COMBINING DISCRETE AND CONTINUOUS SIMULATION

DYNAMO INTO SIMSCRIPT

Ronald Dale Jones

University of Missouri

ABSTRACT

Discrete and continuous simulation can be combined in one program by using a discrete simulation language to simulate, among other things, a continuous simulation program. This amounts to the translation of the continuous language into the discrete language. The translation of Dynamo into Simscript is examined.

I. WHY TRANSLATE DYNAMO INTO SIMSCRIPT?

The ability to define Dynamo in terms of Simscript was the original motivation. I planned to use Fortran to teach Simscript and to use Simscript to teach Dynamo. I also wanted a way to explore the utility of combining interacting discrete and continuous subsystems in one simulation model. The use of Dynamo to add a continuous simulation capability to Simscript was advantageous, I thought, in that one inherits the Dynamo documentation and retains communication with the Dynamo user population. Later I began to suspect that Simscript would provide a useful matrix for a Dynamo program in cases such as the execution of an experimental design. Dynamo is a brilliantly limited language, whereas Simscript provides the power of a general purpose programming language. (1,3,4,5)

Translation of Dynamo into Simscript might be advantageous for those not having a Dynamo compiler, merely to be able to utilize Dynamo, and might be of interest even for those not having a Simscript compiler, since the simulation of Dynamo requires so little of Simscript that the resulting Simscript program could be translated easily into a more widely available language, such as Fortran. However, the value of those uses would be reduced by the lack of the error detection and diagnosis capabilities provided by the Dynamo compiler. I planned to use the Dynamo compiler for debug work prior to translation into Simscript.

II. SIMULATION OF DYNAMO

Dynamo approximates a continuous process of change by trapezoidal integration, where the equations of the continuous system are solved repeatedly at intervals of DT time units. If DT is made sufficiently small, a useful approximation results. Simscript is event oriented; a system being simulated by Simscript changes when and only when an event occurs. The major concept in the simulation of Dynamo by Simscript is for Simscript to schedule, at intervals of DT time units, computational events for the equations of the Dynamo program.

The minimal Simscript program required to simulate Dynamo is relatively simple, consisting of a main program and one event routine, plus a preamble. The only functions of the main program are to schedule the Dynamo event for time zero and to start the simulation. The Dynamo event routine always solves the Dynamo equations, prints any output, reschedules itself for DT time units later, and returns. The preamble defines the Dynamo variables as global and provides other information to the Simscript compiler. The only complexity is that at time zero the event routine initializes the constants and solves the "level" equations of Dynamo through an initialization process.

A flow chart for the event routine is given by Illustration 1. Three subroutines contain the Dynamo equations for (a) initialization;(b) solution of the level equations; and (c) solution of the "auxiliary" and "rate" equations, respectively. Output routines can be added as desired. Any provisions for rerunning the Dynamo model should be added to the main program. Nine Dynamo functions, if used in the Dynamo model, must be simulated by Simscript routines; the other nine Dynamo functions can be simulated by Simscript library functions.

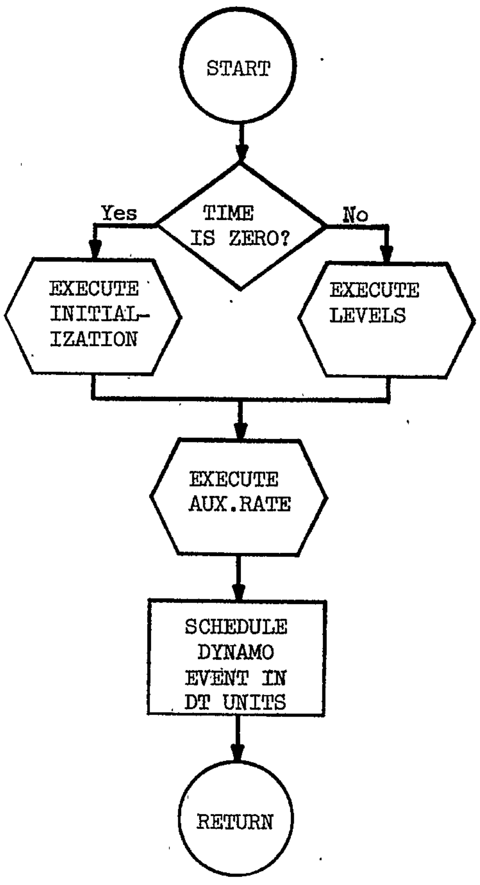
III. TRANSLATION INTO SIMSCRIPT

A manual procedure for translating Dynamo into Simscript is described elsewhere. (2) However, a few remarks are given below to suggest the steps involved.

Translation of a Dynamo program into Simscript involves (a) expansion of any Dynamo macros; (b) conversion of the expanded Dynamo equations to Simscript syntax; (c) use of the equations to complete the three subroutines called by the event routine; and (d) completion of the preamble. Preparation of output routines and modification of the main program to provide rerun capabilities are also potentially involved but are not within the

ILLUSTRATION 1

Event Routine for Simulating Dynamo



scope of this discussion.

A description of the macro expansion process is described in the Dynamo II User's Manual. (4) Conversion to Simscript syntax involves removal of all time subscripts from Dynamo names, modification of some function references, renaming of the "dummy quantities" and unique names created during the macro expansion process, conversion of all constants to real values, and other relatively minor matters of syntax.

All constants and levels are initialized in one subroutine, INITIALIZATION. Equations initializing constants are copied into the subroutine in a straightforward manner but initialization of the levels presents more difficulty. A level cannot be initialized until all the operands in its special initialization equation have been initialized. And all of the operands appearing in the equations of its operands. And so forth until only constants appear as operands.

Each category of equations (levels, rates, etc.) must be ordered so that a variable appearing as an operand within its own category of equations is used as an operand before its own equation is solved. The ordered level equations are placed in subroutine LEVELS; the ordered auxiliary and rate equations are placed in subroutine AUX.RATE, with the auxiliaries preceding the rates.

IV. DISCUSSION

The translation process listed in the Bibliography is limited to the case of the translation to Simscript II.5 of an error-free Dynamo II model for one run and with tabular output. It provides Simscript routines to simulate all Dynamo functions. The process has had but one test, that of the sample program given in the Dynamo User's Manual. (4) No test has been made of most of the function routines. The process was based upon inspection of Dynamo documentation rather than by examination of the Dynamo compiler itself.

BIBLIOGRAPHY

1. Johnson, G. D. Simscript II.5 User's Manual S/360-370 Version. Consolidated Analysis Centers, Inc., Los Angeles, 1972.
2. Jones, Ronald Dale. A Manual Procedure for Translating Dynamo Into Simscript. Mimeo. School of Administration, University of Missouri-Kansas City, 1973.
3. Kiviat, P. J., Villanueva, R., and Markowitz, H.M. The Simscript II Programming Language, Prentice-Hall, Inc., Englewood Cliffs, 1968.
4. Pugh, L. P., III, Dynamo II User's Manual, MIT Press, Cambridge, 1970.
5. Simscript II.5 Reference Handbook, Consolidated Analysis Centers, Inc., Los Angeles, 1971.