

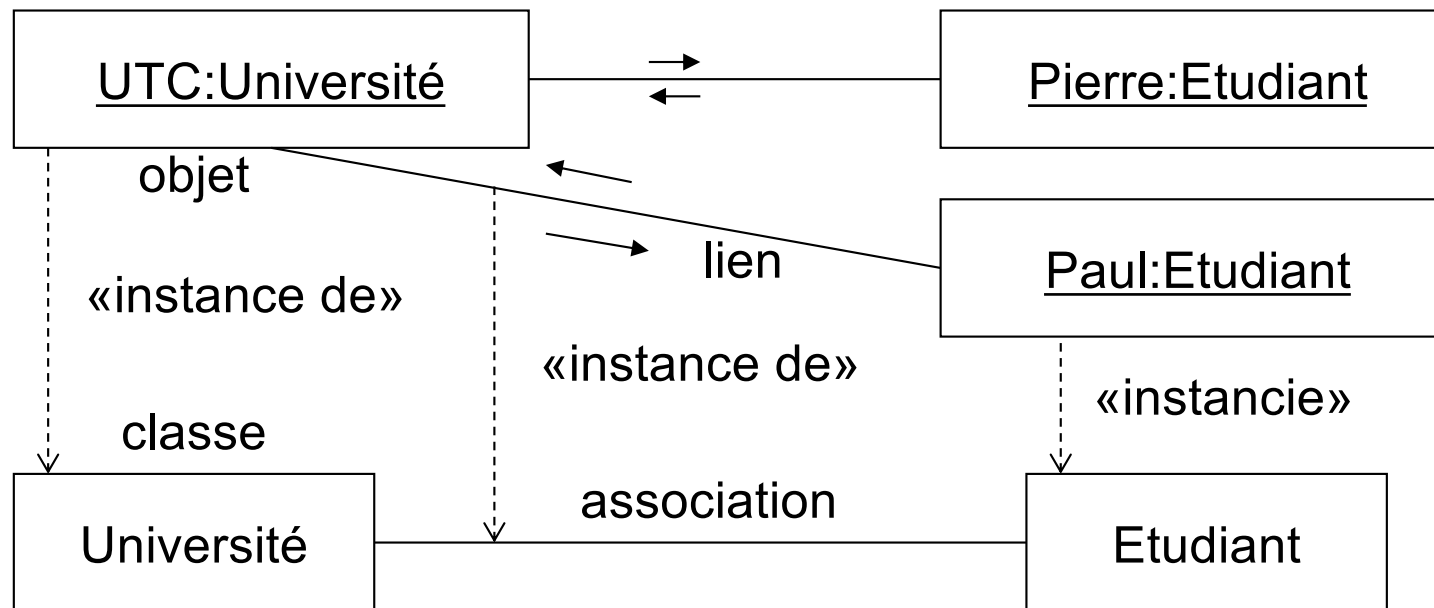
# SysML/UML

## Liens et associations

# Liens, associations

2

- Les messages entre objets circulent le long des liens

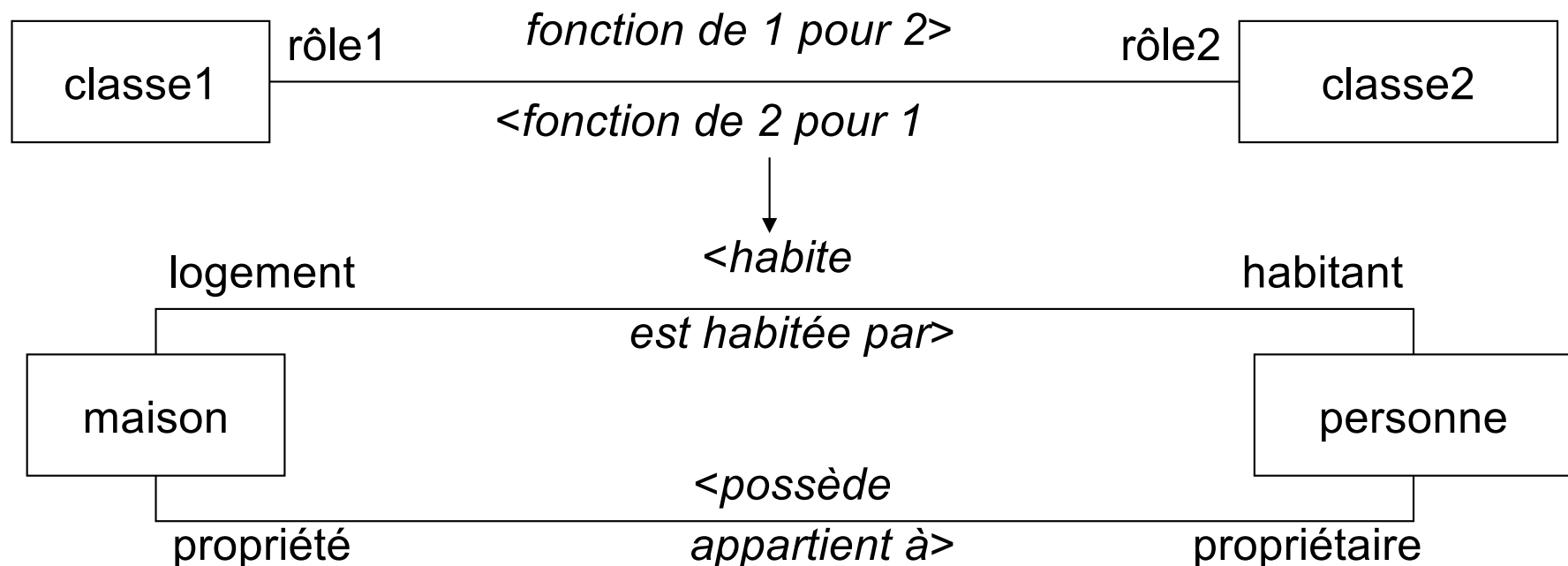


L'**association** exprime la connexion entre classes correspondant à une famille de liens

# Nommage des associations, rôles

3

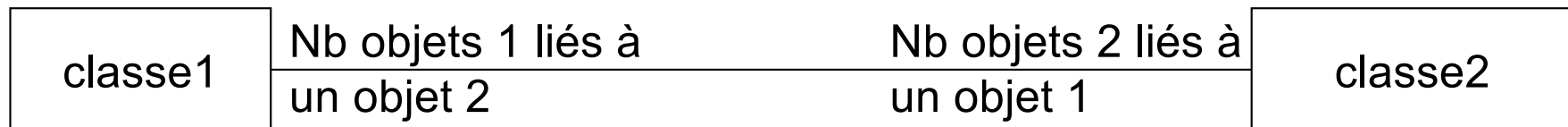
- Il est possible d'avoir plusieurs associations de nature différentes entre deux mêmes classes
- Les associations peuvent être nommées (fonctions : verbes) ainsi que les extrémités d'associations (rôles : substantifs)



# Multiplicité des associations

4

- Les extrémités d'association peuvent porter une indication de multiplicité (équivalent aux cardinalités Merise mais attention syntaxe différente)

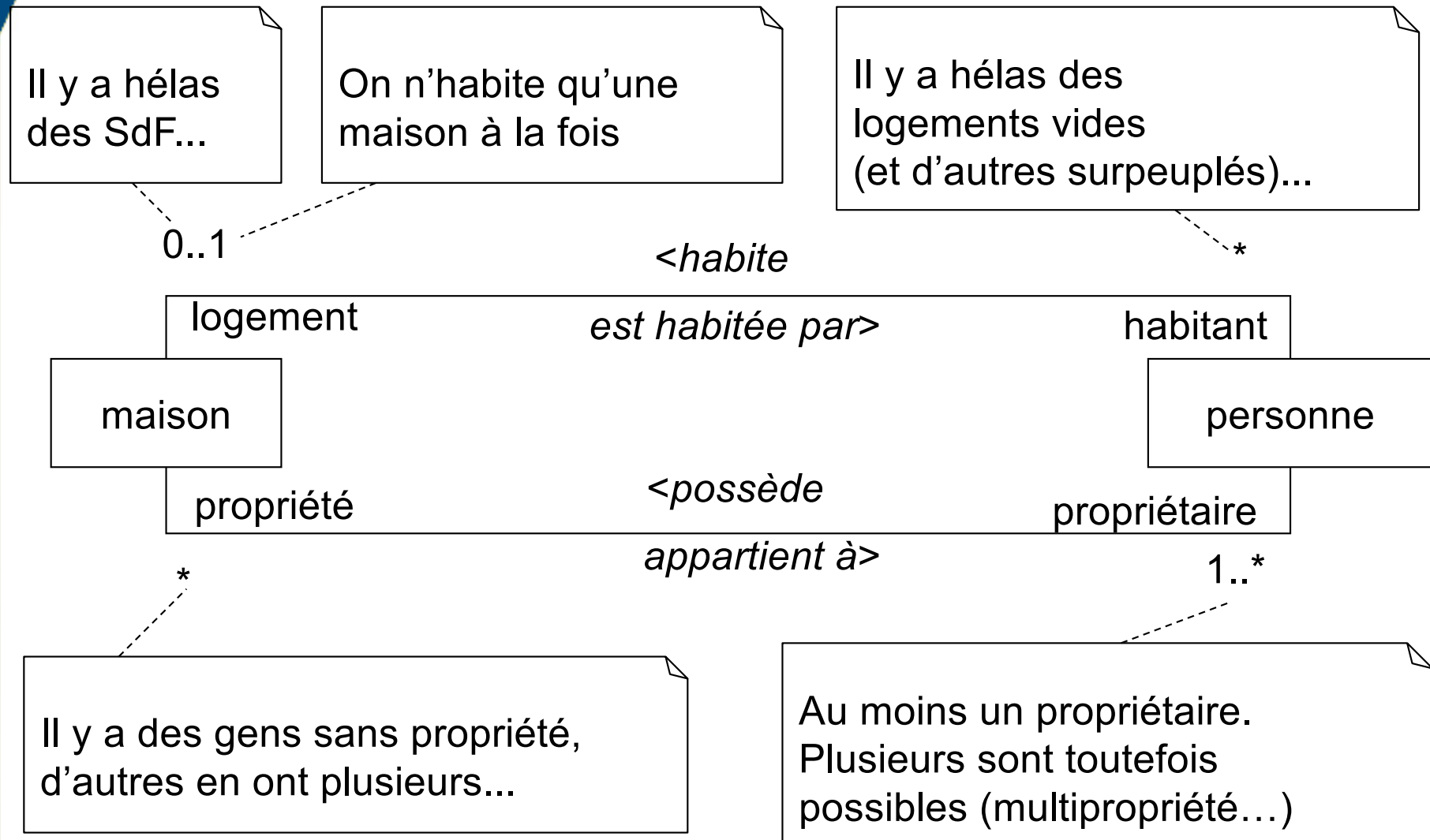


## Expressions valides

N	Exactement N
M..N	Entre M et N (bornes comprises)
M..*	Au moins M (inclus) : pas de borne supérieure
*	Equivalent de 0..*

# Multiplicité des associations

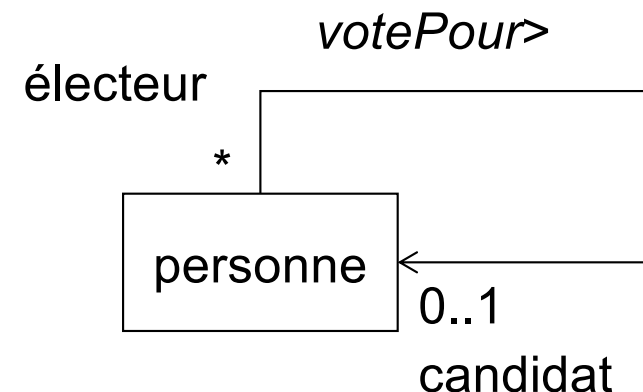
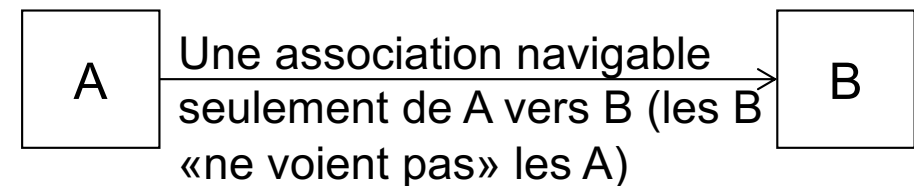
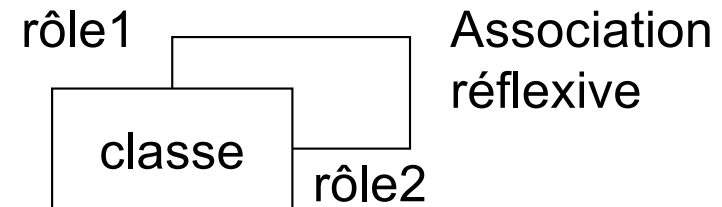
5



# Réflexivité, Navigabilité

6

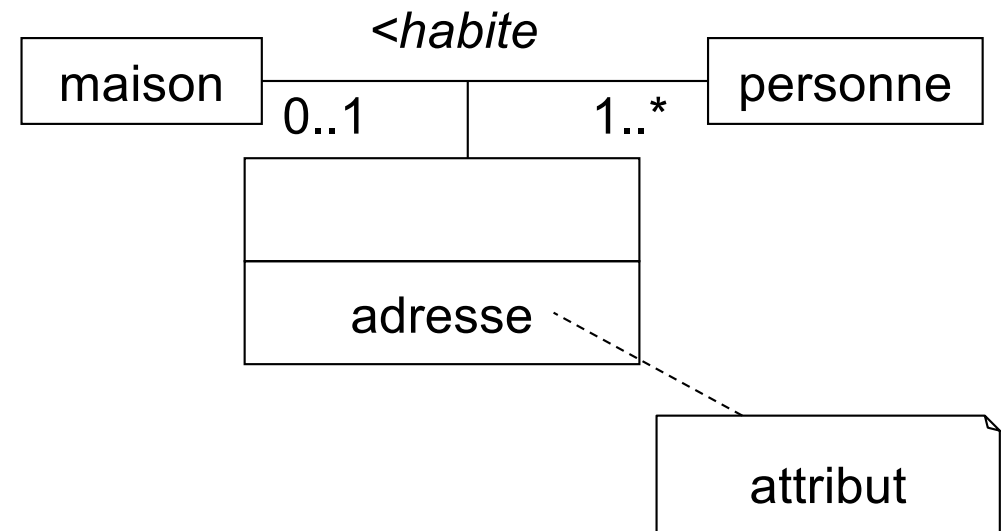
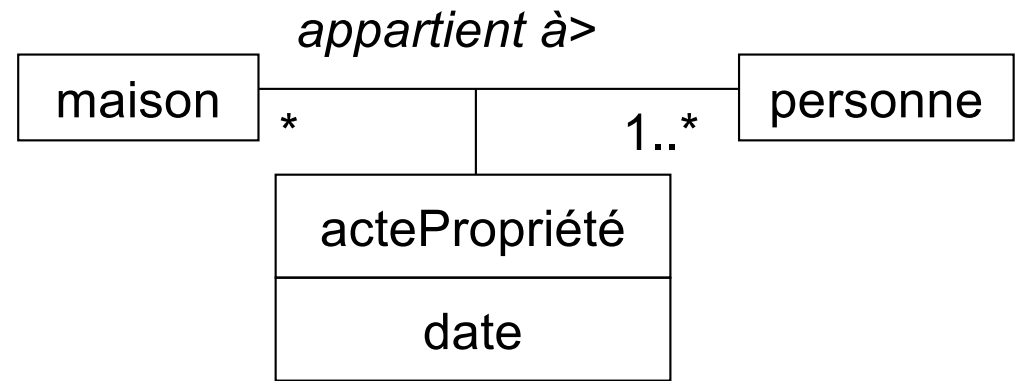
- Une association peut relier une classe à elle même (nom de rôle indispensable)
- On peut aussi restreindre le sens de navigation (sens possibles d'envoi de messages entre objets instances de la classe le long des liens instance de l'association). Par défaut une association est navigable dans les 2 sens.



# Classes-Associations

7

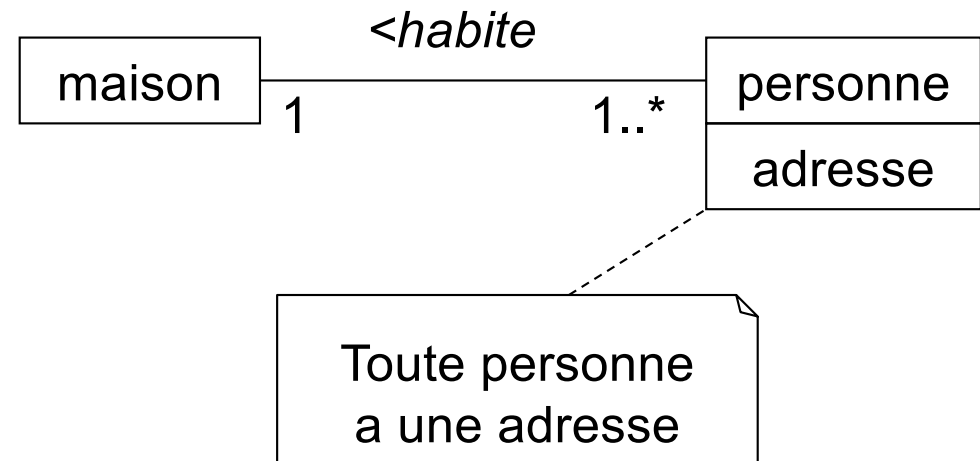
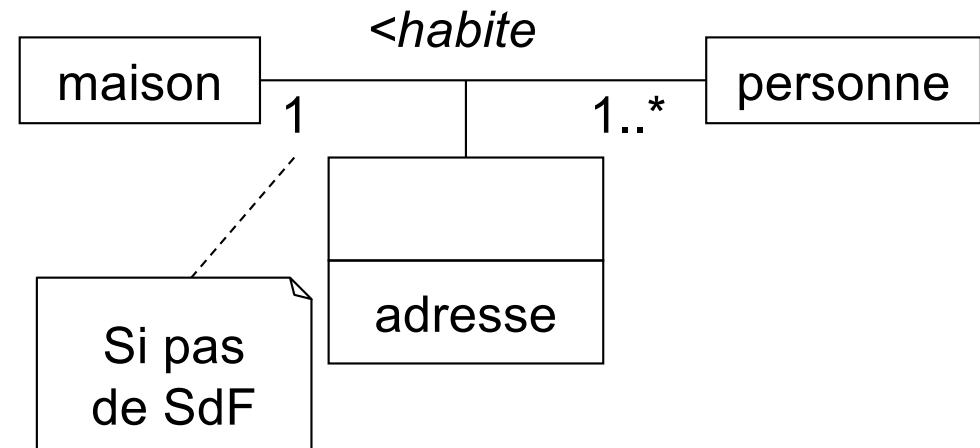
- Il est possible de représenter une association par une classe pour y associer des attributs ou opérations
- Si la classe-association n'a pas elle-même d'associations, le nom est facultatif



# Classes-Associations

8

- Dans le cas d'association 1-N (multiplicité 1 d'un côté), la classe-association est en général transformable en attribut du côté classe de multiplicité N

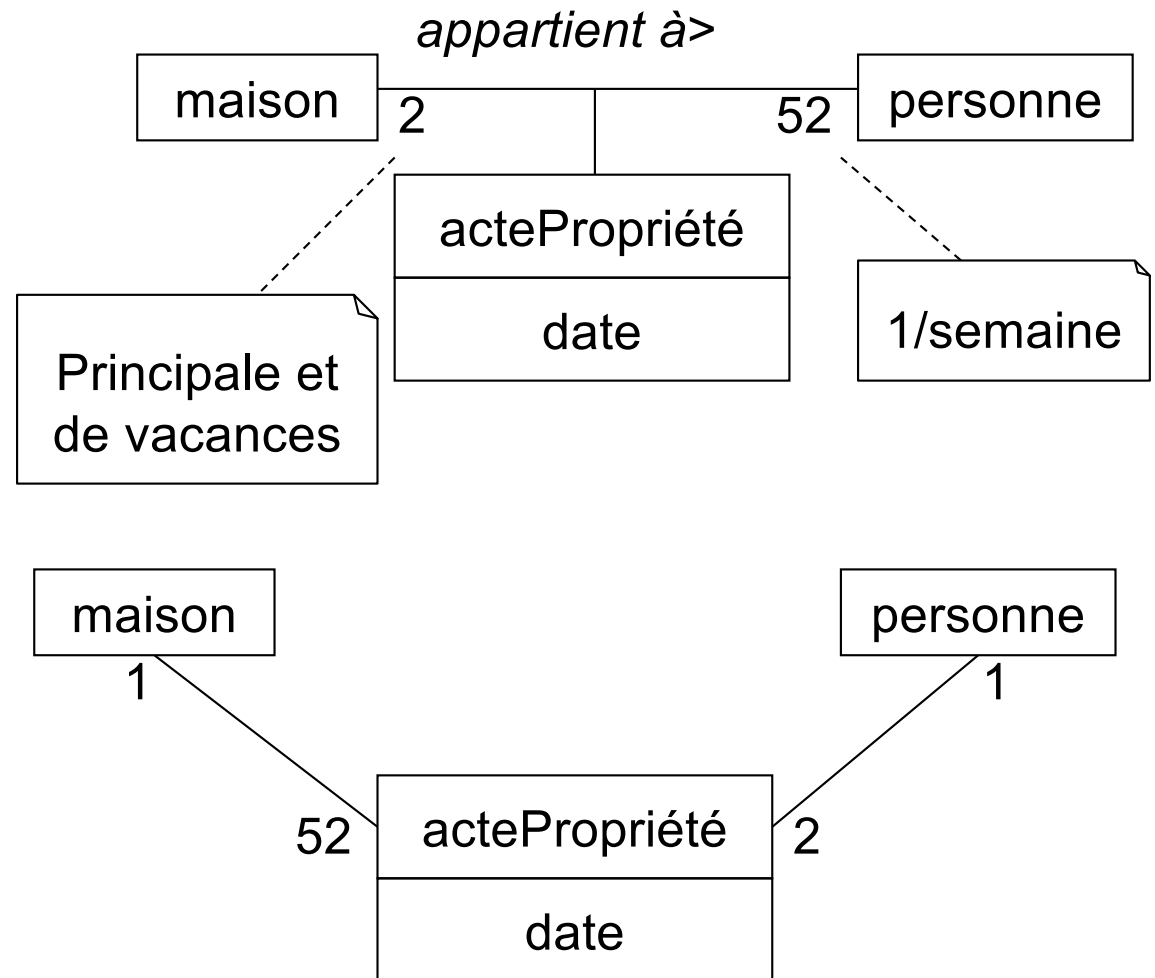




# Classes-Associations

9

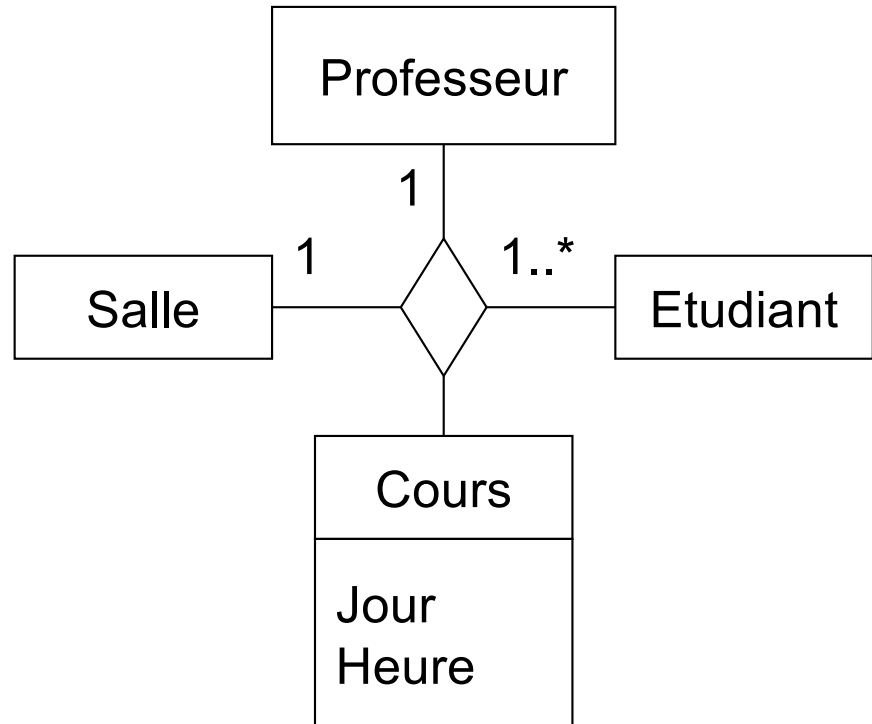
- Dans le cas contraire, on pourra en conception introduire une classe intermédiaire relais (attention au «croisement» des multiplicités en évidence sur cet exemple)



# Arité des associations

10

- En général binaires (relient 2 classes). UML permet toutefois l'expression d'arités supérieures.
- Cette notation est parfois source de confusion dans les multiplicités (un n'étudiant n'a pas qu'un professeur)

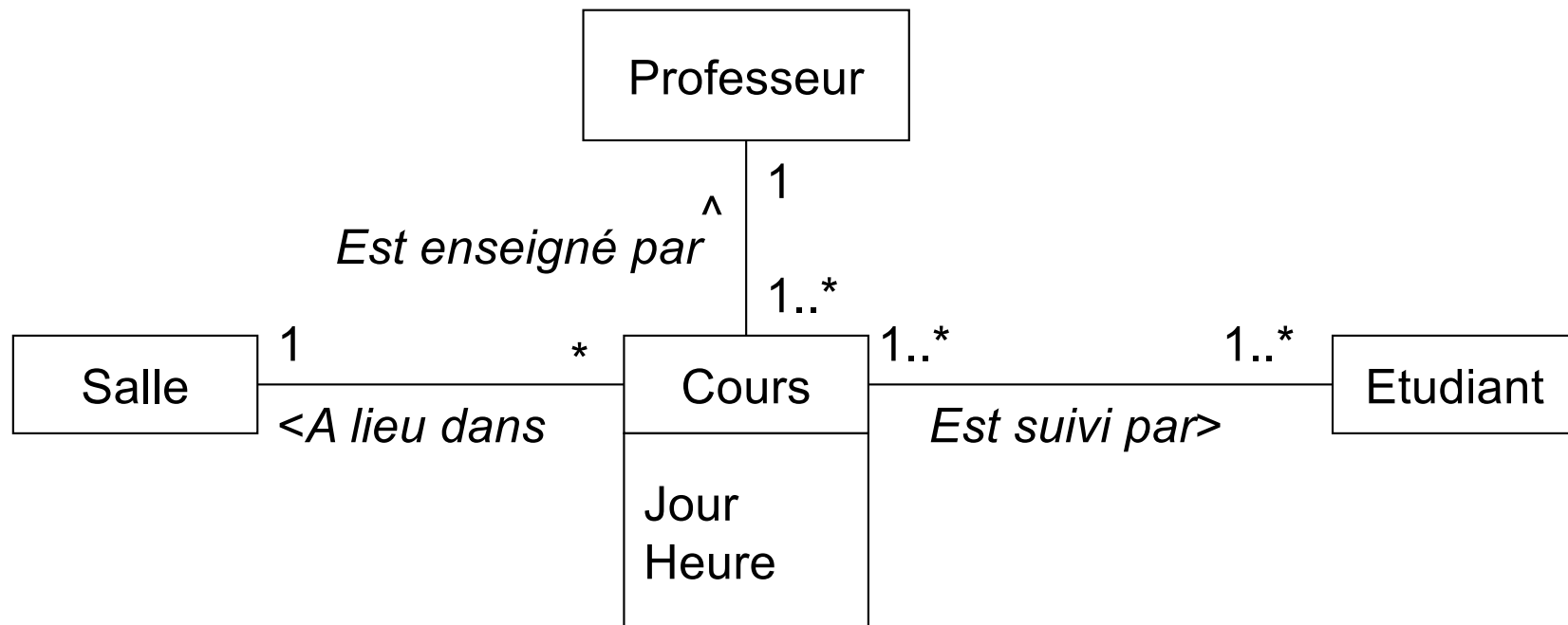


Les association n-aires peuvent souvent être promues au rang de classes-associations

# Arité des associations

11

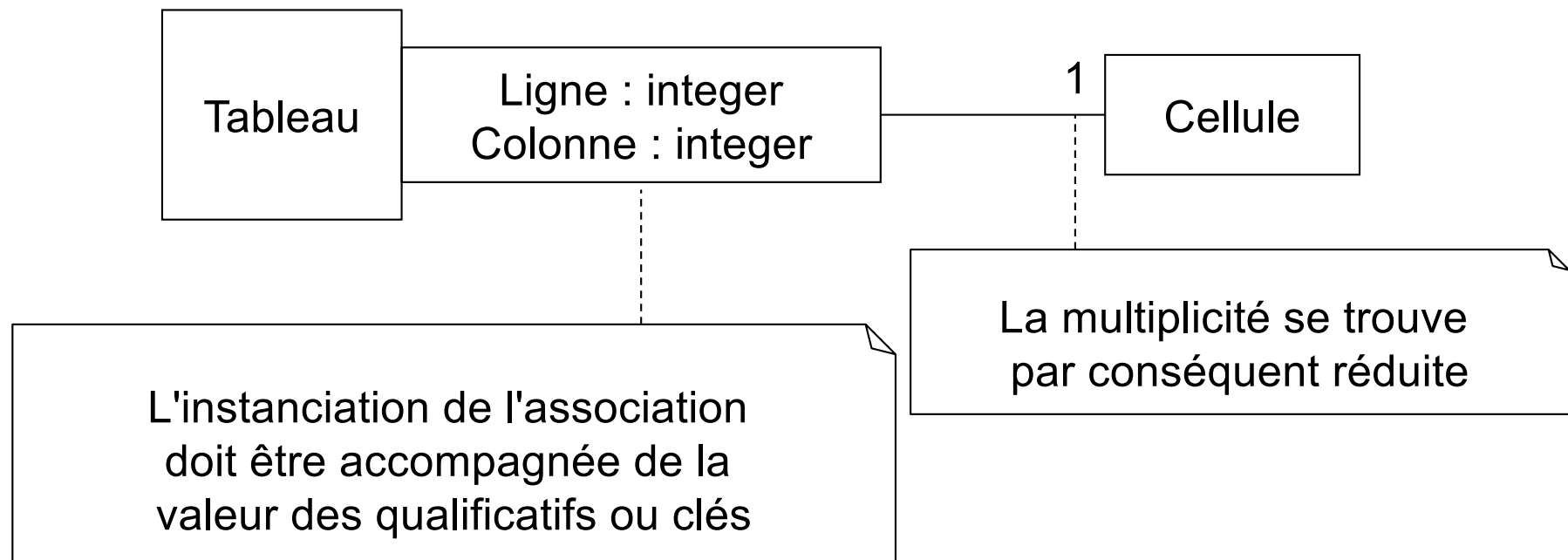
- En conception, la classe-association n-aire devient en général une classe relais :



# Qualification des associations

12

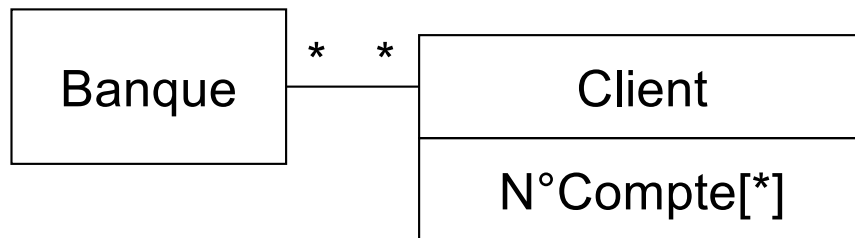
- La qualification ou restriction des associations sert à sélectionner un sous ensemble des instances d'une classe susceptible de participer à l'association :



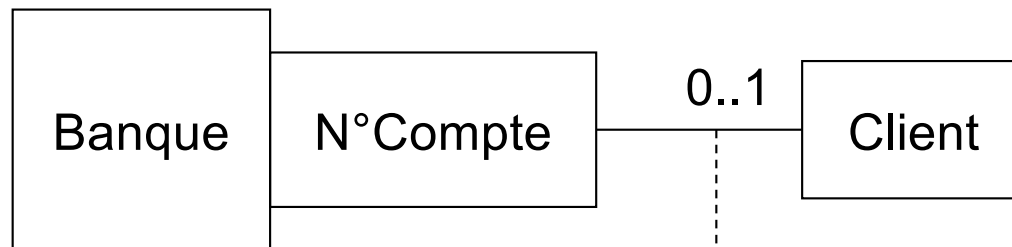
# Qualification des associations

13

- Offre une alternative intéressante au recours aux attributs de la classe destination dans le cas d'attributs pouvant avoir une multiplicité nulle :



Un client peut avoir 0 ou plusieurs comptes



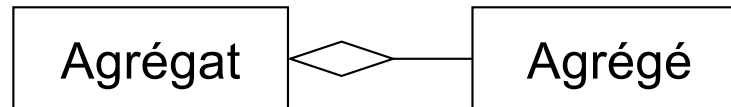
L'association qualifiée a une multiplicité 0..1 côté classe destination

Un N° correspond au plus à un client  
(peut ne correspondre à aucun client)

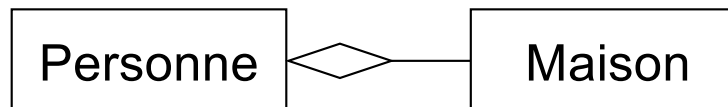
# Agrégations

14

- Associations particulières où une des classes (l'agrégat) joue un rôle prédominant sur l'autre (l'agrégé) :



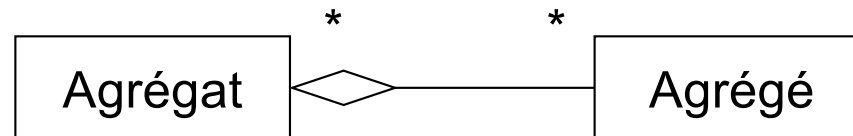
- Souvent utilisé pour exprimer qu'un agrégat possède 0 ou plusieurs agrégés :



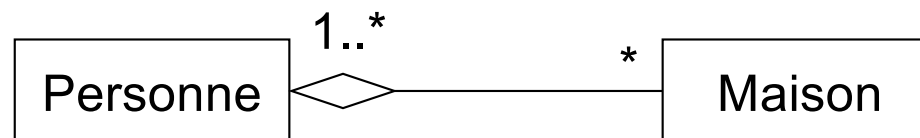
# Agrégations

15

- Il n'y a par contre aucune contrainte de multiplicité ni sur le cycle de vie de l'un par rapport à l'autre :



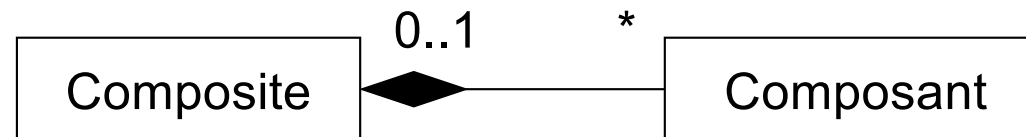
- Une maison peut exister indépendamment d'une personne (et inversement). Une maison peut appartenir à plusieurs personnes (copropriété). Ici le modèle précise qu'il y a au moins un propriétaire :



# Compositions

16

- Agrégations particulières permettant de traduire la notion de composite/composant ou contenant/contenu.
- Implique une contrainte de multiplicité : un composant n'est pas partageable => la multiplicité côté composite est nécessairement 0..1

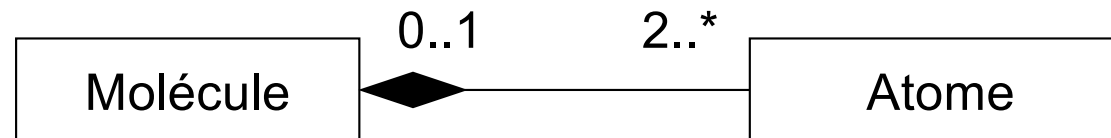




# Compositions

17

- La composition implique également une contrainte sur le cycle de vie : la destruction du composite implique celle de tous ses composants.

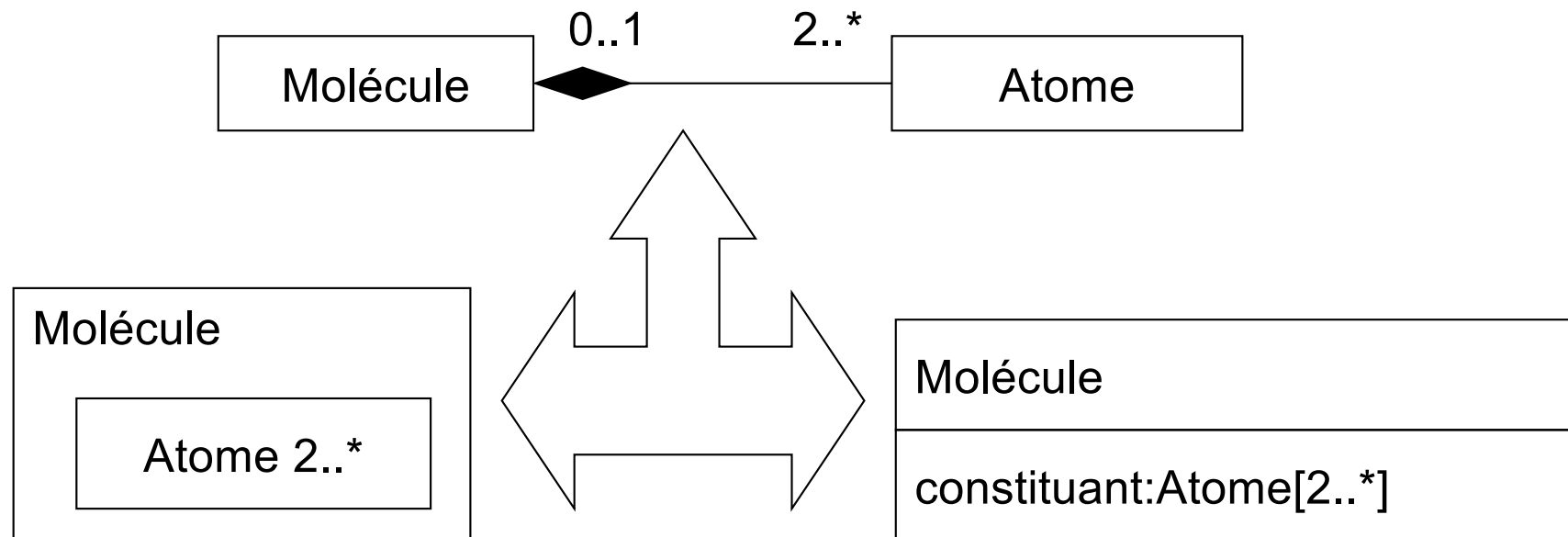


- Dans ce modèle de la réalité la destruction de la molécule implique celle de tous ces atomes (malgré Lavoisier : "rien ne se perd, rien ne se crée, tout se transforme". Il s'agit bien d'un modèle qui peut suffire pour une application particulière).

# Compositions, Attributs

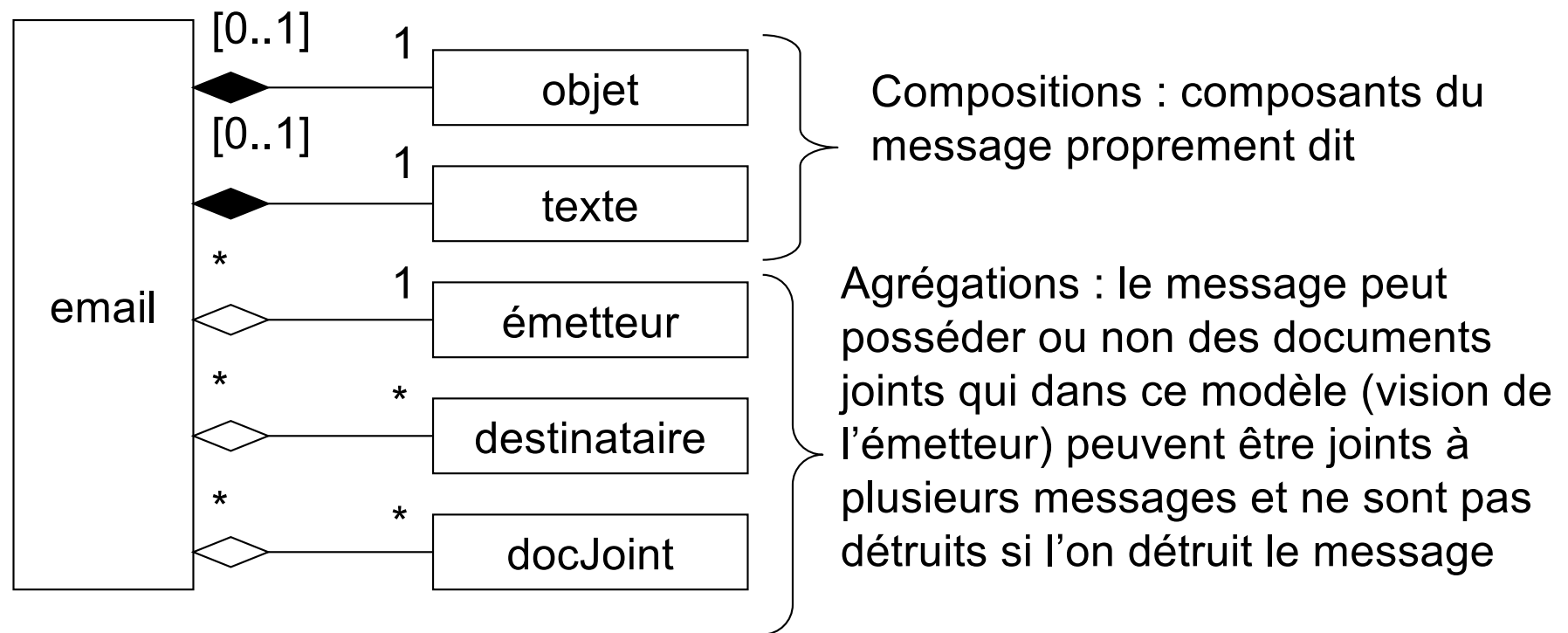
18

- Les compositions peuvent aussi se représenter par des rectangles emboîtés
- Les compositions sont sémantiquement équivalentes aux attributs (à quelques nuances subtiles près...)



# Agrégations et Compositions : Exemple

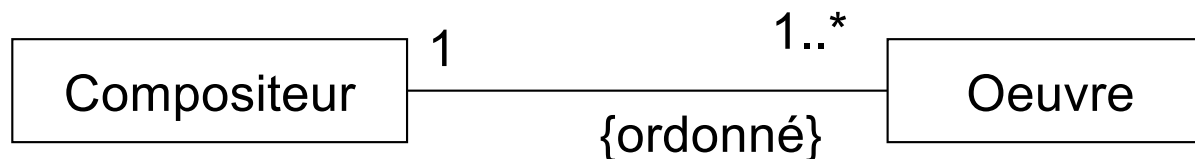
19



# Contraintes sur les associations

20

- Outre la multiplicité, d'autres types de contraintes peuvent être définis :



On souhaite conserver un ordre donné

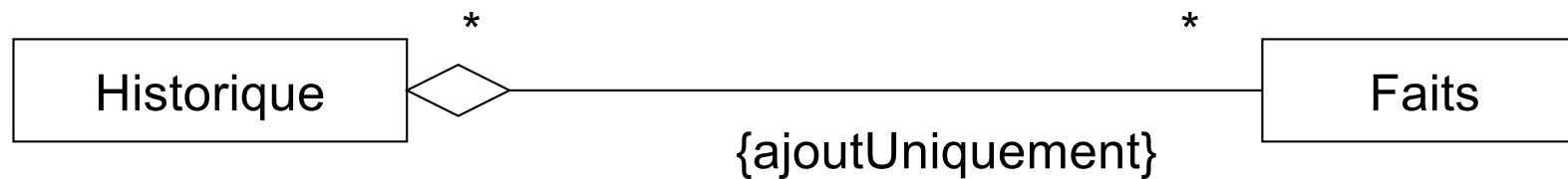


L'instance à l'extrémité mentionnant {gelé} n'est pas modifiable (modèle excluant les remariages...).

{gelé} semble avoir disparu (provisoirement ?) en UML 2.0

# Contraintes sur les associations

21



Un historique ne doit pas être amnésique

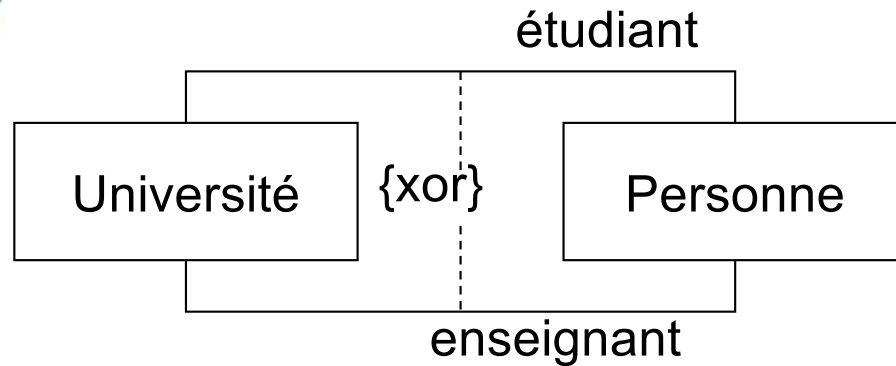


`{individu.DateNaissance < 1/1/1979 and  
individu.Sexe = Masculin}`

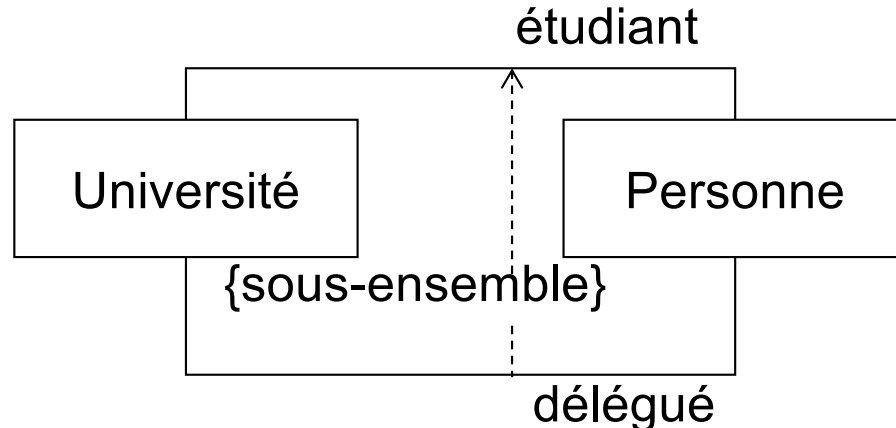
Seuls les individus de sexe masculin et nés avant le 1/1/1979  
peuvent être appelés...

# Contraintes sur les associations

22



On est soit étudiant soit enseignant, mais pas les deux : ou exclusif (xor)



Par contre les délégués sont également des étudiants

Bien d'autres possibilités sont offertes par le langage de contraintes OCL (Object Constraint Language)

# Adaptations SysML

23

Aucune sur ce sujet là :

- **Association** identique (avec sens de navigabilité réduit éventuellement)
- Notions de **multiplicités** identiques avec la même notion d'instanciation de bloc que d'instanciation de classe
- **Agrégation** identique (très utilisée pour faire de la **délégation** : un bloc comporte ou possède un `type_bloc`)
- **Composition** identique (très utilisée en alternative au compartiment « parts » d'un bloc)