



Présentation générale

Concepts & Utilisation

JC. Fabre, M. Rabotin, D. Crevoisier, et al.

LISAH - Laboratoire d'étude des Interactions Sol-Agrosystème-Hydrosystème



This document is licensed
under Creative Commons license

Plan

- 1 **Présentation générale**
 - Contexte & Historique
 - Approche de modélisation
 - Description de la plateforme
- 2 Fonctionnement
- 3 En pratique
- 4 Jeux de données
- 5 Compléments

Le paysage



Modélisation intégrative du paysage

- Modélisation et simulation de processus spatialisés, **interagissant fortement dans le temps et l'espace** *i.e. flux de matière (eau, terre, énergie, ...), activités humaines, ...*
- Approche **pluri-disciplinaire**
- Approche équilibrée entre **modélisation des processus** et **représentation de l'espace**

⇒ Approche **complexe**

⇒ Nécessité d'**outils en support** à la modélisation intégrative du paysage

Historique

Développement du modèle MHYDAS au LISAH depuis 1995

- des concepts de **modélisation spatio-temporelle**,
- une représentation des **éléments du paysage**,
- ciblé sur hydrologie de surface et souterraine, polluants, érosion, agronomie, ...

Nouveaux besoins, nouveaux contextes (2000's)

- montée en puissance de l'**approche par modélisation/simulation**,
- grande **variété de processus spatiaux couplés**,
- **collaborations** de recherche de plus en plus nombreuses (projets sur AO),
- études des **services écosystémiques** dans les **agro-écosystèmes** et au delà

Objectifs de développement d'une plateforme "Paysage"

Un environnement logiciel générique pour la **modélisation spatialisée**, tenant compte des **processus**, des **éléments du paysage** et de leurs **interactions**

Un outil ouvert pour le travail en collaboration, la **capitalisation**, le **partage**, la **réutilisation** des modèles, des connaissances, des compétences

Une démarche d'ingénierie logicielle afin de proposer un logiciel basé sur des **technologies** récentes, maintenables, et pouvant **s'interfacer avec des outils et modèles existants**

⇒ Développement de la plateforme OpenFLUID depuis 2005, première version en juillet 2007

Objectifs de développement d'une plateforme "Paysage"

Un environnement logiciel générique pour la **modélisation spatialisée**, tenant compte des **processus**, des **éléments du paysage** et de leurs **interactions**

Un outil ouvert pour le travail en collaboration, la **capitalisation**, le **partage**, la **réutilisation** des modèles, des connaissances, des compétences

Une démarche d'ingénierie logicielle afin de proposer un logiciel basé sur des **technologies** récentes, maintenables, et pouvant **s'interfacer avec des outils et modèles existants**

⇒ Développement de la plateforme OpenFLUID depuis 2005, première version en juillet 2007

Objectifs de développement d'une plateforme "Paysage"

Un environnement logiciel générique pour la **modélisation spatialisée**, tenant compte des **processus**, des **éléments du paysage** et de leurs **interactions**

Un outil ouvert pour le travail en collaboration, la **capitalisation**, le **partage**, la **réutilisation** des modèles, des connaissances, des compétences

Une démarche d'ingénierie logicielle afin de proposer un logiciel basé sur des **technologies** récentes, maintenables, et pouvant **s'interfacer avec des outils et modèles existants**

⇒ Développement de la plateforme OpenFLUID depuis 2005, première version en juillet 2007

Objectifs de développement d'une plateforme "Paysage"

Un environnement logiciel générique pour la **modélisation spatialisée**, tenant compte des **processus**, des **éléments du paysage** et de leurs **interactions**

Un outil ouvert pour le travail en collaboration, la **capitalisation**, le **partage**, la **réutilisation** des modèles, des connaissances, des compétences

Une démarche d'ingénierie logicielle afin de proposer un logiciel basé sur des **technologies** récentes, maintenables, et pouvant **s'interfacer avec des outils et modèles existants**

⇒ Développement de la plateforme OpenFLUID depuis 2005, première version en juillet 2007

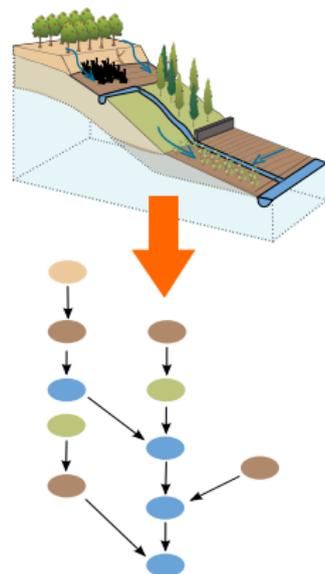
Représentation Numérique du Paysage (RNP)

Représentation sous la forme d'un **graphe**
d'**unités spatiales connectées**

- Les unités spatiales sont réparties en **classes** (surface, fossés, souterrain, ...)
- Les unités spatiales sont **qualifiées** (propriétés géométriques, physiques, ...)

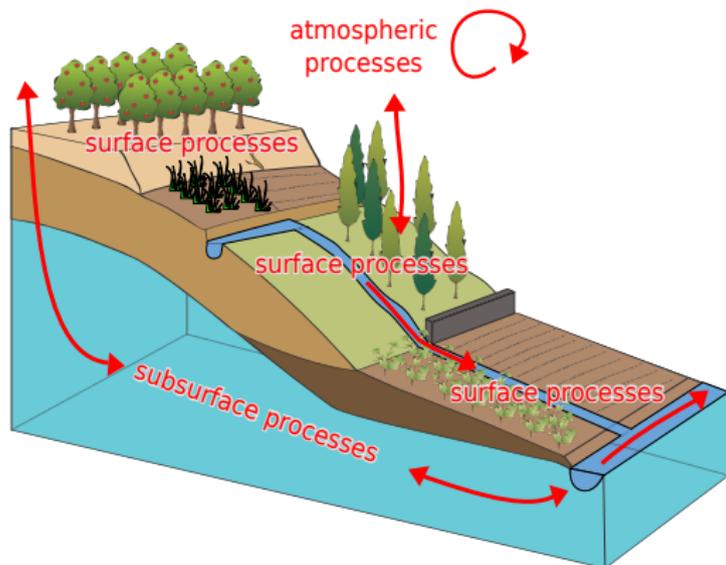
Ce graphe peut être construite en tenant compte

- du croisement de couches d'information spatiales (géométrie, relief, propriétés pertinentes, ...)
- des contraintes liées aux modèles qui vont être appliqués.



Modélisation de processus spatialisés

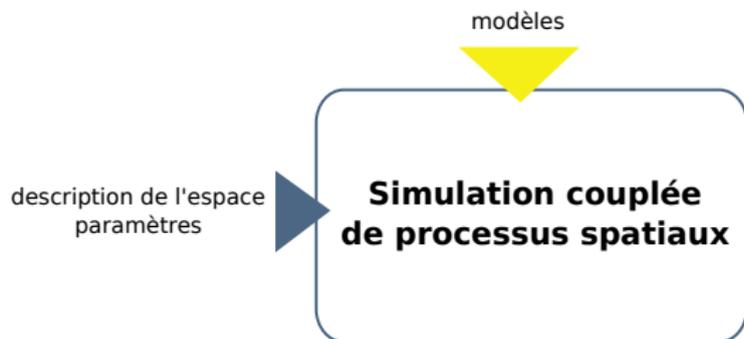
- Les processus sont associés à des unités spatiales
- Ces processus temporels donnent lieu à des **modifications de l'état** des unités spatiales ⇒ **variables d'état** des unités
- L'ensemble des processus couplés forme un **modèle couplé**



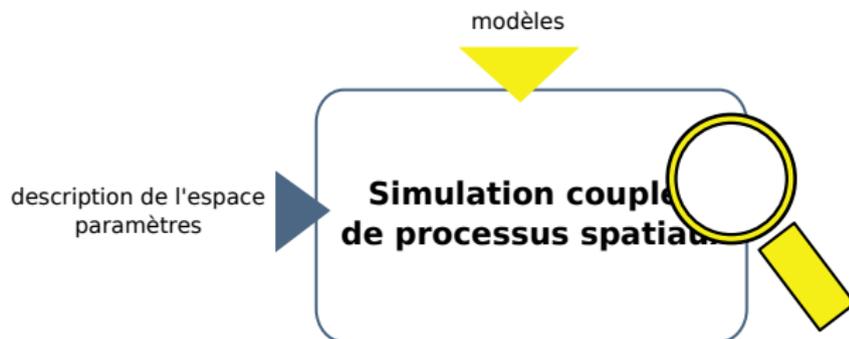
Démarche de simulation

**Simulation couplée
de processus spatiaux**

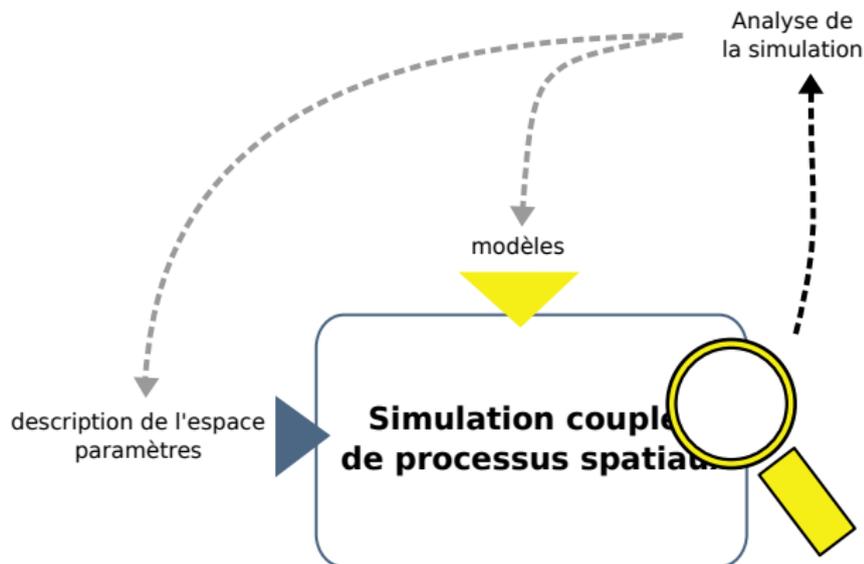
Démarche de simulation



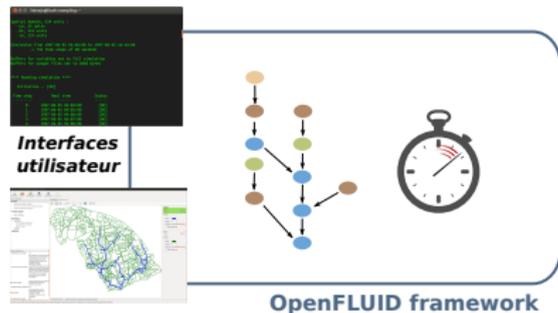
Démarche de simulation



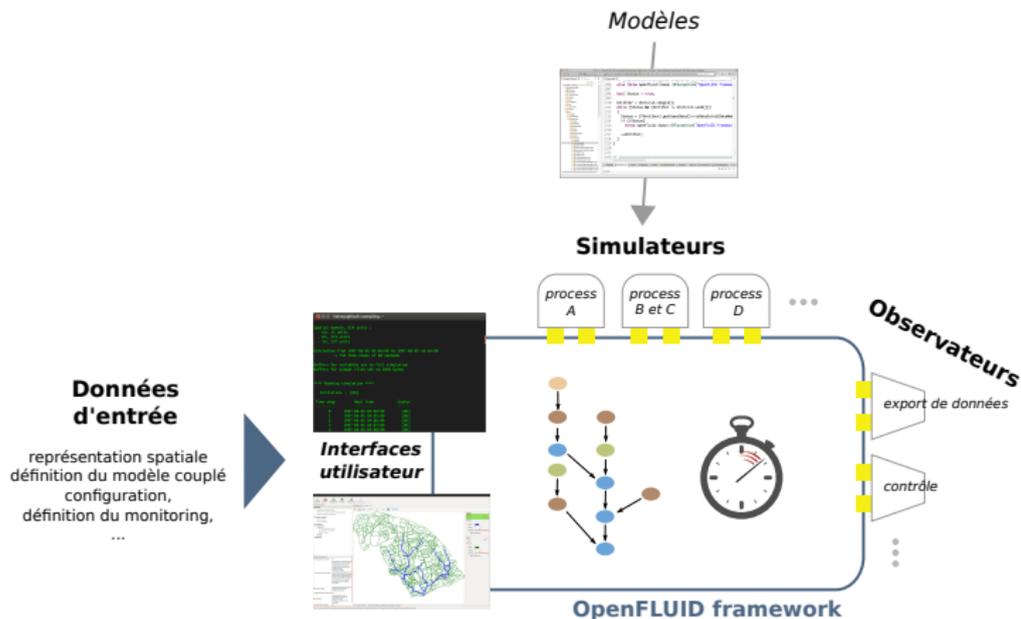
Démarche de simulation



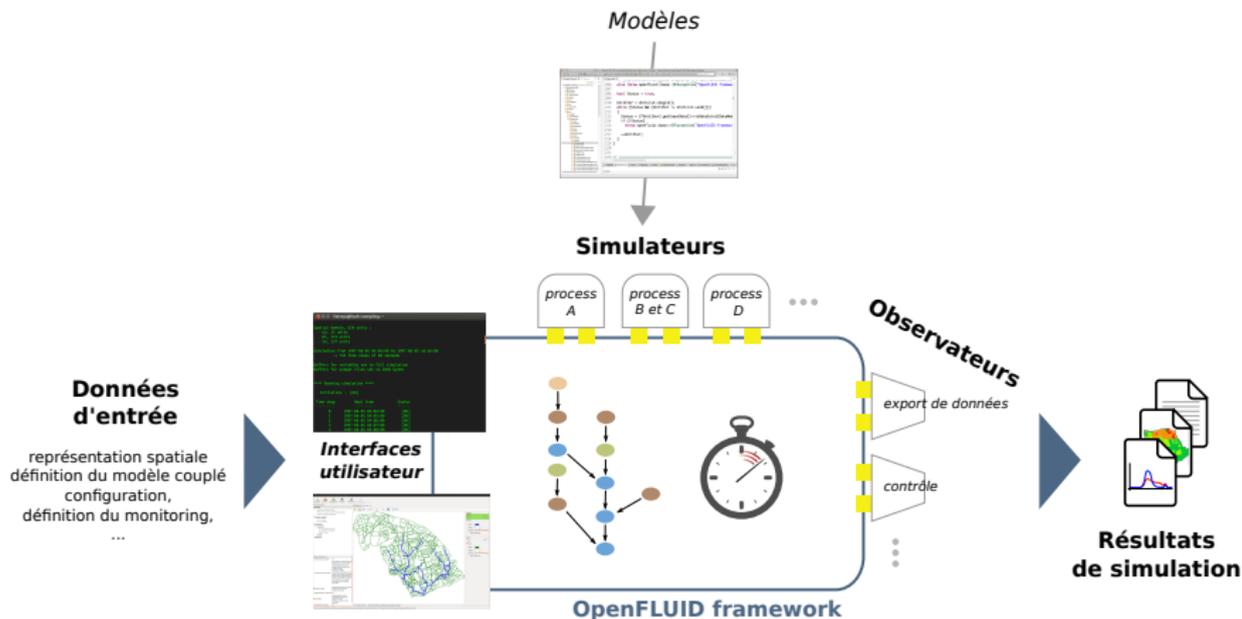
La plateforme OpenFLUID



La plateforme OpenFLUID



La plateforme OpenFLUID



Plan

- 1 Présentation générale
- 2 Fonctionnement**
 - Structures
 - Données
- 3 En pratique
- 4 Jeux de données
- 5 Compléments

Framework OpenFLUID

Structure de construction et couplage de modèles, coeur de la plateforme

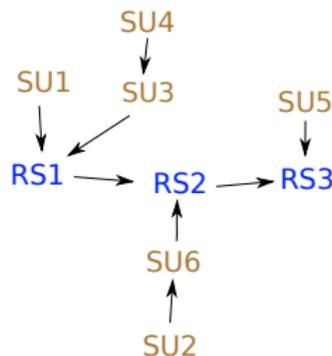
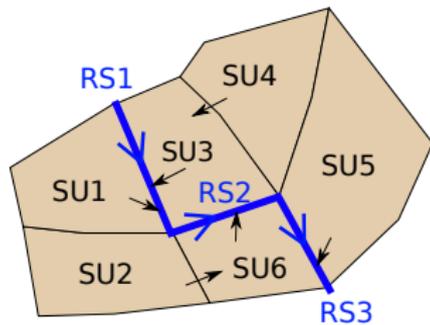
- Représentation de l'espace sous la forme de **graphes d'unités spatiales**, avec multi-échelle possible
- Modélisation couplée basée sur des **simulateurs** branchés dynamiquement lors de la simulation
- Gestion du **monitoring** de la simulation
- Gestion du **déroulement de la simulation** et des échanges entre simulateurs (dans le temps et dans l'espace)

Le framework est un ensemble de bibliothèques logicielles intégrant les fonctionnalités de la plateforme (libopenfluid-xxx)

Représentation numérique du paysage (RNP)

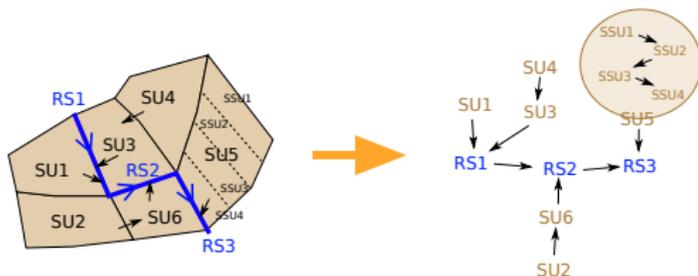
Le paysage est représenté par OpenFLUID sous la forme d'un **graphe orienté**

- Les noeuds sont les **unités spatiales** représentant le paysage, rangés par **classe d'unité**
- Les arcs orientés sont les **connexions** entre les unités spatiales
- Chaque noeud porte des **attributs propres** à l'unité spatiale qu'il représente



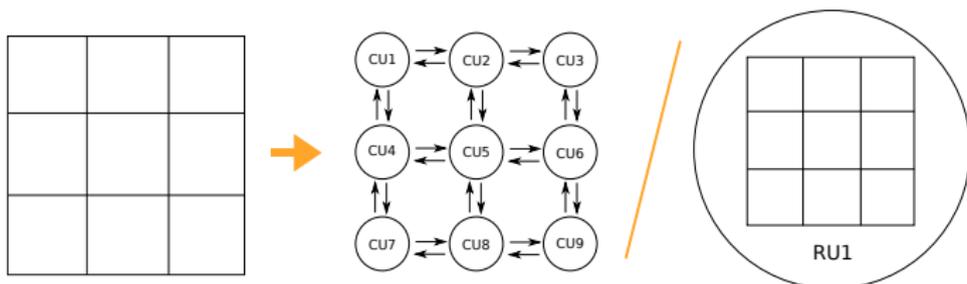
Représentation numérique du paysage (RNP)

Un noeud peut contenir un graphe pour des approches **multi-échelles**



Pour une modélisation orientée **raster**, différentes approches sont possibles:

- 1 unité spatiale par cellule du raster
- 1 unité spatiale portant une **matrice** correspondant au raster



Modèle couplé

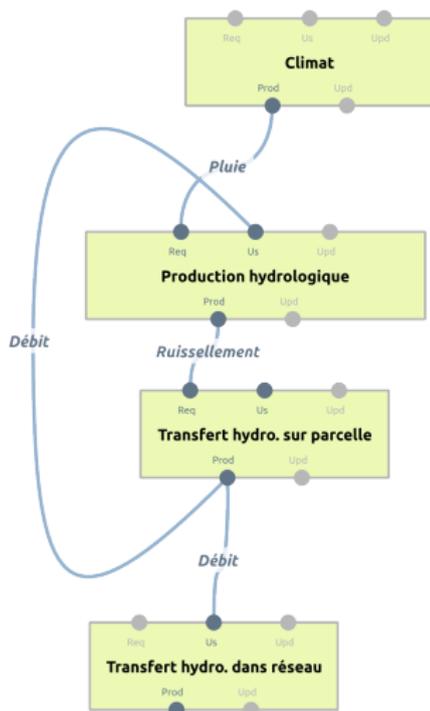
Un modèle couplé est une **liste de simulateurs** qui seront branchés au framework lors de la simulation

Le couplage entre simulateurs se fait par **l'échange spatio-temporel de variables**

Chaque simulateur a son propre pas de temps.

Lors de chaque exécution, il précise la **la planification pour sa réexécution**, parmi les possibilités suivantes:

- une durée en secondes
- le deltat par défaut fixé pour la simulation
- une seule fois à la fin
- jamais



Observateurs OpenFLUID et Monitoring

Un observateur est un code source qui permet de "regarder" les informations de simulation en continu. Il ne peut modifier l'état de la simulation.

Exemples d'observateurs

- export de résultats de simulation (format CSV, BD, ...)
- export de représentations 2D/3D de simulations (vers SIG, GoogleEarth, ...)
- contrôle en continu de simulations
- ...

Un observateur est construit sous la forme d'un plugin pour le framework OpenFLUID.

Le monitoring est une liste d'observateurs qui seront branchés au framework lors de la simulation

Déroulement d'une simulation

Une simulation est définie par une **période de simulation** avec une date de début et une date de fin.

Un **deltat par défaut** est défini pour les simulateurs.

Une **contrainte de planification** peut être appliquée aux simulateurs

- Aucune contrainte
- La planification des simulateurs est **vérifiée** et doit être égale au deltat par défaut
- La planification des simulateurs est **forcée** pour être égale au deltat par défaut (**risqué!**)

Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



Déroulement d'une simulation

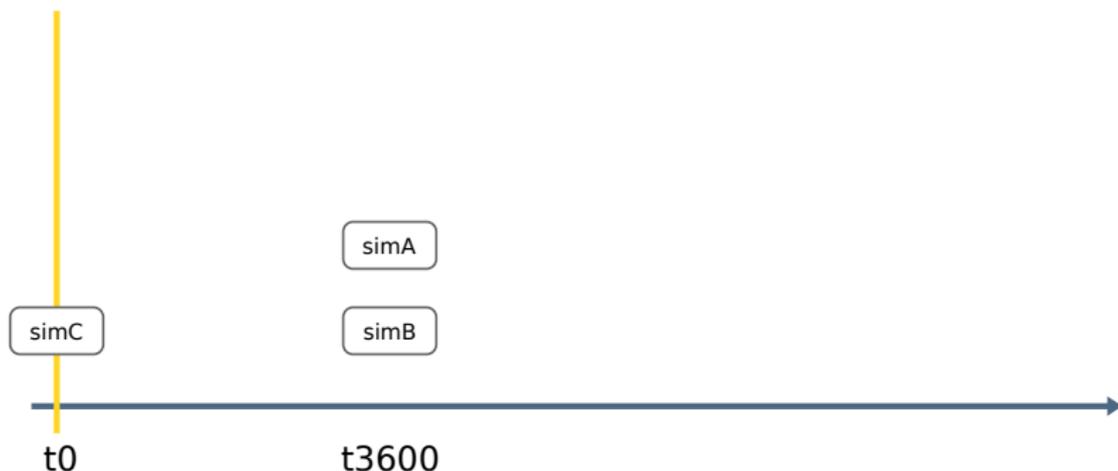
Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



Exécution du simulateur simA à t0

Déroulement d'une simulation

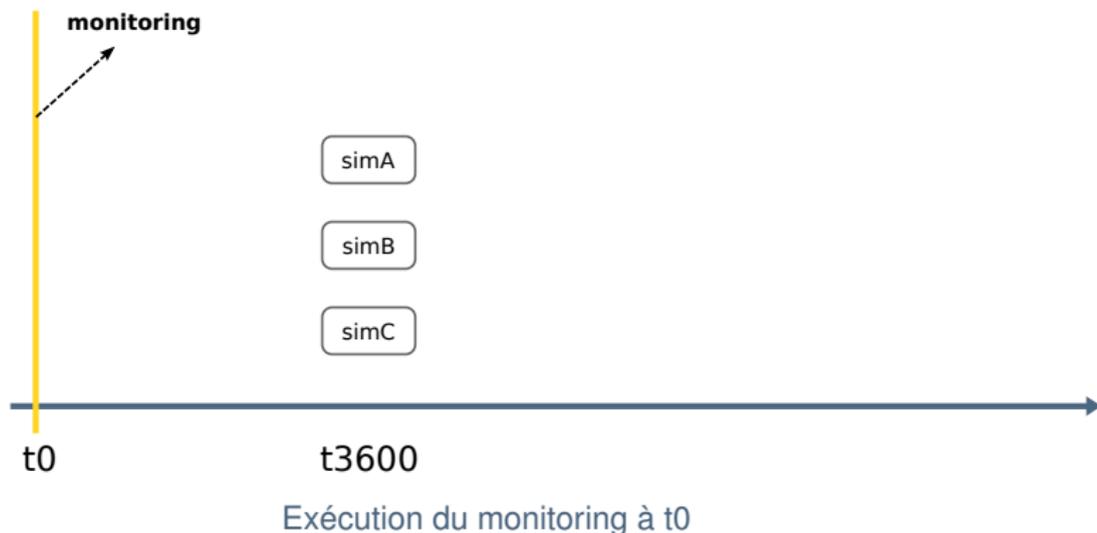
Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



Positionnement de simB pour sa prochaine exécution (**réagencement**)

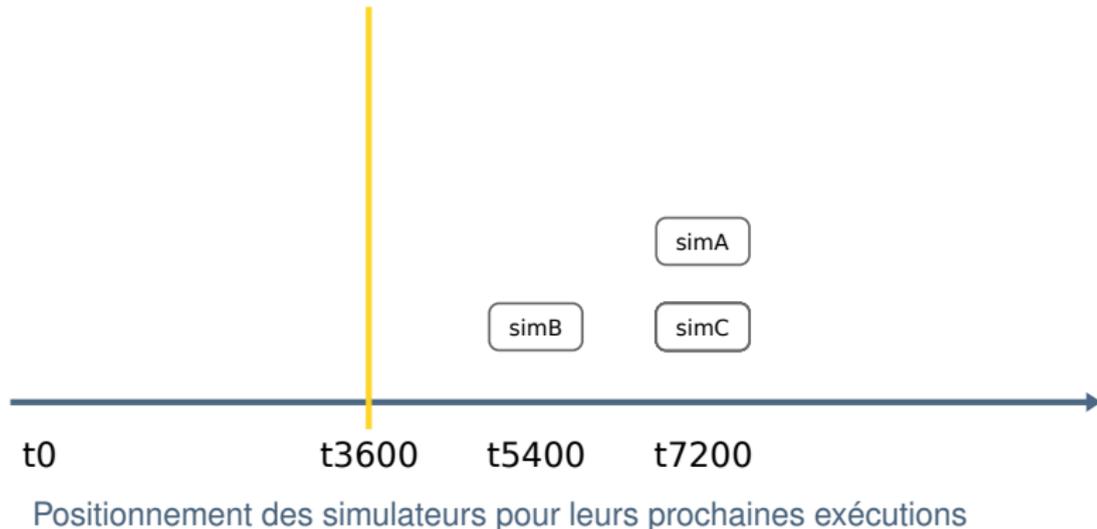
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



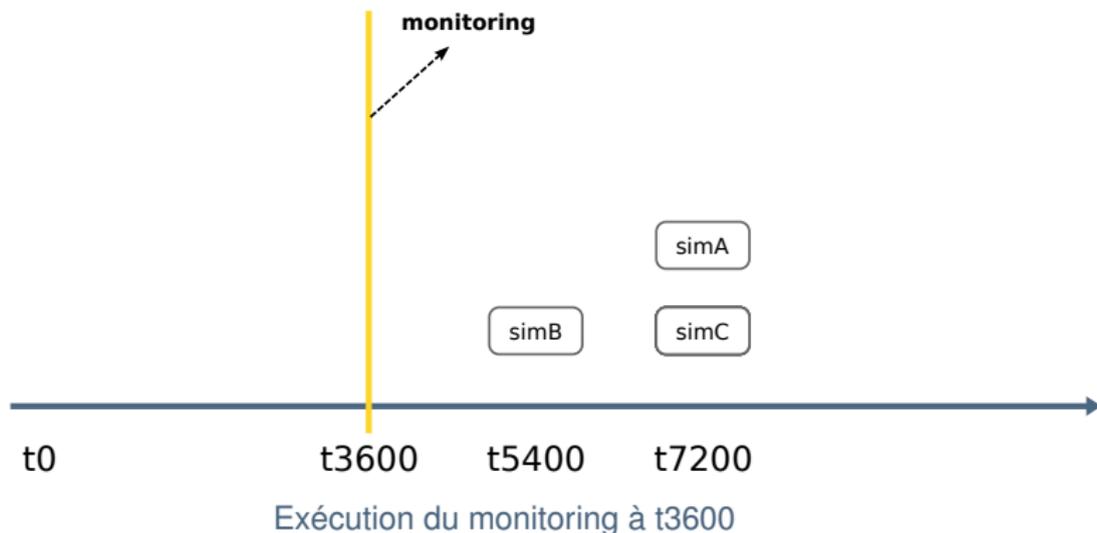
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00 (3600s)**, sans contrainte de planification.



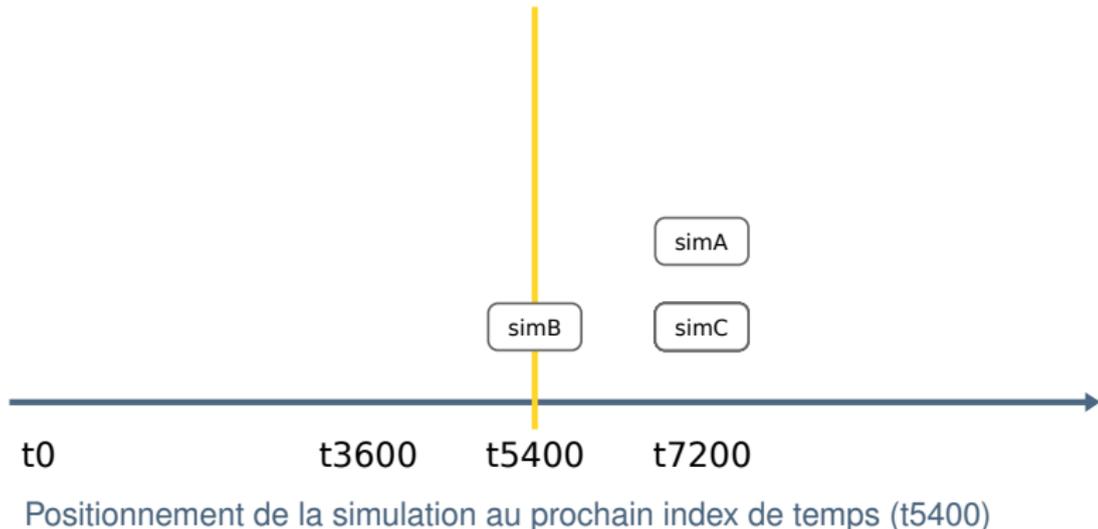
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



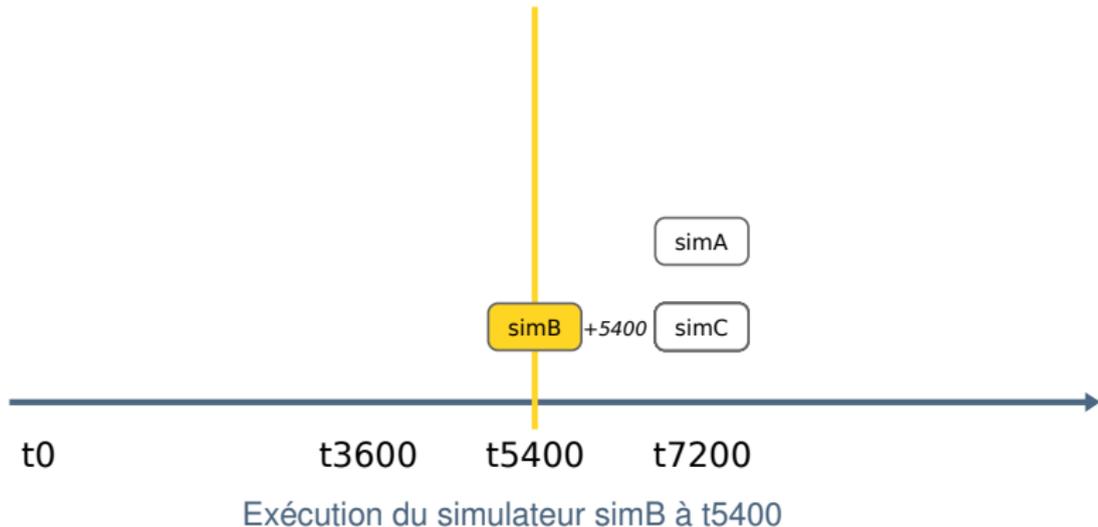
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



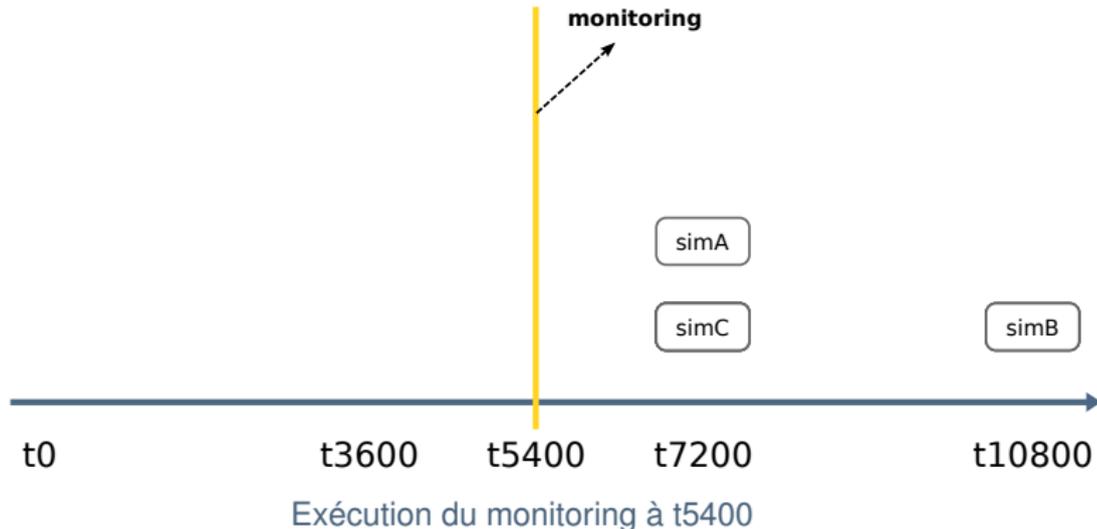
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



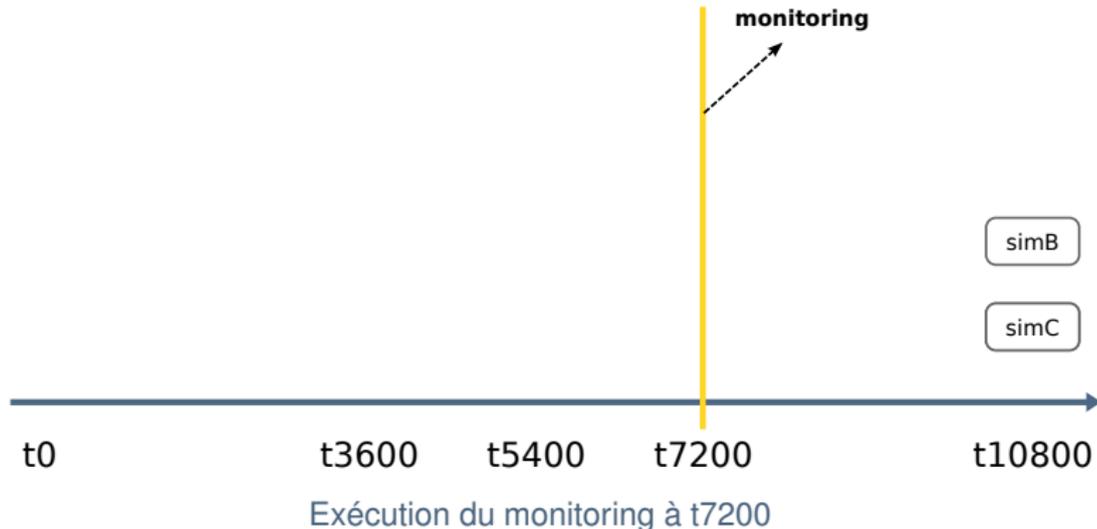
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



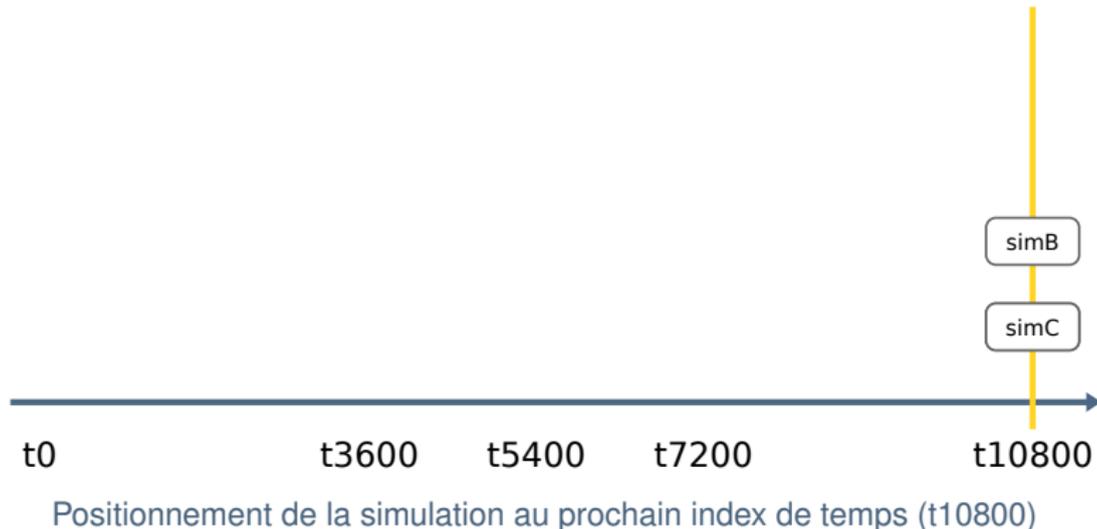
Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,
avec un deltat (dt) par défaut de **1h00** (3600s),
sans contrainte de planification.



Fonctionnalités complémentaires

- Vérification de cohérence des modèles construits
- Parallélisation automatisée des calculs des simulateurs (basée sur l'indépendance entre unités spatiales)
- Gestion des messages d'avertissements et d'erreurs
- Profilage de simulation
- Générateur de documentation scientifique à partir des simulateurs
- Librairie de traitement spatial (OpenFLUID-LandR)
- ...

Types de données OpenFLUID

Les données manipulées par OpenFLUID durant la simulation sont typées

Différents types de données sont définis

Types simples:

- BooleanValue : nombre **booléen** (0 ou 1)
- IntegerValue : nombre **entier** (1,18, 56325874, ...)
- DoubleValue : nombre **réel en double précision** (1.52, 0.000025, ...)
- StringValue : **chaîne de caractères**

Types composés:

- VectorValue : **vecteur** (1D) de nombres en **double précision**
- MatrixValue : **matrice** (2D) de nombres en **double précision**
- MapValue : **liste clé-valeur** de tout autre type de données

Variables

Les variables d'état sont **centrales dans le couplage** car **échangées entre simulateurs** tout au long de la simulation

- elles sont **attachées aux unités spatiales** via le graphe d'espace
- les **valeurs successives sont associées à un index de temps** dans la simulation (instant de production)
- les valeurs des variables sont ajoutées/modifiées par les simulateurs

Une variable d'état est présente sur toutes les unités spatiales d'une même classe d'unité

Les variables d'état peuvent être de n'importe quel type
OpenFLUID

Attributs spatiaux

Les attributs spatiaux sont des **données attachées aux unités spatiales**.

- *morphologie* : surface, longueur, largeur, profondeur, ...
- *propriétés physiques* : conductivité, teneur en eau, ...
- *coefficients* : manning, ...
- *paramètres descriptifs* : occupation du sol, ...

Ils sont stockés sous la forme de chaînes de caractères (StringValue), et peuvent être convertis vers n'importe quel autre type OpenFLUID

Ils ne peuvent **plus être modifiés** une fois la simulation lancée.

Paramètres de simulateurs et d'observateurs

Les **paramètres des simulateurs et des observateurs** sont des paramètres qui s'appliquent à un simulateur ou à un observateur en particulier.

Ils sont stockés sous la forme de chaînes de caractères (StringValue), et peuvent être convertis vers n'importe quel autre type OpenFLUID.

Il est possible de définir des **valeurs globales** pour des paramètres qui s'appliqueraient à plusieurs simulateurs ou observateurs.

Evènements discrets

Les évènements discrets surviennent à un instant précis sur une unité spatiale donnée

- *opérations culturales : labours, épandage de produits, ...*
- *changement d'occupation du sol, rotation de culture, ...*
- *aménagements : curage d'un fossé, ...*

Ils portent des informations qui peuvent être traitées par les simulateurs.

Ces informations sont stockées sous la forme de chaînes de caractères (StringValue), et peuvent être converties vers n'importe quel autre type OpenFLUID

Ils peuvent être connus *à priori* sous la forme d'un **calendrier**, ou **générés** par les simulateurs **au cours de la simulation**.

Banque de données intégrée : Datastore

Le datastore permet d'intégrer des **données non structurées**, en complément des données standards.

Actuellement, le datastore permet d'intégrer des données de type couche géographique **raster** ou **vecteur**.

A terme, intégration de nouvelles sources de données : liens BDs, chroniques, ...

Les données du datastore ne font pas partie des données de simulation, mais viennent en complément.

Générateurs de valeurs de variables

Les générateurs produisent des **valeurs pour des variables**
Ils sont intégrés au framework OpenFLUID, et ne nécessitent pas de simulateurs particuliers

- Générateur de valeurs **constantes** (fixed)
- Générateur de valeurs **aléatoires** (random)
- Générateur de valeurs **à partir d'un fichier**, avec **distribution spatiale** du contenu (inject)
- Générateur de valeurs **à partir d'un fichier**, avec **interpolation temporelle** et **distribution spatiale** du contenu (interp)

Cohérence du modèle couplé

Au travers de la signature de chaque simulateur, la **cohérence du modèle** couplé peut être vérifiée.

Cohérence spatio-temporelle des **variables**

- Une variable requise par un simulateur sur une classe d'unité u doit être produite par un autre simulateur sur l'ensemble des unités spatiales de la classe u
- Une même variable ne peut-être produite que par un seul simulateur

Cohérence spatiale des **attributs spatiaux**

- Un attribut requis sur une classe d'unité u doit être présent sur l'ensemble des unités spatiales de la classe u

La cohérence du modèle couplé est **automatiquement vérifiée par le framework OpenFLUID** à la mise en place du modèle couplé

Plan

- 1 Présentation générale
- 2 Fonctionnement
- 3 En pratique**
 - Utilisation
 - Exemples d'application
- 4 Jeux de données
- 5 Compléments

Application logicielle en ligne de commande

Exécution de simulations en ligne de commande à partir d'un jeu de données d'entrée

- utilisation simple ou en batch, sur cluster de calcul, ...

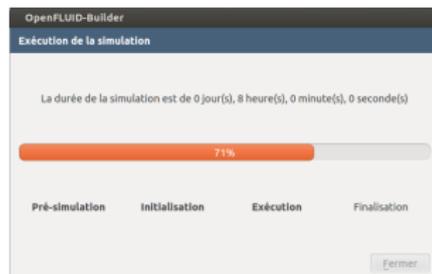
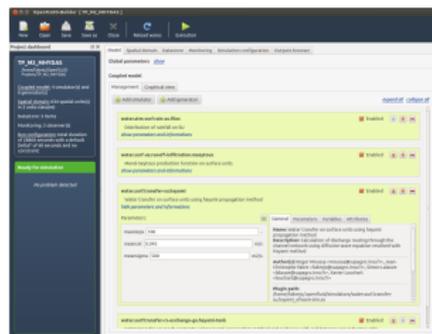
```
fabrejc@lisah-crampling: ~  
* Building spatial domain... [OK]  
* Building model... [OK]  
* Building monitoring... [OK]  
* Initializing parameters... [OK]  
* Preparing data... [OK]  
* Checking consistency... [OK]  
  
Spatial domain, 13 units :  
  - unitsA, 8 units  
  - unitsB, 5 units  
  
Simulation from 2001-01-01 00:03:00 to 2001-01-15 08:01:02  
Default DeltaT is 3597 seconds  
  
Size of buffers for variables is set to 22 (using dataset run configuration)  
  
**** Running simulation ****  
  
  Initialize... [OK]  
  
Progress      Real time      Status  
-----  
0.29%         2001-01-01 01:02:57 [OK]  
0.58%         2001-01-01 02:02:54 [OK]  
0.87%         2001-01-01 03:02:51 [OK]  
1.16%         2001-01-01 04:02:48 [OK]  
1.45%         2001-01-01 05:02:45 [OK]  
1.74%         2001-01-01 06:02:42 [OK]  
2.03%         2001-01-01 07:02:39 [OK]
```

Application logicielle graphique (OpenFLUID-Builder)

Interface graphique pour la construction, la paramétrisation, l'exécution de simulation et la visualisation de résultats

- Construction simple de modèles et de représentations du paysage
- Paramétrisation
- Simulation
- Accès aux résultats

Extensible par ajout de Builder-extensions (plugins) : import/export de données (fichiers, BDs, ...), visualisations, utilisations d'outils externes, ...



Package ROpenFLUID

Utilisation d'OpenFLUID depuis l'environnement GNU R

- Chargement de jeux de données
- Paramétrage (attributs spatiaux, paramètres de simulateurs, ...)
- Exécution de simulations
- Exploitation des résultats

⇒ Profiter des fonctionnalités d'exploration de simulations sous R

- Analyse de sensibilité
- Propagation d'incertitude
- ...

Lancement d'une simulation sous R

```
library('ROpenFLUID')  
ofsim = OpenFLUID.loadDataset('/path/to/dataset')  
OpenFLUID.runSimulation(ofsim)  
data = OpenFLUID.loadResult(ofsim, 'SU', 15, 'water.surf.Q.downstream')
```

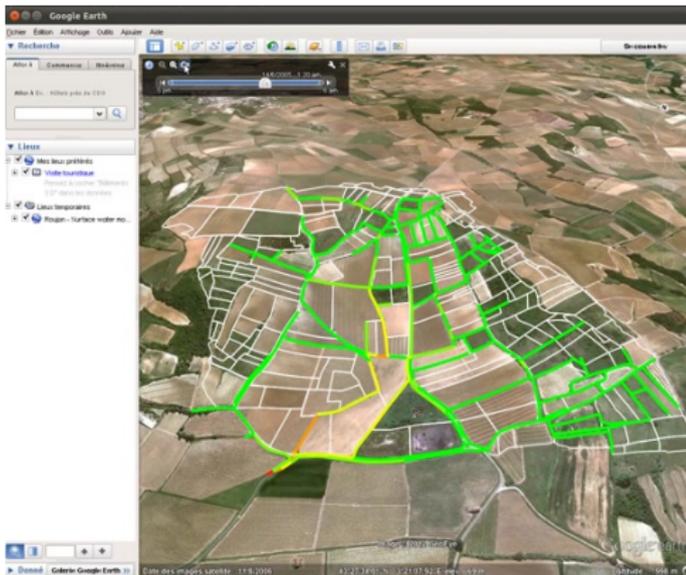
Interfaçage avec des outils externes

Visualisation, SIG:

- VisIt
- Google Earth
- Graphviz
- Qgis, GRASS
- ...

Analyse:

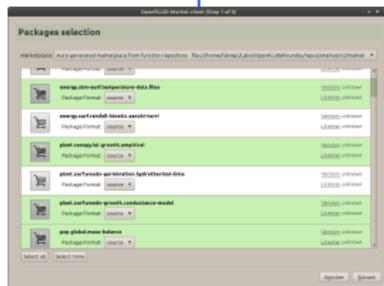
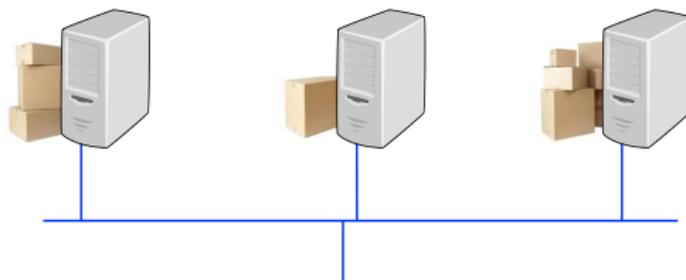
- GNU R
- PEST
- Octave
- ...



Via des observateurs
ou par post-traitement de données de simulation

OpenFLUID Market

Installation **automatisée** de simulateurs, d'observateurs et de jeux de données, mis à disposition sur un dépôt (Marketplace OpenFLUID) **local** ou via **internet** (HTTP)



Ressources OpenFLUID en ligne

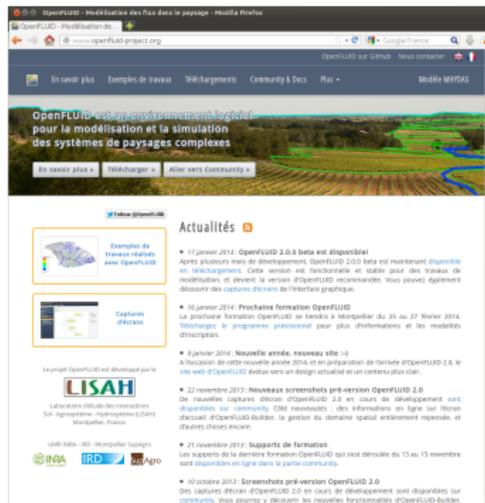
<http://www.openfluid-project.org/>

Une présentation générale

- Projet, applications, téléchargements

Un espace collaboratif (Community)

- Documentations utilisateurs et développeurs
- Guides, exemples de code, ...



@OpenFLUID



Ruissellement et propagation

Parcelle AW6 - Roujan (X. Louchart)



Parcelle AW6 - Roujan :

- 1200 m²
- 1070 unités spatiales

Simulation :

- sur 1 heure
- pas de temps : 10 s

Transfert en réseau hydrographique

Bassin versant de Roujan (*R. Moussa*)

Bassin versant de Roujan :

- 0.91 km²
- ~600 unités spatiales

Simulation :

- sur 7 heures
- pas de temps : 60 s

Plan

- 1 Présentation générale
- 2 Fonctionnement
- 3 En pratique
- 4 Jeux de données**
- 5 Compléments

Jeu de données d'entrée : organisation et formats

Un jeu de données d'entrée OpenFLUID est constitué d'un à plusieurs **fichiers d'entrées standardisés** au format **XML**, portant l'extension **.fluidx**.

Ce(s) fichier(s) doivent définir les sections suivantes

- `<domain>` : définition de la RNP, des paramètres et évènements distribués
- `<model>` : définition du modèle couplé
- `<monitoring>` : définition du monitoring
- `<run>` : configuration de la simulation

Il est possible pour les simulateurs et observateurs de gérer leurs propres fichiers d'entrée.

Jeu de données d'entrée : un seul ⇔ plusieurs fichiers

dataset.fluidx

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <domain>
    <!-- ici la definition et la parametrisation de l espace -->
  </domain>

  <model>
    <!-- ici la definition du modele -->
  </model>

  <monitoring>
    <!-- ici la definition du monitoring -->
  </monitoring>

  <run>
    <!-- ici la configuration de la simulation -->
  </run>
</openfluid>
```

Jeu de données d'entrée : un seul ⇔ plusieurs fichiers

model.fluidx

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <model>
    <!-- ici la definition
           du modele -->
  </model>
</openfluid>
```

domain.fluidx

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <domain>
    <!-- ici la definition
           et la parametrisation
           de l'espace -->
  </domain>
</openfluid>
```

run_monitoring.fluidx

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <run>
    <!-- ici la configuration de la simulation -->
  </run>
  <monitoring>
    <!-- ici la definition du monitoring -->
  </monitoring>
</openfluid>
```

Section <domain>

La section domain est composée de 3 sous-sections:

- <definition> : définition des unités spatiales qui composent la représentation du paysage, avec leur topologie (connectivité)
- <attributes> : attributs distribués, attachés à chaque unité spatiale
- <calendar> : évènements distribués, attachés à chaque unité spatiale

Section <domain>, sous-section <definition>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <domain>
    <definition>
      <unit class="SU" ID="1" pcsorder="1">
        <to class="SU" ID="5" />
      </unit>
      <unit class="SU" ID="5" pcsorder="2">
        <to class="RS" ID="9" />
      </unit>
      <unit class="RS" ID="9" pcsorder="1" />
    </definition>
  </domain>
</openfluid>
```

- déclaration d'une nouvelle unité : balise <unit> avec les attributs précisant la classe de l'unité, son ID et son ordre de traitement
- déclaration d'une connexion avec une autre unité : sous-balise <to> de la balise <unit>
- déclaration d'un lien de parenté avec une autre unité : sous-balise <childof> de la balise <unit>

Section <domain>, sous-section <definition>

The screenshot displays the OpenFLUID-Builder interface for a project named 'TP_M2_MHYDAS'. The interface is divided into several panels:

- Project dashboard:** Provides a summary of the simulation setup:
 - Home: /Users/jc/OpenFLUID-Projects/TP_M2_MHYDAS
 - Coupled model: 4 simulator(s) and 0 generator(s)
 - Spatial domain: 634 spatial units(s) in 3 units class(es)
 - Datatore: 3 items
 - Monitoring: 2 observer(s)
 - Run configuration: total duration of 28800 seconds with a default DeltaT of 60 seconds and no constraint
- Ready for simulation:** A green banner indicating 'No problem detected'.
- Model configuration:** Shows the 'Spatial domain' tab with options to 'Add units class'. Three classes are defined:
 - SU:** Show map style
 - RS:** Show map style
 - GU:** Show map style
- Structure Map:** A list of 18 units. Unit 6 is highlighted in orange. Above the list are buttons to 'Add unit in RS class' and 'Remove unit 6 from RS class'. A 'Process order' dropdown is set to 37.
- Connections:** A table showing connections between unit classes and IDs:

Connection type	Unit class	Unit ID
To	RS	330
To	GU	2
From	RS	9
From	SU	182
- Attributes Events:** A table listing attributes for each unit:

	Ks	height	length	nmanning	slope	width
1	0.003	0.6	11.63	0.2	0.006275	0.3
2	0.003	0.5	7.69	0.2	0.010271	1
3	0.003	0.58	11.24	0.2	0.009435	1.5
4	0.003	0.6	31.99	0.2	0.008409	0.3
5	0.003	0.3	1.69	0.2	0.29153	0.3
6	0.003	0.7	27.54	0.2	0.012636	1.6
7	0.003	0.3	9.35	0.2	0.226715	0.6
8	0.003	0.4	14.24	0.2	0.21442	1.1

Section <domain>, sous-section <attributes>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <domain>
    <attributes unitclass="SU" colorder="area;slope;flowdist" />
    1 4813.344 0.06265 14.366
    2 293.982 0.27129 18.468
  </attributes>
</domain>
</openfluid>
```

- données en colonnes entre balises <attributes> et </attributes> (1ère colonne = ID de l'unité)
- ordre des colonnes (après colonne ID) précisé au travers de l'attribut colorder

Section <domain>, sous-section <attributes>

The screenshot shows the OpenFLUID-Builder interface for the 'TP_M2_MHYDAS' project. The top toolbar includes icons for New, Open, Save, Save as, Close, Reload wares, and Exécution. The 'Project dashboard' on the left provides project details:

- TP_M2_MHYDAS**
Home: /home/fabrice/OpenFLUID-Projects/TP_M2_MHYDAS
- Coupled model: 4 simulator(s) and 0 generator(s)
- Spatial domain: 634 spatial units(s) in 3 units class(es)
- Datstore: 3 items
- Monitoring: 2 observer(s)
- Run configuration: total duration of 28800 seconds with a default DeltaT of 60 seconds and no constraint

A green bar indicates the project is **Ready for simulation** with *No problem detected*.

The main workspace shows the 'Spatial domain' tab with an 'Add units class' button and three unit classes:

- SU** (Show map style)
- RS** (Show map style)
- GU** (Show map style)

The 'Structure' tab displays a map of the domain with a grid of green units. One unit is highlighted in orange. The 'View' panel includes options for 'Show units IDs' and 'Fit to global extent', along with a zooming instruction: 'Use Ctrl key + mouse wheel for zooming'.

The 'Attributes' tab shows a table of data for the selected unit (231):

	Hc	Ks	area	betaMS	flowdist	nmaning	slope	thetain	thetain	thet
227	0.1	2.33E-05	1446.811	1.3	24	0.05	0.23333	0.3	0.2	0.02
228	0.1	8.76E-06	3826.909	1.3	75	0.05	0.016	0.3	0.2	0.02
229	0.1	9.12E-06	7253.936	1.3	113.9	0.05	0.02107	0.3	0.2	0.02
230	0.1	1.99E-06	4694.73	1.3	48	0.05	0.025	0.3	0.2	0.02
231	0.1	2.49E-06	10674...	1.3	64	0.05	0.03125	0.3	0.2	0.02
232	0.1	1.95E-06	6097.226	1.3	13.469	0.05	0.1084	0.3	0.2	0.02
233	0.1	1.87E-05	683.831	1.3	6.7	0.05	0.23881	0.3	0.2	0.02

Section <domain>, sous-section <calendar>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <calendar>
    <event unitclass="SU" unitID="1" date="1997-03-30_12:00:00">
      <info key="molecule" value="diuron"/>
      <info key="percent_area" value="100"/>
      <info key="rate_ha" value="1"/>
    </event>
  </calendar>
</openfluid>
```

- un évènement est déclaré au travers d'une balise <event>, et est associé à une unité spatiale et à une date.
- il porte de 0 à n informations déclarées par une balise <info>, chaque valeur (attribut value) étant identifiée par une clé (attribut key)

Section <model>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <model>
    <simulator ID="water.atm-surf.rain-su.files" />
    <simulator ID="water.surf-uz.runoff-infiltration.mseytoux" >
      <param name="resstep" value="0.000005" />
    </simulator>
  </model>
</openfluid>
```

- déclaration d'un simulateur : balise <simulator> avec attribut ID
- déclaration d'un paramètre de simulateur : balise <param> avec attributs name pour le nom du paramètre et value pour sa valeur

Section <model>

The screenshot displays the OpenFLUID-Builder interface for a project named 'TP_M2_MHYDAS'. The top toolbar includes icons for New, Open, Save, Save as, Close, Reload, and Execution. The main window is divided into several sections:

- Project dashboard:** Provides project details such as 'Coupled model: 4 simulator(s) and 0 generator(s)', 'Spatial domain: 634 spatial units(s) in 3 units class(es)', 'Datatore: 3 items', 'Monitoring: 2 observer(s)', and 'Run configuration: total duration of 28800 seconds with a default DeltaT of 60 seconds and no constraint'. A green status bar indicates 'Ready for simulation' with 'No problem detected'.
- Model Management:** Includes tabs for Spatial domain, Datatore, Monitoring, Simulation configuration, and Outputs browser. A 'Global parameters' link is available.
- Coupled model Graphical view:** A central diagram showing four interconnected simulators: 'water.atm-surf-rain-su.files', 'water.surf-transfer-su.hayami', 'water.surf-uz-runoff-infiltration.mseytoux', and 'water.surf-transfer-rs-exchange-gu.hayami-tank'. Each simulator has 'Req', 'Us', and 'Upd' ports. Interconnections are labeled with data flows: 'water.surf.H.runoff (SU)', 'water.surf.Q.downstream-su (SU)', 'water.atm-surf.H.rain (SU)', 'water.surf.Q.downstream-su (SU)', and 'water.surf.H.infiltration (SU)'.
- View Panel:** Contains options to 'Show variables', 'Fit to all items', and 'Reset to default'. A note suggests 'Use Ctrl key + mouse wheel for zooming'. Below are 'Export as PNG...' and 'Export as SVG...' buttons.

Section <monitoring>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <monitoring>
    <observer ID="export.vars.files.csv">
      <!-- params here -->
    </observer>
    <observer ID="export.vars.files.kml-anim">
      <!-- params here -->
    </observer>
  </monitoring>
</openfluid>
```

- configuration du monitoring
- exports de données de simulations, contrôles, ...

Section <monitoring>

OpenFLUID-Builder [TP_M2_MHYDAS]

New Open Save Save as Close Reload wares Execution

Project dashboard

TP_M2_MHYDAS
Home/Projects/OpenFLUID-Projects/TP_M2_MHYDAS

Coupled model: 4 simulator(s) and 0 generator(s)
Spatial domain: 634 spatial units(s) in 3 units class(es)
Datatore: 3 items
Monitoring: 2 observer(s)

Run configuration: total duration of 28800 seconds with a default DeltaT of 60 seconds and no constraint

Ready for simulation
No problem detected

Model Spatial domain Datatore Monitoring Simulation configuration Outputs browser

Observers list

Add observer [expand all](#) [collapse all](#)

export.vars.files.csv Enabled

Exports simulation variables to CSV files
[show parameters and informations](#)

export.vars.plot.gnuplot Enabled

Plots simulation variables using GNUplot
[hide parameters and informations](#)

Parameters:

graph.g4.title Pluie SU133

graph.g4.ylabel Pluie

output sorties.pdf

persistent 0

serie.s1.label water.surf.Q.downstream-rs372

serie.s1.style lines

serie.s1.unitID 372

General

Name: Plots simulation variables using GNUplot
Description: This observer plots simulation variables using GNUplot. It uses series of values (simulated or from existing data files) that can be plotted in many graphs.
Plotted variables can only be scalar variables (boolean, integer, double).
Parameters for declaring series can be:
serie-<serie-name>-var: the name of the variable to plot (e.g. var.foo)
serie-<serie-name>-unitclass: the unit class (e.g. TU)
serie-<serie-name>-unitid: the unit id (e.g. 17)
serie-<serie-name>-sourcefile: the file to plot (e.g. measured_data.d)
simulation variables
serie-<serie-name>-style: the GNUplot style to use to plot this serie
serie-<serie-name>-label: the label to use for this serie (e.g. foo) inst...

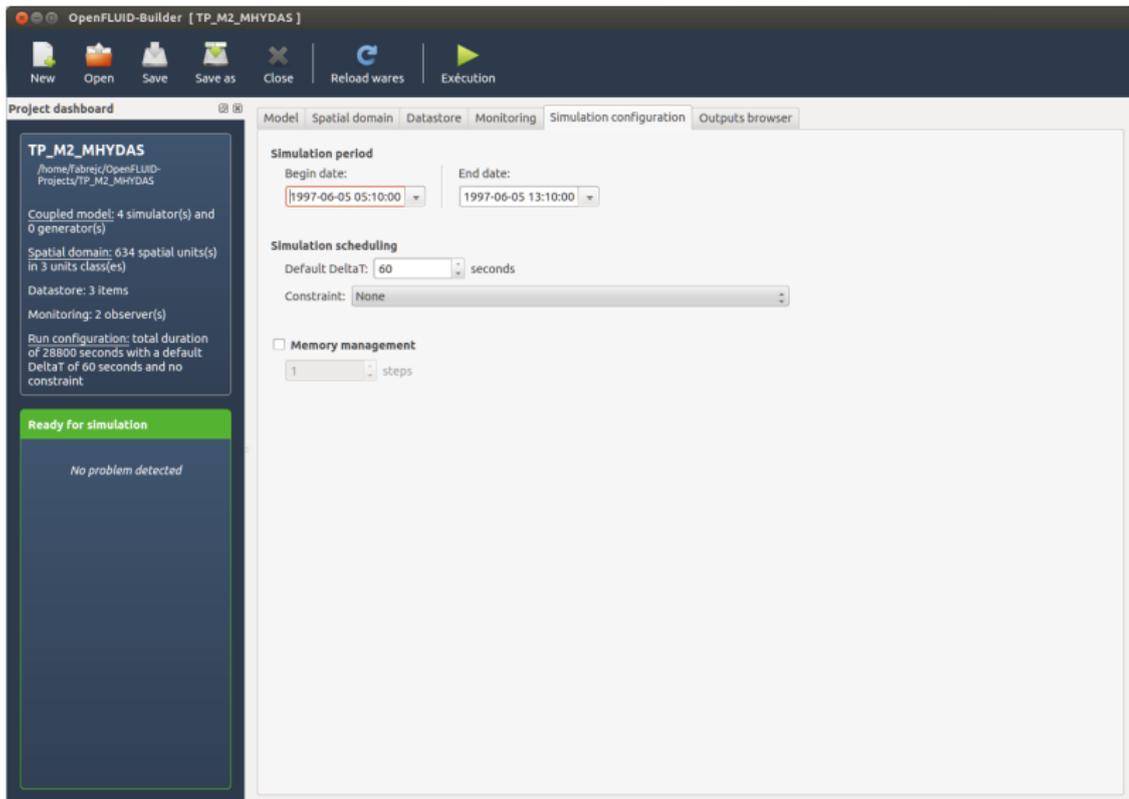
Section <run>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <run>
    <scheduling deltat="60" constraint="none" />
    <period begin="1997-03-29_03:00:18" end="1997-04-01_16:23:21" />
    <!-- <valuesbuffer size="10" /> -->
  </run>
</openfluid>
```

- la configuration de la simulation comprend obligatoirement une période de simulation (<period>) et un pas de temps d'échange (<deltat>)
- la gestion de la mémoire peut être affinée au travers de la balise <valuesbuffer> (optionnelle)

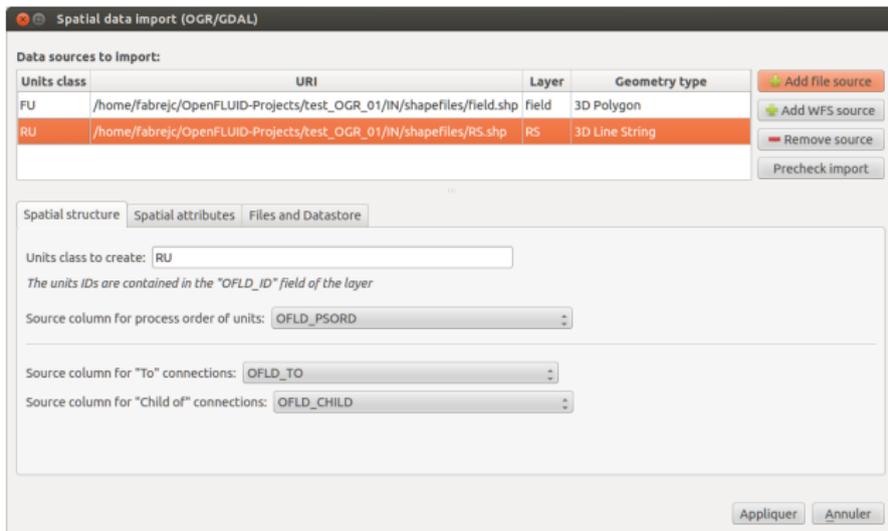
Section <run>



Builder-extension: Import de données spatiales

Extension à OpenFLUID-Builder permettant d'importer une représentation du paysage issue d'une source de données SIG:

- depuis des fichiers locaux (shapefiles, MapInfo, ...)
- depuis des BDs cartographiques (via flux WFS)



Builder-extension: Visualisation du graphe d'espace

Extension à OpenFLUID-Builder permettant de **visualiser la structure de l'espace** sous la forme d'un graphe, mettant en lumière les **connexions spatiales**.

OpenFLUID-Builder [pw6_test]

Nouveau Ouvrir Recharger Enregistrer Enregistrer sous Fermer Exécution

Tableau de projet

pw6_test
Home/Fabric/OpenFLUID-Projects/pw6_test

Modèle couplé: 3 simulateur(s) et 0 générateur(s)

Domaine spatial: 29 unité(s) spatiales dans 2 class(es) d'unités

Datatore: 2 élément(s)

Monitoring: 1 observateur(s)

Configuration de simulation: durée totale de 110256 seconds avec un Delta t par défaut de 60 seconds et aucune contrainte

Prêt à simuler, avec des avertissements

- Paramètres du modèle
 - Le paramètre s pour le générateur training ou prod n'est pas défini

Modèle | Domaine spatial | Datatore | Monitoring | Configuration de simulation | Explorateur des sorties | Spatial graph viewer (GraphViz)

Graph layout
Direction: Top to Bottom

View
Fit view to full graph

Use Ctrl key + mouse wheel for zooming

Export
Export as PNG...
Export as SVG...

The graph visualization shows a hierarchical structure of spatial units (SU) and relationships (RS). The units are represented by ovals, and the relationships are represented by arrows. The graph starts with SU#9 and SU#11 at the top, leading down to various RS nodes, and finally converging to RS#1 at the bottom. The structure is as follows:

```
graph TD
    SU9((SU#9)) --> RS9((RS#9))
    SU7((SU#7)) --> RS9
    SU11((SU#11)) --> RS13((RS#13))
    SU13((SU#13)) --> RS13
    SU14((SU#14)) --> RS5((RS#5))
    SU15((SU#15)) --> RS7((RS#7))
    RS9 --> RS11((RS#11))
    RS10((RS#10)) --> RS11
    RS5 --> RS4((RS#4))
    RS7 --> RS4
    SU12((SU#12)) --> RS8((RS#8))
    RS11 --> RS12((RS#12))
    RS3((RS#3)) --> RS8
    RS4 --> RS8
    SU10((SU#10)) --> RS6((RS#6))
    SU5((SU#5)) --> RS6
    RS12 --> RS6
    SU1((SU#1)) --> RS1((RS#1))
    RS8 --> RS1
    RS2((RS#2)) --> RS1
    SU6((SU#6)) --> RS2
    SU10((SU#10)) --> RS2
    SU4((SU#4)) --> RS2
    RS6 --> RS13
    RS2 --> RS14((RS#14))
    RS14 --> RS1
```

Plan

- 1 Présentation générale
- 2 Fonctionnement
- 3 En pratique
- 4 Jeux de données
- 5 Compléments**
 - **Projet**

Environnement

Répertoires par défaut (sous Unix/Linux) de l'utilisateur

- OpenFLUID personnel :
/home/username/.openfluid
- Données en entrée :
/home/username/.openfluid/OPENFLUID.IN
- Données résultats :
/home/username/.openfluid/OPENFLUID.OUT
- Simulateurs :
/home/username/.openfluid/simulators
- Observateurs :
/home/username/.openfluid/observers

Principales options de la ligne de commande

Option	Description
-i	chemin du jeu de données d'entrée
-o	chemin de sauvegarde des résultats
-w	chemin de projet OpenFLUID
-p	chemins de recherche de simulateurs
-n	chemins de recherche d'observateurs
-q	affichage silencieux pendant la simulation
-v	affichage détaillé pendant la simulation
-k	activation du profiling de simulation
-f	liste des simulateurs disponibles
-r	rapport détaillé sur les simulateurs disponibles
-e	liste des observateurs disponibles
-l	rapport détaillé sur les observateurs disponibles
--version	retourne la version d'OpenFLUID

- Les options de la ligne de commande peuvent être combinées

Buddies

Les **buddies** ou "compagnons" sont des outils périphériques, accompagnant OpenFLUID dans sa mise en oeuvre

4 compagnons sont intégrés à OpenFLUID:

- sim2doc : génération automatique de documentation scientifique pour les simulateurs
- newsim : générateur de code source de simulateurs
- newdata : générateur de jeux de données vides formatés

Utilisation en ligne de commande `openfluid` avec les options `--buddy`, `--buddyopts`, `--buddyhelp`

Buddy sim2doc

Utilisation du code source des simulateurs pour en extraire les informations nécessaires

- Lecture de la signature du simulateur
- Insertion du contenu \LaTeX placé en commentaires du code source entre les balises `<sim2doc>` et `</sim2doc>` (optionnel)

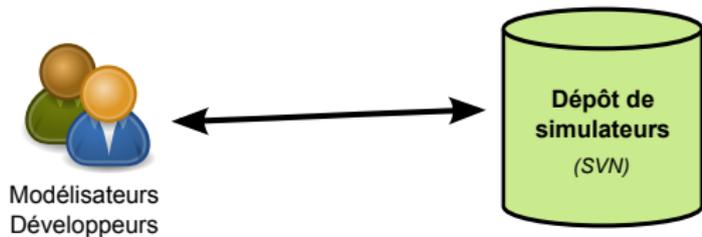
Génère un fichier \LaTeX , pouvant être converti à la volée en PDF ou en HTML

Exemple

```
openfluid --buddy sim2doc --buddyopts \  
  inputcpp=MySim.cpp,outputdir=./doc,PDF=1
```

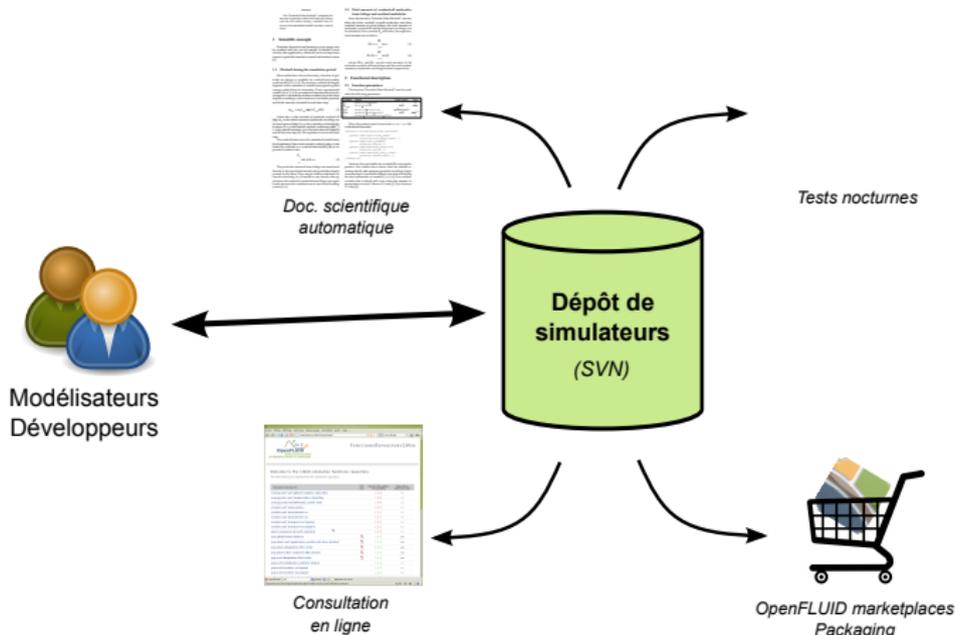
Dépôt de simulateurs

Structure pour la **capitalisation** et le **partage** de simulateurs développés au sein d'un groupe de travail



Dépôt de simulateurs

Structure pour la **capitalisation** et le **partage** de simulateurs développés au sein d'un groupe de travail



Dépôt de simulateurs du LISAH (intranet)

Capitalisation des développements, packaging, tests:

- Structure normalisée, avec outil de gestion du code source
<https://www.umr-lisah.fr/svn/openfluid-functions/>

Navigation dans le contenu:

- Accès à l'historique des développements, aux documentations scientifiques, ...
<http://www.umr-lisah.fr/repos2web/>

Tableau de suivi des tests

- Suite d'outils CMake/CTest/CDash
<http://www.umr-lisah.fr/cdash/>

Quelques éléments de projet

Ingénierie logicielle

- Développement majoritairement en C++
- **Multiplateformes** Linux/Unix, Windows, (Mac OSX)
- **Librairies support:** Boost, Qt, GDAL/OGR, GEOS
- **Environnement de développement/tests/packaging/doc:**
Eclipse, CMake, GCC, CTest, CPack, Doxygen, L^AT_EX
- **Code source géré sous Git,**
centralisé et **disponible sur GitHub** <https://github.com/OpenFLUID>

Cadre juridique

- Dépôt APP, dépôt de la marque OpenFLUID®
- Licence libre (GPLv3) ou sur mesure (commerciale, propriétaire, ...)

Perspectives de développement

- Introduction de **données attachées aux connections** entre unités spatiales
- Validation du module **PyOpenFLUID** permettant d'utiliser OpenFLUID depuis le langage Python
- Consolidation de la **bibliothèque de traitement spatial OpenFLUID-LandR**
- **Accès au datastore** depuis les simulateurs et observateurs
- Amélioration de l'interface utilisateur pour le **paramétrage des observateurs**
- **Evolution du datastore** : liens avec BDs, fichiers de données mesurées, ...
- Amélioration de l'intégration d'**approches orientées raster**
- ...

