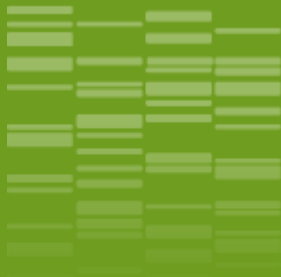


Vers une modularisation du modèle PaSim au sein de la plateforme RECORD

Retour d'expérience

PLAN

- ❖ Introduction
 - ❖ Le modèle PaSim
 - ❖ Pourquoi une plateforme de modélisation ?
- ❖ Contexte
 - ❖ RECORD / BioMa
 - ❖ Pourquoi une librairie déportée ?
- ❖ Modularisation du modèle
- ❖ Conclusion



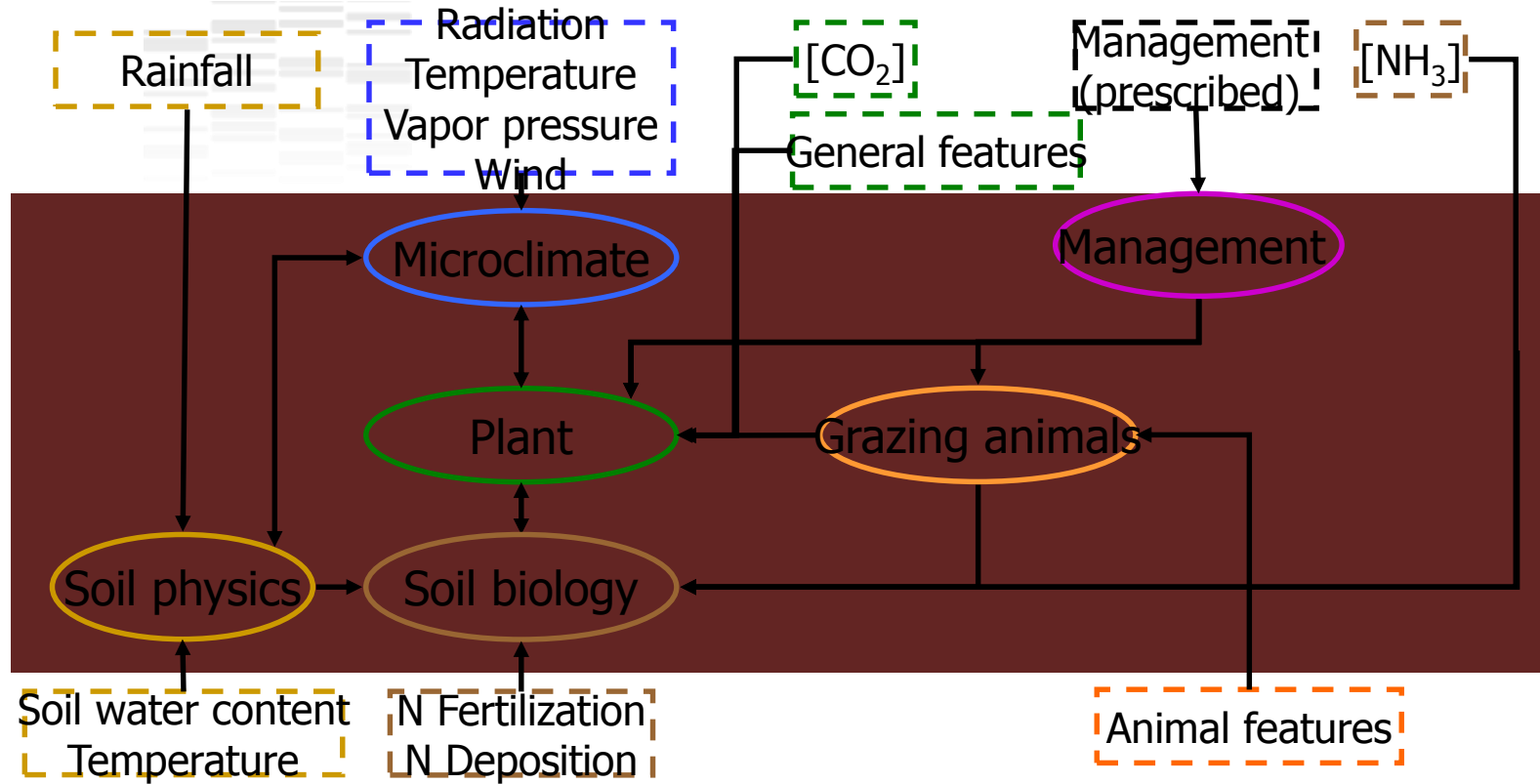
01

INTRODUCTION

LE MODÈLE PASIM : ETAT DE L'ART (1/2)

- Simule les cycles de l'eau, C et N.
- Adapté du modèle HPM et amélioré par Schmidt et al.
- Développer en ADSL puis réécriture en fortran 90.
- Utilise une approche par compartiment pour la végétation
- ~ 250 entrées découpés dans 13 fichiers.
- ~480 sorties possibles.
- Forte interdépendance des différents modules.

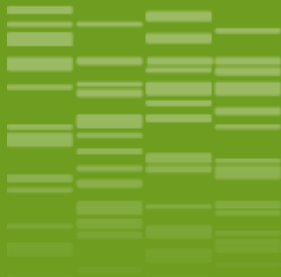
LE MODÈLE PASIM : ETAT DE L'ART (2/2)



POURQUOI UNE PLATEFORME DE MODELISATION ?



- Nouvelle demande en termes de modélisation
- Existence d'un grand nombre de code de simulation indépendants.
- Besoin de représenter la variabilité spatiale à l'intérieur des zones étudiées
- Besoin de simulation d'un nombre de variables important.
- Besoin de connaissances / résultats sur différents types de zones.
- Besoin de mutualiser les codes de simulation existants.



2

CONTEXTE

LA PLATEFORME RECORD

- Plate-forme pour la modélisation et la simulation de système de culture écrite en C++
- Caractéristiques :
 - Basé sur VLE + formalisme DEVS (systèmes dynamiques, événements discrets)
 - Facilité le développement de modèles complexes
 - Promouvoir la réutilisation et le partage des modèles
- Modularité :
 - Maintenabilité des codes plus faciles
 - Approche intégrative de modélisation
 - **Granularité**
- Couplage :
 - Capitalisation des modèles
 - Intégration de nouveaux codes / modèles
- Intégration d'outils d'analyse et de conception dans un environnement commun de modélisation

LA PLATEFORME BIOMA

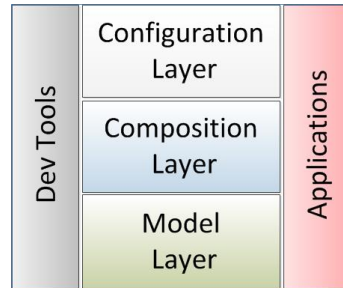


- Plate-forme pour la modélisation et la simulation écrite en C#.

- Caractéristiques :

- Réutilisation de composants.
- Approche modulaire, extensibilité.
- Indépendant de la plateforme.

- Architecture multicouches :



← Adaptateurs pour fonctionnalités avancées

← Modélisation à partir de composants

← **Granularité**, composants

QUELLES SOLUTIONS ?



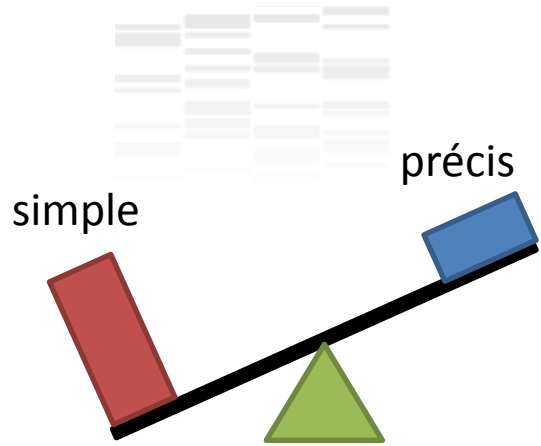
- 2 projets, 1 modèle, plusieurs possibilités.
- Encapsulation « brute » sur les 2 plateformes.
- Réécrire en partie le modèle.
- Réécriture complète du modèle pour chacune des 2 plateformes.

N'y a t-il pas une solution alternative ?

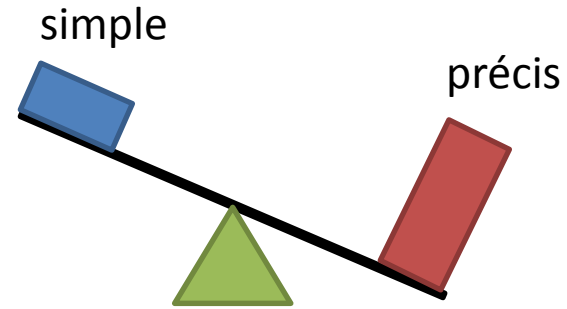
Pourquoi une librairie déportée ?

- Qu'est ce qu'un modèle mathématique ?
- Vision simplifiée d'un système réel.
- Système = groupe d'objet qui interagissent entre eux de manière organisée.
- Résultat de cette organisation est le comportement du système.
- Le modèle devient un outil pour comprendre le système.
- Un modèle mathématique décrit un système en utilisant des principes mathématiques.

Pourquoi une librairie déportée ?

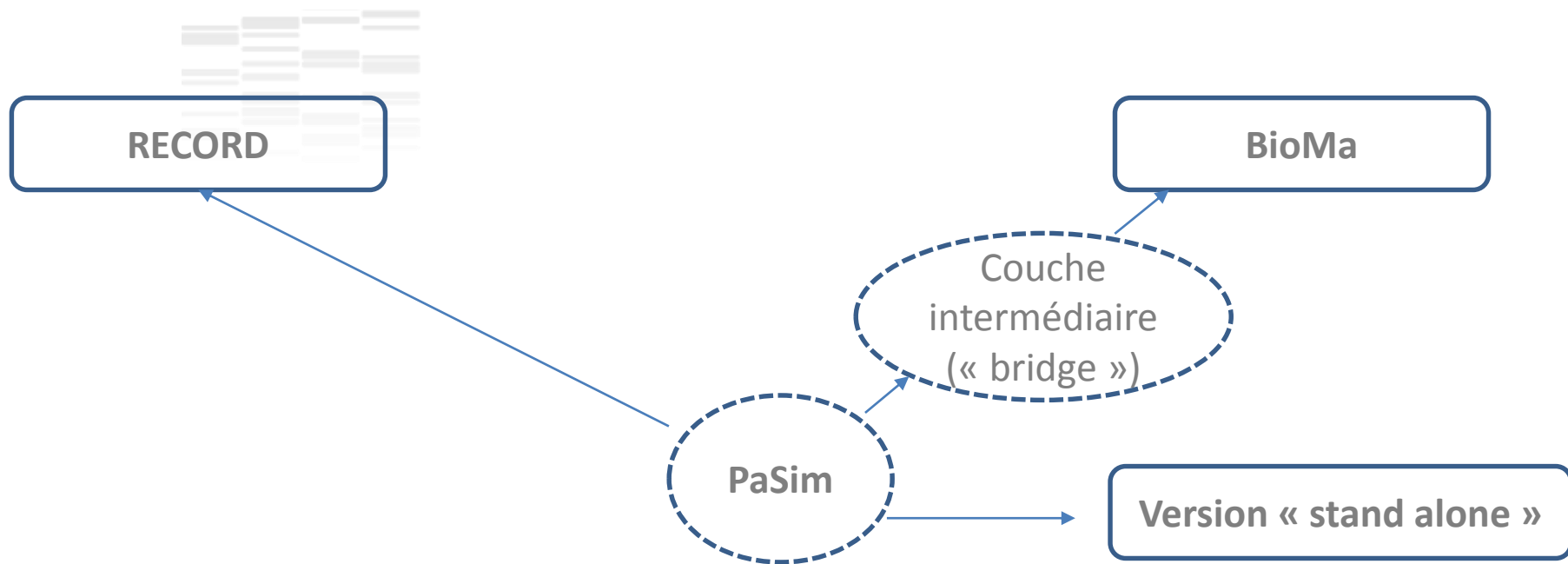


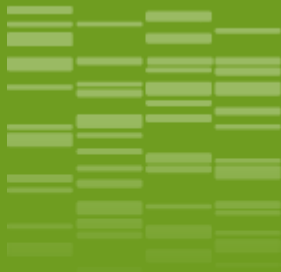
Modèle est simple mais moins précis



Modèle est précis mais moins simple

Pourquoi une librairie déportée ?





3

VERS UNE MODULARISATION DE PASIM

Généralités

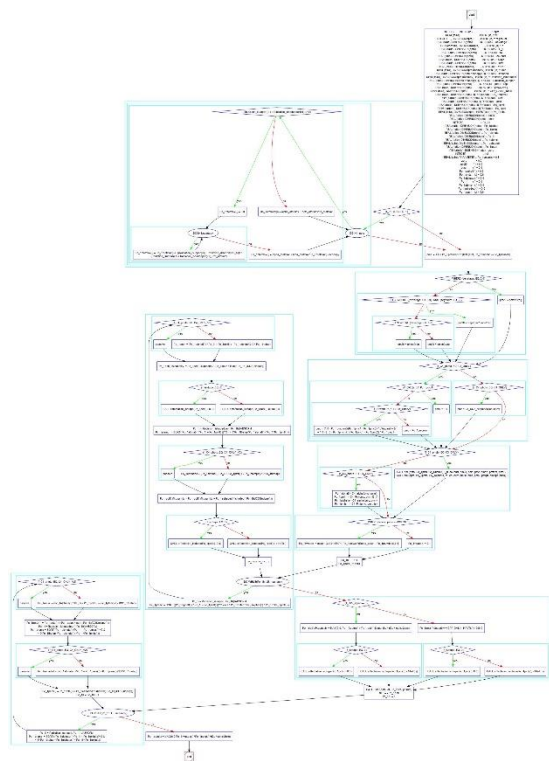
- Choix d'un langage utilisant le concept de paradigme objet : C++
- Modularité
- Facilité d'organisation
- Utilisation des concepts de programmation modernes
- Diviser la complexité en un ensemble logique
- Travailler plutôt sur le sujet que sur le verbe

Granularité du modèle



- Degré de résolution, de finesse, de détail, dans les objets et les processus représentés.
- Pour Pasim, niveau de granularité faible (2/3).
- Analyse du code + UML.
- Extraction d'objets à partir du code source.
- Diagramme de flux.

Graphe de contrôle de flux



Photosynthèse

Sénescence

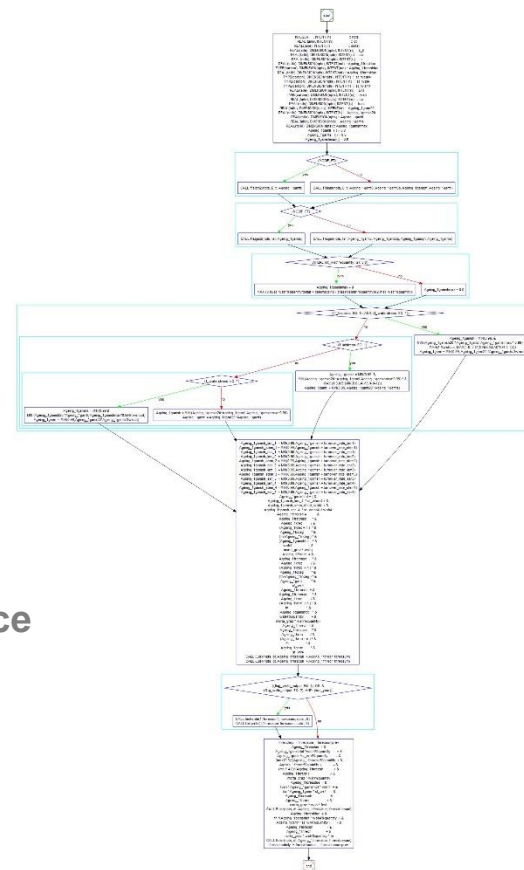
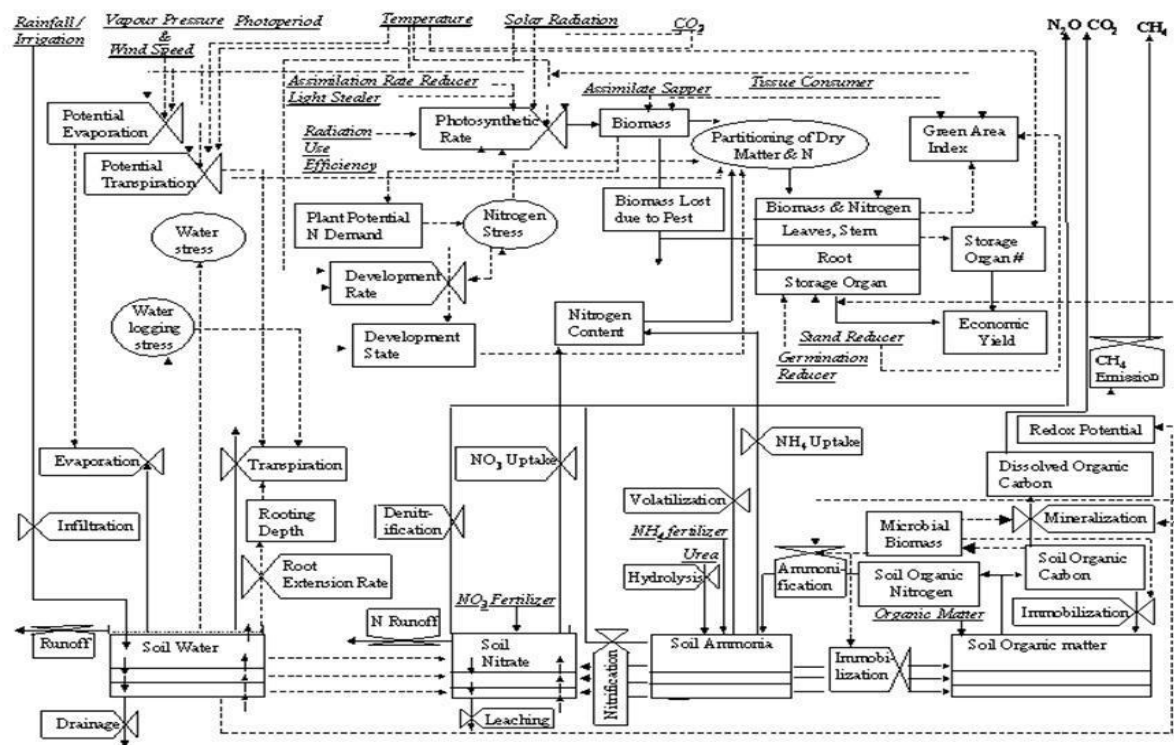


Diagramme de relation



Conclusion



- Le choix d'une librairie déportée apporte une flexibilité dans la réécriture.
- Des tests ont été réalisés sur les 2 plateformes : solution viable
- Le point clé est la granularité du modèle.
- Les premiers tests d'intégration sont concluants
- Travail en cours...



**Merci
pour
votre attention !**