Next-Generation Simulation and Optimization Platform for Forest Management and Analysis

Antti Mäkinen[1], Jouni Kalliovirta[2], and Jussi Rasinmäki[3]

Abstract.—Late developments in the objectives and the data collection methods of forestry create new challenges and possibilities in forest management planning. Tools in forest management and forest planning systems must be able to make good use of novel data sources, use new models, and solve complex forest planning tasks at different scales. The SIMulation and Optimization (SIMO) platform offers a modular forest planning system that adapts to various forest planning tasks by being flexible, adaptable, and extendable. Two different growth and yield simulators for Finnish conditions have been successfully implemented in the SIMO platform.

Introduction

Forest planning and management in Finland relies on certain well-established procedures that define how the data is collected and how it is processed. The basic unit of forest planning is a forest stand, and the conventional forest management objective has been steady timber procurement. The decision-making process in forest management planning is based on evaluating alternative forest development scenarios with a forest planning system. The forest planning system typically consists of a simulator and some optimization method. The simulator predicts the future development for the individual stands, and the optimization method is used to select the best forest management scenario from the number of alternatives produced by the simulator. Because stand-wise data with stand-level mean attributes have been the main data source for forest planning, the data models in the existing forest planning systems have been designed to use mainly stand-level data with a particular set of attributes. The data model dictates what kind of information the system can use and how it is recorded in the system (Paananen 1994, Tokola *et al.* 1997). Also, the prediction models and the program logic have been designed to process the stand-level data. This strong connection between the customary data collection process and the management planning process constitutes a system that can be efficient but, on the other hand, inflexible if changes need to be made.

The objectives in forest management have also been quite simple and conventional, but the settings and the manner in which forestry is practiced have been changing. Today, ecological and social values have to be taken into account in addition to the economic values (Kangas and Kangas 2005). The tools used in forest planning have to be able to adapt to these developments.

Forest planning data is collected through stand-wise inventory mainly by visual assessment and partly by subjectively positioned sample plots. The problem with this kind of data collection process is that the quality of the data is low and the costs are fairly high because it is a labor-intensive method. Given the often poor quality of the data, it is unrealistic to expect reliable prognoses from the planning system. The technical developments have brought new data sources, however, which can provide more precise and accurate forest data with comparable costs. Examples of these new technologies are remote-sensing methods such as aerial photography and laser scanning with fine spatial resolution, even at the single-tree level (Holmgren *et al.* 2003, Korpela and Tokola 2006, Maltamo *et al.* 2005,

[1] Researcher, University of Helsinki, Department of Forest Resource Management, Latokartanonkaari 7 (PL 27) 00014 University of Helsinki, Finland. E-mail: antti.makinen@helsinki.fi.

[2] Researcher, University of Helsinki, Department of Forest Resource Management, Latokartanonkaari 7 (PL 27) 00014 University of Helsinki, Finland. E-mail: jouni.kalliovirta@helsinki.fi.

[3] Researcher, University of Helsinki, Department of Forest Resource Management, Latokartanonkaari 7 (PL 27) 00014 University of Helsinki, Finland. E-mail: jussi.rasinmaki@helsinki.fi.

Næsset and Økland 2002). New field data collection devices such as the laser-relascope can provide more precise and efficient field plot measurements (Kalliovirta *et al.* 2005). The existing forest planning systems cannot take full advantage of these novel data sets because, in most cases, the data do not converge to the system's data model. Changing the data source from stand-level data to tree-level remotely sensed data (e.g.) can be hard or even impossible if the data model is fixed.

Because the forest planning data have traditionally been quite uniform, it has been possible to do most of the growth and yield calculations with a certain limited set of models. This process has been simple and straightforward; all the planning data is of the same type and format and the planner can generate the growth and yield predictions with the same set of models. Some properties of the planning systems, such as the timber prices and the costs of forest operations, have been parametric and thus controllable by the user. Changing the actual mathematical models or the logical rules that define how the simulation process runs inside the system has been challenging and has required a lot of programming. This change has been an obstacle for users wanting to implement modifications in the existing models (e.g., a local calibration of the growth models).

Many users of forest planning systems, especially larger forest companies, operate globally and a forest planning system that could incorporate local growth and yield models anywhere in the world could be of great advantage. If the growth and yield models and the simulation logic are hard-coded in the system, it can be a very demanding task to implement a new set of models from a different geographic area in the system.

A forest planning system, in which new models could be easily implemented, would be a powerful tool also for research purposes. This adaptable forest planning system would allow researchers to test and evaluate new models as a part of larger forest planning system, which would provide instantly the effects of the new model in the whole planning process.

Existing forest planning systems also almost completely lack the control of the quality of data and of the quality of the forest management plans produced with the system. From the user's point of view, it would be essential to have an estimate of the reliability of the forest management plan, because the decisions on forest management in practice base heavily on the management plans. This estimate becomes even more important because the demand for efficiency in forest planning is increasing and the forest planners do not have the time or the resources to thoroughly evaluate the management plans and their quality. A system that could point out the stands in which the input data was likely to be erroneous would help the planners concentrate their efforts to those problematic stands.

A project for developing a SIMulation and Optimization (SIMO) platform for next-generation forest planning started at the University of Helsinki in 2004.[4] The SIMO planning system is implemented as a software platform consisting of a number of modules that include different functions. These modules together constitute a forest planning system that is flexible, adaptable, and extendable. The term "platform" is used here in the sense that various forest models for different conditions and planning tasks can be implemented in the same platform. One part of the project is to implement a set of growth and yield models for Finnish conditions to produce a functional planning system that could by used by forest planners in Finland.

Requirements for the Next-Generation Planning System

The requirements for the next-generation forest planning system were based on interviews with the different forestry organizations that are the users of the existing forest planning systems. The results of the interviews were reviewed and the different requirements cited by the users were summed up into more concise form. The most notable issue that the users repeatedly brought up was the demand for a system that was transparent and that could be easily modified according to the users' needs.

[4] The SIMulation and Optimisation (SIMO) platform is being developed in the SIMO research project, which is financed by UPM-Kymmene Forest, Tornator Oy, Metsämannut Oy, Metsähallitus, Forestry Development Centre Tapio, and Tekes.

In the process of designing the platform, we needed to consider several different aspects. Different users may use different data sources, such as traditional compartment-wise inventory data consisting of stand-level mean attributes or single-tree–level remotely sensed data, in their planning process. In addition, grid-based data, in which forest cover is divided into cells of given size, has been becoming more popular. The ability to accommodate different types of data in the planning process demands flexibility from the planning system's data model.

The planning system should also be capable of solving planning problems of different types and scales. A forest planning problem can be optimizing an operational small-scale stand harvest schedule in which the planner only has to consider maximizing the forest property's net present value. Then again, the planning problem might be a large-scale ecological conservation plan in which the ecological and social aspects as well as the economical values have to be taken into consideration. Often, the inclusion of new objectives in forest planning requires that new kinds of data or models be incorporated in the planning system. Because we cannot exactly foresee the future needs of forest planning systems, the systems should be designed so that it would be as easy as possible to extend them when a need arises.

The main requirements for the SIMO forest planning platform were that the system's data model be flexible and the system be adaptable to different types of planning problems and extendable for possible future needs. More specific requirements were given as well. One important requirement was that the planning system not work like a "black box" in which most of the functionality is hidden and cannot be manipulated by the user. Instead, the system should be transparent and provide the users with access to the different mechanisms inside the planning system and a means to verify the internal workings of the system.

Forest planning systems are often used as a part of or in connection with more extensive data management systems. Because of this connection, interfacing the planning modules to different databases and user interfaces should be made as easy as possible. Because the data models and formats differ between different systems, the data interface to the planning system should be such that it can handle different types of data.

Forest information systems in general include spatial information on the forest stands. Also, tree-level spatial information can be stored with the new and coming data collection methods, such as single-tree remote sensing. The existing forest planning systems do not use this spatial information, however. With the next-generation forest planning system, it should be possible to use the spatial information in the planning process.

Technical Solutions

Platform Design

The aim of this project was to develop a forest planning platform comprising modules that include all of the functionality needed in a forest planning system. This kind of functionality demands a simulator module for creating the forest growth projections and an optimizing module for selecting the management scenario that conforms to the user's objectives. The modularity of the system is important because the users may want to include only some specific forest planning functions in their information systems. A user interface development is not involved in this project because the modules can be integrated behind existing user interface. Here, the simulator module is given the main focus because it can be considered as the core of the planning system and a lot of decent optimization tools are available.

The structure of the simulator is divided here into four logical components: (1) the data model, (2) the models used in the simulations, (3) the simulation logic, and (4) the application logic. The simulation logic comprises the logical rules that define how the models are used to process the data and the application logic is the actual program code that interprets and executes the simulation rules defined in the simulation logic. To produce a flexible, adaptable, and extendable system, the data, the models, and the simulation logic can be modified by the user, but there should be no need to modify the application logic. The user-modifiable components of the simulator are implemented using Extensible Markup Language (XML) documents (Boag *et al.* 2005). The use of XML is proposed here because it is inherently universal and portable, human readable, and easy to process programmatically (McGrath 2003). The syntax for the simulator's XML documents is defined in XML schemas. One advantage of using XML documents is that they contain not only the data values but also the interpretation and metadata for the values. A common vocabulary in all of the simulator's XML documents describes the relationships and works as a glue between the different components of the simulator.

Simulator Module

Data Model. The requirements defined that the simulator's data model should be flexible instead of fixed. The data model proposed here is a hierarchical structure that consists of objects that can have attributes and subobjects. The object-subobject structure is recursive; i.e., the subobjects are objects. The attributes consist of variable-value pairs that define both the interpretation and the value of the attribute. The definition of an object in the data model is abstract and the user can define what the objects actually represent. A typical implementation of this object model could be Finnish national forest inventory plot data with a two-level hierarchy in which a single plot includes a number of trees as subobjects. This kind of hierarchy is pictured in figure 1. Implementing different kinds of data sets on this data model is easy because the objects in the hierarchy can represent almost any real-world object. For the data interface, the use of XML is proposed. Forest data is usually hierarchical and converts into XML format easily, which makes XML a good interface for transporting data between different forest information systems.

Models. The models in the forest planning system describe the natural processes; they also describe the human actions, such as harvests, in the forest. An individual model can be a simple mathematical model describing some stand or tree feature. Predicting the future development of a stand is more complicated and usually involves the use of a number of individual models together. One of the requirements for the next-generation planning system was that implementing new models in the system should be easy. An extendable model base is a set of individual model implementations separated from the application. This kind of separated model base enables straightforward implementation of new models and reuse of old models. Each individual model in the model base consists of three different components: (1) the program code for the model, (2) an XML format description of the model, and (3) an interface between the model and the simulator, generated automatically from the model XML description (fig. 2). The XML description of a model interfaces the separate model implementation to the system and specifies the model's input and output (Kokkonen *et al.* 2003). Currently, models written in C, C++, Fortran, and Python programming languages can be implemented in the model base.

Simulation Logic. The actual simulation is described in the simulation logic, which includes the logical rules on how and in which order the models in the model base are used to process the data. The simulation logic is defined in XML documents as model chains that consist of a hierarchical task structure. The simulation logic is split into tasks that are processed if the condition for the task is satisfied. The conditions in the model chains are logical expressions that can be evaluated against data values. Because the task structure is hierarchical, the tasks can be further split into subtasks. Each task ends up in a single model at the bottom of the task hierarchy. The

Figure 1.—*Hierarchical data in which a plot-level object (1) has tree-level subobjects (2 and 3).*

```
<data>
1.  <object level="plot" id="01">
        <attributes>
            <attr>
                <variable>N_ha</variable>
                <value>1200</value>
            </attr>
        </attributes>
        <subobjects>

2.          <object level="tree" id="011">
                <attributes>
                    <attr>
                        <variable>dbh</variable>
                        <value>13.2</value>
                    </attr>
                </attributes>
            </object>

3.          <object level="tree" id="012">
                <attributes>
                    <attr>
                        <variable>dbh</variable>
                        <Value>10.9</value>
                    </attr>
                </attributes>
            </object>

        </subobjects>
    </Object>
</data>
```

simulator can be modified by changing the models in the model chains, changing the parameters of the models, or changing the conditions of the tasks. All this can be done by modifying the XML files and without touching the program code. A hierarchical task structure of a model chain is shown in figure 3 in XML format and as a more illustrative tree structure.

As a part the development project, we implemented two different forest simulators for Finnish conditions in the SIMO platform: a tree-level simulator that predicts forest growth by using single-tree growth models on a set of trees generated with diameter distribution models and a stand-level simulator that predicts forest growth with models using stand-level mean attributes. The tree-level simulator is based on the Finnish Forest Research Institute's MELA forest planning system models (Hynynen *et al.* 2002) and the stand-level models are by Vuokila and Väliaho (1980).

Application Logic. The application logic covers the rules that define how the input data, the models, and the simulation logic files are processed by the simulator application. The simulator application parses the XML files containing the simulation logic (i.e., the model chains and the simulation parameters) and applies different models to the data according to the logical rules in the model chains. The simulation application itself does not constitute an actual simulator; it needs all the information about the data structure, the models depicting the natural processes, and the simulation as XML input files. Because the simulation logic is separated from the application logic, the simulator can be modified without touching the actual program code. The simulator application is written in the Python programming language. Users should not be required to do any reprogramming on the simulator's application logic to modify the simulator or even to implement new simulators on the platform. The program code is well documented, however,

Figure 2.—*Three model components: (1) Extensible Markup Language definition, (2) interface between model and simulator, and (3) program code implementation.*

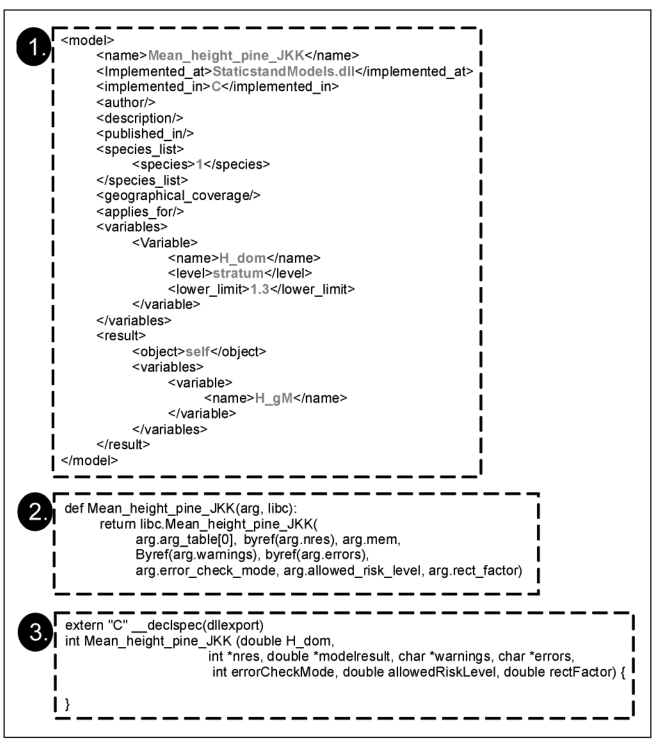
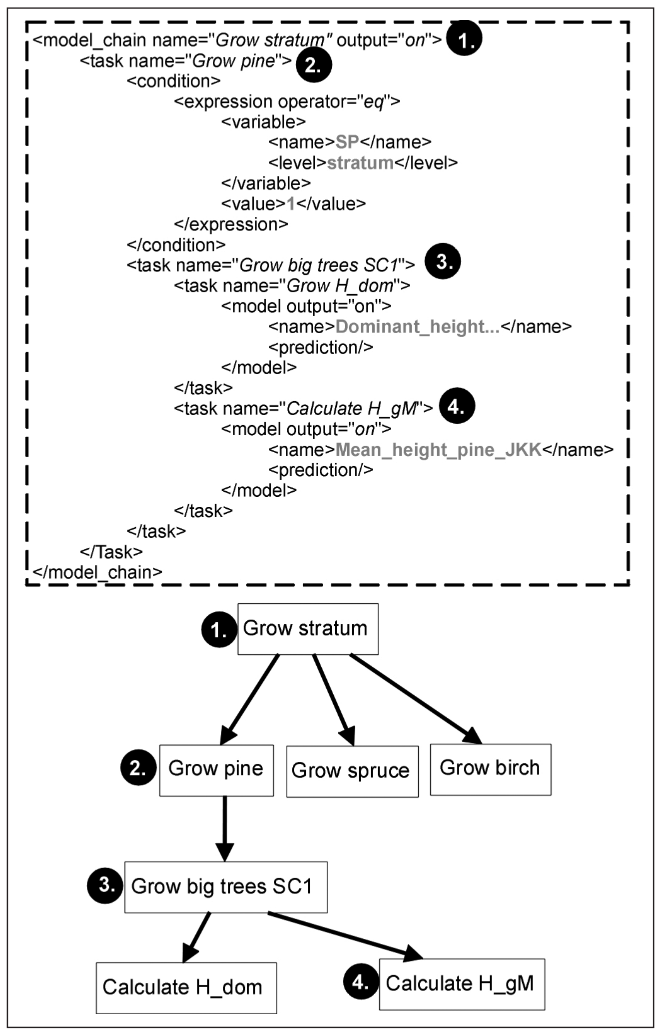


Figure 3.—*A hierarchical task structure in Extensible Markup Language format and as a tree structure.*



and structured for the possible developers who want to extend or modify the application logic.

Other Platform Modules

In addition to the simulator module, the platform includes other modules that constitute different kinds of functionality in the planning system. Other important modules that are part of the platform include optimizer, validator, Geographic Information System (GIS), and reporter.

The optimizer module is basically a programmable interface to another program library. It is an "optimization package" that includes linear programming methods and metaheuristic methods for selecting the best alternative management scenario. This interface can be modified for several different optimization packages because different optimization methods suit different planning problems.

The validator module's main function is to validate the input data and all the files constituting the simulation logic. The use of XML documents as the data exchange format as well as the format for the simulation logic files provides a lot of tools for the automatic validation of the structure and format of the files. Validating the structure and format is easy, but validating the content (i.e., the data values) is a different kind of problem and is strongly linked to the quality control of forest planning data.

Because providing tools for using spatial information in the planning process was one of the objectives of the forest planning system, a module that handles spatial data processing is also being implemented in the platform. The GIS module will be similar to the model base of the platform; it will be a set of functions that is separated from the application. The GIS module will be able to compute basic spatial operations, such as neighborhood operations.

A reporter module is needed to process the simulation results into different formats for the user. Using the reporter module, the user can produce XML format data exchange files, text files, and different kinds of charts. The reporter module can be also used to produce data transfer files to some other information system.

Discussion

The development of a next-generation forest planning system was guided by three main objectives: flexibility, adaptability, and extendability. The modular structure of the platform components and the separation of the application logic from the simulation logic were the most important technical solutions in achieving these objectives. The resulting platform can be seen as an open system that can be modified and extended with new components according to current needs without time-consuming reprogramming (Reynolds 2005). The modules produced within this project include all the functionality needed in a forest planning system, except the user interface.

All the user-modifiable components are implemented as XML documents, which are text files and thus are easily modifiable by the user. The simulator's data model can be modified freely as long as the data is hierarchical. The simulation descriptions can also be modified without touching the program code. The only aspect that calls for programming is implementing new models in the model base. Nevertheless, in many situations, users can take ready-made models and implement those in the model base or reuse models already implemented in the model base.

Modifying the XML documents that define the current simulation task, although fairly easy for even a nonprogrammer, requires acquaintance with the syntax. The different structures defined in the XML schema files are designed to be as simple as possible and therefore the syntax the user needs to learn is quite easy. In addition to human readability, XML documents can be easily manipulated programmatically. A graphical user interface for manipulating the simulator's XML documents would make the system even easier to modify. Because it should be possible to integrate the SIMO modules in existing forest information systems, XML seems like a suitable interface for data transfer. Database contents can be transformed into XML format (e.g., with the standard XML query language, XQuery) (Boag *et al.* 2005).

The transparency of the system was one of the requirements of the development process. The separated simulation logic implemented as XML documents provides a simple and human-readable access to the system and the simulation logic. A separated model base with well-documented model implementations and the XML definitions that include the metadata and the semantics for the models also adds up to the transparency of the system. A well-documented and structured program code is a necessity for an open system that can be continuously developed and extended.

The SIMO modules constitute a general forest planning platform that can be applied to different conditions and various planning tasks. As a means for verifying the adaptability of the SIMO platform, a tree-level simulator and a stand-level simulator were implemented on the platform. The simulation definitions, model chains, and model definitions were constructed for both of the simulators separately from the application development and without touching the program code. The experiences gained from implementing two very different simulators on the same platform are very positive and support the presumption that the SIMO platform constitutes a flexible, adaptable, and extendable forest planning system.

The next likely step in the further development of the SIMO platform is creating graphical user interfaces for running the simulations and for manipulating the XML documents. For a forest planning system to be widely usable, a user-friendly interface is a necessity. Implementing different types of growth and yield models (e.g., models for different vegetation zones or process-based growth models) would provide more valuable experiences on the modifiability of the system.

Literature Cited

Boag, S.; Chamberlin, D.; Fernández, M.F.; Florescu, D.; Robie, J.; Siméon, J., eds. 2005. XQuery 1.0: an XML query language. http://www.w3.org/TR/xquery. (7 January 2007).

Holmgren, J.; Nilsson, M.; Olsson, H. 2003. Estimation of tree height and stem volume on plots using airborne laser scanning. Forest Science. 49(3): 419-428.

Hynynen, J.; Ojansuu, R.; Hökkä, H.; Siipilehto, J.; Salminen, H.; Haapala, P. 2002. Models for predicting stand development in MELA system. Res. Pap. 835. Helsinki, Finland: Finnish Forest Research Institute. 116 p.

Kalliovirta, J.; Laasasenaho, J.; Kangas, A. 2005. Evaluation of the laser-relascope. Forest Ecology and Management. 204(2-3): 181-194.

Kangas, J.; Kangas, A. 2005. Multiple criteria decision support in forest management— fundamentals of the approach, methods applied, and experiences gained. Forest Ecology and Management. 207: 133-143.

Kokkonen, T.; Jolma, A.; Koivusalo, H. 2003. Interfacing environmental simulation models and databases using XML. Environmental Modelling & Software. 18: 463-471.

Korpela, I.; Tokola, T. 2006. Potential of aerial image-based monoscopic and multiview single-tree forest inventory: a simulation approach. Forest Science. 52(2): 136-147.

Maltamo, M.; Mustonen, K.; Hyyppä, J.; Pitkänen, J.; Yu, X. 2005. The accuracy of estimating individual tree variables with airborne laser scanning in a boreal nature reserve. Canadian Journal of Forest Research. 34(9): 1791-1801.

McGrath, R.E. 2003. XML and scientific file formats. Urbana, IL: University of Illinois, Urbana-Champaign, National Center for Supercomputing Applications. 25 p. http://hdf.ncsa.uiuc.edu/projects/nara/XML_and_Binary.pdf. (13 December 2007).

Næsset, E.; Økland, T. 2002. Estimating tree height and tree crown properties using airborne scanning laser in a boreal nature reserve. Remote Sensing of Environment. 79(1): 105-115.

Paananen, R. 1994. A data model for a GIS-based forest information system. Res. Pap. 493. Helsinki, Finland: Finnish Forest Research Institute. 78 p.

Reynolds, K.M. 2005. Integrated decision support for sustainable forest management in the United States: Fact or fiction? Computers and Electronics in Agriculture. 49(1): 6-23.

Tokola, T.; Turkia, A.; Sarkeala, J.; Soimasuo, J. 1997. An entity-relationship model for forest inventory. Canadian Journal of Forest Research. 27: 1586-1594.

Vuokila, Y.; Väliaho, H. 1980. Growth and yield models for conifer cultures in Finland. Communicationes Instituti Forestalis Fenniae. 99.2. In Finnish.