

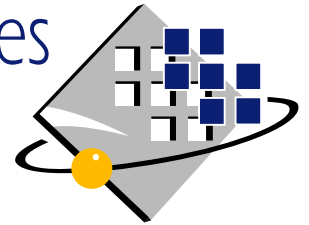


ÉCOLE
SUPÉRIEURE
D'ÉLECTRICITÉ



UNIVERSITÉ
PARIS-SUD 11

Faculté des
sciences
d'Orsay



Composition de modèles pour la modélisation multi-paradigme du comportement des systèmes

Cécile Hardebolle

Thèse

préparée au **Département Informatique de Supélec**,
Ecole Doctorale d'Informatique de l'**Université Paris-Sud XI**

28 nov. 2008

▶ Introduction

Hétérogénéité des modèles

Modélisation multi-paradigme

ModHel'X

Framework expérimental

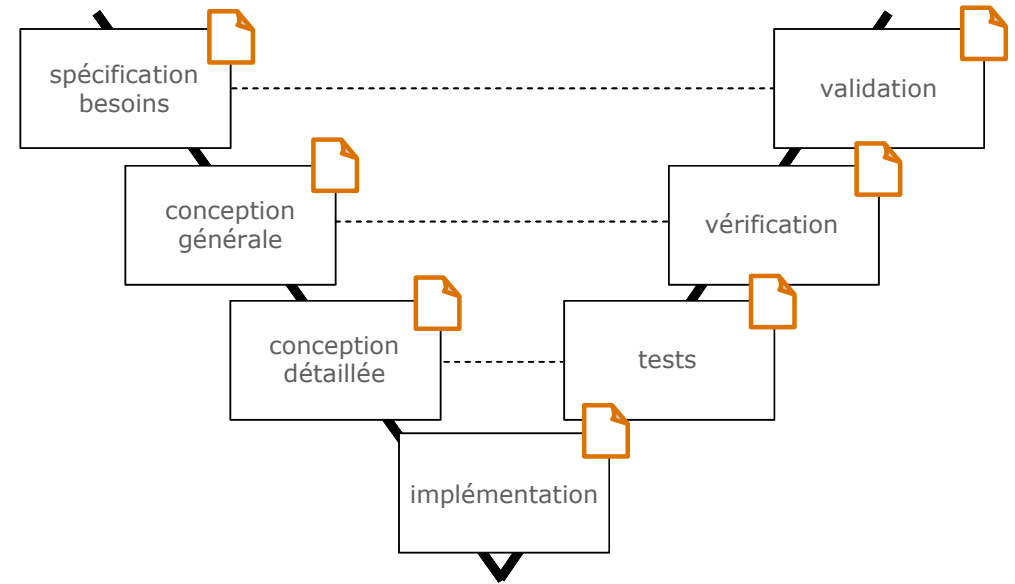
Conclusion & perspectives

Contexte : IDM et systèmes complexes

■ Ingénierie Dirigée par les Modèles (IDM)

1. Modèles informatiques
=
support prépondérant
du cycle de développement

2. Outils informatiques
de bout en bout du cycle

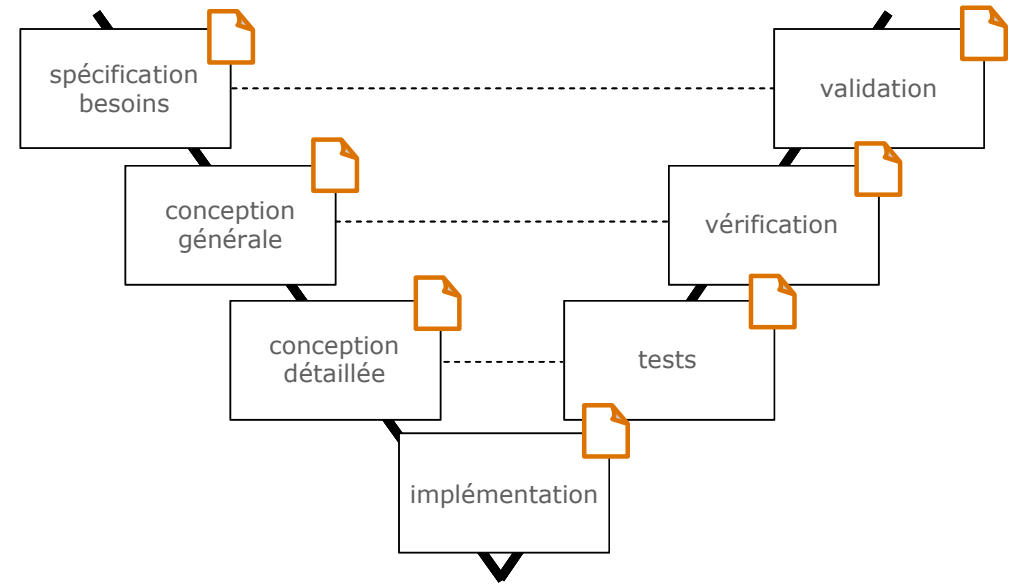


Contexte : IDM et systèmes complexes

■ Ingénierie Dirigée par les Modèles (IDM)

1. Modèles informatiques
=
support prépondérant
du cycle de développement

2. Outils informatiques
de bout en bout du cycle



■ Systèmes complexes

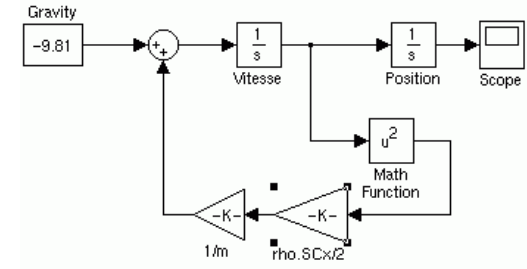
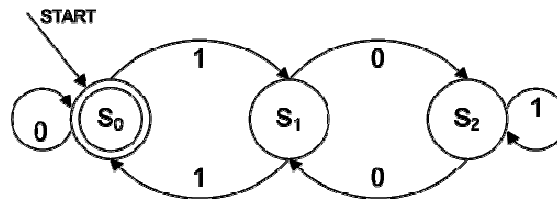
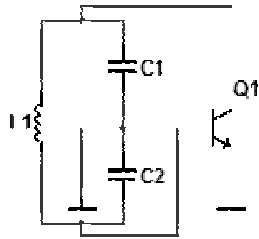
- ▶ Système : ensemble d'éléments en interaction pour **réaliser une fonction donnée** (finalité)
- ▶ Complexe : nombre élevé d'éléments, **hétérogénéité des éléments**, organisation interne, couplage avec l'environnement



➔ **Difficulté de prévoir le comportement global** du système à partir du comportement unitaire de ses éléments

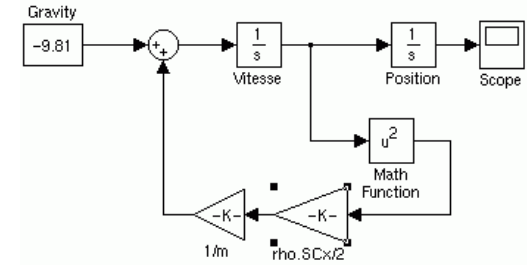
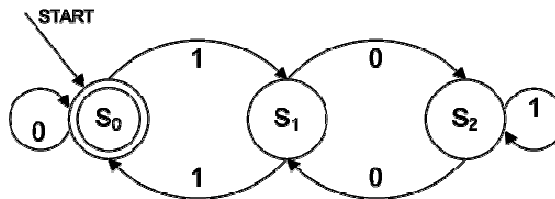
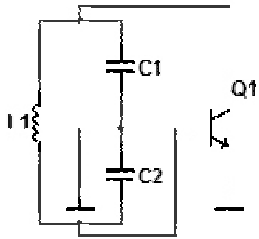
Vocabulaire : modèle, langage de modélisation

- **Modèle** : représentation prédictive du comportement d'un système



Vocabulaire : modèle, langage de modélisation

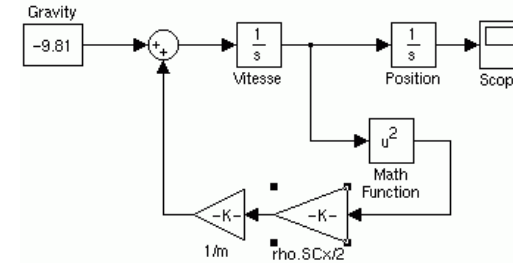
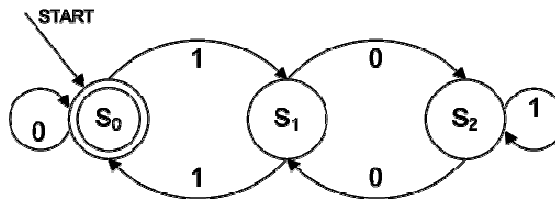
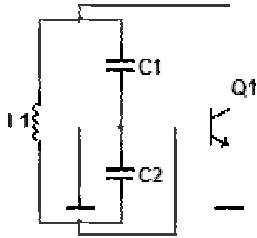
- **Modèle** : représentation prédictive du comportement d'un système



- **Langage** de modélisation : langage informatique permettant de décrire des modèles
 - Syntaxe abstraite : concepts
 - Sémantique : signification
 - Syntaxe concrète : notation
- **Formalisme** de modélisation : langage de modélisation avec syntaxe et sémantique formelles

Vocabulaire : modèle, langage de modélisation

- **Modèle** : représentation prédictive du comportement d'un système



- **Langage** de modélisation : langage informatique permettant de décrire des modèles
 - Syntaxe abstraite : concepts
 - Sémantique : signification
 - Syntaxe concrète : notation
- **Formalisme** de modélisation : langage de modélisation avec syntaxe et sémantique formelles
- **Paradigme** de modélisation : « façon de modéliser »
 - ≈ Famille de langages de modélisation qui partagent des éléments de syntaxe et de sémantique

Problématique de la modélisation multi-paradigme

- Nécessité d'avoir des moyens de raisonner globalement sur le comportement des systèmes complexes

Problématique de la modélisation multi-paradigme

- Nécessité d'avoir des moyens de raisonner globalement sur le comportement des systèmes complexes
 - ▶ Exemple : Mars Pathfinder



source : spaceflightnow.com
(NASA/JPL)

Problématique de la modélisation multi-paradigme

- Nécessité d'avoir des moyens de raisonner globalement sur le comportement des systèmes complexes

- ▶ Exemple : Mars Pathfinder



source : spaceflightnow.com
(NASA/JPL)

- IDM → raisonnement sur les modèles

Intuitivement : **Systèmes** complexes et hétérogènes

↘ **Modèles** complexes et hétérogènes

↘ **Difficulté** de raisonner globalement

= problème de l'hétérogénéité des modèles

Existant

■ Le problème de l'hétérogénéité...

▶ Recouvre différentes réalités

Exemples : matériel/logiciel, temps continu/temps discret...

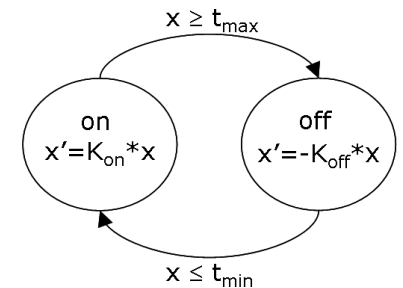
▶ N'est pas nouveau

- Nombreuses **approches spécifiques** existantes

Exemples : VHDL-AMS, automates hybrides, DEVS...

▶ Prend de l'ampleur dans le cadre de l'IDM

- Modèles prépondérants dans le cycle de développement
- Automatisation croissante des traitements sur les modèles
- Multiplication des langages, notamment « domaines-spécifiques »



Existant

■ Le problème de l'hétérogénéité...

▶ Recouvre différentes réalités

Exemples : matériel/logiciel, temps continu/temps discret...

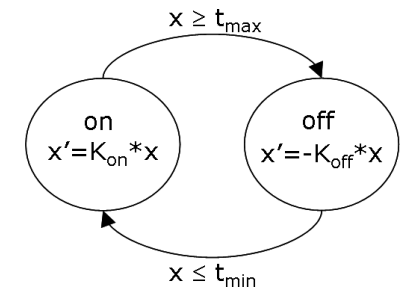
▶ N'est pas nouveau

- Nombreuses **approches spécifiques** existantes

Exemples : VHDL-AMS, automates hybrides, DEVS...

▶ Prend de l'ampleur dans le cadre de l'IDM

- Modèles prépondérants dans le cycle de développement
- Automatisation croissante des traitements sur les modèles
- Multiplication des langages, notamment « domaines-spécifiques »



■ Nécessité

- ▶ d'un **vocabulaire partagé** pour décrire le problème
- ▶ d'une **vue d'ensemble** des approches existantes
- ▶ de **solutions adaptées** au contexte de l'IDM



Introduction

▶ **Hétérogénéité des modèles**

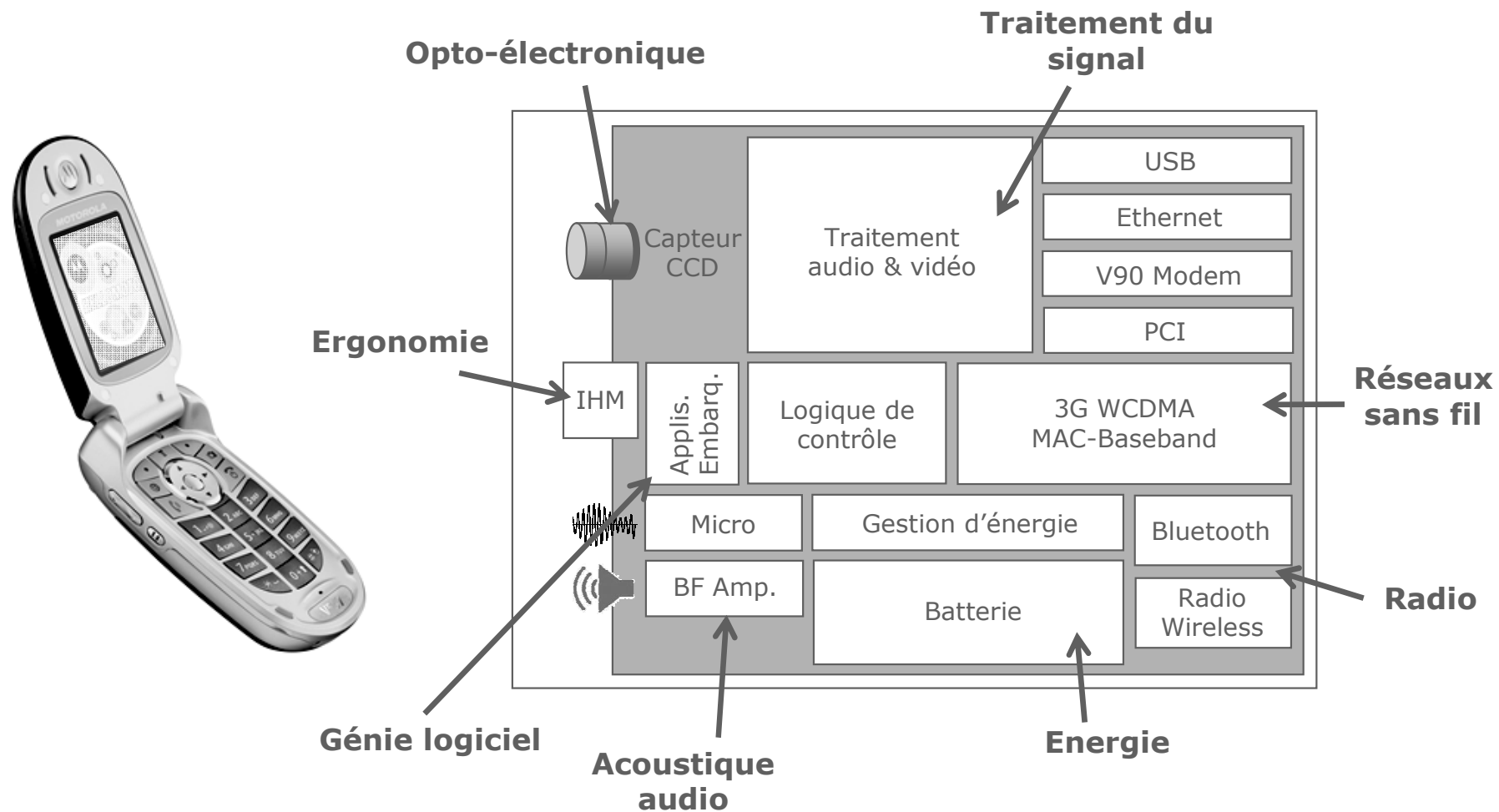
Modélisation multi-paradigme

ModHel'X

Framework expérimental

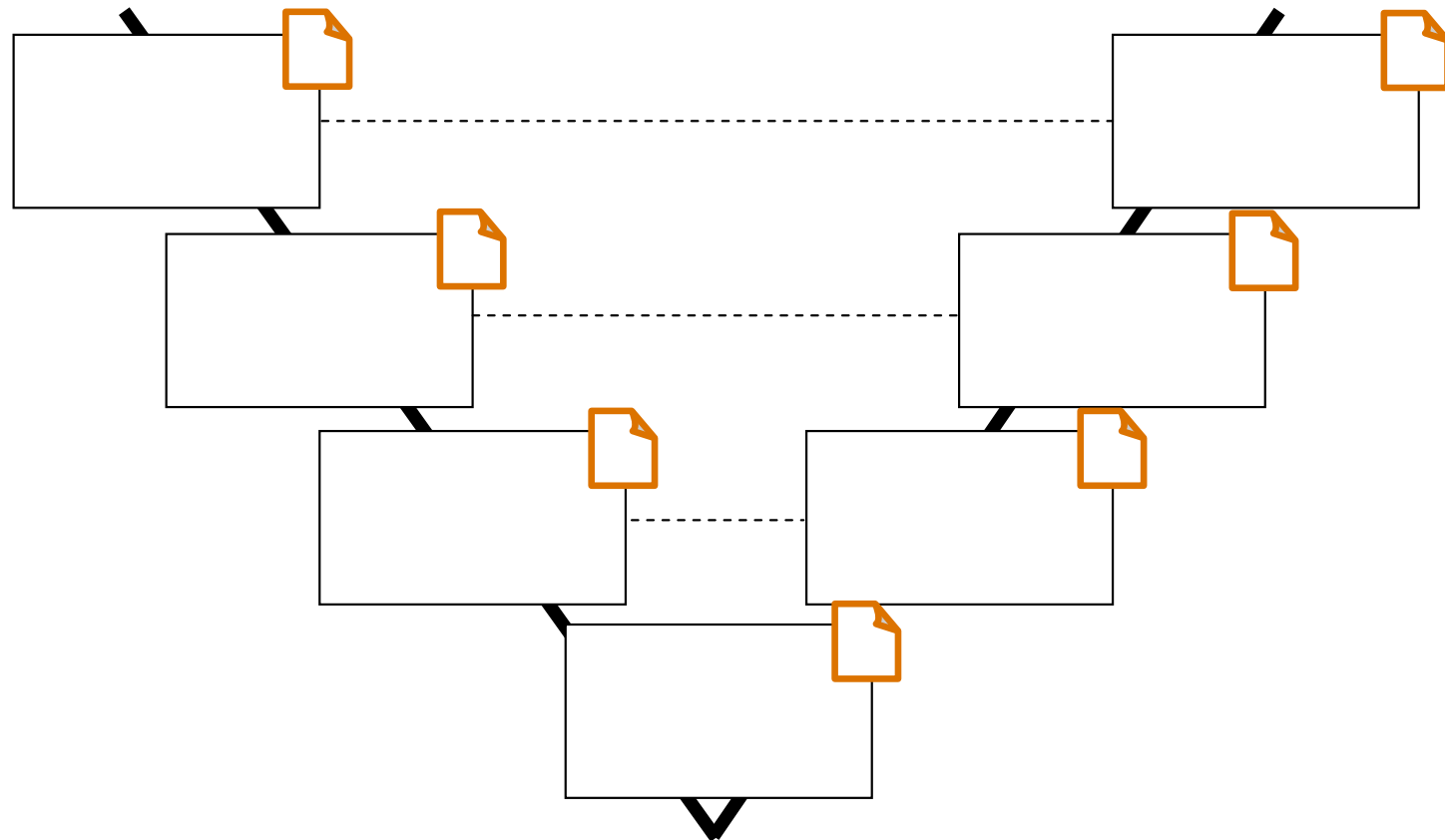
Conclusion & perspectives

Hétérogénéité des modèles : origines

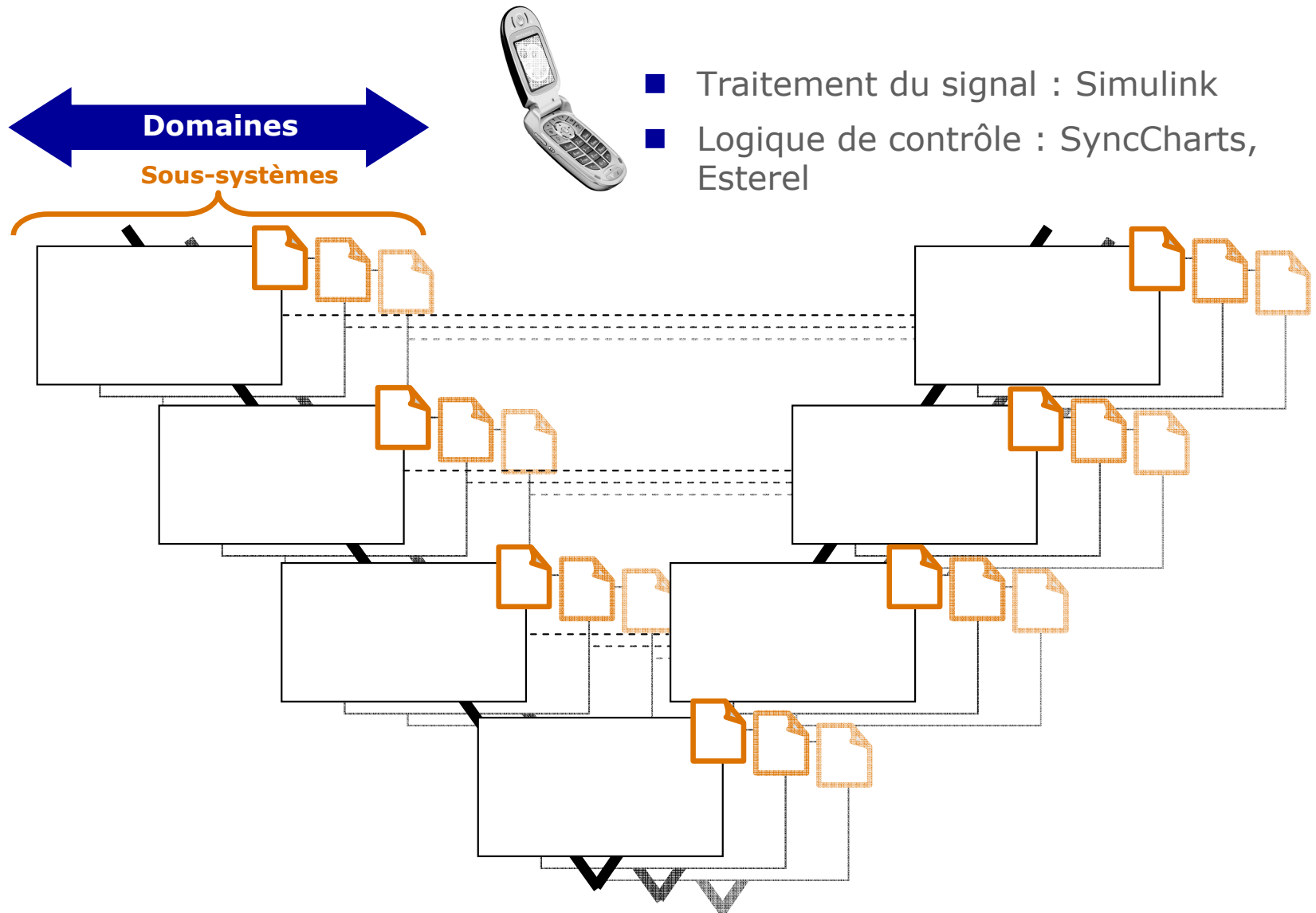


Hétérogénéité des modèles : origines

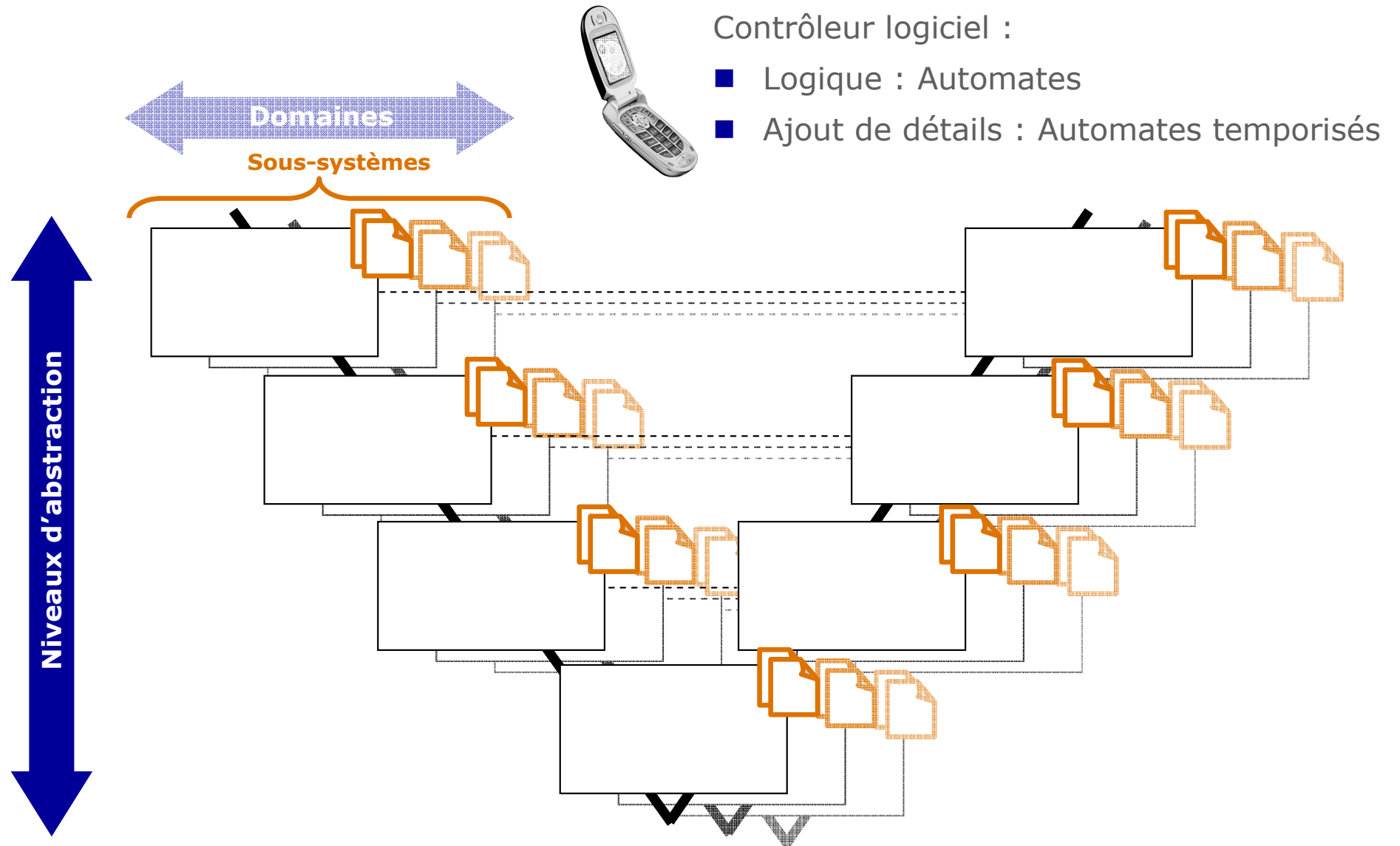
Quels modèles ? Quels langages de modélisation ?



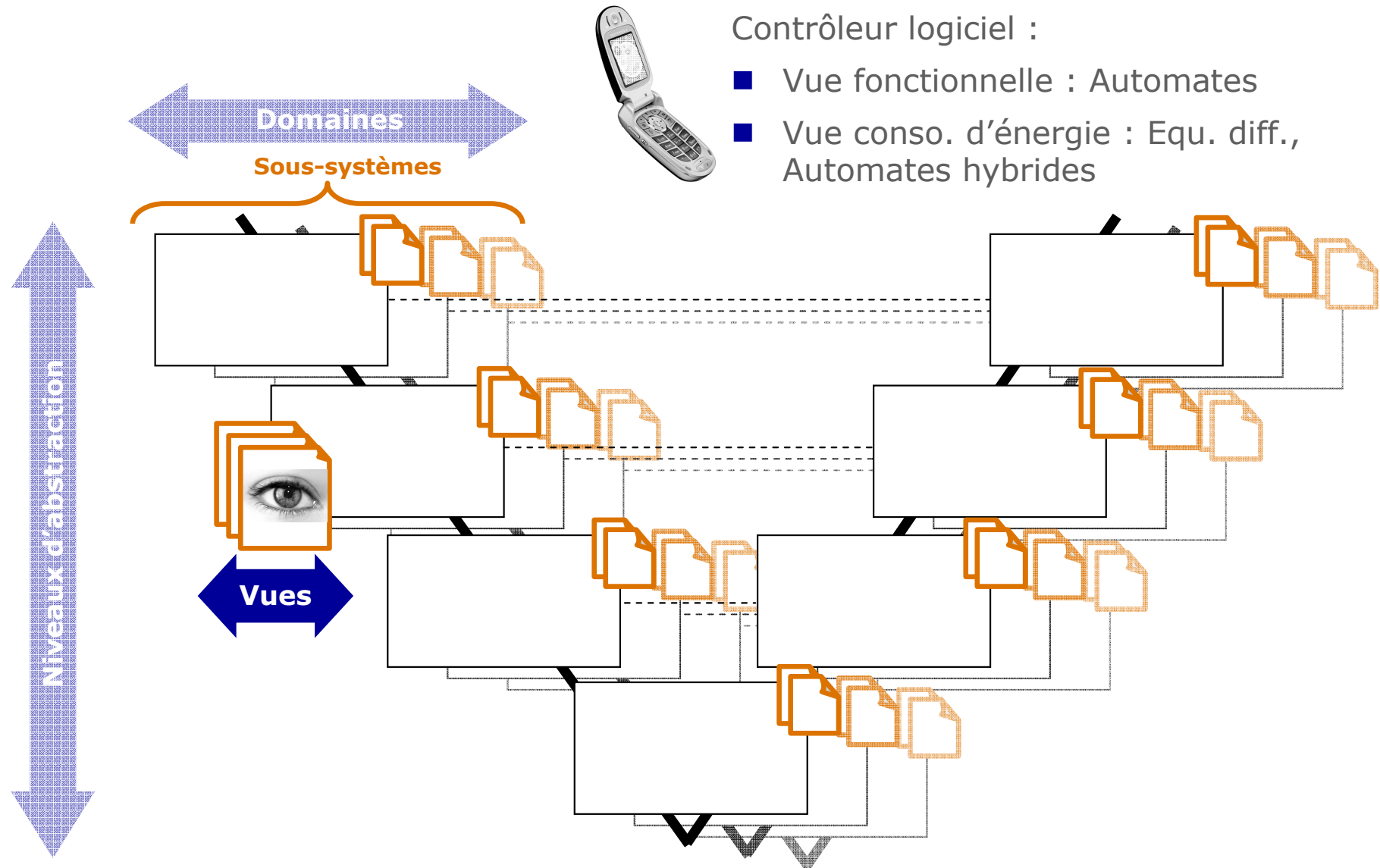
Hétérogénéité des modèles : origines



Hétérogénéité des modèles : origines



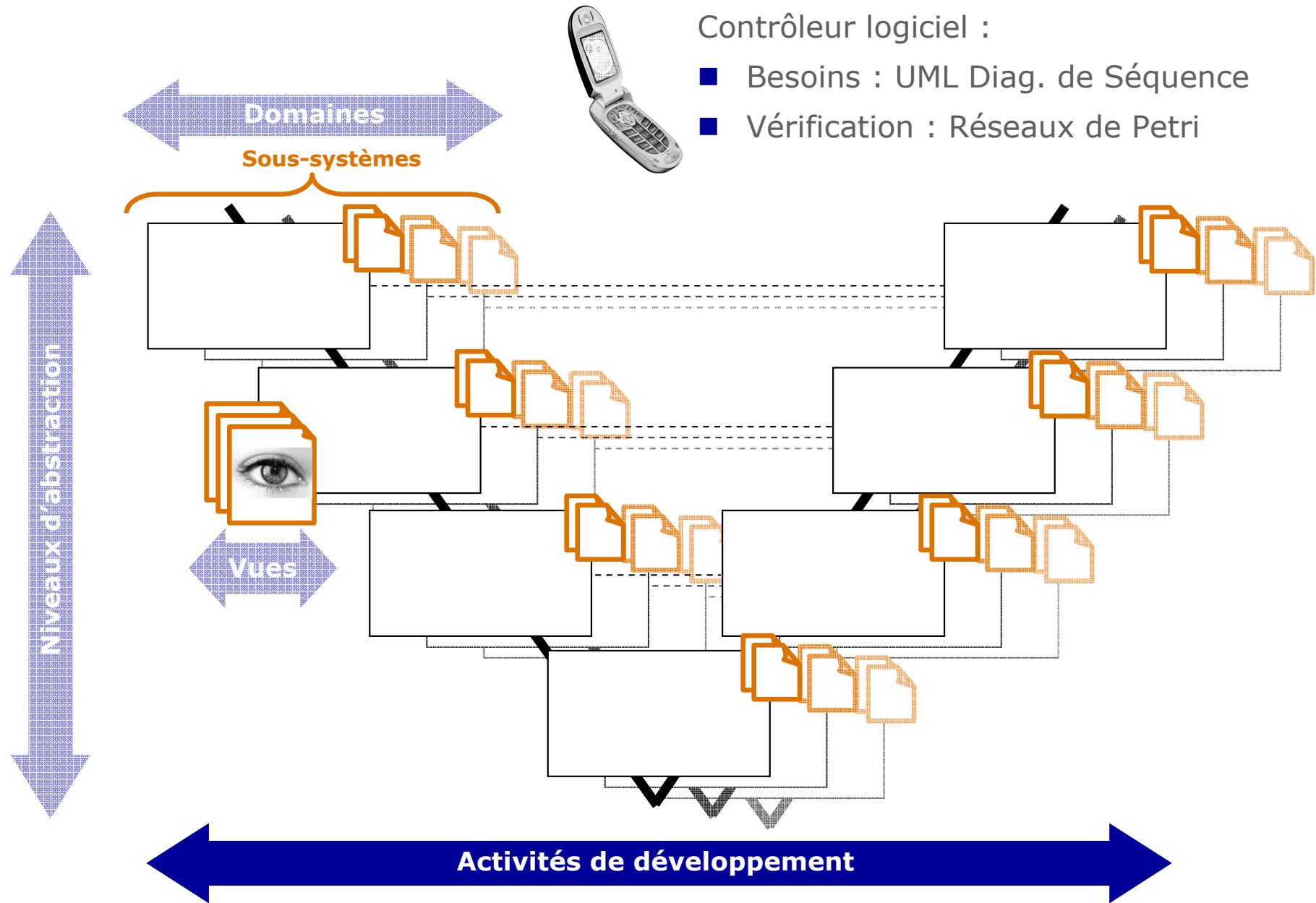
Hétérogénéité des modèles : origines



Contrôleur logiciel :

- Vue fonctionnelle : Automates
- Vue conso. d'énergie : Equ. diff., Automates hybrides

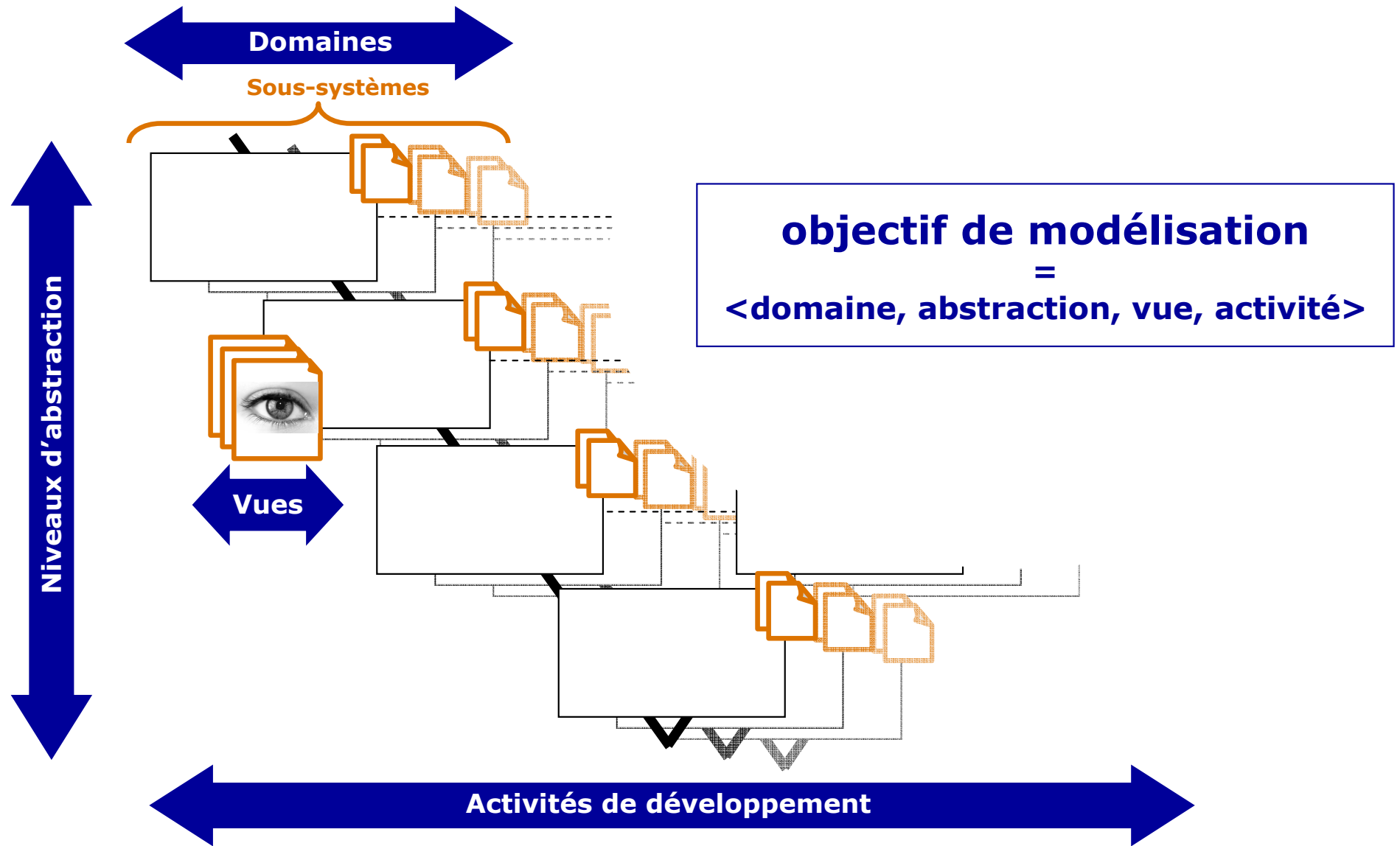
Hétérogénéité des modèles : origines



Contrôleur logiciel :

- Besoins : UML Diag. de Séquence
- Vérification : Réseaux de Petri

Hétérogénéité des modèles : origines



Hétérogénéité des modèles : problème

■ Au cours du cycle :

- ▶ Passage par différents objectifs de modélisation
- ▶ Utilisation de langages de modélisation adaptés
 - A un domaine
 - A un niveau d'abstraction
 - A une vue
 - A une activité

inévitable
&
essentiel

Hétérogénéité des modèles : problème

■ Au cours du cycle :

- ▶ Passage par différents objectifs de modélisation
- ▶ Utilisation de **langages de modélisation adaptés**
 - A un domaine
 - A un niveau d'abstraction
 - A une vue
 - A une activité

inévitable
&
essentiel

■ A un moment donné du cycle :

- ▶ Système = collection de modèles
décrits avec des langages de modélisation différents

Raisonnement global sur le système difficile !



Hétérogénéité des modèles : problème

■ Au cours du cycle :

- ▶ Passage par différents objectifs de modélisation
- ▶ Utilisation de langages de modélisation adaptés
 - A un domaine
 - A un niveau d'abstraction
 - A une vue
 - A une activité

inévitable
&
essentiel

■ A un moment donné du cycle :

- ▶ Système = collection de modèles décrits avec des langages de modélisation différents

Raisonnement global sur le système difficile !



■ Niveau de difficulté du problème :

- ▶ Langages basés sur des paradigmes différents
→ difficultés plus importantes

Русский
精忠报国

Introduction

Hétérogénéité des modèles

▶ **Modélisation multi-paradigme**

ModHel'X

Framework expérimental

Conclusion & perspectives

Modélisation multi-paradigme : axes de recherche

■ Heterogeneous Modeling [Lee 91-08]

- ▶ Hétérogénéité des domaines

■ CAMPaM (Computer Automated Multi-Paradigm Modeling)

- ▶ Hétérogénéité des niveaux d'abstraction [Mosterman 04]
- ▶ Couplage et transformation entre formalismes de modélisation
- ▶ Méta-modélisation

Modélisation multi-paradigme : axes de recherche

■ Heterogeneous Modeling [Lee 91-08]

- ▶ Hétérogénéité des domaines

■ CAMPaM (Computer Automated Multi-Paradigm Modeling)

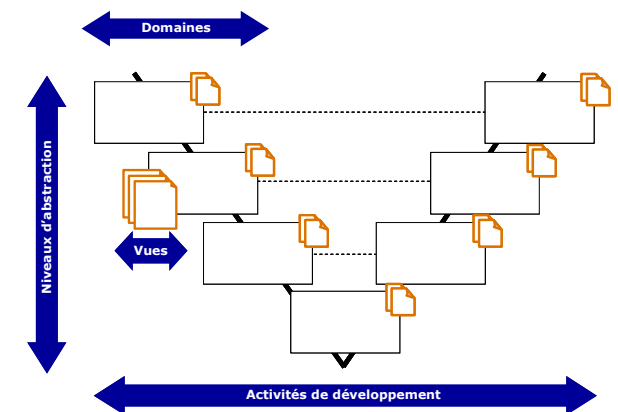
- ▶ Hétérogénéité des niveaux d'abstraction
- ▶ Couplage et transformation entre formalismes
- ▶ Méta-modélisation

[Mosterman 04]

■ Proposition

- ▶ Domaine de la **Modélisation Multi-Paradigme (MPM)**
centré sur 4 axes traitant chacun d'une facette de l'hétérogénéité

- **Domaines** différents : composition
- **Niveaux d'abstraction** différents : conformité
- **Vues** différentes : couplage
- **Activités** différentes : consistance



Modélisation multi-paradigme : approches existantes

- Approches existantes d'horizons très différents
- Proposition
 - ▶ Revue de différents types de techniques
 - ▶ Comparaison de leur façon d'aborder le problème

} Extrait

Modélisation multi-paradigme : approches existantes

■ Approches existantes d'horizons très différents

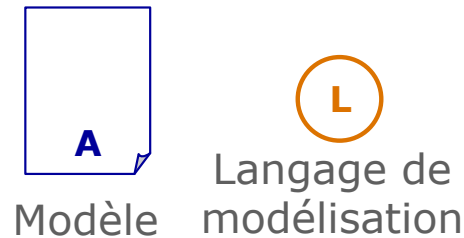
■ Proposition

- ▶ Revue de différents types de techniques
- ▶ Comparaison de leur façon d'aborder le problème

} Extrait

■ Notations

- ▶ Présentation des approches

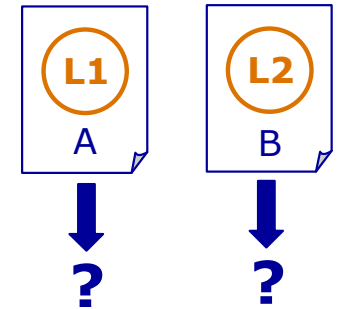


- ▶ Évaluation des approches

- Types de problèmes :
 - Dom** : Domaines différents : composition
 - Niv** : Niveaux d'abstraction différents : conformité
 - Vue** : Vues différentes : couplage
 - Act** : Activités différentes : consistance
- Niveau de traitement :
 - : Problème traité
 - : Problème traité partiellement
 - : Problème non traité

Modélisation multi-paradigme : revue (extrait 1/2)

- Techniques basées sur la transformation des modèles
 - ▶ Problème : choix du langage cible



Modélisation multi-paradigme : revue (extrait 1/2)

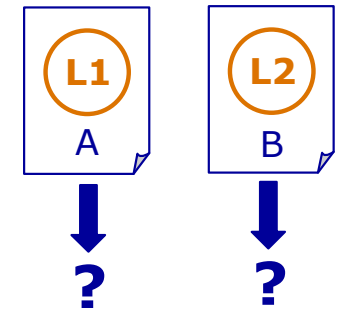
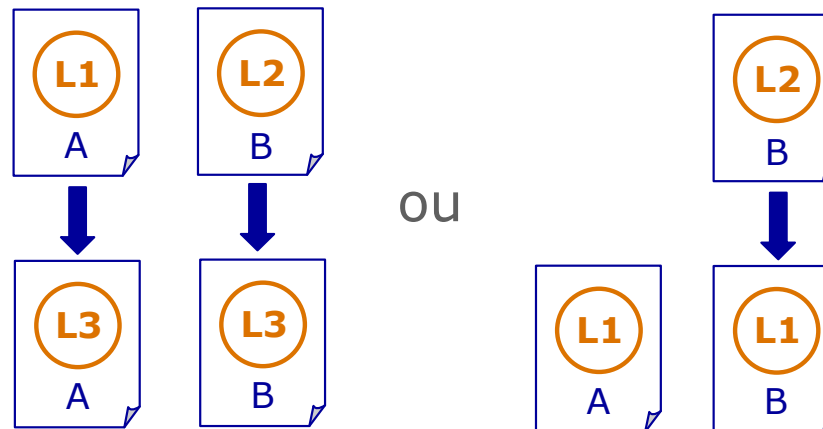
■ Techniques basées sur la transformation des modèles

▶ Problème : choix du langage cible

▶ Transformation vers un langage « existant »

● Exemples :

- ◆ *ATOM3*
- ◆ *GME*
- ◆ *HETS*



Modélisation multi-paradigme : revue (extrait 1/2)

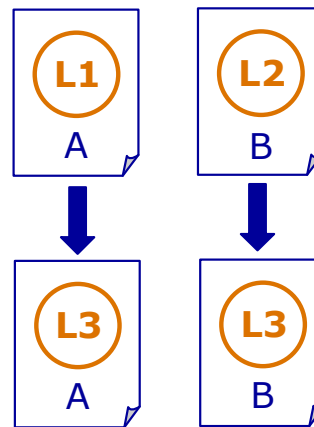
■ Techniques basées sur la transformation des modèles

▶ Problème : choix du langage cible

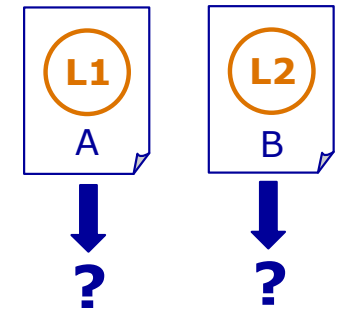
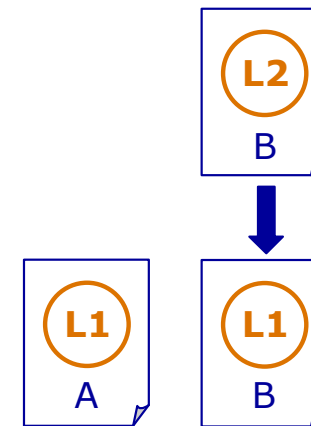
▶ Transformation vers un langage « existant »

● Exemples :

- ◆ *ATOM3*
- ◆ *GME*
- ◆ *HETS*



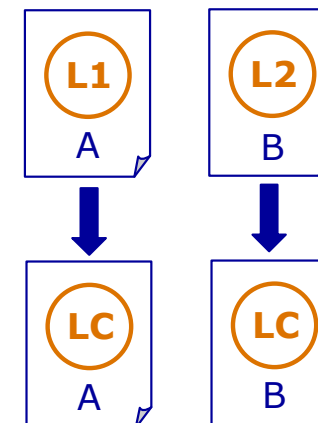
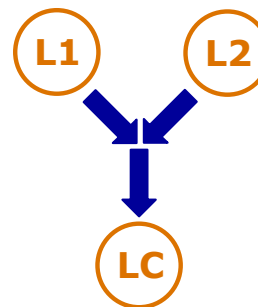
ou



▶ Composition des langages

● Exemples :

- ◆ *GME*
- ◆ *Semantic Units*



Modélisation multi-paradigme : revue (extrait 1/2)

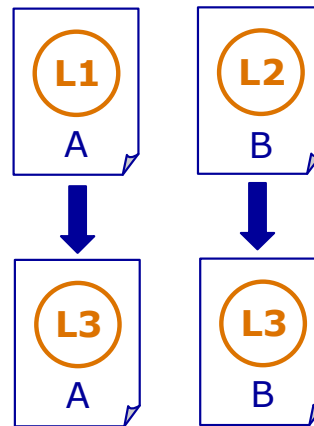
■ Techniques basées sur la transformation des modèles

▶ Problème : choix du langage cible

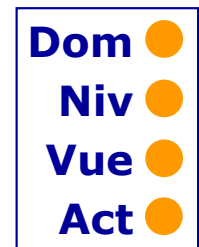
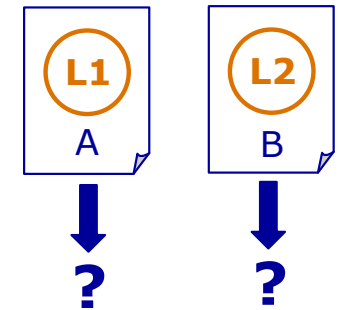
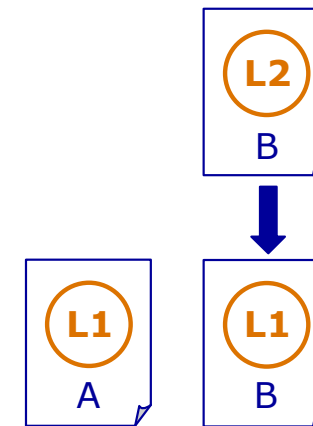
▶ Transformation vers un langage « existant »

● Exemples :

- ◆ *ATOM3*
- ◆ *GME*
- ◆ *HETS*



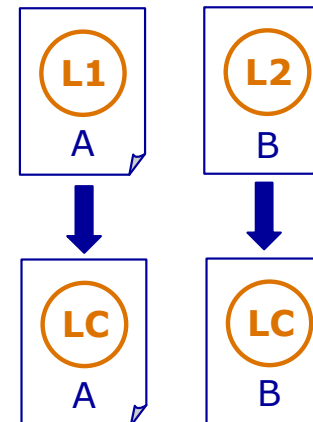
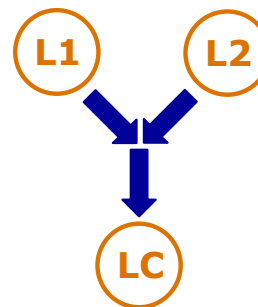
OU



▶ Composition des langages

● Exemples :

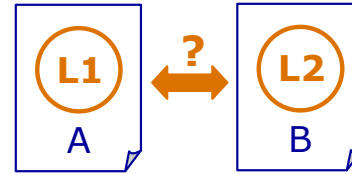
- ◆ *GME*
- ◆ *Semantic Units*



Modélisation multi-paradigme : revue (extrait 2/2)

■ Techniques basées sur la composition des modèles

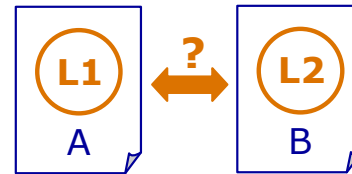
▶ Problème :



Modélisation multi-paradigme : revue (extrait 2/2)

■ Techniques basées sur la composition des modèles

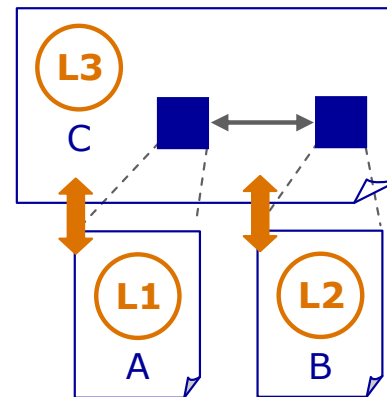
▶ Problème :



▶ Composition verticale :

● Exemples :

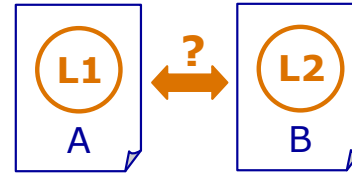
◆ *PtolemyII*



Modélisation multi-paradigme : revue (extrait 2/2)

■ Techniques basées sur la composition des modèles

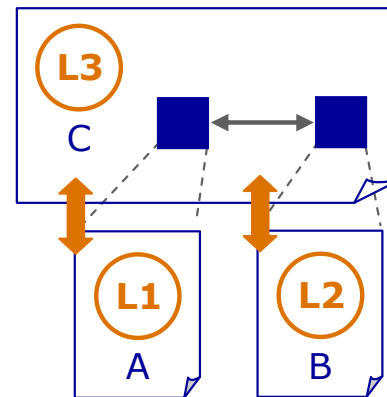
▶ Problème :



▶ Composition verticale :

● Exemples :

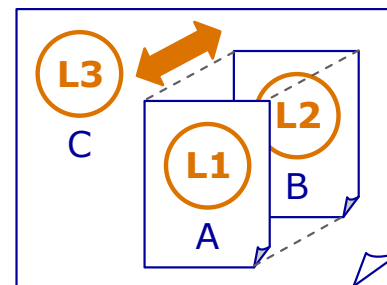
◆ *PtolemyII*



▶ Composition horizontale :

● Exemples :

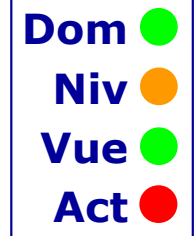
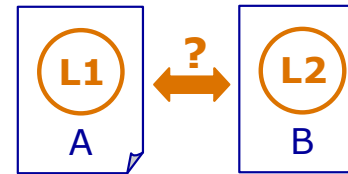
◆ *Rosetta*



Modélisation multi-paradigme : revue (extrait 2/2)

■ Techniques basées sur la composition des modèles

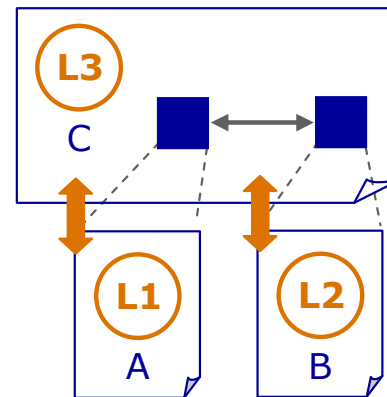
▶ Problème :



▶ Composition verticale :

● Exemples :

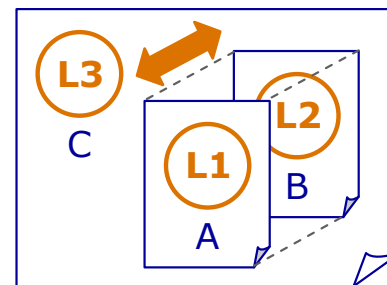
◆ *PtolemyII*



▶ Composition horizontale :

● Exemples :

◆ *Rosetta*




Modélisation multi-paradigme : conclusion

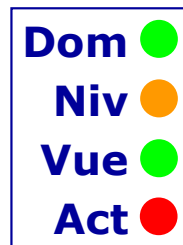
■ Approches existantes

- ▶ Approches **nombreuses et d'horizon très différents**
 - Extrait : **transformation** de modèles et **composition** de modèles
- ▶ Qualités **difficiles à comparer**
 - Extrait : comparaison vis-à-vis des différentes facettes du problème de l'hétérogénéité

■ Conclusion

- ▶ **Aucun type d'approche ne traite tous les problèmes**
- ▶ Les différents types d'approches sont **complémentaires**

 Focus sur la **composition de modèles**
orientée vers le **calcul du comportement**
(exécution, simulation)



Introduction

Hétérogénéité des modèles

Modélisation multi-paradigme

▶ **ModHel'X**

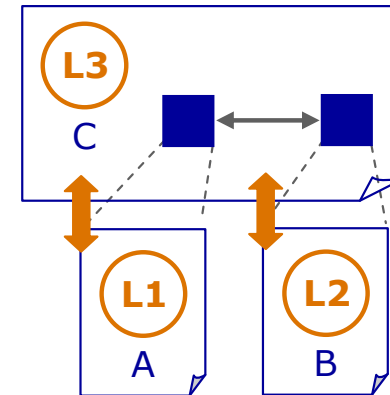
Framework expérimental

Conclusion & perspectives

Composition hiérarchique de modèles orientée exécution

■ Objectif

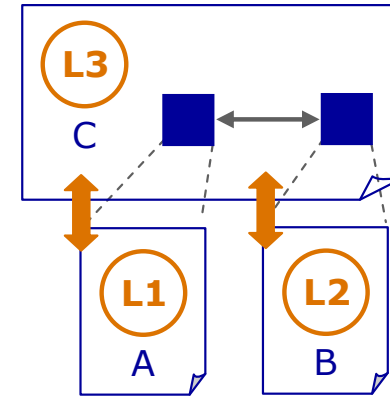
- ▶ Composition structurelle via la hiérarchie
- ▶ Exécution des modèles composés



Composition hiérarchique de modèles orientée exécution

■ Objectif

- ▶ Composition structurelle via la hiérarchie
- ▶ Exécution des modèles composés



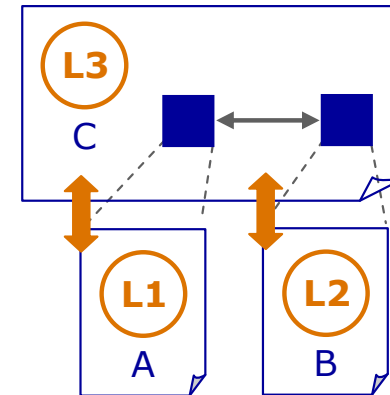
■ Problématiques

1. Comment exprimer la sémantique des langages de façon précise ?
 - Exécution, raisonnement formel ?
2. Comment définir la composition au niveau sémantique ?
 - Mécanisme de composition, sémantique résultante ?

Composition hiérarchique de modèles orientée exécution

■ Objectif

- ▶ Composition structurelle via la hiérarchie
- ▶ Exécution des modèles composés



■ Problématiques

1. Comment exprimer la sémantique des langages de façon précise ?
 - Exécution, raisonnement formel ?
2. Comment définir la composition au niveau sémantique ?
 - Mécanisme de composition, sémantique résultante ?

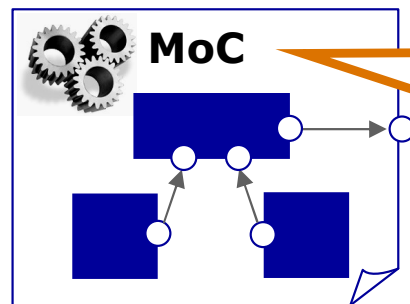
■ Approches existantes

- ▶ Exemple : PtolemyII
- ▶ Principe :
 - Représentation syntaxique unique
 - Support pour l'expression de la sémantique des langages

Modèle de calcul (MoC)

■ Modèle de calcul (MoC) [Lee 91-08, Ptolemy]

- ▶ Ensemble de **caractéristiques communes** à une famille de langages de modélisation
 - Concurrence, synchronisation, communication, temps...
- ▶ **Support pour l'expression de la sémantique** sur une syntaxe unique à base de composants

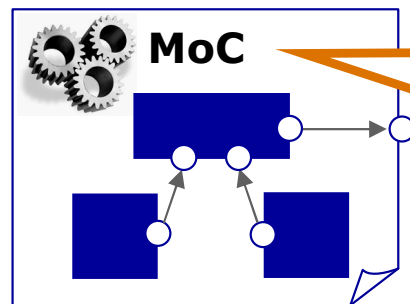


Ensemble des règles qui définissent le comportement d'un modèle en **combinant le comportement des composants**

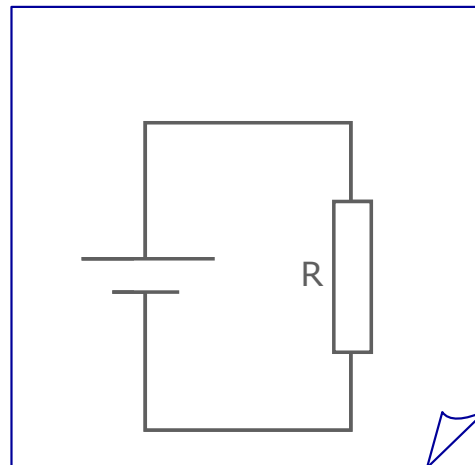
Modèle de calcul (MoC)

■ Modèle de calcul (MoC) [Lee 91-08, Ptolemy]

- ▶ Ensemble de **caractéristiques communes** à une famille de langages de modélisation
 - Concurrence, synchronisation, communication, temps...
- ▶ Support pour l'expression de la **sémantique** sur une syntaxe unique à base de composants



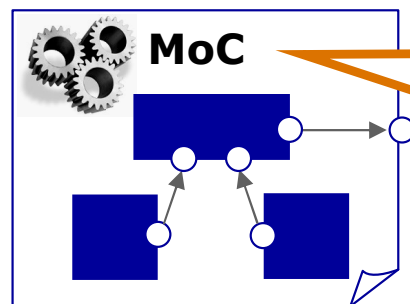
Ensemble des règles qui définissent le comportement d'un modèle en **combinant le comportement des composants**



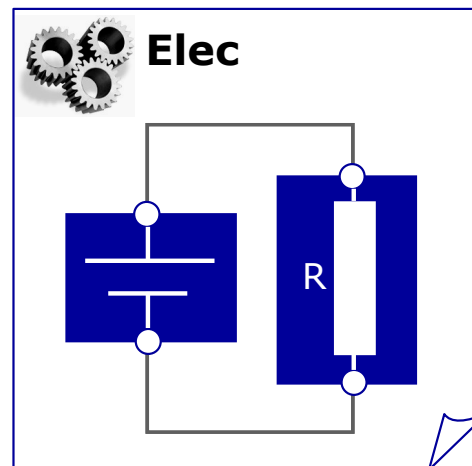
Modèle de calcul (MoC)

■ Modèle de calcul (MoC) [Lee 91-08, Ptolemy]

- ▶ Ensemble de **caractéristiques communes** à une famille de langages de modélisation
 - Concurrence, synchronisation, communication, temps...
- ▶ Support pour l'expression de la **sémantique** sur une syntaxe unique à base de composants

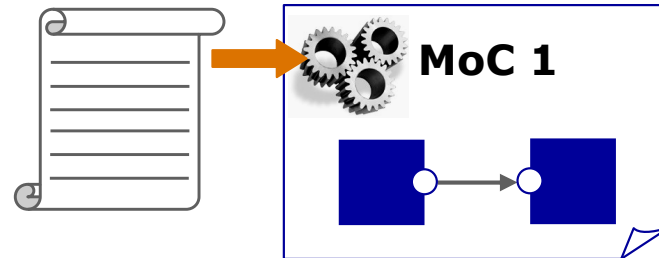


Ensemble des règles qui définissent le comportement d'un modèle en **combinant le comportement des composants**

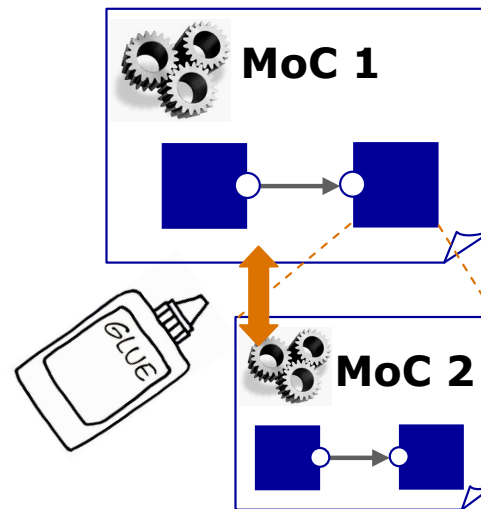


ModHel'X : objectifs

1. Fournir un support pour la **description des MoCs**



2. Fournir un support pour la description des **mécanismes d'adaptation sémantique entre MoCs**

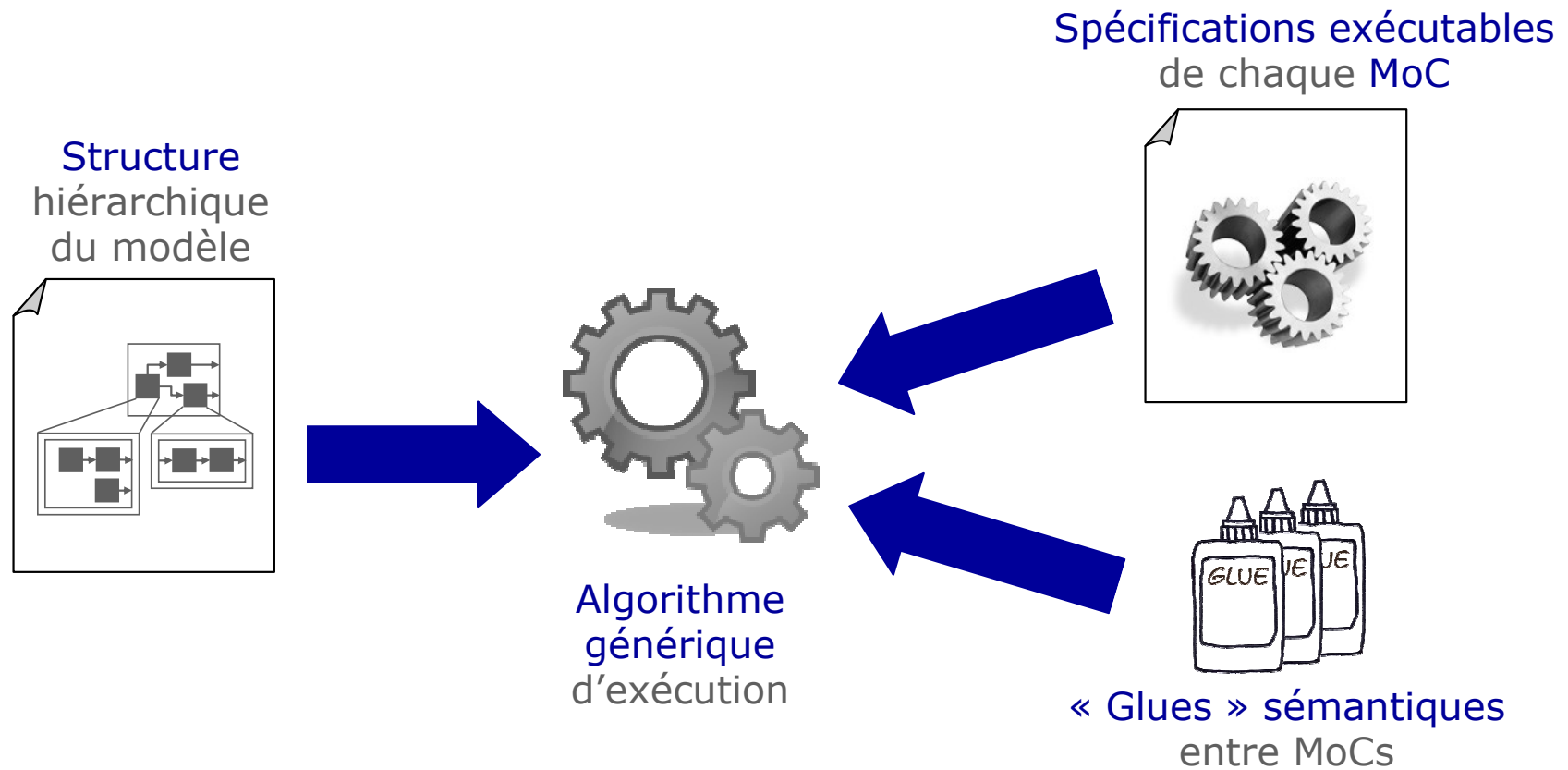


3. Permettre l'**exécution des modèles**

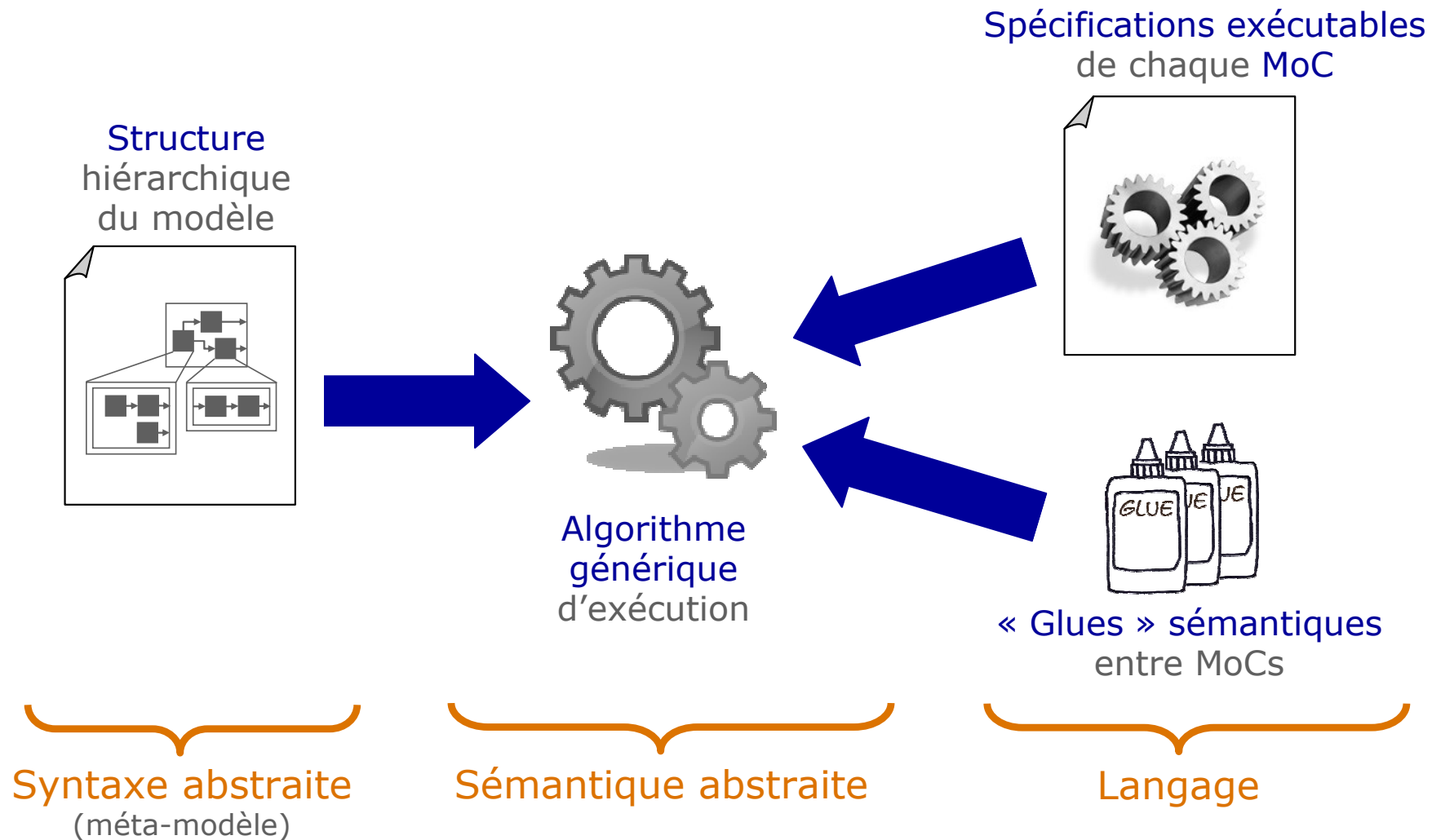
A long terme, permettre le raisonnement formel



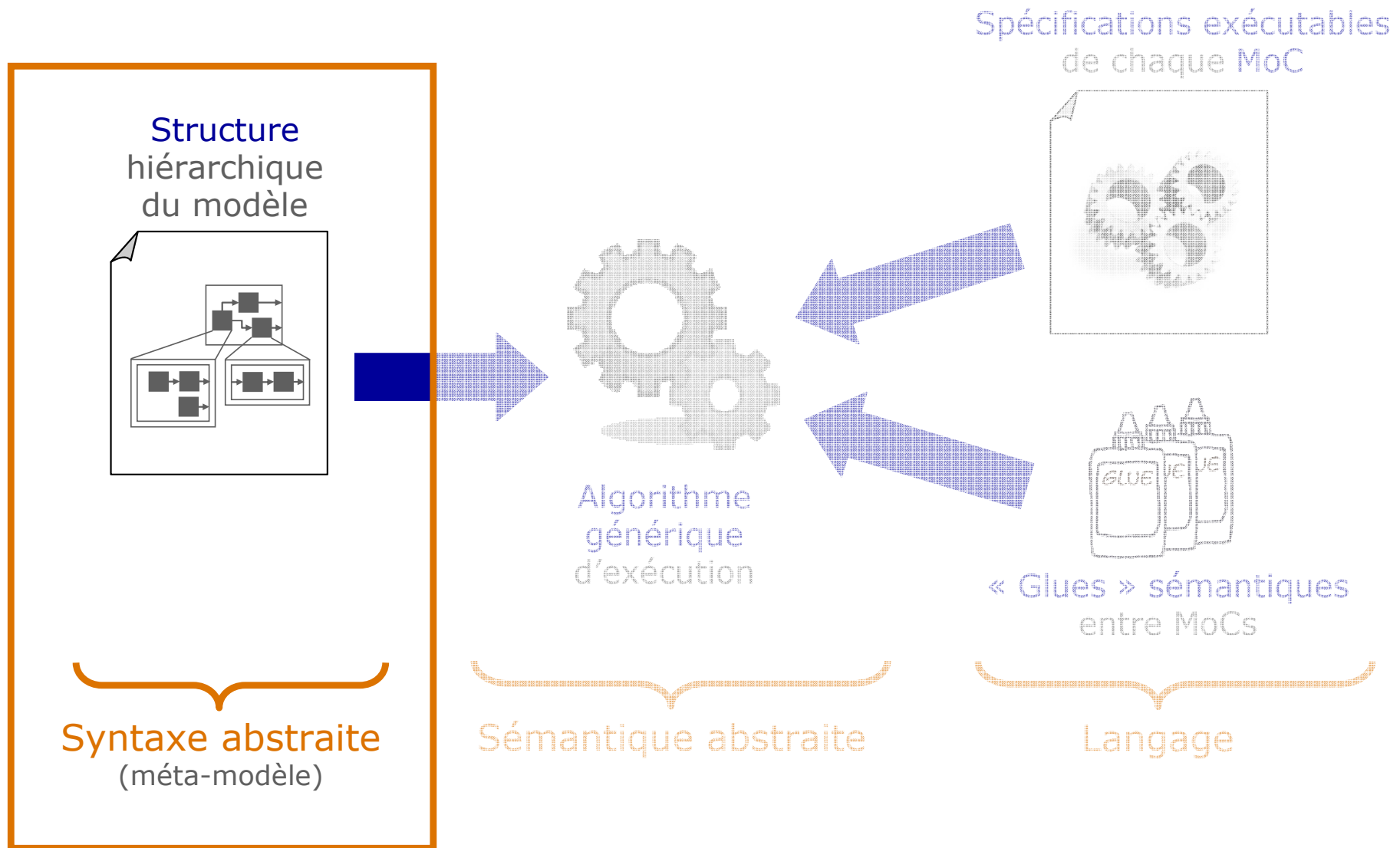
ModHel'X : architecture générale



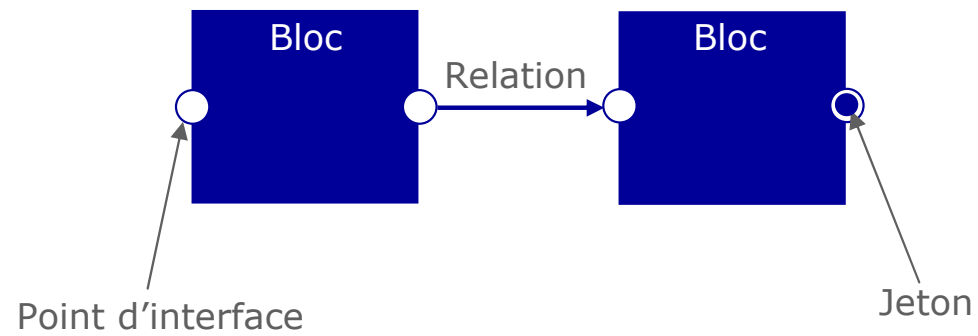
ModHel'X : architecture générale



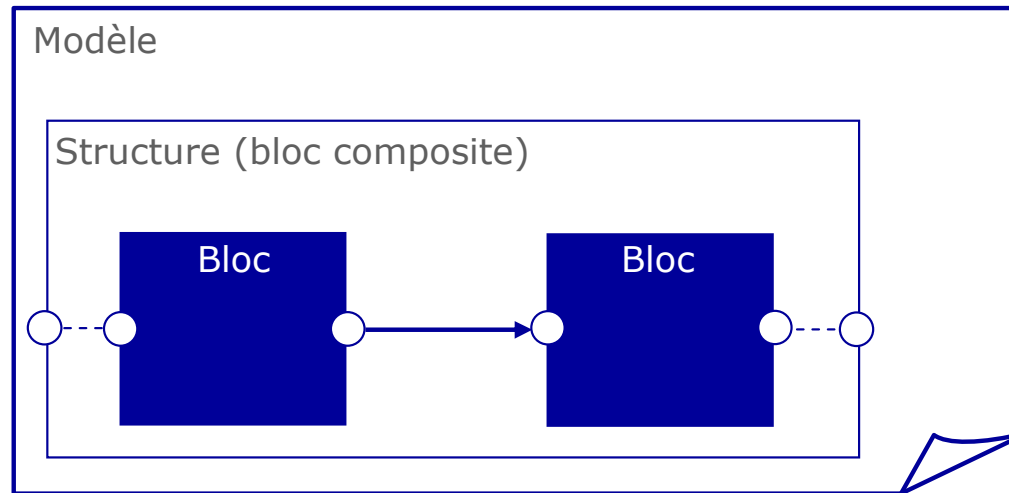
ModHel'X : architecture générale



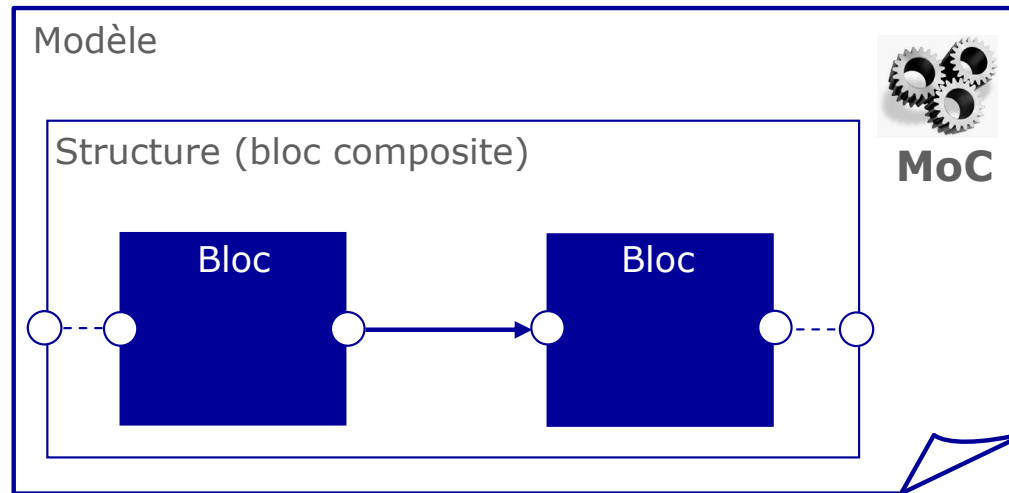
ModHel'X : syntaxe abstraite – blocs, relations, etc.



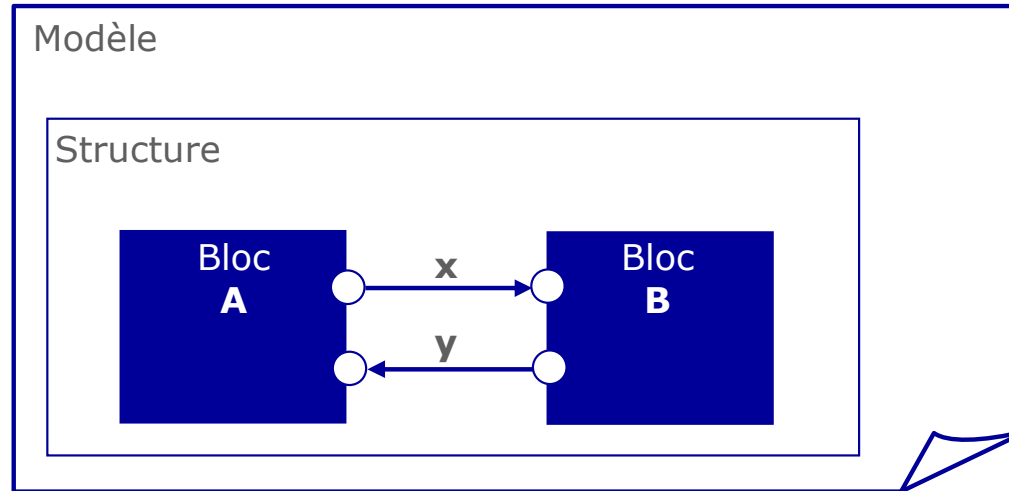
ModHel'X : syntaxe abstraite – modèle



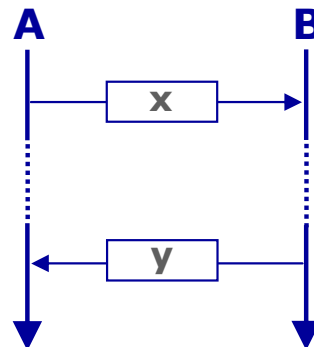
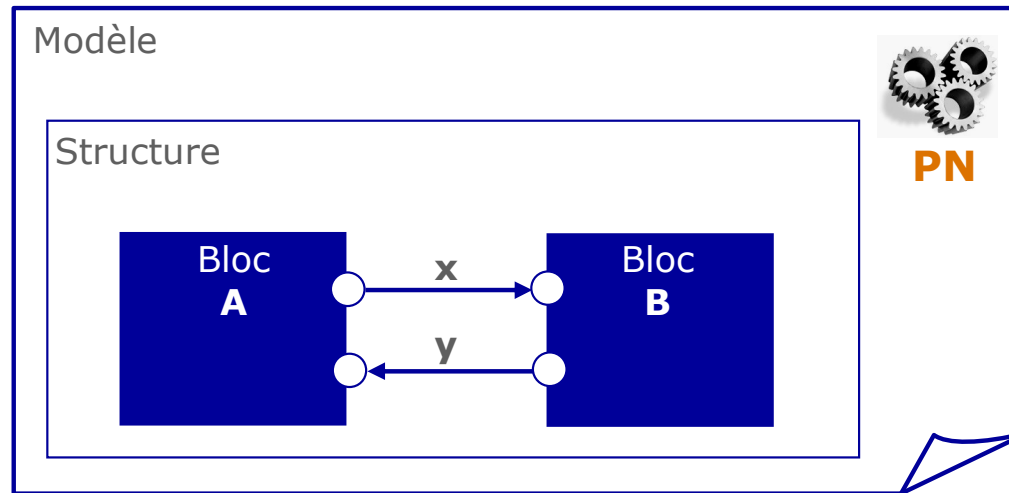
ModHel'X : syntaxe abstraite – MoC



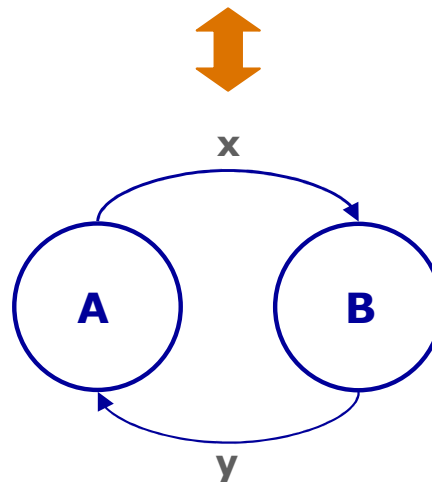
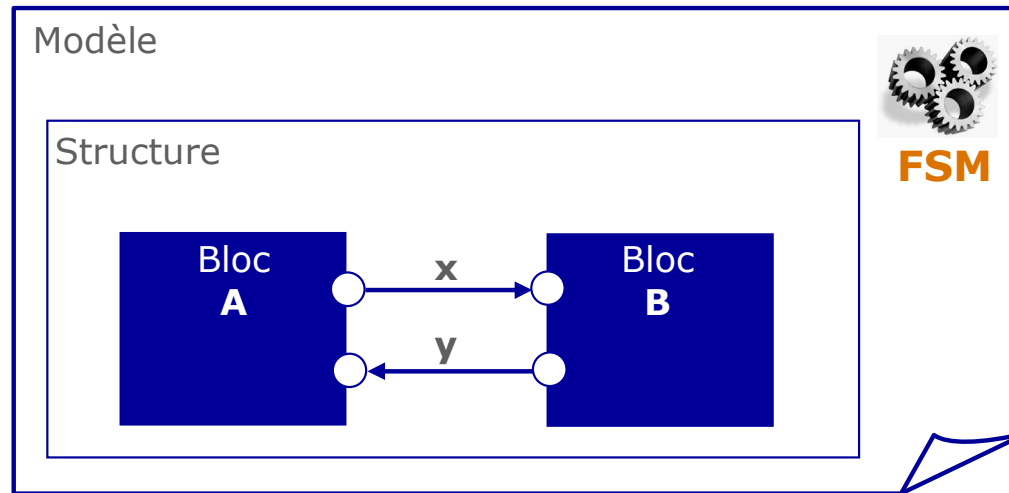
ModHel'X : syntaxe abstraite – MoC



ModHel'X : syntaxe abstraite – MoC

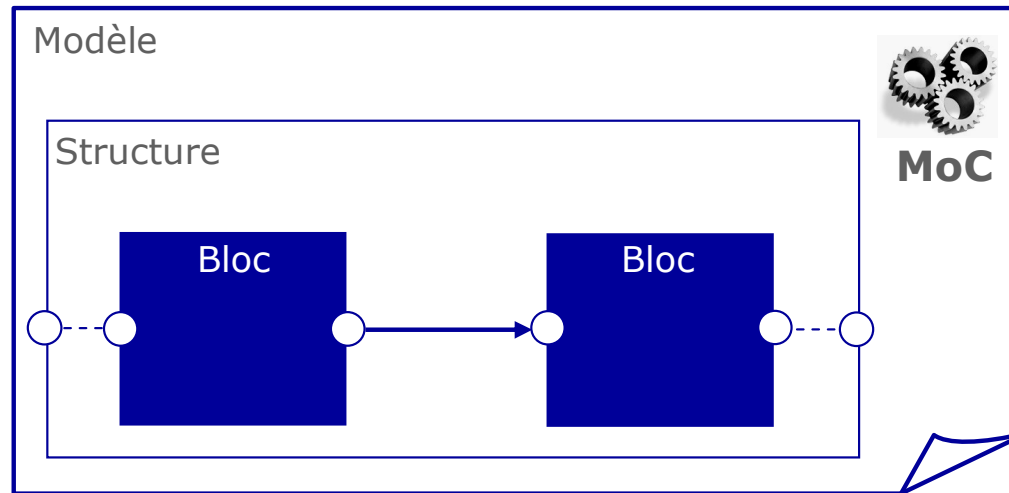


ModHel'X : syntaxe abstraite – MoC

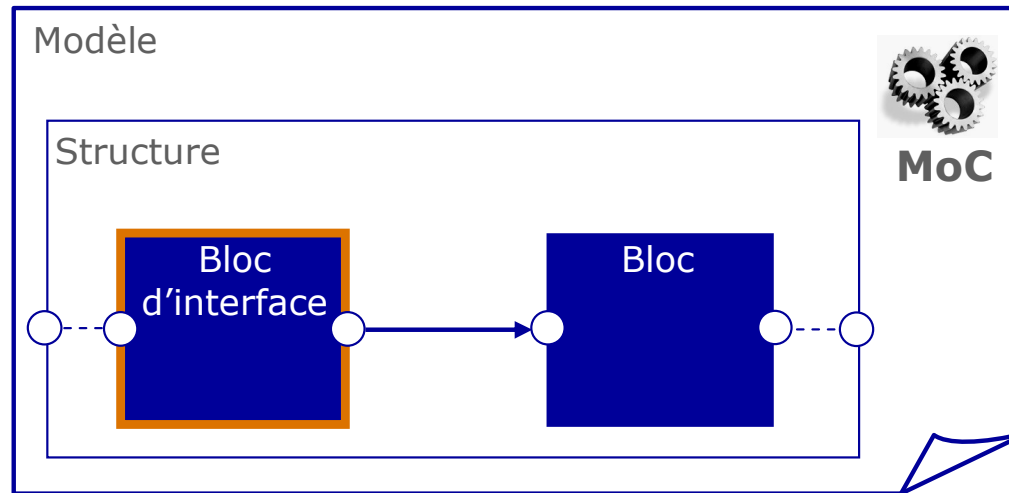


FSM : Finite State Machine

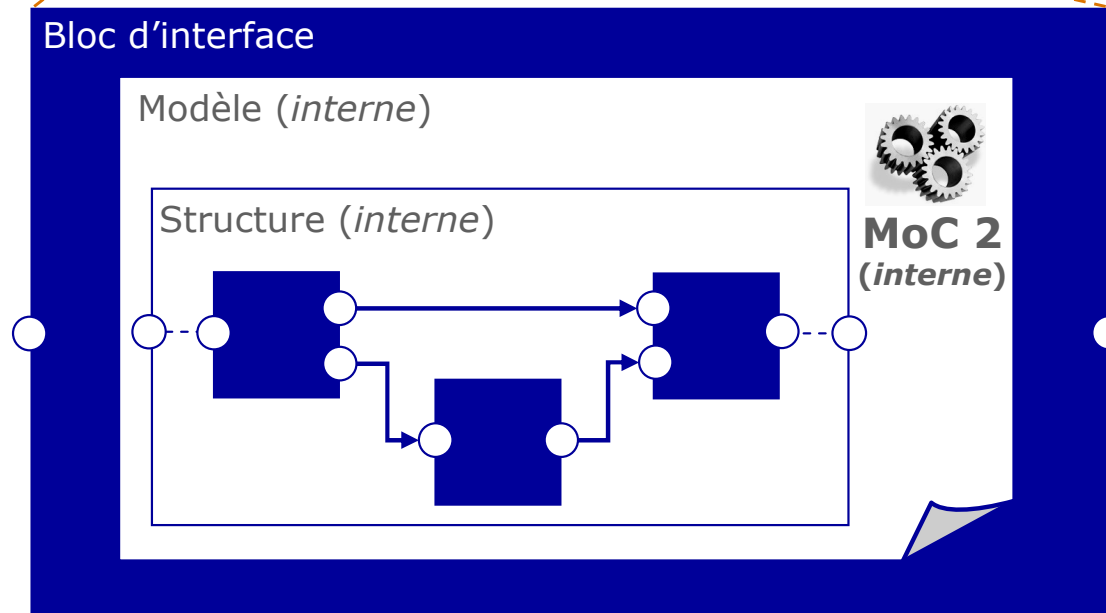
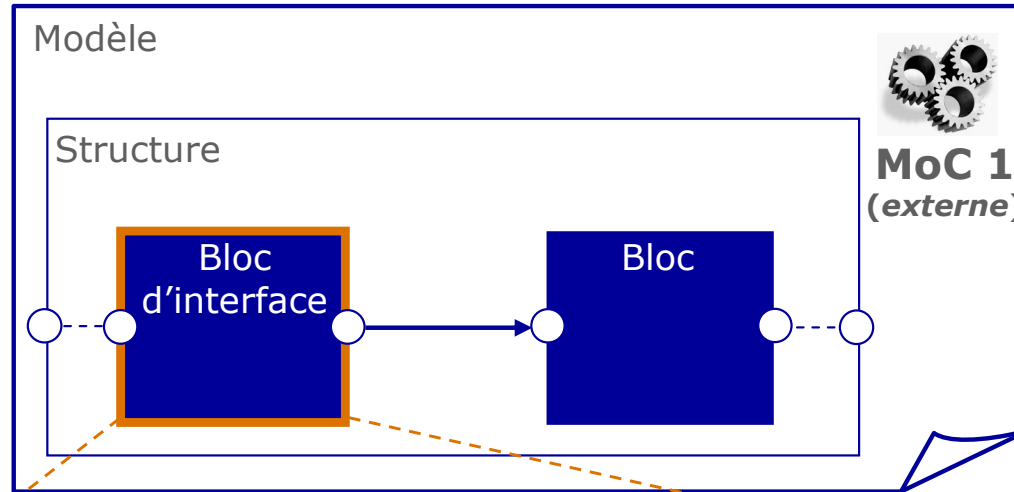
ModHel'X : syntaxe abstraite – bloc d'interface



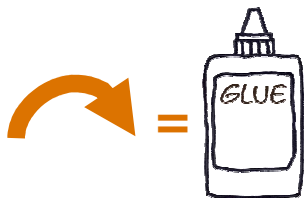
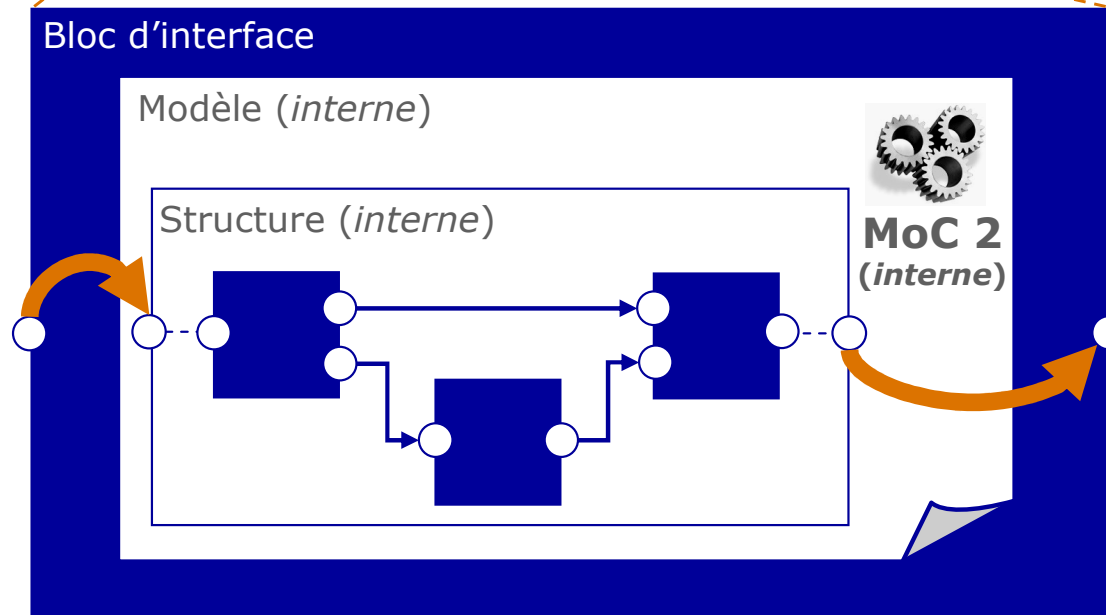
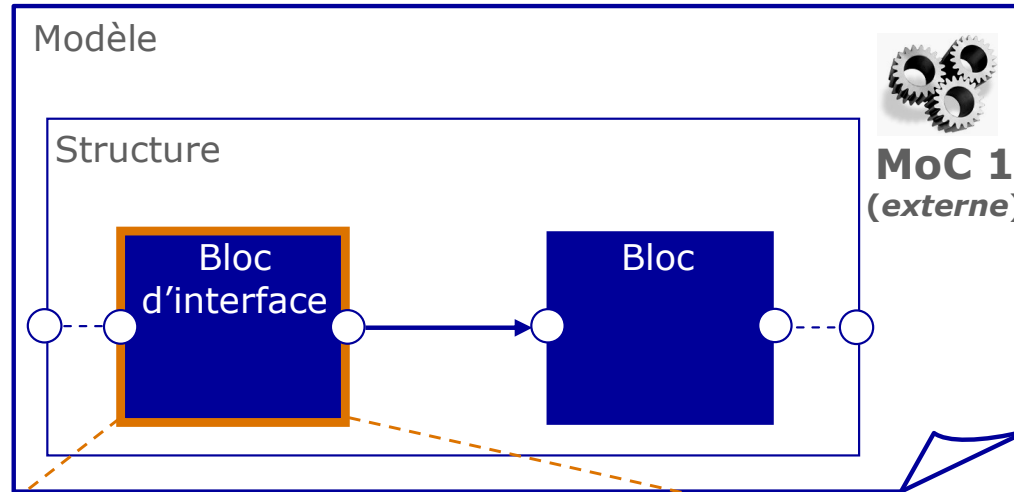
ModHel'X : syntaxe abstraite – bloc d'interface



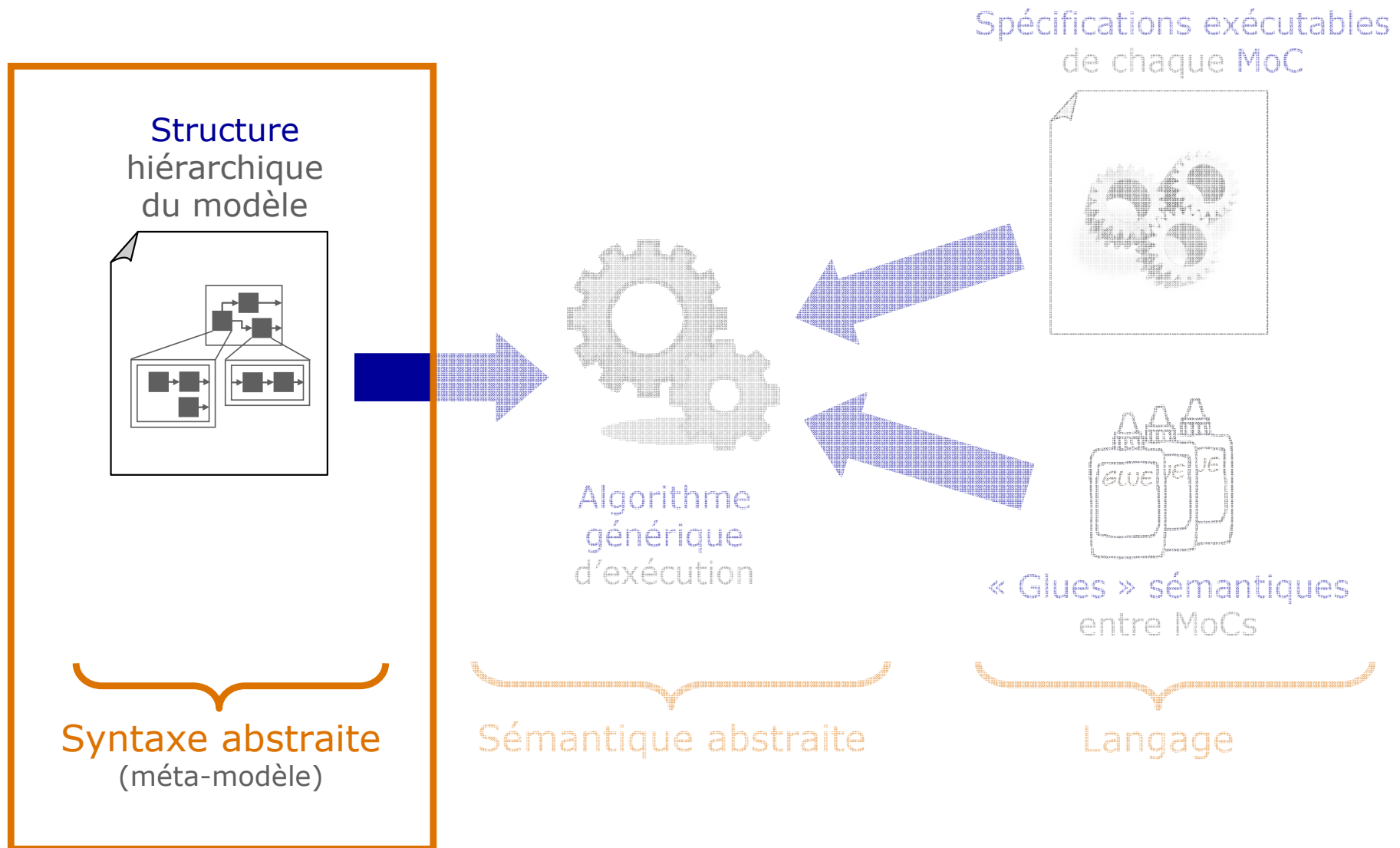
ModHel'X : syntaxe abstraite – bloc d'interface



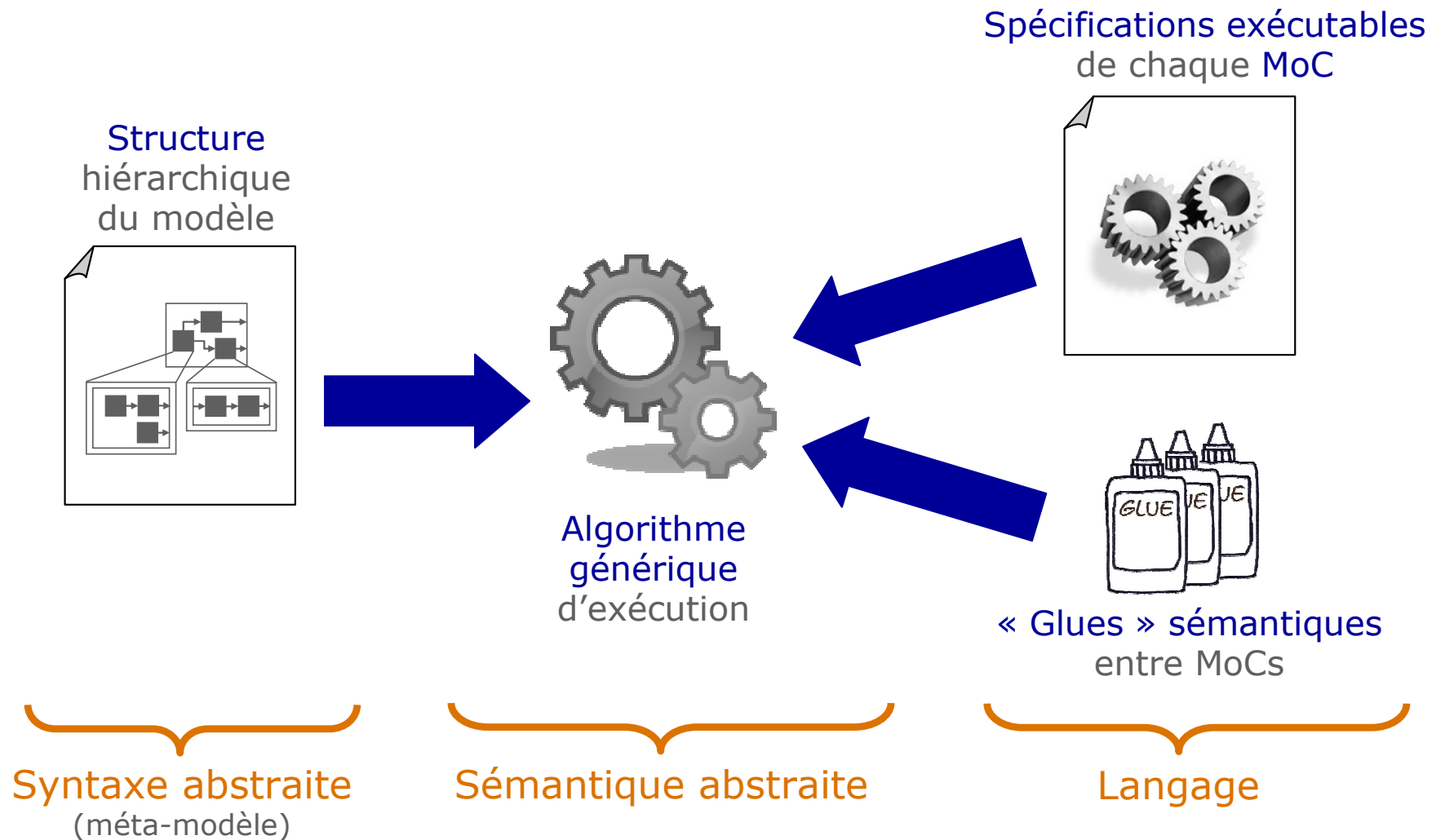
ModHel'X : syntaxe abstraite – bloc d'interface



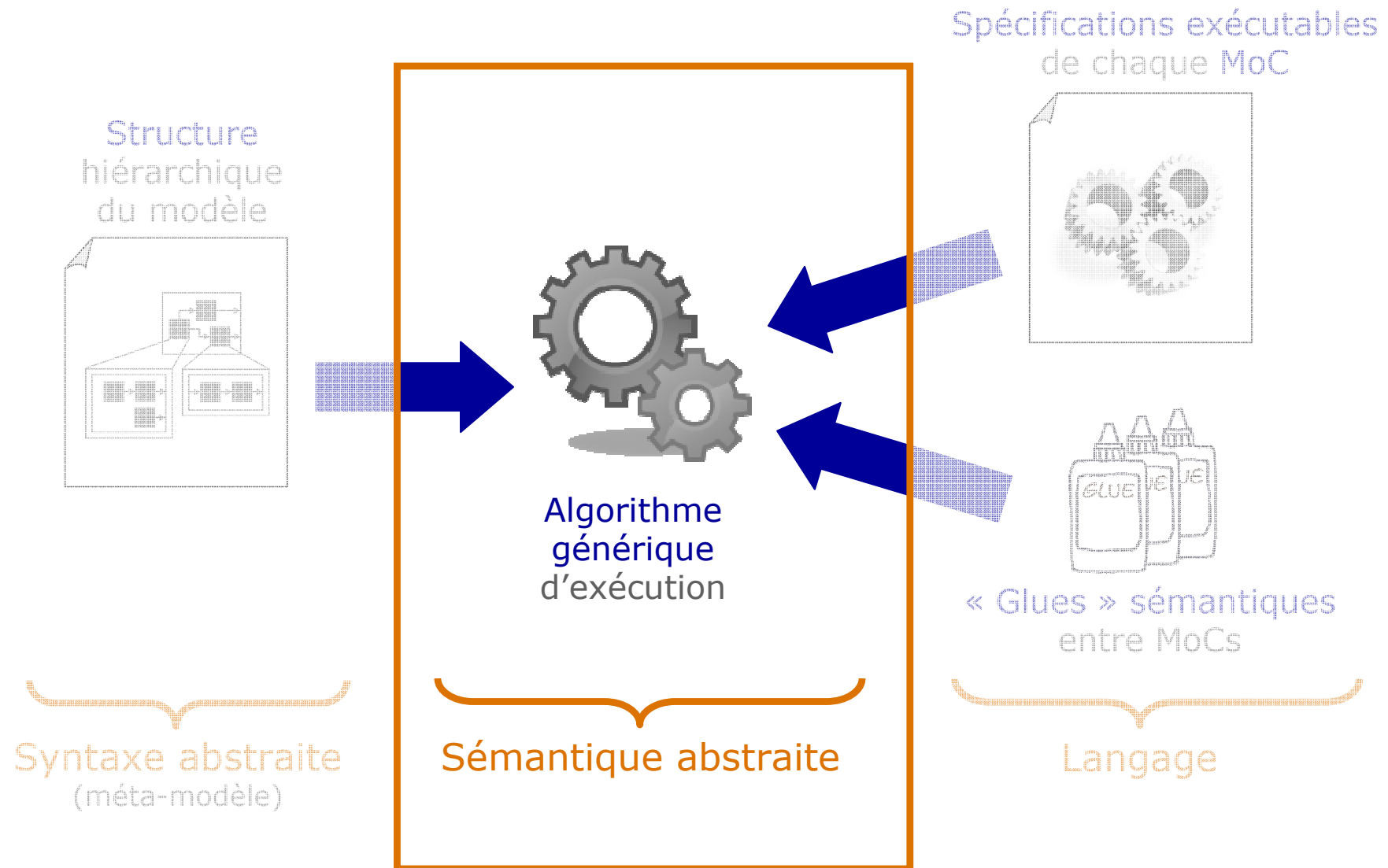
ModHel'X : architecture générale



ModHel'X : architecture générale



ModHel'X : architecture générale



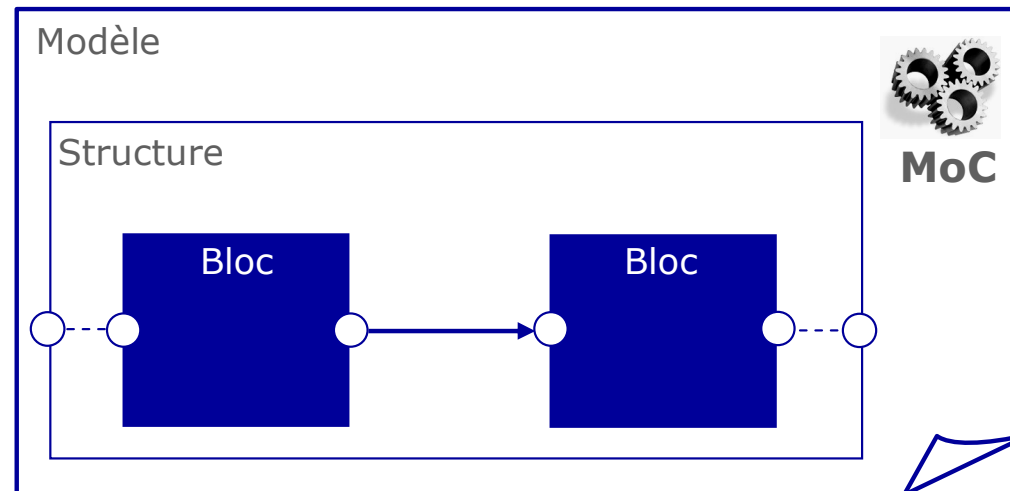
ModHel'X : sémantique abstraite – exécution

■ Problème :

- ▶ Exécuter un modèle décrit dans notre syntaxe
- ▶ Selon les règles données par son MoC

■ Proposition :

- ▶ Algorithme générique pour l'exécution des modèles
- ▶ Personnalisable avec les règles d'un MoC



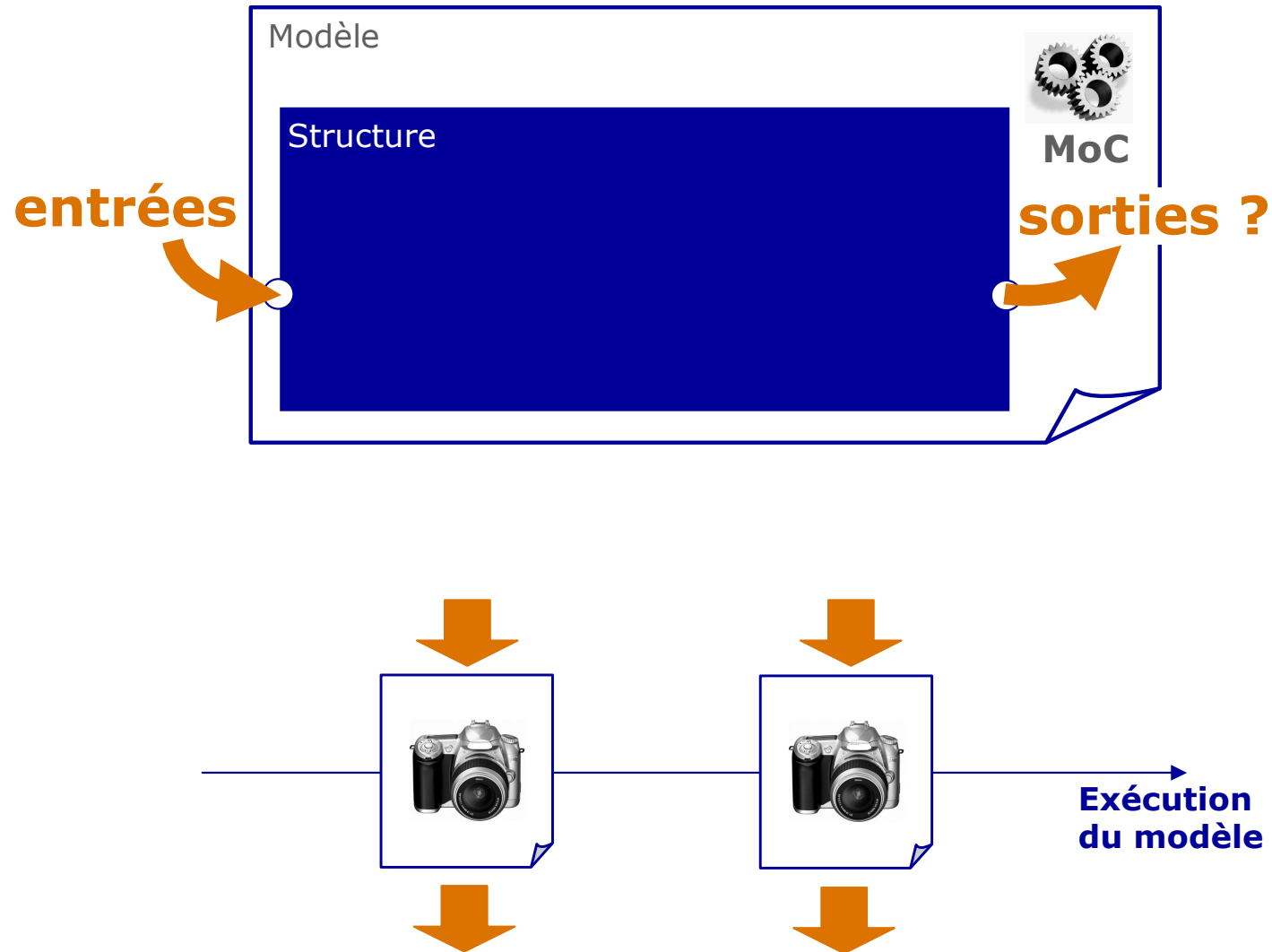
ModHel'X : sémantique abstraite – exécution

- Exécution d'un modèle = **observation de son comportement**



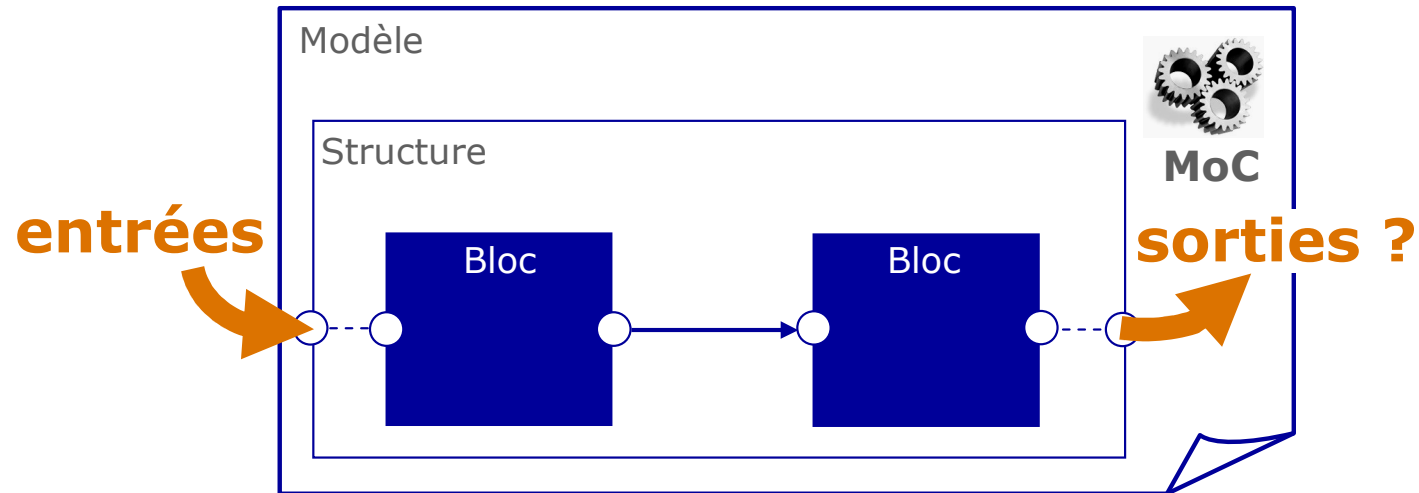
ModHel'X : sémantique abstraite – exécution

- Exécution d'un modèle = **série de snapshots** (observations)

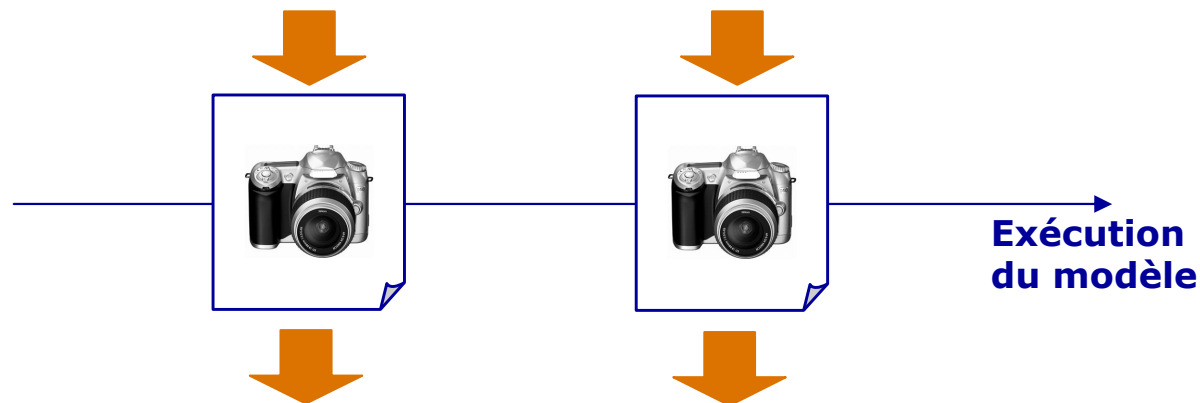


ModHel'X : sémantique abstraite – exécution

- Exécution d'un modèle = **série de snapshots** (observations)

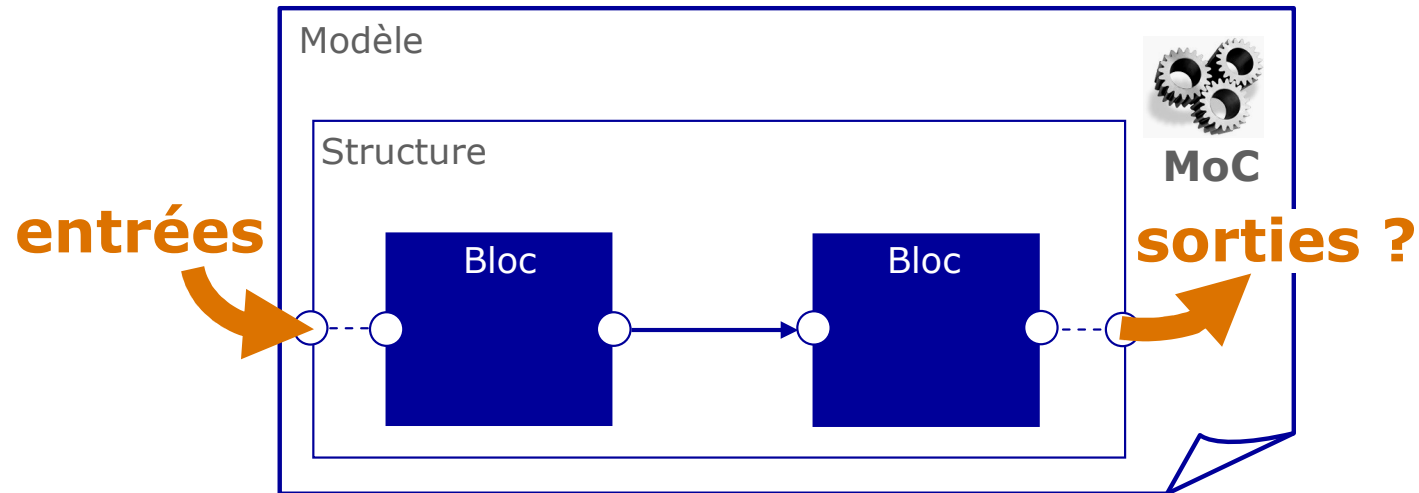


Comment calculer un snapshot ?



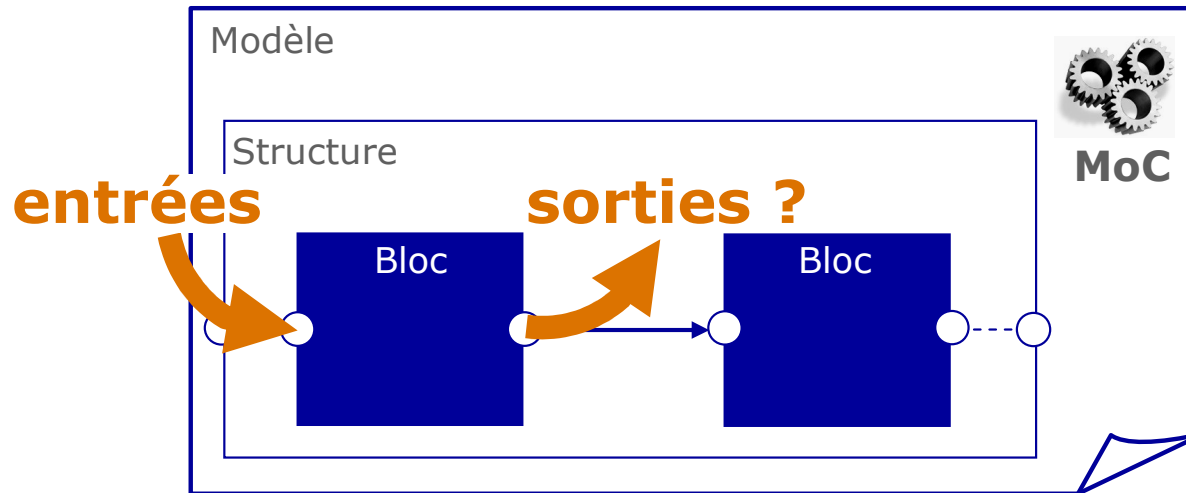
ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle



ModHel'X : sémantique abstraite – snapshot

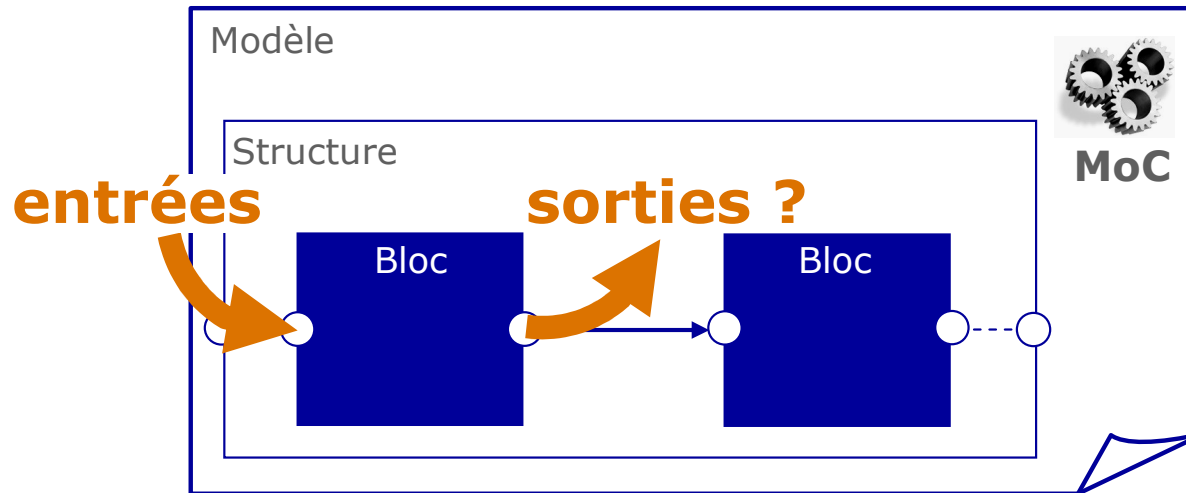
- Snapshot = combinaison d'observations des blocs du modèle



- ▶ Idée : observer les blocs séquentiellement

ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

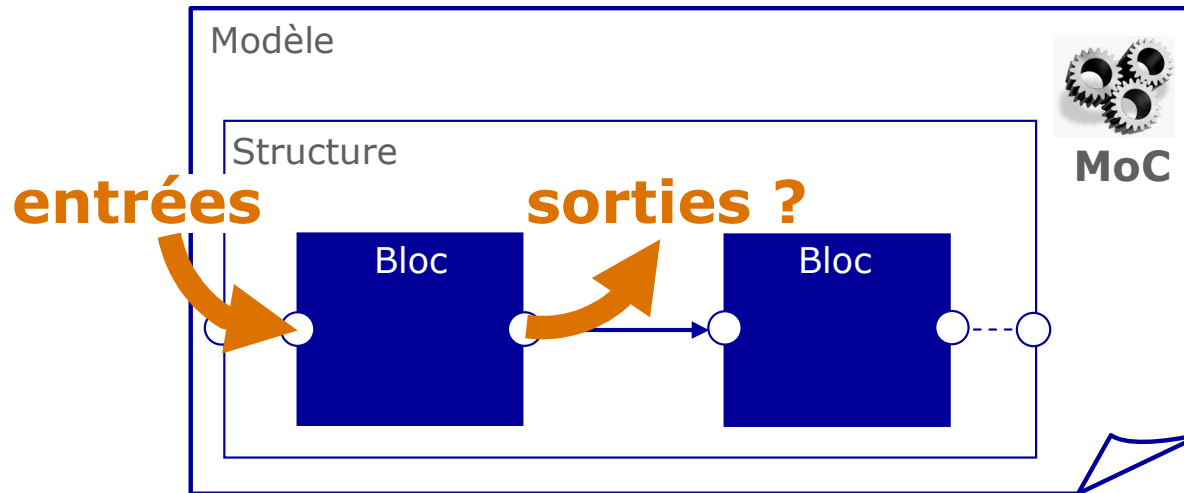


- ▶ Idée : observer les blocs **séquentiellement**

- (**snapDet**)
- Choisir un bloc à observer (schedule)
 - Observer le bloc choisi (update)
 - Propager les données observées (propagate)

ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

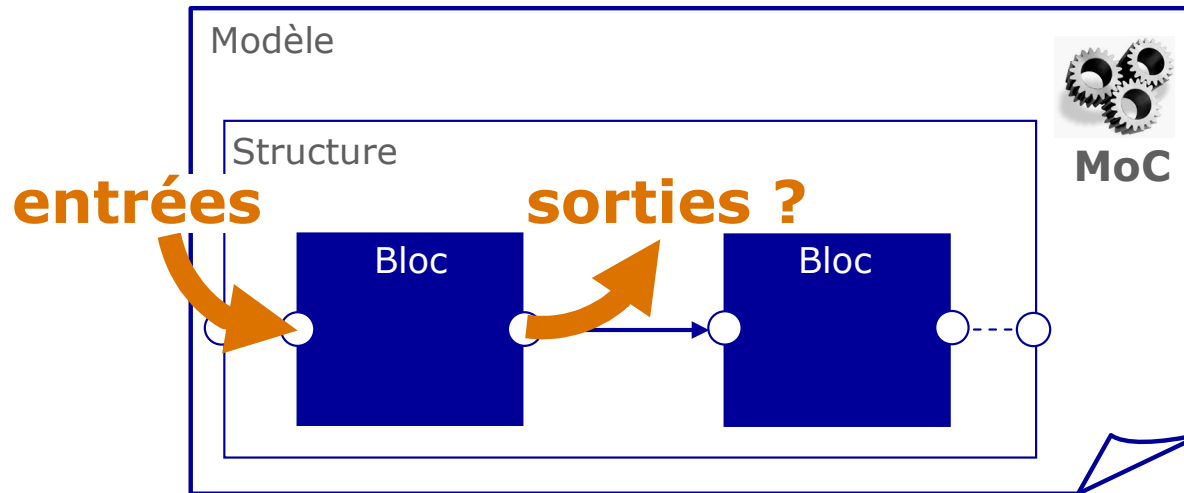


► Questions :

- Dans quel ordre observer les blocs ?
 - ◆ *Ordre topologique ? Fonction des données ?*
- Comment propager les résultats des observations ?
 - ◆ *A quel bloc ? Comment ?*
- Quand arrêter d'observer les blocs ?
 - ◆ *Lorsque tous les blocs ont été observés une fois ? Plusieurs fois ?*

ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle



► Questions :

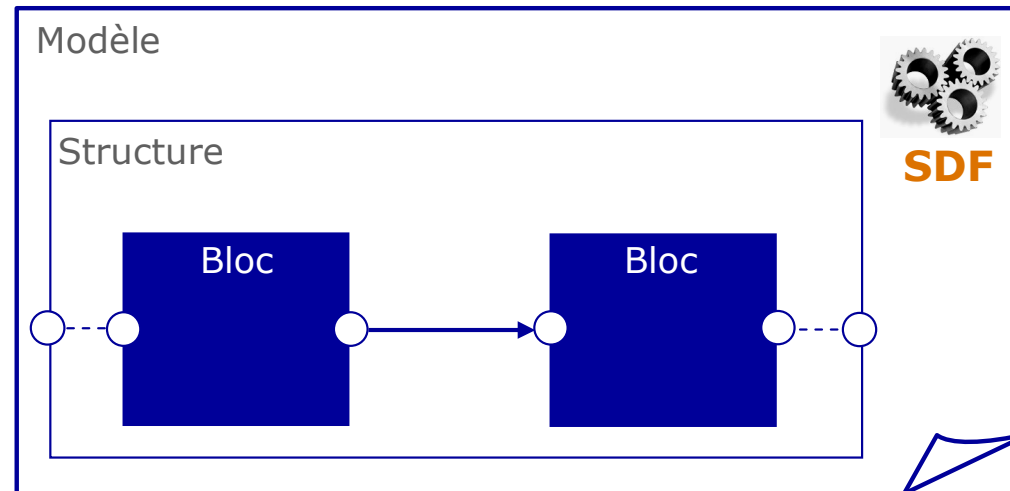
- Dans quel ordre observer les blocs ?
 - ◆ *Ordre topologique ? Fonction des données ?*
- Comment propager les résultats des observations ?
 - ◆ *A quel bloc ? Comment ?*
- Quand arrêter d'observer les blocs ?
 - ◆ *Lorsque tous les blocs ont été observés une fois ? Plusieurs fois ?*

➔ Dépend des règles du MoC !

- Concurrence, synchronisation, communication, temps

ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle



▶ Exemple :

- MoC SDF (Synchronous DataFlow)

- ◆ *Processus de traitement*
- ◆ *Communication par flots de données*

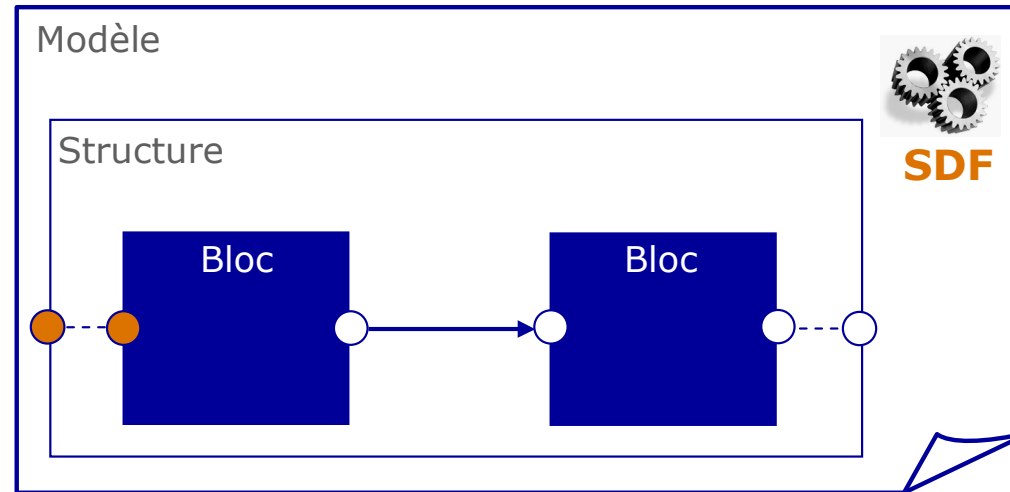
- Exécution pour SDF

- ◆ *Choix des blocs par ordonnancement statique*
- ◆ *Propagation d'ensembles d'échantillons de données le long des relations*

ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

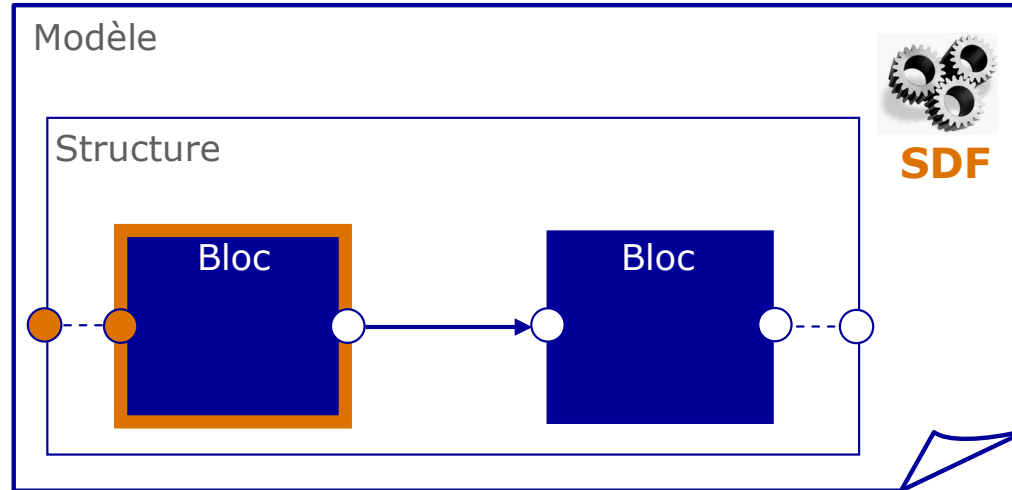
Fourniture
des entrées



ModHel'X : sémantique abstraite – snapshot

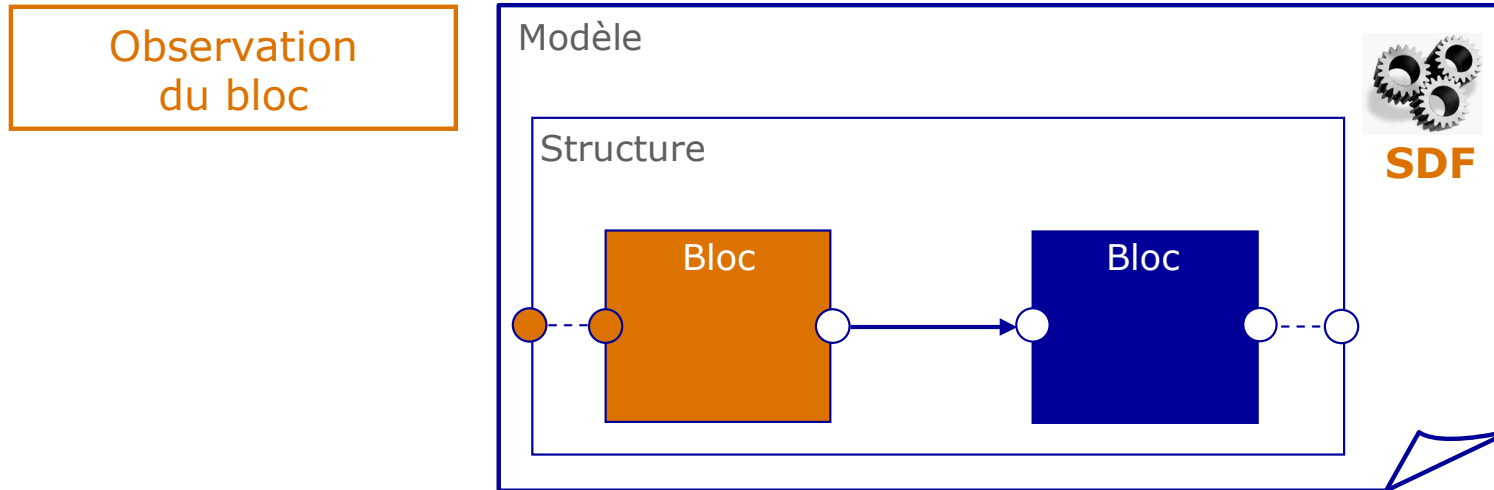
- Snapshot = combinaison d'observations des blocs du modèle

Choix
du bloc



ModHel'X : sémantique abstraite – snapshot

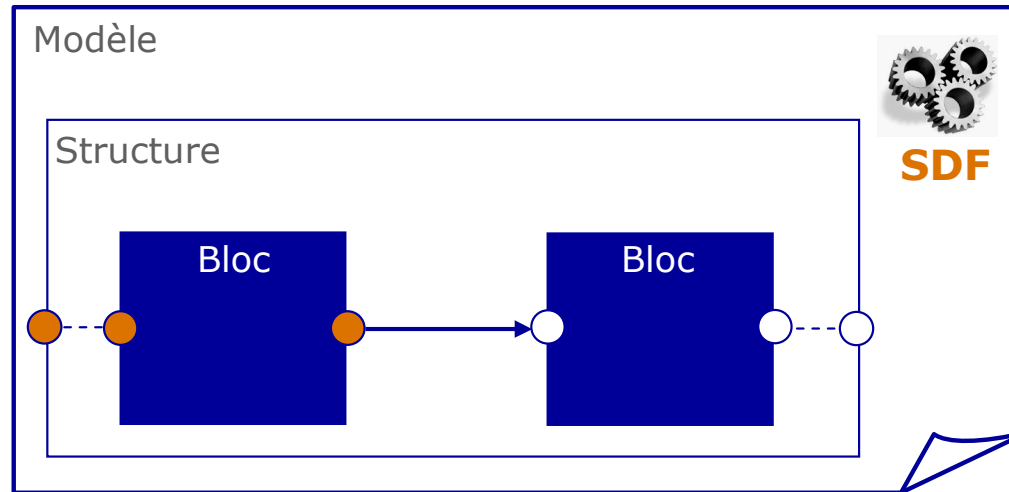
- Snapshot = combinaison d'observations des blocs du modèle



ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

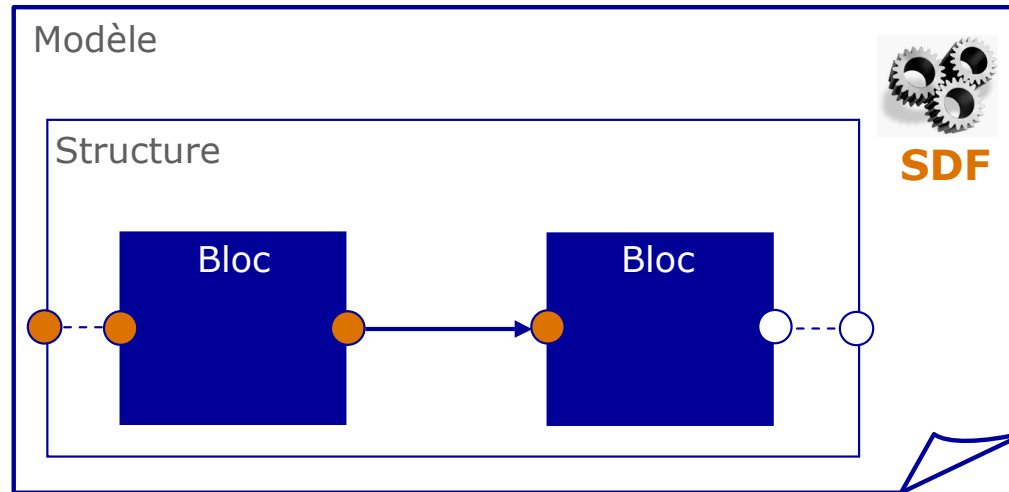
Après observation
du bloc



ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

Propagation
des données

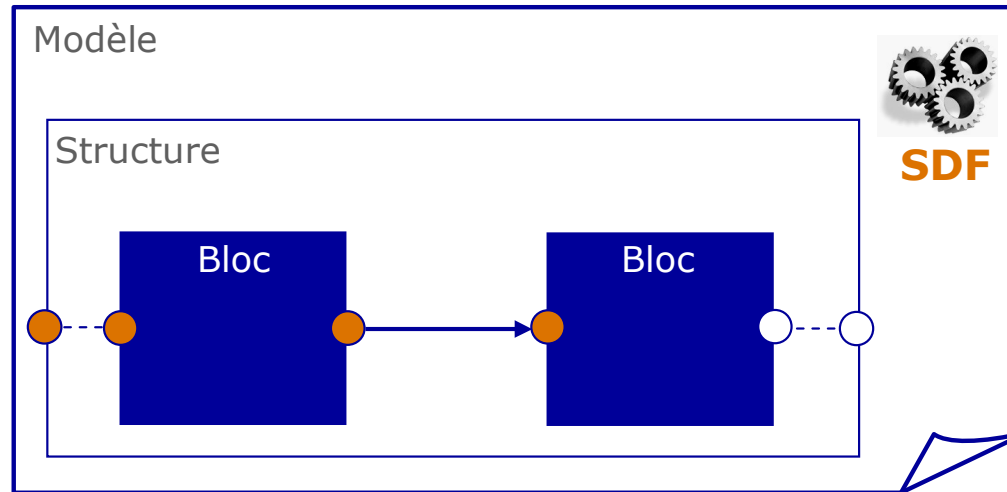


ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

Snapshot
déterminé ?

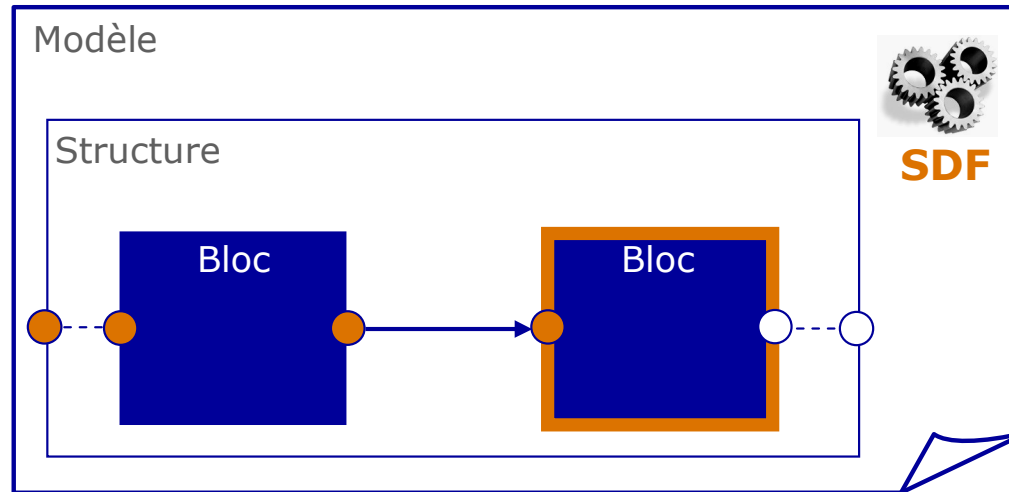
NON



ModHel'X : sémantique abstraite – snapshot

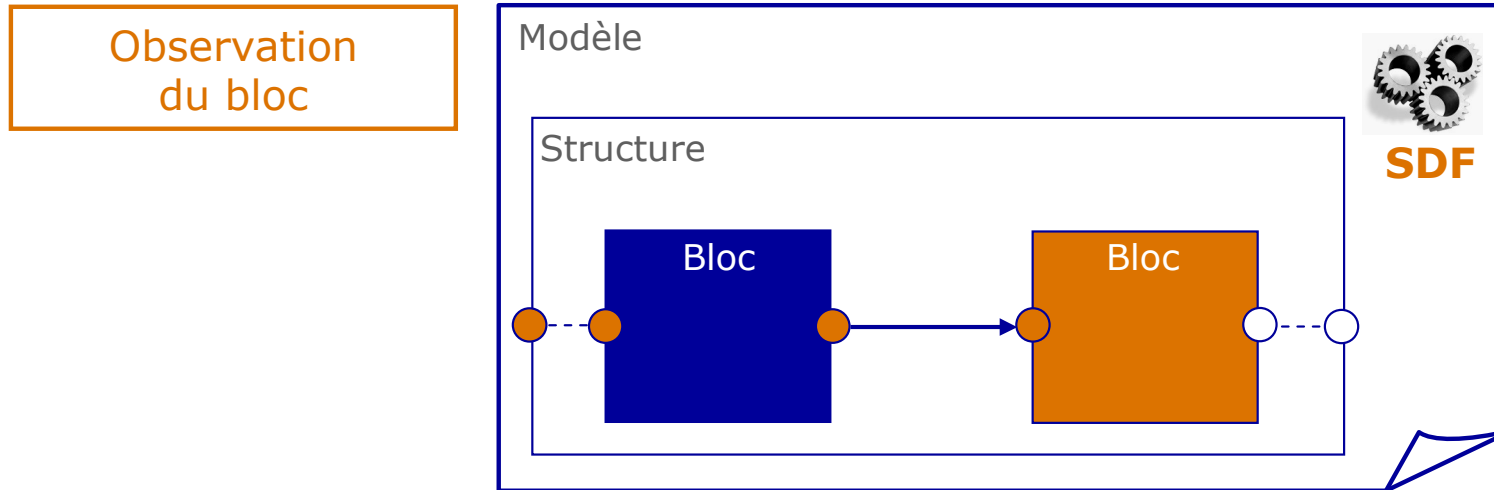
- Snapshot = combinaison d'observations des blocs du modèle

Choix
du bloc



ModHel'X : sémantique abstraite – snapshot

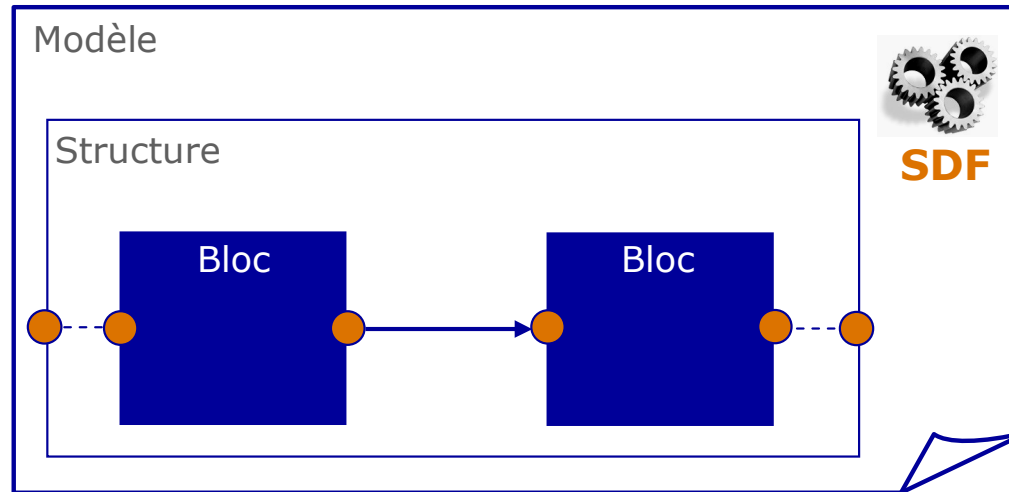
- Snapshot = combinaison d'observations des blocs du modèle



ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

Après observation
du bloc

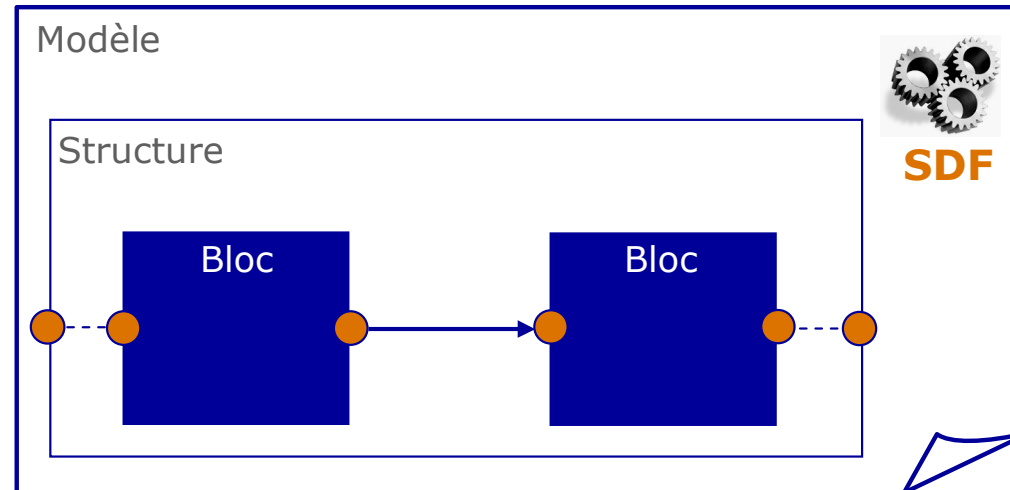


ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle

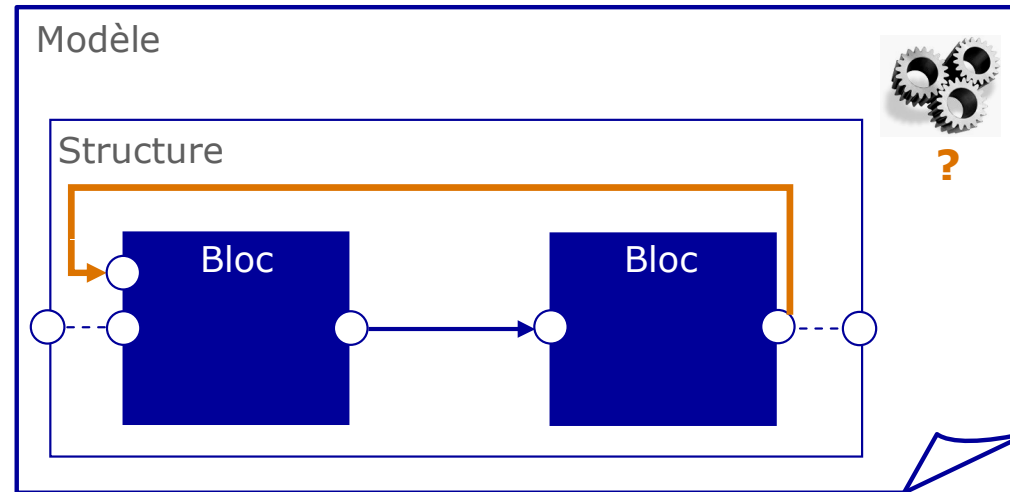
Snapshot
déterminé ?

OUI !



ModHel'X : sémantique abstraite – snapshot

- Snapshot = combinaison d'observations des blocs du modèle



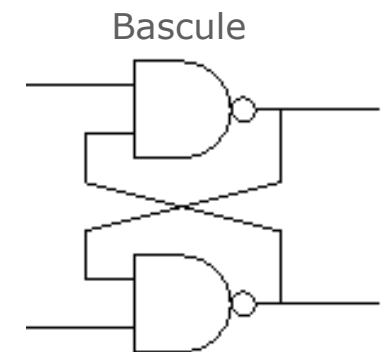
▶ Exemple :

- **Autres MoCs ?**

- ◆ *Boucles de rétroaction*
- ◆ *Propagation instantanée (MoC Synchronous Reactive)*
- ◆ ...

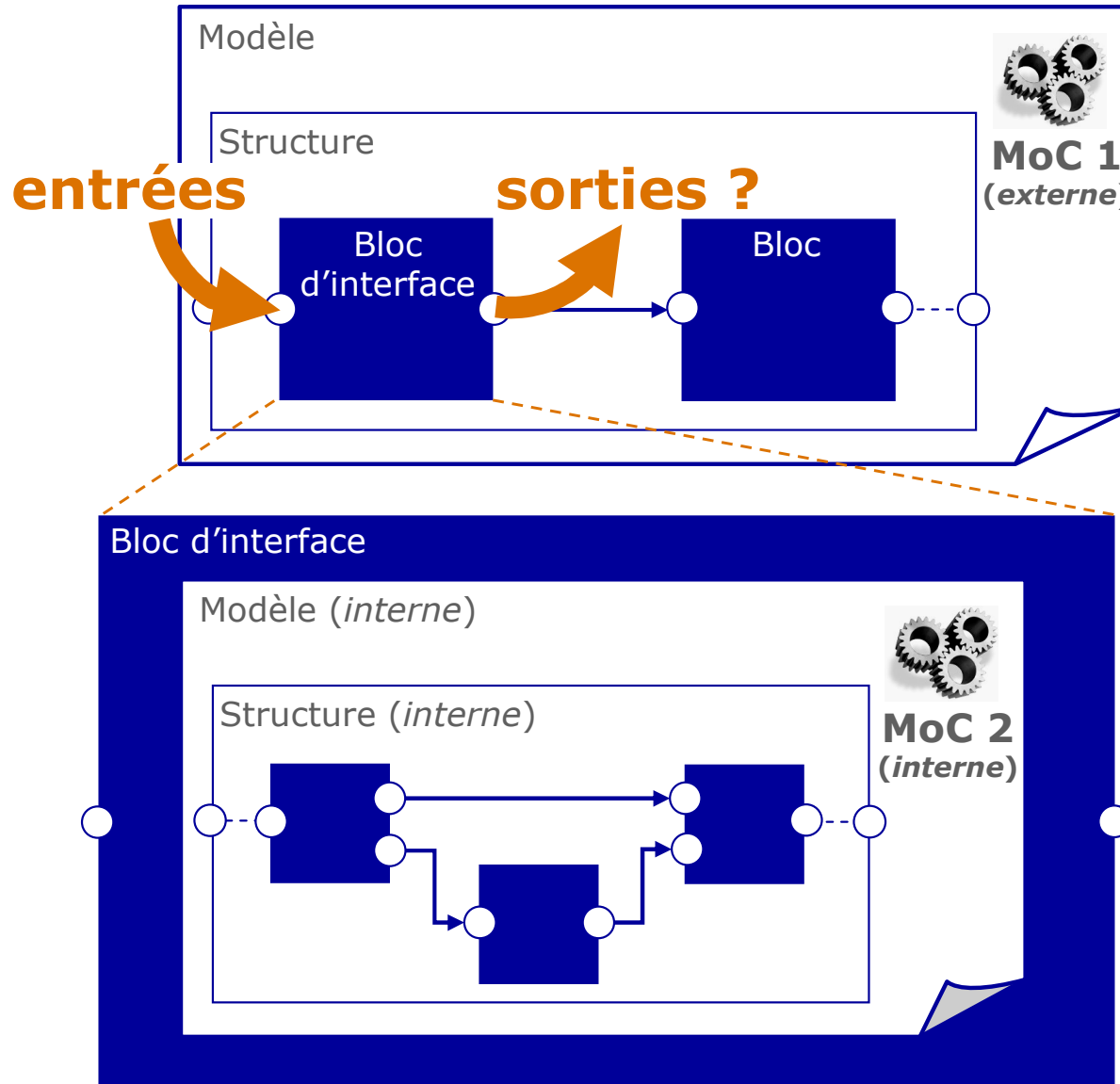
- **Autres règles d'exécution**

- ◆ *Calcul de point fixe*
- ◆ ...



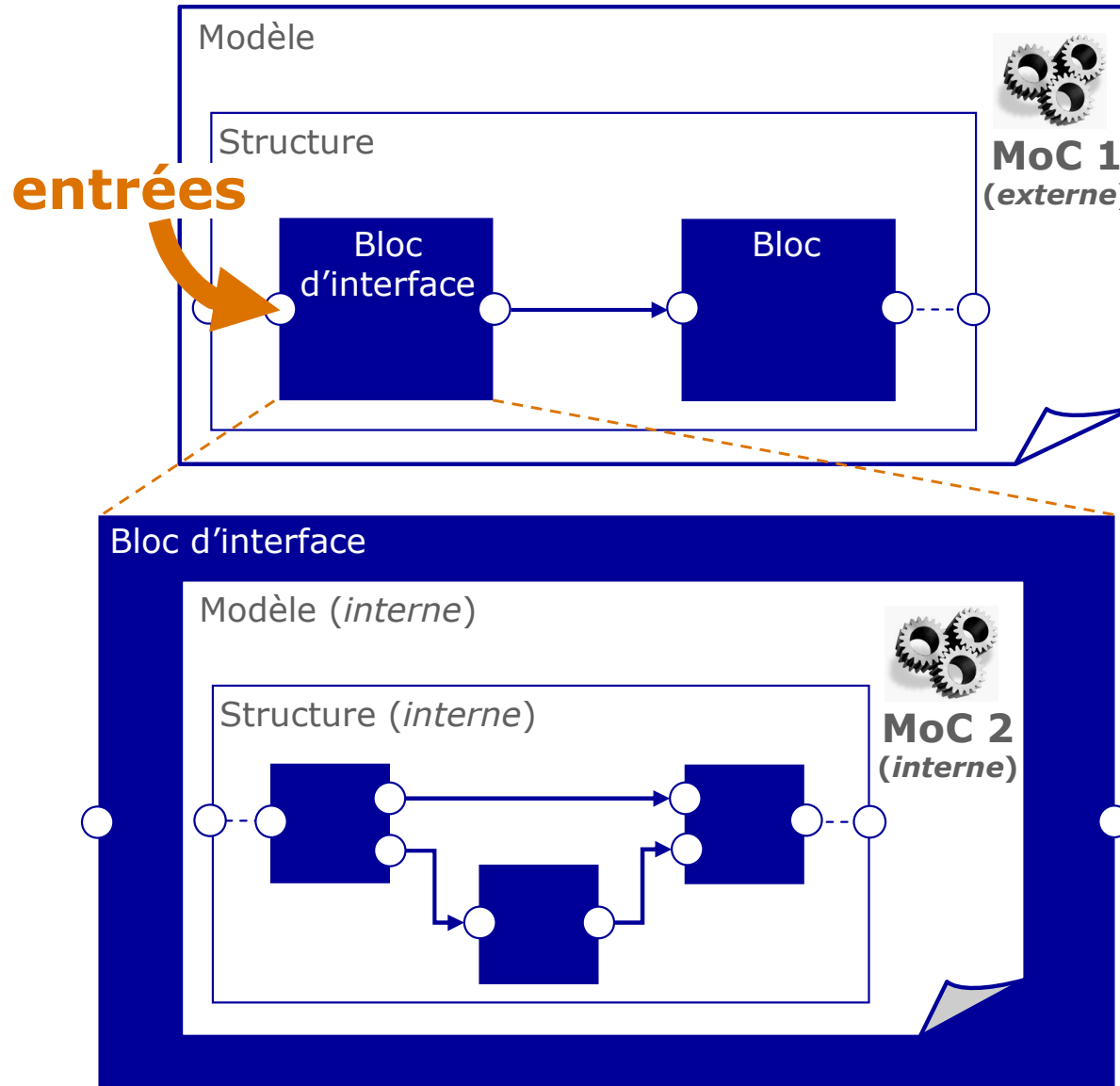
ModHel'X : sémantique abstraite – hiérarchie

- Observation d'un **bloc d'interface** ?



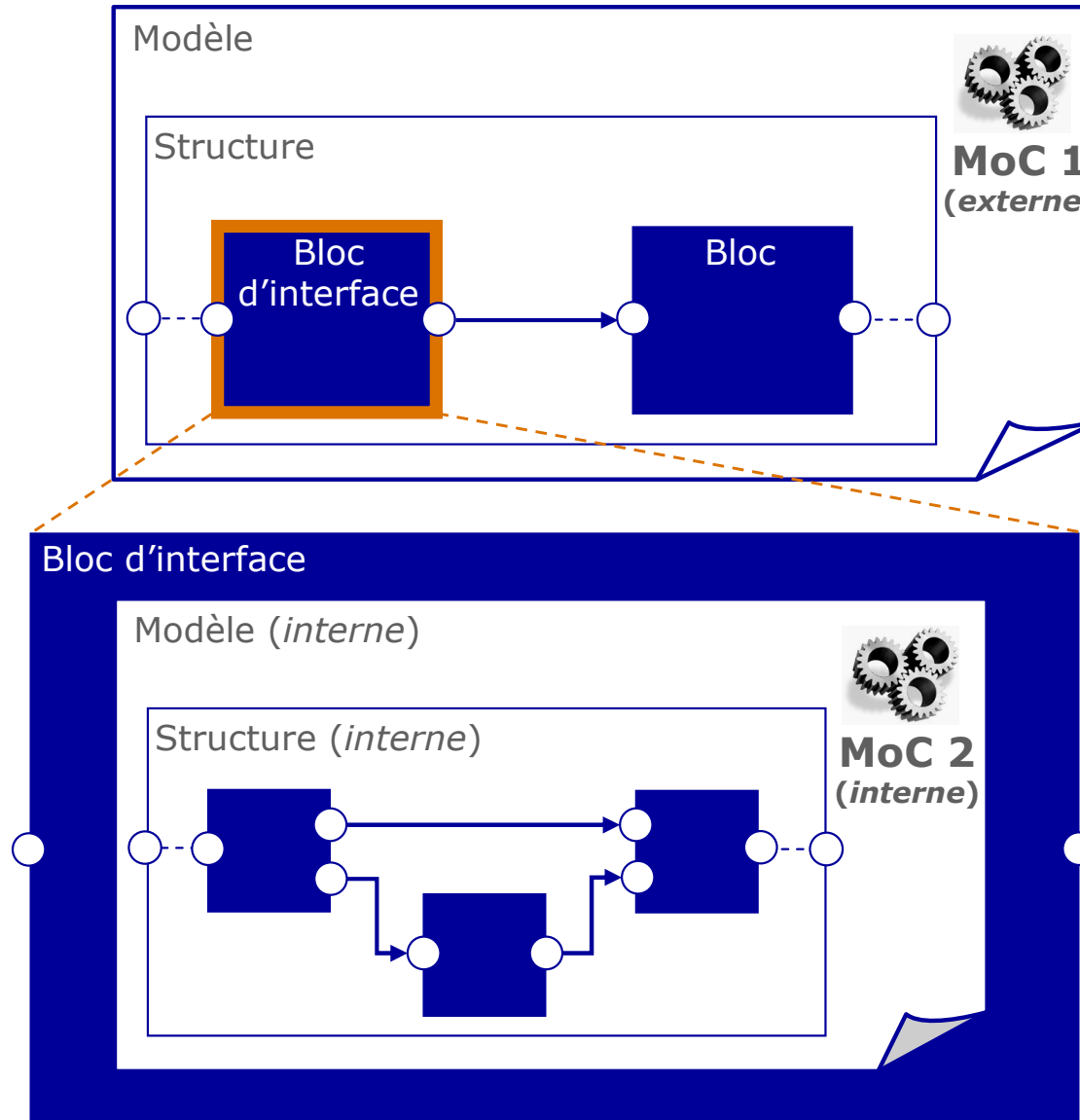
ModHel'X : sémantique abstraite – hiérarchie

- Observation d'un **bloc d'interface** =



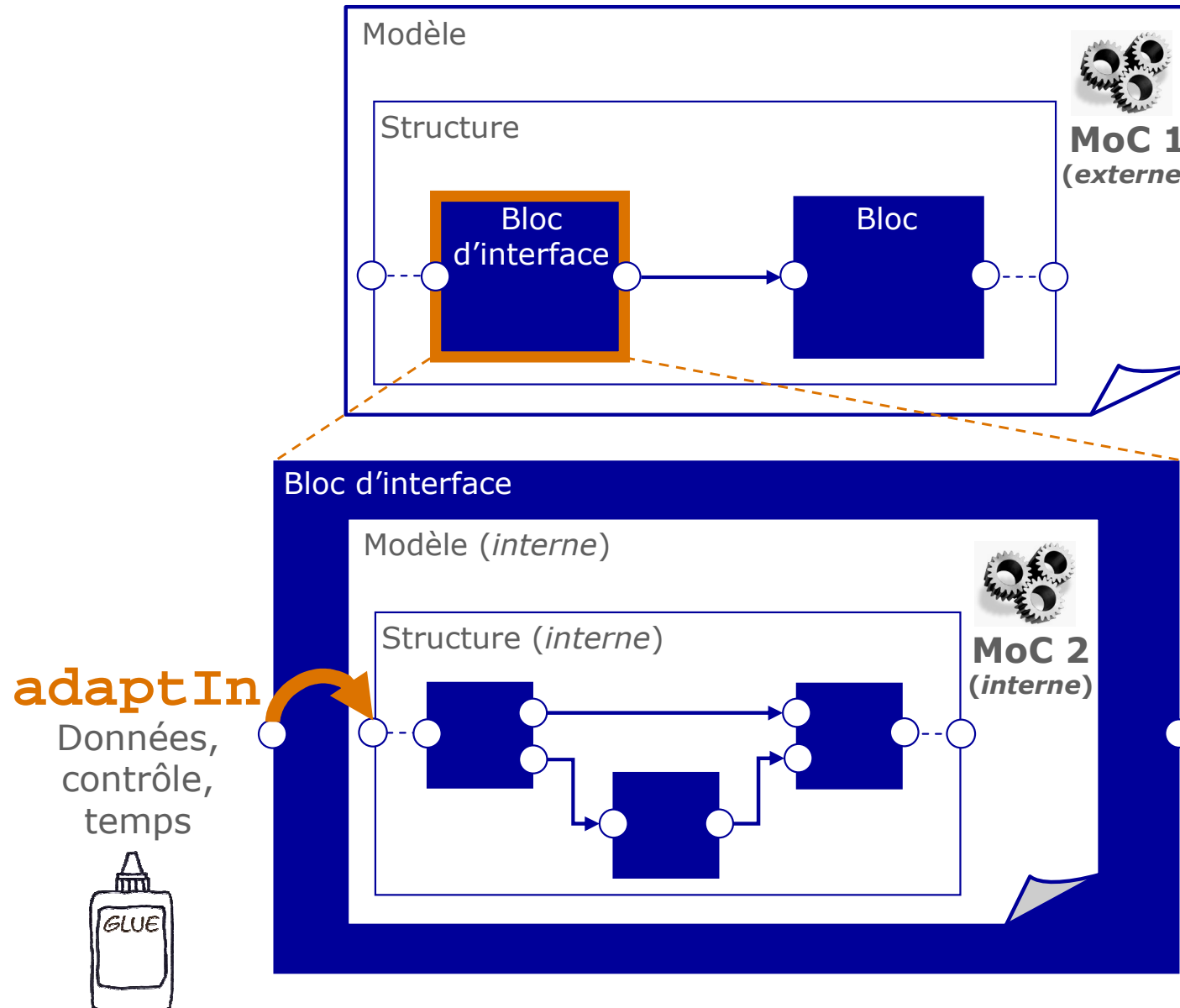
ModHel'X : sémantique abstraite – hiérarchie

- Observation d'un **bloc d'interface** =



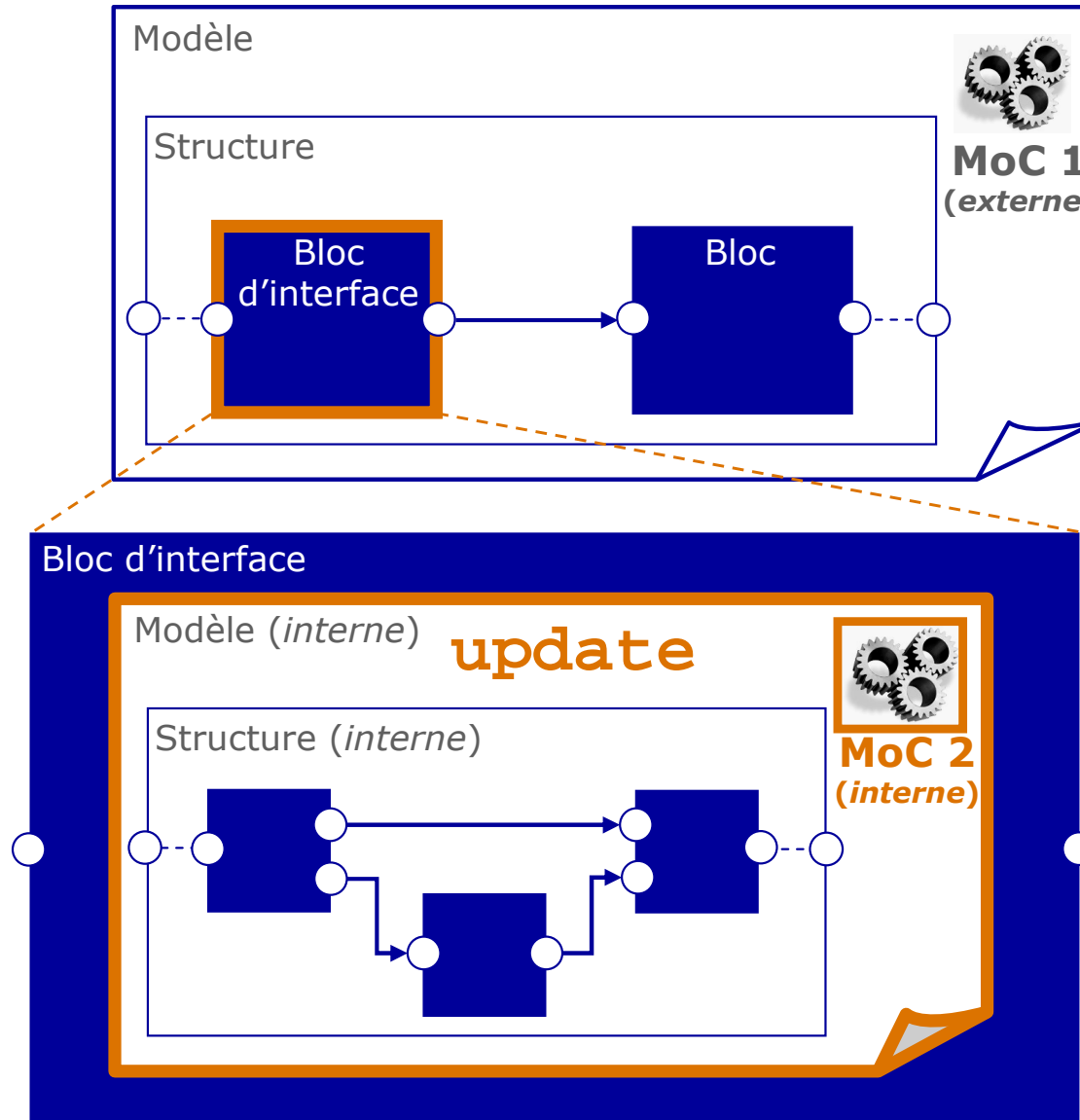
ModHel'X : sémantique abstraite – hiérarchie

- Observation d'un **bloc d'interface** =



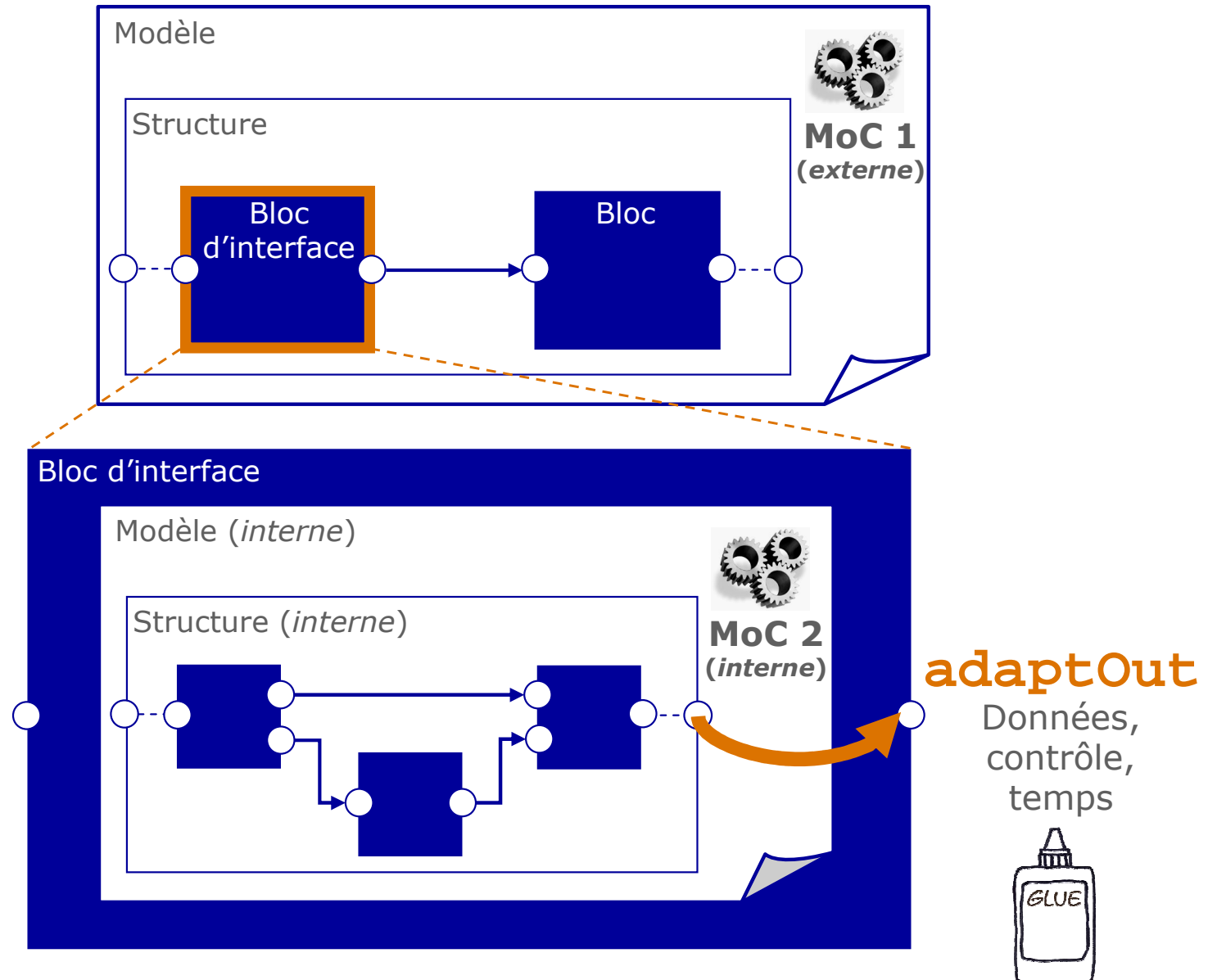
ModHel'X : sémantique abstraite – hiérarchie

- Observation d'un **bloc d'interface** =



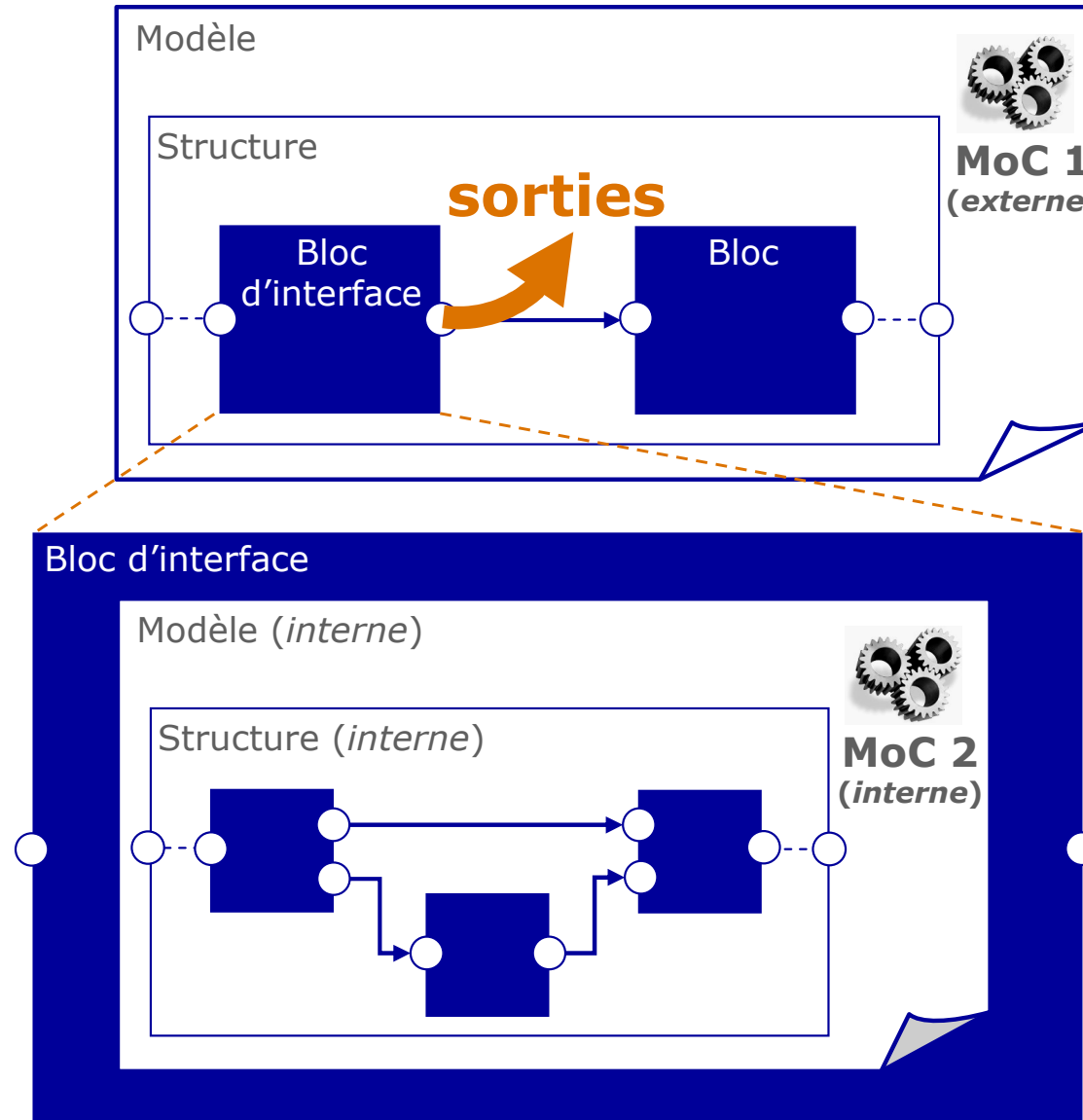
ModHel'X : sémantique abstraite – hiérarchie

- Observation d'un **bloc d'interface** =

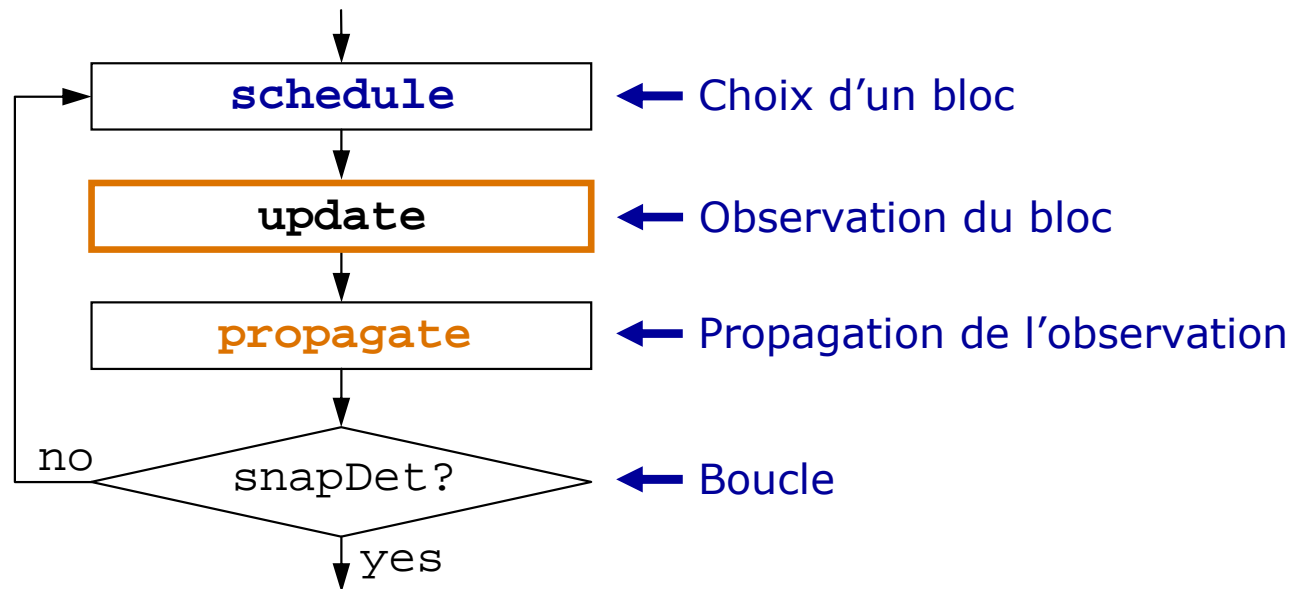


ModHel'X : sémantique abstraite – hiérarchie

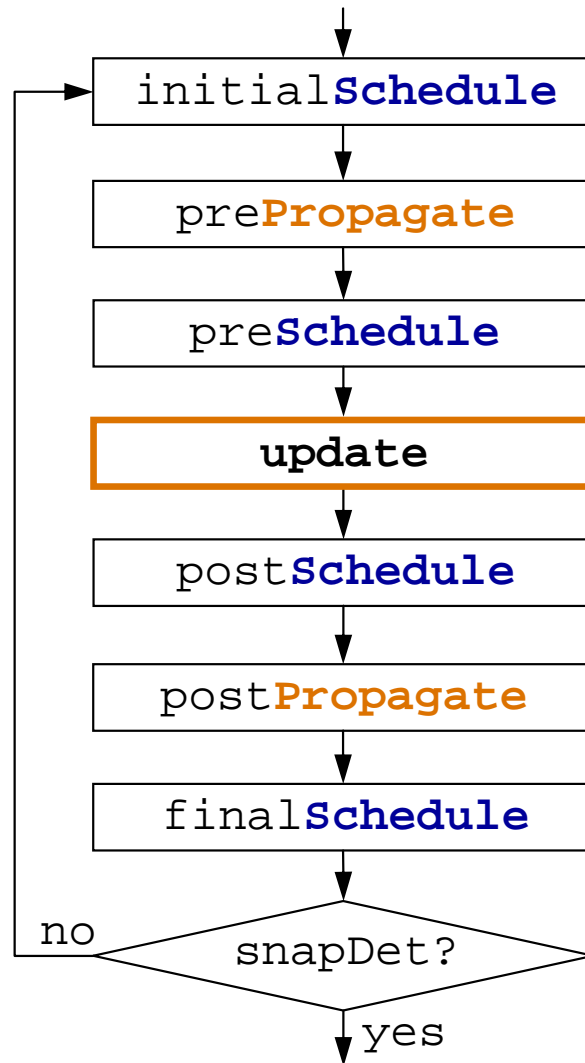
- Observation d'un **bloc d'interface** =



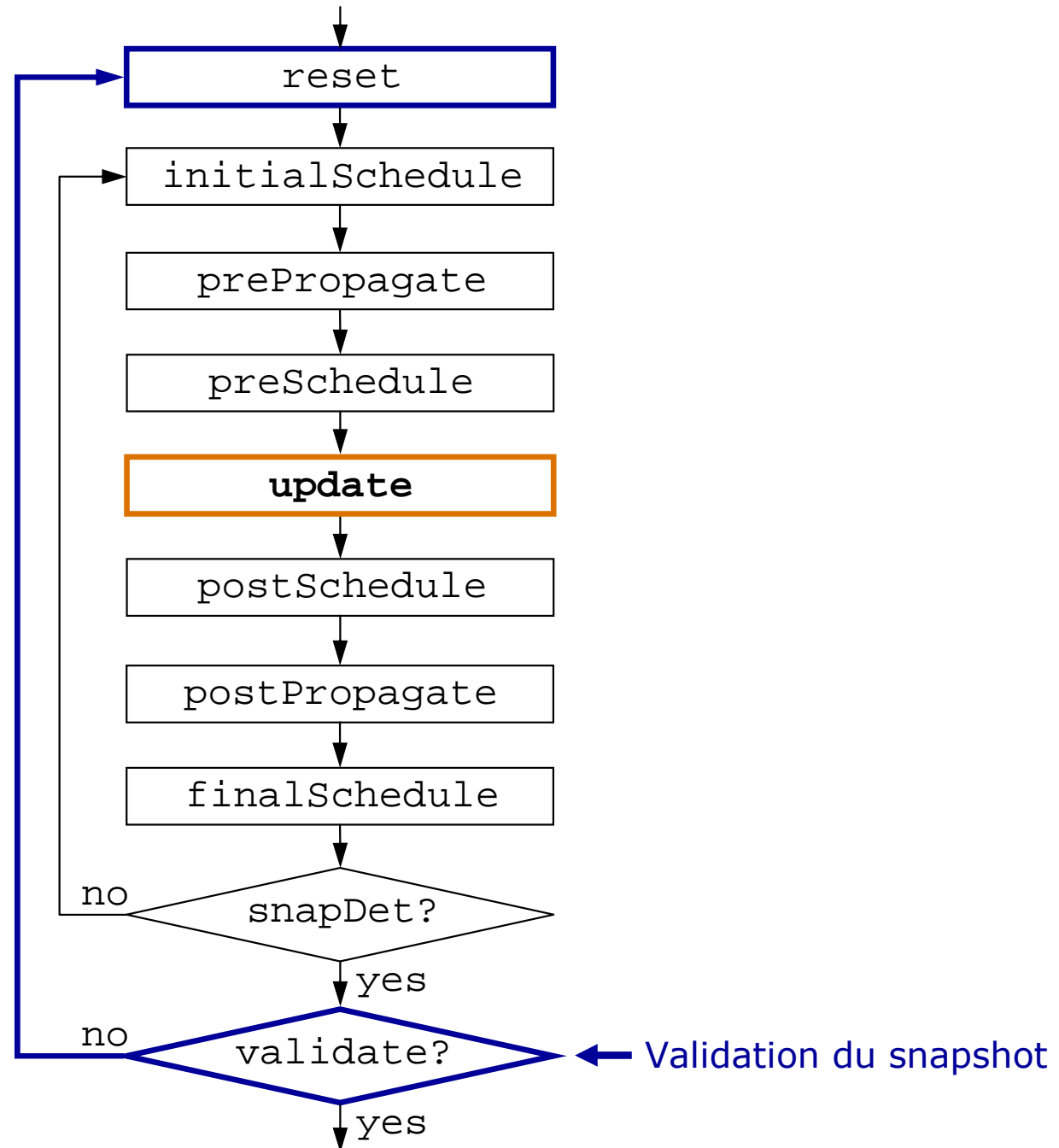
ModHel'X : sémantique abstraite – algorithme



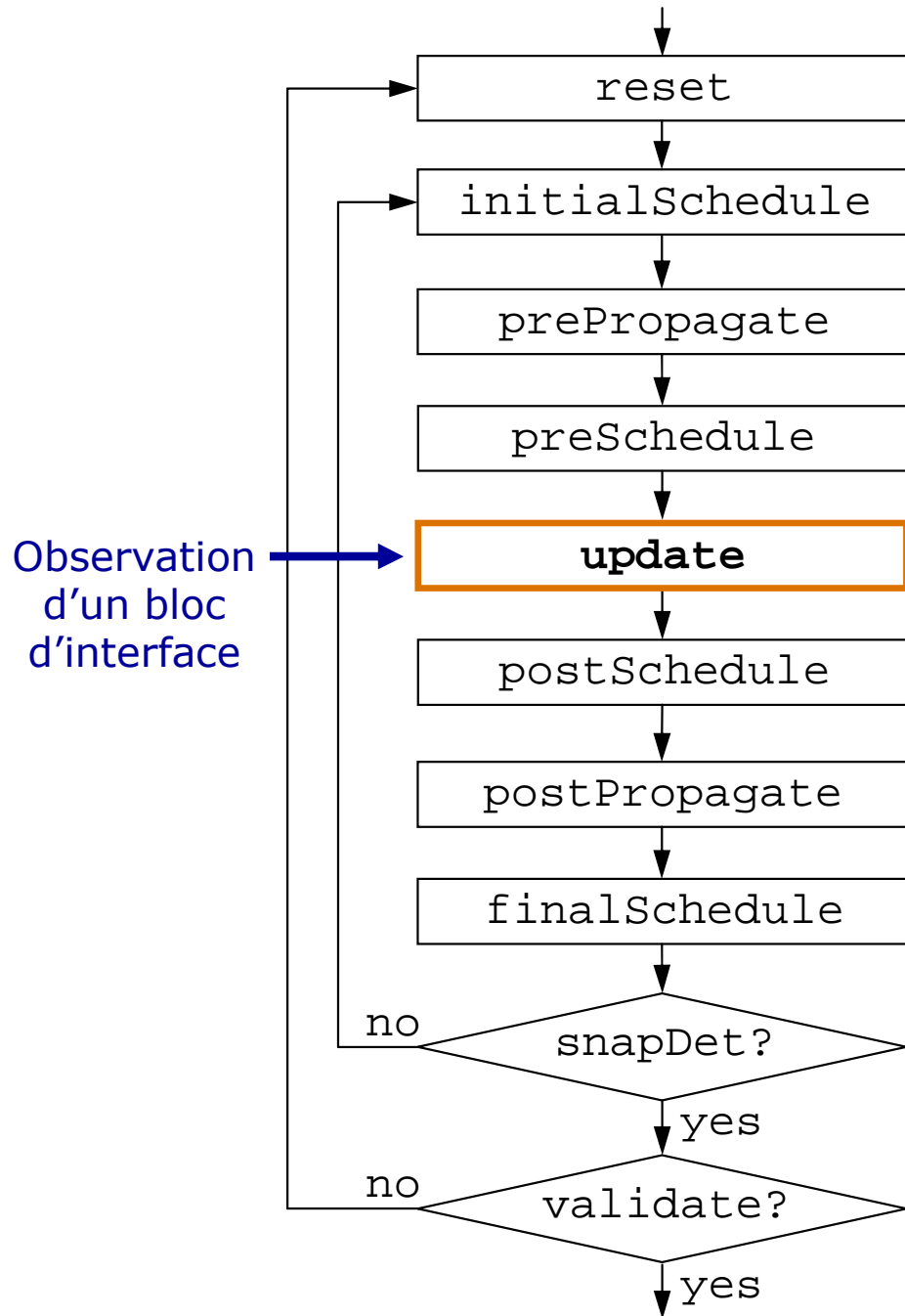
ModHel'X : sémantique abstraite – algorithmme



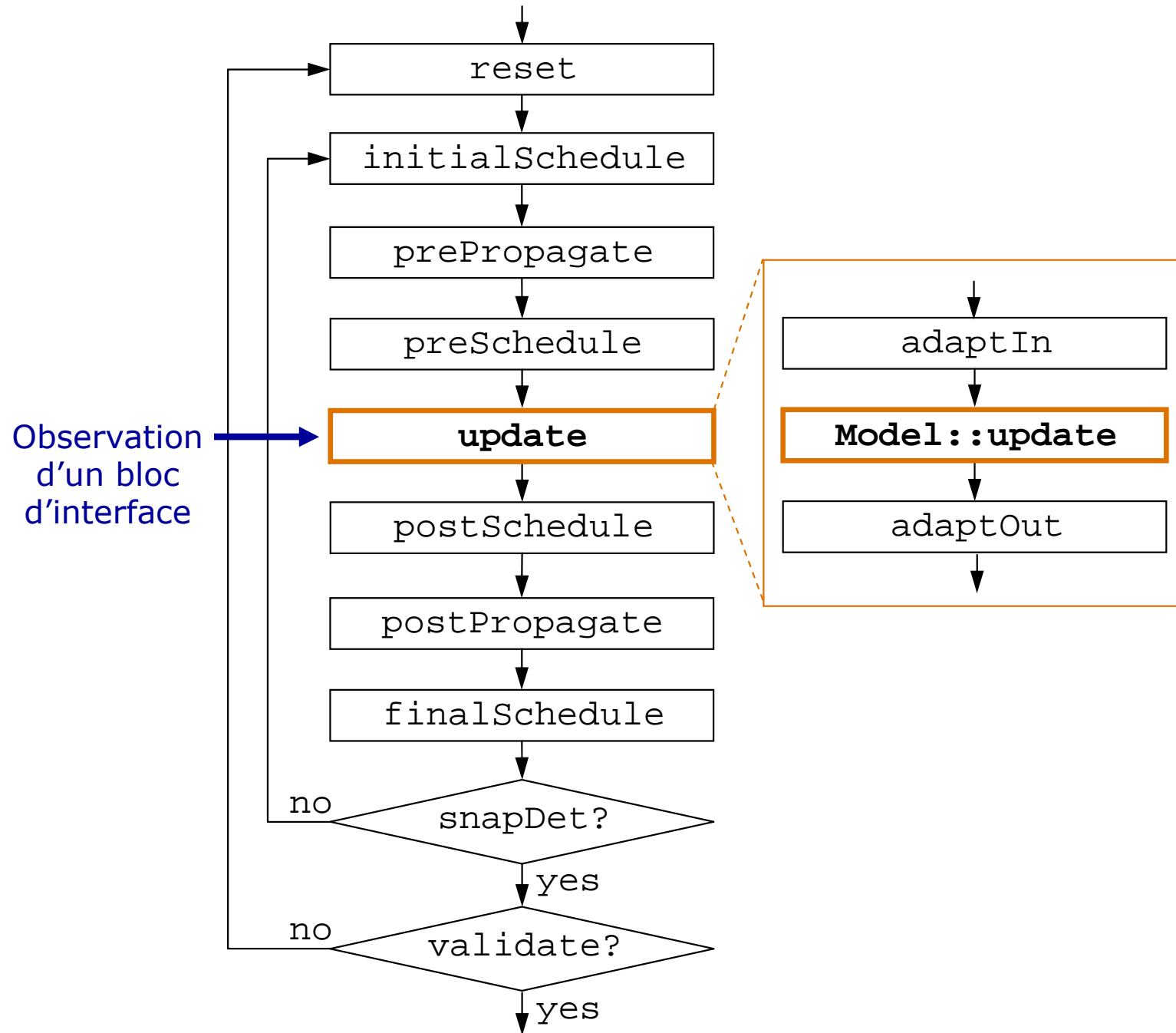
ModHel'X : sémantique abstraite – algorithmme



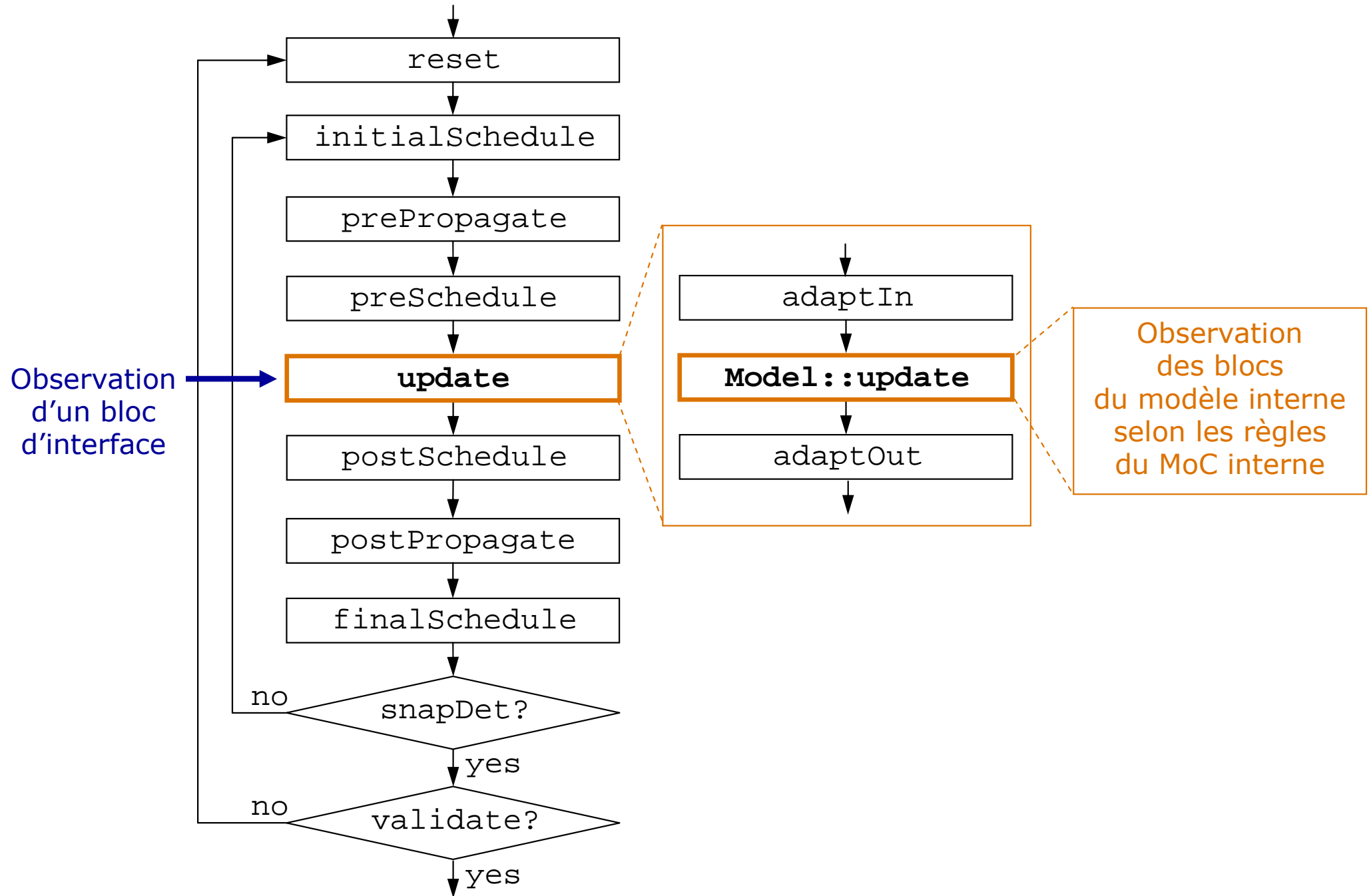
ModHel'X : sémantique abstraite – hiérarchie



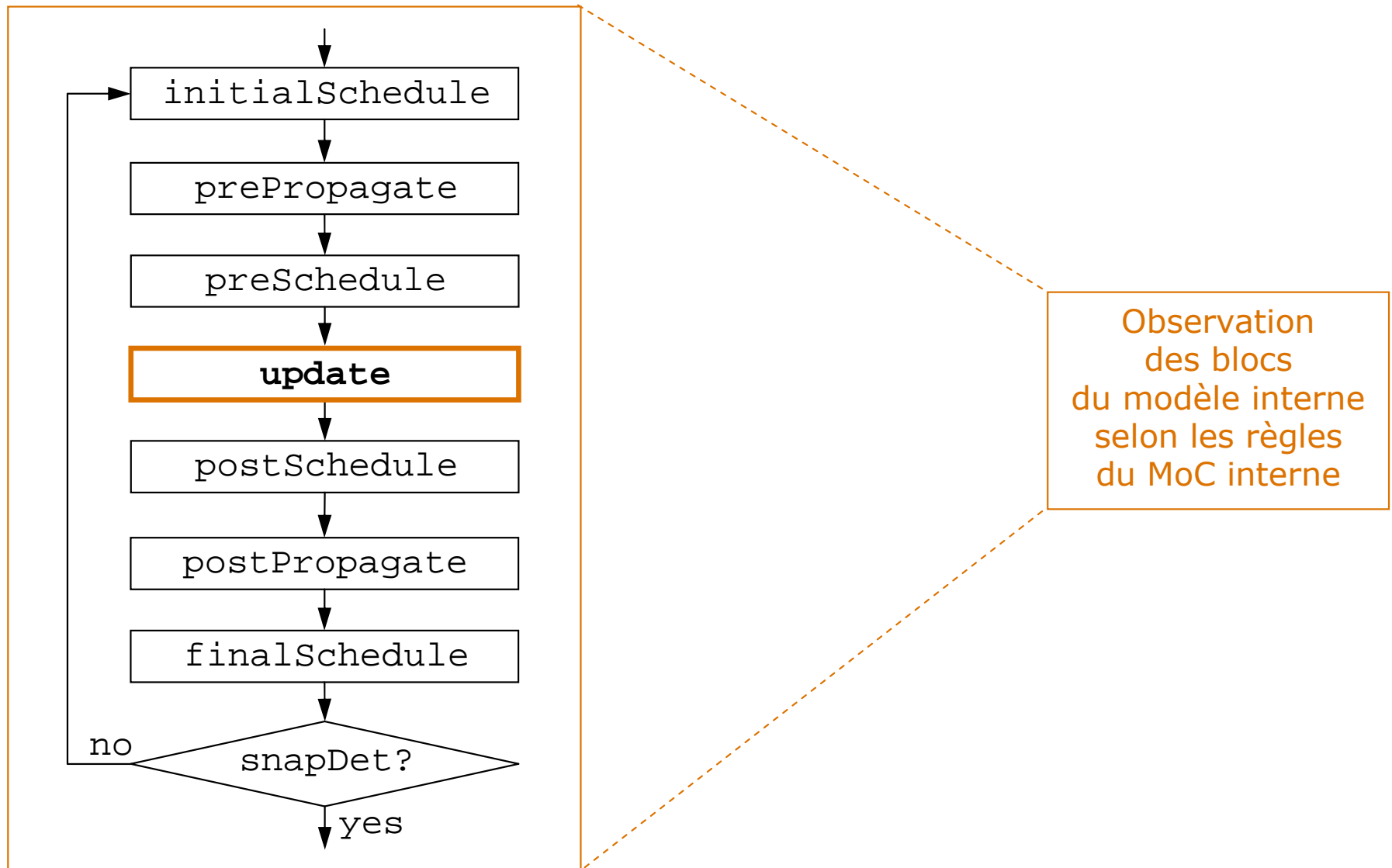
ModHel'X : sémantique abstraite – hiérarchie



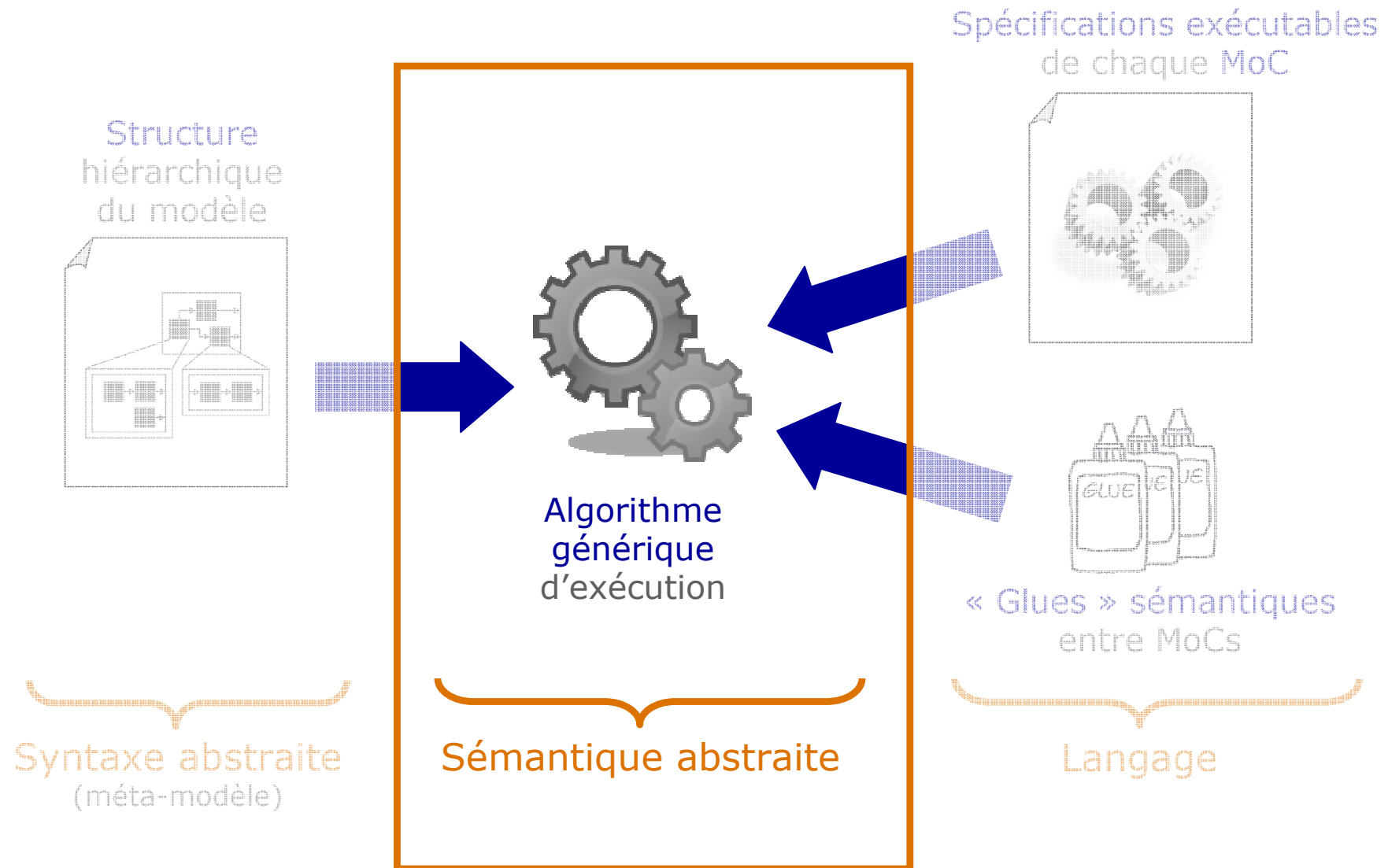
ModHel'X : sémantique abstraite – hiérarchie



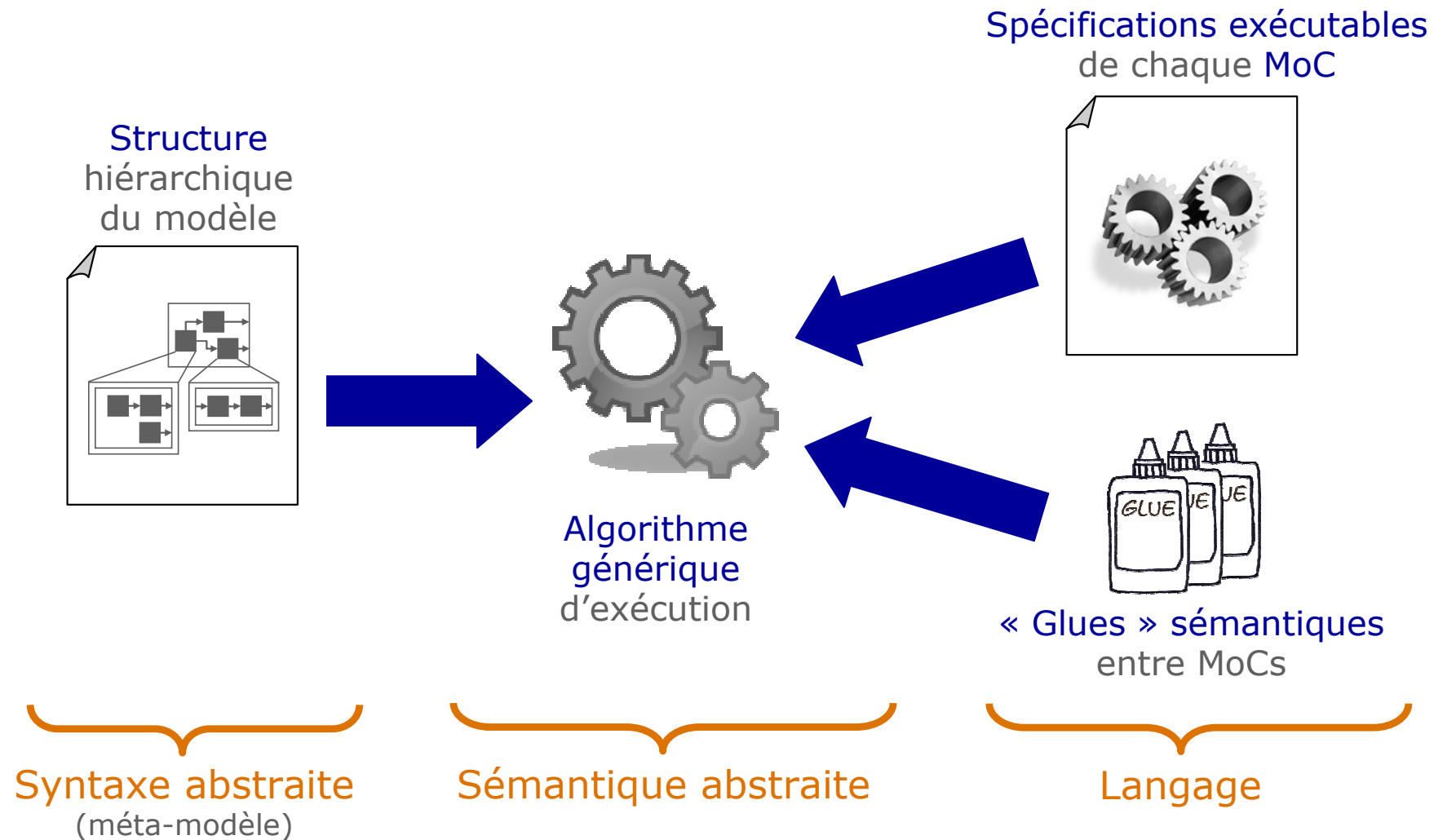
ModHel'X : sémantique abstraite – hiérarchie



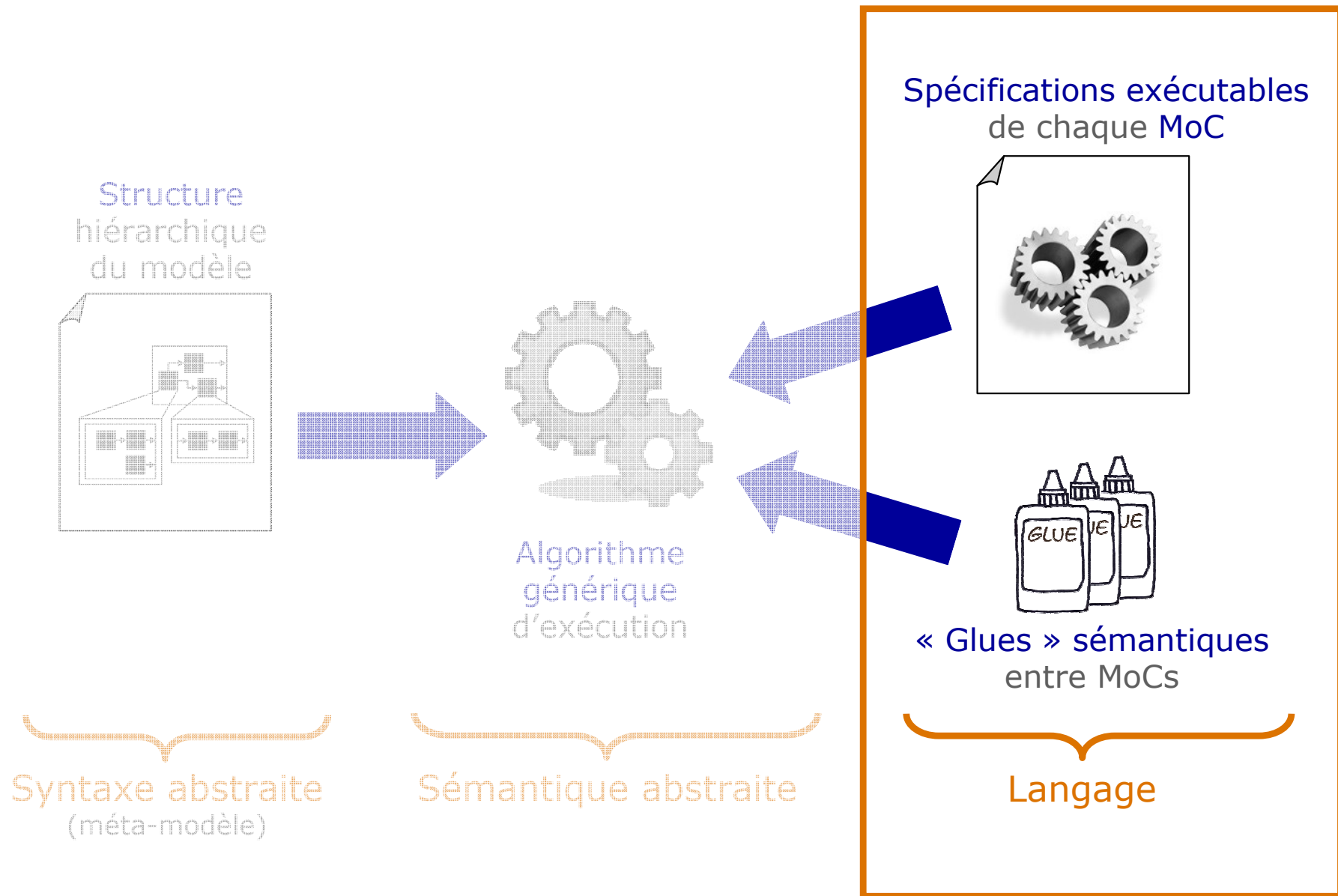
ModHel'X : architecture générale



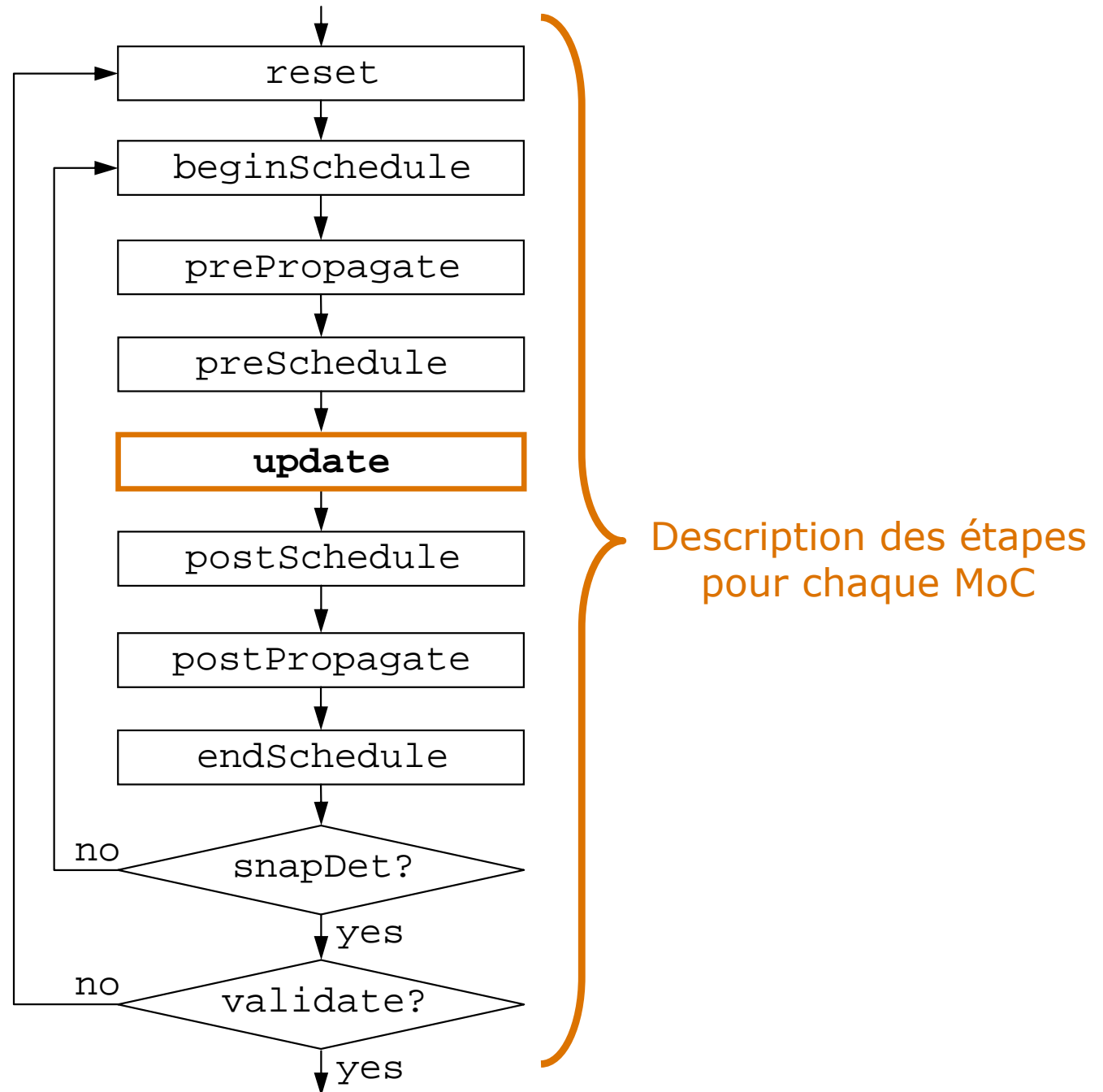
ModHel'X : architecture générale



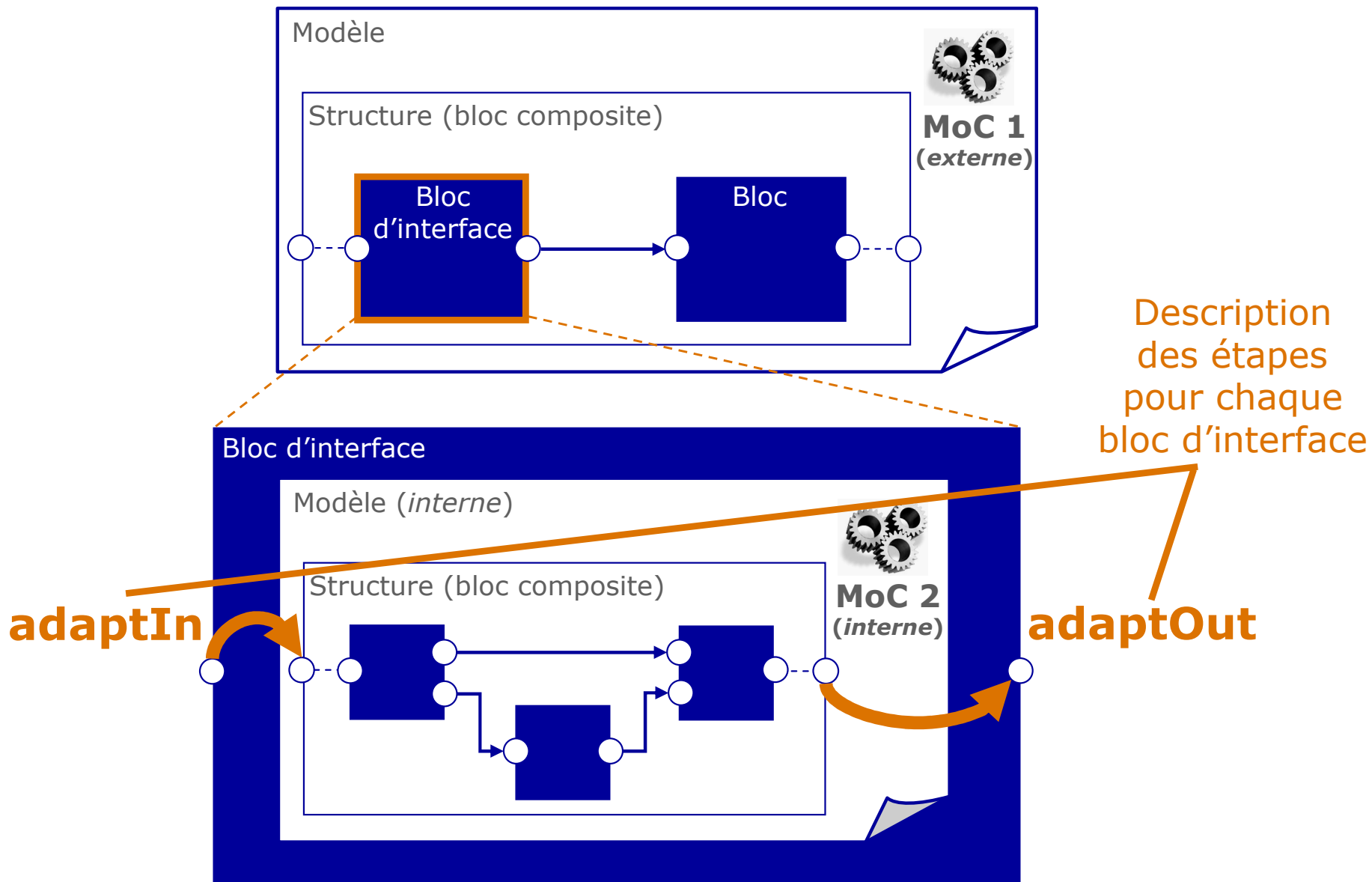
ModHel'X : architecture générale



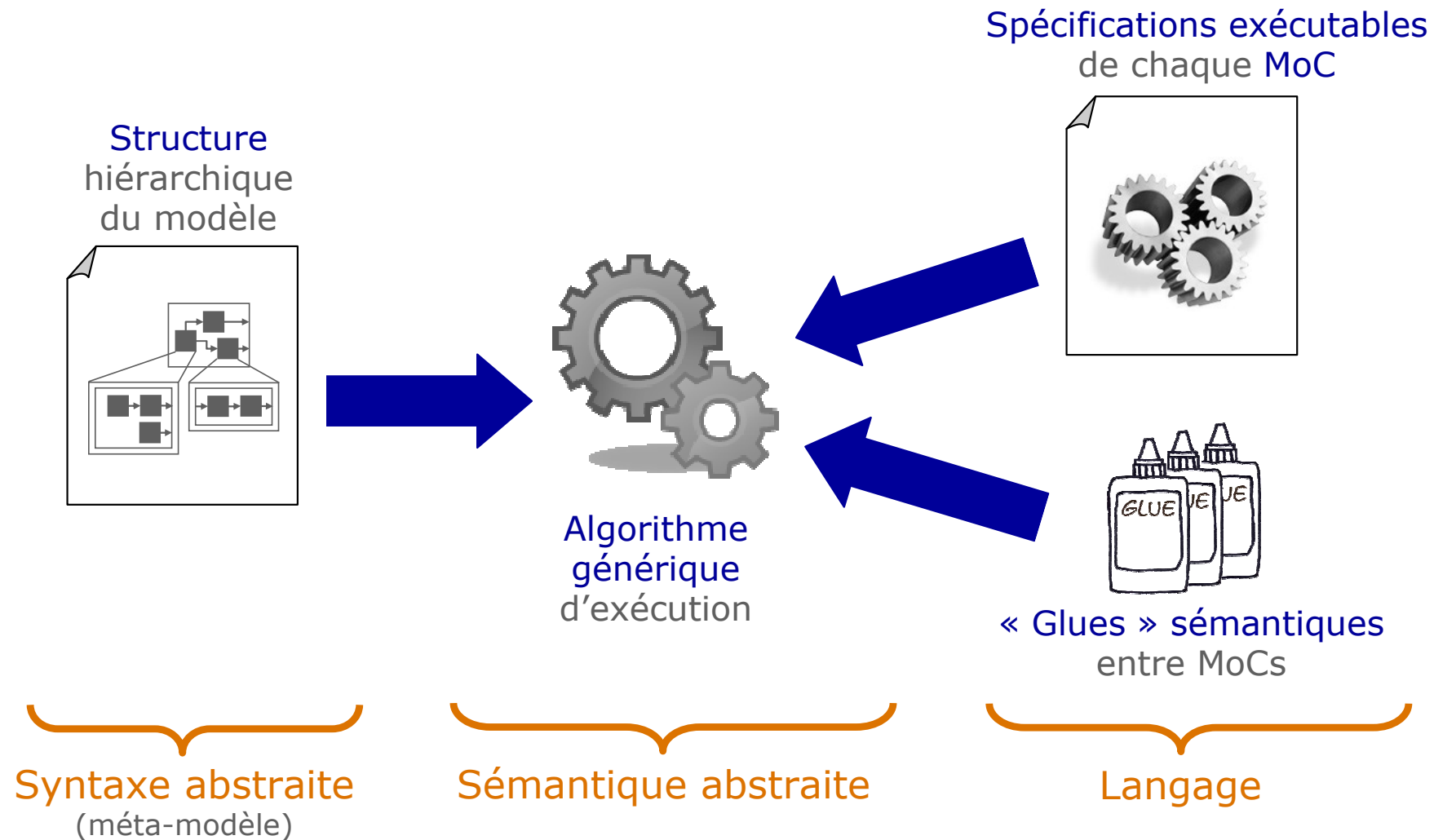
ModHel'X : langage



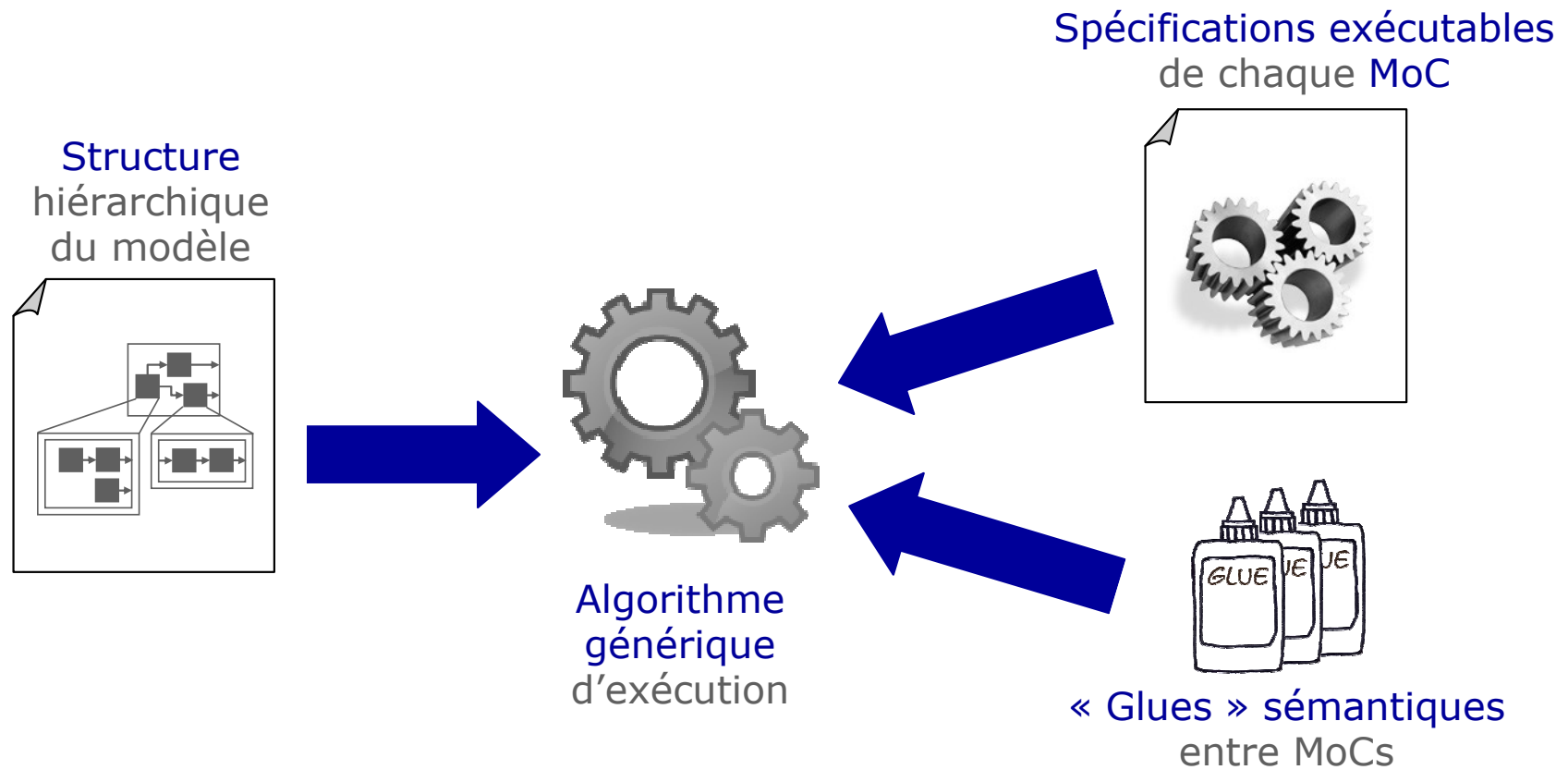
ModHel'X : langage



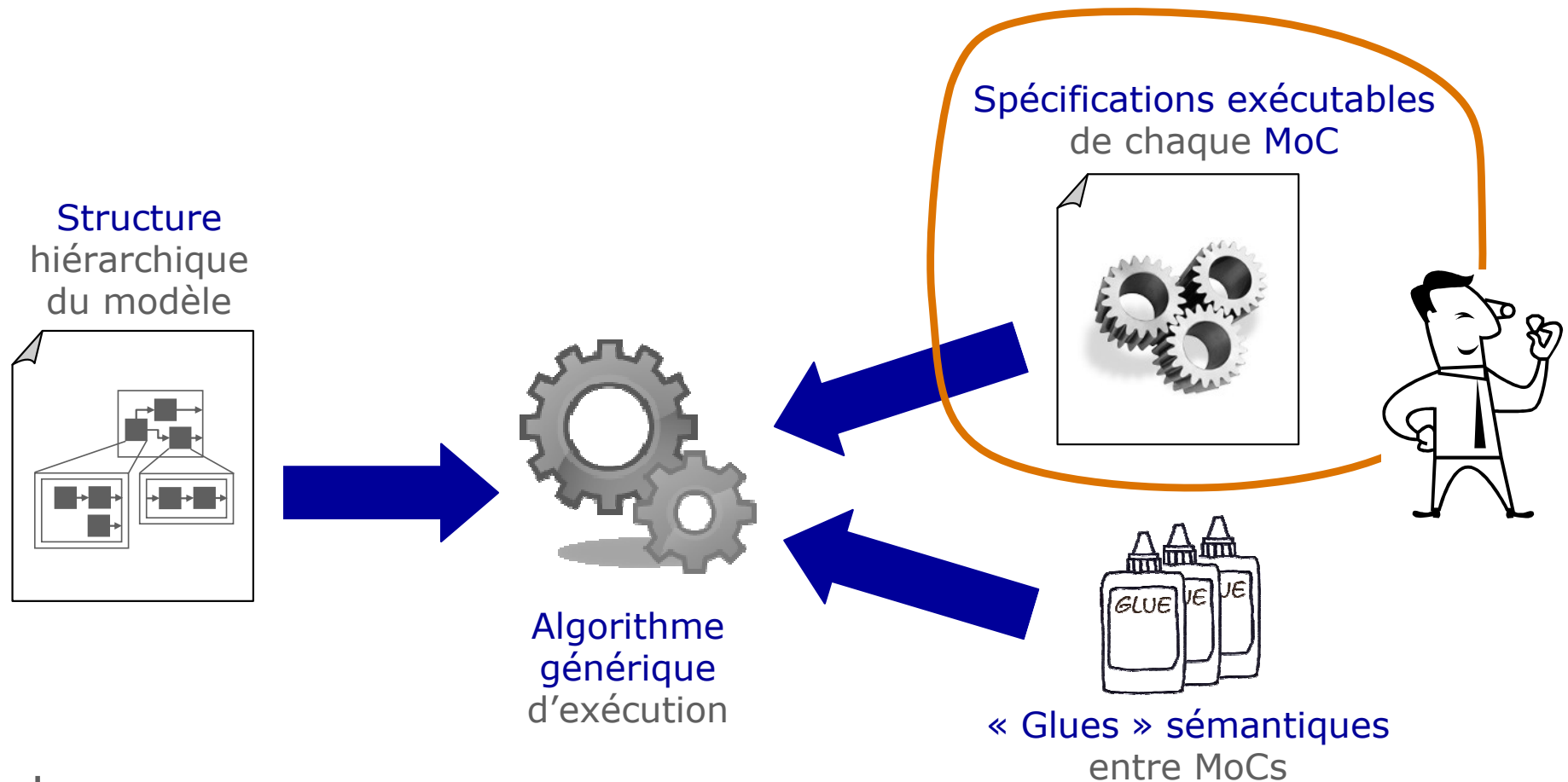
ModHel'X : architecture générale



ModHel'X : workflow



ModHel'X : workflow

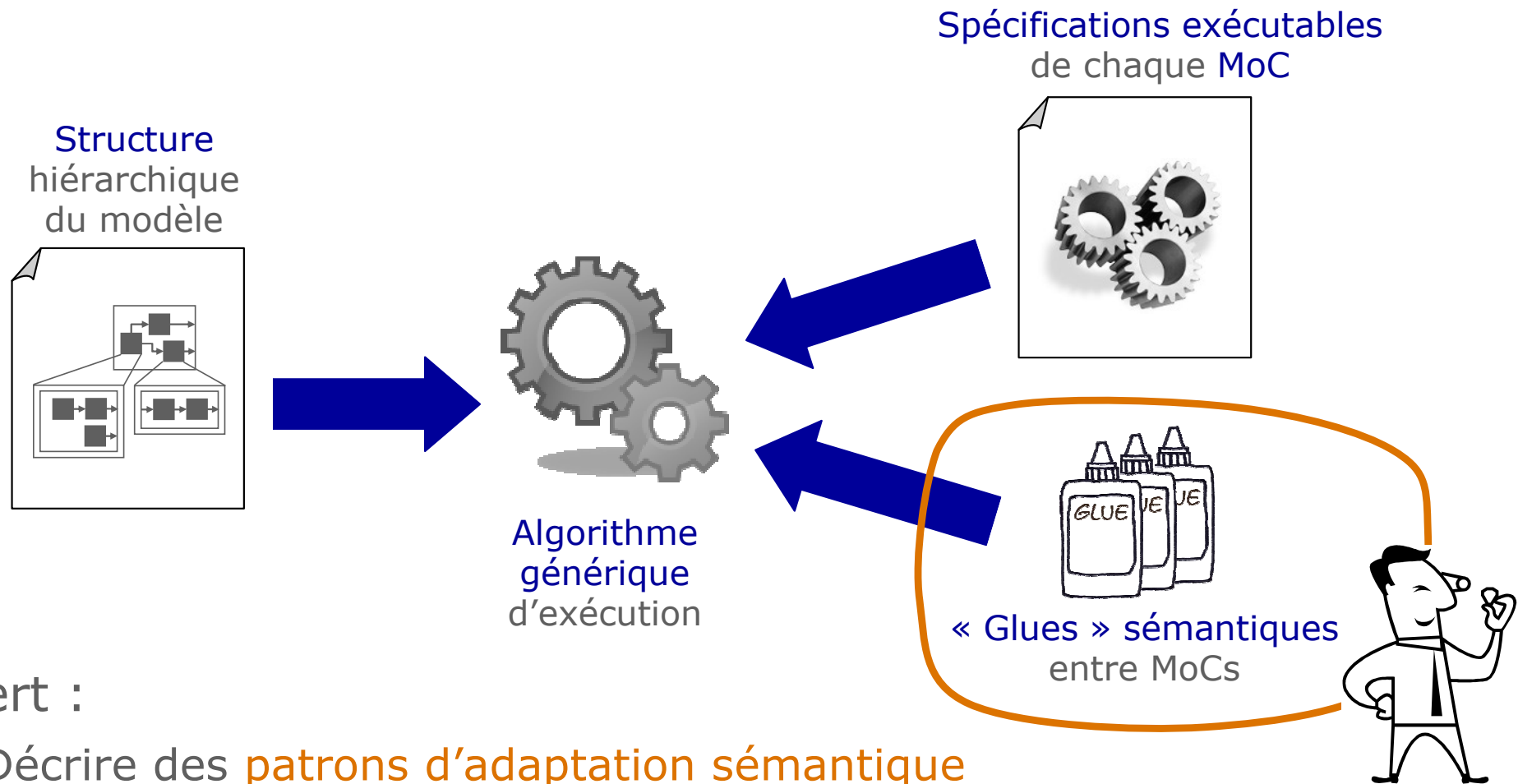


■ Expert :

▶ Décrire des MoCs

- Spécialiser les éléments de la syntaxe abstraite
- Décrire les étapes de la sémantique abstraite

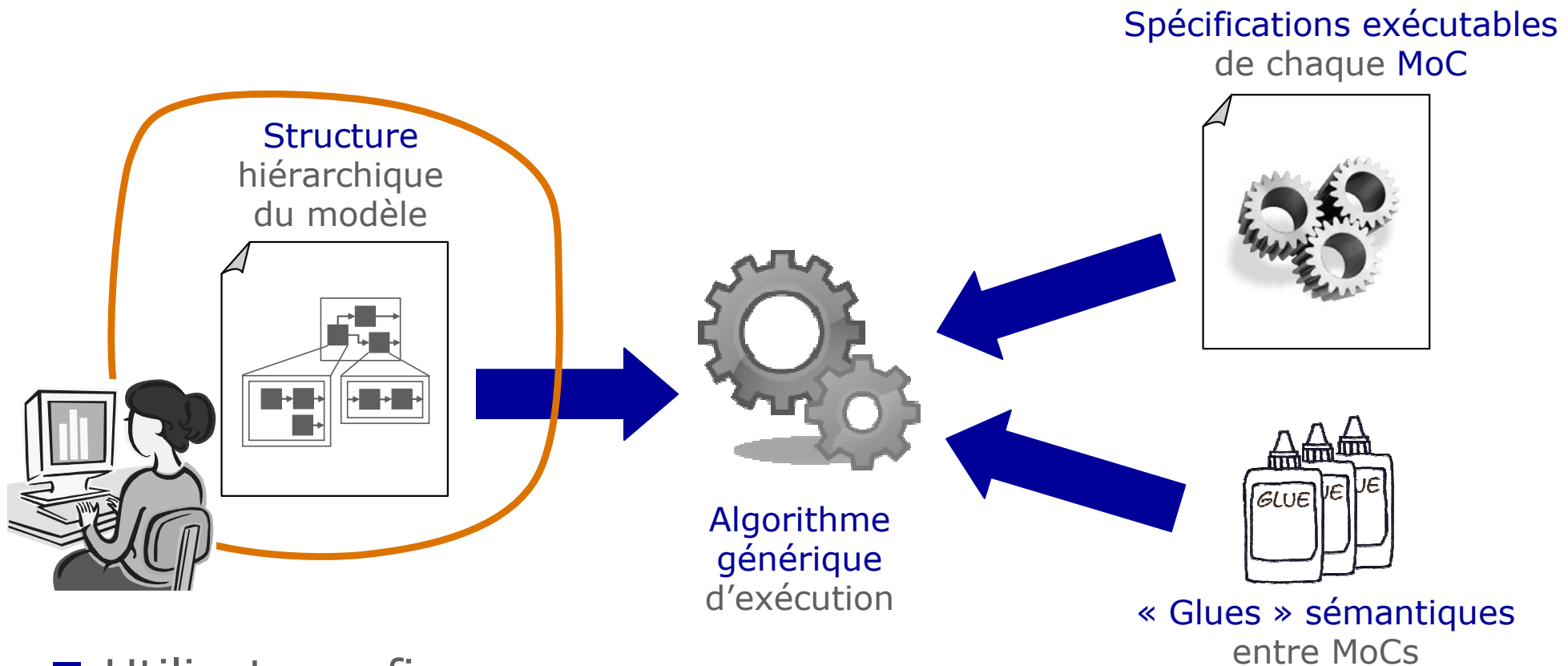
ModHel'X : workflow



■ Expert :

- ▶ Décrire des patrons d'adaptation sémantique
 - Créer un bloc d'interface
 - Décrire les étapes d'adaptation sémantique du bloc d'interface

ModHel'X : workflow



■ Utilisateurs finaux :

▶ Décrire, importer, exécuter des modèles

- Modèles hétérogènes
 - ◆ *Choix d'un patron d'adaptation sémantique*
 - ◆ *Personnalisation du patron*

ModHel'X : positionnement

- Positionnement par rapport à **PtolemyII**

ModHel'X : positionnement

■ Positionnement par rapport à PtolemyII

▶ Adaptation sémantique explicite entre MoCs

- Possibilité pour l'utilisateur de :

- ◆ Choisir

- ◆ Paramétrer

- ◆ Modifier

} *l'adaptation sémantique réalisée entre deux MoCs
(bloc d'interface)*

■ Positionnement par rapport à PtolemyII

▶ Adaptation sémantique explicite entre MoCs

- Possibilité pour l'utilisateur de :

- ◆ Choisir
- ◆ Paramétrer
- ◆ Modifier

} *l'adaptation sémantique réalisée entre deux MoCs
(bloc d'interface)*

▶ Modélisation des MoCs

- Décomposition en étapes pour la sémantique abstraite
- Généralisation de l'algorithme générique d'exécution
 - ◆ Calcul de point fixe
 - ◆ Validation d'un snapshot
 - ◆ Pilotage de l'exécution par n'importe quel niveau de la hiérarchie

ModHel'X : positionnement

■ Positionnement par rapport à PtolemyII

▶ Adaptation sémantique explicite entre MoCs

- Possibilité pour l'utilisateur de :

- ◆ Choisir
 - ◆ Paramétrer
 - ◆ Modifier
- } *l'adaptation sémantique réalisée entre deux MoCs
(bloc d'interface)*

▶ Modélisation des MoCs

- Décomposition en étapes pour la sémantique abstraite
- Généralisation de l'algorithme générique d'exécution
 - ◆ Calcul de point fixe
 - ◆ Validation d'un snapshot
 - ◆ Pilotage de l'exécution par n'importe quel niveau de la hiérarchie

■ Couverture des 4 facettes du problème de l'hétérogénéité

ModHel'X : positionnement

■ Positionnement par rapport à PtolemyII

▶ Adaptation sémantique explicite entre MoCs

- Possibilité pour l'utilisateur de :

- ◆ Choisir
 - ◆ Paramétrer
 - ◆ Modifier
- } *l'adaptation sémantique réalisée entre deux MoCs
(bloc d'interface)*

▶ Modélisation des MoCs

- Décomposition en étapes pour la sémantique abstraite
- Généralisation de l'algorithme générique d'exécution
 - ◆ Calcul de point fixe
 - ◆ Validation d'un snapshot
 - ◆ Pilotage de l'exécution par n'importe quel niveau de la hiérarchie

■ Couverture des 4 facettes du problème de l'hétérogénéité

Couverture **actuelle** :

Dom	●
Niv	●
Vue	●
Act	●

ModHel'X : positionnement

■ Positionnement par rapport à PtolemyII

▶ Adaptation sémantique explicite entre MoCs

- Possibilité pour l'utilisateur de :

- ◆ Choisir
- ◆ Paramétrer
- ◆ Modifier

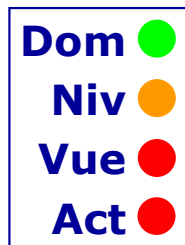
} *l'adaptation sémantique réalisée entre deux MoCs
(bloc d'interface)*

▶ Modélisation des MoCs

- Décomposition en étapes pour la sémantique abstraite
- Généralisation de l'algorithme générique d'exécution
 - ◆ Calcul de point fixe
 - ◆ Validation d'un snapshot
 - ◆ Pilotage de l'exécution par n'importe quel niveau de la hiérarchie

■ Couverture des 4 facettes du problème de l'hétérogénéité

Couverture **actuelle** :



Couverture **cible** :



Introduction

Hétérogénéité des modèles

Modélisation multi-paradigme

ModHel'X

▶ **Framework expérimental**

Conclusion & perspectives

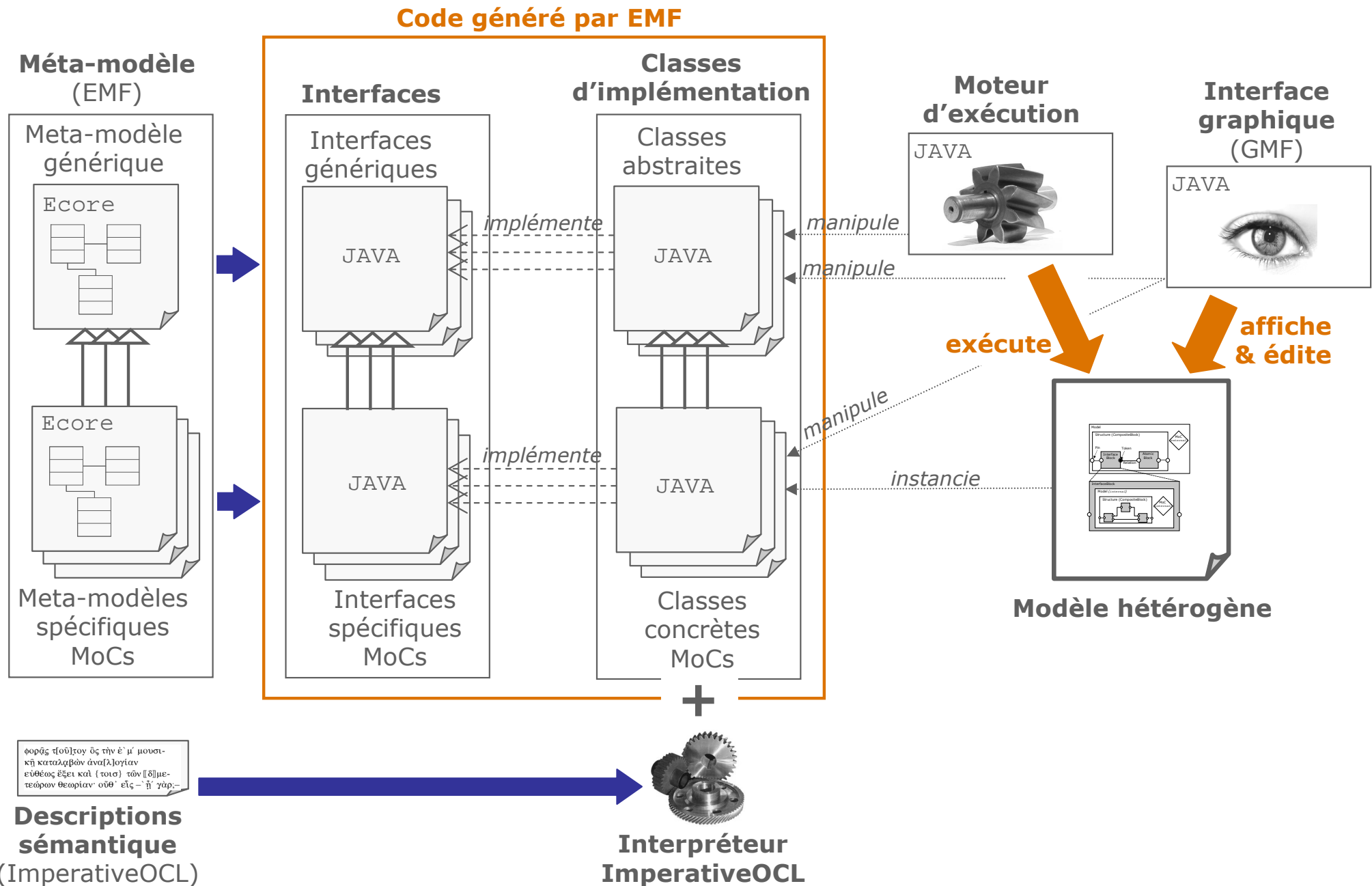
Framework expérimental : architecture cible

- **Syntaxe abstraite** générique (méta-modèle)
 - ▶ Utilisation du **Eclipse Modeling Framework (EMF)**
 - Outillé (génération de code, interface graphique)
 - Facilite l'interopérabilité avec d'autres approches IDM (transformation de modèles, etc.)

- **Sémantique abstraite** générique (algorithme)
 - ▶ **Génération** partielle du code par EMF

- **Langage** pour la description de la sémantique
 - ▶ Utilisation d'un langage **existant**
 - Si possible standard (OMG)
 - Outillé (interpréteur ou générateur de code)

Framework expérimental : architecture cible



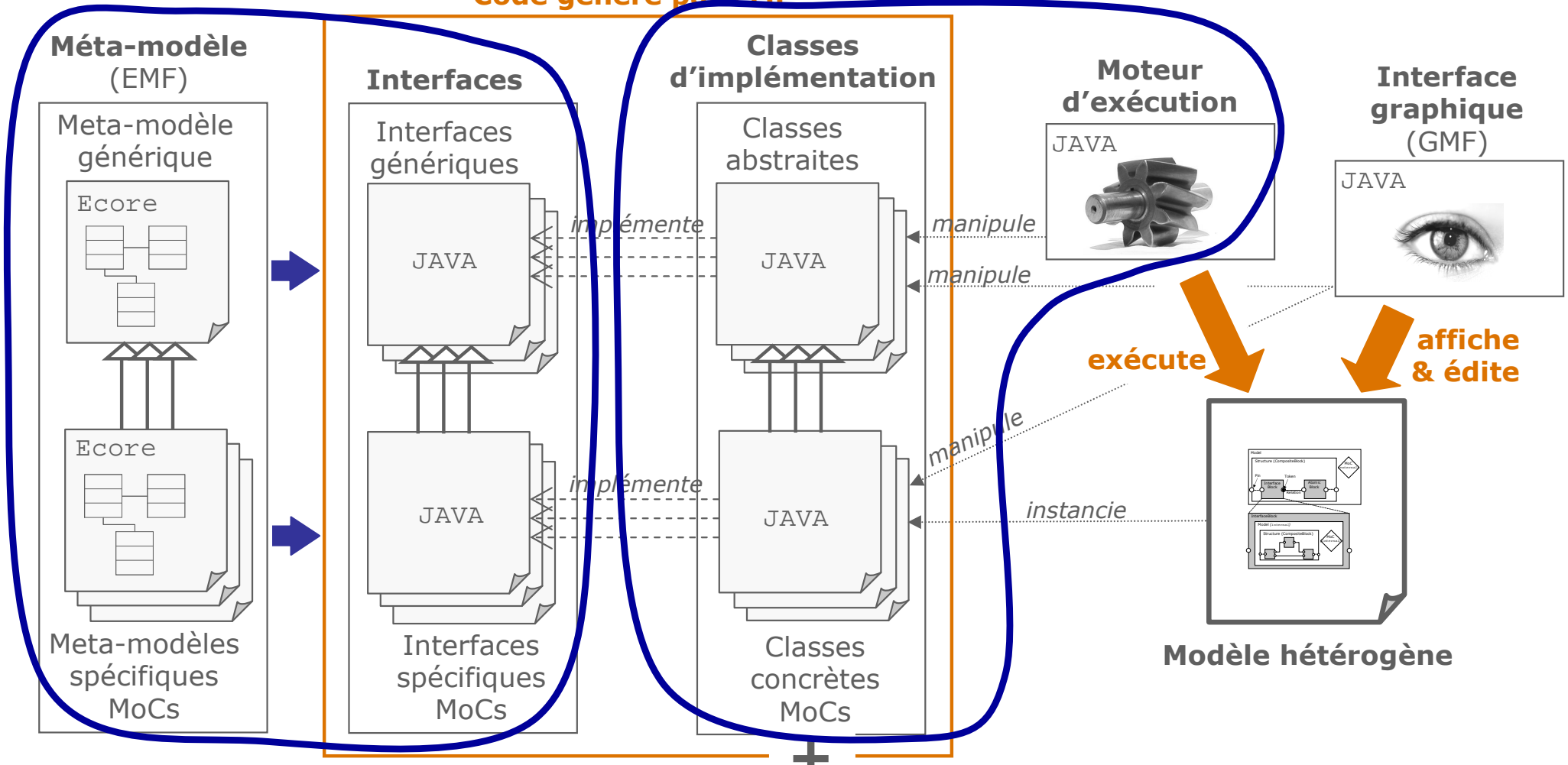
φορὰς ἡσυχίαν ὅς τιν' ἐ' μ' μουσική καταλαβὸν ἀναλογίαν εὐθέως ἔξει καὶ (τοια) τὸν ἴδιον μετέωρον θεωρίαν οὐθ' εἰς - ἢ γάρ...

Framework expérimental : architecture cible

Syntaxe abstraite

Code généré par EMF

Sémantique abstraite

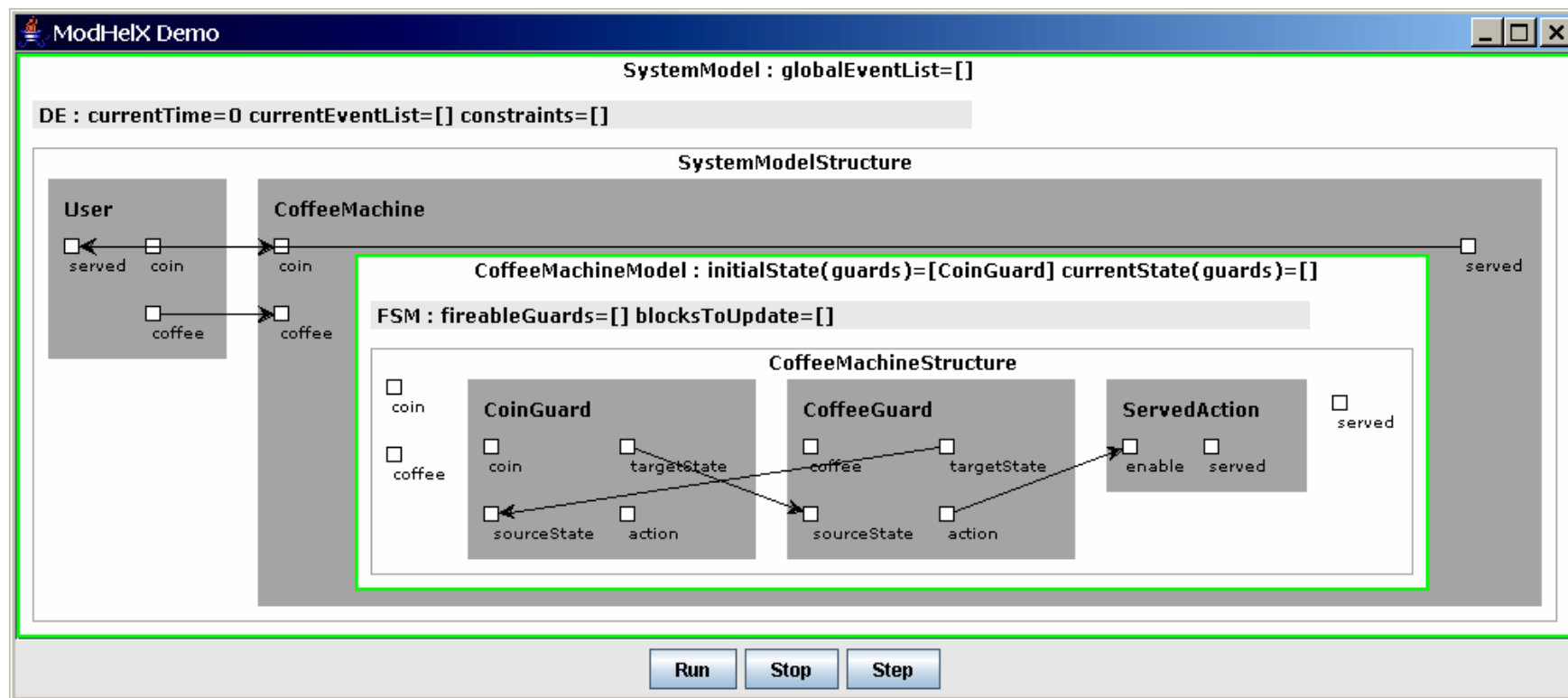


φορές ἡσυχίας ὅς τιν ἐ' μ' μουσική καταλαμβάν ἀναλογίαν εὐθέως ἔξει καὶ τούτῳ τὸν ὀμιλετώρων θεωρίαν οὐθ' εἰς - ἢ γάρ...

Descriptions sémantique (ImperativeOCL)

Interpréteur ImperativeOCL

Langage



<http://wwdi.supelec.fr/logiciels>

■ Validé sur :

- ▶ *charts (modèles modaux), DE, FSM
- ▶ Modèles homogènes et hétérogènes

■ Travail en cours sur d'autres MoCs dans le projet EDONA

Introduction

Hétérogénéité des modèles

Modélisation multi-paradigme

ModHel'X

Framework expérimental

▶ Conclusion & perspectives

Conclusion

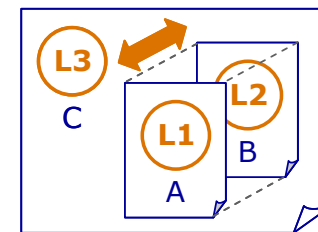
- Une caractérisation des origines du problème de l'hétérogénéité
 - ▶ 4 sources d'hétérogénéité
 - ▶ Notion d'objectif de modélisation

- Une revue des approches de la Modélisation Multi-Paradigme
 - ▶ 4 catégories de problèmes
 - ▶ Différents types d'approches identifiés et comparés

- ModHel'X : une approche de composition de modèles hétérogènes avec adaptation sémantique
 - ▶ Composition hiérarchique de modèles hétérogènes
 - ▶ Support pour la description des MoCs
 - ▶ Adaptation sémantique entre MoCs
 - ▶ Exécution de modèles

Perspectives

- Amélioration du **prototype**
 - ▶ Support EMF
 - ▶ Langage
 - ▶ Interface utilisateur
- Extension de la **gamme de MoCs** implémentés
- Travail sur l'**expression de l'adaptation sémantique** entre MoCs
- Travail sur le **modèle de temps**
- Support pour l'**hétérogénéité des vues**
 - ▶ Composition horizontale
- **Formalisation** des MoCs
 - ▶ Application au test de modèles hétérogènes
 - ▶ Application à la preuve de propriétés



Publications

■ Conférence internationale avec comité de lecture

- [1] F. Boulanger and C. Hardebolle. *Simulation of Multi-Formalism Models with ModHel'X*. Proceedings of the 2008 IEEE International Conference on Software Testing Verification and Validation (ICST 2008). IEEE Computer Society. p. 318-327. April 2008.

■ Workshop internationaux avec comités de lecture

- [2] C. Hardebolle and F. Boulanger. *ModHel'X: A Component-Oriented Approach to Multi-Formalism Modeling*. Models in Software Engineering - Workshops and Symposia at MoDELS 2007, Reports and Revised Selected Papers. LNCS, Springer. p. 247-258. June 2008.
- [3] C. Hardebolle and F. Boulanger and D. Marcadet and G. Vidal-Naquet. A Generic Execution Framework for Models of Computation. Proceedings of the 4th International Workshop on Model-based Methodologies for Pervasive and Embedded Software (MOMPES 2007) at the European Joint Conferences on Theory and Practice of Software (ETAPS 2007). IEEE Computer Society. p. 45-54. March 2007.

■ Journal international avec comité de lecture

- [4] C. Hardebolle and F. Boulanger. *Multi-formalism modelling and model execution*. International Journal of Computers and Applications (IJCA), special issue on the International Summer School on Software Engineering. ACTA Press. To be published in 2009.

■ Publication soumise : journal international avec comité de lecture, 2^e phase de revue en cours

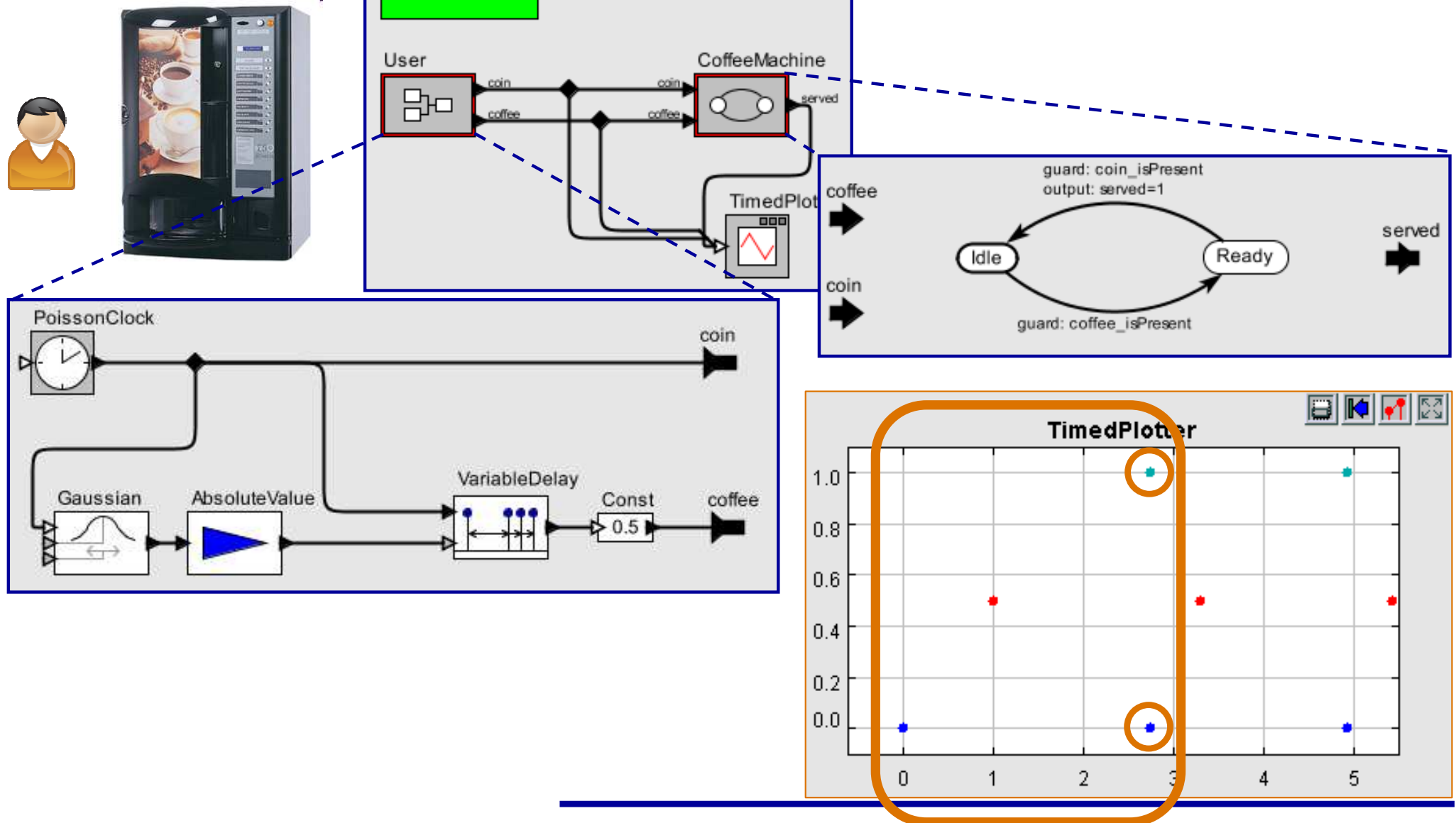
- [5] C. Hardebolle and F. Boulanger. *Exploring Multi-Paradigm Modeling Techniques*. Simulation: Transactions of the Society for Modeling and Simulation International, Special Issue on Multi-Paradigm Modeling. 23 pages. To appear in 2009.

Merci de votre attention...

ModHel'X : positionnement par rapport à PtolemyII

■ Adaptation sémantique explicite entre MoCs

Exemple : modélisation d'une machine à café



ModHel'X : positionnement par rapport à PtolemyII

■ Adaptation sémantique explicite entre MoCs

Exemple : modélisation
d'une machine à café

