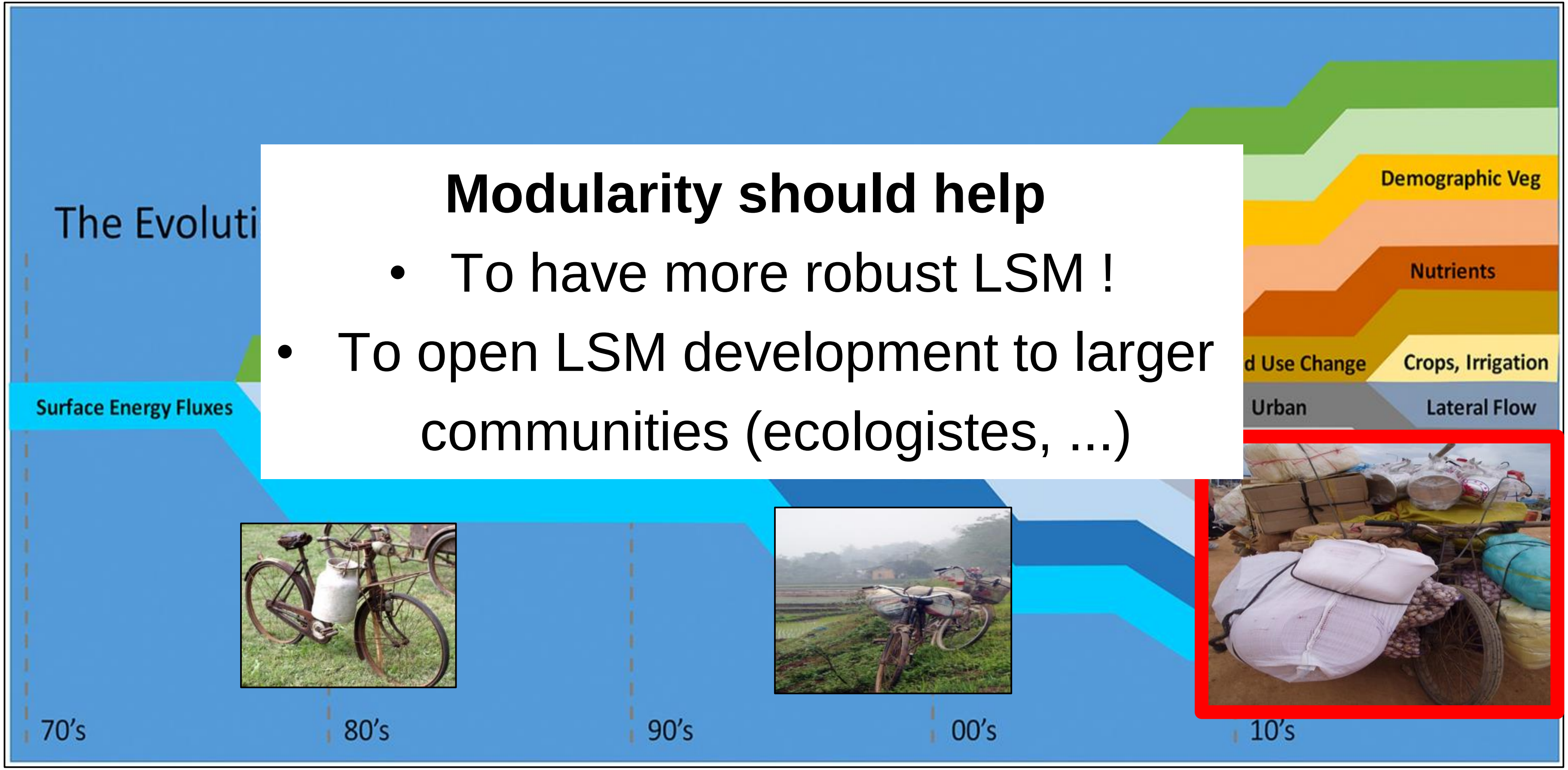
Technical challenges of ‘Modularity’: where we are now ? where we want to go ?

Philippe Peylin with contributions from
Elena S., Reiner S., Veronika G., Dave L., Bill S., R. Fisher,
Eleanor B., Jan P., Akihiko S., ….

Land Surface Modelling Summit : Oxford - Sept 2022

Why developing “Modularity” ?



Modularity should help

- To have more robust LSM !
- To open LSM development to larger communities (ecologistes, ...)

Why developing “Modularity” ?

- **To improve model robustness:**
  - Modularity requires well defined interfaces (including space-time interpolation)

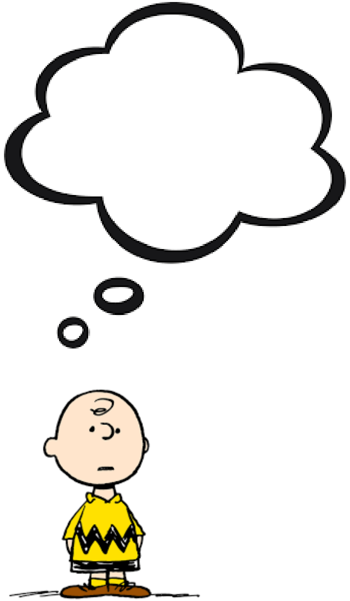
    ⇒ Helps structuring complex LSMs: Extensibility, Maintainability, etc..
  - Idealy Modules can be run in stand alone mode

    ⇒ to be more easily evaluated / calibrated / emulated
  - To better separate the “physics” from “Numerics / computer architecture”

- **For the LSM community:**
  - “Well defined modules” could be more easily exchanged between LSMs
  - Would help to test ‘process - representation’ with different LSM structures

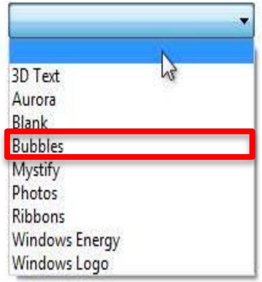
    ⇒ To assess model structural error of specific processes.
  - Facilitate the construction of “Community” land models

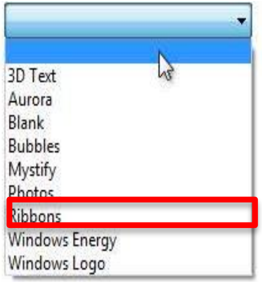
Potential dream : “plug and play” modules !



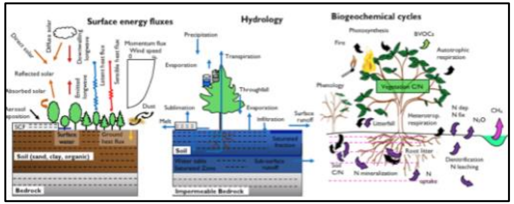
1) Select all model components from a “catalog”

Hydrology Photosynthesis Soil C model ....





2) Generate a new LSM



3) New model simulations ...

However….

- ➔ Modularity is however not easy to achieve !
- ➔ Defining “**Commun module structures”** accross groups is even a much greater challenge!

Some questions for this workshop:

- ➢ Should we try to define such a common structure ?
- ➢ If so, which level of granularity ?
- ➢ How to exchange expertise accross groups ?

A few challenges to define « inter-operable Modules »

- **Coupling between components are crucial !**

  * **Implicit vs Explicit** numerical scheme (Diff Equ.):

  Surface – atmosphere energy coupling !

  * **Feed-backs may complicate interfaces** :

  ex: Soil hydrology – thermics – C dynamics (permafrost) !

  * **Process separation** : Where do you draw the box ?

- **Computing infractructure is critical !**

  * Not all groups use the same **programming language**

  * **Parallelisation** is crucial and strategy differ btw groups (MPI, OpenMP,..)

  * Soon **GPU** may also pose challenges to share module

How to define / exchange « Modules » ?

Module: Set of processes with well defined Inputs / Outputs
can be run in a stand alone mode !

Existing exchange btw LSMs ?

Global LSM

Individual processes

High level model components
(ex. Soil hydrology ; Plant Demography Soil W – E budgets, ...)

Not really done yet !

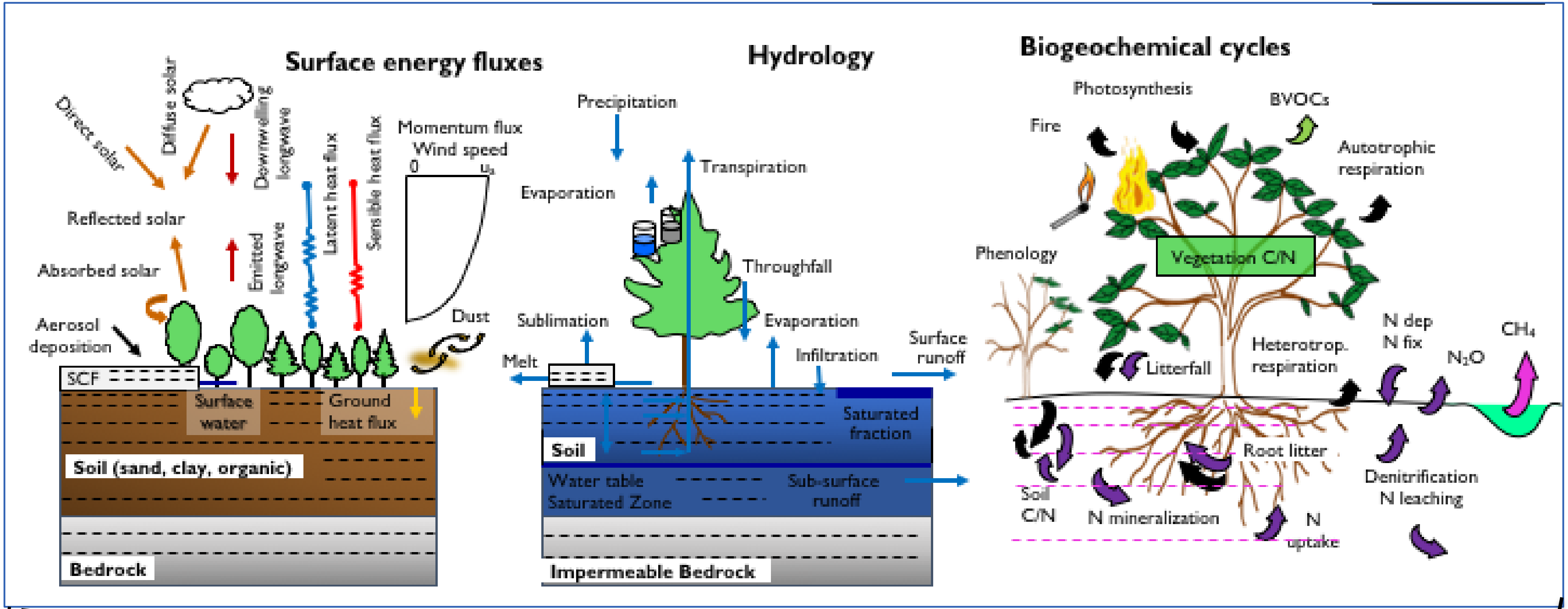
Intermediate level: groups of processes
(ex. Snow dynamic, Leaf level photosynthesis, Soil C dyn., ...)

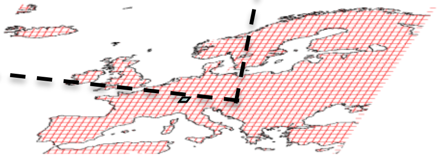
Some examples (usually more difficult than anticipated)

Low level: individual processes
(ex. Temperature sentivity for params., Albedo calculation, Traits description,...)

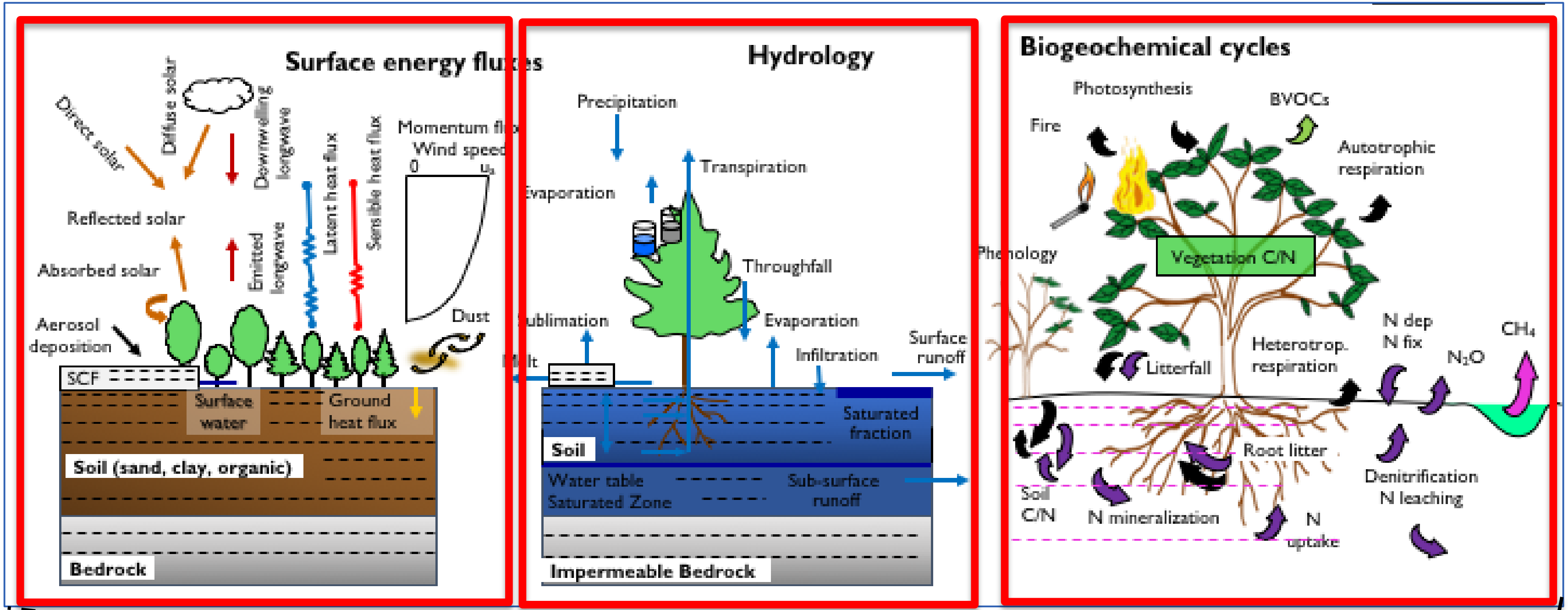
Many examples (curent practice)

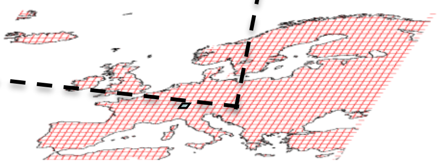
How to define « Modules »





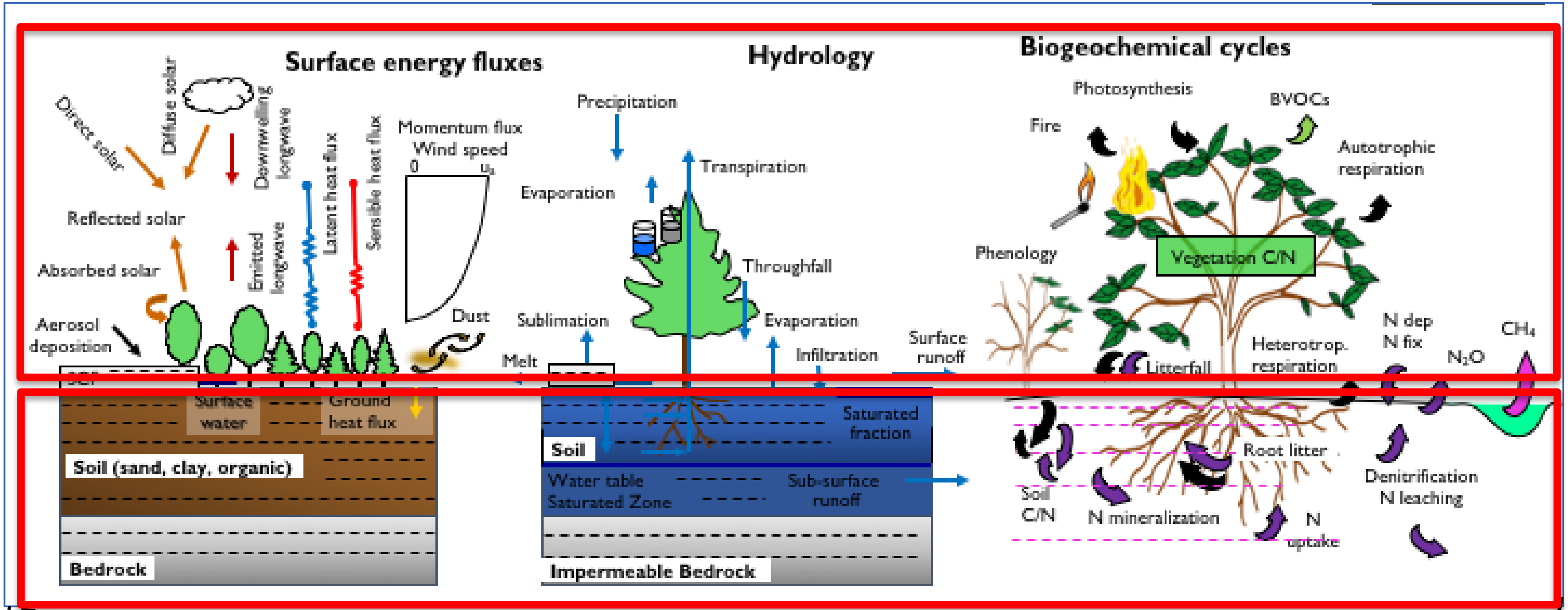
How to define « Modules »

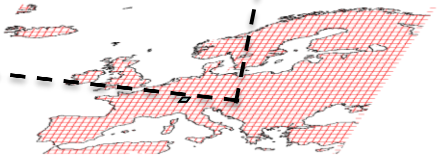




➔ Not optimial : too many interactions between Energy / Hydrology / Biogeochemistry !

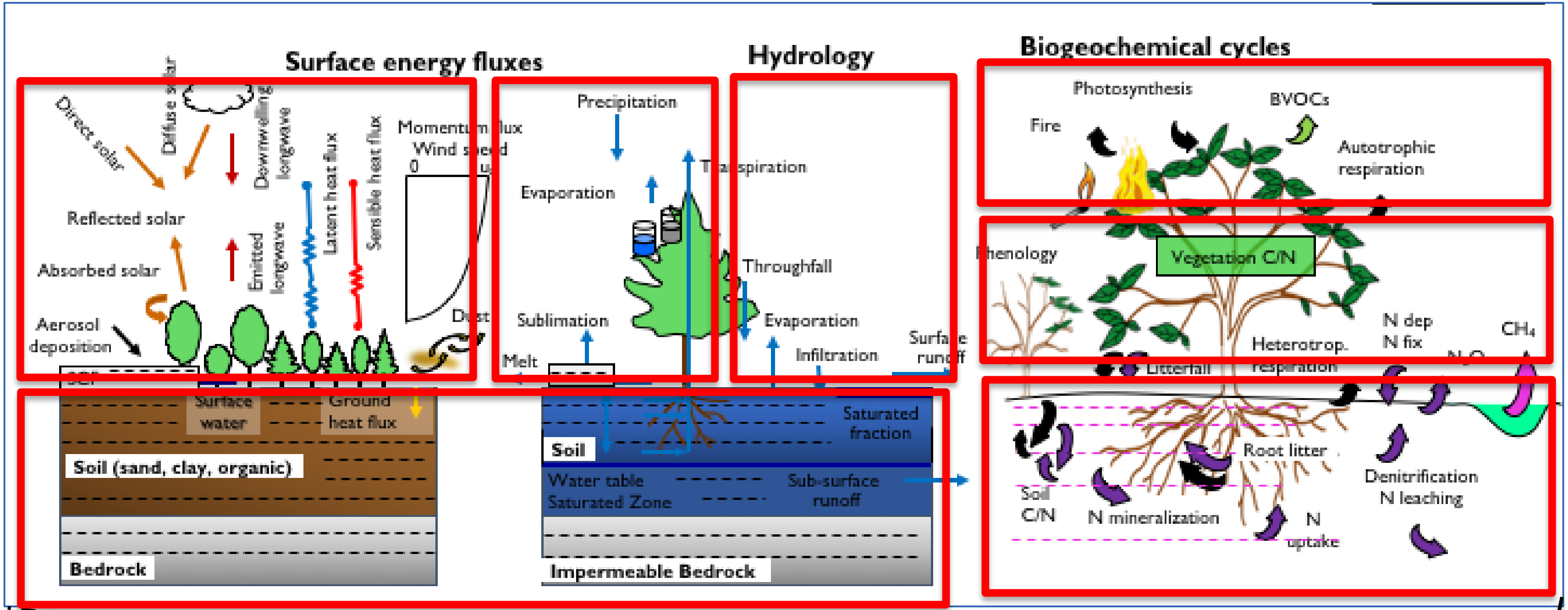
How to define « Modules »

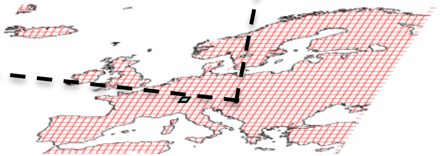




➔ Not optimial : too big modules !

How to define « Modules »





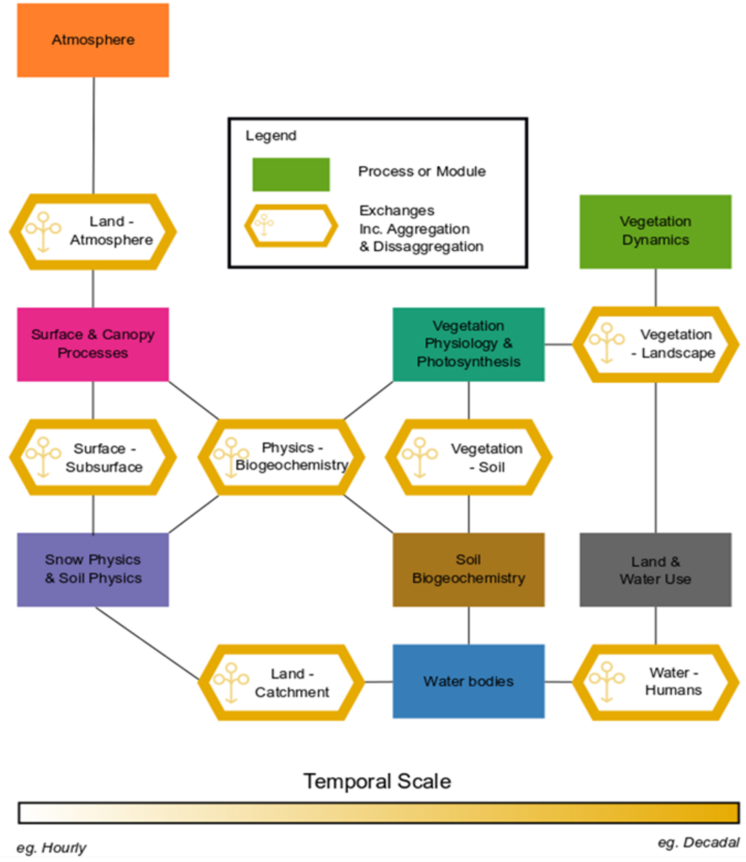
➔ Somewhere we may find good compromises !

Some examples of work around Modularity from different groups

Modularity : example from HYDROJULES model

Issues in the exchanges between modules are mainly to do with **changes of time and space**.

➔ Clustering and solving the process and feedback representation within a module at the same time/space.



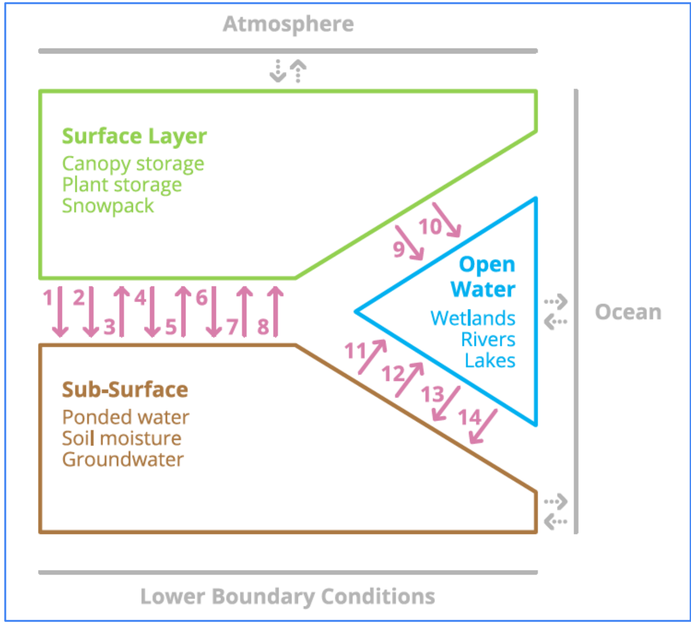
Courtesy from Eleanor B., Simon D., Jan P., ...

Modularity : example from Hydro-JULES community

UniFHy v0.1.1: A community modelling framework for the terrestrial water cycle in Python

Thibault Hallouin[1,2], Richard J. Ellis[3], Douglas B. Clark[3], Simon J. Dadson[3,4], Andrew G. Hughes[5], Bryan N. Lawrence[1,2,6], Grenville M. S. Lister[1,2], and Jan Polcher[7]

- UNIFHY - represents the water cycle as **3 components** each with **distinct time and space resolutions**.
- The coupling framework calculates the exchanges of the variables. Tthe modules can then have any level of complexity.



Courtesy from Eleanor B., Simon D., Jan P., ...

Modularity : example from ICON-Land model

- Integration of **concurrent process** & **surface descriptions** in a flexible way

- **Separate the infrastructure** required to implement physical, bio-geophysical and bio-geochemical land processes **from concrete process implementations accessed by abstract interfaces (Separation of Concerns)**

Flexible, efficient & sustainable software development, maintenance and use

JSBACH is just one (!) concrete implementation of land processes using the ICON-Land framework; ➔ SEE a dedicated poster !

➔ Make use of "object oriented" modelling

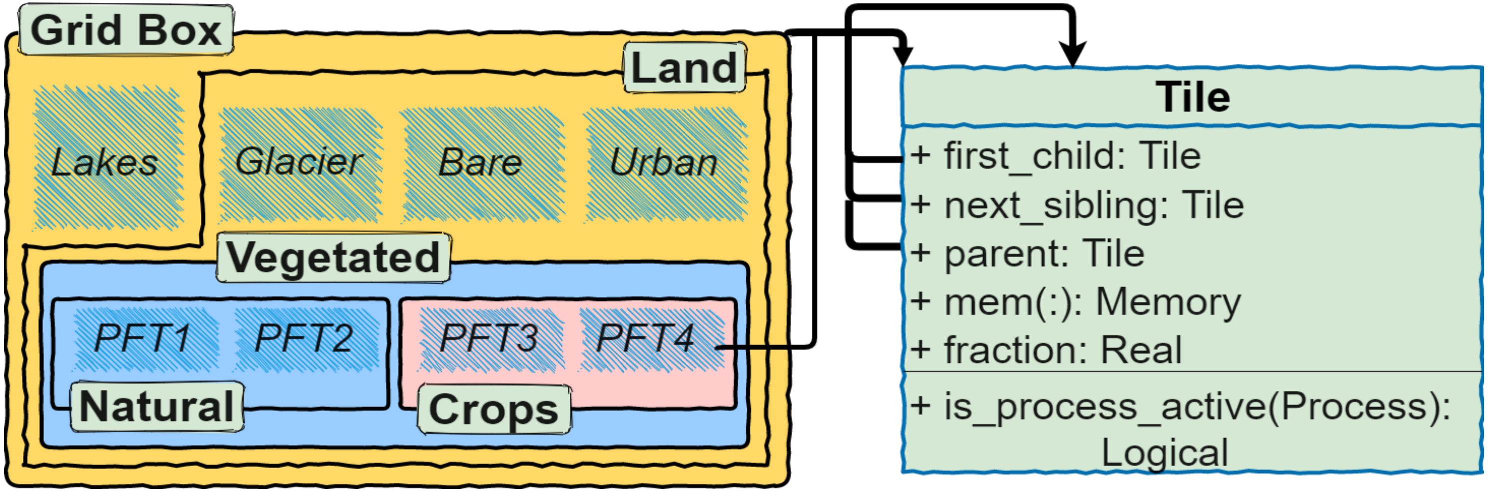
Courtesy from Reiner Schnur...

Modularity : example from ICON-Land model

ICON-LAND FRAMEWORK

Courtesy from Reiner Schnur...

Surface heterogeneity ↔ tiles



Example of surface cover (left) and simplified class diagram for hierarchical tile structure (right)

- Each grid box and its sub-scale cover types are represented by an instance of the ***Tile*** class
- Each tile instance has a cover fraction and instances of the ***Memory*** class for each process (variables)
- Different processes can run on different tile sub-sets (yellow, blue and pink areas)

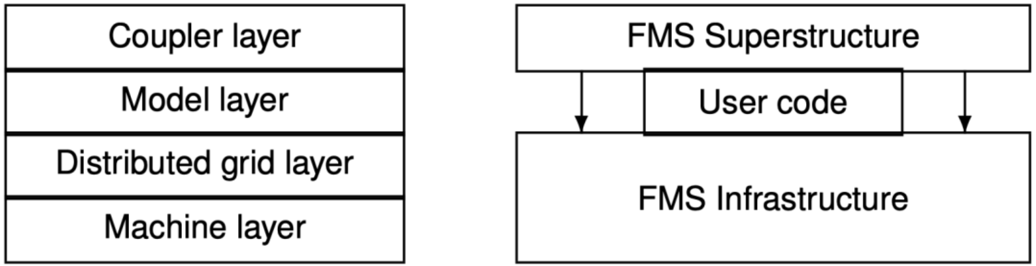
Modularity : example from GFDL group

➔ Try to develop for ESM parts a Flexible Modelling System (FMS)

- Specific work on Interface & I/O
- Strives for Modularity for LSM
- Highly linked to parallelisation and computer structure.

➔ But modularity is tightly linked to key choice of each modeling group !

FMS is in its second decade of active use



- Flexible Modeling System effort began in 1998, when GFDL first moved on to distributed memory machines
- Provided simplified interface to parallelism and I/O: `mpp`. Abstract types for "distributed grid" and "fields".
- Diagnostic output, data override, time manager.
- Component-based design, abstract types for component state vectors, exchange grid.
- "Sandwich" model influential in community.

Courtesy from Elena Schevliakova,...

Modularity : some working directions for CLM

- **Model decomposition into cascade of processes** (ex. Demographie):

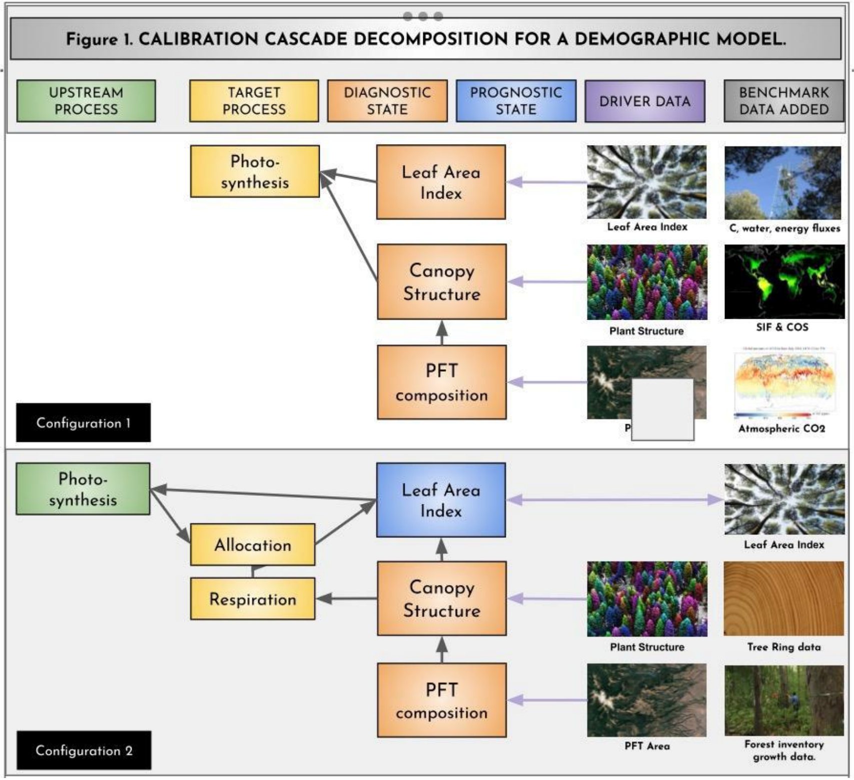
Photosynthesis

Allocation & Respiration

Mortality & Recruitment

Competition

Model calibration at
each step using “data-driven”
input variables or previous step model outputs
(**Calibration Cascade**)

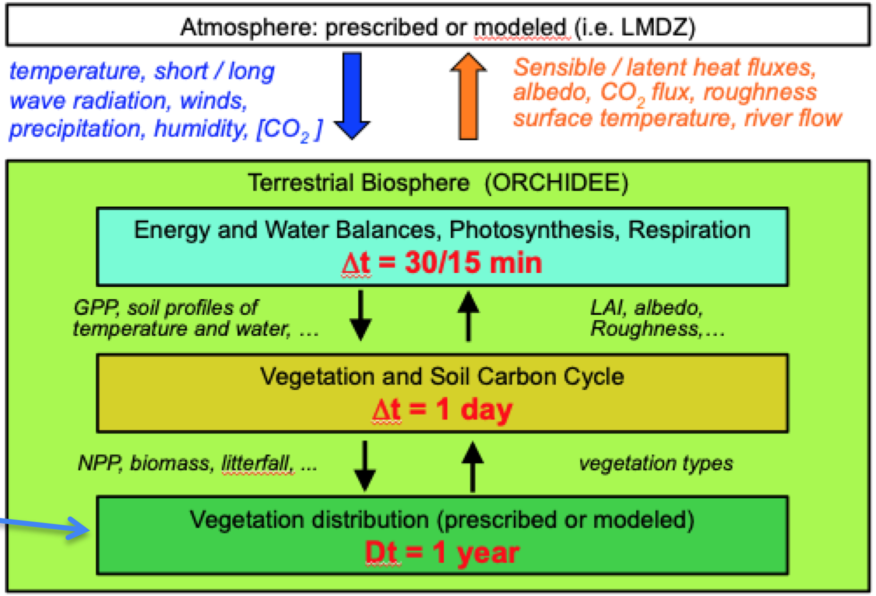


Courtesy from Rosie F., Dave Lawrence, Bill Sacks,...

Modularity : example from ORCHIDEE model

- Energy and Water are grouped (with a short time step)
- Vegetation and soil Carbon (daily time step)
- Vegetation distribution (biogeography) (annual time step)

From LPJ

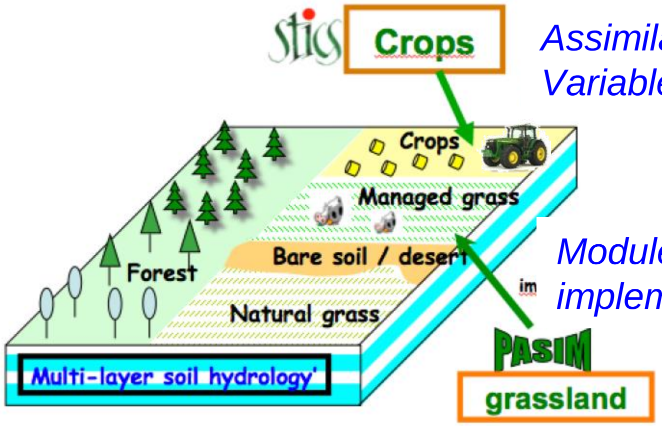


➔ Structure need to be revised given the increasing number of processes linking Carbon – Water – Energy

Modularity : Recent coupling with existing « modules »

➔ We have used modules developed for other models:

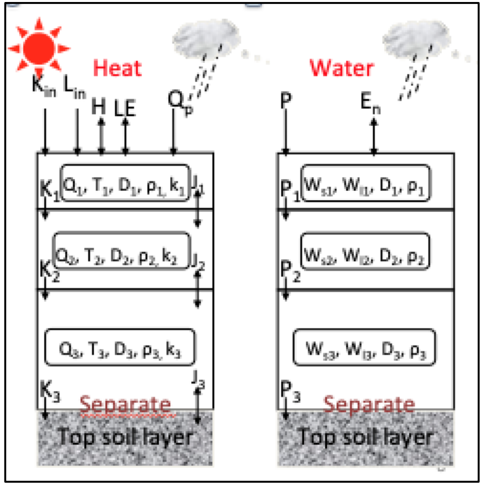
Crop / Grass models from STICS / PASIM



Assimilation of Variables

Modules implementation

3 layers snow model from ISBA-ES



➔ Substantial effort was needed !

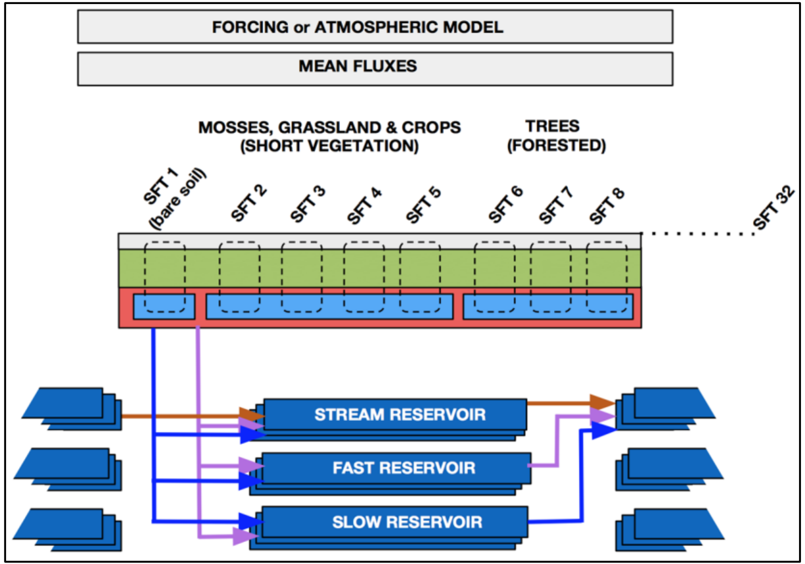
Modularity : example from ORCHIDEE model

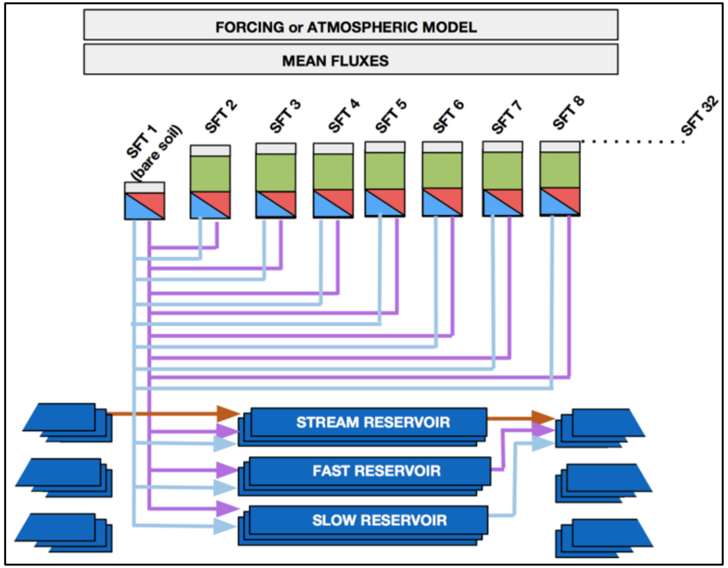
- Treatment of sub-grid heterogeneity is crucial and model specific !!

Nsft C budgets
3 water budgets
1 energy budget
1 atm. column

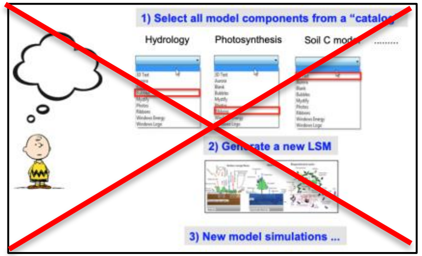
Increase flexibility with
Surface Functional Types (SFT)
= f(soil, land cover, topo.,)

Nsft C budgets
Nsft water budgets
Nsft energy budgets
>1 Atm. Column





Summary / potential way forward...



- “Plug and play” is not realistic
- Defining Common Modularity structure **is a complex challenge** !
  - Agreeing on standards will be a huge social effort !
- Moving toward **Object oriented** programming would help !
  - But not easy for standard modellers !
- **But at least we can progress in several directions:**
  - Defining **standards for agreed Interfaces** (principles, Var. names, units, ...) ex. Coupling with atmosphere (Polcher et al., 1998)
  - **“Technical” modules** may be easier to deal with: Input / Output ;
  - **Sharing expertise** / innovation between groups will help
  - **Try for specific “ common modules”** between a few groups ?

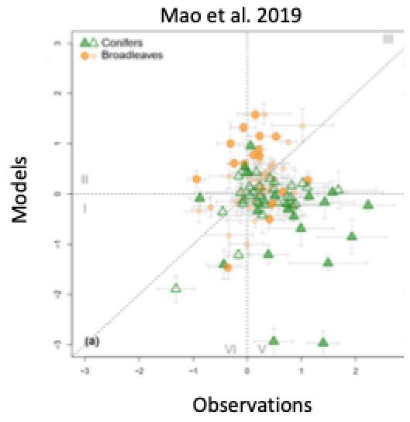
Summary / potential way forward...

- **Potential candidates**
  - Soil Organic Matter dynamic !
  - Water routing scheme
  - Soil water & energy budget
  - Fires (i.e. SpitFire) : But impact on Plant depend on model demography
  - Snow dynamics
  - Model demography ?
  - ….

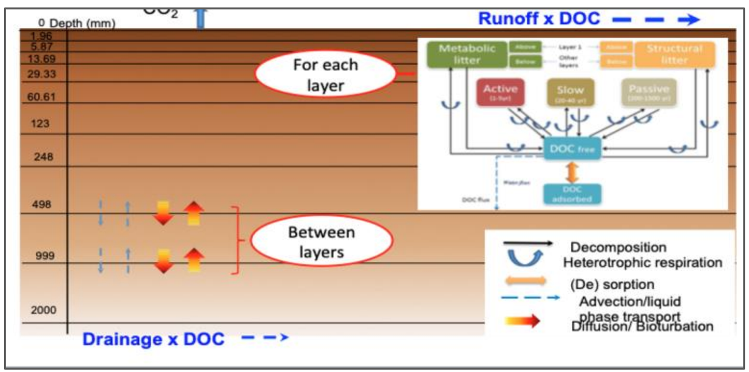
Summary / potential way forward...

- **Soil Organic Matter Dynamic module ?**

- Current model still poorly represented soil carbon stock variations !

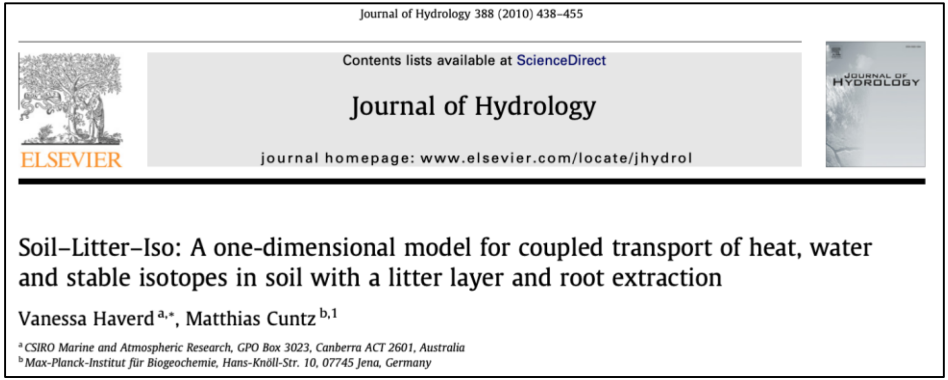


- Need to test different "modeling strategies"
- Interface with litter inputs well defined !
- Outputs could be also standardized !
- But feedbacks between Nutrients availability and plant functionning – litter production will be model specific !



Summary / potential way forward...

- **Soil Water and Energy budget**
  Existing attempt
  (Haverd & Cuntz, 2010)


Journal of Hydrology 388 (2010) 438–455
Contents lists available at ScienceDirect
Journal of Hydrology
journal homepage: www.elsevier.com/locate/jhydrol

Soil–Litter–Iso: A one-dimensional model for coupled transport of heat, water and stable isotopes in soil with a litter layer and root extraction

Vanessa Haverd[a,*], Matthias Cuntz[b,1]
[a] CSIRO Marine and Atmospheric Research, GPO Box 3023, Canberra ACT 2601, Australia
[b] Max-Planck-Institut für Biogeochemie, Hans-Knöll-Str. 10, 07745 Jena, Germany

- **Water routing + heat & C/N coumponds transport !**

OR

- External approach
  (TRIP, CTRIP, CamaFlood,...)
  Easy to "modularize" using Interpolation of runoff / drainage

- Routing within the LSM grid: with sub-grid hydrological transfer units
  More complex to Modularize !