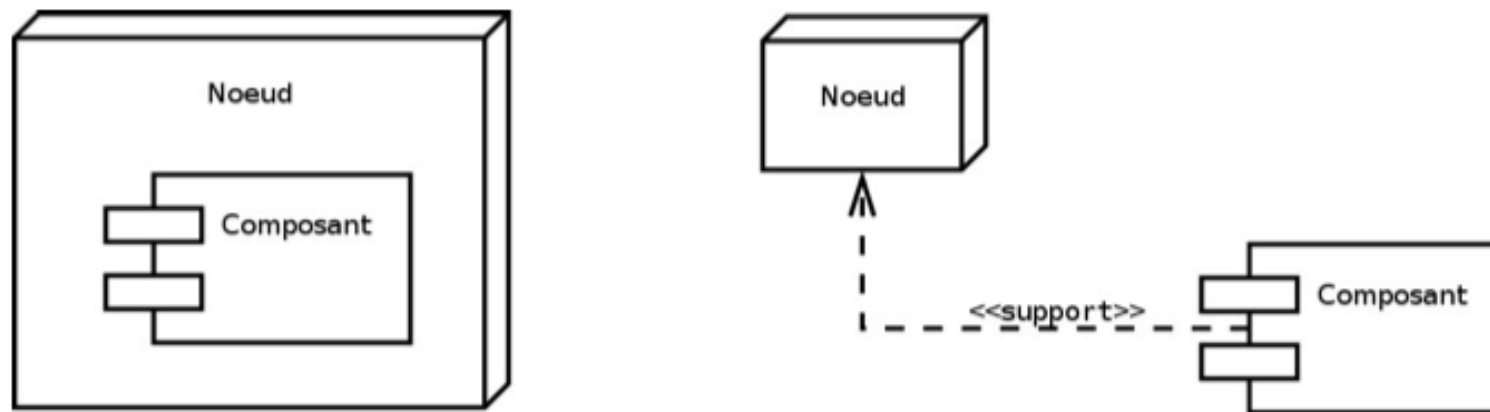
# Diagramme de composants



# Diagramme de composants

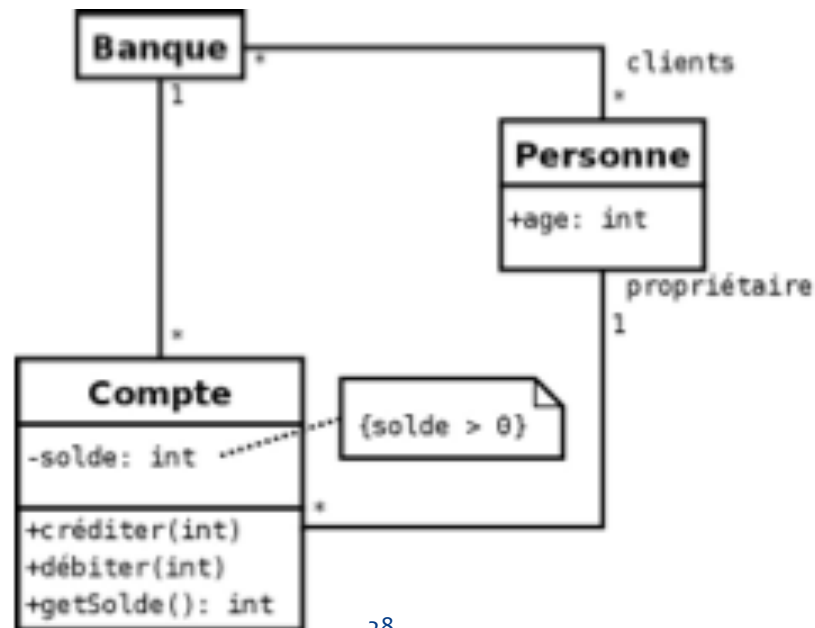


# Diagramme de déploiement



# OCL (Object Constraint Language)

- \* UML dispose d'un langage de définition de contraintes (OCL) afin de normaliser les expressions



# Modélisation UML

- \* Il apparaît nécessaire à l'utilisateur d'UML d'avoir un outil pour représenter les différents diagrammes, afin de respecter le formalisme.
- \* VS 2012 en fait une partie ...