

A bottom-up approach of landscape simulation leading to a generic synchronisation formalism and competition model

Le Chevalier, V. (1); Jaeger, M. (2)

(1) Ecole Centrale Paris – MAS, INRIA Saclay, EPI DigiPlante, Chatenay Malabry, F-92295, France, vincent.le-chevalier@ecp.fr

(2) CIRAD – AMAP, INRIA Saclay, EPI DigiPlante, Montpellier, F-34398, France

Abstract: The maturity of the GreenLab model at the single plant and uniform crop level has led us to consider its application to the landscape scale, where heterogeneity in space and time becomes an issue but where applications are. Our focus remained on plants and the related biophysical processes but we aimed at "functional landscapes" that should eventually achieve some quantitative realism.

We adopted a bottom-up approach, starting with what we knew how to do and expanding according to where our experiments showed potential improvements. The first prototypes had severe shortcomings in software architecture and led us to consider a new simulation formalism for our needs, as well as a new model for competition on resources.

The synchronisation formalism and software architecture is a lot more flexible and generic than what we had originally, but at the same time it lays some drastic constraints on the models themselves and their capacities. It can be seen as a drawback, but we consider this to be a great modelisation tool: it forces the modeler to be perfectly clear about the assumptions of the model, specifically about the time behaviour. The reimplementations of the plant model in this architecture highlighted some interesting points that could bring advances in the modelisation of single plants.

Keywords: landscape; bottom-up; synchronisation; resources; plant model; formalism

Introduction

Our study of "functional landscapes" stemmed from several factors. The first was the maturity of the GreenLab model for crops and uniform stands (Guo et al., 2006; Cournède et al., 2006), leaving us with many questions about how to consider heterogeneity. The

second was the realisation that most of the applications for visualisation, resource management and agricultural planning are at the landscape scale, and ask us to be able to simulate the interaction of our plant model with other environmental models. The study also built upon previous works in volume imaging as introduced in Kaufmann (1991) and Jaeger and Teng (2003).

Landscape simulations are typical complex system simulations, exciting the interest of research communities with a large panel of tools and approaches (Parker et al., 2002; Costanza and Voinov, 2004; Mayer and Sarjoughian, 2007). Given our background and objectives, our main focus remained on the plants inside the landscape, with the incentive to keep modelling work on the rest of the landscape's components to a minimum. This means that we gave more attention to the biophysical aspects than to sociological and political matters for example, and this naturally impacted modelling choices.

We adopted a bottom-up approach, starting with what we knew how to do and expanding according to where our experiments showed potential improvements.

1. Prototypes

Our first prototypes were made to assess the technical feasibility of the approach and explore the behaviour of such simulation objects. The very first prototype (Le Chevalier et al., 2007b) was developed in Liama based on previous experience with voxelisation and imagery, to investigate the potential of visualisation based on the functioning of landscape elements (figure 1). However, some of the models were not as realistic as we wished and that has led us to develop another prototype.

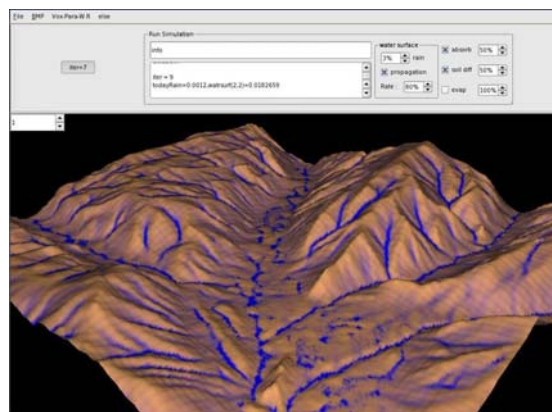


Figure 1: The interface of the first prototype, showing a running simulation. It used the Qt library for GUI design.

The new prototype (Le Chevalier et al., 2007a) had a better runoff simulation, a slightly different plant model, and a modified soil model. It has also been exploited for visualisation, in contexts ranging from scientific visualisation to explore the simulation results, to semi-realistic imaging thanks to several post-processing steps (figure 2). However, it also had several shortcomings, mainly in the simulation part.

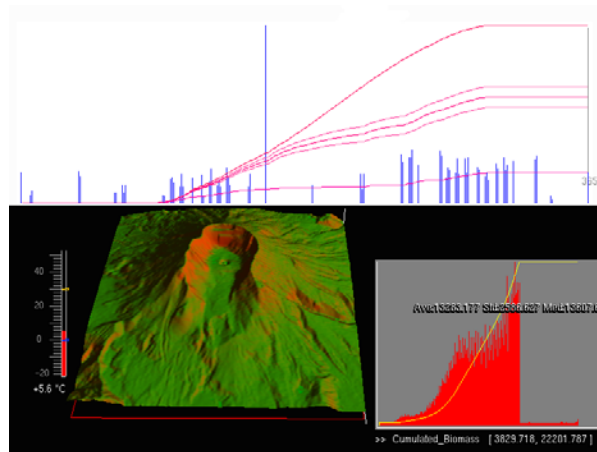


Figure 2: Visualisation of a simulation from the second prototype. It shows a landscape where the interaction of temperature, precipitation, runoff and plant growth creates heterogeneity over the landscape. The curves show the time evolution, while the 3D view show the spatial repartition of plant growth.

2. Shortcomings

The software architecture was very rigid as it was expanded from code that aimed at simulating just runoff. The water resources were not shared, rather owned by each of the processes (soil water infiltration, runoff, plant growth) and copied accordingly as the computation unfolded. It was therefore very difficult to accurately simulate competing processes, for example competition between two plant species. Adding new models was also very difficult, as it was necessary to modify the pre-existing code whose resources was impacted by the new models. There was no global vision of the landscape and specifically of resources; instead, all the information was split and shared between all the different components. The interdependency between components was a very negative point. In short, model composability (Davis et al., 2000) was not achieved at all. Based on this analysis, several axes of work were outlined. First, we had to design a generic way to model the competition for resources among several biophysical processes. Second, we needed a better synchronization architecture to handle varying timesteps that can be the result of resource sharing or of the processes themselves.

3. Resources

The solution we reached concerning resources is based on an interface between two levels. At the landscape level, resources are stored in special simulation objects called Containers. The limitations and structure of these Containers is up to the landscape modeller, though it is of course informed by the processes that interact. At the process level, resources are manipulated by other simulation objects called Accessors, that are in fact virtual

Containers. The processes use them to specify their demands on resource, and to read the level of resource available to them. Between these two levels a layer of Dispatchers assures the communication, and holds the implementation of the competition models.

4. Synchronisation formalism

At the start we thought that the synchronisation problems could be restricted to resource behaviours (saturation or depletion are events that must cause a change in the processes). But as soon as more complex test cases were tackled, it became obvious that this approach was very limiting and left many questions unanswered. Further reflection about our models, both for landscape processes and for resources, prompted us to design a more generic synchronisation formalism. It is similar to the DEVS formalism (Zeigler and Vahie, 1993; Quesnel et al., 2009), but is not based primarily on events and is therefore more intuitive to use for our object of study.

This formalism is based on two family of objects : Models and Caches. They are interconnected according to the flow of information in the simulated system (this information can be about resources, of course). Models are computation objects; they each define three different functions that allow them to evolve, react to changes in their inputs, and lay constraints on the date of the next time step. Caches are objects that abstract the simulation data, provide their inputs to the models and store their outputs. They enforce the synchronicity of the simulation data: it is all stored in an exterior data structure and is updated only once all Models have done their computations (figure 3).

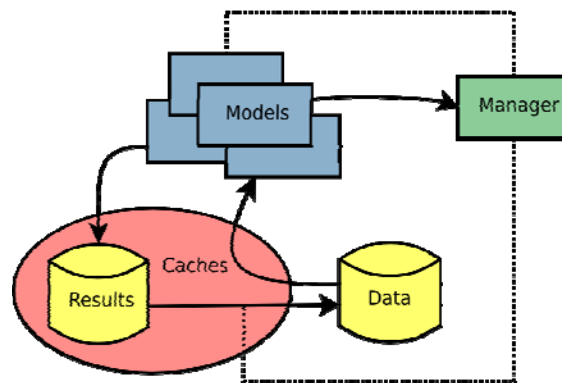


Figure 3: The principle of the new simulation formalism. Data is kept completely separate from the Models, that only interact with it through Caches. The simulation Manager calls the computations of the models to create outputs and to retrieve timestepping information. It also handles the synchronization of Caches, that is the actual change to the simulation data.

This separation of data and computation has several advantages: a consistent simulation is more realistic, stopping and resuming the simulation is easy because its state is always stored in the external data structure, and parallel computations are easier to implement because side-effects on the data are restricted.

Conclusion and further work

The synchronisation formalism and software architecture is a lot more flexible and generic than what we had originally, but at the same time it lays some drastic constraints on the models themselves and their capacities. It can be seen as a drawback, and would be one if we were aiming at a universal simulation platform where any model could be plugged without modification as a black box.

However, we consider this to be a great modelling tool: it forces the modeller to be perfectly clear about the assumptions of the model, specifically about the time behaviour. Any implicit time step readily appears when the model is implemented in this architecture, for example.

In our case, the reimplementing of the plant model in this architecture highlighted some interesting points that could bring advances in the modelling of single plants. In that respect, our study of landscapes brought us full-circle and was worthwhile even from the point of view of the heart of our team's study.

Of course an obvious further application would be a complete landscape, as has been done in the two prototypes. Some of the hydrological models used in the prototypes, especially runoff, are still a problem however, because they relied on the imperfect time synchronisation. Further study is needed as has been done for the plant growth model. Spatialisation should not be a problem given that preliminary tests have shown it to be fairly straightforward. We are also currently exploring the potential of parallel computing in the new architecture.

References

- Costanza R. and Voinov A., 2004. *Landscape Simulation Modeling: a Spatially Explicit, Dynamic Approach*, Modeling Dynamic Systems, Springer.
- Cournède P.H., Kang M.Z., Mathieu A., Barczy J.F., Yan H.P., Hu B.G., de Reffye P., 2006. Structural factorization of plants to compute their functional and architectural growth, *Simulations*, 82(7), p. 427-438.
- Davis P.C., Fishwick P.A., Overstreet M.C. and Pegden, D.C., 2000. Model composability as a research investment: responses to the featured paper, in *WSC '00: Proceedings of the 32nd conference on Winter simulation*, San Diego, CA, USA: Society for Computer Simulation International, p. 1585-1591.
- Guo Y., Ma Y., Zhan Z., Li B., Dingkuhn M., Luquet D., de Reffye P., 2006. Parameter optimization and field validation of the functional-structural model greenlab for maize, *Annals of Botany*, 92(2), p. 217-230.
- Jaeger M., Teng J., 2003. Tree and plant volume imaging - an introductory study towards voxelized functional landscapes, in Jaeger, M., Hu, B.G. (eds.), *Proceedings PMA03 : 2003' International symposium on plant growth modeling, simulation, visualization and their application*, Beijing: Tsinghua University Press, p. 169-181.
- Kaufmann A. (ed.), 1991. *Volume Visualization*, IEEE Computer Society Press.
- Le Chevalier V., Jaeger M., Mei X., Cournède P.H., 2007a. Simulation and visualisation of functional landscapes: Effects of the water resource competition between plants, *Journal of Computer Sciences and Technology*, 22(6), p. 835-845.
- Le Chevalier V., Jaeger M., Mei X., Lesluye A., Cournède P.H., 2007b. A functional landscape prototype to simulate water resource competition between plants, in Fourcaud, T., Zhang, X. (eds.),

- PMA06 - Plant growth Modeling, simulation, visualization and their Applications. November 13- 17, 2006, Beijing, China*, IEEE Computer Society, Los Alamitos, California, p. 124-131.
- Mayer G.R., Sarjoughian H.S., 2007. Complexities of simulating a hybrid agent-landscape model using multi-formalism composability, in *SpringSim '07: Proceedings of the 2007 spring simulation multiconference*, San Diego, CA, USA: Society for Computer Simulation International, p. 161-168.
- Parker D.C., Manson S.M., Janssen M.A., Hoffmann M.J., Deadman P., 2002. Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review, in *Annals of the Association of American Geographers*, 93(2), 314-337.
- Quesnel G., Duboz R. and Ramat E., 2009. The Virtual Laboratory Environment - An operational framework for multi-modelling, simulation and analysis of complex dynamical systems, *Simulation Modelling Practice and Theory*, 17, p. 641-653.
- Zeigler, B.P. and Vahie S., 1993. Devs formalism and methodology: unity of conception/diversity of application, in *Proceedings of the 25th Winter Simulation Conference*, ACM Press, p. 573-579.