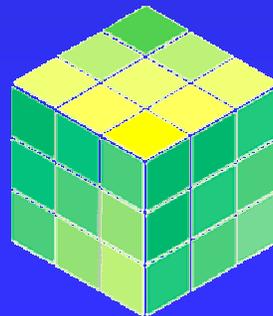


# Création d'un langage d'action pour un logiciel MDA

Soutenance de DRT GEII

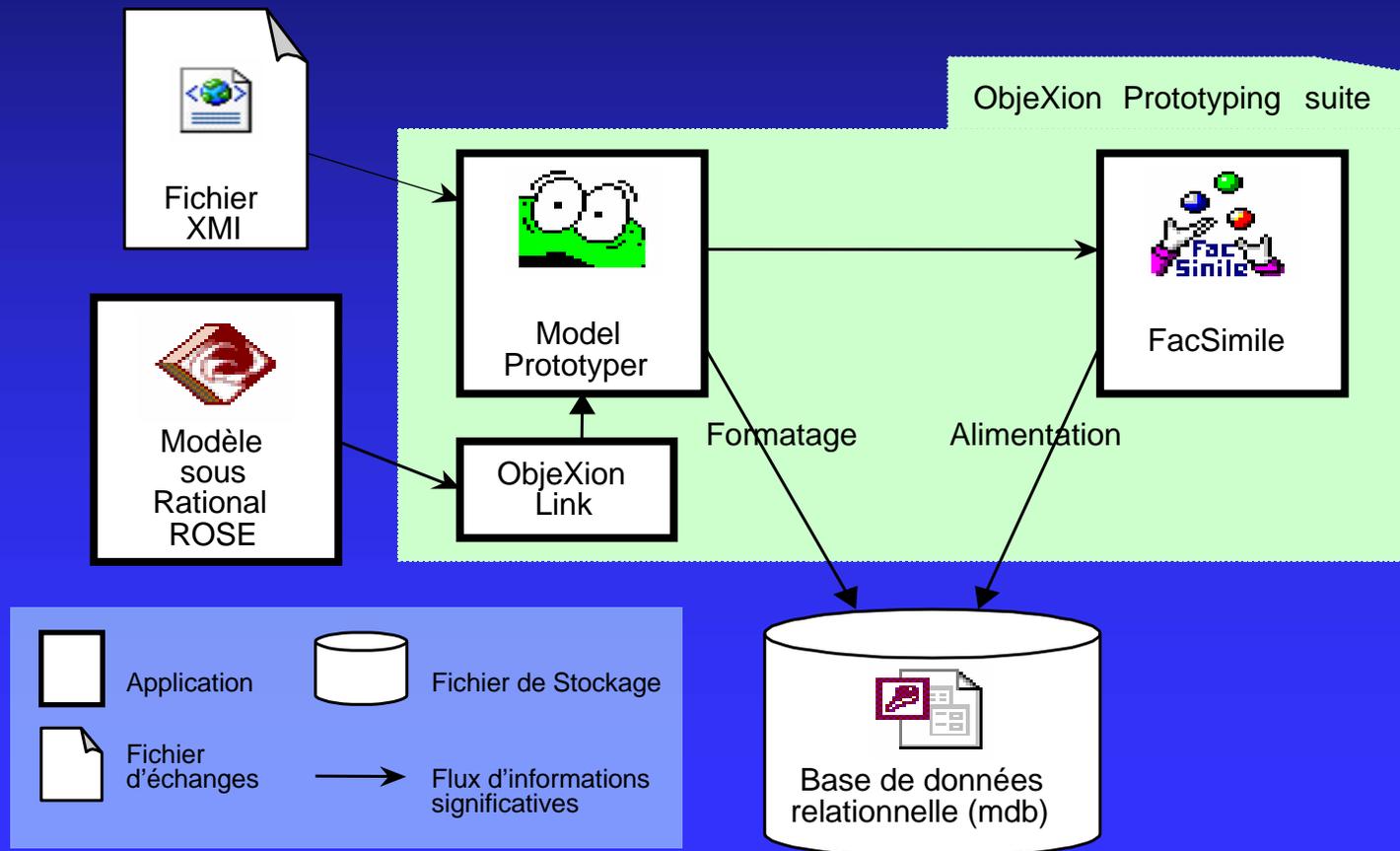
Proposé par l'UHA (Essaim)

Réalisé au sein d'Objexion Software

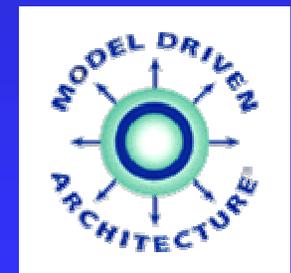
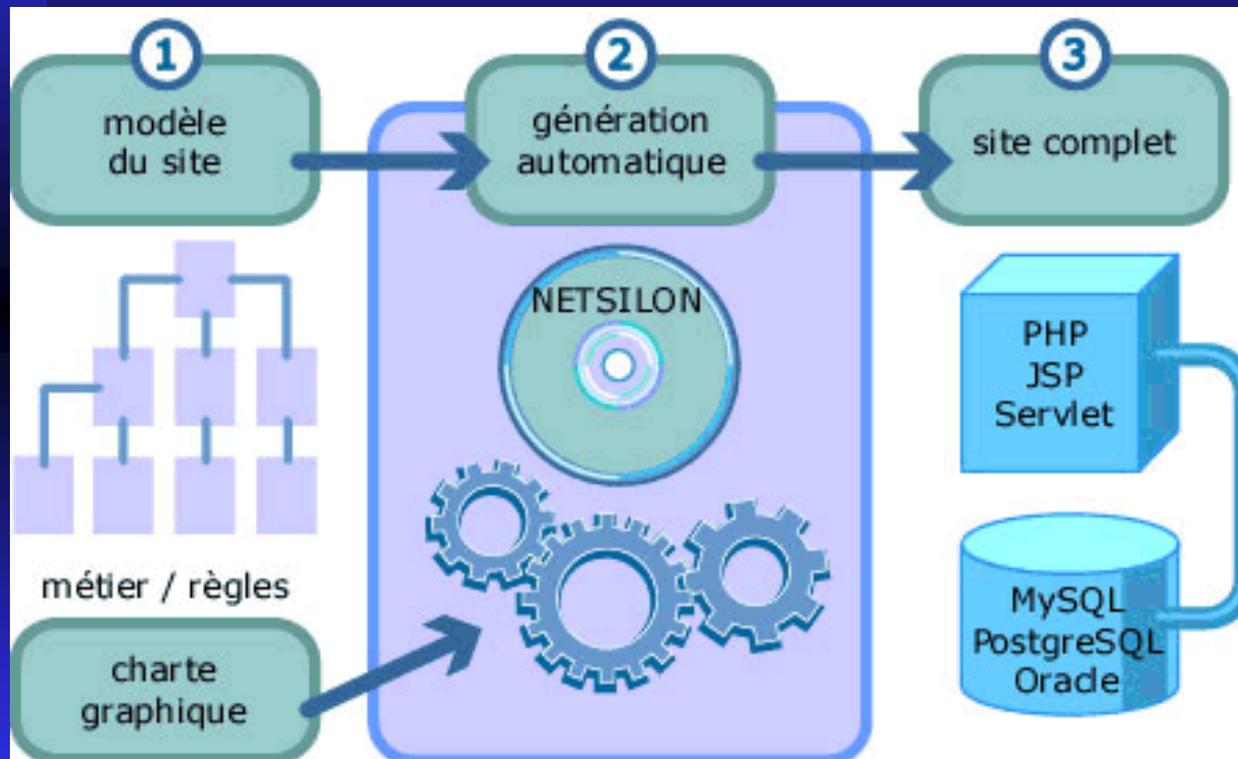


Première partie (résumé):

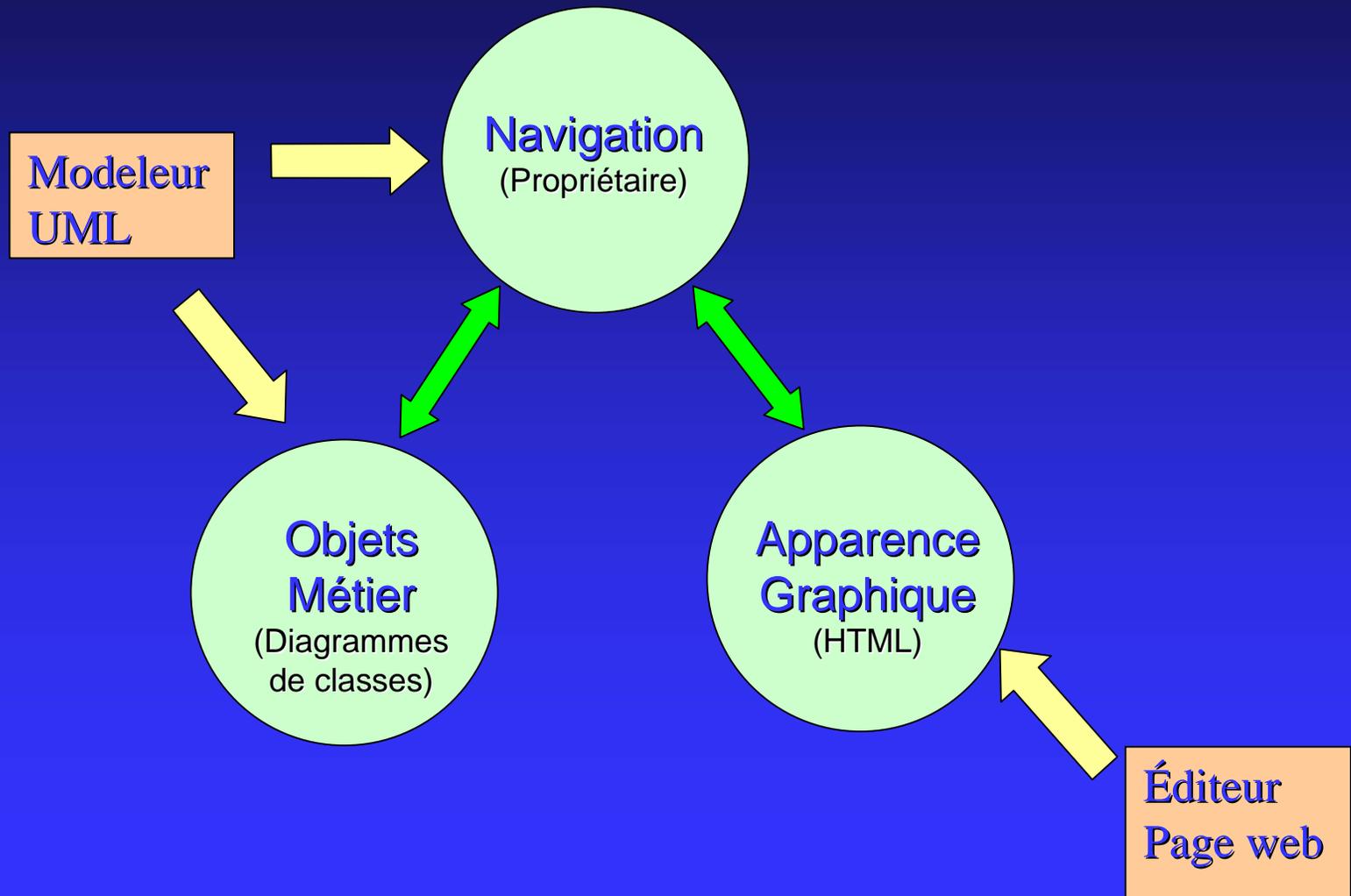
# Développement d'un Interpréteur OCL pour une Machine Virtuelle UML



# Netsilon: présentation



# Netsilon: modélisation



# Création d'un langage d'action pour un logiciel MDA

- Xion
- Optimisation SQL
- Éditeur de code
- Gestion des objets métier

# Xion: besoins

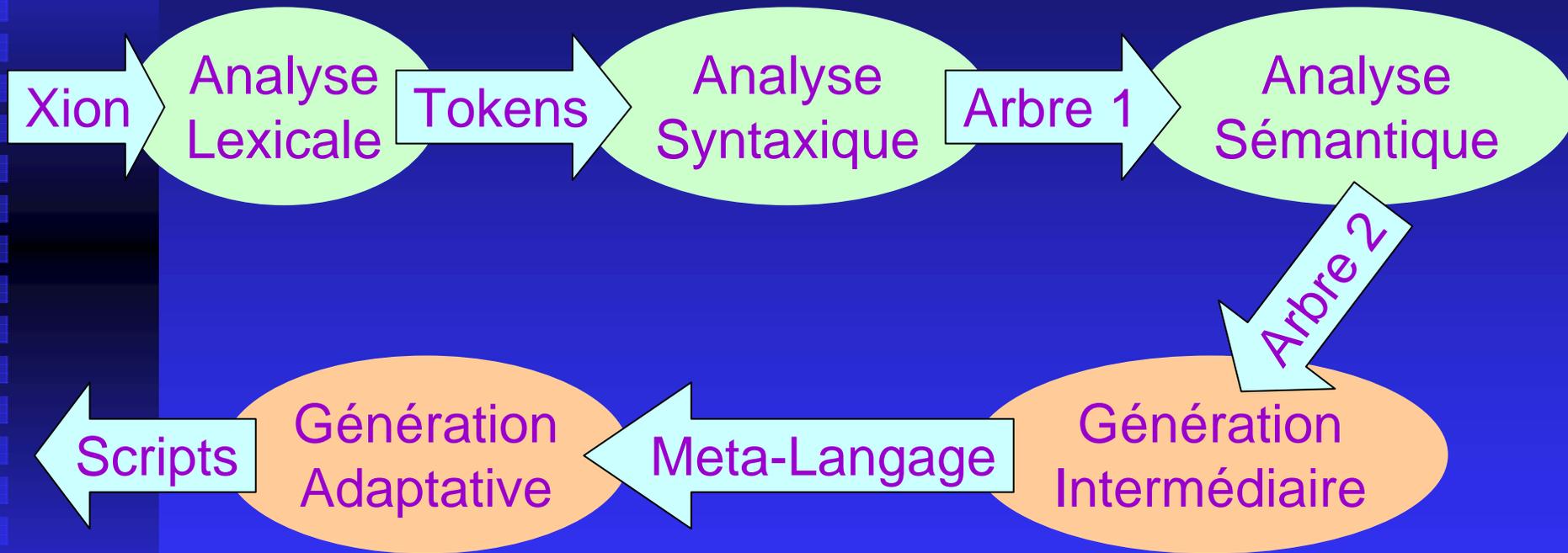
- Description d'opérations à effet de bord
- Navigation dans le modèle métier
- Traduisible en PHP, JSP, Servlet

*Rien n'y répond ! (C++, Java, SDL, OCL,...)*

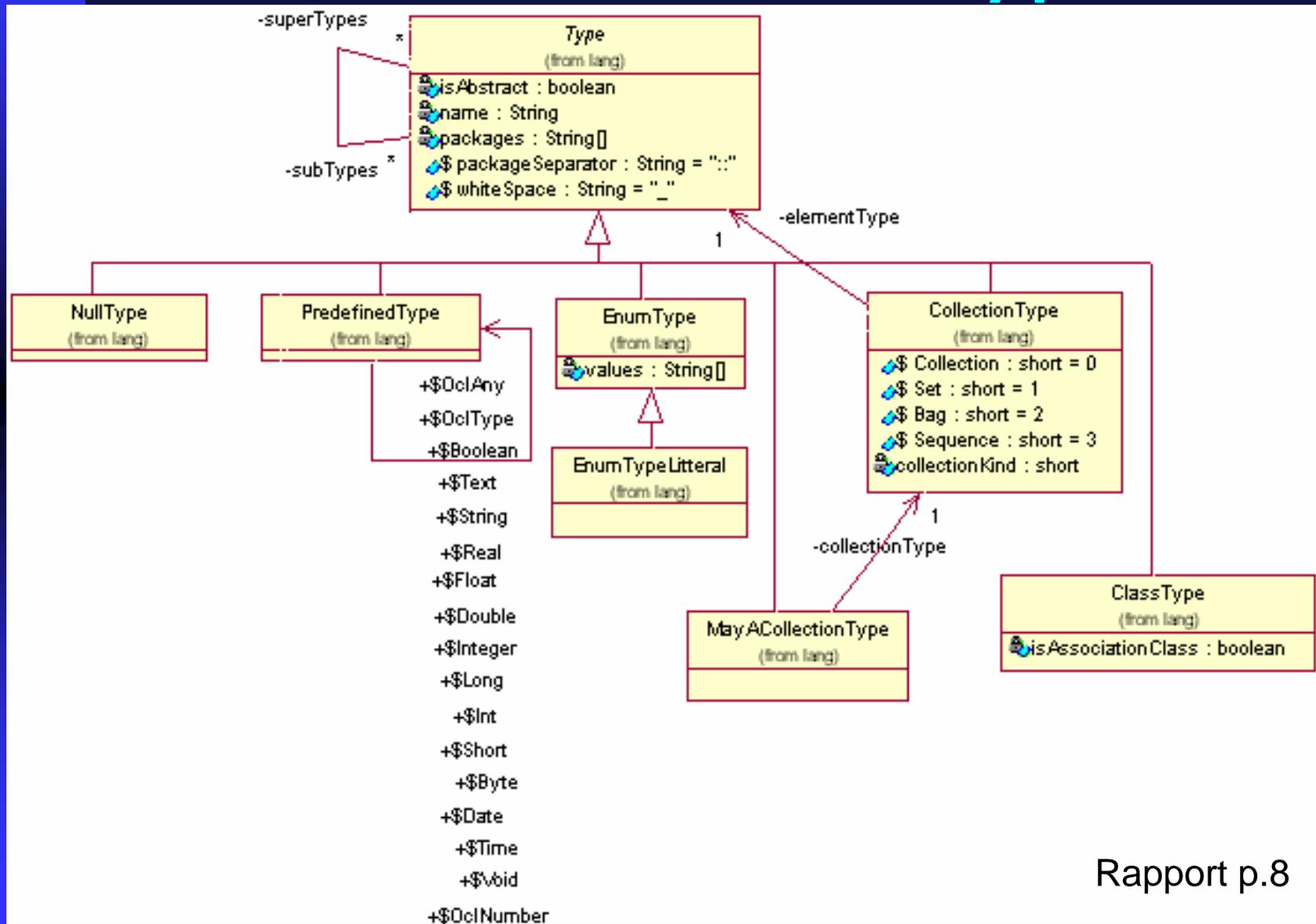
**Solution adoptée:**

**« mélanger » Java et OCL**

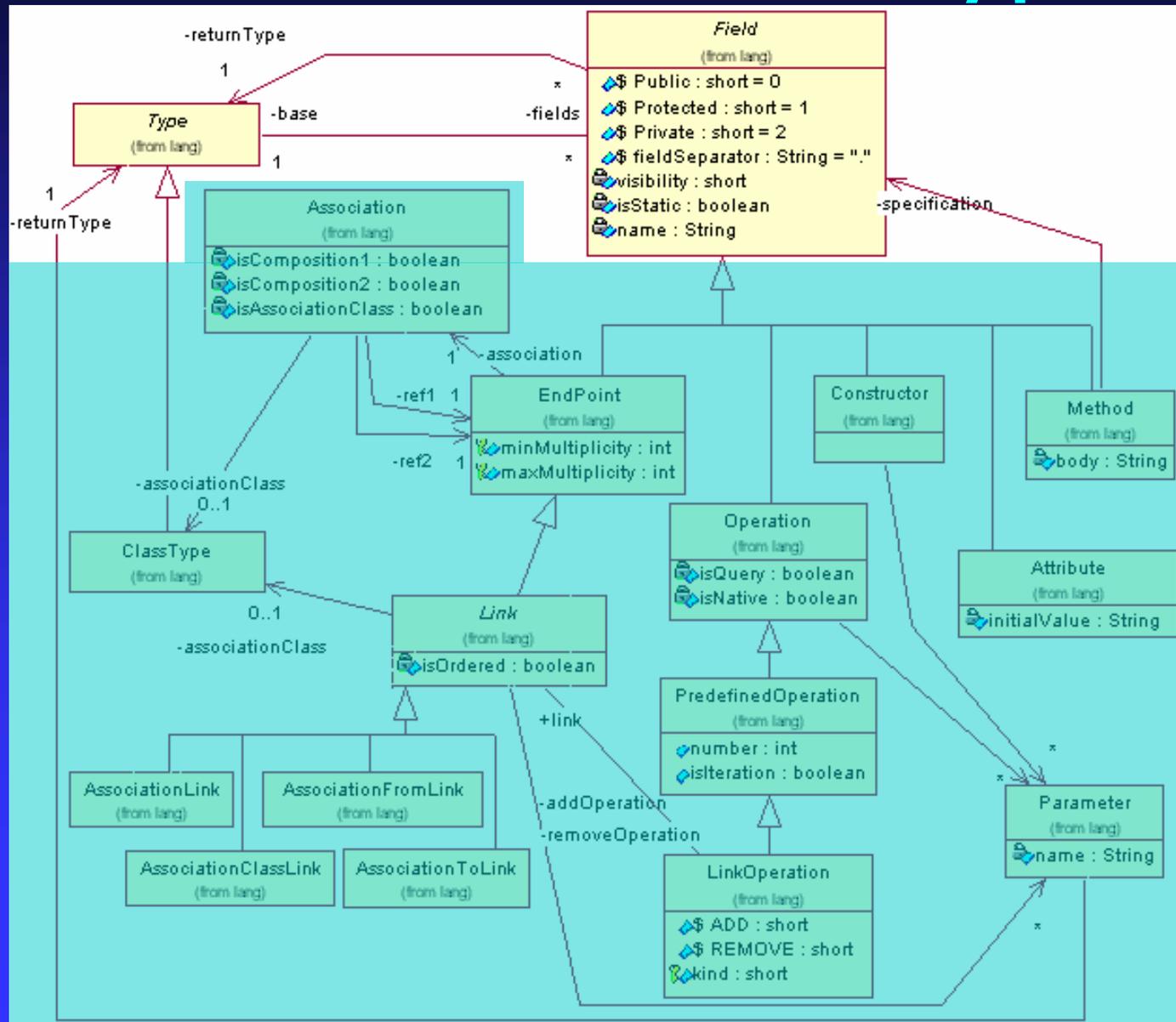
# Xion: réalisation



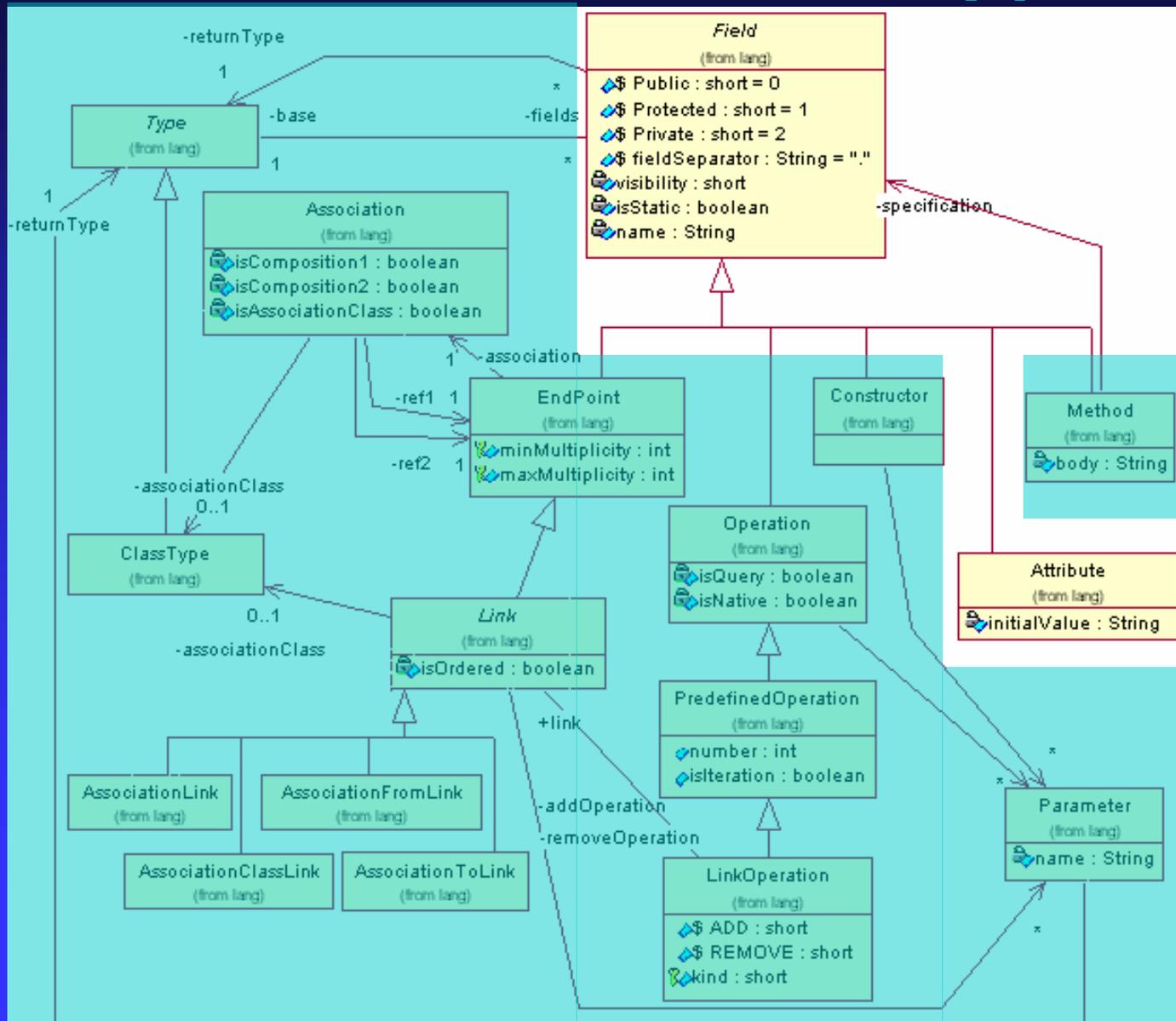
# Xion: contrôle des types



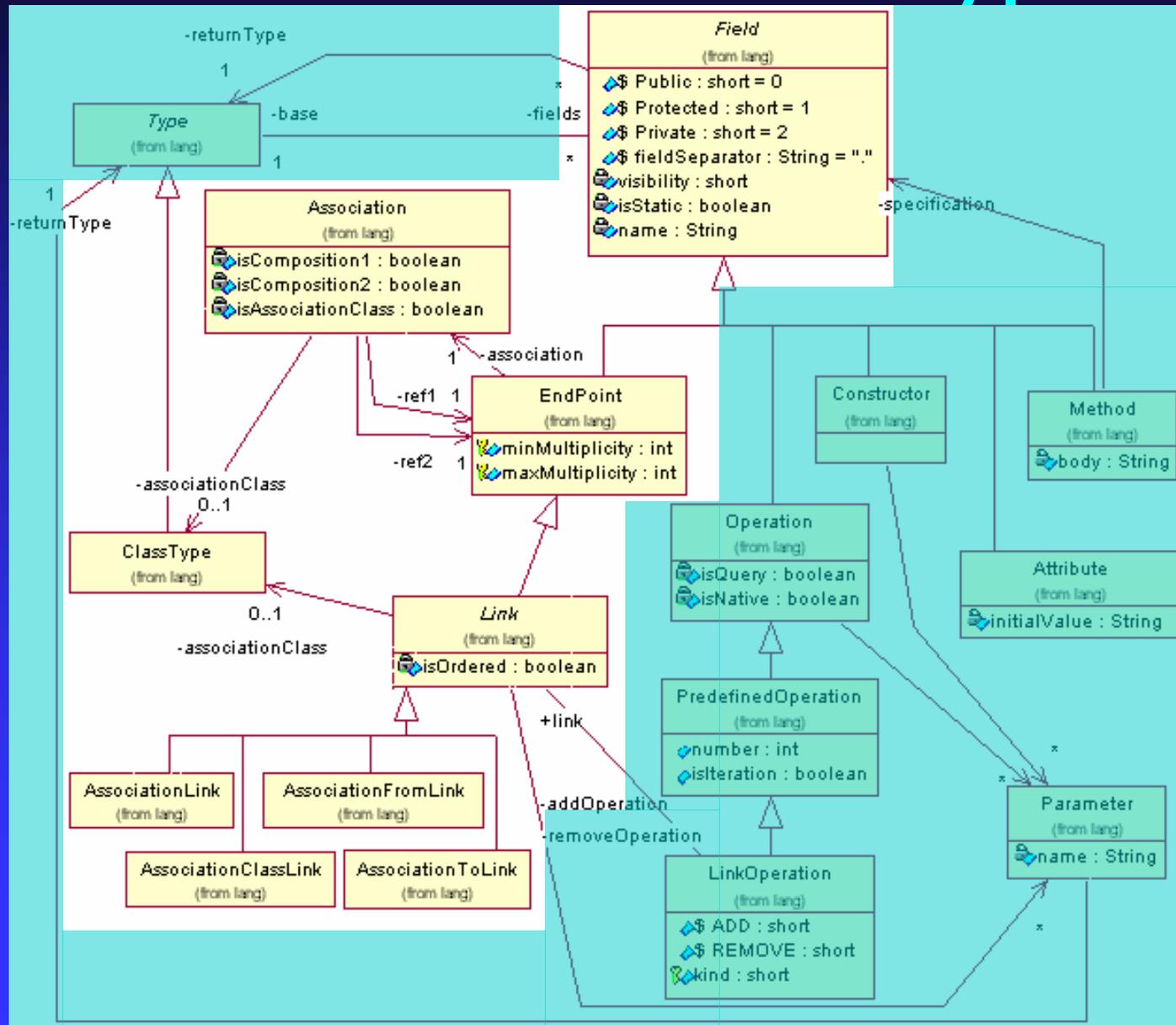
# Xion: contrôle des types



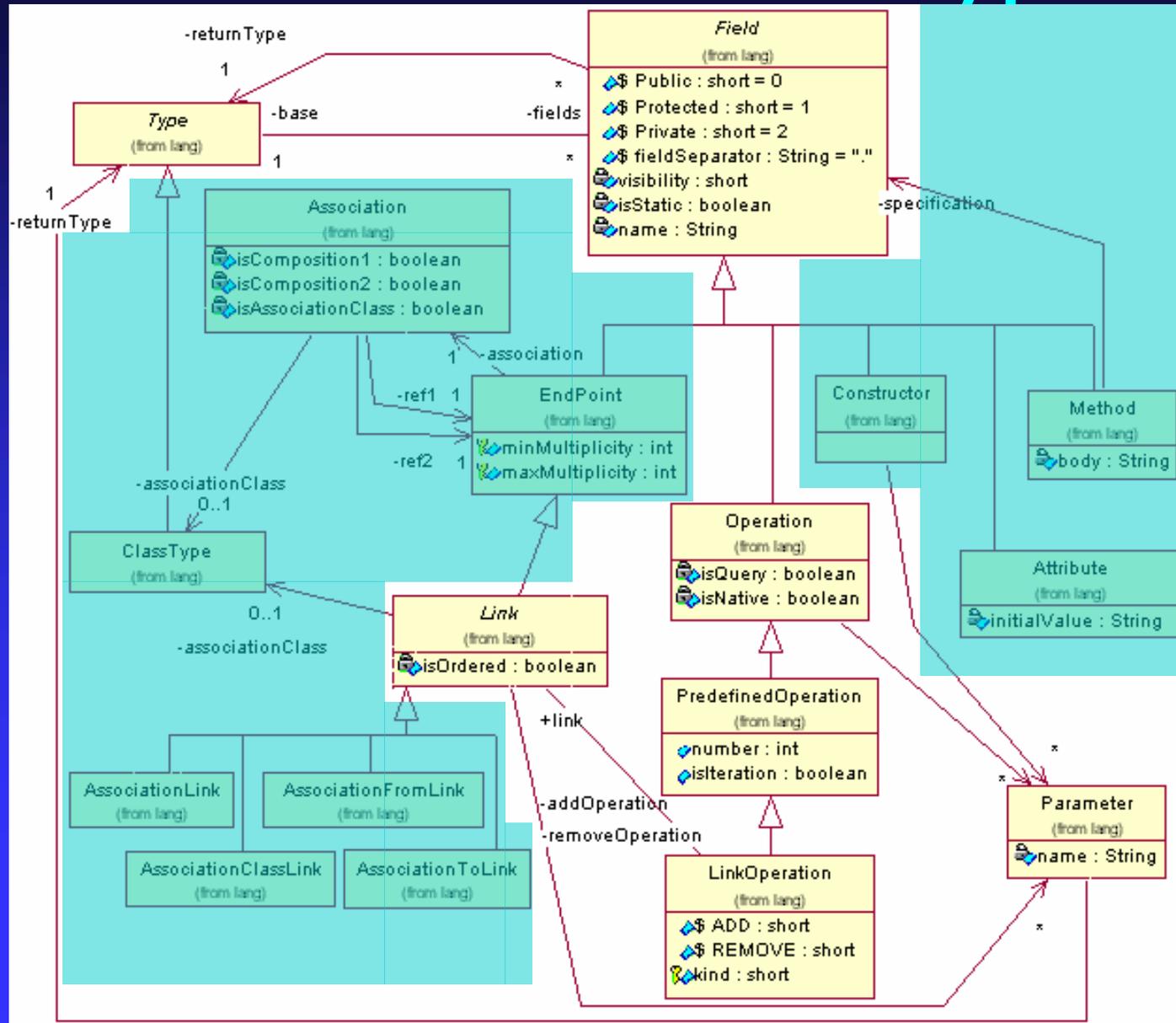
# Xion: contrôle des types



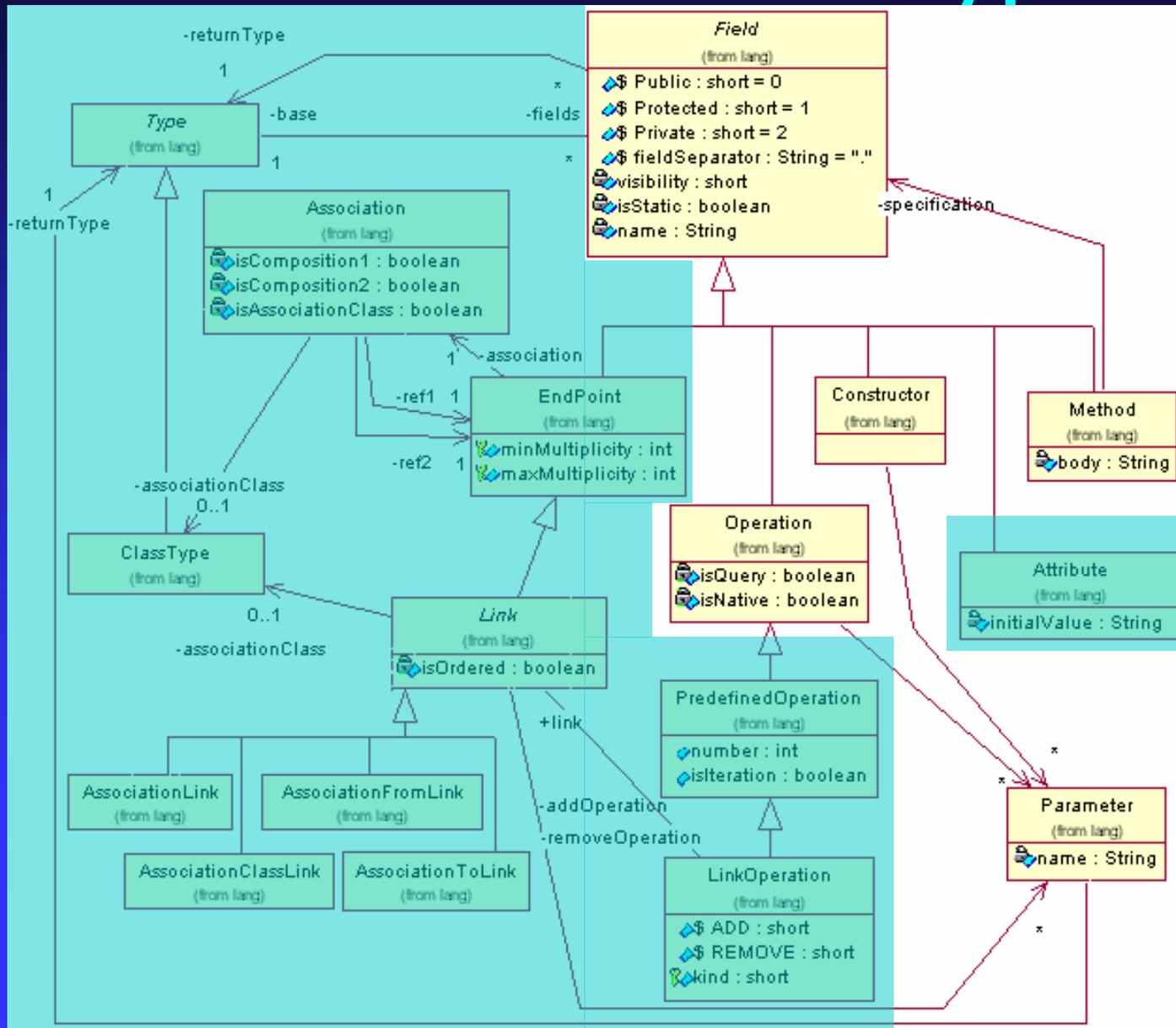
# Xion: contrôle des types



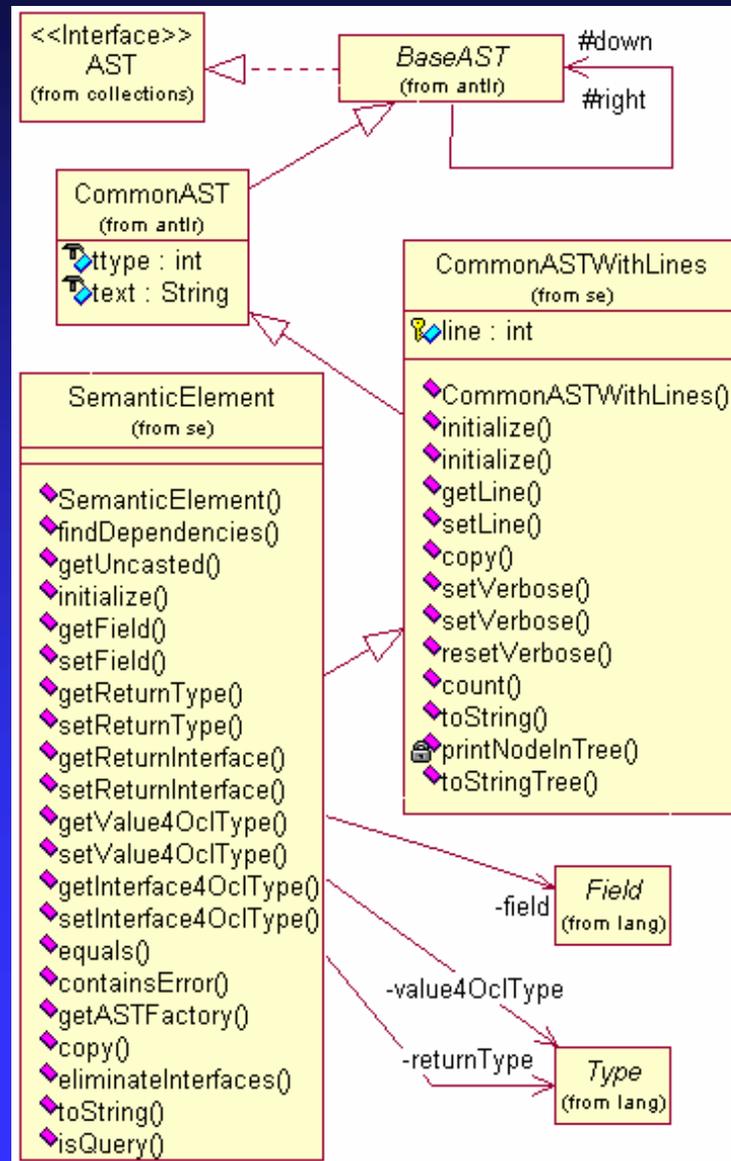
# Xion: contrôle des types



# Xion: contrôle des types



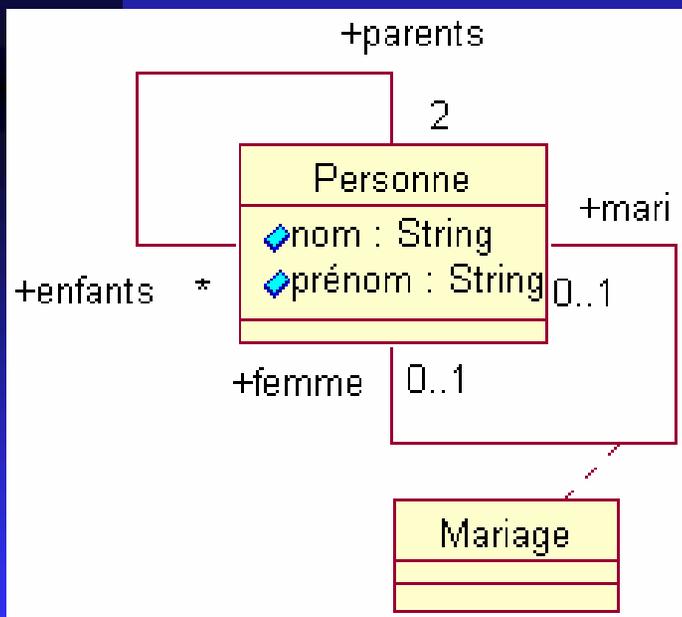
# Xion: arbre abstrait (2)



# Optimisation SQL

Prenons un exemple de modèle métier:

Ce qui donne les tables:



```
personne(OID, nom,  
         prenom)
```

```
mariage(OID, #mari,  
        #femme)
```

```
parents_enfants(#parents,  
                #enfants)
```

# Optimisation SQL

« Personne » est traduit par une classe en langage cible:

```
classe Personne
  attribut oid : String

  fonction get_nom : String
    retourne execute_SQL(
      `SELECT personne.nom
      FROM personne
      WHERE pesonne.OID = ` + oid
    )
  fin get_nom

  fonction set_nom (nom:String)
    execute_SQL(
      `UPDATE personne
      SET nom = ` + nom + `
      WHERE OID = ` + oid
    )
  fin set_nom ...
```

# Optimisation SQL: exemple 1

Un code Xion:

`maPersonne.enfants`

Le script correspondant:

```
execute_SQL(  
    `SELECT enfants  
    FROM parents_enfants  
    WHERE parents = `  
    + maPersonne.oid  
    )
```

# Optimisation SQL: exemple 2

Un code Xion: `maPersonne.enfants->collect(nom)`

Le script correspondant:

```
Ensemble(Personne) tmp1 = execute_SQL(  
    `SELECT enfant  
    FROM parents_enfants  
    WHERE parents = `  
    + maPersonne.oid  
    )  
Ensemble(String) tmp2 = Ensemble vide  
Énumération e = tmp1.éléments  
Tant que e a des éléments faire  
    tmp2.ajoute(e.suivant.get_nom)  
Fin tant que
```

# Optimisation SQL: exemple 2 idéal

Un code Xion: `maPersonne.enfants->collect(nom)`

Le code idéal:

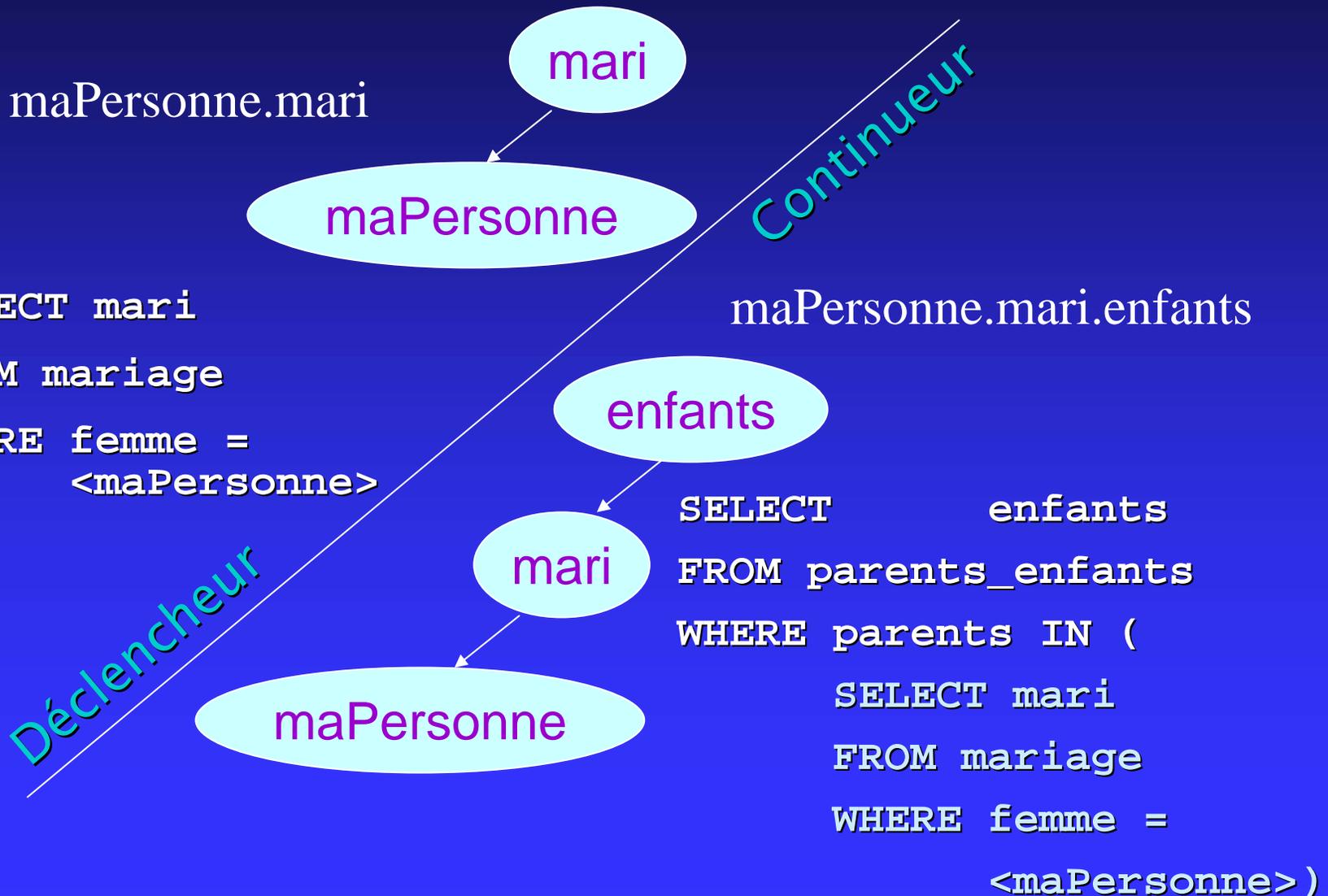
```
execute_SQL(  
    `SELECT personne.nom  
FROM personne  
WHERE personne.OID in (  
    SELECT parent_enfant.enfant  
FROM parent_enfant  
    WHERE parent_enfant.parent =  
    ` + maPersonne.oid + `)`  
)
```

# Optimisation SQL: 4 cas

Basé sur les appels d'opérations  
prédéfinies

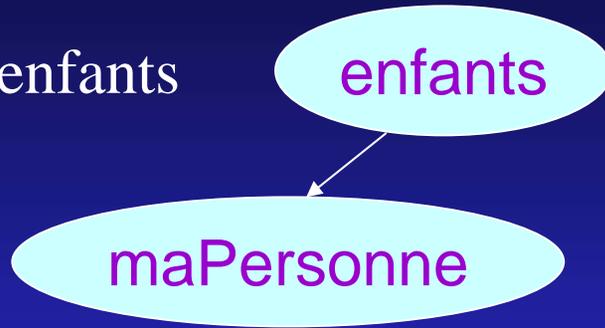
- Déclencheur: quand rien n'a précédemment été optimisé
- Continueur: modifie l'optimisé existant
- Continueur d'itération: modifie l'existant dans une opération d'itération *Xion* comme « select » ou « sortedBy »
- Continueur de « collect »: modifie l'existant dans un « collect »

# Optimisation SQL: les liens



# Optimisation SQL: les liens

maPersonne.enfants



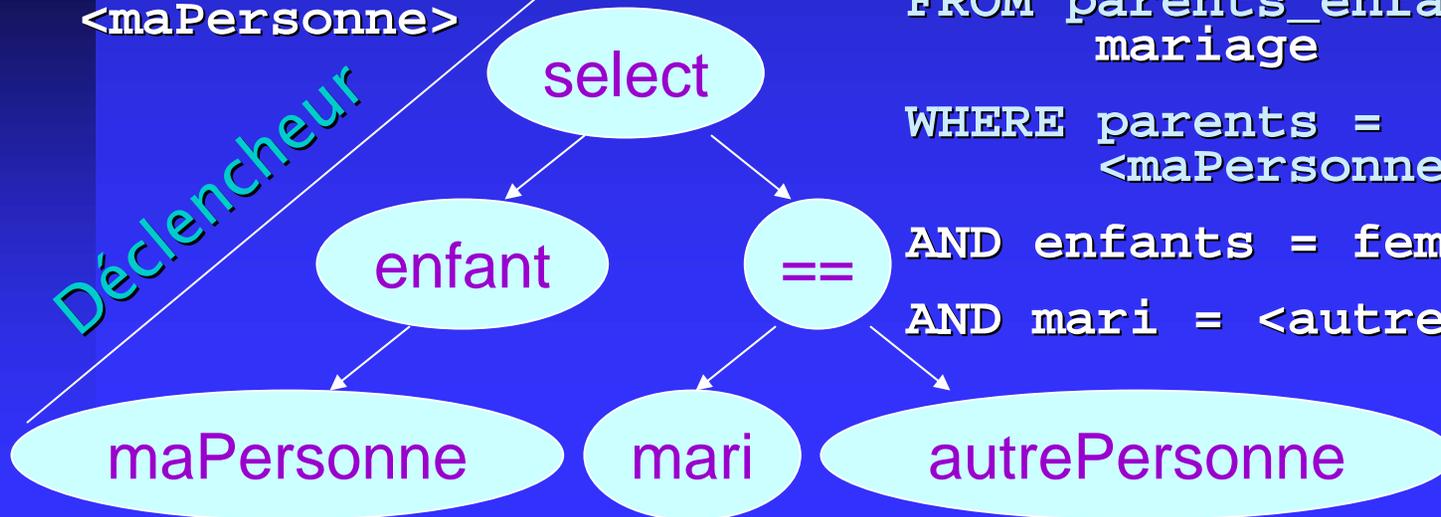
maPersonne.enfants->  
select(mari ==  
autrePersonne)

*Continuateur d'itération*

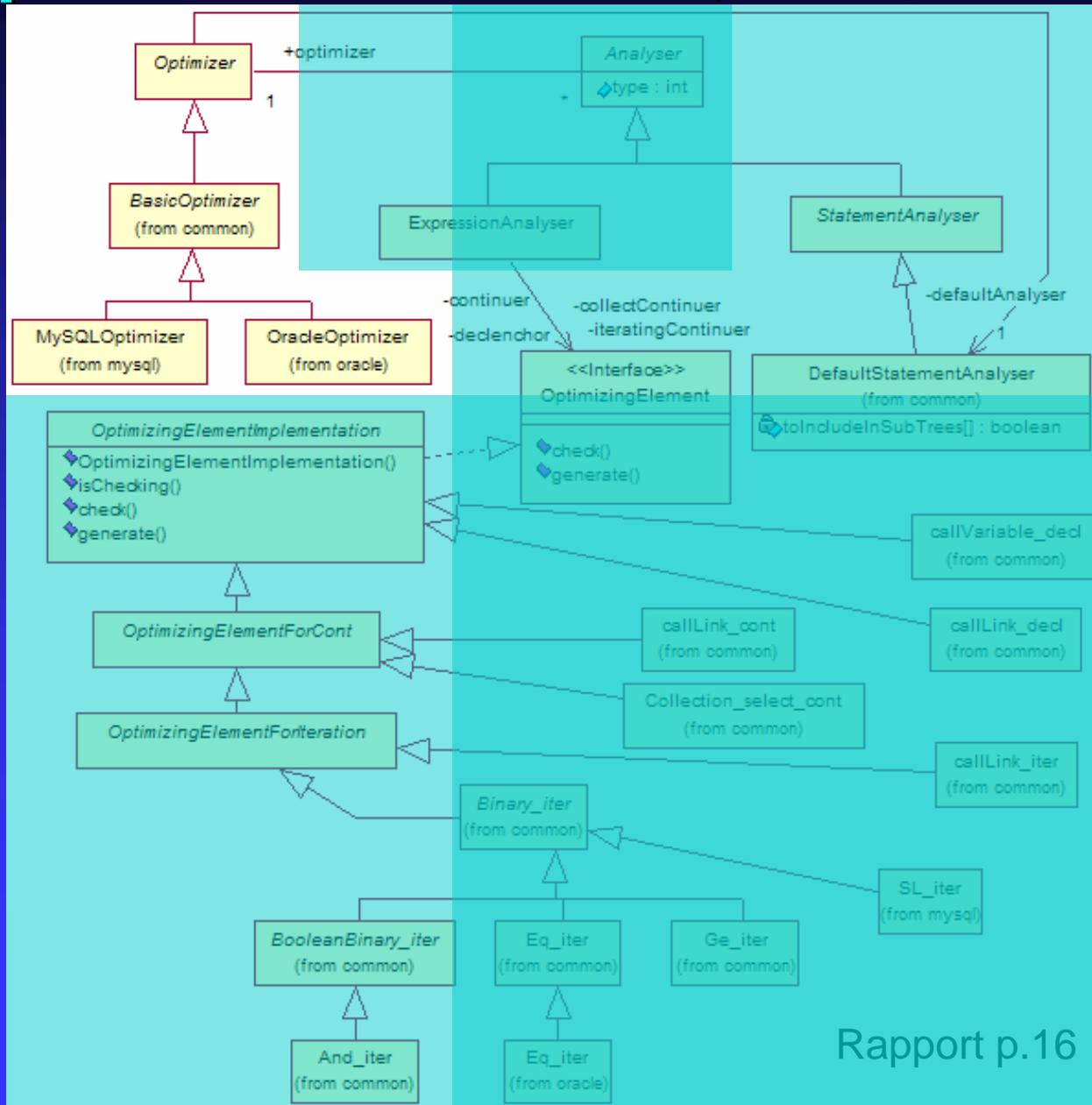
```
SELECT enfants  
FROM parents_enfants  
WHERE parents =  
<maPersonne>
```

```
SELECT      enfants  
FROM parents_enfants,  
      mariage  
WHERE parents =  
      <maPersonne>  
AND enfants = femme  
AND mari = <autrePersonne>
```

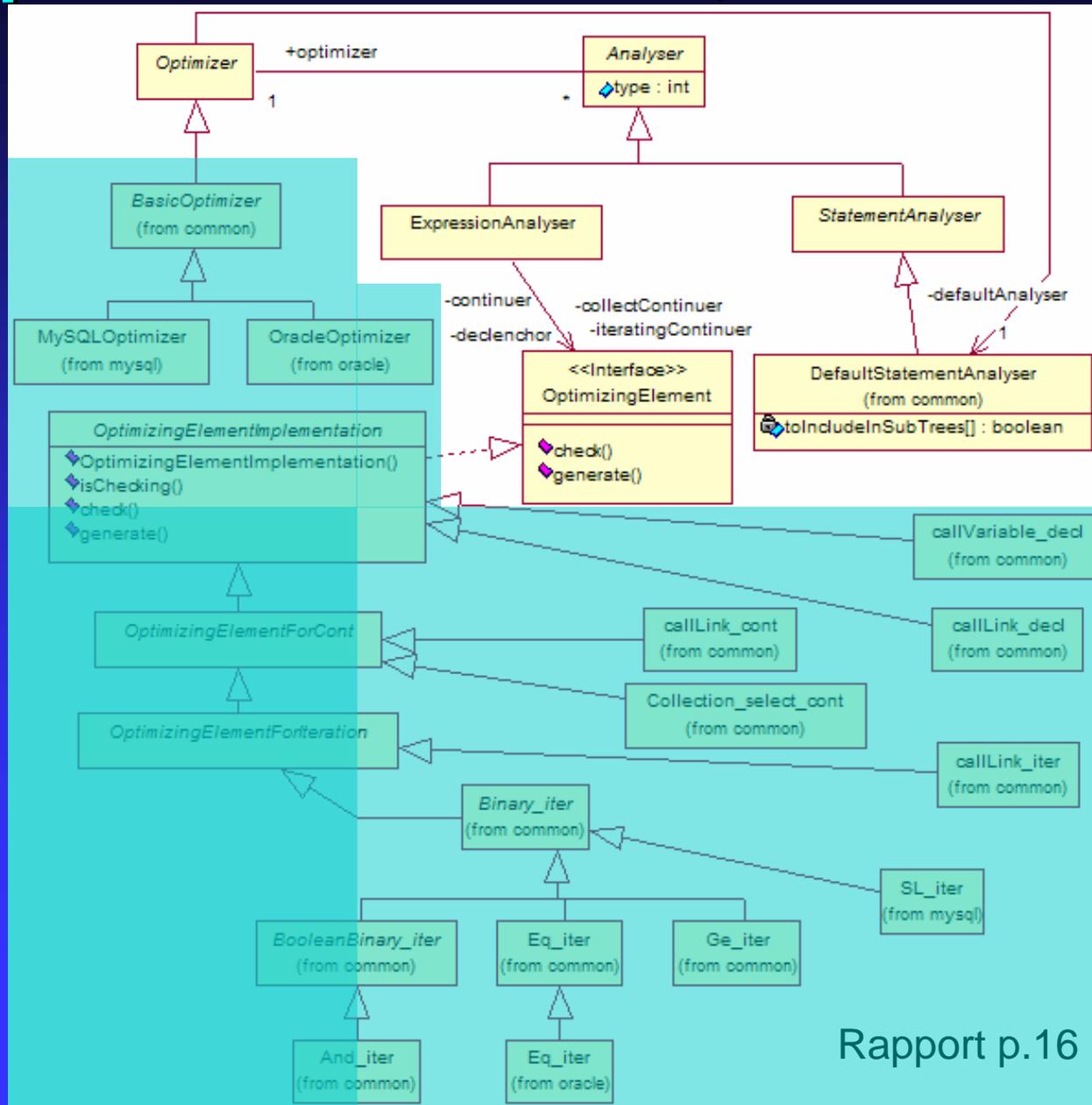
*Déclencheur*



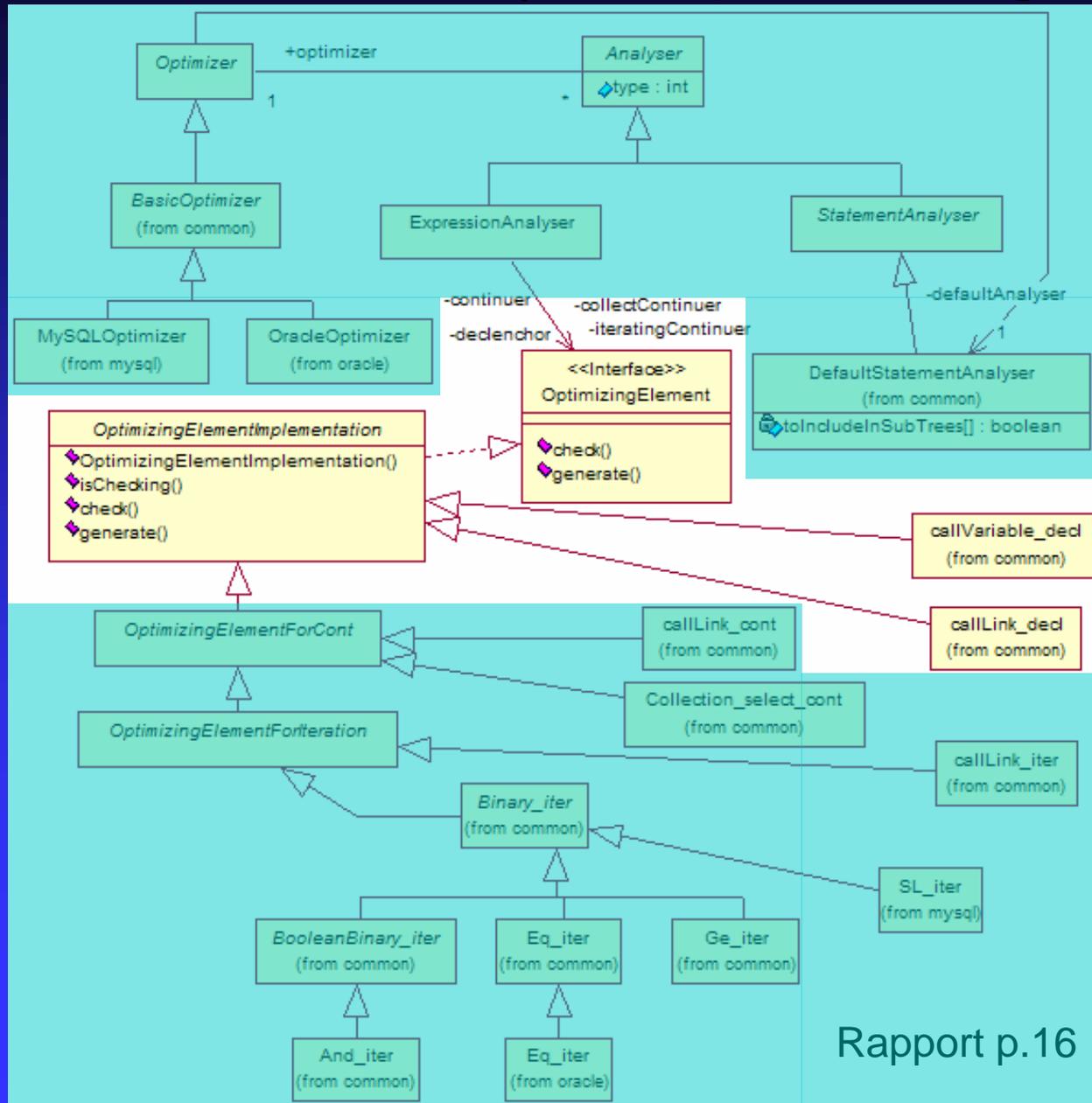
# Optimisation SQL: Optimizer



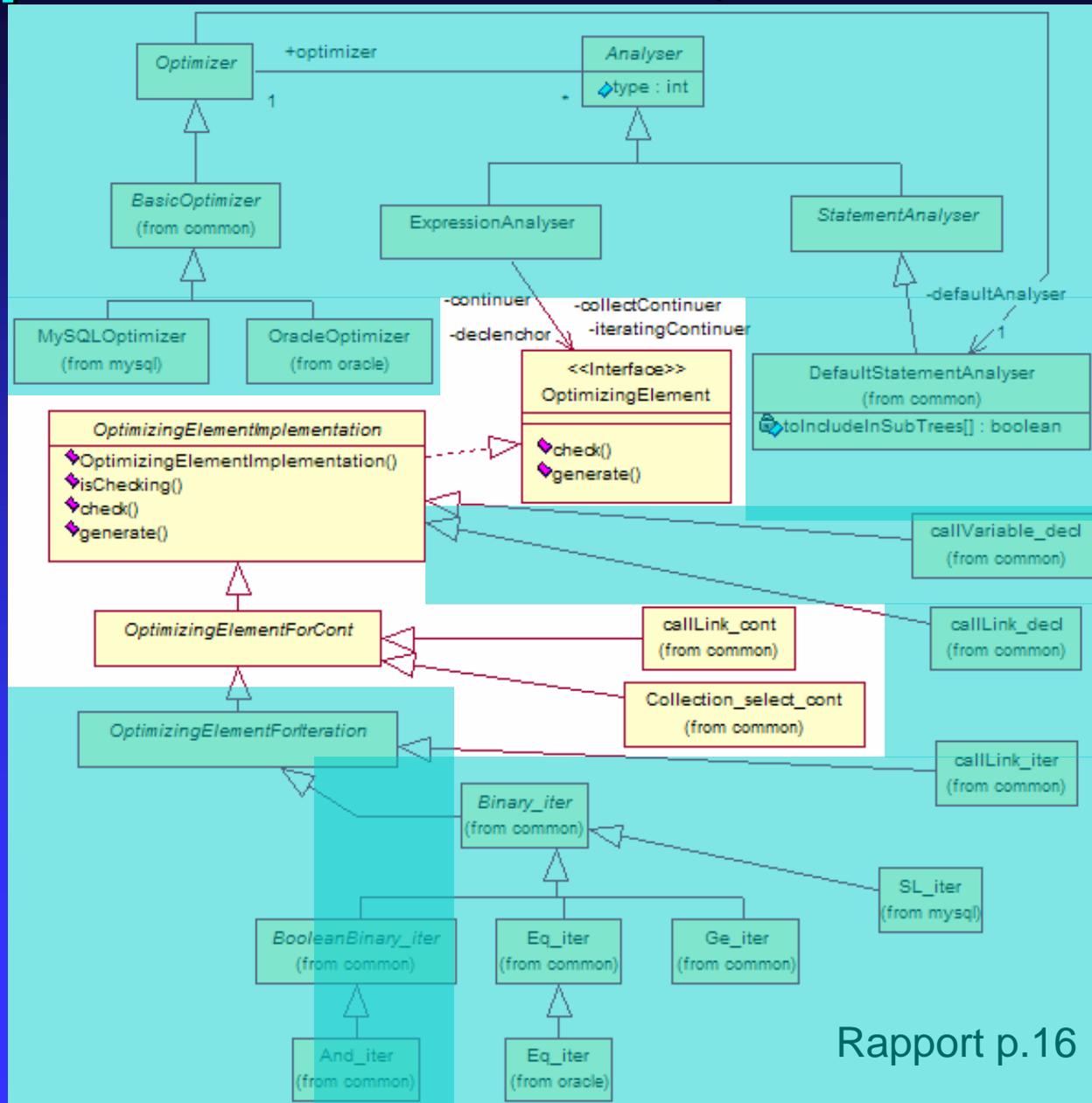
# Optimisation SQL: Analyzer



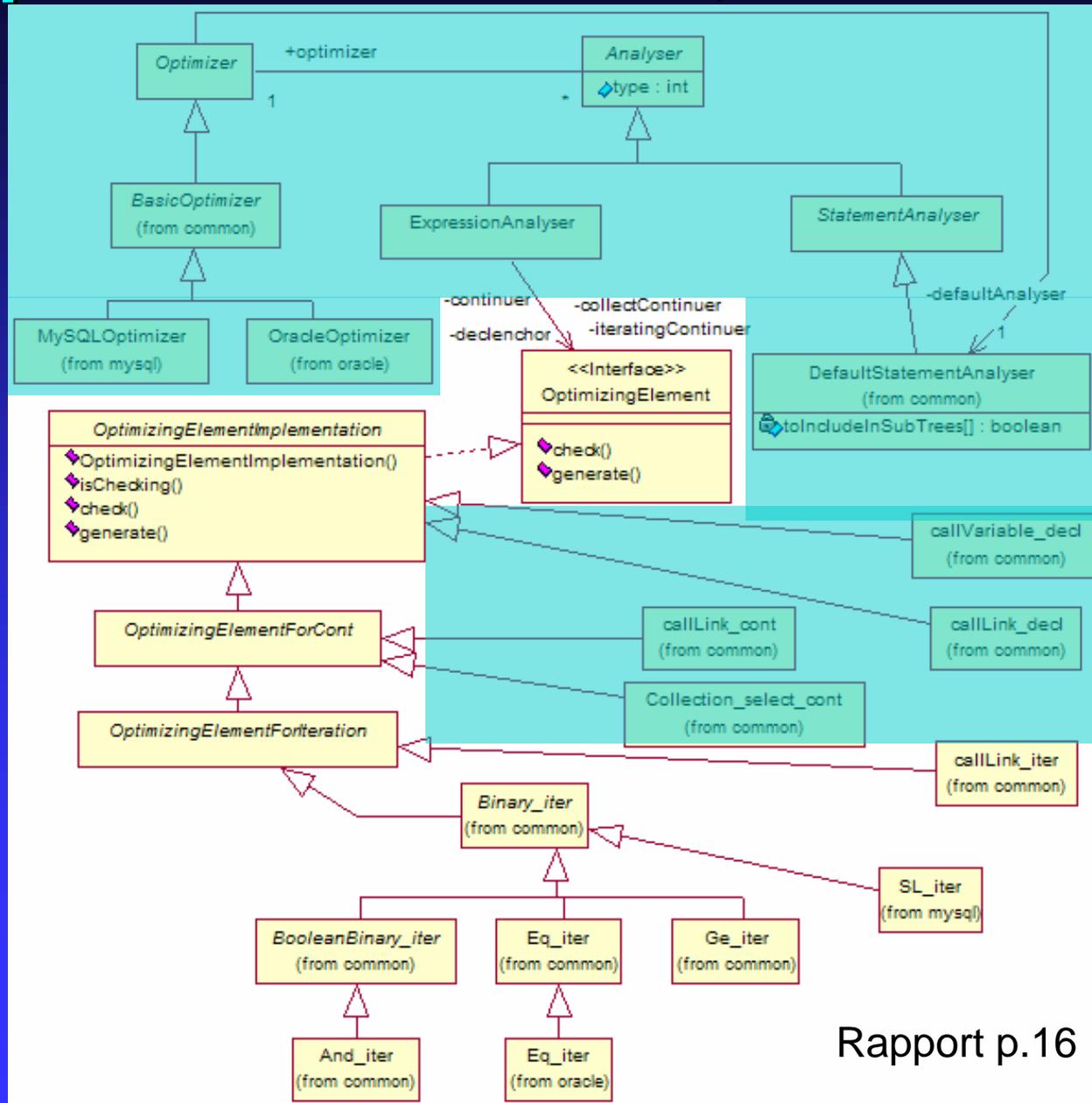
# Optimisation SQL: OptimizingElement



# Optimisation SQL: Continueurs



# Optimisation SQL: Itérateurs



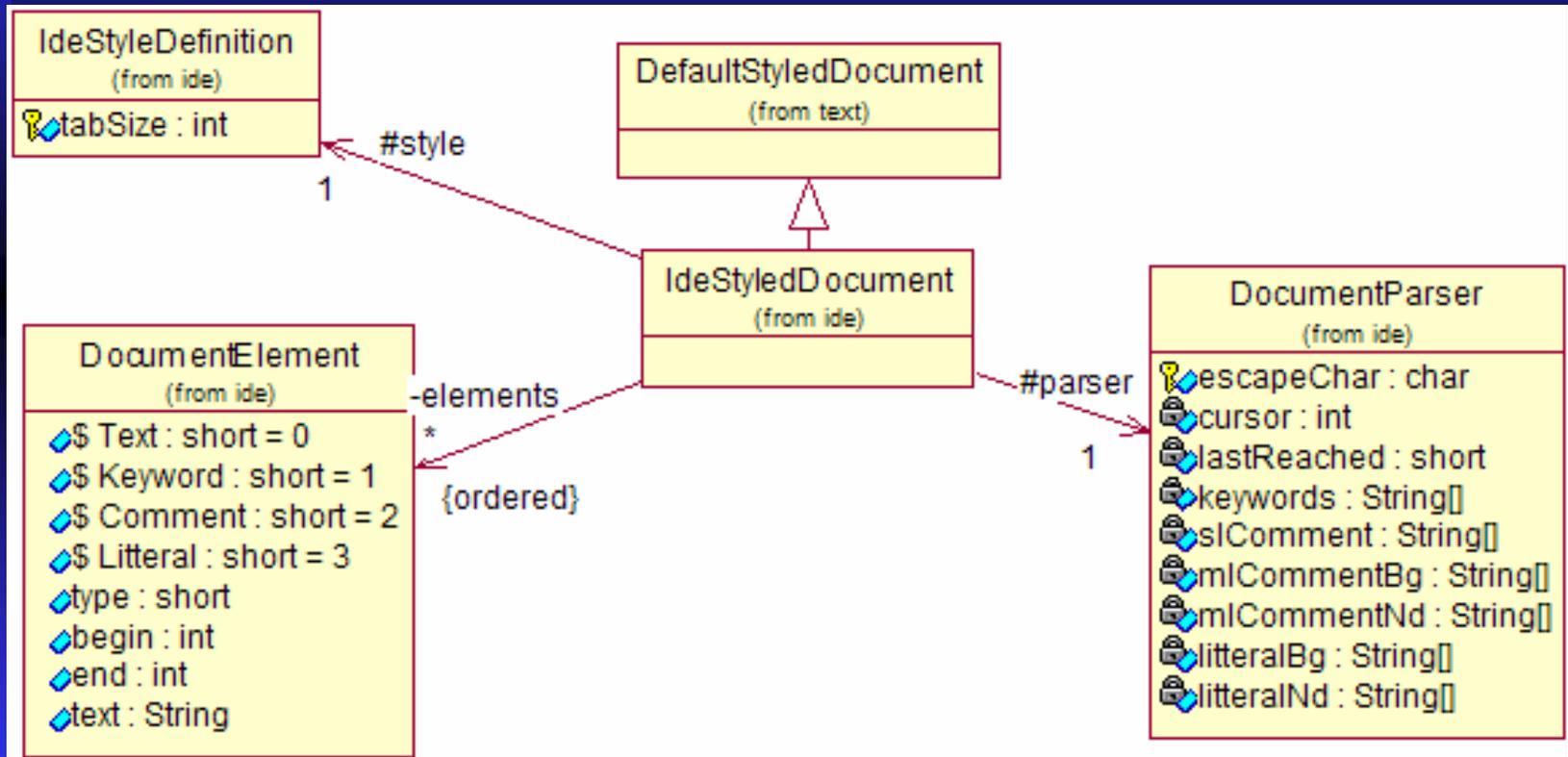
# Éditeur de code

Netsilon est innovant

=> Difficile à appréhender

- Fonctionnalités classiques
- Coloration syntaxique
- Complétion sémantique

# Coloration syntaxique



# Complétion sémantique

Liste les champs accessibles d'un objet

Réutilisation des premières couches du compilateur

Mise en forme du texte transmis au compilateur

```
Integer i = 0;
```

```
this.setAttribute(i.max(21).
```

Devient

```
Integer i = 0;
```

```
i.max(21);
```

```
mesPersonnes->select (i:
```

```
    i.mari.
```

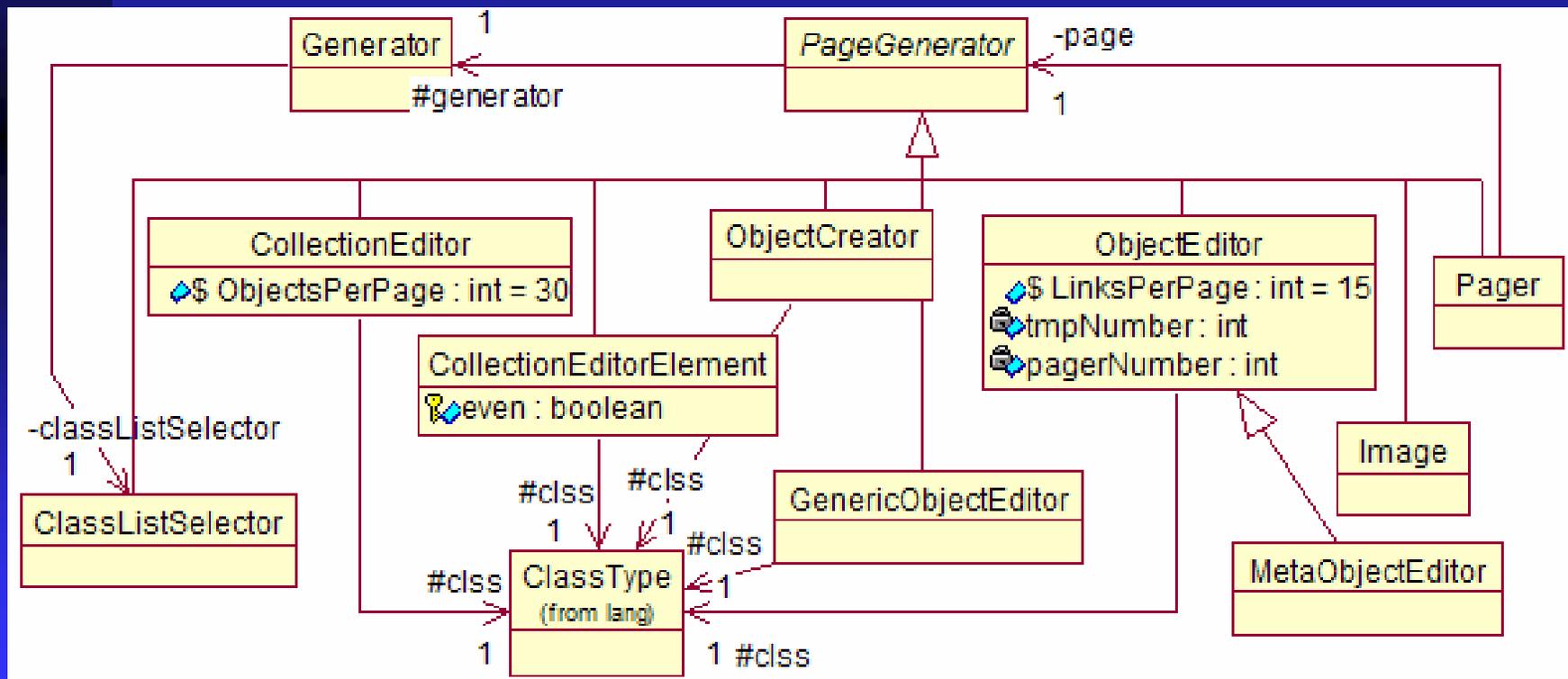
Devient

```
Personne i;
```

```
i.mari;
```

# Gestion des objets métier

Pour nourrir la base de données d'objets



# Conclusion

Xion est incontournable pour modéliser une application web avec Netsilon.

Pas de langage d'action... Pourquoi pas Xion ?

Xion, a l'instar des autres langages, ne résout pas tous les problèmes: besoin d'extension au langage par profilage.