

# SimExplorer / OpenMOLE

Et l'étude de vos modèles de calcul ne connaîtra plus de limites

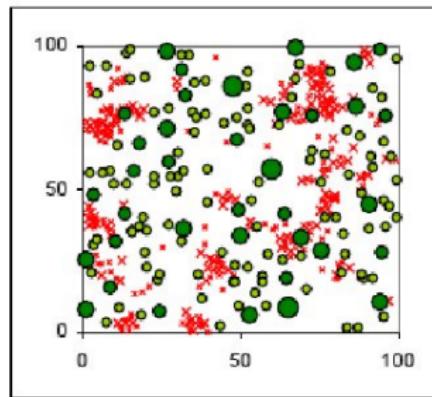
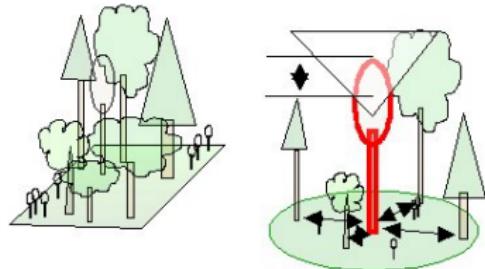
Nicolas Dumoulin

Cemagref — LISC, Aubière, France

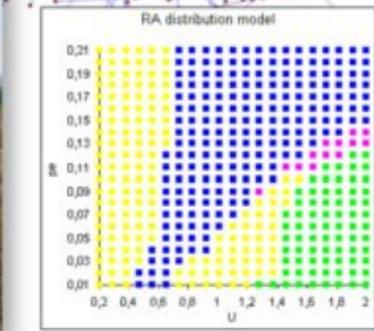
Giens, le 8 Juin 2010



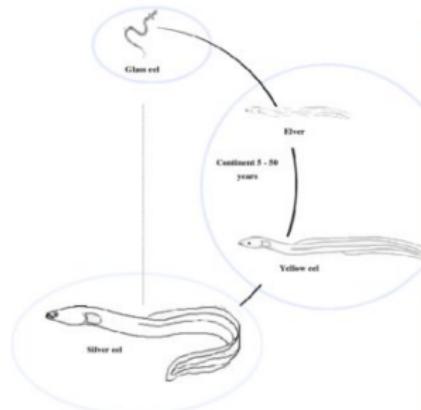
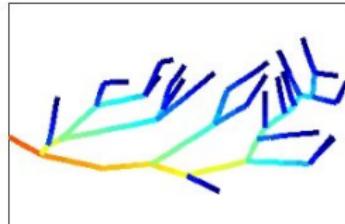
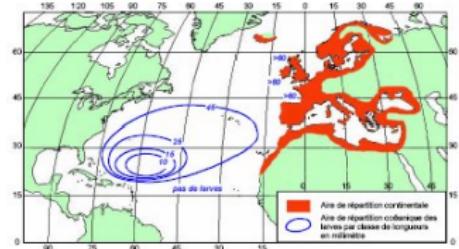
# Un système complexe à modéliser ...



# Un système complexe à modéliser . . .



# Un système complexe à modéliser . . .



A screenshot of the 'SimAquatic' software interface. The main window displays a simulation setup with 'Simulation parameters' and 'Environment file'. On the right, there is a plot titled 'Estuary' showing a blue-shaded area representing water flow or habitat. Below the main window, a 'Console' window displays command-line output related to the simulation.

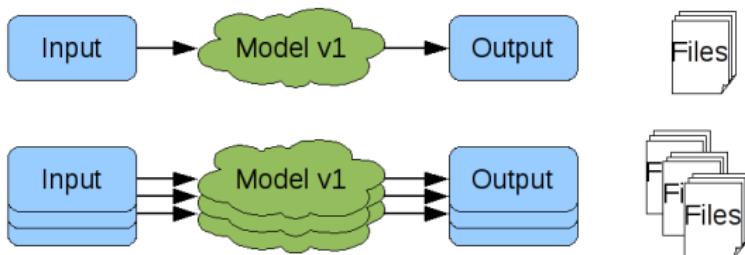
```
[2400; 17.-18] T=772.94 ux 0.0 uy 0.0 bed -12.81 surf 3.444; 0.0; rane: (223027.788, 96036.538)
[2400; 18.-19] T=772.94 ux 0.0 uy 0.0 bed -10.81 surf 3.444; 0.0; rane: (22334.788, 96036.538)
[2400; 19.-18] T=772.94 ux 0.0 uy 0.0 bed -12.81 surf 3.444; 0.0; rane: (22334.788, 96036.538)
[2400; 20.-18] T=772.94 ux 0.0 uy 0.0 bed -12.81 surf 3.444; 0.0; rane: (223027.788, 96036.538)
```

# On improvise ?



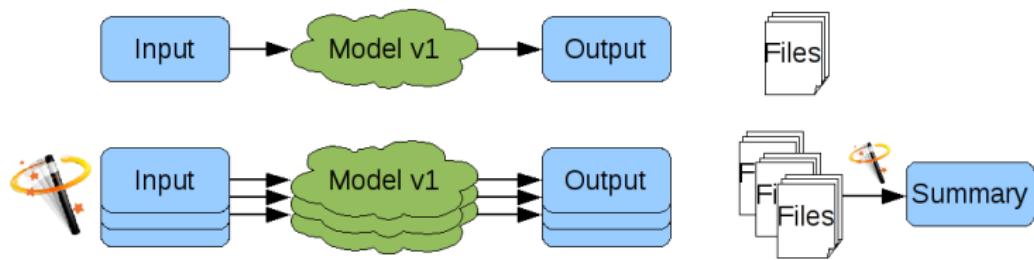
The simulation needs input and gives output

# On improvise ?



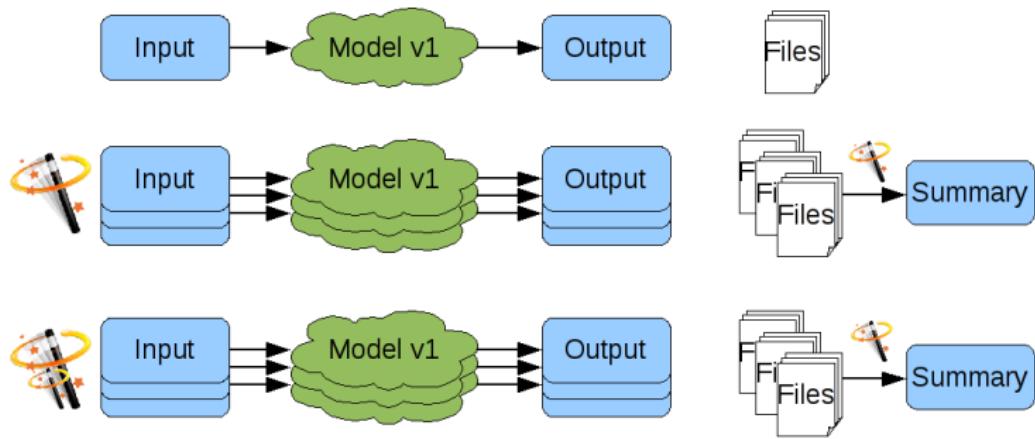
To explore the model, we need to launch several simulations

# On improvise ?



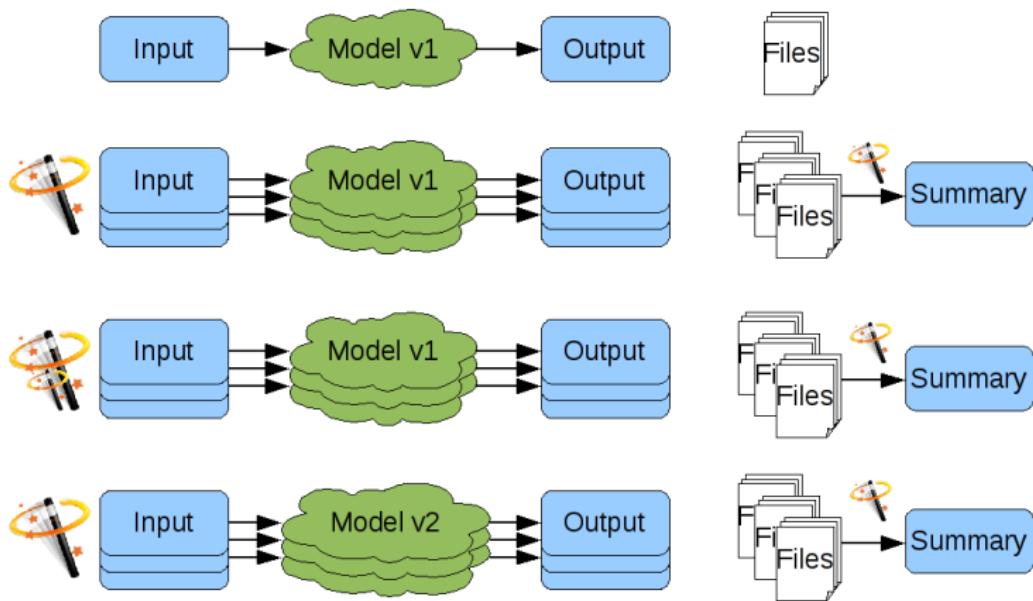
Design of experiments literature helps to build scenarii

# On improvise ?



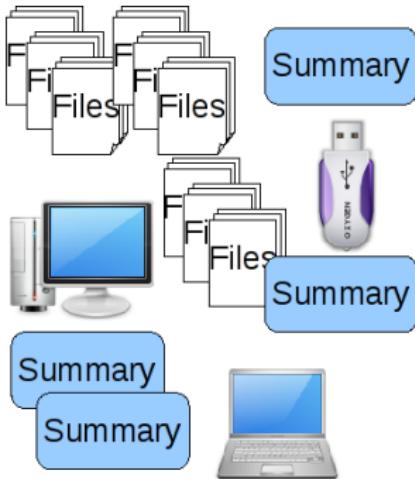
An exploration may bring to design a new exploration (more accurate)

# On improvise ?



Analysing the model behaviour may bring to conclude to modelling errors

# Les problèmes ne sont pas loin



- Which version of the model was used ?
- How to reproduce or modify this exploration ?
- What were the parameters of the model for these simulations ?
- What was the conclusion of this exploration ?

# Nécessité d'explorer l'espace des paramètres

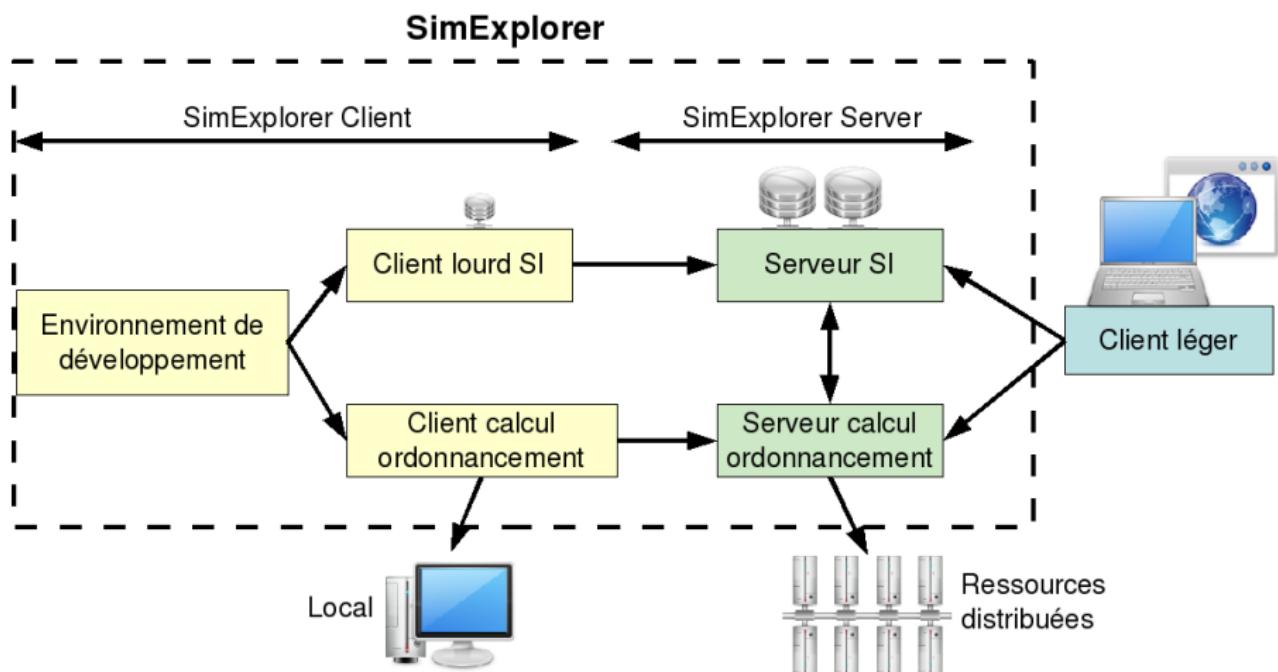
- ➊ Analyses de sensibilité (pour détecter les variables significatives)
- ➋ Détection de trajectoires typiques
- ➌ Calibration (paramètres minimisant une erreur)
- ➍ Calcul de surfaces de réponses (paramètres pour lesquels le modèle satisfait certaines propriétés)
- Nécessité de développer pour chaque modèle une application *ad hoc* pour l'explorer !

# Besoin d'un outil pour

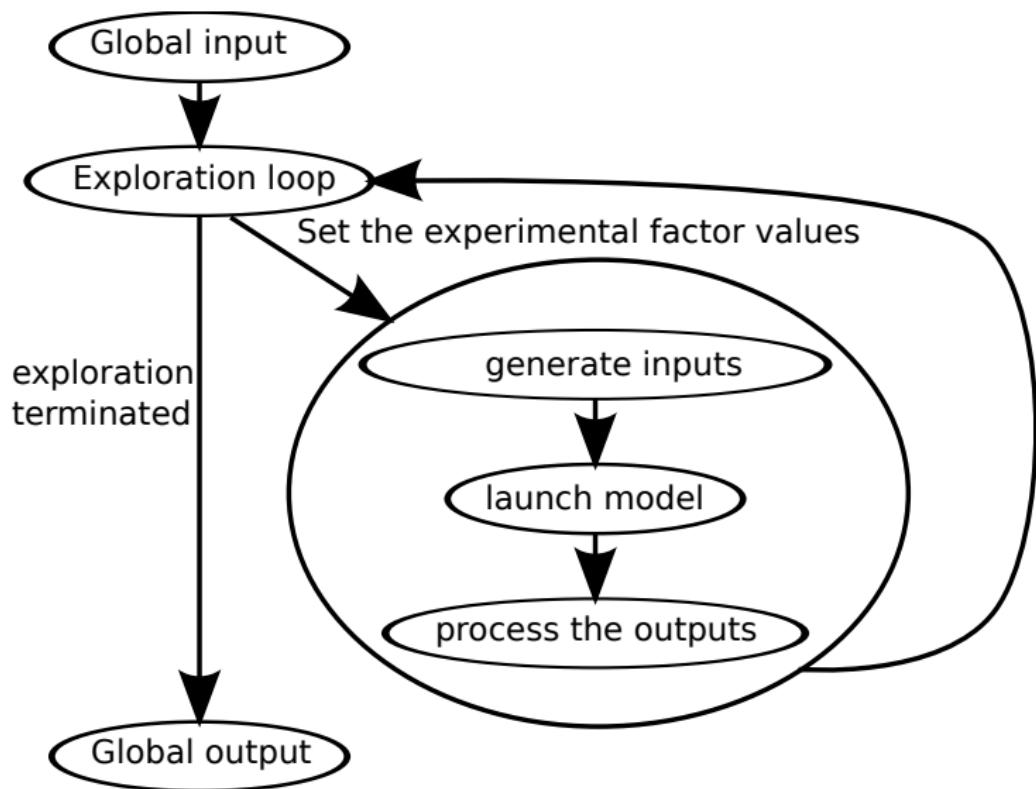
- Éviter de développer un outil *ad-hoc* pour chaque modèle
- Assister la conception de scénarios d'exploration
- Faciliter l'utilisation de clusters ou grilles
- Faciliter l'accès aux bibliothèques de plans d'expériences et d'analyse statistique
- Gérer la reproductibilité / traçabilité des expériences (qualité scientifique)

- Financement :
  - Investissement interne depuis plusieurs années
  - Projet région LifeGrid (financement FEDER)
  - Projet Européen PATRES
- Collaborations :
  - Réseau national Mexico
  - LIMOS (fédération TIMS)
  - Institut des systèmes complexes Paris Île de France.

# Architecture du logiciel



# Exploration séquentielle



# Exemple d'utilisation : IDE

The screenshot shows the SimExplorer 0.2.RC8 IDE interface. The main window is titled "SimExplorer 0.2.RC8".

**Applications View:** Shows a tree structure of applications under "bacteria". The "Exploration\_loop" node is selected.

**Editor View:** Displays configuration options for the selected application.

- CentralCompositeDesign:** Set to "Central Composite Design (CCD) options".
- Replication:** Set to "Replication of factorial points".
- Replicates of axial (star) points:** Set to 1.
- number of center points:** Set to 4.
- Alpha:** Options include "rotatable", "spherical", "practical", "face centered", and "other".

**Output View:** Shows the status "Exploration completed!".

**OutputStructure Window:** Shows the "outputStructure" is of type "Complex".

**Input Structure Window:** Lists input variables:

- inputStructure: Complex
- bi: Double
- di: Double
- alphapi: Double
- seed: Integer
- Ni: Integer
- Npi: Integer
- sigma2i: Double

**Variables Window:** Lists variables:

- input: System
- output: System
- factors: System

**Factors Window:** Lists factors:

- e: Integer, RangeInteger
- f: Integer, RangeInteger
- a: Double, RangeDouble (selected)
- b: Double, RangeDouble
- c: Double, RangeDouble
- d: Double, RangeDouble

# Exemple d'utilisation

The screenshot shows the SimExplorer 0.2.RC8 interface. The main window has tabs for 'Exploration' (selected), 'Output', and 'IDE Log'. The 'Exploration' tab displays several panels:

- Applications:** Shows a tree view of applications like 'bacteria', 'Exploration\_Step', 'Input\_processor', 'TemplateFileGenerator', 'Modal\_launcher', 'SystemEventLauncher', 'Output\_processing', 'GroovyProcessor', and 'Final\_processing'.
- Editor:** A panel for defining sampling methods, showing 'CentralCompositeDesign' selected under 'Central Composite Design (CCD) options'. It includes sections for 'Replication' (number of factorial points, replicates of axial (star) points, number of center points), 'Alpha' (radio buttons for rotatable, spherically, practically, face centered, or other), and 'Factors' (a table with columns Type, Domain, and RangeDouble).
- OutputStructure:** A panel for defining output structures, showing 'outputStructure' selected under 'Type'.
- Variables:** A panel listing variables: 'input' (System), 'output' (System), and 'factors' (System).
- Factors:** A panel listing factors: 'a' (integer, RangeInteger), 'b' (double, RangeDouble), 'c' (double, RangeDouble), and 'd' (double, RangeDouble).

To the right, there is a 'Simexplorer SI' window showing a list of 'Exploration applications' with details like name, type, language, and code.

Below the interface, three small diagrams labeled (a), (b), and (c) are shown:

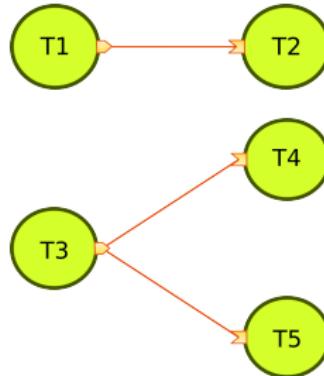
- (a) Machine locale: A 2D scatter plot with axes from -0.00 to 0.00, showing a single diagonal line segment.
- (b) Cluster de 15 nœuds: A 2D scatter plot with axes from -0.00 to 0.00, showing a cluster of 15 horizontal line segments forming a downward-sloping curve.
- (c) Grille Auvergrid: A 2D scatter plot with axes from -0.00 to 0.00, showing a grid of vertical and horizontal line segments forming a grid pattern.

At the bottom right, there is a 3D scatter plot with axes from 0 to 1024, showing a complex, multi-colored surface with a yellow-to-blue gradient and numerous small arrows indicating local gradients or vectors.

FIG. 9: Diagrammes de Gantt de l'exploration du modèle de test (60 tâches).

# Naissance d'OpenMOLE

- Conception séquentielle limitée difficilement adaptable ou extensible
- Remise à plat des concepts pour aboutir à une approche par chaîne de traitement modulaire (*Workflow*)
- Relation séquentielle entre des tâches
- Représentation parallèle



Objectif : Rendre les tâches indépendantes de leur environnement d'exécution

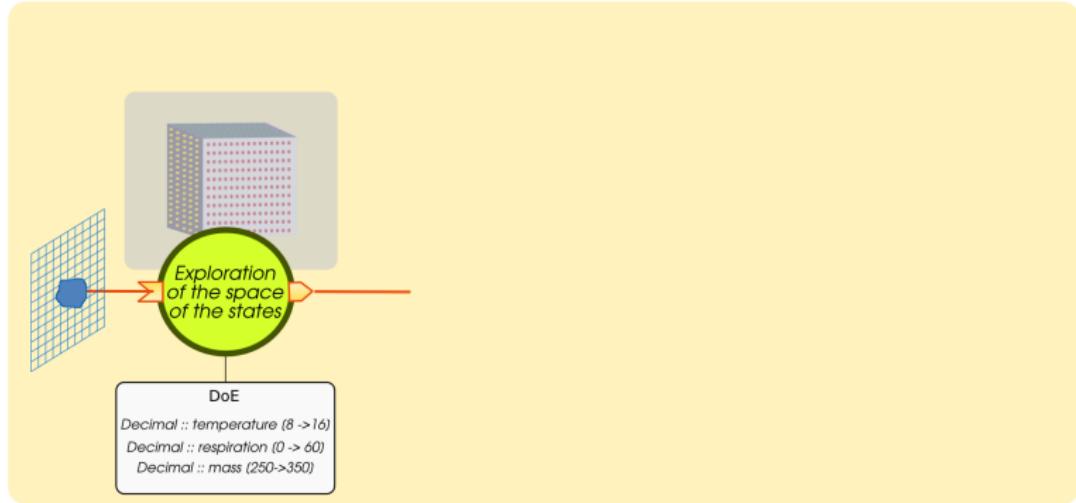
# Les concepts d'OpenMOLE

- Les **tâches** sont assimilables à une fonction et définit ses variables et fichiers d'**entrées/sorties**
- La **capsule** accueille une tâche et peut être reliée à une autre capsule à l'aide de **transition**
- Les transitions peuvent être conditionnelles
- L'ensemble constitue un *workflow*
- L'exécution se fait en désignant la capsule de départ, transformant le **workflow** en **MOLE**
- Un gestionnaire de plugin permet d'étendre et spécialiser ces concepts

# Application à la viabilité



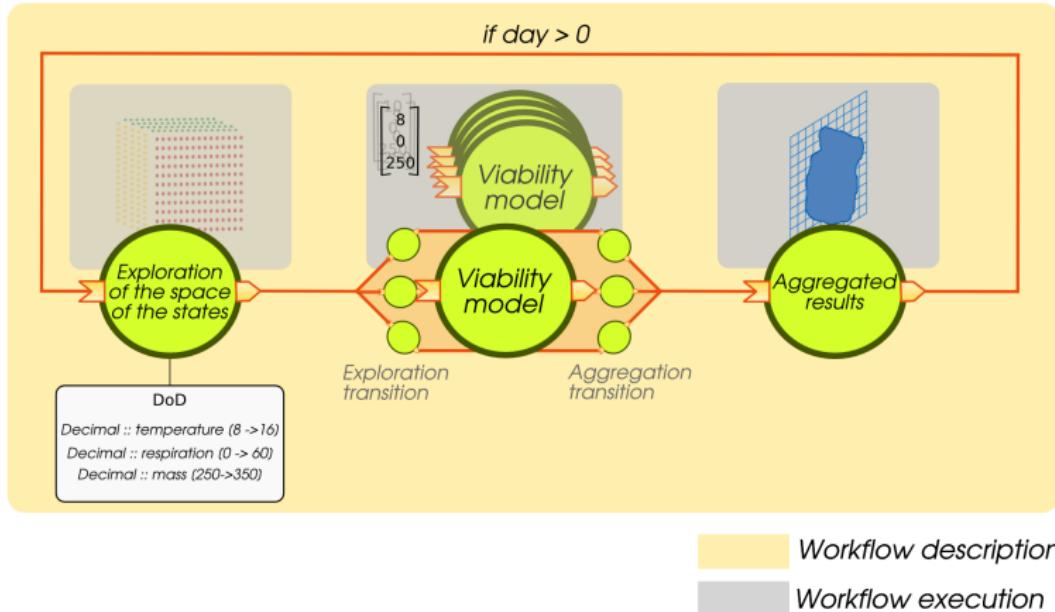
# Application à la viabilité



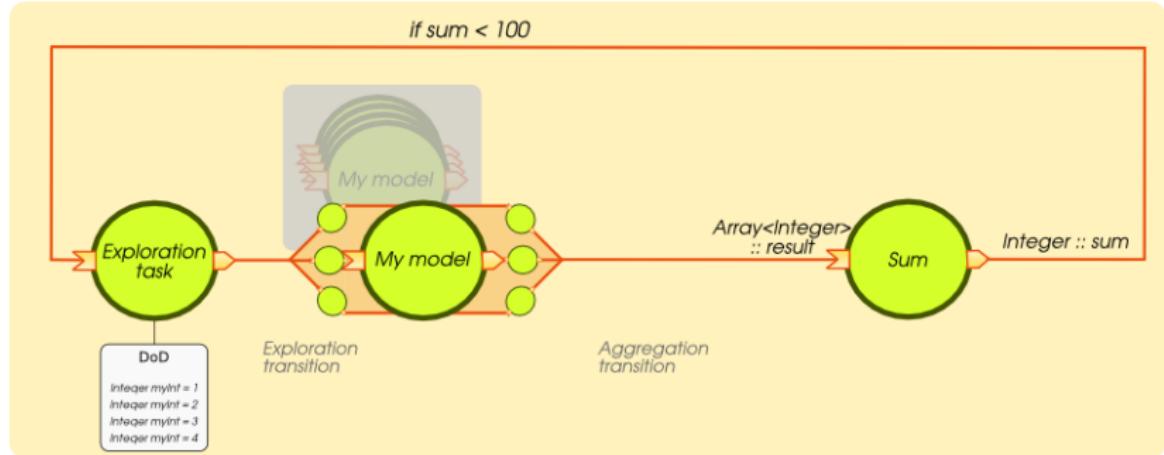
Workflow description

Workflow execution

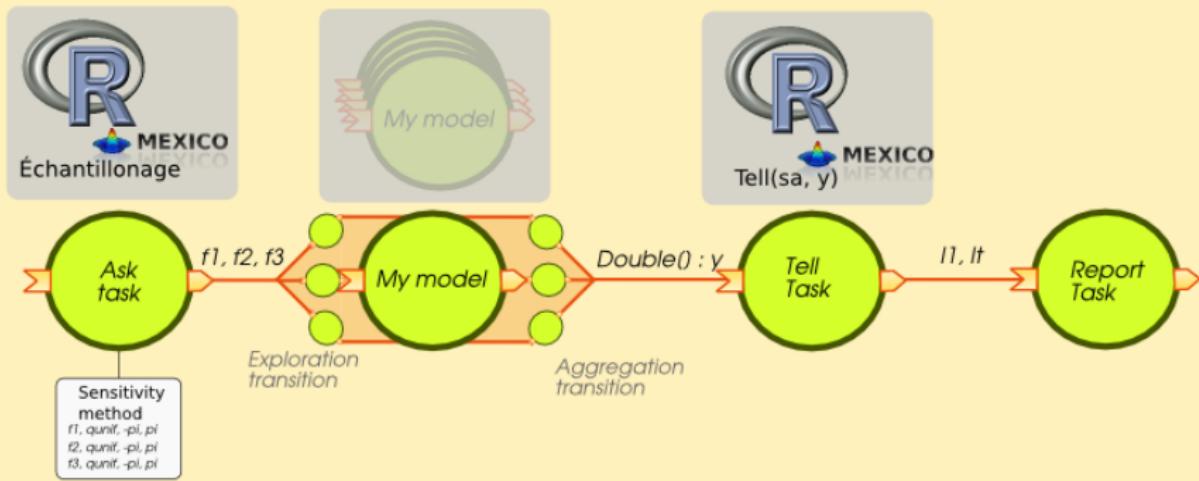
# Application à la viabilité



# Workflow d'exploration



# Workflow pour sensitivity



# Workflow pour sensitivity

```
import org.openmole.plugin.task.groovy.GroovyTask
import org.simexplorer.openmole.plugin.task.sensitivity.*

sensitivity.addFactor("f1", Double, new RFunctionDomain("qunif","-pi","pi"))
sensitivity.addFactor("f2", Double, new RFunctionDomain("qunif","-pi","pi"))
sensitivity.addFactor("f3", Double, new RFunctionDomain("qunif","-pi","pi"))

modelTask = new GroovyTask("model")
modelTask.setCode('sleep(3)\ny = Math.sin(f1) + 7.0 * Math.pow(Math.sin(f2), 2) + 0.1 * →
    ↪ Math.pow(f3, 4) * Math.sin(f1)')
sensitivity.setModelTask(modelTask)

reportTask = new GroovyTask("report")
reportTask.setCode('println "First order = ${l1}\nTotal order = ${lt}"')
sensitivity.setReportTask(reportTask)

execution = sensitivity.fast99(1000)
execution.start()
```

# Workflow pour sensitivity sur une grille

```
import org.openmole.plugin.task.groovy.GroovyTask
import org.simexplorer.openmole.plugin.task.sensitivity.*

sensitivity.addFactor("f1", Double, new RFunctionDomain("qunif","-pi","pi"))
sensitivity.addFactor("f2", Double, new RFunctionDomain("qunif","-pi","pi"))
sensitivity.addFactor("f3", Double, new RFunctionDomain("qunif","-pi","pi"))

modelTask = new GroovyTask("model")
modelTask.setCode('sleep(3)\ny = Math.sin(f1) + 7.0 * Math.pow(Math.sin(f2), 2) + 0.1 * →
    ↪ Math.pow(f3, 4) * Math.sin(f1)')
sensitivity.setModelTask(modelTask)

reportTask = new GroovyTask("report")
reportTask.setCode('println "First order = ${l1}\nTotal order = ${lt}"')
sensitivity.setReportTask(reportTask)

import org.openmole.plugin.environment.glite.GliteEnvironmentDescription
sensitivity.setModelEnvironment(new GliteEnvironment("vo.iscpif.fr",
    "voms://grid12.lal.in2p3.fr:20013/O=GRID-FR/C=FR/O=CNRS/OU=LAL/CN=grid12.lal.in2p3. →
        ↪ fr",
    "ldap://topbdii.grif.fr:2170"))
)

execution = sensitivity.fast99(1000)
execution.start()
```

# Conclusion

- Plateforme flexible, ouverte et collaborative
- SimExplorer et OpenMOLE diffusés en GPL
- Applications 100% portables (bientôt)
- Accès aux ressources de calcul distribuées

<http://www.simexplorer.org>

<http://www.openmole.org>