

HMIN112M : Modélisation conceptuelle UML

I.Mougenot

UM

2020

Notion de modèle

Modèle : abstraction d'une réalité pour répondre à un besoin

- 1 différents modèles
- 2 modèle statique ou contemplatif : cherche à modéliser les entités du domaine d'intérêt
- 3 modèle dynamique ou numérique : cherche à modéliser les processus qui s'appliquent aux entités du domaine d'intérêt

Un modèle est une simplification : les modèles sont utiles, à défaut d'être exacts

Modèle statique

La description de la structure des entités et de leurs relations est centrale dans le modèle

- modèle dit aussi structurel ou conceptuel
- plusieurs approches sous-tendent la modélisation : modèle Entité-Association (1976) et méthode Merise, notation objet UML (années 1990, issue du génie logiciel)
- les approches privilégient une représentation graphique / visuelle des mondes considérés

Différents modèles

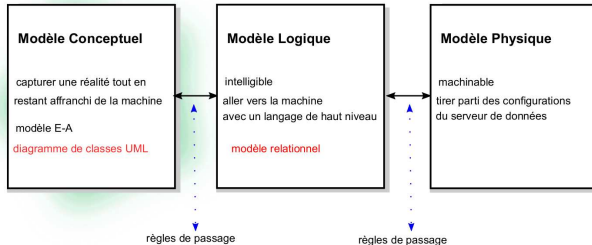


Figure: Une approche s'appuyant sur trois types de modélisation

Notation UML

Différents diagrammes sont proposés pour traiter des aspects fonctionnels, structurels et statiques d'un système, parmi lesquels (non exhaustif) :

- ① aspect fonctionnel : cas d'utilisation
- ② aspect structurel : diagrammes de classes, diagrammes d'objets
- ③ aspect dynamique (ou comportemental) : diagramme de séquences, diagramme d'états-transitions, diagramme d'activités
- ④ aspect implantation : diagramme de composants

Outils d'aide à la conception

Un large éventail de logiciels gratuits ou payants pour faciliter les activités de modélisation

- ① ArgoUML
- ② Dia
- ③ Umbrello
- ④ PowerAMC
- ⑤ Rational Rose
- ⑥ ...

Définitions classe et objet

Ces définitions sont générales au paradigme objet (langages de programmation en particulier)

- objet = un identifiant (donné par le système) + un état (ensemble d'attributs valués) + un comportement (ensemble d'opérations qui s'appliquent à l'objet)
- classe = moule à objets : la classe regroupe les objets qui partagent les mêmes types d'états et les mêmes types de comportement
- Par exemple, la classe Etudiant regroupe tous les objets qui ont des propriétés comme un numéro INE.
- le terme objet possède des termes équivalents comme instance ou individu

Université : monde modélisé

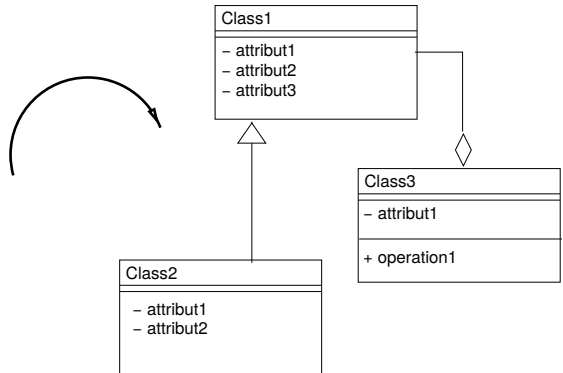
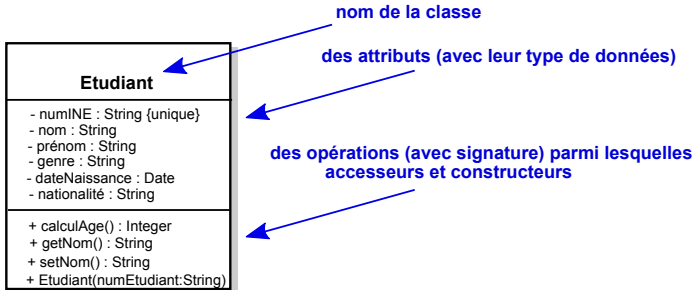


Figure: image extraite d'un article du journal Libération

Notion de classe dans UML



notion de portée (principe d'encapsulation) :

privée (-) : accessible uniquement au niveau d'un objet de la classe

publique (-) : accessible par tout objet

**protégée (#) : accessible uniquement au niveau d'un objet de la classe
ou d'une classe dérivée**

Figure: Représentation de la classe Etudiant

Notion d'objet dans UML

objet ou
individu ou
instance
de classe

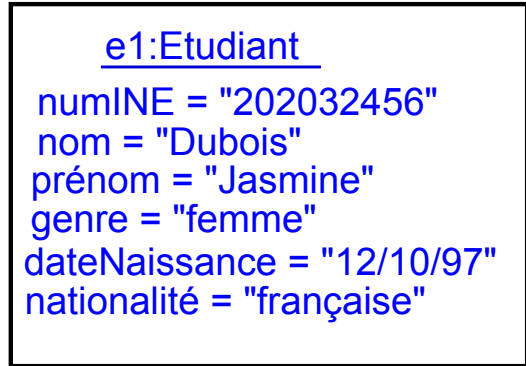


Figure: Représentation d'un objet de la classe Etudiant

Notion d'association

Les associations posées entre les classes permettent d'exprimer les inter-relations des objets de ces classes

- les associations possèdent un nom (souvent un verbe qui désigne une action) et un sens de lecture
- les extrémités des associations sont enrichies de multiplicités qui indiquent le nombre de rôles min et max joués par un objet dans une association
- la signification de l'association et des objets qui y sont impliqués, peut être étendue par des rôles

Notion d'association dans UML

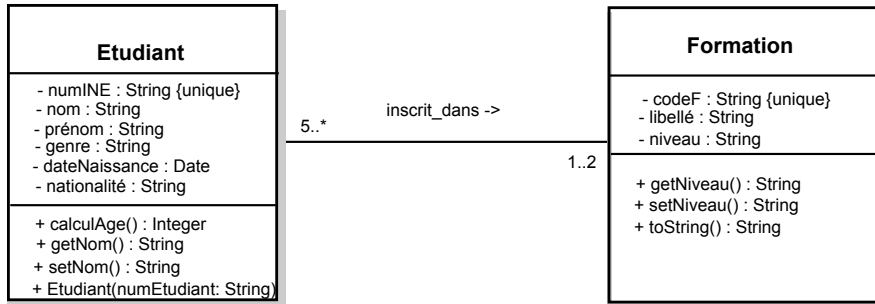


Figure: Représentation de l'association inscrit_dans (à l'instant présent)

Des précisions

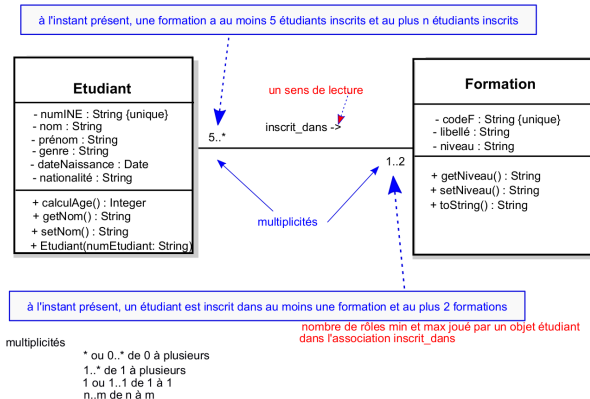


Figure: Autour de l'association inscrit_dans

Diagramme d'objets conforme au diagramme de classes

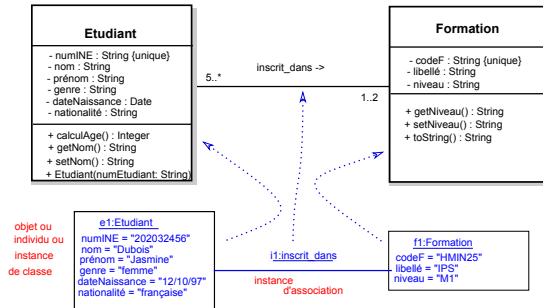


Figure: Exemple diagramme de classes / diagramme d'objets

Notion de classe association

Une association munie d'une ou de plusieurs propriétés :

- prend un statut particulier :
 - elle est élevée au rang de classe, elle est dite "chosifiée ou réifiée"
 - elle reste toutefois également une association

Classe association en UML

Traduire la temporalité : les multiplicités sont également modifiées dans le diagramme

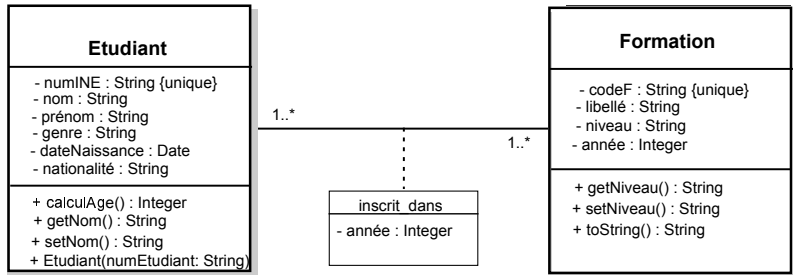


Figure: Une classe association pour exprimer les inscriptions dans le temps

Diagramme de classes enrichi

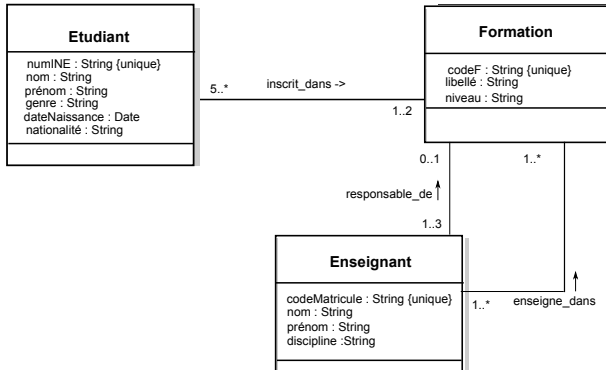


Figure: Plusieurs associations peuvent être définies entre deux classes

Diagramme de classes enrichi

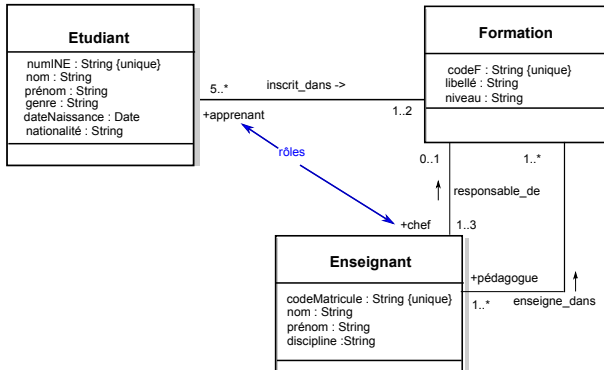


Figure: Des rôles aux extrémités des relations viennent préciser la sémantique de la fonction jouée par l'objet

Même diagramme dans le temps

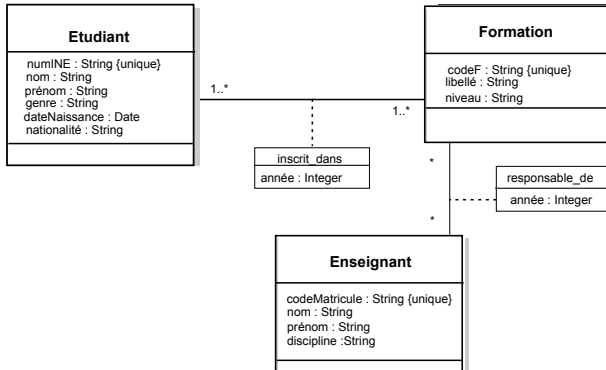


Figure: Plusieurs classes associations avec des attributs temporels

Notion d'arité d'une association

Une association peut être munie de 2 ou plus extrémités

- arité d'une association = nombre d'extrémité de l'association
- le plus souvent, l'association a deux extrémités, elle est dite binaire ou d'arité 2
- il peut arriver, lorsque plus de deux objets sont engagés dans la même association en simultané, de devoir définir des associations d'arité > 2 , dites alors n-aires
- le nom, le sens de lecture et les multiplicités de l'association ne sont alors pas mentionnés

Notion de relation n-aire

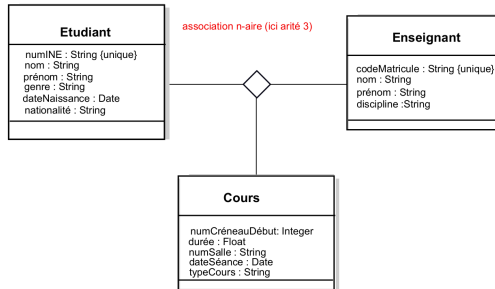


Figure: L'association possède trois extrémités

Réduire la complexité du modèle

tenter de revenir sur des associations binaires

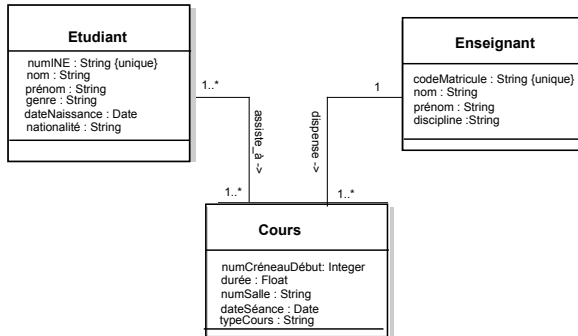


Figure: Traduire la relation n-aire en relations binaires

Réduire la complexité du modèle

ou bien exploiter une classe association

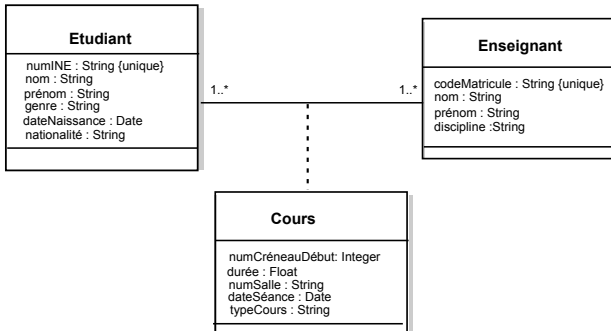


Figure: Traduire la relation n-aire avec une classe association

Aspects structurels : association réflexive

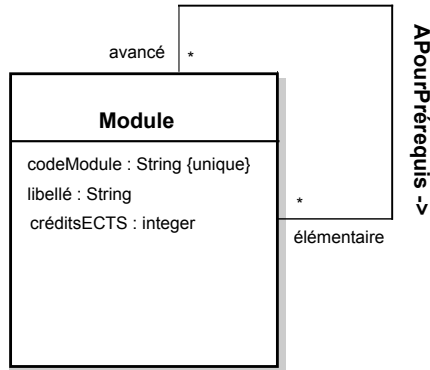


Figure: Une association dont les deux extrémités pointent sur la même classe

Association réflexive

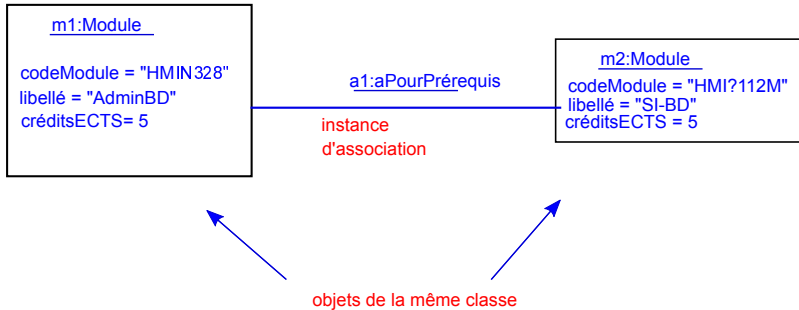


Figure: Exemple de diagramme objet pour deux objets impliqués dans une association réflexive

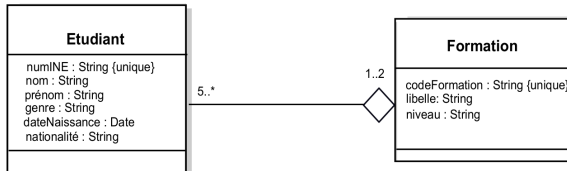
Associations spécifiques

Traduire la paronymie et l'héritage au travers d'associations dédiées

- agrégation : relation tout/partie
- composition : même sémantique que l'agrégation mais avec une dépendance existentielle supplémentaire
- héritage : généralisation / spécialisation

Agrégation (à l'instant t)

Relation d'agrégation : traduire la notion de tout/partie

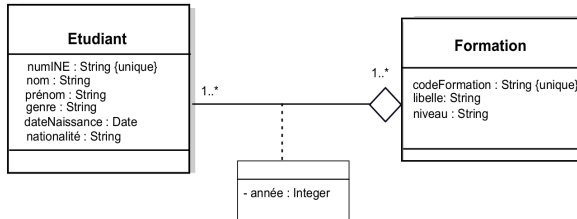


Un étudiant fait partie d'une à deux formations. Une formation est un tout et a pour sous-parties au moins 5 étudiants voire plus.

Figure: Formation envisagée comme une collection d'étudiants

Agrégation et classe association (dans le temps)

Relation d'agrégation : éventualité de classe association

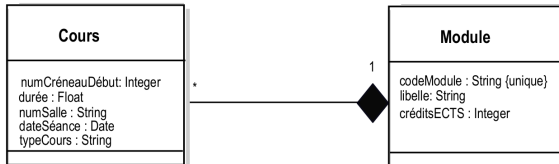


Dans le temps, un étudiant fait partie de 1 à plusieurs tous (formations)

Figure: Formation envisagée comme une collection d'étudiants

Composition

Relation de composition : tout/partie avec dépendance forte

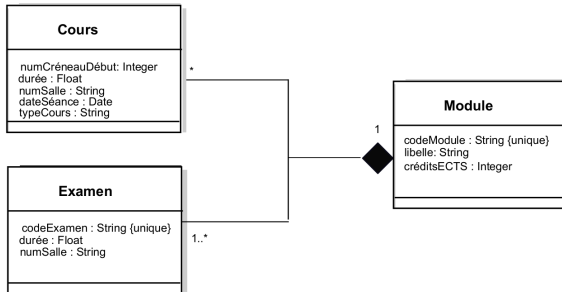


Un cours est une partie inhérente du module. Sans le module, le cours n'a pas de raison d'exister

Figure: Un cours : obligatoirement une sous-partie d'un module

Composition

Relation de composition : tout/partie avec dépendance forte

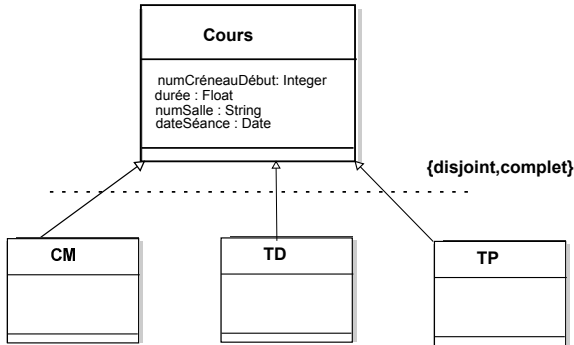


Un cours est une partie inhérente du module. Sans le module, le cours n'a pas de raison d'exister
Il en va de même pour l'examen

Figure: Plusieurs relations de composition : entre cours et module et

Héritage

Relation d'héritage : généralisation / spécialisation

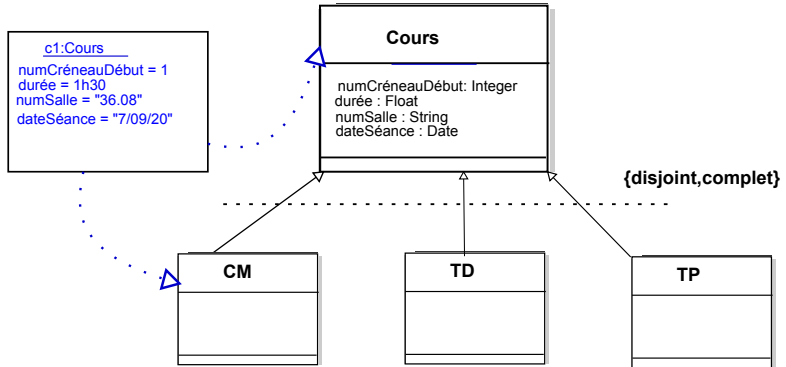


Un cours magistral est une spécialisation de cours. C'est un cours mais il peut avoir ses propres spécificités
Il en va de même pour un TD ou un TP

Figure: Un CM est aussi un cours à un niveau plus général

Aspects structurels : héritage

Relation d'héritage : généralisation / spécialisation

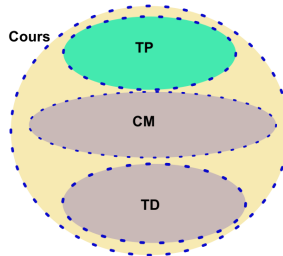


c1 est une instance à la fois de Cours et de CM

Figure: Une objet de la classe CM est aussi un objet de cours

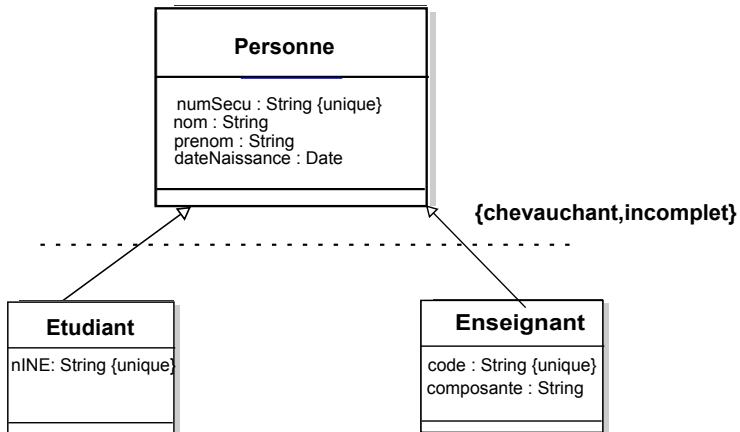
Partition

Notion de partition : quand les classes filles sont disjointes et sont toutes connues



les intersections entre classes filles sont vides : $TD \cap TP = \emptyset$ et $TD \cap CM = \emptyset$ et $CM \cap TP = \emptyset$
les unions entre les classes filles correspondent à la classe mère : $TP \cup TD \cup CM = \text{Cours}$

Autre exemple d'héritage (sans partition)



Une personne peut être tout à la fois un étudiant et un enseignant (chevauchant)
Une personne peut n'être ni un étudiant, ni un enseignant (incomplet)

Exemple d'objet peuplant plusieurs classes

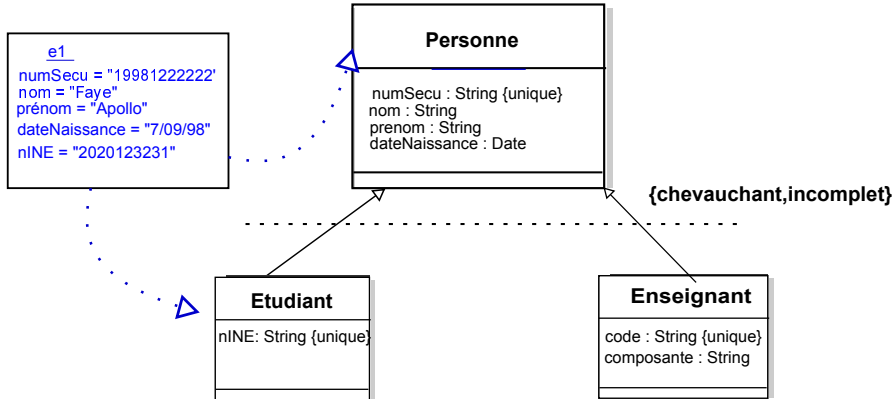


Figure: Sur la base du modèle précédent

Les contraintes s'appliquant à un diagramme de classes

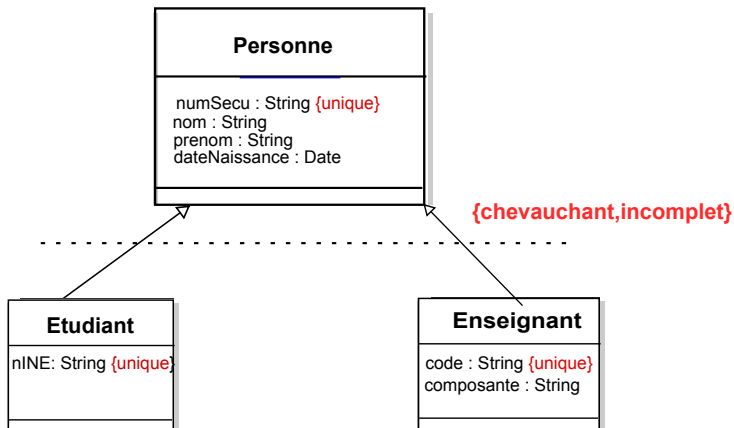


Figure: Pseudo-langage OCL (Object Constraint Language)

Stéréotypes

Stéréotype : étendre le pouvoir d'expression d'UML

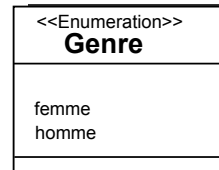
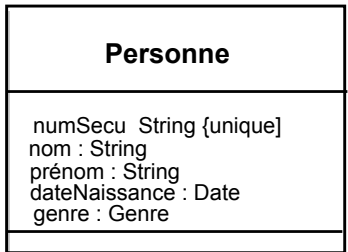


Figure: Arborescence de classes chevauchantes

Attributs et associations dérivées

attributs et associations dérivées

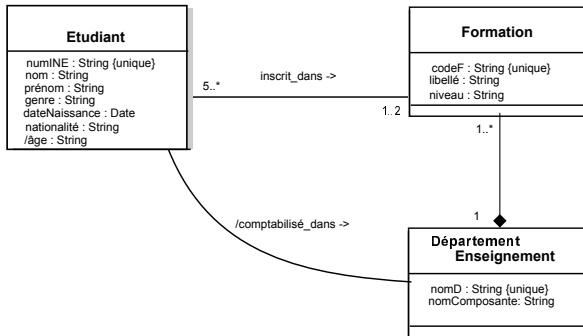


Figure: Calculées à partir d'autres attributs ou associations