Expert System Dynamics Modeling with GURU

John J. Uhran, Jr.
Nasir Ghiaseddin
Ramzi Bualuan
University of Notre Dame

ABSTRACT

The purpose of this paper is to present a possible way in which a marriage between Artificial Intelligence and Modeling can take place. More specifically it is the purpose of the paper to explore some basic concepts related to Artificial Intelligence and by using an expert system shell called GURU to aid in the development of system dynamics models. The concept is one of going from data base to knowledge base to models and to examine the line of reasoning that is used in formulating the problem. Simple examples will explore the potential of this approach.

INTRODUCTION

A. Background

System Dynamics was first introduced by Forrester in 1953 (Forrester 1959) with a small but active group of researchers continuing to explore the potential which this technique has for solving a variety of difficult problems. This approach to modeling is one which allows the details to the micro level parts of the problem to be modeled in a macro way such that the answers are reasonable and accurate. It has lent itself ideally to the solution of many socio-economic problems (Forrester, 1969 and Meadows 1974) which in the past have not been easily resolved using other techniques. The approach over the years has evolved in such a way that it allows interesting and useful answers to come out of careful evaluations of models that on the surface appear simple -only several states -but in fact are fairly complex because of the inter-related non-liniar feed back loops that are necessary to make the models valid. Useful studies, both theorectical and practical, continue to evolve from basic concepts.

About the same time that this technology began to evolve, another had its roots in a 1956 meeting at Dartmouth College in which several researchers across the country met to discuss the concepts of artificial intelligence. The power of computing that was developing at the time was the impetus, but only a small core of researchers continued the work in this area until recently. The whole area of artificial intelligence is now blooming in numerous directions creating many interesting topics related to the original concepts. In particular, expert systems have shown themselves to be a particularly valuable tool.

In recent years, researchers have begun to examine the ways in which models evolve. System dynamics has been no stronger to this idea (Gould 1985). In particular, the marriage of system dynamics with expert systems seems a natural one (Shannon and Mayer 1985).

Obviously in order to evolve the correct form as well as the correct values associated with a model, data must be evaluated. Thus data bases become one of the key ingredients of an expert system application to modeling. Bringing these two fields together is natural since the methodology of System Dynamics Modeling is somewhat heuristic in its approach as is the use of expert systems. The latest expert systems known to system simulation professionals can be built into the software and the ability of the program to remember the results of simulations enable it to aid in the possible development of new rules and heuristics.

B. Problem Explanation

In this paper we would like to show how software can facilitate two primary improvements in simulation modeling; 1) a streamlining of the modeling process through a reduction in the number of state variables necessary to arrive at a useful model and 2) an increase into the insights of the system structure that might be directly derivable from data and from the interaction of the expert system in the modeling process. We will establish certain rules and guidelines and then use a software package called GURU as an expert system shell so that it may establish for us the actual model itself as well as the line of reasoning of the model development for ultimate solving using a standard simulation package. Examples of a data base modeler, the modeling language and system dynamics will all be demonstrated.

II. SYSTEMS DYNAMICS MODELS

A. Modeling a system.

A system, in order to be simulated, must first be represented by a model. For a model to be accurate and complete, it must contain all the elements that would make it behave as though it were the system itself.

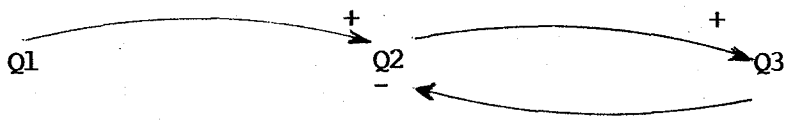
In System Dynamics, a system can be completely modeled with a set of first order equations, written in an appropriate simulation language such as NDTRAN (Uhran and Davisson 1984) or DYNAMO (Pugh). The equations would contain the necessary information about the components of the system, specifying their types, their values, and interconnections. Before equations are written though, a Flow Diagram (FD), (See Figure 1) is set up. In it, all the components of the system are represented by their respective symbols, and are connected to represent their effect and influence on each other.

When the system dynamics model of a system is required, it is frequently hard to know what the types of components are beforehand. This is why a Casual Loop Diagram (CLD) is more appropriate to start with and easier to draw. Here all that is needed is the connectivity among the various elements represented, and this can be done with the help of a signed digraph. It should be noted that a lot of information is lost representing a system by a CLD. The types of the elements are not specified and their values are not given. Only influences affecting the elements is given in the simpliest form.

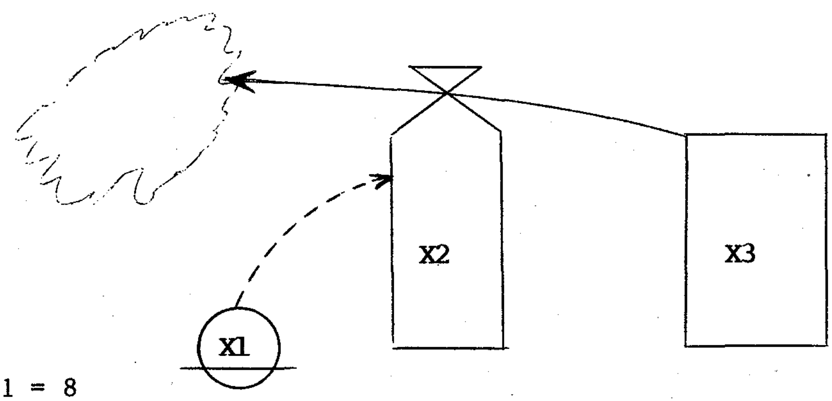
Figure 1 is an example showing the relationship between the CLD and FD. It is a simple first order non-linear system whose solution is of the form $y=a/x$.

Example:

Causal Loop Diagram:



Flow Diagram:



Equations: Q1 = 8

Q3 = Q2

Q2 = Q1 / Q3

Figure 1. A Simple Example

B. Influence graphs.

The Causal Loop Diagram, as we mentioned above, gives us an idea of how different elements of a system influence each other. The sign at the end of the arc indicates the effect the increase in one variable has on the other. In the previous example for instance, it can be seen that an increase in Q1 produces an increase in Q2, which in turn would produce an increase in Q3. However, as Q3 tends to increase, Q2 would tend to decrease, which indicates that we have a negative feedback loop.

The relations between various variables are usually not governed by laws or expressed by equations, but rather determined experimentally. In the course of this paper, we shall assume that these relations are of a monotonic nature. This assumption is often justifiable, and in cases where it is not, we can then consider only small variations in the variables around a stable point. In light of the above, the influence that one variable has on another can then be represeneted by a simple incfluence graph instead of a signed arc joining those two variables. As data are taken experimentally, the relationship between two variables can be observed and the graph determined.

III. COMPUTER EMPLEMENTATION

A. From Causal Loop Diagrams to Flow Diagrams.

In order to go from a CLD to a FD, however, the rules and heuristics that govern System Dynamics must be strictly followed. Through these rules, the entities in the CLD can be systematically classified into their respective types, and the whole FD can be drawn.

The methodology that was followed to perform the above task is the Burns (Burns 76, 77 and 79) algorithm. Burns uses SD rules to set constraints on the couplings between different entities and on the relationships among them. When these rules are applied, the entities become known, and the ambiguity involved becomes minimal for small systems.

The entities are called quantities, and may be of the following types:

1. Rates
2. States (Levels)
3. Auxiliaries
4. Inputs
5. Outputs
6. Parameters

The quantities are connected via couplings, either:

1. Information (I) couplings
2. Flow (F) couplings

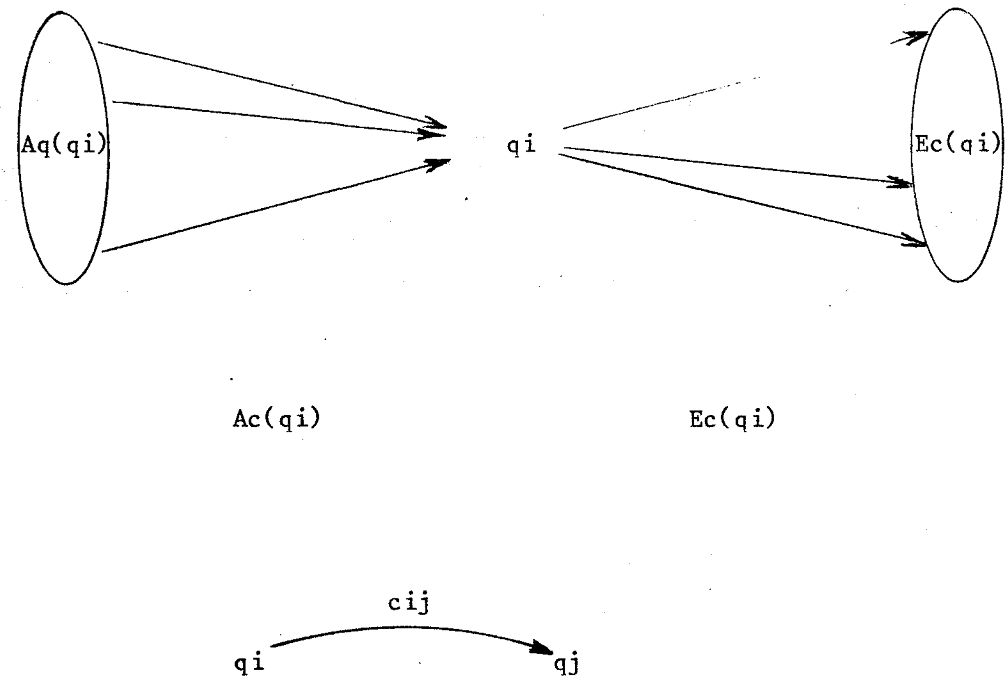
Simply put, the Flow couplings are those that connect rate quantity to a state quantity, whereas the Information couplings are considered to be all the other couplings.

B. Burns Algorithm

The Causal Loop Diagram is represented mathematically with what Burns (Burns 1976, 1977, and 1979) refers to as a Square Ternary Matrix (STM). The STM is a N x N matrix (N is the number of components considered). Each dimension of the matrix lists all the quantities in the system, and the cells of the matrix are the couplings of the CLD, i.e. they indicate how the various quantities are coupled to each other. A 0 (or empty cell) indicates that there is no coupling connecting the two quantities, whereas a 1 or -1 signifies that the two quantities are related.

Initially, all the quantities of the system are unclassified, i.e. their values (types) are unknown. As each rule, theorem, axiom, or supposition is applied, the quantities and couplings become known one by one, which will then help classify other variables.

In order to represent how a quantity is related to adjacent quantities, some attributes are created. These attributes are the "affector" Ac(qi), and "effector" Ec(qi), concepts shown in Figure 2. Then a set of rules must be developed. A fundamental principle is applied, the consistency supposition, which states that all couplings directed towards a quantity must be of the same type (Information of Flow), and identically so for all the couplings directed away from a quantity. In other words, all the elements of the set Ac(qi) are of the same type, and similarly for all the elements of the set Ec(qi).



cij : affector coupling of qj

cij : effector coupling of qi

qi : affector quantity of qj

qj : effector quantity of qi

Figure 2. Concepts Required by Burns.

The importance of the consistency supposition should be obvious: whenever a quantity qi becomes classified, and since the inward and outward couplings of a known quantity are of determined type, then all the couplings associated with qi will be known. As more couplings become classified, the types of remaining unknown quantites can be determined with more ease.

A straightforward way to find the type of a quantity is by looking at its affector and effector couplings. In Table 1, the rules that govern how couplings and quantites are related is given. These rules are necessary and sufficient conditions for the relations to hold.

Table 1

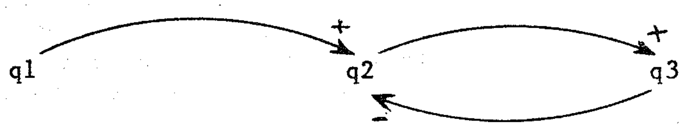
I ---->	RATE	----> F
F ---->	STATE	----> I
I ---->	AUXILIARY	----> I
I ---->	OUTPUT	
	INPUT	----> I
	PARAMETER	----> I

An output is a quantity that has no outward coupling, (i.e. whose Ec is an empty set). Similarly, if a quantity has no inward coupling (its Ac is an empty set), then it is either an input or a parameter.

Besides the rules that relate quantities to couplings there are rules that govern the most important structure of a System Dynamics model, the feedback loop. Briefly stated, a feedback loop (minor submodel) must consist of at least a rate, a state, a Flow coupling, and an Information coupling. Frequently associated couplings going into and/or out of the loop and also included.

To illustrate the above concepts, the simple example of Figure 1 will be used. The system consists of only three quantities, and contains a feedback loop. Figure 3 now includes the initial state of the STM diagram.

CLD :



STM :

	1	2	3
1		+1	
2			+1
3		-1	

Figure 3. CLD with its Initial STM.

The following is a step-by-step analysis of the evolution of the Burns algorithm as it is applied to solve the above problem.

Since the signs of the couplings have no effect on the Flow diagram, and do not come into consideration until the model is to be computerized, we will then drop them. Evolution of the STM is given in Figure 4. (Explain meaning of rows & columns in STM.)

By inspection of the STM, we can easily determine that q1 is an input (or parameter; since the difference is usually decided by the user himself, we shall consider inputs only). This comes from the fact that column 1 is all empty.

Since q1 is an input, coupling c12 must therefore be an Information coupling (I) Figure 4a.

Applying the consistency supposition, coupling c32 must then be an I, Figure 4b.

	1	2	3
1		I	
2			1
3		1	

(a)

	1	2	3
1		I	
2			1
3		I	

(b)

	1	2	3
1		I	
2			F
3		I	

(c)

Figure 4. Evolution of an STM.

We can see that q2 and q3 form a feedback loop (c32 and c23 are symmetrical with respect to the main diagonal, and are nonzero). One of those quantities must then be a rate, and the other a level. Since column 2 is all I's, q2 must be a rate; q3 therefore becomes a state, and c23 is a Flow coupling. The final STM is shown in Figure 4c.

With this information the solution becomes,

q1 : INPUT
q2 : RATE
q3 : STATE

and thus the Flow Diagram can be drawn as in Figure 1.

III. A MODIFIED BURNS ALGORITHM USING GURU.

The Burns algorithm is a solid tool for converting a CLD into a FD. It incorporates all the rules of System Dynamics, keeps user intervention to a minimum (see Ramos' (Ramo 1983) modification for a better performance of the Burns algorithm for large systems), and makes sure the representation is consistent.

The computer implementation of the algorithm was done with the help of an Expert System shell called GURU (GURU Reference Manual). Among the advantages of using an Expert System over a conventional procedural computer language are the fact that it allows us to write all the rules in an IF..THEN manner, which is useful for the purpose of understanding. In addition, GURU is user-friendly, powerful, versatile, and allows for user intervention when unable to continue. It makes the addition, modification and deletion of production rules easy to perform.

The main advantage of using a tool as GURU, however, is its ability to give its "line of reasoning". In other words, it can give a trace through all the rules that it fired in finding its solution.

The version of the Burns algorithm that is to be implemented on GURU has been simplified further. It works very well with small systems, and it only breaks down with systems that are complicated. It assumes the model is valid and correct. It bypasses many of the details that Burns includes, and stresses only the important rules. It works remarkably well with systems whose feedback loops contain only two elements. It should be noted that Burns' complicated example given in his last paper does not contain any feedback loop with more than two elements.

The following page lists the eight suggested steps in the simplified algorithm. Note that this algorithm is not procedural (as Burns' is). Each step contains many similar rules (one for each quantity), and may be applied at any time if the premises of one of its rules is true. Except for steps 1, 2 and 8, the order of the other steps is therefore dependent on the problem.

B. Simplified algorithm.

1. Find all outputs. These are the quantities for which the corresponding rows of the STM are empty (all zeros). Make the corresponding columns Information columns.

2. Find all inputs (or parameters). These are the quantities for which the corresponding columns of the STM are empty. Make the corresponding rows Information rows

3. Apply consistency supposition for Information couplings. For each Information row (column), change every 1 to an I and make its column (row).

4. Find auxiliaries. Qi is an auxiliary variable if row i and column i are both of the Information type.

5. Find feedback loops (two elements). If two entries of the STM are symmetrical with respect to the main diagonal and are nonzero (by now, they should be 1 and I), they indicate a feedback loop. the row of the cell with I will indicate a state, its column a rate. The 1 changes to F.

6. Apply consistency supposition to Flow couplings. Similar to step 3.

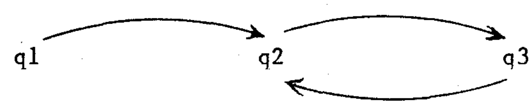
7. After all the above steps are fully implemented and iterated, remaining 1's should be F's. The preceeding steps can then be applied again.

8. When all quantities are known, the execution terminates.

IV. SOLUTION OF PROBLEMS USING GURU.

A. PROBLEM NO. 1

Problem No. 1 is depicted in figure 5(a). Our expert system written in GURU will start with the initial STM (figure 5b) and will achieve the final STM (figure 5c). The line of reasoning used by the expert system is contained in appendix A. A detailed explanation of the problem solving process is followed:



(a)

STM:

	1	2	3
1		1	
2			1
3		1	

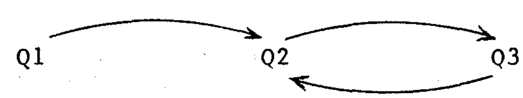
Initial
(b)

	1	2	3
1		I	
2			F
3		I	

Final
(c)

Figure 5. Problem #1.

Initially



Q1 : Unknown
Q2 : Unknown
Q3 : Unknown

STM:

	1	2	3
1		1	
2			1
3		1	

Since the 1st column is empty, this indicates that Q1 is an input (because it has no inward couplings) => rule INPT1 can fire.

INPT1 When this rule fires, it sets Q1 to INPUT, and makes outward couplings from Q1 (OUTW1) of I type. This is in order to apply the consistency supposition later.

STM: (unchanged)

	1	2	3
1		1	
2			1
3		1	

Q1 : Input
Q2 : Unknown
Q3 : Unknown

We can now apply the consistency supposition to row 1 (because OUTW1 has just been set to I)

CSIRW1 When this rule fires, all elements of row 1 will become I, and, for each of these elements, its corresponding inward couplings will be of the I type (again, in order to apply CS later)

In our problem, then, T12 becomes I and INW2 is set to I

STM:

	1	2	3
1		I	
2			1
3		1	

Q1 : Input
Q2 : Unknown
Q3 : Unknown

Since T12 is I, therefore T32 should also become I (CS) (That's why INW2 was set to I above). This will be done by rule CSICL2, which can now fire.

CSICL2 When this rule fires, all "1" elements of column 2 will become I, and their corresponding outward couplings will be of the I type.

Here, then, T31 becomes I and OUTW3 is set to I

STM:

	1	2	3
1		I	
2			1
3		I	

Q1 : Input
Q2 : Unknown
Q3 : Unknown

Since OUTW3 has been set to I, that will enable CSIRW3 to fire.
Since there are no other element on that rown (3), CSIRW3 will have no effect.

CSIRW3 The firing of this rule has no effect.

STM:
(unchanged)

	1	2	3
1		I	
2			1
3		I	

Q1 : Input
Q2 : Unknown
Q3 : Unknown

As we can see, CS cannot be applied anymore. We therefore have to look for feedback loops.

Two-element feedback loops can be detected on the STM with nonzero elements symmetrical with respect to the main diagonal.

We can see that Q3 and Q2 form such a loop. FDBK32 can then fire.

FDBK32: When this rule fires, T23 becomes F, Q2 is a rate, and Q3 a state. This is because Information couplings go from state to rate, while flow couplings go from rate to state.

STM:
(final)

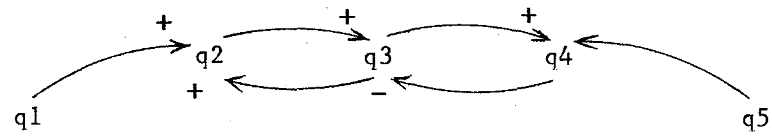
	1	2	3
1		I	
2			F
3		I	

Q1 : Input
Q2 : Unknown
Q3 : Unknown
(solution)

Since all quantities have been classified, execution stops.

B. PROBLEM NO.2

The causal loop diagram of problem No. 2 depicted in figure 6a. Similar to problem No.1, the expert system will start with the initial STM (figure 6b) and will produce the final STM (figure 6c). The line of reasoning is contained in Appendix B.



STM:

(a)

	1	2	3	4	5
1		1			
2			1		
3		1		1	
4			1		
5				1	

Initial
(b)

	1	2	3	4	5
1		I			
2			F		
3		I		I	
4			F		
5				I	

Final
(c)

Figure 6. Problem #2.

V. CONCLUSION

What we have tried to emphasize in this paper is the validity of building expert systems into system dynamics modeling. The advantages are significant even though, as indicated here, applied to very simple examples. Once the Approaches above are understood, the second step is to go back further into the problem that is the data base and to use the expert system to develop the CLD itself. This is a problem which we are attempting to do at this time and the results are proving promising. This process, the integration of an expert system with its line of reasoning heuristic rules and ability to remember to a certain events, becomes a much more powerful tool for the overall modeling process.

Appendix A

Problem 1 line of reasoning

Initially: quantities Q1, Q2, Q3 are unknown

STM:

	1	2	3
1		1	
2			1
3		1	

(Refer to the supplements to the report for the rules' effects on the STM.)

1st rule INPT1 (see if Q1 is an input: this will be the case if column 1 is empty)

effects: Q1 is an input
outward couplings from Q1 are of information type (I)

2nd rule CSIRWI (apply consistency supposition to row 1)

effects: nonzero elements of row 1 must become I
T12 = I
=> inward couplings to Q2 must be of I type

3rd rule CSICL2 (apply consistency supposition to column 2)

effects: T32 becomes I

4th rule CSIRW3 (apply consistency supposition to row 3)

effects: None

5th rule FDBK32 (detect a feedback loop between Q3 and Q2)

effects: T23 becomes F
Q2 becomes Rate
Q3 becomes State

Appendix B

<u>Problem 2 line of reasoning</u>

1st rule <u>INPT5</u> (see if Q5 is an input)

<u>effects</u>: Q5 is an input
couplings outward from Q5 are of I type.

2nd rule <u>CSIRW5</u> (apply consistency supposition to row5)

<u>effects</u>: T54 becomes I
couplings inward to Q4 are of I type.

3rd rule <u>CSICL4</u> (apply consistency supposition to column4)

<u>effects</u>: T34 becomes I
couplings outward from Q3 are of I type.

4th rule <u>CSIRW3</u> (apply consistency supposition to row 3)

<u>effects</u>: T32 becomes I
couplings inward to Q2 are of I type.

5th rule <u>CSICL2</u> (apply consistency supposition to column 2)

<u>effects</u>: T12 becomes I

6th rule <u>INPT1</u> (see if Q1 is an inputs)

<u>effects</u>: Q1 is an input
couplings outward from Q1 are of I type.

7th rule <u>CSIRW1</u> (apply consistency supposition to row 1)

<u>effects</u>: none

8th rule <u>FBTRY</u> (find feedback loops)

<u>effects</u>: Q2 is a rate
Q3 is a state
Q4 is a rate

REFERENCES

[1] Burns, J.R. "A preliminary approach to automating the process of simulation model synthesis", Proc. Seventh Pittsburgh Conf. Modeling and Simulation, Apr. 1976.

[2] Burns, J.R. "Converting signed digraphs to Forrester schematics and converting Forrester schematics to differential equations", IEEE Trans. Syst., Man, Cybern., vol SMC-7, no. 10, pp. 695-707, Oct. 1977.

[3] Burns, J.R. "An algorithm for converting signed digraphs to Forrester schematics", IEEE Trans. Syst., Man, Cybern., vol. SMC-9, no. 3, pp. 115-124, Mar. 1979.

[4] Forrester, J.W., Industrial Dynamics, MIT Press, Cambridge, Mass, 1959.

[5] Forrester, J.W., Urban Dynamics, MIT Press, Cambridge, Mass., 1969.

[6] Gould, J.M. "Artificial Intelligence: A Tool for System Dynamics." Proceedings of the 1985 International Conference of the System Dynamics Society, Keystone, Colorado, pp. 317-330.

[7] GURU Reference Manual, Micro Data Base Systems, Inc., Lafayette, Indiana.

[8] Meadows, D.L., etal., Dynamics of Growth in a Finite World Wright-Allen Press, 1974.

[9] Pugh, A.L. DYNAMO User's Manual, M.I.T. Press.

[10] Ramos, J.M. "Some modifications to the Burns Algorithm", IEEE Trans. Syst., Man, Cybern., vol. SMC-13, no.1, pp. 108-110, Jan./Feb. 1983.

[11] Shannon, R.E., R. Mayer, and H.H. Adelsberger, 1985. "Expert Systems and Simulation." Simulation (June): pp. 275-284.

[12] Uhran, J.J., and W.I. Davisson, "The Structure of NDTRAN - A Systems Simulation Language", IEEE Transactions on Systems and Cybernetics, vol. SMC-14, No. 6, Nov/Dec 1984, pp. 837-846.