



OpenFLUID

Software Environment
for Modelling Fluxes in Landscapes

Présentation générale

Utilisation d'OpenFLUID

JC. Fabre, M. Rabotin, D. Crevoisier, X. Louchart, F. Colin, R.
Moussa, A. Libres

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

Modélisation intégrative du paysage

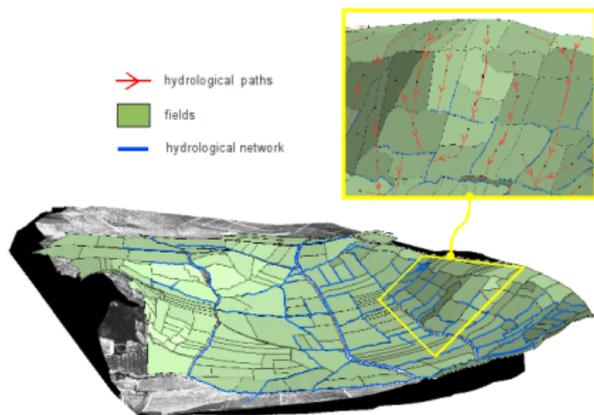
- Modélisation et simulation de processus spatialisés, interagissant fortement dans le temps et l'espace
i.e. flux de matière (eau, terre, énergie, ...), activités humaines, ...
- Approche pluri-disciplinaire
- Approche équilibrée entre processus et espace
- Approche complexe

⇒ Nécessité d'outils-support de la modélisation intégrative du paysage

Représentation Numérique du Paysage (RNP)

Représentation sous la forme d'unités spatiales connectées, issues

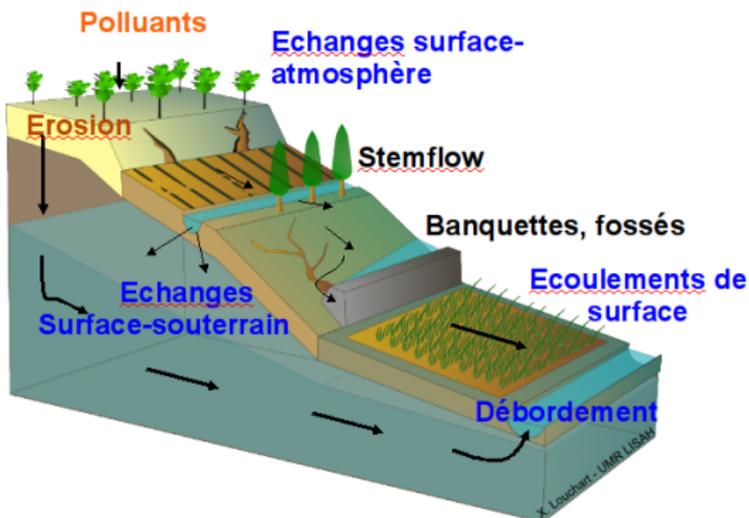
- du croisement de couches d'information spatiales (géométrie, relief, propriétés pertinentes, ...)
- des contraintes liées aux modèles qui vont être appliqués.



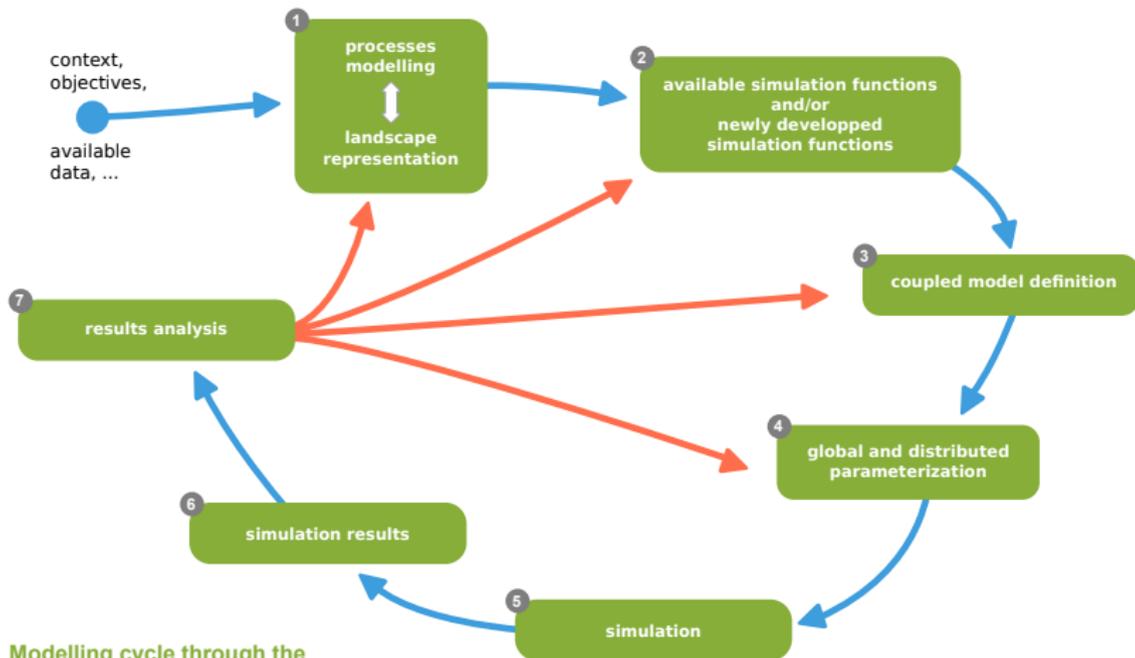
- Représentation finale sous forme d'un graphe orienté
- Classification des unités (surface, fossés, souterrain, ...)

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



Modelling cycle through the OpenFLUID framework

Objectifs d'une plateforme "paysage"?

Un environnement logiciel pour la modélisation intégrative

- Ouvrir à la pluri-disciplinarité, proposant des **formalismes "aussi génériques que possible"**
- Construire des **modèles adaptés** aux objectifs donnés

Un outil pour la recherche

- Utiliser et réutiliser des modèles
- Partager un **environnement logiciel unique**
- Définir des **standards** pour le développement et les données
- Etre dans la lignée Open-Source

Une démarche d'ingénierie logicielle

- Profiter des **technologies** récentes ou émergentes
- Améliorer la maintenance, l'évolutivité, la qualité
- Introduire des **méthodes de projet adaptées** (AGILE)

Un lien pour une communauté

- Partager des **compétences**, des **connaissances**, des **réalisations**

Objectifs d'une plateforme "paysage"?

Un environnement logiciel pour la modélisation intégrative

- Ouvrir à la pluri-disciplinarité, proposant des **formalismes "aussi génériques que possible"**
- Construire des **modèles adaptés** aux objectifs donnés

Un outil pour la recherche

- **Utiliser** et **réutiliser** des modèles
- Partager un **environnement logiciel unique**
- Définir des **standards** pour le développement et les données
- Etre dans la lignée Open-Source

Une démarche d'ingénierie logicielle

- Profiter des **technologies** récentes ou émergentes
- Améliorer la maintenance, l'évolutivité, la qualité
- Introduire des **méthodes de projet adaptées** (AGILE)

Un lien pour une communauté

- Partager des **compétences**, des **connaissances**, des **réalisations**

Objectifs d'une plateforme "paysage"?

Un environnement logiciel pour la modélisation intégrative

- Ouvrir à la pluri-disciplinarité, proposant des **formalismes "aussi génériques que possible"**
- Construire des **modèles adaptés** aux objectifs donnés

Un outil pour la recherche

- **Utiliser** et **réutiliser** des modèles
- Partager un **environnement logiciel unique**
- Définir des **standards** pour le développement et les données
- Etre dans la lignée Open-Source

Une démarche d'ingénierie logicielle

- Profiter des **technologies** récentes ou émergentes
- Améliorer la maintenance, l'évolutivité, la qualité
- Introduire des **méthodes de projet adaptées** (AGILE)

Un lien pour une communauté

- Partager des **compétences**, des **connaissances**, des **réalisations**

Objectifs d'une plateforme "paysage"?

Un environnement logiciel pour la modélisation intégrative

- Ouvrir à la pluri-disciplinarité, proposant des **formalismes "aussi génériques que possible"**
- Construire des **modèles adaptés** aux objectifs donnés

Un outil pour la recherche

- **Utiliser** et **réutiliser** des modèles
- Partager un **environnement logiciel unique**
- Définir des **standards** pour le développement et les données
- Etre dans la lignée Open-Source

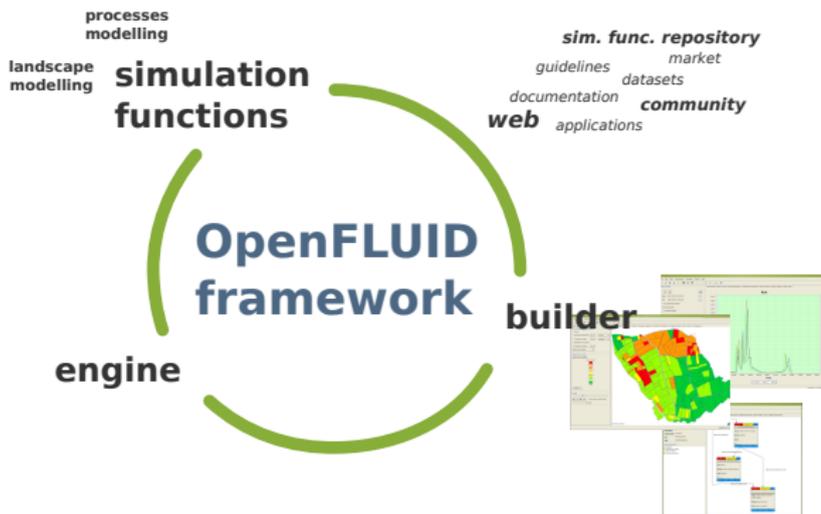
Une démarche d'ingénierie logicielle

- Profiter des **technologies** récentes ou émergentes
- Améliorer la maintenance, l'évolutivité, la qualité
- Introduire des **méthodes de projet adaptées** (AGILE)

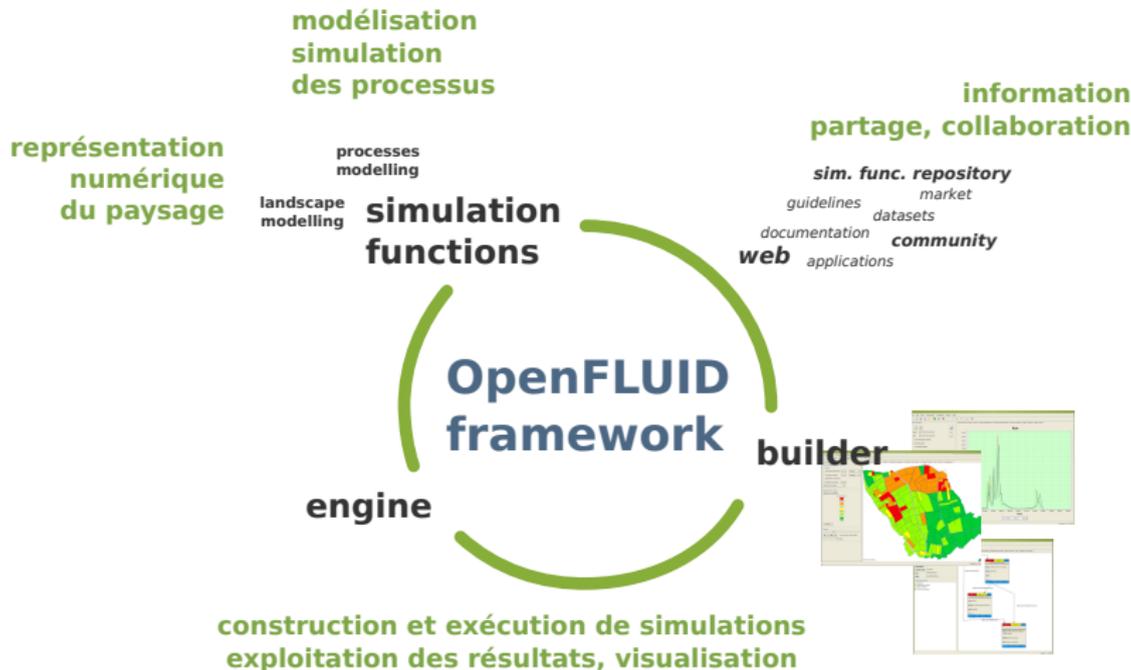
Un lien pour une communauté

- **Partager** des **compétences**, des **connaissances**, des **réalisations**

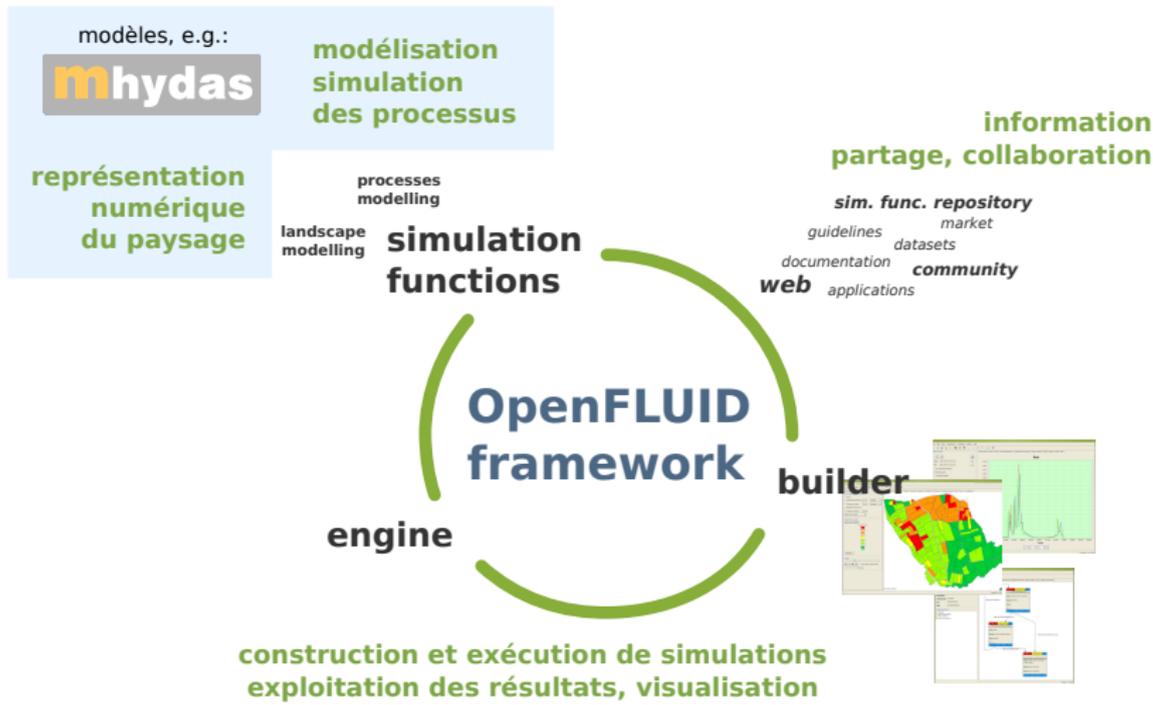
La plateforme OpenFLUID



La plateforme OpenFLUID



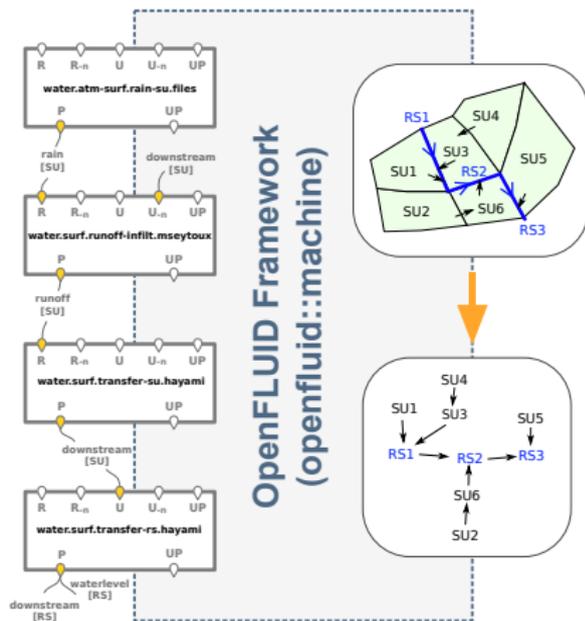
La plateforme OpenFLUID



Framework OpenFLUID

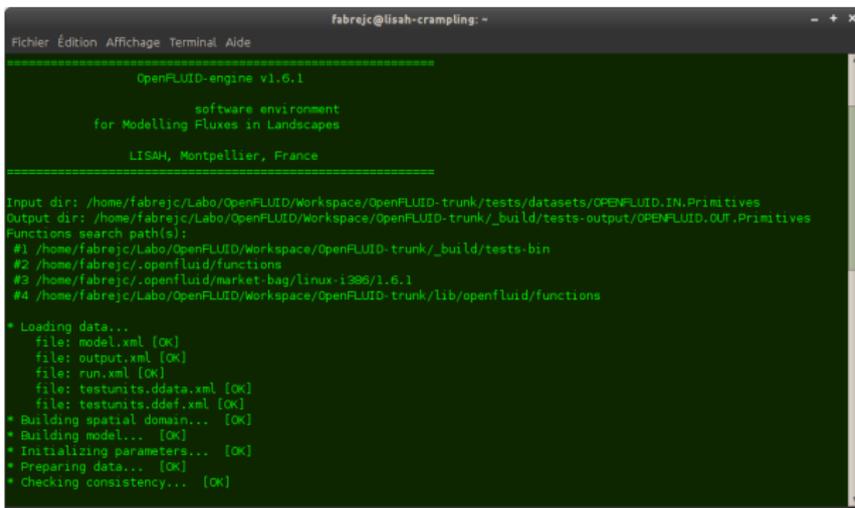
Structure de construction et couplage de modèles, implémentant le cadre de modélisation défini

- Représentation de l'espace sous la forme de **graphes connexes d'unités spatiales**, avec multi-échelle possible
- Modèles **branchés dynamiquement** via mécanisme de plug-ins -**fonctions de simulations**- pour former des modèles couplés
- Une fonction de simulation intègre un ou plusieurs processus spatiaux (transferts, évolutions)
- Echanges de **variables** et **événements** dans l'espace et dans le temps



Application logicielle OpenFLUID-Engine

Exécution de simulation 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: ~
Fichier  Édition  Affichage  Terminal  Aide
-----
OpenFLUID-engine v1.6.1

software environment
for Modelling Fluxes in Landscapes

LISAH, Montpellier, France
-----

Input dir: /home/fabrejc/Labo/OpenFLUID/workspace/OpenFLUID-trunk/tests/datasets/OPENFLUID.IN.Primitives
Output dir: /home/fabrejc/Labo/OpenFLUID/workspace/OpenFLUID-trunk/_build/tests-output/OPENFLUID.OUT.Primitives
Functions search path(s):
#1 /home/fabrejc/Labo/OpenFLUID/workspace/OpenFLUID-trunk/_build/tests-bin
#2 /home/fabrejc/.openfluid/functions
#3 /home/fabrejc/.openfluid/market-bag/linux-1386/1.6.1
#4 /home/fabrejc/Labo/OpenFLUID/workspace/OpenFLUID-trunk/lib/openfluid/functions

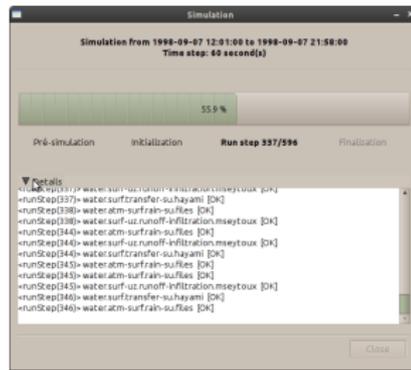
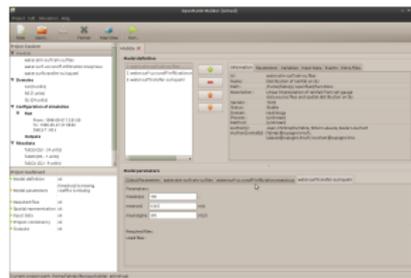
* Loading data...
  file: model.xml [OK]
  file: output.xml [OK]
  file: run.xml [OK]
  file: testunits_data.xml [OK]
  file: testunits_def.xml [OK]
* Building spatial domain... [OK]
* Building model... [OK]
* Initializing parameters... [OK]
* Preparing data... [OK]
* Checking consistency... [OK]
```

Application logicielle OpenFLUID-Builder

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

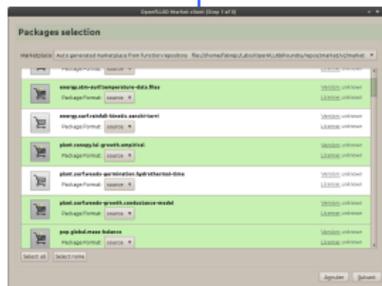
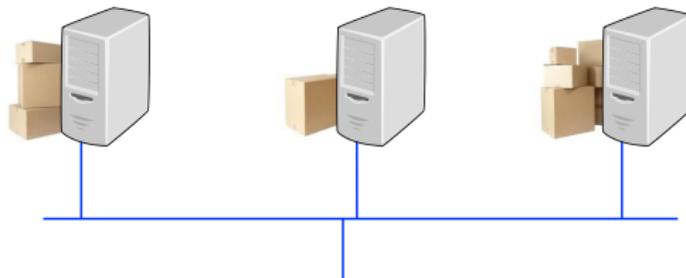
- Construction simple de modèles et de représentations du paysage
- Paramétrisation
- Simulation
- Traitement des résultats

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



OpenFLUID Market

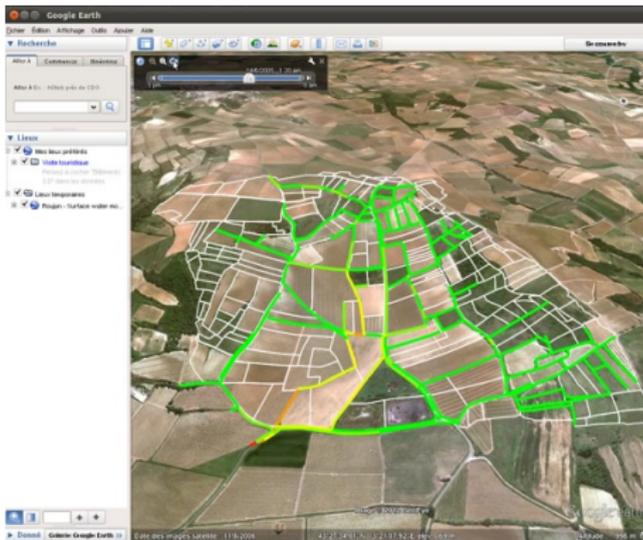
Installation **automatisée** de modèles à partir de packages **sources** ou **binaires**, mis à disposition sur une Marketplace OpenFLUID **locale** ou via **internet** (HTTP)



Interfaçage avec des outils externes

Visualisation, SIG:

- VisIt
- Goole Earth Hot!
- Graphviz
- Qgis, GRASS
- ...



Analyse:

- PEST
- Octave
- ...

Ressources OpenFLUID en ligne

<http://www.umr-lisah.fr/openfluid/>

Une présentation générale

- Projet
- Applications
- Téléchargements

Un espace collaboratif (Community)

- Documentations utilisateur, développeur
- Guides, conventions
- Exemples de codes
- ...



Plan

- 1 Présentation générale
- 2 Exemples d'applications**
 - Echelle parcellaire
 - Echelle du bassin versant
- 3 En pratique
- 4 Jeux de données
- 5 Compléments

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

Evolution des teneurs en pesticides dans le sol

Bassin versant de Neffiès (X. Louchart)



Bassin versant de Neffiès :

- 4 km²
- ~800 unités spatiales

Simulation :

- sur 3 ans, selon pratiques de pulvérisation observées
- 10 insecticides et fongicides
- pas de temps : 24 H

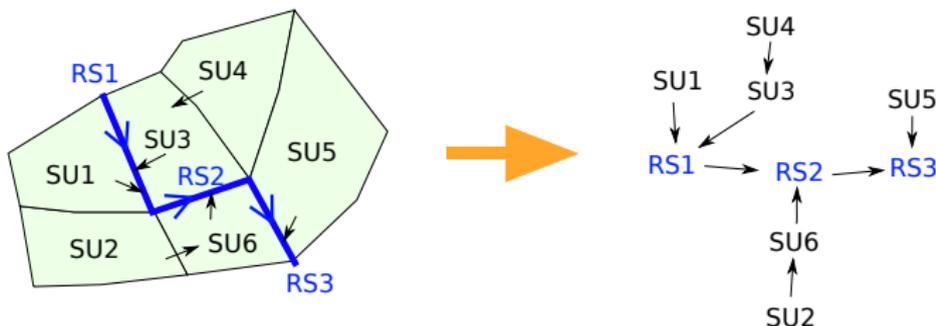
Plan

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

Représentation numérique du paysage (RNP)

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

- Les noeuds sont les unités spatiales représentant le paysage
- Les arcs orientés sont les connexions entre les unités
- Chaque noeud porte des propriétés propres à l'unité qu'il représente



Simulation de processus

Les processus sont représentés par des **fonctions de simulation** qui utilisent et produisent des valeurs de **variables d'état** sur les unités spatiales.

Elles manipulent également des **données d'entrée** distribuées, des **paramètres de fonctions**, des **événements discrets**.

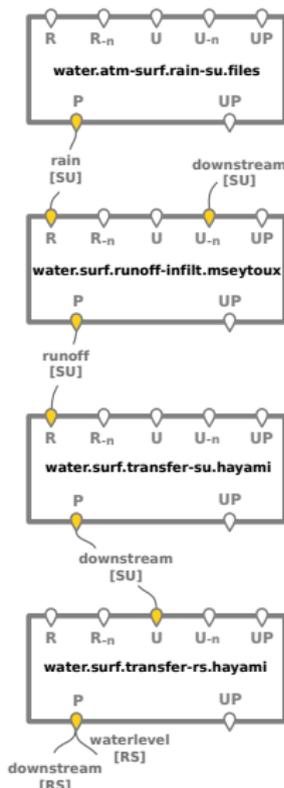
- Chaque fonction de simulation déclare les variables utilisées et produites, ses données d'entrée et ses paramètres de fonction
- Les fonctions de simulation s'engagent à respecter ce qu'elles déclarent

Modèle couplé

Les modèles sont définis par une **liste ordonnée de fonctions de simulation**

- Les fonctions de simulation peuvent avoir **leur propre pas de temps de calcul**
- Le **pas de temps d'échange** entre fonctions de simulation est **fixe**

Un modèle couplé doit être **cohérent** pour être simulé



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)
- DoubleValue : nombre **réel en double précision** (1.52, 0.000025, ...)
- IntegerValue : nombre **entier** (1,18, 56325874, ...)
- 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

Plan

- 1 Présentation générale
- 2 Exemples d'applications
- 3 En pratique
- 4 Jeux de données**
 - Formats
 - Simulations
- 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

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

Il est possible pour les fonctions de simulation 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>

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

  <domain>
    <!-- ici la definition et la parametrisation de l espace -->
  </domain>

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

  <output>
    <!-- ici la configuration des sorties -->
  </output>
</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_output.fluidx

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <run>
    <!-- ici la configuration de la simulation -->
  </run>
  <output>
    <!-- ici la configuration des sorties -->
  </output>
</openfluid>
```

Section <model>

Exemple

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

- déclaration d'une fonction : balise <function> avec attribut fileID
- déclaration d'un paramètre de fonction : balise <param> avec attributs name pour le nom du paramètre et value pour sa valeur

Section <model>

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)
- <inputdata> : paramètres 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 shows the OpenFLUID-Builder [virtual] interface. The Project Explorer on the left shows the following structure:

- Model
 - water.atm-surf.rain-su.files
 - water.surf-uz.runoff-infiltration.mseytout
 - water.surf.transfer-su.hayami
- Spatial domain
 - CU (9 units)
 - RS (7 units)
 - SU (24 units)
- Simulation
 - Run
 - Outputs
 - Results

The Simulation section is expanded to show the Run subsection. The Project Dashboard at the bottom left shows the following status:

- Model definition: ok
- Model parameters: threshold is missing, coeffks is missing
- Required files: ok
- Spatial representation: ok
- Input data: ok
- Project consistency: ok
- Outputs: ok

The main workspace displays the Unit Classes table for the Spatial domain:

Unit Classes	ID	Process Order
CU	2	1
RS	4	1
SU	6	1
	7	1
	9	1
	1	2
	3	2
	5	2
	8	2

Current project path: /home/fabrejc/Bureau/builder_prj/virtual

Section <domain>, sous-section <inputdata>

Exemple

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

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

Section <domain>, sous-section <inputdata>

The screenshot displays the OpenFLUID-Builder software interface. The main window is titled 'OpenFLUID-Builder [virtual]' and features a menu bar (Project, Edit, Simulation, Help) and a toolbar with icons for New, Open, Save, Close, Map View, and Run. The interface is divided into several panels:

- Project Explorer:** Shows a tree view of the project structure, including Model, Domain, Simulation, and Results.
- Project Dashboard:** A summary table of project status:

Category	Status
Model definition	ok
Model parameters	threshold is missing coeffks is missing
Required files	ok
Spatial representation	ok
Input data	ok
Project consistency	ok
Outputs	ok

- Inputdata Table:** A table with columns: Unit ID, area, betaMS, flowdist, hc, ks, nmanning, slope, thetains, thet. The table contains 24 rows of data.

Unit ID	area	betaMS	flowdist	hc	ks	nmanning	slope	thetains	thet
24	166.2	1.3	55	0.1	0.000001	0.05	0.01	0.2	0.3
23	134.2	1.3	45	0.1	0.000001	0.05	0.01	0.2	0.3
22	369.9	1.3	170	0.1	0.000001	0.05	0.04	0.2	0.3
21	288.9	1.3	128	0.1	0.000001	0.05	0.06	0.2	0.3
20	259.8	1.3	105	0.1	0.000001	0.05	0.04	0.2	0.3
19	203.5	1.3	224	0.1	0.000001	0.05	0.07	0.2	0.3
17	481	1.3	99	0.1	0.000001	0.05	0.03	0.2	0.3
16	510.8	1.3	130	0.1	0.000001	0.05	0.05	0.2	0.3
15	159.8	1.3	230	0.1	0.000001	0.05	0.15	0.2	0.3
13	127.1	1.3	108	0.1	0.000001	0.05	0.02	0.2	0.3
12	335.1	1.3	140	0.1	0.000001	0.05	0.03	0.2	0.3
11	617.5	1.3	147	0.1	0.000001	0.05	0.06	0.2	0.3
10	163.3	1.3	137	0.1	0.000001	0.05	0.05	0.2	0.3
8	171.1	1.3	160	0.1	0.000001	0.05	0.04	0.2	0.3
6	284.3	1.3	112	0.1	0.000001	0.05	0.1	0.2	0.3
3	233.6	1.3	130	0.1	0.000001	0.05	0.05	0.2	0.3
1	135.3	1.3	50	0.1	0.000001	0.05	0.02	0.2	0.3
18	153.7	1.3	79	0.1	0.000001	0.05	0.02	0.2	0.3
14	293.9	1.3	65	0.1	0.000001	0.05	0.02	0.2	0.3
9	119.2	1.3	43	0.1	0.000001	0.05	0.01	0.2	0.3
7	123.4	1.3	66	0.1	0.000001	0.05	0.01	0.2	0.3
5	264.7	1.3	110	0.1	0.000001	0.05	0.05	0.2	0.3
2	270.5	1.3	66	0.1	0.000001	0.05	0.02	0.2	0.3
4	228.2	1.3	85	0.1	0.000001	0.05	0.01	0.2	0.3

Current project path: /home/fabrej/Bureau/builder_prj/virtual

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, à une date donnée
- il porte de 0 à n informations déclarées par une balise info, chaque valeur (attribut value) étant identifiée par une clé (attribut key)

Section <run>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <run>
    <deltat>60</deltat>
    <period begin="1997-03-29_03:00:18" end="1997-04-01_16:23:21" />
    <!-- <valuesbuffer steps="10" /> -->
    <!-- <filesbuffer kbytes="5" /> -->
  </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 des balises <valuesbuffer> et <filesbuffer> (optionnelles)

Section <run>

The screenshot displays the OpenFLUID-Builder [virtual] interface. The main window is titled "OpenFLUID-Builder [virtual]" and features a menu bar (Project, Edit, Simulation, Help) and a toolbar with icons for New, Open..., Save, Close, Map View, and Run... (a green play button).

The interface is divided into several panels:

- Project Explorer:** Shows a tree view with "Model", "Domain", and "Simulation". Under "Simulation", the "Run" configuration is selected. It displays the simulation period: "From : 1998-09-07 12:01:00", "To : 1998-09-07 21:58:00", and "Delta T : 60 s". Below this, there is an "Outputs" section and a "Results" section.
- Project Dashboard:** A table showing the status of various project components:

Model definition	ok
Model parameters	threshold is missing coeffks is missing
Required files	ok
Spatial representation	ok
Input data	ok
Project consistency	ok
Outputs	ok
- Time Management:** A section for configuring the simulation time. It includes:
 - Delta T:** A spin box set to 60 seconds.
 - Period Begin:** A date field set to 1998-09-07 12:01:00 and a time field set to 0 h, 0 m, 0 s.
 - Period End:** A date field set to 1998-09-07 21:58:00 and a time field set to 0 h, 0 m, 0 s.
- Memory Management:** A section for configuring memory usage. It includes:
 - Files Buffer:** A spin box set to 2 KB.
 - Values Buffer:** A checkbox (unchecked) and a spin box set to 1 steps.

The status bar at the bottom indicates the current project path: "Current project path: /home/fabrejc/Bureau/builder_prj/virtual".

Section <output>

Exemple

```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <output>
    <files colsep="_" dtformat="%Y-%m-%dT%H:%M:%S" commentchar="#"
      precision="7">
      <set name="full" unitsclass="SU" unitsIDs="*" vars="*" />
      <set name="oneunit" unitsclass="SU" unitsIDs="1" vars="*" />
      <set name="onevar" unitsclass="RS" unitsIDs="*" vars="water.surf.H.level-rs" />
    </files>
  </output>
</openfluid>
```

- configuration de jeux de données de sortie (<set>)
- "customisation" des sorties : format de dates, choix des unités, choix des variables, précision décimale, ...

Section <output>

The screenshot displays the OpenFLUID-Builder [virtual] application window. The interface includes a menu bar (Project, Edit, Simulation, Help), a toolbar with icons for New, Open..., Save, Close, Map View, and Run..., and a Project Explorer on the left. The Project Explorer shows a tree view with Model, Domain, Simulation, and Run. The Run section displays simulation dates and a Delta T of 60 s. Below this is an Outputs section and a Results section. A Project Dashboard at the bottom left lists various components and their status: Model definition (ok), Model parameters (threshold is missing, coeffs is missing), Required files (ok), Spatial representation (ok), Input data (ok), Project consistency (ok), and Outputs (ok). The main workspace is divided into two panes: Files formats and Sets definition. The Files formats pane contains a table with columns for Format name, Column separator, Date format, and Comment character. The Sets definition pane contains a table with columns for Set name, Unit class, Unit IDs, Variables, Format name, and Precision. The current project path is shown at the bottom: /home/fabrejc/Bureau/builder_prj/virtual.

OpenFLUID-Builder [virtual]

Project Explorer

- Model
- Domain
- Simulation
 - Run
 - From : 1998-09-07 12:01:00
 - To : 1998-09-07 21:58:00
 - Delta T : 60 s

Outputs

Results

Project Dashboard

- Model definition ok
- Model parameters threshold is missing
coeffs is missing
- Required files ok
- Spatial representation ok
- Input data ok
- Project consistency ok
- Outputs ok

Files formats

Format name	Column separator	Date format	Comment character
Format #1		%%Y%%m%%dT%%H%%M%%S	%
Format #2	:	%%Y%%m%%d%%H%%M%%S	#
Format #3		%%Y%%m%%d%%H%%M%%S	%

Sets definition

Set name	Unit class	Unit IDs	Variables	Format name	Precision
fullGU	GU	*	*	Format #1	5
fullRS	RS	*	*	Format #1	5
fullSU	SU	*	*	Format #1	5

Current project path: /home/fabrejc/Bureau/builder_prj/virtual

Environnement

Répertoires par défaut (sous Unix/Linux) pour un utilisateur tom

- OpenFLUID personnel :
/home/tom/.openfluid
- Données en entrée :
/home/tom/.openfluid/OPENFLUID.IN
- Données résultats :
/home/tom/.openfluid/OPENFLUID.OUT
- Fonctions de simulation :
/home/tom/.openfluid/functions

Variables d'environnement

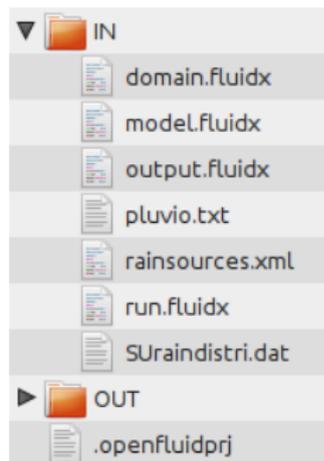
- \$OPENFLUID_FUNCS_PATH : chemins(s) de recherche de fonctions de simulation

Notion de projet OpenFLUID

Un projet OpenFLUID est un répertoire qui rassemble au minimum:

- d'un jeu de données d'entrée (sous-répertoire IN), contenant notamment les fichiers `.fluidx`
- d'un ou plusieurs jeu de résultats (sous-répertoire(s) OUT)
- des métadonnées du projet (fichier caché `.openfluidprj`)

Nom du projet, auteurs, dates de création et de modification



OpenFLUID Builder recherche des projets existants dans un répertoire par défaut, qui peut être configuré depuis l'application

Plan

- 1 Présentation générale
- 2 Exemples d'applications
- 3 En pratique
- 4 Jeux de données
- 5 Compléments**
 - Ligne de commande
 - Buddies
 - Dépôt de fonctions de simulation
 - Perspectives de développement

Principales options de la ligne de commande OpenFLUID-Engine

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 fonctions de simulation
-q	affichage silencieux pendant la simulation
-v	affichage détaillé pendant la simulation
-k	activation du profiling de simulation
-f	liste des fonctions disponibles
-r	rapport détaillé sur les fonctions disponibles
-u	rapport détaillé sur une(des) fonction(s) particulière(s)
--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:

- **func2doc** : génération automatique de documentation scientifique
- **convert** : convertisseur de formats de jeux de données d'entrée
- **newfunc** : générateur de code source de fonctions de simulation
- **newdata** : générateur de jeux de données vides formatés

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

Buddy func2doc

Utilisation du code source de la fonction de simulation pour en extraire les informations nécessaires

- Lecture de la signature de la fonction de simulation
- Insertion du contenu \LaTeX placé en commentaires du code source entre les balises `<func2doc>` et `</func2doc>` (optionnel)

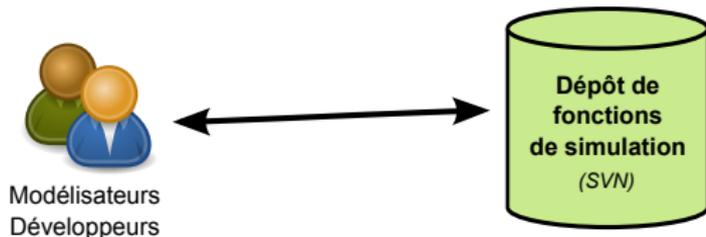
Génère un fichier latex, pouvant être converti à la volée en PDF ou en HTML

Exemple

```
openfluid-engine --buddy func2doc --buddyopts \  
  inputcpp=MyFunc.cpp,outputdir=./doc,PDF=1
```

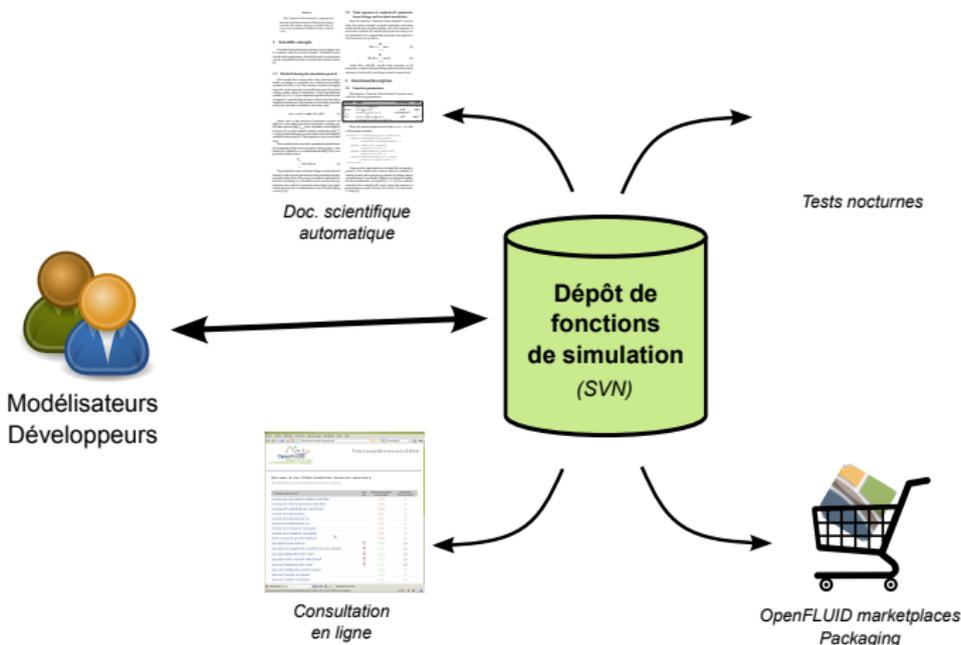
Dépôt de fonctions de simulation

Capitalisation et partage des fonctions de simulation développées au sein d'un groupe de travail



Dépôt de fonctions de simulation

Capitalisation et partage des fonctions de simulation développées au sein d'un groupe de travail



Références & Ressources



J.C. Fabre, X. Louchart, R. Moussa, C. Dagès, F. Colin, M. Rabotin, D. Raclot, P. Lagacherie, and Voltz M.
OpenFLUID: a software environment for modelling fluxes in landscapes.
In *LANDMOD2010*, INRA, CIRAD, page 13pp, Montpellier, France, 2010. Quae.



Site web OpenFLUID.
<http://www.umr-lisah.fr/openfluid/>.



Site web OpenFLUID Community.
<http://www.umr-lisah.fr/openfluid/community/>.