MODEL RESTRUCTURING

Robert L. Eberlein
James H. Hines
Pugh-Roberts Associates

ABSTRACT

Models are continually changed and altered, and it is clear that more effective techniques for accomplishing this can make modeling easier and faster. One common way that models are changed is to remove selected pieces of structure from the model. This type of model alteration can be done for debugging a model, for tuning a model, and for performing policy analysis. In this paper we present techniques that automate the processes involved in removing components of model structure. By taking out some of the drudgery of changing a model, and guaranteeing consistency in the resulting altered model, the techniques greatly enhance modeling fluency and speed. These techniques are implemented in a software package that is part of Professional DYNAMO.

INTRODUCTION

Often in the process of model development, analysis, and presentation it is desirable to alter the feedback structure of a model. Such alterations can range from the addition or deletion of a few measurement equations, to the complete rebuilding of a model's feedback structure. Modification of structure is time consuming and error prone, In this paper we outline a procedure that allows more automated structural alteration when the goal is to remove or inactivate selected model components. The automated procedures save the modeler time in the initial alteration of structure, and allow easy replication of the restructuring process.

We begin the paper by reviewing typically encountered situations in which structural alteration can be useful, and the common features of these situations. We use these common features to characterize the information necessary to restructure a model, and the nature of the restructured model. Information is recast into the form necessary to use the restructuring program that is available as a supplement to Professional DYNAMO Plus. Following this, we present the theory of model restructuring more formally, discussing both the required information, and the algorithmic approach used to accomplishing restructuring. Finally, we present an example illustrating how the restructuring program described can be applied to an actual model.

THE STRUCTURE OF MODELS

It is true that structure determines behavior, and clearly modelers determine structure. Still, it is often the case that modelers are faced with the task of finding out why the structure they have created is behaving the way it is. Forrester (1961 Appendix N) describes a situation in which a number of decision makers, working to accommodate seasonality, in fact induce seasonality. Forrester (1971) describes why things often work counter-intuitively, and the importance of understanding surprising model behavior is widely recognized.

Homer (1983) describes techniques for dealing with the behavior of specific components or substructures in a model as a means of validation.

There are two important points that are common to the above issues. First, the behavior of individual structures within a model is not sufficient to determine the overall behavior of the model. Second, it is much easier to evaluate the behavior of an individual piece of structure than that of the full model. If surprising or irregular behavior can be traced to a specific component of the model structure, its validity can be judged in the context of that component of model structure.

There are three phases in the model development and analysis process where delineating key structures driving behavior is of great importance. The first, is at the initial model development stage, when it is necessary to just get the model working without numerical overflow. The second, is in the validation stage, where it is necessary to have the model produce good, or at least sensible, results. And the third phase, is in the understanding of what policies are likely to be useful and why. The issues are somewhat different, but there is a great deal of overlap.

Problems arising in the early stages of model construction are often very coarse in nature. A model may "blow up," with exponentially increasing variables causing simulations to abort early on. The sources of such behavior may be obvious (for example sign reversals in accounting identities) but they can still take time to track down. Problems cannot always be isolated when the bad output from a sector influences the rest of the model, because the input to the sector is also likely to show the problem behavior.

Finding the source of odd and inexplicable behavior can be accomplished by the successive removal of pieces of model structure, until the structure, or linking of structures, responsible for the problem is found. Structures and components within a model can be removed by making the key variables that feed back into the rest of the model exogenous, or in some cases constants. When this is done the output of different model sectors can be reviewed; if some outputs of a sector display problematic behavior, while the inputs to the sector seem okay, there is a problem in the sector.

To actually decouple elements of structure, variables that switch between endogenously generated values and exogenously specified values are often inserted into a model. The resulting equations tend to complicate the model and make maintenance more difficult. There is also a tendency to clutter the model with all the names of the exogenous variables and the associated switches that together simply act as a substitutes for endogenous variables. In writing the new equations all locations where switches and alternative variables are necessary must be identified. This is very time consuming and tends to interrupt the flow of tracking down behavior.

Closely related to this process of tracking down problematic model behavior is that of adjusting model parameters to attain acceptable historical fit. In the early stages of this process it is often desirable to reduce the amount of feedback to make the problem more tractable (Homer 1983). The difficulty in adjusting model parameters, with all model feedback active, arises for the same reason that problematic behavior is hard to trace. Historical inaccuracies in any variable are likely both caused by and causing inaccuracies in other variables. Adjusting a set of parameters appropriate to one or more variables may simply be compensating for the fact that there are other variables in the model that are not right. In such situations finding parameters that simultaneously make a large number of variables historically accurate can be very time consuming.

The solution to this problem of parameter adjustment is precisely that used for problematic behavior. Feedback can be cut out using switches that substitute exogenous values, usually historical values, for different variables. With these switches in place the modeler can work on individual components of model structure, As the components become acceptable the parameter values obtained can be incorporated into the full model, and feedback restored. The full model can then be used as the basis for final parameter adjustment which, normally, will be slight.

After problematic behavior has been removed from a model and the parameters have been adjusted to the satisfaction of the modeler the model should be an effective tool for analyzing and explaining problems and policies. Much of this understanding and explanation is in terms of fundamental feedback paths. Understanding comes about through working with the model and conducting a variety of different experiments. Typically, this experimentation involves the isolation of structures that are key to causing the problems, and making policies work. This can again be accomplished by deactivating selected structures with switches, where such switches are likely to be more selective than in the earlier modeling phases.

There are also more analytical ways of arriving at the fundamental elements of structure responsible for determining behavior. Perez-Arriaga (1981) outlines techniques for identifying important model levels, Forrester (1982) outlines techniques for determining important model causal loops. In this work there is a, possible implicit, removal of structure. The removal is not completely the same as that we are describing, though Eberlein (1986) applies our approach to restructuring in this context.

The common element in the above discussion is the alteration and selective removal of feedback paths. In small models this can often be easily accomplished by the addition of switches or the explicit alteration of structure. As models become larger, however, the difficulty of selectively removing structure increases very rapidly; the number of potentially altered variables as well as the number of times those variables are used becomes very burdensome. The partial removal of feedback through, for example, omission can also cause difficulties because the model will be different from what it was thought to be. For all sizes of models techniques that automate the alteration of model structure and enforce consistency have great potential benefits.

The cutting of model feedback links by the creation of switches does not necessarily take complete advantage of the less complicated feedback structure that results. If the full model is still retained for simulation purposes, even though many variables are being ignored, the computational requirements of simulation will be as high as for the original model. It is possible to make a model smaller by pulling out selected pieces entirely, however, the difficulty of fixing up loose ends and later putting the pieces back in, tends to make this impractical. Even when large quantities of structure have been made inactive, simulation can still be as time consuming as it was for the original model.

AUTOMATED RESTRUCTURING

The approaches used in altering the feedback structure of a model typically involve two steps: first, the identification of a set of structures or variables of particular interest, and second, the severing of feedback links. In automating restructuring we emulate this, working with groups of model variables to be retained, and groups to be made exogenous. In this section we specify the form and content of the information required for our automatic restructuring algorithm to work. The software we have developed then creates a restructured model based on this information. The discussion in this section is heuristic and explanatory, the formal theory behind the algorithms used to accomplish restructuring is presented in the next section. For a more detailed discussion of the actual procedures involved in performing a restructuring the reader is referred to the program documentation (Professional DYNAMO 1987).

In restructuring a model attention is focused on those elements causing trouble, or being compared to data, or of policy interest, that are closely related. Normally, variables that have short causal paths connecting them would be grouped together. In such a grouping a few key variables will be the most important to consider. Such variables are often those for which data are available, or generally important concepts for the system being modeled. In the approach to restructuring we use, these key variables will be the ones that need to be identified by the modeler. Subsidiary variables, closely related to the variables of interest, need not be specified; they will be kept automatically if needed.

In severing feedback links we work with the model variables, just as switches that move a variable from endogenous to exogenous do. Often, making a variable exogenous entails cutting more than one causal link, nonetheless, the conceptual process is still one of cutting links. The selection of variables to be made exogenous is, in some sense, the opposite of the selection of variables to focus on. Key variables outside of the area of focus, that influence the area of focus are clear candidates to be made exogenous.

The modeler needs to take a more active role in deciding what is to be made exogenous. While what is important to retain is usually quite clear, the feedback links existing between the unimportant elements and the important elements of a model can be subtle, and difficult to find. To effectively exclude elements of structure all of these feedback links need to be identified. The amount of structure to be retained is made larger by feedback. This is an advantage, in that important things will be kept, but a disadvantage in that unimportant things will also tend to be kept. It will often be necessary to make more than just a few important key variables exogenous.

Following the above logic, the inputs into a model restructuring are two lists, the first of variables to be retained in the restructured model and the second of variables to be set exogenous. The restructuring algorithms then determine the restructured model. We outline the steps involved.

First, and optionally, the modeler may wish to trace through the causal structure to find the largest possible list of exogenous variables. A variable is effectively exogenous if its value can be determined based solely on other exogenous variables. That is, a variable can be made exogenous if it is not involved in any feedback. When a modeler sets a variable to be exogenous, it is possible that other variables are effectively made exogenous. Depending on the model and purpose, setting all possible variables as exogenous may be desirable. This practice can reduce the size of a model by eliminating intermediate computations on exogenous variables. On the other hand, if a small number of exogenous variables drive many more, the required number of exogenous variable inputs to a model can be driven up. Under these circumstances, if the mechanism for automatically determining additional exogenous variables is invoked it can result in a larger list of variables to be set exogenous.

Given the lists of variables to be retained and variables to be set exogenous a new simpler restructured model can be assembled that contains only the variables, endogenous and exogenous, required to compute the retained variables. The restructuring starts with the list of variables to be retained and identifies all of the model variables involved in feedback loops with the retained variables. In addition, all exogenous variables necessary to compute the retained variables, and the variables on which they depend, are identified. Feedback links involving exogenous variables are not used in use in identifying the variables to retain. Rather, when an exogenous variable is encountered no further dependencies are considered. The resulting model contains only what is essential to computing the retained variable given values for the exogenous variables.[1]

The equations for all the kept endogenous variables are the same as those in the original model. The equations for the exogenous variables are no longer needed, since the variables will be input exogenously. Any other variables and their equations are simply dropped. The resulting model, with more limited feedback, serves the requirements of isolating the sources of dynamic behavior, and, because extraneous variables are simply dropped, the model size will normally decrease substantially. The model that results is a working DYNAMO model (in either compiled form or as a model listing) that can be used as any other model.

While the resulting model is definitely smaller, it is likely to have a number of exogenous variables that were not in the original model. In order to simulate the model values are required for these exogenous variables. The preferable source of these values will depend on the purpose of the restructuring. For the purposes of building and validating models the most appropriate source of exogenous variables is the available historical data. When historical data are not available, or it is not desirable to use them, the original model can be used to generate the exogenous variables. This can be done by simulating the original model and saving values for the needed exogenous variables. Finally, it will often be appropriate to set exogenous variables to some constant value when working with the restructured model.

There are a couple of points relative to arrays and macros that should be mentioned. The discussion in this paper is all stated in terms of variables, and if there are neither macros nor arrays in a model the meaning of this is clear. When there are arrays the entire array is treated as a single variables. Thus if one element of an array is kept, or made exogenous all are. Second, the nature of macros is to increase the number of variables that are hidden - those variables that exist and are used for computation but are never seen by the modeler. To be consistent with this hiding, macros, like arrays, are all or none. If something within a macro is kept, the whole macro is kept.

---

[1] The determination of initial conditions may involve the inclusion of additional variables as well as the alteration of some active equations to initial value equations.

The model resulting from the restructuring we have outlined has a minimal number of variables. For the purposes of computation this is preferable to a model that removes feedback but retains all variables. However, sometimes it is preferable to cut the feedback in a model while, at the same time, retaining all of the model variables. This permits the inspection of what the model would have produced for a variable's values. Most commonly, this would be useful when parameters are being adjusted on the basis of a variable, but that variables value is not desired to influence any other variables in the model.

To accommodate situations in which computed values are also desired we define an intermediate class of variables which we refer to as mixed variables. For the purposes of restructuring mixed variables are treated as endogenous variables. For the purposes of simulation all equations using a mixed variable use an exogenously supplied value, but the simulated value is created for review. Mixed variables have the further advantage that they can be applied selectively to elements of an array. This allows mixing of endogenous and exogenous impacts for a given subscripted variable.

The approach to restructuring we have outlined is straightforward and intuitive. The resulting models are consistent with Professional DYNAMO Plus and can be created in either compiled form or as a new set of model equations. The use of an algorithm in the creation of the models guarantees consistency, so that the models are always well formulated. Such an automated technique can substantially reduce the time needed to alter a models structure.

FORMAL THEORY

Intuitively the ideas behind restructuring a model are very straightforward. First determine all the variables that need to be kept, second determine what variables can be made exogenous and third find all the feedback loops that go through the variables to be kept but not the exogenous variables. The feedback loops found in this way are what need to be kept. If one goes to a stock and flow diagram and cuts and traces in the above manner the result will be a restructured model. However, short of actually going to a causal loop diagram, some formalism is required to state fully the techniques of restructuring. We present in this section a brief description of the algorithms used to accomplish restructuring. The discussion essentially repeats that of the previous section in a more formal manner.

In the presentation of the formalism we use x,y and z to represent any variables in a model. A name enclosed in curly braces (as in {RETAIN}) is used to denote a set. When the name of a set is in upper case it indicates the set is defined globally, lower case indicates a definition local to a function or operator. The symbol ∈ is used to indicate that a variable is an element of a set, while ∉ indicates that a variable is not an element of a set. ∪ indicates the union of two sets and ∩ the intersection. {a} - {b} indicates the intersection of {a} with the complement of {b}, that is, any elements of {b} that are in {a} are removed.

Prior to beginning the restructuring it is assumed the two sets {RETAIN} and {SET} have been defined: {RETAIN} consists of all variables that are to be retained, {SET} of all variables that are to be set exogenous.

We first define -> (arrow) as a boolean operator such that x -> y is true if there is a direct link from x to y and false otherwise. Note that x -> y does not imply y -> x, and will, in fact, rule this possibility out in many cases. The arrow operator could equivalently have been defined as true if x appears in the equation defining y, and as false otherwise.

The goal of the restructuring is to arrive at a new set

$$\{KEEP\} = \{y \mid y \rightarrow z_1,\ z_1 \rightarrow z_2 \ldots z_{n-1} \rightarrow z_n,\ z_n \rightarrow x,$$

$$x \in \{RETAIN\},\ z_1,\ldots,z_n \notin \{SET\}\} \qquad (1)$$

of variables that are kept in the restructured model and a second set

$$\{EXOG\} = \{KEEP\} \cap \{SET\}\ , \qquad (2)$$

of variables required as exogenous inputs into the model.

The set {KEEP} is clearly a finite set, but a practical way of arriving at it needs to be specified. We use the arrow operator in order to define the set valued function

$$\{define(\{list\})\} = \{z \mid z \rightarrow x,\ x \in \{list\}\ \}\ . \qquad (3)$$

{define({list})} gives the set of all variables that directly enter into the determination of every element of {list}. The {define()} function can, in turn, be used to set up the recursive function {include({add},{list})} with the following algorithm

Step 0: If {add} is empty go to step 5

Step 1: {current} = {define({add})} ∪ {list}

Step 2: {new} = {define({add})} − [{list} ∪ {SET}}]

Step 3: {include} = {include({new},{current})}

Step 4: finished

Step 5: {include} = {list}

Step 6: finished

The set {include} returned by this function is given in either step 3 or step 5. This set contains all of the variables that are needed to calculate {add}, plus all variables in {list}.

To see that {include()} will terminate after a finite number of recurs first note that if any variable x is an element of {define(add}} then x will be an element of {list} on successive recursions. Since there are only a finite number of variables and recursion continues only as long as {define(add)} is non-empty it follows that there can be only a finite number of recursions.

With the function {include} thus defined we can set

{KEEP} = {include({RETAIN},{RETAIN})} (4)

To show that this generates the right model we note that if y -> z and z ∈ {KEEP} either y ∈ {RETAIN} or y ∈ {SET} or because y ∈ define(z) at some recursion y ∈ {current} and therefore y ∈ {KEEP}. In particular, if y -> z_1, z_1 -> z_2, ... z_{n-1} -> z_n, z_n -> x, x ∈ {RETAIN}, and $z_1,\ldots,z_n \notin$ {SET}} then y ∈ {RETAIN}.

In discussing the procedures for restructuring a model we considered the possibility of making an exhaustive list of potentially exogenous variables. We do this using the recursive boolean function trace({check},{hold}) intended to determine whether a set of variables is completely exogenous.
trace({check},{hold}) is defined with the following algorithm

Step 1: {potential} = {check} ∪ {hold}

Step 2: {new} = {define({check})} - {SET}

Step 3: if {new} is empty go to Step 8

Step 4: {current} = {new} ∩ {hold}

Step 5: if {current} is empty go to step 10

Step 6: trace = FALSE

Step 7: finished

Step 8: trace = TRUE

Step 9: finished

Step 10: trace = trace({new},{potential})

Step 11: finished

To see that trace() will terminate after a finite time note that {hold} is a growing set that is bounded above by the total number of variables which is finite. If, in any recursion, the set does not actually increase in size the algorithm terminates because {new} is empty. Since {hold} is bounded in size and the recursion terminates as soon as it stops growing it follows that the recursion must terminate.

If there is a causal path from x to y Step 2 of the algorithm guarantees that the path will be followed to y. Feedback from x to y similarly would return to the point where x ∈ {define(z)} so that x ∈ {new}, but since x ∈ {hold} the routine will return FALSE. Conversely, if x is exogenous, it ultimately depends only on {SET} (and constants and possibly time) so that {new} will eventually be empty and the procedure will return true.

The function trace is inclusive in the sense that if any element of {check} is determined not to be exogenous all will. Thus it is necessary to invoke trace for every variable in a model individually in order to determine if the variable can be made exogenous.

We have set out a procedure for arriving at a restructured model given an initial collection of variables that can be made exogenous ({SET}) and variables that are important ({RETAIN}). So far we have only dealt with the actual variables to keep or discard, but the equations for the variables follow directly. If x ∈ {KEEP} and x is not exogenous then {define(x)} ∈ {KEEP} so that the original equation for x can be retained. If x has been made exogenous the equation for x can be discarded. Note that {define()} will be different if used for initialization equations then if used for active equations. Thus the active and initial condition inclusions can be determined separately by the execution of {include()} with two distinct {define()} functions. Mixed variables, as discussed in the previous section, do not require any alteration in feedback structure but only a rewriting of equations.

The above discussion outlines formally what is required in order to restructure a model. The algorithms, because they require little numerical computation, are quite fast. By construction the resulting model will be complete and well formulated. No simultaneous equations or indeterminate initial conditions will arise when the model is restructured. The resulting model can stand on its own as a model of part of a process. The existing exogenous variables only represent a redefinition of the model boundary.

APPLYING RESTRUCTURING

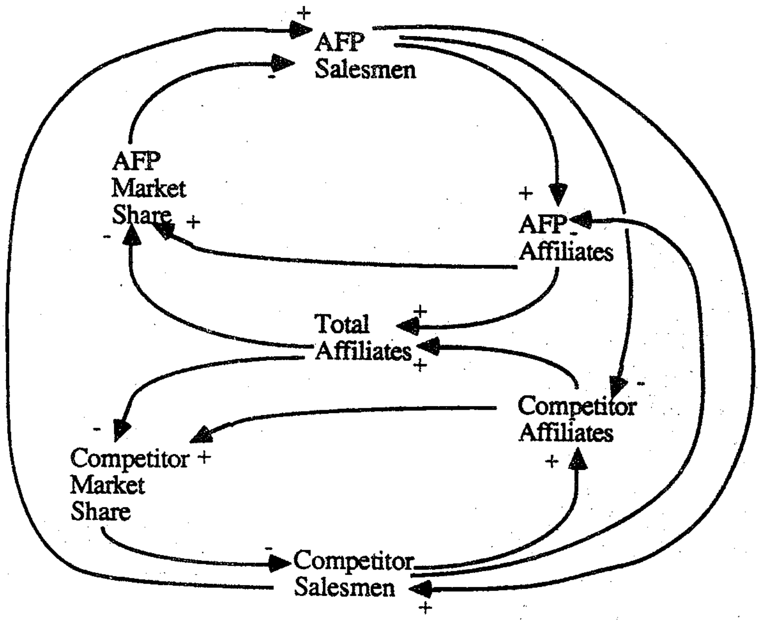
Automated restructuring is proving its worth in the modeling effort ar Pugh-Roberts Associates. A recent example serves to illustrate the power of the techniques. In late 1986 and early 1987 we developed a model for one of the principle pension fund managers in Chile's privatized social security system. Early in the validation process the model was tuned to historical data. Subsequent tests of the dynamic responses of the model, however, revealed that the formulation for salesmen was flawed. After rewriting the salesman equations, we were faced with the problem of retuning the model.

The relevant feedback structure of the model altered by the new equations is sketched in Figure 1. We use the fictional name "AFP" for our client in order to preserve confidentiality. Among the factors affecting the decision to hire salesmen is the market share of affiliates (workers affiliated with a particular pension fund manager) and the strength of competitors' sales forces. Sales force, in turn, affects the number of affiliates as well as salesmen hiring.

Affiliates and salesmen form a tight complex of feedback loops. Model output after the alteration of the sale force structure indicated substantial differences between the simulated and historical values for AFP's sales-force, the sales-force of competitors and the number of AFP affiliates. We felt that retuning to achieve a better fit would like yield more accurate parameter values. A better fit would also bolster confidence in both the structure and the new parameters.

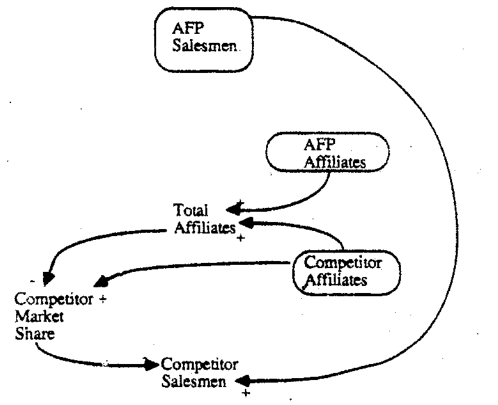
To retune we used the restructuring program now available in Professional DYNAMO Plus to break the structure into the three simpler ones illustrated in Figure 2. the first structure takes affiliates and AFP's salesmen as exogenous, making it straightforward to discover the parameters governing competitors' hiring decision. The second structure takes affiliates and competitors' salesmen as exogenous, permitting the easy uncovering of the parameters affecting AFP's hiring. Finally, AFP's and competitors' salesmen are taken as exogenous,

Figure 1

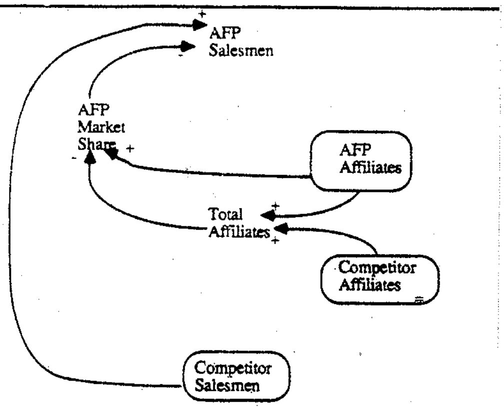


Relevant Structure for New Salesmen Equations

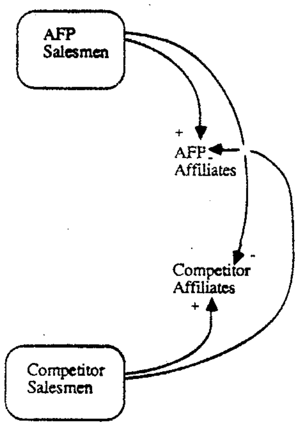
Figure 2



Structure 1: AFP Salesmen, AFP Affiliates and Competitor Affiliates are exogenous in order to tune the Competitor Salesmen decision.



Structure 2: AFP Affililiates, Competitor Affiliates, and Competitor Salesmen are exegonous in order to tune the AFP Salesmen decision.



Structure 3: AFP Salesmen and Competitor Salesmen are made exogenous in order to tune the response of Affiliates to Salesmen.

permitting us to get a fix on the parameters governing the affiliates' response to the activities of the salesmen.

Having decided how to break up the complex of feedback loops, we next chose sources for the exogenous inputs. We used historical data on affiliates and AFP's salesmen as the exogenous input to the first structure, achieving a very good fit for competitors' salesmen. Because the data on competitors' salesmen are yearly, and available only from 1984, we used the continuous output of the tuned first structure (beginning in 1982) as the exogenous input to the second structure. Finally, we used the continuous sales-force output from the first two structures as the exogenous input to the third structure in order to find values for parameters governing affiliates' response to salesmen.

Putting the parameters resulting from tuning the three individual structures resulted in a good fit for the full model, differing little from the fit for the individual structures. If we had tried to tune the full model all at once things would not have gone as smoothly. For example, putting in the correct parameter values for the competitors' salesmen decision without altering other parameters in the original model does not improve the fit over the untuned model. The set of parameters that would improve the fit using the full model would be different, and on tuning the other components of the model would need to be altered. Full model tuning would have been a repetitive and time consuming process of tuning and retuning the same parameters.

Cutting feedback loops and using exogenous data for tuning purposes got us a better fit faster. The restructuring program provided a quick easy means for breaking the model up into smaller pieces and, in combination with Professional DYNAMO's translation options, provided a quick means of putting in the proper exogenous data were they were needed.

CONCLUSIONS

Part of modeling is changing models. In this paper we have outlined a technique for accomplishing many common changes in an efficient and straightforward manner. The alteration of model structure can be accomplished automatically by selectively making variables exogenous, thus cutting feedback links. Because the approach guarantees the resulting model to be consistent, the potential for error is substantially reduced over less automated techniques. The techniques can be applied in the model building and validation stage as well as the policy analysis phase. In all cases the techniques serve to streamline many of the cumbersome alterations that might otherwise be employed. Through the use of an example we have illustrated some of the strengths of these techniques.

REFERENCES

Eberlein, R.L. (1986) "Identifying and Displaying Important Feedback Paths," The Proceedings of the 1986 International Conference of the System Dynamics Society, Seville, Spain

Forrester, J.W., (1961) Industrial Dynamics, The MIT Press, Cambridge, Mass.

Forrester, J.W., (1971) "Counterintuitive Behavior of Social Systems," Technology Review, 73 pp 52-68.

Forrester, N.B., (1982) A Dynamic Synthesis of Basic Macroeconomic Theory: Implications for Stabilization Policy Analysis, PhD Thesis, Sloan School of Management, MIT, Cambridge, Mass.

Homer, J.B. (1983), Partial Model Testing as a Validation tool for System Dynamics, The proceedings of the 1983 International System Dynamics Conference, Boston, Massachusetts pp 919-932

Perez-Arriage J.I. (1981), Selective Modal Analysis with Applications to Electric Power Systems, PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass.

Professional DYNAMO (1987), Supplement to Professional DYNAMO Plus Reference Manual, Pugh-Roberts Associates, Cambridge, Mass.

Richardson, G.P, and A.L. Pugh (1975) Introduction to System Dynamics Modeling With Dynamo, MIT Press, Cambridge, Mass.