

# Modélisation du comportement des systèmes 1

Frédéric Boulanger

Novembre 2019

# Objectifs de ce cours

## Les modèles

- ▶ À quoi sert un modèle ?
- ▶ Qu'est-ce qu'un modèle ?
- ▶ Comment construire un modèle ?
- ▶ Qu'est-ce qu'un métamodèle ?

# Objectifs de ce cours

## L'ingénierie dirigée par les modèles

- ▶ Pourquoi utiliser des modèles ?
- ▶ Quand utiliser des modèles ?
- ▶ Comment choisir un outil de modélisation ?
- ▶ Comment utiliser au mieux les modèles ?

# Objectifs de ce cours

## Syntaxe et sémantique

- ▶ Systèmes, modèles, langages et métamodèles
- ▶ Modélisation structurelle / modélisation comportementale
- ▶ Syntaxe concrète, syntaxe abstraite
- ▶ Sémantique concrète, sémantique abstraite
- ▶ Modèles de calcul

# Objectifs de ce cours

## Composabilité et compositionnalité

- ▶ Approche système pour les systèmes complexes
- ▶ Décomposition, implémentation, recomposition
- ▶ Propriétés émergentes
- ▶ Vérification, validation, correction par construction

# Objectifs de ce cours

## Paradigmes de modélisation

- ▶ Langages, modèles de calcul, modèles d'exécution
- ▶ Paradigmes usuels :
  - ▶ Temps continu
  - ▶ Événements discrets
  - ▶ Réseaux à flots de données (SDF, KPN)
  - ▶ Automates
  - ▶ Modèles du parallélisme : synchrone, threads
  - ▶ Sémantique constructive du synchrone

# Modèles, ingénierie dirigée par les modèles

# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ augmentation de la complexité des systèmes
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée

# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ **augmentation de la complexité des systèmes**
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ **augmentation de la complexité des systèmes**
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ souplesse du logiciel par rapport à la mécanique
- ▶ plus de fonction pour une énergie



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ **augmentation de la complexité des systèmes**
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ souplesse du logiciel par rapport à la mécanique
- ▶ plus de... pour un...



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ **augmentation de la complexité des systèmes**
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ couplage du logiciel par rapport



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ augmentation de la complexité des systèmes
- ▶ **numérisation des données**  $\Rightarrow$  **traitement uniforme**
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ augmentation de la complexité des systèmes
- ▶ **numérisation des données**  $\Rightarrow$  **traitement uniforme**
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ augmentation de la complexité des systèmes
- ▶ **numérisation des données**  $\Rightarrow$  **traitement uniforme**
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

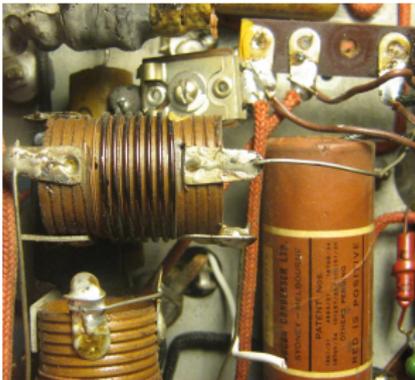
- ▶ augmentation de la complexité des systèmes
- ▶ **numérisation des données**  $\Rightarrow$  **traitement uniforme**
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

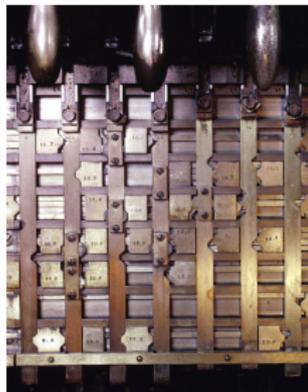
- ▶ augmentation de la complexité des systèmes
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ **souplesse du logiciel par rapport à la mécanique ou l'électronique**
- ▶ plus de fonctionnalités dans un volume donné, pour une énergie donnée



# Modèles, ingénierie dirigée par les modèles

## Importance croissante des modèles informatiques

- ▶ augmentation de la complexité des systèmes
- ▶ numérisation des données  $\Rightarrow$  traitement uniforme
- ▶ souplesse du logiciel par rapport à la mécanique ou l'électronique
- ▶ **plus de fonctionnalités dans un volume donné, pour une énergie donnée**



# Sources de la complexité

## Nombre de composants d'un système ?

- ▶ montre mécanique, cockpit d'avion de ligne → complexes
- ▶ cristal de silicium → simple

# Sources de la complexité

## Nombre de composants d'un système ?

- ▶ montre mécanique, cockpit d'avion de ligne → complexes
- ▶ cristal de silicium → simple

## Interactions entre les composants ?

- ▶ montre mécanique, cockpit d'avion de ligne → complexes
- ▶ neurones, fourmis → complexes
- ▶ circuit RLC → simple

# Sources de la complexité

## Nombre de composants d'un système ?

- ▶ montre mécanique, cockpit d'avion de ligne → complexes
- ▶ cristal de silicium → simple

## Interactions entre les composants ?

- ▶ montre mécanique, cockpit d'avion de ligne → complexes
- ▶ neurones, fourmis → complexes
- ▶ circuit RLC → simple

## Complexité

- ▶ interactions non linéaires entre les composants
- ▶ exemples : trois aimants

# Complexité

## Gestion de la complexité

- ▶ décomposition en sous-systèmes
- ▶ réalisation des sous-systèmes
- ▶ recomposition des sous-systèmes (intégration)

## Abstraction

- ▶ ignorer les détails au niveau global
- ▶ ajouter des détails lors de la réalisation
- ▶ définir des interfaces, masquer les détails internes

## Modélisation

- ▶ outil pour l'abstraction
- ▶ permet de ne considérer que les détails pertinents

# Ingénierie dirigée par les modèles (IDM)

## *Model Driven Engineering (MDE)*

- ▶ centrer la conception sur les modèles
- ▶ raffinement = transformation de modèle
- ▶ conception = raffinement des spécifications jusqu'à l'implémentation

## Model Driven Architecture (MDA)

- ▶ approche MDE proposée par l'OMG
- ▶ modèle indépendant de la plateforme (PIM)
- ▶ + modèle de la plateforme  
⇒ modèle spécifique à la plateforme (PSM)

# Qu'est-ce qu'un modèle ?

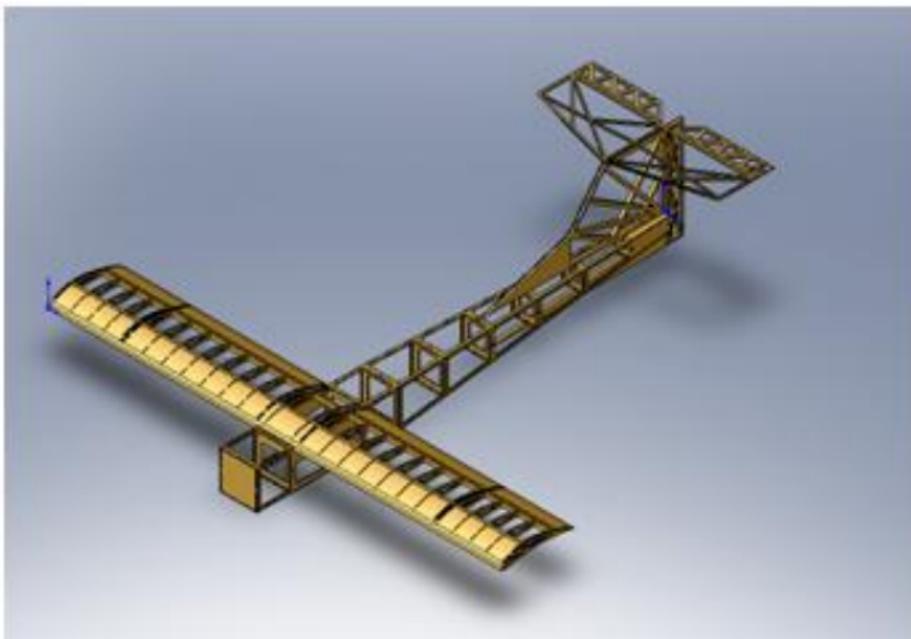
# Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand



## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose



## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose
3. Une description schématique permettant de répondre à des questions sur un système



## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose
3. Une description schématique permettant de répondre à des questions sur un système
4. **Le style ou la conception d'un équipement**



## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose
3. Une description schématique permettant de répondre à des questions sur un système
4. Le style ou la conception d'un équipement
5. Exemple à imiter, référence



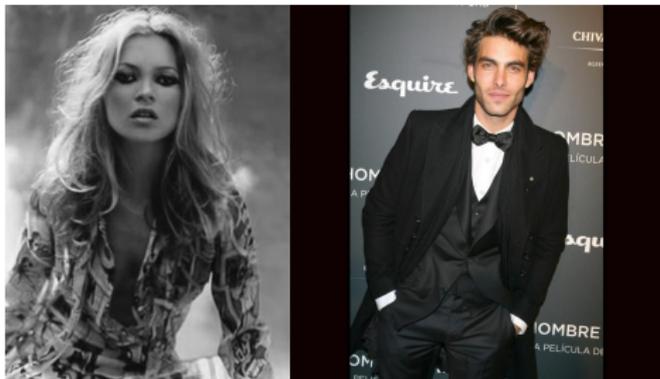
## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose
3. Une description schématique permettant de répondre à des questions sur un système
4. Le style ou la conception d'un équipement
5. Exemple à imiter, référence
6. Ce qui sert de sujet à un artiste



## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose
3. Une description schématique permettant de répondre à des questions sur un système
4. Le style ou la conception d'un équipement
5. Exemple à imiter, référence
6. Ce qui sert de sujet à un artiste
7. Une personne employée à présenter des articles de mode



## Qu'est-ce qu'un modèle ?

1. Un petit objet qui représente un objet plus grand
2. Une maquette qui sert de plan pour construire quelque chose
3. Une description schématique permettant de répondre à des questions sur un système
4. Le style ou la conception d'un équipement
5. Exemple à imiter, référence
6. Ce qui sert de sujet à un artiste
7. Une personne employée à présenter des articles de mode

Un modèle est une **représentation simplifiée** d'un système, établie dans un certain **objectif**. Un modèle doit permettre de répondre à des questions que l'on se pose sur le système.

# Typologie des modèles

## Aspect modélisé

- ▶ modèle structurel (diagramme de classes)
- ▶ modèle comportemental (automate)

## Objectif

- ▶ modèle explicatif
- ▶ modèle prédictif

## Utilisation

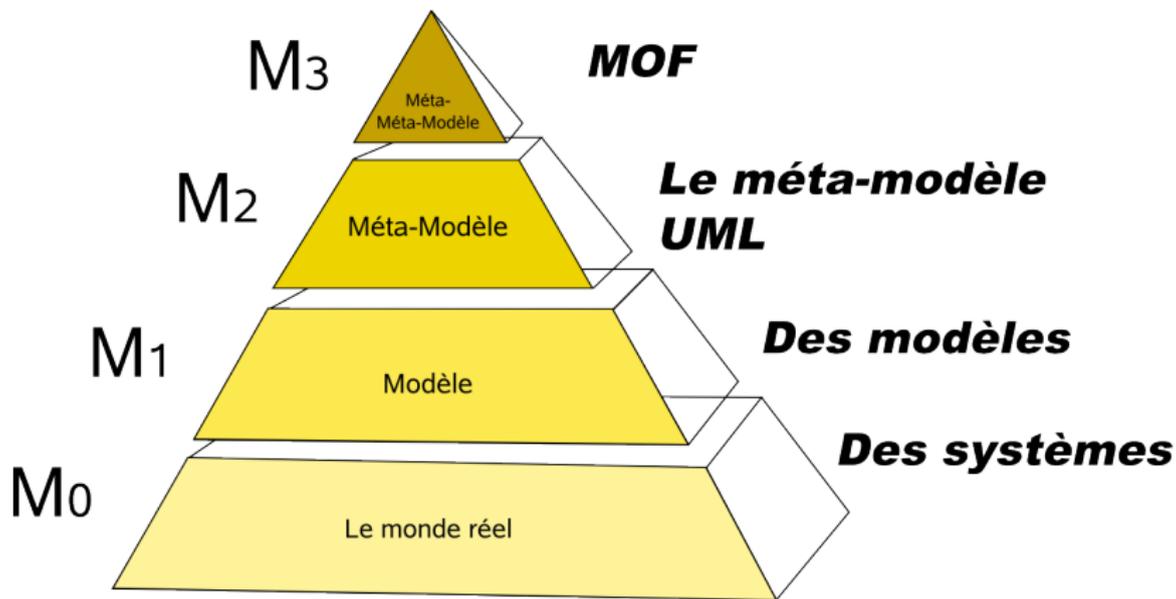
- ▶ modèle constructif (exécutable ou approximable)
- ▶ modèle d'analyse (système pré-existant)
- ▶ modèle de conception (système à concevoir)

Un modèle d'un modèle est-il un métamodèle ?

# Systèmes, modèles, langages

- ▶ un modèle est une représentation d'un système
- ▶ un modèle peut être étudié en tant que système (ingénierie des modèles)
- ▶ il existe des modèles de modèles
- ▶ un modèle est décrit dans un langage de modélisation
- ▶ un langage de modélisation peut être étudié en tant que système (ingénierie des langages)
- ▶ il existe des modèles de langages de modélisation
- ▶ un langage de modélisation est décrit par une grammaire (syntaxe concrète) ou un métamodèle (syntaxe abstraite)

# La pyramide UML



# Systèmes, modèles, langages

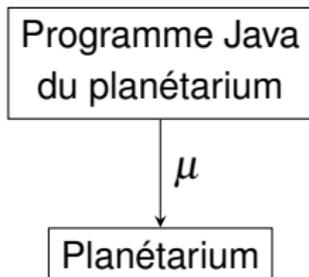
Planétarium

# Systèmes, modèles, langages

Programme Java  
du planétarium

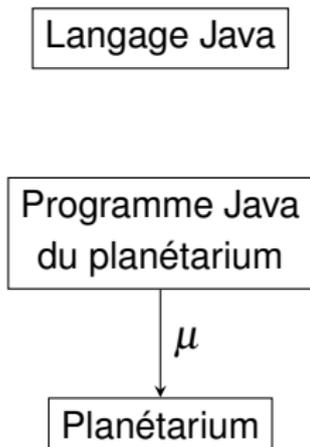
Planétarium

# Systèmes, modèles, langages



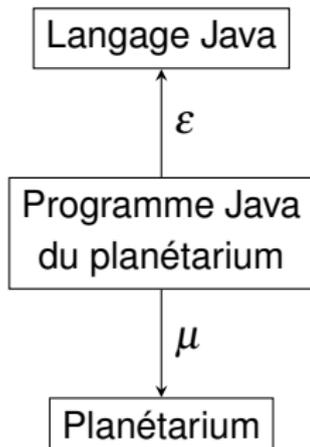
$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

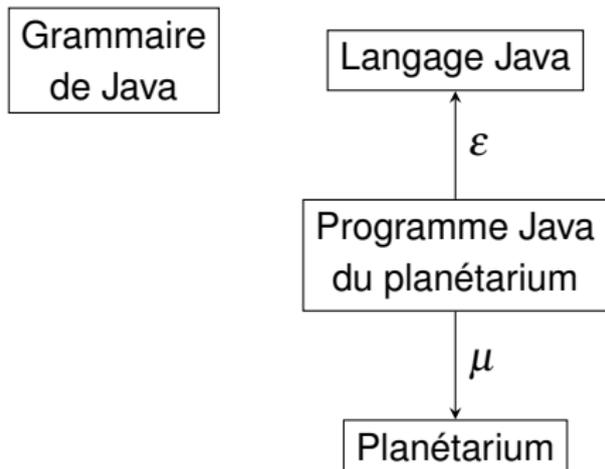
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

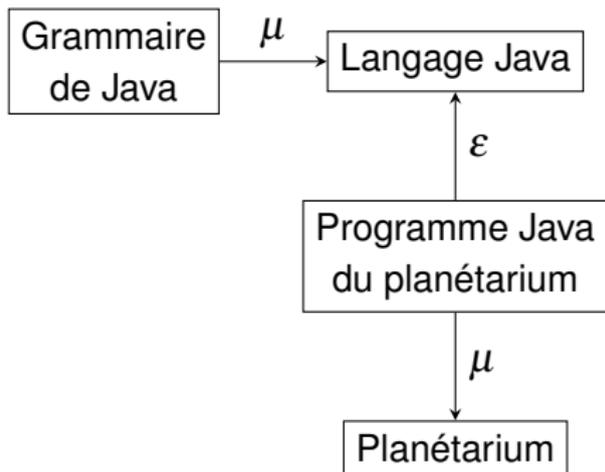
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

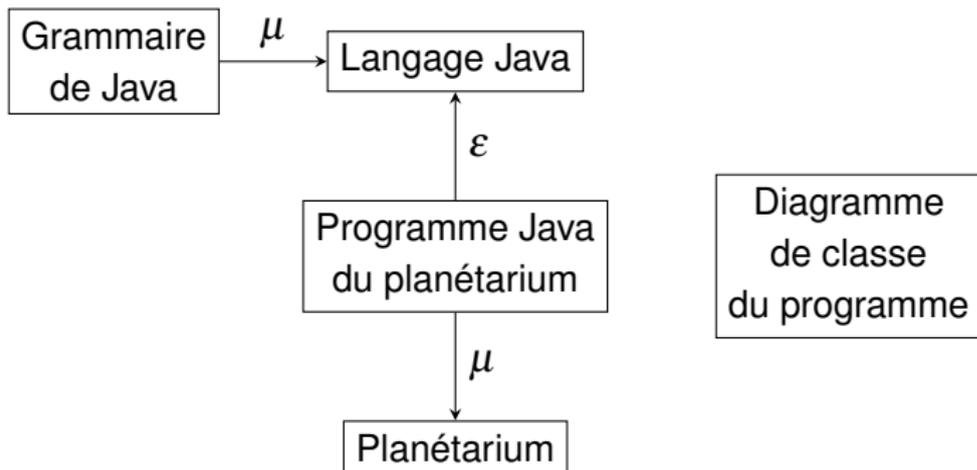
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

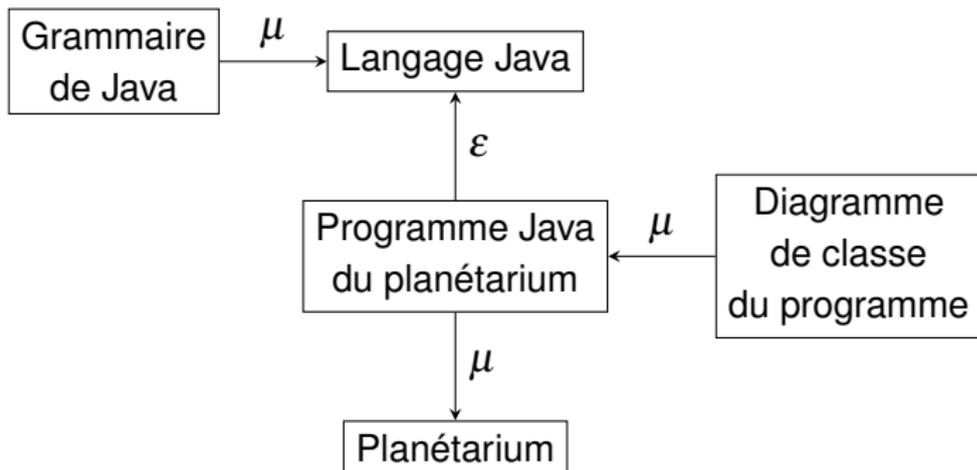
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

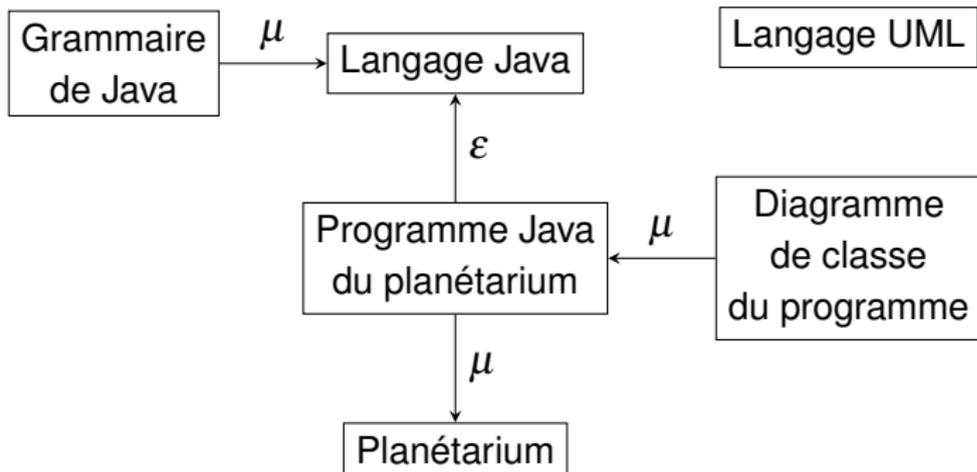
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

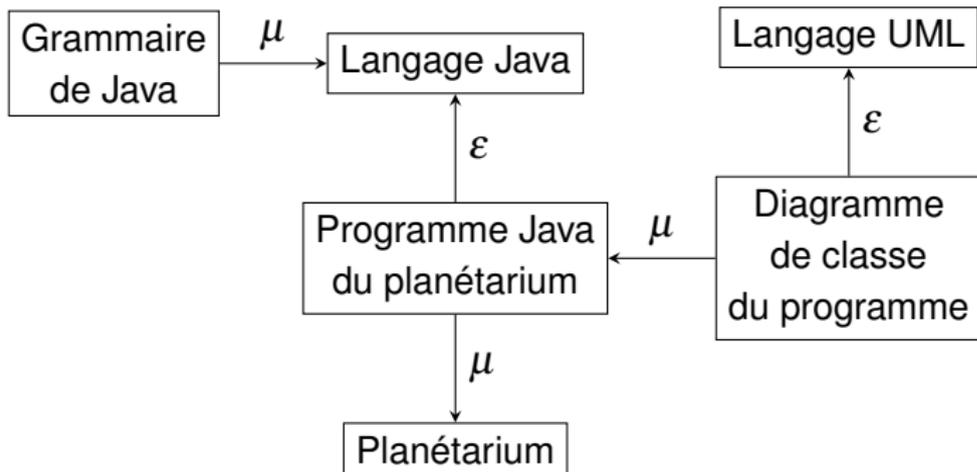
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

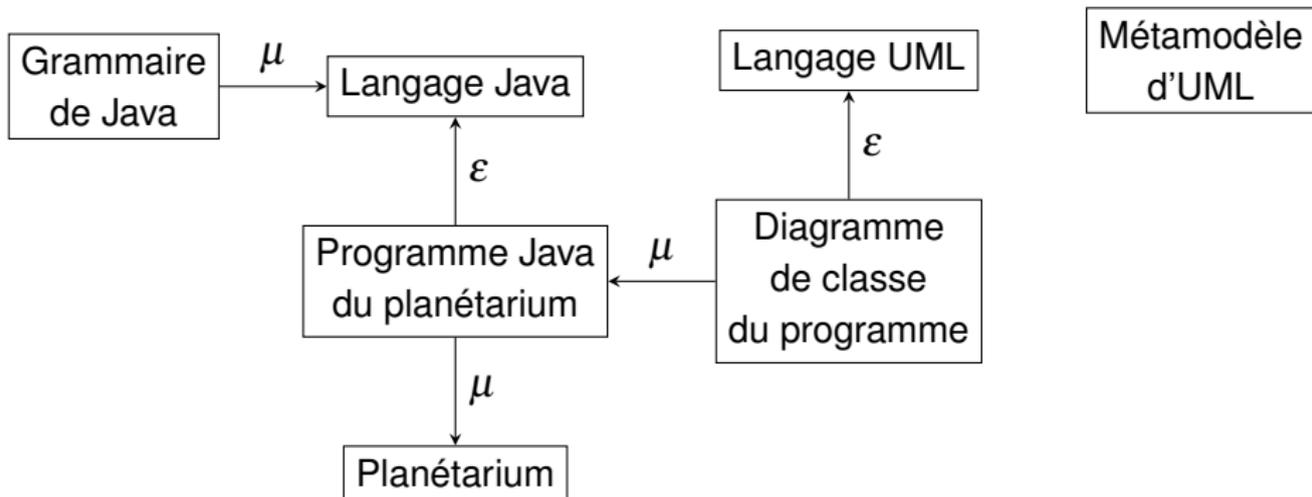
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\epsilon} L$  M appartient au (est écrit dans le) langage L

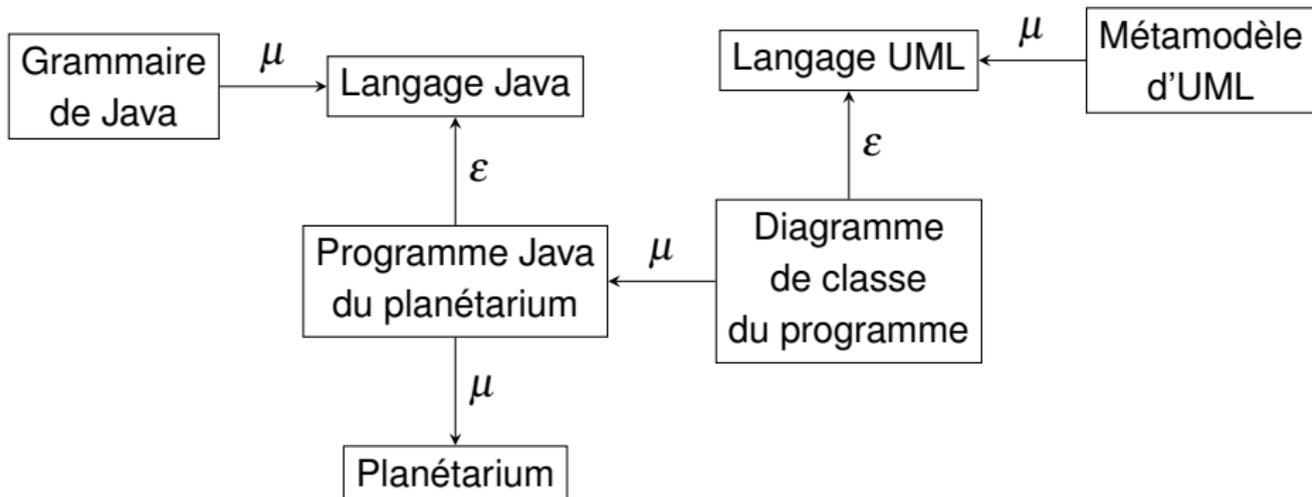
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\epsilon} L$  M appartient au (est écrit dans le) langage L

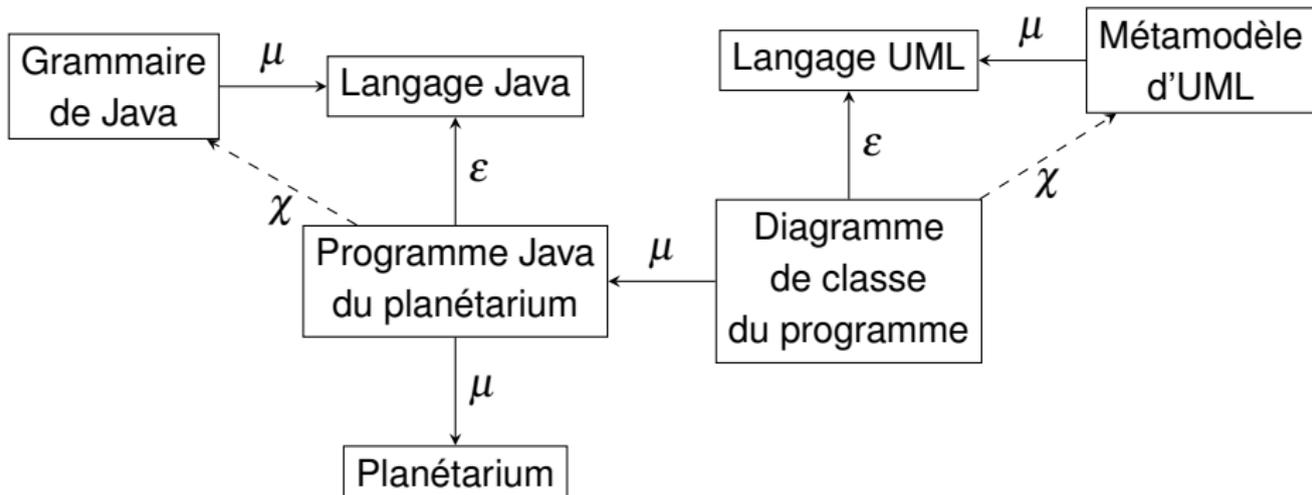
# Systèmes, modèles, langages



$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\epsilon} L$  M appartient au (est écrit dans le) langage L

# Systèmes, modèles, langages

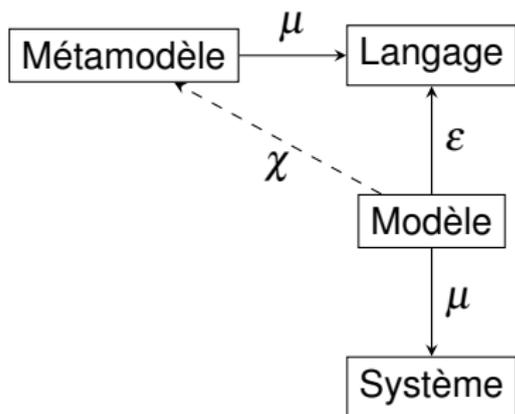


$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

$M \xrightarrow{\chi} MM$  M est conforme au métamodèle MM

# Systèmes, modèles, langages

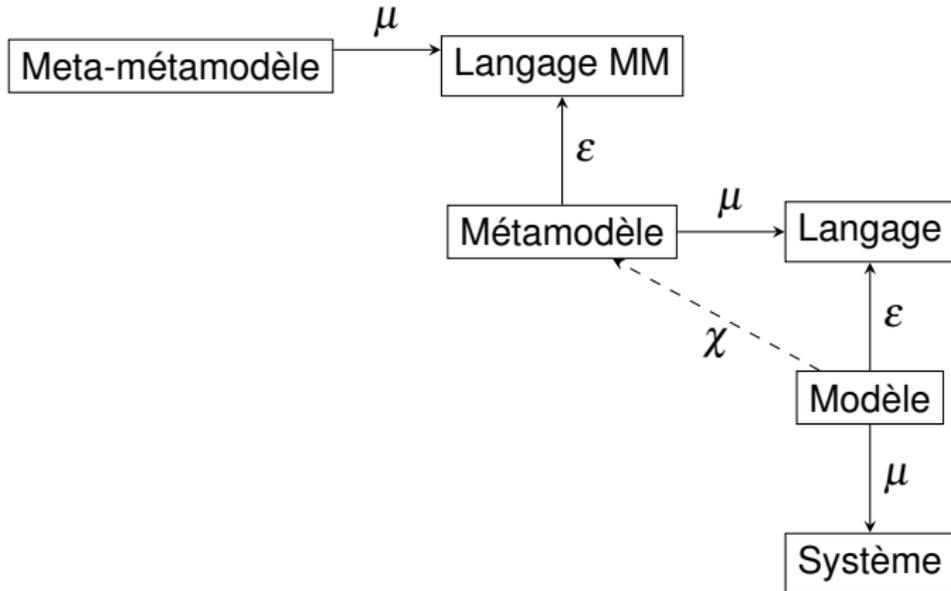


$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

$M \xrightarrow{\chi} MM$  M est conforme au métamodèle MM

# Systèmes, modèles, langages

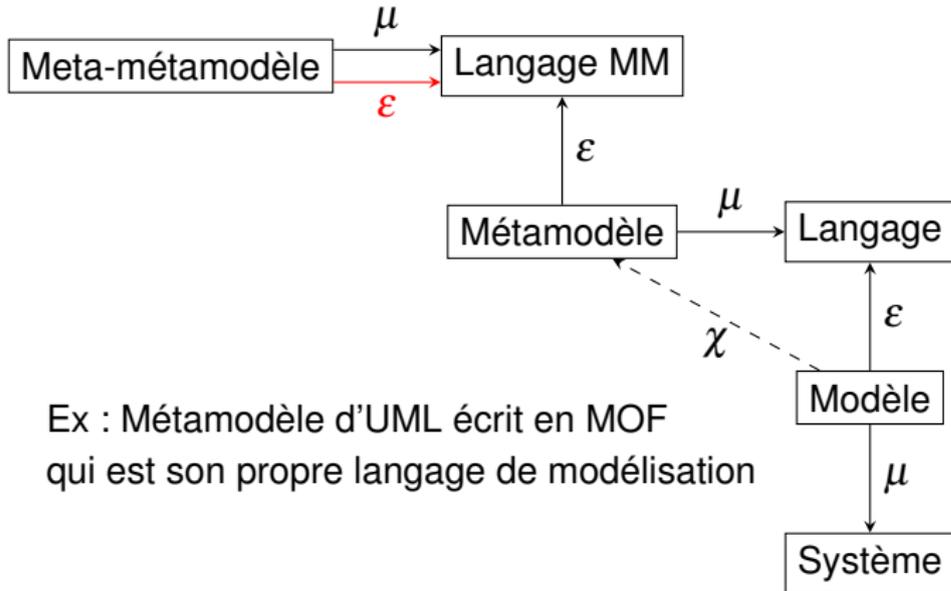


$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

$M \xrightarrow{\chi} MM$  M est conforme au métamodèle MM

# Systèmes, modèles, langages



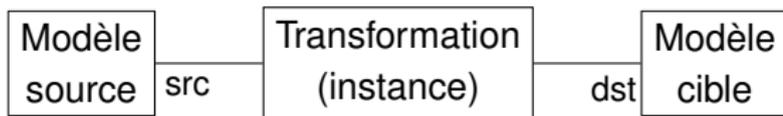
Ex : Métamodèle d'UML écrit en MOF  
qui est son propre langage de modélisation

$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

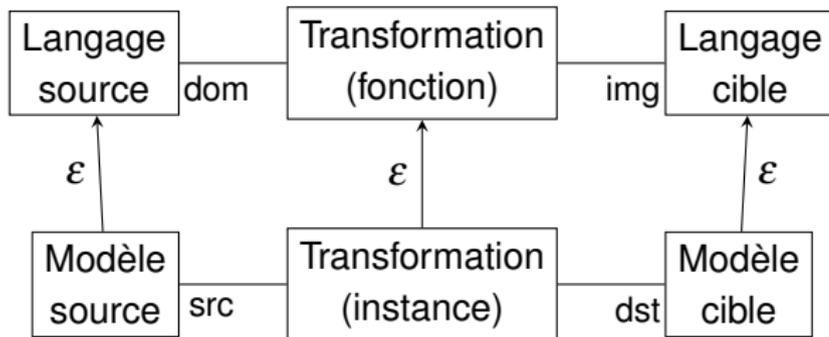
$M \xrightarrow{\epsilon} L$  M appartient au (est écrit dans le) langage L

$M \xrightarrow{\chi} MM$  M est conforme au métamodèle MM

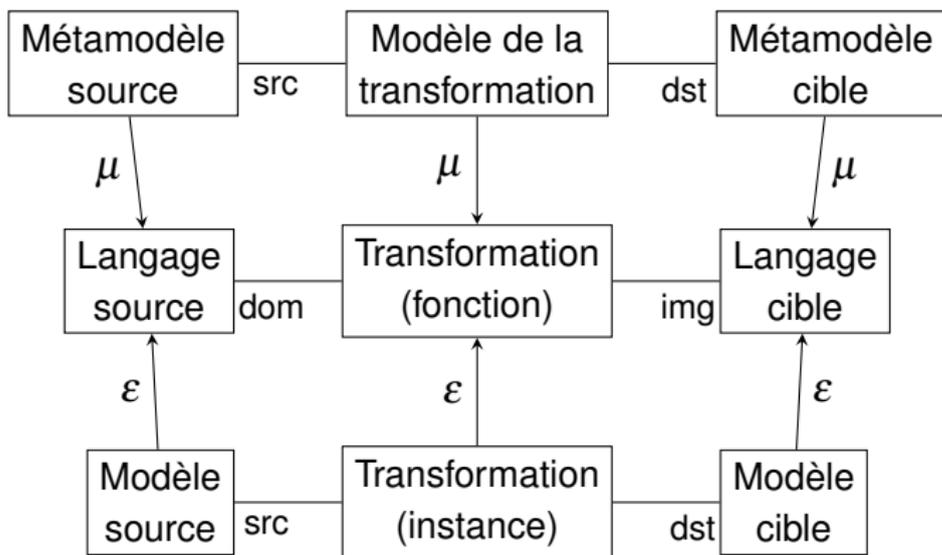
# Mégamodèle d'une transformation de modèles



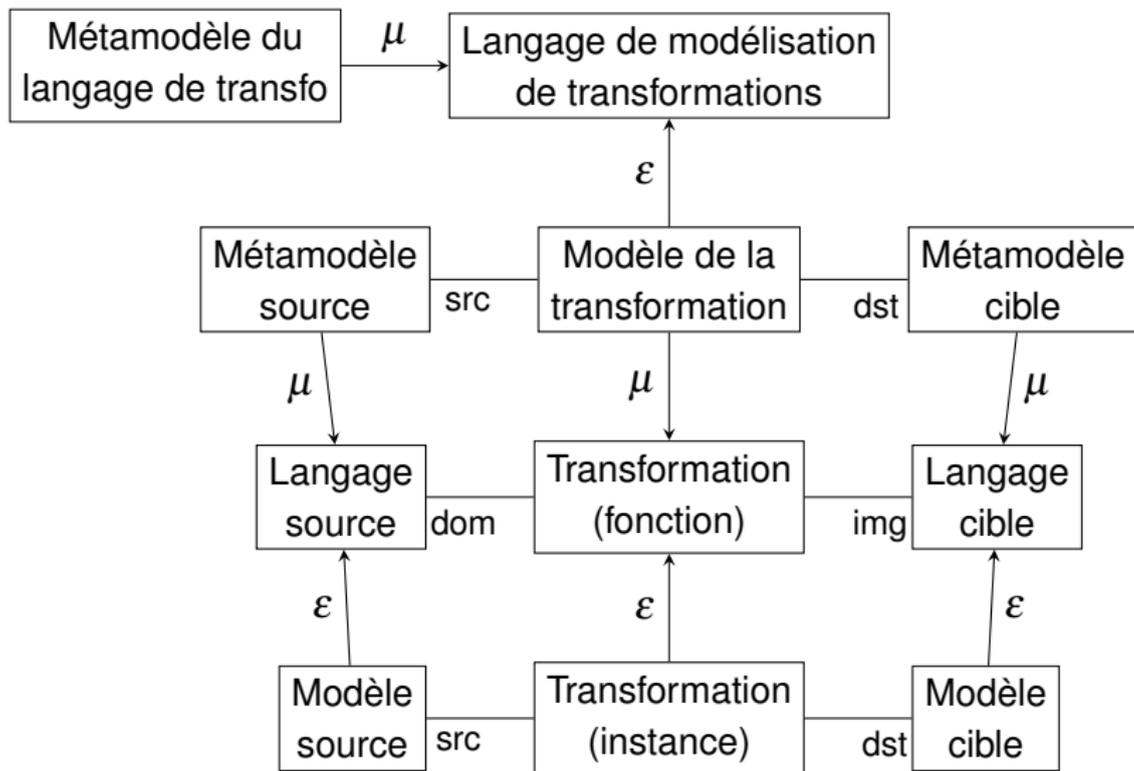
# Mégamodèle d'une transformation de modèles



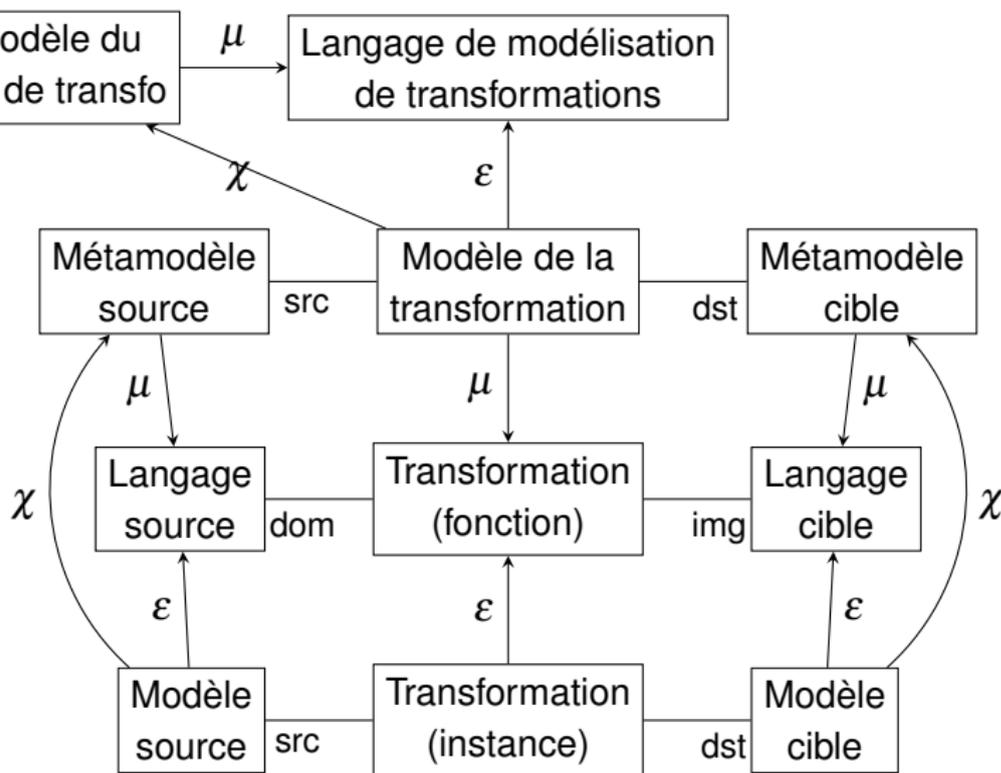
# Mégamodèle d'une transformation de modèles



# Mégamodèle d'une transformation de modèles



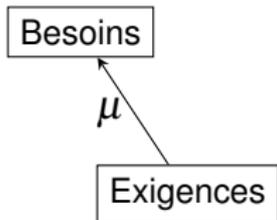
## Mégamodèle d'une transformation de modèles



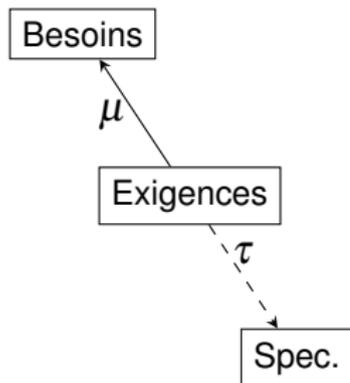
# Cycle de vie d'un système

Besoins

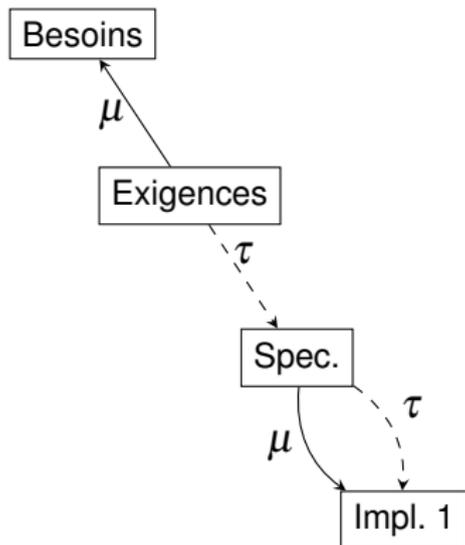
# Cycle de vie d'un système



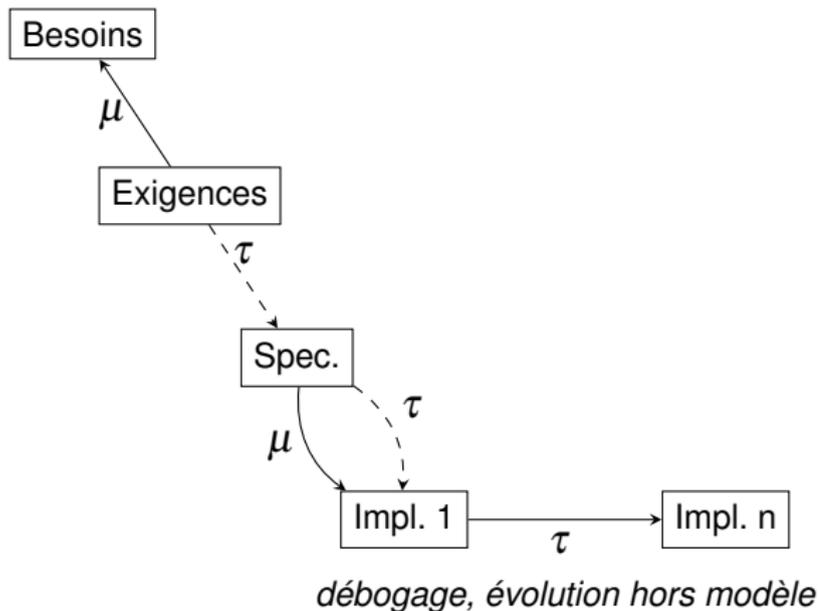
# Cycle de vie d'un système



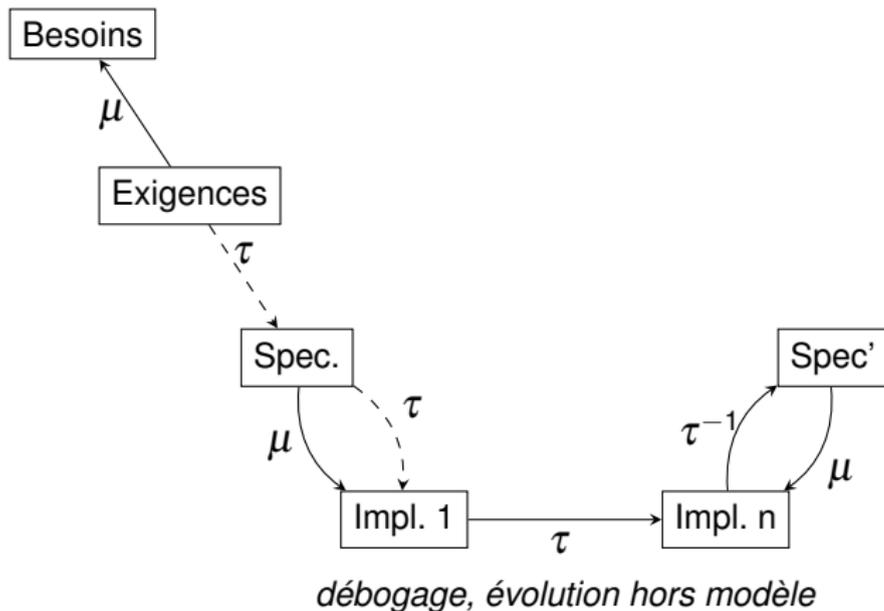
# Cycle de vie d'un système



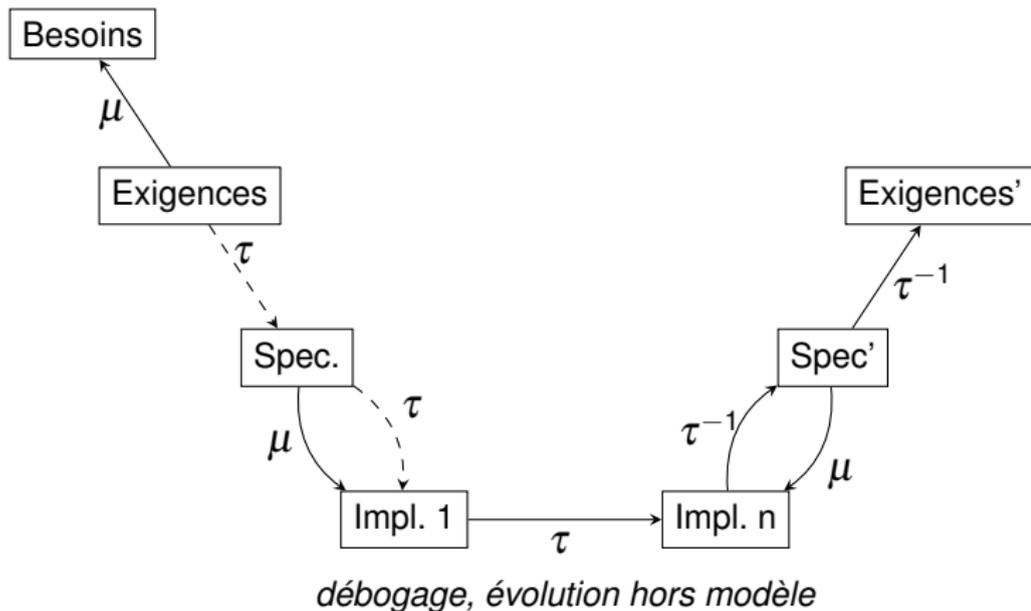
# Cycle de vie d'un système



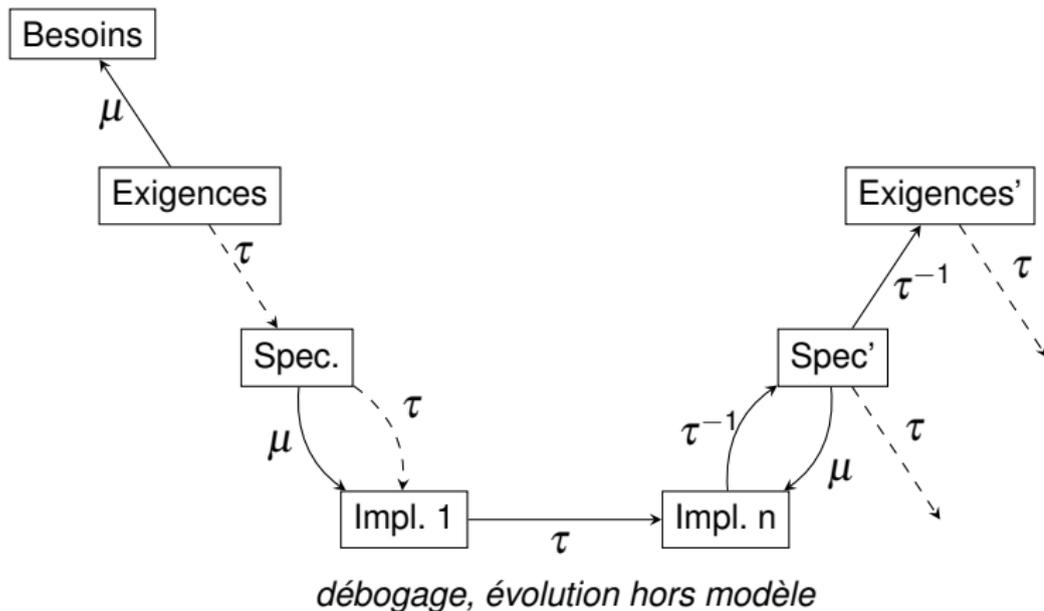
# Cycle de vie d'un système



# Cycle de vie d'un système



# Cycle de vie d'un système



# Modèles, modèles... (pourquoi l'homme de Néanderthal a disparu)

