

# An overview of software platforms and tools for model integration, collaborative development and resource sharing

S. Dufour

UMR AMAP

23/10/2009

# Plan

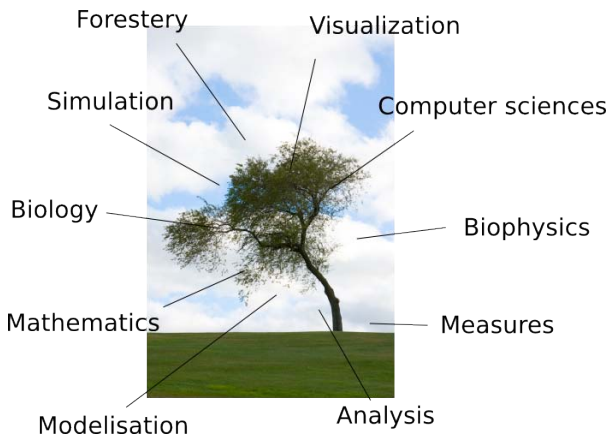
- 1 Software platform challenges
- 2 Modeling approaches
- 3 Tools to co-develop
- 4 Community
- 5 Conclusion

# Software platform challenges

...

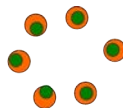
# Pluri-disciplinary approach

Plant/Tree modeling is based on different domains  
⇒ Need of collaboration



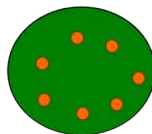
# Modeling strategy

- 1 Construct the best model (efficient & simple) for each new situation

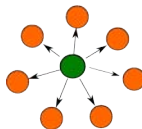


● Model  
● Question

- 2 A general unified model



- 3 Defining common phenomenon, concepts and methods :
  - Common to different situations
  - Extensible



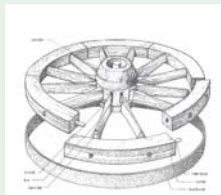
# Development method

## Experimentation vs Production

- Research model  $\neq$  Industrial products  
⇒ models should be usable and modifiable at the same time
- Adapt development methodology

## Limited development resources

- The same tools/libraries are often re-develop
- **Do not reinvent the wheel**, but reuse existing tools
- Exemple : the 3D viewer



# Platform challenges

## Goals

- Rapid and **easy development** of model
- **Reuse** softwares & tools
- **Share** development and exchange experiences
- **Synergy** in the community
- Ensure **software quality** (portability, installation, support...)
- Improve **diffusion** and valorisation

## Expected results

- Perenity
- "Standardisation"
- Cost reduction

# Modeling approaches

...



# Type of platform

**Platform = an approach + organisation**

## Different approach

- Generic model
- Data structure
- Methodology
- Software libraries
- Component integration
- Simulation formalism

## Different organisation

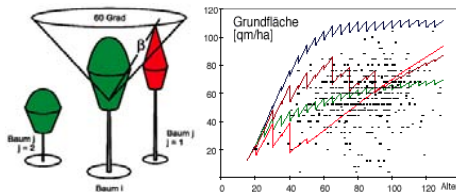
- Centralized vs Distributed development
- One developer or modeler vs a Community
- Open vs Closed license

# Around a model

**Model** : Sylva

**Theme** : Forestry

- **Forest growth simulator**
- **Multi-species** : *Picea abies*, *Abies alba*, *Pinus sylvestris*, *Fagus sylvatica*, *Sessile Oak*, *Quercus petraea*, *Alnus glutinosa*
- **Flexible** : simulate diverse intermingling and structured stands, as well as tending regimes and regeneration methods.



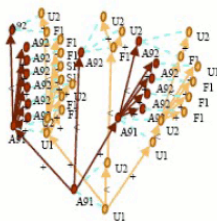
# Data Structure

**Platform** : AmapMod

**Data Structure** : MTG (Multi Scale Tree graph)

**Theme** : Plant architecture analysis

- Query Language (AML) + Statistic tools + Analysis tools



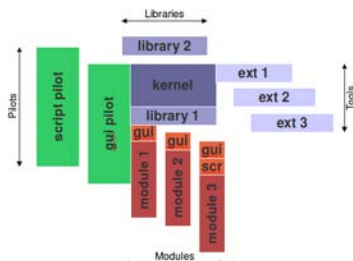
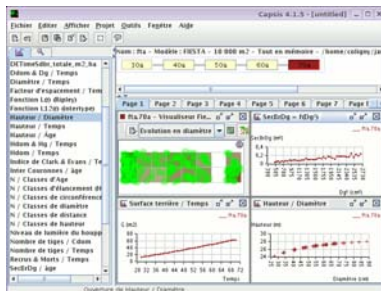
Godin, C., et al., 1999. Exploration of plant architecture databases with the AMAPmod software illustrated on an apple-tree bybird family. *Agronomie*, 19(3/4) : 163-184.

# Methodology

**Platform** : Capsis

**Theme** : Forestry growth

- **Multi-Model** : stand level, individual, spatialized, ...
- **Software architecture** : kernel, models, plugins , ...
- Scenario and intervention
- Common **charter**, common methodology

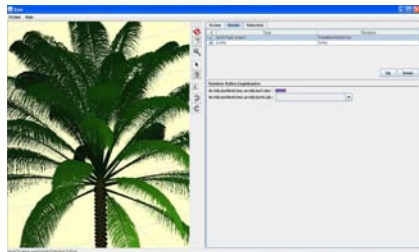
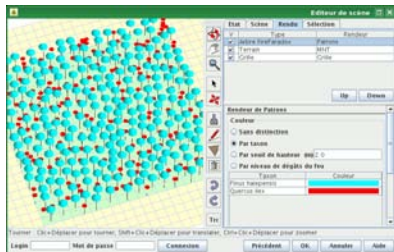


# Set of libraries

**Platform** : Jeeb

**Theme** : plant architecture modeling

- A set of **shared librairies** and application
- **Common tools**, data structure, algo (ligth, sketching...)...
- Plugin architecture, shared code
- Dedicated applications (Xplo, Simeo...)







# Tools to co-develop

...



# Project management (Forge)

## Share information

- Bugs
- Features
- Roadmap
- Technical documentation
- Activity

The screenshot shows the Forge project management interface for the 'Capsis' project. The page is divided into several sections:

- Overview:** Features a 'Capsis' logo and a description: 'Computer Aided Prediction of Strategies in Bioscience'. It includes a 'Homepage' link and a 'Download' link.
- Issue tracking:** Lists '25 open on 18' bugs, '11 open on 23' features, and '0 open on 2' support requests. A 'View all issues' link is provided.
- Members:** Lists the project manager (François de Coligny) and several developers, including Antoine Besson, Céline Michéaux, and others.
- Latest news:** Announces 'Capsis version 4.1.7' and 'Capsis version 4.1.7 roadmap', both released by François de Coligny.

The interface includes a navigation menu at the top (Overview, Activity, Roadmap, Issues, New Issue, News, Wiki, Responding, Settings) and a search bar. The footer contains the text 'Powered by Redmine & © 2014-2015 Jean-François de Coligny'.

# Sharing code

## Versioning system

- All developer are **synchronised**
- All changes are **recorded**
- ⇒ Improve traceability, error recovery, bug finding...
- Exemple : SVN, Mercurial, Git, Bazaar...

The screenshot shows the Capsis web interface. At the top, there are navigation tabs: Overview, Activity, Roadmap, Issues, New Issue, News, Wiki, Repository, and Settings. The 'Repository (Subversion)' section is active, displaying a table of repository items. Below this, the 'Latest revisions' section shows a list of recent commits with columns for ID, Date, Author, and Comment.

Name	Size	Revisions	Age	Author	Comment
branches		815	25 days	Sébastien Dufour-Roσσas	add branches directory
branches		523	2 days	Sébastien Dufour-Roσσas	* Add a description label in new scenarios dialog...
branches		236	74 days	François de Colligny	add eclipse config files
branches		523	3 days	Sébastien Dufour-Roσσas	* Add a description label in new scenario dialog...
branches		515	4 days	François de Colligny	updated comments in request form for Page editor
branches		495	8 days	Sébastien Dufour-Roσσas	* New refreshing build generation... (1) New et...
branches		508	5 days	François de Colligny	new tool de off in Project - new description
branches		524	3 days	Sébastien Dufour-Roσσas	update arguments of new scenario dialog (feature #44)
branches		508	5 days	Sébastien Dufour-Roσσas	Remove unnecessary for scenario generator
branches		508	5 days	Sébastien Dufour-Roσσas	add project and -diagnose capsis files
branches		512	45 days	author	Member help user identifier and url instead...
branches		515	8 days	Sébastien Dufour-Roσσas	* New refreshing build generation... (1) New et...
branches		508	20 days	Sébastien Dufour-Roσσas	update build and build tool with complete help fe...

ID	Date	Author	Comment
523	2020-02-09 09:28 PM	Sébastien Dufour-Roσσas	* Add a description label in new scenarios dialog * Add a description field in dialog
522	2020-02-09 08:26 PM	Sébastien Dufour-Roσσas	improve arguments of new scenario dialog (feature #44)
521	2020-02-09 08:07 PM	Sébastien Dufour-Roσσas	the help dialog
520	2020-02-09 12:47 PM	François de Colligny	add a warning if not enough space to place more (highly processor !!!)
519	2020-02-09 08:54 AM	François de Colligny	some more comments - a cleanup
518	2020-02-09 08:54 AM	François de Colligny	2 connectors - a scenario comment
517	2020-02-09 03:44 PM	François de Colligny	Added a step, still under development, must be considered in context, relative balance algorithms and collections
516	2020-02-09 12:54 AM	François de Colligny	Added missing Agreements property files
515	2020-02-09 10:52 AM	François de Colligny	Updated comments in request form for Page editor
514	2020-02-09 12:02 AM	Sébastien Dufour-Roσσas	Change description label of 927 toolbar

# Build / Deployment

## Developer task should be automated and **standardised**

- Build complex and large projects (ant, cmake, scons...)
- Manage dependencies between subprojects (maven, ivy, ...)
- Portability

## Deployment, diffusion and update

- Installer creation
- Web diffusion
- Package Manager (ex : apt)



# Software quality : Providing tests

## Fonctional Testing

- *Outputs* are correct for a given *input*
- Tests are automated  $\Rightarrow$  test reports
- Refactoring & improvements are possible

## Continuous integration

- Model are tested continuously on a server
- Ensure validity **after modifications**
- Detect indirect errors (integration problem)





# Community

...

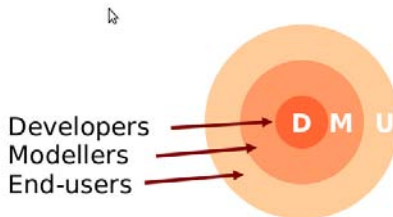
# Roles

## Standard model

- **Developer** : create & diffuse the software
- **User** : install & use the software

## Scientific platform model (cf Capsis)

- **Developer** : develop platform kernel and shared libraries
- **Modeler** : develop models, reused shared tools
- **User** : use the model with its own data



# Charter (Capsis)

- **Free kernel** (free license)
- **Modelers are responsible** of their model and their validation
- They can have **support from developers** : lib, training, tools...
- Models and source code are **accessible** in the modeler community
- Respect of **intellectual property** : License / Copyright
- Distribution & **Decentralisation** : modeler can distribute their own models and provide support on them
- ...



# Methodology : Agile development

## The Agile Manifesto [Fowler, Highsmith - 2001]

### Principles

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Collaboration** over contract negotiation
- **Responding to change** over following a plan



### In practice

- Small iteration & Immediate feedback
- Coding / Modeling meetings
- Good practice (code review...)
- **Keep it simple !**

# License / Open Source

- Published models are accessible  $\Rightarrow$  verification
- Code is perennial, models & lib can be reused and improved...
- **Open source is not public domain (Copyright)**

## Strategy

- Non viral Free license for core modules and shared libs (LGPL like)
- Open or commercial or dual licensing for models



# Conclusion

## A software platform ?

- **Common framework** / paradigm / data structure...
- **Shared** tools and libraries (mutualisation & reuse)
- Developer / Modeler / User **community**
- Common **rules** (charter, license...)

