UNIVERSITÉ PARIS SUD

L3 informatique et MIAGE

Année 2016-2017

Génie logiciel avancé

Rappels d'UML et expression des contraintes

Delphine Longuet

delphine.longuet@lri.fr

http://www.lri.fr/~longuet/Enseignements/16-17/L3-GLA

Modélisation

Modèle : Simplification de la réalité, abstraction, vue subjective

- modèle météorologique, économique, démographique...

Modéliser un concept ou un objet pour :

- Mieux le comprendre (modélisation en physique)
- Mieux le construire (modélisation en ingénierie)

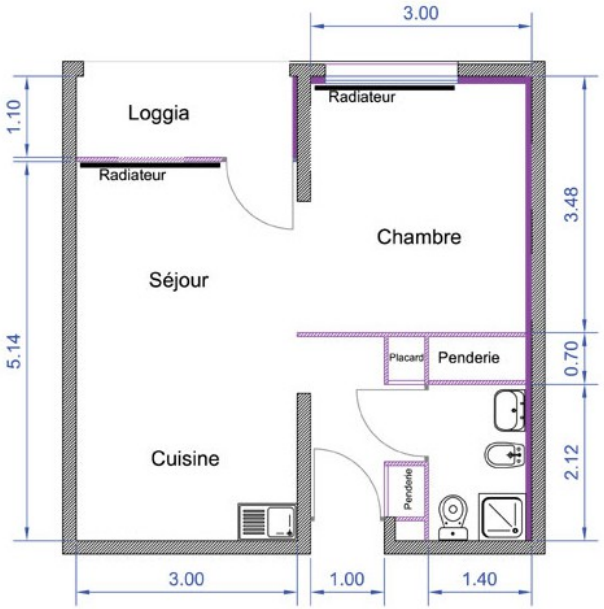
En génie logiciel :

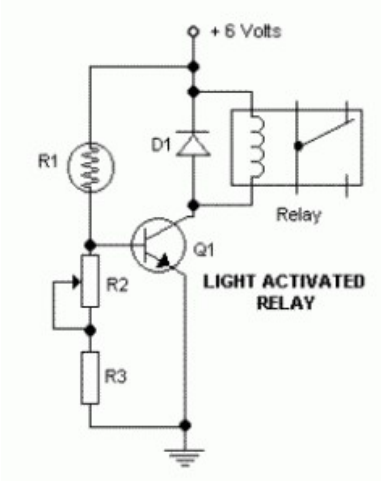
- Modélisation = spécification + conception
- Aider la réalisation d'un logiciel à partir des besoins du client

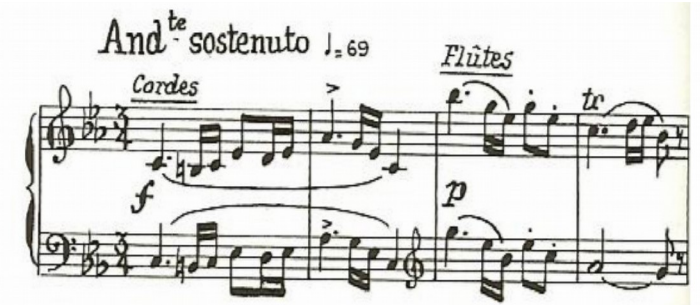
Modélisation graphique

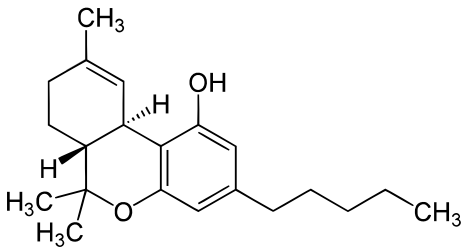
Principe : « *Un beau dessin vaut mieux qu'un long discours* »

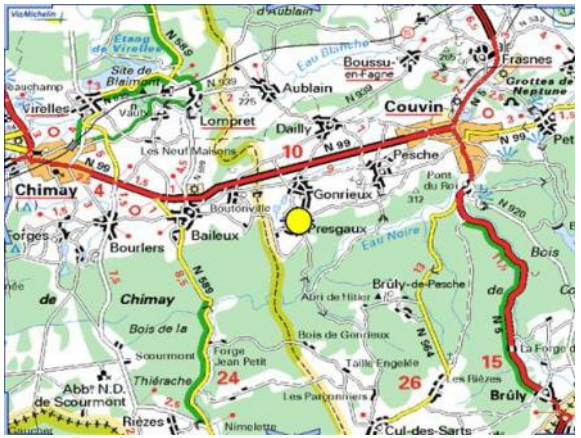
Seulement s'il est compris par tous de la même manière











UML : Unified Modeling Language

Langage :

- Syntaxe et règles d'écriture
- Notations graphiques normalisées

... de modélisation :

- Abstraction du fonctionnement et de la structure du système
- Spécification et conception

... unifié :

- Fusion de plusieurs notations antérieures : Booch, OMT, OOSE
- Standard défini par l'OMG (Object Management Group)
- Dernière version : UML 2.5 (juin 2015)

En résumé : Langage graphique pour visualiser, spécifier, construire et documenter un logiciel

Pourquoi UML ?

Besoin de modéliser pour construire un logiciel

- Modélisation des aspects statiques et dynamiques
- Modélisation à différents niveaux d'abstraction et selon plusieurs vues
- Indépendant du processus de développement

Besoin de langages normalisés pour la modélisation

- Langage semi-formel
- Standard très utilisé

Conception orientée objet

- Façon efficace de penser le logiciel
- Indépendance du langage de programmation (langages non objet)

Diagrammes UML

14 diagrammes hiérarchiquement dépendants

Modélisation à tous les niveaux le long du processus de développement

Diagrammes structurels :

- Diagramme de classes
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement
- Diagramme de paquetages
- Diagramme de structure composite
- Diagramme de profils

Diagrammes comportementaux :

- Diagramme de cas d'utilisation
- Diagramme états-transitions
- Diagramme d'activité

Diagrammes d'interaction :

- Diagramme de séquence
- Diagramme de communication
- Diagramme global d'interaction
- Diagramme de temps

Exemple fil rouge : station service

Une station service TATOL a des postes de distribution de deux types :

- automatiques (par carte bancaire) ouverts 24h/24
- manuels (utilisables seulement si la station est ouverte)

Chaque poste est identifié par un numéro.

Chaque poste peut délivrer 3 sortes de carburant. À chaque instant, il en délivre au plus une sorte.

Chaque poste est muni de 3 compteurs :

- la quantité de carburant servie
- le prix au litre (qui ne change pas pendant la journée)
- le prix total à payer

Les compteurs affichent 0 s'il n'y a pas de distribution en cours.

Exemple fil rouge : station service

La station dispose de 3 citernes (une par type de carburant) avec :

- le niveau courant de carburant
- un niveau d'alerte pour prévenir le gérant qu'elle va être vide
- un niveau minimal qui, une fois atteint, ne permet plus de délivrer du carburant

Les niveaux d'alerte et minimaux sont les mêmes pour toutes les citernes.

Le système doit garder une trace de tous les achats effectués depuis la dernière ouverture de la station.

Un opérateur ouvre la station quand il arrive et la ferme quand il part.

Quand une livraison de carburant a lieu, l'opérateur met à jour le niveau de la citerne. Une livraison fait toujours repasser les niveaux au-dessus du niveau d'alarme.

Exemple fil rouge : station service

Pour utiliser un poste automatique :

1. Un client insère sa carte bancaire et tape son code
2. Le système authentifie la carte auprès du service bancaire
3. En cas de succès, le client choisit un type de carburant et se sert d'un certain nombre de litres
4. Le système enregistre l'achat, envoie un ordre de débit au service bancaire et remet à zéro les compteurs du poste

Pour utiliser un poste manuel :

1. Le client choisit son carburant et se sert
2. Les caractéristiques de l'achat sont envoyées à l'opérateur
3. Lorsque le client a payé en indiquant son numéro de pompe, le pompiste enregistre l'achat et remet à zéro les compteurs du poste

Les postes restent bloqués tant que les compteurs n'ont pas été remis à 0.

Description des cas d'utilisation

Objectif : Comprendre les besoins du client pour rédiger le cahier des charges

Principe :

- Définir les limites du système
- Définir l'environnement du système : les utilisateurs ou éléments qui interagissent avec le système
- Définir les utilisations principales du système : à quoi sert-il ?

Éléments constitutifs :

- Diagrammes des cas d'utilisation
- Description textuelle des cas d'utilisation
- Diagrammes de séquence des scénarios d'utilisation

Cas d'utilisation

Fonctionnalités principales du système du point de vue extérieur

Acteur : Entité qui interagit avec le système

- Personne, chose, logiciel, extérieur au système décrit
- Représente un rôle (plusieurs rôles possibles pour une même entité)
- Identifié par le nom du rôle

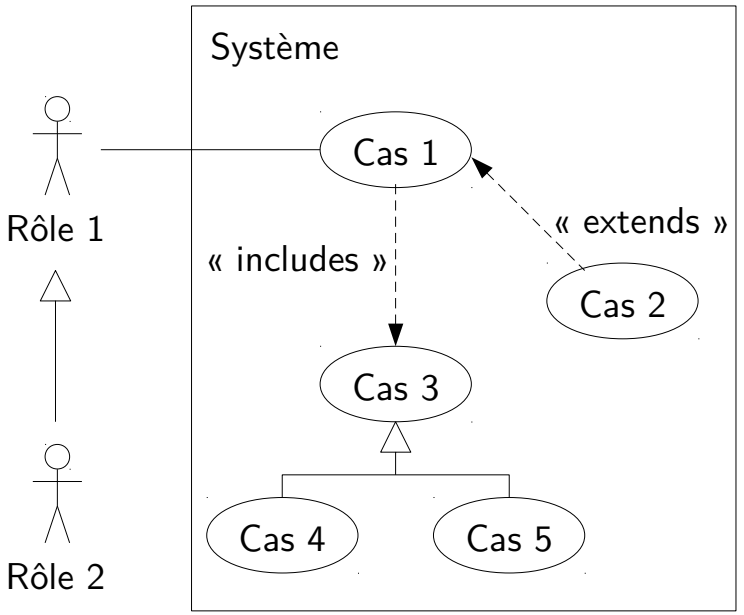
Cas d'utilisation : Fonctionnalité visible de l'extérieur

- Action déclenchée par un acteur
- Identifié par une action (verbe à l'infinitif)

Vision du système centrée sur l'utilisateur

Spécification des cas d'utilisation

Diagrammes des cas d'utilisation + Description textuelle



Cas 1
Acteur : Acteur A
Contexte :
Entrées :
Sorties :
Scénario principal :
1.
2.
3.
Variantes :
1a.
1b.
3a.

+ Scénarios d'utilisation

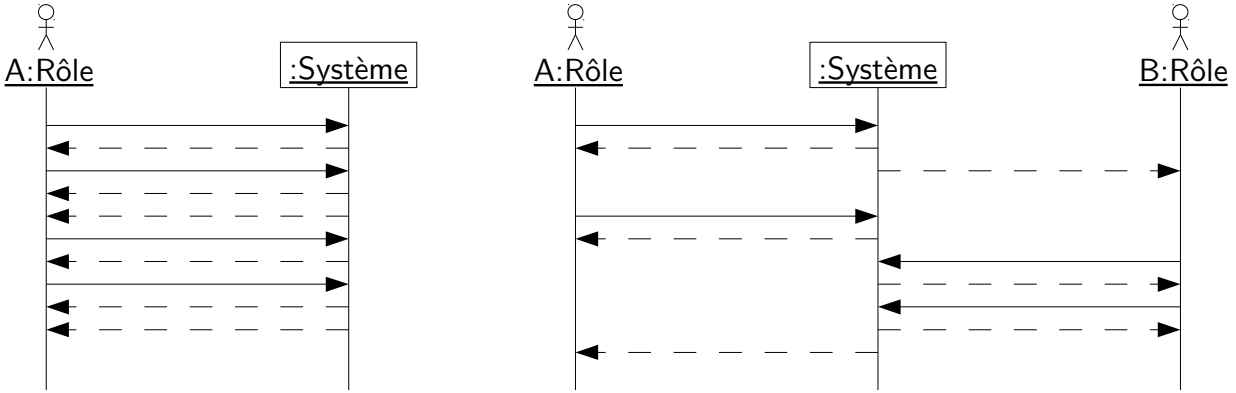


Diagramme de séquence (analyse)

Représentation graphique de la chronologie des échanges de messages entre les acteurs et le système

- Pas de messages entre acteurs (en dehors du système)
- Pas de messages internes au système (conception)
- Pas de boucle ou de branchement

En phase d'analyse

- Messages informels (pas des appels de méthodes)
- Noms des messages liés aux cas d'utilisation
- Mise en avant des données utiles au scénario (arguments)

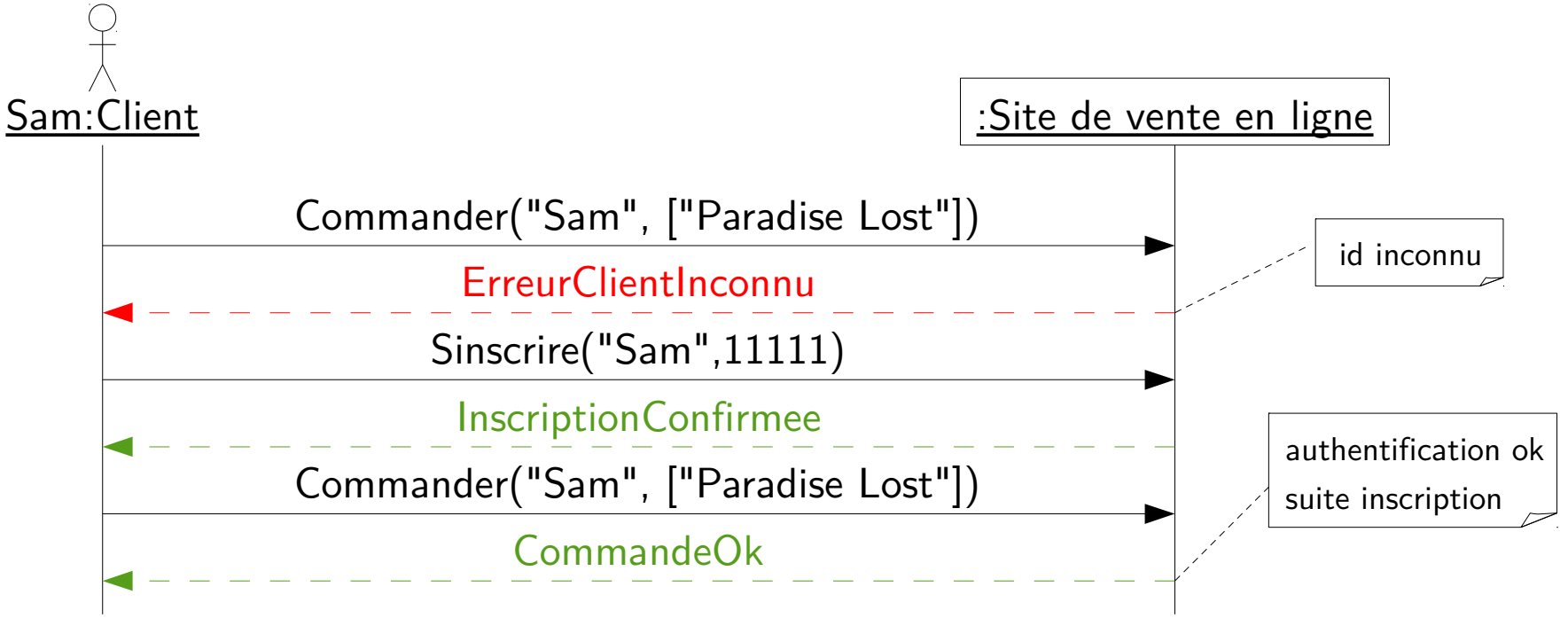
Objectifs

- Décrire un cas d'utilisation particulier
- Illustrer les interactions entre cas d'utilisation (scénario d'utilisation)

Scénario d'utilisation concret

Principe : Variables remplacées par des valeurs concrètes pour

- illustrer les différents scénarios d'un cas d'utilisation
- mettre en évidence les relations entre les différents cas
- construire des scénarios d'utilisation complexes pour le test



Mise en évidence de la nécessité d'être inscrit pour pouvoir commander

Objets et classes

Conception orientée objet : Représentation du système comme un ensemble d'objets interagissant

Diagramme de classes

- Représentation de la structure interne du logiciel
- Utilisé surtout en conception mais peut être utilisé en analyse

Diagramme d'objets

- Représentation de l'état du logiciel (objets + relations)
- Diagramme évoluant avec l'exécution du logiciel
  - création et suppression d'objets
  - modification de l'état des objets (valeurs des atributs)
  - modification des relations entre objets

Diagramme de classes

Représentation de la structure statique interne du système :

- les classes d'intérêt
- les relations entre classes
- les attributs (et les opérations)

En phase d'analyse :

- type de données simple (bool, int, real, string) ou primitif (Date)
- pas d'opérations en dehors de celles des cas d'utilisation, qui ne sont pas rattachées à une classe à ce niveau d'abstraction
- pas de visibilité autre que public

Limites du diagramme de classes

Diagramme de classes : structure du système en termes d'objets et de relations entre ces objets

Ne permet pas de représenter :

- Valeurs autorisées des attributs
- Conditions sur les associations
- Relations entre les attributs ou entre les associations

Expression des contraintes liées au diagramme :

- Notes dans le diagramme
- Texte accompagnant le diagramme
- OCL : langage de contraintes formel associé à UML
- Dans ce cours : logique mathématique orientée objet

Contraintes, invariants

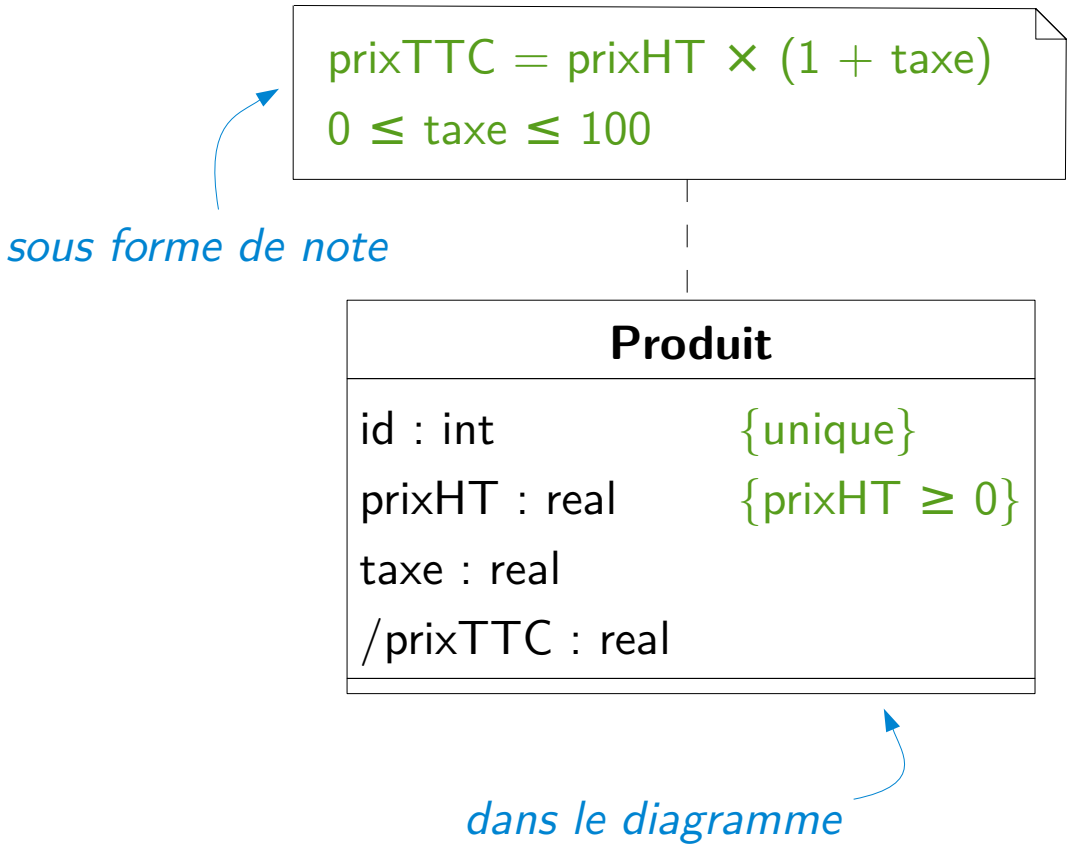
Propriétés :

- Portant sur les éléments du modèle
- Doivent être vérifiées à tout instant
- En général, restriction sur les diagrammes d'objets possibles à partir du diagramme de classes
- Héritage des contraintes de la super-classe vers les sous-classes

Contraintes présentes dans le diagramme :

- Type des attributs
- Multiplicités des associations

Contraintes sur les attributs

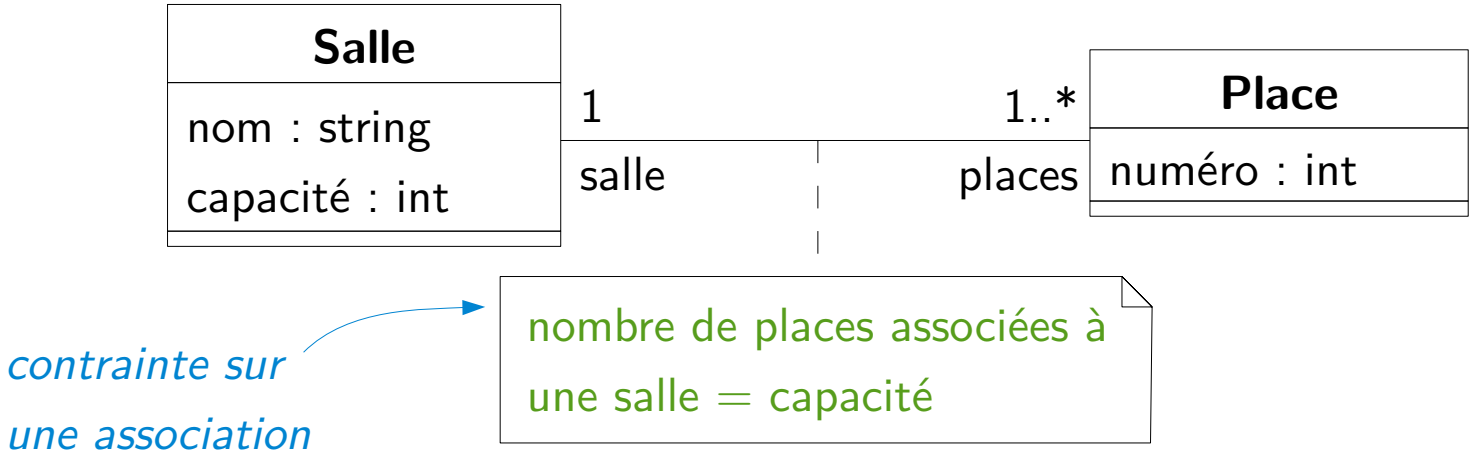


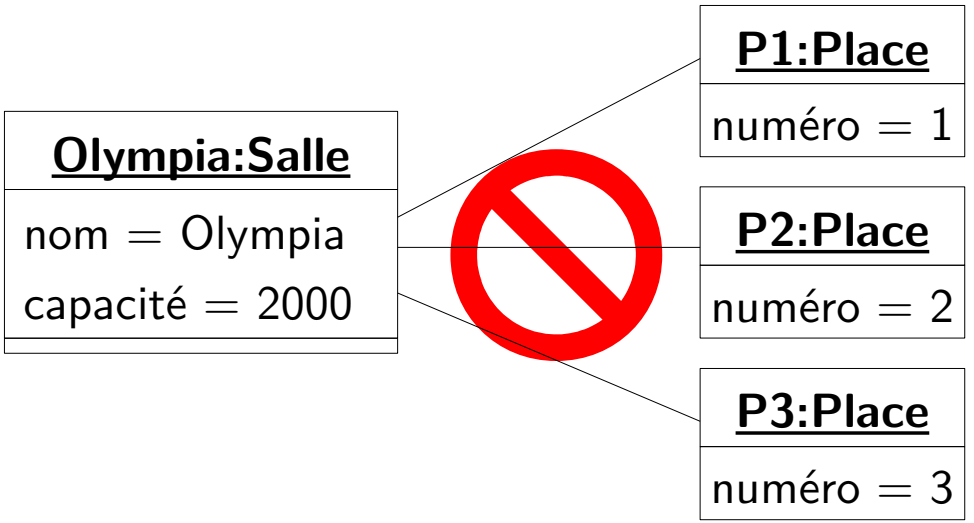
Contraintes sur la classe Produit :

- L'identifiant est unique
- Le prix hors taxe est toujours positif ou nul
- La taxe est comprise entre 0 et 100
- Le prix TTC est égal au prixHT augmenté de la valeur de la taxe appliquée au prixHT

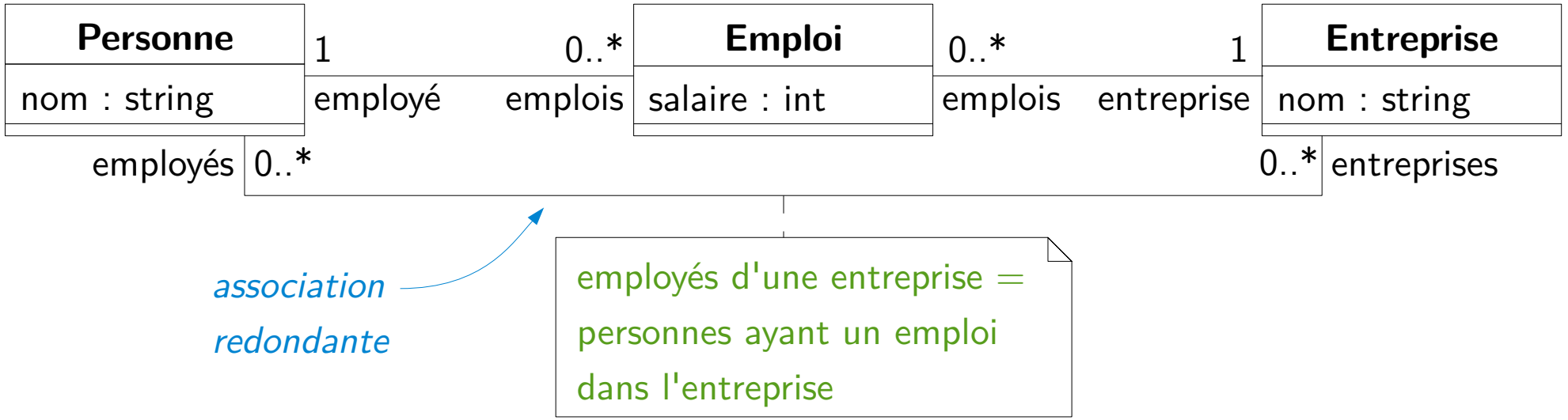
dans un document annexe

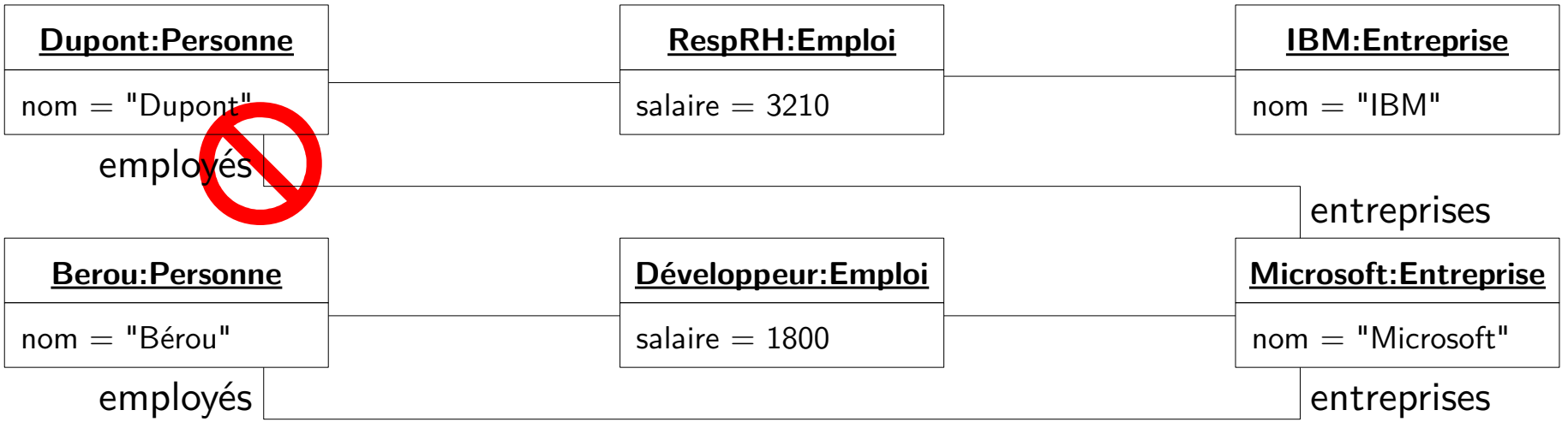
Contraintes sur les associations



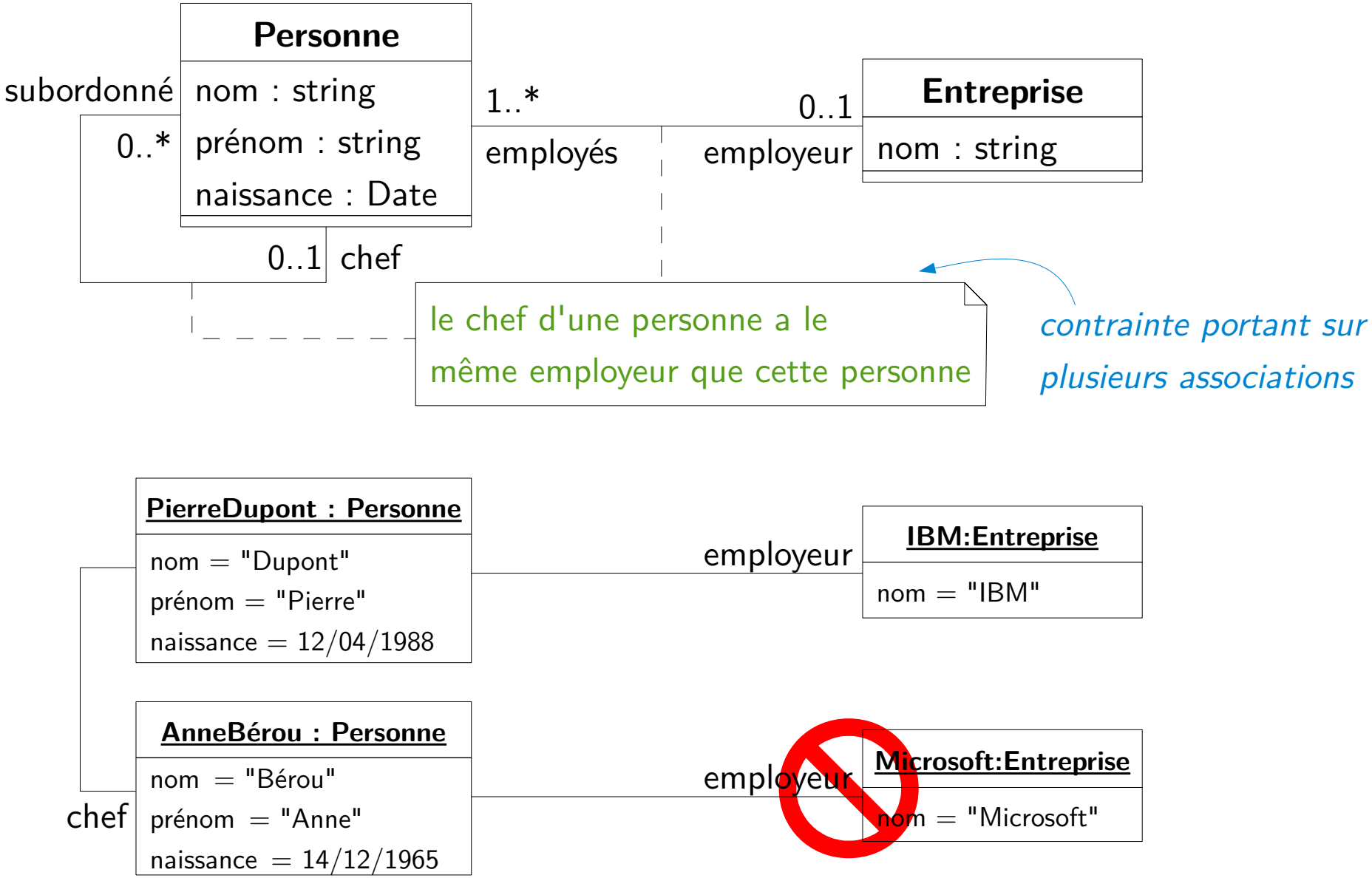


Contraintes sur les associations

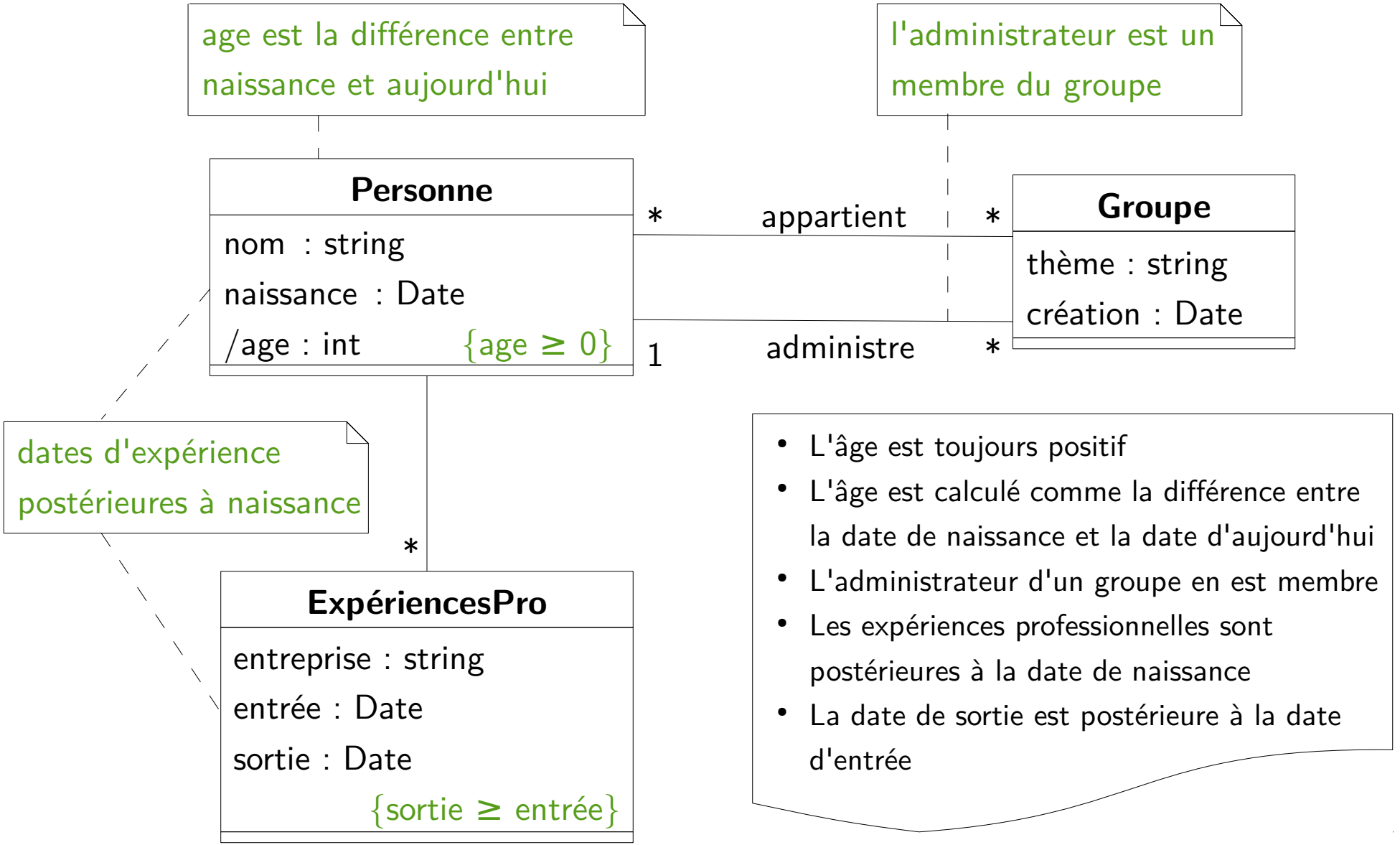




Contraintes sur les associations



Contraintes, invariants



Expression des contraintes

Langage de contraintes : logique mathématique orientée objet

Base : logique mathématique et théorie des ensembles

$\forall c \in A, \exists \ldots$

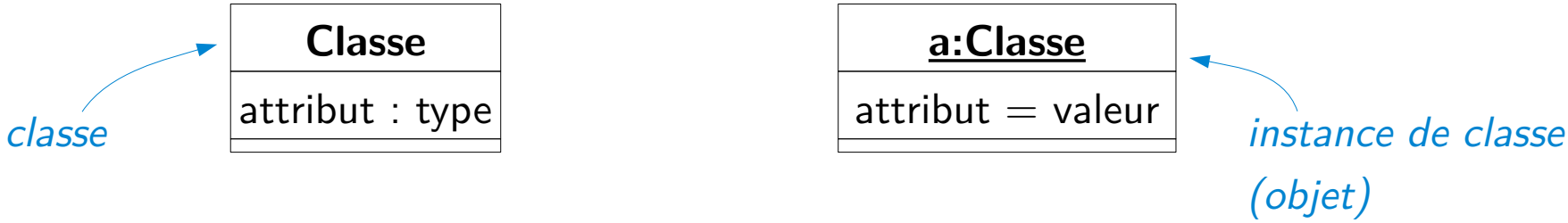
$\{ c \in C \mid \ldots \}$

$c > 0 \Rightarrow \ldots$

Ajout : termes dénotant

- des valeurs : utilisateur.nom
- des objets : personne.employeur
- des ensembles de valeurs : client.vehicules.couleur
- des ensembles d'objets : cinema.salles

Navigation dans le diagramme de classes

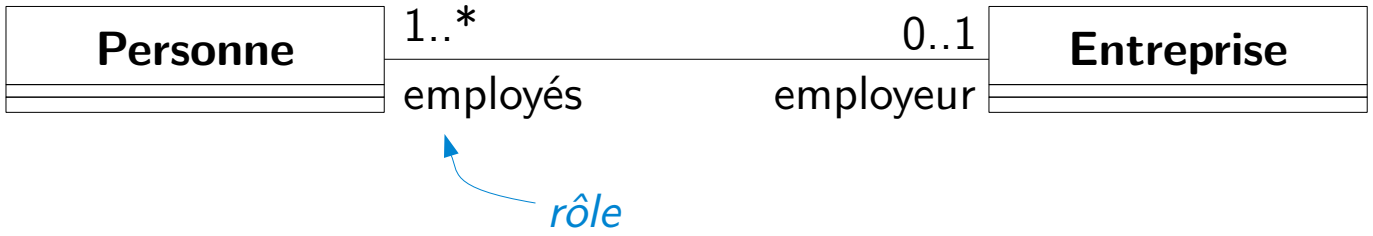


a.attribut : valeur de l'attribut de **a**

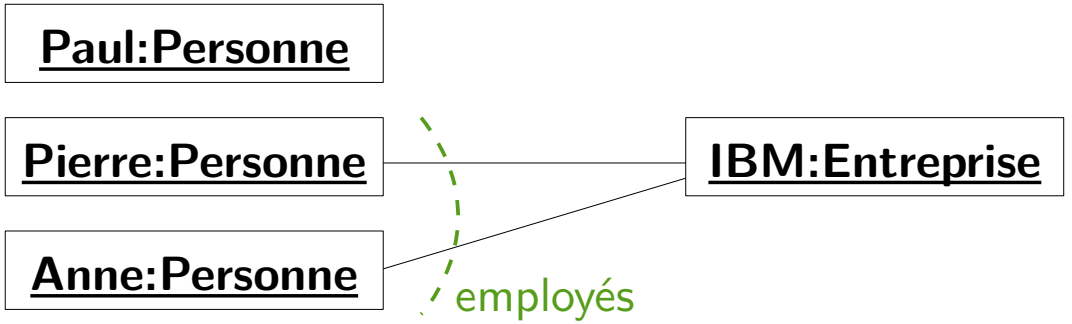
pile:Produit
id = 7846210
prixHT = 2,03
taxe = 20,0
prixTTC = 2,49

$$\text{pile.prixHT} \geq 0$$
$$0 \leq \text{pile.taxe} \leq 100$$
$$\text{pile.prixTTC} = \text{pile.prixHT} \times (1 + \text{pile.taxe})$$

Navigation dans le diagramme de classes

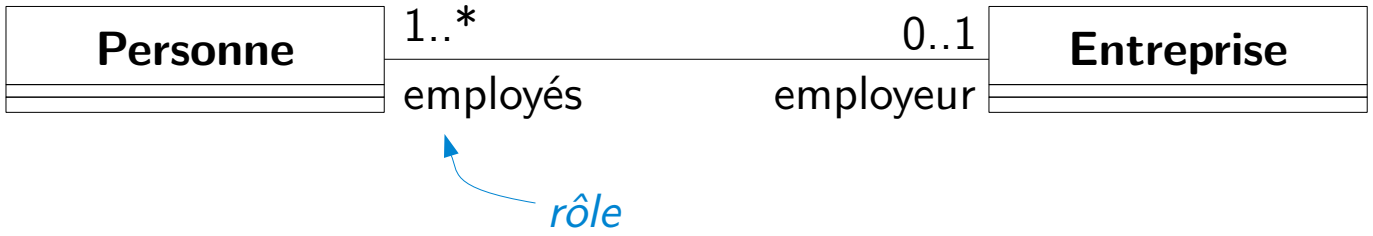


a.role : ensemble d'objets liés à **a** par l'association de terminaison **role**

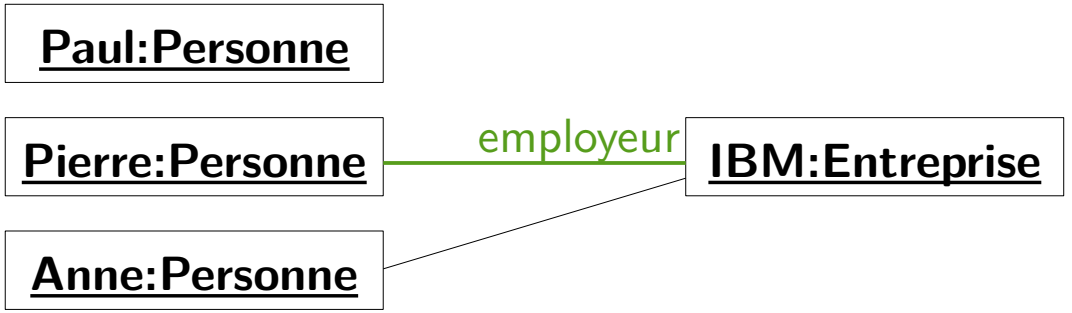


IBM.employés = {Pierre, Anne}

Navigation dans le diagramme de classes



a.role : ensemble d'objets liés à **a** par l'association de terminaison **role**

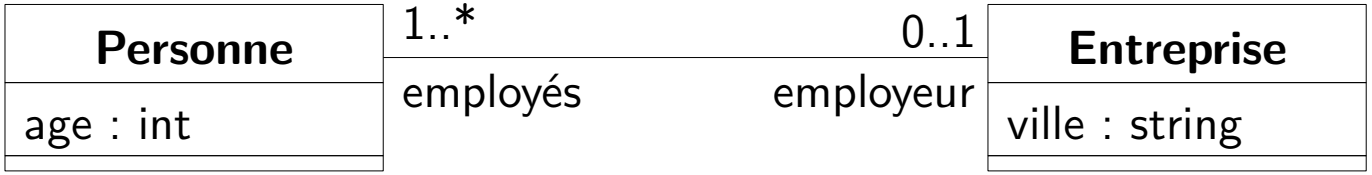


IBM.employés = {Pierre, Anne}
Pierre.employeur = IBM
Paul.employeur = ∅

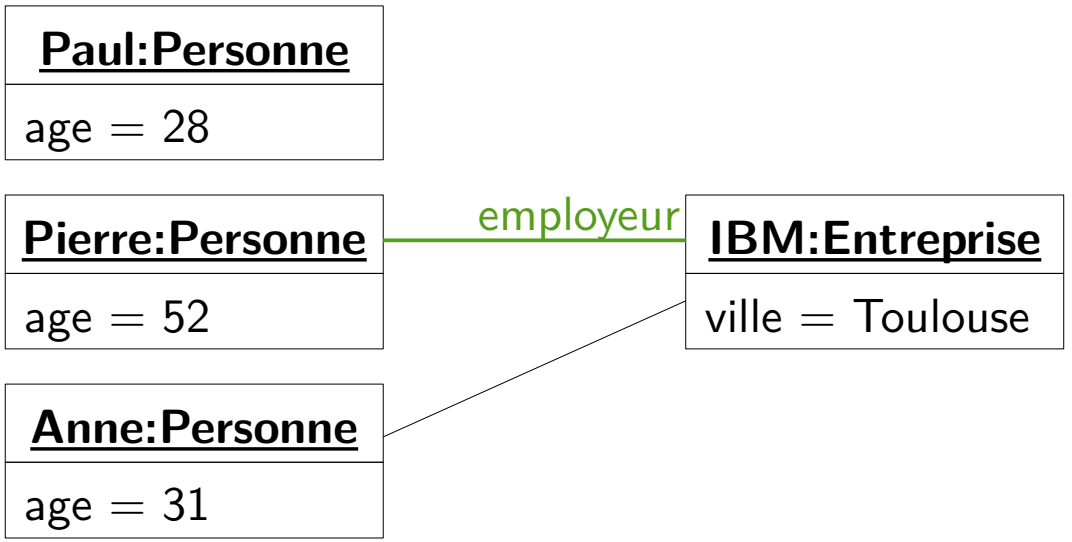
Multiplicité 0..1 ou 1 : **a.role** est un singleton, noté sans accolade par convention

Si a n'est lié à aucun objet par l'association, **a.role** = ∅

Navigation dans le diagramme de classes



a.role.attribut : ensemble des valeurs de l'attribut pour les objets liés à **a** par l'association de terminaison **role**



Pierre.employeur.ville = Toulouse

IBM.employés.age = {52,31}

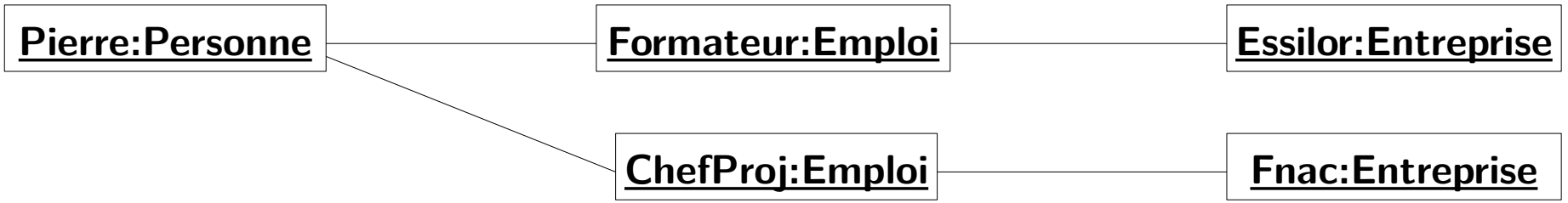
Paul.employeur.ville

Si **a** n'est lié à aucun objet par **role**, **a.role.attribut** n'est pas défini

Navigation dans le diagramme de classes

a.role1.role2 : ensemble des ensembles d'objets liés par **role2** à chaque objet lié à **a** par **role1**

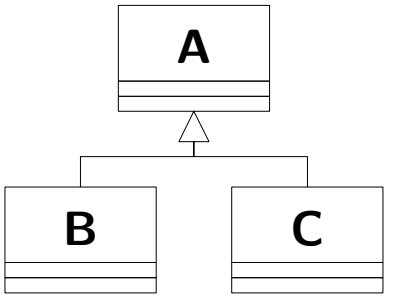
Par convention, ensemble « aplati » : { {a,b}, {c}, {d} } → {a,b,c,d}



Pierre.emplois.entreprise = {Essilor,Fnac}

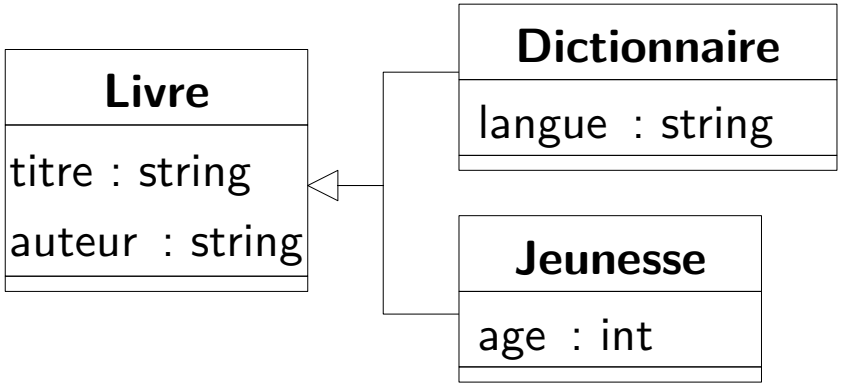
Ensembles d'objets

Classe = ensemble de ses instances

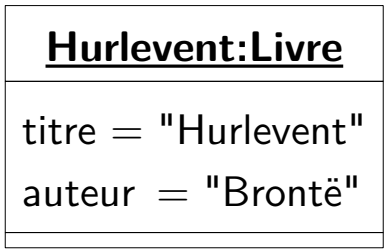


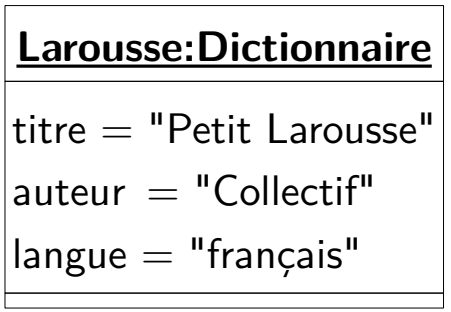
Hiérarchie : $B \cup C \subseteq A$, avec $B \cap C = \emptyset$

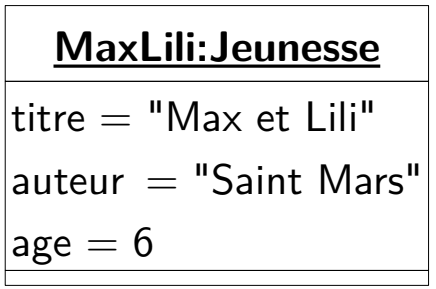
Certains objets de A peuvent n'être des objets ni de B ni de C



Hurlevent $\in$ Livre
MaxLili $\in$ Livre
Larousse $\in$ Livre
Larousse $\in$ Dictionnaire
Hurlevent $\notin$ Jeunesse







Expression des invariants

Invariant : Contrainte portant sur tous les objets d'une classe

Forme générale d'un invariant :

$$\forall\ a \in \text{Classe},\ P(a)$$

Produit
id : int
prixHT : real
taxe : real
prixTTC : real

Le prix hors taxe d'un produit est toujours positif

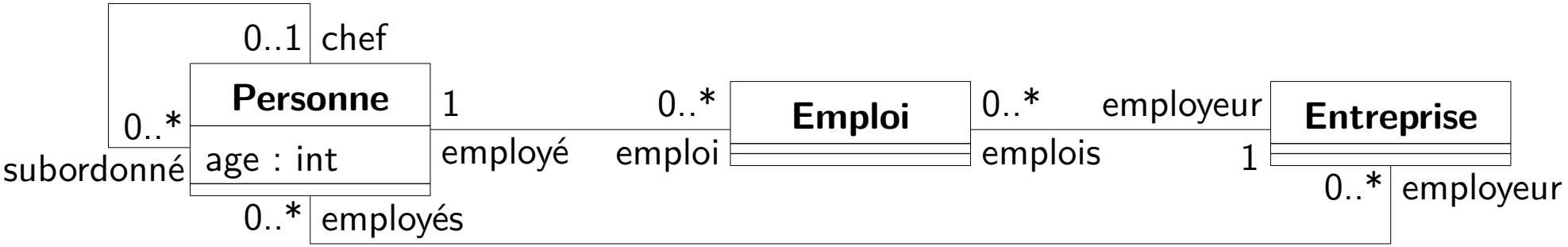
$$\forall\ p \in \text{Produit},\ p.\text{prixHT} \geq 0$$

Le prix TTC d'un produit est égal à son prixHT augmenté de la valeur de la taxe appliquée au prixHT

$$\forall\ p \in \text{Produit},\ p.\text{prixTTC} = p.\text{prixHT} \times (1 + p.\text{taxe})$$

En particulier, dans une hiérarchie, héritage des invariants par les sous-classes

Expression des invariants



Les employés de l'entreprise sont les personnes ayant un emploi dans l'entreprise

$\forall$ e $\in$ Entreprise, e.emplois.employé $=$ e.employés

Le chef d'une personne a le même employeur que cette personne

$\forall$ p $\in$ Personne, p.employeur $=$ p.chef.employeur

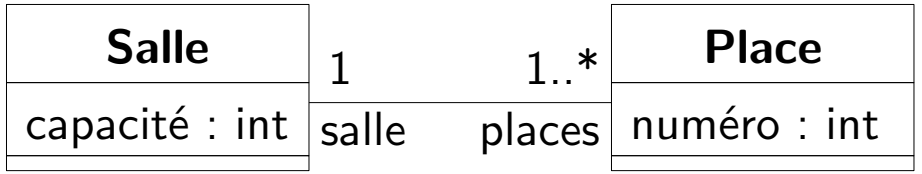
Tous les employés d'une entreprise ont plus de 16 ans

$\forall$ e $\in$ Entreprise, $\forall$ a $\in$ e.employés.age, a $\geq$ 16

Une entreprise n'a jamais plus de 10 employés de moins de 18 ans

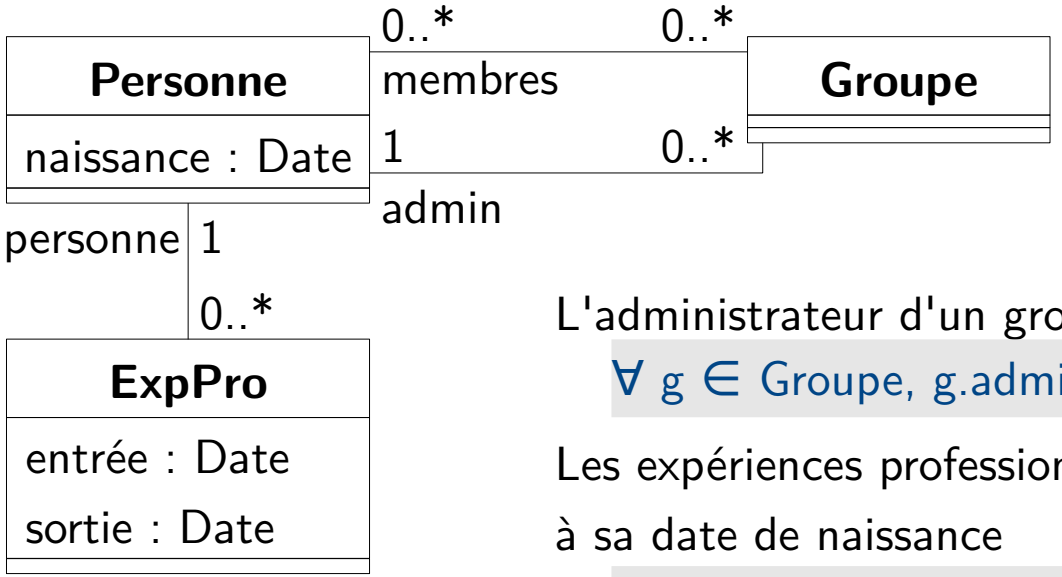
$\forall$ e $\in$ Entreprise, $|\{p \in e.employés \mid p.age < 18\}| \leq 10$

Expression des invariants



Le nombre de places de la salle égale sa capacité

$\forall\ s \in \text{Salle},\ |s.\text{places}| = s.\text{capacité}$

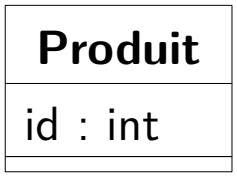


L'administrateur d'un groupe en est un membre

$\forall\ g \in \text{Groupe},\ g.\text{admin} \in g.\text{membres}$

Les expériences professionnelles d'une personne sont postérieures à sa date de naissance

$\forall\ e \in \text{ExpPro},\ e.\text{entrée} \geq e.\text{personne.naissance}$



L'identifiant d'un produit est unique

$\forall\ p1, p2 \in \text{Produit},\ p1 \neq p2 \Rightarrow p1.\text{id} \neq p2.\text{id}$