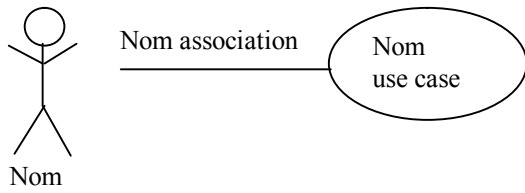


UML : Modèle statique

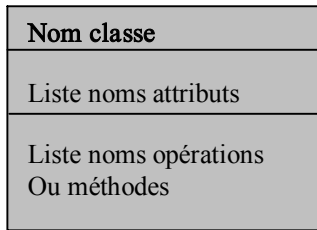
NIAM

Acteur, use case, association

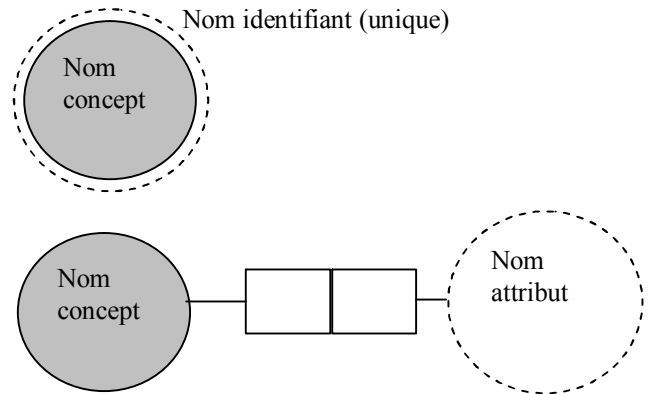


concepts absents

Classe, attribut, opération



Concept ou ensemble, identifiant



L'identifiant est absent.

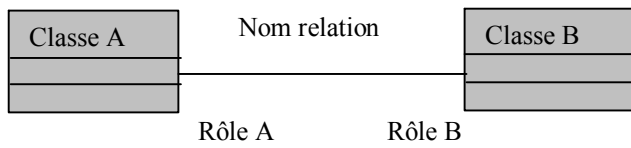
Il est repéré par un attribut à qui on donne implicitement le statut de principal.
A n'en pas douter, l'évolution d'UML comblera cette absence.

Concept opération absent

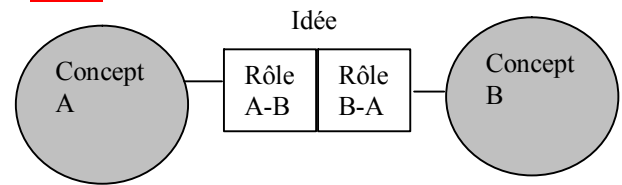
UML : modèle statique

NIAM

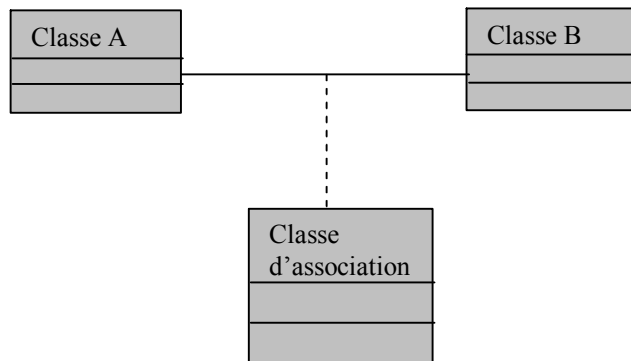
Association (entre deux classes) Rôles



Idée (ou relation binaire) entre deux concepts (ensembles) Rôles

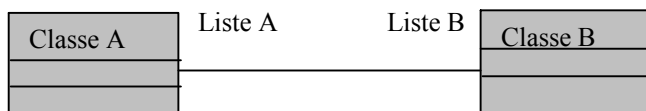


Classe d'association



Ce concept absent de NIAM est très pratique pour qualifier le nouvel ensemble constitué par les couples d'objets pris dans l'association A-B

Cardinalités : ensemble des entiers positifs



Exemples = 0..1 ; 1 ; 1..3 ; *

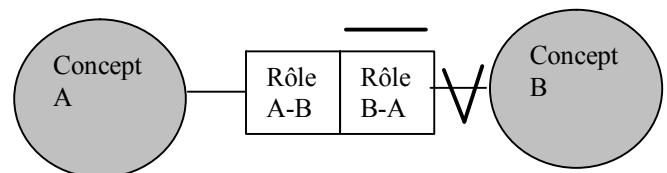
* = plusieurs,

On peut de plus exprimer des contraintes de cardinalité sur chaque ensemble.

(Voir plus loin)

Les cardinalités du sens de l'association B vers A sont obligatoirement l'inverse de celles de A vers B.

Cardinalités seulement binaires



Absence de symbole : pas de contrainte

Unicité

—

V

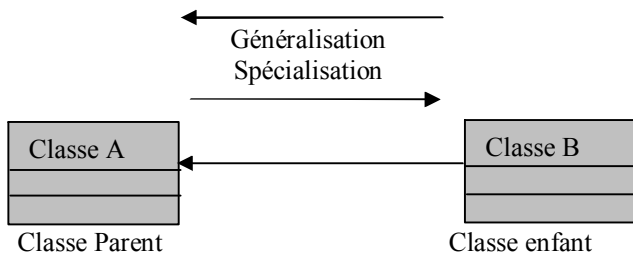
Totalité

Les cardinalités du sens de l'association B vers A sont indépendantes de celles de A vers B .

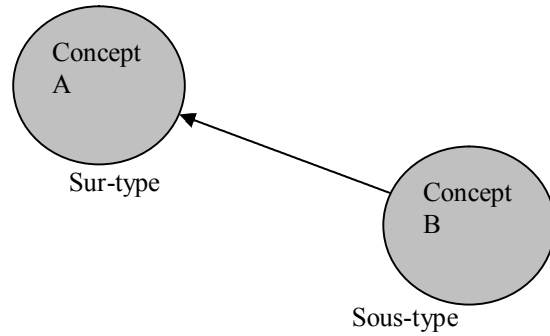
UML : modèle statique

NIAM

Généralisation-spécialisation



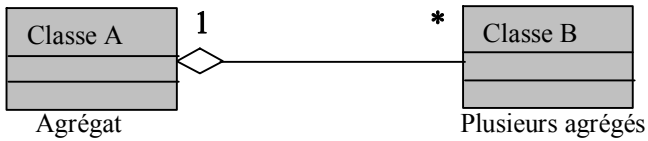
Relation d'héritage



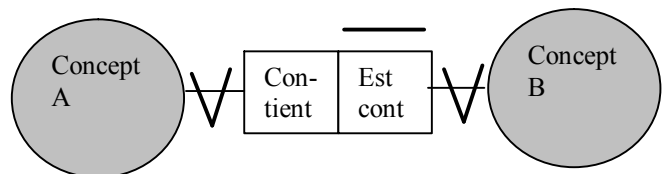
Classe abstraite (sans instances)

Abstract super-type (sans occurrences)

Association particulière d'agrégation (composé de, composant ou faisant partie de)

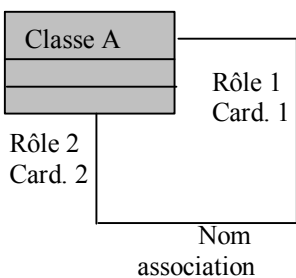


NIAM ne distingue pas de noms particuliers pour une idée. Ce sont les rôles qui définissent la sémantique de la relation. Cette relation d'agrégation se traduit par le schéma :

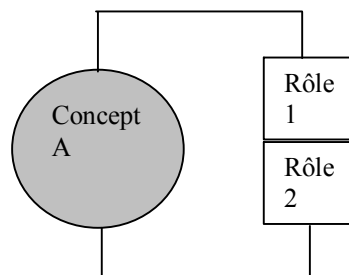


Le symbole flèche « losange » se place toujours du côté de l'agrégat, le contenant.
 Traduction en phrase élémentaires :
 Un objet de la classe A, par exemple un Appartement, est composé de plusieurs pièces.
 Plusieurs pièces font partie d'un appartement.
 La sémantique de cette association implique deux propriétés
 - les associations opèrent une partition exclusive dans l'ensemble agrégé (une pièce ne peut appartenir à deux appartements)
 - si on supprime un appartement, les pièces associées disparaissent.
 Pour conserver explicitement la trace du contenu de chaque Appartement, il faudrait créer une classe d'association.

Association unaire

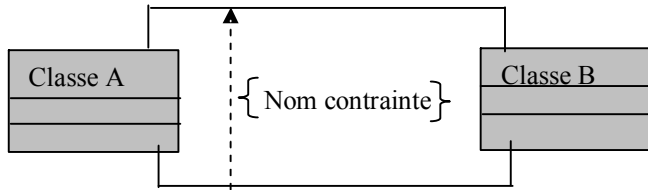


Idée portant sur le même concept



UML : modèle statique

Contraintes entre deux associations

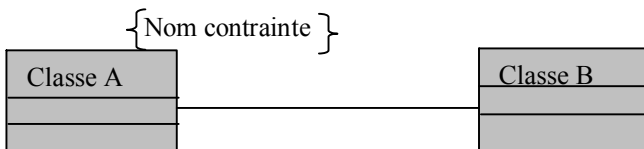


Se note entre accolades.

Contrairement à NIAM, les contraintes, leur nom, leur signification dans UML sont libres.

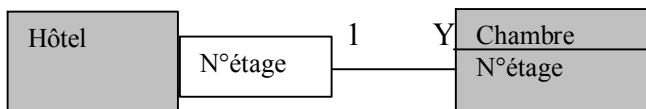
Attention de ne pas en abuser : l'informaticien devra les traiter !

Contrainte portant sur les instances d'une classe :



Le principe de qualification libre d'une contrainte sur les occurrences d'un concept n'existe pas dans NIAM

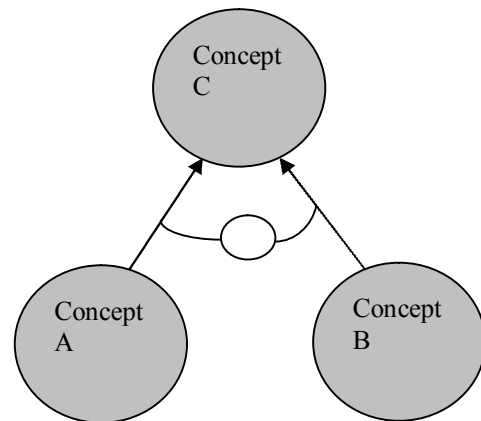
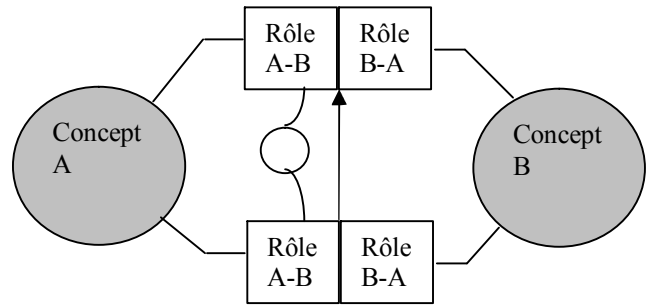
Les qualificateurs. Prenons un exemple



Traduction : Dans un hôtel, il y a des chambres, qui portent entre-autre un numéro d'étage comme attribut. Le qualificateur permet de compter les chambres par étage :
Pour un numéro d'étage, il y a Y chambres.

NIAM

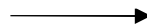
Contraintes entre deux idées ou entre deux relations d'héritage



Le symbole 

Peut contenir une contrainte

- d'unicité
- d'égalité
- d'exclusion
- de totalité

Le symbole 

Est une contrainte de sous-ensemble.

Dans NIAM il vaut mieux créer un concept à la place d'un attribut chaque fois que l'on doit utiliser un identifiant comme ici le numéro. Le schéma NIAM correspondant à cet exemple comportera donc 3 concepts (Hôtel, Chambre et étage), et deux idées. Le schéma est moins concis, mais gagne en clarté. C'est d'ailleurs vrai aussi pour UML.