# NED-2: An Agent-Based Decision Support System for Forest Ecosystem Management

**Donald Nute[a] (dnute@uga.edu), Walter D. Potter[a], Frederick Maier[a], Jin Wang[a], Mark Twery[b], H. Michael Rauscher[c], Peter Knopp[b], Scott Thomasma[b], Mayukh Dass[a], Hajime Uchiyama[a], and Astrid Glende[a]**

[a] *Artificial Intelligence Center, The University of Georgia, Athens, GA 30605, USA*

[b] *Northeastern Research Station, USDA Forest Service, P.O. Box 968, Burlington, VT 05402-0968, USA*

[c] *Bent Creek Experimental Forest, Southern Research Station, USDA Forest Service, Asheville, NC, USA*

**Abstract:** Decision making for forest ecosystem management can include the use of a wide variety of modeling tools. These tools include vegetation growth models, wildlife models, silvicultural models, GIS, and visualization tools. NED-2 is a robust, intelligent, goal-driven decision support system that integrates tools in each of these categories. NED-2 uses a blackboard architecture and a set of semi-autonomous agents to manage these tools for the user. The blackboard integrates a Microsoft Access database and Prolog clauses, and the agents are implemented in Prolog. A graphical user interface written in Visual C++ provides powerful inventory analysis tools; dialogs for selecting timber, water, ecological, wildlife, and visual goals; and dialogs for defining treatments and building prescriptive management plans. Users can simulate management plans and perform goal analysis on different views of the management unit, where a view is determined by a management plan and a point in time. Prolog agents use growth and yield models to simulate management plans, perform goal analyses on user-specified views of the management unit, display results of plan simulation using GIS tools, and generate hypertext documents containing the results of such analysis. Individual agents use meta-knowledge to set up and run external simulation models, to load rule-based models and perform inference, to set up and execute external GIS and visualization systems, and to generate hypertext reports as needed, relieving the user from performing all these tasks.

**Keywords:** ecosystem management, model management, decision support system, knowledge based system, blackboard architecture, Prolog.

## 1. INTRODUCTION

The goals of modern managers of forested lands include not only timber production but ecological, wildlife, water, recreational, and other objectives. The personal computer puts a broad range of decision support tools at the fingertips of the forest manager, including spreadsheets and databases for recording and manipulating inventories, growth and yield simulators for projecting future conditions of forested land, optimization programs for maximizing timber production, financial analysis systems for tracking income from forest products and other uses of forested land, and geographical information systems and visualization tools to help the manager understand the impact of different management decisions. More recently, knowledge based systems have been developed to provide silvicultural prescriptions to meet specific timber goals, to diagnose forest health problems, and to evaluate whether forested lands satisfy a range of timber, wildlife, water, visual, and other goals. The effort required to identify and to utilize the appropriate decision support tools while keeping in view the myriad goals the forest manager is trying to achieve can be overwhelming. A likely result is that the forest manager will use only the few decision support tools that he feels comfortable with and will concentrate on a severely limited set of goals because he does not know how to manage for anything more.

What is needed to improve the situation? If we could provide the forest manager with an assistant who understood how to use the various decision support tools, then the manager could concentrate more of his attention on developing a more comprehensive set of management goals and on making management decisions based on the analyses provided and clarified by the management tools to better achieve this larger set of goals. The objective of the Northeast Decision Model project of the U.S.D.A. Forest Service is to develop such an assistant, or rather a group of assistants, in the form of an agent-based software system that integrates a suite of the decision support tools that are most useful to forest managers into a complete goal-driven decision support process that leads the manager through the process in a natural and intuitive way. For a summary paper providing background on intelligent software agents and their application in environmental management, see [Smith and Mackanass 1996]. This paper also cites several other papers that take an agent-based approach to developing decision support systems for different domains within environmental management. While this work collectively shows the value of such approaches, none of the systems we have found in the literature attempts to integrate as broad a range of decision support tools into as single decision process as NED-2, the system we will describe.

NED-2 is an Intelligent Information System designed to provide decision support for forest ecological system management in the eastern United States. The latest product of the Northeast Decision Model project and the successor of NED-1[1] [Twery et al. 2000], NED-2 provides a platform for integrating databases, knowledge bases, simulations, geographic information systems, visualization tools, and other knowledge sources and knowledge tools. NED-2 has an open agent-based architecture that facilitates future integration of additional knowledge sources and tools. The U.S.D.A. Forest Service developed NED-2 in collaboration with researchers from several universities.

We will describe a session with NED-2 to illustrate the NED decision process. Then we will describe the NED-2 architecture in detail. Finally, we will summarize the current status of the NED-2 prototype and the work that remains to be done to complete a full implementation of the NED-2 design.

## 2. THE NED DECISION PROCESS

The NED decision process is driven by information about the management unit, but it is also goal driven [Rauscher et al. 2000]. The first step in the process is to record inventory data into a database management system or a spreadsheet. The next step is to clearly define the management goals for the forested land to be managed. In case studies conducted by the NED team, landowners showed strong interest in incorporating wildlife, recreational, ecological, and other non-timber goals when they realized there were effective, economical ways to pursue these goals. A goal can only be effectively pursued if there are tools available for evaluating whether the goal will be satisfied by a particular management plan. In the process of making management decisions, a user may define alternative goal sets and use them to evaluate several management plans.

Managers typically think of the land they manage as a collection of distinct *stands* where each stand occupies an area of uniform composition, age, and condition. For effective management, the user will need to record the species and size of each overstory tree on at least one sample plot on each forested stand. Spreadsheets or other tools can be used to calculate basal areas, timber volumes, and other stand statistics that will drive many of the management decisions.

To manage for recreational value, wildlife, and other non-timber goals, a manager must also record information about the shrub and ground layers and about other physical characteristics of each stand such as the presence of ponds or streams, the amount of coarse woody debris available, and the presence of snags or hollow trees.

A user may not have current inventory on all stands, and the most recent inventory for different stands may not have been taken in the same year. To establish a common baseline year for planning purposes, the user may need to use a growth and yield simulator that can grow each stand to the baseline year and simulate any historical treatments that may

[1] NED-1 and other software from the NED project is available free of charge from http://www.fs.fed.us/ne/burlington.ned.

have been performed on each stand since inventory was taken.

Management happens through treatments applied to the land. The development of management plans is guided by the goals that have been selected. Silvicultural treatments fall into categories such as fertilizing, thinning, and harvesting. But within these categories, specific treatments can be designed in many ways. The user might use a silvicultural knowledge based system to recommend treatments to include in a plan (see for example [Nute *et al.* 1995].) A treatment plan might be developed as a text document using a word processor.

Once the user has developed one or more possible treatment scenarios, he can set up a growth and yield simulator to simulate the plan on each stand in his management unit. Simulators typically require that tree data be converted to a special format before it can be used by the system. Different simulators will often require specific codes to represent tree species in the data. Control files with special codes denoting treatments and parameters for treatments must also be provided. If the system does not simulate treatments, then the user will have to grow each stand up to the point a treatment is to be performed, use some other tool to simulate the treatment, and grow the stand again using the post-treatment data. Some simulators such as FVS have an interface [Crookson 1997] that assists the user in building these control files. When the simulations are run, the simulator will produce output files that will typically be in a format and using tree species codes specific to that simulator. The user will want to get this data back into whatever spreadsheet or database management tool he is using to store his inventory data.

The experiences of the NED research team suggest that managers sometimes do not try to manage for anything except timber goals because they do not realize that there are tools available to manage for wildlife, visual, or other goals. To manage for some of these goals, they must first be *operationalized*. For example, the user does not know how to manage for owl habitat or for fall color unless he can translate these goals into specific variables that can be measured or observed in the forest. Earlier work by the NED team produced NED-1, a decision support system that includes a knowledge based system that analyzes a wider range of forest management goals into specific conditions that must be satisfied at the stand or management unit level. To use NED-1 to analyze the output from his simulation runs, the user must create a separate data file in the NED-1 format for each treatment plan and each year of data in the treatment plan he wants to examine. Each year of each treatment plan must be loaded into NED-1 and analyzed separately. This is what has been done in the NED case studies that have been conducted. [Rauscher 2001] Alternatively, the user could try to analyze his plans manually, looking at each stand for percentages of a specific kinds of ground cover, distributions of plant species, presence of temporary or permanent ponds, or whatever other combination of requirements might correspond to a wildlife, water, visual, or ecological goal.

Once the user has built, simulated, and performed goal analysis on one or more possible management plans, he may want to view the results in various ways. Some useful tools are geographical information systems and visualization tools such as the Stand Visualization System [McGaughy 1997], or EnVision [McGaughy 1998]. Once again, data will have to be converted into a format that each of these tools can use.

The user will probably want to do some financial analysis on the different plans he is considering. For this, he will need data on costs and returns. Finally, the user will want to put together a document summarizing the results of this process.

The decision process we have described would use a text editor, spreadsheet and/or database management system, growth and yield simulator, treatment simulator (which will often be the same as the growth and yield simulator,) silvicultural expert systems, knowledge based goal analysis tools, financial analysis tools, geographical information systems, and visualization tools. Tools of these kinds are now available, and new tools are being built all the time. But the average forest manager, such as the private land owner or the forestry consultant, is unlikely to make use of more than a few of these tools. Each of these tools requires a considerable investment of time and effort to learn how to use the tool. Once the user is familiar with the tool, its interface, and its required data formats, there remains the challenge of converting data between a large number of formats and managing all the data that are generated. Finally, the user must be able to digest all this information, make intelligent decisions based on it, and formulate a report or planning document that summarizes the data and the decisions in cogent form.

This is what an intelligent model management system should do for the forest manager. It should provide a single, intuitive interface for the user to learn. It should know how to set up and run a variety of third-party decision tools. It should provide a way for the user to build goal sets and design alternative treatment plans easily and quickly It should be able to translate data from one format to another transparently and with minimal action by the user. It should be able to use the goals and plans the user has specified to control the different simulations and goal analyses it performs. It should manage all the data generated so the viewer can review it in a single format. It should then be able to collect the data and conclusions into standard sets of reports that the user can use for final decision making and planning.

## 3. THE NED-2 ARCHITECTURE

NED-2 integrates a sophisticated user interface, databases, simulations, knowledge bases, hypertext documents, geographical information systems, and visualization tools into a single decision support system. We wanted an open system that would allow us to incorporate additional simulations, knowledge bases, and other decision support tools easily. This would not be possible if integration of each component required extensive procedural programming. Instead, we decided to build intelligent agents each of which knew how to use a class of decision support tools. These agents are developed in Prolog, a high-level logic programming language. As an example, the NED-2 simulation agent knows that a growth and yield simulation requires input in a given format, requires control codes to simulate treatments and set stop conditions, and writes output in a specific format.

The central organizing principle for NED-2 is the blackboard [Ni 1989] (Figure 1). Unlike object oriented or mediator architectures, agents do not directly invoke each other in a blackboard system. Instead, tasks that need to be done are posted to a blackboard. Agents also post results of their activities to the blackboard where they are accessible by all other agents. Agents watch the blackboard continually. A particular set of facts and/or tasks listed on the blackboard can prompt an agent to do some work. If an agent completes a task on the blackboard, it erases that task from the “to do” list. An agent may also begin a task, then discover that something needs to be done that is beyond its capability before it can complete the task. It can put the new task on the blackboard and wait until another agent

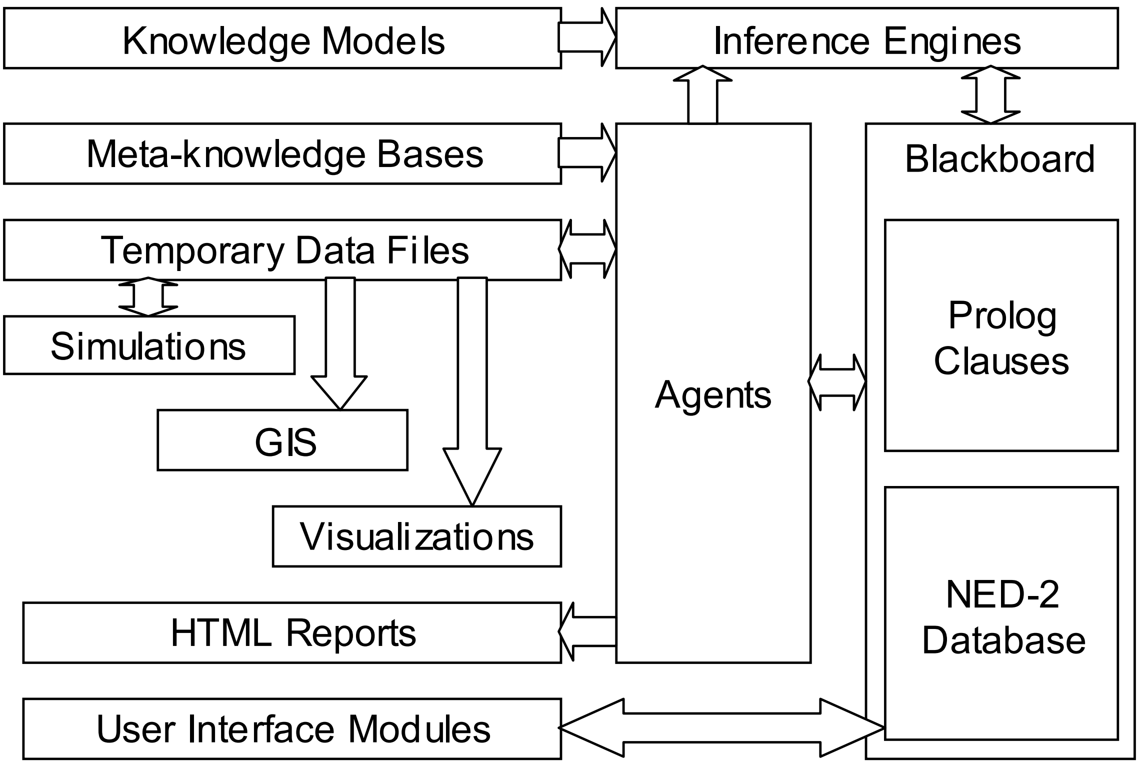


**Figure 1: The NED-2 Architecture**

performs it before completing the original task.

Blackboard architectures have certain advantages over other agent architectures. In object oriented or mediator architectures, an agent (or object) that needs help in performing some subtask must address the request for help to a specific agent that can perform that subtask. It must also provide that assistant with the specific information it will need to perform the subtask. So each agent must know what other agents can do and what they need to do it. With a blackboard architecture, an agent simply posts a request on the blackboard. The request is visible to every other agent and all information the assisting agent needs is also on the blackboard. This reduces the knowledge requirements for agents.

Blackboard architectures are more flexible than rigid hierarchical architectures. Organizing agents into hierarchical trees prevent agents from accepting assignments from any agents except their immediate supervisors. In object oriented programming, an object can respond to the request of any other object that knows how to address it. Using a blackboard architecture, an agent can respond to any agent that can write a request on the blackboard.

The NED-2 blackboard has three components: a set of Prolog clauses, a Microsoft Access database, and a set of Prolog routines. The simplest part of the NED-2 blackboard is represented by the two Prolog predicates **`fact/4`** and **`request/1`**.

Facts are stored as AOV (attribute-object-value) triples. One of these is the first argument in a clause for the predicate **`fact/4`**. The other three arguments are the source for the fact, a fact characteristic index (usually the atom **`true`** but might in some cases be a confidence factor, fuzzy number, or some other value,) and a time stamp. A request can be the name of a task to perform or a list of tasks to be performed. AOV triples and tasks may be fully described given the expressive power provided by Prolog. For example,

```
fact(trees_per_acre([stand(17),
species(oak),    size_class(9)],
22),true,user,2345235).
```

Here the object is abstract: it is the oak trees of 9” diameter on stand 17. The attribute is the number of those trees per acre, and the value is 22. The other three values can be used to represent a degree of confidence in the answer (simply “true” in this case since the information is provided by the user,) the source of the information (here, the user,) and an internal time stamp indicating when the fact was put on the blackboard.

Requests can have a simple form like

```
request([analysis]).
```

Requests can also contain complex plans; these will be discussed below in the section on NED-2 planning agents.

NED-2 uses a Microsoft Access database to store inventory and other information. This database has been integrated into the NED-2 blackboard. When a NED-2 agent wants to know something, it calls the Prolog predicate **`known/1`**. The argument provided to **`known/1`** is an AOV triple where the attribute and the object are specified but the value is an uninstantiated variable. **`known/1`** matches this incomplete AOV triple against triples contained in clauses for **`fact/4`**. If it doesn’t find a corresponding fact there, it converts the triple into an SQL query and seeks the information in the NED-2 database.

Agents place facts on the blackboard as Prolog clauses by asserting them. Agents formulate SQL queries to place data in the database.

## 4. NED-2 AGENTS

Each agent in NED-2 possesses both the procedural and the declarative knowledge to perform a particular step in the NED decision process. The procedural knowledge for a task is captured in procedural codes written in C++ or Prolog or both. Often these are procedures for executing a piece of software developed by a third party such as a simulation, a geographical information system, or a knowledge based system. In some cases, the agent knows how to operate a class of external models, for example, growth and yield simulators. Each member of this class will have its own requirements and knowledge about these requirements is provided in a declarative meta-knowledge file. Essentially, a meta-knowledge file provides semantics for the external simulator or other model by interpreting its data model in terms of the internal data model of NED-2.

The list of functions currently supported by NED-2 include inventory, inventory analysis, goal selection, treatment definition, baseline generation, plan development, plan simulation, goal analysis, GIS display, and report generation. Each of these functions requires the actions of one or more agents for completion. The current list of NED-2 agents, all written in Prolog, include the following.

Interface agent
Treatment Definition agent
Simulation Agent
Goal Analysis Planning agent
Timber Goal Analysis agent
Wildlife Goal Analysis agent
Water Goal Analysis agent
Visual Goal Analysis agent
Ecology Goal Analysis agent
GIS agent
Report Generation agents

We provide additional details about how these agents are organized as we describe individual agents below.

## 4.1 The NED-2 Interface Agent

The NED-2 user interface agent consists of control codes written in Prolog and a large set of C++ and Prolog modules. The Prolog control code reads the blackboard and determines when a particular user interface function is required. It then calls the appropriate C++ or Prolog interface module. A few of the interface modules are called directly by one of the other NED-2 agents. The NED-2 interface modules currently include the following.

User Preferences module
Inventory module
Goal Selection module
Treatment Definition module
Baseline Development module
Plan Development module
Report Selection module
GIS module

Each of these modules provides the procedural knowledge to perform a particular interface function. The User Preferences module allows the user to choose the units of measurement, formulas for computing volumes, and other preferences to be used in the system. The Inventory Module is a spreadsheet for entering data about trees in the overstory, understory data, and physical characteristics of stands and plots. This spreadsheet automatically calculates basal areas, volumes, and other important values. Although referred to as the Inventory module, this module can also display data generated through simulation of treatment plans. The Goal Selection module allows the user to select management goals for timber, wildlife, water, visual quality, and ecology from a pre-defined list. Before a goal can be included in a meaningful way, rules must be added to a knowledge based system that define the satisfaction conditions for the goal in terms of variables that can be measured or observed on the land.

The next three modules allow the user to develop alternative treatment plans for his land. The Treatment Definition module displays available treatments that can be included in plans including default parameters. Through the Treatment Definition dialog, the user can either accept the default parameters for thinnings, shelter cuts, and other treatments or modify the parameters to suit his preferences. The Baseline module lets the user select a year to be used as the base year in any alternative treatment plans he constructs. There must be inventory data on each stand at or prior to the baseline year. If he has defined some treatments, the user can also include these in the Baseline dialog to show historical treatments that have been implemented on a stand since the most recent inventory. The Planning module presents a time line to the user for each stand in the management unit. The user can attach any treatment he has defined to the timeline for any stand using the Planning dialog.

The final two modules let the user decide how to view the data and analyses provided by the system. The Report Selection module provides the user with a list of available stand level and management unit level reports. Here the user can customize the set of reports he wants to generate. At the time reports are actually generated, a second dialog lets the user specify the treatment plan and the year that the reports should be based on. The GIS module lets the user merge data and results of goal analysis with a shape file to view this information in ArcView. The dialog lets the user choose the variables and goals that can be used to color the stands in a map of his management unit. He can include layers for inventory and baseline. He can also include separate layers for different years of a single treatment plan, or separate layers for different treatment plans at a single year.

## 4.2 The NED-2 Treatment Definition Agent

Many growth and yield simulators will also simulate various silvicultural treatments. The Forest Vegetative Simulator (FVS) incorporates a suite of growth and yield simulators developed by different researchers. This suite of simulators is combined with a uniform method for simulating silvicultural treatments and a single interface called SUPPOSE for setting up a simulation run [Crookston 1997]. In the current version of NED-2, we have incorporated the northeastern and southern variants of FVS. However, NED-2 is designed to make it easy to incorporate other simulators as they become available.

The Treatment Definition agent responds to a request placed on the blackboard by the Interface agent. This agent helps the user set up the parameters for different silvicultural treatments that a given growth and yield simulator supports. A Prolog knowledge base provides an initial list of available simulators, specifies which treatments are supported by each simulator, and establishes default parameters for each of these treatments. Called the default treatment knowledge base, this file also includes meta-knowledge about the control information required for each treatment by each simulator. This control information is used by the Simulation agent described below.

The Treatment Definition agent provides a dialog for the user to review default treatment parameters for any available simulator. The user must accept or modify the default values before the treatment becomes available for him to use in developing a baseline or a treatment plan. Once the user has selected all the treatments he will use, these are stored in a separate Prolog knowledge base. He can recall and revise this personal treatment knowledge base later and reuse it for other management units.

The Treatment Definition agent also creates a table in the data file for the current NED-2 project. This table provides information to the Baseline and Planning modules about the available simulators and the treatments that have been defined for them.

This agent simplifies the task of setting up parameters for treatments to be included in treatment plans. A single interface is used to define treatments regardless of which simulator the user selects. So the user does not have to learn different interfaces. Nor does the user need to know anything about the specific control codes used by a simulator. And the user only has to set up his treatment parameters once. After they have been stored in his personal treatment definition knowledge base, they are available for reuse with that or any other management project in the future.

### 4.3 The NED-2 Simulation Agent

If the user directly requests that a plan be simulated, then the Interface agent places a request for that plan to be simulated on the blackboard. The Goal Analysis agent may also place a request for one or more plans to be simulated on the blackboard if the user has requested a report or a GIS display involving a plan that has not yet been simulated. The Simulation agent responds to these requests.

The Simulation agent is responsible for constructing the baseline and simulating the management plans that the user constructs. Both inventory, i.e., field measured, and simulated data are stored as "snapshots" in the NED-2 database. Each simulated snapshot represents data for a stand under a treatment regime for a single year. Snapshots include information about the overstory, understory, and ground layers. Growth and yield models generate overstory information; knowledge-based systems generate understory and ground layer information. The Interface agent uses the Inventory module to display snapshots and side-by-side comparisons of data in different snapshots.

Meta-knowledge is knowledge about knowledge. Agents that operate external models need meta-knowledge to know when and how to use a knowledge source or a knowledge tool. For a particular simulation program, we develop a meta-knowledge base that provides the simulation agent with the tables needed to translate data between the NED-2 format and the format of the simulation. This meta-knowledge base also tells the simulation agent what control codes the simulation understands. The users' personal treatment definition knowledge base provides information about the parameters to be used for particular treatments. So all procedural knowledge for running simulations is written into the Simulation agent and specific knowledge needed to run individual simulations is stored in these declarative meta-knowledge bases.

The first growth and yield models integrated with NED-2 are the Northeastern and Southern variants of FVS [Crookston 1997]. The Simulation agent sets up and calls these growth and yield models as needed. The user selects the simulator he wants to use for each stand or NED-2 will use meta-knowledge to recommend a simulator based on information about the management unit. Minimally, NED-2 evaluates the user's choice of simulator and notifies the user if it detects a problem. For example, NED-2 will alert the user that the Northeastern variant of FVS may not be appropriate for use in Georgia.

Rule-based knowledge models will be used to predict changes in the understory and ground layer of stands over time. Causal models are not available for these layers at present, but it is essential to provide usable data for these layers as input to wildlife and other models. Members of the NED Core team are developing rules that will predict what these layers look like based on historical data and the properties of the overstory layer.

The advantages of an intelligent agent that can run a variety of simulation programs for a user should be obvious. The user builds a single set of treatment definitions that are stored in his own knowledge base. Then, working with a single intuitive dialog, he can set up a schedule of treatments for every stand in his management unit. While only two regional variants of FVS are currently available through NED-2, we anticipate adding simulators for pine plantations or other special situations. This will allow the user to select different simulators for different stands and different purposes. But the user will not need to know anything about how the data must be prepared for these simulators or about the codes these simulators require to control growth and simulation of treatments. All of this is transparent to the user. Nor will he have to set up separate simulation runs for each stand and each alternative treatment plan. The Simulation agent will accomplish all of this for him when it simulates his treatment plans.

### 4.4 NED-2 Goal Analysis Agents

NED-2 knowledge based models are rule sets written in Prolog and used in conjunction with one of the NED-2 inference engines. Standard forward and backward chaining inference engines and a fuzzy backward chaining inference engine are used in NED-2. Other inference engines can be added easily.

Goal analysis agents use fuzzy rule sets to evaluate how well a stand or management unit satisfies management goals. These rules analyze a goal into desirable future conditions. Four fuzzy categories indicate how well a goal is met: fails, nearly passes, barely passes, and passes. NED-2 agents can perform goal satisfaction analysis on any set of snapshots representing the management unit at a point in time under a management plan.

NED-2 includes a knowledge model for wildlife habitat requirements based on research for the northeastern United States. Alternative wildlife models are being developed for the Great Lakes region and for the southeastern United States. Goal analysis agents will use a meta-knowledge base to select the appropriate wildlife model for a user.

Other knowledge models ranging in size from quite small to medium sized are used in NED-2 in many places. For example, knowledge models are used to evaluate the biological diversity and the "patchiness" of management units. Different agents in the system know when and how to employ these knowledge models to complete tasks requested by other agents or by the user.

When the user requests either a GIS display or a set of reports, NED-2 must determine which plan(s) and year(s) should be included, what simulations need to be run, and what goal analyses need to be performed. The Goal Analysis Planning agent responds to any request for GIS display or reports and builds a plan to generate the information needed for the display or reports. First, this agent looks at the blackboard to see which reports and goals the user has selected, which treatment plans the user has constructed, and which of these plans has been simulated. If there are no plans or if other prerequisite tasks have not been performed by the user, the agent provides the user with instructions for performing the tasks that must be performed before a GIS display or reports can be generated.

If the preliminaries have all been completed, the Goal Analysis Planning agent invokes the Goal Analysis Planning interface module where the user can select a plan and year to be used for reports, or can select a plan *or* a year for a GIS display. (More details on the GIS display are explained below.) The agent then determines which simulations need to be run, and which goals need to be analyzed for which plans and which years. It then puts a complex request on the blackboard that contains a plan for running the necessary simulations and performing the appropriate goal analysis. If a GIS display has been requested, then this request is added to the end of the plan. If reports have been requested, then the Goal Analysis Planning agent determines which reports should be written and puts requests for these reports at the end of the plan.

Without the Goal Analysis Planning agent, the GIS agent, and the Report Generation agents described below, the user would have to set up each simulation, run goal analysis on each plan for each year where it is required, and then either manually convert data to the format needed to be merged with a shape file or incorporate the data into reports that he composes himself.

### 4.5 The NED-2 GIS Agent

The output from NED-2 analyses can be displayed using a geographical information system. The GIS agent invokes the GIS interface module. In the dialog displayed by this module, the user can decide whether to view a single plan with different layers representing different years, or to view a single year with different layers representing alternative treatment plans for that year. In this dialog, the user also selects the data variables and goals he wants to use to color a map of the stands in the management unit. Then the GIS agent creates a temporary data file containing the information the user wants to view. The agent calls a Microsoft Visual Basic wrapper that starts the GIS (ArcView) and tells it to join the temporary data with a shape file the user provides. In the GIS, the user selects stand level variables (forest type, size class, satisfaction of some stand-level goal, etc.) to display on the map.

### 4.6 The NED-2 Report Generation Agents

NED-2 generates a variety of reports organized into a multi-page hypertext document. Each Goal Analysis agent generates an HTML document summarizing the results of its analysis and puts a fact on the blackboard indicating that the analysis has been performed and recording the name of the summary HTML file. Other specialized Report Generation agents respond to requests on the blackboard and write HTML files that contain both stand level and management unit level reports on vegetation, physical features, patch analysis, and other information useful in making management decisions. Each of these agents also records its actions on the blackboard. Finally, an agent responsible for writing the top level file for the hypertext document collects information from the blackboard on the files that have been generated for the report and constructs an HTML page with all the links to the other reports to be included in the document. This agent then executes the user's default Web browser and loads the top level page of the report into the browser.

The user can save a set of reports to a new folder, making it possible to refer to the set without having to generate them again in NED-2. The user can cut and paste text and tables from these reports into any word processor together with images from the GIS displays. This greatly simplifies the task of creating a custom report that includes the analysis and final management plan.

## 5. CONCLUSIONS

The intended users of NED include anyone who is interested in management of forest land, but principally those responsible for the individual management decisions on specific units of land. Case studies using the NED decision process on management units ranging from 85 to 15,000 acres have been initiated and some of these have been completed. In these studies, NED-1 is used and the functions of NED-2 not available in NED-1 are performed by an experienced NED-1 user. The approach in NED-2 is to provide independent agents using a blackboard architecture that can perform the integrating tasks of the NED-1 user in these case studies. This approach allows great flexibility while keeping the development effort efficient. We do not ask NED-2 to recommend treatment plans to the user; instead, NED-2 provides the user with access to a wide range of tools for generating alternative treatment plans, simulating these plans, evaluating how well these plans achieve management goals, and displaying the results of this analysis in a variety of formats. The integrated software system helps users to gain a general understanding of their situation while using data collected from their forests to help analyze specific questions. The key is that through this approach we are able to help people consider multiple benefits and the tradeoffs among them.

NED-2 ver. 0.2 is now complete. It includes all user interface components described here. This version can simulate management plans using the northeastern and southern variants of FVS, and it can perform goal analysis on simulated data for future years. Hypertext reports have been enhanced. At the time of this writing, we expect to release a beta version of NED-2 before the end of 2002. This version will not include Envision or other visualization tools, but plans are in place to incorporate these tools in a future version of NED.

The evaluation of NED-2 will use the case study method. Rauscher [2001] has completed an initial study using NED-1 for inventory analysis and for goal analysis. In this study, simulation and visualization were performed independently of the NED-1 system. Rauscher is engaged in additional case studies for management units ranging from a few hundred to a few thousand acres. He is currently moving the software base for these studies from NED-1 to NED-2. Results will appear in future publications.

## 6. ACKNOWLEDGEMENTS

More than 100 people have contributed to the development of NED software through participation on committees, testing of preliminary versions, or providing financial or moral support. Suffice it to say that without their hard work and dedication, NED and NED-2 would not have been possible.

## 7. REFERENCES

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

McGaughey, R. 1997. Visualizing forest stand dynamics using the stand visualization system. *Proceedings*, of the 1997 ACSM/ASPRS Ann. Conv. and Exp., April 1997, Seattle, WA, vol. 4, 248-257, Bethesda, MD; Am. Soc. For Photogrammetry and Remote Sensing.

McGaughey, R. 2000. ENVISION – environmental visualization system, http://forsys.cfr.washington.edu/svs.html.

Ni, H. Penny. 1989. Blackboard systems. In Avron Barr, Paul R. Cohen, and Edward A. Feigenbaum (eds.), *Handbook of Artificial Intelligence, Vol. IV*. Reading, MA: Addison-Wesley.

Nute, D., Rauscher, H. M., Perala, D.A., Zhu, G., Chang, Y., and Host, G. M. A toolkit approach to developing forest management advisory systems in prolog. *AI Applications* **9** (3) (1995):39-58.

Potter, W. D., Liu, S., X. Deng, Rauscher, H. M. 2000. Using DCOM to support interoperability in forest ecosystem management decision support systems. *Computers and Electronics in Agriculture* 27: 335-354.

Rauscher, H. Michael, Lloyd, F. Thomas, Loftis, David L., and Twery, Mark J. 2000. A practical decision-analysis process for forest ecosystem management. *Computers and Electronics in Agriculture* 27: 195-226.

Rauscher, H. Michael. 2001. The NED decision process for sustainable forest management. The Deer Hill Tree Farm Case Study. Final Report. Asheville, NC: U.S.D.A., Forest Service, Bent Creek Experiment Station.

Smith, S., and Mackaness, W. 1996. The role of intelligent agents in integrated environmental management. *1st International Conference on GeoComputation University of Leeds UnitedKingdom,* 17 - 19 September 1996.

Twery, Mark J., Rauscher, H. Michael, Bennett, Deborah J., Thomasma, Scott A., Stout, Susan L., Palmer, James F., Hoffman, Robin E., DeCalesta, David S., Gustafson, Eric, Cleveland, Helene, Grove, J. Morgan, Nute, Donald, Kim, Geneho, and Kollasch, R. Peter. 2000. NED-1: integrated analyses of forest stewardship decisions. *Computers and Electronics in Agriculture* 27: 167-193.