"DYNAMICA" ISSN 0306-7564 Volume 4, Part 1

System Dynamics or Super Dynamics?

by

Dr. Raimo Keloharju
The Helsinki School of Economics

Abstract

In order to enable optimization of system dynamics models three hierarchies are proposed: I. Model, II. Optimizing procedures, III. (artificial) Intelligence. Using these levels two iterations are proposed. The optimization procedure of level II leads to a "small" iteration. When this has ended a transition is made via level III to other optimization procedures - "grand iteration". This hierarchial approach leads to a division of labour in system dynamics, as model structuring can now be made an automatic process.

Introduction

When Industrial Dynamics was invented it focused explicitly on rules underlying decision making processes. This was a true contribution to the art of Management Science, as managerial policy making instead of individual decisions could now be explored in normative environments.

Twenty years have now elapsed from those days but in the meantime control engineering principles of dynamicists have very effectively isolated them in a subculture, impenetrable to any outsider. The paradigm they have and the means for analysis they use have been adopted from control engineering, a black-box to a typical manager and to a non-specialist. One might ask, therefore, whether the price dynamicists have paid for their professional independence was too high or perhaps unnecessary.

System dynamicists have shown considerable willingness to apply their tool to a very broad range of social phenomena but astonishingly little effort has been given to modify the tool and paradigm. Potential explanations might vary depending on the inquirer's own role and biases. A social psychologist might refer to a weak group ego of the dynamicis a Marxist researcher to the power of the controlling class of control engineers, and a business economist to the leadership of the founder. My primary explanation is that outsiders from other scientific groups usually do not know system dynamics thoroughly enough to be able to change it in a way that commands acceptance or understanding. Insiders may know but still only recommend control engineering studies in order to convert the outsider to an insider. The tiny group of volunteers indicates, however, that interscientific barriers should be removed some other way.

One might also raise the question, why has an optimizing approach not been used more widely because Control Engineering background seems to be so common in this field. To reply in system dynamics terms I would say that the natural delays in the evolutionary process of system dynamics itself are quite long. It takes time to recognize that optimal control theory has developed and offers now new clues, it takes time to draw the necessary conclusions from this development and finally, it takes both time and resources to modify the tool.

A fresh and hopefully constructive approach might be to ask if the modelling process could be simplified. Why not hide, for example, some control engineering aspects from the user. If this proves to be a successful approach a non-specialist is able to use system dynamics effectively, releasing professionals to deal with problems of a higher order. Actually we have for long been ready for that but it requires a new kind of tool - more general and more versatile than before. Perspectives ahead of us will be demonstrated below and thid endogeneous growth process has no foreseeable end. As system dynamics in the old-established simulation sense is going to be just one extremely important part of a bigger whole, perhaps the word Super Dynamics might describe the new field lying ahead.

The Missing Link

A model builder without control engineering background probably feels that the new policy formation phase in system dynamics is difficult, confusing and extremely time-consuming at the best. Even with the loop-analysis method he is likely to get into trouble as "there is little in the way of explicit rules to guide one, except that increasing delay/or reducing gain are the key to improving stability, where that is a desirable system characteristic".[1]

Suppose for a moment, however, that we have an algorithmic method which automatically scans through all conceivable model structures and parameter values and then selects the 'best one' according to some measure. Some optimizing versions of Dynamo-language, e.g. SDRDYN are already available. SDRDYN has an SDR (Search Decision Rules) - heuristic algorithm as the main program allowing 'optimization' of a pre-selected set of model structures and parameter values after a process of iteration[2]. This approach as such does not work in real life applications, however, because of the number of parameters involved. It means that the model must be subdivided somehow.

To make the idea clearer let us see first what analogies might be drawn from the Simplex algorithm of Linear Programming, which changes a basis repetitively until an optimum solution has been found. A basis refers to those variables which have a non-zero value in the latest solution and therefore, are in an activated state. The objective function never deteriorates during the iteration process, which has been proved to be converging in L.P.

We would need an algorithm where each basis defines a model structure - parameter value combination that is being explored. At the same time there are nonbasic model structure and parameter value variables but they should retain the latest values from heuristical optimizing process in order to manage the procedure.

A small example will clarify the concepts of basic and non-basic in system dynamics. Let us suppose that parameters P1,P2 and P3 in the equation below have been defined as variables for the optimizing SDR-algorithm:

R RATE.KL=P1*AUX1.K+P2*AUX2.K+P3*AUX3.K

If P1 and P2 are in the basis the algorithm will only search various P1- and P2-values within some predefined upper and lower bounds, treating P3 as a constant. One possible basis change might then be to remove P2 from the basis, adding P3 instead. When P1 and P3 are being searched P2 would retain the latest value it has received.

The Simplex algorithm determines incoming and outgoing variables but an heuristic algorithm could be bu9lt to do the same. Actually there is even no reason to restrict the total number of changing variables to two. It can be guaranteed that no worsening of the attained objective function value will occur as the search algorithm always stores the best value so far found in a computer memory. In this way we have link connections to all important search results from the past.

In order to prevent confusion let us call a change-of-basis calculation is system dynamics as grand iteration. This is a heuristic procedure which, at least in principle, can make use of all information produced by the system before. Let us call that part of the model artificial intelligence algorithm as it is computerized and might have learning abilities.

An artificial intelligence algorithm defines the rules that change the basis. The incoming variable selection for a basis change operation might utilize an analogy from line-balancing in production planning. In one heuristic line-balancing algorithm the total work content will be assigned to various work stations by using random task selection. This procedure is then repeated several times using biased task selection based on earlier successes and failures in this experimentation process.[3] This kind of repetitive procedure might be useful also in selecting incoming variables in Super Dynamics. Figure 1 below compares both approaches:

Line Balancing	Super Dynamics	Comments
Task	Basic variable	
Order of task selection	Set of variables selected	Solution to be found
Number of work stations	Objective function related to time	Criterion used

Fig. 1. Comparison of some concepts in line-balancing and Super Dynamics

The outgoing variable(s) can be determined by examining 'historical' values of all variables during the iteration process of SDR-algorithm. As the algorithm tries to work in the order of diminishing returns those variables that are in good control are candidates for leaving the basis. This is opposite to statistical quality control philosophy where an out-of-control situation triggers action. Nevertheless, the notion of control limits might still be applicable.

One might also ask whether removal of variables under control works if the system variables interact strongly. Without further experimentation and research with different models nobody can give any definite answers to this question. Let us suppose, however, that we have a case where the removal idea fails. As the number of outgoing variables is by no means restricted to one, broadening the selection rule might help. Perhaps the new rule should treat strongly interacting variables as an aggregate in addition to the ordinary aspect of relative changes.

The goal of artificial intelligence is to replace human brain. One might temporarily work the other way round, however, to demonstrate the utility of presented ideas before commiting any new software changes. Some simple change-of-basis rules will now be selected and the experimenter executes then these rules from a time-sharing terminal. This means that he, instead of the artificial intelligence algorithm, provides all grand iterations.

The changing of a variable might be done at any time, or after some specific number of iterations. As the second alternative is easier, to follow, I think, we will proceed that way this time.

The relative change of each basis variable is the change in percentages from the initial value before the latest search process and should be calculated after a certain number of iterations. A small relative change indicates that a variable either is of secondary importance to the optimization process or already has a value close to 'optimum'. This makes it an outgoing variable candidate.

Both the objective function value and relative changes are important for grant iterations. The time for next grand iteration has arrived when the objective function value is becoming stabilized. An estimation error in this respect might effect both the final solution from the whole process as well as computation costs. The outgoing variable related decision is based on relative changes and only this aspect matters as the objective function value is now already given.

If the algorithm really chooses a variable it then moves to the group of non-basis variables, i.e., parameters. The simplest way of choosing incoming variables is to do it randomly but there are no technical limits - only economical ones - for sophistication.

For example, one might estimate the change in the objective function that results from changing each non-basic parameter by 1%. If there are n non-basic parameters, this requires n runs of the model. For small changes in the parameters, the change in the objective function is linear. The problem is now to pick the most promising candidates to enter the basis. The criterion is simply the maximum improvement to be expected in the objective function, assuming that linearity still holds over a wider range of parameter values.

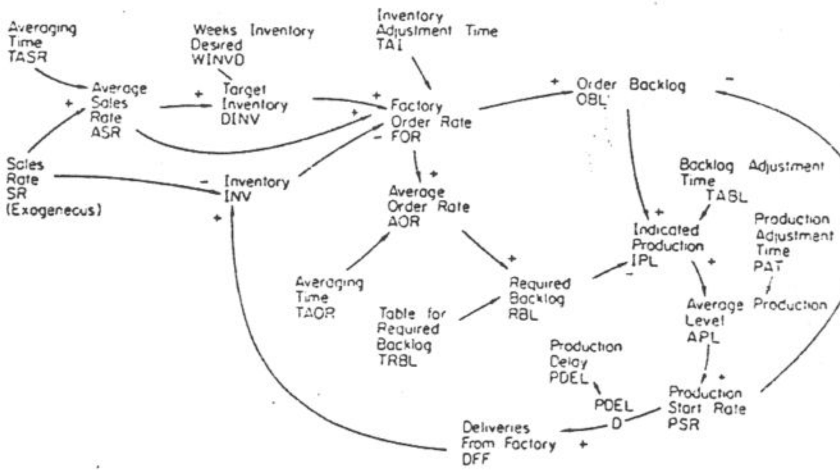
In an example of an application that follows, the relative changes were calculated by a pocket calculator and two basis variables with the smallest relative changes were removed from the basis. The incoming variables were selected randomly by using random number tables.[5] The same procedure was then repeated again from the very beginning but this time removing those variables which promise most improvement to the objective function value.

An Example of an Application

Coyle demonstrates in his "Management System Dynamics" how system behaviour can be improved by the loop-analysis method[6]. He explains the application as follows:-

> "A company has two departments. Distribution hold a stock with which to meet sales, and replenish the stock by placing orders on Manufacturing. Manufacturing adjust their production against the backlog of unfilled orders, delivering the finished goods to Distribution's stock, after a delay. For simplicity in this example, distribution are allowed to have negative stocks and Manufacturing regard the backlog as depleted when the work has started."

Fig. 2 below shows the influence diagram of the basic system that needs correcting.



<u>Fig. 2. Influence Diagram of the Basic System</u>

Reproduced from Coyle, p. 206.

Coyle makes changes in the model in several phases, and in this way gradually improves model behaviour. Let us now suppose that a non-specialist would try to do the same. He would first make a list of all conceivable changes. To give a specific example: Factory order rate (FOR) was defined in the basic system as follows:-

FOR.KL=ASR.K+(DINV.K-INV.K)/TAI

The non-specialist might now reason that FOR could be a function of order pipeline instead of inventory <u>or</u> a function of both. The equation for FOR will thus be changed:

FOR.KL=ASR.K+A1*(DINV.K-INV.K)/TAI+A2*(PLD.K-PLA.K)/TAPL, where

PLD = pipeline desired
PLA = pipeline actual
TAPL = time to average pipeline

We have now two new parameters, A1 and A2. When A1 = 1 and A2 = 0 the equation is simplified to the original one. On the other hand, if A1 = 0 and A2 = 1 the model structure has changed from what it was. In both cases we have an ordering policy that is <u>pure</u> as it relies only on one alternative. It might be reasonable, however, to pursue both choices simultaneously, i.e., to have a <u>mixed</u> policy. This happens when A1 0 and A2 0.

Let us suppose now that our non-specialist selected exactly the same model changes as Coyle did in his pedagogical example, just to get a yardstick for later comparison. In this way we have a set of potential model structures and parameters without having any idea how to select the right combination out of this mess.

The list below shows those model equations that have changed:-

```
R   FOR.KL=ASR.K+A1*(DINV.K-INV.K)/TAI+A2*(PLD.K-PLA.K)/TAPL
A   RBL.K=A3*TABHL(TRBL,AOR.K,50,150,25)+(1-A3)*TABHL(TRBL,APL.K,
X   50,150,25)
A   IPL.K=A4*(OBL.K-RBL.K)/TABL+A5*AOR.K+A6*APL.K
A   PLD.K=(A7*ASR.K+(1-A7)*AOR.K)(A8*WPLD+(1-A8)*DDR.K), where
```

FOR = factory order rate
ASR = average sales rate
DINV = desired inventory
INV = inventory
TAI = time to adjust inventory
PLD = goods desired in production pipeline
PLA = goods in production pipeline
TAPL = time to adjust pipeline
RBL = required level of backlog
TRBL = table of required backlog
AOR = average order rate at factory
APL = actual production level
IPL = indicated production level for backlog control
OBL = actual order backlog
WPLD = weeks pipeline desired
DDR = delivery delay recognized by distribution

The model has now 8 new parameters (A1,...,A8) and each of them might be selected as a variable for the SDR-algorithm. Let us call them decision parameters in order to distinguish them from usual system dynamics parameters. A value of zero for a decision parameter would cut off an information flow and change the structure of the model. The model also has many ordinary parameters, like TAI,TAPL, etc. Most of them are controllable, as management can change their values within some limits without making any physical changes in the real world system. Production delay, PDEL, is fixed, however, for modelling purposes, as a production process must be assumed as given in the short run. Ordinary parameters are, therefore, either controllable or fixed.

Let us suppose now that the following 9 parameters are controllable and will be selected for a closer examination. The allowable range for all of them is assumed to be from one to 15 weeks.

TASR = time to average sales rate
TAI = time to adjust inventory
TAPL = time to adjust pipeline
TAOR = time to average order rate
TABL = time to adjust backlog
PAT = time to adjust to planned production level
TAP = production averaging time
TSDD = time to smooth delivery delay
WPLD = weeks pipeline desired

We have now selected 17 parameters, of which 8 are decision parameters. This model is so small that all parameters can be given to the search process at the same time. Let us call this alternative simultaneous approach. The other alternative, especially proposed in this paper, is to give only some but not all alternatives at the same time to the search algorithm. This strategy will now be named sequential approach. In the area of plant layout heuristics there is a similar kind of division. A construction procedure starts with an empty layout and gradually assigns departments to locations until the layout is complete. An improvement procedure starts with a complete layout and attempts to improve on it by changing the locations of departments.

The number of variables in the basis and the number of leaving or entering variables are both decisions that should be made. Let us decide, for the sake of illustration, that the basis size is going to be 7 variables and that 2 variables will be changed in each grand iteration.

The initial basis was build randomly, all 17 variables thus having exactly the same chances of being selected. As to the initial values for all parameters, they were selected in such a way that the model equaled Coyle's basic system.

Relative changes of the basic variables were calculated (using a pocket calculator) after 30 iteration rounds. Two variables with smallest relative changes were removed from the basis and two others were added randomly. This was the first grand iteration. After another 30 iterations the same procedure was repeated again but without accepting any variables that had left the basis earlier. In 6 grand iterations all model variables had already been tried and with the results shown in Figure 3. When a variable leaves the basis it will retain the latest value it had. For example, A7 changed only from 1 to 0.88 in the first grand iteration. The relative change was not more than 12% and A7 was removed. After the first grand iteration the line of A7 in Fig. 3 is empty as A7 is now a parameter with a value of 0.88.

Quadratic cost components were added to the model for purposes of defining the objective function. The form of the objective function was taken as that which gave the best results in an earlier study.[7] It consisted of weighted terms of production rate variations, inventory variations, and inventory errors from a target value, all summed over a reasonably long period of time.

First part of two pages

Variable	one			two			three		
	Initial value	New value	Change § in p.c.	Initial value	New value	Change in p.c.	Initial value	New value	Change in p.c.
$0 \leq A1 \leq 4$	1	.12	88	.12	.07	42			
$0 \leq A2 \leq 4$							0	.17	∞
$0 \leq A3 \leq 1$									
$0 \leq A4 \leq 4$	1	2.28	128	2.28	4.0	75	4.0	4.0	0
$0 \leq A5 \leq 1$									
$0 \leq A6 \leq 1$	0	.01	0†						
$0 \leq A7 \leq 1$	1	.88	12						
$0 \leq A8 \leq 1$									
$1 \leq TASR \leq 15$							4.0	1.0	75
$1 \leq TAI \leq 15$				4.0	1.0	75	1.0	1.32	32
$1 \leq TAPL \leq 15$									
$1 \leq TAOR \leq 15$	4	8.48	112	8.48	15	77	15	15	0
$1 \leq TABL \leq 15$									
$1 \leq PAT \leq 15$							3	5.20	73
$1 \leq TAP \leq 15$				4.0	2.16	46			
$1 \leq TSDD \leq 15$	4	2.40	40	2.40	5.48	128	5.48	5.02	8.4
$1 \leq WPLD \leq 15$	6	4.32	28	4.32	2.30	47	2.30	2.12	7.8
Objective Function:									
Initial value		110 750.‡			750.07			433.42	
Final value		750.07			433.42			400.11	

Variable	Grand Iteration Number: four			five			six		
	Initial value	New value	Change in p.c.	Initial value	New value	Change in p.c.	Initial value	New value	Change in p.c.
A1									
A2	0	.17	∞	.17	.05	71	.05	0	100
A3							1	0	100
A4									
A5				0	.35	∞	.35	.40	14.3
A6									
A7									
A8	1	.31	69	.31	.27	13			
TASR	1	1	0						
TAI	1.32	1.29	2.3						
TAPL							4	13.64	241
TAOR									
TABL				4	2.16	46	2.16	2.67	24
PAT	5.20	4.24	18.5	4.24	4.83	14	4.83	1.00	79
TAP									
TSDD	5.02	7.64	52	7.64	7.03	8.0			
WPLD	2.12	1.86	12.3	1.86	12.84	590	12.84	14.99	17
Objective Function:									
Initial value		400.11			371.69			277.33	
Final value		371.69			277.33			216.84	

† It was arbitrarily decided that an increase in a parameter value from zero to less than 0.1 would be regarded as zero increase

‡ in thousand monetary units

§ only the absolute value was considered

Figure 3. Demonstration model changes in six grand iterations with 30 iterations in each. The basis has 7 variables.

Any variable changes were permitted when all variables had already been tried in a basis. Figure 4 shows the speed of convergence towards an 'optimum' that was found earlier by using the simultaneous approach and 150 iterations.

Grand Iteration	0[†]	1	2	3	4	5	6
Above 'Optimum' in p.c.	5749.	29.00	12.54	10.81	9.33	4.42	1.28

Grand Iteration	7	8	9
Above 'Optimum' in p.c.	1.25	0.61	0.14

[†] the starting condition

Figure 4. Convergence of Solution

The sequential approach and the simultaneous approach both gave similar time series and they look very satisfactory. As Figure 5 demonstrates, recovery from a step input was fast in the sequential approach. To find any significant differences we have to go deeper into detail and have to focus on individual parameter values in both cases.

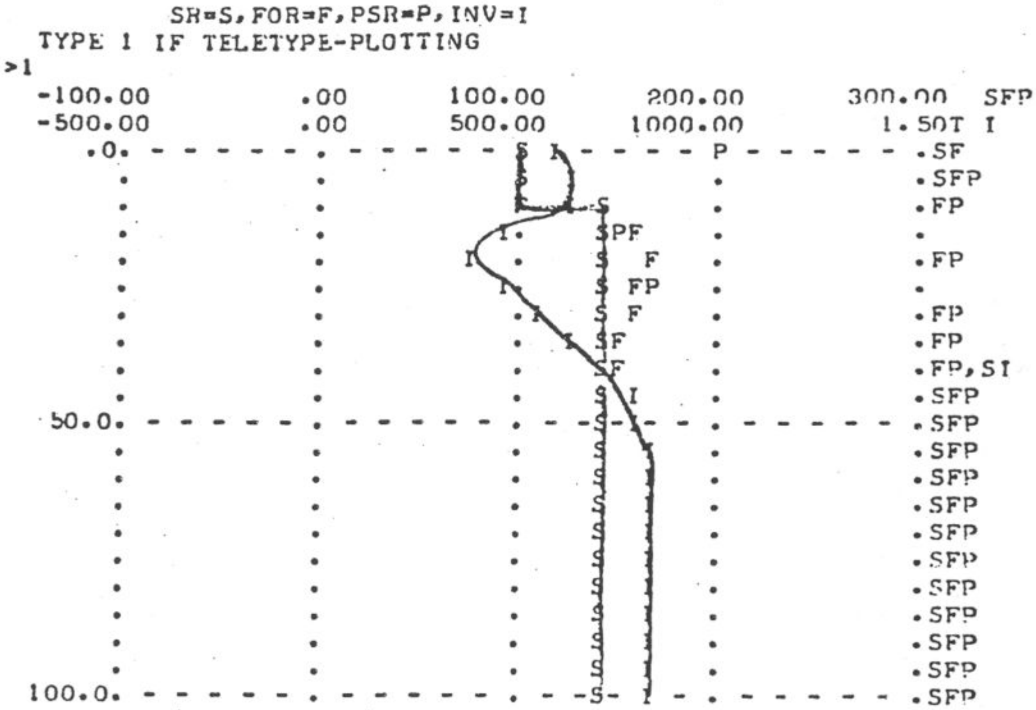


Figure 5. Some time-series from sequential approach after 9 grand iterations

Fig. 6 summarized the most crucial parameter variations between both approaches. Comparison with the equations of FOR and PLD given before shows that the sequential approach controlled delivery delay, the simultaneous approach controlled pipeline effects and inventory discrepancy. It looks as if we have here a case where highly different policies were of equal value. This is encouraging, as a hill-climbing algorithm can never guarantee that the lowest one of multiple minima has been found.

Final parameter value	Sequential approach	Simultaneous approach
A1	.07	1.00
A2	0	.72
A8	.27	1.00

Fig. 6. Major policy differences between sequential and simultaneous approaches

Let us now see what happened when perturbation of parameters was tried in order to find out the incoming basis variables. Each parameter was changed at a time by $\pm$ 1% when predetermined boundaries allowed it. When the lower limit of a parameter was zero and the parameter had that value, the change of 1% from the upper limit was made. An index value was calculated for each parameter in the following way:

$$I_j = \Delta O_j \frac{p_{B,j} - p_j}{|\Delta p_j|} \quad , \quad \text{where}$$

I_j = index value

ΔO_j = improvement in the objective function value, when the value of parameter j was changed

$p_{B,j}$ = boundary value of parameter j in the direction which improves the objective function value

p_j = change in the value of parameter j

Two parameters with highest index values were then selected as incoming basis variables. Figure 7 summarized the results received. The completed grand iteration number one from Figure 3 was selected as the starting point. Numerical values in parentheses refer to the values of incoming (+) and outgoing (-) basis variables. The values of the "Objective Function Change" column indicate the cost effect of each parameter change tried.

Variable	Grand Iteration Number				
	New value	Objective function change †	New value	Objective function change †	New value
A1	0.12		0.28		0.29
A2		-0.72	+(0)	0.30	0.02
A3		0.44		-0.04	
A4	2.28		4.00		4.00
A5	+(0)	15.03	0.26		0.25
A6	-(0.01)			0.01	
A7	-(0.88)			0	
A8		0		0	
TASR		1.18	+(0)	0.13	2.54
TAI		3.67		0.05	
TAPL		0		0	
TAOR	8.48		-(15.00)		
TABL	+(4.00)	4.51	1.00		1.00
PAT		0.48		0.15	
TAP		0.19		0.03	
TSDD	2.40		-(2.40)		
WPLD	4.32		1.02		1.03
Objective Function:					
Final value	750.07†		241.94		238.60

Figure 7. Model changes by using parameter perturbation approach

† in thousand monetary units

Figure 7 clearly shows an unexpected development in the optimizing process, i.e. very rapid initial improvement leads to a less recommended local optimum. In this specific case it was better to proceed at a lower speed by selecting some parameters of minor importance to the basis. The more efficient approach, on the other hand, moved towards a solution which never allowed experimentation with many variables. There is a limit to the use of simultaneous approach, related to model size, model structure and to the stste of the art in electronic data processing. This means that we cannot always use a simultaneous approach. Even when it is possible to do so, computational efficiency should be examined in both frameworks.

The results of a search procedure depend on the initial values of the basis variables. Experimentation with them is likely to lead to a good enough solution, but at the expense of significantly increased computation time. More generally speaking, we should notice that either the perturbation approach or the sequential approach might initially fail. Construction of artificial intelligence algorithms might therefore utilize a combination of the concepts of random selection, parameter perturbation and initial value perturbation.

Final Remarks

In the experiment reported the search process was quided by the assumed cost function. It had three additive terms, which should be balanced somehow to give enough weight to all components of the cost function. Cost function parameters cannot be judged in complete isolation from the model structure and policies, however, as these have repercussions on real cost balance achieved during a simulation run.

A highly simplistic step input was tried in the model. In a real world more complicated driving functions and noise might be expected. As both these examples show, much more research work will be needed. Especially artificial intelligence algorithms hopefully will be taken as a challenge by S.D. specialized control engineers.

This paper has shown that by dividing the variables into basic and non-basic ones the sequential approach is extendable to problems of practical size. This immediately brings up the question of the real significance of the idea.

The starting point was the notion that use and understanding of System Dynamics has required undesirable expertise. Every effort to reduce this is highly welcome, as the age of Industrial Democracy is going to bring new groups into the decision-making process. At that time survival of individual management tools might depend on the understanding required by a user. Now we know that this challenge can be met.

The flexibility of the technique opens up, however, new vistas in other directions also. Until now Systems Theory has not been successful in providing models which could structurally change when time passes. This handicap has led to the use of scenarios in models that reach far out into the future. The experience so far available indicates, however, that optimizing models end up with mixed policies that are very suitable for continuous changes. Therefore, model structuring can be an automated and <u>continuous</u> process in system dynamics. In practical terms, this process would be directed together by the artificial intelligence algorithm and the objective function of a model.

The objective function is that part of the model that really guides the evolutionary process produced by the model. At a higher abstraction level, we might say that any objective function corresponds to policy making by rate equations in ordinary system dynamics terms. In practical terms, optimization required that some parameters were treated as variables. In the same way, policy making by an objective function is more general than policy making in system dynamics. <u>The objective function itself is likely to change</u> as time passes. Again technically speaking, this is no problem. In SDYDYN, for example, any model equation can be accepted as the objective function. Therefore, it is possible to let the future shape an evolutionary process of the objective function. System Dynamics has already given promises as a tool for futurologists[8], but the prospects look even brighter.

Discussion has so far been confined to a single-objective case, although multi-objective decision making is likely to be increasingly in focus in the years to come. This is an area where the sequential approach might prove to be very valuable, as the system and the objective function are not independent of each other, but interrelated. Instead of having only a set of potential basis variables for selection purposes, as so far implied, we would have a set of objective function elements, too. A decision maker might then repetitively change the objective function by using information on model structure and model behaviour. This level of sophistication leads to interactive modelling from a terminal, but it implies that all previous phases of conceptual developmen have already been implemented by making them fully automatic.

References

1) Coyle, R.G. (1977) Management System Dynamics, John Wiley & Sons, p. 237.

2) Keloharju, R. (1977) Multi-objective decision models in system dynamics. Dynamica, Part I and II.

3) Arcus, A.L. (1963) An Analysis of a Computer Method of Sequencing Assembly Line Operations. Unpublished Ph.D. dissertation, UCLA.

4) Biermann, Alan W. (1976) Approaches to Automatic Programming. In Advances in Computers, Vol. 15 by Morris Rubinoff and Marshall C. Yovits (Editors), Academic Press.

5) The Rand Corporation (1955) A Million Random Digits. Ill. U.S.A.

6) Coyle, R.G. Op. cit., Chapter 8

7) Keloharju, R. Op.cit.

8) Bloom, M.L. (1977) Time-Dependent Event Cross-Impact Analysis: Results from a New Model. Technological Forecasting and Social Change 10, 181-201.