SIMULATING WITH SIMSCRIPT*

Yen W. Chao
Consolidated Analysis Centers Inc.
Arlington, Virginia

The SIMSCRIPT programming system is especially designed to facilitate the writing of simulation programs. Digital simulations generally consist of a numerical description of "status", which is modified at various points in simulated time by "events". SIMSCRIPT simulations consist primarily of a collection of "event routines" written by the user describing how different kinds of events in a particular simulated world affect current status and cause future events. Status is described in terms of various "entities", "attributes", and "sets" as specified by the user.

Simulation is presently being used as an analysis and management tool in numerous fields as manufacturing, logistics, economics, transportation, and military operations. Unfortunately, the development of simulation programs, using conventional programming techniques, can be extremely time consuming. The SIMSCRIPT programming system, on the other hand, is especially designed to facilitate the writing of simulation programs.

For the industrial engineer or operations research analyst, the SIMSCRIPT programming language serves as a convenient notation for formulating simulation models. For the programmer, it reduces programming time severalfold as compared to simulations written in FORTRAN and permits relatively easy program modification and expansion. If the analyst and programmer are not the same person, SIMSCRIPT greatly simplifies the problem of communication since the model and the computer program are written in a notation readily understood by both.

There are three aspects of SIMSCRIPT which enable it to reduce the programming time required for simulation. These are its world-view of the model to be simulated; its method of communicating to the computer the world to be simulated; and some features which are useful for programming in general, and thus for simulation programming in particular.

In this tutorial session, two examples (one coded in SIMSCRIPT I. 5[2] and one coded in SIMSCRIPT II. 5[3]) will be discussed in order to demonstrate these three aspects--the SIMSCRIPT world-view, program structure and special programming features of the two most widely used versions of the language.

SIMSCRIPT'S World-View

SIMSCRIPT provides a world view that allows models to be structured in terms of the concepts listed in Fig. 1.

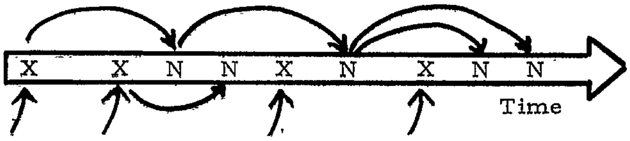
STATUS {
Entities
  Temporary
  Permanent
Attributes
Sets

EVENTS {
Exogenous
Endogenous

Fig. 1-Basic Concepts in SIMSCRIPT World-View

As of any moment of time, the simulated world has a status characterized in terms of how many of which type of entities exist; what are the current values of their attributes; what sets the various entities belong to, and who are the members of the sets which they own. For the sake of programming efficiency, we make a distinction between temporary and permanent entities. Temporary entities can be created and destroyed (can appear and disappear) in the course of a simulation run. Permanent entities do not come and go, but stay with a run from start to finish.

Part of the SIMSCRIPT world-view is an event--a point in time when status changes. Exogenous events are caused from outside of the simulation process. Endogenous events are caused by prior occurrences inside the simulation. Thus, as illustrated in Fig. 2, during the course of the simulation, the exogenous events (vertical arrows) occur at predetermined times, perhaps causing one or more subsequent endogenous events which, in turn, may cause still other endogenous events.



X Exogenous Event
N Endogenous Event

Fig. 2-Schematic Representation of Exogenous and Endogenous Events through Time

The SIMSCRIPT supplied timing routine automatically orders all events in time sequence so that the most imminent event occurs next. Simulated time in the model is advanced from event to event rather than at fixed intervals of time.

SIMSCRIPT Program Structure

A simulation model written in SIMSCRIPT I. 5 usually consists of four parts:

1. Definition Form
2. Events List
3. Event Routines
4. Data { Initialization / Exogenous Events

Definition forms are filled out to describe the variables of the system (i. e. entities, attributes and sets). The names of the variables are recorded, along with their data formats (i. e. integer, decimal, and packing). The events list provides names for each type of exogenous and endogenous event. Event routines written by the user describe the precise changes to the system when an individual event occurs. The initial value of all permanent variables (the number of each type of permanent entity and value of each permanent attribute) are established by data specified by the model builder in the initialization forms. Exogenous event data directly controlled by the user are ordered chronologically according to the sequence of event occurrences. The definition form, events list and event routines are used in compilation by SIMSCRIPT. After compilation of the simulation model, usually only the initialization and exogenous event data are changed for different experimental runs.

Rather than fixed forms, SIMSCRIPT II.5 uses free-form, English-like sentences that make the program easy to write and easier to read. Definition forms are substituted by PREAMBLE statements. Initialization forms are replaced by a user written initialization read routine.

SIMSCRIPT Programming Features

Most of the important features of FORTRAN (e.g. DO LOOP, IF, GO TO, etc.) are contained within SIMSCRIPT so the user has the capability of FORTRAN, but also has all the simulation oriented features of SIMSCRIPT. The timing mechanism, random look-up tables and statistics collection instructions of SIMSCRIPT greatly facilitate the programming of a simulation model. Generalized list-processing capabilities, dimension-free memory, and report generator features of SIMSCRIPT are particularly useful for the development of most computerized models. In summary, SIMSCRIPT is one of the most powerful simulation packages now available on a wide variety of computers.

This conference presentation will describe in detail a single model programmed in both SIMSCRIPT I.5, and SIMSCRIPT II.5 in order to illustrate the different features of the two SIMSCRIPT languages.

*This paper is an updated verions of the 1966 RAND memorandum[1] by Dr. Harry M. Markowitz, the inventor of SIMSCRIPT.

1. Geisler, M., and Markowitz, H., "A Brief Review of SIMSCRIPT as a Simulating Technique," RM-3778-PR, the RAND Corporation, August 1963.

2. Hausner, B., Karr, H., and Markowitz, H., "SIMSCRIPT: A Simulation Programming Language," Prentice-Hall, 1963.

3. Kiviat, P., Markowitz, H., and Villaneuva, R., "The SIMSCRIPT II Programming Language," Prentice-Hall, 1968.