SHULEI SUN
Wildlife Habitat Management in NED-2
(Under the direction of DONALD NUTE)

This thesis describes the development of a knowledge-based system to evaluate forested land for availability of wildlife habitat. NED is an acronym derived from "Northeast Decision Model". The NED DSS, a collection of software products, was created to help resource managers to build up goals, access current and potential conditions, and create long-term, landscape-level management plans for forest properties in the eastern United States. An initial version of NED had been released, which achieved only part of the functions we hope eventually to capture in NED. Throughout this thesis, I will focus on the wildlife component of NED and its successors. I will describe the weakness of first version in four aspects: 1) intelligent module manager, 2) data management, 3) reporting utility, and 4) knowledge representation. Then, I will discuss how we implement solutions to these problems.

INDEX WORDS: NED, Wildlife habitat, Desire future conditions (DFCs), Knowledge-based representation, Prolog

WILDLIFE HABITAT MANAGEMENT IN NED-2

by

SHULEI SUN

BMED, Taishai Medicine University, P.R. China, 1993

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

2001

© 2001

Shulei Sun

All Rights Reserved

WILDLIFE HABITAT MANAGEMENT IN NED-2

by

SHULEI SUN

Approved:

Major Professor: Donald Nute

Committee: Charles Cross
Eileen Kraemer

Electronic Version Approved:

Gordhan L. Patel
Dean of the Graduate School
The University of Georgia
December 2001

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to Dr. Donald Nute, my major professor, for his constant support and encouragement and invaluable and advice throughout this challenging thesis project. I also wish to thank my two other committee members, Dr. Charles Cross and Dr. Eileen T. Kraemer, for their support and advice.

I thank my family for their love and concerns, especially my husband, Yang, for his continuous support and encouragement.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS ……iv

CHAPTER

1 INTRODUCTION……...……....1

1.1 FUNCTIONAL SPECIFICATION FOR NED ……..……..…4

1.2 PREVIOUS WORK ON NED ..……...……..5

1.3 NED-1 ARCHITECTURE AND FUNCTIONALITY ……..…...……..7

2 THE WEAKNESS OF NED-1……..12

2.1 INTELLIGENT MODULE MANAGER ……..……12

2.2 DATA MANAGEMENT ……...……...15

2.3 REPORTING UTILITY ……...……16

2.4 KNOWLEDGE REPRESENTATION ……17

3 IMPLEMENTATION ……..……20

3.1 INTELLIGENT MODULE MANAGER ……..…...20

3.2 CONVERTING THE WILDLIFE RULES TO DSSTOOLS FORMAT…….22

3.3 REPORT GENERATION ……..…….24

3.4 DATABASE MANAGEMENT ……...……...25

4 SUMMARY OF RESULTS ……27

BIBLIOGRAPHY ……..……29

CHAPTER 1

INTRODUCTION

This thesis addresses the problem of development of a knowledge-based system to evaluate forested land for availability of wildlife habitat. This evaluation has taken place within the broader context of multiple-use forest management, and the knowledge-based system I developed is incorporated into a larger decision support system for forest management.

Forest ecosystems are complex hierarchies (Twery et al.2000). To obtain an ideal management result, multiple aspects of the system need to be monitored and worked on. It is key for the best use of the forest. Focusing on only one species or area of problem can cause very bad outcomes. Since there are so many factors to be considered, environmental management is difficult to perform. To make decisions on environmental subjects, we need to be aware of several important issues, which include influences of the human activity on the environment, concerns about human health, economic costs, the relevant laws or regulations, fairness, and many others. Generally speaking, all of the interrelated factors stated above have impacts on any decisions made regarding the environmental management. For example, how can we manage timber yield if it threatens an already endangered species? How can we increase the productivity of land without doing damage to the environment?

The answer often lies in the use of decision support systems (DSS), which are the computer programs helping managers make decisions in situations where human

judgment is important but where limitations on the ability of judgment impede decision-making (Turban, 1993). A DSS analyzes data and presents it so that the user can make decisions more easily. The aim of a DSS is to provide foresters and ecologists with the knowledge, skills, tools, and consulting services needed in the whole process of their decision-making. A DSS provides decision makers with a set of opportunities directed toward improving the effectiveness and productivity of managers and professionals, rationalizing the decision making process within an organizational context, and providing a powerful aid to better allocate scarce resources for land management. DSSs have been in use, in one form or another, for nearly 20 years, but with the power increase and sophistication of computers, DSSs have become a vital component of land management. As natural resource management moves from a compartmentalized approach to multiple-use management and then to a more complex ecological approach wherein the interaction among components must be known and considered, the need for more powerful DSSs is obvious (Rauscher, 1999). A recently released compendium of DSSs for ecosystem management describes 24 systems (Mowrer, 1997). The Northeast Decision Model (NED) is one of them.

The NED DSS, a collection of software products, was created to help resource managers to build up goals, assess current and potential conditions, and create long-term, landscape-level management plans for forest properties in the eastern United States. The USDA Forest Service's Northeastern Research Station is leading the development effort in cooperation with collaborators from other units in the USDA Forest Service, in universities, and in the private sector (Nute et al. 2000).

The NED system converts common goals into specific and well-matched goals, and with these goals the NED system is being developed to put forward specific management proposal for units of land. Anyone who is interested in management of forested land in the Northeast, particularly those who assume the responsibility of individual management decisions on specific units of land, can be a client of the NED system. NED is designed for two major user groups. The first group is consulting foresters, and the second group is public forest resource managers such as district managers of national forests.

Among the practices of forest management, tree cutting is the most traditional and direct technique for foresters, but foresters also use planting, burning, and other activities to meet their goals. As a comprehensive system, NED provides a large amount of information to the end user. The information specifies, the possible management goals for a specific subject, the conditions necessary to meet those goals, and possible management techniques that can improve the conditions.

A major threat to the maintenance of wildlife diversity is the failure to recognize the impact of management decisions on habitat conditions of various habitat types (Cleveland *et al.* 1998). Compounding this problem is the fact that most forest managers are not trained in evaluating forests for wildlife habitats. Each wildlife species has a specific habitat; the habitat provides the requirements for survival and breeding. Based on food, cover, and reproductive requirements, several habitat suitability models have been developed for many individual species. The habitat suitability models of the U.S. Fish and Wildlife Service (Clark and Hutchinson 1989) is such a model. However, the

majority of these habitat models focus on major game species. One model for New England wildlife uses matrices instead of suitability indices to relate information on forest type and vegetative and physical structural features to over 300 bird, mammal, amphibian, and reptile species in New England (Cleveland *et al.* 1998). However, to find out the wildlife habitats in an individual stand by reading all the matrices and indices is very time-consuming. Faster tools are required.

The NED system includes such a tool. It provides users with either a list of habitats used by an individual species or a list of species derived from habitats. My research focuses on the wildlife habitat aspect of forest management. A rule-based intelligent system is used to define wildlife goals and to evaluate forested land to see if habitat is available for wildlife species included in the system. This system is based on the New England Wildlife model.

1.1. FUNCTIONAL SPECIFICATION FOR NED

In NED, wildlife management focuses on increasing species richness. Part of the reason is that some people want to satisfy their hobbies, such as bird watching or animal hunting. Wildlife has great ecological, social, and economic importance as well. Bacteria, plants, algae, fungi, and invertebrates are involved in essential ecological processes such as oxygen generation, nitrogen fixation, nutrient cycling, waste decomposition, water cleansing, and soil formation. Many wildlife species are harvested and processed to produce a variety of commercial products, ranging from food and

fashions to medicines, paper, and construction materials. Recreational activities including fishing, hunting, and wildlife study and photography provide much enjoyment and substantial economic benefit.

Every species of wildlife exists within a specific set of environmental conditions: climate, food availability, shelter, etc. This set constitutes the species' "home" or habitat. Without the particular conditions that a species' habitat affords, it cannot exist. These conditions may be more or less broad. The habitat of humans and cockroaches, for example, covers an extremely wide range of ecological regions, due to their great adaptability.

NED gives the user tools for wildlife management, particularly habitat management (Twery *et al.* 2000). When using the system, the users specify a desired increase in the richness of the wildlife species present in the forest among their goals, then they select certain properties based on the goals. They can also select whether to maintain, enhance, or create habitat for certain species.

1.2. PREVIOUS WORK ON NED

Development of the NED system was begun in 1987 by a group of researchers in the Northeastern Forest Experiment Station (Rauscher *et al.* 1997). They worked on creating a program to promote new ways of sharing ideas and organizing information. One of the ideas put forward was to develop a computer program that would combine all the previously independently produced growth and yield models developed by scientists

within the Station (Marquis 1990). A main purpose of the project was to develop an easy-to-use program to provide the user with summary information and expert prescriptions for any forest type in the northeastern United States.

To develop this comprehensive system, a considerable amount of time and resources were required. Several expert panels were formed to work on collecting information for this complicated resource system. Each group consisted of research scientists and practitioners; the coordinators of the NED project attended all meetings. Within each committee, the definitions of potential desirable goals for that particular resource, plus a description of the conditions necessary to achieve each goal, were developed through a series of meetings and correspondences (Rauscher *et al.* 1995).

In order to get useful feedback from the potential users, which will help with the design of the program, NED's developers released several independent software programs, which represent different stages of development of the program. The initial freestanding programs such as NED/SIPS (Simpson *et al.* 1995), NEWlLD (Thomasma *et al.* 1998), and the Forest Stewardship Planning Guide (Alban *et al*. 1995) have had a large body of users, generated considerable comment, and had great influence on the design of NED.

An analytic version of NED, NED-1, was released in March of 1999. NED-l is being used in its beta-testing form at several colleges and universities to replace the DOS-based NED/SIPS in their classes (Twery *et al.* 2000). NED-1 implements the proposed decision method through the goal satisfaction analysis stage (Nute *et al.* 2000) and makes summary reports and evaluates conditions for forest property using the user-supplied forest inventories.

1.3 NED-1 ARCHITECTURE AND FUNCTIONALITY

The major components of the NED-1 architecture are shown below. The main program of NED-1, which consists of the user interface and data management modules, is written in C++ utilizing the application framework C++/Views. The user accesses NED-1 functional elements through the user interface. The inference server is written in both C++ and Prolog, and the knowledge representation is written in Prolog.

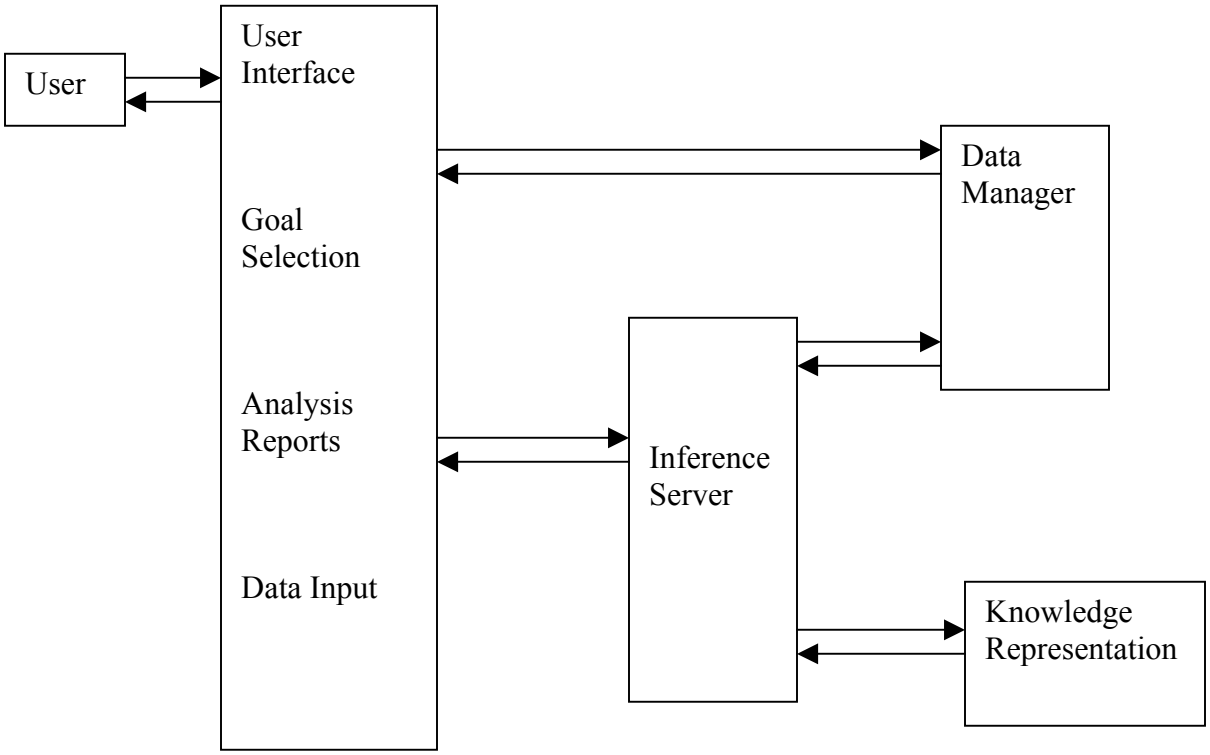

Figure 1.1 The major components of the NED-1 architecture

NED-1 is a Windows-based program that is used to analyze the forest-inventory data collected about various forest resources. The categories it covers include visual quality, ecology, forest health, timber, water, and wildlife. The primary function of NED-l is to evaluate the degree to which individual stands or the management unit as a whole provides the conditions required to accomplish specific goals (Twery *et al*. 2000). NED-1 utilizes an extensive hypertext system to offer the user a large amount of information on managing a forest property. This includes information about resource goals, the conditions that will be needed to achieve those goals, and the data used to evaluate the real time conditions of the forest. NED-l is designed to access all sorts of information through a single interface. The information is related to the multiple resources and multiple goals defined based upon the Forest Stewardship Planning Guide, the analysis of wildlife habitat, the timber inventory summary, and economic analysis supported by NED/SIPS. NED-1 is helpful in multiple-stand management service. It can provide analysis for the multiple-stand as a whole as well as separately for an individual stand, and the user can evaluate conditions across the entire property.

NED-1 uses a proprietary database management system. The data manager of NED-1 consists of data dictionaries and data objects. It functions as a central data storage unit that receives data through user input. It allows the user to select goals for analysis and reports for generation, and then passes the data to its analysis and report-generation modules. The data dictionary stores the data in the form of static metadata that contains the properties of each variable in the database. The function of each data dictionary is to

create one or several DataObjects. Each DataObject contains a list of the values of the variables stored as well as flags indicating the sources of each value.

The data dictionary designed for NED-1 allows easy definition of variables by using a dictionary entry. As mentioned before, it uses a proprietary format developed in C++. getDisplayString, getDatasource, and getDataValue are three major functions in the data manager. In the NED-1 database, any data element can be accessed using one or more of those functions. They all have the same seven arguments; so, the programmer can access the hierarchical data structure to retrieve any piece of data. Since one function cannot presume that data are present in every data element, two steps are used to retrieve data. The first step is to find out whether data are present and the second step is to retrieve the real value.

This discussion of the way data is handled in NED-1 is important because NED-2 will take a different approach to data management and this will affect the way we implement wildlife habitat analysis in NED-2.

In NED-1, the goal-selection function lets the user select wildlife species goals from a Goal List displayed in the user interface dialog box (see Figure 1.2). Selecting goals starts the analysis. From the selected goals, NED-1 generates a set of Desire Future Conditions (DFCs), and evaluates how well current forest conditions satisfy these DFCs. A goal is a desirable condition, a situation that someone is willing to allocate resources (time, effort, money, etc..) to achieve. ‘Desired Future Condition’, or ‘DFC’, is a technical term to signify goals that are not analyzed in terms of other goals in the goal

structure (Nute et al. 2000). A DFC is a special kind of goal, and is a specific criterion to be used for analyzing higher-level goals.

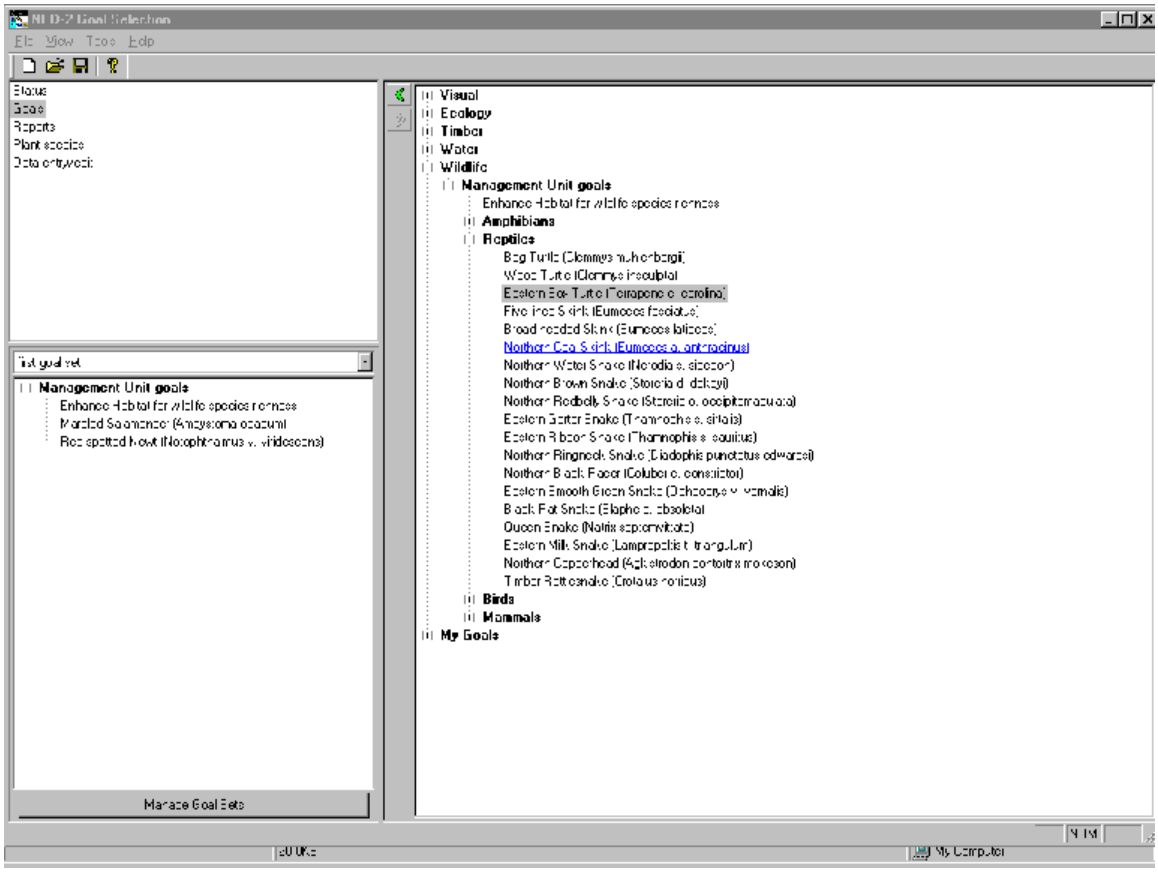

Figure 1.2 User interface dialog box

NED-1 can make different types of reports after analyzing the current conditions in the data management module or comparing those conditions to some DFCs. There are several general tabular report formats for which the user must choose the variable that appears in the table, which potentially produce a diverse set of reports. Two of the most important reports are the Stand Goal Analyses and Management Unit Goal Analyses. A

dialog box allows the user to select which reports to generate for which stands. A Reports List view holds the selected list and allows the user to view the reports individually or to print one or all of the selected reports. All NED-1 reports are created from specifications that are written in C++. These specifications are stored in a data file external to NED-1.

Although NED-1 is a very useful DSS for forest management, it still has some weaknesses in its design. In the next chapter, I will discuss some of the problems with the system. In chapter 3, I will discuss how we implement solutions to these problems. In the last chapter, I will summarize my results. Throughout this thesis, I will focus on the wildlife component of NED-1 and its successors.

CHAPTER 2

THE WEAKNESS OF NED-1

During the years of the implementation and use of NED-1, people realized that NED-1 still left much to be desired. Since 1999, scientists in artificial intelligence and related field have been working on upgrading the old NED-1 system to make it more like a truly intelligent model management system, and so far a lot of practical problems have been spotted, which include database connection, module control, reporting utility, and the knowledge base format. These problems are detailed below. In the next chapter, I will describe how changes in NED-2 address these problems.

2.1. INTELLIGENT MODULE MANAGER

A DSSTools application consists primarily of one or more knowledge bases and one or more domain control modules (DCMs). The DCMs are semi-autonomous agents; they can call any of the inference engines available in DSSTools, and have permission to read the blackboard and to put information and requests on the blackboard. The blackboard itself is a global database in memory where the system stores information provided by the user, conclusions inferred during a consultation, and requests for particular modules to perform some task. DCMs are independent of each other. Only one DCM is active at a given time. A DCM examines the data on the blackboard, performs

the subtask, and writes the results onto the blackboard where they become available to every DCM. After a DCM terminates, each DCM has a chance to check the blackboard to see if it should perform its task. This makes the design and maintenance of the system much easier.

NED-1 has three kinds of DCMs. A set of facts and rules, which provides the knowledge to do some essential function in NED-1, is paired with most of these DCMs. Each DCM, plus its associated knowledge base and the inference engine that it uses, constitutes a small knowledge based system in NED-1.

NED-1 has two main components, a main C++ program that provides system control and a front-end that communicates with the user and an Inference Server written in Prolog and using DSSTools. These two components communicate with each other through a Dynamic Linked Library (dll). The Logic Core DCM in the Inference Server receives requests from the NED-1 C++ front-end program. For example, the Inference Server could be requested to evaluate the DFCs for the wildlife goals that have been selected or to provide a list of wildlife species for which suitable habitat can be found on the management unit. Although the NED-1 front-end program has all the data about the management unit, it does not know what data will be required to complete any of these requests. At the same time, the DSSTools component of NED-1 has no direct access to the forest data; but it knows how to solve different problems. It requests information from the NED-1 front-end program as it is needed. The NED-1 front-end is informed when the inference is complete. It can then call the Inference Server Interface to read the results from the blackboard.

NED-l requires the user to provide detailed information about the management unit including tree inventories for representative plots for each stand in the forest. The NED-1 front-end program had already been developed in C++ before the developers made a decision about how to provide the inference capabilities that the system would need. Then the client-host structure described here was added to NED-1. A big problem in the NED-1 architecture is that the procedural front-end program handles model management. The interface and simulation components are separated from the knowledge based systems in NED-1 and Prolog would have to go through C++ to call a simulation when we extend the capabilities of the system in the next phase of NED development. Since the communication link available with Prolog running in the background is very limited, this approach creates an awkward bottleneck. NED-2 is aimed at adding a crucial function of truly intelligent model management to the DSSTools part of the system. Prolog provides the top-level control in NED-2. Data entry and other dialogs included in the NED-1 front-end program are divided into different procedural modules to assist modular development. In NED-2, these modules are called by Prolog.

In NED-2, the intelligent module manager (IMM) is a mechanism controlling the modules of NED. The blackboard of the IMM stores knowledge about the intermediate data and user requests received from the interface written in C++ (Plug-and-Play or PnP modules). The IMM registers PnP modules, intelligently controls execution of PnP modules, monitors and communicates with various user interfaces, and provides data and control data access.

2.2. DATA MANAGEMENT

As mentioned before, NED-1 uses a proprietary database. The data manipulation routines are written in C++, and the goal analysis is written in Prolog. Data retrieval by Prolog is difficult because Prolog can only access the data through the C++ program. In NED-2, a commercial DBMS is used to manage the data. Prolog can access this data directly. Using a commercial database has the additional advantage that the NED developers do not have to maintain the database management system.

For database manipulation, the DAODLL (Data Access Object Dynamic Linked Library) was developed by Geneho Kim using C++. The DAO provides a framework for using code to create and manipulate databases. The DAO supplies a hierarchical set of objects that use the Microsoft Jet database engine to access data and database structure in a MicroSoft Access database. In an intermediate NED prototype between NED-1 and NED-2, programs could simply call functions in the DAODLL to read, update, and delete data from an MS Access database. For efficiency, all PnP modules called functions in the DAODLL directly. One advantage of this method is that PnP modules do not have to implement their own routines to access MS Access databases.

However, the DAODLL is designed for MS Access 97, and the DAO cannot access databases in MS Access 2000. To retain the advantages of the design and make it compatible with MS Access 2000, we use the ODBC driver for SQL to access the databases in NED-2. LPA (Logic Programming Associates) provides the ProData toolkit with built-in predicates to load and query databases. ODBC is an open standard that

provides a common set of Application Programming Interface (API) calls to manipulate databases. Some advantages to the ODBC API over native database APIs are as follows. (1) ODBC provides a single API to any database with an ODBC driver. If a developer wants to use Sybase, the API calls are the same when using ODBC. When using native database APIs, the developer would have to know the Sybase API. (2) Applications can be easily ported from one database to another. This enables you to access data from more than one type of database from the same application. (3) Applications built with ODBC can take advantage of the majority of the functionality of the database and are as fast as applications developed using native database drivers.

One of my contributions was to rewrite many of the Prolog routines that used DAO. With the new data structure, we query the data with Prolog using SQL and an ODBC driver. This is a standard way to manage the relational databases. In this process, I also implemented the routines to generate the help files in HTML format from special databases defining system variables, goals, and rules. Another member of the NED programmer team, Fred Maier, implemented routines to link the database to the Blackboard.

2.3. REPORTING UTILITY

In NED-1, two of the most important reports are the Stand Goal Analysis and the Management Unit Goal Analysis. However, these two reports are the only documents in NED-1 to be produced by the knowledge-based systems described earlier. This indicates

that the utilization efficiency of the knowledge base module is low. In NED-2, we will also use knowledge-based systems implemented in Prolog to generate reports comparing the performance of different treatment plans over time.

All reports and documents in NED-l will be converted to HTML files and accessed using a Web browser. In NED-2, reports are generated as HTML files by DCMs using a rule-based system. This will enhance the quality and flexibility of the reports.

Geneho Kim developed DCMs to generate reports. In Kim's codes, goal analysis and report generation functions are intertwined in a single DCM. We have separated these two functions. Analysis will be performed by one DCM and the results put on the blackboard. Then, another DCM will compile the results into reports.

2.4. KNOWLEDGE REPRESENTATION

In NED-1, a decision support system toolkit (DSSTools) written in Prolog was used for developing rules of the knowledge base (Twery *et al*., 2000; Nute *et al.,* 2000.) DSSTools provides a blackboard architecture for cross module communication and the routines to maintain it (Nute and Zhu, 1996, Engemore and Morgan, 1986) Working with DSSTools, NED-1 can run external simulation models, use a hypertext system to generate detailed explanations, write the logic of the inferring process to a trace file that the user can later examine, and prompt the user to input data. Interested readers will find a full explanation for each DSSTools component in Nute and Zhu (1996).

However, the rules for wildlife management in NED-1 are not in the format that DSSTools requires. The rules are created at run time using HAMMOD, a table that contains a set of habitat combinations for each species. Cleveland *et al.*, (1998) developed wildlife Habitat Assessment matrix. The assessment matrix lists habitat structural features across the top in columns and functional group numbers in rows. If a functional group uses a structural feature then a number is put at the row and column intersection. Even if suitable structural features are available in the forest stand, some wildlife species will not inhabit particular forest types or size classes. So, individual tables are used to list those wildlife species. The effect of this assessment procedure is a list of species that use the structural features present in a stand. A Prolog program has been developed that uses the assessment matrix. It automatically codes the answers, checks the database, and gives a list of individual wildlife species whose habitat requirements are met. Fuzzy values are used in the wildlife rules to evaluate how well the forest satisfies habitat requirements for a species. The idea behind fuzzy reasoning is that an object may belong to a class "more or less" (Nute *et al*. 2000). The threshold, which is a value associated with the observable variable in a desired future condition (DFC), can have this kind of fuzziness. A DFC is an observable variable together with a desired value for that variable. (See Nute *et al.* 1999 for details). The actual value for the variable is regarded as definitely satisfying the DFC if it is at least 5% above the threshold, it is regarded as definitely failing the DFC if it is at least 5% under the threshold, and it is reported as marginally satisfying or marginally failing the DFC if it is within 5% above or below the threshold. Computation of these fuzzy values is integrated into the wildlife

rules in NED-1. In NED-1, the wildlife ‘rules’ are really a procedure for interpreting the tables described above rather than specific habitat rules for specific species. The representation of the knowledge is so complex that it will be difficult to add new knowledge to the system. A major development effort in converting NED-1 to NED-2 is to rewrite the rules used for wildlife management according to the specification of DSSTools. NED-2 wildlife rules are separated from the HAMMOD tables and and the computation of fuzzy values. Dr. Donald Nute developed the fuzzy inference engine used to interpret these stand-alone rules and my research focused on creating a program that read the tables and wrote individual species habitat rules in the DSSTools format.

CHAPTER 3

IMPLEMENTATION

The architecture for NED-2 is radically different from the architecture for NED-1 as illustrated by (Nute *et al.* 2000). NED-2 has more system functionality. This thesis describes some important development efforts in the conversion process, particularly for wildlife management. The basic NED-2 architecture is a blackboard architecture with Domain Control Modules (DCMs) looking at the blackboard and handling requests for services. The blackboard itself is a global database in memory in which the system stores information provided by the user, conclusions inferred during a consultation, and requests for some task to be performed. One or more DCMs manage the interface written in C++ (PnP modules). Others perform wildlife goal analysis (inference), simulation (setting up and calling external programs), etc. For the details of the NED-2 architecture, see (Nute *et al.* 2000).

3.1. INTELLIGENT MODULE MANAGER

In order to add more intelligence to NED, we made changes in the procedural components of NED that make it easier to move overall control of NED to DSSTools. These modules are called by DCMs.

The user interface control DCM controls all the PnP modules. It gets requests from the user through the PnP modules and puts the requests on the blackboard. Analysis DCMs perform goal satisfaction analysis after it is requested. After management objectives and information about current conditions in the forest become available, goal satisfaction analysis can be done. The aim of goal satisfaction analysis is to find out how well the forest satisfies the user's objective at a chosen time. Simulation DCMs run external simulation programs to produce snapshots of stands at future times under different management plans. Report DCMs collect data and results from analysis and integrate them into HTML reports. Data visualization DCMs display data and results visually by outputting them to GIS or other systems.

The user picks wildlife species of interest through the NED-2 main window. When the user finishes the selection, all the information is saved in the working database. After that, the analysis DCM calls the NED fuzzy values inference engine to see if habitat for those species is present for a particular view of the forest. A view is a list of snapshots that depict the management unit as a point in time, either the present or a simulated future. Results of this analysis are put on the blackboard so that any module that uses those facts can see them. The Wildlife Reports DCM obtains all the information about user-selected goals including facts from analysis and compiles them into an HTML summary.

In NED-2, the intelligent module manager (IMM) is a mechanism controlling the modules of NED. The blackboard of the IMM stores knowledge about the intermediate data and user requests received from the PnP modules. The IMM registers PnP modules,

intelligently controls execution of PnP modules, monitors and communicates with various user interfaces, and provides data and control data access.

3.2. CONVERTING THE WILDLIFE RULES TO DSSTOOLS FORMAT

In NED-1, a set of procedures encoded as forward chaining rules reads two text files that contain sets of habitat combinations for each species. These procedures also manipulate fuzzy values, asserting and retracting them as computation proceeds. In order to make self-contained backward chaining DSSTools rules, a program was developed using Prolog to generate the new rules based on the forward-chaining production rules and the text files.

The first step of the rule generating procedure is reading the two text format files and producing a list of habitat requirements associated with each species code and a list of species codes and names. In the second step, the rule generator generates rules for each species in the species code list. At the same time, Donald Nute built a custom backward-chaining inference engine for NED-2 using fuzzy values. I provide the following rule as an example of the wildlife habitat rules, and I will also describe how the inference engine works using the example.

```
rule(nedcf, stand_analysis, wi_ham_marbled_salamander, 1.0,
      (
       habitat_available([snapshot(S1), species(marbled_salamander)],
       passed) :-
            member_of([`SNAPSHOT` = S1], stand_ham_type,
                  [`red_maple`, `white pine/red oak/red_maple`]),
            (
            equal([`SNAPSHOT` = S1],stand_temp_ponds,1);
            equal([`SNAPSHOT` = S1],stand_perm_ponds,1)
            ),
```

```
        more([`SNAPSHOT` = S1],stand_woody_debris,50),
        more([`SNAPSHOT` = S1],stand_rocky,25),
        more([`SNAPSHOT` = S1],stand_litter,30)
),
[`This rule determines a degree of confidence that a snapshot
of`, `a stand provides habitat for Marbled Salamander. `]).
```

When the backward-chaining inference engine is called, a list of goals is sent to the inference engine. The goals are in the form of attribute-object pairs. For example, `habitat_available([snapshot(S), species(marbled_salamander)])` is one of such pairs. `snapshot(S)` will represent a specific stand at a specific time under a specific management plan. The snapshot is a set of conditions on a stand. A threshold, which is usually 0.5, is also sent to the inference engine. The threshold is used as a value that the fuzzy value must meet before a conclusion is added to the blackboard. Then the inference engine will use the wildlife habitat rules to determine whether any of the snapshots mentioned in the goal set provide habitat for the associated species with a fuzzy value greater than the threshold. We can interpret this as the degree to which a snapshot-species pair belongs to the class of pairs for which the snapshot provides habitat for the species. When one of these pairs that belonged to this class has high enough degree, the attribute-object-value (AOV) triple succeeds with the value `passed`, and the fact

`habitat_aviable([snapshot(S),species(Species)],passed)`

is put on the blackboard where report generators and other DCMs can find it. The name of the rule used in getting the result and the actual fuzzy value related to the result are also put on the blackboard along with the AOV triple.

3.3. REPORT GENERATION

When the inference engine finds the results and puts them on the blackboard, then the report generator takes the facts from the blackboard and puts them in the reports. The user can view the selected reports using a Web Browser. When this module is called, HTML reports are generated for all the user-selected goals. The goal analysis function in the NED-2 system will be enhanced. The purpose of this analysis in NED-1 is to find out how well the forest currently satisfies the user's objectives. For example, the analysis can determine whether the DFCs for the wildlife goals that have been selected are met, or give a list of wildlife species for which appropriate habitat can be found on the management unit. NED-2 will also be able to perform goal analysis on simulated future states of the forest.

The report-generating DCM generates an HTML page that shows all the user-selected goals and calls the Reports Generation model to generate all the selected reports. The report page contains the user-selected goals and a short description of each goal. There is a link in each goal name to an individual page describing requirements for that goal. The quality and flexibility of the reports are improved with a rule-based system for generating reports.

In NED-1, the analysis and report generation were intertwined. The analysis is run during report generation. While in NED-2, the analysis and report generation are separated from each other. When the analysis is done, all facts are put on the blackboard

where the report generator can get all the information for the reports. Separating these functions will simplify maintenance and expansion of NED-2.

3.4. DATABASE MANAGEMENT

NED-2's Data Manager Module (DMM) is responsible for providing data manipulation and data access control. When the user opens a file, the blackboard utilities will automatically look in the database for information, effectively extending the blackboard to include the specified database. The loaded data can be used by any module. Since NED-2 uses MS Access databases to store data, this module provides accessibility to MS Access databases.

The DMM has both a C++ and a Prolog component. Several modules in the interface PnP can read from and write to the database. Inventory is entered, management plans are developed, and goals and reports are selected using the PnP. The results are written to the databases by the PnP. Prolog must access the databases to get information to feed to simulations, to perform analyses, and to incorporate into reports or visual displays. The Prolog component of the DMM is incorporated into the utilities that handle the blackboard. Thus to DCMs in the system, information stored as temporary Prolog clauses or as values for fields in the databases are accessed in the same way. The databases effectively become part of the Prolog blackboard, making it possible for the DCM developer to ignore the actual source of the information the DCM will use.

Another function of the DMM is to make sure that all the data are up-to-date. Each PnP module declares data request type (read only or modifiable). When a modifiable data request is made, the data is transferred to the module that requested the data, then all data requests are blocked until the module finishes its task. Hence, any modifiable request must tell the DMM when its request is finished. Similarly, when a module is in the middle of a long computation that has the possibility of modifying data, it informs the DMM of the starting and ending of the computation so that the DMM can block any other data access during the computation

CHAPTER 4

SUMMARY OF RESULTS

NED is a collection of decision-making support tools for forest management for multiple values and purposes. A variety of different tools have been developed and distributed to help a diversity of users. NED-1 analyzes current situations and contains user goal selection for comparison of existing conditions with desired future conditions. The inferential part of the decision process in NED-1 applies knowledge-based systems and a blackboard architecture. However, the original implementation of NED-1 has its weaknesses, such as data retrieval problems, the intertwining of declarative and procedural knowledge, and module management handling by the procedural front-end program. These weaknesses result in the inefficiency of the NED-1 system and the non-applicability of some of the AI techniques that might otherwise be used.

NED-2 revises the NED architecture. The blackboard architecture is used not only in the inference server, but also in the entire NED system. With the new NED architecture, NED-2 uses MS Access databases to store data and the system can access the data in a more efficient and standard manner. Since it separates declarative knowledge and procedures, it is easier in NED-2 to modify (add/delete) knowledge. NED-2 coordinates rule-based systems, visualization tools, and the simulations in an intelligent way. NED-2 uses Web browsers to display reports, so users can easily choose to view either a list of selected reports or an individual report.

With the rigorous design of the software implementation, it is very easy to add or delete variables, modify goals and DFCs, change the input formats, modify, add or delete knowledge, and change output reports. The purpose of NED is to enable users to use the different functions supported by the NED DSS in a variety of manners, and give users the flexibility to find solutions in an easy manner and to change previous solutions under new conditions.

BIBLIOGRAPHY

Alban, L.M., Thomasma, S.A., Twery, M.J., 1995. Forest stewardship planning guide user's manual (Version 1.00) [Computer Program]. Gen. Tech. Rep. NE-203. U. S. Department of Agriculture, Forest Service, Northeastern Forest Experiment Station, Radnor, PA, 15pp.

Clark, F.B., J.G. Hutchinson (Eds.). 1989. Center hardwood notes. USDA Foresty Service NC-Notes 9.02.

Cleveland, H.M., J.C. Finley. 1998. Assessing Forest Wildlife diversity in Pennsylvania. Northern Journal of Applied Forestry. Volume 15, Number 2: 77-85.

Engemore, R., Morgan, T. (Eds.), 1986, Blackboard Systems. Addison-Wesley, Reading, MA.

Harvey, H.M., 1994. Inventory process for determining forest stand habitat structure and assessment model for wildlife habitat. M.S. thesis. The Pennsylvania State University. University Park, PA, 221 pp.

Harvey, H.M. and Finley, J.C., 1995. Stand level assessment -- a piece of the biodiversity puzzle. In: Finley, J.C. and Jones, S.B., Editors, 1995. Proceedings of the 1995 Penn State Forest Resources Issues Conference, The Pennsylvania State University, University Park, PA, pp. 43¯50.

Kim, Geneho, Donald Nute, and H. Michael Rauscher. In press. *DSSTOOLS 2000: A Toolkit for the Development of Decision Support Systems in Prolog*. USDA FOREST SERVICE, Southern Research Station, Asheville, NC, GTR-SRS-XXX.

Hoffman, R.E., Palmer, J.F., 1996. Silviculture and forest aesthetics within stands. The New York Center for Forestry Research and Development, Pub. #2, State University of New York, College of Environmental Sciences and Forestry, Syracuse, NY.

Hoffman, R.E., Palmer, J.F., Harrilchak, M.A., 1996. Silviculture and forest aesthetics: an annotated bibliography. The New York Center for Forestry Research and Development, Pub. #4, State University of New York, College of Environmental Sciences and Forestry, Syracuse, NY.

Marquis, D.A., 1990. A multi-resource silvicultural decision model for forests of the northeastern United States. In: XIX IUFRO World Congress, 5-11 August 1990, Montreal, Canada, Division 1, Volume 1, Proceedings, pp. 419-431.

Mowrer, H. T. (Technical Compiler), 1997. Decision support system for ecosystem management: an evaluation of existing systems. General Technical Report RM-GTR-296. U. S. Department of Agriculture, Forect Service, Rocky Mountin Research Station, Fort Collins, CO, 154 pp.

Nute, Donald, Zhu Goujun. 1996. DSSTOOLS: a toolkit for development of decision support system in PROLOG. Master of Science Thesis. University of Georiga. Athens. P. 106.

Nute, Donald, Nath, Shree; Rosenberg, Gregg; Verma, Brahm; Rauscher, H. Michael; Twery, Mark J.; and Grove, Morgan. 1999. Goals and a Goal Orientation in Decision Support Systems for Ecosystem Research Organizations Conference: The Application of Scientific Knowledge to Decision-making in Managing Forest Ecosystems, Asheville, NC, 3-7, 1999.

Nute, Donald, 2000. a Multi-criterial Decision Support System for Forest management.

Rauscher, H.M, Twery, M., Palmer, J., Hoffman, R., Stout, S., Steinman, J., Kollasch, P., Bennett, D., Thomasma, L., Hornbeck, J., Worth, C.V., 1995. Northeast Decision Model Design Document. AI Appl. 9(3): Insert: 3633 kb; 813 chunks; 978 links (electronic).

Rauscher, H.M., 1999. Ecosystem management decision support for federal forests of the United States: a review. For. Ecol. Manage., Jan. 1999.

Rauscher, H.M., Lloyd, F.T., Loftis, D.L., Twery, M.J., 2000. A practical decision analysis process for forest ecosystem management. Comput. Electron. Agric., 27, 195-226.

Simpson, B.T., Kollasch, R.P., Twery, M.J., Schuler, T.M., 1995. NED/SIPS user's manual: Northeast decision model stand inventory processor and simulator (Version 1.00) [Computer program]. Gen. Tech. Rep. NE-205. U.S. Department of Agriculture, Forest Service, Northeastern Forest Experiment Station, Radnor, PA, 103 pp.

Steinman, J., 1996. ForestHealth. Unpublished report on file with USDA Forest Service, Burlington, VT.

Thomasma, S.A., Thomasma, L.E., Twery, M.J., 1998. NEWILD (version 1.0) User's Manual. [Computer program]. Gen. Tech. Rep. NE-242. U.S. Department of Agriculture, Forest Service, Northeastern Research Station, Radnor, PA, 28 pp. [1 computer disk; user's manual].

Turban, E., 1993. Decision Support and Expert Systems; Management Support Systems. MacMillan Publishing, New York, 833pp.

Twery, M. J., Deborah J. B., Kollasch, R. P., Thomasma, S.A. 1997 Annual Convention & Exposition Technical Papers. Volume 4. 331pp.

Twery, Mark J.; Rauscher, H. Michael; Bennett, Deborah J., 2000. NED-1: integrated analyses for forest stewardship decisions. Computers and Electronics in Agriculture. 27: 167-193pp.