OpenAlea

A platform for plant modelling, analysis and simulation

C. Pradal, F. Boudon, N. Donès, JB Durand, P. Barbier de Reuille, C. Fournier, H. Sinoquet, C. Godin

Virtual Plants
Plant modelling

Biological objects

Measure

Modelling

Simulation
Pluri-disciplinary research

- Visualisation
- Biophysics
- Ecophysiology
- Statistical analysis
- Forestry
- Modelling
- Computer science
- Simulation
- Software engineering
- ...

...
Modelling strategy

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

2. **A general unified model**

3. **Defining common phenomenon, concepts and methods:**
   - Common to different situations
   - Extensible
Python as a modelling language [Sanner 06]

Python as a scripting environment

Python as a component framework

- 3D scene graph
- Visualisation
- Graph Data structure
- Plant models

Python as a software Bus
OpenAlea

**ALEA**: *Atelier Logiciel en Ecophysioologie et en Architecture 3D des plantes*
- use, evaluate and build experimental models

**Users**
- biologists and modellers

**Components**
- various tools and models for studying plant architecture and its development

**Partners** (French research institutes)
- INRA, CIRAD, INRIA, LABRI, INPG.
Objectives

**Share knowledge**
- Reuse softwares & tools
- Share development between various team
- Exchange experience & data
- Share training effort

**Component based software architecture**
- Integration of existing softwares & tools
- Rapid development of new models
- Quality rules
OpenAlea community

**Designers**
- kernel development
- modellers counsel, training and assistance

**Modellers**
- Models and tools development and integration
- users counsel, training and assistance

**Users**
- define scenario and provide feedback

**Open Source Community**
- Free kernel licence: LGPL
- Distribution rules
Architecture

- Users
  - Shell
  - Interface
    - GUI
- Designers
  - Python Language
  - Kernel
    - Generic data structures
      - MTG
      - Trees
      - Sequences
  - Wrappers (C, C++, Fortran)
- Modelers
  - Python packages
    - Scientific package
      - Python packages
        - Python Community
      - New models
        - Python
        - C, C++, ...
      - Existing tools
        - C, C++
        - Fortran
        - ...
  - Analysts, eco-physio, ...
  - Python Community
  - Simulation, eco-physio, ...
  - Analyse, eco-physio, ...

- Python Community
  - Interface
    - Shell
    - GUI
  - Python packages
    - Scientific package
      - Python packages
Data structures

Common data structures
- Sequences
- Graph (Topology)
  - Tree
  - Directed graph
  - Hierarchical tree (MTG)
- Scene graph (Geometry)
  - Hierarchical scene graph

Interfaces and adaptors
- Zope.Interface
- Well defined interfaces
- Different implementations
Multiscale representation of trees

Multiscale Tree Graphs (MTG)

(Godin, Caraglio 1998)
Development tools

**OS**
- Linux and Windows

**Languages**
- C, C++, Fortran and Python

**Wrappers**
- Swig, Boost.Python and f2py

**GUI**
- PyQT

**Test**
- py.test (Python) and Boost.Test (C++)

**Documentation**
- epydoc and doxygen
Building and Installing

**Scons**
- Build C, C++ components and their wrappers

**Scons_utils**
  - Extend `scons` for OpenAlea developers
  - Hide the complexity of the build system
  - Default options for each tools (Linux, Windows):
    - opengl, boost.python, qt, bison, flex, compilers, ...
  - Unify the build process for the different packages
    - `ALEALibrary('mylib','*.cpp')`
    - `ALEAInclude('mylib','*.h')`
    - `ALEAWrapper('mywrapper','*.cpp')`
    - `ALEAProgram('prog','*.cpp')`

**Packaging**
- Extend distutils with scons as a build system
Working together

Collaborative development
- gforge
- Subversion
- Wiki

Coding sprint
- Pair programming
- Specification and development

Training
- Elearning platform for modelers and users
  - Python
  - 3D Plant Architecture
- France, Thailand and Brasil
Components

**AMAPmod (Godin, Guédon et al.)**
- Analysis of plant architecture

**PlantGL (Boudon, Pradal et al.)**
- Plant Geometric Library & 3D viewer

**TreeAnalysis (Durand et al.)**
- Hidden-Markov Tree Models

**RATP (Sinoquet et al.)**
- Radiative transfer, transpiration and photosynthesis

**Archimed (Dauzat et al.)**
- Modelling biophysical processes on 3D plant models

**PyCaRiBu (Fournier, Chelles et al.)**
- Simulation and radiative transfer (Maize and wheat)

**Merrysim (Barbier de Reuille et al.)**
- Simulation of virtual meristems
AMAPmod – AML/C++

Plant → Code → MTG

Observation Digitalization Python

Extraction Models & tools

Form ... Classes ... Decompos ... Topology ... Features ... Code ...
/P1/U1 +U1 ...

10110000101 00110011101 11101101111 ...

40034010 4446132101133 2300141111
PlantGL – C++

Meristem simulator (P. Barbier de Reuille)

Python

Hemispheric view (C. Parveaud)

Geometry
  - Group
  - Primitive
  - Transformed
    - Curve
    - Surface
    - Volume
    - Planar Model
      - SOR
      - Box
      - Extrusion
      - Hull
      - ...
Biological objects at different scales
TreeAnalysis – C++

Hidden-Markov Tree model

- Long, Medium & short Growth Unit (GU)
- Alternation vegetative & flowering GU.

R.A.T.P. – Fortran

Radiation Absorption, Transpiration, Photosynthesis

- **Structure**: LAD, inclinaisons
- **Radiation transfer**: leaf irradiance and light regime, leaf N content, gsmax, Vcmax, Jmax
- **Wind Speed**: Boundary layer conductance
- **Energy Balance**: stomatal conductance, transpiration, leaf temperature
- **Photosynthesis**: net CO₂ assimilation
Modelling biophysical processes on 3D plant models

- reflectance
- irradiation
- light transmission
- photosynthesis
- energy budget
- sap flow
- transpiration
PyCaRiBu – C++

Maize simulation

- Radiative transfer
- Geometric Interpreter
- Plant Simulation (LSystem)
- Parameters Estimation
Conclusion

Toward a visual programming environment?

➢ Vision [Sanner et al. 02]
➢ Orange [Demsar et al., 04]
➢ TraitsUI [anvisage, enthought]

Python as a modelling language
- Easy to learn, even for botanists.

Python as a software bus
- Glue together Fortran, C and C++
- Large scientific community
Acknowledgements

- **OpenAlea community**
  INRIA Sophia (VirtualPlants),
  UMR AMAP (Montpellier),
  UMR PIAF (Clermont-Ferrand),
  UMR EPC (Paris-Grignon),
  Labri (Bordeaux),
  Joseph Fourrier University (Grenoble)
  UMR SAGAH (Angers),
  UMR BEPC (Montpellier),
  CEMAGREF (Clermont-Ferrand),
  INRA (Orléans), INRA (Lusignan), INRA (Nancy)

- **Guido von Rossum & Python community**