



OpenFLUID : Une plateforme logicielle
pour la modélisation spatio-temporelle
des paysages

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

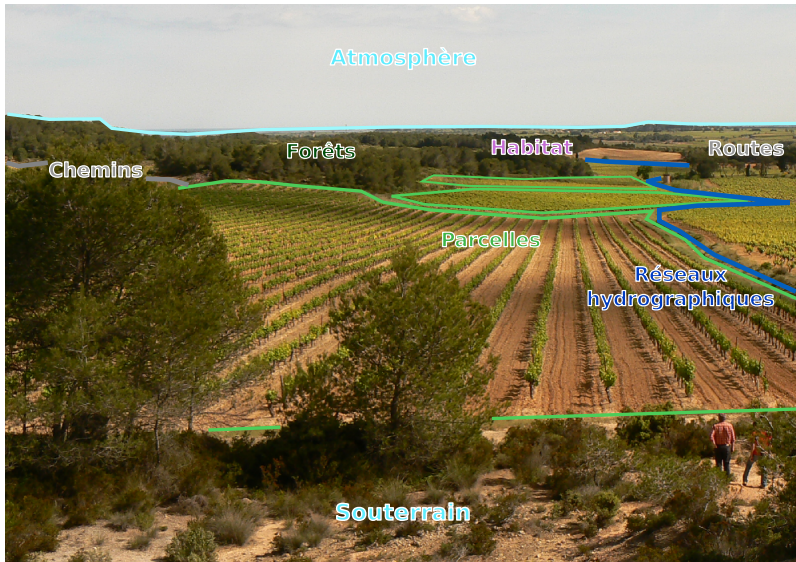
- 1 Panorama
- 2 Concepts
- 3 Plateforme logicielle
- 4 Partage de modèles
- 5 Exemples d'applications
- 6 En savoir plus

Panorama

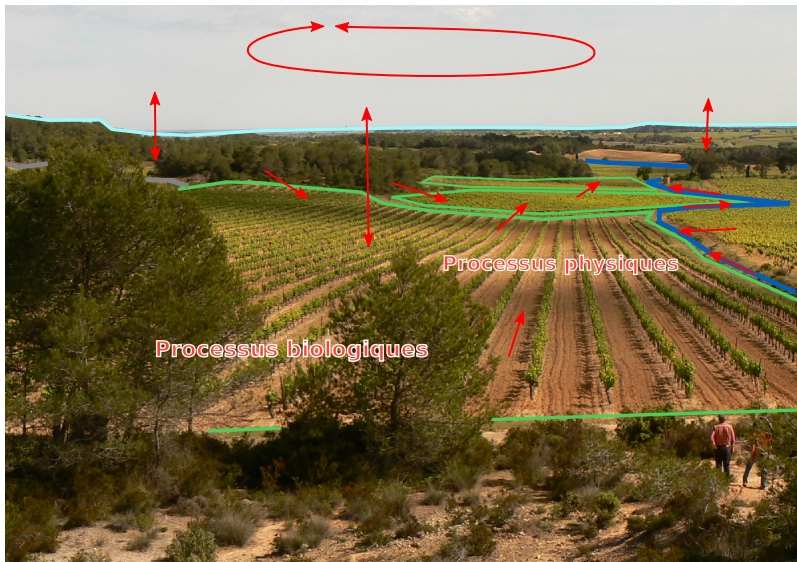
Le paysage



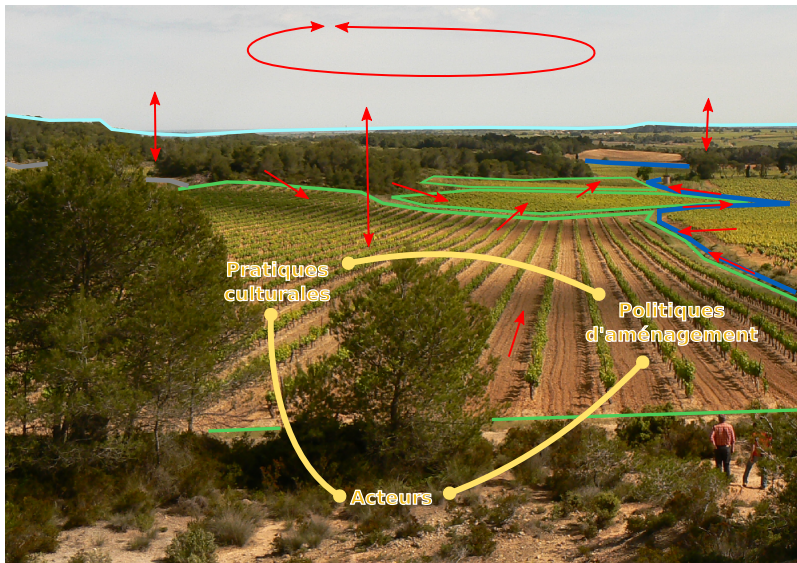
Le paysage



Le paysage



Le paysage



Etude du paysage

Le paysage est un **système complexe** (agrosystème, agroécosystème), où de nombreux phénomènes **interagissent fortement dans le temps et l'espace**

Cadre d'étude et d'évaluation de nombreux **services écosystémiques** (MEA, 2006)

Etude de l'organisation et du fonctionnement du paysage
⇒ modélisation de la **structure** et de la **dynamique** du système

Selon les objectifs:

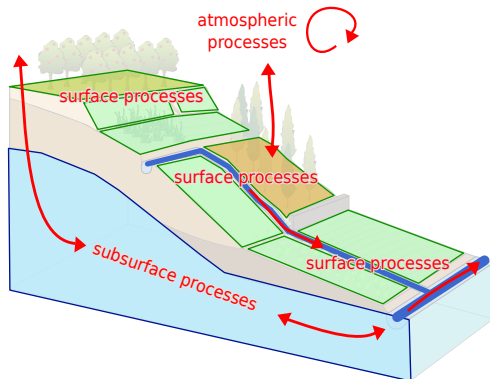
- choix des échelles,
- détermination des éléments pertinents dans le paysage, et relations entre ces éléments,
- identification des processus majeurs à prendre en compte,
- ...

Modélisation du fonctionnement du paysage

Représentation **numérique**
du paysage

+

Modélisation des **processus**
en **interaction**



Modélisation du fonctionnement du paysage

Représentation **numérique** du paysage

- **éléments du paysage**
(parcelles, routes, fossés, nappes, ...)
- **propriétés** de ces éléments
(géométrie, propriétés physiques, ...)
- **relations/connexions** entre ces éléments (topologiques et hiérarchiques)

+

Modélisation des **processus** en **interaction**

- **dynamiques locales et spatiales** des processus
(transferts, évolutions, décisions, ...)
- **couplage** entre les processus (interactions, rétroactions)



Concepts

Vers une plateforme de modélisation...

Début 90's: développement de modèles spatio-temporels

1995: développement du modèle MHYDAS

2000-2005: couplages de plus en plus nombreux, "explosion" des codes sources de modèles

2005-2006: réflexion pour le développement d'une **plateforme de modélisation**:

⇒ moderniser l'infrastructure de modélisation, normaliser les développements, se doter d'un socle commun pour la modélisation au LISAH

Août 2006: première ligne de code OpenFLUID

Juillet 2007: première formation OpenFLUID (interne au LISAH)

What is OpenFLUID?

Plateforme logicielle pour la modélisation et la simulation du **fonctionnement spatio-temporel des paysages**, principalement focalisée sur les flux

- Intégration de représentations numériques de l'espace
- Branchement de modèles, couplage automatique

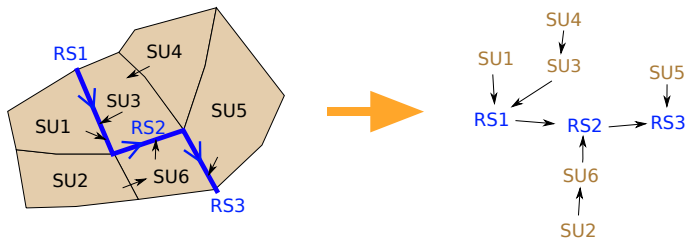
Construction de modèles couplés **en fonction des objectifs de modélisation**, à partir de modèles disponibles

Socle collaboratif pour le développement, capitalisation, réutilisation de modèles dans des groupes de travail

Représentation de l'espace sous OpenFLUID

L'espace est représenté par OpenFLUID sous la forme d'un **graphe orienté**

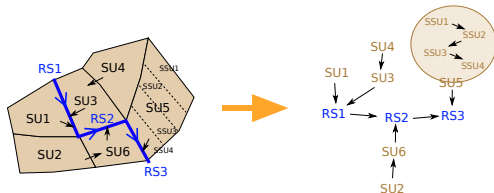
- Les noeuds sont les **unités spatiales** composant l'espace, rangés par **classes d'unités spatiales**
- Les arcs orientés sont les **relations** entre les unités spatiales
- Chaque noeud porte des **attributs propres** à l'unité spatiale qu'il représente



Nombreux **algorithmes** disponibles sur les graphes
(calculs de connexité, parcours, parallélisation, ...)

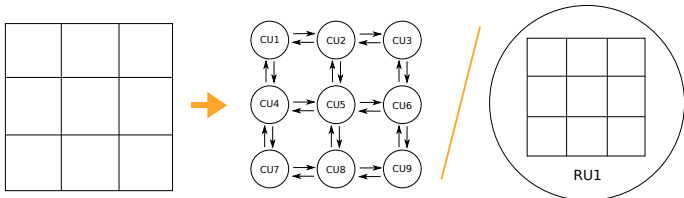
Représentation de l'espace sous OpenFLUID

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



Représentation de la dynamique sous OpenFLUID

La dynamique est représentée par des **simulateurs** OpenFLUID
= **codes de calcul** développés à partir de modèles mathématiques

- Il simule un ou plusieurs processus spatiaux (transferts, évolutions, ...)
- Il fait **évoluer les états** des unités spatiales
- Il peut utiliser l'ensemble des informations de simulation disponibles pour ses calculs (attributs spatiaux, variables, ...)

Un simulateur est construit sous la forme
d'un **plugin** pour la plateforme OpenFLUID:

- développé "from scratch"
- **encapsulation d'un code de calcul** existant



Possibilité de **parallélisation automatique des calculs**, basée sur la structure du graphe d'espace

Représentation de la dynamique sous OpenFLUID

Signature des simulateurs et couplage

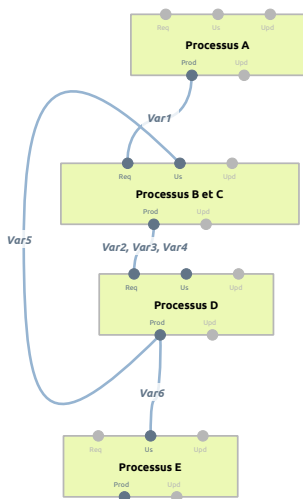
Chaque simulateur déclare son **comportement** au travers de sa **signature**:

- **Variables** requises et/ou utilisées en entrée, produites et/ou mises à jour en sortie
- **Attributs spatiaux** requis ou utilisés
- Pas de temps, modifications spatiales possibles, ...

ainsi que des **métadonnées**

- Nom de l'auteur, description, version ...

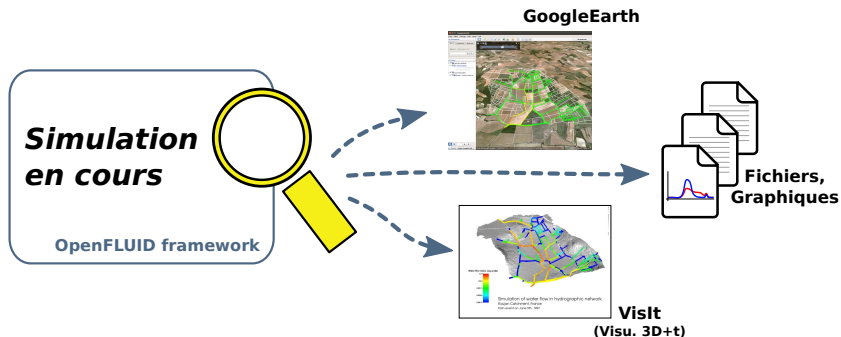
Le couplage de simulateurs est **réalisé et vérifié automatiquement** par la plateforme à partir des signatures des simulateurs mobilisés



Observation des simulations OpenFLUID

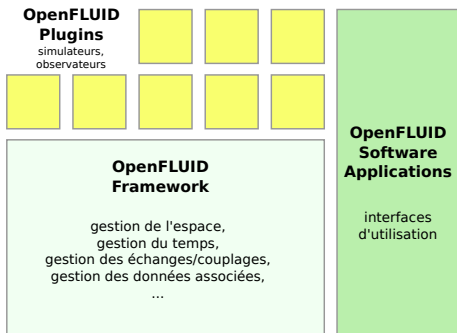
Les données de simulation peuvent être observées tout au long des simulations via des **observateurs** (plugins)

- export de données, contrôles, ...
- une simulation peut comporter de 0 à n observateurs qui composent le **monitoring** de simulation



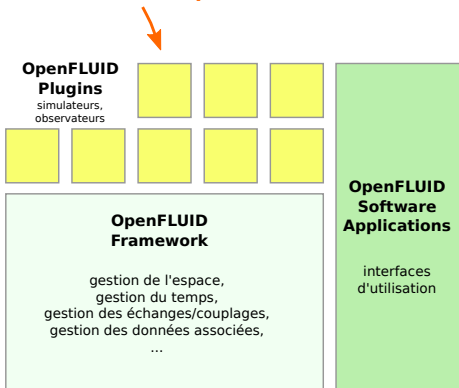
Plateforme logicielle

Infrastructure logicielle OpenFLUID



Infrastructure logicielle OpenFLUID

**Connaissance scientifique
et thématique**



**Mise en
application,
jeux de données**

**Ingénierie logicielle
pour la modélisation spatio-temporelle**

Utilisation en ligne de commande

Utilisation en ligne de commande

⇒ console, cluster de calcul, scripting, ...

```

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]

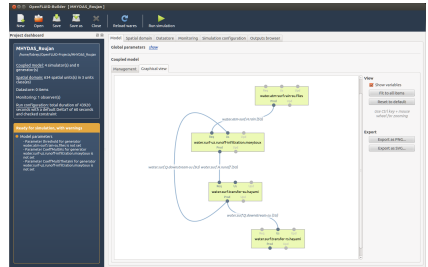
Progress          Real time          Status
-----
  0.29% 2001-01-01 01:02:57      [OK]
  0.58% 2001-01-01 02:02:54      [OK]
  0.87% 2001-01-01 03:02:51      [OK]
  1.16% 2001-01-01 04:02:48      [OK]
  1.45% 2001-01-01 05:02:45      [OK]
  1.74% 2001-01-01 06:02:42      [OK]
  2.03% 2001-01-01 07:02:39      [OK]

```

Utilisation en interface graphique

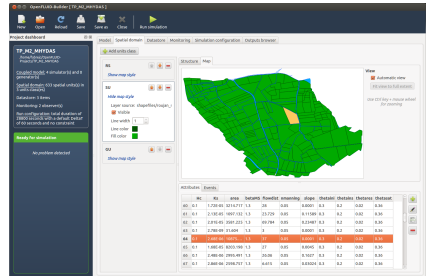
Interface graphique OpenFLUID-Builder

- Préparation des données
- Paramétrage des simulations
- Exécution des simulations
- Exploitation des résultats



Extensible par ajout de plugins Builder-extensions:

- import de données,
- aide au paramétrage,
- visualisation,
- ...



Utilisation en environnement externe

Package **ROpenFLUID**: utilisation d'OpenFLUID depuis l'environnement GNU R

- Paramétrage, exécution de simulations, exploitation des résultats

⇒ Profiter des fonctionnalités d'**exploration de simulations** sous R

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')
```

Module **PyOpenFLUID** (en cours de validation): Utilisation d'OpenFLUID depuis le langage Python (scripting, web, maths appli, ...)

Développement d'**applications spécifiques**, embarquant le moteur de simulation OpenFLUID

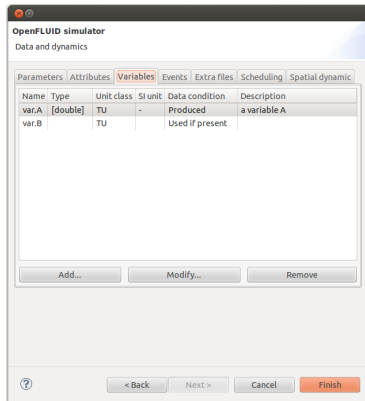
Environnement de développement de simulateurs

Utilisation de l'environnement de développement Eclipse + plugin OpenFLUID pour Eclipse

- Assistant à la définition de la signature
- Génération automatique du code source (sans code de calcul)
- Pré-configuration de la compilation

Outil sim2doc pour la documentation scientifique des simulateurs OpenFLUID

- Génération automatique en PDF ou HTML à partir de la signature du simulateur et des commentaires du code source (avec syntaxe $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$)



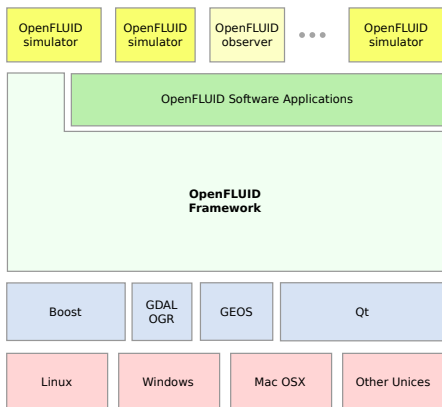
Architecture logicielle

Framework:

- Architecture **objet en C++**
- **API** complète
- Support aux applications logicielles standards et aux plugins
- Basé sur des bibliothèques de référence dans le domaine

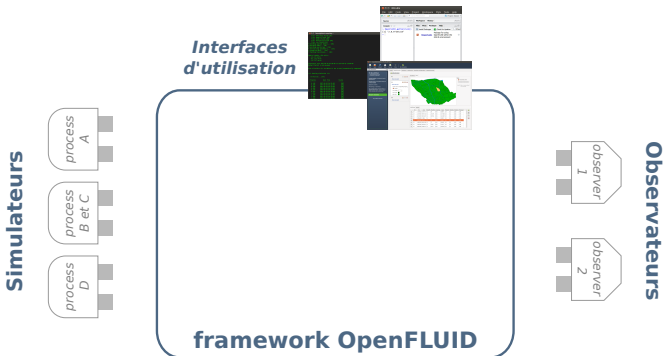
Plugins simulateurs et observateurs:

- C++ ou langage compatible (C, Fortran, ...)
- **Capitalisation** et **réutilisation** pour différentes simulations

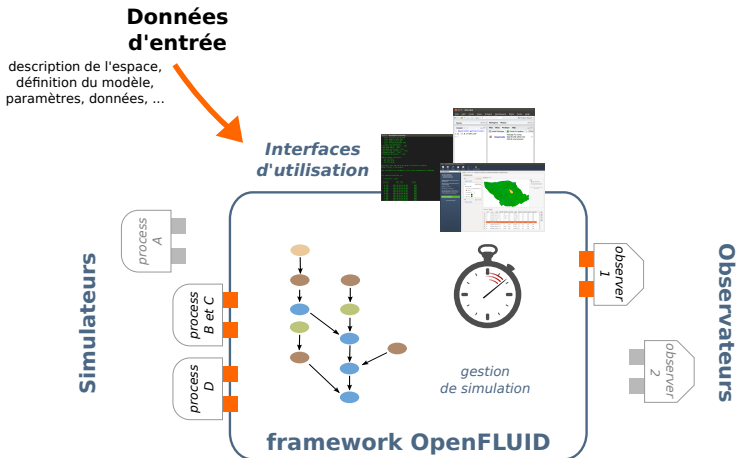


*OpenFLUID est disponible
sous licence libre GPL
ou sous licence propriétaire*

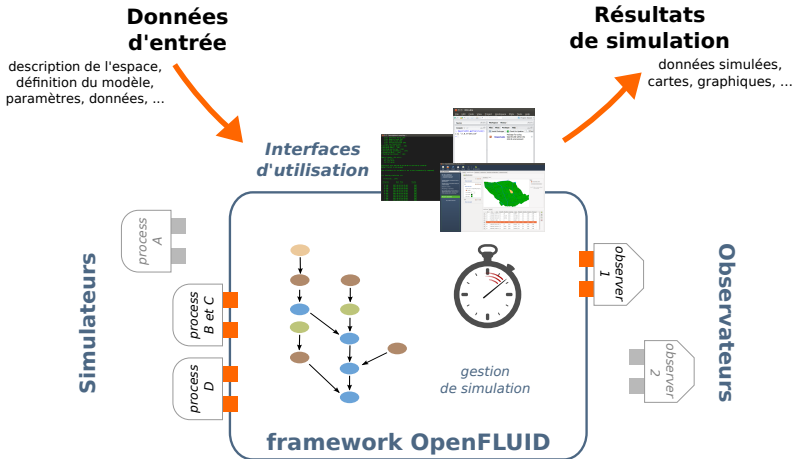
Démarche générale de simulation



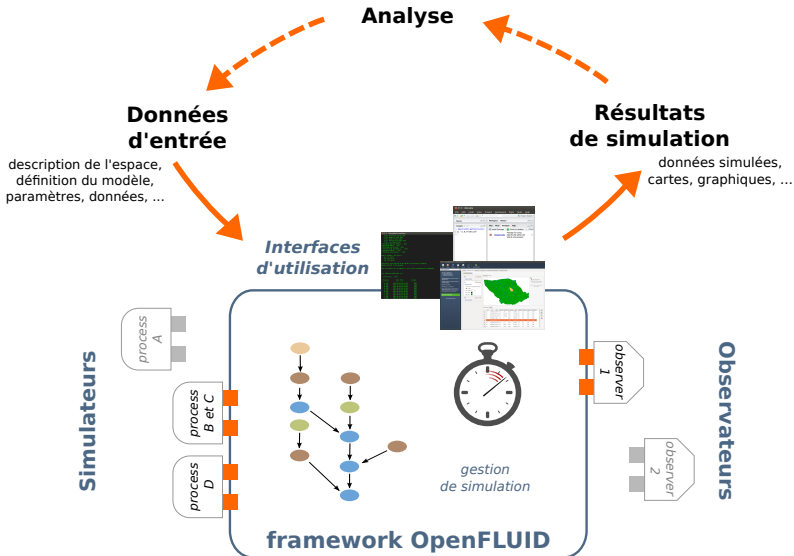
Démarche générale de simulation



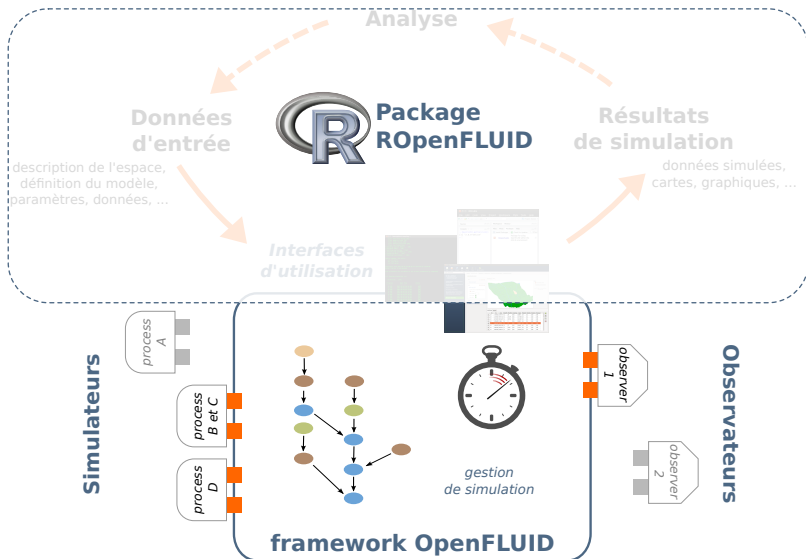
Démarche générale de simulation



Démarche générale de simulation



Démarche générale de simulation



Partage de modèles

OpenFLUID-WaresHub

Espace en ligne pour la **capitalisation** et le **partage** de simulateurs, observateurs, extensions graphiques

- gestion du **code source** (git)
- gestion des **tâches** (bugs, évolutions, review)
- gestion des **droits d'accès**
- mise à disposition de la **documentation associée**
- **méta-information** : licence, contributeurs, contact(s), ...

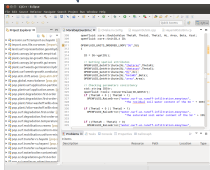
⇒ **Automatisation de traitements** (tests, packaging, ...)

Déployable par projet, groupe de travail, labo, ...

server
OpenFLUID-WaresHub



web service



Devel. Environment
local

Exemples des modèles développés au LISAH

The screenshot shows the LISAH WaresHub interface in a Mozilla Firefox browser. The page title is "LISAH WaresHub" and the user is logged in as "fabrejc" with a "Sign out" button. The main content area features a search bar and three buttons: "Simulators 85", "Observers 4", and "Builder-extensions 4". Below these is a "Reload informations" link. A search bar with "Searched terms" and a "Search" button is present. The main list displays several simulators with their IDs, descriptions, and OpenFLUID compatibility levels.

ID	Doc	OpenFLUID compatibility
energy.atm-surf.daily-min-max-temperature-data.files	Compute daily min and max temperature data from files 📄 📄 📄	★ 1.7 and previous
energy.atm-surf.global-radiation-data.files	Compute Global Radiation (GR) data from files 📄 📄 📄	★ 2.0 and previous
energy.atm-surf.temperature-data.files	Generate temperature data from files 📄 📄 📄	★ 2.0 and previous
energy.surf.rainfal-kinetic.zanchi-torri	Compute the kinetic energy of rainfall from Zanchi and Torri (1980) method 📄 📄 📄	★ 2.0 and previous
erosion.surf.attenuation.-		★ 2.0 and previous
erosion.surf.detachment-rs.-		★ 2.0 and previous

Liste des simulateurs disponibles

Exemples des modèles développés au LISAH

The screenshot shows a web browser window displaying the LISAH WaresHub interface. The page title is 'LISAH WaresHub' and the user is logged in as 'fabrejc'. The main content area shows a project page for 'simulators / water.surf.transfer-rs.hayami'. The project description is 'Discharge routing through the ditch network using diffusive wave equation resolved with Hayami method and discharge conversion to water height'. The documentation is available as a PDF. The OpenFLUID version(s) are listed as 2.0 and 1.7, 1.6, 1.5, 1.4.2, 1.4.1, 1.4.0. There are 4 contributors listed: Jean-Christophe Fabre (32 commit(s)), Simon Lalauze (23 commit(s)), Xavier Louchart (2 commit(s)), and Aline Libres (1 commit(s)). The current branch is 'openfluid-2.0'. The page includes tabs for 'General information', 'Commits history', and 'Issues'. There are also tags for '#water', '#surface', '#network', '#transfer', and '#hayami'. The development status is 'stable' and the license is 'GPL-3.0'. The contact information is 'contact@openfluid-project.org'. The footer of the page reads 'LISAH-WaresHub - contact'.

Hydrologie de surface intra-parcellaire

Parcelle AW6 - Roujan (*EU Life-Aware, X. Louchart*)



Parcelle

- 1200 m^2 , 1070 unités spatiales
- 4 simulateurs

Simulation des chemins de l'eau en intra-parcellaire

- 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

Aménagement d'un réseau de fossés

Bassin versant de Roujan (*thèse F. Levavasseur*)

Scenarios d'aménagement du réseau de fossés sur le bassin versant de Roujan (0.91 km²)

- évacuer l'eau des parcelles
- limiter les pertes en sol dues à l'érosion
- minimiser la longueur du réseau → minimiser les coûts de maintenance

Faible densité
38 fossés

Haute densité
322 fossés

Densité moyenne
223 fossés

Diagnostic pollution sur aire de captage

Puissalicon (*FEDER Phyt'Eau, collab. LISAH - BE Envilys - Groupe Eurofins*)

Simulation du **devenir et du transfert de produits phytosanitaires**,
intégrant **pratiques agricoles et processus hydrologiques**

Teneur en herbicides dans le sol sur 1 an

Transfert d'herbicides dans le réseau
hydrographique durant 1 événement de crue

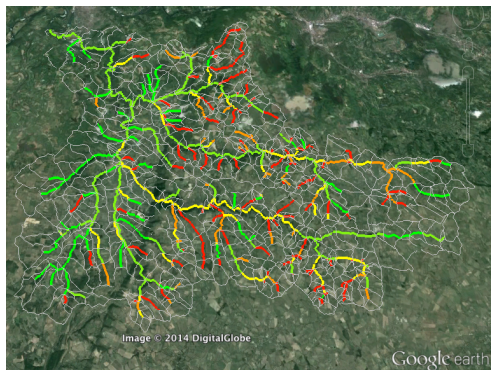
82ha, 279 parcelles, 112 tronçons de réseau, 11 simulateurs couplés

Impact hydrologique de retenues et prélèvements

Diège (collaboration LISAH - BE CEREG, transferts de technologie et de compétences)

Evaluation de l'impact hydrologique de retenues en tenant compte des prélèvements pratiqués

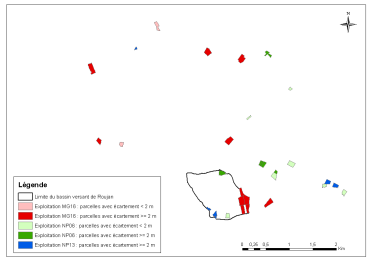
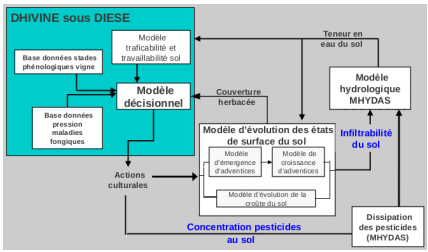
- 165km², 1115 unités spatiales
- 8 simulateurs couplés
- simulations sur 30 ans, $\Delta t = 1$ journée
- différents scénarios d'aménagement



Couplage hydrologie - décision agronomique

La Peyne - (thèse N.Paré, collab. BIA Toulouse, ANR GeDuQuE, ANR SP3A)

Construction d'un **modèle couplé pression-impact** pour l'**expérimentation virtuelle de pratiques culturales** à l'échelle de petits bassins versants



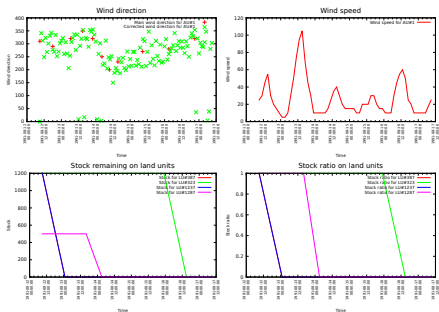
Encapsulation du framework DIESE (Rellier et al., BIA Toulouse) dans un simulateur OpenFLUID

Couplages de 15 simulateurs (biophysique + décision)

Génération dynamique d'**événements discrets** au cours de la simulation

Propagation d'incendie

Haute vallée de l'Orb - (Exemple 'proof of concept')



Modèle simplifié de propagation d'incendie

Simulation sur grille (1516 unités spatiales, ~9500ha),
Couplage de 2 simulateurs

Recalcul en cours de simulation du graphe de connexions spatiales en fonction
des directions de vent

En savoir plus

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.



Positionnement de la simulation à t_0

Déroulement d'une simulation

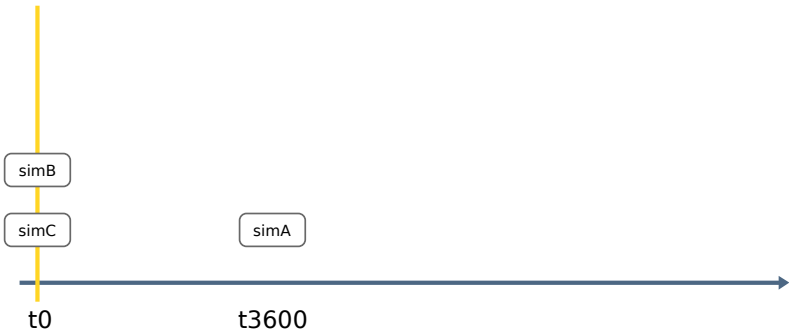
Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00 (3600s)**, sans contrainte de planification.



Exécution du simulateur simA à t0

Déroulement d'une simulation

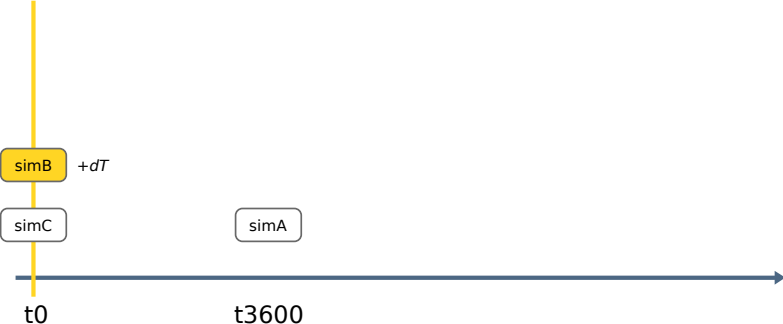
Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00 (3600s)**, sans contrainte de planification.



Positionnement de simA pour sa prochaine exécution

Déroulement d'une simulation

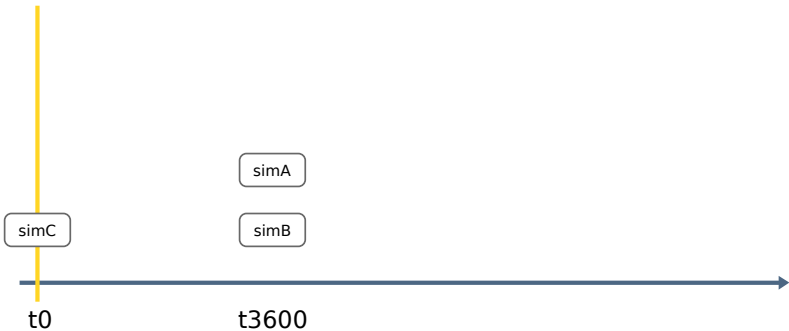
Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00** (3600s), sans contrainte de planification.



Exécution du simulateur simB à t0

Déroulement d'une simulation

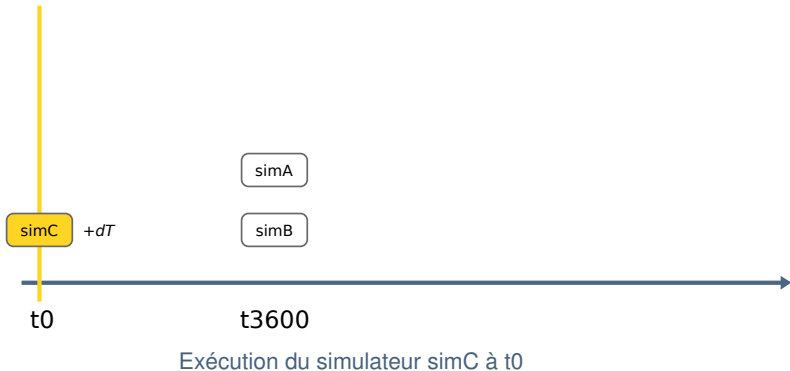
Exemple de simulation comprenant 3 simulateurs, avec un deltat (dt) par défaut de 1h00 (3600s), sans contrainte de planification.



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

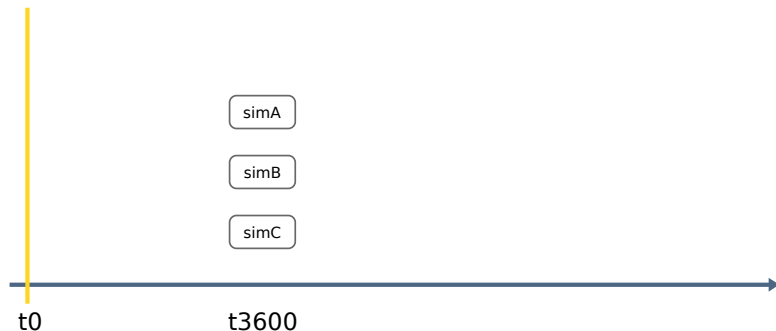
Déroulement d'une simulation

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



Déroulement d'une simulation

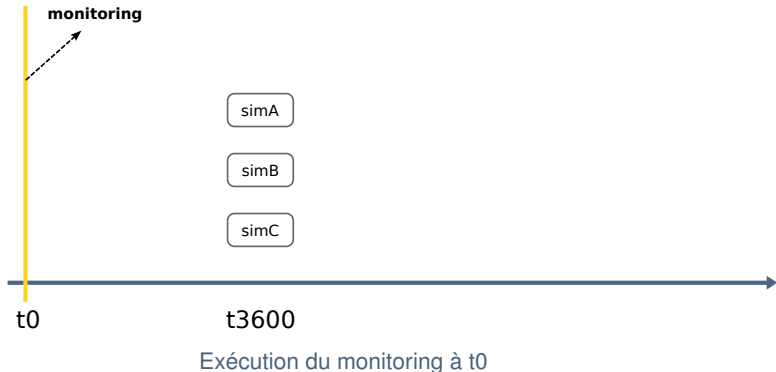
Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00 (3600s)**, sans contrainte de planification.



Positionnement de simC pour sa prochaine exécution

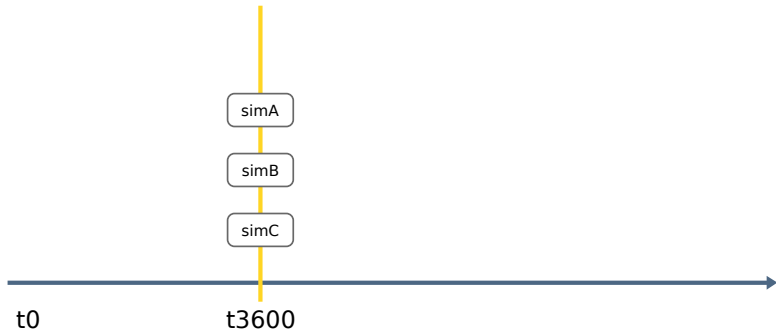
Déroulement d'une simulation

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



Déroulement d'une simulation

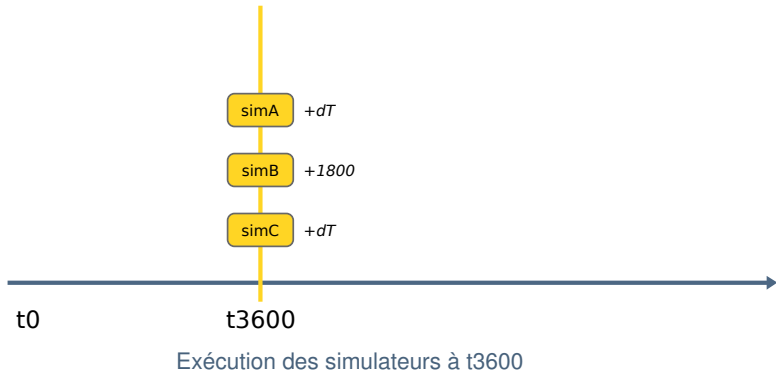
Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00 (3600s)**, sans contrainte de planification.



Positionnement de la simulation au prochain index de temps (t_{3600})

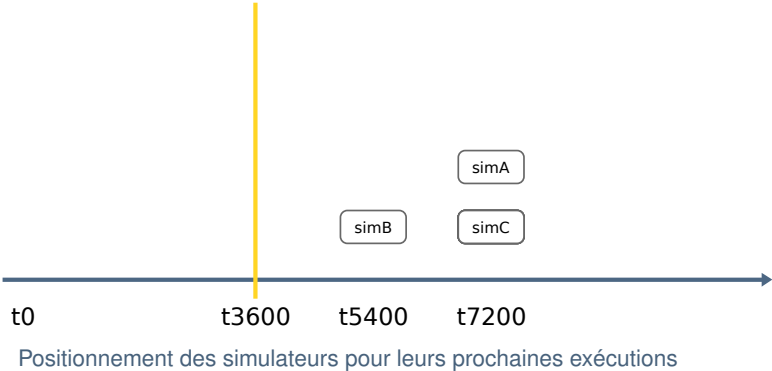
Déroulement d'une simulation

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



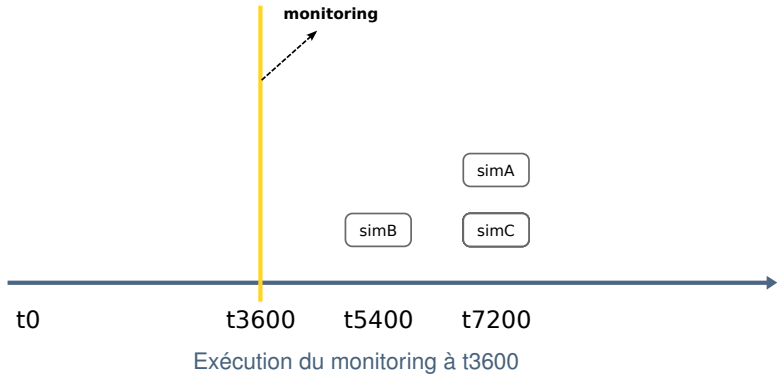
Déroulement d'une simulation

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



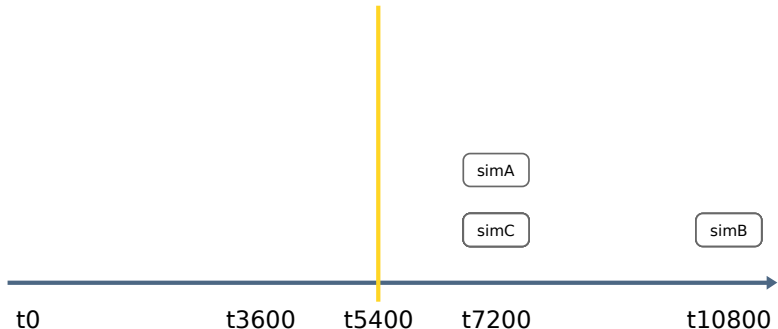
Déroulement d'une simulation

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



Déroulement d'une simulation

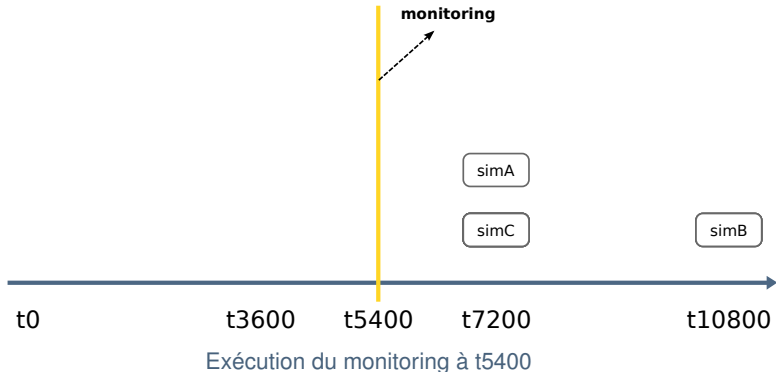
Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00** (3600s), sans contrainte de planification.



Positionnement de simB pour sa prochaine exécution

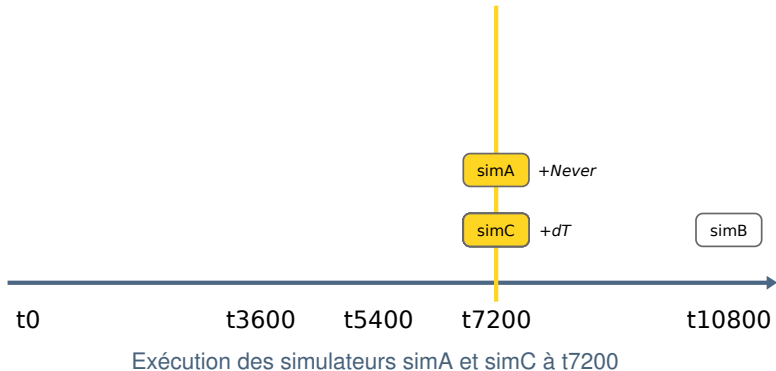
Déroulement d'une simulation

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



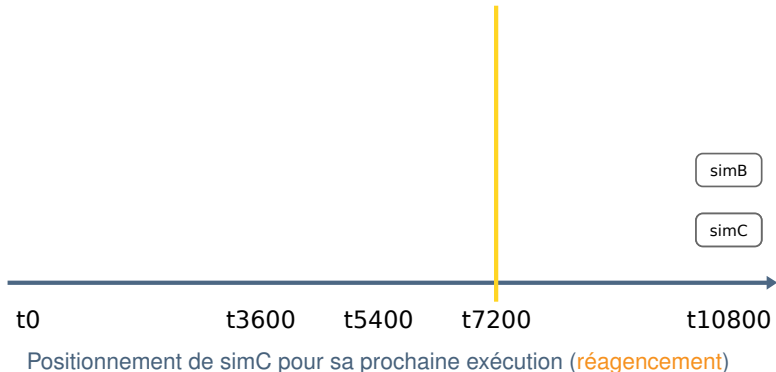
Déroulement d'une simulation

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



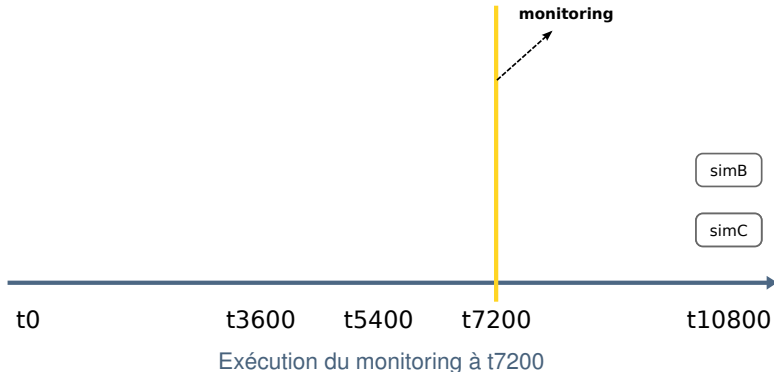
Déroulement d'une simulation

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



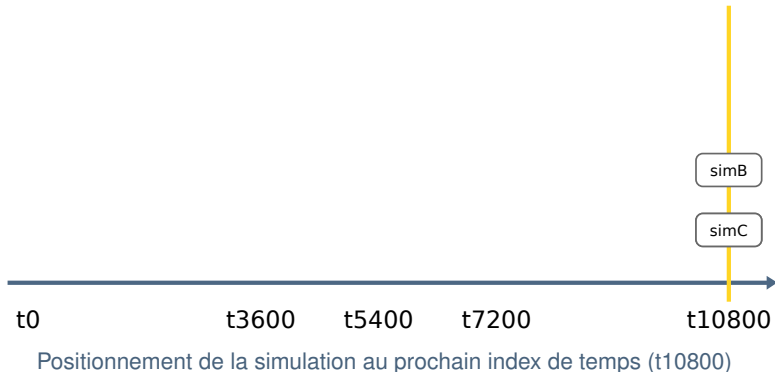
Déroulement d'une simulation

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.



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

Fonctionnalités complémentaires

- 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)
- ...

Développements en cours et à venir

En cours: développements axés sur la facilité d'utilisation

- Editeur **intégré** pour le développement de modèles (DevStudio)
- **Assistants de paramétrage** des simulateurs et observateurs
- Explorateur de **logs** et du **profiling** de simulation
- Protocole FluidHub pour la mise en place de web services autour d'OpenFLUID
- ...

A moyen et long terme

- Installations d'OpenFLUID sur serveurs accessibles sous la forme de web service (SimHub)
- Paramétrage sur connexions
- Approches individus centrées
- Nouveaux schémas de couplage, itérations
- ...

Références



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.



J.-C. Fabre, M. Rabotin, D. Crevoisier, A. Libres, C. Dagès, R. Moussa, Ph. Lagacherie, D. Raclot, and M. Voltz.

OpenFLUID: an open-source software environment for modelling fluxes in landscapes.

In *Vol. 15, EGU2013-8821-1, EGU General Assembly 2013*, Vienna (Austria), 7-12 avril 2013.



M. Rabotin, J.-C. Fabre, A. Libres, P. Lagacherie, D. Crevoisier, and R. Moussa.

Using graph approach for managing connectivity in integrative landscape modelling.

In *Vol. 15, EGU2013-8851, EGU General Assembly 2013*, Vienna (Austria), 7-12 avril 2013.

Téléchargements, Actualités, Docs, Supports de formation ...

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



@OpenFLUID

Dépôt GitHub (code source, suivi des développements, ...)

<https://github.com/OpenFLUID/>

Rapport d'activité 2014

http://www.openfluid-project.org/resources/docs/misc/OpenFLUID_RapportActivite2014_final.pdf