Proceedings of the 7th International Conference on Functional-Structural Plant Models, Saariselkä, Finland, 9 - 14 June 2013. Eds. Risto Sievänen, Eero Nikinmaa, Christophe Godin, Anna Lintunen & Pekka Nygren. http://www.metla.fi/fspm2013/proceedings. ISBN 978-951-651-408-9.

Rule-based integration of LIGNUM into GroIMP

Katarína Smoleňová[*], **Michael Henke**, **Yongzhi Ong** and **Winfried Kurth**
Department Ecoinformatics, Biometrics and Forest Growth, Georg-August University of Göttingen,
37077 Göttingen, Germany
**correspondence:* ksmolen@gwdg.de

Highlights: LIGNUM is a functional-structural tree model combining the use of L-systems for structural development and the programming language C++ for modelling metabolic processes and radiation regimes. We show how both the structural and functional part of LIGNUM, used for a Scots pine model, can be translated into the rule-based language XL, thus offering new possibilities for model reuse and comparison.

Keywords: LIGNUM model, GroIMP, XL, FSPM integration

INTRODUCTION

LIGNUM is a generic functional-structural plant model (FSPM) designed and parameterized for modelling coniferous (e.g. Perttunen et al. 1998) as well as deciduous (e.g. Perttunen et al. 2001) tree species. These application areas justify the generic design of the LIGNUM model, which is implemented with the programming language C++. A tree is described in terms of four main structural units: tree segment, bud, branching point, and axis. While in conifers the foliage is modelled as a layer of a tree segment, in deciduous trees the leaves are modelled explicitly as ellipses or triangles (Perttunen 2008). Simplified modelling of root biomass is included. New structure is created at each time step (usually one year) by applying L-system rules, written in the language L (Perttunen and Sievänen 2005). The underlying data structure is a list. In LIGNUM, four generic algorithms for tree traversal have been identified: *ForEach*, *Accumulate*, *AccumulateDown*, and *PropagateUp* (Perttunen 2008). These are sufficient to cover the computation of all considered metabolic processes taking place throughout the tree, e.g., photosynthesis, respiration, senescence, or iterative allocation of new photosynthates into the organs. Light distribution is simulated by computing the mutual shading between all segments or by using a voxel space model. When modelling a concrete (new) species, new structural units, in the sense of C++ classes, need to be specified, e.g., *CfTreeSegment*, *PineSegment*, *ScotsPineSegment*, and also the rewriting rules for the architecture modelling, as well as species-specific metabolic processes. This design differs from another generic FSPM, GreenLab (Yan et al. 2004), where a fixed number of organ types is specified and architecture is modelled by a dual-scale automaton, corresponding to one generic rewriting rule (Smoleňová et al. 2012) – these are reparameterized for modelling different species, from simple plants to trees. Concerning metabolic processes, different implementations of submodels are available in GreenLab, too.

The question of reimplementation of the whole LIGNUM model using L-systems was raised in Perttunen and Sievänen (2005), but such a solution seemed not to be optimal because of the complexity of FSPMs. Note that the older version of LIGNUM applied to Scots pine (*Pinus sylvestris* L.) (Perttunen et al. 1996) was already translated into L-systems using the tool GROGRA (Kurth 1999). There, predefined sensitive functions were used inside the L-systems to simulate response to environment. The software platform GroIMP and its underlying language XL (eXtended L-system language) provide advanced features that allow to specify even complex relations addressed by FSPMs in terms of rule-based modelling. Here we show how to translate both, the structural and the functional part of LIGNUM (as described in Perttunen (2008)) into XL rules, with focus on the model parts necessary for the simulation of Scots pine.

IMPLEMENTATION DETAILS

The model presented here is implemented in the open-source platform GroIMP, version 1.4 (www.grogra.de), written in Java. The modelling language of GroIMP, the language XL, extends Java and allows combination of rule-based, object-oriented and imperative programming. The underlying data structure is a graph (see Kniemeyer (2008) for technical details).

The following structural components of LIGNUM were specified as XL modules (analogous to L-system symbols) with associated attributes and methods:

(1) modules for trees in general: *Tree*, *TreeSegment*, *Bud*, all extending *TreeCompartment*,

(2) modules specific for conifers, pine and Scots pine: *CfTreeSegment* (extends *TreeSegment*), *PineSegment* (extends *CfTreeSegment*), *PineBud* (extends *Bud*), *ScotsPineTree* (extends *Tree*), *ScotsPineSegment* (extends *PineSegment*), *ScotsPineBud* (extends *PineBud*).

Because of the general graph data structure, the main structural components *BranchingPoint* and *Axis* that were inevitable in LIGNUM were inherently expressed in XL.

The rules for architecture development were translated in a straightforward way. It was possible to write the XL rules in a more compact way, using *for*-loops for generation of new buds, which are not supported in the language L.

Generic algorithms of LIGNUM that are used to traverse the modelled tree and to perform functional computations were translated using selected features of the language XL (Kniemeyer 2008), e.g.:

(1) *Query expressions* are used to search for a specific pattern, enclosed in asterisked parentheses. The query

(sps:ScotsPineSegment, (sps.getLGAage() == 0) *)*

finds all newly created organs of type *ScotsPineSegment* with age 0, where age is accessed by the method *getLGAage()* and *sps* represents the current instance of *ScotsPineSegment*. Alternatively, age could be accessed using *sps[LGAage]*. Different edge patterns can be used inside of the queries to specify relations between searched organs, or to define transport in basipetal or acropetal direction.

(2) *Aggregate methods* collect multiple values and return one single value as result. Standard aggregate methods are for example *count*, *sum*, *first*, *last*. Aggregate methods can be combined with queries. The expression

sum((ScotsPineSegment *).getLGAP())*

searches for all organs of type *ScotsPineSegment*, obtains their photosynthate values (this is accessed by the method *getLGAP()*), and returns the sum of them.

(3) *Execution rules* are typically used to execute imperative statements for searched organs or update their attribute values, without changes in the topology. The following rule

sps:ScotsPineSegment ::> sps.photosynthesis();

computes photosynthesis for each *ScotsPineSegment*. In this case, the photosynthesis method was implemented inside the module *TreeSegment*.

A number of methods necessary for simulating the following metabolic processes have been translated into XL: photosynthesis, respiration, senescence, and allocation of photosynthates connected with tree segment elongation and diameter growth.

Two light models used in LIGNUM for computation of light distribution in single trees as well as the sky model (Perttunen 2008) were adapted into GroIMP. The first light model approach computes received radiation by tree segments influenced by shading caused by other segments. The second, voxel space model, reduces the radiation computation complexity by operating on voxels (regular boxes) into which the space where the tree grows is discretized. The light models were translated directly into GroIMP framework code (similarly as the light models already provided by GroIMP, i.e. the Monte-Carlo reversed path tracer, GPU-based raytracer supporting also spectral light simulation), but it would be possible to rewrite them in XL.

The following execution rule demonstrates how the light models can be called to compute radiation distribution in the tree:

sps:ScotsPineSegment ::> radiation.eval(sps);

where *radiation* is an instance of one of the light model classes. The amount of incoming and absorbed radiation by a tree segment can then be obtained by calling *sps.getLGAQin()* and *sps.getLGAQabs()* respectively.

DISCUSSION AND OUTLOOK

The advanced features of the modelling language XL allow to translate even complex models like LIGNUM into rule-based models. While structural development was translated using rules similar to classical L-systems, further features (e.g., graph traversal, query and aggregate expressions, execution rules) were necessary for simulation of metabolic processes. The two radiation models and the sky model were added to GroIMP, and therefore their code is hidden from the user, although an implementation in XL would also be possible. Furthermore, besides the fact that rule-based description of plant growth and development is more intuitive, the rule-based integration of LIGNUM into GroIMP supports model reuse and opens new possibilities for testing and comparison of different submodels. On one hand, the submodels already existing in XL could be integrated into the translated LIGNUM along with potential extensions. This will make a direct comparison, for instance of different hypotheses and approaches of light simulation or biomass partitioning, possible. On the other hand, LIGNUM submodels could be integrated into other XL models. However, both possibilities would require some model adaptations, either in LIGNUM or in (existing) XL models. Another interesting issue concerns model comparison for one selected species, simulated using different models, in terms of generated output values, based on the same input data set (cf. Baey et al. 2012). This way, one could compare outputs (e.g. biomass) generated by GreenLab and LIGNUM simulations of the same species, e.g. Scots pine.

ACKNOWLEDGEMENTS

This research was funded by the German Research Foundation (DFG) under project identifier KU 847/8-1. We thank A. Brinkmann for his work on translating LIGNUM's light models into GroIMP. We also thank J. Perttunen and R. Sievänen for explanation of the LIGNUM model and for hosting the first author at Finnish Forest Research Institute Metla, Vantaa.

LITERATURE CITED

Baey C, Song L, Cournède PH, Lemaire S, Maupas F. 2012. Evaluation of the predictive capacity of five plant growth models for sugar beet. In: Kang MZ, Dumont Y, Guo Y, eds. *Proc. IEEE 4th International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications*. IEEE, Beijing, 30–37.

Hemmerling R, Kniemeyer O, Lanwert D, Kurth W, Buck-Sorlin G. 2008. The rule-based language XL and the modelling environment GroIMP illustrated with simulated tree competition. *Functional Plant Biology* **35**: 739-750.

Karwowski R, Prusinkiewicz P. 2003. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science* **86**: 134–152.

Kniemeyer O. 2008. Design and implementation of a graph grammar based language for functional-structural plant modelling. PhD Thesis, BTU Cottbus, Germany.

Kurth W. 1999. *Die Simulation der Baumarchitektur mit Wachstumsgrammatiken*. Berlin: Wissenschaftlicher Verlag.

Perttunen J. 2008. The LIGNUM functional-structural tree model. PhD Thesis, Helsinki University of Technology, Finland.

Perttunen J, Sievänen R. 2005. Incorporating Lindenmayer systems for architectural development in a functional-structural tree model. *Ecological Modelling* **181**: 479–491.

Perttunen J, Nikinmaa E, Lechowicz MJ, Sievänen R, Messier C. 2001. Application of the functional-structural tree model LIGNUM to sugar maple saplings (*Acer saccharum* Marsh) growing in forest gaps. *Annals of Botany* **88**: 471–481.

Perttunen J, Sievänen R, Nikinmaa E, Salminen H, Saarenmaa H, Väkevä J. 1996. LIGNUM: a tree model based on simple structural units. *Annals of Botany* **77**: 87–98.

Perttunen J, Sievänen R, Nikinmaa E. 1998. LIGNUM: a model combining the structure and the functioning of trees. *Ecological Modelling* **108**: 189–198.

Smoleňová K, Henke M, Kurth W. 2012. Rule-based integration of GreenLab into GroIMP with GUI aided parameter input. In: Kang MZ, Dumont Y, Guo Y, eds. *Proc. IEEE 4th International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications*. IEEE, Beijing, 347–354.

Yan HP, Kang MZ, de Reffye P, Dingkuhn M. 2004. A dynamic, architectural plant model simulating resource-dependent growth. *Annals of Botany* **93**: 591–602.