ACHIEVING STRUCTURAL SENSITIVITY BY AUTOMATIC SIMPLIFICATION

Raimo Keloharju, The Finnish Academy of Science, Ari Luostarinen, Hewlett-Packard, Finland.

Raimo Keloharju is Associate Professor of Management Systems at the Helsinki School of Economics. He is on leave 1983-84 to work on a Finnish Academy of Science project "Living in the Information Society". He received a BSc(Eng) from Helsinki Institute of Technology in 1952, a MBA from Indiana University in 1960 and a DBA from the Helsinki School of Economics in 1968.

Ari Luostarinen is a field engineer at Hewlett-Packard in Finland. Before that he worked as the head of system design for the Computing Centre at the Helsinki School of Economics. He received a BSc in Computer Science from the Helsinki University in 1977 and a MBA from the Helsinki School of Economics in 1983.

**The authors gratefully acknowledge helpful comments from Dr. R G Coyle on some of the ideas given in this paper.*

ABSTRACT

This paper describes how redundant relations from a system dynamics model can be removed by using a special purpose software. The new methodology developed was applied to a recent publication of a forest study.

1. INTRODUCTION

It is generally assumed by system dynamicists that there are two major problems with system dynamics methodology. According to Legasto and Macariello (1980),[1] they are:

- the enormity and laboriousness of the sensitivity analysis effort
- the lack of an objective function by which alternative policies could be evaluated.

As far as the first problem is concerned, system dynamicists already do sensitivity analysis but deal only with changes in individual parameters. To quote Starr[2]:

"Changes in groups would introduce an overwhelming number of combinations and hence are never done on an all inclusive basis".

Explicit objective functions are not commonly used in SD although there is some support for the idea. For example, Sharp[3] has argued as to why objective functions should be used and Keloharju [4,5,6] has worked with them.

We can, therefore, see that only the first problem of Legasto and Macariello is real; the second problem has been solved but it seems that the solution is not widely known. What is perhaps more important is that the problems are interrelated: the objective function allows a kind of sensitivity analysis which is not piecemeal. But there is now another theoretical problem since use of the objective function requires optimization and initiates a series of methodological changes. In this paper we shall try to explain how this analysis can be performed.

Structural sensitivity by automatic simplification, which is our main concern, came only after a very long and arduous period of research. The tracing of this technical development process requires a whole book and it has been reported elsewhere[7]. Here we shall take the methodology as a black-box, but concentrate on the theory and an application example. We simplify for two reasons: To understand a model and to carry out structural sensitivity analysis.

To understand in non-technical terms what simplification is all about, let us start from optimization.

In order to optimize a model, some objective function has to be proposed. In ordinary optimization, the objective function points model behaviour in some acceptable direction. In simplification, the model behaviour should remain unchanged even though the model gets simpler. If that happens, the parts that have been removed are redundant.

When system dynamicists do sensitivity analysis, they change model parameters one by one. This can be based on trial and error or on the known formal properties of the system equations. The trial and error procedure can be made systematic by having the objective function guide the search.

Optimization can be done in parameter space with some model parameters being treated as variables that have to be changed. We can go further by adding new artificial parameters to model equations as additional multipliers on structural relations. Model structure is then transformed into a variable.

Additional parameters have to be added to the simulation model, and in such a way that they control the contribution of some model parts which have been chosen intuitively. Using these parameters the modeler makes hypotheses about the importance of various model parts. For example, that a rate depends on intuitively chosen functions of two or more levels. Optimization then determines the importance of each relation in the final (optimized) rate formulation. In order to solve the problem technically, some heuristic optimization algorithm is needed.

So far we have treated the optimization process as a full sampling of parameters; all those parameters which had to be manipulated were chosen simultaneously. Could we do this more economically? At least we could try by adopting some sequential selection plan where the same parameter may be used more than once. It is important that a potential selection rule may use any kind of information from the modeling process. The rules draw on the resources of theory, intuition and imagination.

The purpose of the selection rule is to remove those parts of the model structure which are not needed to generate the reference behaviour. A step-by-step procedure similar to 'inverted' stepwise multiple regression-analysis is used but now it is applied to feedback models in a somewhat heuristic way. When simulation of the reduced model shows that you are still reproducing the behaviour adequately, the sample-size may be reduced.

Continue to decrease the number of active parameters as long as the results from the selection are acceptable. What you will finally receive is a model which has the simplest acceptable structure, and even more important: one that the computer has found for you.

A colleague of ours expressed the same in a slightly different way by saying: "It seems to me that you keep throwing out parts from your radio receiver until the sound disappears. Then you pick up the last part and that magic box plays again".

2. THEORY

The quality of a model is related to model acceptability and again it is based on two yardsticks: the model itself and model output. As the model simplifies reality, three types of errors are likely to occur:

(a) some model relations do not agree with their real world counterparts

(b) some necessary relations are missing

(c) some relations are redundant

In the SD approach, points (a) and (b) are dealt with explicitly. Point (a) is relatively easy to deal with by observing, inquiring and discussing. Model output is used to make sure that point (b) does not cause any problems; if the dynamic hypothesis is wrong, the model does not behave as it should. Finally, all agree that the model should be as simple as possible. However, so far there has been no formal measure to cover point (c).

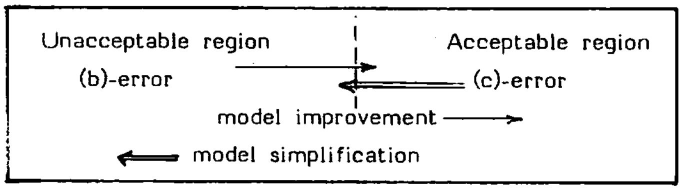

Figure 1: The improvement and simplification contrasted

Redundancy means that a model has casual relationships which are not needed. When a model is being developed, it is very difficult and time consuming to test which model elements really are important. Therefore, the finished product usually has some redundancy although system dybamicists favour model simplicity, at least in principle. A reduction in redundancy means that model structure gets simpler.

Is redundancy then a fault if it does not affect model behaviour? The answer is found by returning to the opening sentence above: The quality of a model is related to model acceptability . . . Therefore the right question to ask now is, is redundancy acceptable?

The users want to understand why the model behaves as it does. That means that, at least from the modeler's standpoint, (c) is a flaw; he needs to know the cimplest model that describes the problem of his client. Nevertheless, it is still possible that the modeler prefers a discussion with the user in terms of another model-version, which has redundancy.

Then he would have a large, redundant, public model, and a small efficient, private model, which is where the analysis is really done. Suppose now that we have some means available to simplify a model. How could we know that the simplification we made is justifiable? The answer is seen when we return to the errors (b) and (c) above. They belong to the same 'deviation' category, indicating that the model contains either 'too little' or 'too much'. Therefore the occurrence

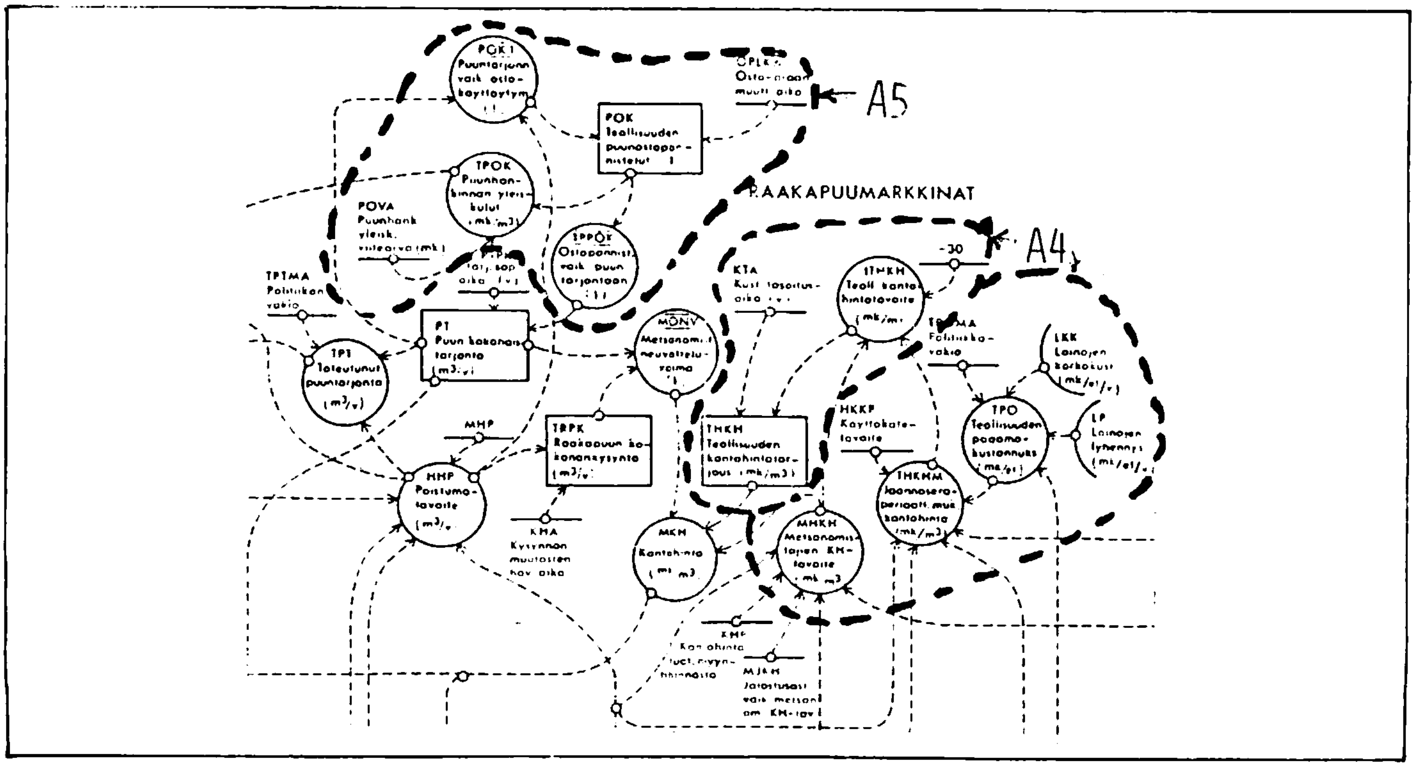

Figure 2: The forest-reserves sector

of both types of error should be evaluated in the same way, i.e. from model behaviour.

When a model is being constructed, the process continues until the model behaves in a proper way. When the model is being simplified it behaves in a proper way as long as the simplifications are justifiable. Figure 1 illustrates the relationship of errors (b) and (c).

As we now have a clear meaning of how simplification is related to improvement, let us proceed still further. Figure 2 calls forth new questions such as

(1) How can we decide where to stop simplifying?

(2) Could we move the boundary between 'acceptable' and 'unacceptable'?

(3) Could we combine improvement and simplification?

(1) The boundary has to be crossed both in improvement and in simplification in order to know where to stop. In simplification you have to return to the acceptable region. The problem is that you do not know which version is the last 'acceptable' before you found the first 'unacceptable'.

(2) The boundary cannot be moved when the model should reproduce some real world behaviour. Since, if you moved it, model behaviour would be wrong.

When the purpose is to improve the current situation, the boundary becomes more flexible. Increased co-operation and understanding between the user and the modeler is now possible because the new tool is flexible. The acceptable area of operation is likely to grow as understanding deepens.

In figure 1 we tried to avoid either (b)-error or (c)-error. But the model construction may proceed recursively by first focusing on error (b), then focusing on error (c), again on error (b), etc. The information that is used to remove the (c)-error should also guide is in the (b)-stage that follows. This is possible when the information is related to structural sensitivity of the model.

(3) The quality of the model depends on its behaviour and its simplicity. We can thus improve the model by simplification when the proper behaviour has already been found, or we can improve the behaviour and at the same time simplify it.

Sensitivity analysis shows how the model responds to changes. On the basis of this we can classify into four groups:

(1) *Robust* models respond optimally to major changes

(2) *Vulnerable* models may perform optimally for some, but not all major changes

(3) *Tolerant* models ignore minor changes, e.g. noise in demand

(4) *Sensitive* models respond even to minor changes.

Although robustness is not an operational concept, it is the goal for all modeling efforts in SD. In optimization, the aspiration level of the modeler is lower as the target is now to construct vulnerable models. Vulnerability is an operational concept.

There is a philosophical difference between robustness and vulnerability. In both cases, the future is assumed to be unpredictable and it is assumed that things will go wrong. However the nature of the response is different. Builders of robust models believe that a model can be robust and they try to find that model. Builders of vulnerable models believe that the model is only an imperfect estimate of some unpredictable process and, therefore, the model should change whenever new information from the process suggests it. Besides ex-ante information, the latter group uses ex-post information for model updating. They are Bayesians in spirit but system dynamicists in practice.

3. APPLICATION – EXAMPLE

DYSMOD software (Dynamic Simulation Model Optimizer and Developer), which was developed at the Helsinki School of Economics in 1979-81, extends the use of Dysmap to heuristic optimization by simulation[8]. But it also provides an opportunity for some advanced model development procedures, such as model simplification, and this is done so as to decrease model opaqueness.

A recent simulation study was chosen to test the simplification idea in practice. We decided to simplify a model which was developed by H. Seppala, J. Kuuluvainen and R. Seppala and which was published in a book named "The Finnish Forest Sector at the Crossroads"[9] (in Finnish). The book reports on the development of a medium-size SD-model and certain experiments made using the model.

In this study, the reference behaviour was based on the following variables:[10]

- the labor supply in the Finnish forest economy
- the forest balance, i.e. the difference between harvests and forest growth
- the production of the forest industry
- the loan percentage from turnover in the forest industry

The model was first run in the simulation mode to find the reference behaviour for measurement purposes. This reference behaviour from the original simulation model was approximated with four table-functions with time as the independent variable. The table-functions were then matched with the points of 0, 20, 40 and 60 years. For each of the four variables, the cumulative sum of the squared deviation from the reference mode was calculated as a percentage of the reference mode value. It was defined as a percentage percentage because some tested variables were different orders of magnitude. The piecewise linear approximation was thus quite rough but it worked. The value of the objective function at time 60 was 0.185, resulting from linearization error.

The sum of the four sums was then used as the objective function to be minimized. The program was written in Fortran and was attached as a subprogram to the model.

The estimated social significance was the main criterion in choosing parts for decoupling in simplification. The simplicity of the technical solution also affected the choice.

The model has 9 simplification parameters. Each of them either retains or removes some part of the model, depending on the value the parameter receives in the simplification process.

- L1 is related to conditions of loan-granting. Two mutually exclusive alternatives are considered: quarantees and profitability.
- A1 is related to structural conditions of raw material supply
- A2 determines how forest ownership affects forest supply. The forest is owned either by farmers or investors.
- A3 is the sales price of the final product
- A4 controls the pricing mechanism in the raw material market. It covers the full range from the buyer's to the seller's market.
- A5 controls the effect of a supply deficit in the supply on buying behaviour. The relationship was delayed in the original model. Here the value of one for A5 makes the delay-time infinite and, therefore, cuts off the connection.
- A6 determines the effect of the depreciation life time on production costs.
- There is a delay between the change in the operating cash margin and the depreciation period of the manufacturing capacity. A7 controls the delay-time.
- A8 allows us to examine whether the model is sensitive to an increased use of machines in harvesting or not. Again the delay-time is the focus of control.

Figure 2 is from the book mentioned above. The test is in Finnish but it is of no great importance as we are not going to discuss the details. The figure shows the model sector which describes the forest-reserves market. The dotted areas were added afterwards and they show those parts of the model which were chosen for simplification.

The top part is removed if simplification parameter A5 receives the value of zero. The lower part consists of two parts but only one of them may be removed. The choice depends on whether A4 receives the value of one or zero.

The computer needs some information for simplification and this information is given interactively from a terminal. The interactive program first requests the basic data that is needed in optimization, information such as the objective function and the optimization parameters with their ranges. The parameters are of two types: ordinary optimization parameters and simplification parameters.

Then the computer asks questions which are related to the guidance of the optimization process, i.e. the length of simulation, the number of interactions and the calibration of the optimization algorithm.

The simplifier subsystem of DYSMOD now becomes active. In ordinary optimization, the optimization algorithm treats the simulation model as a subsystem. The 'simplifier' treats both of them as its subsystem.

The simplifier asks certain questions so as to get the information it needs for self-regulation. The simplification process proceeds step by step. Each step consists of optimization and of an estimation concerning the significance of the parameters, of sorting out the parameters into a ranking order on the basis of their significance and, finally, of decreasing the group of optimization parameters by one.

At the end of each step, the computer prints out some information. Figure 3 gives an example of such a print-out.

```
90 · .155E+00  .251E+00 .887E+00 .503E+00 .895E+00 .244E+00 .500E+00
               .500E+00 .250E+00 .504E+01 .250E+00 .271E+00 .100E+02
               A6       A1       A2       Y        A3       L1
               A4       A5       TRSA     A7       A8       LA
91   .142E+00  .252E+00 .916E+00 .500E+00 .896E+00 .248E+00 .500E+00
               .500E+00 .250E+00 .502E+01 .250E+00 .284E+00 .100E+02
A1     ROUNDED      .916 --> 1.000
THE SIMPLIFYING PROCESS WILL CONTINUE WITH
THE FOLLOWING VALUES :
A4       =  .5000
A2       =  .5000
L1       =  .5000
A6       =  .2525
A5       =  .2500
A3       =  .2475
A7       =  .2500
A8       =  .2837
A1       = 1.0000   ( OUTSIDE THE BASE )
THE FIRST SIMULATION RUN :
               A6       A2       Y        A3       L1       A4
               A5       TRSA     A7       A8       LA
91   .142E+00  .252E+00 .500E+00 .896E+00 .248E+00 .500E+00 .500E+00
               .250E+00 .502E+01 .250E+00 .284E+00 .100E+02
TYPE 1 IF YOU WANT TO PLOT THE RUN
```

Figure 3: The report from a simplification step

In the top part of figure 3, there is some information from the last optimization round taken before the simplification starts. The numbers from left to right refer to the iteration rounds, to the value of the objective function and to the values of the optimization parameters. The last iteration round, which has a heading showing the names of the parameters, gives the best value of the objective function. The rounding-off information follows the last iteration round.

The list of new parameter values is very useful because parameters have been listed in order of their significance. For example, A4 is the most important simplification parameter when the criteria is a change in the value of the objective function. When the simplification procedure has ended, the ranking list of the remaining parameters suggests where the model should be enlarged.

The modeler also receives the plotting of the graphs after each simplification step. By comparing plottings from consecutive steps, he will see where to stop the simplification procedure.

The model proved to be very insensitive to most of the simplification parameters as the model behaviour practically unchanged. A noticeable change occured, however, when A4 was round off to one. Figures 4 and 5 show the results.

The plots describe the following variables:

- labor supply (T)
- forest balance (*)
- production (B)
- loan percentage (L)

Parameter A4 controls the pricing mechanism in the raw material market. In figure 5, parameter A4 obtains the value of one, indicating that there is a complete seller's market. This involves heavy increases in raw material price and, therefore,

- raw material purchases slow down. As a result the forest balance is disturbed
- liquidity deteriorates as shown by the increased share of loans.

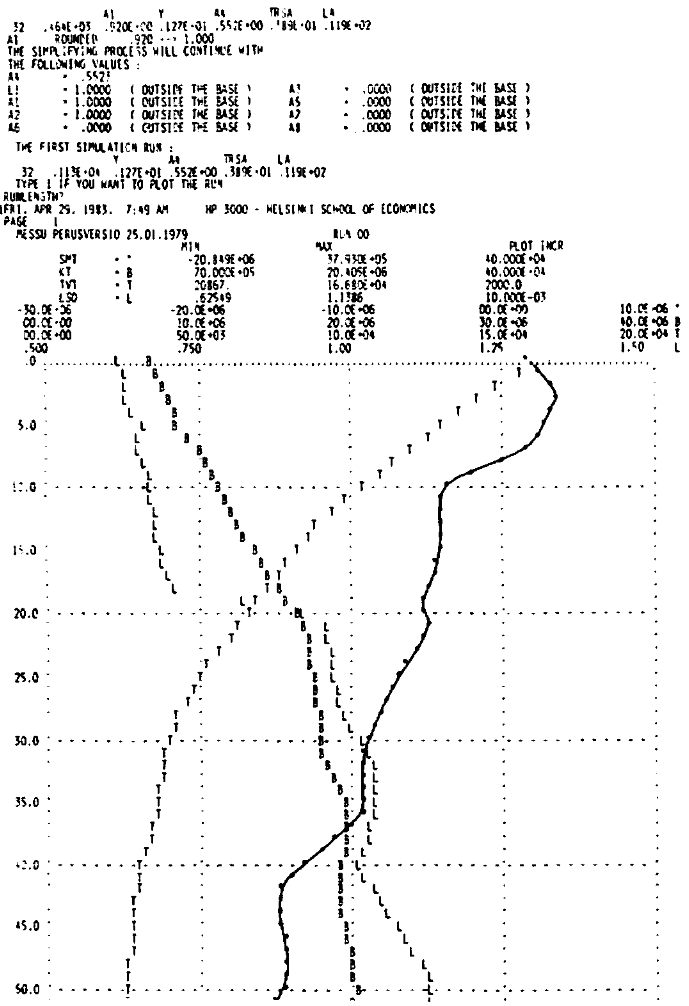

Figure 4: Model behaviour before A4 was rounded off to zero

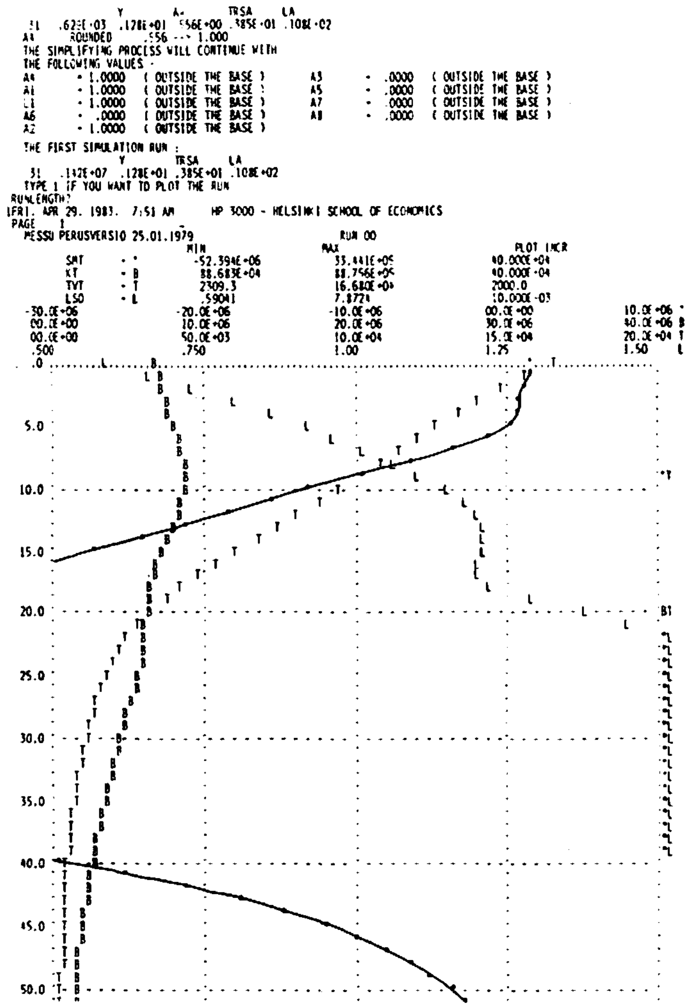

Figure 5: Model behaviour when A4 was rounded off to zero

4. DISCUSSION OF RESULTS

The comparison of parameter rankings from all simplification steps showed that the significance of each parameter remained generally quite stable during the simplification process. Parameter A1 was the only exception.

Figure 6 summarizes the roundings off and the corresponding values of the objective function. The value of the objective function summarizes the information from the model behaviour to the computer. The modeler bases his simplification decisions on more complete information as seen in figure 4 and 5.

The comparison in figure 4 and 5 showed that the model behaviour changed significantly when parameter A4 was rounded off to one at simplification step 9. Therefore the modeler now knows what parts of the model are redundant; all those parts that had been removed from the model by the simplifier. For example, A5 was rounded off to zero at the seventh simplification step. The dotted area marked by 'A5' thus disappears from the model.

The study showed that A4 was the most important parameter in sensitivity. This means that by including a more detailed analysis of the alternatives available in price negotiations, the model might generate very valuable information for decision makers. For methodological reasons the original simulation study did not lead to this recommendation.

The simplification task could have been accomplished interactively. However, the size of the task was quite big for a HP 3000 series computer. That is why the runs were made on a batch-basis overnight. The whole simplification procedure required about 3 hours cpu time.

1.	A1	0.916 ---> 1	.142E+00 ---> .142E+00
2.	A8	0.200 ---> 0	
	A2	0.750 ---> 1	.120E+00 ---> .677E+01
	A1	1.000 <--- 1	
3.	A1	0.998 ---> 1	.717E+00 ---> .120E+01
4.	A3	0.245 ---> 0	
	A7	0.253 ---> 0	.773E+00 ---> .729E+03
	A1	1.000 <--- 1	
5.	A1	1.000 ---> 0	.262E+00 ---> .263E+03
6.	L1	0.517 ---> 1	
	A6	0.399 ---> 0	.222E+00 ---> .336E+04
	A1	1.000 <--- 1	
7.	A5	0.275 ---> 0	.749E+03 ---> .240E+04
8.	A1	0.920 ---> 1	.464E+03 ---> .113E+04
9.	A4	0.556 ---> 1	.629E+03 ---> .142E+07

Figure 6: Numerical data from all simplification steps

CONCLUSIONS

This study was the first attempt to simplify an SD-model automatically via heuristic optimization. The results clearly show that automatic simplification is a feasible approach and adds a new dimension to the tools available for the use of the system dynamicist.

But when does a simplified model start behaving differently? Perhaps the only safe way to answer the question is to say that it depends on the purpose of the model. If this is true, simplification ends where modeling begins. Again one feedback loop has been closed.

REFERENCES

1. LEGASTO, Augusto A. and MACARIELLO, Joseph (1980), *System Dynamics: A Critical Review*. TIMS Studies in the Management Sciences, Vol. 14.

2. STARR, Patrick J. (1980), *Modeling Issues and Decisions in System Dynamics*. TIMS Studies in the Management Sciences, Vol. 14.

3. SHARP, J.A. (1979), *Optimal Control Theory as a Framework for the Interpretation of System Dynamics*. Dynamica, Vol. 4, Part 3.

4. KELOHARJU, Raimo (1977), *Multi-objectice Decision Models in System Dynamics*. Dynamica, Vol. 3, Part I and II.

5. KELOHARJU, Raimo (1978), *System Dynamics or Super Dynamics?* Dynamica, Vol. 4, Part I.

6. KELOHARJU, Raimo (1980), *General Frame of Resources, Structure and Trade-off*. Dynamica, Vol. 6, Part I.

7. KELOHARJU, Raimo (1982), *Relatively Dynamics*, Second Revised Printing. Helsinki School of Economics, Working Paper F-33.

8. LUOSTARINEN, Ari, (edited by R.G. Coyle, 1982), *Dysmod User's Manual. Bradford University*.

9. SEPPALA, H., KUULUVAINEN, J. and SEPPALA, R. (1980), *The Finnish Forest Sector at The Crossroads* (in Finnish). Folia For. 434:1-122.

10; LUOSTARINEN, Ari (1981), *The Finnish Forest Sector Model as a Testing Ground for the simplification Method* (in Finniah). Unpublished MSc Thesis, Helsinki School of Economics.