Langage et Compilation

langage, syntaxe et sémantique par traduction

Julien Deantoni

Syllabus (1/3)

-
  -
  -
  -
-
  -
  -

Syllabus (2/3)

-
  -
  -
    -
    -
-
  -

Syllabus (3/3)

-

Introduction

langage

nom masculin

(de langue)

1. Capacité, observée chez tous les hommes, d'exprimer leur pensée et de communiquer au moyen d'un système de signes vocaux et éventuellement graphiques (la langue).
   SYNONYME :
   parole

2. Tout système structuré de signes non verbaux remplissant une fonction de communication : Langage gestuel. Langage animal.
   SYNONYMES :
   idiome - langue

3. Manière de parler propre à un groupe social ou professionnel, à une discipline, à un domaine d'activité, à quelque chose : Le langage administratif.
   SYNONYMES :
   argot - jargon - lexique - vocabulaire

4. Manière particulière de s'exprimer d'un groupe, de quelqu'un, jugée par rapport à une norme : Langage populaire, familier.

5. Manière de s'exprimer, choix des termes lorsqu'on s'adresse à quelqu'un : Quand il ne sera plus en colère, il tiendra un tout autre langage.

6. Expression propre à un sentiment, une attitude : Le langage de la raison.

7. Ensemble des procédés utilisés par un artiste dans l'expression de ses sentiments et de sa conception du monde.

8. Ensemble de caractères, de symboles et de règles qui permettent de les assembler, utilisé pour donner des instructions à un ordinateur.

https://www.larousse.fr/dictionnaires/francais/langage/46165

Introduction

8. Ensemble de caractères, de symboles et de règles qui permettent de les assembler, utilisé pour donner des instructions à un ordinateur.

https://www.larousse.fr/dictionnaires/francais/langage/46165

♦ *Langage de programmation.* ,,Langage préétabli utilisé pour écrire les programmes d'un ordinateur déterminé`` (MESS. *Télém.* 1979). *Langage symbolique.* ,,Langage de programmation utilisant des codes mnémoniques pour représenter les instructions machines`` (*Informat.* 1972). [Langage symbolique :] *code intermédiaire entre le langage machine et les langages externes (langues naturelles ou langages documentaires), et permettant au programmeur de communiquer aisément avec l'automate. C'est pourquoi les langages symboliques sont aussi appelés langages de programmation* (COYAUD, *Introd. ét. lang. docum.*,1966, p. 14).

https://www.cnrtl.fr/definition/langage

Introduction

8. Ensemble de caractères, de symboles et de règles qui permettent de les assembler, utilisé pour donner des instructions à un ordinateur.

https://www.larousse.fr/dictionnaires/francais/langage/46165

♦ *Langage de programmation.* ,,Langage préétabli utilisé pour écrire les programmes d'un ordinateur déterminé`` (MESS. *Télém.* 1979). *Langage symbolique.* ,,Langage de programmation utilisant des codes mnémoniques pour représenter les instructions machines`` (*Informat.* 1972). [Langage symbolique :] *code intermédiaire entre le langage machine et les langages externes (langues naturelles ou langages documentaires), et permettant au programmeur de communiquer aisément avec l'automate. C'est pourquoi les langages symboliques sont aussi appelés langages de programmation* (COYAUD, *Introd. ét. lang. docum.*,1966, p. 14).

https://www.cnrtl.fr/definition/langage

Langage de programmation

Un **langage de programmation** est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, de significations, mais aussi d'un environnement de traduction censé rendre sa syntaxe compréhensible par la machine [1,2].

https://fr.wikipedia.org/wiki/Langage_de_programmation

Introduction

Langage de programmation

Un **langage de programmation** est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, de significations, mais aussi d'un environnement de traduction censé rendre sa syntaxe compréhensible par la machine[1,2].

https://fr.wikipedia.org/wiki/Langage_de_programmation

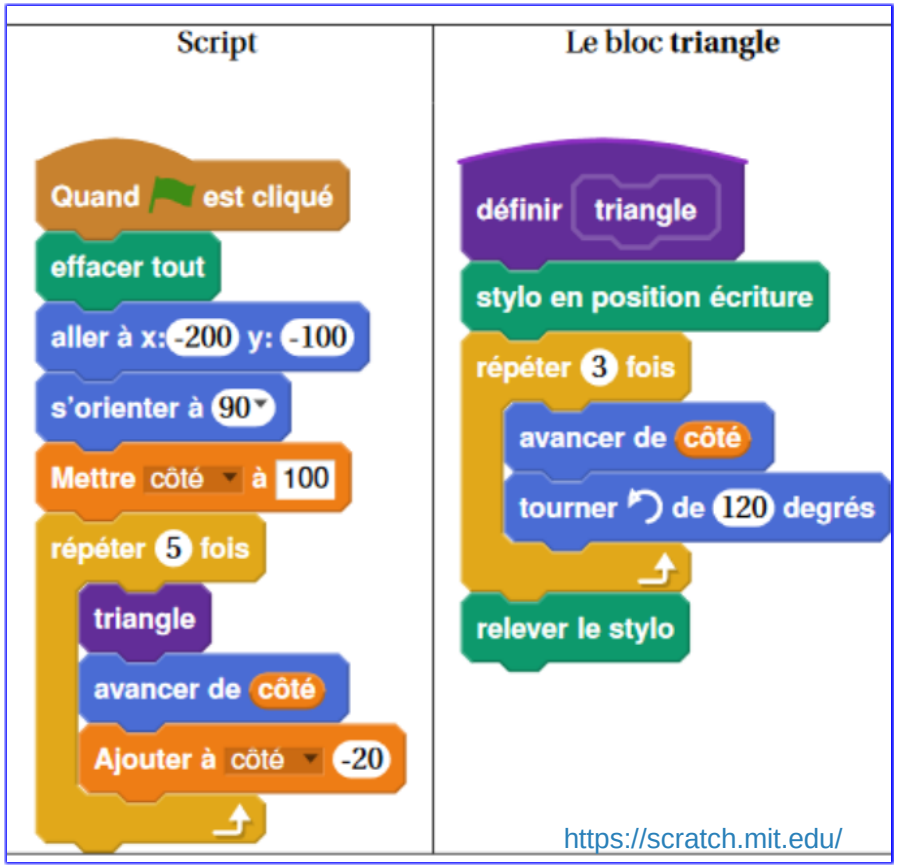
-
-
-

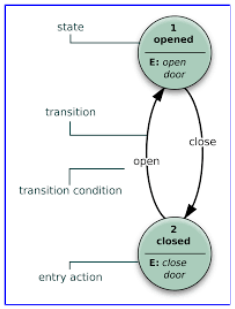
Introduction

Langage de programmation

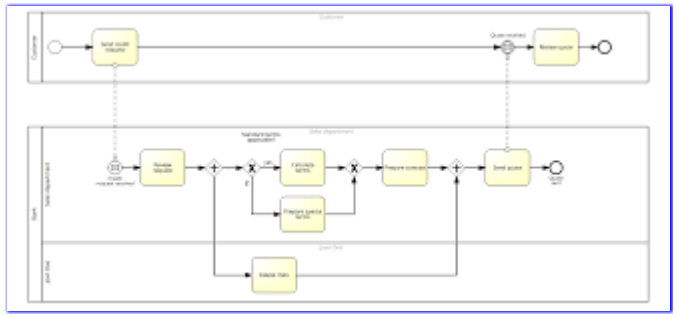
Un **langage de programmation** est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, de significations, mais aussi d'un environnement de traduction censé rendre sa syntaxe compréhensible par la machine [1, 2].

https://fr.wikipedia.org/wiki/Langage_de_programmation





State charts



BPMN

Quid de scratch et autres langages graphiques ?

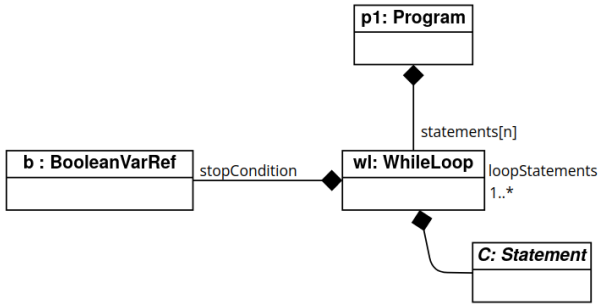
Introduction

-
-
  -
-

Programme :

```
while (b)
do
    C ;
done
```

A program p1 in a textual concrete syntax



Domain Model instance corresponding to p1

Exécuter C de manière répétée (et séquentielle), aussi longtemps que l'expression *b* est vraie.

- 1) évaluer l'expression *b.*
  - si *b* == vrai, exécuter C et retourner à 1)
  - si *b* == faux, sortir.

Behavioral semantics of p1

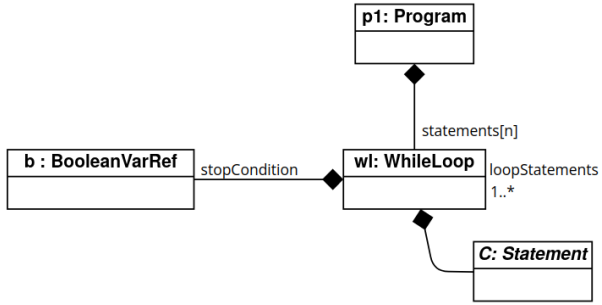
Syntaxe abstraite

Modèle d'une syntaxe abstraite:

Programme :

```
while (b)
do
        C ;
done
```

A program p1 in a textual concrete syntax



Domain Model instance corresponding to p1

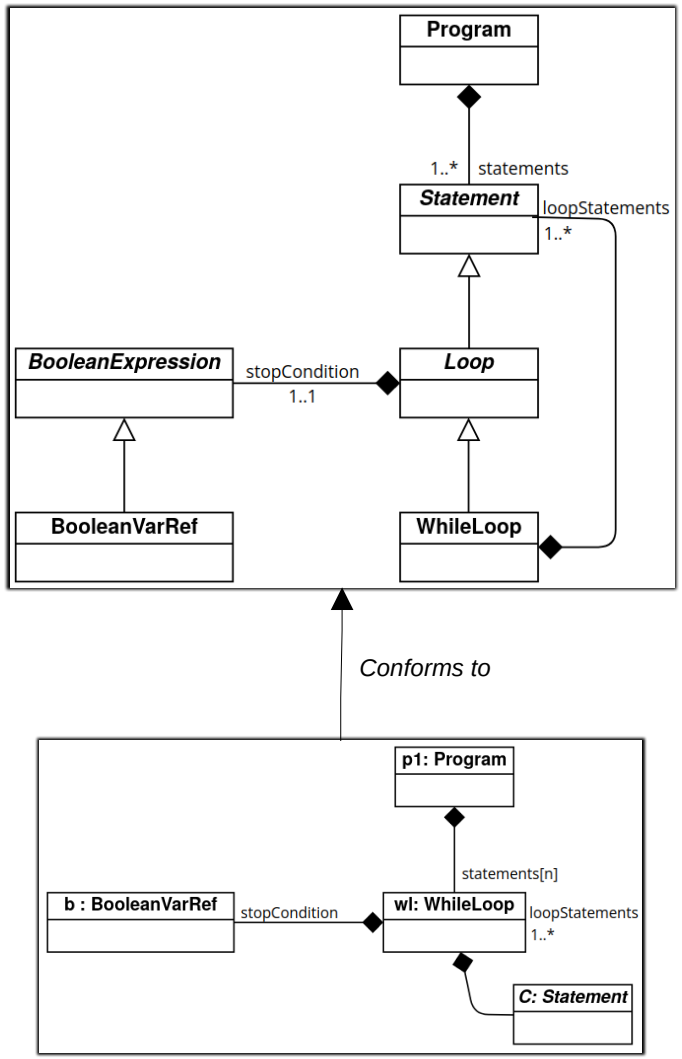
Exécuter C de manière répétée (et séquentielle), aussi longtemps que l'expression *b* est vraie.

- 1) évaluer l'expression *b.*
  - si *b* == vrai, exécuter C et retourner à 1)
  - si *b* == faux, sortir.

Behavioral semantics of p1

Syntaxe abstraite

Modèle d'une syntaxe abstraite:



Domain Model instance corresponding to p1

C'est la définition (partielle) d'un domain model, ici implémentée par un métamodèle

modèle

Programme :

```
while (b)
do
        C ;
done
```

A program p1 in a textual concrete syntax

Exécuter C de manière répétée (et séquentielle), aussi longtemps que l'expression *b* est vraie.

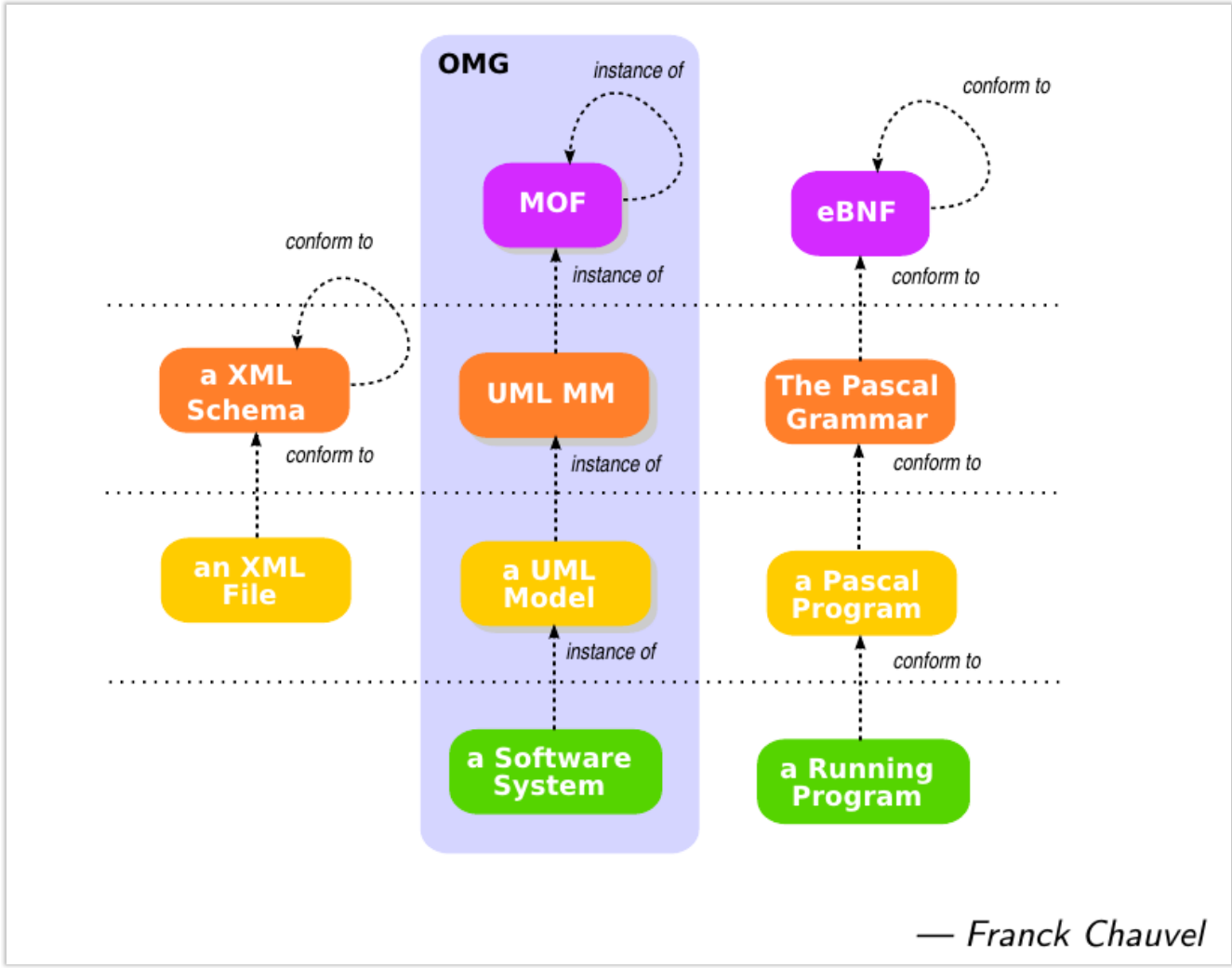
- 1) évaluer l'expression *b.*
  - si *b* == vrai, exécuter C et retourner à 1
  - si *b* == faux, sortir.

Behavioral semantics of p1

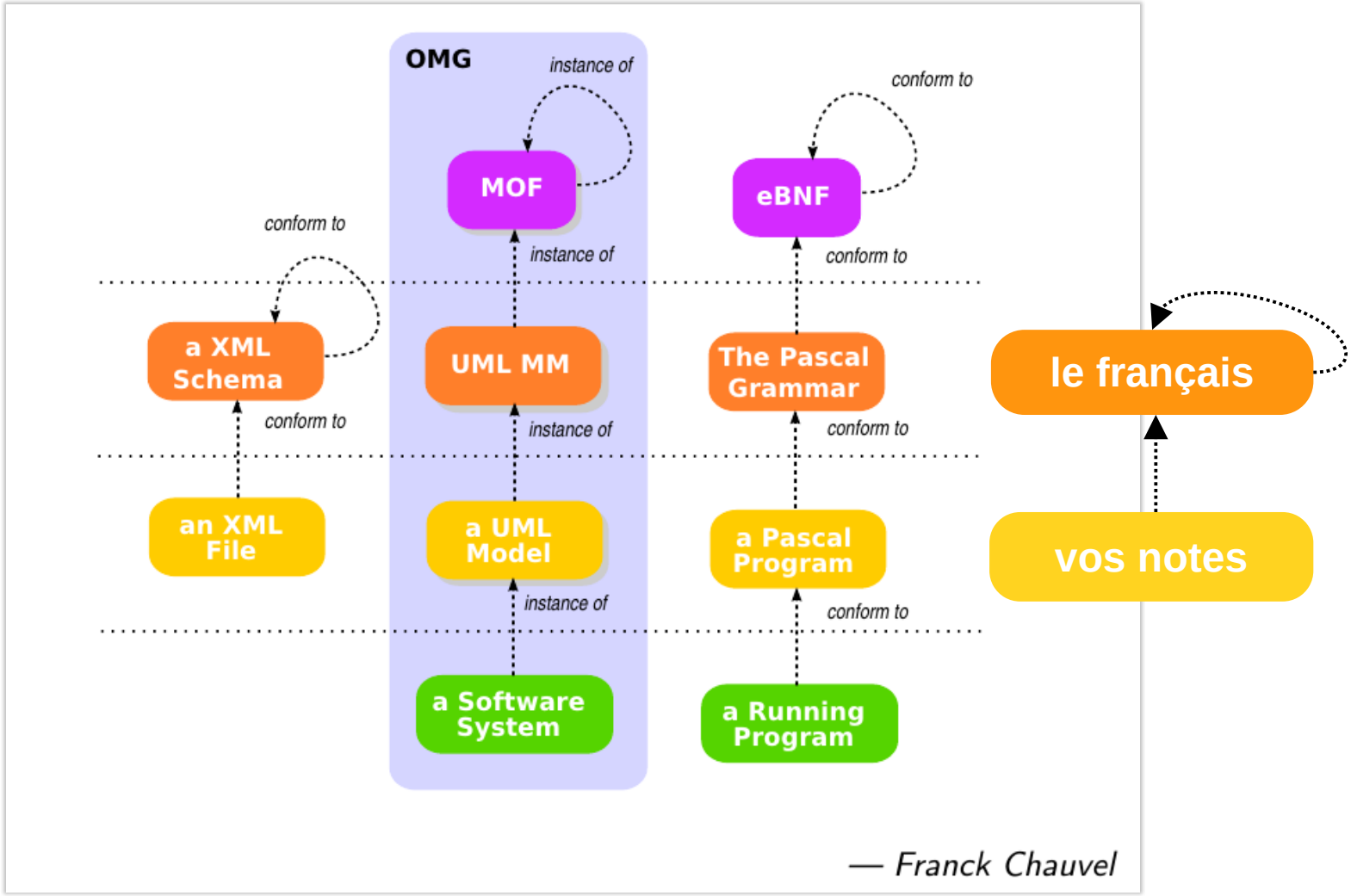
Modèle et métamodèle

-
  -
  -
  -
-
  -
  -

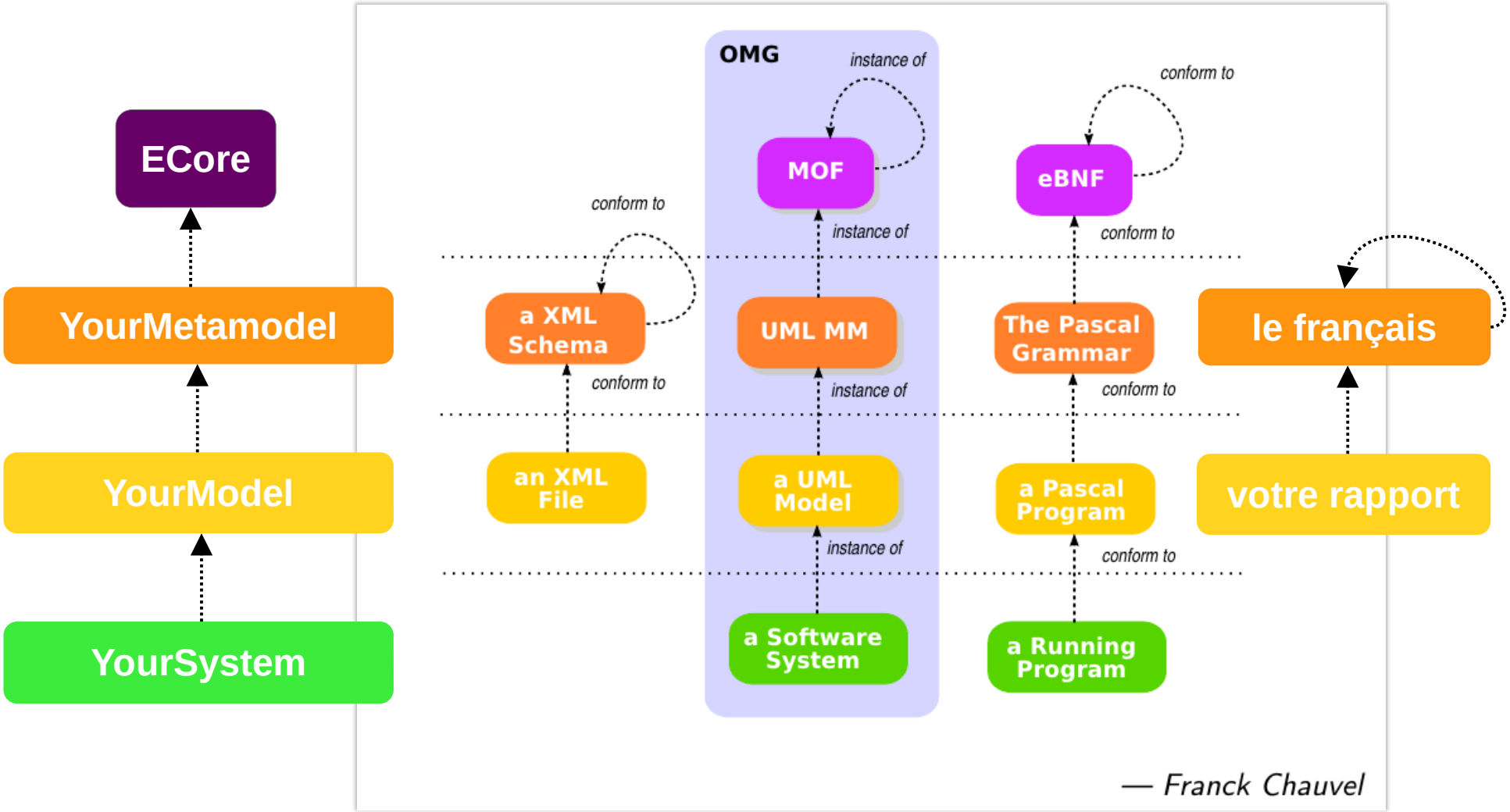
Modèle et métamodèle



Modèle et métamodèle

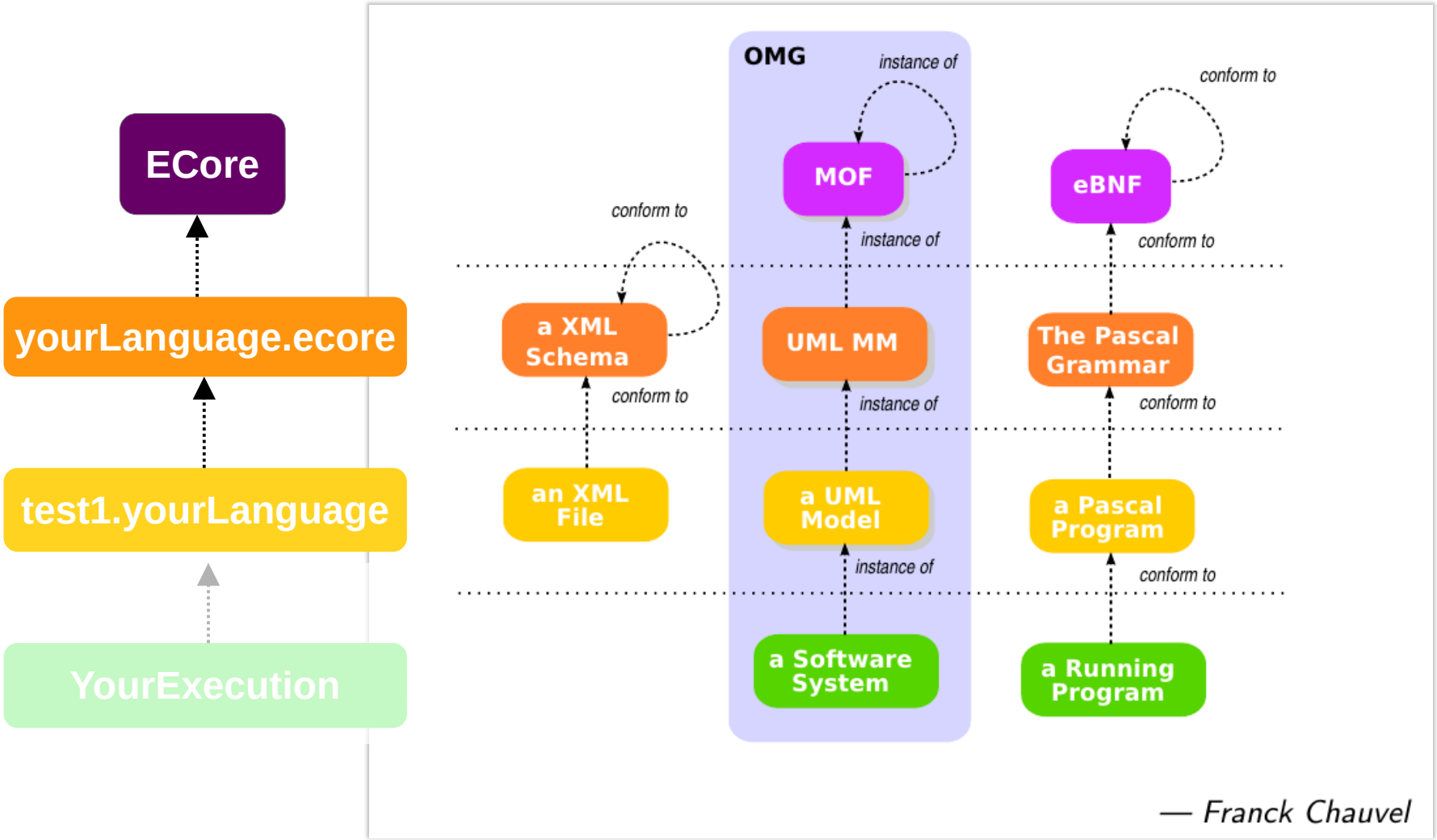


Eclipse Modeling Framework

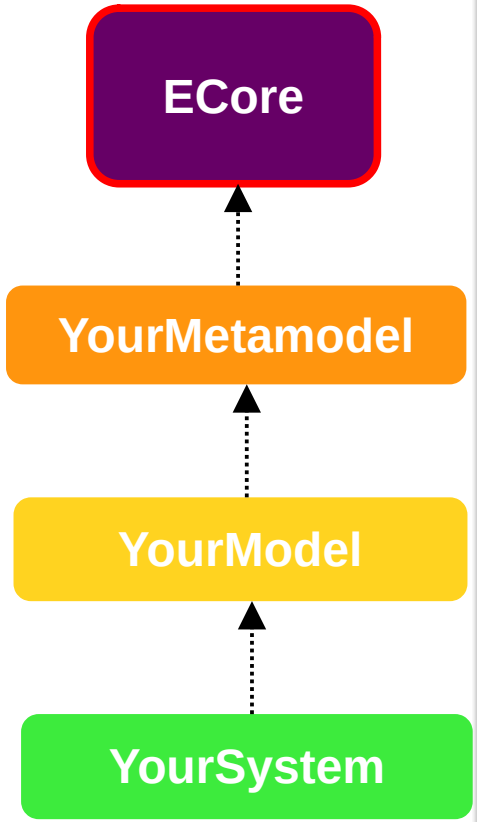


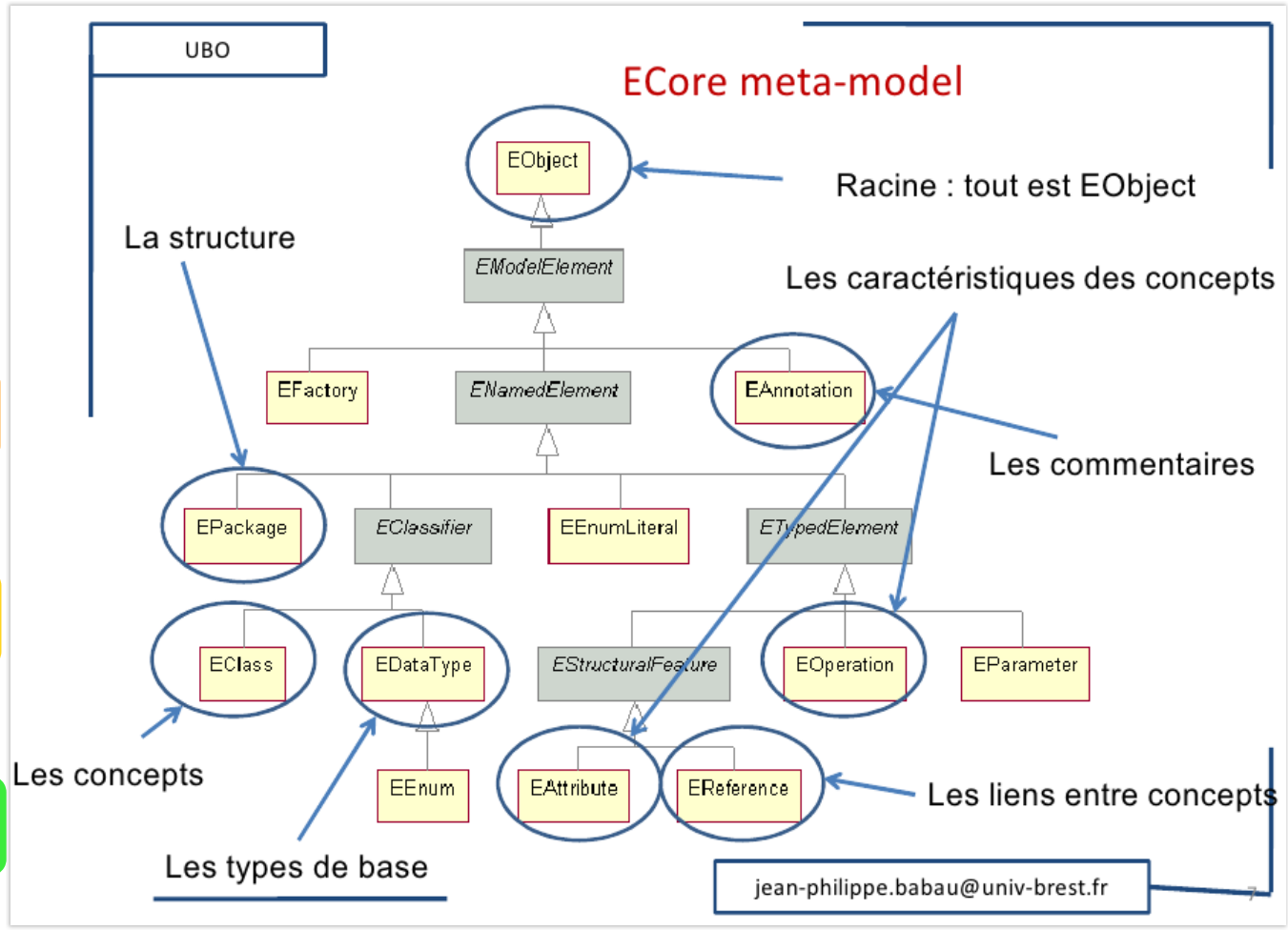
Eclipse Modeling Framework

Dans quelques heures ?

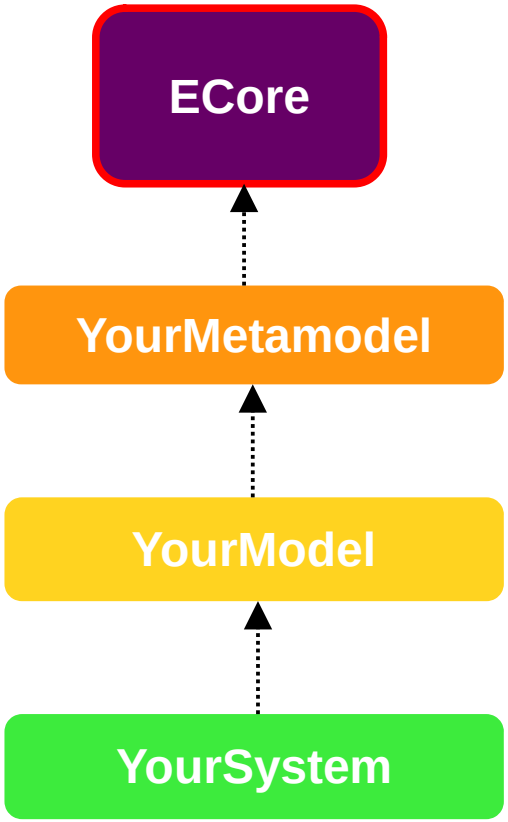


Eclipse Modeling Framework





Eclipse Modeling Framework



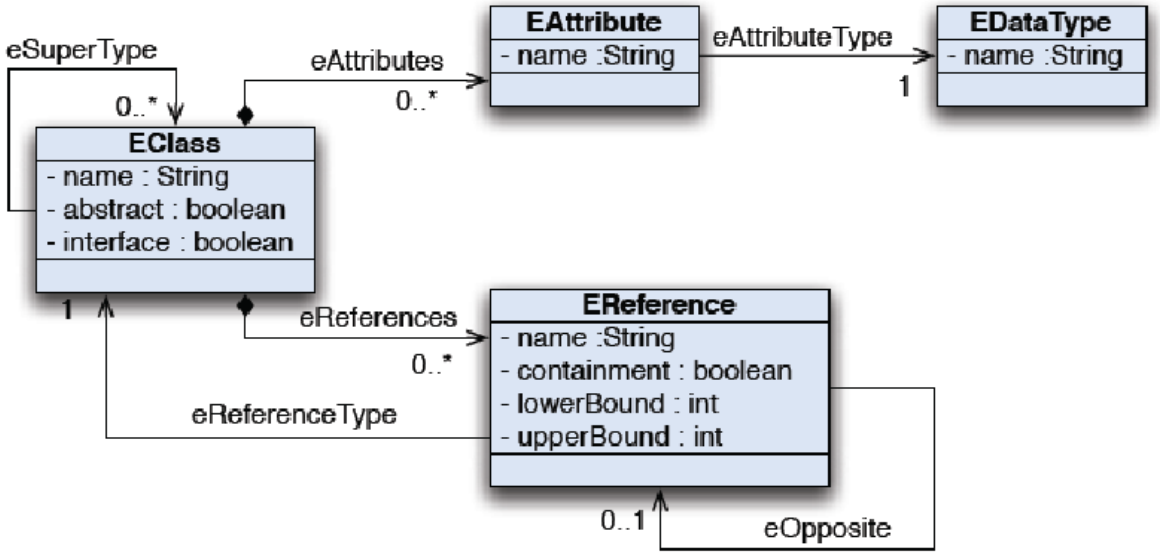
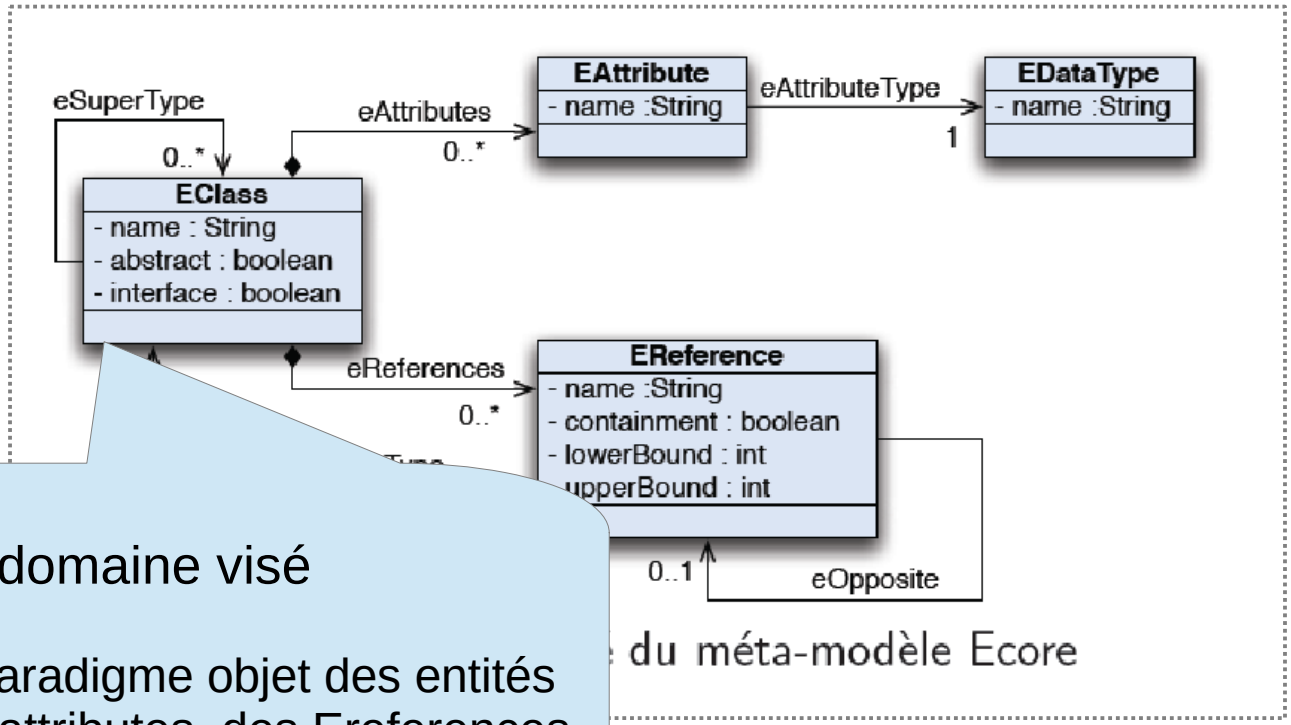


FIG.: Extrait simplifié du méta-modèle Ecore – Sébastien Mosser

Eclipse Modeling Framework

ECore

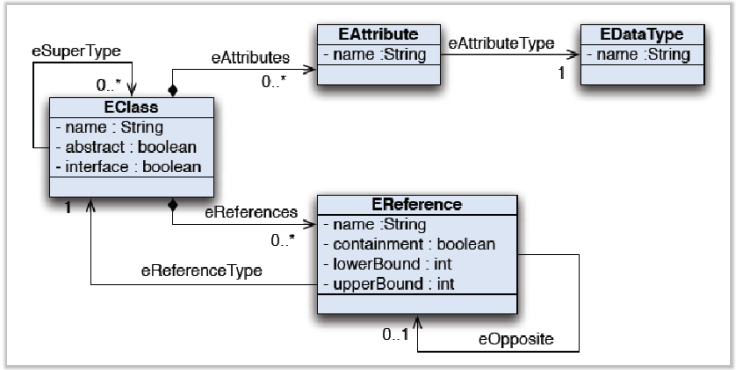


Modélise un concept du domaine visé

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

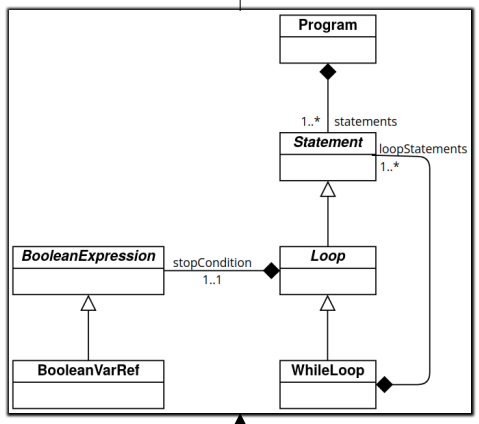
Syntaxe abstraite

Métamodèle d'une syntaxe abstraite:



Conforms to

Modèle d'une syntaxe abstraite:



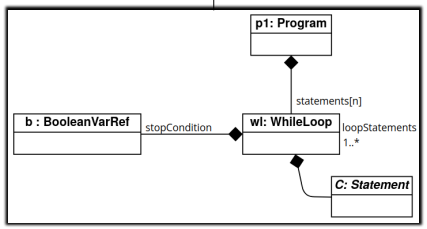
Conforms to

C'est la définition (partielle) d'un domain model, ici implémentée par un métamodèle, dont le modèle est ecore

programme

```
while (b)
do
        C ;
done
```

A program p1 in a textual concrete syntax



Domain Model instance corresponding to p1

modèle

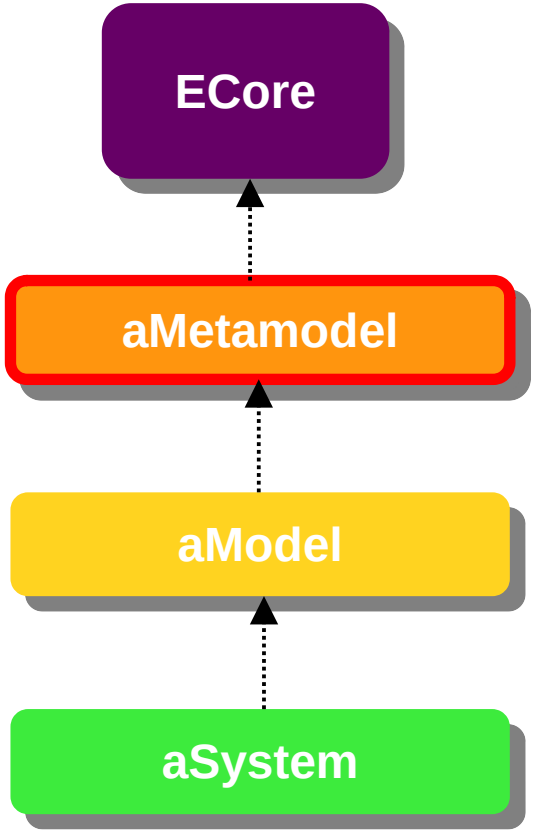
Exécuter C de manière répétée (et séquentielle), aussi longtemps que l'expression *b* est vraie.

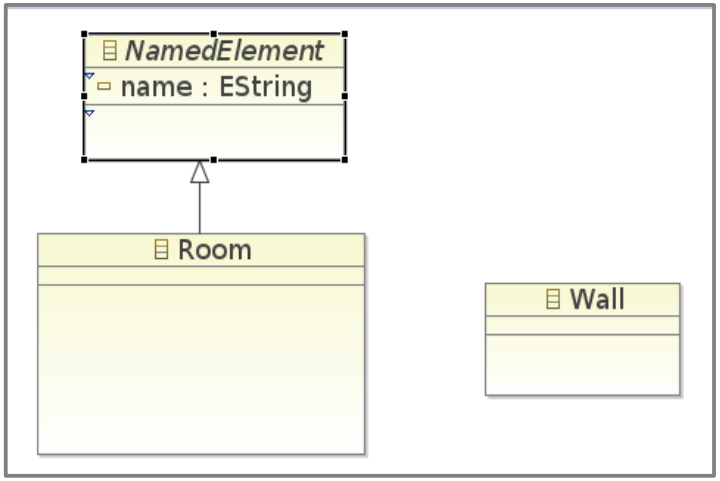
- 1) évaluer l'expression *b.*
  - si *b* == vrai, exécuter C et retourner à 1
  - si *b* == faux, sortir.

Behavioral semantics of p1

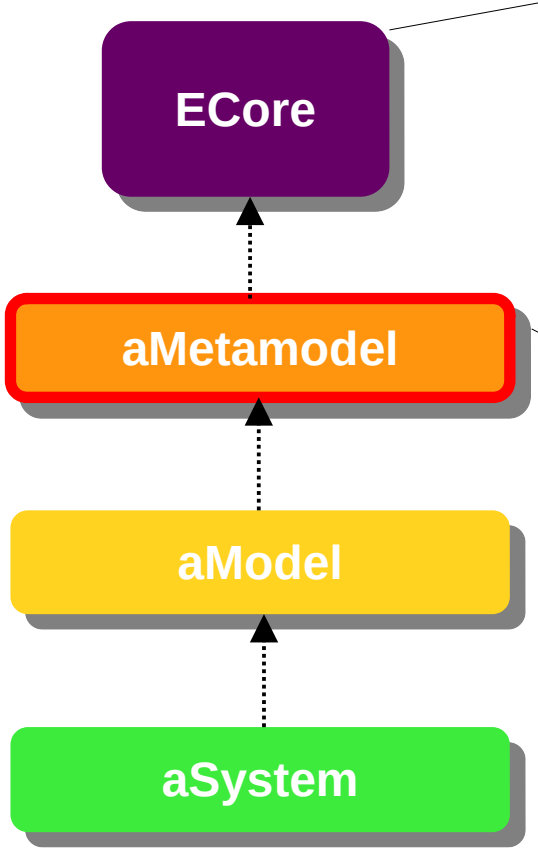
Exemple (très) jouet

Eclipse Modeling Framework





Eclipse Modeling Framework



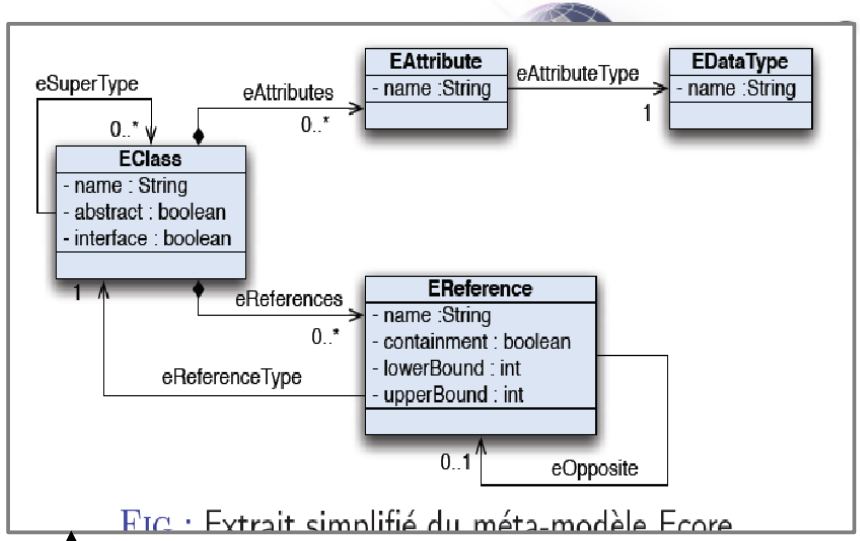
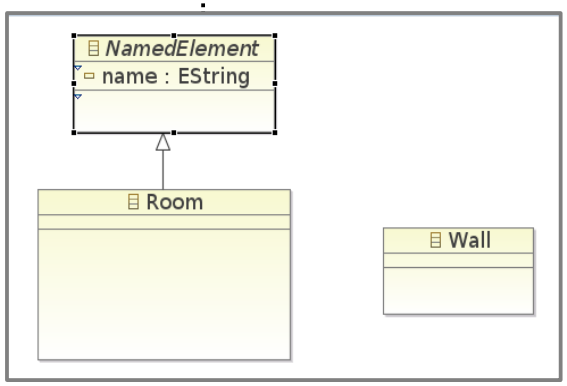
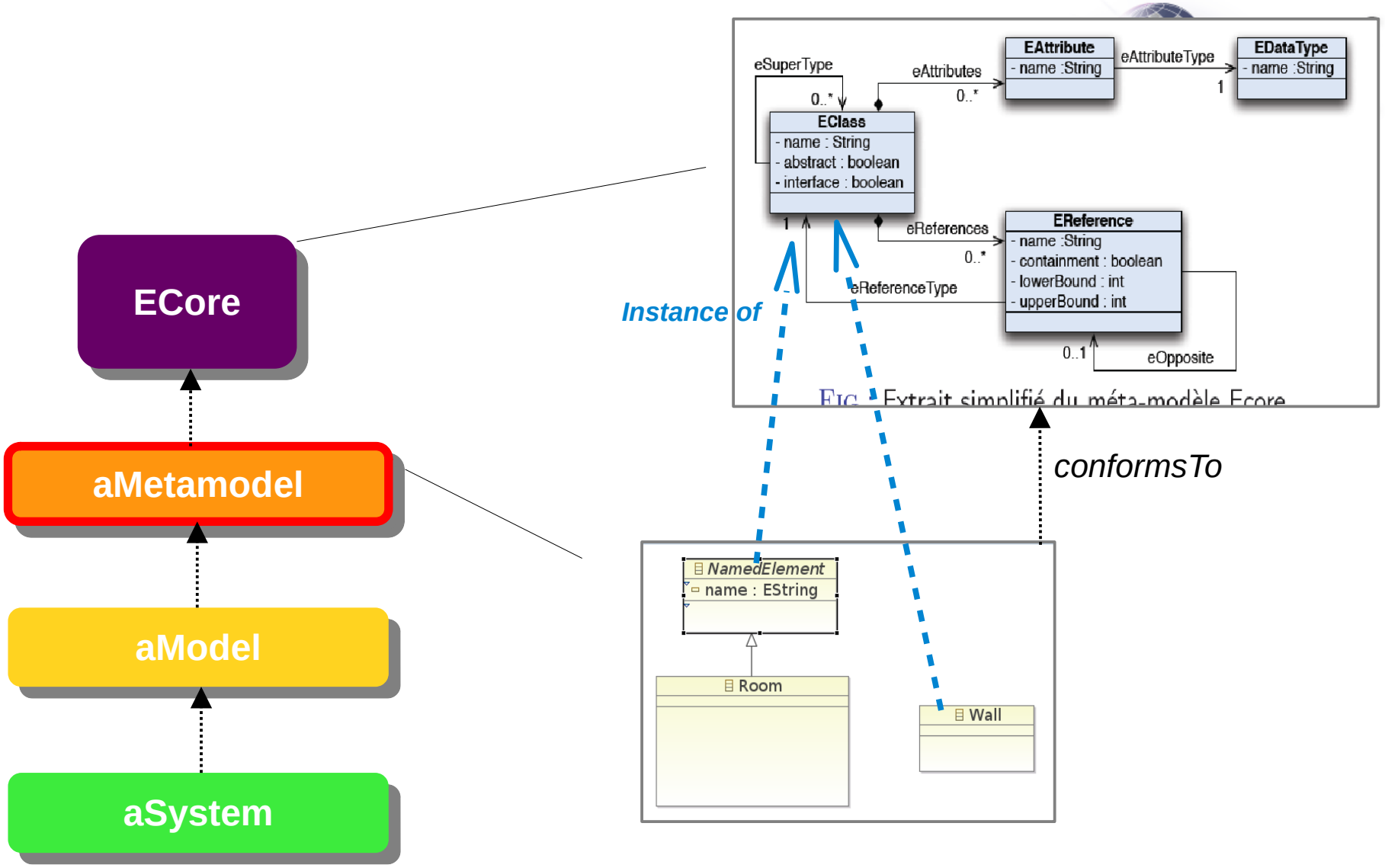


FIG. : Extrait simplifié du méta-modèle Ecore

conformsTo



Eclipse Modeling Framework



Eclipse Modeling Framework

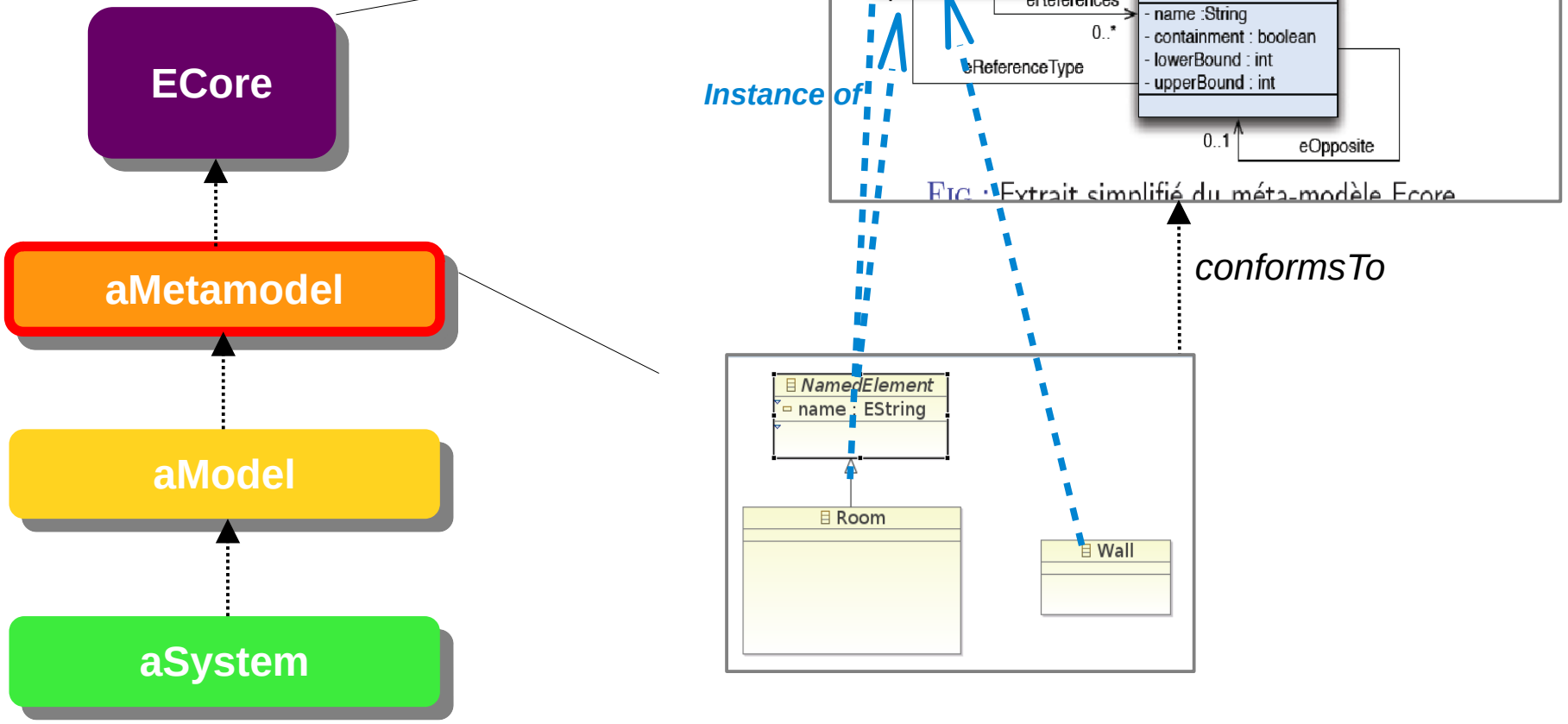


Fig. : Extrait simplifié du méta-modèle Ecore

Eclipse Modeling Framework

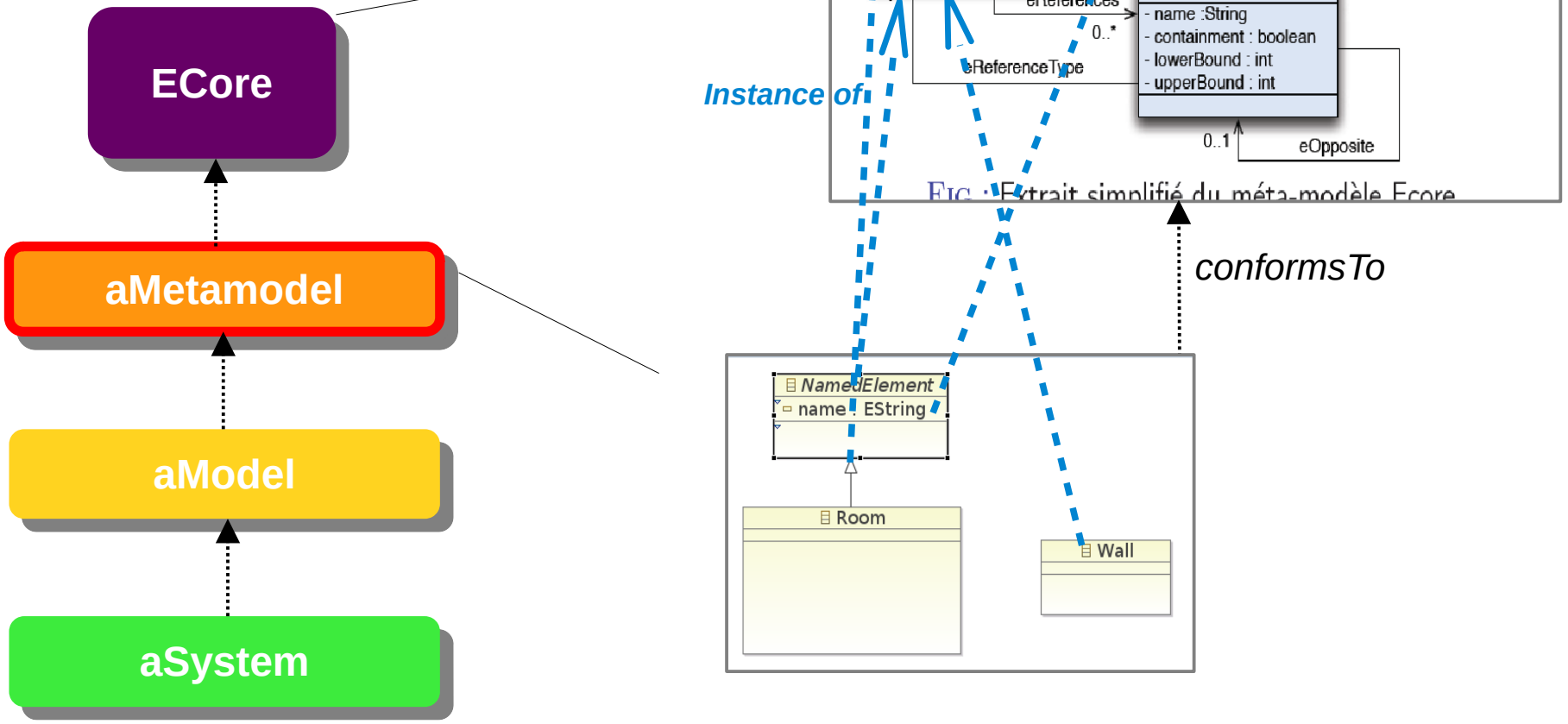
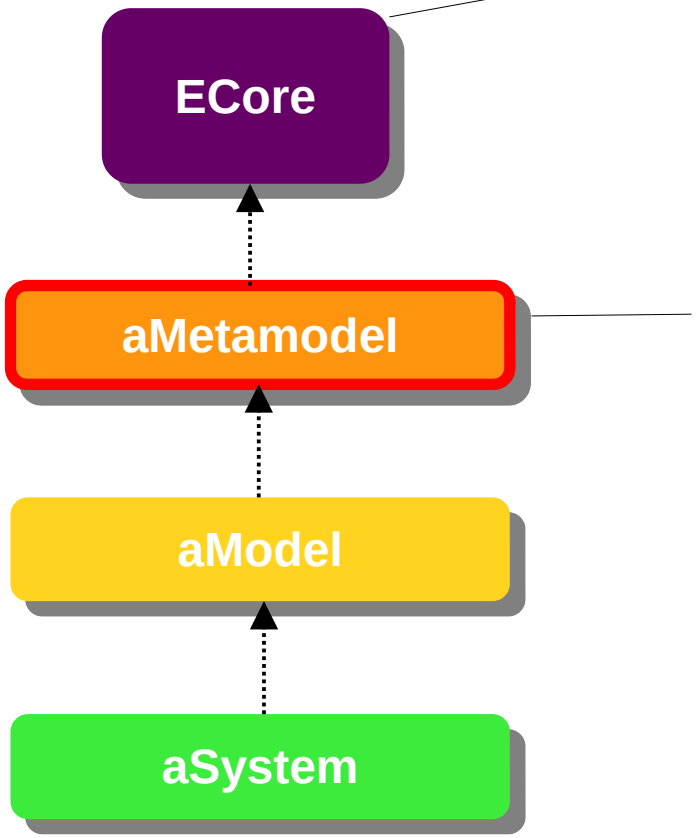
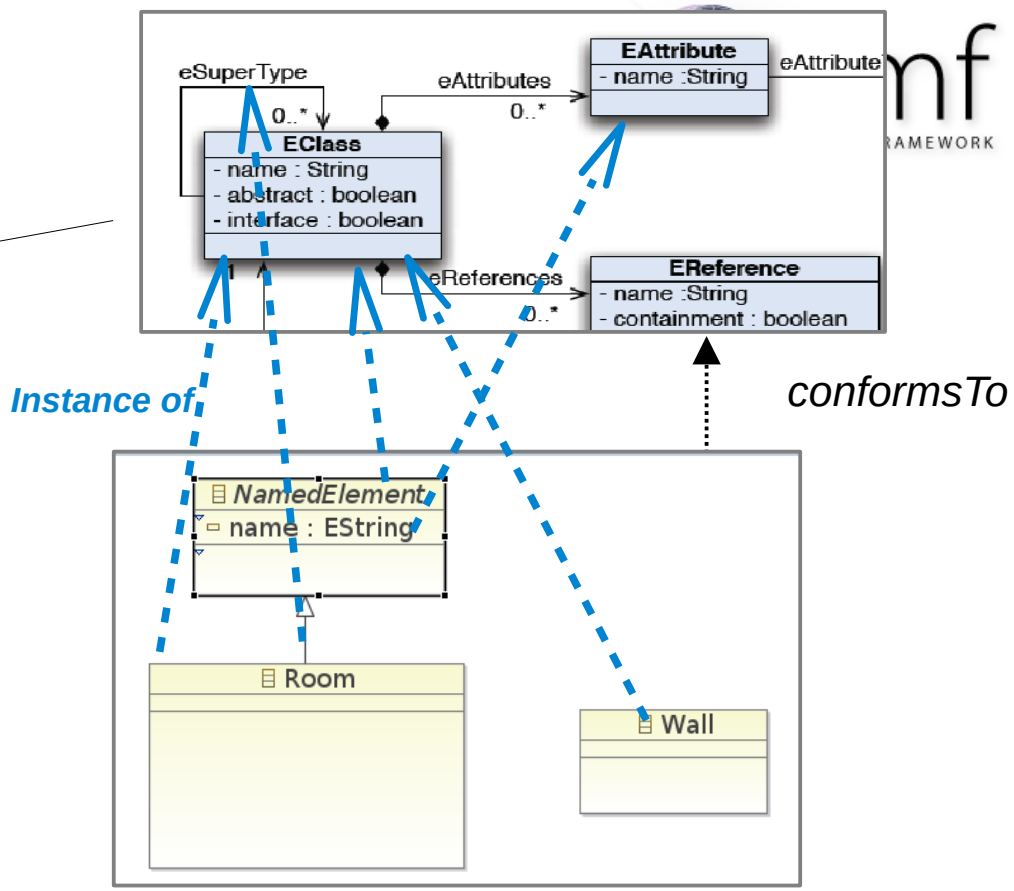


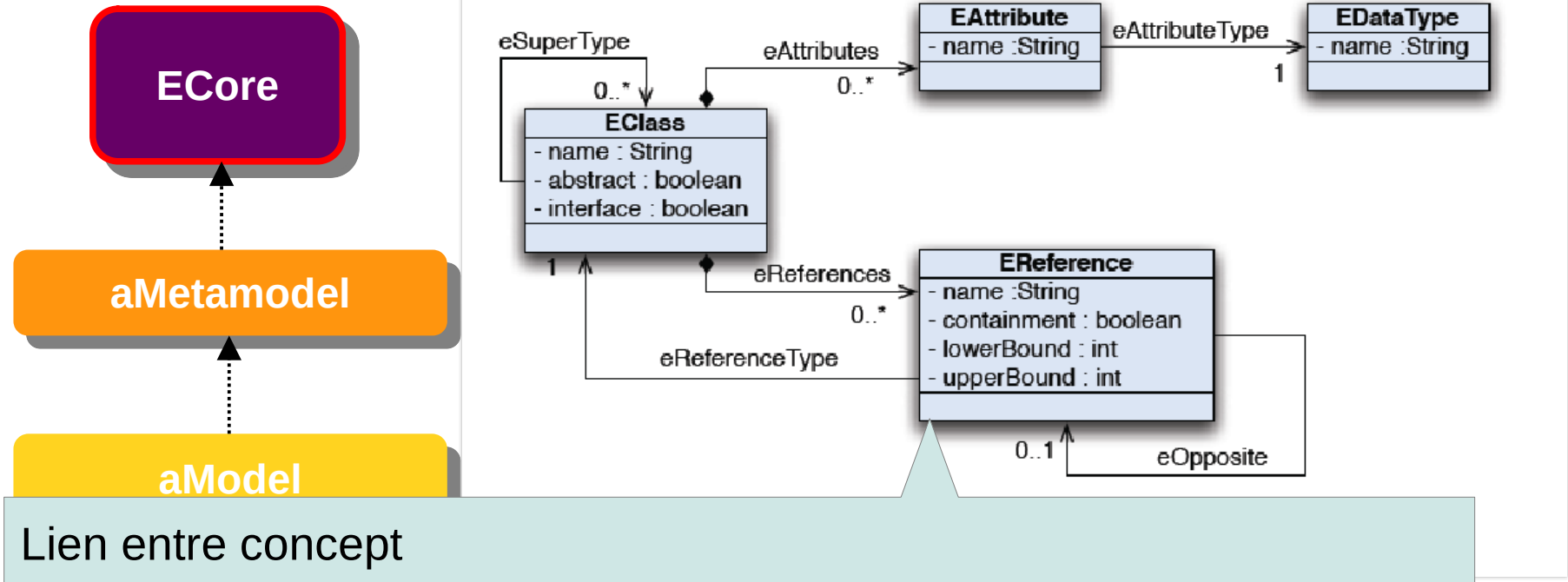
Fig. : Extrait simplifié du méta-modèle Ecore

Eclipse Modeling Framework





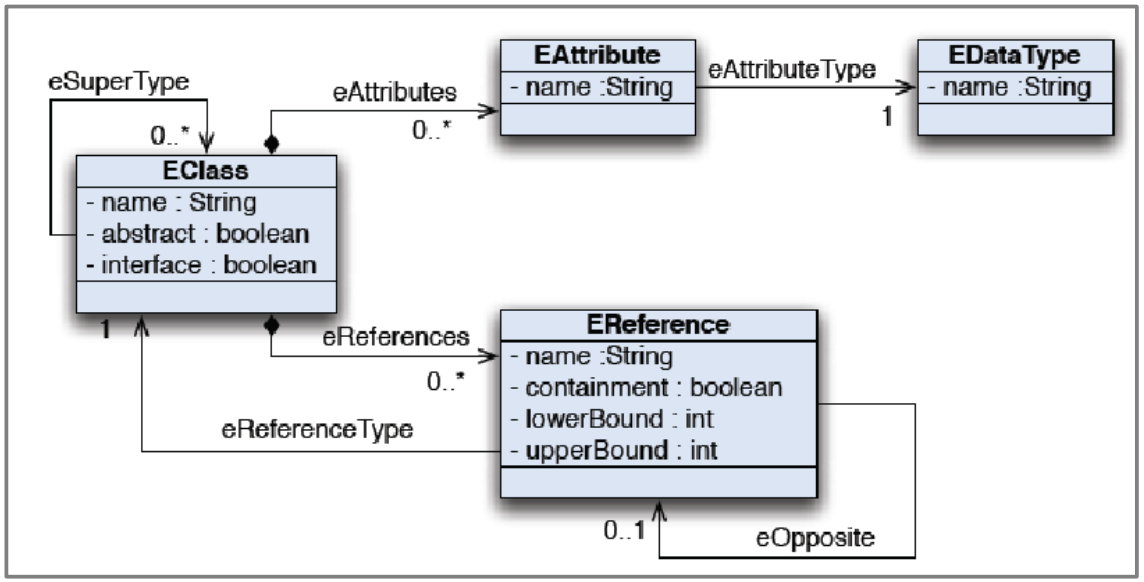
Eclipse Modeling Framework

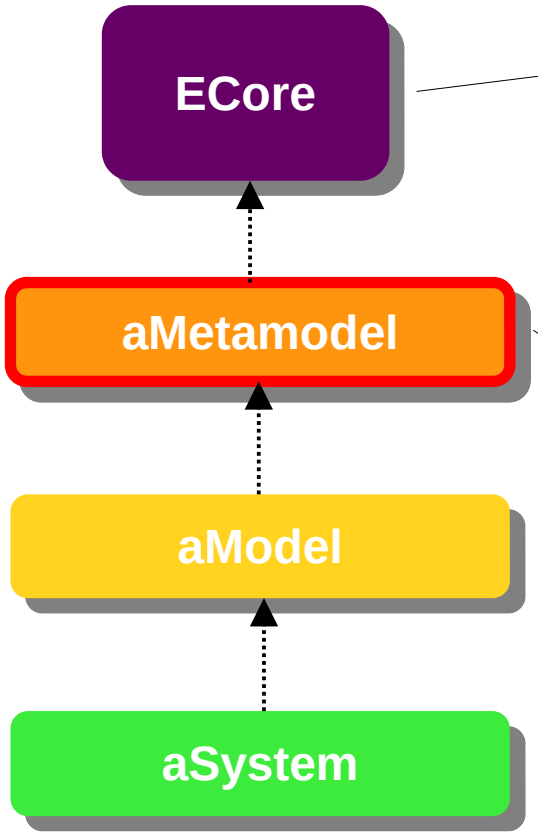


Lien entre concept

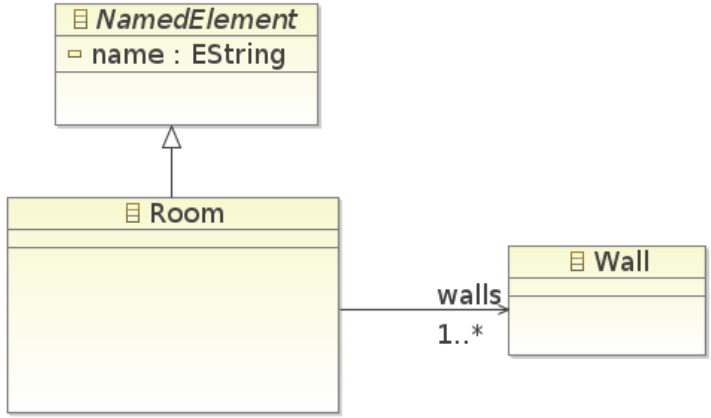
- Si “containment” est vrai alors equivalent à EAttribute pour les types non primitifs, sinon association simple.
- Caractérisé par un nom et une arité
- Eopposite assure que les instance d'un côté et de l'autre “correspondent”

Eclipse Modeling Framework

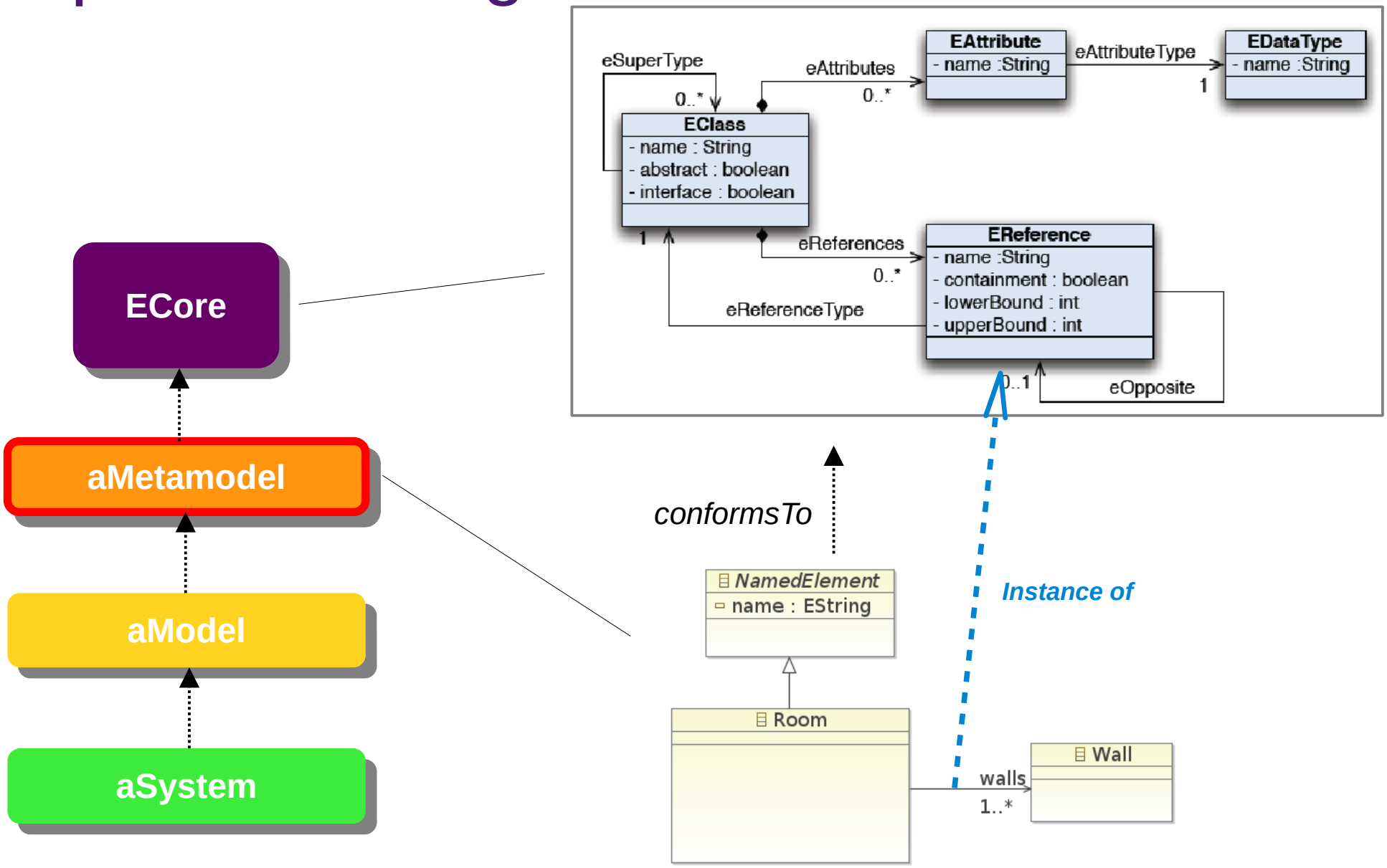




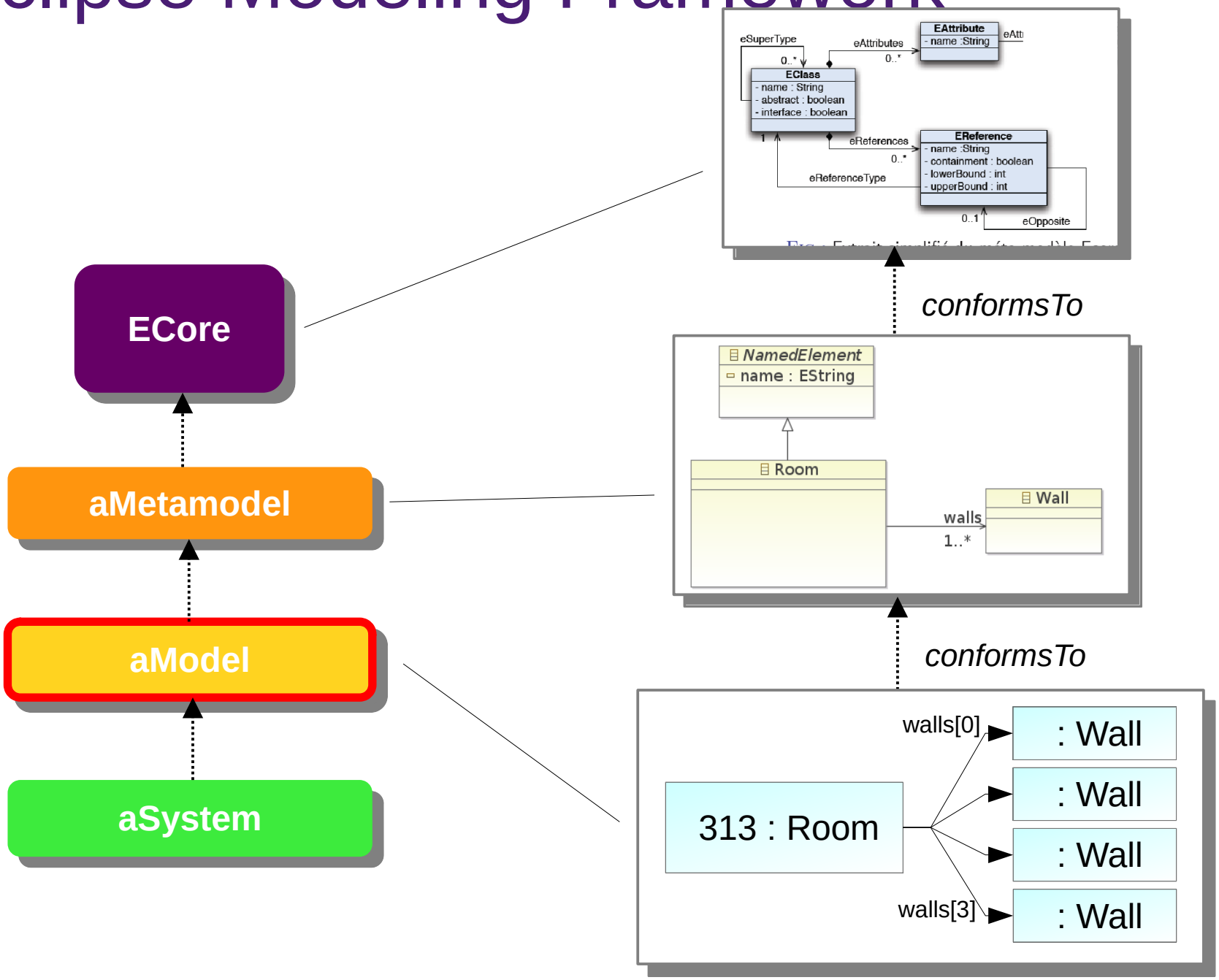
conformsTo



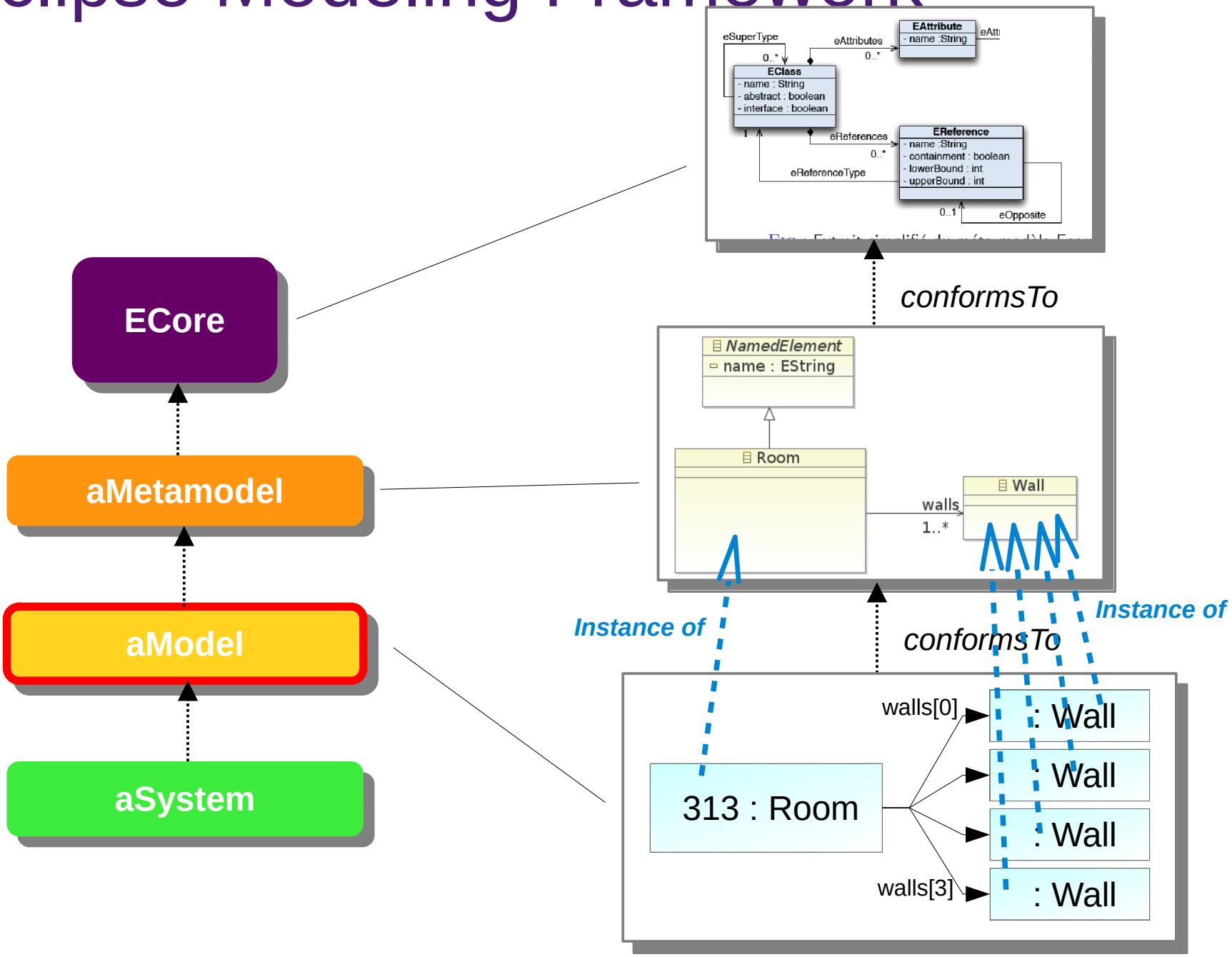
Eclipse Modeling Framework



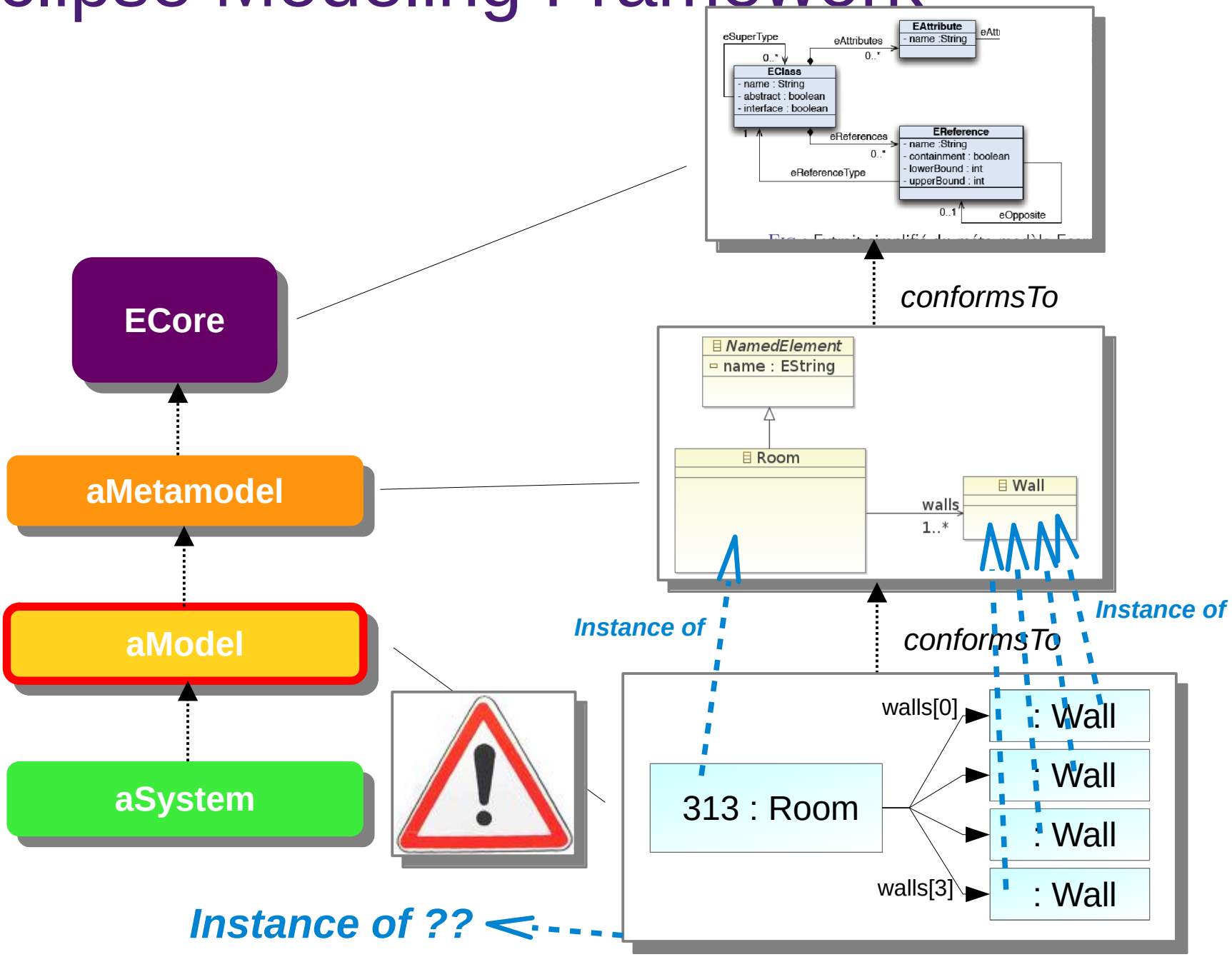
Eclipse Modeling Framework



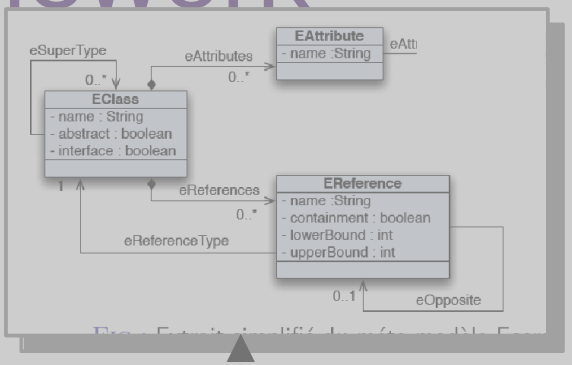
Eclipse Modeling Framework



Eclipse Modeling Framework



Eclipse Modeling Framework



ECore

aMetamo

aMod

aSystem

Contraintes d'un métamodèle:

- Il doit toujours y avoir (au moins) une EClass racine !
- À partir de cette racine, il doit exister un chemin de “containment” vers toutes les classes concrètes.

ance of

313 : Room

: Wall

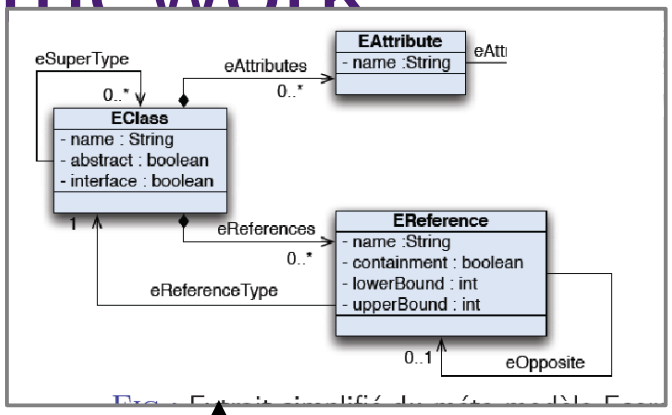
: Wall

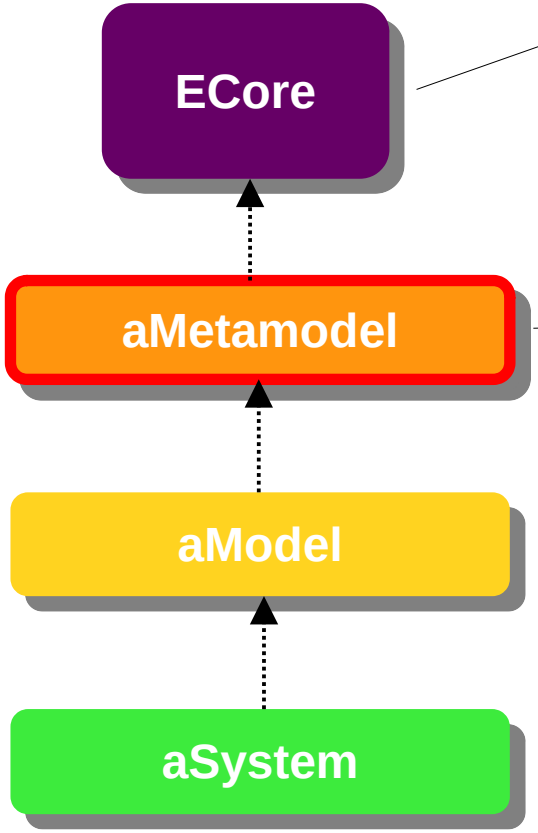
: Wall

walls[3]

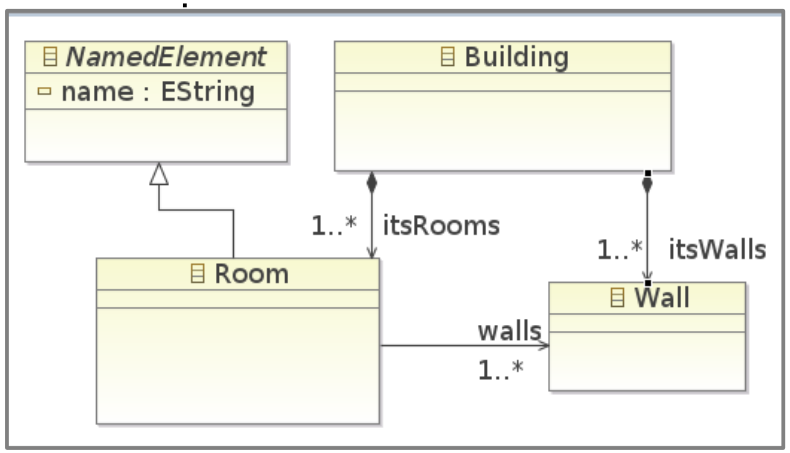
Instance of ??

Eclipse Modeling Framework

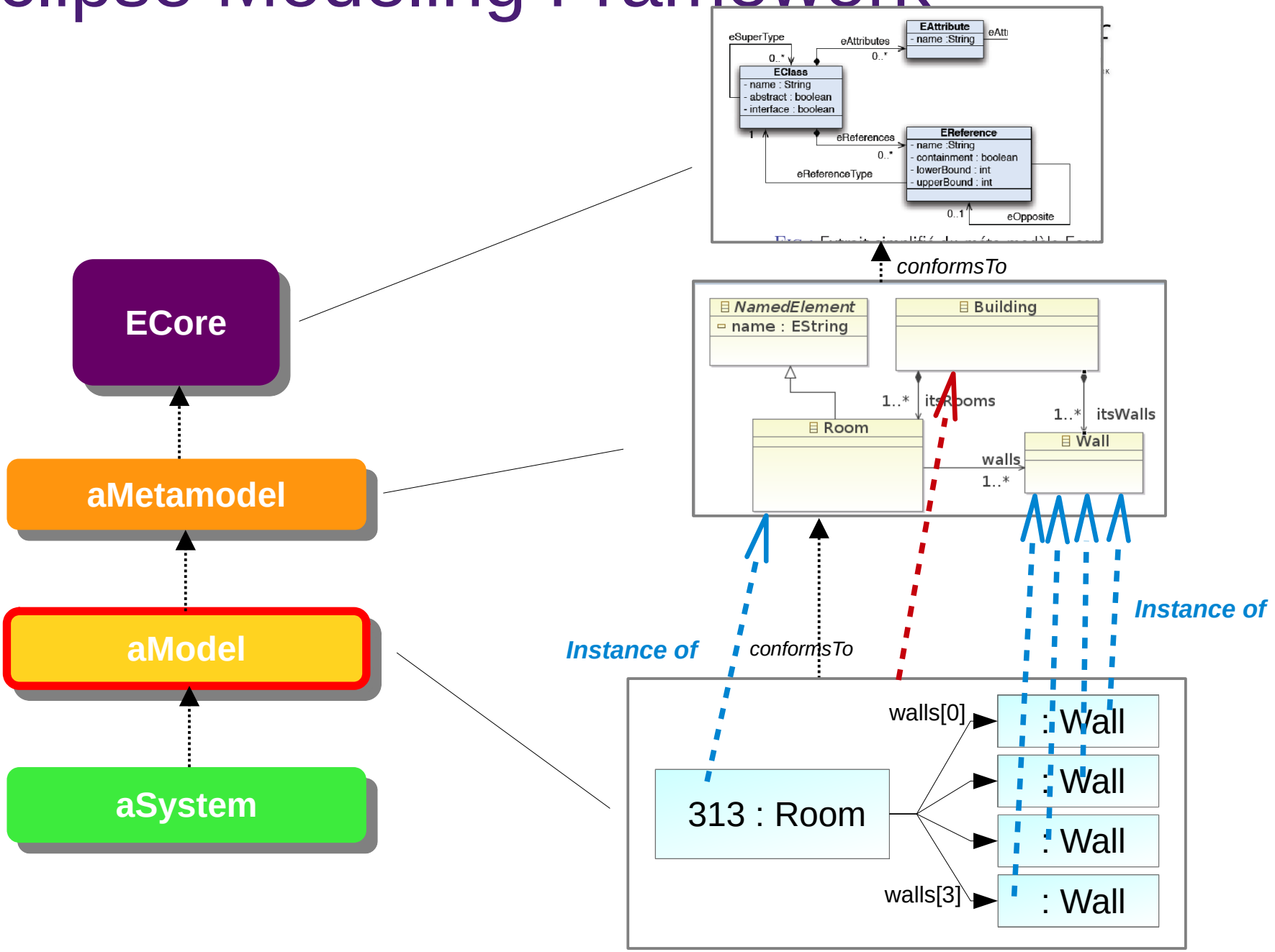




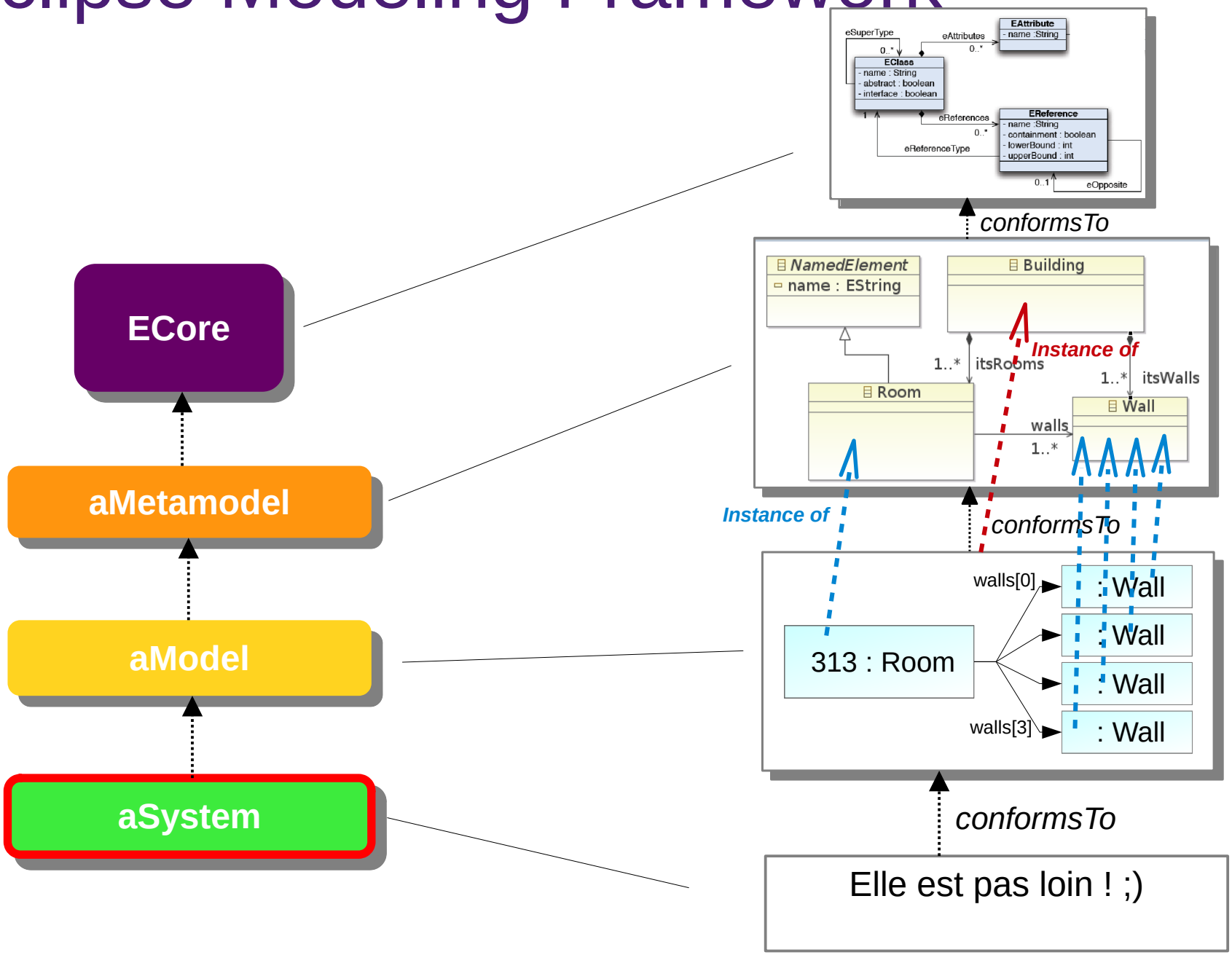
conformsTo



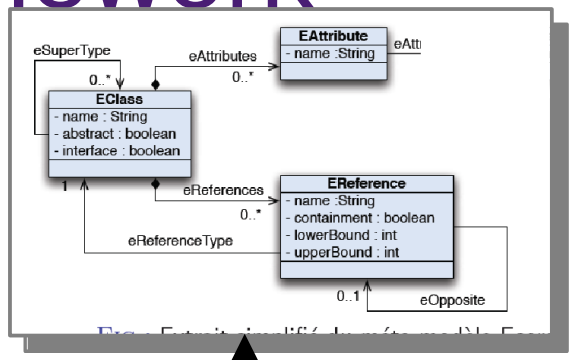
Eclipse Modeling Framework



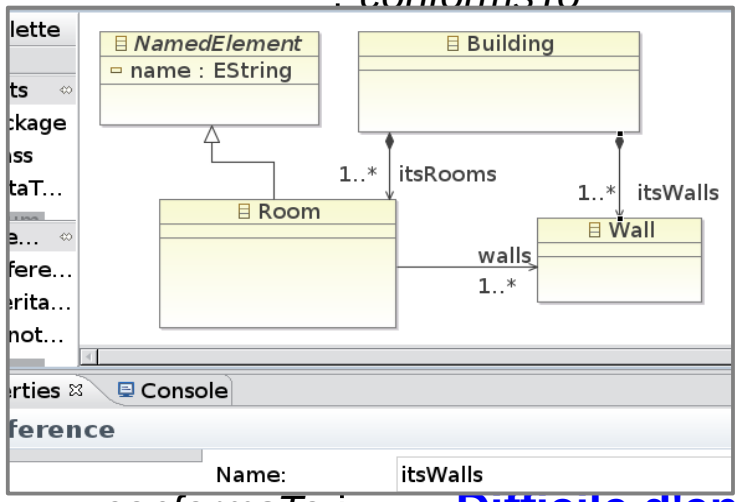
Eclipse Modeling Framework



Eclipse Modeling Framework

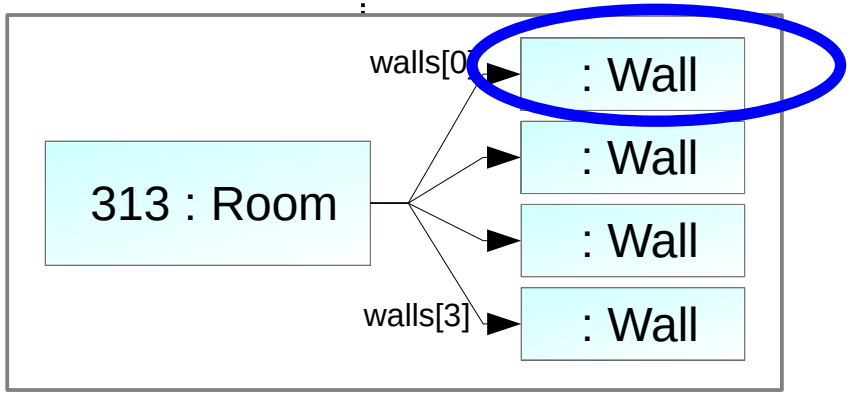


conformsTo

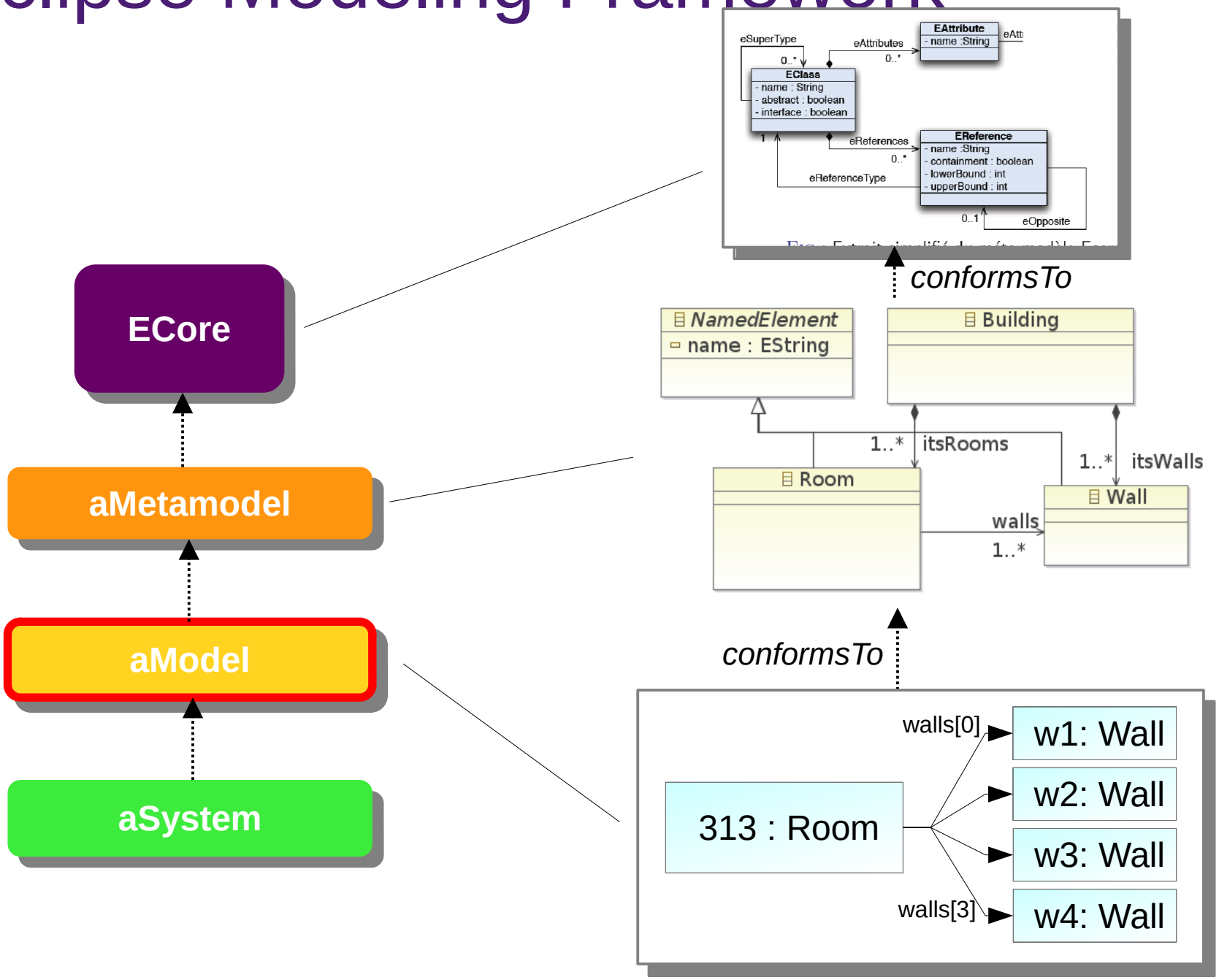


conformsTo

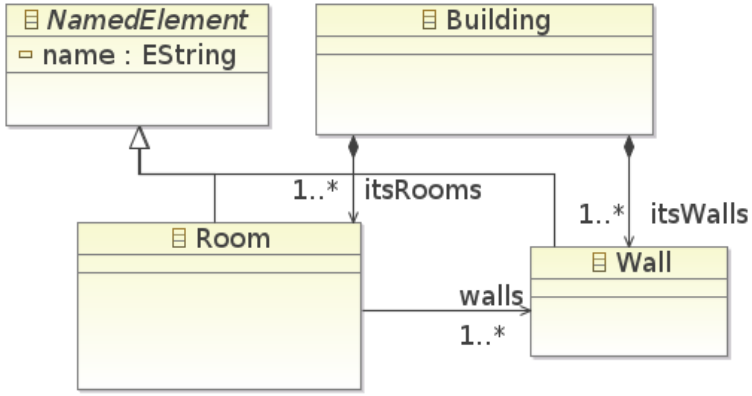
Difficile d'en parler....



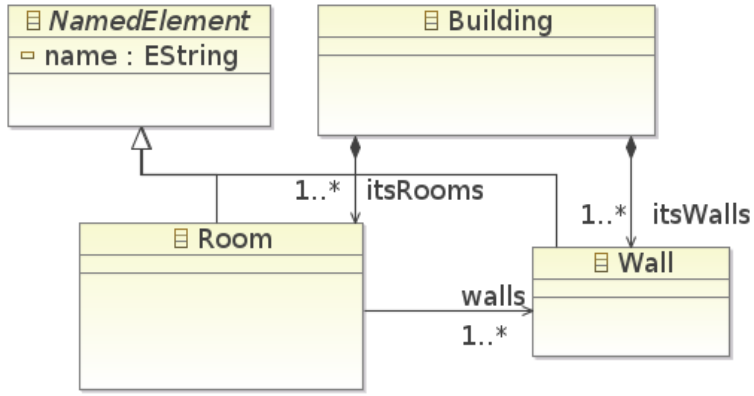
Eclipse Modeling Framework



Avantages de la méta-modélisation



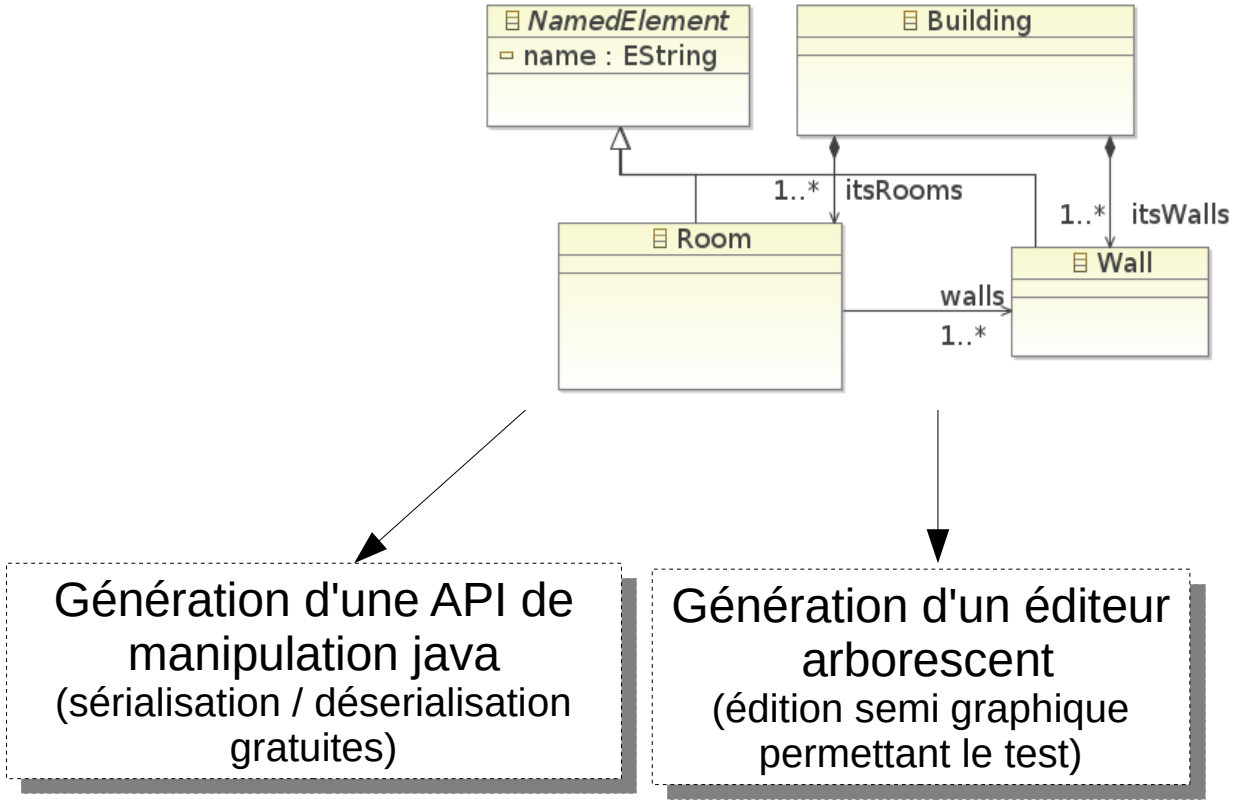
Avantages de la méta-modélisation



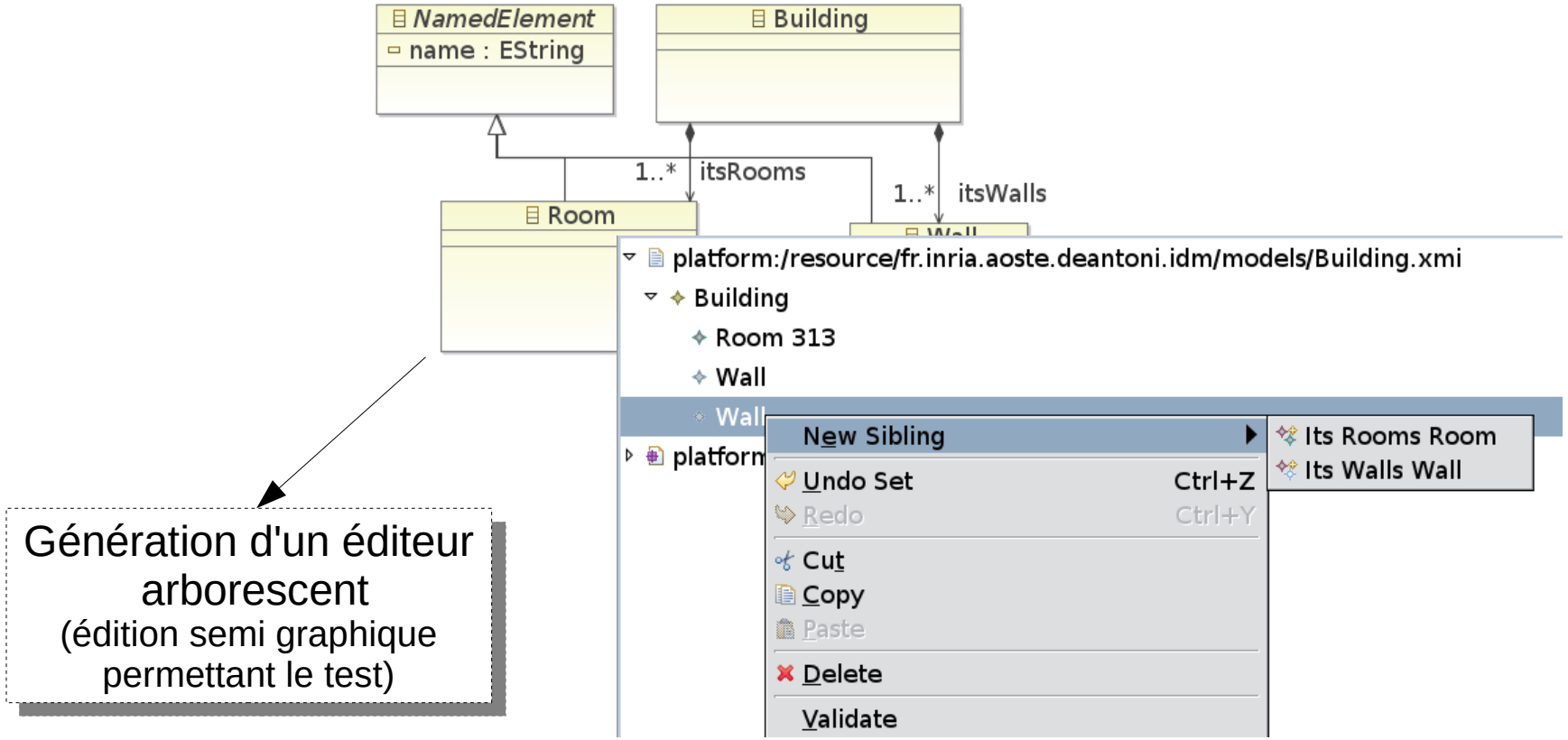
Génération d'une API de manipulation java
(sérialisation / déserialisation gratuites)

Manipulation aisée des modèles en java, gain de temps pour sauvegarder les modèles

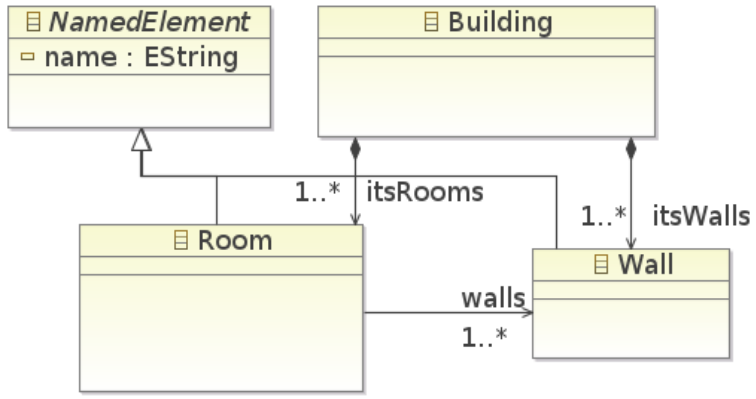
Avantages de la méta-modélisation



Avantages de la méta-modélisation



Avantages de la méta-modélisation

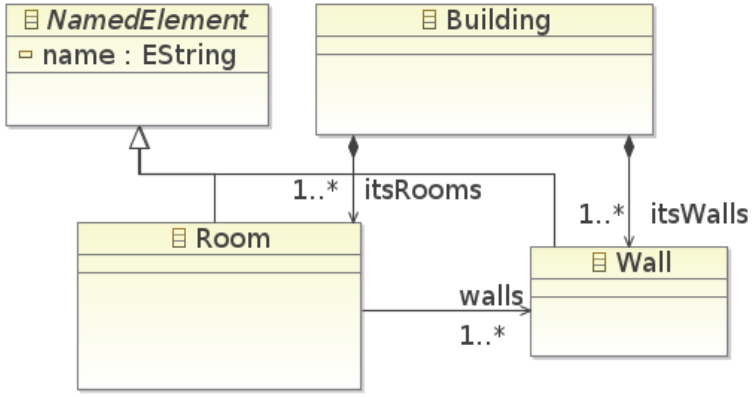


Génération d'une API de manipulation java
(sérialisation / déserialisation gratuites)

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 de la méta-modélisation

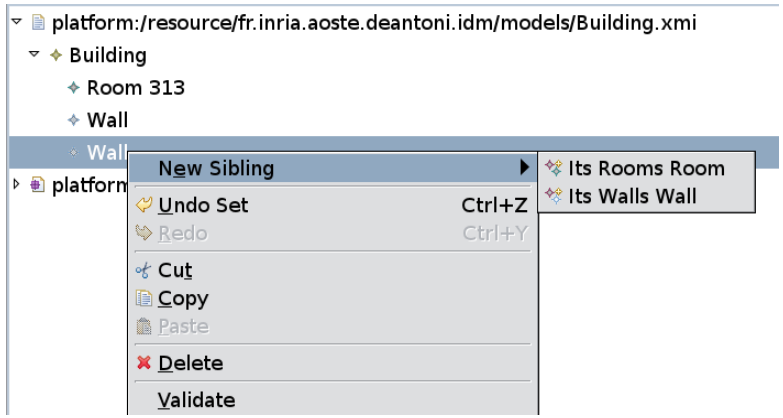


Outil de création d'une syntaxe concrète...

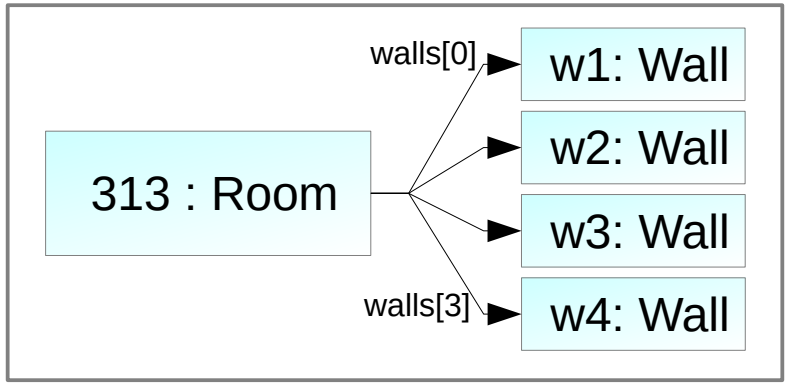
Accès aux outils basés sur les métamodèles:

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

Syntaxes concrètes et abstraites (caricature)

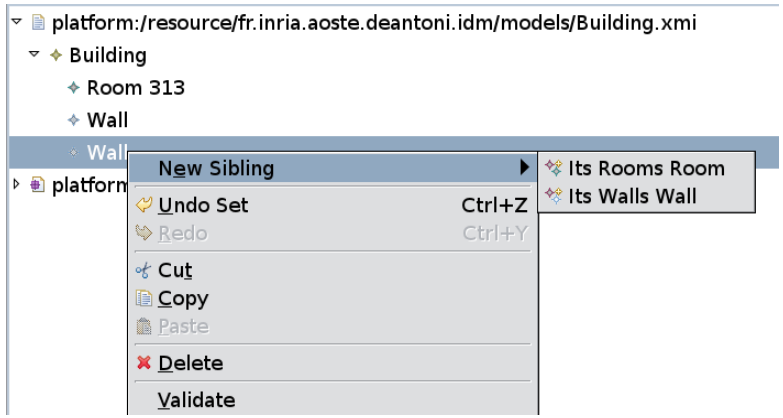



```
Building{
    Wall w1;
    Wall w2;
    Wall w3;
    Wall w4;
    Room 313 (w1,w2,w3,w4);
}
```

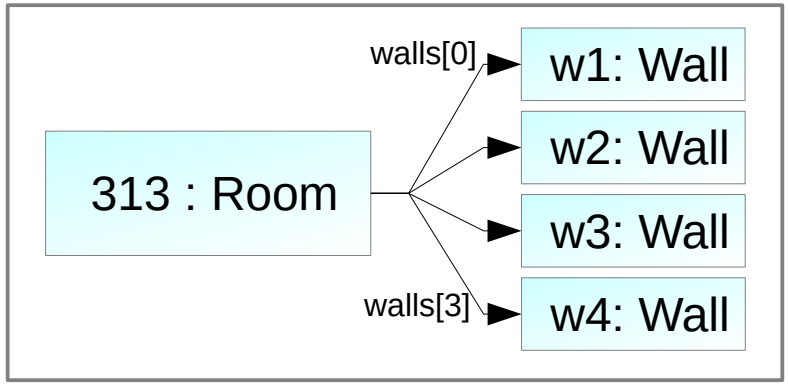


Syntaxes concrètes et abstraites (caricature)

- On peut avoir plusieurs syntaxes concrètes pour une même syntaxe abstraite




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



Bon ok mais...

Mise en oeuvre de EMF

2. Créer un “ecore modeling project”

```
fr.univcotedazur.l3ia.lenomduprojet.model
```

Faire le métamodèle EMF

finaliser le métamodèle EMF

`fr.univcotedazur.l3ia.lenomduprojet`

`http://univcotedazur.fr/l3ia/lenomduprojet/v0.1`

Générer l'API java et le code de l'éditeur

Utiliser l'éditeur

Bon...