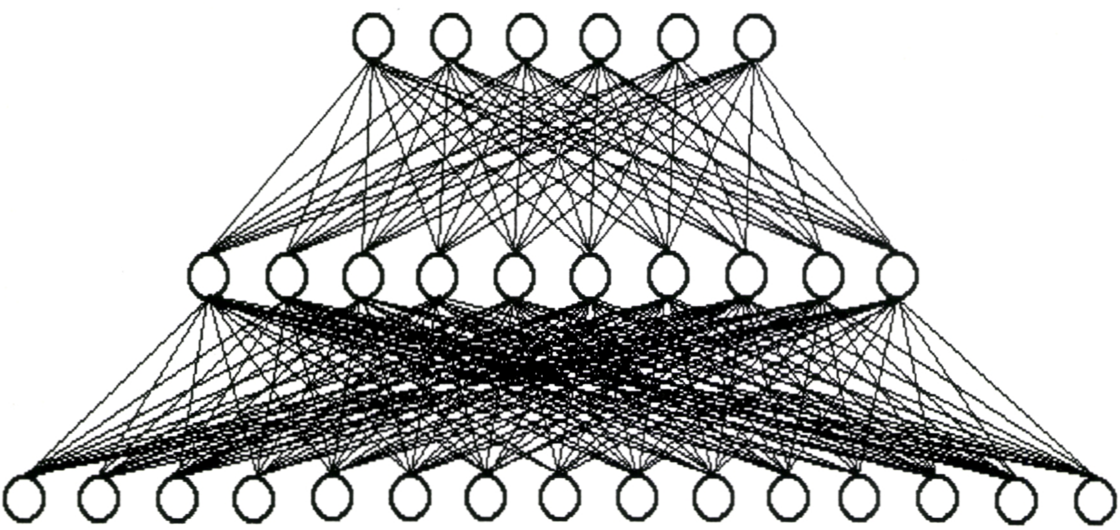
METSÄNTUTKIMUSLAITOKSEN
TIEDONANTOJA 395

Current Advances in the Use of Computers in Forest Research

Workshop of the IUFRO Working Party S4.11-03
Joensuu, Finland, February 14, 1991

Hannu Saarenmaa
Editor



Helsinki 1991

METSÄNTUTKIMUSLAITOKSEN TIEDONANTOJA
[BULLETINS OF THE FINNISH FOREST RESEARCH INSTITUTE]
VOL. 395

Current Advances in the Use of Computers in Forest Research

Workshop of the IUFRO Working Party S4.11-03
Joensuu, Finland, February 14, 1991

Edited by
Hannu Saarenmaa

Helsinki 1991

Finnish Forest Research Institute
Department of Mathematics
Unioninkatu 40A
00170 Helsinki
Finland

ODC 946.3 + 945.14
ISBN 951-40-1181-3
ISSN 0358-4283

Helsinki 1991

Hakapaino Oy, Helsinki 1992

Contents

Reports from the Nordic graduate course on "Information Systems in Forest Management Planning" held in Mekrijärvi on February 4-13, 1991

Workshop reports

Preface

The IUFRO working party Computers in Forest Research has had annual workshops in the North America since 1986. These sessions have been a lively forum for exchange of ideas and reports of ongoing work. In Europe, the working party has been less active. The first European workshop was held in 1989.

This volume reports theachievements at the second workshop that took place at the University of Joensuu, Finland, on February 14th, 1991. Eleven papers were presented during one day. These summarize current activites in the research of the use of computers in the solving problems in forestry and forest research. Emphasis is in various methods from the field of artificial intelligence. Neural networks seem to especially promising in the building of predictive models. Another trend present in the papers is integrated systems. Forestry problems are so multifaceted that they normally require various approaches such as expert systems, databases, GIS, and mathematical programming to be combined. Modern object-oriented tools let scientists do that -- not easily, but nevertheless with solid modules and user intefaces.

Prior to the workshop on February 4-13, a graduate teaching course about these integration themes with the title Information systems in forest management planning was held at the Mekrijärvi Research Station of the University of Joensuu. The first five papers in these proceedings were compiled during that course. All the papers in this volume represent ongoing work. Indeed, in this science of making computer applications to solve forestry problems projects rarely become finished, since products and results evolve during work. A computer application is a moving target: when you begin, you cannot predict what the final product will be, since many external factors like user input and rapid development of technology affect the results. Because of this, reporting ongoing work is important part of the process.

The workshop remained entirely Scandinavian, although we tried to attract participants from all over Europe. In future, the working party has to develop a more comprehensive mailing list of active participants in Europe. Judging from the current activies, the topics that will dominate in next workshops are connected to object-oriented modeling.

The course was attended by 24 participants and 4 teachers. It was sponsored by the Nordic Council of Ministers. Professor Rihko Haarlaa and professor Simo Poso of the University of Helsinki handled the financing. The teachers were Dr. Peter Kourtz, Canadian Forestry Service, Petawawa, Ontario; Dr. Dougals K. Loh, Range Science Department, Texas A&M University; Ms. Tuula Nuutinen, Department of Geography, University of Edinburgh, Scotland; and the undersigned. Practical matters were handled by Mr. Jari Jämsä, Mr. Teijo Palander, and the staff at Mekrijärvi. Course was also supported by Viltec Oy and Scandinavian Softline Technology Oy. My sincerest thanks go to all that made the course and the workshop to happen.

Helsinki, March 20, 1991

Hannu Saarenmaa

Estimating inoptimality losses in harvesting decisions using neural networks

Tomas Lämås, Göran Ståhl and Tomas Thuresson
Swedish University of Agricultural Sciences
Department of Biometry and Forest Management Planning
Umeå, Sweden

Abstract

In this paper, the problem of assigning priorities to different harvesting treatments, to all the compartments in a forest holding, is addressed in a new way by using neural computing.

One of the traditional ways of performing forestry planning is to make a strategic plan based on a sample of the compartments in a forest. These sample compartments are inventoried by using a very accurate objective method. In the strategic plan total cutting levels for different time periods are decided. The problem then is to decide optimal treatments for all the compartments in the forest. Because of high costs for objective inventory methods, ocular methods are used when data are collected from all the compartments.

The results from the strategic plan should then be transferred to the operational short-term planning. This is traditionally done by regression analysis, where stand data are predictor variables and the inoptimality loss for a certain treatment the predicted variable. An inoptimality loss is the loss incurred if a compartment is treated at the wrong time or by the wrong method. The losses are expressed in net present values.

In this study, neural computing was used instead of regression analysis for the prediction of inoptimality losses. Neural computing is considered to be one branch of artificial intelligence. The results show that neural computing in this case performed slightly better than regression analysis. The study, however, is very restricted and should be looked upon as a pilot study.

1. Introduction

1.1 Background

An important matter in forestry is the choice of treatment for each forestry compartment. The questions are when, how and where to carry out different silvicultural and harvesting treatments.

In this paper, we will study a question concerning the choice and timing of harvesting treatments. Large values can be saved if this is done correct.

The planning concept used here is described before the details of the problem are presented. In summary, forestry planning can be divided into strategic planning and operational planning. In the strategic planning, we are interested in identifying the over-all forestry alternatives. Important factors considered are total cutting volumes in different time periods, total net incomes in different periods, etc. To perform strategic planning it is not necessary to use data from all the compartments in a forest holding. A sample is quite sufficient, and is most often used. Data from objective inventory methods are reliable, but expensive to collect. In our problem, strategic planning is based on a sample of compartments.

In operational short-term planning, on the other hand, the problem is to assign the correct treatment to all compartments in a forest. To accomplish this, data is needed from all the compartments. Most commonly, subjective inventory methods are used when stand data from the whole forest are required. Objective methods are too expensive to use. Subjective data, however, are often subject to large errors (e.g. the systematic error is often large). This is one of the reasons why subjective data are not used in the strategic planning.

In the concept of planning used in this paper, the results from the strategic planning are used to provide guidelines for the operational planning. The traditional way of doing this is to make regression functions that predict inoptimality losses (Jacobsson, 1986) for different treatments for all compartments. These functions are based on data from the sample compartments. Predictor variables are the subjectively collected stand data that are available for all compartments.

An inoptimality loss is the loss incurred if a compartment is treated at the wrong time or by the wrong treatment (e.g. clear cutting instead of thinning). The inoptimality losses are expressed as losses in net present value. If a compartment is treated optimally the inoptimality loss will be zero.

Inoptimality losses are used in order to determine a compartment's priority for different treatments. The total harvesting levels (volume, net income, etc) are set in the strategic plan.

1.2 Aim of the study

The aim of this study is to compare two different methods of predicting inoptimality losses for compartments that are not included in the sample. The traditional regression approach provides the basis for this comparison. The regression results are then compared with the results obtained by using neural computing (Anon, 1990).

2. Methods and data

Two different methods of predicting inoptimality losses have been used. The first one is regression analysis and the second is neural computing.

2.1 Regression functions

Regression functions, predicting inoptimality losses for thinning, clear cutting and no treatment, have been made. The functions have been constructed by using multiple linear regression analysis. A major problem when constructing these functions is the difficulty of finding a realistic model as a basis for the regression. This makes the construction of functions very hazardous. Very often a lot of different models are tested. Then, the model that provides the best fit is selected. By selecting regression models in this way, there is a significant risk that a model that happens to suit only the existing data well is chosen. When the function is used for prediction, large errors in the choice and timing of treatments can result.

In this study, regression functions were determined without any particular model as a basis. Stand data and some transformations of stand data were used in MINITAB's BREG (best regression) command. By using this command, MINITAB finds functions that provides the best fit to existing data (the fit is measured as adjusted R-square). The best function for all possible numbers of prediction variables is presented[1] (Table 1). We then chose the funtion that had the largest R-square value. The only model type tested was the simple additive model, that can be described as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$$

Table 1. Regression functions for predictions of inoptimality losses (IL), (SEK/ha). Volume is expressed in cubicmetres/ha, site index in metres, and diameter in millimetres. The coefficients are presented as well as the residual errors of the functions.

	Regression coefficients		
Variables	IL-thinn.	IL-clear cut	IL-no treatment
Constant	0	19923	- 3443
Age	- 41.5	0	0
Vol. pine	64.2	184	27.5
Vol. spruce	24.5	0	0
Vol. hardwoods	0	- 53.8	11.7
Site index	0	2887	128
Diameter	0	- 287	0
Tot vol/diameter	- 1115	- 16296	0
Residual error	2019	5520	1058

1. MINITAB provides some results as decimal numbers, and some as integers without any decimals.

The fit of the regression functions are rather poor, which is expressed by high residual errors. One reason to this is that sample compartments of all ages have been used for the calculation of each function. Usually, only old compartments are used when the IL-function for clear cutting is constructed. Also, when the IL-function for thinning is constructed, only middle- aged compartments are used. In this case, however, our material was very limited. Therefore, the functions were made without selecting a specific subset of the sample compartments.

2.2 Neural computing

The other method used, for prediction of inoptimality losses for different treatments, was neural computing (Anon, 1990).

Neural computing is often referred to as a kind of artificial intelligence. The basic idea of neural computing is to imitate the way the human brain is thought to work.

Many theories of human intelligence are described by Minsky (1985). One theory is the knowledge-lines (K-lines) concept for learning and storing knowledge. In this theory the brain is supposed to consist of a many "unintelligent agents", that cooperate in a way that makes the system of agents intelligent. The brain is described as a lot of "agents", connected to each other in a network. The learning process is then a "flow of data" through the network. Knowledge is described as a very specific flow of data from one agent to another. In the human brain the agents could be the neurons in the nervous cells and the network between them would then be the axons.

Neural computing imitates this pattern. In a typical neural network computing system, data are entered at the "input level". Then, data pass through "hidden layers" of "processing elements" (the "agents" above) that are connected to each other in certain ways, according to the kind of network used. The specific network used in this study is called a "back-propagation network", (Anon, 1990). The discriptions of neural networks that follow apply to this kind of network.

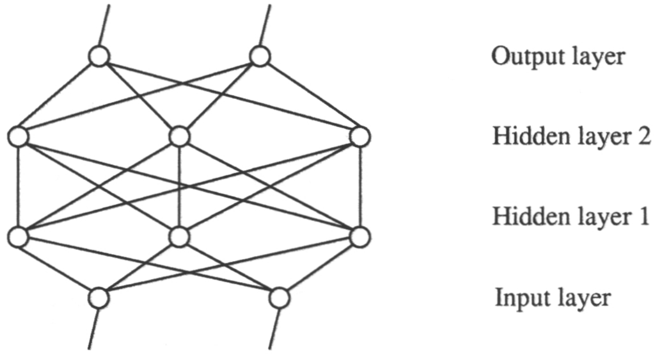
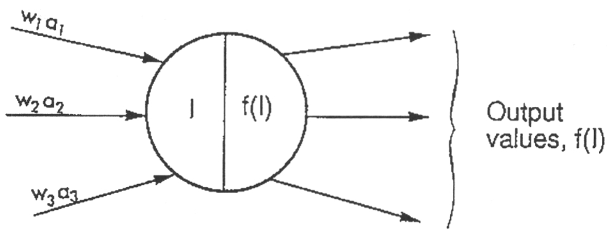


Figure 1. Outlines of a back-propagation neural network. The number of processing elements in each layer might vary as well as the number of hidden layers



w_i = weighting factors $\quad$ $I = \Sigma\, w_i\, a_i$
a_i = incoming values $\quad$ $f(I)$ = transformation function

Figure 2 illustrates the details within a single processing element.

Each input processing element corresponds to a specific input variable. Each processing element, including the input elements, are connected by arcs to all the processing elements in the layer above. This structure continues until the last layer, which is the output layer (see fig. 1). The processing elements of the output layer correspond to output factors. Unlike regression, there can be many such outputs.

Along each arc in the network, there is a weighting factor. Processing proceeds through the input level, to the hidden levels and finally to the output level. As a value passes through each arc joining two processing elements, it is modified by the weighting factor associated with the arc. Within a specific processing element, the weighted values being received from each arc into that processing element are summed (see fig. 2). This sum is further modified by a nonlinear transfer function and the result is then forwarded along the outgoing arcs.

The knowledge that the neural network contains is encoded in the many weights. The values of the weights are determined by an iterative process involving paired input and output sets. These input/output sets are supplied by the user and correspond to data sets similar to those used in regression analysis.

The process of defining the weights (called training or learning) begins by assigning a random number, usually between zero and one, to each of the weights. For a specific iteration a set of input values is presented to the input processing elements. The values are then propagated through the network, being modified by the weights, summation process and transfer functions. After the final transformation associated with each output processing element, the results are compared to the output values corresponding to the input set.

The difference between the network's predicted output and the observed output of the training set is called the error. With each iteration, this error is redistributed back to each processing element in a manner that credits the specific processing element with its share of the responsibility for the total error. Gradient calculus methods are used to propagate this error backward through the network to each processing element. The weight of each arc is then adjusted in proportion to this error. The rate of this adjustment is under control of the user. By trial and error adjustment of this rate, the network can be made to converge toward very small errors.

The adjustment of the weights is by very small increments. Only through an iterative process are the weights modified in a way that brings the predicted values near to the observed values. In order to achieve a large number of iterations, the original training set of input/output pairs is randomly sampled many times. This process is repeated often several hundered thousand times in order to achieve convergence.

Compared to regression functions, neural computing is said to be less susceptible to missing data or outliers, that often lead to prediction errors in regression analysis (Anon, 1990).

2.3 Description of the study's network

In this study, a back-propagation network consisting of two hidden layers was used. The input layer had 6 elements into which stand data were fed. The stand data used were volume (/ha) of pine, spruce and hardwoods respectively, age, site index, and diameter. The first hidden layer consisted of 10 processing elements and the second of 6 elements. The output layer had 3 elements; the inoptimality losses associated with thinning, clear cutting and no treatment. The original inoptimality losses were modified by rescaling to a smaller range which seemed to improve the results.

As the network was constructed it was set to run through the training data 30 000 times.

2.4 Training and test data

Data sets were used from a sample of compartments taken from a forest holding in the county of Småland in Sweden. Out of approxamitely 400 compartments, 56 compartments were inventoried by an objective circular plot method. For various reasons only 35 of these compartments could be used in this study. For every sample compartment inoptimality losses for thinning, clear cutting and no treatment were calculated. This was done by using the Forest Management Planning Package (Jacobsson et al, 1987). The interest rate used was 2.5 %.

The inoptimality losses are treated as true, even if they may contain some random errors.

The data were divided into two parts. One part (20 compartments) was used to train the neural network and to build the regression functions. The other part (15 compartments) was used to evaluate the results. For these 15 compartments, predictions of inoptimality losses were carried out by using both the "trained" neural net and the regression functions. The predictions were then compared to the true inoptimality losses.

At the end of the neural network training process, using the training data, the weights in the network structure are stored in a way that can easily be reconstructed. Each input/output pair from the test set is then put through this network where the identical processes of weighting, summing and transforming are carried out once. This parallels the process of using a regression equation. The result is an output set that can be compared to the output of the original input/output pair of the test data set..

Table 2. Deviations between true inoptimality losses an inoptimality losses predicted by using regression functions and neural computing. The results are expressed as root mean square errors (RMSE). The unit of the values is SEK/ha.

Treatment	Prediction results (RMSE)	
	Regression	Neural computing
No treatment	1533	702
Thinning	2741	2463
Clear cutting	11030	7304

3. Results and discussion

In Table 2,, the results are presented as the root mean square error (RMSE) of the deviations between predicted values and true values.

The results show that neural computing seems to have performed slightly better than regression analysis. The prediction capability, however, seems to be rather limited for both methods on this particular data set. The root mean square errors are quite large.

These results should not be taken as general. The study is very restricted and should be regarded as a pilot study.

An interesting comparison is to study the residual errors from the regression analysis (Table 1) and the RMSE's presented in Table 2. The residual errors from the regression analysis are considerably lower then the RMSE's in Table 2. This shows the problem of making regression functions without a sound model as a basis.

The same comments can be made for neural computing. The predictions on training data worked incredibly well. For prediction purposes the results were poorer. The alternatives provided by the software to adjust the network to perform in different ways were, however, not used to the full extent in this study. Another matter related to the performance of the neural network is the number of training iterations used. In our study 30 000 iterations were used, while professional users seldom use less than several hundered thousand iterations.

In conclusion, it appears that neural networks could be very useful for various tasks concerning forest management planning, as well as in other forestry applications. This is especially the case when no model to describe a system is evident.

In these cases, however, it is important to evaluate the prediction results carefully. The approach used here to divide the material into two parts (one for training and one for testing), is a very simple way of evaluating the result. A more efficient way would be to use cross-validation. Then, a larger part of the data would take an active part as training data for the neural network. When cross-validation is used, data is processed through the same model many consecutive times. Each time one or more observations are left out and used to test the predicting capabilities. This procedure would lead to more accurate evaluations of the results.

Acknowledgements

The authors wish to thank Dr Peter Kourtz of the Petawawa National Forestry Institute, Forestry Canada, who introduced neural computing to us. He also helped us with this project. We would also like to thank Dr Hannu Saarenmaa of the Finnish Forestry Research Institute, Helsinki, Finland, who arranged the graduate course during which this paper has been written.

References

Anon, 1990. Neural Computing (manual). Neural Works Professional II/Plus and Neural Works Explorer. NeuralWare Inc, Pittsburgh, USA.

Jacobsson, J. 1986. Optimization and data requirements. - A forest management planning problem. Swedish University of Agricultural Sciences, Faculty of Forestry, Department of Biometry and Forest Management Planning, Umeå, Sweden.

Jacobsson, J, Jonsson, B & Kallur, H. 1987. Indelningspaketet, - ett helhetsgrepp på skoglig planering. Skogsfakta, inventering och ekonomi nr 17. Swedish University of Agricultural Sciences, Faculty of Forestry.

Minsky, M. 1988. The Society of Mind. Simon and Schuster, New York, USA.

Estimation of forest stand characteristics using neural networks and satellite remote sensing

Henrik Feychting, Olle Hagner and Mats Nilsson
Swedish University of Agricultural Sciences
901 83 Umeå, Sweden
Fax: +46 90 118116

Abstract

In this study Neural Networks were compared to regression analysis techniques for estimation of forest stand characteristics based on satellite remote sensing and field inventory data, both at sample plot and forest stand level. The performance of the two methods was evaluated using the coefficients of correlation between observed and predicted values. The respective performances were found to be similar although the estimates derived using the Neural Network models were slightly less precise.

1. Introduction

Modern forest management requires accurate and up to date information on the forest resources. Inventory methods based on satellite remote sensing exploits the relationships between field measurements of forest variables and spectral signatures in satellite imagery for prediction of forest characteristics. The technique may be used for both standwise inventory (Hagner 1990) and larger scale inventories such as national forest inventories (Tomppo 1990).

Neural networks are human attempts to simulate and understand what goes on in nervous systems, with the hope of capturing some of the power of these biological systems. Neural networks are built of neurons and are usually arranged in layers. The neurons in a layer are connected to many other neurons in other layers. The total input each neuron receives is processed and sent to the neurons in the next layer. Artificial neural networks are trained in an iterative process where weights are assigned to each inter-neuron connection. Several different strategies may be used in the training process (NeuralWare 1990).

The scope of this study was to assess the possibilities of using Neural Networks instead of regression analysis techniques for estimation of forest stand characteristics based on satellite remote sensing and field inventory data.

2. Material

2.1 Data

Two separate data sets were used. The first set consisted of field measurements on 226 permanent national forest inventory (NFI) sample plots from the Northen parts of Sweden and corresponding radiance data acquired with the LANDSAT TM sensor (date of acq. 21-06-1989). 175 of the plots were used for model calulations and the remaining 50 for evaluation. The following variables were registered on each plot:

-stem volume per hectare
-mean age
-site index
-tree species composition
-LANDSAT TM pixel values (bands 1-5 and 7)

The second data set consisted of stand variables for 80 reference stands. Three estimates of each variable were derived using separate methods; 1) subjective field inventory (15 minutes / stand) 2) regression estimates using Landsat TM radiance counts. 3) sample plot survey (20 circular sample plots/stand). The data was split into two parts, one used for modeling and one for evaluation. The variables registered were:

- stem volume per hectare
- mean age
- mean diameter
- tree species composition
- site index (subjective field inventory only)
- surveyor

2.2 Software

A Neural Network Software package called " NeuralWorks Professional II/PLUS" running on a SUN SPARC/IPC workstation was used (NeuralWare Inc. 1990) for the analysis.

3. Methods

Estimates derived by neural network and regression analysis were compared using the coefficient of correlation for observed and predicted values of stand variables.

3.1 Regression analysis

Twelve multiple linear regression models were used for estimation of stand characteristics, six for each data set.

3.2 Neural Network

The two network models tested (figure 1 and 2) were of so called "back-propagation" type (Rummelhart 1986). The hyperbolic tangent function was used as transfer function in each processing element (NeuralWare,Inc 1990).

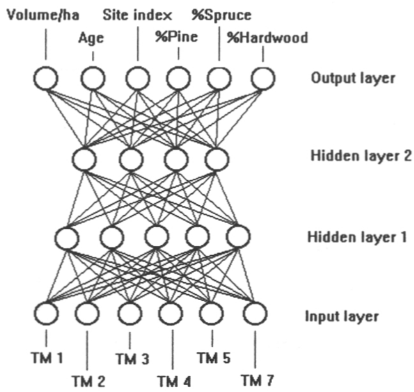


Figure 1. The Neural Network model used for plotwise estimation of forest characteristics. The six nodes in the input layer correspond to Landsat-TM (bands 1-5 and 6). The network has two hidden layers with 5 and 4 nodes respectively. The 6 output nodes correspond to estimates of plot variables (volume/ha, mean age, site index, %pine, %spruce and %broadleaves).

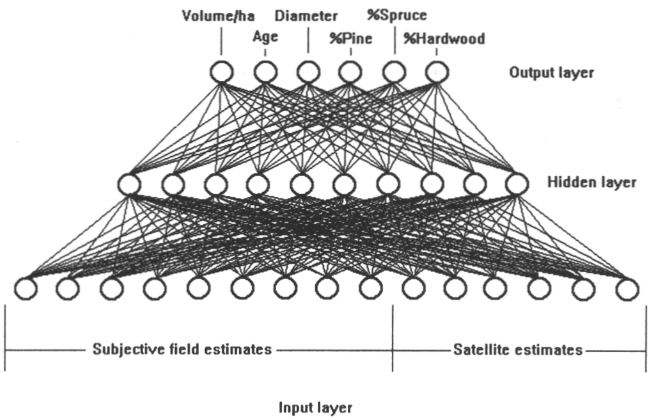


Figure 2. The Neural Network model used for standwise estimation of stand characteristics. 15 nodes in the input layer correspond to field inventory estimates and regression estimates based on Landsat-TM radiance values. The hidden layer contains 10 nodes connected to 6 nodes in the output layer, corresponding to output estimates of stand variables.

4. Results

The coefficents of correlation between obeserved and predicted values for various variables derived with neural network and regression models are shown in tables 1 and 2. The statistics were calculated for independant data sets not used for the construction of prediction models.

Table 1. Coefficients of correlation between measured sample plot variables and predicted values based on Landsat radiance data using one neural network model or six multiple linear regression models. The number of training and test samples were 176 and 50 respectively.

Plot variables	Coefficient of correlation (r)	
	Neural Networks	Regression
Stem volume/ha	0.58	0.60
Mean age	0.56	0.53
Site index	0.13	0.11
Prop. pine	0.60	0.63
Prop. spruce	0.54	0.67
Prop. hardwoods	0.72	0.72

Table 2. Coefficients of correlation calculated for obseved and predicted values of stand variables using Neural network and multiple linear regression models. The data used for estimation consisted of subjective field inventory and satellite remote sensing data. The correlation statistics are based on 40 independent reference stands.

Stand ariables	Coefficient of correlation (r)	
	Neural Networks	Regression
Stem volume/ha	0.83	0.87
Mean age	0.57	0.68
Mean diameter	0.81	0.90
Prop. pine	0.91	0.94
Prop. spruce	0.92	0.94
Prop. hardwoods	0.71	0.84

5. Discussion

The results indicate that the Neural Networks technique is an interesting alternative to regression analysis for estimation of stand characteristics using satellite remote sensing. The method seems to be very flexible, offering a magnitude of variation opportunities e.g. network design, learning strategies, error propagation methods, scaling, and others.

References

Hagner, O. 1990. Use of digital satellite data for stand delineation and estimation of stand varaibles by regression analysis and field inventory. Proceedings ISPRS commission VII, mid term symposium, September 17-21, 1990, Victoria, B.C. Canada.

NeuralWare, Inc. 1990. Users manual, Neural Computing. Penn Center West, Suite 227, Pittsburgh, Pennsylvania 15276.

Rummelhart, D.E. & McClelland,J.L. (Editors) 1986. Parallel Distributed Processing: Explorations in the Microsructure of Cognition. Volume I, Foundations. MIT Press.

Tomppo, E. 1990. Designing a satellite image-aided national forest survey in Finland. Proceedings SNS/IUFRO workshop "The usability of remote sensing for forest inventory and planning", 26-28 February 1990. Umeå, Sweden. Report 4, Remote Sensing Laboratory Swedish University of Agricultural Sciences. ISBN 91-576-4208-7.

Modelling the risk of moose browsing in forest plantations with neural networks

Ari Nikula and Jouni Väkevä[1]
Finnish Forest Research Institute, Rovaniemi Research Station
P.O.Box 16, SF-96301 Rovaniemi, Finland
nikula@roi.metla.fi

Abstract

Moose damage in forest plantations is mainly function of plantation characteristics, surrounding stands and vicinity of culture areas and roads. This study was concentrated to model the browsing intensity depending on tree species growing on the plantation. The model was built by using neural networks. The densities of 8 common tree species on a plantation were given as input data, and the output data produced by the model were the numbers of browsed pines and birch/hectar. The model was built and trained by using data collected from 125 plantations. Data from 124 plantations were used to test the model. Results of this preliminary study was that neural networks model was able to make predictions better or at least equal to those given by linear regression models. Neural networks was seen as promising tool to model this kind of complex system otherwise difficult to handle.

1. Introduction

Neural networks are human attempts to simulate and understand what goes on in nervous systems. When developing neural network solutions to problems, neither the knowledge nor the explicit rules for processing the knowledge are coded by the programmer. Neural networks, in contrast to being programmed, are trained. It learns the rules for processing the knowledge. This is done by adjusting the weight values in a highly connected network based on the example data. Neural networks can be thought as a very general model that is parametrized by the adjustable weights (see e.g. Rumelhart & McClelland 1986, Anderson & Rosenfeld 1988).

The basic unit in an artificial neural networks is referred as a “prosessing element”, PE (Fig. 1). It is analogous to the biological neuron cell. A processing element has many input paths (dendrites) and combines the values of input paths usually by a simple summation. The result is an internal activity level for the processing element. The combined input is then modified by a transfer function and is sent via output paths (axon) to other processing elements. Data is processed in numerical form in the network. Every processing element give certain weight to the data according to the transfer function.

---

1. Finnish Forest Research Institute, Department of Mathematics, Unioninkatu 40 A, 00170 Helsinki, Finland

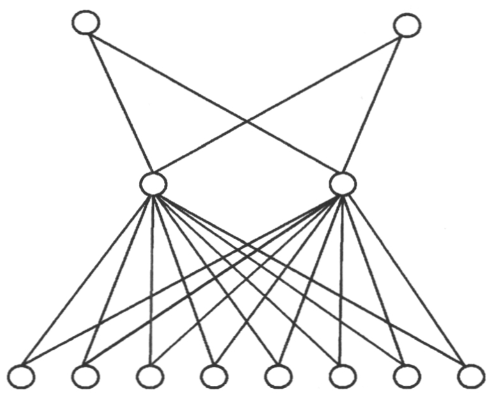


Figure 1. Schematic presentation of the neural network model used in the analysis.

A neural network consist of many processing elements, which have many connections to each others and are organized into groups called layers or buffers. Two of the layers have connections to the outside world. The data is presented to the network in an input buffer and the response to a given input is holded in an output buffer. Between these layers there are usually from one to three hidden layers.

The aim of this study was to model the browsing intensity of moose on pine and birch using neural networks. The major interest was to make some preliminary evaluations of the applicability of neural networks in this kind of problems.

2. Material and methods

Data used in this analysis was collected from three forestry planning areas in southwestern Finnish Lapland. Total of 249 randomly selected plantations were measured. Number of trees/hectare and number of browsed trees/hectare in each plantation was calculated for 8 tree species or families (pine,birch, aspen, rowan, Salix spp., juniper and Caprea spp.). For building and training the neural network models data from 125 randomly selected plantations were used. Data from 124 plantations were used to test model. For building the model also with classified data, we then classified both datasets in 11 classes (cutpoints; 250, 500, 750, 1000, 1250, 1500, 2000, 3000, 5000, 10000, 30000).

For training the neural network model we used back propagation method (Rumelhart & al. 1988). Back-propagation is a technique for solving the "credit assignment" problem. This means a way to determine the processing element or inter-connections that should be adjusted in case of error. Back-propagation solves this by assuming that all processing elements and connection elements are somewhat to blame for an erroneous response. Responsibility for the error is affixed by propagating the output error backward through the connections to the previous layer. This propagation is repeated until the input layer is reached.

For both classified and unclassified data the input level consisted of 8 processing elements (number of trees/hectare or density classes for 8 species) and the output level of 2 (number of browsed trees/hectare or browsing density classes for pine and birch). The best neural network model was achieved with one hidden layer and two processing elements in it. Models were trained with 15000-100000 random samples. R^2 of the difference of the input and output values were used as a measure when adjusting the model to input data. Models giving the highest R^2 values were used (0.74 for classified and 0.57 for unclassified data). Models were then tested against independent data sets and R^2 values were used as a comparison. Linear regression models were calculated for both classified and unclassified data sets. Models were calculated separately for number of browsed pine and birch/hectare.

3. Results and discussion

For the unclassified data the neural network model could explain 52% of the variation and for the classified data 72.8% . When tested against independent data sets fitness of models were 37% and 77.8% respectively. Linear regression models could explain 25% for unclassified pine browsing data and 27% for classified data. When using browsing of birch as a dependent variable linear regression model could explain 56% for unclassified and 39% for classified data.

When modelling the foraging patterns of moose on heterogeneous food resources several approaches has been used. Belovsky (1978) presented an optimal foraging theory. According to optimal foraging theory animals tend to intake food plants in relation to their digestibility and nutrient value. This means that diet is fitted to available food resources such that animals always prefer ‘best’ food plants available. Resent studies of vertebrate foraging patterns suggest that a mechanistic functional model for such systems must account for the response with respect to both plant attack rate and the biomass consumption rate (see f.ex. Lundberg and Danell 1990, Åström 1990). Löyttyniemi & Piisilä (1983) showed that moose damage in pine plantations was a function of plantation characteristics, surrounding stands and vicinity of culture areas and roads. However, they could explain only about 30% of the variation of the damage.

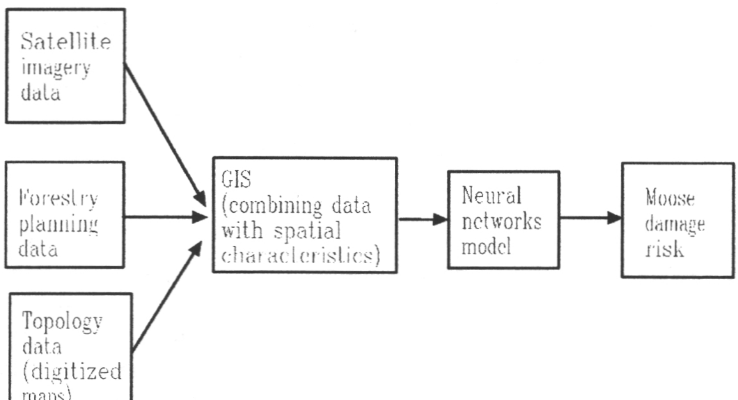


Figure 2. Estimating the risk of moose damage with neural networks as a part of forestry planning process.

Theories and studies like above provide valuable information about behaviour of herbivores like moose, but are difficult to use as such in integrated pest management. There are at least two reasons for this. Associations between variables are nonlinear and discrete. Secondly, theories like functional response theory or optimal foraging theory do not take into account other variables possibly affecting the behaviour of pest or variables used in forest planning.

Neural network model can be used as a part of forest planning system. Satellite imagery data of a certain area can be put in a GIS-system. Stand boundaries created by GIS-system routines or stand boundaries from forest planning maps are added to the image. If we assume moose damage risk as a function of plantation characteristics and spatial data like characteristics of a surrounding stands, this data could be used as a characteristics of a certain plantation. Neural network model could then be build upon this data, assuming that moose damage data of same plantations is also known. In a planning stage imagery data and forestry planning data are combined in the same manner and analysed with neural network model (Fig. 2). Neural network models provide an effective tool when modelling phenomena improperly known or having discrete and non-linear data. In this study, although being preliminary, neural network models were able to make predictions better or at least equal to those given by linear regression models. Because neural network models were build and tested using two variables at the same time, it is somewhat difficult to compare R^2 values directly to those computed from regression models. However, neural networks models can be seen as effective tool in modelling complex systems otherwise difficult to handle. On the other hand, neural networks may be difficult to use to analyze problems or interactions between factors within a problem. Knowledge of the behaviour of factors and their interrelationships is important e.g., in integrated pest management to be able to affect on factors increasing or decreasing the risk of damage.

References

Anderson, J. A., Rosenfeld, E. (eds.) 1988. Neurocomputing: foundations of research. The MIT Press, 1988.

Åström, M. 1990. Foraging time in herbivores: time per prey and day, so the plants may pray. Dissertation. Swedish University of Agricultural Sciences. Department of Wildlife Ecology. Rapport 18: 27pp.

Belovsky, G.E., 1978. Diet optimization in a generalist herbivore: the moose. Theor. Popul.Biol. 14:105-134.

Lundberg, P., Danell, K., 1990. Functional response of browsers: tree exploitation by moose. Oikos 58: 378-384.

Löyttyniemi, K.,Piisilä,N., 1983. Moose (Alces alces) damage in pine plantations in district forestry board Uusimaa-Häme area. Folia Forestalia 553: 23 pp.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. 1988. Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. and the PDP Research Group (Eds.) 1988. Parallel Distributed Processing. Explorations in the microstructure of cognition. Volume 1: Foundations.

Simulation of forest stand development - an object oriented approach

Taneli Kolström[1] and Hannu Salminen
Finnish Forest Research Institute, Rovaniemi Research Station
P.O.Box 16, SF-96301 Rovaniemi, Finland
salminen@roi.metla.fi

Abstract

The paper introduces a stand development simulation system based on the object-oriented approach. The system includes object-class and object-models for presenting individual tree and stand development. Birth, growth, death and seeding models are encapsulated in objects as well as cuttings and various silvicultural operations. The prototype presented uses tree-level distance-dependent development models. Model- and rule-based reasoning is planned to control the system. Object-oriented approach offers a flexible and convenient way to construct ecological models and systems based on them. Presented prototype is unable to show the computing efficiency of planned simulation system until some relevant details have been added to it.

1. Introduction

The aim of this study is to describe the structure and functioning of forest stand; it is a simulation model of the actual system. There are pragmatic reasons which urge to develop an application for research purposes and also for forestry practice (Bossel & Schäfer, 1990). Researches do need a "skeleton" to manage and use growth and development models, and forest management systems needs applications to produce forecasts and predictions of the future development of stands.

The simulation model is based on the development of individual trees and stands. The development consists of birth-, growth-, and death-models, which reflect the competition caused by the environment of trees. Environmental changes caused by various controlling actions like cuttings and other silvicultural operations are also included the model. It is possible to concetrate also on stand-level algorithms, but the developed prototype is constructed to be more detailed. Tree- and stand-level growth models can be used in basic growth actions, as controllers in parallel use, or as a combination, e.g. feeding parameters to each other.

1. Finnish Forest Research Institute, Joensuu Research Station, P.O.Box 68, SF-80101 Joensuu, Finland, taneli@joyl.joensuu.fi

11. Discrete event simulation

Simulation is understood as combining development of algorithms and programs with real or artificially generated data in a computer system. Individual tree-level models may present how the detailed components of growth interact, but there is too much complexity to predict the behaviour of a whole forest area without the help of computer simulation. Simulation offers a possibility to make experiments and test theories as well as predict the development of a stand far into the future.

The development of a stand contains concurrent processes of single trees. So far the parallelism used in the prototype of simulator is rather sequential than concurrent due to the single-processor computers in use. Real concurrent parallelism eventually needs a multi-processor-computer, although object oriented languages offer a way to present parallel processes in a convetional platform.

The basic guideline in building a simulation system of forest development is to imitate the "data structures" and activities of the nature itself. From this point of view, continuous activities should be modelled as well as discrete state transitions. In practice the development of stand is constructed as a parallel discrete event simulation (=PDES, see Fujimoto 1990). Alike the control processes in a real stand (like solar radiation, transportation of water and nutrients), mechanisms needed in parallel continuos simulation are not known well enough. Even if they are known, pragmatic reasons lead to simplify the system.

In the prototype all events are designed to use same timestep. This simplification makes it easier to build a parallel system. If the system is expanded to include processes with different timesteps, interactions should be examined in more detail.

The purpose and the role of the system define the abstraction level, i.e how detailed models are used. If different abstraction levels are used in the same system, it must be done with special care.

12. Simulation based on object-oriented approach supported with reasoning

The philosophy of information systems (IS) science is based on and closely connected to classic philosophy (logic etc.), so the methods and methodologies of IS-science may in some extent be applied in forest research also. The general approach in growth and yield studies and in building information systems can be constructed by the same way. The questions to be presented are; what is the domain we are interested in, what are the 'things' (objects, entities etc.) in it, what are the attributes and behaviour/functions of the 'things', and what are their relations to each other. On the other hand, many methods and techniques in information systems area are established for business applications. In these cases the difference in conceptual backgrounds must be considered (Kaila & Marshall 1991).

Besides the theoretical aspect, tools for building information systems can be used in making forest simulation systems, presenting and describing them, and delivering possible executable applications.

Out of the different methodologies of building information systems, object oriented approach seems suitable for forest modelling purposes because:

1. It emphasizes understanding and modelling rather than task- or process-oriented aims. In any OO-oriented application, objects (classes and instances) do make up most the system. They are the items of interest in each phase (analysis, design, implementation) of building an information system (Korson & McGregor 1990). It forces us to examine the-structure and functioning of the domain, i.e. to understand it.
2. It makes it possible to imitate the "data structures" from nature (Bossel & Schäfer 1990).
3. It is capable to parallel simulation.
4. It has tools for presenting tentative theories without the demand of applicable system (prototype) and it makes cooperation with other modellers and researchers easier. Object oriented programming even allows sharing objects between different simulators. Rapid prototyping is possible because of usage of feasible object-libraries and software reuse (Kaila & Marshall 1991).
5. It uses iterative way of solving problems (prototyping and reuse) (Korson & McGregor 1990). Object-classes are designed as collections of data and the set of allowable operations on that data. That means fewer connections between classes (weak coupling) and good modularity (Korson & McGregor 1990). Due to inheritance and polymorphism it is fairly easy to extend and modify an OO-system.

As reasoned above, object oriented system includes a considerable amount of information about the domain. Thus it can be considered as a description of the structure and behaviour. According to Tsatsoulis (1991), "Message passing between objects does not necessitate any reasoning. In themselves, OO-techniques do not add any intelligence to the simulation environment, and they must be supported by reasoning system". On the other hand, Bobrow and Stefik (1986) defined, that "an intelligent system should embody and apply information about itself so that it can assist in its own continuing development". In that meaning, object oriented systems are "intelligent" even without any reasoning. Anyhow, reasoning can improve systems behaviour and add the intelligence in a convenient way. In the simulation of forest development, reasoning is needed at least when simulating human decisions and constructing some control actions (like cuttings and silvicultural operations).

2. Simulation of the development of a forest stand

21. Methodology and tools

There are several suggestions how to build object-oriented information systems, but there is no methodology and no 'tools' have yet gained universal acceptance (Boehm 1988, Shlaer & Mellor 1988, Bailin 1990, Coad & Yourdon 1990, Gibson 1990, Henderson-Sellers & Edwards 1990, Iivari 1990, 1991, Korson & McGregor 1990, Booch 1991). Tools are here understood as defining working processes, guiding rules and supporting software. Thmethod presented by Booch (1991) is a good candidate for standard 'tool' of object-oriented design. We followed the spiral-model of Iivari (1990, 1991) and at different levels of modelling the modifications presented by Kaila & Marshall (1991). Identifying object-classes was done with the help of guidelines described by Korson & McGregor (1990).

Since object-oriented analysis (OOA), object-oriented design (OOD), and implementation(object-oriented programmming, OOP) have no district boundaries, a CASE-tool should be integrated into OOP. This type of software was not available. We are convinced that designing a system with the help of different kind of diagrams (see Booch 1991) is useful, but it must not be separated from the implementation. Prototype and actual analysis, design and implementation of the system was done with KAPPA^{TM} Application Development System version 1.1.

22. Description of simulation system

221. Organizational level

Organizational level defines the organizational role and context of the information system (Iivari 1991). Kaila and Marshall (1991) divided the software applications produced in scientific research into three categories; "1) Application software developed as research tools, 2) information systems and application software developed for use within the research community, and 3) information systems and application software developed to provide solutions to applied forestry problems". Kaila and Marshall (1991) underline that organizational needs are more extensive in the third category than in the first two categories.

So far the planned simulation system is used mainly for reasearch purposes and has a minor strategical position among the information systems of the Finnish Forest Research Institute when considered as a standalone system. When integrated into decision-making applications it might have greater responsibility (Saarenmaa et al. 1990). The final aim is to produce applications for forestry practice. Only then a simulation system can be considered as a part of the corporate information system.

222. Conceptual/infological level

Conceptual/infological level defines an "implementation independent" specification for the information system (Iivari 1991). In this level the "vision" of forest development simulation is presented. It should be also remembered that this vision will change during the development process of the system.

Simulator is planned to include reference-database for different growth models, each of which can be chosen according to the rules controlling the model-base. In the prototype each stand is simulated separately, but other approaches are possible as well. If border-stands are considered without simplifications or modifications (like mirroring) stands cannot be treated as independent.

In each stand there are trees with location, age, species and different types of dimensions. Factors affecting to the growth will be encapsulated into an abstract object presenting the local environment of each tree. Different actions like cuttings, regenerations, and other silvicultural operations are controlled by a decision-maker called 'forester'. The simulation system is not planned to decide, which treatment schedule is the best, although it can be connected to some decision-support systems.

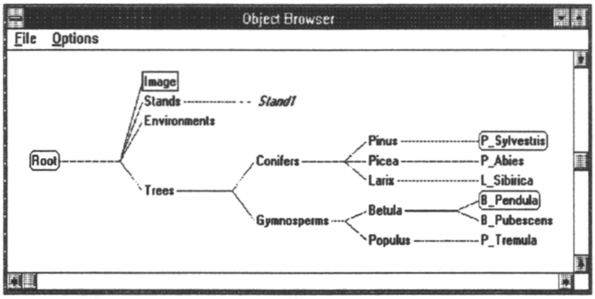


Figure 1. Entity-objects.

Input data can be either tree-level or stand-level. In the latter case, tree-level data will be generated through distribution functions and stand-level growth models will control the simulated development. The main output of simulator is planned to be a set of possible development predictions for each stand.

Objects of a system can be divided into five categories: entities, events, inputs, outputs, queries and interfaces (Iivari 1991). The entities of the simulation system are the classes: forest areas, stands, trees, seeds and local environments (Figure 1) The events-category consists of various controlling actions divided into subclasses such as cuttings, regenerations, fertilizations, ditchings, cleanings, prunings, and site preparations (Figure 2). The inputs-category has not yet been constructed. The data import module will base on the input's object-class as well as data output bases on the 'output's' object-class. All database operations use different queries clarified in objects belonging to the queries-class.

Tree-objects are further devided according to main species. In the prototype parent-object 'Trees' has attributes 'age', 'alive', 'basal area', 'height', 'breast height diameter', 'environment', and 'location'. There are three biological methods in trees-object; 'aging', 'dying' and 'growing'. In addition there is a method to describe the structure of a tree ('calculate dimensions'). The method 'draw a tree' is used in the mapping of trees in the interface.

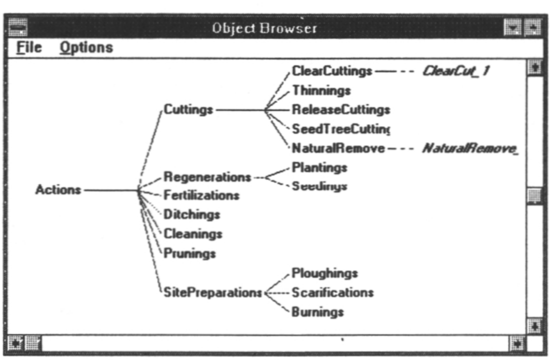


Figure 2. Event-objects.

In the prototype the only actions constructed were cuttings. Parent-object 'cuttings' has tree attributes; 'removal percentage', 'remove basal area', and 'removed basal area'. In the prototype removal is presented as a proportion ('removal percentage') of the total basal area. It could as well base on density, location of trees, or another silvicultural fact. The three attributes are manipulated by three methods; 'cut a stand' receives a message from forester-object and sends messages to methods 'select trees to be cut' and 'cut selected trees', which actually are responsible for accessing instances in 'Trees'-object-class. In the prototype trees to be cut are selected on a random-basis. Because all trees have location and some relevant attributes, more intelligent cutting methods can easily be constructed.

Input, output, query, and interface objects have more technical purposes than entities and events. They are not presented detailed in this paper.

223. Datalogical /technical and implementation level

Datalogical/technical level defines the technical implementation for the information system (Iivari 1991). Prototype was build with KAPPA^{TM} Application Development System version 1.1(1). It can use dBase-files(2) and C-libraries and dynamic data exchange in MS-Windows-environment (3). Pascal- and Fortran subroutines can be called through C-modules.

3. Discussion

Object-oriented approach offers a flexible way to construct simulation systems. Entities representing "real world" and necessary abstract objects (like stands, environments) can be described after some simplifications. Data structures imitated directly from the domain makes future expansion possible. Various kinds of cuttings and silvicultural operations can be simulated in a very detailed manner if wanted. The simulation system can accept stand- or tree-level development models including distance dependent spatial models. Ecophysiological process-models will require more detailed object-structure, but it seems also possible to connect them into the system.

Various tools could help OO-work especially in analysis and design phases, but they should be integrated with the programming module. So far no CASE-software with integrated OO-support is available.

The prototype represents an ongoing work. A reference data-base for growth models is under work. C-, Fortran-, and Pascal-libraries will also be collected. The efficiency and the simulation speed of the system are correlated to the chosen abstraction level. They can be speeded up by using C-libraries instead of Kappa's KAL-functions. Efficiency can be obtained also by using connections to servers or distributing processes into more powerful workstations.

References

Bailin, S.C. 1989. An object-oriented requirements specification method. Communications of the ACM, Vol. 32, No. 5.

Boehm, B.W. 1988. A Spiral Model of Software Development and Enhancement, Computer. May 1988.

Booch, G. 1991. Object oriented Design with Applications. The Benjamin/Cummings Publishing Company. 580 p.

Bossel, H. & Schäfer, H. 1990. New directions in forest modelling: process dynamics and integrated knowledge processing. Proceedings of XIX IUFRO World Congress, Montreal, August 5-11, 1990, Div. 6:95-104.

Coad, P. & Yourdon, E. 1990. Object Oriented Analysis. Yourdon Press, Prentice Hall.

Fujimoto, R.M. 1990. Parallel Discrete Event Simulation. Communications of the ACM. October 1990. Vol.33, No. 10. p.30-53.

Gibson, E. 1990. Objects - Born and Bred. Byte, October 1990. p. 245-254.

Henderson-Edwards, B. & Sellers, J.M. 1990. The Object-Oriented Systems Life Cycle. Communications of the ACM. October 1990. Vol.33, No. 10. p.142-159.

Iivari, J. 1990 Hierarchical Spiral Model For Information System and Software Development. Part 2.: design process. Information and Software Technology, Vol. 32, No.7.

Iivari, J. 1991. Object-oriented Information Systems Analysis. A framework for object identification. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on Systems Sciences (ed. Bruce D. Shriver). Vol II. Software Technology track. IEEE Computer Society Press. p. 205-218.

Kaila, E. & Marshall, L. 1991. Passing Information Systems Across the Demarcation Line Between Forest Research and Forestry Practice. Accounting, Management and Information Technologies (submitted).

Korson, T. & McGregor, J.D. 1990. Understanding Object-oriented: A Unifying Paradigm. Communications of the ACM. September 1990. Vol.33, No. 9. p.40-60.

Saarenmaa, H., Lehto, K., Kaila, E., Salminen, H. & Pöntinen, J. (eds.). 1990. Metsäntutkimuslaitoksen tietojärjestelmästrategia, Työryhmän mietintö. Metsäntutkimuslaitoksen tiedonantoja 350, 82 pp.

Shlaer, S. & Mellor, S. 1988. Object-Oriented Systems Analysis. Modelling the World in Data. Yourdon Press, Prentice Hall. 144 p.

Tsatsoulis, C. 1991. A Review of Artificial Intelligence in Simulation. Sigart Bulletin. Vol.2 No 1. p. 115-121.

The design of a data management system for research measurements

Olavi Kurttio, Ari Talkkari[1] *and Ari Turkia*[1]
Finnish Forest Research Institute, Department of Forest Genetics
P.O.Box 18, SF-01301 Vantaa, Finland

Abstract

In this paper a description of a data management model is given. The database based on the model presented here can handle data from any type of field or laboratory measurements. A small part of the Finnish Forest Research Institute's data model was used as a starting point of this work. The data model consists of 17 normalized entities and their relationships. Performance will probably be the most difficult point in the practical database application. This is due to large number of keys compared to actual data values. Anyhow, irrelevant key values are avoided. Instead, most keys have a reasonable meaning for researcher. To be useful, this method will require very careful planning of measurement activities.

1. Introduction

Research and measurement data management is an important part of the activities of every research organization. Usually there have been no uniform methods or tools to store and manage different kinds of data from various sources. Instead, data sets are too often stored with shortsighted methods. This may later cause difficulties in data management and use. When valuable data is stored in a database, users can easily retrieve suitable parts of it for further processing. The main advantage of a well-designed relational database is in data management, which can be handled in a consistent manner, without a need to operate with several files of different formats. Database users can either have a copy of the data they need, or they can create views into the actual databases. Standard methods of data storage and retrieval provided by properly designed relational databases are necessary for long-term data utilization. Measured data stored in a database is no longer disposable.

In this paper a description of a formal model for managing data from any type of field or laboratory measurement is given. A schematic representation of the placement of measurement data management system in a general research organization is shown in Figure 1. At first the contradiction of planning a system that is simultaneously general and specific seemed to be an almost impossible task from philosophical point of view. In any case, using information engineering methodology and relational database concepts this system was possible to describe.

There exists also another discrepancy in the present work; the approach to data modelling

1. University of Joensuu, Department of Forestry, P.O. Box 111, SF-80101 Joensuu, Finland

became - partly unintentionally - somewhat object-oriented, but the application will be build using relational database technique, which in principle is not suitable for strictly object-oriented approach.

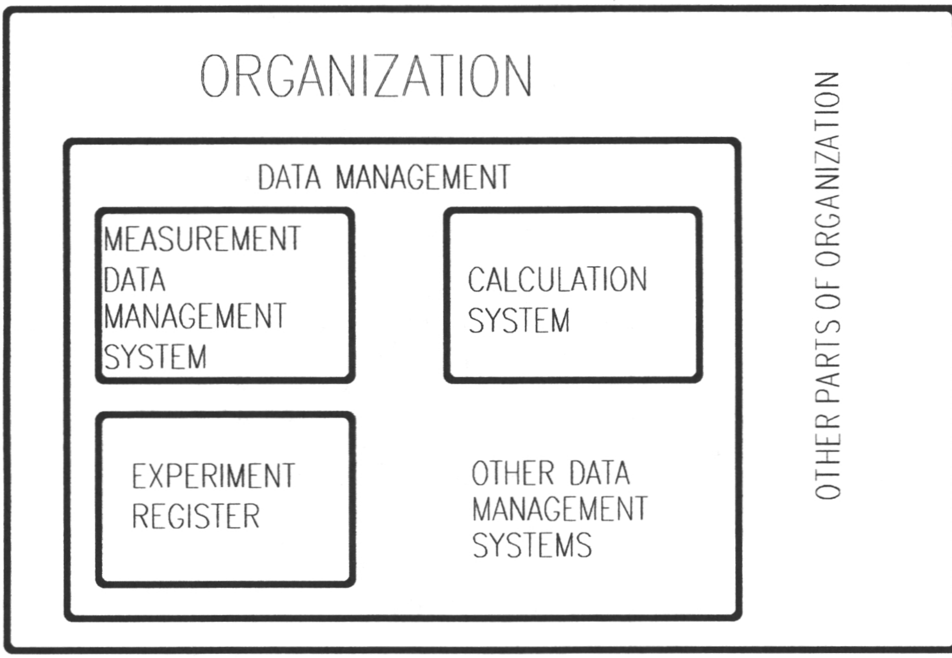


Figure 1. The relationships of the measurement data management system to some other databases in a research organization.

2. Material and Methods

The initial framework of the work was a small part of the Finnish Forest Research Institute's (FFRI) data model, which contains some 330 entities (Saarenmaa & al. 1990). That model was developed using a methodology called SRM ("Systeemien Rakentamismenetelmä"), which is a quite close relative to James Martin's information engineering methodology (IEM) (e.g. Mikkonen & Soini 1988, Martin 1989, Roukala & Soini 1989).

In the present work, only the entity diagram technique of SRM was applied in the development of data model. Other steps in data modelling were not used, because they were carried out extensively in FFRI's work. The aim of our work was to develop this part of FFRI's model into a more general and suitable one, that could be easily transformed into a real relational database system.

The data model consists of entities, their relationships, and attributes. Usually entities can be considered as tables in relational database, and relationships show table connections. Attributes represent fields in a database table. In SRM methodology, entities can be described as concrete or abstract objects, of which the organization needs or obtains some

information. All types of entities have some common features:
- an entity may have several properties,
- an entity must have a identifying key,
- entities are defined to the third normal form.

Entities can be classified into three main types; characteristic, associative and kernel entities. A characteristic entity describes, classifies or clarifies some other entity (kernel, see below), and it represents some repeated characteristic of the kernel. It does not exist alone. An associative entity represents many-to-many relationships. It can have characteristic entities attached. If an entity is neither characteristic nor associative, is considered as a kernel entity, which is independent.

Basic types of relationships are one-to-one (1:1), one-to-many (1:M), many-to-many (M:M), and their conditional (C) forms. Totally there are nine different types, although some are quite seldom used. Many-to-many relationships can not occur in a normalized database. They are removed by creating an associative entity.

The CASE (Computer Aided System Engineering) tool used in this work was System Architect by Popkin Software. Only the entity-relationship component of System Architect was used in modelling work.

3. Results

The main result is, of course, the data model or entity-relationship diagram, which is presented in Figure 2. The interpretation of the model is quite laborious and can not be done extensively in this paper. The attributes of entities are presented in Appendix 1. Data normalization was carried out extensively; entities are in third normal form. Due to this, many of the entities consists of key items.

Basic aims in building the data model was data retrieval with relevant key fields and values. That caused additional complexity in the model, because many key fields should contain only appropriate data, and the database program should not be allowed to give arbitrary values in such cases.

4. Discussion

The data model presented in this work differs from the part of FFRI model that was used as the starting point because of a more data-oriented approach. The naming convention used in the present data model should not be confused with that of object-oriented design. For example, our objects should be considered as logical ones, that are measured, instead of objects in pure o-o sense.

We "hand-tested" the model using some data from ordinary forest management measurements and some artificial measurements, but the working database prototype is under development. The database for practical use will be build after the prototype has been thoroughly tested during spring 1991.

It is probable that the presented model is not very efficient, but performance was not the

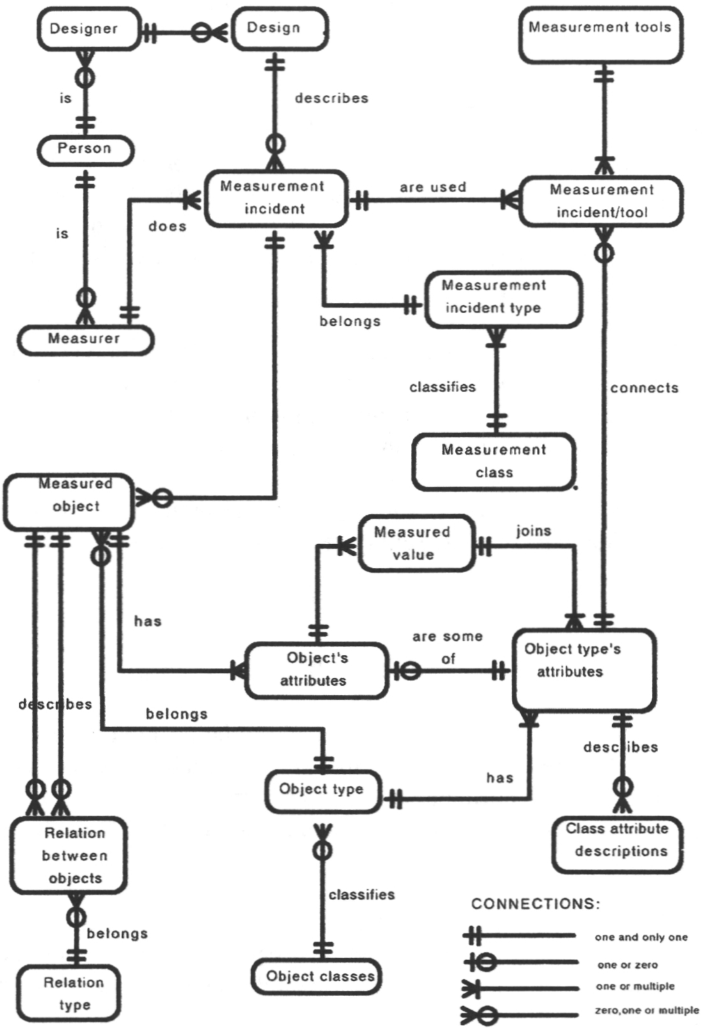


Figure 2. A data model for measurement data management

major concern. More important aspects were generality and robustness. Besides, a poorly normalized database (with few large tables) may seem efficient from file- oriented point of view, but in practice such a construction is not much better in this respect. Furthermore, management of such a database will become increasingly difficult after some time.

Some entities in the model are obviously very important, e.g. "Measurement incident", "Object type" and "Measured object". The essential table in the model contains the measured value and all the necessary keys for its identification (Table 1).

The first three rows of this hypothetical relational table (Table 1; "MEASURED VALUE") are measured in the same measurement incident (e.g. one field group during one day). First two are measurements of same object type (e.g. relascope plot), and the third one is e.g. a tree in the plot. First two are different objects ("plots"), but the measured attribute is the same (e.g. stand type). As can be seen from Table 1, the model can accept all types of data, character strings as well as numeric values. This table contains only the measured value; it's interpretation - what does this value mean - is in another table in the database (Object type's attributes).

The model was not thoroughly tested because of the short time frame. The data model may have to be modified after the real database prototype is tested. Most likely this might be due to performance reasons, which may force us to de-normalize some parts slightly. The number of actual data values compared to large number of key values needed may seem too small, but most of key values must be determined and placed in the database before data input phase.

The model managed well to handle various types of data, but there are still many problems in the conceptual side of measurement object classification. These problems could be solved only by the researcher, who is responsible of planning the experiment or measurement. Our model can give the platform to use any classification of measurement objects.

Table 1. An example of table with some possible values (see explanation below)

MEASURED VALUE

Measurement Incident Number *	Object Type Number *	Measured Object Number *	Attribute Number *	Measured Value
18	10	11	3	3
18	10	12	3	5
18	11	1	2	220
88	15	70	10	30
89	15	70	10	27
77	23	345	3	John Smith

* = Key field

References

Martin, J. 1989. Information engineering, I. Introduction. Prentice Hall, Englewood Cliffs, New Jersey

Mikkonen, M. & Soini, T. 1988. Yrityksen tietosuunnittelu. 344 pp, Weilin+Göös, Espoo.

Saarenmaa, H., Lehto, K., Kaila, E., Salminen, H. & Pöntinen, J. (eds.) 1990. Metsäntutkimuslaitoksen tietojärjestelmästrategia. Työryhmän mietintö. Metsäntutkimuslaitoksen tiedonantoja 350, 82 pp.

Soini, T. 1985. Tietoanalyysi. 269 pp. Weilin+Göös, Espoo

Appendix 1. Contents of the entities in the data model

CLASS ATTRIBUTE DESCRIPTIONS	
ObjectTypeNumber	Int
AttributeNumber	Int
MeasuredValue	Char(40)
ValueDescription	Char(40)
COMPUTABLE OBJECT VALUES	
ComputableAttributeNumber	Int
ComputableAttributeName	Char(40)
DESIGN	
DesignNumber	Int
DesignerNumber	Int
Description	Char(400)
DESIGNER	
DesignerNumber	Int
PersonNumber	Int
MEASURED OBJECT	
MeasuredIncidentNumber	Int
ObjectTypeNumber	Int
MeasuredObjectNumber	Int
MEASURED VALUE	
MeasurementIncidentNumber	Int
ObjectTypeNumber	Int
MeasuredObjectNumber	Int
MeasuredValue	Char(40)
MEASUREMENT CLASS	
MeasurementClassNumber	Int
MeasurementClassName	Char(40)
MEASUREMENT INCIDENT	
MeasurementTypeNumber	Int
MeasurementIncidentNumber	Int
MeasurementYear	Smallint
MeasurementMonth	Smallint
MeasurementDay	Smallint
MeasurementTime	Char(10)
MeasurerNumber	Int
DesignNumber	Int
MEASUREMENT INCIDENT TYPE	
MeasurementIncidentTypenumber	Int
MeasurementIncidentTypeName	Char(40)
MeasurementClassNumber	Int
MEASUREMENT INCIDENT/TOOL	
MeasurementTypeNumber	Int
MeasurementIncidentNumber	Int
MeasurementToolNumber	Int
ObjectTypeNumber	Int
AttributeNumber	Int
MEASUREMENT TOOLS	
MeasurementToolNumber	Int
MeasurementToolName	Char(40)
MEASURER	
MeasurerNumber	Int
PersonNumber	Int
OBJECT'S ATTRIBUTES	
MeasurementIncidentNumber	Int
MeasuredObjectNumber	Int

ObjectTypeNumber	Int
AttributeNumber	Int
OBJECT CLASSES	
ObjectClassNumber	Int
ObjectClassName	Char(40)
OBJECT TYPE ObjectTypeNumber	Int
ObjectTypeName	Char(40)
ObjectClassNumber	Int
OBJECT TYPE'S ATTRIBUTES	
ObjectTypeNumber	Int
AttributeNumber	Int
AttributeName	Char(40)
AttributeUnit	Char(10)
PERSON PersonNumber	Int
PersonFamilyName	Char(40)
PersonGivenName	Char(20)
RELATION BETWEEN OBJECTS	
MeasurementIncidentNumber1	Int
ObjectTypeNumber1	Int
ObjectNumber1	Int
MeasurementIncidentNumber2	Int
ObjectTypeNumber2	Int
ObjectNumber2	Int

Rule-based reasoning in the selection of forest regeneration method

Taneli Kolström and Jarmo Ahonen[1]
The Finnish Forest Research Institute, Joensuu Research Station
P.O.Box 68, SF-80101 Joensuu, Finland
taneli@joyl.joensuu.fi

Abstract

This paper introduces a rule-based expert system for the selection of forest regeneration methods. The rules are based on the guidebook of silviculture of the company Enso-Gutzeit Oy (1988). The system consists of 300 rules written in Lisp, an inference engine (Backtrack1 algorithm by Nilsson 1980), and an interface build in KEE. The presented system is in prototype phase. This type of a system handles information more systematically than a human decision-maker. By this way in the decision-making one can reduce errors and failures.

1. Introduction

Silvicultural decision-making is quite errorprone and complex area to operate. This is partly due to the fact that most silvicultural decisions do not implement the best possible solutions and results to the problems we are working on. In long term this causes remarkable economical losses. For example, in 1988 the number of seedlings used for complementary planting was 27 million, i.e., 12 % of the total number of seedlings used for plantings in Finland (Aarne & al. 1990).

Guidebooks have been made to decrease the faultness of decisions (see e.g. Enso-Gutzeit 1988). On the other hand they have decreased failures in decisions but on the other hand they have made the decision-making more monotous than it was earlier. For example, in artificial forest regeneration certain regeneration chains are dominating. The need for more advanced decision support systems is obvious.

Human decision-making is subjective. It can not take very many facts into the consideration. Computer-assisted decision-making has advances that improve human decision-making in that respect. Computers are capable to handle unlimited amount of information, to pick out the essential facts from vast amounts of information, and in best cases able to solve problems by themselves by using knowledge-based models of the area (Kaila & Saarenmaa 1990).

---

1. University of Joensuu, Department of Computer Science, P.O.Box 111, SF-80101 Joensuu, Finland, jahonen@ujocs.joensuu.fi

The methods offered by artificial intelligence (AI) have been recently applied in decision-support systems, especially in expert systems. Rauscher and Hacker (1989) list 74 research projects concerning AI applications in natural resource management. Those projects spread widely over the whole sector of natural resource management. Mansur (1989) has made an expert system of artificial regeneration in Mozambique. Rice et al. (1989) have constructed a rule-based system for species selection in artificial regeneration.

The aim of this study is to make a prototype of expert system that is able to help in the selection of forest regeneration methods. The system is based on rule-based reasoning.

2. Methods

21. Forest regeneration

The forest regeneration can be described as a chain of decisions following each other (Parviainen & Lappi 1983). The realization of a silvicultural decision support system requires formulation of the decision-making process into a programmable form.

Traditionally there has been separated two main lines in forest regeneration. The basic selection is done first between natural regeneration and artificial regeneration. Another main issue is the selection of a tree species for next tree generation.

There is two main issues that define the result of regeneration (Parviainen & Lappi 1983): i) The conditions in the regeneration area and nature define the constraints and possibilities. For example, to these belong site, previous tree generation, and stoniness. These conditions and limits restrict the regeneration area. They are influenced by changes in the environmental conditions. ii) The actions made by decision maker such as soil preparation, regeneration cutting, and cleaning of sapling stand. They can be chosen quite freely but they can cost.

The aimed decision support system of forest regeneration requires that the usual decision-making can be modified in form of rules. The guidebooks describe usually forest regeneration problem almost as rules (Enso-Gutzeit Oy 1988). In this work the guidebook of silviculture of company Enso-Gutzeit Oy (1988) was used as an aid in generating rules.

22. The computer model of decision-making

The guidance for the regeneration problem presented by Enso-Gutzeit Oy (1988) was similar to the rule model of human problem solving as presented by Newell and Simon (1972).

Rules were written as knowledge structures of lisp as follows:

```
(d-rule fertile-peatland ()
 (and (is-soil-type 'peat)
 (is-fertile))
(include-asked-data 'Y 'fertile.peatland),
```

where the (and (...) (...)) is the condition part of the rule and (include-asked-data ...) is the consequence part. Especially the inconsistencies of rules were checked. Inconsistencies can be defined as conflict, redundancy and subsumption (Buchanan & Shortliffe 1985).

Conflict is a situation where two rules are succeeded in the same identical database. The results of these two rules are conflicting in a way that is in inconsistency. For example the rules

```
IF fruit is yellow THEN fruit is grape
 and
IF fruit is yellow THEN fruit is citron
```

are conflicting to each other. To solve the conflict we focused conditions of the rules until the conflict disappear.

Redundancy is a situation where the same rule or rules giving a same functional result are present more than once in the knowledge base. Redundancy is not risky for the reasonability of the knowledge base, though it can cause remarkable uneffectiness in praxis. Due to the criteria of effectivity we tried to exclude redundancy.

Subsumption is third situation of inconsistency. In subsumption another rule of a pair succeeds more easily than the another although the consequences of both are same. Rules have often subsumptions, if conditions are confused or inadequate. For example, the legendary rules generated from the penguin problem (Kimbrough & Adams 1988)

```
IF Tweetie is a bird THEN Tweetie can fly
 and
IF Tweetie is a bird AND Tweetie is not a penguin
THEN Tweetie can fly
```

are in relation of subsumption. To exclude subsumptions the rules itself and their condition parts were defined as simple as possible and their field of activities limited.

A rule-based knowledge system operates mainly as follows (Barr & Feigenbaum 1981):

1) Find all rules whose condition parts are TRUE and make them applicable.
2) If more than one rule is applicable, then deactivate any rule whose action adds a duplicate symbol to the context list.
3) Execute the action of the rule. If no rule is applicable, then quit.
4) Reset the applicability of all rules and return to step 1.

The used algorithm was a forward-chaining version of BACKTRACK1 algorithm by Nilsson (1980). In forest regeneration the initial situation is known but the final situation is unknown. The use of backward chaining means that one has to select a hypothesis and then try to prove it true. Another reason for the selected method was the easy implementation. In implementation we added a system of explanation, which tells to the user the basis of reasoning and proposed choice.

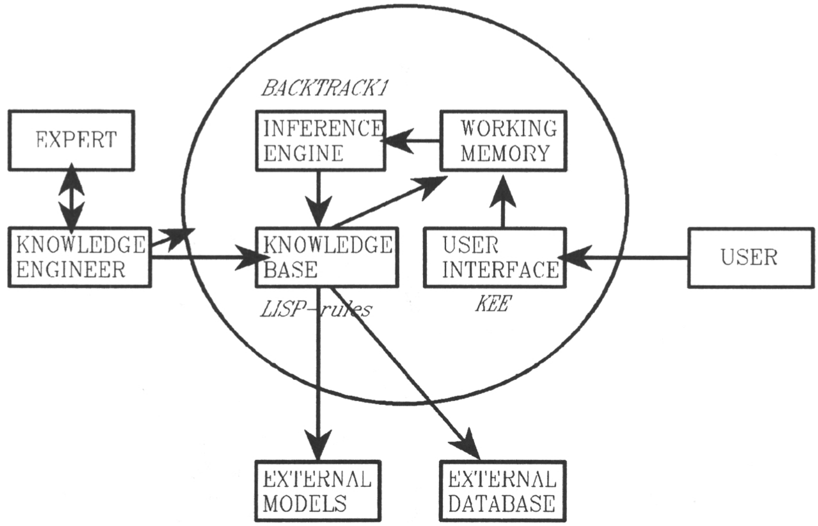


Figure 1. Structure and fuctioning of an expert system (Saarenmaa 1989). In the schematic the parts mentioned in the text are typed italics.

23. Construction of the system

The main system was built by using LISP and the interface was done by using KEE. The work was done in a SUN 4/110 workstation. Figure 1 summarizes the components of the expert system. Totally the knowledge-base consists of 300 rules written in Lisp.

The implemented interface is in Figure 2. The basic information of a regenerated area is selectable in the interface and the results of the reasoning are shown. During the reasoning the system may ask from the user more specific details in dialog window if needed. The reasoning algorithm is implemented directly as a LISP-function, which is activated by an active-image option of KEE. The desired values are moved from KEE to an independent object. Next step is the activating of reasoning. After the reasoning the objects of KEE are updated and we are back in an interactive state.

After the reasoning user can look the explanations of the selected method in a dialog window. User can freely follow the matched rules from the results back to the initial situation.

3. Results

The validation of an expert system involves the comparison of system recommendations with known input data and results from case studies. We compared the results of the system with choices of some experts and found out some charasteristics of system functioning:
i) Choices made by ES were reasonable compared to those of experts.
ii) The functioning of ES demands the user go through all the details affecting the case.

ES could also recommend unused solutions like natural regeneration of birch. Natural regeneration of birch is not directly mentioned in instructions of Enso-Gutzeit Oy (1988).

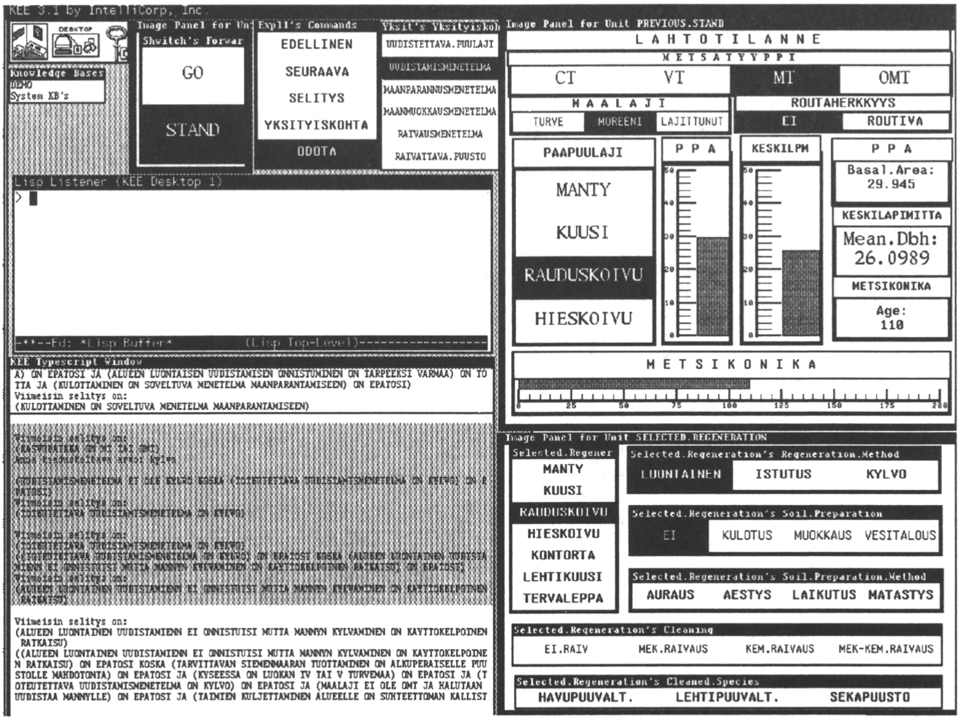


Figure 2. The interface of implemented system.

On the other hand it is not restricted out of choices.

We expected the time and space demands of the algorithm to grow exponentially with the growth of search space, but this did not happen. The implemented algorithm did not need to backtrack to overstate the changes in knowledge-base made by rules. The situation did not change although we made a physical reorganization of rules. The polynomical reason of time demand remains open.

4. Discussion

This expert system is a trial to use rule-based reasoning in selection of forest regeneration method. The used ‘data’ was the guidebook of silviculture of Enso-Gutzeit Oy (1988). This means that the ‘data’ was more or less heuristic rules (rules of thumb) that combine both results of research and experience from the field. Compared with the traditional guidebooks we believe that this type of expert system does not bring any new information. But it does help in using this information more systematically than a guidebook does (Waterman 1985). By this way it improves decisions in the selection of forest regeneration method.

This expert system is a prototype and it is not aimed to use in everyday forestry yet. Perhaps the best area to use this type of expert system in everyday forestry is silvicultural planning. Usually the planned actions like cutting and planting are stored in knowledge-base. This type of expert system may act as a background daemon in storing planned actions to check the reasonability of actions. By this way it can eliminate the unreasonable regeneration methods from becoming implemented.

References

Aarne, M., Uusitalo, A.M. & Herrala-Ylinen, H. (eds.). 1990. Metsätilastollinen vuosikirja 1989. Yearbook of forest statistics 1989. SVT Maa- ja metsätalous 1989:1. Folia Forestalia 760:1-246.

Barr, A. & Feigenbaum, E.A. (Eds.). 1981. The Handbook of Artificial Intelligence, Vol. I. William Kaufmann Inc., Los Altos. 409 pp.

Buchanan, B.G. & Shortliffe, E.H. (Eds.) 1985. Rule-Based Expert Systems. Addison-Wesley Publishing Company, Massaschusetts. 748 pp.

Enso-Gutzeit Oy. 1988. Metsienhoito-ohje. Guide for silviculture. (in Finnish). Enso-Gutzeit Oy, metsätoimiala.

Kaila, E. & Saarenmaa, H. 1990. Tietokoneavusteinen päätöksenteko metsätaloudessa. Summary: Computer-aided decision-making in Forestry. Folia Forestalia 757, 34 pp.

Kimbrough, S.O. & Adams, F. 1988. Why Nonmonotonic Logic, Decision Support Systems 4.4:111-127.

Mansur, E. 1989. Selecção de Espécies para Reflorestamento em Moçambique, com auxilio de computador. FAO MOZ/86/003.

Newell, A. & Simon, H.A. 1972. Human Problem Solving. Prentice-Hall, New Jersey. 736 pp.

Nilsson, N.J. 1980. Principles of Artificial Intelligence. Tioga Publishing Company, Palo Alto. 476 pp.

Parviainen, J. & Lappi, J. 1983. Laskentamalli metsänviljelyketjujen vertailemiseksi. Summary: A calculation model for the comparison of artificial regeneration chains. Folia Forestalia 549:1-24.

Rauscher, H.M. & Hacker, R. 1989. Overview of AI Applications in Natural Resource Management. A paper presented at the Artificial Intelligence and Growth Models for Forest Management Conference, Vienna, Austria, September 18-22, 1989.

Rice, D.A., Nix, J.N. & Gellespie, A.J.R. 1989. A simple rule-based sustem for species selection in artificial regeneration. AI Applications in Natural Resource Management 3(1):38-43.

Saarenmaa, H. 1989. Artificial intelligence concepts, techniques, and applications in forestry. Proceedings of Artificial intelligence and growth models for forest management decisions. Vienna, Austria, September 18-22, 1989. pp. 190-201.

Waterman, D.A. 1985. A Guide to Expert Systems. Addison Wesley Publishing Company, Massachusetts. 419 pp.

Utilizing relational database management system in timber sale planning

Tuula A. Nuutinen
Department of Geography, University of Edinburgh
Drummond Street, Edinburgh EH8 9XP, Scotland, U.K.
tel. +44-31-650 2541
tan@geovax.ed.ac.uk

Abstract

This paper will describe a prototype of timber sale planning system designed to evaluate a relational database management system (RDBMS). The evaluation was an initial step of the project "The Research Forest Database and Planning System". The project aims to develop an integrated decision support system (DSS) for the Finnish Forest Research Institute (FFRI). The purpose of the evaluation was to find out the critical features of the RDBMS in an integrated software environment. For example, the integrated DSS may need some advanced resource modeling operations (such as growth simulation) which can be implemented only using a procedural high-level language. The evaluation process consisted of three parts: (1) the development of a research forest database for one research area, (2) the development of an interface for the forest simulator (MELA), and (3) the development of an "SQL-calculator" for timber sale planning. The database was developed on VAX/VMS using the ORACLE software. The interface for MELA was programmed using Embedded SQL. The "calculator" was programmed using interactive SQL. The evaluation indicates that an RDBMS kernel together with a query language interface provides an "easy-to-use" toolbox for the end-user. The Embedded SQL interface allows programmers to access data in the database for their own applications. For advanced site-specific analysis an operational interface between an RDBMS and a GIS is recommended.

1. Introduction

1.1 Background

The Finnish Forest Research Institute (FFRI) has some 140 000 hectares of state forest at its disposal. 80 000 hectares are commercially managed forests. On these there are 20 000 experimental plots. The total forest area is divided into so called research areas for management purposes. A research area consists of one or several farms. The size of research areas varies from 590 hectares to 14,127 hectares. The forest plan of research area is made every ten years in Southern Finland and every 20 years in Northern Finland. The plan is based on forest inventory using aerial photographs and field survey. The inventory stands and experiments are delineated on the map. The stand characteristics are measured and the treatment proposals are made in the field.

The resulted forest plan guides annual short-term planning (e.g. harvesting scheduling and timber sale planning) in research areas. The problem of short-term planning is how to take the frequent changes in land-use and the continuous change of the forest state into account in the plan implementation. The current system has only limited amount of routines for monitoring stands reserved for experiments or stands leased from experiments after field inventory. However, the departments which have experiments under or near planned treatments should be informed about harvesting in case the treatments should violate the experiments. On the other hand, there are no routines to "manually" update the changes after treatments or "automatically" update the growth of trees in the stand database. A corporate database integrated with planning applications would provide both researchers and research areas with better management tools.

1.2 Relational database management system technology

Traditional planning tools are a forest plan and a calculator. The modern software tools for application development such as relational database management systems (RDBMS) allow end users to create and maintain their own plan "calculators".

For example, ORACLE includes an RDBMS kernel plus a complete set of integrated software productivity tools. Tools include e.g. a screen forms manager, a report writer, an application generator and a query language and a programming language interface. A standard language, SQL (Structured Query Language), is used to interact with databases. SQL is an English-like language which includes commands for query (retrieving data from the database), data manipulation (inserting, updating and deleting data in the database), data definition (adding new tables to the database), and data control (preventing access to private data in the database). The main advantages of SQL are: a simple data structure ("table"), powerful operations (e.g. relational join), reduced training costs (English-like language) and application portability (e.g. applications for minicomputer end-users can be developed on PC).

There are, however, some limitations in SQL: it is non-procedural (i.e. most statements are executed independently of preceding or following statements) and there are several restrictions for "built-in functions" (e.g. SUM). Because of the limitations the constructs such as "loops" or "if/then" pairs have to be implemented using high-level programming languages. SQL statements can then be combined (embedded) with a program written e.g. in FORTRAN (Date 1989).

1.3 The aim of the study

The Finnish Forest Research Institute (FFRI) started a project "The Research Forest Database and Planning System" in the summer 1990. The initial plan was to develop an integrated decision support system (DSS) based on a corporate database (an integrated stand and experiment database) and a forest simulator (MELA). The plan was derived from the traditional view that an organization should have a tabular resource database which is compatible with existing business management functions. An integrated DSS may, however, need also resource modeling operations such as treewise growth simulation for database update or site-specific queries for the analysis of the long-term impacts of treatments. These modeling operations can often be implemented only using a procedural high-level language or a specific "toolbox" such as a GIS (Geographical Information System).

Because "anything worth doing is worth doing poorly until you learn to do it well", the aim of this evaluation study was to test the capabilities of relational database management system (RDBMS) as a data management and planning tool. The purpose of this paper is to describe the critical features of the RDBMS in an integrated software environment. The application of timber sale planning was chosen for the evaluation because it contains the characteristic features for an integrated decision support system: a corporate database integrated with simulation routines to model complex relationships.

The evaluation process contains three parts: (1) the development of a research forest database for one research area, (2) the development of an interface for the forest simulator (MELA), and (3) the development of an "SQL-calculator" for timber sale planning.

2. The data model

The data model of the study (Figure 1) was based on the current standwise forest inventory system and the practice of delivery timber sale.

A RESEARCH AREA consists of one or several FARMS and each farm belongs to one and only one research area. A farm can contain zero, one or more STANDS, while every stand must be located inside of one farm. Inside a stand zero, one or more description (sample) TREES may be measured. Every tree grows inside one stand.

If there is an EXPERIMENT inside an inventory stand, an "artificial stand" is delineated around the experiment and the "new stand" has a land-use code "experiment". One stand may contain zero, one, or more (adjacent or overlapping) experiments and each experiment may be located inside one or several stands.

A CUTTING UNIT is a group of stands cut at the same time using the same method. A part of a stand or the whole stand may belong to a cutting unit. Different parts of a stand may belong to different cutting units.

Timber from a cutting unit is extracted to a STORAGE POINT on the roadside. A storage point can be also seen as a node in the transportation network. After SALE a FOREST COMPANY transports the timber from the storage point.

3. The database

A database of the KOLI research area was created for the evaluation. The database was developed on VAX/VMS using the ORACLE software. In the following, the database organization is described and the choice of foreign and primary keys reasoned. (A primary key identifies uniquely records in a relation. A foreign key is the primary key of some other relation. Oxborrow 1987.)

The database consists of entity tables, relationship tables, and look-up tables. The entity tables of the inventory database are FARM, STAND, and TREES. Inventory data were

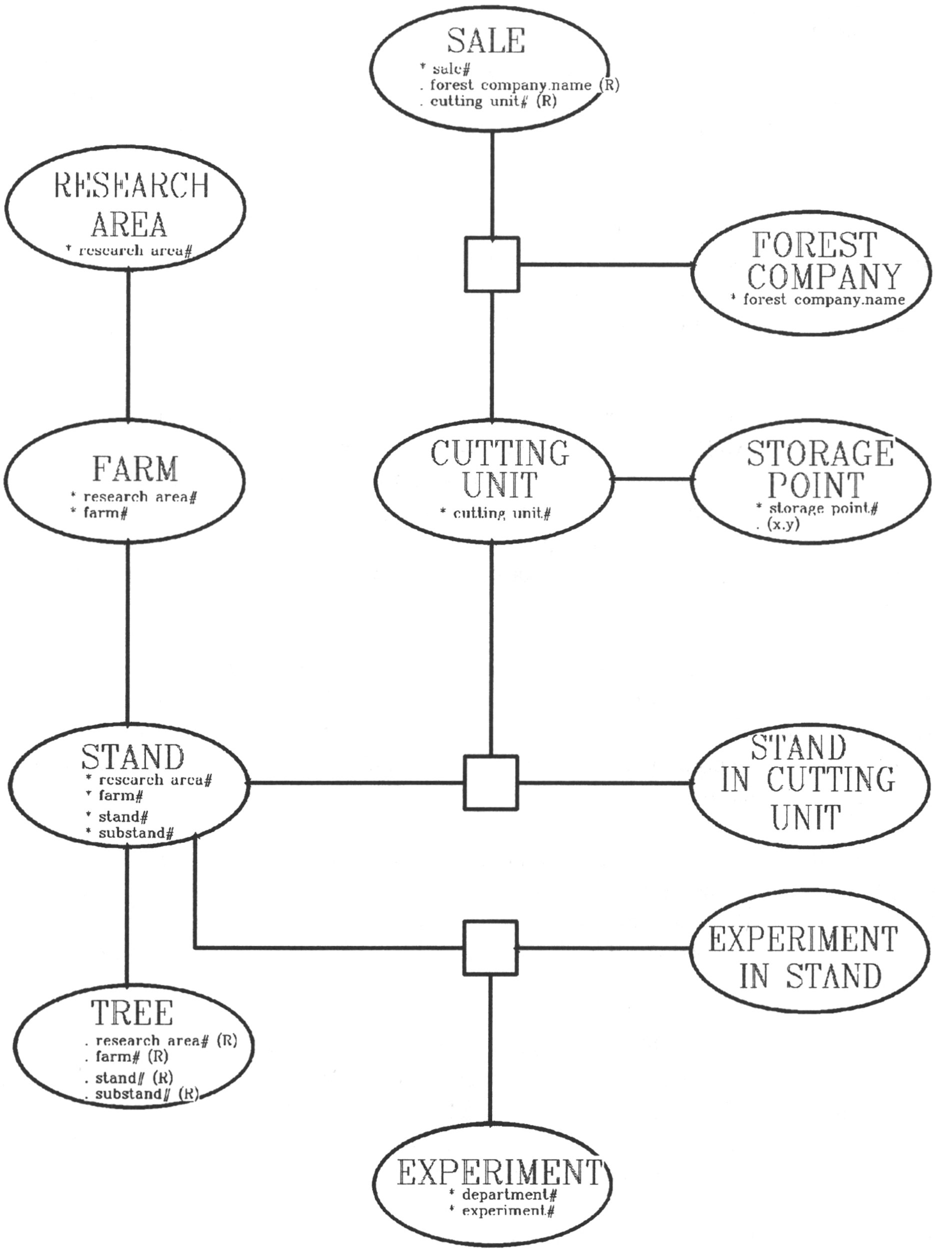


Figure 1. The FFRI research forest data model.

loaded from ASCII-files using ODL ORACLE Data Loader (Gittings and Healey 1987). In addition, tables for EXPERIMENT, CUTTING_UNIT, and STORAGE POINT were added for planning purposes. A temporary experiment "register" containing the treatment proposals was derived from paper documents.

Relationship tables were used to define the connection between entities. Relationship tables were EXPERIMENT_IN_STAND (a list of stands where a given experiment is located), STAND_IN_CUTTING_UNIT (a list of stands which belong to a given cutting unit) and SALE (a table which contains the name of the company which has bought a given cutting unit). Look-up tables were created for codes such as land-use and tree species to give a text description for integer codes used in inventory.

The relationship between a research area and a farm is one- to-many. A primary key of a FARM is a combination of research area number and farm number. (In addition, a map sheet number is given as a foreign key for a map sheet which is a physical disk storage unit of both map and attribute data in the current FFRI inventory system. The key for a map sheet is utilized in data loading.)

The relationship between a farm and a stand is one-to-many conditional (1:Mc). The primary key of a STAND is a combination of research area number, farm number, stand number and substand number. The combination can be used as a key for a polygon in the GIS stand coverage. Because the current system is designed to work independently from a GIS, the coordinates of the reference point and the area of polygon are stored together with other stand characteristics in the stand table. The stand key is utilized to import the stand data from a GIS.

The relationship between a stand and a description tree is one-to-many conditional (1:Mc), because every sample tree is inside a stand and every stand has zero, one or several sample trees but it is not necessary to have any trees in a stand. The relationship is reflected in the TREES table by the combination of attributes research area number, farm number, stand number and substand number.

The relationship between a stand and an experiment is many-to-many conditional (Mc:Mc). The relationship is captured in the model by the relationship table EXPERIMENT_IN_STAND, which has foreign keys for both a stand and an experiment. The primary key of an EXPERIMENT is a combination of department number and experiment number.

The primary key of CUTTING_UNIT is the cutting unit number. The relationship between a stand and a cutting unit is many- to-many conditional (Mc:Mc). The relationship is captured in the model by a relationship table, STAND_IN_CUTTING_UNIT, in which the attribute cutting unit number refers to a cutting unit. In addition, a key for a stand and the percentage of a stand that belongs to that specific cutting unit are stored.

The relationship between a storage point and a cutting unit point is one-to-many conditional. The relationship is reflected in the model by a foreign key for a storage point in the CUTTING_UNIT table. The primary key of STORAGE is storage point number. The coordinates of a storage point are imported from a GIS coverage.

In this study it is assumed that every cutting unit can be sold only to one forest company. The underlying relationship (one-to-one conditional) gives rise to an incident object, SALE. The primary key of a SALE is sale number. The SALE has a foreign key for cutting unit.

4. The interface for MELA

MELA contains a general forest simulator based on individual trees (Siitonen 1983). The input data are read from a file which contains 31 stand variables for each stand and 10 tree variables for every description tree. Because there are normally only 1-9 description trees per stand, a "tree group" which represents a description tree has to be generated to create a more realistic forest. Because interactive SQL does not provide tools for 1) loops (e.g. find all the trees for a given stand and return to read next stand) or 2) procedural statements (e.g. different functions for different tree species), the Embedded SQL had to be used. Pro*FORTRAN programs were used to link together SQL constructs (used to retrieve data from database) and FORTRAN modules (used to generate tree groups).

The MELA simulator is then used to update the stand data and the updated "view" can be imported to the database.

The data exchange between two systems usually includes some code conversions. The code conversions should be automated using the RDBMS data dictionary whenever possible. In this case, however, conversions were based on complex many-to-many relationships and FORTRAN statements had to be used. In the future, the knowledge based extensions to RDBMS could be used in the code conversions.

5. The calculator

The aim of the design of the "calculator" was to test the power of interactive SQL as a decision support tool. This prototype does not contain optimization routines, it attempts to imitate the manual planning process. In timber sale planning, a user might want 1) to print out the current state (e.g. land-use, treatment proposal) of a given stand to determine if it should be cut this year, or 2) to list the total volume of chosen cutting units for the forest company before sale.

In this study, SQL macros were designed to produce inventory reports for stands and summary views of cutting units for the timber sale. (A view is only a virtual table that does not exist but looks to the user as if it did. For example, the view acts as a kind of window through which the forest company can see only the total volume of timber at a storage point - derived from standwise inventory data.)

SQL*PLus commands and relational joins were used 1) to calculate the total volume of a given stand, 2) to insert a given stand into a cutting-unit, 3) to define the storage point where the timber shoud be extracted, and 4) to calculate the statistics of a given cutting unit (i.e. the up-to-date total volume) for timber sale. The stand volume was calculated by summing up the products of the tree volume, the number of trees represented by a tree and the area of a stand. The tree volume was calculated as a function of diameter and height. For the cutting-unit volume the stand volume was updated to the current date by using standwise percent growth model.

The limitations of SQL forced to make simplifications in resource models. Conditional statements would be required to apply different volume functions for different tree species.

Because the interactive SQL does not contain procedural capabilities, conditional statements could not been used and only one common volume function for each species was used.

The unstandardized inventory data caused problems. For example, either basal area or number of trees represented per each description tree is recorded in the field. In the calculation process a conditional statement would be required to determine whether the formula based on basal area or number of trees could be used.

There were also some problems due to the restrictions for "built-in functions". The functions SUM and AVG accept only scalar arguments and nested function references cannot be used inside a call. Therefore, the grouping of trees within a stand was done using so called grouping functions, such as GROUP BY. This function, however, requires that all the fetched columns are within "group" . For example, if the total volume of a stand is calculated by grouping according a stand, an individual stand variable is not within "group".

6. Discussion

In this study, a research forest database, an interface for MELA forest simulator, and a few template queries for the decision support in timber sale planning were developed. In the evaluation process, some knowledge for the design of new inventory, for the design of corporate databases, and for the integration of subsystems were collected.

The created database will serve as a subsystem in the FFRI research forest database and planning system. In the future the database will be integrated with the corporate experiment register. At the moment the relationships between experiments and inventory stands are captured in a relationship table. This simplification requires that the subsystems are updated simultaneously whenever changes in the inventory stands or experiments occur. If the relationship was captured as two separate GIS coverages, the subsystems could be used independently.

This prototype could be easily transferred to any organization which has an RDBMS and an existing inventory database. For example, those forestry units who sell timber as cooperative units could produce a common "sale view" by combining the individual annual cutting unit views. The common "sale view" could then be used to assist in chaining the cutting units along the same route to get the forest companies to raise unit price for timber.

The queries, here referred to as a "calculator", are only templates which an end-user can tailor for his/her own purposes and link with the commercial software package (e.g. INGRES or ORACLE) available. To increase the applicability of the system, the input forms should be modified to include error checking. The on-line help based on the data dictionary should be developed. In the current system, the location of stands and storage points are given as coordinates and the user enters the associative objects i.e. relationship tables (e.g. stands in cutting unit) explicitly. This is a simplification which is both tedious and unrealistic. It should be possible to mark the location of storage point on the map or delineate cutting units ignoring the stand boundaries.

The analysis also suggests that timber sale planning is a spatial problem and, some spatial operations such as AREA, DISTANCE, ADJACENT, CONTAINS, ENCLOSED BY, INTERSECT are required. For example, in timber sale planning it is necessary to analyze site-specific information to evaluate the short- and long-term impacts of treatments on visual and recreational use and wildlife habitat (Bobbe 1987; Reisinger et al. 1990). For simple spatial operations Herring (1988) and Goh (1989) have developed spatial extensions to the SQL.

Most forestry organizations have an existing mapping system or a GIS. The integration of an RDBMS and a GIS via an operational interface such as ARC/INFO RDBI would provide users with two fully equipped toolboxes that work on real corporate databases. The tabular toolbox would provide input forms and report generators for the advanced decision support systems and the spatial toolbox could be used e.g. to improve resource inventories, to analyze the complex spatial relationships, and to illustrate results of planning process on the map.

In conclusion, the evaluation indicates that the RDBMS-tehnology supports end-users by providing an easy-to-use toolbox (interactive SQL) to utilize existing databases. However, some modeling operations can be implemented only using Embedded SQL. In addition, an operational interface between subsystems such as an RDBMS and a GIS would increase the applicability of the system.

References

Bobbe, T. J. 1987. An application of a Geographic Information System to the timber sale planning process on the Tongass National Forest - Ketchikan area. In: GIS '87 Conference Proceedings, San Francisco: 554-562.

Date, C. J. 1989. A Guide to the SQL Standard. Addison-Wesley Publishing Company.

Gittings, B. M. & Healey, R. G. 1987. An Introduction to the Oracle Relational Database Management System.

Goh, P-C. 1989. A graphic query language for cartographic and land information systems. Int. J. Geographical Information Systems. 3(3):245-255.

Herring, J. R., Larsen, R. C. & Shivakumar, J. 1988. Extensions to the SQL query language to support spatial analysis in a topological data base. In: GIS/LIS '88 Proceedings: 741-750.

Oxborrow, E. 1986. Databases and Database Systems. Concepts and Issues. Chartwell-Bratt. Studentlitteratur.

Reisinger, T. W., Kenney, D. P., Goode, K. B. 1990. A Spatial decision support system for opportunity area analysis on the Jefferson National Forest. In: GIS/LIS '90 Conferences Proceedings. Anaheim, California, November 13-15, 1990. pp. 733-740.

Siitonen, M. 1983. A Long Term Forestry Planning System Based on data from the Finnish National Forest Inventory. In: Forest Inventory for Improved Management. University of Helsinki, Department of Forest Mensuration. 17:195-207.

Stochastic Stand Level Optimization Using Scenarios

Lauri Valsta
Finnish Forest Research Institute
Unioninkatu 40 A, 00170 Helsinki, Finland
valsta@metla.fi

Abstract[1]

A flexible model for stochastic optimization is developed that can be used with forest stand simulation models. Stochasticity is represented by a large set of scenarios, each of which is an outcome of stochastic processes. The optimization model is defined in control variable space and it includes the timing, intensity and type of thinning, and rotation length for an even-aged stand. The objective function is nondifferentiable and the optimum solution is found using the direct search method of Hooke and Jeeves.

The stand growth simulator consists of single-tree growth and mortality models, and models for tree stumpage value and logging costs. The model structure would also allow for silvicultural treatments, provided that satisfactory models exist for them. The stochastic environment is described by yearly growth rate leves, growth rate trend, and the probability of a catastrophe, such as wild fire or windthrow.

Numerical results in the case of a risk neutral decision maker show that the optimum rotation is shortened with an increasing probability of a catastrophe. Further, an increasing growth rate variation has mixed and weak effects that depend, in particular, on the tree mortality model. Conversely, increasing risk taking shortens the optimum rotation, given the model set used.

Stochasticity affects the objective function value in the following way (a risk neutral case): an increasing risk of a catastrophe reduces the expected present net value whereas increasing growth variation slightly increases the expected present net value.

About 100 scenarios were required to get a satisfactory representation of the stochastic phenomena. Because of local optima, the optimization had to be replicated a few times to ensure a global optimum solution. When available, the algorithm can efficiently utilize parallel processing.

1. To appear in: Proceedings of The 1991 Symposium on Systems Analysis in Forest Resources. March 3-7, 1991, Charleston, South Carolina, USA. Society of American Foresters.

Knowledge representation in the diagnosis of biotic damage on scots pine - a rule-based expert system

Jouni Väkevä and Hannu Saarenmaa
Finnish Forest Research Institute, Department of Mathematics,
Unioninkatu 40 A, 00170 Helsinki, Finland
vakeva@metla.fi

Abstract

An expert system has been built at the Finnish Forest Research Institute to improve the possibilities of forest managers to diagnose common biotic damages on Scots pine. This rule-based system was implemented by using the development tool Personal Consultant Plus. This presentation describes the knowledge presentation of the diagnostic task. The main principles of constructing deductive rules required to recognize the damaging agents are discussed. The structure and functioning of the system are presented shortly too.

1. Introduction

Consultative expert systems offer a new way of solving diagnostic problems in the area of forest pest management. They can be built to mimic human experts and also to serve as a source of knowledge typically described in books. Several expert systems have been built for diagnosis and management of pests in agriculture but only a few are in operation in forestry (e.g. Schmoldt & Martin, 1986; Clark & Davis 1989; Rauscher & Hacker, 1989; Schmoldt, 1989; Saarenmaa, 1990). At the Finnish Forest Research Institute, we have built an expert system to improve the possibilities of forest managers to diagnose common biotic damages. The product, called "Pest manager", has been built to consult in the diagnosis of insect and fungal damage on Scots pine (Pinus sylvestris). The current version of the approach includes 28 insect species or groups, 6 other animals and 12 pathogens.

The system has been built by using the expert system shell Personal Consultant Plus. It is a tool for building rule-based expert system applications on personal computers. PC Plus has been transformed from medical expert system Mycin (Buchanan & Shortliffe, 1984) for common use. Mycin was constructed specially for diagnostic problem solving, which is the main reason why PC Plus was chosen for the implementation of this work.

The term "rule-based" means that the knowledge is represented as sets of rules which are expressed as IF-THEN statements, an example below. During the problem solving the IF portions (antecedents) of the rules are compared with the facts given by the user. When they match, the rules fire and the actions or conclusions specified by the THEN portions are performed.

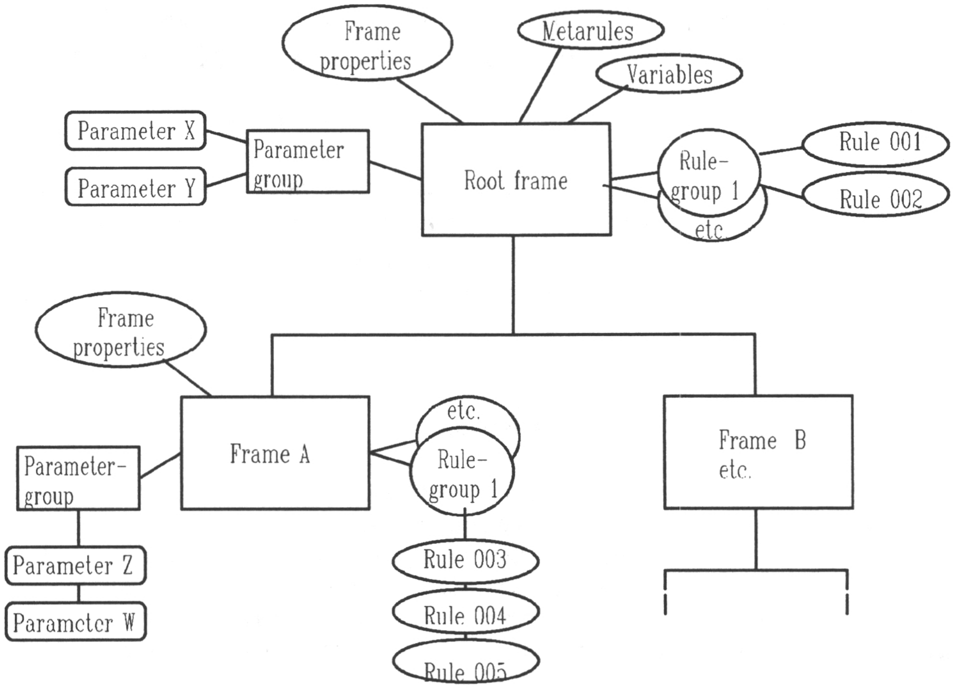


Fig. 1. A simplified overview of the data structures in Personal Consultant Plus.

```
RULE036
If 1) tree species is SCOTS PINE, and
   2) damaged part of tree is CROWN,
Then it is definite (100%) that potential damaging agent
is DIPRIONINAE.
```

This paper is concentrated in the knowledge representation of the diagnostic task. Our purpose is to describe the principles of constructing deductive rules to recognize the damaging agents. The structure and functioning of the system will be discussed too.

2. Development tool

In Personal Consultant Plus parameters, variables, rules, metarules, certainty factors and frames have the main role in knowledge representation (Fig. 1). Rules are constructed by using parameters and variables. Parameters can get numeric or textvalues, also graphical pictures can be assigned to them. Metarules are used to control basic rules during the reasoning. Certainty factors are a mechanism for handling uncertain knowledge. Frames can be used to segmentate the knowledge base (Personal Consultant Plus 1988).

Table 1. Verbal explanations for the numerical values of certainty factors.

Certainty factor (%)	Translation
100	it is definite that ...
80... 100	there is strongly suggestive evidence that ...
50.. 80	there is suggestive evidence that ...
0... 50	there is weakly suggestive evidence that ...
0	no evidence
-50... 0	there is weakly suggestive evidence that ... is not ...
-80... -50	there is suggestive evidence that ... is not ...
-100... -80	there is strongly suggestive evidence that ... is not ...
-100	it is definite that ... is not ...

2.1 Certainty factors

Uncertain or incomplete information can be processed by using certainty factors (CF). A certainty factor can be assigned to a value of a parameter, when the developer or the end user of the system can not be definite about the value. A CF can get numerical values between -100 % and 100 %, and verbal explanation can be given to all the values (Table 1).

When diagnosing damaging agents on a tree, certainty factors can be used to accumulate evidence from many rules pointing to a certain agent. Called the theory of evidence the principles of combining CFs have been rigorously defined by Shafer (1976). Some of the principles are illustrated in Figure 2.

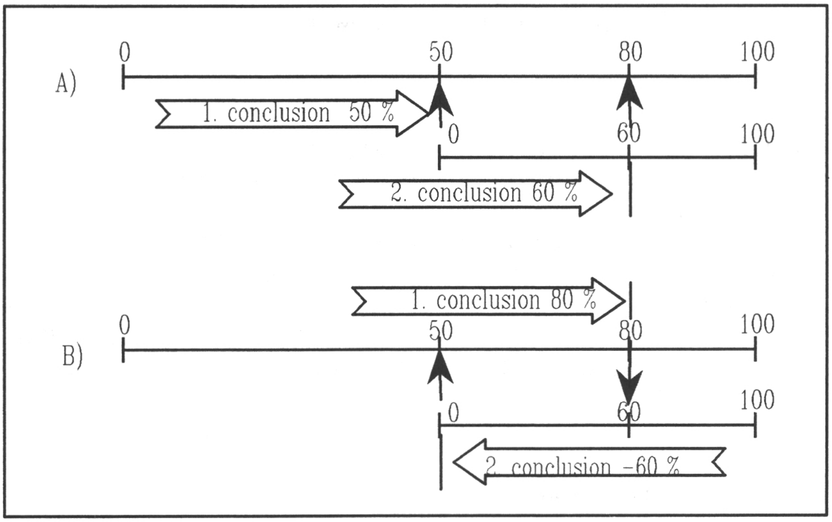


Fig. 2. Combining of positive (A) and negative (B) certainty factors.

2.2 Recognition rules

Knowledge about the damaging agents has been modified into deductive rules, which are called as recognition rules in this paper. They are grouped into three categories: (1) rules of potential agents, (2) description rules of symptoms and damage, and (3) description rules of certain insect species. In the following chapter the main principles of rule construction are discussed. All the rules of genus *Tomicus* and species *T. piniperda* (Linnaeus) and one rule of species Neodiprion sertifer (Geoffroy)are provided as examples of the implementation of the principles.

3. Rule structure

3.1 Potential agents

In this approach the first goal in solving every diagnostic problem is to find out the potential damaging agents. Conclusions about the potential agents can be drawn from the information about the environmental factors and the damaged tree. The end user must give this information as initial data in the beginning of each consultation session. Questions are made about the geographical location, observation month, habitat, height of the tree, and damaged part of the tree. Tree species is not asked because it is supposed to be Scots pine. In the rules this information is described with statements consisting of a parameter and its value.

The main purpose of potential agent -rules is to restrict the number of possible species in the next stage of problem solving. Thus it is not necessary to define such parameters in rules, which are not restrictive. For example, if a species lives all over the country, it is not necessary to write anything about the geographical location into the rule. However one parameter is obligatory in every rule of this type. The damaged part of the tree must be written into every rule because the value of the parameter is needed in control the reasoning later. An example of this type of rules is given below (rule 075).

```
   RULE075
   =======
  If 1) tree species is SCOTS_PINE, and
     2) 1)1) height of the tree is greater than or
            equal to 1, and
            2) damaged part of the tree is CROWN, or
          2) 1) height of the tree is greater than or equal to 3,
            and
            2) damaged part of the tree is TRUNK/BRANCHES,
  Then it is definite (100%) that potential damaging agent
  is TOMICUS_SPP.
```

3.2 Descriptions of symptoms and damage

When all the potential damaging agents have been discovered, the processing of the description rules of symptoms and damage begins. The potential agents will be handled as hypothesis of actual damaging agents affecting the tree. The hypothesis will be confirmed or rejected according to the information of symptoms or damage observed on the tree.

In this approach the tree was divided to subparts, which is natural way to classify symptoms and damage, because damaging agents are usually specified to live in certain parts of the tree (e.g. Saalas, 1949; Graham 1963; Coulson & Witter, 1984). The main parts of tree are crown, trunk and branches, and roots. Crown is divided into three subparts: buds, shoots and needles. Because it is not always possible to specify the damage to these three parts of the crown, three common types of damage of the crown are offered as alternatives: crown partly dead, growth disturbance and needle loss. Trunk/branches are divided to three subparts: bark, phloem and wood. When symptoms and damage was modelled into the rules, the basic principle was that in one rule describes only one or two damages of a certain part of the tree.

After construction of antecedents of a rule, an appropriate certainty factor was assigned to the conclusion. The value of a certainty factor is based on developer's deliberation and experts opinions. When many rules are fired for one damaging agent, certainty factors are combined automatically.

Rules 258, 076 and 184 have been written to describe the typical damage caused by *Tomicus* spp.

```
 RULE258
 =======
If 1) potential damaging agent is TOMICUS_SPP., and
   2) height of the tree is greater than 3, and
   3) symptom/damage in the crown is CROWN_PARTLY_DEAD, and
   4) location of the damage inside the crown is UPPER_PART,
Then there is weakly suggestive evidence (25%) that damaging
agent is TOMICUS_SPP..

 RULE076
 =======
If 1) potential damaging agent is TOMICUS_SPP., and
   2) 1)1) symptom/damage in the shoots is HOLLOW, and
        2) special feature is not the gallery inside
        the shoot full of sawdust, or
      2) 1) symptom/damage in the shoots is CUT, and
        2) special feature is some shoots have fallen
        to the ground,
Then it is definite (100%) that damaging agent
is TOMICUS_SPP..

 RULE184
 =======
If 1) potential damaging agent is TOMICUS_SPP., and
   2) symptom/damage on the bark is SAWDUST, and
   3) 1) observation month is 4, or
   2) observation month is 5, or
```

```
   3) observation month is 6, and
   4) type of the sawdust is FLOURY, and
   5) colour of the sawdust is MOTTLED,
Then it is definite (100%) that damaging agent
is TOMICUS_SPP..
```

The damage described in the rules above can be caused by *T. piniperda* or *T. minor* (Hartig), but it is too difficult to distinguish between the species in those cases. The difference between the species is clear when it is possible to look at egg galleries. The most common gallery types are shown to the user as a graphic picture and the user can select the gallery, which resembles the observed one (Fig. 3). The user can assign certainty factor lower than 100 % to the answer, if s/he can not be sure about the type of the gallery. In the case illustrated in Figure 3 the user has chosen egg-galleries 1, 3 and 7 with different certainty factors.

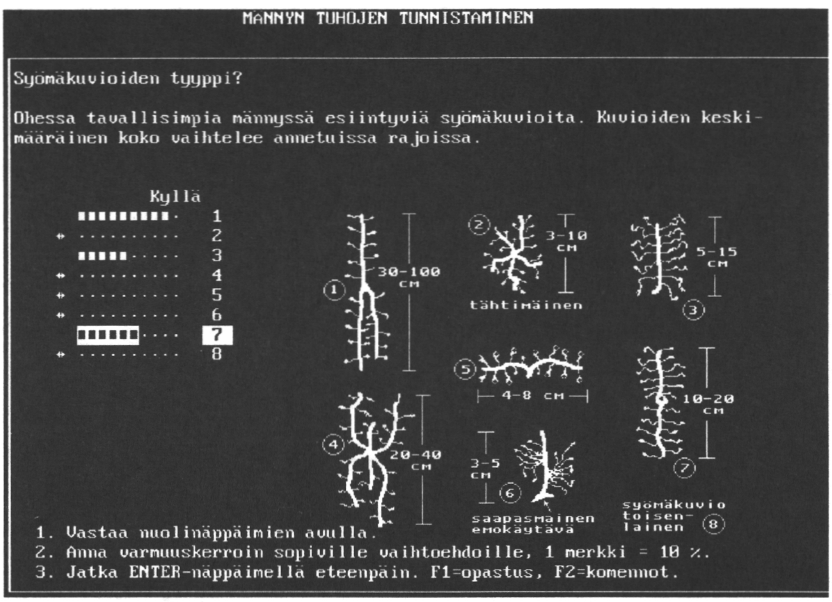


Fig. 3. Example of using graphics in Pest Manager.

For T. piniperda two rules have been written about egg galleries. Even if the observed galleries resemble type one, it is not possible to conclude for certain that the species is T. piniperda, as there are some other species constructing similar galleries. Thus the certainty factor in the rule 181 is 80 %. When one has observed that the egg gallery is located under thick bark and the mother gallery is pitchy, the rule 182 matches and the certainty factor increases to 100 %.

```
 RULE181
 =======
If 1) potential damaging agent is TOMICUS_SPP., and
   2) symptom/damage under the bark is MARKS_OF_EATING, and
   3) sy?ntij?ljet is EGG-GALLERIES, and
```

```
   4) type of the egg-gallery is 3,
Then there is strongly suggestive evidence (80%) that
damaging agent is TOMICUS_PINIPERDA.
```

```
 RULE182
 =======
If 1) potential damaging agent is TOMICUS_SPP., and
   2) symptom/damage under the bark is MARKS_OF_EATING, and
   3) mark of eating is EGG-GALLERIES, and
   4) type of the egg-gallery is 3, and
   5) thickness of the bark is THICK, and
   6) special feature is mother gallery of type-1
      egg gallery is pitchy,
Then it is definite (100%) that damaging agent
is TOMICUS_PINIPERDA.
```

3.3 Insect recognition

The purpose of the third type of rules is to identify the insect species found on the tree. Because it is possible to find a large number of pests and many other species on a tree, it doesn't make sense to try to write rules for every pest. Rules have been written for a few species to make the set of damage description rules more complete. For example when it is not possible to make definite conclusions about Diprioninae species, it can be useful to ask about the signs of larvae. For example in the rule 043 is described the signs of *Neodiprion sertifer* larvae.

```
 RULE043
 =======
If 1) it is suspected that damaging agent
      is NEODIPRION SERTIFER, and
   2) insects observed, and
   3) type of insects is LARVA, and
   4) lenght is less than or equal to 26, and
   5) 1) colour is GREENISH, or
      2) colour is YELLOWISH, and
   6) special feature is head is black and
      there are light and dark stribes on the body,
Then it is definite (100%) that damaging agent
is NEODIPRION SERTIFER.
```

4. Frame structure and functioning

All the rules have been stored in frames, which have been arranged in a hierachical order (Fig. 4). There is total number of 250 rules in Pest Manager (version 1.0, 1991). This kind of modular structure makes it possible to control the problem solving in a way that resembles the thought process of a human expert.

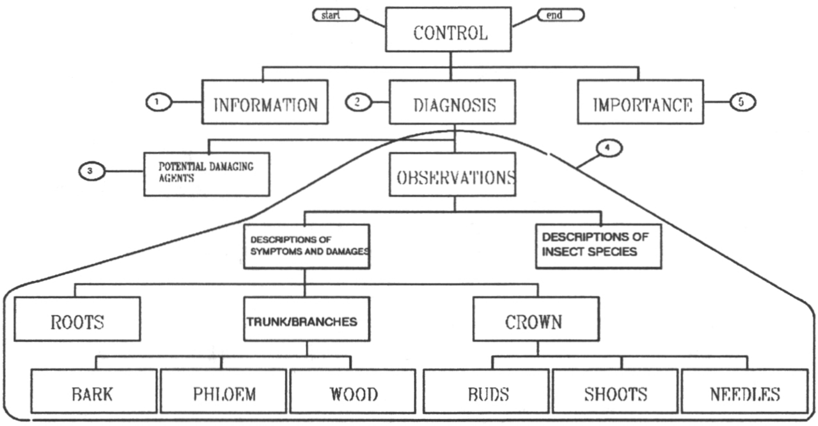


Fig. 4. The frame structure of the Pest Manager and the main steps (1-5) of a consultation session.

A consultation session can be divided to five main steps as numbered in the Figure 4. Consultation begins from the control frame, which controls the whole consultation from the beginning to the end. In step (1) the user can get general information about the Pest Manager. If information frame is activated, the user can check the list of damaging agents, or gets information about the general functioning of the system, or is given advice how to make observations in the field. Diagnosis frame controls the diagnosis in step (2). In the next step (3) the potential damaging agents frame is activated and the initial data concerning site factors and tree characteristics are asked. When potential damaging agents have become discovered, step (4) begins. All the necessary frames under the observations frame are activated and the questioning of observations proceeds dynamically. At the end of step (4), the user is given the damaging agents the PM has identified with their respective certainty factors. If no agent has been identified, then the list of potential damaging agents is printed. In the last step (5) user can activate importance frame and short text files about the importance of the identified agents are shown to the user. Consultation ends in the control frame.

Two main strategies have been used to control the functioning. In the first instance metarules are used to give certain utility values to the recognition rules. All the rules in one frame are given an utility value which differs from other frames, for example metarule 003 is used to assign utility value 70 to the rules in frame "shoots". Due to utility values all the rules of one frame are processed before activating next frame, otherwise the query would jump back and forth from a frame to another.

```
 MRULE003
 ========
If 1) damaged part of the tree is CROWN, and
   2) put any OBJRULES which meets the condition:
   SHOOT SYMPTOMS/DAMAGES is mentioned in the
   rule into SHOOT-RULES,
Then set the UTILITY of (VALUE-OF SHOOT-RULES) to be 70.
```

The second main control strategy is eliminating the frames, which are not relevant in problem solving. All recognition rules give values to the same parameter; "damaging agent", and so all the frames could be activated without any control mechanism. Depending on the previous answers given by the user, frames not relevant are eliminated by using the function "dontconsiderframe". For example if needles have been damaged, only frames "crown" and "needles" are activated, and other tree part frames are not.

5. Discussion

Rule-based reasoning has proved to be useful way to solve diagnostic problems in this work. When an off-the-shelf development tool is used, it is possible to built many small prototypes in a short time and concentrate in the problem solving process instead of programming. Rule language in Personal Consultant Plus is simple, thus rule construction and modifying is easy. The capability to use certainty factors has been proved to be a convenient mechanism to handle uncertain knowledge. When it is not possible to make definite conclusions, the system provides more or less probable answers.

According definitions (e.g. Harmon & King 1985; Waterman 1986; Luger & Stubblefield 1989) real expert systems are programs, which are able to solve problems at least at the same level as human expert. But when expert systems and human experts are compared, we have to remember that techniques to model human knowledge are still poorly known. Use of deduction rules is restricted to problems where it is possible to find out if-then connections between things. Often the problem solving must be simplified before it can be described as rules. In the diagnosis of biotic damaging agents this problem was faced too. Human experts do not think exactly with rules, when they are examining damaged trees. They make visual observations and compare the situation to the personal experiences. If the problem is new one, they can use deep knowledge and general theories to find out the solution (e.g. Luger & Stubblefield 1989).

There is no evidence how well the present system can solve diagnostic problems in reality. At this time the system has been delivered to experts for evaluation. Field tests will be the next step in developing the system. After getting feedback from end users the system can be modified to practical expert system product. The goal of the project is to produce a new medium for increasing the knowledge of damaging agents (Saarenmaa 1987, 1989). Probably it is possible to reach level of human experts in diagnosis of certain damaging agents, but the whole tree dying process is too complicated to diagnose with rule-based system. Too many rules are needed to take into account all possible variations occuring in the nature.

In this approach symptoms and damage have been expressed in rules with words and some graphical pictures, and sometimes it has been rather difficult to find appropriate expressions.

In many cases one or two authentic pictures could compensate number of rules. They could make the query shorter and reduce the possibilities to misunderstanding. Pictures are also fascinating to look at. Technically there are no problems to use pictures in this kind of system. Also multimedia and hypermedia techniques offer interesting possibilities for diagnostic problem solving.

References

Buchanan, B. G. & Shortliffe, E. H. 1984. Rule-Based Expert Systems, The MYCIN experiments of the Standford heuristic programming project. Addison Wesley Publishing Company. 748 p.

Clark, J.L. & Davis, J.R. 1989. A selective bibliography of expert systems in natural resource management. AI Applications in Natural Recource Management. 3 (3): 1-18.

Coulson, R.N. & Witter, A.J. 1984. Forest entomology, ecology and management. John Wiley & Sons. 669 p.

Graham, K. 1963. Concepts of forest entomology. Reinhold Publishing Corporation, New York. 388 p.

Harmon, P. & King, D. 1985. Expert systems, artificial intelligence in business. John Wiley & Sons, Inc. 283 p.

Luger, F. G. & Stubblefield, W. A. 1989. Artificial intelligence and the design of expert systems. The Benjamin/Cummings Publishing Company, Inc. 660 p.

Personal Consultant Plus 1988. Reference guide. Texas Instruments Incorporated.

Rauscher, H.M. & Hacker, R. 1989. Overview of artificial intelligence applications in natural resource management. In: Burkhart, H., Raucher, M. & Johann, K. (edit.). Artificial intelligence and growth models for forest management decisions. Publication No. FWS-1-89 School of Forestry and Wildlife Resources Virginia Polytechnic Institute and State University Blacksburg, Virginia 24061. 21 s.

Saalas, U. 1949. Suomen metsähyönteiset sekä muut metsälle vahingolliset ja hyödylliset eläimet. Werner Söderström Oy. 719 s.

Saarenmaa, H. 1987. Tietokone "asiantuntijana". Metsä ja puu, 1/-87.

- 1989. Tavoitteena metsätietoasema. Metsä ja Puu, 3/-89.

- 1990. Frame- and rule-based knowledge representation in an expert system for integrated management of bark beetles. Tiivistelmä: Kehys- ja sääntöpohjainen tietämyksen esitys kaarnakuoriastuhojen integroidun hallinnan asiantuntijajärjestelmässä. Silva Fennica 24(2): 249-260.

Schmoldt, D.L. & Martin, G.L. 1986. Expert systems in forestry: utilizing information and expertise for decision making. Computers and Electronics in Agriculture, 1 (1986) p. 233-250.

Schmoldt, D.L. 1989. Development and evaluation of an expert system for diagnosing pest damage of red pine in Wisconsin. Forest Science, 35 (2): 364-387.

Shafer, G. 1976. A mathematical theory of evidence. Princeton, NJ: Princeton University Press.

Waterman, D.A. 1986. A quide to expert systems. Addison-Wesley Publishing Company. 419 p.

Operational forest management planning with logic programming

Hannu Saarenmaa, Erkki Kaila[1], Tuula Nuutinen[2] and Taneli Kolström[3]
Finnish Forest Research Institute
Unioninkatu 40A, 00170 Helsinki, Finland
Hannu.Saarenmaa@metla.fi

Abstract

Phases of operational forest harvest planning are connected with the backward chaning reasoning and backtracking mechanism of Prolog. The main goals which are sustainable and increasing supply of timber are realized when the age-class distribution of the forest area is even and when poorly productive stands are replaced by high-yield stands. Planning is done in an opportunistic way. After choosing a stand for cutting, the program selects a proper combination of regeneration methods and estimates subsequent plant damage.

The program is written in Turbo Prolog on a microcomputer. Separate windows in the graphic user interface show the path of reasoning and a map of the forest area. It was concluded that alone the inference mechanisms of Prolog are not sufficient for model integration.

1. Introduction

The present high level of forest production in Finland is by and large the result of an effectively implemented set of opportunistic rules for silviculture. A sustainable and increasing supply of timber has been the ultimate goal of forestry. The rules that implement this have been written in the guidelines of forestry organizations. In essence, they pursue a stable age-class distribution and replace poorly productive stands by higher-yield stands. A systematic approach to forest ameliaration has improved the growing conditions over large areas.

Although modern planning methods such as simulation and optimization have been studied extensively, actual decision-making about silvicultural operations is still largely opportunistic. The connection to long-term planning is often vague because the varying needs of the landowner and the fluctuations of timber prices play major roles in harvest decisions. This

1. Finnish Forest Research Institute, Rovaniemi Research Station, Eteläranta 55, 96300 Rovaniemi, Finland, kaila@roi.metla.fi
2. Department of Geography, University of Edinburgh, Drummond Street, Edinburgh EH8 9XP, Scotland, U.K., tan@geovax.ed.ac.uk
3. Finnish Forest Research Institute, Joensuu Research Station, P.O.Box 68, 80101 Joensuu, Finland, taneli@joyl.joensuu.fi

is supposed to cause inoptimality losses (cf. Lämås et al. 1991). On the other hand, the opportunistic planner uses up-to-date data, which is more accurate than the predictions and constants used by long-term planning models which normally have been compiled years before.

There are many approaches by which short-term and long-term plans can be combined. One approach is to generate a list of candidate stands with optimization and then do more detailed operational planning among them (e.g. Hokans 1984, Integrated... 1989, Nuutinen 1990, Loh 1991) Detailed planning means combining harvest units, assessment of environmental effects, and decisions about renewal methods.

Merging long-term and short-term planning is possible if we can ensure that they share goals and constraints. This is possible to implement with logic programming (cf. Stefik 1981 a, b). With logic programming, we backward-chain in a tree-like structure (Fig 1) from a common high level goal to the end nodes that define the detailed actions. In one branch are the results of long-term planning and in the other the executable actions of the short-term problem solving. The constraints that are set early in the search are in effect throughout it, but may be systematically revised with the backtracking mechanism of Prolog. This process is parallel to hierarchical activity models (Martin 1989).

The objective of this paper is to explore the integration of various planning phases via goal-driven reasoning. We present a dependency network of subgoals to be satisfied in forest regeneration and we implement them in Prolog. The current system is a research prototype and cannot be used as a serious management tool. Our goal is to explore the application of some classical artificial intelligence planning methods to forest management planning and to discuss the findings.

2. Methods

2.1 Artificial intelligence and planning

Planning involves formulating a set of actions to achieve a goal. These actions can be drawn up beforehand and retrieved from a database or they can be generated during the planning process. Rule-based reasoning, in which the rules form a closed search space, is of the former kind. However, it is a special case and not very widely applicable for planning problems. Most true planning systems, such as those of robotics, do not care to generate the complete search space. Instead they have operators that expand the search from one node or action to another and selectively search only the relevant part of it (Fikes and Nilsson 1971, Cohen and Feigenbaum 1982).

Because of this generality, planning is still poorly handled in the expert systems world. Most planning systems can, however, be classified under the wider definition of knowledge-based systems (see e.g. Winston 1984). One can, in fact, plan quite well with a rule-based system where the constraints for search are stored. This is generally the case with forest management, which is guided by legislation.

Rule-based planning works by backward-chaining from one or several goals to end nodes while constraints are propagated. For instance, when a Prolog goal matches, a pointer sets

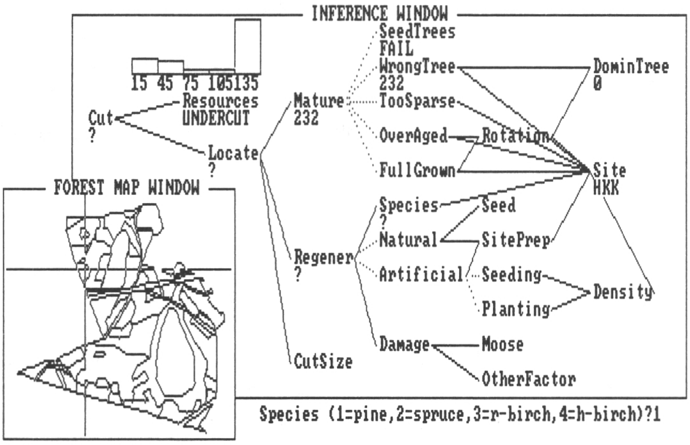


Figure 1. The user interface shows the rule network. This is the actual screen of the system on a PC with a 640*200 pixel CGA. Solid lines represent AND-type and dotted lines OR-type dependencies.

the values of parameters so that these will be in effect while the rest of the search is done; in this way the rest of the search is constrained. The constraints can also be called interactions between subproblems (e.g. Stefik 1981a, b).

A planning problem is best solved by dividing it into subproblems; within forestry these subproblems are steps in decision making. First, the top-level goals such as the amount of timber are decided and then lower level goals such as where to find it and what actions to take to regenerate the stand are addressed. This approach fits within the definition of opportunistic planning (Hayes-Roth 1980), in which small trial steps are made constantly and backtracking is heavily involved. Logic programming is an appropriate representation of this process.

2.2 Prolog

Invented in France in 1971 and polished to its present definition in Edinburgh (Clocksin and Mellish 1981), Prolog is a high-level AI language for logic programming. It has a built-in inference engine that tries to prove clauses true or unknown. To prove a clause, it searches for its antecedents and tries to prove them in turn. The search is over when all the end nodes, facts with no antecedents, are found. If the original clause was true, the next one is executed, but if it failed, then the inference engine backtracks to where it last branched.

A Prolog program can proceed either down or up and via multiple paths, which makes it very peculiar to use in the beginning. It can also be used inversely by asking the program which conditions would realize a given goal. The inference engine can be forced to inhibit backtracking with the "cut" code in the program. However, its use compromises the pure declarative logic by introducing an imperative programming style.

Compiled Prolog is efficient even in microcomputers and it does not use as much memory the way LISP does. There are numerous implementations available in the marketplace. We used Turbo Prolog because of its graphics capabilities and good windowed user inteface. Its manual provides a good introduction to Prolog (Anon. 1986). Turbo Prolog requires that variables be typed, which makes it non-standard but, on the other hand, very fast. The source code for the present program is available upon request from the first author.

3. The program

A coarse goal-driven function model of the opportunistic planning in the Finnish forestry was implemented with Prolog. The program has a graphic interface so that the planning process can be visually monitored all the time (this is illustrated in Fig 1 which should be referred to while reading this section). When backward-chaining from the goals, colored question marks appear under the nodes showing the current reasoning path. When the search returns, the values found replace the question marks. In another window, map of the forest area is shown. A colored crosshair points to the current stand.

The forest is described in a Prolog database with the same attributes that are used in forestry practice. There are two relational tables: one holding the stand data retrieved from an operational stand database in a mainframe computer, and another holding the coordinates which have been retrived from a mapping system. The databases are automatically searched by Prolog for stands relevant to the the current subgoal.

The ultimate goal in the model is to have a desired amount of timber. Cutting is possible if we have a forest area where the age-class distribution allows cutting now without creating a gap that would make cutting impossible at some future time. The distribution is checked, shown graphically, and classified as either stable (allowing cutting) or unstable (suggesting resting). This subgoal approaches the goals of long-term planning. In the present system we have not developed interpreting long-term plans more deeply; they are the subject of future work.

If the age-class distribution did not fail, the model starts to search for a stand to be cut. The stand must be mature, capable of being regenerated, and its cutting must not conflict with multiple-use values (e.g., too large a clearcut area). When deciding about maturity, the model considers the following:
- seed trees that have done their job
- stands with a dominant tree species inappropiate for the site
- stands too sparse for the tree species on the site
- over-aged stands on the site
- stands that have no faults but are full-grown

This priority list is used in practice and it approaches sorting by value-growth and by the difference between actual and potential yield on the site. From the subgoal there is an OR-type relation to one of these potential mature stands for cutting. By an OR-type relation we mean that only one antecedent needs to succeed for the goal to succeed. As soon as one is found, it is highlighted in the forest map window. The name of the stand appears under the branch where it was found in the inference window. The stand is automatically marked in the database by the Prolog inference engine, and the process goes on to the regeneration branch.

The order in which decisions about regeneration are made is not definitely closed. There are two possibilities: either we first decide the new tree species, or alternatively we decide the regeneration method. If we opt for artificial regeneration, the outcome of these two alternative decision paths is the same; if natural regeneration is preferred, the resulting regeneration chain might be different. The present model is built so that the tree species is decided first (Fig 1), but either option is possible in principle. Turbo Prolog does not allow asserting rules in run-time and thus the user cannot be given this decision; however, on a true Prolog system this rule can be asked for. Asserting rules in run-time can be useful if an expert user has strong opinions and wants to modify the reasoning paths.

After the tree species is selected, the conditions for natural regeneration are checked. If there is seed available for the new tree species in the present stand and the soil will not prohibit the development of young seedlings, natural regeneration is selected. Should the soil condition fail, a site preparation method is asked for. Should the seed production fail, the program backtracks to select another tree species.

If natural regeneration fails for the selected tree species, artificial regeneration is called for. Site preparation and either seeding or planting to a minimum density are chosen. Artificial regeneration never fails. (Except in the case of poor work, which is ignored here. See discussion.)

Subsequent damage by a variety of factors can, of course, thwart the results of all regeneration efforts. To assess the probabilities of these events, models can be embedded in the system or called as from outside. At present we have not included these.

Fig 1 shows clearcut size as an example of multiple-use values. It is related to the particular stand that is being evaluated as a subject for cutting. However, because multiple-use values do not conform to the ultimate goal of timber harvesting, they should form another parallel reasoning path. Both timber values and multiple-use values can then be connected at a higher level of forest utilization.

4. Discussion

The research that led to this model started from the need to have a general framework for models that predict damage to plantations by forest pests. Such models include mathematical functions for the different mortality agents (Näslund 1985) and models of moose behavior and resource utilization (Saarenmaa et al. 1986, Saarenmaa et al. 1988). We did not want a stand-alone model that would have planned damage avoidance correctly while ignoring other measures or implementing them incorrectly. The present model not only provides such a top-level scheme but can also incorporate other forest regeneration models (see Fig 1). Such a model is "VILJO", by means of which a forester can compare alternative permutations of forest regeneration methods (Parvianen et al. 1985).

As yet, the present model integrates different subgoals only on the conceptual level. We do not have separate optimization models as one branch for long-term planning. The use of age-class distribution was meant to point out this need and provide some long-term stability. The present approach is to interpret long-term plans the same way by which middle management has its own decision authority within a corporate strategy. The actual mechanism when implemented might be a planning algorithm providing a set of possible actions that are close to the optimum; the forest manager using the short-term problem solver would then select a few of these to be executed.

Both the short-term and long-term branches of the system integrate many different kinds of models. There are submodels for selecting a stand for cutting, regenerating it and providing estimates of plant damage. The models can be used as external routines. However, logic programming as such provides a possibility of replacing them entirely. This might be necessary, since these models issue the same problems as expert systems (Parviainen et al. 1985, Hämäläinen et al. 1985). Some of the regeneration chains have already been built into the present model.

We had to structurize our thinking and distinguish the processes that 1) cause mortality in the old stand, 2) produce the new one, and 3) cause mortality in the new stand. These correspond to the subgoals "mature", "regeneration", and "damage", respectively. This led to such surprising rules as "artificial regeneration never fails". This is not simplistic at all, but only a way to re-organize knowledge so that it can be made functional. Such a re-organizing is necessary if we want to integrate existing models.

The present model does not incorporate thinning. However, there is no obstacle to including such a subgoal in the "mature" branch. With that it would include most short-term forest management operations. Long-term investments such as fertilization and drainage can be included in the branch which checks for age-class distribution and resource stability.

There are two search spaces in forestry: management rule-bases and forest databases. Visual display of reasoning paths in both of these is crucial for understanding and decision making. Consequently, the user interface of the present system accounts for more than 70% of the 24 Kbyte code. Although the present system is very cheap and efficient, the small resolution of microcomputer screens constitutes a serious limit for these programs. However, without a good graphic interface, this type of model would be nearly useless.

Model integration appears to be a valuable goal for developers of computer-based decision aids. A superb example in this direction has been provided by Kourtz (1987). He used Prolog to tie together a number of FORTRAN models and to provide the reasoning mechanisms for fire management. The CHAMPS program (Raushcer 1988) integrates various tools for stand-level forest planning with Prolog. In agriculture, some Prolog-based integrated systems have already been delivered (Stone & Toman 1988). However, large Prolog systems are difficult to maintain, and for this reason most projects have dropped it lately (we too) in favor of object-oriented approaches. After all, it is very simplistic to base model integration on an inference method only.

Acknowledgements

We thank all those who have commented this research and paper. Especially the advice of Michael Rauscher has been useful.

References

Anon. 1986. Turbo Prolog. Owners handbook. 221 pp. Borland International Inc., Scotts Valley, California.

Clocksin, W.F. & Mellish, C.S. 1981. Programming in Prolog. 297 p. Springer-Verlag.

Cohen, P.R. & E.A. Feigenbaum. 1982. Handbook of artificial intelligence. Vol. 3. 639 p. William Kaufmann Inc.

Fikes, R.E. & Nilsson, N.J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence 2, 189-208.

Hämäläinen, J., Kaila, S. & Keskinen, S. 1985. Laskentasysteemi metsänviljelyn menetelmien vertailuun. Summary: Calculation system for comparing forest regeneration methods. Metsätehon Katsaus 18/1985, 1-4.

Hayes-Roth, B. 1980. Human planning process. Rand Corporation, Rep. R-2670-ONR, Santa Monica, California.

Hokans, R.H. 1984. An artificial intelligence application to timber harvest schedule implementation. Interfaces 14(5): 77-84.

Integrated Forest Resource Management System (INFORMS) 1989. System Design Workshop Report. U.S. Forest Service, Report RFQ RM-89-64, 58 p.

Kourtz, P. 1987. Expert system dispatch of forest fire control resources. AI Applications in Natural Resource Management 1, 1-8.

Lämås, T., Ståhl, G. & Thuresson, T. 1991. Estimating inoptimality losses in harvesting decisions using neural networks. In: the present volume, p. 5-12.

Loh, D.K, Chu, Y.-T., Holtfrerich, D.R. & Choo, Y.K. 1991. Integrated resource management automation (IRMA). In: the present volume, p. 69-77.

Martin, J. 1989. Information engineering. Vol. I. 208 p. Prentice Hall, Englewood Cliffs, New Jersey.

Näslund, B.-Å. 1985. Simulering av skador och avgång i ungskog och deras betydelse för beståndsutvecklingen. Summary: Simulation of damage and mortality in young stands and associated stand development effects. Swedish University of Agricultural Sciences, Department of Silviculture, Report 18, 1-147. Umeå.

Nuutinen, T. 1990. GIS in harvesting scheduling of private forests in Finland. GIS/LIS '90 Proceedings, p. 432-439.

Parviainen, J., Sokkanen, S. & Ruotsalainen, M. 1985. Metsän uudistamisen vaihtoehtoja vertaileva laskentaohjelma "VILJO". ("VILJO" a program for comparison of forest regeneration chains). Metsäntutkimuslaitoksen tiedonantoja 179, 1-93. Finnish Forest Research Institute, Joensuu Research Station.

Rauscher, H.M. 1988. Using expert systems for stand level forest planning. Minnesota's Timber Supply: Perspectives and analysis Conference, Grand Rapids, Minnesota, September 21-22, 1988. 14 p.

Saarenmaa, H., Packard, J.M. & Grant, W.E. 1986. An artificial intelligence approach to animal behavior and resource utilization. 4th International Congress of Ecology. August 11-16, 1986. Syracuse, New York.

- , Stone, N.D., Folse, L.J., Packard, J.M., Grant, W.E., Makela, M.E. & Coulson, R.N. 1988. An artificial intelligence modelling approach to simulating animal /habitat interactions. Ecological Modelling 43: 125-141.

Stefik, M. 1981a. Planning with constraints (MOLGEN: Part 1). Artificial Intelligence 16, 111-140.

- 1981b. Planning and meta-planning (MOLGEN: Part 2). Artificial Intelligence 16, 141-170.

Stone, N.D. & Toman, T.W. 1988. COTFLEX: An integrated expert and database system for decision support in Texas cotton production. 7 pp. AAAI Workshop on Integration of expert systems techniques with conventional programming in agriculture. San Antonio, Texas, August 10-12, 1988.

Winston, P.H. 1984. Artificial Intelligence. Addison-Wesley, Reading, Massachusetts. 524 p.

Integrated Resource Management Automation (IRMA)

Douglas K. Loh, Yi-Te Chu, David R. Holtfrerich & Yew K. Choo
STARR LAB
Department of Rangeland Ecology and Management , Texas A&M University
College Station, Texas 77843-2126, USA
loh@starr20.tamu.edu

1. Introduction

Decision making in natural resource management is becoming an increasing difficult task that requires the balancing of a number of environmental and socioeconomic concerns. The volume of information that must be analyzed and the complexity of the decision making process demands that computerized systems be developed to provide decision support services (Loh et al. 1988). An integrated systems approach that couples database management, geographic information systems, simulation models, and expert systems is needed (Loh and Rykiel 1990). We refer to such an approach as integrated resource management automation and describe a demonstration system, IRMA, that has been developed to convey the design concept of integrated systems.

IRMA stands for Integrated Resource Management Automation. It is a USDA Forest Service/Nicolet National Forest and Forest Pest Management Technology Development Project. The IRMA project supports the Forest Pest Management Strategic Plan for Forest Health by integrating pest management considerations into project level planning. IRMA is intended for resource specialists at the field level of the Forest Service. It will make available to managers and resource specialists an integrated user interface between the field level project planner and corporate computer tools such as GIS, Relational Databases, and Expert Systems (USDA Forest Service 1989).

2. Components of IRMA

IRMA is divided into five sub-systems: the user interface system (UIS), the management information system (MIS), the database management system (DBMS), the geographic information system (GIS), and the connection management system (CMS). The object oriented programming paradigm and artificial intelligence techniques have been used to integrate these sub-systems into a uniform window environment with a mouse-driven user interface. Such integration provides the concurrent processing and multitasking capabilities needed for a "full service" information processing system.

IRMA functions by allowing a team of individuals at a district to combine in a graphic system spatial and tabular information concerning management directions for a particular geographic area where implementation of management practices is being considered. Once all relevant information has been entered, the team performing the analysis can examine the consequences of alternative management activities in this particular area. These proposed actions are compared with the management directions for compliance. Finally, IRMA assists the analysis team in evaluating the pros and cons of each alternative project set.

The IRMA system integrates spatial and tabular data along with specific management strategies. All relevant information is downloaded into a fact base in a personal computer for integration. A rule base written in CLIPS (C Language Production System) takes input from the fact base and fires the rules accordingly. The results generated by the rule firing process provide the basis for supporting management decisions.

2.1 User interface system

The user interface is designed around Microsoft Windows/386 and exploits many of its powerful features. User responses are input by clicking a mouse to select options from pull-down menus. Other systems such as expert system shells, word processors and spreadsheets can be easily run from the windows environment along with the IRMA system.

2.2 Management information system

This module is the reservoir that holds domain specific knowledge. It consists of two main parts. The expert system base contains the inference engine and the rule bases, while the model base contains quantitative knowledge such as simulation models and evaluation functions. System modification can be made by developers, system maintenance personnel, and users whenever necessary.

2.3 Database management system

This part of the system provides domain independent data from existing sources such as forest inventory information in the Map Overlay and Statistical System (MOSS) and in the Vegetation Management Information System (VMIS).

2.4 Geographic Information System

Basic functions associated with standard GIS permit analysis and display of the spatially referenced data as well as interaction with analytical procedures used in many settings of natural resource management. These functions include generating maps, intersecting data themes, creating buffers, zooming in on sub-areas of a map and scrolling horizontally and vertically across maps.

2.5 Connection management system

This module manages the communication between computers by using standard protocols, remote logins and file transfers. However, the user deals only with a graphics workstation through the UIS. The underlying process is transparent to the user.

3. System Requirements

3.1 Hardware

IRMA is designed to work with the IBM PC/386 and compatibles. An Intel 80387 math coprocessor is needed for advanced calculations. Your computer should have at least 4M of RAM (6M if intended to upgrade to UNIX environment,) one 1.2M floppy drive and a hard disk with at least 70M of free space. A mouse driver has to be installed in your AUTOEXEC.BAT or your CONFIG.SYS file. A color monitor with EGA (Enhanced Graphic Adapter) is recommended for screen display and a printer/plotter is needed for generating hard-copy output. Other recommendations include tape drives for backups and an Ethernet card for communication.

3.2 Software

Microsoft Windows/386, XXCLIPS (a modified version of CLIPS), and a database management system (currently R:BASE) need to be installed before operating IRMA.

4. System Functions

IRMA uses the Microsoft Windows environment as the standard interface. Pull-down menus are used to make choosing and understanding commands easier. In addition, a window environment is used to display multiple items at the same time. Figure 1 illustrates the user interface with the initial IRMA screen.

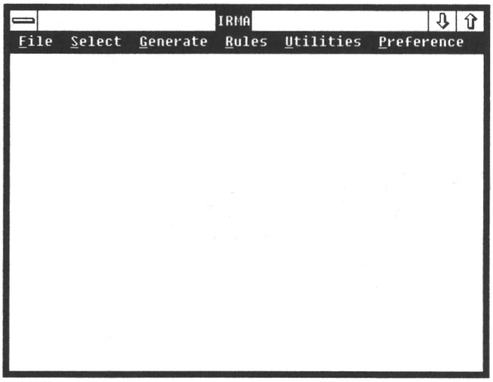


Figure 1. The user interface with the initial screen of IRMA.

The functions provided by IRMA are grouped into six categories: File, Select, Generate, Rules, Utilities, and Preference.

4.1 File

The first item in the menu bar is **File** which includes the following functions to perform activities at the file level:

The **New Session** function is used to locate the database to work with when you start a new session.

The **Open, Save**, and **Save As** functions are used to open and save a session. All settings, layers, and scores generated can be saved and resumed.

The **Print preview** and **Print** functions are used to view the content of the window, rearrange it on a page, and print hard copies.

The **Copy Screen** function is used to copy the content of the window to the system clipboard. You may then paste it to other applications to create your own report.

The **Export** function is used to export a file in MOSS format. This file can be imported back to MOSS and used as a data layer.

The **Exit** function is used to exit the system. The user can still work on other applications under the window environment after exit from IRMA.

4.2 Select

The second item in the menu bar is **Select** which includes functions to display information in a tabular format as well as in a graphic format. You can select target areas by entering display criteria and/or by overlapping different map layers.

The **Map** function is used to create and display a map of your interest. Figure 2 shows an example of the Map function.

The **List Subjects** function is used to show a subject listing of the GIS layers currently plotted on the screen. These are the same subject names used in MOSS.

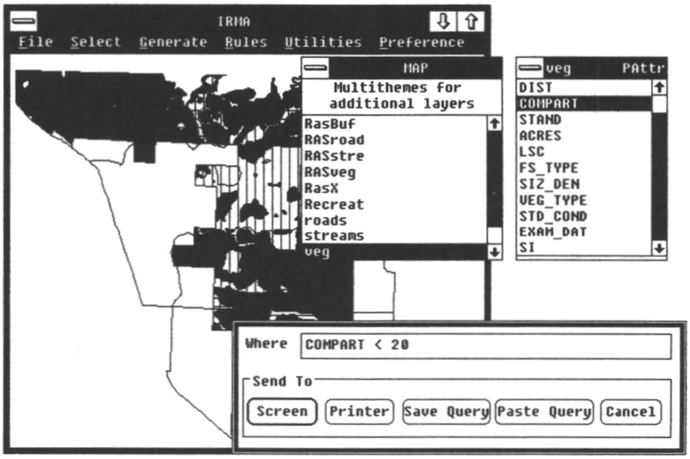


Figure 2. An example of using the Map function to display a map of an area.

The **Quick Query** function is used to obtain minimal quick information about any polygon, point, or line on the screen (Figure 3).

The **Table** function is used to display a table containing the columns (attributes) from the database (Figure 4). The values in each column can be summed up by using the Sum Columns function in the system menu.

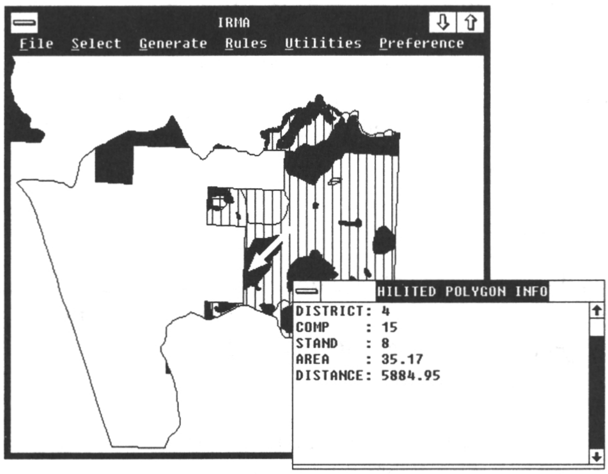


Figure 3. An example of Quick Query. The polygon of interest is highlighted and the information about the polygon is displayed in a separate window.

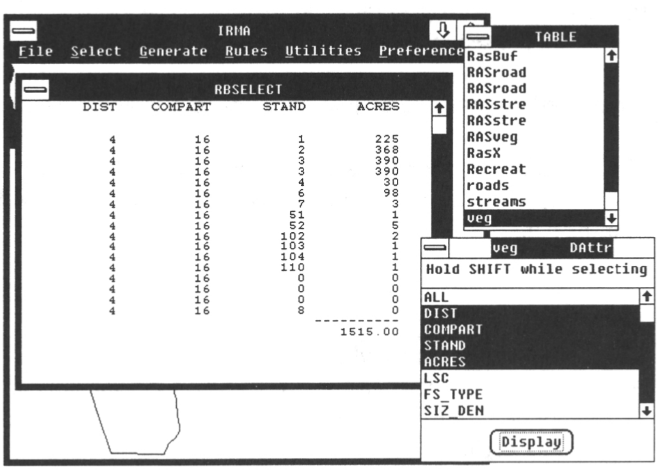


Figure 4. An example of a table listing four columns (DIST, COMPART, STAND, ACRES) in the Veg (Vegetation) table.

The **Report** function is used to display a set of programmed queries. The output is a map showing the candidate stands which includes a report listing the FORPLAN candidate stands, their relevant attributes (acres, Mbf/acre, total Mbf, etc.), along with the methods of harvest selected for the current period.

The **Previous Queries** function is used to recall a query that the user has saved for later use.

The **Do MultiThemes** function is used to add layers to the existing map. By activating this function, any new map will be plotted as a layer in addition to the existing map.

The **Zoom History** function is used to display the history of the zooming activities. The user is allowed to select a part of the map and zoom it as many times as desired. The system will keep track of the ten most recent zooming activities to be viewed by the user at any time.

4.3 Generate

The third item in the menu bar is **Generate** which includes functions to create and display intersections, buffers, differences, and unions of map layers. Both raster and vector formats are available for generating intersections and buffers while raster format is currently used in generating differences and unions.

The **Intersect** function is used to generate an intersection between two map layers. The results will be saved in a file so that they can be displayed anytime without having to be regenerated.

The **Buffer** function is used to generate and display a buffer for a map layer. It is useful when the user needs to locate the region within a certain distance of a road, body of water, or rare plant.

The **Difference** function is used to generate and display the region which is in the first (base) map layer but not in the second.

The **Union** function is used to unite and display two sets of polygons together.

The **Raster** and **Vector** functions are processing options for the user to select. The raster option uses grid cells to represent data while the vector option uses patterns of points, lines and area to represent the data.

4.4. Rules

The fourth item in the menu bar is **Rules** which contains functions to access the rule base of the system. The rule base takes into consideration all forest plan standards and guidelines, issues, concerns, opportunities specified from the public, and special management strategies to aid in decision making.

The **Evaluate** function is used to initiate the rule base. The system will take user inputs and automatically load the stands currently in your working area to form the fact base.

The **Result** function is used to display the output from running the rulebase (Figure 5). The score tables contain certainty factors of each management alternatives for the stands. The certainty factors range from -1 to +1 to indicate the preference of each alternative. The reasoning tables contain the rules that have been fired during the reasoning process.

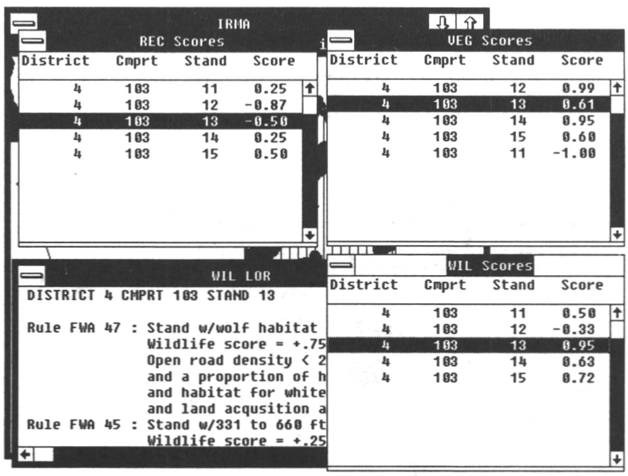


Figure 5. A sample result of the rulebase reasoning. Scores for each alternatives for the stands are displayed along with the line of reasoning table for comparison and references.

4.5 Utilities

The fifth menu item in the menu bar is **Utilities** which contains extra capabilities to aid the user in analysis.

The **Help** function is used to access the on-line help capabilities provided by the system. It gives helpful instructions on the meaning and use of a specific function.

The **Legend** function provides the description of the map including symbols and colors.

The **Distance** function is used to locate a point, draw a line, and/or draw a polygon anywhere on the map. The system will show the UTM (Universal Transverse Mercator) coordinates of a point and display the distance and area of a line or a polygon interactively (Figure 6).

The **Layer Area+Dist** function gives you the area and distance of a layer. The units used by the function are selected by the user in the **Display Unit** function under **Preference**. It is also useful in locating the buffered and/or intersected areas.

4.6 Preference

The sixth item on the menu bar is designed to serve different user needs. You may use these functions to set display unit, arrange layout order, set line width, and select a text font. The preferences you set will be used by the system throughout the entire session.

The **Display Unit** function gives the user the flexibility to select units used for distance and area display (Figure 7).

The **Layer Layout** function is used to rearrange multiple layers on the screen. The system will re-plot the map according to the order assigned.

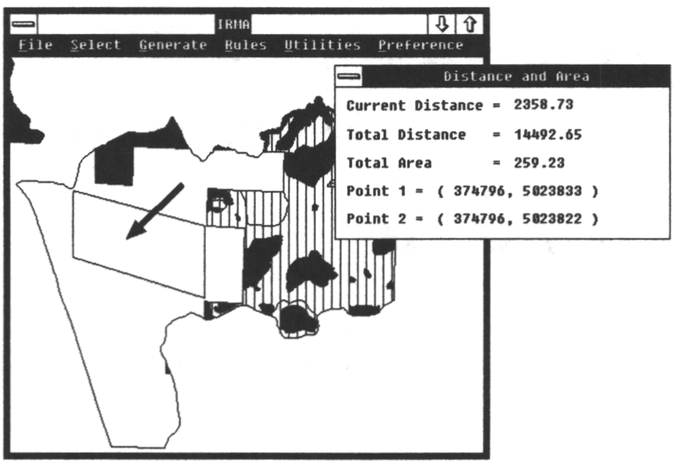


Figure 6. An example of the Distance function. The UTM coordinates of the point, the distance and area are displayed interactively as the user points and draws on the screen.

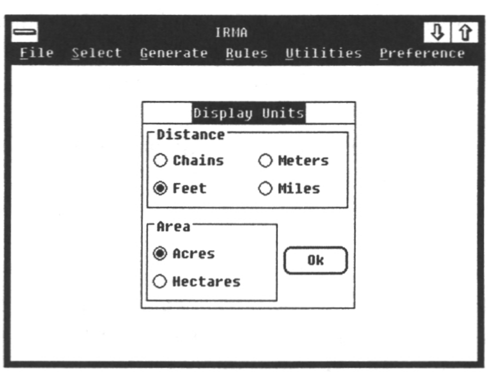


Figure 7. The dialog box which asks the user to select display units for distance and area.

The **Line Width** functions allow you to select the width of the lines for display. **Line Width 1** displays lines in one pixel wide and **Line Width 2** displays lines in two pixels wide.

The **Text Font** function lets the user to select the font used in printing and table displaying.

Conclusions

Because IRMA is designed in the Microsoft Windows environment, in addition to the functions described above, any commercial software products that runs under the same environment are easily accessed by the user. Word processing software and Spreadsheet packages can all run parallel with IRMA. IRMA is just a "window" which provides an integrated view to other software products from the perspective of a project planner.

References

Loh, D. K., English, K. J., Choo, Y. K., Chu, Y. T. and Sun, M. 1988. Integrated natural resource management automation- the knowledge shop. Proc. Resource Technology 88, American Society for Photogrammetry and Remote Sensing, Falls Church, Va., pp. 158-167.

Loh, D. K. and Rykiel, E. J. Jr. 1990. Integrated resource management systems: Coupling expert systems with database management and geographic informations. Abstracts of the Plenary, Symposium Papers and Posters. V International Congress of Ecology. Yokohama, Japan. August 23-30, 1990. 503 pp.

USDA Forest Service. 1989. Integrated Forest Resource Management System: System Design Workshop Report. RFQ RM 89-64. USDA Forest Service. 58 pp.

Decision Support Systems in Farm Management: the Potential of Integrating Expert Systems with Conventional Problem Solving Techniques

Pentti Hyttinen
University of Joensuu, Department of Forestry
P.O. Box 111
SF-80101 Joensuu, Finland

Abstract

The objective of this study was to assess the potential for integrating expert systems (ES) with current problem solving techniques in farm management -- especially, in whole farm planning including agriculture and forestry as well as other sources of income. First, the decision-making environment of a farm was described, and the current state of technologies used in supporting farm manager's decision-making was surveyed. Researchers call for integration of expert systems and conventional problem solving techniques such as linear programming (LP). It is assumed that integrated knowledge- based systems would make existing conventional applications more accessible. This assumption was tested by developing a prototype of an expert system that serves as a front end to a LP-based whole farm planning method. Tests with the prototype indicate that this approach has several advantages compared with conventional methods.

1. Introduction

In Finland, farming can be seen as a combination of agriculture and forestry. An average farm consists of 13 hectares of arable land and 37 hectares of timberland (Kettunen 1990). Throughout agriculture and forestry, the drive for more efficient production, harvesting and processing has become of major concern. Problem solving techniques, such as modelling, simulation, and optimization have been used extensively to help scientists and practitioners understand and control biological systems.

By their nature, however, most of these systems are difficult to define quantitatively. Many of the models and simulations that have been developed lack a user interface that enables people other than developer to use them (Whittaker & Thieme 1989). For example, in linear programming, it is not uncommon that problems have thousands of constraints and activities. Matrix generator systems have been designed to help in this process. However, the amount of manual labor involved is still very great and the formu lation process is subject to errors which are difficult to detect.

This paper examines the potential for integration of expert systems and conventional problem solving techniques to relieve the problems faced with current methods. First, the decision-making environment of a farm will be described. Then, the potential for computer-aided decision support in farm environment is discussed. As a practical example, a prototype of an expert system that serves as a front end to a LP-based whole farm planning method will be developed and evaluated.

2. The decision-making environment on a farm

An accurate definition of farm management can be found, for example, in Kay's (1986) text book: "Farm management is the decision-making process whereby limited resources are allocated to a number of production alternatives to organize and operate the business in such a way as to attain some objectives." This definition identifies most of the characteristics of the management activities found on farms. Farm management can be thought of, then, as being a decision-making process. The decisions are concerned with allocating the limited resources of land, labor, and capital among alternative and competing uses.

Farm management is becoming an increasingly complex task. Consider, for example, the development of new technology and the many changes which have taken place in tractors, machinery, and other equipment. Therefore, farm management is a continual decision process because of the continual changes taking place in global and national economy, and in an individual business. As new changes come about, farm managers must be prepared to identify them and make the correct decisions in response. Without a timely and correct response, a manager cannot expect to survive in a dynamic economy.

Weather, insects, diseases, and government policies are examples of factors which place the farm manager in a position of making decisions in an environment of risk and uncertainty. While there is a certain amount of risk and uncertainty in every business endeavor, the farm manager's decision-making environment contains more factors creating risk and uncertainty than most types of business. The unpredictable nature of the farming environment is difficult to control and assess. This explains why tradition, heuristics, and common sense have always had such a big role in farming decisions (Gauthier & Kok 1989).

3. The potential for computer-aided decision support

According to Gauthier and Kok (1989), recent developments in microelectronics and software technology have the potential to greatly increase the capabilities and popularity of computer-based methods to address questions that rise in everyday farm management. Modern computer systems are able to monitor and regulate a wide variety of farm events and processes, supervise the majority of standard farming operations, and especially, help farm executives make better decisions. At the strategic level, decision support systems (DSS) can be useful in areas such as machinery selection and sizing, planning of crop and timber rotations, feeding or breeding programs, scheduling and design of land improvement work, financial planning, marketing, and so on. These tools could be used, for example, to help determine which crops to plant and where to plant them, what equipment or land to purchase and when to do this, how much milk, eggs or livestock should be produced and when, whether diversification of the enterprise might be profitable, etc.

In farm management, many decision-making processes rely substantially on heuristics and on information that is incomplete or which has certain level of associated fuzziness or uncertainty. The ability to use heuristics has traditionally been a characteristic of human intelligences, although computer programs that can represent and use heuristics are becoming more common (Harmon et al. 1988). Adopting computerized DSSs constitutes an intermediate step in the progress towards the complete automation of a control process by keeping the final decision-making authority in the hands of the farm manager. According to Gauthier and Kok (1989), this transitional stage will be needed for as long as the knowledge and procedures involved in making a decision are not sufficiently well understood and until the cognitive structures and control regimes used by intelligent artifacts become much more sophisticated.

As mentioned in introduction, the utilization of current farm management problem solving techniques has suffered from the complexity of the applications. A number of models have been developed within academic and research organizations that have not been used to their potential because they are difficult to apply. As a result, several scientists are now integrating knowledge-based system (KBS) technology with conventional problem solving techniques in order to increase the robustness and usability of their systems (e.g. Binbasioglu & Jarke 1986, Pai-Chun et al. 1987, Gauthier & Kok 1989, Childers 1990, and Tucker 1990).

In all, agriculture is seen as an area of enormous potential for applications of integrated knowledge-based systems and conventional technologies. For example, excellent databases are available for information ranging from historical weather data to individual dairy cow records. Complex simulations have been developed to describe phenomena ranging from plant growth to economic systems. Such investments are a valuable asset as knowledge sources for knowledge-based decision making.

4. An example of an expert system assisting LP-formulation

4.1 Background

In Finland, most agricultural products have been overproduced since early 1970's (Kettunen 1990). The Government has set quotas, which along with blocked markets make it difficult for farmers to maintain an adequate level of income with current production lines. Most farmers must consider the possibilities to restructure their production. In many cases, the problem could be resolved just by changing the emphasis to field crops other than those being cultivated now, or by putting more emphasis on forestry. Thus, the need for whole farm planning has become more and more obvious.

In 1988, a linear-programming based model for whole farm planning was developed at the University of Joensuu, Finland (Hyttinen 1988). The model was named KOTKA (= Eagle, an abbreviation of Finnish words 'Koko tilan kalkyyli' which stand for 'Whole farm calculation' in English). The KOTKA software runs on IBM PC/AT and compatibles, and it can be used on an individual farm to determine the optimal level of specialization in different production alternatives.

4.2 Problems with the LP-based system

Some problems, however, have arisen during the test use of the KOTKA model. The bottleneck has been the determination of all the possible production alternatives relevant on a farm in question. It has proven to be too intricate and time consuming. There is a wide variety of variables to be taken into account in order to be capable of enlisting all the alternatives. The first group of these variables refers to the location of the farm, the second to the type of soil, the third to the farmer himself, the fourth to the external environment, and so on.

In addition, it has been difficult to set the constraints to the LP-model. Although a certain crop may be able to be cultivated, the maximum area of it may be limited by various types of restrictions. The availability of labor and machinery as well as other resources may form limits to the production. Basically, these restrictions are easy to handle within the LP-model, but the problem is that they may be difficult to recognize when putting data into the model. Finding out all the special cases of legislation, restrictions, quotas, subsidies, and so on takes time and tries patience. On an average farm, the size of the LP problem can easily grow to a 1,000 by 500 matrix. Thus, human errors are not uncommon.

Another difficult problem area refers to the complexity of the farm manager's decision making. Although the decision making problem in question is basically economic in nature, the problem includes much more than just economic aspects. The manager's background, attitudes, education and social pressures, among other things, seem to strongly affect the final decision of production alternatives. Many alternatives, even very profitable ones, may be out of question due to these non-monetary variables. Furthermore, farmers may have certain preferences, and they conduct the searching process so that only the alternatives they subjectively prefer will be considered as relevant alternatives. So, many reasonable alternatives can be out of evaluation due to this misleading way of thinking. Farmers, like most people in general, are incrementalists: they resist big changes.

4.3 Proposed solution: A front-end expert system

These problems could be relieved integrating an expert system with the KOTKA model. As mentioned earlier, it is assumed this kind of integration can make existing conventional applications more accessible. In this study, the assumption was tested by trying to solve a part of the problems faced with the KOTKA model by building a prototype of an expert system to produce alternatives and constraints to the LP model. The prototype system recommends the potential crop alternatives and sets acreage constraints for each crop. The expert system was developed for IBM PC/ATs or compatibles. CLIPS expert system language was used as a development tool. The details and features of CLIPS are described in Giarratano & Riley (1989). An end user interface developed by Engel et al. (1989) was used to make the software easy to use. The basic knowledge and procedures required to determine the production alternatives and constraints were acquired from text books on agricultural economics (e.g. Ryynänen & Pölkki 1984). Moreover, some domain experts in Finland were interviewed.

The following is a list of factors that must be taken into account when determining whether a crop could be cultivated. After each factor, the variables describing the factor are listed, and the explanation how these variables are handled in this prototype expert system:

1. Climate * The sum of thermic (over +5 C) temperatures, the length of growing season, and precipitation.

- In current version, the user have to give these. It would be possible to build a database of these data. In that case the user needs to give the coordinates only.

2. Soil * Quality class (includes nutrient content, particle-size distribution, etc.).
- The area in each quality class is asked.

3. Risks * Frost, flood, plant diseases, weed, pests, etc.
- In this prototype, only frost and flood are taken into account, in a simple and primitive way. Liability to weed is taken into account in soil quality classes. The real problem is with diseases and pests because they do not follow any rules, they just come and destroy the crop when they decide to do so.
- In developing this system, the problem in handling risks was how to find out the certainty factors or probabilities for each risk situation. Finally, the solution was to use probabilities and questions like "In how many years out of ten ..." The advantage of this approach is that this kind of questions is much easier to understand for a farm manager compared to question like "What is the probability of ..."
- The farm manager is asked to give his estimate of the probability of freezing and flooding during the growing season, as well as the lowest level of probability he accepts. The probabilities of receiving normal yield under these risk situations can be derived from field experiments.

4. Farmer * Knowledge and skills, the condition of health (e.g. the farmer may be allergic to some crop).
- Not handled in this version.

5. Social environment * Attitudes, trends and fads.
- Not handled in this version. On the other hand, the use of this ES prevents some of the negative effects of these factors -> no alternatives are left out for these questionable reasons.

6. Government policies * Legislation, quotas and restrictions.
- Agreements on restriction of production can be handled with this system.

In the prototype version, only the most common crops and their varieties are treated as alternatives. As a result of running this expert system, the potential crops will be listed for each quality classes of arable land. Each of these crop alternatives may have an upper and/or a lower area limit. In current version, the output information is stored into a file that has to be converted to an MPS input file before running the linear programming part of the whole farm planning system.

5. Evaluation of the approach

When fully developed, this expert system would be of great help in using the KOTKA whole farm planning model. To be a complete prephase in running the KOTKA whole farm planning model, all the potential production alternatives in crop cultivation, in animal husbandry, in forestry, and in other possible sources of living on a farm should be included in this expert system.

This approach has several advantages compared to the situation in which the potential crops are searched through tedious and time consuming thinking process one by one. By using this system, it would be more likely that all the relevant - and only the relevant - crop alternatives are taken into account. Moreover, this ES speeds up the LP process, because all the irrelevant alternatives are left out. Furthermore, the probability to get irrelevant outcomes - for the reason that the user has forgot to pay attention to some significant factor - and along with it, the need for reruns of the model decreases.

References

Binbasioglu, M., and Jarke, M. 1986. Domain specific DSS tools for knowledge-based model building. Decision Support Systems. Elsevier Science Publishers BV. 2:213-223.

Childers, R.E. 1990. GOSSYM-COMAX cotton management for the 90's. Project report. Texas Agricultural Extension Service. College Station, TX. 5 p.

Engel, B.A., Jones, D.D., Rewerts, C.C., and Rogers, J.B. 1989. Expert system development environment. An ASAE Meeting Paper No. 89-3513. 4 p.

Gauthier, L., and Kok, R. 1989. Integrated farm control software: Functional requirements and basic design criteria. AI Applications in Natural Resource Management 3(1):27-37.

Giarratano, J.C. and Riley, G.D. 1989. Expert systems: principles and programming. PWS-KENT Publishing Company. Boston. 632 p.

Harmon, P., Maus, R., and Morrisey, W. 1988. Expert systems tools and applications. John Wiley and Sons, New York.

Hyttinen, P. 1988. Using linear programming in planning the management of combined forestry and agricultural firm (in Finnish). A thesis for the degree of Licentiate in Agriculture and Forestry. University of Joensuu, Finland. 132 p.

Kay, R.D. 1986. Farm management: Planning, control and implementation. Second edition. McGraw-Hill, Inc. New York. 370 p.

Kettunen, L. 1990. Finnish agriculture in 1989. Publications of the Agricultural Economics Research Institute no. 60a. Helsinki. 52 p.

Pai-Chun, M., Myrphy, H.M., and Stohr, E.A. 1987. Computer-assisted formulation of linear programs. IMA Journal of Mathematics in Management. Oxford University Press, Inc. 1:147-161.

Ryynänen, V., and Pölkki, J. 1984. Agricultural Economics (in Finnish). The Department of Agriculture and Forestry. University of Helsinki. 355 p.

Tucker, T., Shannon, R., and Lovellette, R. 1990. A comparison of the development methodologies between conventional systems and expert systems. AI Applications in Natural Resource Management 4(2):27-32.

Whittaker, A.D. and Thieme, R.H. 1989. Integration of expert systems and conventional problem solving techniques in agriculture. AI Applications in Natural Resource Management 3(1):49-52.

Hakapaino Oy, Helsinki 1992