The NED-2 Forest Ecosystem Management DSS: The Integration of Even-aged Red Pine, Aspen and Uneven-aged Loblolly Pine Prescription Models

by

Xia Qu

(Under the direction of Walter D. Potter)

Abstract

NED-2 is a multi-agent, intelligent, goal-driven decision support system for the forest ecosystem management developed by the USDA Forest Service and Institute for Artificial Intelligence at the University of Georgia. It has integrated many different forest management tools and models including vegetation growth and yield models, wildlife models, management models for timber, ecology, water and visual quality goals, GIS reporting tool, HTML report generating tools, etc.

This thesis describes recent work on the integration of even-aged red pine, aspen and uneven-aged loblolly pine prescription models into NED-2. These prescription models combine the expert knowledge with the existing growth and yield models to achieve the goal of automatically thinning for timber management.

Index words: blackboard architecture, knowledge-based system, FVS, prescription model

The NED-2 Forest Ecosystem Management DSS: The Integration of Even-aged Red Pine, Aspen and Uneven-aged Loblolly Pine Prescription Models

by

Xia Qu

B.M., Beijing Forestry University,
Beijing, China, 2006

A Thesis Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Master of Science

Athens, Georgia

2008

The NED-2 Forest Ecosystem Management DSS: The Integration of Even-aged Red Pine, Aspen and Uneven-aged Loblolly Pine Prescription Models

by

Xia Qu

Approved:

Major Professor: Walter D. Potter

Committee: Donald Nute
Khaled Rasheed

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2008

Dedication

This thesis is dedicated to my parents and my younger sister

Acknowledgments

I would like to thank Dr. Walter D. Potter, my major professor, Dr. Donald Nute, the Principal Investigator of the NED-2 project, and Dr. Khaled Rasheed for being my committee members and advising my study in the AI Program. Also, I would like to express my sincere gratitude and respect to Dr. Nute, for his support, concern and advice in my two years' research on the NED-2 project. Without him, I could not finish my research and this thesis.

I would like to thank Drs. Mark J. Twery, Scott Thomasma, and Pete Knopp-members of the USDA Forest Service-for their help and support on my work on the NED-2 project.

I would also like to thank my friends and classmates, as well as AI alumni-Bob, Cesar, Karan, Tom, Eric, Han -for bringing me a pleasant life at UGA.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

NED-2 is a multi-agent, intelligent, goal-driven decision support system for forest ecosystem management developed by the USDA Forest Service and Institute for Artificial Intelligence at the University of Georgia. It is a blackboard system, combined with different intelligent agents [15]. NED-2 is an integrative system that incorporates decision support tools including commercial software and components developed both by the NED-2 development team and by third parties within the Forest Service.

Typically, a blackboard system is made of three components, including a set of semi-autonomous agents which contain software routines that help to solve problems, a blackboard which contains information that agents need to solve problems, and a control mechanism which provides the agents access to the blackboard [9].

In NED-2, the blackboard consists of a Microsoft Access database, a set of Prolog clauses, and a set of Prolog routines [10]. All of the information and data are posted on the blackboard.

Agents in NED-2 contain the knowledge needed to assist decision processes. These agents do not invoke each other directly; instead, they communicate with each other through the blackboard [15]. When a task is demanded, it will be posted on the blackboard as a request. At the same time, information about the system will also exist on the blackboard. All agents will watch the blackboard continually. When an agent sees a request on the blackboard that it can fill, the agent will take control of the system and start to implement its function. During the running of the agent, the information on the blackboard might be changed or deleted while new information might also be added onto the blackboard. When the agent is done with its task, the request that triggered it will be erased from the blackboard or

changed depending on whether new tasks are requested. Then all agents will begin another round of watching for control by checking the information on the blackboard.

By using a blackboard architecture, each agent in NED-2 does not need to worry about other agents. This makes the agents more independent, and also makes NED-2 more flexible to integrate function models in the future.

Many models have been integrated into NED-2, including growth and yield models, wildlife models, and management models for timber, ecology, water and visual quality goals. Among these, growth and yield models play a central role in the forest management simulation. They can simulate the growing situation of the tree crops as time changes and project inventory data into the future for multiple treatment plans. With aid of simulation models, it is possible for NED-2 to evaluate how well different treatment plans will meet management goals over time.

Naturally, as time changes and the trees grow higher and bigger, the number of the trees in a stand will go down. This decrease in the number of trees in a stand is also considered in the simulation model and it can be represented as the mortality rate. However, the death rate of the trees in nature which is based on the mortality rate is low and the number of trees that die from natural factors is small. Since the resource for trees in a stand is limited, as time goes by and trees become higher and bigger, the growth rate will go down dramatically due to the competition for resources among those trees in the stand. In this situation, a thinning treatment is usually suggested in forest management to make sure that the remaining trees in the stand can maintain a high growth rate and grow into the desired timber as soon as possible.

Prescription models are designed to implement the functions of automatic periodical thinning based on expert knowledge [14]. They can compute the time when a thinning treatment should be applied and the volume of this thinning treatment according to the present situation and the expert knowledge.

In the next chapter, we will summarize the expert knowledge needed for the management of even-aged stands of red pine and aspen, and uneven-aged stands of loblolly/short-leaf pine. In Chapter 3, we will describe the NED simulation process. With these two chapters as background, we will explain in Chapter 4 how we integrated a prescription system into NED for even-aged stands of red pine and aspen, and in Chapter 5 how we integrated a prescription system into NED-2 for uneven-aged stands of loblolly/short-leaf pine. Chapter 6 will discuss the current status of these prescriptive systems in the NED development cycle.

Chapter 2

Managing Even-aged Red Pine, Even-aged Aspen, and Uneven-aged Loblolly/Short-leaf Pine Stands

2.1 Introduction

Red pine, aspen and loblolly/short-leaf pine are important forest types in the United States. A forest stand has various functions in different areas from ecology maintenance to commercial uses, but timber management is still the main concern for many forest managers. All research in this thesis is based on the implementation prescriptive systems for maximizing this management goal.

Sivilcultural systems can be divided into two types, for even-aged and uneven-aged stands, which are based on the age class distribution of the trees in the system. For even-aged stands, the trees in the stand are about the same age; while the trees in uneven-aged stands vary significantly in ages. In the following part of this chapter, management for even-aged stands of red pine and aspen as well as uneven-aged stands of loblolly/short-leaf pine will be discussed.

2.2 Managing Even-aged Stands of Red Pine and Aspen

Red pine is one of the most common species of trees in the Lake States region of the United States. It is a highly valuable species because of its multiple uses. Up to one million acres were planted in this species between 1970 and 2000. Another important forest species in the Lake States region is aspen. It is mostly found in the states of Minnesota, Wisconsin and Michigan with up to 13 million acres, but one third of the aspen forest grows far below potential [12].

The management of these two forest types can be implemented by growing in even-aged or uneven-aged stands, but since red pine and aspen both are shade intolerant and they grow best in full sunlight, even-aged management can give better results [2]. Therefore, even-aged management is widely used by forest managers.

In order to form an even-aged stand, all the trees in the forest need to be regenerated at the same time after a clear-cut. As the trees grow in the stand, periodic thinning of trees will be recommended to put the growth on the best trees available, remove disease and injured trees, and increase the yield of timber [2]. The thinning treatment will be triggered when a certain condition is satisfied.

2.2.1 Timber objectives and goal dbh

The management objectives considered in this thesis are to improve yields of timber and to shorten the harvest time. The related regulation methods are based on the timber objective. Typically, for red pine, seven timber objectives are considered, which include pulpwood, cabin logs, small poles, piling and large poles, small saw timber, large saw timber, and multiple products. The timber objective is related to tree diameter at breast height (DBH). For a specified timber objective, there is a goal DBH value. When the trees in the stand achieve the goal DBH, they can be harvested for this timber objective. The relations between each timber objective and the corresponding goal DBH for red pine are given in Table 2.1. Similarly, the corresponding relations between the four timber objectives and goal DBHs for aspen are also provided in Table 2.2.

When trees in the stand arrive at the goal DBH, a clear-cut will be triggered, which also implicate that the management for this generation of trees is over and a new generation of trees can be initiated in the same stand. Forest managers will decide the species of trees to be regenerated. Aspen can be naturally regenerated from sprouts. In order to make the stand a typical even-aged one, we repress the natural regeneration until a clear-cut is activated.

Table 2.1: Timber Objectives and Goal DBH for Red Pine

Timber Objective	**Goal DBH(Inches)**
Pulpwood	10
Small Poles	12
Cabin Logs	15
Piling and Large Poles	15
Small Saw Timber	15
Multiple Products	16
Large Saw Timber	20

Table 2.2: Timber Objectives and Goal DBH for Aspen

Timber Objective	**Goal DBH(inches)**
Biomass	5
Pulpwood	7
Saw timber	12
Special products	14

After the clear-cut, natural regeneration will be activated, and the number of sprouts for the new generation depends on the number of the trees that are applied the clear-cut.

Before reaching the goal DBH, trees in the stand need periodic thinning in order to maintain a high growth rate. The thinning time and thinning volume can be determined by the values of some measures of the stands.

2.2.2 Thinning for red pine

For red pine, the periodic thinning method is determined by several measures of the stand, including basal area (BA), minimum basal area (MIN_BA), maximum basal area (MAX_BA), and the desired residual basal area (Residual_BA). Basal area is a measure of stand density,

defined as the total cross-sectional area of the trees in a stand at breast height, measured in square feet per acre. It can be calculated by the formula based on the number of trees in the stand and the DBH of each tree.

$$BA = \sum_i \pi \cdot (\frac{DBH_i}{2} \cdot 0.083333)^2 = \sum_i 0.005454 \cdot DBH_i^2$$

Trees in the stand need spaces and resources to grow. If the number of trees is too small, they will grow rapidly, but the yield of timber will not be satisfactory. Minimum basal area and maximum basal area are the thresholds of basal area for the trees in the stand to keep a high growth rate and also make full use of the stand resources. They are determined by trees per acre (TPA). For a different size class of trees in the stand, the formulas for computing these measures are also different. Table 2.3 summarizes the information for the computation.

Table 2.3: Values for MIN_BA and MAX_BA in Red Pine

		Seedling		Sapling	Pole	Small Saw Timber	Large Saw Timber
Age		(0,8]	> 8				
DBH		[0,2]	[0,2]	(2,5]	(5,9]	(9,15]	> 15
TPA	MIN	400	400	400	formula(2.1)	formula(2.3)	formula(2.5)
	MAX	2000	1600	1200	formula(2.2)	formula(2.4)	formula(2.6)
BA	MIN	0	0	0	MIN_TPA*DBH*DBH*0.005454		
	MAX	160	160	160	MAX_TPA*DBH*DBH*0.005454		

Pole:

$$MIN\_TPA = 43560/((0.866 \cdot DBH + 3.73 + DBH \cdot 0.443) \cdot (DBH + 3.73 + DBH \cdot 0.443)) \quad (2.1)$$

$$MAX\_TPA = 43560/((0.866 \cdot DBH + 1.47 + DBH \cdot 0.053) \cdot (DBH + 1.47 + DBH \cdot 0.053)) \quad (2.2)$$

Small Saw Timber:

$$MIN\_TPA = 43560/((0.866 \cdot DBH + 4.18 + DBH \cdot 0.393) \cdot (DBH + 4.18 + DBH \cdot 0.393)) \quad (2.3)$$

$$MIN\_TPA = 43560/((0.866 \cdot DBH + 2.53 - DBH \cdot 0.0543) \cdot (DBH + 2.53 - DBH \cdot 0.0543)) \tag{2.4}$$

Large Saw Timber:

$$MIN\_TPA = 43560/((0.866 \cdot DBH + 4.22 + DBH \cdot 0.391) \cdot (DBH + 4.22 + DBH \cdot 0.391)) \tag{2.5}$$

$$MIN\_TPA = 43560/((0.866 \cdot DBH + 3.117 - DBH \cdot 0.0867) \cdot (DBH + 3.117 - DBH \cdot 0.0867)) \tag{2.6}$$

According to the values of the above several measures of the stand and thinning conditions, we can decide whether a thinning treatment is needed. In order to decide the thinning volume, residual basal area is used. It will tell how much of the basal area should be left after the thinning. Table 2.4 shows thinning rules for red pine.

Table 2.4: Thinning Rules for Red Pine

Timber Objective	Conditions		How to Thin(Residual BA)
Not Large Saw Timber	BA>MIN_BA+MAX_BA	Yes	1/2*BA
		No	1/2*(MIN_BA+MAX_BA)
Large Saw Timber	BA≥ 2*MIN_BA	Yes	1/2*BA
		No	MIN_BA

2.2.3 Thinning for aspen

For aspen, several more measures are used to decide thinning treatment, including minimum bi (MIN_Bi), maximum bi (MAX_Bi), mean July air temperature (Jtemp), and trees per hectare (Nn). Bi is a parameter dependent on minimum merchantable top diameter [13]. It is used to compute the variables related to thinning rules. For different timber objectives, the value settings are also different. Values for MIN_Bi and MAX_Bi with the timber objective are collected in Table 2.5. Jtemp is set to 17 °C. Nn can be computed by using tree DBH of the stand,

$$Nn = 4088000 \cdot (1 - 0.7022^{2.54 \cdot dbh}) \cdot 0.8213^{Jtemp} \cdot (2.54 \cdot dbh)^{-1.657} \tag{2.7}$$

Table 2.5: Bi Range for Different Timber Objectives in Aspen

Timber Objective	Range of bi	
	bi_max	bi_min
Biomass	1.1	0.9
Pulpwood	0.9	0.7
Saw timber	0.8	0.6
Special products	0.7	0.5

This value needs to be transformed into English units, since we use English units for other variables, so that, TPA and BA can both be obtained. The relations for these variables are displayed in Table 2.6.

Table 2.6: Values for MIN_BA and MAX_BA in Aspen

Size Class	**Seedling**	**Sapling**	**Pole**	**Small Saw Timber**	**Large Saw Timber**
DBH	[0,2.0]	(2.0,5.0]	(5.0,9.0]	(9.0,15.0]	$DBH > 15.0$
Nn	Equation 2.7				
MIN_TPA	$0.405 \cdot 147^{1-bi\_min} \cdot Nn^{bi\_min}$				
MAX_TPA	15000	$0.405 \cdot 147^{1-bi\_max} \cdot Nn^{bi\_max}$			
MIN_BA	$MIN\_TPA \cdot DBH \cdot DBH \cdot 0.5454$				
MAX_BA	100	$MAX\_TPA \cdot DBH \cdot DBH \cdot 0.5454$			

In aspen management, there are no specific thinning rules. In order to maintain a high growth rate, we can treat MAX_BA as the trigger, which means that if BA is greater than MAX_BA, a thinning treatment will be activated. Correspondingly, the target can be set as the MIN_BA; that is, when a thinning treatment is triggered, the thinning will be executed until Residual_BA is MIN_BA.

The management rules for even-aged red pine and aspen can be illustrated in Figure 2.1. Thinning rules vary based on different situations.

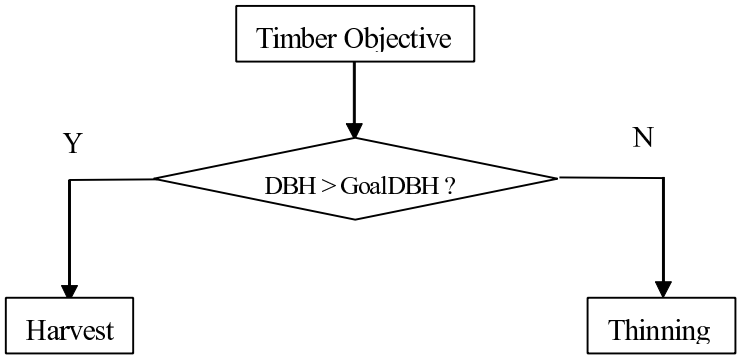

Figure 2.1: Management Rules for Red Pine and Aspen

2.2.4 Managing Uneven-aged Stands of Loblolly/Short-leaf Pine

Loblolly/short-leaf pine is an important forest type in the southern part of the United States. It has commercial value and also plays an important role in maintaining a balanced ecosystem. For the last half century, the forest industry has paid a great deal of attention to the even-aged management of loblolly. However, some private owners in the south of the United States also have a large percentage of the forest lands, which might not be properly stocked to be managed using the even-aged system. Therefore, in order to provide a periodic income, an uneven-aged management is needed [16].

There are several different kinds of uneven-aged stand based on the structure of trees' DBH class distribution. One of these is a balanced uneven-aged trees structure, which has many small trees, some medium trees, and a few large trees in the stand. It can be seen as a reverse J-shape, as shown in Figure 2.2. The management for uneven-aged loblolly/short-leaf pine stands is based on the balanced stand. If a stand is not balanced, it is better to manage the stand as an even-aged stand.

The management for uneven-aged loblolly/short-leaf pine is aimed at developing a balanced structure of size classes through thinning and then to maintain this structure through periodic harvesting. Stocking information is used to determine thinning treatments. Stocking

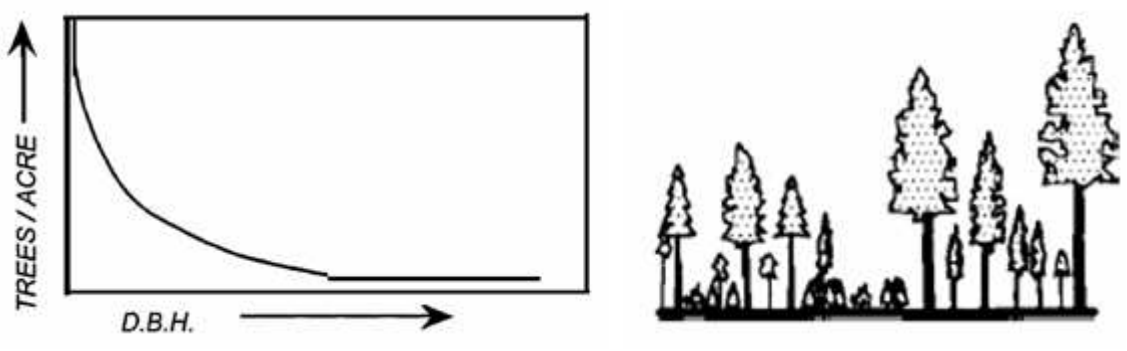

Figure 2.2: Balanced Uneven-aged Stand

is measured separately for the merchantable and sub-merchantable components [1]. Sub-merchantable trees are seedlings and saplings with a DBH of less than 5 inches; while merchantable trees are those trees that are bigger than saplings, which means that their DBH is no less than 5 inches. Using the stand merchantable basal area, we can divide stands into 4 stocking situations - inadequate stocked, under stocked, fully stocked and overstocked - which can be seen in Table 2.7

Table 2.7: Stocking Information

Stocking	**Merchantable Basal Area (square feet)**
Inadequate	$(0, 5]$
Under stocked	$(5, 45]$
Fully stocked	$(45, 75]$
Over stocked	> 75

There is no clearcut in uneven-aged stand management. In order to maintain a balanced structure of the stand and obtain a continuous timber outcome, periodic thinning is scheduled. Growing merchantable basal area (Merch_BA) between the cutting cycles will be considered for deciding whether thinning should be adopted for an uneven-aged stand. An inadequately stocked stand has enough space to grow; so no thinning is needed. For

other stocking situations, thinning will be activated. Table 2.8 shows rules for thinning in uneven-aged stand of loblolly/short-leaf pine.

Table 2.8: Thinning Rules for Uneven-aged Loblolly/short-leaf Pine

Stocking	**Residual_BA**
Inadequate	No Thinning
Under stocked	$BA - 75\% \cdot Growth\_Merch\_BA$
Fully stocked	$BA - (Merch\_BA - 45)$
Over stocked	

Chapter 3

Simulating Treatment Plans in NED-2

NED-2 is a multi-agent, intelligent, goal-driven decision support system for forest ecosystem management developed by the USDA Forest Service and Institute for Artificial Intelligence at the University of Georgia. It integrates many different forest management tools and models including vegetation growth and yield models, wildlife models, management models for timber, ecology, water and visual quality goals, a GIS reporting tool, and HTML report generating tools.

3.1 NED-2 planning module

A key feature of NED-2 is that it can simulate the growth of a forest under alternative sivilcultural treatment plans [11]. In NED-1, the first generation of the NED system, the user could only work with inventory data. NED-1 did not include any models for growing forests or for simulating the user's treatment plans. In NED-2, we have incorporated several variants of the Forest Vegetation Simulator, a set of models for simulating growth and treatment plans for forests in different regions of the United States. By simulating different treatment plans for the same management unit, users can evaluate the different plans and decide which plan best meets their management goals. In the following chapters, we will explain how prescriptive systems can develop and run simulations for treatment plans for even-aged red pine and aspen stands and for uneven-aged loblolly/short-leaf pine stands. But first we need to understand how a user can develop his own treatment plan and then use FVS to simulate it.

The first step in the simulation process for NED-2 is to input inventory data for the management unit. A management unit can be divided into several stands. Each stand will represent a part of the management unit that has similar characteristics and that will be managed together according to a single treatment plan. For each stand, data includes physical characteristics of the stand and an inventory of trees taken on one or more plots in the stand. Inventory is the basis for implementing all processes and functions in NED-2. It reflects the observed situation for each stand in the management unit at a specific point in time. For our discussion in this chapter, we will use a simple management unit with two stands called North Stand and South Stand.

Typically, inventory for different stands in the management unit is taken in different years [8]. In order to simplify the plan simulation process, all plans need to start at the same year. Therefore, before a treatment plan can be developed, users must pick a baseline year which is no earlier than the latest inventory year for all stands in the management unit. In order to generate data for the baseline year, growth of the trees in each stand needs to be simulated from the inventory year to the baseline year. Figure 3.1 shows the NED-2 user interface with the Develop Baseline function active. Notice that the interface has four sections: the caption and menu bars at the top plus three panes. The top left pane always shows a menu of NED-2 functions. We will call this the functions menu. The bottom left pane displays information and further options for the function chosen in the functions menu. We will call this the function control pane. The large pane on the right displays information generated by the function we have chosen and allows us to interact with this information. We will call this the action pane. In this case, the action pane displays a table showing all of the stands in the management unit and all years for which data are available up to the baseline year. When this table is first displayed, only the years for which inventory data has been collected for one or more stands is displayed. Here inventory for 1999 was entered for the North Stand and 2001 for the South Stand. By default, the latest year when inventory is available is highlighted in yellow and this will become the baseline year for any plans the

user develops. In our example, the user has added the year 2008 to the table making it the baseline year. By clicking on the Generate Baseline function in the functions menu, the user can prompt NED-2 to generate simulated date for all cells in the table except gray cells.

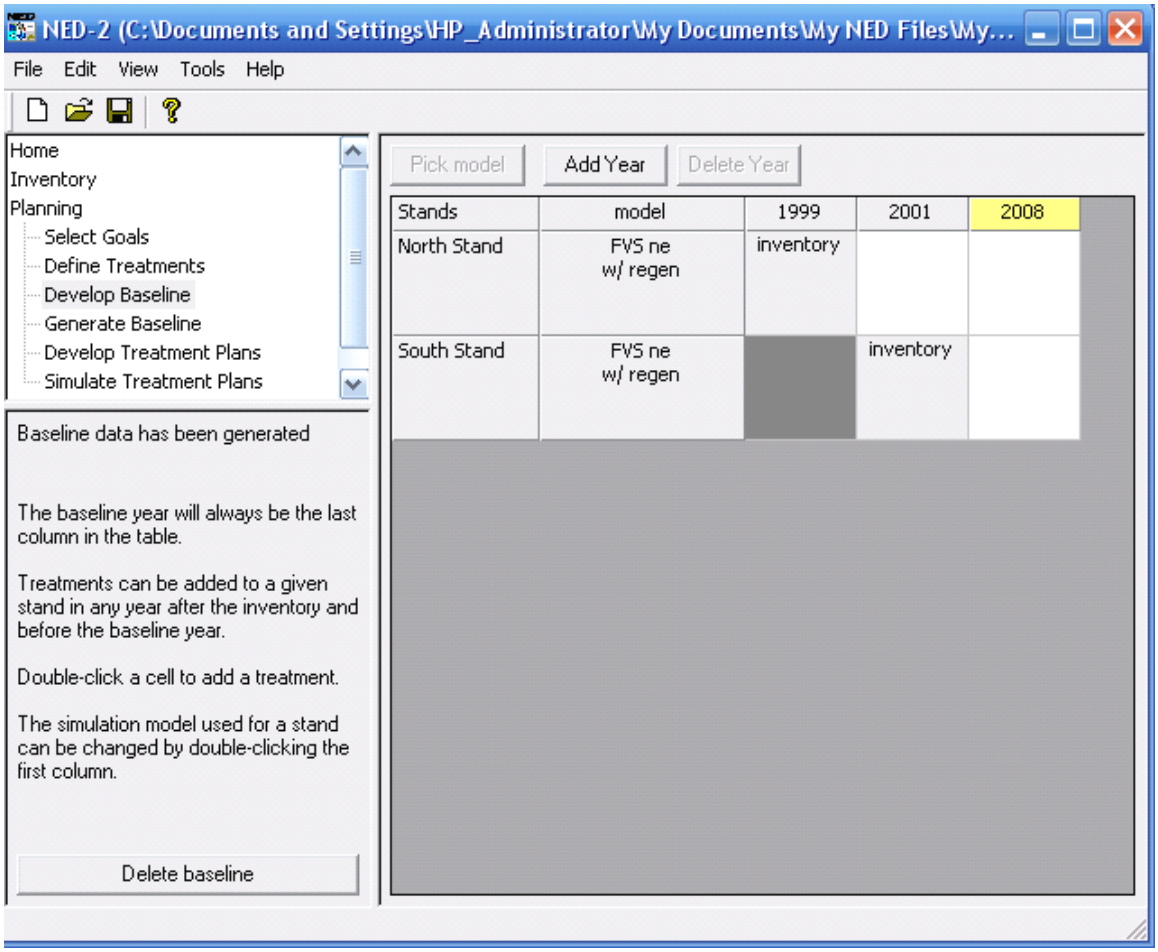

Figure 3.1: Baseline Year Generated for Two Stands with Different Inventory Years

After baseline year generation, users can create simulation plans and add plan years to them. A plan year is a time for which the growth data for trees will be recorded during simulation. Normally, plan years will represent growing cycles of a set length such as every five or ten years. But the user can set up his growing cycles to be different lengths. A desired treatment will be put into the plan year by the user. Figure 3.2 shows part of the NED-2 window for the Develop Treatment Plans function. Here we see the table for a 30-year plan for two stands with 10-year growth cycles. The plan starts from the baseline year of 2008 and

ends in the final year of 2038. The user builds his treatment plans by double-clicking on a cell and selecting the treatment he wants to include in his plan from a pop-up dialog. When he does this, an icon appears in the cell to show the treatment that has been selected. This plan, contains an icon for the North Stand, in the year of 2028 that stands for a thinning treatment; while for the South stand, an icon in 2018 stands for a clear cut treatment.

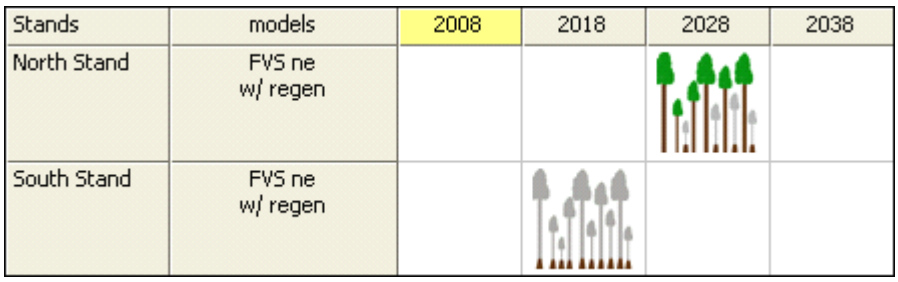

Figure 3.2: Simulate Plan

Based on these settings, a simulation plan can be implemented by running the selected growth and yield models. In this example, the Northeast variant of FVS has been selected for both stands.

3.2 NED-2 DATA MODEL

All data entered in NED-2 are kept in a Microsoft Access database. This database is called the NED-2 working file and contains several tables. In NED-2, a treatment plan is referred to as a scenario. Information for all plans is stored in the database tables [*Scenarios*] and [*Scenario_designs*] (see Figures 3.3 and Figure 3.4). Plans are identified by their SCENARIO numbers, which are assigned automatically. One special scenario does not represent a user plan. Scenario 0, the baseline scenario, represents all of the data for the years prior to and including the baseline year. In Figure 3.3, we see three scenarios: the baseline scenario and two treatment plans defined by the user.

For each stand in a plan, simulation sequence is controlled by SEQUENCE numbers, which are increasing positive integers for each year after the baseline year. For years from the baseline year backward to the inventory year, SEQUENCE is assigned a decreasing

	SCENARIO	scenario_id	scenario_desc	scenario_date_created
+	0	Baseline	BaselineYear=2008; BaselinePreTreatment=FALSE	07/29/2008
+	1	Grow Only	Plan created 07/29/2008 at 12:56:17	07/29/2008
+	2	Simulation with Treatments	Plan created 07/29/2008 at 12:56:33	07/29/2008
▶		ID		

Figure 3.3: Scenarios Table

SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
0	0	-3	0	inventory	inventory_entered	1999
0	0	-2	6	inventory	pseudo_plots	1999
0	0	-1	8	grow	grow	2001
0	0	0	9	grow	grow	2008
0	1	-2	1	inventory	inventory_entered	2001
0	1	-1	7	inventory	pseudo_plots	2001
0	1	0	10	grow	grow	2008
1	0	1	14	grow	grow	2018
1	0	2	15	grow	grow	2028
1	0	3	16	grow	grow	2038
1	1	1	11	grow	grow	2018
1	1	2	12	grow	grow	2028
1	1	3	13	grow	grow	2038
2	0	1	-999	grow	grow	2018
2	0	2	-999	grow	grow	2028
2	0	3	-999	treatment	FVS_ne_2	2028
2	0	4	-999	grow	grow	2038
2	1	1	-999	grow	grow	2018
2	1	2	-999	treatment	FVS_ne_1	2018
2	1	3	-999	grow	grow	2028
2	1	4	-999	grow	grow	2038
▶						

Figure 3.4: Scenario_designs Table

integer starting from 0. Even if the baseline year is the same as the inventory year for a stand, there will be two different sequence numbers for the stand. This is because NED-2 does some pre-processing on the plot data for each stand before plans are simulated. The baseline year is always represented by the sequence number 0.

The snapshot is a key concept in the NED-2 data model. It represents what a stand looks like at a particular point in time under a given treatment plan [11]. Initially, all snapshots are set to -999 except those for inventory years which are assigned unique integers. A snapshot number of -999 in the [*Scenario_designs*] table indicates that no data has been generated

for this stand and plan at this point in time. After simulation, each -999 is replaced by a unique value by the simulation agent. In any year when treatments are included, two snapshots will be created for the same year, one representing growth since the last plan year (the pre-treatment snapshot), and the other representing the stand after the treatment (the post-treatment snapshot). The snapshot number links together data in different tables in the NED-2 working file. The data in Figure 3.4 with SCENARIO value 2 reflects information for the plan in Figure 3.2 before its simulation.

Information about physical stand characteristics is also needed for NED-2. This information is kept in the [*Stand_header*] table. This table includes the stand area, the site index, and the site species. The site index and site species together provide the growth simulation model with information about the stand's productivity. The site index indicates how high a tree of the site species will grow in 50 years. Some information of the [*Stand_header*] table is shown in Figure 3.5.

		STAND	stand_id	stand_area	stand_site_spp	stand_site_index
	+	0	North Stand	138.600006	QUAL	57
	+	1	South Stand	40.5	QUPR2	68
▶			ID			60

Figure 3.5: Stand_header Table

Detailed information about the trees for each stand in the management unit is stored in the [*Overstory_obs*] and [*Understory_obs*] tables. These data are grouped by snapshot numbers, which reflect trees' situation at some particular time under different treatment plans. Figure 3.6 shows [*Overstory_obs*] data for the inventory year for our two stands North Stand and South Stand. Information in this table includes the overstory tree id generated by the system, species, and dbh, and so on. Table [*Understory_obs*] contains the same kind of information for trees that are in the understory. After simulation, results generated for trees through simulating growth and treatments are written back into these two tables under new snapshot numbers.

	SNAPSHOT	CLUSTER	OVER_OBS	tree_obs_id	tree_spp	tree_dbh	tree_stems_per	tree_ba
+	0	0	1	1: 2	QUPR2	15	8.1487330863045	1.22718463030859
+	0	0	2	1: 3	QUPR2	25	2.93354391106962	3.40884619530165
+	0	0	3	1: 4	QUAL	27	2.51504107601991	3.97607820219984
+	0	0	6	1: 7	FRAM2	14	9.35441298172711	1.0690141668466
+	0	0	7	1: 8	QUCO2	19	5.07885026154713	1.96894956240623
+	0	0	8	1: 9	QUCO2	21	4.1575168807676	2.40528187540484
+	0	0	9	1: 10	QUAL	17	6.34416935784953	1.57625048070748
+	0	1	0	2: 1	QURU	8	28.6478897565393	0.349065850398889
+	0	1	1	2: 2	QUAL	19	5.07885026154713	1.96894956240623
+	0	1	2	2: 3	QUAL	20	4.58366236104628	2.18166156499306
+	0	1	3	2: 4	QUAL	17	6.34416935784953	1.57625048070748
+	0	1	4	2: 5	ACRU	6	50.9295817894032	0.196349540849375
+	0	1	6	2: 7	QUAL	20	4.58366236104628	2.18166156499306
+	0	1	7	2: 8	QUAL	17	6.34416935784953	1.57625048070748
+	0	1	8	2: 9	QUAL	19	5.07885026154713	1.96894956240623
+	0	1	9	2: 10	QUAL	10	18.3346494441851	0.545415391248264
+	0	2	0	3: 1	NYSY	18	5.6588424210448	1.76714586764438
+	0	2	1	3: 2	LITU	16	7.16197243913482	1.39626340159556
+	0	2	2	3: 3	QURU	19	5.07885026154713	1.96894956240623
+	0	2	3	3: 4	QUAL	19	5.07885026154713	1.96894956240623
+	0	2	4	3: 5	LITU	22	3.7881507116085	2.6398104936416
+	0	2	5	3: 6	LITU	16	7.16197243913482	1.39626340159556
+	0	2	6	3: 7	LITU	20	4.58366236104628	2.18166156499306
+	0	2	7	3: 8	NYSY	17	6.34416935784953	1.57625048070748
+	0	2	8	3: 9	LITU	10	18.3346494441851	0.545415391248264
+	0	2	9	3: 10	QUAL	17	6.34416935784953	1.57625048070748
+	0	2	10	3: 11	TSCA	9	22.6353696841792	0.441786466911094
+	1	0	0	1: 1	QUPR2	8	28.6478897565393	0.349065850398889
+	1	0	1	1: 2	QUPR2	21	4.1575168807676	2.40528187540484
+	1	0	2	1: 3	QUPR2	9	22.6353696841792	0.441786466911094
+	1	0	3	1: 4	QUPR2	14	9.35441298172711	1.0690141668466
+	1	0	4	1: 5	QUVE	20	4.58366236104628	2.18166156499306
+	1	0	5	1: 6	QUPR2	28	2.33860324543178	4.27605666738639
+	1	0	6	1: 7	QUPR2	22	3.7881507116085	2.6398104936416
+	1	0	7	1: 8	CAAL27	16	7.16197243913482	1.39626340159556
+	1	0	8	1: 9	QUAL	24	3.1830988618377	3.14159265359
+	1	0	9	1: 10	QUPR2	15	8.1487330863045	1.22718463030859
+	1	0	10	1: 11	QUVE	25	2.93354391106962	3.40884619530165
+	1	0	11	1: 12	QUVE	24	3.1830988618377	3.14159265359
+	1	0	12	1: 13	QUAL	11	15.152602846434	0.659952623410399
+	1	0	13	1: 14	QUPR2	19	5.07885026154713	1.96894956240623
+	1	0	14	1: 15	QUAL	28	2.33860324543178	4.27605666738639
+	1	1	0	2: 1	QUPR2	15	8.1487330863045	1.22718463030859

Figure 3.6: Overstory_obs Table

3.3 Forest Vegetation Simulator (FVS)

The simulation process in NED-2 is implemented by the growth and yield models. FVS is one of the simulation models that have been integrated into NED-2. It uses forest inventory data and simulation models to project the growth of forest stands under different conditions [15].

FVS was developed by the USDA Forest Service in the 1970s, and since then it has been used in the U.S. forest management field. FVS is a family of forest growth simulation models, and has more than 20 variants dealing with forest types in the different regions of the United

States [15]. Among these FVS variants, 8 of them have now been integrated into NED-2, including Northeast, Southern, Central States, Lake States, Blue Mountain, East Cascades, Inland Empire and Pacific Northwest Coast variants.

In FVS, there are two ways to schedule a treatment in a simulation plan. The most common way, which has already been included in the simulation process in NED-2, is to assign a specified treatment in a desired year (see the example in Figure 3.2). In this method, the treatment is deterministic, that is, it only happens in the assigned year with a specified treatment illustrating the thinning. The FVS treatment statement using this method for the plan in Figure 3.2 for the two stands (thinning to residual basal area of 60 for North Stand in 2028 and a clear-cut for South Stand in 2018) is given below,

North Stand:

```
thinbba          2028       60.0        1.00         0.0       999.0         0.0       999.0
```

South Stand:

```
thindbh          2018        0.0       999.0        1.00         ALL         0.0         0.0
```

The above treatment statement consists of a keyword followed by 7 parameters. There are several kinds of treatments which describe different thinning methods. In the given example, thinbba and thindbh are used, which stand for thin from below to basal area target and thin from a dbh range respectively. The 7 parameters give detail information about thinning settings, including thinning year, thinning trees dbh range, residual basal area and/or thinning species, and so on [7].

The other way to schedule a treatment in a simulation plan is conditional scheduling, which can be implemented by the FVS event monitor. It specifies a set of conditions and a set of treatments. The conditions describe an event. When a certain condition is satisfied, that is, the event is triggered, a corresponding treatment will be scheduled. Therefore, by checking conditions, different treatments will be scheduled dynamically.

Specifically, an event starts from the keyword IF, followed by a logical expression which can be arithmetic operators, logical operators (GT, LT, EQ, AND, OR), and certain vari-

ables. The keyword THEN, which follows the logical expression, signals the scheduled activity when the logical expression is true. And the event ends at keyword ENDIF [6]. Besides these keywords, another one called COMPUTE will be needed to define a variable which can be expressed as a function of event monitor variables [5].

A simple example that helps illustrate the FVS event monitor method is: If before-thinning basal area (BBA) is greater than 120 and cycle year is greater than 2018, then thin from below to a residual basal area 60. This conditional thinning can be simulated with the following FVS statement,

```
IF
BBA GT 120 AND YEAR GT 2018
THEN
THINBBA          0.0        60
ENDIF
```

When FVS simulates a plan with the above FVS statement inside, these conditions will be checked for each cycle year in the plan. If conditions are satisfied for some cycle year, the described thinning treatment will be scheduled.

Compared with the first method that has been implemented in NED-2, the FVS event monitor is more complicated in scheduling a treatment. For a simulation plan in NED-2, treatment is predefined either by the system or by users. When a treatment is selected for a plan in a certain year, it is specified. Users will have enough information about this treatment, for example, they will know that the treatment is limited to happen in that specified year, and when it happens they will also have an idea about the volume of trees that will be cut. While for the second method by using the FVS event monitor, treatments are scheduled under conditions and users have no idea about the exact simulation process, for example, they do not know the exact year that a thinning or harvest treatment happens. Also, more than one cycle year can be scheduled with the same treatments as long as the conditions for that year are satisfied.

For the prescription models we are going to develop, the FVS event monitor is needed for the implementation of automatically dynamic treatments.

3.4 NED-2 RUNS FVS

Running FVS needs two files as input, a tree data file (*.fvs) and a keyword file (*.key). These two files can be produced by predicate *mdb2fvs/6* in NED-2. After running, four different output files will be generated. Among these output files, tree list file (*.trl) contains detailed trees information and needs to be interpreted and stored back into the NED-2 database. This can be implemented by predicate *fvs2mdb/14*.

Tree data file

The tree data file (*.fvs) contains information about the trees in the stand for the baseline year, which includes all tree records from tables [*Overstory_obs*] and [*Understory_obs*] in the baseline year. Based on this, tree growth can be predicted. Specifically, sub-predicate *mdb2fvs_fvsFileFormat/4* in *mdb2fvs* is in charge of writing this file. It will look for information including the tree species, dbh, stems per acre and tree alive for each tree in the stand in the baseline year in the two tables. Figure 3.7 is an example of an fvs file.

```
        COLUMNS
         1         2
12345678901234567890
====================
0001001   11.1CO 15.8
0001002    2.1CO 26.2
0001003    2.1WO 28.1
0001004   13.1WA 14.9
0001005    7.1SO 19.9
0001006    5.1SO 22.3
0001007    8.1WO 17.8
0001501  132.1DW  3.1
0001502  119.1AB  4.6
0001503  120.1AB  4.6
0002001   37.1RO  8.8
0002002    7.1WO 19.9
0002003    5.1WO 21.0
0002004    8.1WO 17.8
0002005   66.1RM  6.6
0002006    5.1WO 21.0
0002007    8.1WO 17.8
0002008    7.1WO 19.9
0002009   24.1WO 10.8
```

Figure 3.7: fvs File

In the example fvs file in Figure 3.7, each row is a record of an individual tree in table [*Overstory_obs*] or [*Understory_obs*]. Each record consists of plot id (columns 1-4), tree id (column 5-7), stems per acre (column 8-13), tree history (column 14), the species alpha code (column 15-17), and the dbh (column 18-21). Trees in [*Overstory_obs*] and [*Understory_obs*] have almost the same kind of information representation as displayed in the fvs file but tree id in column 5 is different (0 for overstory trees and 5 for understory).

Keyword file

The key file (*.key) controls the simulation process. It is written in a specific format and can be produced by the sub-predicate *mdb2fvs_keyFileFormat/8* for each stand in the plan.

```
                                    COLUMNS
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
================================================================================
Comment
Starting year for simulation is 2008
Ending year for simulation is   2038
Min and Max inventory years are 2008 2008
Common cycle length is          30
End
*
StdIdent
Stand North Stand at Sample Forest
*
InvYear         2008
ModType            1
*
StdInfo                             114       304         7         5
Design          -1.0       1.0       0.1         2
SiteCode          WO        57
*
NumCycle           3
TimeInt            1        10
TimeInt            2        10
TimeInt            3        10
*
NoTriple
thinbba         2018      60.0      1.00       0.0     999.0       0.0     999.0
CutList            0        3.         1                            0
TreeList           0        3.         1         1         0         0         0
*
Open               2
0000102.fvs
TreeData           2         1
Close              2
*
SPLabel
   All, &
   !StandsInNoDefinedGroup
Process

STOP
```

Figure 3.8: key File

Typically, a keyword file (Figure 3.8) contains 4 parts, the header (row 5-22), the time interval (row 23-27), the treatments (row 28) and the body (row 29-42)[8]. For each part, a corresponding predicate is responsible for locating information in several database tables as well as writing information into the keyword file. The third part, treatments, is the most important part in the keyword file. It decides the treatments in the plan and helps to achieve the desired goals. Two ways of scheduling treatments discussed above are included in this part. It is produced by the predicate *mdb2fvs_writeKeyFile_C/3* with the information given from table [Scenario_designs].If there is no treatment used in a simulation plan, this part will be ignored.

Tree list file

Tree list file (*.trl) is the most important output file generated by FVS. It contains detailed information about all individual trees that are projected at any future time under a given scenario [8]. An example of a tree list file is given in Figure 3.9.

In the tree list file, each record consists of a header and a body. The header is the first line of the record starting with '-999'. It gives general information about the cycle in the stand. Two kinds of records are contained in the tree list file, a tree list and a cut list. They can be distinguished by the parameter in column 81 in the header ('T' for the tree list while 'C' for the cut list).The body part contains detailed information about each individual tree in the cycle year (trees started with 'ES' stand for new regenerated ones). For each year in the plan, if there is no treatment, a unique tree list record will be displayed; otherwise, two records for the same cycle year, respectively representing the tree list before the treatment and the cut list in the treatment.

The generated data in the tree list file will be stored back into a NED-2 database file. The information relevant to NED-2 includes the tree number (column 6-8), the species alpha code (column 15-16), the plot id (column 27-29), stems per acre (column 31-38), and the dbh (column 49-53) [8]. The predicate *fvs2mdb/14* is in charge of writing these data back into

```
                                                  COLUMNS
                                                                                                      1         1         1
         1         2         3         4         5         6         7         8         9         0         1         2
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
===============================================================================================================================
-999   19    1 2018 Stand                    NONE NE 6.21 08-08-2008 13:35:34 T 10 0.2000000E+01 10.19.2000 00000000000
    2005   15 RM 26  3  0   2   27.893     5.107   7.0  0.35  55.7  1.9 46 15.4  0  18.14     0    4.1    0.0      0.0  0  0   0
    1502    9 AB 40  3  0   1   44.718    14.782   5.0  0.41  47.9  2.7 37 10.2  0  12.28     0    0.0    0.0      0.0  0  0   0
    1503   10 AB 40  3  0   1   45.643    14.357   5.0  0.41  47.9  2.7 37 10.2  0   7.37     0    0.0    0.0      0.0  0  0   0
    1004    4 WA 42  3  0   1    6.319     0.181  15.6  0.61  71.0  1.0 42 33.1  0  36.67     0   33.5   25.3    160.8  0  0   0
    1003    3 WO 55  3  0   1    0.966     0.034  29.3  1.06  68.9  0.9 40 40.2  0 100.00     0  153.4  141.4    900.9  0  0   0
    1007    7 WO 55  3  0   1    3.737     0.263  18.4  0.56  68.0  1.0 39 26.1  0  59.00     0   52.7   41.1    252.8  0  0   0
    2002   12 WO 55  3  0   2    3.336     0.164  20.7  0.71  68.4  1.0 39 29.0  0  77.42     0   67.9   61.5    381.2  0  0   0
    2003   13 WO 55  3  0   2    2.392     0.108  22.0  0.93  68.5  1.0 40 30.9  0  82.44     0   85.0   69.8    436.3  0  0   0
    2004   14 WO 55  3  0   2    3.719     0.281  18.4  0.56  68.0  1.0 39 26.1  0  53.51     0   52.7   41.1    252.8  0  0   0
    2006   16 WO 55  3  0   2    2.399     0.101  22.0  0.94  68.5  1.0 40 30.9  0  87.49     0   85.1   69.8    436.6  0  0   0
    2007   17 WO 55  3  0   2    3.754     0.246  18.4  0.56  68.0  1.0 39 26.1  0  48.05     0   52.7   41.1    252.7  0  0   0
    2008   18 WO 55  3  0   2    3.323     0.177  20.7  0.70  68.4  1.0 39 29.0  0  71.23     0   67.9   61.5    380.7  0  0   0
    2009   19 WO 55  3  0   2   10.609     1.391  11.4  0.51  63.9  1.4 36 16.7  0  30.03     0   17.3    9.2     49.2  0  0   0
    1005    5 SO 60  3  0   1    3.290     0.210  20.7  0.69  63.5  0.6 45 40.5  0  65.09     0   67.5   52.7    324.4  0  0   0
    1006    6 SO 60  3  0   1    2.400     0.100  23.6  1.19  63.5  0.6 48 45.8  0  93.29     0   90.2   81.7    506.7  0  0   0
    1001    1 CO 64  3  0   1    5.004     0.496  16.4  0.57  62.1  0.6 43 24.3  0  42.54     0   39.8   24.5    159.0  0  0   0
    1002    2 CO 64  3  0   1    0.958     0.042  27.5  1.12  62.4  0.5 49 40.4  0  96.42     0  113.1  101.4    696.8  0  0   0
    2001   11 RO 67  3  0   2   15.943     2.557   9.3  0.45  61.3  1.7 41 17.8  0  24.10     0    9.3    0.0      0.0  0  0   0
    1501    8 DW104  3  0   1   53.257    12.743   3.2  0.09  33.9  3.0 33  7.7  0   2.36     0    0.0    0.0      0.0  0  0   0
-999   10    2 2018 Stand                    NONE NE 6.21 08-08-2008 13:35:34 C 10 0.2000000E+01 10.19.2000 00000000000
    2005   15 RM 26  3  0   2   27.893     0.000   7.0  0.35  55.7  1.9 46 15.4  0  18.14     0    4.1    0.0      0.0  0  0   0
    1502    9 AB 40  3  0   1   44.718     0.000   5.0  0.41  47.9  2.7 37 10.2  0  12.28     0    0.0    0.0      0.0  0  0   0
    1503   10 AB 40  3  0   1   45.643     0.000   5.0  0.41  47.9  2.7 37 10.2  0   7.37     0    0.0    0.0      0.0  0  0   0
    1004    4 WA 42  3  0   1    6.319     0.000  15.6  0.61  71.0  1.0 42 33.1  0  36.67     0   33.5   25.3    160.8  0  0   0
    2004   14 WO 55  3  0   2    2.933     0.000  18.4  0.56  68.0  1.0 39 26.1  0  53.51     0   52.7   41.1    252.8  0  0   0
    2007   17 WO 55  3  0   2    3.754     0.000  18.4  0.56  68.0  1.0 39 26.1  0  48.05     0   52.7   41.1    252.7  0  0   0
    2009   19 WO 55  3  0   2   10.609     0.000  11.4  0.51  63.9  1.4 36 16.7  0  30.03     0   17.3    9.2     49.2  0  0   0
    1001    1 CO 64  3  0   1    5.004     0.000  16.4  0.57  62.1  0.6 43 24.3  0  42.54     0   39.8   24.5    159.0  0  0   0
    2001   11 RO 67  3  0   2   15.943     0.000   9.3  0.45  61.3  1.7 41 17.8  0  24.10     0    9.3    0.0      0.0  0  0   0
    1501    8 DW104  3  0   1   53.257     0.000   3.2  0.09  33.9  3.0 33  7.7  0   2.36     0    0.0    0.0      0.0  0  0   0
-999   22    2 2028 Stand                    NONE NE 6.21 08-08-2008 13:35:34 T 10 0.2000000E+01 10.19.2000 00000000000
ES020011   11 RM 26  2  0   1   27.893     0.000   1.0  0.00   5.5  0.0 63  3.9  0   0.24     0    0.0    0.0      0.0  0  0   0
ES020012   12 RM 26  2  0   1   27.893     0.000   1.0  0.00   5.5  0.0 63  3.9  0   0.96     0    0.0    0.0      0.0  0  0   0
    2008    1 WO 55  3  0   2    3.153     0.169  21.4  0.68  69.1  0.7 40 30.1  0  39.58     0   80.3   66.0    410.8  0  0   0
    1003    3 WO 55  3  0   1    0.932     0.034  30.4  1.02  69.6  0.6 41 42.0  0 100.00     0  165.5  152.4    974.7  0  0   0
    2006    4 WO 55  3  0   2    2.302     0.097  23.0  0.89  69.3  0.7 41 32.4  0  73.48     0   92.8   76.1    478.8  0  0   0
    1007    7 WO 55  3  0   1    3.505     0.231  19.1  0.63  68.7  0.8 40 27.2  0  12.32     0   56.8   44.2    274.1  0  0   0
    2004    8 WO 55  3  0   2    0.734     0.052  19.1  0.64  68.7  0.8 40 27.2  0  14.65     0   56.8   44.3    274.2  0  0   0
    2003    9 WO 55  3  0   2    2.288     0.104  23.0  0.88  69.3  0.7 41 32.4  0  62.85     0   92.7   76.0    478.4  0  0   0
    2002   10 WO 55  3  0   2    3.179     0.157  21.4  0.68  69.1  0.7 40 30.2  0  52.30     0   80.4   66.0    411.3  0  0   0
ES020015   15 WO 55  2  0   1   10.609     0.000   1.0  0.00   5.5  0.0 63  4.5  0   0.47     0    0.0    0.0      0.0  0  0   0
```

Figure 3.9: tree list File

database tables. Because of the new generated data, most of the database tables need to be inserted with new records. While for table [*Scenario_designs*] it just needs to be updated, for scenarios are given before the simulation process.

After the simulation, the “View Stand Snapshots” model in NED-2 can display information for the pre-treatment and post-treatment of each year in the plan by consulting the database. This kind of information is similar to an FVS tree list file but much easier for users to read.

Chapter 4

Implementing Prescriptions for Even-aged Red Pine and Aspen Management

The management of even-aged red pine and even-aged aspen are both based on stand timber objectives. When a specified timber objective is selected, a corresponding set of prescription rules will be applied to the trees in the stand. According to these prescription rules, a series of conditions will be checked. When a condition is satisfied, the corresponding treatment will be triggered. By executing these prescription treatments, the desired timber objective can be achieved efficiently. The prescription model that we developed can implement this management process.

Each even-aged prescription model is a knowledge-base combining prescription rules with timber objectives. These rules describe a relation between the growing situations of all trees in the stand and the appropriate number of these trees. Upper and lower bounds for the number of trees are explicitly stated under different circumstances. When the number of the trees in the stand stays in this range, these trees can make good use of the site resources and maintain a high growth rate. Otherwise, when the stand density becomes too high, corresponding thinning treatments will be provided. With the aid of these rules, the desired timber objective can be achieved with good yields but in less time.

To get the treatments from the prescription model, plan simulation is also involved in this process. Similarly to the simulation process we discussed in NED-2 in the previous chapter, we need to use a growth and yield model to simulate the growth of the stands. Since red pine and aspen are two kinds of forest types that are widely planted in the Lake States region of the United States, we are going to use the LS variant of FVS as the growth and yield model.

Compared to a treatment plan in NED-2, a prescription model has an additional knowledge base which contains the prescription rules. These rules are the integrated knowledge from experts' years of experience and research. Based on these rules, the prescription model can compute corresponding treatments automatically.

The explicit prescription rules for red pine and aspen are different, but the implementation processes are similar. Therefore, we will only describe the implementation for even-aged red pine management in detail.

4.1 FMAS

The Forest Management Advisory System (FMAS) is a decision support system for the management of even-aged stands of red pine and aspen. It integrates knowledge based systems for the treatment prescriptions of red pine and aspen with growth and yield simulation models [17]. The project was supported by the U.S.D.A. Forest Service, designed by Dr. Donald Nute of the University of Georgia and Dr Michael Rauscher of the Forest Service, and implemented by graduate students Guojun Zhu and Yousong Chang in 1995 [3]. The knowledge base used in the system was provided by the research scientists in the U.S.D.A. Forest Service.

In FMAS, rules for the management of red pine and aspen are written in a format which can be used by a forward chaining inference engine to compute the missing data. The forward chaining rules in FMAS are in the following format:

```
Rule(fc, Environment, Name, Value, (Conditions then Actions), Explanation)
```

Parameters in this rule format respectively stand for the kind of inference engine used, the rule set, the rule's unique name, rule strength, content, and explanation. The prescription rules for the management of even-aged red pine and aspen stands in NED are extracted from these FMAS rules. Some examples are given below.

1. Rules for goal DBH

```
rule(fc,objective,dbh_goal_2,true,
    (
    goal(dbh_goal),
    product(‘Large sawtimber‘),
        then,
    thin_ba_x(0.7),
    dbh_goal(20),
    remove(dbh_goal)
    ),[‘For Large sawtimber, the thinning basal area multiplier‘,
      ‘is  0.7, dbh_goal is estimated to be 20.‘
     ]).
```

This rule provides an easily measured harvest standard (goal DBH) for a particular timber objective. In this example, for large saw timber, the goal DBH value is 20 inches. Similar rules are also provided for other timber objectives. When the DBH for the stand achieves its desired timber objective goal DBH, a clear-cut will be activated and the management for the red pine stand will terminate.

2. Thinning Rules

```
rule(fc,implement,treat_8,true,
    (
    goal(‘Thin‘),
    product(‘Large sawtimber‘),
    min_ba(Minba),
    ba(Ba),
    test(Minba > Ba/2),
        then,
    residual_ba(Minba),
    remove(‘Thin‘)
    ),
    [‘When treatment is thin, and product is Large sawtimber,if min_ba ‘,
       ‘is greater than  basal area/2, then, residual_ba estimated to be ‘,
       ‘as same as min_ba.‘
      ]).
```

Rules in this category contain the detailed thinning treatments under all circumstances. In this example, for the timber objective of large saw timber, it gives a treatment (thinning trees until Residual_BA is MIN_BA) under a certain circumstance (BA is less than 2*MIN_BA).

Also, treatments for other circumstances are provided. Based on these rules, treatments can always be concluded from tree information.

3. Variables Rules

```
rule(fc,stand_eval,size_4,true,
    (
    goal(size),
    dbh(D), test(D>5.0), test(D=<9.0),
    X1 is 43560/((0.866*D+1.47+D*0.053)*(D+1.47+D*0.053)),
    places(X1,0,MxT),
    X2 is 43560/((0.866*D+3.73+D*0.443)*(D+3.73+D*0.443)),
    places(X2,0,MnT),
    X3 is MxT*D*D*0.005454,
    places(X3,0,MxB),
    X4 is MnT*D*D*0.005454,
    places(X4,0,MnB),
        then,
    max_tpa(MxT),
    min_tpa(MnT),
    max_ba(MxB),
    min_ba(MnB),
    size(‘Pole‘),
    remove(size)
    ),
    [‘When dbh is between 5.0 and 9.0, max_tpa is estimated to be ‘,
    ‘~M43560/((0.866*Dbh+1.47+Dbh*0.053)*(Dbh+1.47+Dbh*0.053)),‘,
    ‘~Mmintpa is estimated to be ‘,
    ‘~M43560/((0.866*Dbh+3.73+Dbh*0.443)*(D+3.73+Dbh*0.443)), ‘,
    ‘~Mmax_ba is estimated to be ‘,
    ‘~MMax_tpa*Dbh*Dbh*0.005454, and min_ba is estimated to be ‘,
    ‘~MMin_tpa*Dbh*Dbh*0.005454, and size is estimated to be Pole. ‘
    ]).
```

These rules determine the values for variables that are involved in the treatments rules. This example gives the value for MAX_TPA, MIN_TPA, MAX_BA and MIN_BA when DBH is in the range 5.0 to 9.0 inches.

These three examples show the most common forms of the rules used in FMAS. Rather than return to the original scientific papers from which these rules were derived, we backward-engineered the rules in FMAS to extract the rules used in NED. All of the NED rules for red pine are summarized in English in Chapter 2.

4.2 FVS event monitor

The prescription model provides treatments automatically based on the desired timber objective. The model was implemented by applying the prescription rules obtained from FMAS to the growth and yield model (FVS).

In FVS, a simple way of scheduling a treatment in a simulation plan is to schedule the specific treatment in the desired year. In this way, the treatment is observable for users. In the prescription model, however, treatments are determined by checking different conditions and computing related parameters. In this model, it is impossible to predict the years when a treatment needs to be executed or the volume to cut. Therefore, the simple method of scheduling treatments for specific years cannot be used to implement the prescription model. Another way of scheduling a treatment is conditional scheduling, using the FVS event monitor. We implemented the red pine and aspen prescription models using this method.

In the area of forest management, a certain number of variables are commonly used in deciding treatments. In FVS, some of these variables are already pre-defined, for example, BBA (Before-thinning basal area per acre) and ATPA (After-thinning trees per acre). Users can use these variables to define conditional treatment plans directly. However, there are other variables that appear in the red pine and prescription rules but are not defined in FVS, for example, MIN_BA (minimum basal area) and MIN_TPA (minimum trees per acre). In order to make sure that the prescription model can schedule treatments exactly based on the prescription rules, these variables need to be defined in FVS. The keyword COMPUTE is used to define this kind of variable so that they can be used to define conditional treatments in the same way as other variables that are predefined in FVS.

The prescription rules we obtained from FMAS can be rewritten in the FVS keyword format. With the aid of the FVS event monitor, treatment prescriptions can be implemented by the simulation when the triggering conditions are satisfied.

Figure 4.1 shows part of the treatments statements for red pine in the FVS keyword format for the multiple products timber objective. In rows 1-12, variables following the keyword COMPUTE are defined, so that, they can be used in the following FVS conditional treatment definition. Rows 14-36 list several events that trigger thinning treatments. Each event starts its conditions from the keyword IF, and its consequence of corresponding prescription treatment follows after the keyword THEN. In the consequence part of the event monitor, only those FVS keywords that include a parameter indicating the cycle year in which the corresponding treatment is to be performed can be used with conditional functions.

```
1  COMPUTE            0
2  TrtYear = 2018
3  BA_TARG = 1/2*BBA
4
5  MIN_TPAP = 43560/((0.866*BADBH+3.73+BADBH*0.443)*(BADBH+3.73+BADBH*0.443))
6  MAX_TPAP = 43560/((0.866*BADBH+1.47+BADBH*0.053)*(BADBH+1.47+BADBH*0.053))
7  MIN_BAP = MIN_TPAP*BADBH*BADBH*0.005454
8  MAX_BAP = MAX_TPAP*BADBH*BADBH*0.005454
9  BA_TRIGP = MIN_BAP + MAX_BAP
10 BA_TARGP = 1/2*BA_TRIGP
11 ......
12 END
13 *
14 IF
15 YEAR GE TrtYear AND &
16 BADBH GE 0 AND BADBH LE 2.0 AND &
17 BBA GT 160
18 THEN
19 THINBBA          0.0     PARMS(BA_TARG, 1.0, 0.0, 999.0, 0, 999)
20 ENDIF
21 *
22 IF
23 YEAR GE TrtYear AND &
24 BADBH GE 0 AND BADBH LE 2.0 AND &
25 BBA LE 160
26 THEN
27 THINBBA          0.0     PARMS(80, 1.0, 0.0, 999.0, 0, 999)
28 ENDIF
29 *
30 ......
31 IF
32 YEAR GE TrtYear AND &
33 BADBH GT 16
34 THEN
35 THINBBA          0.0       0.0
36 ENDIF
```

Figure 4.1: FVS treatment definition in keyword format

4.3 Integration

The difference between the prescription model and the already existing treatment plan simulation process in NED-2 is that the prescription model is knowledge based and it also uses a new way of scheduling the treatment. In order to integrate this new model, NED-2 needs to be able to deal with the knowledge base as well as the conditional scheduling of treatments. Two predicates, *mdb2fvs* and *fvs2mdb*, which are responsible for the communication between FVS and the NED-2 database, need to be changed for this integration.

4.3.1 mdb2fvs

We have discussed a little about *mdb2fvs* in the previous chapter. It contains the code to produce two kinds of FVS input files: a tree data file (*.fvs) and a keyword file (*.key) from the NED-2 database. The structure of the tree data file for the prescription model is the same as it is for the standard simulation process in NED-2. Therefore, no changes are needed for the production process of the data file. However, the simulation processes are different for the prescription model and the standard NED-2 treatment plan. Since the keyword file contains the instructions that FVS uses to simulate treatments, changes are needed in the code that writes this file for simulations where the prescription models will be applied.

In NED-2, information in the keyword file statements for FVS is taken mainly from the database [*Scenario_designs*] table. For the prescription model, there is also a similar relation between the keyword file and this table. Figure 4.3 shows the database table [*Scenario_designs*] for the prescription plan in Figure 4.2. In the *scenario_designs_treatment_id* field in the table, we see the selected timber objective. In fact, we have a separate rule for each timber objective for red pine, and also for aspen. The user selects the prescriptive treatment that includes the timber objective he wants to achieve.

A keyword file consists of 4 parts (the header, the time interval, the treatments and the body), produced by 4 different sub-predicates. The third part (the treatments) controls the process of treatments execution and is the most important in the keyword file. From our

Stands	models	2008	2018	2028	2038	2048	2058
stand 1	FVS ls w/ regen						

Figure 4.2: Plan with prescription model

	SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
	0	0	-1	0	inventory	inventory_entered	2008
	0	0	0	1	inventory	pseudo_plots	2008
	1	0	1	-999	grow	grow	2018
	1	0	2	-999	prescription	Red pine prescription - Small poles	2018
	1	0	3	-999	grow	grow	2028
	1	0	4	-999	grow	grow	2038
	1	0	5	-999	grow	grow	2048
	1	0	6	-999	grow	grow	2058
▶							

Figure 4.3: Table Scenario_designs

previous discussion of *mdb2fvs*, we know that the difference for each treatment plan comes from the treatments part in the keyword file. This treatments part is produced by the sub-predicate *mdb2fvs_writeKeyFile_C/3*. The auxiliary predicate used to add the conditional prescription to the keyword file in this process is defined as following,

```
mdb2fvs_writeKeyFile_C_x([[Prescription,TrtYear,_]|Rest],Variant,TmpTxt,TrtTxt) :-
    prescription_KB(Prescription,KB),
    absolute_file_name(kbs(KB),Path),
    !,
    ensure_loaded(Path),
    prescription(Prescription,ForestType,TrtYear,Txt),
    cat([TmpTxt,Txt],NewTmpTxt,_),
    mdb2fvs_writeKeyFile_C_x(Rest,Variant,NewTmpTxt,TrtTxt).
```

To implement the prescription model, the corresponding prescription rules need to be loaded and added into the treatments part. The predicate *prescription_KB/2* locates the file that keeps all of the prescription rules so that the system can load these rules. Specifically, the

prescription rules are stored in the redpine.kb knowledge base file under the folder KBs. They are written in the following form:

```
prescription(Treatment,'PIRE',TrtYear,Text)
```

The first parameter is the timber objective written in the form of 'Red pine prescription - *' (* is the name of the treatment objective:, Small poles, for example), and 'PIRE' is the species code for red pine. The third parameter TrtYear states the beginning year that the prescription rule starts to apply to the stand. The last parameter is the content of the prescription rule. This is the exact text of the set of variable definitions and conditional treatments that will be inserted into the keyword file (see Figure 4.1).

After loading the prescription rules, the set of rules for the selected timber objectives can be retrieved and written into the treatments part of the keyword file. Whenever the Prolog agent sees a red pine or aspen prescriptive treatment in the [*Scenario_designs*] table, it uses this process to produce the keyword file containing the conditional scheduling treatments to be interpreted by the FVS event monitor.

4.3.2 FVS2MDB

Similar to *mdb2fvs*, *fvs2mdb* is also used for communication between the NED-2 database and FVS. It is responsible for interpreting new data in the tree list file which is generated by FVS and writing these data back into the NED-2 database.

Since we use the FVS event monitor for scheduling conditional treatment, the thinning actions cannot be predicted before the running of the simulation. Treatments produced by the FVS event monitor are not indicated in the original [*Scenario_designs*] table as they would be if the user scheduled a specific treatment in a specific year. In this case, new snapshots which represent these new treatments need to be created and inserted into the correct positions within the database table. The order of the snapshots is based on the cycle years in which the treatments are scheduled to happen. At the same time, those snapshots

with cycle years after these treatments' cycle years need to have new SEQUENCE values calculated for them. Therefore, all snapshots are sorted in the order of their cycle years.

Figure 4.3 shows the [*Scenario_designs*] table settings before simulation of a prescriptive treatment for red pine. Since a timber objective (Red pine prescription - Small poles) is selected in year of 2018, only one new snapshot which is related to this prescription is added into the snapshots list in the same scenario. After simulation, all treatments that are triggered by satisfying the required conditions need to be included in the [*Scenario_designs*] table so that snapshots representing all of the tree growth and treatments can be updated for the related database tables.

The process of updating snapshots is implemented by predicate *fvs2mdb*, specifically, its auxiliary predicate *fvs2mdb_updateScenario_designs/6*. For the simple case where specific treatments are scheduled for specific years, this predicate is in charge of creating new snapshot records and inserting the corresponding snapshot number into the [*Scenario_designs*] table at the point where the treatment appears. For the prescription model, since the treatments are triggered at a year later than the year where the user entered the prescription, the code also needs to be able to add complete new records to the [*Scenario_designs*] table which represent tree treatments generated by the prescription rules.

Part of the code is changed in the predicate *fvs2mdb_updateScenario_designs/6* in NED-2 to accomplish this. Therefore, new records representing the prescription treatments can be inserted into the [*Scenario_designs*] table. Specifically, the predicate checks the FVS output tree list file. When it comes to a cut list which represents that a prescription treatment is triggered, it will change the [*Scenario_designs*] table by inserting a new record which represents this treatment, and change the SEQUENCE values for the records that are scheduled to happen after this treatment. In order to distinguish prescription treatments and other treatments, we also insert the value 'prescription' in the *scenario_designs_function_id* field in the [*Scenario_designs*]. This field has the value 'grow' or 'treatment' for standard, non-prescriptive NED-2 plans.

Figure 4.4 shows the updated [*Scenario_designs*] table before a prescription treatment is processed. Figure 4.5 shows that a new record related to a prescription treatment which happened in 2028 has been inserted into the database table. Also, the SEQUENCE values for the records that are scheduled to happen after the year 2028 are revalued.

SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
0	0	-1	0	inventory	inventory_entered	2008
0	0	0	1	inventory	pseudo_plots	2008
1	0	1	2	grow	grow	2018
1	0	2	-999	prescription	Red pine prescription - Small poles	2018
1	0	3	3	grow	grow	2028
1	0	4	-999	grow	grow	2038
1	0	5	-999	grow	grow	2048
1	0	6	-999	grow	grow	2058

Figure 4.4: [Scenario_designs] partly updated before a new record is added

SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
0	0	-1	0	inventory	inventory_entered	2008
0	0	0	1	inventory	pseudo_plots	2008
1	0	1	2	grow	grow	2018
1	0	2	-999	prescription	Red pine prescription - Small poles	2018
1	0	3	3	grow	grow	2028
1	0	5	-999	grow	grow	2038
1	0	6	-999	grow	grow	2048
1	0	7	-999	grow	grow	2058
1	0	4	4	prescription	prescription	2028

Figure 4.5: a new snapshot record is added into [Scenario_designs] table

Figure 4.6 shows the [*Scenario_designs*] table after all prescription treatments have been added into the table. Even though the prescription starts from 2018, no treatments happened in this year. Therefore, the snapshots for growing and treatment in this year are the same and SNAPSHOT value for the prescription snapshot is set to be the same as the growing snapshot. The prescription starts in 2018, however, the real prescription treatments are triggered in 2028, 2038 and 2048.

After inserting all of the new records that correspond to the triggered treatments (Figure 4.6) into the [*Scenario_designs*] table, the NED-2 "View Stand Snapshots" function in the NED-2 interface can display all snapshots including prescription treatments in the appropriate order.

SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
0	0	-1	0	inventory	inventory_entered	2008
0	0	0	1	inventory	pseudo_plots	2008
1	0	1	2	grow	grow	2018
1	0	2	2	prescription	Red pine prescription - Small poles	2018
1	0	3	3	grow	grow	2028
1	0	5	5	grow	grow	2038
1	0	7	7	grow	grow	2048
1	0	9	9	grow	grow	2058
1	0	4	4	prescription	prescription	2028
1	0	6	6	prescription	prescription	2038
1	0	8	8	prescription	prescription	2048

Figure 4.6: [Scenario_designs] table completed with updating records

4.4 Aspen Regeneration

Besides the prescription rules, another difference between the prescription models for aspen and red pine is that for aspen there is a regeneration process. When a harvest or clear-cut is executed, the management model for a red pine stand terminates; while for an aspen stand, the site is open and sprouts will come out from the roots of harvested plants. The prescription rules continue to apply to the newly generated aspen.

FVS has a built-in regeneration model and regeneration occurs throughout the simulation. However, for the prescription model we want to suppress this "establishment" regeneration since it interferes with the computation of the variables that trigger treatments. The commands to do this are included in the text that is written into the keyword file. A new regeneration process will be triggered when the clear-cut is executed. The number of newly regenerated aspens is estimated by a linear interpolation of the number of trees that were clear-cut.

Chapter 5

Implementing Prescriptions for Uneven-aged Loblolly/Short-leaf Pine Management

5.1 Introduction

Loblolly/short-leaf pine is an important forest type in the Southern United States. It not only provides commercial value to forest owners, but also plays an important role in maintaining a balanced ecosystem. For the last half century, the forest industry has paid much attention to the even-aged management of this forest type. However, many private forest owners in the Southern United States also have forest lands. Many of these forests are not well stocked to be managed as even-aged stands. In order to provide these private forest owners a periodic income, another type of management that aims for uneven-aged stands is needed [16]. Researchers from the USDA Forest Service have summarized the expert experience and research on uneven-aged silviculture of loblolly/short-leaf pine stands. These results are a good guide for the management of uneven-aged loblolly/short-leaf pine [1].

Since expert knowledge is available for the management of uneven-aged loblolly/short-leaf pine, it is possible to implement a dynamic computer advisory system which can make full use of this knowledge. In 2000, an expert system for this was developed by Mingguang Xu. This system realized the function of management for uneven-aged loblolly/short-leaf pine with the aid of expert knowledge. However, forest owners may need other functions and a single expert system may not satisfy their needs. NED-2 is a decision support system which can provide different functions to meet various forest management goals. It is beneficial to integrate the prescription model for the management of uneven-aged loblolly/short-leaf pine into NED-2.

Similar to the prescription models we discussed in the previous chapter, the prescription model for the management of uneven-aged loblolly/short-leaf pine also uses the FVS simulator but with the southern variant. However, there is an important difference between the even-aged and uneven-aged. For the first two models, all variables that the prescription rules depend on can be fully computed or expressed in FVS, which makes it possible to use the conditional scheduling function of the FVS event monitor to implement these two models. But for the uneven-aged loblolly/short-leaf pine prescription model, one of the most important variables used in the prescription rules is the difference between the merchantable basal area for the present year and that of its previous cycle year, and this value cannot be computed or expressed in FVS.

Because of the difference we discussed above, the prescription model for the management of uneven-aged loblolly/short-leaf pine needed to be implemented in a different way. We will discuss the implementation process in the following part of this chapter.

5.2 Simulation Schema

Since the prescription model for uneven-aged loblolly/short-leaf pine cannot be implemented by just using FVS, we will separate the whole process into two sub-processes: condition checking and simulation. These two sub-processes will be called repeatedly.

Condition checking is an auxiliary process for simulation. Specifically, it provides treatment information by computing the growth value of merchantable basal area and deciding when and how to thin the trees based on the knowledge base. The simulation process will be implemented in the FVS simulator, combining corresponding treatment information obtained from the previous condition checking. For one scenario plan of the prescription model, these two processes will be executed more than once.

To illustrate the whole process, we make a scenario plan with n cycles of years. Suppose we select the 1^{st} cycle year to start the prescription model. For the following series of figures from Figure 5.1 to Figure 5.4, the black line arrow indicates the year when the prescription

model starts, while the red dotted arrow indicates the year when conditions are satisfied and the specified treatment is applied.

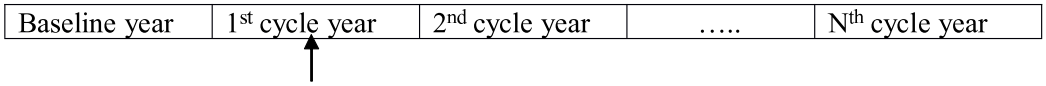

Figure 5.1: a scenario plan with prescription model inside

The prescription model will first run the FVS simulator for all cycle years without any treatments. This process is in fact a pure growth simulation. Using the simulated growth data for each cycle year, we can compute the change in merchantable basal area for any cycle year. With this information available, NED-2 can proceed to the conditions checking sub-process. This sub-process will check the treatment conditions for each cycle year starting from the one when the prescription model is selected until it finds the first cycle year which satisfies the condition. Correspondingly, a treatment will be triggered by the condition and the detailed information for the treatment will be computed based on the knowledge base. Treatment information will be copied back into FVS and the simulation sub-process will restart with this obtained treatment included. These two sub-processes will be repeated until all cycle years have been checked. For each round of the simulation sub-process, all treatments obtained up to this round will be included in the FVS treatment part.

The process for the scenario plan in the loblolly/short-leaf pine prescription model is described as below.

1. Use FVS to simulate the entire scenario plan without any treatment inside (Figure 5.1)

2. Condition checking: find the first cycle year when a prescription treatment is triggered by checking the conditions outside of FVS (In this example, some treatment-triggering condition is satisfied in the 2^{nd} cycle year)

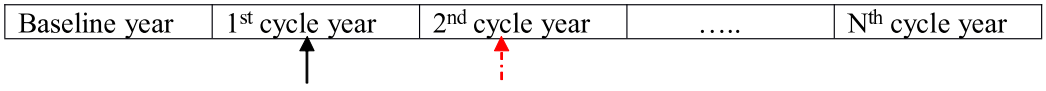

Figure 5.2: prescription model: condition checking

3. Simulation: simulate the entire scenario plan again with the treatment obtained from step 2 included.

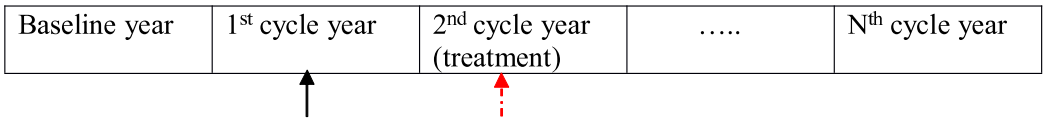

Figure 5.3: prescription model: simulation process with a treatment

4. Repeat steps 2-3 until all cycle years have been checked for the prescription rules, and for each round of running step 3, all the treatments obtained including the former ones will be kept and executed.

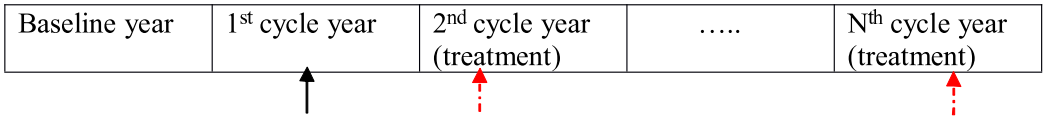

Figure 5.4: prescription model

5.3 LOBLOLLY AGENT AND IMPLEMENTATION

In order to implement the previously discussed simulation schema for the prescription model of loblolly/short-leaf pine, we introduced another agent called **loblolly** into NED-2. This agent controls the simulation process by running between the FVS simulator and NED-2 computation.

The running process for the loblolly agent can be summarized in Figure 5.5 (the dotted square indicates the loblolly agent).

When a loblolly prescription is posted in a certain cycle year within a scenario plan (Figure 5.6 *Scenario_designs* table), as the simulation starts, a request for loblolly prescription will be inserted into the blackboard requests list ahead of all others. This is done by the predicate *mdb2fvs*, specifically, by the sub-predicate *mdb2fvs_writeKeyFile_C* (Figure 5.7). When a loblolly prescription is found in the treatment list, it will first be removed from the treatment list. A new request for loblolly prescription will be inserted into the beginning of

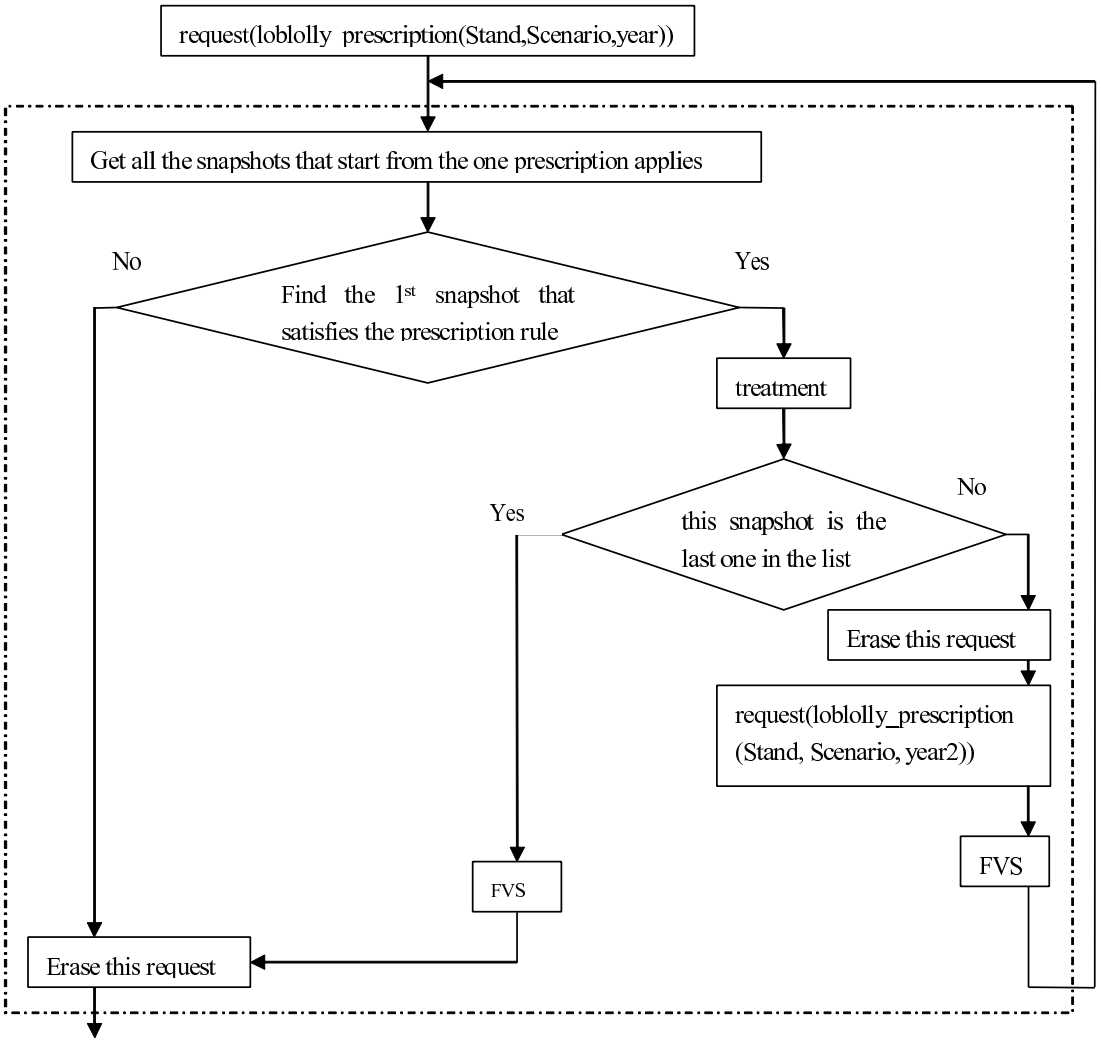

Figure 5.5: Loblolly agent

the request list on the blackboard. This new request contains detailed information about the prescription, including *scenario_plan_id* and *stand_id* as well as the starting time when this prescription applies.

Since no thinning treatment is left in the treatment list, this round of simulation will continue without treatment by the FVS simulator.

After the first round of growth simulation, the simulation agent will release control of the NED process. Now, all agents will start a new round of control watching by checking

SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
0	0	-1	0	inventory	inventory_entered	2008
0	0	0	1	inventory	pseudo_plots	2008
1	0	1	-999	grow	grow	2013
1	0	2	-999	treatment	Loblolly Prescription	2013
1	0	3	-999	grow	grow	2018
1	0	4	-999	grow	grow	2023
1	0	5	-999	grow	grow	2028

Figure 5.6: Table Scenario_designs before prescription model runs

```
mdb2fvs_writeKeyFile_C(Scenario,Stand,Treatments,Variant,TrtTxt) :-
    remove(['Loblolly Prescription',StartYear,_],Treatments,NewTreatments),
    retract(request([fvs_plan|Rest])),
    assert(request([fvs_plan,loblolly_prescription(Scenario,Stand,StartYear)|Rest])),
    mdb2fvs_writeKeyFile_C_x(NewTreatments,Variant,``,TrtTxt)
   ;
    mdb2fvs_writeKeyFile_C_x(Treatments,Variant,``,TrtTxt).
```

Figure 5.7: mdb2fvs_writeKeyFile_C

request information on the blackboard. Since the first item in the requests to do list is the request for loblolly prescription, that is, *loblolly_prescription(Scenario,Stand,StartYear)*, the loblolly agent will take control.

First, the loblolly agent will find all snapshots (starting from the cycle year that the prescription model applies to the end of the scenario plan) that need to be checked by the prescription rules. Among all of these snapshots, it will look for the first one that satisfies treatment conditions in the knowledge base and find the corresponding treatment. The treatment conditions use the change in merchantable basal area in one cycle year. Triggered treatment will be written into the blackboard as a fact for the future use in the following form,

```
fact(treatment([snapshot = current],Treatment(Residual_BA)), _, _, _ ).
```

The treatment is in the form of a structure with the FVS keyword thinaba and a parameter for this keyword representing the desired residual basal area. This information decides the thinning treatment and can be used in the next round of the FVS simulation subprocess. The new treatment will be rewritten into the database table *Scenario_designs* (Figure 5.8) by loblolly agent.

	SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
	0	0	-1	0	inventory	inventory_entered	2008
	0	0	0	1	inventory	pseudo_plots	2008
	1	0	1	2	grow	grow	2013
	1	0	2	2	prescription	Loblolly Prescription:'thinaba'101.51	2013
	1	0	3	3	grow	grow	2018
	1	0	4	4	grow	grow	2023
	1	0	5	5	grow	grow	2028
▶							

Figure 5.8: Prescription treatment information in Scenario_designs table

If the cycle year for the obtained treatment is the same as the year when the prescription model starts, the record with "Loblolly Prescription" filled in the field of *scenario_designs_treatment_id* will be replaced by the new treatment; otherwise, a new record will be inserted into this table as a treatment snapshot. When a new record is inserted, the SEQUENCE values for records whose *scenario_designs_year* are after the year of this new record need to be recalculated so that simulation will happen in the right order. The treatment prescription in the *scenario_designs_treatment_id* column is in the form of "Loblolly Prescription:thinaba***" (***represents the value of the residual basal area as shown in Figure 5.8).

With the treatment prescription indicated in the *Scenario_designs* table, FVS can be called by the loblolly agent to resimulate the plan. In this process, two files must be created as input for FVS running. Similar to the previous prescription model, no change is needed for the tree list file; while for the key file, some work needs to be done to translate the prescription treatment in the *Scenario_designs* table into the format that FVS can read. This is implemented inside of the predicate *mdb2fvs_writeKeyFile_C_x*, which is in charge of the treatment keyword part. Two clauses,

```
cat(TreatmentList,TreatmentParts,[22,1,7,1]),
TreatmentList = ['Loblolly Prescription:',_,Treatment,_,BA]
```

make sure that we can get treatment information from the form 'Loblolly Prescription:thinaba*** ' which exists in the database. With this information extracted from the database, the keyword file entry can be created for the corresponding prescriptive treatment. In order to maintain the structure of the uneven-aged loblolly stand and get a sustainable periodic income, some amount of loblolly pines planting can be prescribed after this thinning treatment. An example of a prescriptive treatment in the keyword file is displayed in Figure 5.9.

```
NoTriple
thinaba         2013    101.51      1.00       0.0     999.0       0.0    999.0
ESTAB           2013
PLANT           2013        LP        50
END
CutList            0        3.         1                            0
TreeList           0        3.         1         1         0         0         0
```

Figure 5.9: Treatment part in the keyword file

In the simulation sub-process, predicate *mdb2fvs* will also check whether there are some cycle years left after the last prescription treatment year; if there are, the old loblolly prescription request will be replaced by a new one in the beginning of the request list on the blackboard. The starting year for this new loblolly prescription request will be the first year after the previously generated prescription treatment. After FVS simulation, the loblolly agent will finish its task and release control of the NED process. If the first request in the request list on the blackboard is still a request for loblolly prescription, the loblolly agent will be activated again and the two sub-processes will be repeated again. When the loblolly agent cannot find a snapshot that satisfies the prescription rules, it will release control of processing.

For each round that the loblolly agent runs, it always looks forward for new snapshots that have not been assigned any treatments. If a treatment is triggered, it will be added

into the *Scenario_designs* table with a specific snapshot. All previous generated treatments will still be kept in the table. In the last round of simulation, this table (Figure 5.10) will contain all prescription treatments that are prescribed to trees in the stand based on the prescription rules.

SCENARIO	STAND	SEQUENCE	SNAPSHOT	scenario_designs_function_id	scenario_designs_treatment_id	scenario_designs_year
0	0	-1	0	inventory	inventory_entered	2008
0	0	0	1	inventory	pseudo_plots	2008
1	0	1	2	grow	grow	2013
1	0	2	3	prescription	Loblolly Prescription:'thinaba'101.51	2013
1	0	3	4	grow	grow	2018
1	0	4	5	prescription	Loblolly Prescription:'thinaba'83.01	2018
1	0	5	6	grow	grow	2023
1	0	6	7	prescription	Loblolly Prescription:'thinaba'76.56	2023
1	0	7	8	grow	grow	2028
1	0	8	9	prescription	Loblolly Prescription:'thinaba'67.25	2028

Figure 5.10: Table scenario_designs in the final round

With all prescription treatments available, the final round of FVS simulation will get results for this complete set of prescriptive treatments.

Chapter 6

Conclusions

This thesis discusses the implementation process for integrating prescription models of even-aged red pine and aspen stands, and uneven-aged loblolly/short-leaf pine stands, into NED-2. These prescription models can plan automatic periodical thinning treatments based on related expert knowledge.

In NED-2, FVS was used to simulate a predefined treatment on a specific year which cannot be changed during the simulation process in a scenario plan. For the prescription models, a conditional treatment function is also integrated into NED-2, which makes it possible to do the dynamic treatment prescription based on the prescription rules.

However, these prescription models will actually not be turned on until NED-3. The reason for this is that the NED user interface has to be modified to delete all the post-prescriptive-treatment snapshots from the working file if the user's plan is changed in any way. Those changes to the NED user interface are not included in NED-2 and will not be implemented until the next version of NED. However, this did not interfere with the testing program for the prescription models. We were able to test all three prescription models in a modified version of NED-2. This allowed us to insure that the models actually worked as intended. The only function that was missing was the ability for the user to "undo" a simulation based on one of the prescription models if he changed a treatment plan for a stand.

Bibliography

[1] Baker, James B.; C., M. D.; Guldin, James M.; Murphy, Paul A.; Shelton, Michael G. (1996) Uneven-aged silviculture for the loblolly and shortleaf pine forest cover types. General Tech. Report SO-1 18, SRS, USDA, Forest Service, Asheville, NC 28802.

[2] Benzie, J. W. (1977) Manager's handbook for red pine in the north-central states. General Technical Report NC-33. St. Paul, MN, U.S. Dept. of Agriculture.

[3] Chang, Y. (1994) A Red Pine Forest Management Advisory System. Artificial Intelligence Center. Athens, The University of Georgia. Master: 51.

[4] Cheng, Z. (2005) The NED-2 Forest Ecosystem Management DSS: The Integration of The Stand Visualization System, The Loftis REGEN Model, and Other Extensions. Artificial Intelligence Center. Athens, The University of Georgia. Master: 71.

[5] Crookston, N. L. (2002) User's Guide to the Event Monitor: Part of the prognosis Model Version 6.

[6] Dixon, G. E. (2002) Essential FVS:A User's Guide to the Forest Vegetation Simulator. Fort Collins, CO, United States Department of Agriculture, Forest Service, Forest Management Service Center: 210.

[7] Dyck, M. G. V. (2002) Keyword Reference Guide for the Forest Vegetation Simulator. Fort Collins: 107.

[8] Glende, A. (2004) The NED Forest Management DSS: The Integration of Growth and Yield Models. Artificial Intelligence Center. Athens, The University of Georgia. Master: 96.

[9] Maier, F. W. (2002) Notes on a Blackboard: Recent Work on NED-2. Artificial Intelligence Center. Athens, The University of Georgia. Master: 94.

[10] Nute, Donald; P.,W. D.; Maier, Frederick; Wang, Jin; Twery, Mark; Rauscher, H.Michael; Knopp, Peter; Thomasma, Scott; Dass, Mayukh; Uchiyama, Hajime; Glende, Astrid. (2004) NED-2: an anget-based decision support system for forest ecosystem management. Environmental Science and Artificial Intelligence 19(9): 13.

[11] Nute, Donald; J. B.; Cheng, Z; Potter, Walter D; Loftis, David; Twery, Mark; Thomasma, Scott; Knopp, Peter. (2006) Interleaving Growth and Regeneration Models in the NED-2 Decision Support System for Forest Ecosystems. iEMSs Third Biennial Meeting: Summit on Environmental Modelling and Software, Burlington, VT, July 10, 2006.

[12] Perala, D. A. (1977) Manager's Handbook for Aspen in the North-Center States: 30.

[13] Perala, D. A.; H., George E.; Jordan, James K.; Cieszewski, Christopher J. (1996) A Multiproduct Growth and Yield Model for the Circumboreal Aspens. Northern Journal of Applied Forestry 13(4): 7.

[14] Twery, Mark J.;K., P. D.;Thomasma, Scott A.; Rauscher, H.Michael; Nute, Donald E.; Potter, Walter D.; Maier, Frederick; Wang, Jin; Dass, Mayukh; Uchiyama, Hajime; Glende, Astrid; Hoffman, Robin E.. (2005) NED-2: A decision support system for integrated forest ecosystem management. Computers and Electronics in Agriculture 49.

[15] Wang, J.; Potter, W. D; Nute, D.; Maier, F; Rauscher, H. Michael; Twery, M. J; Thomasma, S; Knopp, P (2002) An Intelligent Information System for Forest Management: NED/FVS Integration. Second Forest Vegetation Simulator Conference. Fort collins, CO, Proc. RMRS-P-25, Ogden: 17.

[16] Xu, M. (2000) Design and Implementation of an Expert System (LOBLOLLY-ES) for the Management of Uneven-aged Loblolly-Shortleaf Pine Forests. Artificial Intelligence Center. Athens, The University of Georgia. Master: 26.

[17] Zhu, G. (1995) DSSTOOLS: A Toolkit for Development of Decision Support Systems in Prolog. Artificial Intelligence Center. Athens, The University of Georgia. Master: 116.