Building modular FSPM under OpenAlea: concepts and applications

Christian Fournier, Christophe Pradal, Gaëtan Louarn, Didier Combes, Jean-Christophe Soulié, Delphine Luquet, Frédéric Boudon, Michaël Chelle

Building modular FSPM under OpenAlea: concepts and applications

Christian Fournier[1*], Christophe Pradal[2*], Gaëtan Louarn[3], Didier Combes[3], Jean-Christophe Soulié[4], Delphine Luquet[4], Frédéric Boudon[2] and Michaël Chelle[5]

[1]INRA, UMR 759 LEPSE, 2 place Viala, 34060 Montpellier cedex 01, France. [2]CIRAD, INRIA, EPI Virtual Plant, UMR DAP, Avenue Agropolis, 34398 Montpellier Cedex 5, France. [3]INRA, UR4P3F, F-86600 Lusignan, France. [4]CIRAD, UPR104 AIVA, Avenue Agropolis, 34398 Montpellier Cedex 5, France. [5]INRA, UMR1091 EGC, F-78850 Thiverval-Grignon, France.

[*]These authors equally contribute to the work.

Corresponding author: Christian.Fournier@supagro.inra.fr

Keywords: FSPM, modularity, Wheat, Maize, rice, vine, simulation, plant architecture, phylloclimate

Introduction

The OpenAlea platform (Pradal *et al.,* 2008) was designed to facilitate the integration and inter-operability of heterogeneous models to get comprehensive FSPMs. It relies on Python gluing capabilities, that allow non intrusive integration of programs written in various languages (Fortran, C, C++, R, L-system); and on the dataflow computing paradigm, that promotes decomposition of applications into independent components that can be recombined dynamically into customized workflows. Still, a plugable collection of components is not by itself a solution to the modularity problem in FSPM modeling. First, heterogeneities between components inputs and outputs can lead to exponential needs for specific adaptors and converters to get functional assemblies. Second, several ways exist to decompose models into independent components. This can lead to incompatibilities or difficulties for re-assembly into comprehensive models. Last, users of the platform may find difficult to build applications, without some knowledge on how a simulation has to be reasoned within the data-flow computing paradigm. Here, we propose a modeling strategy to help for building coherent, yet modular FSPM under OpenAlea. We first define the key concepts of this strategy, illustrate how they can be used under Visualea and how it lead to a first set of reusable components resulting from various ecophysiological studies.

Modeling strategy

The modeling strategy is to: (i) decompose the application into sub processes of different type (see below); (ii) use a data structure as a central pivot for the computations, and (iii) delegate the control of the simulation to the dataflow. It uses the following functional entities:

- The Multi-scale Tree Graph (*MTG,* Godin and Caraglio, 1998) encodes the plant architecture (topology and geometry) at different scales, and with any type of properties characterizing plant entities (*e.g.* internal state, phylloclimate, mechanical stress). This central data-structure is created and updated by the other components that will either modify the topological structure (*topological builder*), compute a geometric representation (*geometric solver*) and add or modify the properties associated to the topological elements (*process models*). The MTG implementation currently available in OpenAlea allows for such an encoding, and provides facilities for conversion into similar data structure, *e.g.* bracketed string of L-system.

- The *topological builder* is defined as an operator that transforms the topology of the MTG using graph rewriting operations. Various implementations exists: *L-studio-VLab/cpfg-lpfg* (Prusinkiewicz, 2004), *GroIMP/XL* (Kniemeyer, 2008) or *L-Py* (Boudon, 2010). It may use parameters computed by the different *process models* and the topological information.

- The *geometric solver* computes the geometry from the MTG. It starts from partial information contained in the MTG, i.e. topology, specific properties (*e.g.* position, length, *etc.*) and user defined constraints. And it completes them by computing geometric information for all the entities of the plant. Two types of solver exist. The first one relies on relative information i.e. for instance the position and orientation of an entity are defined relatively to its parent. In this case, Turtle geometry can be used to resolve the system. The second type uses global positioning information. PlantFrame (Pradalet al., 2009) may be used in this case to complete missing information. Resulting geometry is stored in the MTG, to ensure the correspondence between MTG entities and their geometric counterparts.
- The *process models* compute properties associated to topological elements on a static structure based on partial information (topology, geometry and properties) contained in the MTG, or provided by the user. For instance, this category includes biological models that compute fluxes within the plant or environmental models that compute phylloclimatic variables.
- The *control manager* coordinates and schedules the different components. In OpenAlea, the control manager is a dataflow, which is build and executed through the graphical environment, Visualea. A 'for' node allow to process the MTG using a dataflow several times to model feedback.

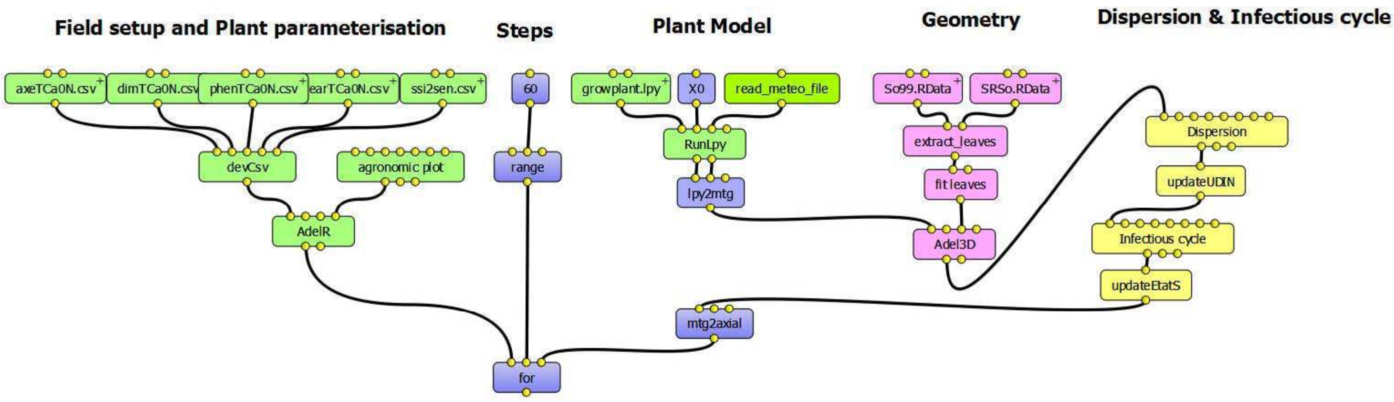

Figure 1: A simulation of plant-disease interactions build under Visualea, based upon the modeling strategy defined here. The different colors indicates the category the component refers to: topological builder (green), geometric solvers (pink), and process models (yellow). The data circulating between boxes is an MTG, that is transformed into an L-system string before and after each L-system iteration (blue boxes). The control manager is the dataflow, which define the order of computation between the modules. At the bottom, the 'for' node make it possible to apply the right part the dataflow on the MTG several times.

Illustration under Visualea

Figure 1 illustrates how this strategy is applied to an FSPM that simulate the interactions between a pathogen and wheat plants (Septo 3D Robert *et al.*, 2008). A small set of young plants (1 meter square) is first build from experimental data as an MTG (the axiom). The reconstructed structure enters into a loop where it is iteratively modified by an L-system encoding for plant development (addition of new organs + growth of existing ones). Then, a geometric solver generates a 3D reconstruction of the plot according to the MTG and its properties (such as spores produced on leaves). A physical model (process model) estimates the dispersal of pathogen spores on the 3D structure and updates the property encoding for the amount of spores deposited on leaves. A second process model simulates infectious cycle of lesions in leaves. It updates lesion deposit and reduction of green area of organs.

Application to the share of components in ecophysiology

Another benefit of the strategy was to enhance the share of models between a small community of ecophysiologists. All had interests in reusing parts of an integrated model implemented on L-studio, that computes Graminae 3D development (Adel, Fournier 2003) and light distributions on plant organs (Caribu, Chelle 1998, 2004). New contexts for reuse for these models were : (i) to complete a rice FSPM model that lacked of 3D geometric representation and 3D light computation (Ecomeristem, Luquet et al, 2008), to couple Caribu with (ii) an L-system model of grass development built under Lpy (Graminel, Verdenal *et al.*, 2008) , (iii) a 3D vine reconstruction model (Topvine, Louarn et al., 2008) and a photosynthesis model (Farquhar et al., 1980), and (iv) digitisation acquisition procedures (for wheat, maize, and vine). A first solution was to develop specific code for each models. However, such a solution was not satisfying as it lead to multiple implementation of similar utilities (e.g. the aggregation of Caribu output by organs) and was not scalable (e.g. inclusion of new rules in ADEL for compatibility with rice).

A better solution was found using the concepts presented above. First, Caribu was enriched with small utilities to become a process model that accept an MTG as input and returns it with new properties as output. These utilities allow for user-specification of optical properties and of aggregation routines that tranform irradiance values computed at the level of geometrical shapes into irradiances at the level of MTG nodes. Using MTG traversal routines, light distribution was further integrated (*e.g.* at plant or canopy level) whenever needed. Second, Adel was decomposed into a topological builder able to simulate plant development in terms of topology and dimension of organs, and into a geometric interpreter that takes an MTG as input with properties describing organ length. As such, the reconstruction tool had easily been extended to handle rice reconstruction. Finally, generic conversion routines were set up between outputs of an L-system and MTG. These include a conversion from L-system string to MTG (with possibility to define modules coding for scales) and a tool to associate L-system geometric output to the mtg. As such, L-systems program can be used either as topological builders, geometric solver or both. These new components were used to recompose the original application (Adel +Caribu), and develop the new ones.

Ecomeritem becomes a repetitive loop that performs sequentially the following operations: *i)* Ecomeritem setup, one day simulation, and production of the topology (XML file) converted into a MTG, *ii)* build of the 3D representation from the MTG, using Adel geometrical model *iii)* energy balance computation using Caribu and aggregation at plant level, and *iv)* new EcoMeristem setup using the values defined at the previous step, and one day simulation. Graminel was directly implemented with tools allowing conversion from L-system to MTG and vice-versa. For Topvine, who already outputs an MTG, the coupling was straightforward, and consists mainly in harmonizing naming convention. The coupling with the photosynthesis model was implemented as a new process model that converts an MTG with caribu output properties and photosynthetic parameters into carbon gains. A similar approach was used to couple with MTG coming from digitization

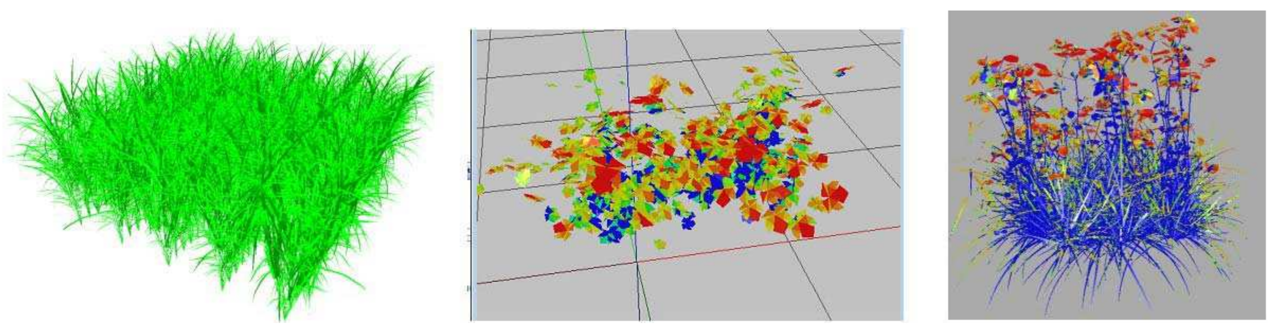

Figure 2 :Illustrations of applications based on the same set of components and that followed a common strategy. (Left) rice reconstruction with Adel coupled to Ecomeristem, (Middle), light distribution on a vine (TopVine + Caribu), (Right) Grasses growing under alfalfa and competing for light (Graminel, MTG reconstruction and Caribu).

Conclusion

This paper presented how modular FSPMs could be build under OpenAlea. It uses a functional decomposition which is closed to those implemented in other formalism, like the trio “production rules”, “homomorphism”, “external models” in the Open L-System strategy. However, it is more general because none of these entities is defined within a given formalism (for example, topological builders can be L-systems, graph rewriting systems or regular procedural code). This flexibility, associated with the integration capabilities of OpenAlea proved very efficient to share models at the scale of a small community of modelers.

Acknowledgements

This project is supported by Agropolis Fondationand by ARC INRA-INRIA Alinea program

References

Boudon F., Cokaeler T., Pradal C., Godin C., L-Py, an open L-systems framework in Python, pending publication.

Chelle, M., Andrieu, B. and Bouatouch, K., 1998. Nested Radiosity for Plant Canopies. The Visual Computer, 14: 109--125.

Chelle, M., Hanan, J. and Autret, H., 2004. Lighting virtual crops : the CARIBU solution for open L-systems. In: C. Godin et al. (Editors), 4th International Workshop on Functional-Structural Plant Models, Montpellier, pp. 194.

Farquhar GD, von Caemmerer S, Berry J. (1980) A biochemical model of photosynthetic CO_2 assimilation in leaves of C_3 species. *Planta* 149: 78-90.

Fournier C., Andrieu B., Ljutovac S. & Saint-Jean S. (2003) ADEL-wheat: a 3D architectural model of wheat development. In: *Plant Growth Modeling and Applications* (eds B.-G. Hu & M. Jaeger), pp. 54-66. Tsinghua University Press and Springer, Beijing, China.

Godin, C. et Y. Caraglio. 1998. A multiscale model of plant topological structures. *Journal of theoretical Biology* 191: 1-46.

Kniemayer O. (2008) Design and implementation of a graph grammar based language for functional-structural plant modelling. *PhD Thesis*, BTU Cottbus.

Louarn G, Lecoeur J, Lebon E. (2008) A 3D statistical reconstruction model of grapevine (*Vitis vinifera*) simulating canopy structure variability within and between cultivar/training system pairs. *Annals of Botany* 101: 1167–1184.

Luquet D, Dingkuhn M, Kim HK, Tambour L, Clément-Vidal A (2006) EcoMeristem, a Model of Morphogenesis and Competition among Sinks in Rice. 1. Concept, Validation and Sensitivity analysis. *Functional Plant Biology* **33**, 309-323.

Pradal C., Dufour-Kowalski S., Boudon F., Fournier C. & Godin C. (2008) OpenAlea: a visual programming and component-based software platform for plant modelling. *Functional Plant Biology*, **35**, 751-760.

Pradal C, Boudon F, Nouguier C, Chopard J & Goding C (2009) PlantGL : a Python-based geometric library for 3D plant modelling at different scales. *Graphical Models* **71**, 1-21.

Prusinkiewicz P. (2004) Art and science for life: Designing and growing virtual plants with L-systems. *Acta Horticulturae*, **630**, 15-28.

Robert C., Fournier, C., Andrieu, B., Ney, B. (2008) Coupling a 3D virtual wheat plant model with a Septoria tritici epidemic model (Septo3D): a new approach to investigate plant-pathogen interactions linked to canopy architecture. *Functional Plant Biology*, **35**, 997-1013.

Verdenal A., Combes D. & Escobar-Gutiérez A. (2008) A study of ryegrass architecture as a self-regulated system, using functional-structural plant modelling. *Functional Plant Biology*, **35**, 911-924.