



## Cours 2 : Le modèle navigationnel

Riadh ZAAFRANI - Rim Samia KAABI

Septembre 2014

3ème Année Ingénieur

Spécialité Génie du Logiciel et des Systèmes d'Information (GLSI)

1

## Objet-Relationnel (OR)

- Implémentation du NFNF Non First Normal Form avec interfaces (méthodes) et oid pour les objets relationnels : La technologie OR := oid + types complexes + méthodes
- Types complexes implémentés. Table avec attribut : atomique et complexe: (Type Abstrait de Données: TAD (prédéfini) ou UDT (défini par l'utilisateur));
- Table peut avoir des imbrications (Oracle 8, limite 1niv. , illimité avec Oracle9i, 10g, ...);

2

## Objet-Relationnel (OR)

- **Jointure** peut avoir une formulation **implicite** en utilisant l'attribut de type **REF**;
- **Formulation** plus simple des requêtes; mais les mises à jour peuvent être plus complexes.
- **Héritage et surcharge** ajoutés progressivement ....? Visibilité des attributs partiellement sous contrôle via les vues .. **pour devenir véritablement OO**.

3

## Comment passer du conceptuel au modèle objet-relationnel ?

- La passage à l'OR **peut être plus facile à comprendre** par un détour vers le modèle navigationnel (toujours de niveau conceptuel) proposé par Christian Soutou « Notation graphique proposée par C. Soutou, Objet Relationnel sous Oracle8, Eyrolles, 1999 »

**C'est un passage utile pour des fins didactiques seulement.**

4

## Comment passer du conceptuel au modèle objet-relationnel ?

---

- Le modèle navigationnel (Mnav) est une représentation graphique de l'OR
- Ce modèle Mnav joue le rôle de pivot ou d'étape entre le MCD et le modèle d'implantation objet. C'est un outil pour la modélisation à caractère essentiellement didactique.

5

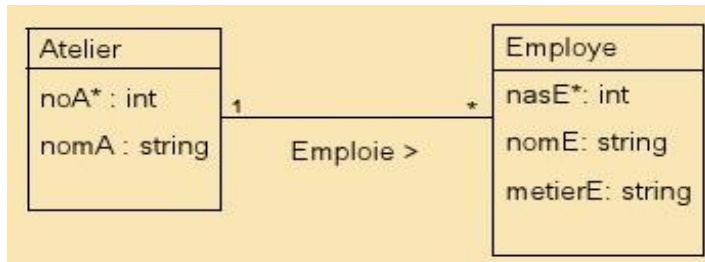
## Rôle du modèle navigationnel

---

- **Caractéristiques du Mnav :**
  - Facilite le passage à l'OR par la visualisation des passages : conceptuel UML=> navigationnel => objet-relationnel
  - Favorise la visualisation des structures complexes et des chemins de navigation.
  - Mais ne représente pas bien le partage des structures.
- Pourquoi le navigationnel ? Le modèle introduit les liens logiques qui sont de type REF (implémentation de la Ref est toujours indéterminée au niveau conceptuel).

6

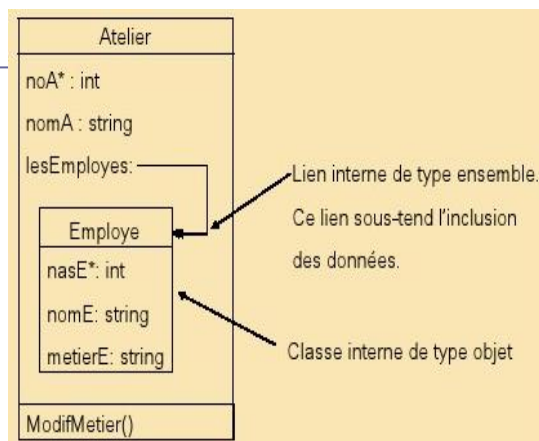
## Mnav versus MR



Le MR: Atelier (`noA*:int`, `nomA: string`) FN3  
Employe(`nasE*`, `nomE`, `metierE`, `noA`) FN3

7

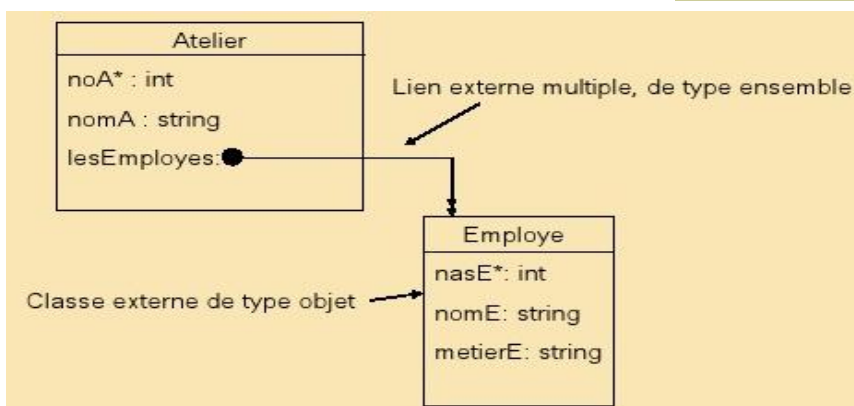
## Mnav correspondant au MR Atelier-Employe



Mnav représente visuellement et directement les données complexes sans normaliser les structures de 1er niveau (plate).  
Avantages: accès direct aux employés d'un atelier!

8

## Autre Mnav correspondant au MR Atelier-Employe



Lien externe utilise une référence pour pointer sur un ou plusieurs employés

9

## Résumé des conventions d'écriture pour le Mnav

- La notion de **classe externe** correspond à une structure de données complexe équivalente à la classe UML. La structure interne ou **classe interne** est aussi une structure de données mais imbriquée dans une classe.
- Le nom d'une classe commence par une lettre majuscule : **Employe**
- Le nom d'un attribut ou d'une méthode commence par une minuscule: **matriculeE**. Ce nom n'a pas d'accent ni autres symboles orthographiques;

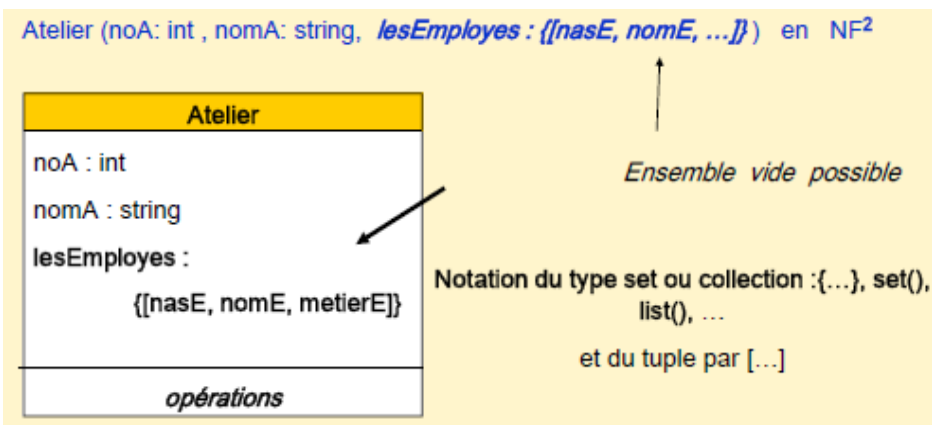
10

## Résumé des conventions d'écriture pour le Mnav

- Le nom d'une association UML commence par une majuscule : **EffectueS**;
- Le type d'un attribut est défini par son nom suffixé par **\_t**
- Le lien interne et externe est une **REF** représente une référence logique sans égard à leur implantation physique; Le lien peut-être simple ou multiple.

11

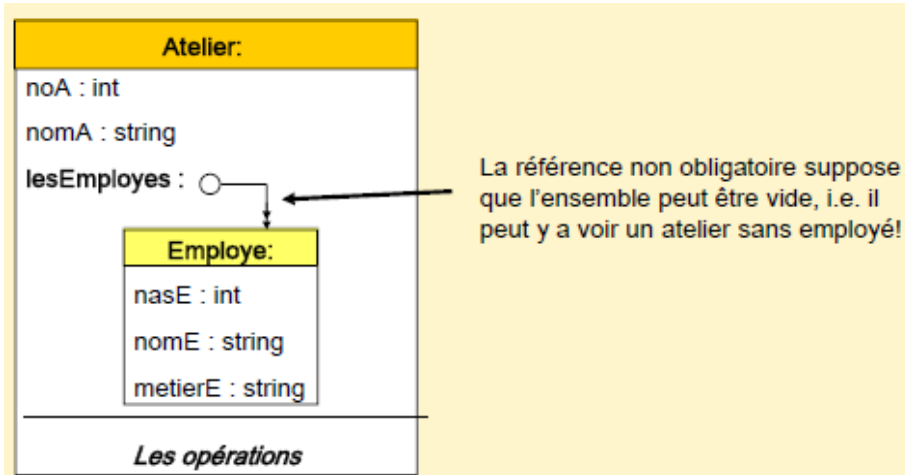
## Classe UML augmentée d'un attribut d'ensemble



- ✓ Attribut **lesEmployes** a un type collection (set) dont les éléments sont des objets d'une autre classe : Employe
- ✓ Une collection ou ensemble peut être une structure typée : ensemble, liste, bag, array, ...

12

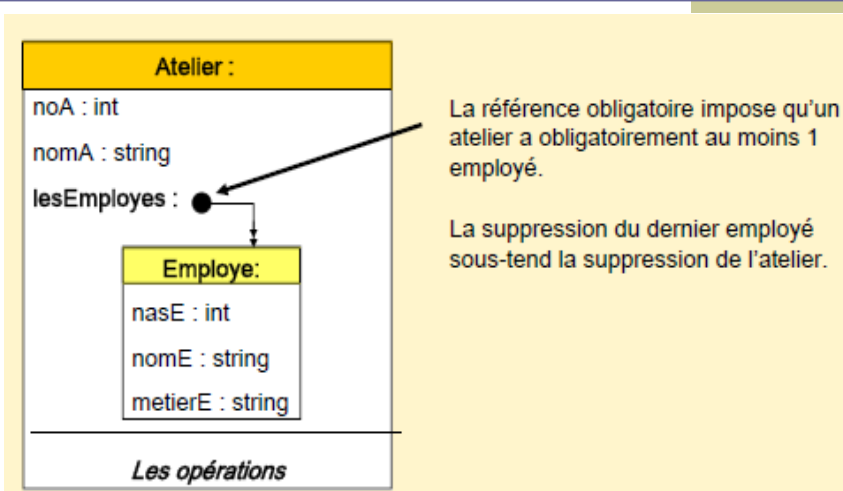
## Modèle navigationnel équivalent



- ✓ Attribut `lesEmployes` a un **lien interne multiple** sur des objets d'une classe `Employee`.
- ✓ Les objets `Employee` sont inclus dans la classe `Atelier`.

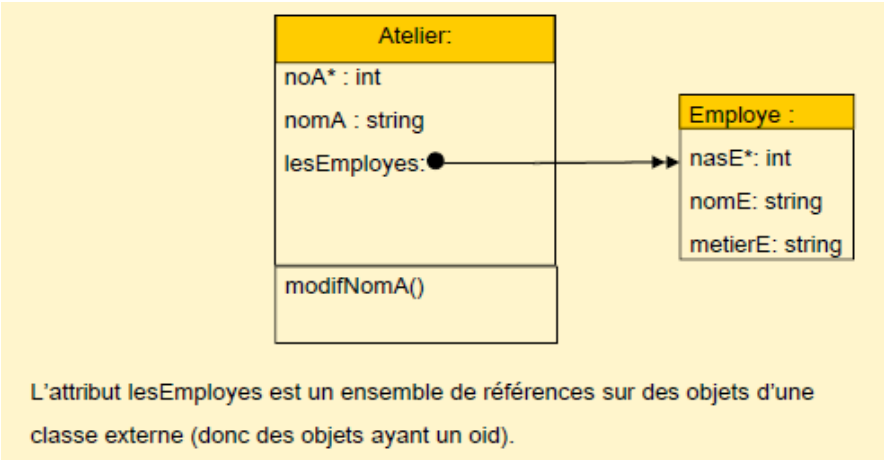
13

## Modèle navigationnel équivalent



14

# Modélisation avec un lien externe



15

# Représentation tabulaire du Mnav avec lien interne multiple

Atelier:	noA: string	nomA: string	lesEmployes : { }		
			nasE : int	nomE: string	metierE: string
	A20	Atelier B	123	Arthur	fraiseur
			234	Sylvie	tourneur
	A30	Atlier C	256	Pierre	perceur
			456	Serge	ponceur

✓ Cette vision de la classe instanciée repose sur la notion de table relationnelle. La table Atelier a un attribut d'ensemble, lesEmployes dont le type est une sous-table (ensemble imbriqué) formée avec les attributs typés : nasE, nomE, metierE.

16

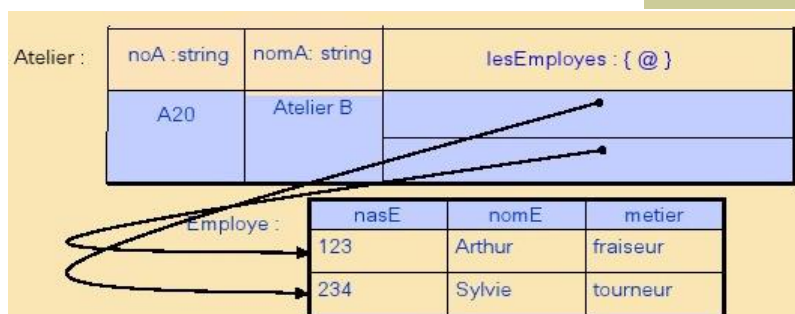


## Nouveau Type pour implémenter le lien externe Ref

- Attribut de navigation entre les classes externes.
- Le type Ref pour un attribut spécifie que ce dernier est **une référence** à un objet (i.e. une structure avec un oid ou l'équivalent).
- Il est possible de contraindre l'attribut de type Ref pour le limiter à cibler une sorte d'objets seulement et au besoin, à ne pas être NULL.

17

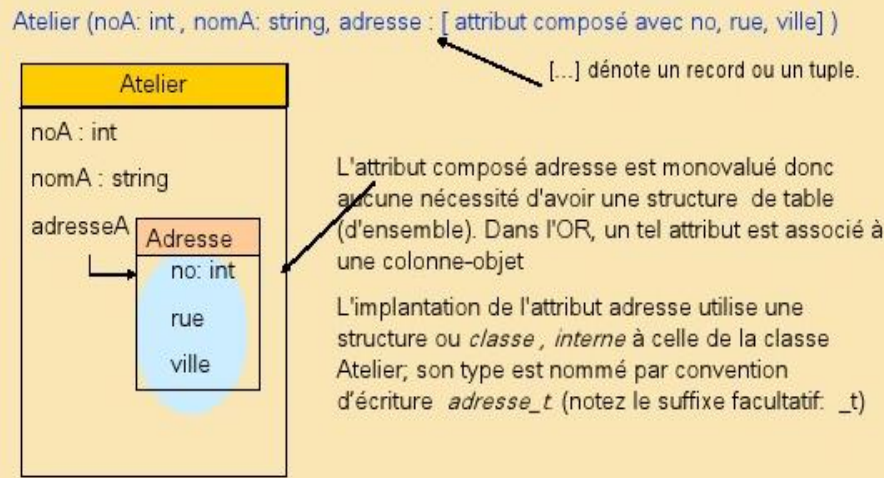
## Mnav : Exemple de collection avec un attribut de type REF



- L'accès à un employé à partir de la table Atelier sous-tend **une navigation sans calcul de jointure**. Par contre, trouver le noA de l'employé123 sous-tend une jointure avec (si elle existe) la référence.

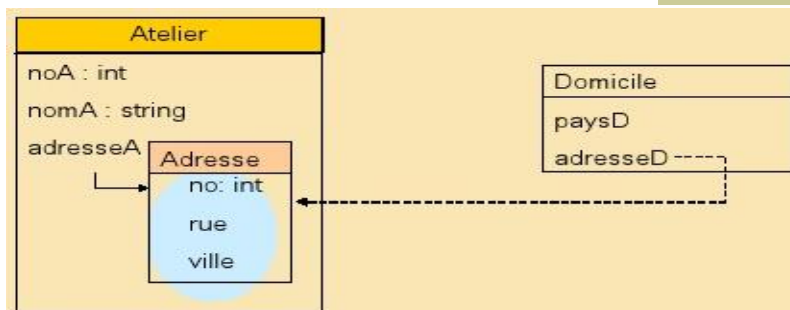
18

## Attribut composé dans une table: lien simple et classe interne



19

## Inconvénient: Partage d'une classe interne difficile



- Le partage de la structure d'un attribut composé (dite classe interne) avec une autre table (ou classe Mnav) est **difficile** à représenter dans le Mnav : partage de adresse par les classes Atelier et Domicile. Possibilité d'utiliser une astuce comme un lien fait d'un pointillé??

20

## Attributs et liens du Mnav

---

### ■ Un attribut peut être :

- simple (chaîne de caractères, nombre, ...)
- Composé d'attributs (appelé aussi une structure interne et noté[...])
- référence (logique : Ref)
- collection (collection de structures (avec attribut composé) ou des références) notée : {...}
- Combinaisons des types précédents ← TAD

21

## Attributs et liens du Mnav

---

### ■ Lien intra (interne) et inter (externe):

- **Interne (intra):** à l'intérieur d'une même classe (intra-classe) incorporant alors une structure complexe composée et typée; c'est un **lien non-navigational** -- (avec Oracle: correspond à un objet-colonne, OC inclus dans la table);
- **Externe (inter):** entre deux classes (explicite) en incorporant des valeurs de Ref. C'est un **lien navigational**.

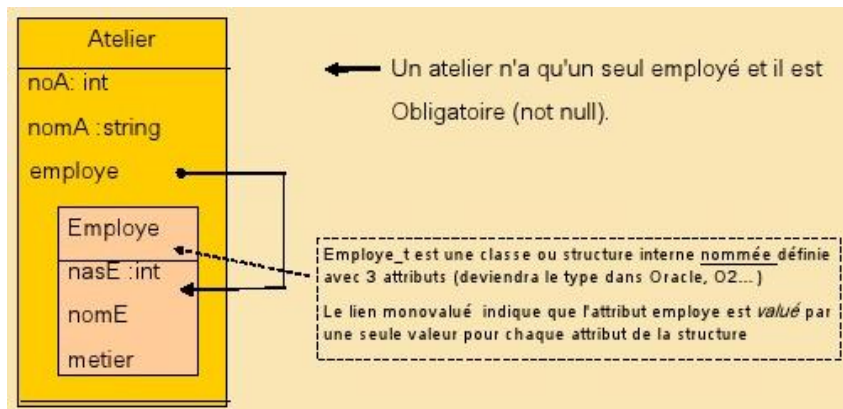
22

# Lien simple et multiple

- Un **lien simple** peut être
  - 1) **Interne**: pour associer à un attribut une seule valeur complexe (dit objet-colonne)
  - 2) **Externe**: pour associer un objet externe appartenant à une autre classe.
- Un **lien multiple** (sous-tend l'ensemble\*) associe plusieurs instances (ou valeurs).
  - 1) **Lien interne** pour associer plusieurs objets-colonnes ou valeurs complexes incluses dans la classe.
  - 2) **Lien externe** pour associer plusieurs objets
- \*En Oracle, un ensemble est une collection (terme plus générique).

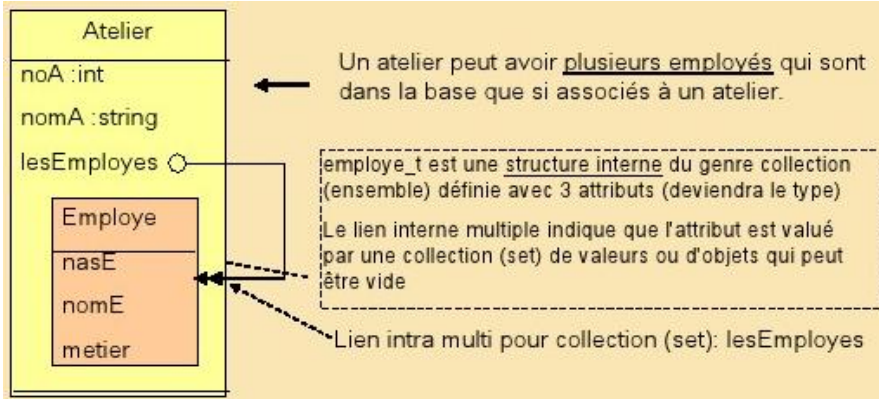
23

## Autre exemple de l'attribut composé: lien interne simple



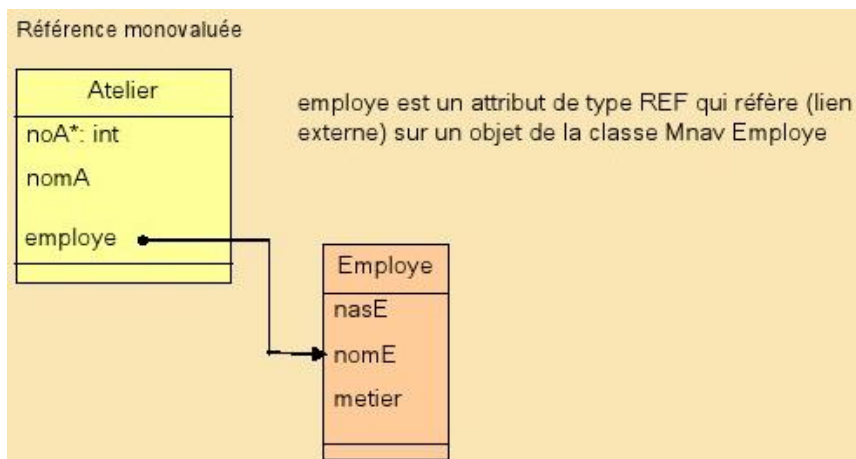
- La structure de stockage est généralement du genre contiguë et incluse avec la structure physique de Atelier. 24

## Autre exemple de l'attribut composé: lien interne multiple

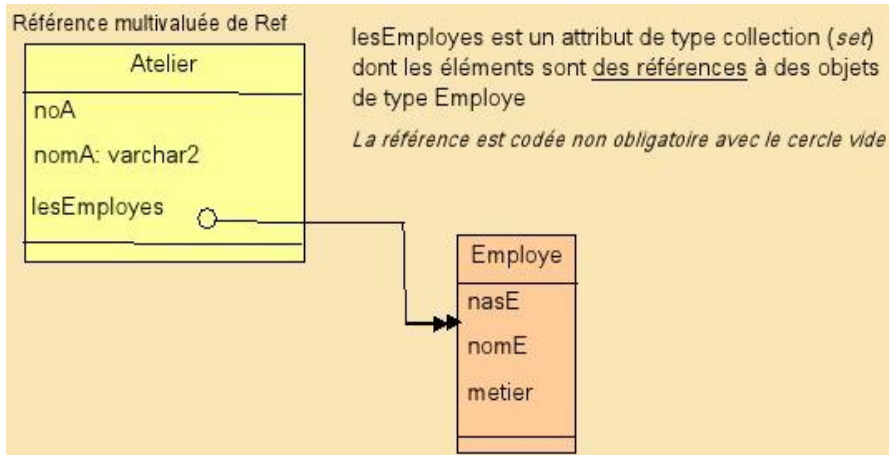


- Dans Atelier, l'attribut lesEmployes a une structure (type) dont les éléments sont des employés (employe\_t) imbriqués dans la classe Atelier.

## Autres implémentations possibles : Le Mnav a un attribut composé et un lien externe simple



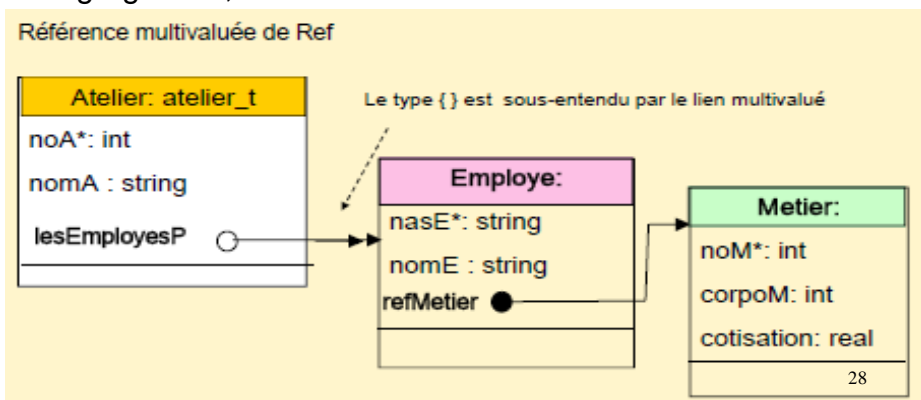
## Représentation graphique de l'attribut d'ensemble: avec lien externe multiple



27

## Cascade des liens avec le modèle navigationnel (Mnvg)

- Représentation explicite par les liens externes: des structures de données complexes, des hiérarchies et des agrégations;



28

## Avantages escomptés du modèle navigationnel (Mnav)

---

- Le passage du DC-UML vers le modèle OR est souvent plus simple à concevoir via le Mnav lorsque le modèle est petit.
- Un choix plus judicieux des liens entre les classes d'objets est possible;
- Une description plus explicite des associations n-aires.
- La visualisation des chemins par les liens rend (souvent) plus facile la formulation des requêtes;

29

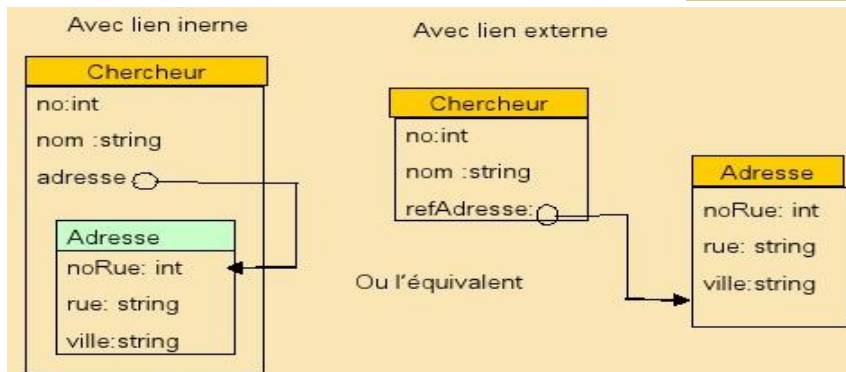
## Inconvénients du modèle navigationnel

---

- Le partage d'un même type structure interne par plusieurs classes n'est pas représenté adéquatement.
- Il demeure possible (mais **déconseillé**) de débiter la conception d'une base OR par l'élaboration directe d'un schéma navigationnel, donc sans DC-UML.
- Peut être parfois plus difficile de raisonner du Mnav que de raisonner à partir d'un MCD;
- **Etape non essentielle ajoutée : passage direct du DC-UML vers le OR possible sans transiter par le modèle Mnav.**
- C'est un modèle didactique...

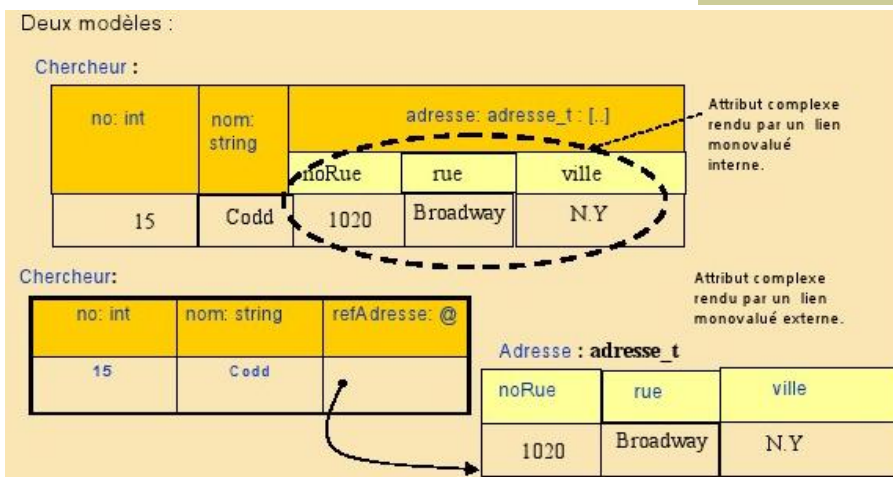
30

## Exercice : Modèle Mnav du Chercheur



- 1) Quelle est leur représentation logique tabulaire dans les 2 cas ?
- 2) Quelle information fournit le 1<sup>er</sup> modèle Mnav au sujet des multiplicités du diagramme de classes correspondant? 31

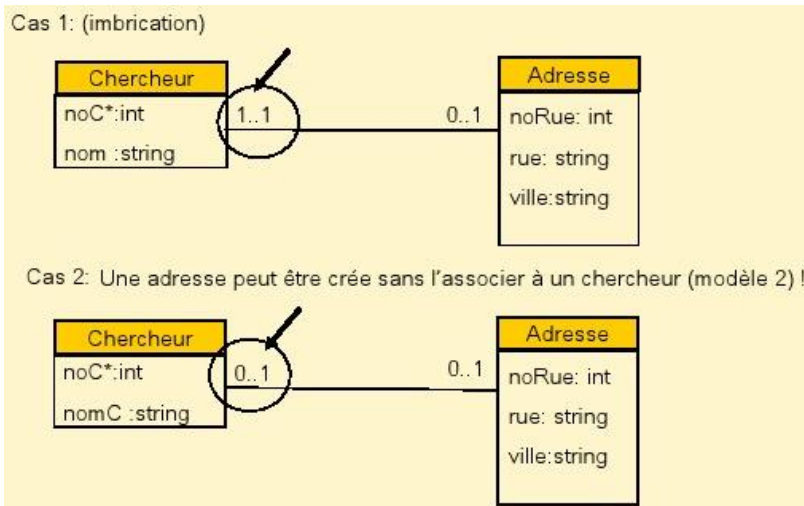
### 1. Quelle est leur représentation logique tabulaire dans les 2 cas ?



32



## 2. Quelle information fournit le 1<sup>er</sup> modèle Mnav au sujet des multiplicités du diagramme de classes correspondant?



33

## Sommaire de la représentation graphique

- Dans un Mnav, le lien simple est représenté par une flèche simple (→) et le lien multiple par une flèche à double pointe (→→).
- Le lien interne (intra) associe un attribut de structure complexe participant à la définition de la classe.
- Le choix entre un lien interne ou externe est basé sur des considérations d'exploitation : efficacité et le sens de la navigation exprimé par la flèche dans le diagramme de classes UML.

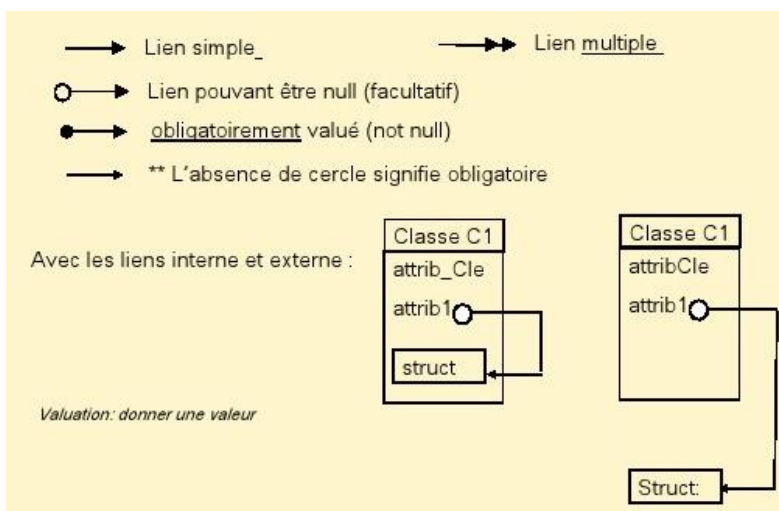
34

## Sommaire de la représentation graphique

- En bref: le **lien externe** évite la formulation explicite des jointures, mais rend les mises à jour et ajouts plus complexes : les **REF** doivent être ajoutées et des contraintes sur les Ref doivent être gérées correctement : **is dangling\*** ou **is not null**. \*Lorsque l'objet référencé est supprimé, la référence devient invalide. On parle alors de dangling ref. Le prédicat IS DANGLING permet de savoir si une ref est valide ou non.
- Le **lien Interne** facilite la suppression des objets : suppression d'une instance de Chercheur entraîne celle de Adresse (son adresse).

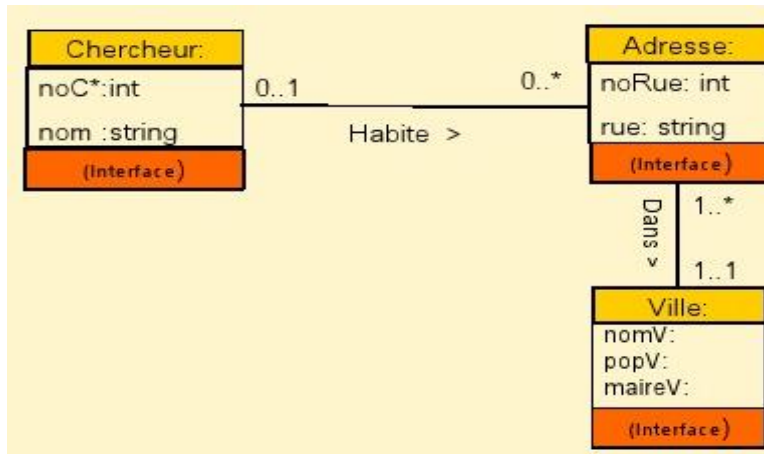
35

## Valuation de l'attribut Ref



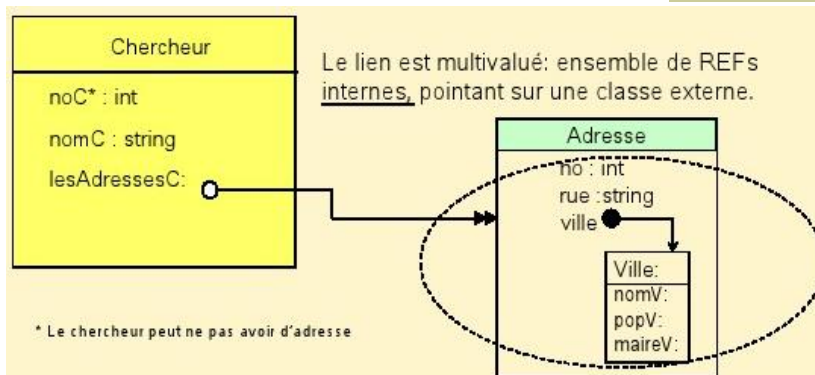
36

## Cascade d'associations (Exemple)



37

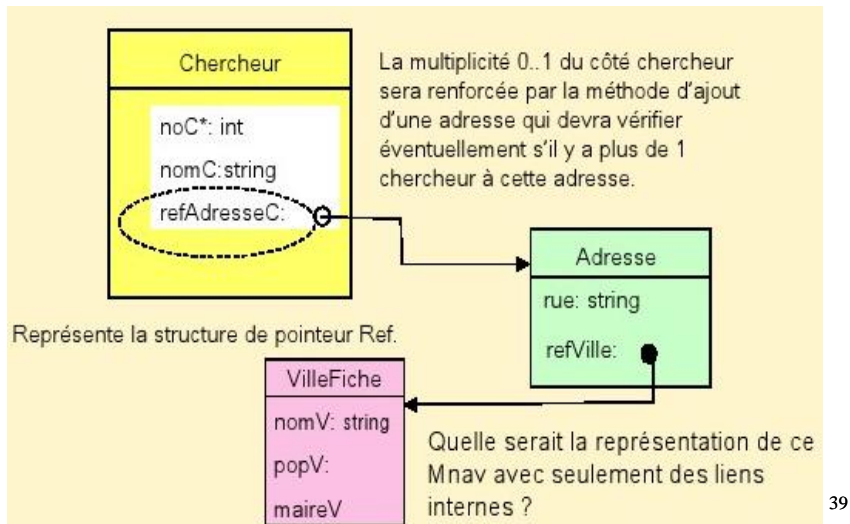
## Modélisation avec un seul lien externe multiple



- ✓ L'attribut **lesAdressesC** est une collection ou `set()` de liens externes de type Ref (ou de référence) sur la structure **Adresse** qui est externe à la classe **Chercheur**. Ce lien externe se concrétise par un ensemble de références dans la classe **Chercheur**.

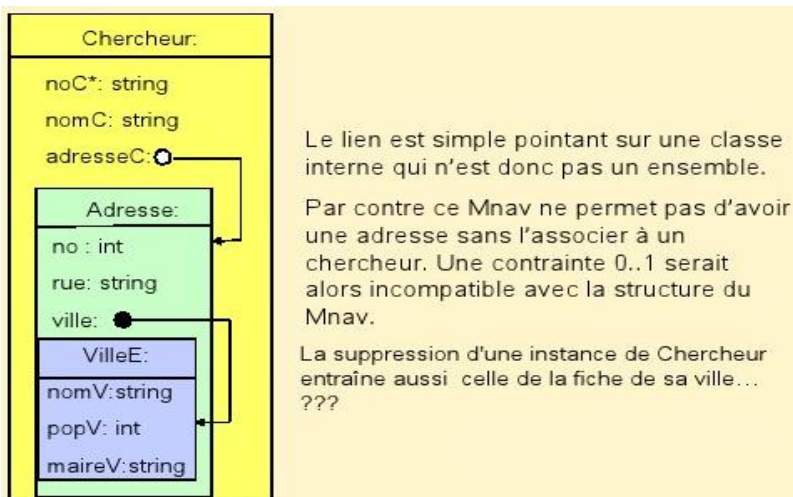
38

## Modèle avec deux liens externes simples à plusieurs niveaux



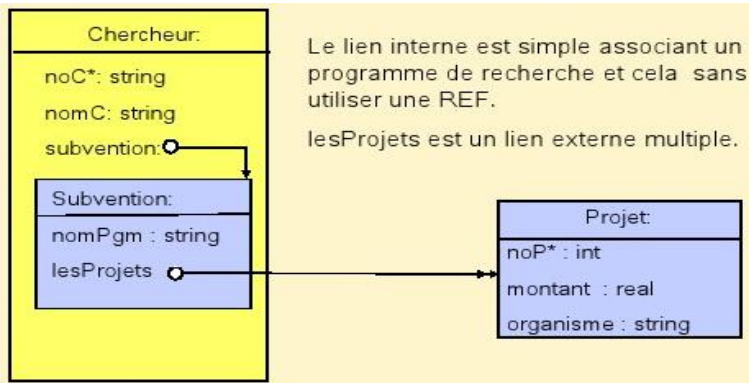
39

## Modèle avec un lien interne simple à plusieurs niveaux



40

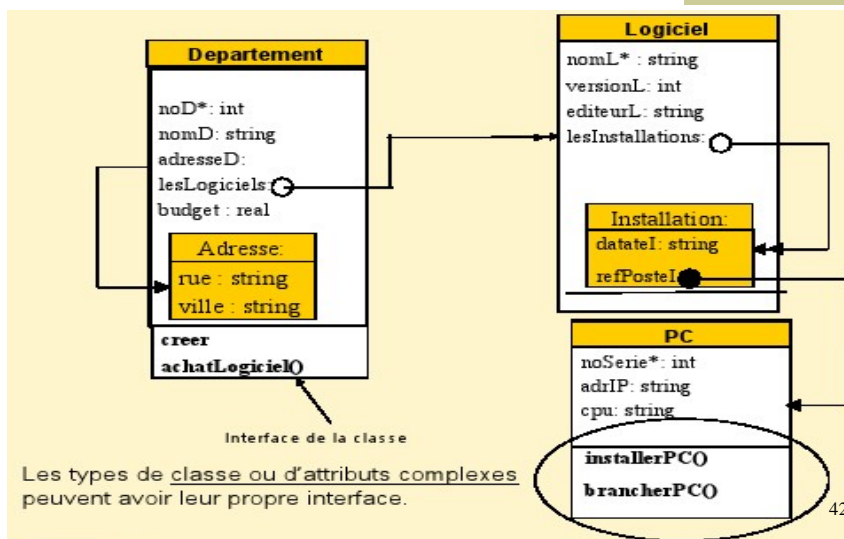
## Combinaison de lien interne simple et externe multiple



- ✓ La suppression d'une instance de Chercheur entraîne la suppression de sa subvention mais pas des projets subventionnés.

41

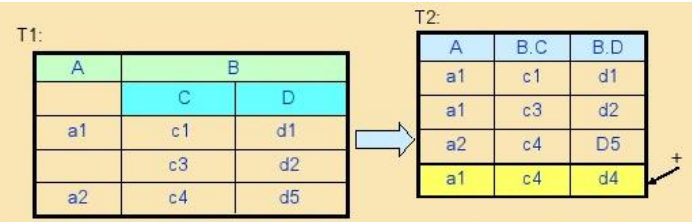
## Ajout de l'interface dans un Mnav



42

# Unnesting d'une table en Oracle : Table()

- SQL-DML manipule des valeurs de tables de **premier niveau**.
- Une table imbriquée peu importe le niveau doit être logiquement **aplatie (Unnested)** pour avoir accès aux valeurs avec SQL. Cette opération sous-tend une jointure entre la table et la sous-table!



- Insert intoT2 ( A, B.C, B.D) values (a1, c4, d4); **OK si table préalablement aplatie**
- Insert intoT1 ( A, B.C, B.D) values (a1, c4, d4); **NON, car table imbriquée**
- Select B fromT1; **OK pour certaines sélections (mais fournit des objets-colonnes)**

43

## Unnesting et le Stockage physique d'une table imbriquée

T1

A	B	
	C	D
a1	c1	d1
	c3	d2
a2	c4	d5

Pour faire le unnesting: faire une jointure entre la table mère et la sous-table

```

Select  x.*, w.*
From    T1 x, Table (x.B) w;

Select  z.A, z.D
From    (Select  x.*, w.*
From    T1 x, Table (x.B) w) z;

```

Ta

A	ref-Interne
a1	p0001
a2	p0002

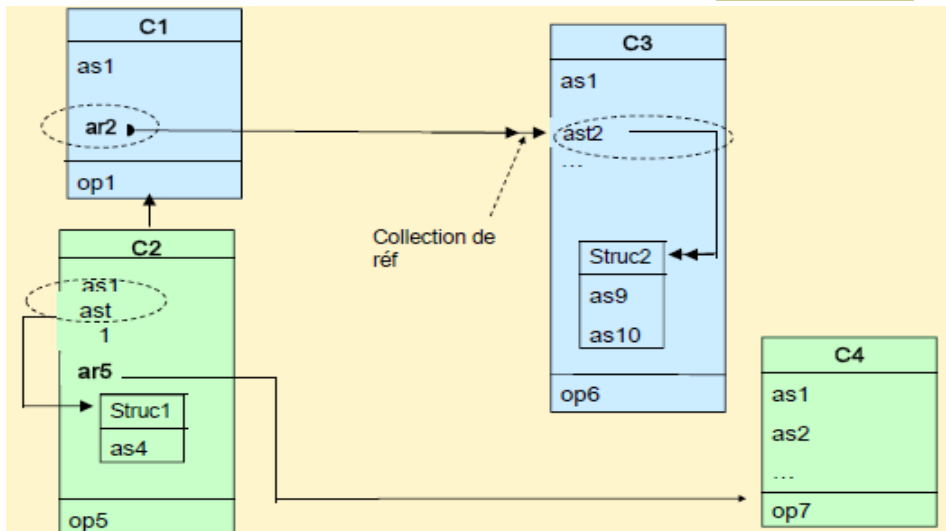
Tb

C	D	ref-interme
c1	d1	p0001
c3	d2	p0001
c4	d5	p0002

T1 := Ta |x| Tb

44

## Résumé de la notation graphique du du Modèle navigationnel (Soutou. C)



## Formalisme : attributs et références

- Les types possibles pour les attributs sont les suivants :
  - **as** : un attribut simple ayant un type atomique ;
  - **ac** : attribut composé = structure struc1 du genre tuple ;
  - **ar** : attribut de type référence vers une classe ;
  - **Struc** est une classe interne avec structure composée d'un attribut simple et d'une référence vers une classe ;
  - **Un attribut de collection** est à associé à un lien interne ou externe
  - **Un attribut de REF** représente un lien externe

## Passage du diagramme de classe UML au modèle navigationnel (Mnav)

- L'implantation du DC-UML en Mnav peut se faire de diverses façons. Le critère de validité du modèle cible est que le modèle navigationnel obtenu ne soit pas en contradiction avec la sémantique du DC-UML et de ses contraintes.
- Les contraintes qui ne sont pas renforcées par le Mnav le seront par trigger ou par SQL dans une méthode.

47

## Du DC-UML vers le navigationnel : règles de passage

- Transformation d'une **classe UML**:
  - Chaque classe du DC-UML, sauf les classes-associations, est transformée en une classe externe du Mnav.
  - Chaque attribut composé UML devient une classe interne avec un lien interne simple.
  - Chaque attribut d'ensemble est rendu par un lien interne multiple ou un lien externe multiple.

48



## Du DC-UML vers le navigationnel : règles de passage

### ■ Identifiant de classe du Mnav:

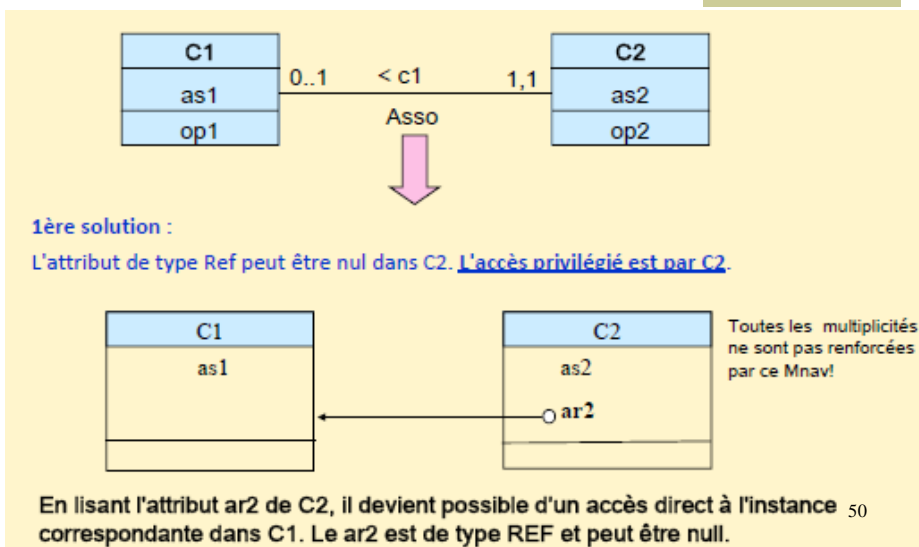
- Chaque classe du Mnav (sauf les sous-classes dans l'arbre d'héritage) et la classe d'association, peut avoir un identifiant de valeur. Souvent pratique pour la recherche.
- Si aucun attribut ne convient comme clé, on peut en ajouter une clé, si possible significative. Ce n'est pas une exigence du modèle.

### ■ Transformation des associations :

- Plusieurs possibilités selon le genre d'association: 1-1, 1-\* et \*-\*.
- Chacune étant spécifiée de préférence par le DDL-SQL du système utilisé.

49

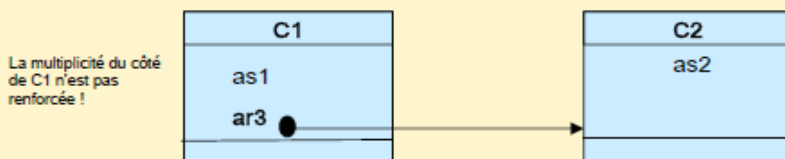
## Transformation de l'association 0..1 – 1..1



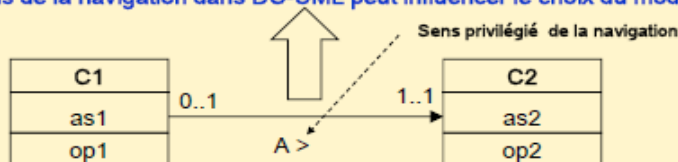
50

## Transformation de l'association 0..1 – 1..1 (Suite)

- 2ème solution : accès via C1 (si le sens de l'association est modifié)  
L'attribut de référence ar3 ne peut pas être nul.



- Sens de la navigation dans DC-UML peut influencer le choix du modèle :

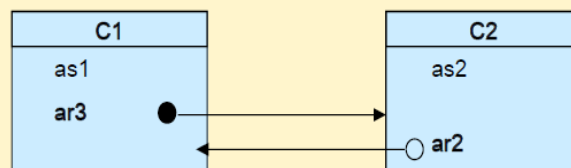


Le sens de la navigation est indiqué dans le DC-UML suggérant la transformation appropriée. Si le **sens de la navigation est absent** dans l'association : assumer alors les 2 sens sont équi probables

51

## Transformation de l'association 0..1 – 1..1 (Suite)

- 3e solution pour les 2 sens de parcours: 2 liens simples



ar3 est un attribut de C1 du genre REF monovalué non nul qui réfère à la structure de C2 (donc aux objets de la classe C2).

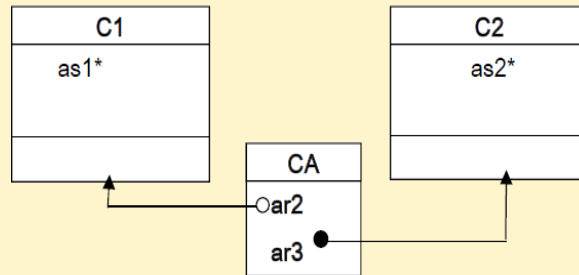
ar2 : est un attribut du genre Ref qui peut être nul et qui pointe sur un objet de C1.

- La suppression exige cependant celle de 2 références.
- Ce Mnav exprime le renforcement des 2 contraintes.

52

## Transformation de l'association 0..1 – 1..1 (Suite)

4e solution ([dite universelle](#)): utilisation d'une 3e classe CA et 2 liens mono et externe. Toutes les multiplicités sont renforcées.

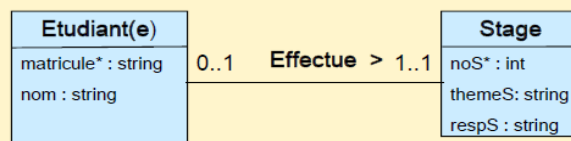


➤ Le modèle peut évoluer plus facilement vers un autre type d'association sans pénaliser l'interrogation et la manipulation de la base. Cette solution est valable pour tous les types d'association.

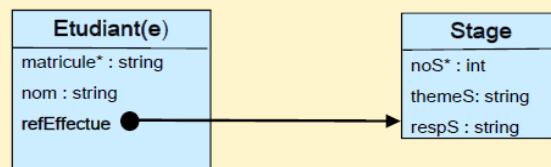
53

## Transformation de l'association 0..1 – 1..1 et interrogation

➤ Réalisation des stages étudiants



Solution:



- 1- Quels sont les étudiants sans stage?
- 2- Quels sont les stages sans titulaire?

Le calcul de la réponse à la question 2 a une formulation plus complexe faisant appel à une différence d'ensemble.

54

## Transformation de l'association 0..1 – 1..1 et interrogation

1- Aucun puisque la contrainte de multiplicité impose que tout étudiant dans la base ait un stage.

2- Pour calculer la réponse il faudra faire une différence:

```
Select s.noS -- tous les stages dans la base
```

```
From Stage s
```

- (Minus)

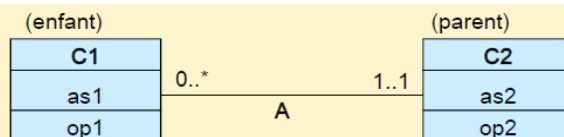
```
Select e.refStage.noS
```

```
From Etudiant e
```

```
Where e.refEffectue is not null;
```

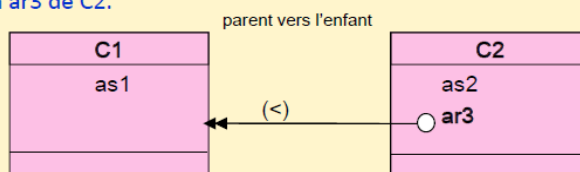
55

## Association de multiplicité 0..\* – 1..1



Aucun sens de navigation formulé dans le DC-UML : les 2 probables

- 1ère solution : un lien multiple. La multiplicité minimale 0 figure à la source du lien ar3 de C2.

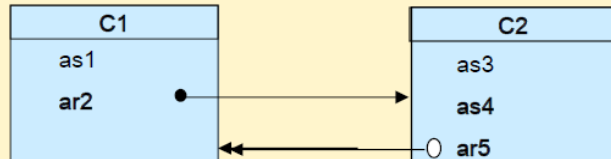


L'accès par C2 est privilégié (<), mais la suppression d'un objet C2 n'entraîne pas celle de C1. La suppression de C1 devra être bloquée parce qu'elle peut engendrer une incohérence avec un enfant dans la base sans un parent.

56

## Association de multiplicité 0..\* – 1..1 (Suite)

➤ 2e solution : un lien multiple et un lien mono inverse.



ar5 est un attribut d'ensemble (éventuellement vide) de références sur C1.

ar2 est un attribut de type Ref qui référence C2.

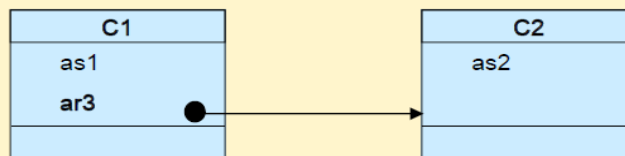
La suppression d'un objet de C2 doit éventuellement entraîner celle d'un objet de C1. La suppression d'un objet C1 est possible.

\*\* Plus il y a de liens externes, plus la mise à jour sera généralement complexe.

57

## Association de multiplicité 0..\* – 1..1 (Suite)

➤ 3e solution \*\*\* : un lien monovalué non nul vers le parent C2



Chaque instance de C1 est liée à une instance de C2 par l'attribut ar3 de type Ref et est obligatoire.

La suppression de C1 se limite à cette opération.

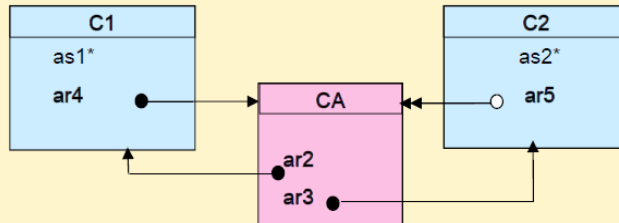
La suppression de C2 doit être bloquée par une contrainte si elle est référée par un objet de C1. La source du lien ar3 doit avoir obligatoirement une valeur de Ref valide.

La suppression de C2 devrait éventuellement entraîner aussi celle de C1 qui s'y réfère et cela au cours d'une même transaction.

58

## Association de multiplicité 0..\* – 1..1 (Suite)

- 4e solution (universelle): utilisation d'une troisième classe. Cette classe porte le nom de l'association et contient deux liens simples pour implémenter l'association.



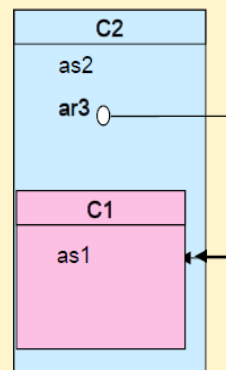
Sans les liens vers CA, la navigation (donc l'accessibilité) ne serait possible qu'à partir de CA. Cette solution fournit un Mnav plus lourd à gérer! La suppression de C2 peut impliquer plusieurs changements: suppression dans CA et dans C1.

Que serait le modèle si la classe externe CA avait aussi les 2 clés comme attributs?

59

## Association de multiplicité 0..\* – 1..1 avec un lien interne multiple ??

Autre solution non équivalente au UML-DC



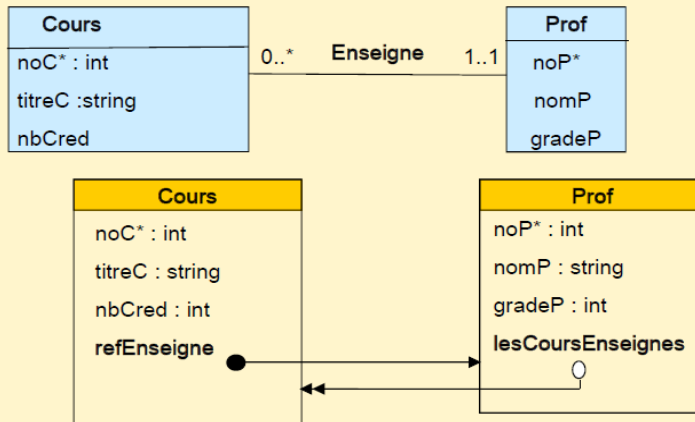
Le lien interne et multiple pour ar3 sous-tend un ensemble formé avec des objets-colonnes de type C1 imbriqués.

\*\* L'accès est privilégié par C2 pour trouver les C1. La suppression de C2 entraîne celle de C1. La suppression de C1 n'entraîne pas celle de C2.

60

## Exercice : Association 0..\* – 1..1

### ➤ Exemple : Cours-Prof



Quelles sont les contraintes du Mnav à l'ajout et à la suppression d'un cours?

61

## Exercice : Association 0..\* – 1..1

À l'ajout d'un cours:

Le cours doit référer à un prof par une référence obligatoire.

Éventuellement, un prof doit dispenser ce nouveau cours.

À la suppression d'un cours:

Le cours est supprimé

Il doit être aussi supprimé de la liste du prof concerné.

À l'ajout et à la suppression d'un prof. ?

Ajout: ajout simple.

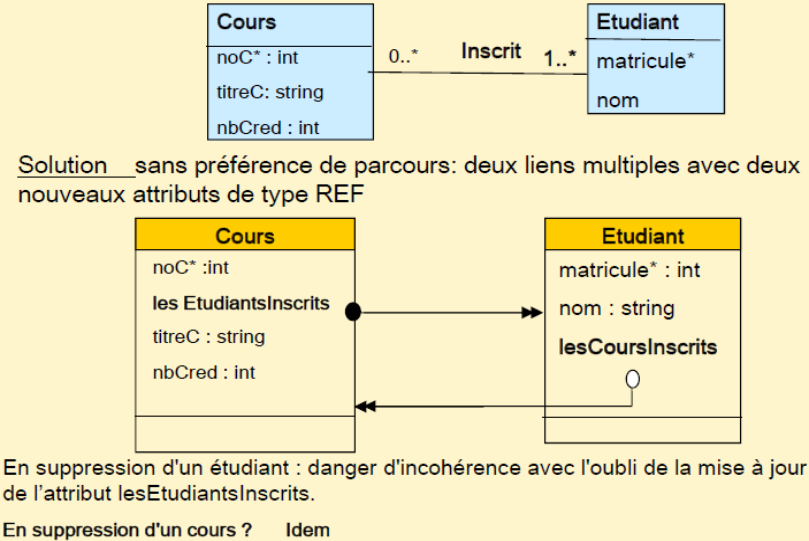
Suppression: Dans une même transaction:

- suppression de tous les cours enseignés par ce prof.
- suppression du prof

Ces contraintes devront être implémentées soit par trigger soit par des clauses SQL intégrées dans une méthode.

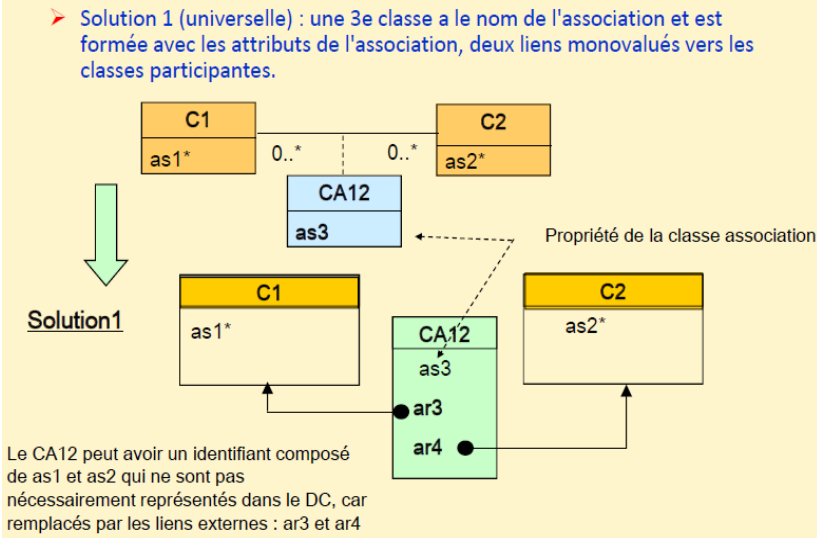
62

# Association 0..\* – 1..\* sans propriétés propres



63

# Association 0..\* – 0..\* avec propriétés (attributs)

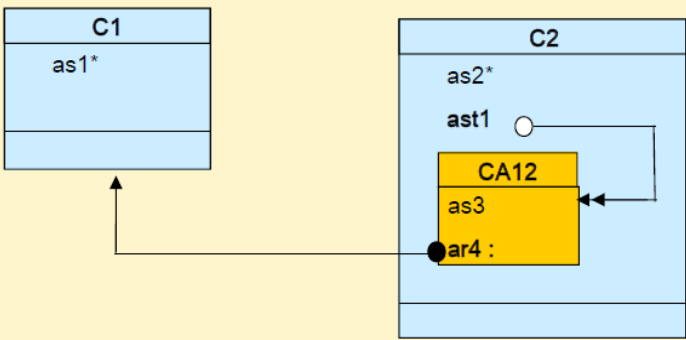


64



# Association 0..\* – 0..\* avec propriétés (suite)

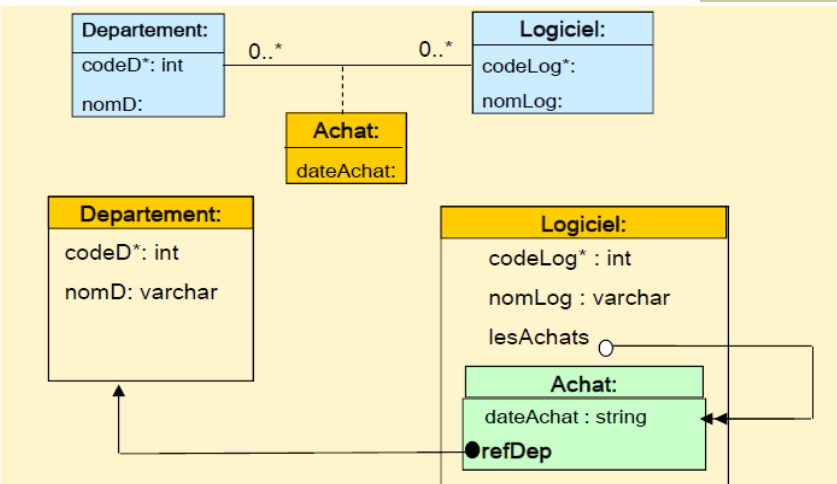
➤ Solution 2: une structure interne avec un lien multiple et un lien externe simple.



La suppression d'une instance de C1 sous-tend la mise à jour de ar4 à NULL. Or comme c'est interdit par le diagramme, il faut donc aussi supprimer l'instance de CA12. Qu'en est-il de l'ajout de C1?

65

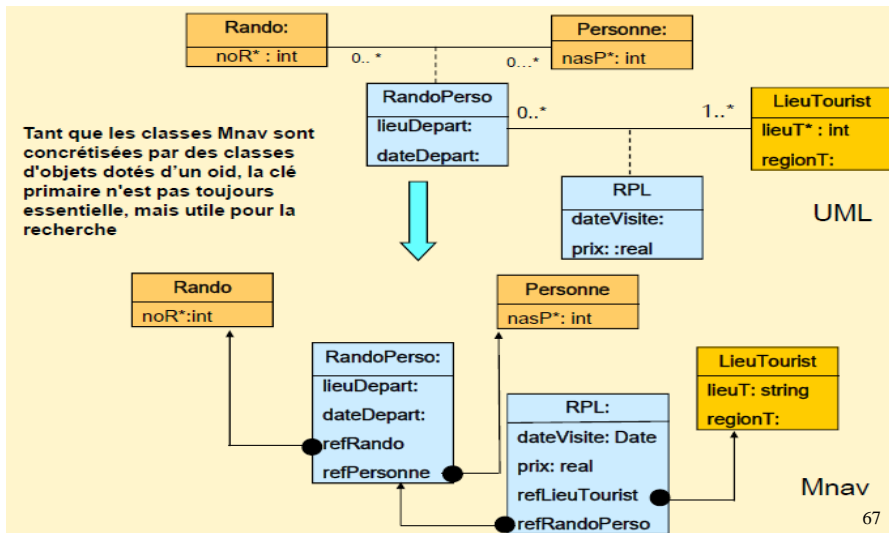
## Exemple : Association 0..\* – 0..\*



Les achats de logiciels distincts faits par un département sont représentés par la collection interne dont les éléments ont la structure achat\_t

66

## Exemple d'une classe association participant à une autre association (*solution universelle*)



## Quelques interrogations avec parcours par référence

Pour ne pas perdre de vue la réalité de l'exploitation subséquente du modèle, voici quelques clauses SQL pour interroger et mettre à jour cette base.

- Quelles sont les régions où est parti le randonneur 3456?

```
Select rp.refRando.noR
From RandoPerso rp
Where rp.refPersonne.nasP = 3456;
```

- Quels sont les lieux touristiques visités par le randonneur 3456?

```
Select t.refLieuTourist.lieuT
From RPL t
Where t.refRandoPerso.refPersonne.nasP = 3456;
```

68

## Mise à jour avec parcours par les références

- Le randonneur 3456 qui a quitté le 15-10-2010 n'a pas parcouru la région 5 mais la région 6. Une mise à jour s'impose.

```
Update RandoPerso rp set rp.refRando = (select Ref® From Rando r
                                         Where r.nor = 6)
Where rp.dateDepart = to_date(2010-10-15, 'YY-MM-DD')
and rp.refPersonne.nasP = 3456
and rp.refRando.noR = 5 ;
```

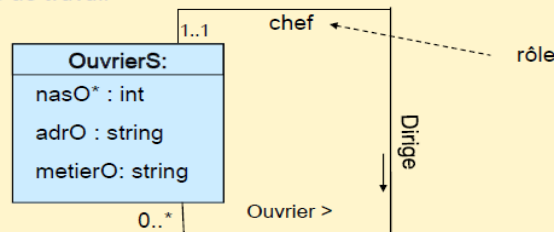
### Remarque:

Dans ce Mnav, une randonnée de personne ne peut être ajoutée s'il n'y a pas eu la visite d'un lieu touristique. Donc l'ajout d'une randonnée doit nécessairement inscrire la visite d'un lieu touristique. Pour renforcer la contrainte.

69

## Association réflexive 1..1 – 0..\* Modèle ouvrier spécialisé

- Exemple : Poste de travail



**Solution 1 :** 1 lien externe : mono

Quels sont les ouvriers dont le chef est 25?

```
Select o.*
From OuvrierS o
Where o.refOuvrierSChef.nas= 25;
```

Quel est le nas du chef de l'ouvrier 50?

```
Select o.refOuvrierSChef.nasO From Ouvrier o Where o.nasO = 50;
```

70

# Association réflexive 1..1 – 0..\* : les 2 parcours privilégiés

➤ **Solution 2 :** les 2 accès ou parcours privilégiés

Mnav : 1 lien externe simple  
1 lien externe multiple

Quels sont les ouvriers dont le chef est 25?

```
Select o.nasO
From OuvrierS o
Where o.refChef .nasO = 25;
```

71

## Le choix à privilégier pour les associations

Pour l'association	Modèle navigationnel
1..1 – 1..1	Solution 1 ou 2 (un lien monovalué)
1..1 – 0..* (sans préférence de navigation)	Solution 3 (lien monovalué de enfant vers le parent)
0..* – 0..* <u>sans</u> propriétés	Solution 2 (2 liens mono + 2 liens multi + CA)
0..* – 0..* avec propriétés	Solution 1 (2 liens mono + CA)
n-aire sans propriétés	Solution 2 (structures dans une classe)
n-aire avec propriétés	Solution 1 (n+1) ème classe)

Les contraintes non exprimées dans le Mnav devront être renforcées par les méthodes, les triggers et les clauses DDL propres au DDL objet.

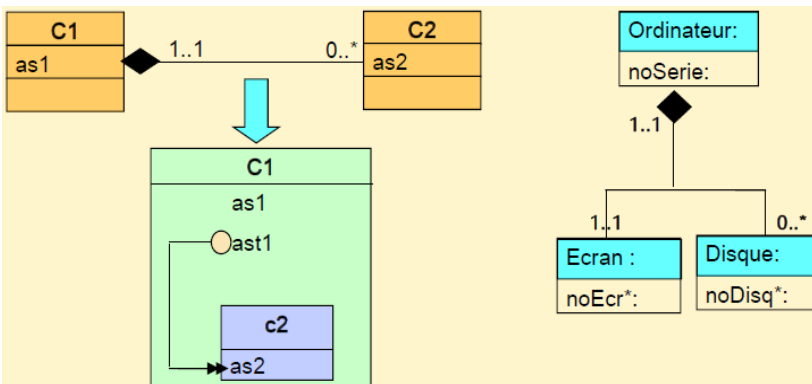
72

## Agrégation : composition et partage

- Agrégation de composition et de partage : en premier, ce sont des associations dotées de contraintes supplémentaires au regard de la suppression.
  - Composition (forte) : la suppression de l'agrégat entraîne celle des instances agrégées et chaque instance du composant est en association qu'avec une seule instance de l'agrégat.
  - Faible (ou partagée) : la suppression de l'agrégat n'entraîne pas celle d'un composant qui peut être éventuellement en association avec une autre instance de l'agrégat ou avec une entité d'une autre classe.
- La représentation d'une agrégation :
  - Soit par des structures dans la classe composée (agrégat) avec des liens simples ou multiples;
  - Soit par une classe composée reliée aux différentes classes (externes) avec des liens simples ou multiples externes.

73

## Agrégation de composition forte



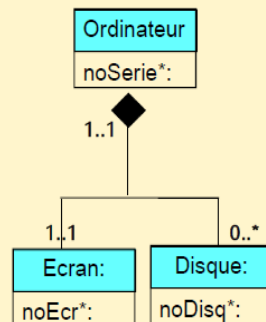
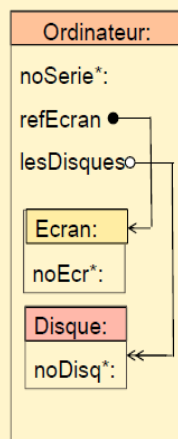
La suppression d'une instance de C1 entraîne celle des instances de C2 associées. La création de C2 doit se faire en même temps que celle de C1. Conforme à la sémantique de l'agrégation forte.

Quel est le Mnav pour l'agrégation de droite?

74

# Mnav pour une agrégation multiple

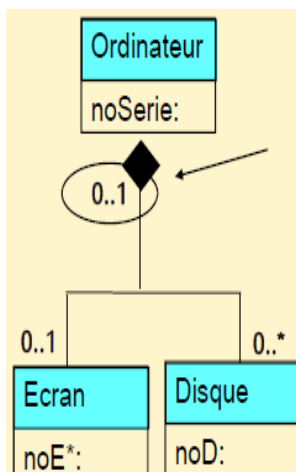
Solution avec 2 liens internes :



La modélisation avec une agrégation forte sous-tend la suppression dans la BD de l'écran et des disques qui le composent, lorsque l'ordi est supprimé.

75

## Composition forte avec 0..1



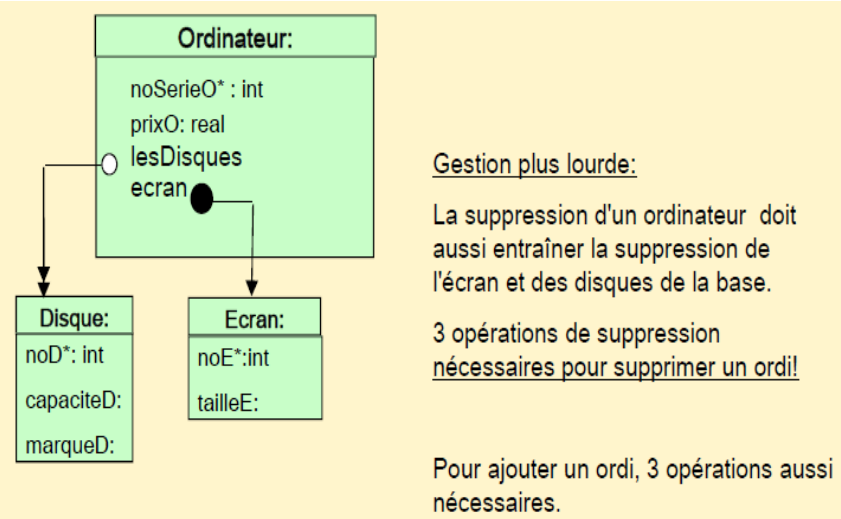
En ajout: un écran est inscrit dans la BD sans obligation d'être associé à un ordinateur. Idem pour le disque.

Un ordi est fait de 0 ou 1 écran et de 0 ou plusieurs disques.

En suppression: par contre, l'ordinateur implique celle de l'écran et des disques.

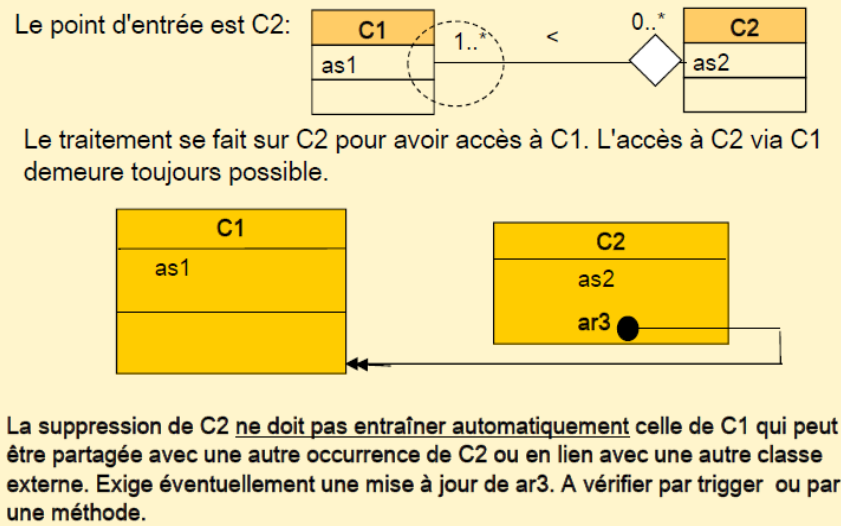
76

# Autre implémentation de l'agrégation de la composition forte



77

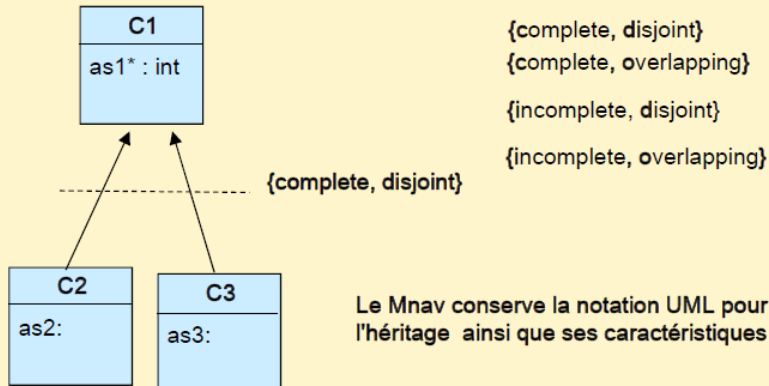
# Agrégation partagée pour une association 1..\* - 0..\* orientée



78

# Héritage dans le Mnav

L'héritage se formalise avec la même sémantique que celle de UML; il est spécifié directement avec le schéma objet de Oracle. L'héritage aura des propriétés : exclusivité (d), overlapping (o), total (c) et partiel (i).



79

## En conclusion

Le Mnav a pour but de visualiser les structures possibles conduisant à l'objet et en évitant autant que possible les jointures.

Il permet aussi de renforcer par sa structure certaines contraintes du modèle, mais pas toutes. Les autres seront renforcées par les méthodes et les triggers.

Les contraintes de multiplicité autres que les classiques, exemple de (0..5), restent à être renforcées soit par des triggers classiques soit par la logique même des méthodes qui encapsulent les données.

Ces méthodes peuvent aussi s'appeler l'une l'autre afin de profiter de leur logique propre par le partage.

Le Mnav met aussi en évidence la notion de type d'ensemble sans toutefois préciser s'il s'agit d'un ensemble ordonné, le `varry()` ou non le `set` ou la `sous-table`.

80