Behavior Research Methods & Instrumentation
1975, Vol. 7 (2), 226-228

Modeling on the simulation writer interactive program

ROBERT L. STOUT, Independent Consultant
136 Irving Avenue, Providence, Rhode Island 02906

The Simulation Writer Interactive Program (SWIP) is an extension of the Michigan Experimental Simulation Supervisor program series, and is designed to permit individuals who are not skilled programmers to create numerical simulation models, primarily for instructional purposes. SWIP elicts from the user information about the manipulable (independent), central (intermediate), and observable (dependent) variables in the user's model, as well as any necessary tables and control information. A powerful editor is available within SWIP for making modifications. SWIP deduces the structure of the user's model from the information about the variables, and produces an interpreted version of the model which is used to generate simulated data. A variety of model structures are possible in SWIP, including multivariate models, models for repeated measures experiments, and timeseries models. A context-dependent question-answering facility is also available.

The primary goal of the Simulation Writer Interactive Program (SWIP) is to provide a maximally flexible and easily usable interactive program for the implementation of instructional simulation models. SWIP is intended for use by instructors in the behavioral, biological, and physical sciences who wish to create their own numerical simulation models for classroom use, but who are not proficient in writing computer programs. SWIP can also be used by instructors who wish to teach modeling techniques to their advanced students.

SWIP is an extension of the Michigan Experimental Simulation Supervisor (MESS) program series. From the point of view of a student exploring an already developed model, there is little difference between SWIP and the earlier programs in the MESS series. Although some modifications have been made in the syntax for describing experiments and in the nomenclature used, the general format of input and output for students is, in most cases, the same in SWIP as in the earlier programs. Since the use of the MESS system by students has been described elsewhere (Stout, 1974; Stout & Main, 1973), that aspect of SWIP will not be discussed further here.

In the earlier versions of MESS, implementing a simulation model meant writing a FORTRAN subroutine compatible with the MESS supervisory program to produce the simulated data, and preparing a data deck to provide control information and to describe the manipulable variables of the model so that the supervisory program could interpret student input. Although relatively little programming was involved in implementing a model (the models implemented to date comprise between 100 and 500 FORTRAN statements, including a generous proportion of comment statements), the fact that programming was required at all was sufficient to discourage many would-be model builders from striking out on their own.

It would appear that the principal factor deterring more people from building simulation models is the necessity of learning the variety of relatively arbitrary conventions and procedures necessary to write a model in a standard computer language and to make it work. Formulating a coherent model of behavior is, in any case, a difficult information-processing task, but the job of translating such a model into a standard computer language and testing and debugging it can amplify by a large factor the amount of information processing to be done. Any system that hopes to make simulation readily available to a large body of users must drastically reduce the burden of extra knowledge of special conventions and procedures that a model builder must carry with him as he goes about his task. SWIP was designed with the goal of minimizing human information-processing and memory requirements.

The principal features of SWIP designed to ease the model builder's burdens are as follows:

(1) The program combines the functions of editor, compiler, and execution controller so that the user need not deal separately with an editor to update source code, a compiler to update object code, and a loader to load and run the program.

(2) SWIP prompts the user for all necessary information; moreover, if a user should fail to understand a prompting message, he can request one or more levels of more detailed explanation of what is being asked simply by entering a question mark.

(3) Only the minimum amount of information necessary to specify the model unambiguously is required; in particular, the order of computation of variables is deduced by the program from interdependencies among the variables. The model builder, therefore, is not required to state explicitly the order in which the variables are to be computed. This feature also allows the model builder to begin his model by specifying his dependent variables and then work back to the independent variables. The capability of specifying variables in any order can be of considerable use to someone who wants to create his model at the terminal; however, it is debatable whether one should encourage modelers to make up their models on line.

In addition to the major features listed above, there are a number of program facilities of less importance, such as the ability to order the program to back up one or more steps if a mistake has been made; however, a detailed description of every facet of the program is beyond the scope of this paper.

When implementing a model in SWIP, four kinds of information must be entered:

(1) General parameters such as the maximum number of subjects per experiment or experimental group; (2) descriptions of the variables defining the model; (3) tables (if needed); and, (4) information defining illegal combinations of values of manipulable variables so that errors in student input can be detected and flagged.

The critical step in model building is of course describing the variables which define the model. There are three kinds of variables in the SWIP system: manipulable, central, and observable variables. Manipulable variablesare those that are at least potentially available to the students. A student specifies the conditions to be used in an experiment by assigning values to the manipulable variables. Observable variables are used to communicate the outcome of an experiment to the student; observable variables are usually dependent variables. Central variables are also usually dependent variables, at least in a sense. Central variables depend on manipulable variables (and/or random functions), but their values cannot be observed by students. For example, in a simulation on motivation and performance, students might be able to manipulate subjects' needs for achievement by selecting subjects whose test scores on need for achievement fall within a certain range. A subject's test score range would then be a manipulable variable. From this manipulable variable one might calculate a "true" underlying need for achievement score for each subject by transforming the manipulable variable and adding random variation; this "true" score would then be a central variable. One would then use the "true" score to calculate the subject's performance in some experimental task; this score would be an observable variable. Every model must have at least one manipulable and one observable variable.

Every variable has a name and optionally one or more "abbreviations." These "abbreviations" are specified in a simple string-processing format so that a wide range of variant spellings and abbreviations can be used in place of the generic name. Words or abbreviations are limited to 79 characters in length. Manipulable variables are further characterized by lists of keyword values and/or ranges of legal numeric values. An attempt is being made to have SWIP handle vector valued manipulable variables, but that feature is not fully implemented at this time. Central and observable variables are characterized by the formulae used to calculate their values. For any one central or observable variable, there may be a number of different procedures (in the present implementation, up to 100) for calculating values for the variable. Logical expressions called "decision formulae" are used to decide which computational procedure is to be used on any one occasion. After a value has been calculated for a variable, it may be rounded to any precision and clipped or censored. Observable variables are the same as central variables except that a little more information is required about how the variable should be labeled on output and what value, if any, represents missing data.

In addition to information about how to compute the value of a central or observable variable, SWIP also needs information about the level at which the variable must be computed. In SWIP, the level of a variable indicates at what depth in a series of nested DO loops values for the variable should be calculated. How the concept of level is used is perhaps best explicated by examples. In one of the models implemented in MESS Version 3, the outcome of one condition is the number of persons diagnosed schizophrenic in the group defined by the manipulable variable settings. Thus, there is one score per set of cases, and in SWIP this dependent variable would be described as being calculated at the "set" level. In most of the other models implemented to date, one or more scores are produced for each subject; these dependent variables would be described as being computed at the "case" level, "case" being a more neutral word than "subject." In other models, data may be gathered on a given individual subject over the course of several trials under varying conditions. In such repeated measures experiments, the dependent variables are calculated at what is called the "trial" level. For example, in an A-B-A-C paired-associate list transfer paradigm, the dependent variable might be the number of trials to criterion for each subject for each set of lists. This dependent variable would have to be calculated at the "trial" level, since it must be calculated once for the A-B pairing, and then again for the A-C pairing for the same subjects. The nomenclature is confusing, but no better alternative seems to present itself.

Another level, or actually set of levels, allowed in SWIP, is the "series" level. This level permits SWIP to deal with the large variety of timeseries models which are found in many disciplines. In economics, for example, one might want to simulate changes in the consumption, investment, and governmental components of the gross national product (GNP) over a series of points in time. The output from such a model might look like:

```
TIME      CONSUMP      INVEST     GOV

  1          70          20        40
  2          80          25        40
  .
  .
  .
 25         130          44        80
```

The paired-associate transfer model discussed above could be transformed into a timeseries model if the dependent variable were the number of correct responses for each presentation of a list. There would then be as many scores for each subject for each condition as there were trials to criterion. The output from this model might look like:

```
.
.
.
CASE = 5, COND = E
NO. CORRECT

3  5  6  9 10  9 10 10 10

CASE = 5, COND = F
NO. CORRECT

2  5  5  7  8  8  7  9 10 10 10

 .
.
.
.
```

The principal difference in SWIP between a repeated measures model and a timeseries model is that in a repeated measures model the experimental conditions (the settings of the manipulable variables) may change from one trial to the next, whereas in the series model data points are generated across time under constant settings for the manipulable variables. A second difference is that the number of trials in a repeated measures experiment is fixed by the student's design, while the length of a timeseries may vary from one case to the next.

Up to this time, our instructional simulation systems have been able to handle comfortably only a fraction of the experimental designs currently in use, particularly when one considers disciplines other than psychology. SWIP cannot handle every type of design, but it does extend our capabilities significantly

Obviously, SWIP has to be a very large program. It is written in IBM FORTRAN IV (G); including all auxiliary programs and subroutines, SWIP comprises about 10,000 FORTRAN statements including comments and occupies approximately 50,000 32-bit words of storage when loaded into core. SWIP also requires direct-access disk files, one for the program's messages and other data (approximately 80,000 characters), and one for each model. Also, since SWIP interprets computational formulae instead of compiling them, it runs slowly when simulating an experiment; hence, in its present form it is much more suitable as a vehicle for developing models than for routine use by students exploring models. To overcome this disadvantage, it is planned at this time to write another program that will accept as input a SWIP model file and produce as output a FORTRAN subroutine to be used with a modified version of the supervisory program. It seems likely that most users who have smaller computers will have to develop their models on a nearby large computer system and then transport the final product to their own systems to be run on a much smaller student oriented version of the supervisor.

The capacity of SWIP is limited largely by the size of five storage vectors in common. Space is dynamically allocated within these arrays when a model is interpreted, so that the size of a model is restricted principally by the total size of these arrays. Other capacity limitations are now in effect on the currently operating system—principally that the number of variables is limited to 100 and the number of tables to 50—but these restrictions are being eliminated in the version now being implemented at Brown University.

The transportability of SWIP has not been fully determined at this time, but some of the obstacles to transportability presented in earlier versions are now being removed, and it seems likely that it will be possible to implement SWIP in any system having sufficient memory and a direct-access file system which allows variable length binary records. It will, of course, always be easier to transport the program to machines whose memory architecture is similar to that of the IBM 360 (8-bit bytes, 32-bit words).

SWIP is still a system very much in development. Not all of the features are fully implemented, and we are still in the process of discovering those parts of the process of modeling that cause the most difficulty for potential users of the system. When these problem areas are identified, the program can be extended at times to help alleviate the difficulties. One drawback of SWIP is that it is too general and too restricted at the same time. It is too general because modelers who wish to implement simple, straightforward numerical models find themselves confronted with options they had not considered, and of whose significance they are unsure. The system is also too restricted in that SWIP does not have available some simulation techniques, such as those requiring the solution of systems of linear equations or the integration of partial differential equations. For the latter case, it may be possible to develop special versions of SWIP to handle special classes of simulation problems. It is less easy to see what to do about the inexperienced modeler who has trouble selecting a representation for his model. It may be that such people need to start with a much more restrictive simulation system which allows fewer choices.

Hopefully, systems like SWIP will enable us to make available a wider variety of instructional simulation models and to provide a flexible vehicle for teaching elementary modeling. They should also allow us to learn more about the psychology of model construction so that we can provide sophisticated systems for modelers in all scientific disciplines.

REFERENCES

Stout, R. L. Modeling and the Michigan Experimental Simulation Supervisor: An overview and some prospects. Behavior Research & Instrumentation, 1974, 6, 121-123.

Srout, R. L., & Main, D. B. Guide for student users of the Michigan Experimental Simulation Version 3 on MTS. Ann Arbor: Project EXTEND, The University of Michigan, 1973.