

PLECOTUS, an Open Source Framework for Modelling Forest Growth and Interaction

Richard Zinck¹

Abstract

The aim of the PLECOTUS project is to speed up development and validation of object-oriented forest growth and interaction models and to provide a platform for their integration. It does so by defining a set of common types and a generic object-oriented view on forest growth and interaction. The project plans to offer analysis and visualization tools as well as adapters to existing tools that can be employed by all models implementing the framework. The framework itself is independent of unit systems, e.g. the metric system, and provides maximum freedom in the implementation of a specific forest growth and interaction model.

1. Introduction

Modelling forest growth and yield has a long tradition. Generally, forest growth and yield models are developed to assist in decision-making by providing information. Content and quality of this information vary from model to model. Yield tables, for example, provide information only for pure, even aged stands.

A forest stand is a complex ecosystem which reacts to disturbances such as timber harvesting and road construction. How does it react? What kind of reaction is favourable, which is to be avoided? What happens in the long run if certain types of usage are employed? Conducting field studies to answer these questions for all kinds of silvicultural interventions and stand types is tedious and virtually impossible because of sparse resources and the number of possible settings. Forest growth models are developed to answer the above and other questions concerning forest resource assessment, efficiency of forest utilization, evaluation of forest damage and last but not least the comparison of management strategies.

Typically, development time is divided between creating a formal forest growth model, choosing a platform for implementation and validating the model. A lot of time, too much time in the opinion of many forest scientists, is spent with implementation details. Choosing and learning about an implementation platform, debugging code, designing and writing GUI's and tools is time consuming, nerve wrecking and produces little scientific outcome. Thus, it is no surprise that many simulation models are poorly maintained and usable only by domain experts. Serious efforts have been made to reduce this problem using component technology (Lemm/Erni/Thees 2002) and open source development (Nagel 2002). Nevertheless, a number of problems remain. Firstly, while modelling the same part of nature with the same goals and while validating models with similar data, interfaces of two forest growth models are most likely to be incompatible. Secondly, the problem of incompatibility is aggravated by the numerous ways in which details of forest growth are modelled (stochastic, deterministic, structured or 'black box' etc.). Thirdly, every simulation model runs on its own, often obsolete platform, integrating models without the use of middleware technology is virtually impossible. The PLECOTUS project strives to solve these problems in an elegant yet practical manner while leaving maximum freedom to the model developer.

¹ Center for Informatics, Georgia Augusta University Goettingen, Germany.
email: zinck@math.uni-goettingen.de, Internet: <http://www.yapok.de>

2. Structure

To solve these problems one needs to be aware of differences between typical forest growth models. Common types include stand models, where the basic simulated object is the forest stand, and single-tree models where individual trees are simulated and stand attributes are aggregated from their features. Both single-tree and stand models are supported by the framework.

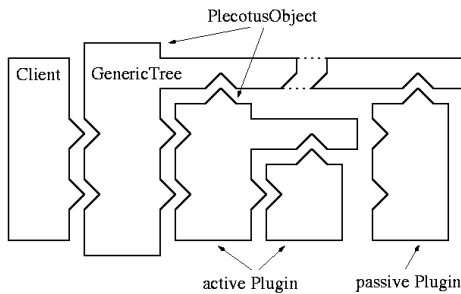


Figure 1

Simulation objects implement their interface by forwarding method calls to a plugin belonging to a certain model. Plugins can be exchanged at runtime and stored with the object.

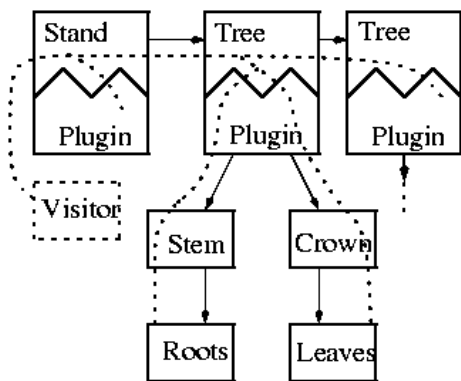


Figure 2

Visitor traversing a sample stand object (simplified). Specific actions can be performed for every object type encountered in the simulation, providing fine grained control.

Implemented as threads are basically software agents which can interact with each other and the forest stand.

Time Model. Inherited behaviour from 'PlecotusObject' includes the time-chain, a mechanism that elegantly solves interdependencies between state transitions of simulation objects in time. In a sequential program, objects change state one after another, hence information about the prior state of an object is lost while it might still be needed. The time-chain provides a transparent infrastructure for storing and retrieving previous states of a simulation object. It can be enlarged and shortened dynamically to remember any number of previous states. Since visitors traverse the aggregated model

Model Granularity. The level of supported abstraction is not fixed but scalable, e.g. the growth process of a tree can be modelled from molecular level to 'black box' representation while a forest stand can be part of a landscape or not. The types 'Tree' and 'Stand' have been defined in an hierarchical concept based on attributes typically collected in field studies and behaviour commonly encountered in single-tree forest growth models. The model, however, decides on how these types are implemented and what objects are used internally, e.g. a tree can be depicted as consisting of a crown, stem, branches, roots and leaves interacting with atmosphere and soil or as an abstract entity.

To support models of different granularity and their integration at runtime all simulation objects inherit a plugin-mechanism from the class 'PlecotusObject' (Fig. 1). A 'GenericTree' object, for example, comprises all tree attributes typically collected in field studies and offers basic services such as a growth operation, calculation of stem and crown profiles etc. How this basic functionality is implemented is left to the active model plugin. Clients, however, rely solely on the interface and reference of a 'GenericTree' object to satisfy their needs.

Internal Interaction and Communication. Interaction with a forest stand is modelled using the visitor pattern. Foresters, beetles etc. can be interpreted as 'visitors' to a stand. A visitor traverses the hierarchical model structure enabled by the plugin-mechanism (Fig. 2). Actions, such as thinning, are taken during the visit of a stand. The model of a visitor going into a stand is very intuitive. Furthermore, the question of which visitor is responsible for what action is decoupled from the action itself, e.g. a forester might assume the role of a 'StandThinningVisitor' or 'StandValueAssesingVisitor'. Thus, actions and visitors can be reused more easily. It is further possible to design hierarchies of interventions, a 'FemelHarvestingVisitor', for example, might inherit from 'HarvestingVisitor'. Note that visitors relying on types, not implementation (plugins), can be exchanged between models. Furthermore, visitors imple-

structure during simulation, they can serve as time schedulers. Hence, simulation objects can change at their own pace, as long as they can be synchronized.

Unit System Independence. To support the use of regional unit systems such as the metric system, all computations done by simulation objects must be independent of a concrete unit system. This is achieved by using objects for physical entities such as Length, Surface and Temperature, e.g. a diameter calculation function returns an object of type 'Length' from which the diameter can be queried in meter or yard.

Typical User Session. The user generates a stand from a file complying to a fixed format (csv, XML) and chooses a model to work with. Plugins are initialized and activated in the background. The next step consists of selecting visitors to interact with the forest stand, such as a 'BeetleInvasionVisitor' or 'ForestGrowthVisitor' according to your purpose. Effects are then analysed using statistical tools and visualizations as needed including visitors such as 'ValueAssessingVisitor' or 'BiodiversityAssessingVisitor'. In this way management scenarios can be explored, compared and documented using one or more models.

3. Outlook

By offering a powerful open source platform for development of forest growth models and associated tools the PLECOTUS project provides the unique opportunity to unite the efforts of research groups scattered among Europe and beyond. A large number of existing models can be implemented in the framework and seamlessly be extended to landscape simulation models as has been done for FVS (Forest Vegetation Simulator, Crookston/Havis 2002) in LMS (Landscape Management System, Carey/Elliot/Lippke/Sessions/Chambers/Oliver/Franklin/Raphael 1996). Furthermore, the clear architecture and interfaces of the PLECOTUS framework allow straightforward integration into a decision-support systems as proposed by Döllerer (2003). The foundations for the PLECOTUS framework have been laid by Zinck (2003), a forest growth model for northern Germany is being implemented.

Bibliography

- Carey, A., Elliot, C., Lippke, B., Sessions, J., Chambers, C., Oliver, C., Franklin, J., Raphael, M. (1996): Washington Forest Landscape Management Project—A pragmatic ecological approach to small-landscape management., Report No.2, Washington State Department of Natural Resources, 99p.
- Crookston, N., Havis, R., (2002): Second Forest Vegetation Simulator Conference, February 12-14, Fort Collins, CO. Proc. RMRS-P-25, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 208 p.
- Döllerer, M. (2003): Implementierung von Entscheidungsunterstützungssystemen auf Komponentenbasis, Tagungsband der 14. Tagung der Sektion Forstliche Biometrie und Informatik der DVFFA in Tharandt 2002, 27-33.
- Lemm, R., Erni, V., Thees, O. (2002): Komponentenbasierte Software-Entwicklung: neue Perspektiven für forstliche Modellierung und Informationsverarbeitung, in: Schweizerische Zeitschrift für Forstwesen, Vol. 153, pp. 3-9.
- Nagel, J. (2002): Das Open Source Entwicklungsmodell - eine Chance für Waldwachstumssimulatoren. Deutscher Verband Forstlicher Forschungsanstalten- Sektion Ertragskunde, Jahrestagung Schwarzbürg 13-15. Mai 2002, pp. 1-6.
- Zinck, R. (2003): Object-Oriented Analysis of Forest Growth and Yield, Technical Report of the Institute for Informatics, Georgia Augusta University, Goettingen.