International Congress on Environmental Modelling and Software

3rd International Congress on Environmental Modelling and Software - Burlington, Vermont, USA - July 2006

Jul 1st, 12:00 AM

Interleaving Growth and Regeneration Models in the NED-2 Decision Support System for Forest Ecosystems.

Donald Nute

Mark Twery

Julian Bishop

Zhiyuan Cheng

David Bucugnani

See next page for additional authors

Presenter/Author Information
Donald Nute, Mark Twery, Julian Bishop, Zhiyuan Cheng, David Bucugnani, Walter D. Potter, David Loftis, Scott Thomasma, and Peter Knopp

Interleaving Growth and Regeneration Models in the NED-2 Decision Support System for Forest Ecosystems

Donald Nute[a], Julian Bishop[a], Zhiyuan Cheng[a], Walter D. Potter[a], David Loftis[b], Mark Twery[c], Scott Thomasma[c], and Peter Knopp[c]

[a]Artificial Intelligence Center, University of Georgia, Athens, Georgia, USA
[b]Bent Creek Experimental Forest, Southern Research Station, USDA Forest Service, Asheville, NC, USA
[c]Northeastern Research Station, USDA Forest Service, Burlington,VT, USA

Abstract: NED-2 is a goal-driven system designed to help manage timber, wildlife, visual, and ecological goals for a forested ecosystem. The basic approach of the decision process modeled by NED-2 is to develop alternative management plans for the stands in a management unit, to simulate these plans over time, and then to analyze the results of the simulation to see how well the management goals are achieved by the alternative plans. The basic simulation tool used in the system is the USDA Forest Service Forest Vegetation Simulator (FVS.) FVS provides a regeneration component, but a need was recognized for making available alternative regeneration models. The first effort in this direction was to integrate a competitive model developed by David Loftis and implemented as a program called REGEN. This model uses pre-disturbance inventories of existing regeneration sources and information about new seedling establishment, particularly light-seeded species from the seedbank or from trees in areas adjacent to a stand. The stochastic model uses a knowledge base that allows ranking the competitive abilities of different species, taking into account the origin of the regeneration source—new seedling, stump-spout, or different sizes of pre-existing seedlings. Different knowledge bases can be developed for different forest types and regions. This model must be interleaved with FVS when the user desires to use the REGEN model. Individual NED-2 software agents control the FVS and REGEN systems. This paper describes how these agents communicate using a blackboard architecture to synchronize the operations of these two models. The task is made more complicated because regeneration on one stand can affect the results on regeneration on an adjacent stand at a later time.

***Keywords**:* Regeneration, Growth and Yield Models, Decision Support System, Ecosystem Management, Forest Management.

1. INTRODUCTION

NED-2 is a decision support system for managing forested ecosystems (Nute et al. 2005, Twery et al. 2005.) A key feature of the NED-2 system is the simulation of the growth of stands of forested land under alternative silvicultural treatment plans. Some silvicultural treatments such as clear-cutting will open the overstory enough to trigger natural regeneration. Growth and yield models, such as the Forest Vegetation Simulator (FVS) (Crookston et al. 1997) used in NED-2, incorporate a regeneration model. But users may prefer to use a different regeneration model, especially when that model has been calibrated for local conditions or includes features not included in FVS. In this paper, we describe how the regeneration model developed by David Loftis (1989, 1990) was integrated with FVS in NED-2. While this exercise involved particular simulation and regeneration models, the issues and methods described apply more widely to integration of other pairs of models.

Simulation is only one step in the decision model implemented by NED-2, but it is an essential step. The NED-2 decision process is goal-driven, and the goals that are considered by the system include timber, wildlife, visual, and ecological goals. After entering inventory information and selecting a set of management goals, NED-2 leads the user through a series of steps to guide development of a management plan. The basic approach is for the user to create alternative silvicultural treatment plans, simulate them, and analyze them to see how well they achieve the desired management goals. The agent-based architecture used in NED-2 is designed to facilitate integration of third-party decision tools as well as decision tools developed by the NED-2 development team. As the user proceeds through the steps of the NED-2 decision process, the different decision tools are made available and NED-2 performs the necessary data conversion among the formats required by the different decision tools. The NED-2 decision model and architecture are described in detail in (Nute et al. 2005) and (Twery et al. 2005).

In this paper we describe how treatment plans are created and simulated in NED-2. Then we describe the Loftis regeneration model (REGEN) and its implementation. Next we discuss the basic method for integrating FVS and REGEN in NED-2. Finally, we discuss how the regeneration model had to be modified work within the context of NED-2.

2. SIMULATING TREATMENT PLANS IN NED-2

The first step in the NED-2 decision process is entering forest information including an inventory of overstory, understory, and ground plots for each stand in the management unit. (By a stand we mean a forested or non-forested area with the same silvicultural characteristics throughout the area, and by a *management unit* we mean a collection of stands that are being considered together for the purposes of management.) Next, the user must establish a baseline year for defining silvicultural treatment plans. The baseline year can be no earlier than the latest year for which stand inventory has been entered. The user must also decide which simulation model will be used for each stand. At present, several regional variants of FVS are available in NED-2.

Figure 1 shows the matrix that NED-2 uses to configure the baseline. Rows in the matrix correspond to stands in the management unit and columns correspond to years. The column headed "models" indicates whether the user has selected growth, treatment, and regeneration models for each stand. By double-clicking on a cell in this column, the user accesses a dialog where he can select options. To select the Loftis REGEN model for a stand, the user must also select a knowledge base to use with the model. The purpose of these knowledge bases is explained below.

The dark gray cells in the matrix indicate years for which there is no data available for a stand. The first white column in each row will correspond to the inventory year for the stand. In this example with only five stands, we have inventory for 1995, 1999, and 2001. The baseline year can be 2001 or any year after 2001. In this example, the user has added the current year, 2006, to the baseline matrix. Conversion to grayscale has obscured it, but the header for the 2006 column is in yellow, indicating that this is the year that has been selected for the baseline year.

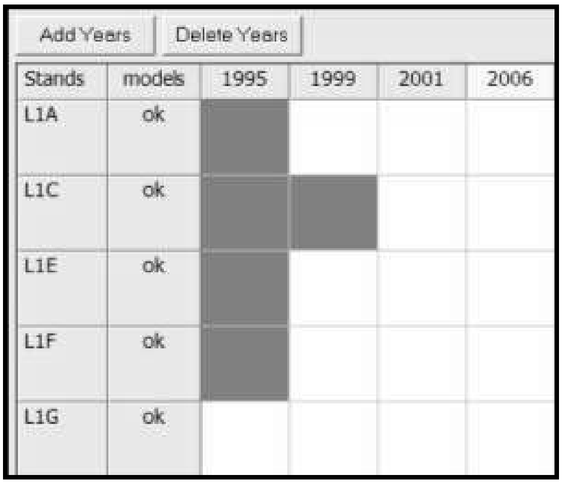

Figure 1. NED-2 baseline development matrix

Once these initial tasks have been completed, NED-2 to generates data for the baseline year. If the baseline year is the same year as the inventory year for a stand, then the inventory data is used as the baseline year data for that stand. For all other stands, NED-2 runs the appropriate variant(s) of FVS on the inventory data and simulates stand change up to the baseline year. This simulated data becomes the baseline year data for these stands.

The user creates a set of user-defined treatments that will be used to guide NED-2. NED-2 provides a set of standard treatments with default parameters that the user may add to the treatment set or modify as necessary. NED-2 also provides tools for defining various custom cuts that the user may want to include in the treatment set.

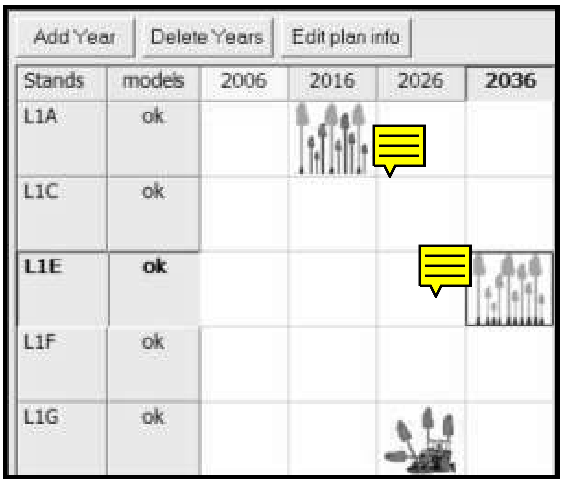

Figure 2. NED-2 plan development matrix

After the treatment set is created, the user creates one or more treatment plans using the NED-2 plan development dialog. Figure 2 shows the matrix used in this dialog to create a treatment plan. First the user specifies the years covered by the plan.

This example (Figure 2) shows a 30-year plan with a 10-year treatment cycle. Selected treatments are indicated on the matrix by icons. Multiple treatments can be scheduled in the same year. Once a plan is developed, it becomes part of the user's working file.

After the first plan is created, the user can create a second, third, etc., in the same manner. For convenience, earlier plans can be edited, or they can be copied as a starting point for an alternative plan.

Before simulating treatment plans, the NED-2 simulation agent checks to make sure all information needed to simulate all existing plans is available. If no stands have been entered, if a baseline year hasn't been generated, if no plans have been created, or if some other necessary data is missing, the simulation agent writes an HTML file listing all missing data and opens it in the user's default Web browser. This allows the user to make all necessary corrections at on time before continuing.

If all data needed for simulation are found, a dialog asks the user to specify which plans are to be simulated, and for which stand each plan is to be simulated. Thus, a user can easily simulate a single plan on a single stand, all plans on all stands, or any combination. After the user has specified which plans and stands to simulate, the simulation agent executes the appropriate FVS variant to simulate tree and stand change. FVS creates an output file that shows the tree data for each year in the plan. In years where treatments are scheduled, FVS provides both pre-treatment and post-treatment data.

The simulation agent converts the FVS output back into the NED-2 data model. A key concept of this data model is a *snapshot*. A snapshot represents what a stand looks like at a particular point in time under a particular treatment plan. There will be one snapshot for each stand for each year where the plan does not include any silvicultural treatment for the stand. In years where one or more treatments were scheduled, there will be two snapshots, one before and one after the treatments are performed.

As was mentioned before, FVS incorporates a regeneration model. Regeneration can be turned off during an FVS run by including the appropriate key words in the FVS control file. Without some mechanism for interleaving an alternative regeneration model, the user's only options are to accept the FVS regeneration model or to have no regeneration take place during simulations.

3. THE LOFTIS REGENERATION MODEL

The Loftis regeneration model (Loftis, 1989; 1990) requires a pre-disturbance inventory of regeneration sources. The model also requires information about stumps left after tree removal and the presence of light-seeded species in the area. Many of these data are stored in the NED-2 inventory for understory and ground level plots. If these data have not been entered, then regeneration using the Loftis model will be invalid.

The Loftis model is competition-driven. Using a knowledge base developed for a specific set of species and site conditions, such as an ecological classification unit, the model predicts the number and species of tree that will form the overstory ten years after a regeneration event. The model is stochastic and produces slightly different results when run on the same data multiple times. The model has been implemented as REGEN, a Prolog inference engine with an Excel interface (Boucugnani, 2005.) REGEN was designed so a user could easily run the model multiple times using a variety of plot sizes. The system generates useful statistics based on the results of these runs.

For the purposes of the NED-2 project, an important feature of REGEN is that the inference engine is a self-contained Prolog program. Since the blackboard architecture and the agents for NED-2 are also written in Prolog, this simplified integration of the regeneration model into NED-2. The inference engine takes a set of Prolog clauses as input and produces a set of Prolog clauses as output. To run the model in NED-2, it was necessary to write a regeneration agent that could convert data from the internal NED-2 model into a set of clauses the REGEN engine could use, and then convert the Prolog clauses the REGEN engine produced back into the NED-2 data model. Providing input to the REGEN engine was relatively simple, although, interpreting the output of the regeneration model raised some questions.

The plan development dialog in NED-2 was modified to accommodate the REGEN model. The user must not only specify which growth simulator to use for each stand, but must also specify which regeneration model to use: the regeneration function built into FVS, the Loftis model, or none. If the Loftis model is specified for a stand, then the knowledge base that contains the regeneration rules for that location and forest type must also be specified.

4. INTEGRATING SIMULATION AND REGENERATION

The first task for integrating REGEN with NED-2 was to design a method that would allow NED-2 to interleave the FVS growth simulator with the REGEN engine. We already had a simulation agent in NED-2 that was able to run FVS. Now we needed a regeneration agent that was able to run the REGEN engine. And we needed a method for the two to coordinate their activities.

An advantage of an agent architecture is that one agent does not need to know very much about how another agent works. The Loftis regeneration model is designed to be used after a major disturbance has removed essentially all of the overstory. Knowledge of the conditions that trigger regeneration in the Loftis model fall within the domain of the regeneration agent, not the simulation agent; so the entire process begins when the simulation agent uses FVS to simulate data all stands for a plan from beginning to end, ignoring the possibility that regeneration might take place on any stands where the Loftis model has been selected by the user. When the simulation agent is finished, it puts facts on the blackboard indicating which stands it has simulated.

Next, the regeneration agent sees the facts on the blackboard indicating which stands were recently simulated. It then begins examining all of these stands from the first year of the simulation looking for a stand that satisfies the triggering conditions for the Loftis model. It identifies the earliest year where regeneration is triggered on any stand and it runs the model on a single stand where regeneration begins in that year. Then it modifies the snapshots for that stand for the year that comes ten years after regeneration is triggered, and it updates all snapshots for that stand for subsequent years. Finally, it puts a fact on the blackboard indicating that it ran the Loftis regeneration model on that stand in that year.

Now the simulation agent sees the message left by the regeneration agent. It re-simulates the affected stand from the post-regeneration year to the end of the plan and puts this information on the blackboard. The regeneration agent then examines all the stands starting from the plan-year when the previous regeneration event occurred until it finds another stand where regeneration is triggered. This process continues, working forward from the beginning to the end of the plan, until the regeneration agent can find no more stands where the Loftis model is triggered. At this point, it cleans up the notes on the blackboard and the full simulation with regeneration is complete.

It might seem more efficient to allow the regeneration agent to run the Loftis model on all stands where regeneration is triggered in any year, and then allow the simulation to re-simulate each of the affected stands from its post-regeneration year forward to the end of the plan. But this cannot be done because regeneration may be affected by adjacent stands. If a light-seeded species is represented in the overstory of a neighboring stand, then seedlings from that species are placed in the regeneration stock for the target stand even if that species is not already in the target stand. But the light-seeded species might only have arrived in the neighboring stand as a result of an earlier regeneration event on the neighboring stand. We designed this back-and-forth interleaved method to allow for this possibility. Although the circumstances where this is needed may be rare, we do not think that the repeated alternation between the two agents as they work from the beginning to the end of the treatment plan slows down the system significantly.

We said that the regeneration agent "modifies" the snapshot representing the stand as it looks ten years after regeneration was triggered. Remember that our method for simulating a treatment plan is to first simulate all growth and treatments for the plan without regard for regeneration. Then the regeneration agent determines at which points regeneration is triggered and runs the Loftis regeneration model. Any stem appearing in the pre-regeneration snapshot with a dbh of at least 1.5" is considered a residual from the regeneration-triggering treatment and is not treated as one of the stems that compete during regeneration. But users typically enter many stems with dbhs of less than 1.5" in the inventories for the understory plots describing stand conditions before the regeneration event. These stems will have been grown by FVS in the first stage of the simulation. The regeneration model picks the stems and seedlings that survive following regeneration. If a survivor comes from a "large" understory stem (a stem over 4' tall with a dbh of less than 1.5"), the regeneration agent randomly chooses stems in the target snapshot and marks them to survive. After choosing all the "survivors" in this category, all other stems in this category in the target snapshot are removed. For this class of stems, then, the regeneration agent actually removes tree records rather than adds tree records during regeneration. This method has the advantage that the dbh of the selected stem has been determined by FVS during growth simulation. The regeneration agent does not need to calculate dbhs for these stems. For other stems that come from stump sprouts, from root suckers, from seedlings, or from "small" or "medium" stems without a dbh, the regeneration agent creates a new record in the target snapshot and gives it a dbh of 1.5". Later, we will be able to insert a model that varies the sizes of these newly-created stems if a different method for calculating dbh distributions is adopted.

Another problem can arise in interpreting the outputs from the REGEN model when the user has selected a cycle length of more than ten years in defining the treatment plan. Then the regeneration agent must modify the snapshot for the first year that comes at least ten years after the regeneration-triggering event. Suppose, for example, the years 2005 and 2025 are included in a plan, but no years between 2005 and 2025 are in the plan. If regeneration is triggered at 2005, there is no target snapshot for the year 2015 for the regeneration agent to modify and the regeneration agent must modify the snapshot for 2025.

Cycles can also be too short for the model. If there are snapshots at 2005, 2010, and 2015, and regeneration is triggered at 2005, the regeneration agent modifies the snapshot for 2015 and marks the 2010 snapshot as being "in regeneration". If the user tries to look at data for the 2010 snapshot for this stand, NED-2 tells him that no data is available because the year is part of a regeneration event.

When the Loftis model is run as a stand-alone tool, the assumption is that the overstory for the stand has been disturbed sufficiently to allow regeneration. This issue is more complex in the NED-2 context. Obviously, a clearcut should trigger regeneration, but shelterwood cuts designed to promote regeneration are also common silvicultural treatments. This prompted us to design a complex set of triggering conditions for our regeneration agent. All of these conditions depend on residual basal area after a treatment. The regeneration model applies the following three tests on each snapshot in the given order.

1. If the residual basal area for the stand in year Y is greater than 50 ft^2/acre and less than 60 ft^2/acre, and if the residual basal area for the stand is less than 20 ft^2/acre within five years after Y, then regeneration is triggered in year Y.

2. If the residual basal area for the stand in year Y is greater than 20 ft^2/acre and less than 50 ft^2/acre, and if the residual basal area for the stand is less than 20 ft^2/acre in the period Y + 5 years to Y + 10 years, then regeneration is triggered in year Y.

3. If the residual basal area for the stand is less than 20 ft^2/acre in year Y, then regeneration is triggered in year Y.

Since the triggering conditions for the Loftis model are defined in terms of the basal area of a stand, it can certainly happen that a stand will still satisfy the triggering condition ten years after a regeneration event. This is a situation that was not considered in developing the REGEN core engine since it was assumed that the new trees would form a closed canopy within ten years. Consequently we designed the regeneration agent to repress further regeneration events for thirty years after a regeneration event takes place.

Cheng (2005) provides a more detailed description of the simulation and regeneration agents and the method they use to interleave the growth and simulation models in NED-2.

5. CONCLUSIONS

One might expect that integrating a computational model for simulating growth and treatments with another computational model for regeneration would be straightforward. And the basic mechanics, as we have described them, were reasonably straightforward. But the regeneration model we used in this project was designed for a different context than the one in which we were using it. This difference raised a number of questions about how our NED-2 regeneration should apply the regeneration model and how it should interpret the model's outputs.

As we noted, users of the stand-alone version of the Loftis regeneration model to predict species composition after the implementation of a regeneration harvest would naturally provide input data for a stand where some disturbance will open the overstory sufficiently to trigger regeneration. So REGEN, the software based on the Loftis model, does not need to incorporate triggering conditions for regeneration. In the NED-2 context where users might schedule shelterwood cuts to promote regeneration, the triggering conditions had to be developed to include events that occur *after* the treatment that actually triggers regeneration. Finally, regeneration might be triggered again within five or ten years of the end of a previous regeneration event. To prevent spurious regeneration events, the regeneration agent will not run the REGEN model again until at least thirty years after an earlier regeneration event.

All of these details had to be determined through knowledge acquisition with a domain expert, in this case the author of the regeneration model.

The methodology developed here can be used to integrate other regeneration models or to integrate the Loftis regeneration model with other simulators when they are included in NED-2. One simple proposed modification to REGEN is for the user to provide a tree list that represents the conditions expected for the stand following a regeneration event. Such a user model must also provide triggering conditions for the regeneration event(s). Given this base of knowledge, the regeneration agent could apply that type of simple user's model much as it now applies the Loftis model.

A problem that requires further research concerns the information about the understory and ground level vegetation that is used to drive the Loftis model in NED-2. At present, there is no model for the simulating changes in understory and ground level vegetation over time. NED-2 uses an "eternal" model which treats this information as unchanging. One approach would be to use a data-driven model, at least for the purposes of regeneration. A set of files representing standard understory and ground level inventory information for a stand of a given forest type at different ages could be developed. Then the regeneration agent would use the age and forest type of a stand to pick the appropriate file to use to develop the input for the REGEN model.

The current integration of the Loftis regeneration model into the NED-2 decision support system provides both a usable application and a proof of concept. Our plans are to continue our investigation of the feasibility of integrating growth and treatment simulation models with regeneration models along the lines suggested in this paper.

6. REFERENCES

Bugnani, David. 2005. *REGEN: A Modular and Generalized Forest Regeneration Agent.* Master's Thesis, Artificial Intelligence Center, the University of Georgia, Athens, GA 30602, USA.

Cheng, Zhiyuan. 2005. *The NED-2 Forest Ecosystem Management DSS: The Integration of the Stand Visualization, the Loftis REGEN Model, and Other Extensions.* Master's Thesis, Artificial Intelligence Center, the University of Georgia, Athens, GA 30602, USA.

Crookston, Nicholas L. Suppose: an interface to the Forest Vegetation Simulator. In: Teck, Richard; Moeur, Melinda; Adams, Judy. 1997. Proceeding: Forest vegetation simulator conference. 3-7 February 1997, Fort Collins, CO. Gen. Tech. Rep. INT-GTR-373. Ogden, UT: U.S.D.A., Forest Service, Intermountain Research Station, 1997.

Loftis, David L. 1989. Species composition of regeneration after clearcutting Southern Appalachian hardwoods. In James H. Miller (compiler), *Proceedings of the Fifth Biennial Southern Silvicultural Research Conference; November 1-3, 1988; Memphis, TN.* General Technical Report SO-74, New Orleans, LA :U.S. Department of Agriculture, Forest Service, Southern Forest Experiment Station, 253-257.

Loftis, David L. 1990. Predicting post-harvest performance of advanced red oak reproduction in the Southern Appalachians. *Forest Science* 36, 908-916.

Nute, Donald, Potter, Walter D., Cheng, Zhiyuan, Dass, Mayukh, Glende, Astrid, Maier, Frederick, Routh, Cy, Uchiyama, Hajime, Wang, Jin, Witzig, Sarah, Twery, Mark, Knopp, Peter, Thomasma, Scott, and Rauscher, H. Michael. 2005. A method for integrating multiple components in a decision support system. *Computers and Electronics in Agriculture* 49, 44-59.

Twery, Mark J., Knopp, Peter D., Thomasma, Scott A., Rauscher, H. Michael, Nute, Donald E., Potter, Walter D., Maier, Frederick, Wang, Jin, Dass, Mayukh, Uchiyama, Hajime, Glende, Astrid, and Hoffman, Robin E. 2005. NED-2: a decision support system for integrated forest ecosystem management. *Computers and Electronics in Agriculture* 49, 24-43.