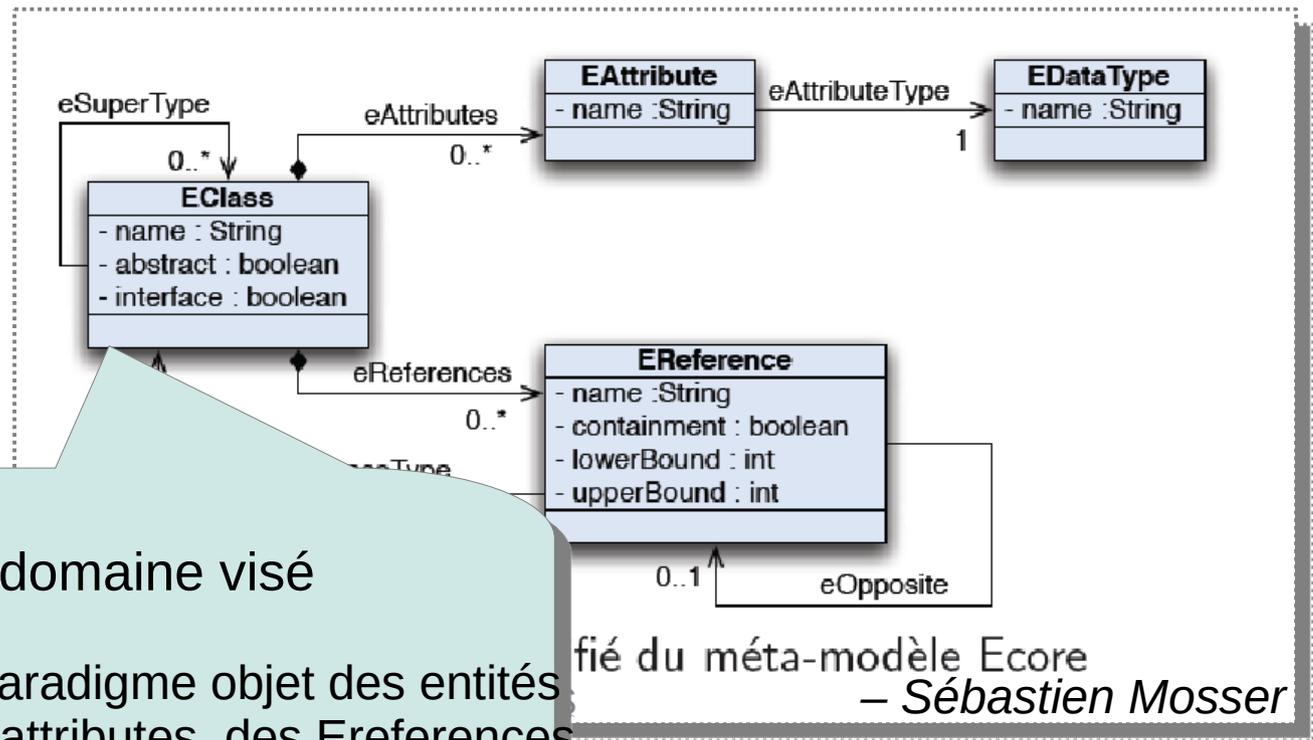


# Outillage de la syntaxe concrète: XText

# Eclipse Modeling Framework



**ECore**

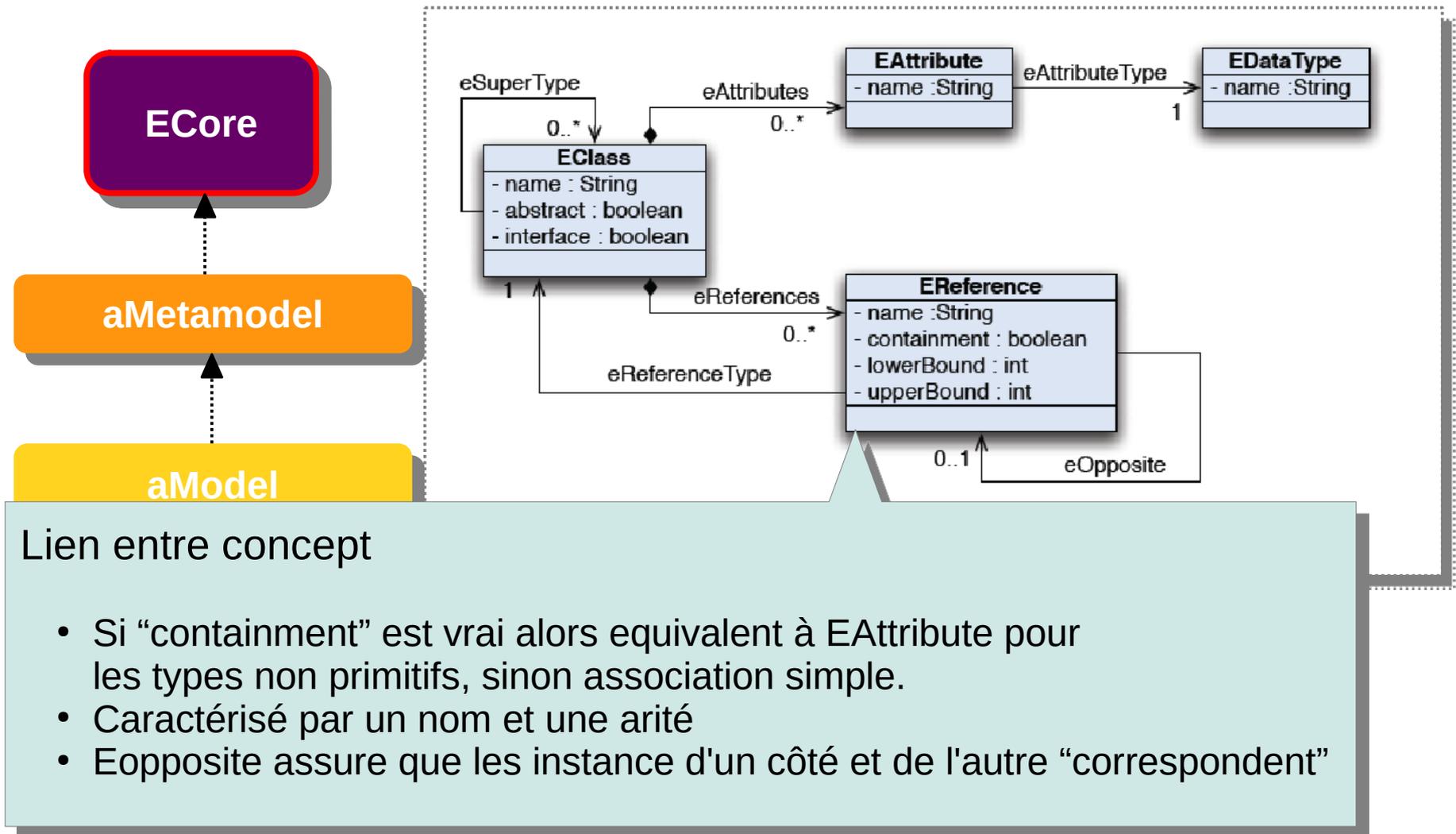


Modélisé du méta-modèle Ecore  
 – Sébastien Mosser

Modélise un concept du domaine visé

- Modélisation selon le paradigme objet des entités
- Caractérisée par des Eattributes, des Ereferences (et des EOperations)

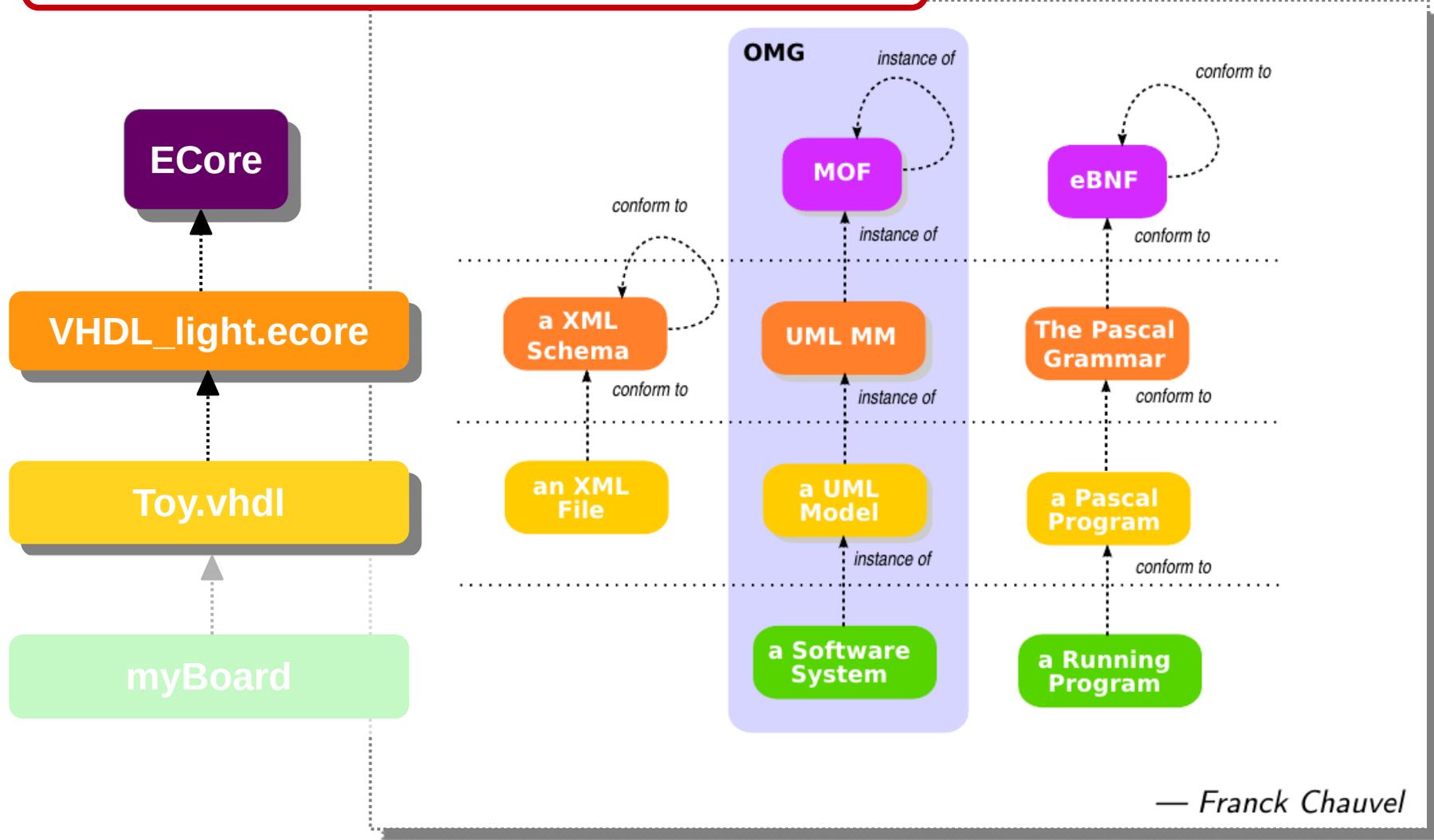
# Eclipse Modeling Framework



# Eclipse Modeling Framework

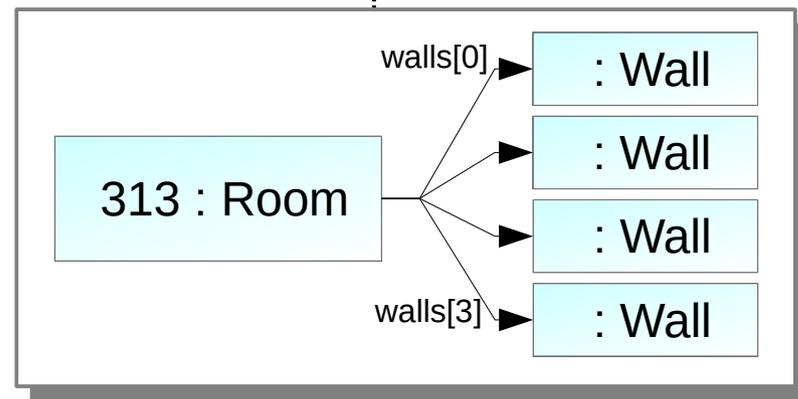
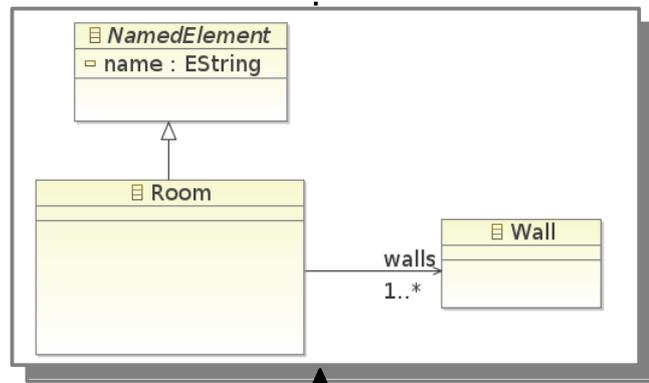
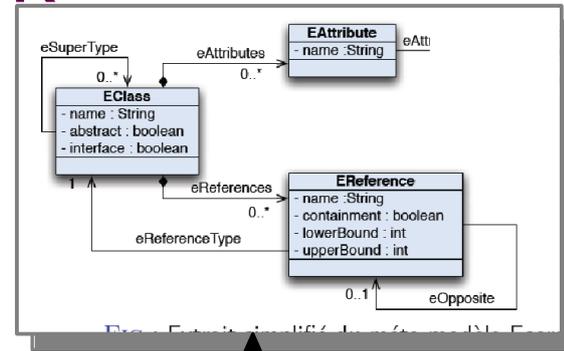
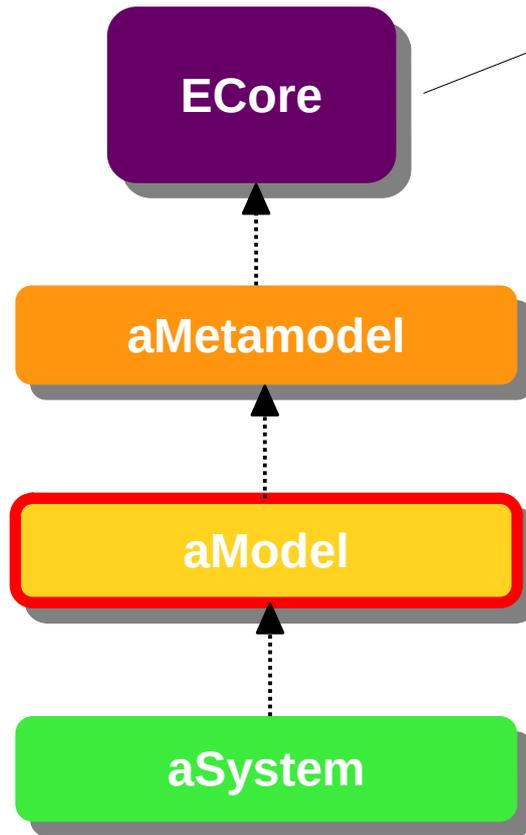


**Vous avez un métamodèle et des modèles**



— Franck Chauvel

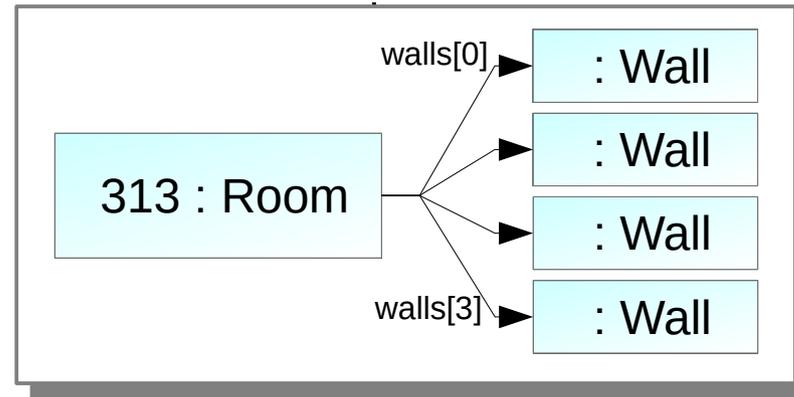
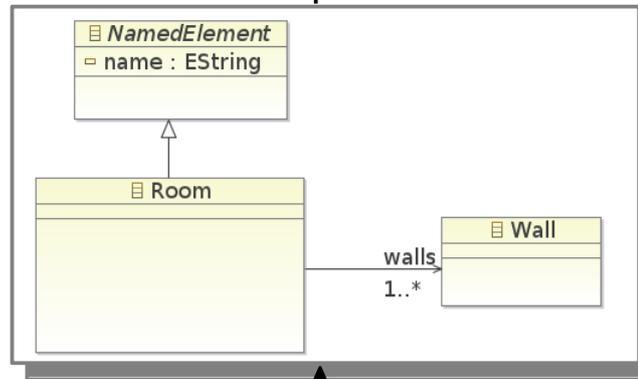
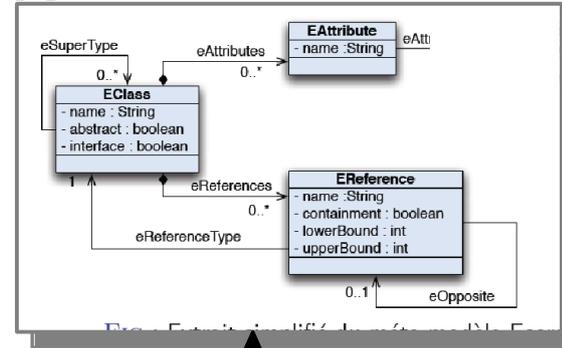
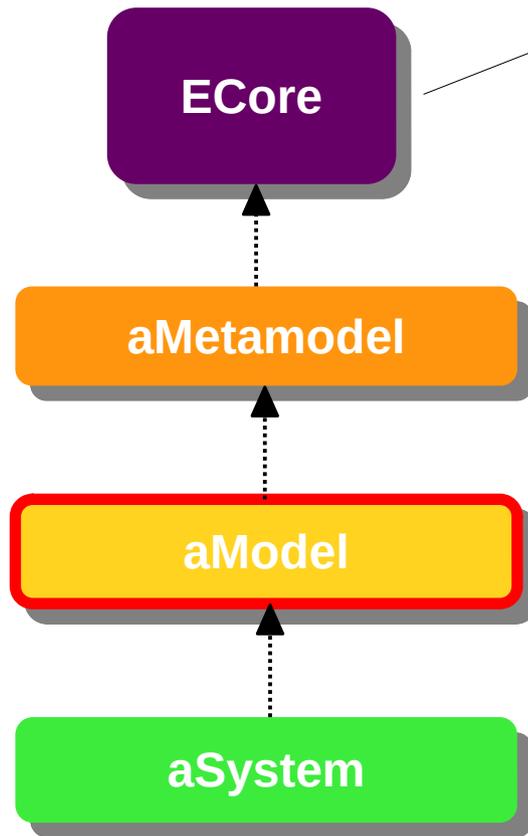
# Eclipse Modeling Framework



*conformsTo*

*conformsTo*

# Eclipse Modeling Framework



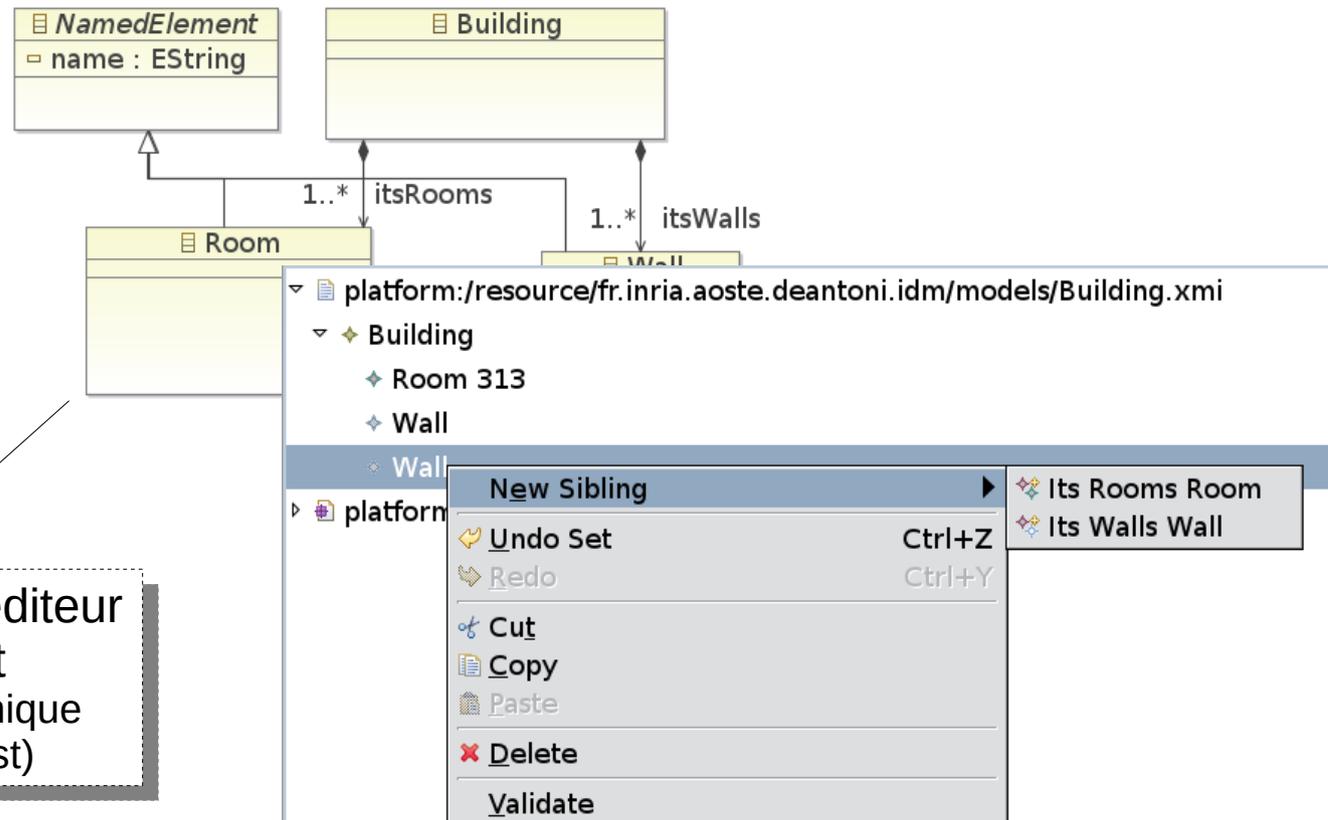
*conformsTo*

*conformsTo*

**Syntaxe concrete graphique...**

# Avantages techniques de la méta-modélisation

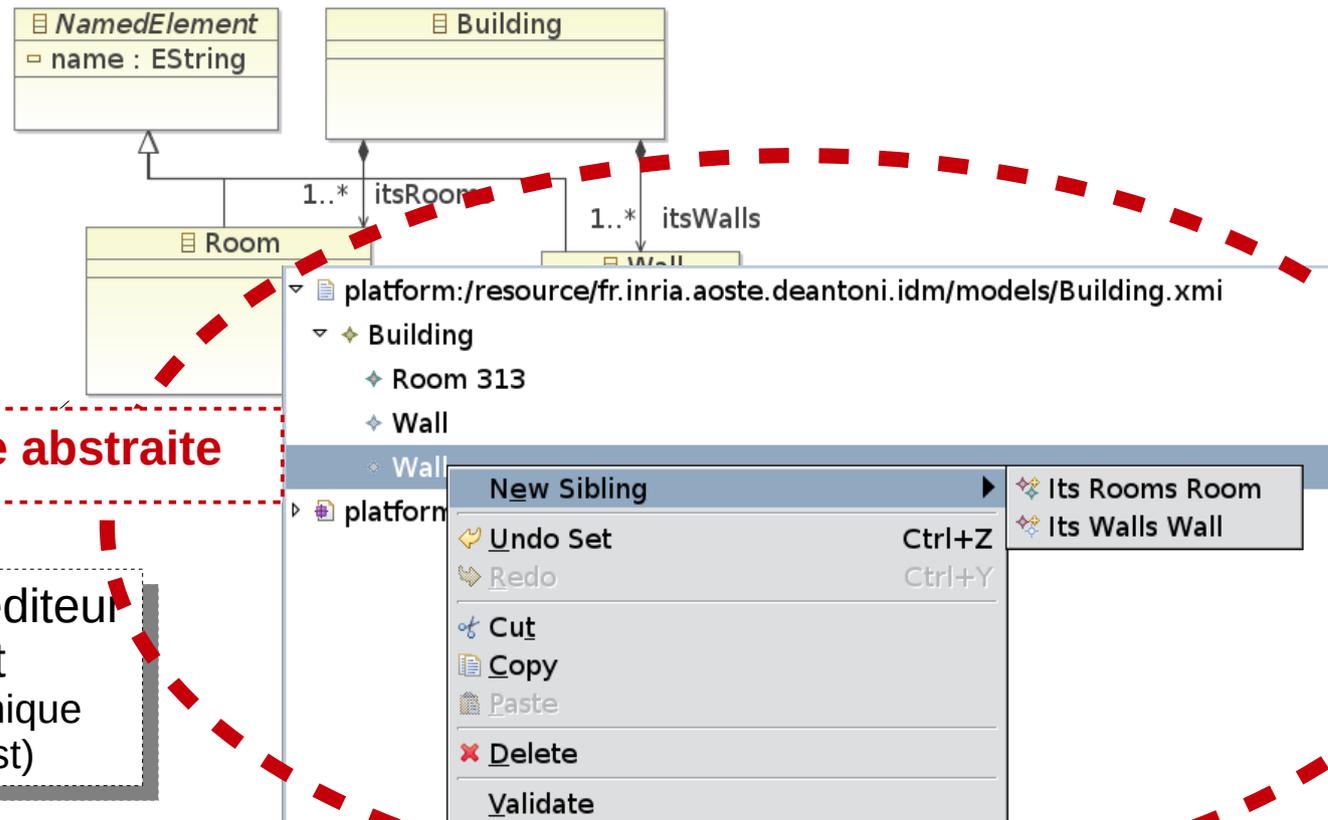
- Pourquoi : Pour faciliter la manipulation de modèles (éditeurs, transformations, ...)



Génération d'un éditeur arborescent (édition semi graphique permettant le test)

# Avantages techniques de la méta-modélisation

- Pourquoi : Pour faciliter la manipulation de modèles (éditeurs, transformations, ...)

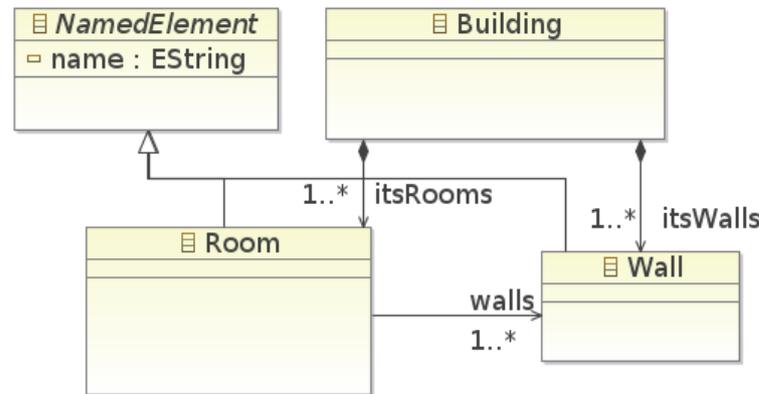


**Éditeur de la syntaxe abstraite**

Génération d'un éditeur arborescent (édition semi graphique permettant le test)

# Avantages techniques de la méta-modélisation

- Pourquoi : Pour faciliter la manipulation de modèles (éditeurs, transformations, ...)



Génération d'une API de manipulation java (sérialisation / désérialisation gratuites)

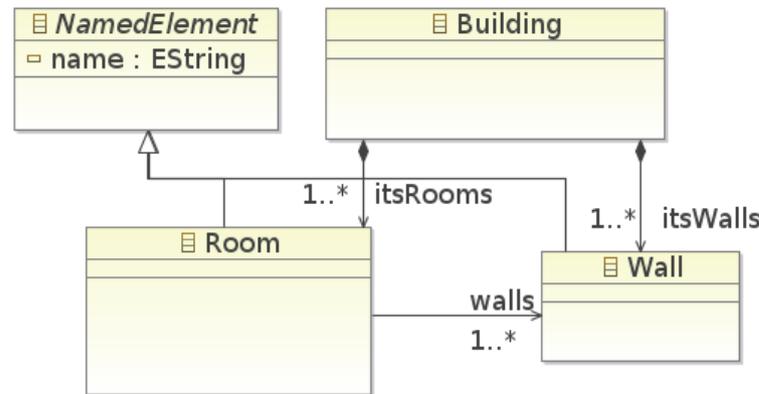
Génération d'un éditeur arborescent (édition semi graphique permettant le test)

Accès aux outils basés sur les métamodèles:

- Xtext
- sirius
- Transformation modèle à modèle
- Transformations modèle à texte
- ...

# Avantages techniques de la méta-modélisation

- Pourquoi : Pour faciliter la manipulation de modèles (éditeurs, transformations, ...)



Outil de création d'une syntaxe concrète...

Accès aux outils basés sur les métamodèles:

- Xtext
- Sirius
- Transformation modèle à modèle
- Transformations modèle à texte
- ...

# Syntaxes concrètes et abstraites (caricature)

- Syntaxe abstraite

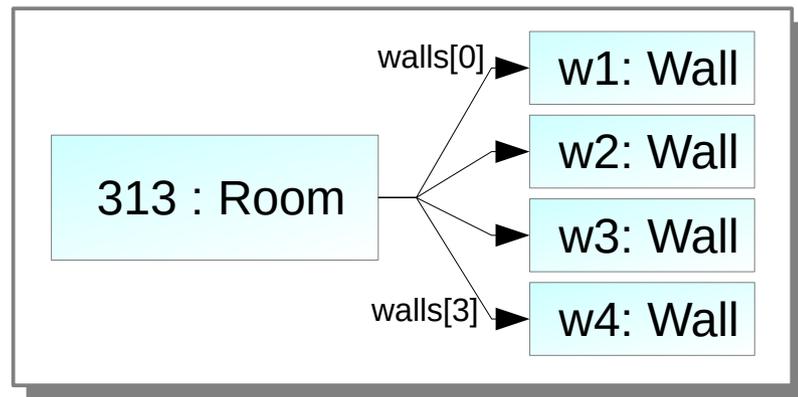


- Syntaxe concrète

textuelle

```
Building{
  Wall w1;
  Wall w2;
  Wall w3;
  Wall w4;
  Room 313 (w1, w2, w3, w4);
}
```

graphique



# Syntaxes concrètes et abstraites (caricature)

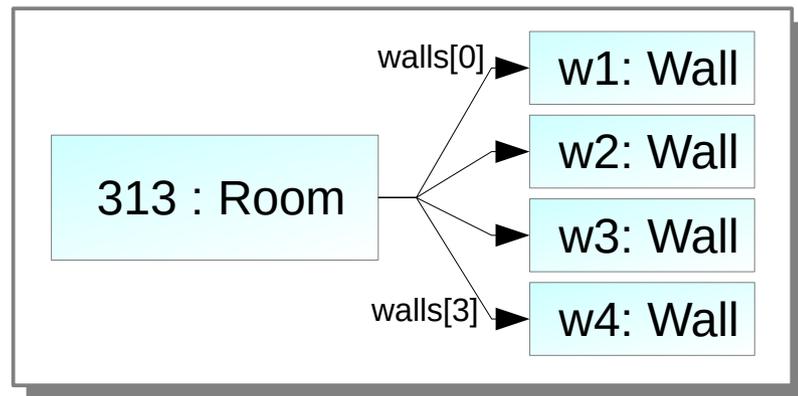
- Syntaxe abstraite



**Xtext** est un framework d'aide à la création de syntaxe concrète textuelles et des outils associés...

```
Building{
    Wall w1;
    Wall w2;
    Wall w3;
    Wall w4;
    Room 313 (w1, w2, w3, w4);
}
```

graphique



# Xtext en un transparent

- Ici: <http://www.eclipse.org/Xtext/>
- À partir d'une grammaire BNF étendue, on obtient:
  - ANTLR3-based **parser**
  - EMF-based **metamodel**
  - Eclipse **editor** with
    - syntax highlighting
    - code completion
    - customizable outline
    - code folding
    - real-time constraint checking
  - A Web editor
  - A LSP server

# Xtext en un transparent

- Ici: <http://www.eclipse.org/Xtext/>
- À partir d'une grammaire BNF étendue, on obtient:

- ANTLR3-based **parser**
- ~~EMF-based **metamodel**~~ →
- Eclipse **editor** with
  - syntax highlighting
  - code completion
  - customizable outline
  - code folding
  - real-time constraint checking
- A Web editor
- A LSP server



Éviter de générer un méta-modèle à partir de la syntaxe concrète, par contre baser la grammaire sur le méta-modèle est une bonne pratique

# Spécification de la BNF étendue

Name of your grammar

Reused grammar

```
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"

import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
    {VDHL_Specification}
    ( '#-'
        (libraries+=Library)* &
        (modules+=Module)* &
        ((entities+=Entity)*)? &
        (( architectures+=Architecture)* )?
    );

Library returns Library:
    'library' name=EString ';'
;

Module returns Module:
    'use' library=[Library|EString] ( '.' path+=EString )* ';'
;

[SE-1.5]
```

# Spécification de la BNF étendue

Métamodèle(s) utilisé(s) dans la grammaire

```
VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
    {VDHL_Specification}
    ('#-'
        (libraries+=Library)* &
        (modules+=Module)* &
        |(entities+=Entity)* &
        (( architectures+=Architecture)* )?
    );

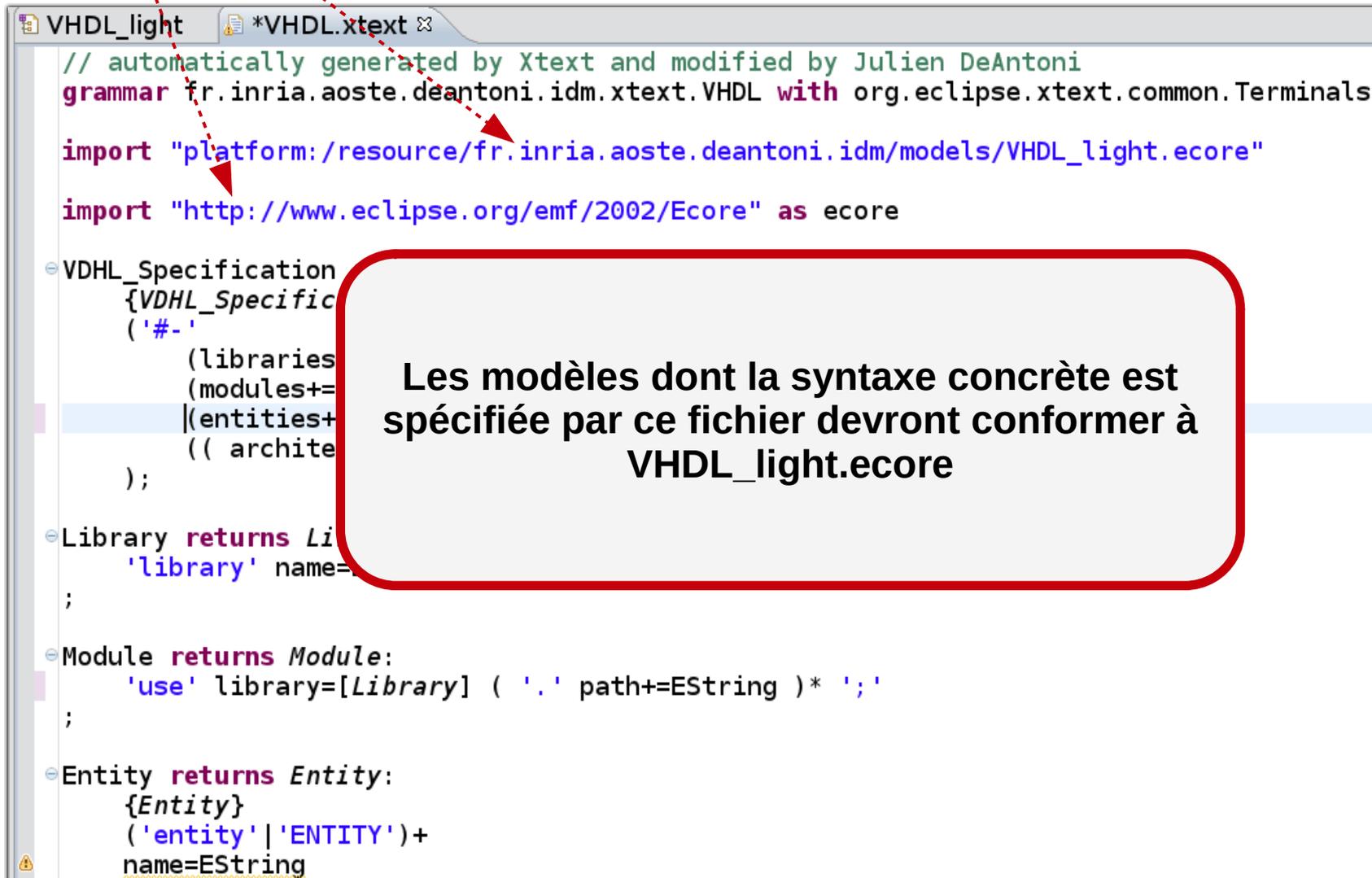
Library returns Library:
    'library' name=EString ';'
;

Module returns Module:
    'use' library=[Library] ( '.' path+=EString )* ';'
;

Entity returns Entity:
    {Entity}
    ('entity'|'ENTITY')+
    name=EString
```

# Spécification de la BNF étendue

Métamodèle(s) utilisé(s) dans la grammaire



```
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification
  {VDHL_Specific
    ('#-'
      (libraries
      (modules+=
|entities+
      (( archite
    ));

Library returns Li
  'library' name=

;

Module returns Module:
  'use' library=[Library] ( '.' path+=EString )* ';'
;

Entity returns Entity:
  {Entity}
  ('entity'|'ENTITY')+
  name=EString
```

Les modèles dont la syntaxe concrète est spécifiée par ce fichier devront conformer à VHDL\_light.ecore

# Spécification de la BNF étendue

Nom d'une règle grammaticale

Type de l'élément généré par cette règle  
Concept de VHDL\_light.ecore

```

VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"

import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
    {VDHL_Specification}
    ( '#' -
        (libraries+=Library)* &
        (modules+=Module)* &
        |(entities+=Entity)* &
        (( architectures+=Architecture)* )?
    );

Library returns Library:
    'library' name=EString ';'
;

Module returns Module:
    'use' library=[Library] ( '.' path+=EString )* ';'
;

Entity returns Entity:
    {Entity}
    ('entity'|'ENTITY')+
    name=EString
    
```

# Spécification de la BNF étendue

Mot clef

```
VHDL_light *VHDL.xtext x
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
    {VDHL_Specification}
    ('#-'
        (libraries+=Library)* &
        (modules+=Module)* &
        (entities+=Entity)* &
        (( architectures+=Architecture)* )?
    );

Library returns Library:
    'library' name=EString ';'
;

Module returns Module:
    'use' library=[Library] ( '.' path+=EString )* ';'
;

Entity returns Entity:
    {Entity}
    ('entity'|'ENTITY')+
    name=EString
```

# Spécification de la BNF étendue

Mot clef

Ajout dans une collection nommée 'librairies' d'un élément généré par la règle nommée 'Library'

```

VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
{VDHL_Specification}
('#-'
 (libraries+=Library)* &
 (modules+=Module)* &
 (entities+=Entity)* &
 (( architectures+=Architecture)* )?
);

Library returns Library:
'library' name=EString ';'
;

Module returns Module:
'use' library=[Library] ( '.' path+=EString
;

Entity returns Entity:
{Entity}
('entity'|'ENTITY')+
name=EString

```

Affectation d'une propriété de type EString

# Spécification de la BNF étendue

Mot clef

Ajout dans une collection nommée 'librairies' d'un élément généré par la règle nommée 'Library'

```

VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext:VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
{VDHL_Specification}
('#-'
 (libraries+=Library)*
 (modules+=Module)* &
 |(entities+=Entity)* &
 (( architectures+=Architecture)* )?
);

Library returns Library:
'library' name=EString ';'
;

Module returns Module:
'use' library=[Library] ( '.' path+=EString
;

Entity returns Entity:
{Entity}
('entity'|'ENTITY')+
name=EString

```

Cette règle *match* un nombre indéterminé de fois

Cette règle *match* 0 ou 1 fois (juste pour l'exemple)

Affectation d'une propriété de type EString

Cette règle *match* plus de 1 fois (juste pour l'exemple)

# Spécification de la BNF étendue

Mot clef

Ajout dans une collection nommée 'librairies' d'un élément généré par la règle nommée 'Library'

```

VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext:VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
{VDHL_Specification}
('#-'
 (libraries+=Library)* &
 (modules+=Module)* &
 (entities+=Entity)* &
 (( architectures+=Architecture)* )?
);

Library returns Library:
'library' name=EString ';';

Module returns Module:
'use' library=[Library] ( '.' path+=EString
);

Entity returns Entity:
{Entity}
('entity'|'ENTITY')+
name=EString
    
```

Cette règle *match* nombre indéterminé de fois

Les règles dans le bloc peuvent être saisies dans un ordre quelconque

Cette règle *match* 0 ou 1 fois (juste pour l'exemple)

Affectation d'une propriété de type EString

Cette règle *match* plus de 1 fois (juste pour l'exemple)

# Spécification de la BNF étendue

Mot clef

Ajout dans une collection nommée 'librairies' d'un élément généré par la règle nommée 'Library'

```

VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
{VDHL_Specification}
('#-'
 (libraries+=Library)* &
 (modules+=Module)* &
 (entities+=Entity)* &
 (( architectures+=Architecture)* )?
);

Library returns Library:
'library' name=EString ';'
;

Module returns Module:
'use' library=[Library] ( '.' path+=EString
;

Entity returns Entity:
{Entity}
('entity' | 'ENTITY')+
name=EString

```

Cette règle *match* nombre indéterminé de fois

Les règles dans le bloc peuvent être saisies dans un ordre quelconque

Cette règle *match* 0 ou 1 fois (juste pour l'exemple)

Affectation d'une propriété de type EString

Cette règle *match* plus de 1 fois (juste pour l'exemple)

Alternative (ici utilisée entre mots clefs mais ce pourrait être entre règles)

# Spécification de la BNF étendue

Mot clef

Ajout dans une collection nommée 'librairies' d'un élément généré par la règle nommée 'Library'

```

VHDL_light *VHDL.xtext
// automatically generated by Xtext and modified by Julien DeAntoni
grammar fr.inria.aoste.deantoni.idm.xtext.VHDL with org.eclipse.xtext.common.Terminals

import "platform:/resource/fr.inria.aoste.deantoni.idm/models/VHDL_light.ecore"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

VDHL_Specification returns VDHL_Specification:
{VDHL_Specification}
('#-'
 (libraries+=Library)* &
 (modules+=Module)* &
 |(entities+=Entity)* &
 (( architectures+=Architecture)* )?
);

Library returns Library:
'library' name=EString ';'

Module returns Module:
'use' library=[Library] ( '.' path+=EString

Entity:
'ENTITY'+

```

Cette règle *match* nombre indéterminé de fois

Les règles dans le bloc peuvent être saisies dans un ordre quelconque

Cette règle *match* 0 ou 1 fois (juste pour l'exemple)

Affectation d'une propriété de type EString

Cette règle *match* plus de 1 fois (juste pour l'exemple)

Alternative (ici utilisée entre mots clefs mais ce pourrait être entre règles)

**Cross-référence !!**  
**[Library] est ici un type et non une règle**

# Spécification de la BNF étendue

Mot clef

Ajout dans une collection nommée 'librairies' d'un élément généré par la règle nommée 'Library'

Besoin de plus d'infos ?

<https://www.eclipse.org/Xtext/index.html>

mbre  
s

peuvent être  
saisies dans un ordre quelconque

```
(( architectures+=Architecture)* )?
```

```
Library returns Library:  
  'library' name=EString ';' ;
```

```
Module returns Module:  
  'use' library=[Library] ( '.' path+=EString
```

Cette règle *match* 0 ou 1 fois  
(juste pour l'exemple)

Affectation d'une propriété de type  
EString

Cross-référence !!  
[Library] est ici un type et  
non une règle

```
Entity:
```

```
'ENTITY'+
```

Cette règle *match* plus de 1 fois  
(juste pour l'exemple)

Alternative (ici utilisée entre mots clefs  
mais ce pourrait être entre règle)