

SMA : principes, enjeux et perspectives



P. MATHIEU
philippe.mathieu@univ-lille.fr

Université de Lille, CRISTAL

JFMS, Cargèse – 2 nov 2020

- 1 Principes
- 2 Les plateformes
- 3 Large échelle
- 4 Multi-échelles
- 5 L'approche orientée interactions

Un système multi-agents c'est quoi ?

Un système dans lequel interagissent des agents autonomes dotés de leurs propres caractéristiques et de leur propre comportement

“Agent” = 4 notions essentielles

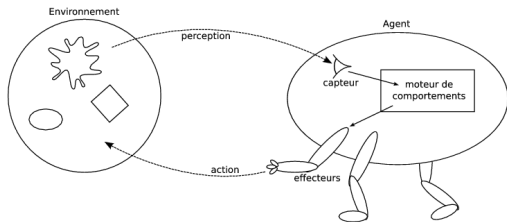
- Environnement
- Autonomie
- Comportement
- Interactions

Le système multi-agents :

- le système qui donne vie aux agents et leur permet d'interagir

Principes

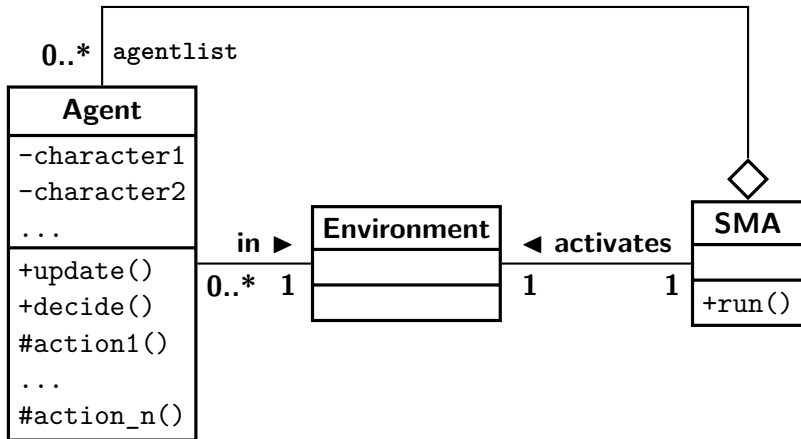
Comment fonctionne un agent ?



©Russel&Norvig

à chaque invocation :

- 1 Perception
- 2 Decision
- 3 Action



```
class Agent:
    def __init__(self,name) :
        self.name=name

    def decide(self):
        print("Bonjour ! My name is "+self.name)

    def update(self):
        return
```

```
sma = SMA()
sma.addAgent(Agent("paul"))
sma.addAgent(Agent("kim"))
sma.addAgent(Agent("Lisa"))
sma.run(10000)
```

```
import random
class SMA:
    def __init__(self):
        self.resetTicks()
        self.agentList = []

    def resetTicks(self):
        tick=0

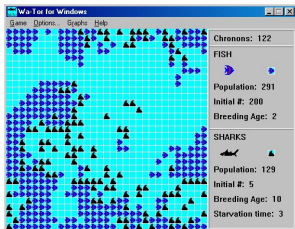
    def addAgent(self,ag):
        self.agentList.append(ag)

    def run(self,rounds):
        for i in range(rounds):
            self.runOnce()

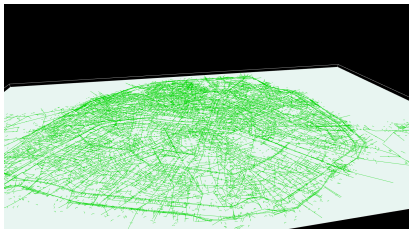
    def runOnce(self):
        self.tick+=1
        random.shuffle(self.agentList)
        for ag in self.agentList :
            ag.decide()
        print("tick "+str(self.tick)+" ended")
```

https://github.com/cristal-smac/mas_basics

Discrete Grid



GIS



Social networks



Agents physiques

- Limitée aux accointances
 - ▶ métaphore humaine, sociale
 - ▶ nécessite en général des actes de langages, de la négociation
- Broadcasting général
 - ▶ Tout le monde connaît toute l'information
- Stigmergie
 - ▶ communication bio-inspirée
 - ▶ L'environnement devient support de communication

	Fair scheduling Explicit agent handling by the simulation engine	Unfair scheduling Delegated to the language or operating system
Only 1 process all sequential	Simulated simultaneity possible (influence/reaction) Fair access to info	no simultaneity
N physical processes real execution simultaneity	increase execution time for agents	real time reasoning

- massive-software.com 2002,
- EA/Maxis Glassbox 2012



VIDEO massive.avi

- La recherche en SMA est en général multi-disciplinaire ; on offre des outils pour les autres disciplines
- En général on étudie un phénomène macroscopique bien identifié ...

La question de recherche !

Trouver le comportement le plus simple possible à mettre aux agents pour reproduire ce phénomène macroscopique ;

Appliquer le **Rasoir d'Ockham** , Etre fidèle au **KISS principle**,
Principe de parcimonie

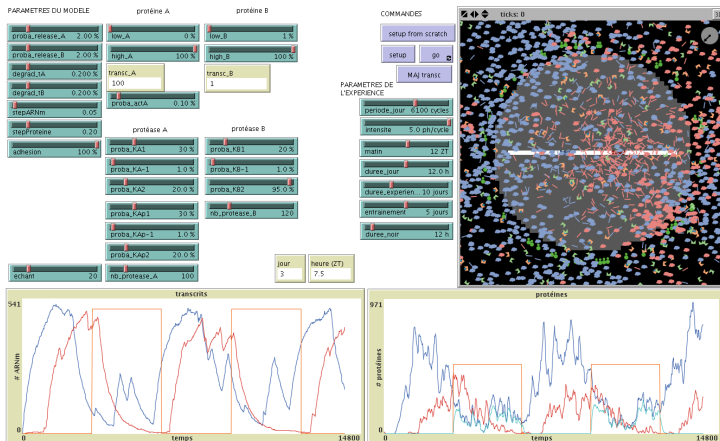
Le SMA permet de tester des hypothèses dans la discipline observée



Partout où il y a des entités et des comportements

- Biologie
- Ecologie
- Supply chain
- Transport
- Finance
- Théorie des jeux computationnelle
- Commerce
- Resolution de problèmes
- Formation
- Systèmes complexes

Liste non exhaustive



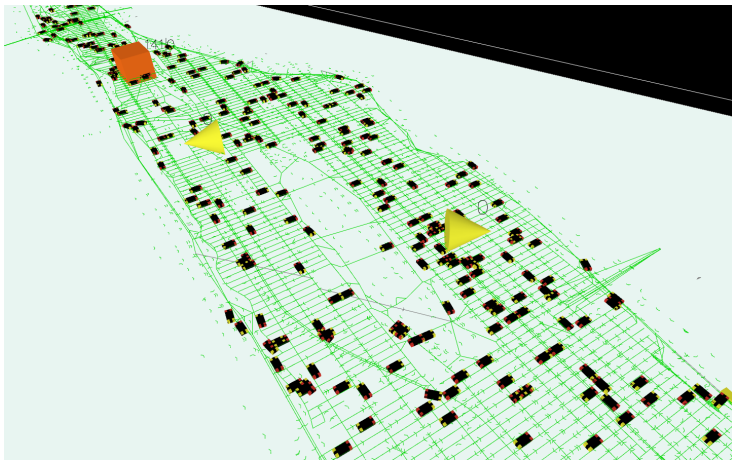
DEMO RegulationGenetique [Picault, JFSMA 2007]

- 1 Principes
- 2 Les plateformes**
- 3 Large échelle
- 4 Multi-échelles
- 5 L'approche orientée interactions

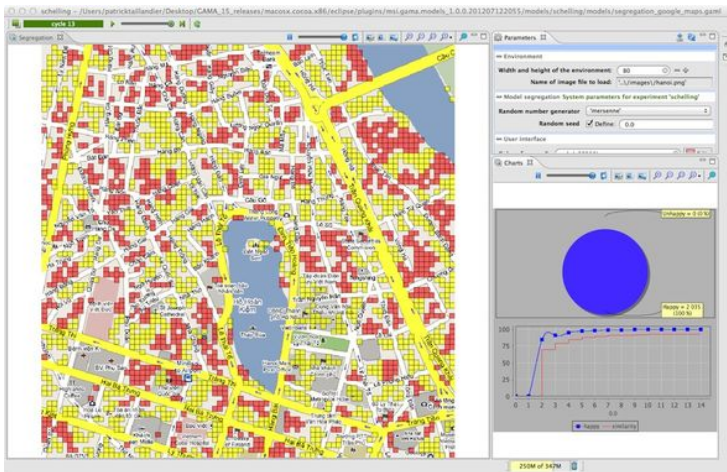
Certaines sont génériques

- Jade
- Netlogo
- Repast
- Gama
- Mason
- Madkit

D'autres plus spécifiques **Sumo**, **MatSim**, **Jason**, **ATOM**, ...
et plein d'autres plus confidentielles ... **Mesa**, **Cormas**, ...



VIDEO TrafficGen [Bonhomme et al, PAAMS 2015]



[Download](#)

Développée par l'IRD/UPMC au sein de l'unité mixte internationale
UMMISCO

- GAMA permet de construire des simulations à base d'agents spatialement explicites
- manipulation de données GIS
- Langage de modélisation en XML : GAML
- Plusieurs couches de description,
- Permet de manipuler des objets géographiques : Les routes, les bâtiments les villes peuvent être considérés comme des agents
- Une forme géométrique vectorielle peut être associée à chaque agent
- Nombreuses primitives sur ces formes (calculs de surfaces, d'intersection, de rotation, de distances, de plus court chemin, etc ...)
- Fonctionnalités multi-échelles (Agent émergent, et primitives d'agrégation, de fusion, d'ajout)

Plusieurs projets s'appuient sur GAMA, dans les domaines du transport, de la planif urbaine, de l'épidémiologie ou de l'environnement



Détails

Projet ESCAPE : Exploring by Simulation Cities Awareness on Population Evacuation (ANR 2016-2020)

- 1 Principes
- 2 Les plateformes
- 3 Large échelle**
- 4 Multi-échelles
- 5 L'approche orientée interactions

Jusqu'où va t-on avec un ordinateur récent (2018) ?

le cas ATOM [Mathieu et al, PAAMS 2012, Physica-A 2019]

- Atom est un marché financier artificiel
- Traders “intelligents” qui interagissent à travers un marché régissant plusieurs titres
- Atom est multi-agents, multi-ordres, multi-titres, intra et extra-day

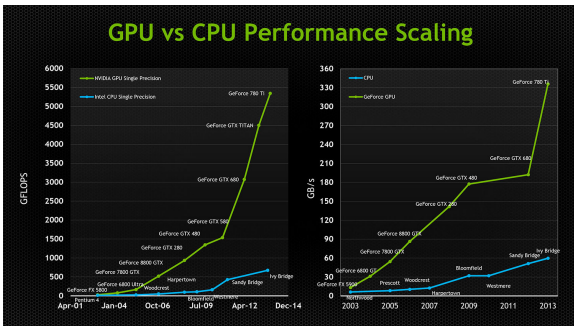
Vitesse

1000 ticks avec 1000 agents qui envoient 1 LimitOrder à chaque demande (donc 1.10^6 ordres, 500.000 prix) en 10 sec
100Hz

How to download ATOM



- Supercalculateur
- Clusters de CPU
- Grilles de calcul
- GPGPU



GPGPU

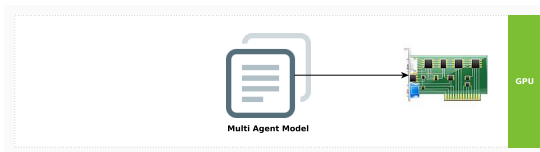
General-Purpose computing on Graphics Processing Units

Utiliser l'architecture massivement parallèle des cartes graphiques pour programmer

Large échelle

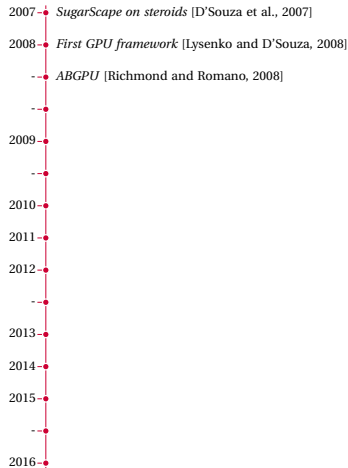
Deux types d'approches

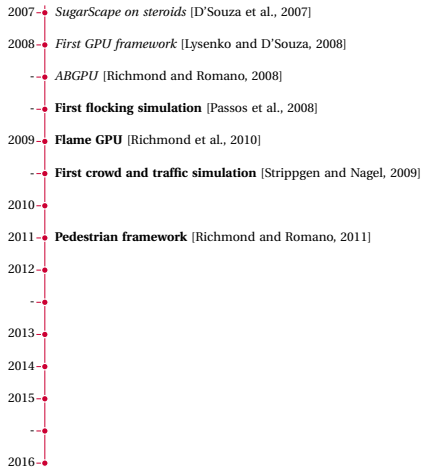
All-in-GPU

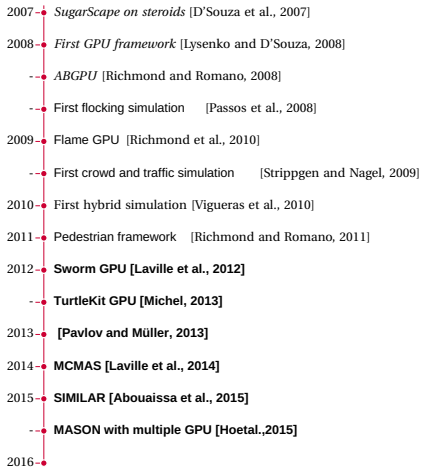


Hybrid









Idée générale :

- Continuer dans l'hybride
- Déléguer les calculs liés à l'environnement au GPU
- Le CPU s'occupe des agents (spécifique), le GPU s'occupe des calculs répétitifs de l'environnement
- Typiquement : gradients, phéromones, moyennes à chaque endroit de l'espace

Philosophie

Identifier les calculs des agents qui peuvent être transformés en dynamique environnementale et donc réalisée par des modules GPU.

Cette approche est mise en place dans **TurtleKit GPU** [Hermellin & Michel JFSMA'15 MABS'15] avec de très bonnes perfs !

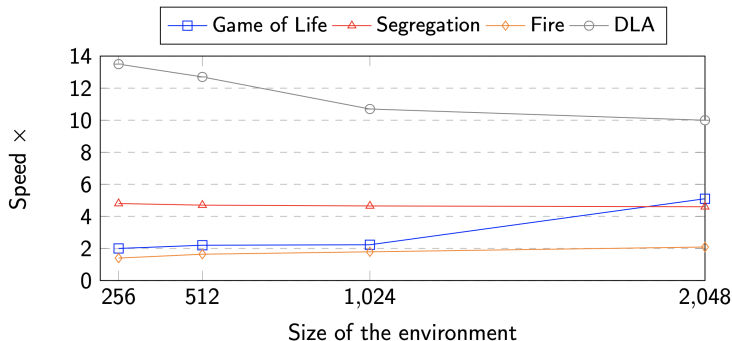
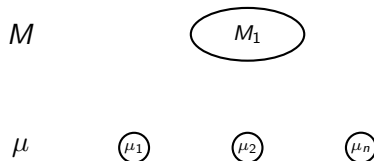


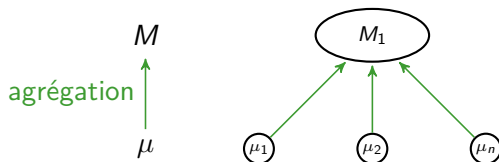
Figure: Coefficient of acceleration (between CPU and CPU+GPU versions)

- 1 Principes
- 2 Les plateformes
- 3 Large échelle
- 4 Multi-échelles**
- 5 L'approche orientée interactions

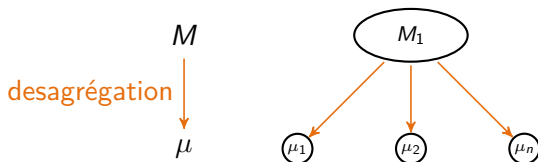
- On considère dans un modèle 2 niveaux relatifs : μ et M
- Un agent de M représente une agrégation d'agents de μ
- Simulation multi-niveaux : passage d'un niveau source (S) vers un niveau cible (T)



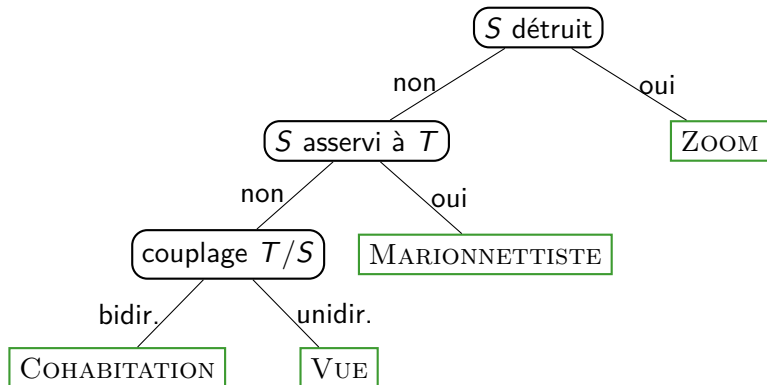
- On considère dans un modèle 2 niveaux relatifs : μ et M
- Un agent de M représente une agrégation d'agents de μ
- Simulation multi-niveaux : passage d'un niveau source (S) vers un niveau cible (T)



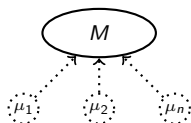
- On considère dans un modèle 2 niveaux relatifs : μ et M
- Un agent de M représente une agrégation d'agents de μ
- Simulation multi-niveaux : passage d'un niveau source (S) vers un niveau cible (T)



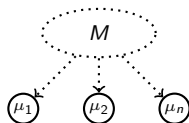
- Autonomie relative des agents des niveaux source et cible
 - ▶ sélection d'action (*decide*)
 - ▶ dynamique d'état (*update*)



- Le ou les agents du niveau source sont détruits à l'issue de la transformation



(a) Agrégation



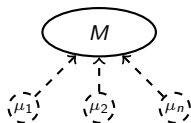
(b) Désagrégation

.....→ (dés)agrégation destructive

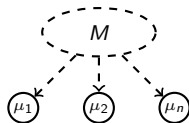
⊙ agent détruit

- Exemples en simulation de trafic routier :
 - ▶ Agrégation d'agents véhicules dans un flux [Bourrel :2003a]
 - ▶ Désagrégation dynamique d'un flux en agents véhicules [Bouha :2015]

- Le ou les agents du niveau source perdent leur autonomie comportementale à l'issue de la transformation



(a) Agrégation



(b) Désagrégation

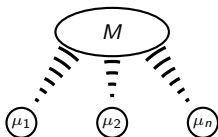
-----> délégation comportementale

() agent marionnette

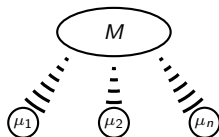
- Exemples :

- ▶ Agrégation de chasseurs en escouades [Mathieu :2013]
- ▶ Désagrégation : méthodologie top-down de SWARM [Minar :1996]

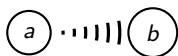
- L'état des agents du niveau cible est calculé à partir de celui du niveau source



(a) Agrégation



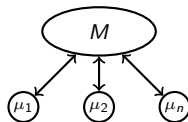
(b) Désagrégation



Délégation d'état : l'état de l'agent b est calculé à partir de celui de l'agent a

- Exemples :
 - ▶ Agrégation d'agents pour visualiser des phénomènes émergents
 - ▶ Désagrégation : *Zoom in* dynamique [Sewall :2011]

- Cohabitation des agents des niveaux sources et cibles

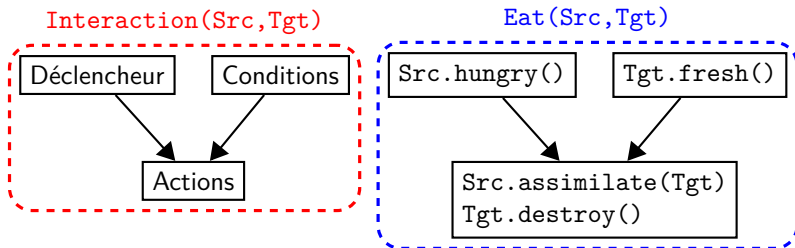


←→ échange d'information bidirectionnel

- Exemples :
 - ▶ Agrégation de larves de diptères [Morvan :2008]
 - ▶ Désagrégation : Génération de véhicules en aval d'un flux [Bourrel :2003a]

- 1 Principes
- 2 Les plateformes
- 3 Large échelle
- 4 Multi-échelles
- 5 L'approche orientée interactions

- focalisation excessive sur l'individu
 - ▶ forte dépendance comportement/agent
 - ▶ différenciation « agents » / « objets » (ressources)
 - ▶ complique la révision des modèles
- caractère fortement procédural du comportement
 - nécessité de séparer déclaratif/procédural :
 - ▶ simplifier l'acquisition d'expertise
 - ▶ meilleure intelligibilité du modèle
 - ▶ réutilisabilité accrue
 - ▶ homogénéité des concepts



Déclencheur : motivations implicites/buts explicites visés par les actions

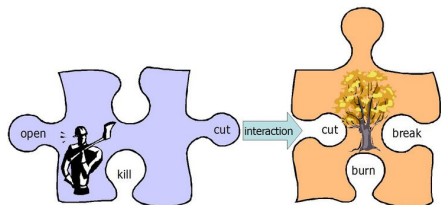
Conditions : conditions logiques requises pour l'exécution des actions

Actions : liste des actions à effectuer

L'approche orientée interactions

3 idées clefs

- toute entité pertinente est un **agent**
= entité dotée de capacités de **perception** et **d'action**
- tout comportement est une **interaction**
= **règle conditions/actions** impliquant plusieurs agents
- un **moteur générique** détermine quelles interactions peuvent avoir lieu



agents caractérisés par leur capacité à **effectuer** (**source**) ou **subir** (**cible**) une interaction
(Mathieu et al 2001)

Matrices d'interaction et de mise à jour

SOURCES \ TARGETS	∅	Trees	Grass	Sheep	Goats	Wolves
Trees						
Grass	[2]Grow (0) [2]Spread (1)					
Sheep	[2]Die (3) [2]Wander (0)		[2]Eat (2; 0)	[2]Mate (1; 0)		
Goats	[2]Die (4) [2]Wander (0)	[2]Climb (3; 1)	[2]Eat (2; 0)		[2]Mate (1; 0)	
Wolves	[2]Die (4) [2]Wander (0)			[2]Eat (3; 1)	[2]Eat (2; 1)	[2]Mate (1; 0)

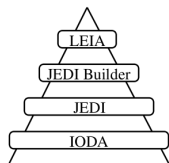
	UPDATE
Trees	
Grass	
Sheep	[2]Age (2) [2]BecomeSick (1)
Goats	[2]Age (2) [2]BecomeSick (1)
Wolves	[2]Age (1)

- représentation synthétique des comportements
- actions issues d'une décision vs. changement spontané d'état

L'approche IODA

La séparation déclaratif/procédural permet :

- d'**automatiser** l'implémentation des modèles et l'analyse des simulations
- d'explicitier les **biais de simulation** (Kubera et al 2008)



IODA Interaction-Oriented Design of Agent simulations
méthodologie et cadre de modélisation

JEDI Java Environment for the Design of agent Interactions
plateforme hautement paramétrable
(Kubera et al 2011)

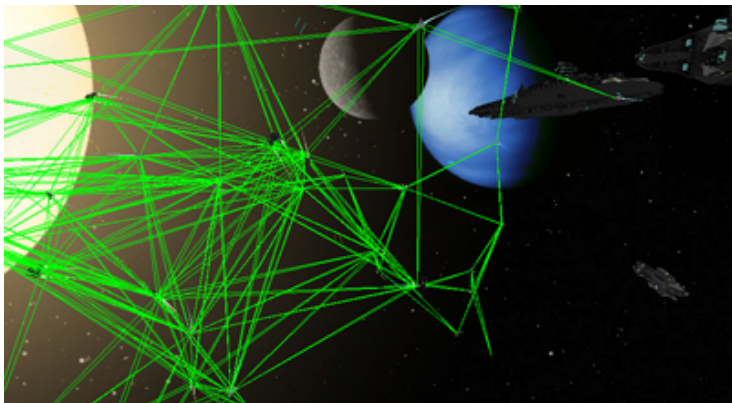
JEDI-Builder Générateur de code
modèle IODA → simulation JEDI

LEIA LEIA lets you Explore Interactions for your Agents
Explorateur de l'espace de simulation
(Gaillard et al 2010)

Galaxian

- démonstration dans un contexte "Ecran 180"
- implémentation de IODA sous le moteur de jeu Unity 3D

(Mathieu & Picault 2013)



Enjeux et Perspectives

- Tirer parti des architectures
- Automatiser les différentes formes de multi-échelles
- Améliorer les méthodologies : de la modélisation au code
- Mélanger les paradigmes : multi-modélisation
- Identifier de nouveaux domaines applicatifs

Questions ?

