

Using an enhanced general-purpose language for teaching simulation

Andrew F. Seila
Department of Management Sciences
University of Georgia
Athens, GA 30602

At the University of Georgia, two introductory courses in simulation are offered in the College of Business Administration: one to advanced undergraduates (usually seniors) who are majoring in Management Science, and the other to beginning graduate students (Masters and Ph.D.). Students in the graduate course may be majoring in any area of business (Management, Insurance, Finance, etc.), and other academic areas such as Forestry or Agriculture, as well as Management Science. For the past approximately six years I have been using Pascal, augmented by SIMTOOLS as the software for teaching these courses. SIMTOOLS is a collection of procedures, functions and related data structures that allow one to develop discrete-event simulations in Pascal with approximately the same amount of effort that would be required with SIMSCRIPT II.5. Students are expected to know Pascal before starting any of these courses. Most students who take the undergraduate simulation course have taken two courses in Pascal. If graduate students have not used Pascal prior to taking the course, they are expected to learn it on their own.

SIMTOOLS has facilities for (i) random variate generation, (ii) entity and list management, (iii) resource management, (iv) event sequencing, (v) system tracing, and (vi) data collection. The random variate generation facilities include a state-of-the-art random number generator, routines for seeding the random number generator, and twelve functions returning random variates from the most common continuous and discrete distributions. The Pascal RECORD type is used to represent entities. Procedures are available for creating and deleting entities, and for counting the number of entities of each type that have been created, deleted and currently exist in the system. List management facilities include a procedure to initialize a list, several procedures to place entities into a list, depending upon the discipline to be used, functions returning the status of the list (full or empty), the number of entities currently in the list, and pointers to the first and last entities in the list. Also included are procedures to remove an entity from a list and search a list for an entity meeting a user-specified criterion. Resource management facilities include procedures to initialize a resource, use and replenish a resource, and check for a specific amount of a resource.

SIMTOOLS sequences events by creating a list of event notices organized in order of the time of occurrence of the event. Facilities for event sequencing include procedures to schedule and cancel events, search for a particular event, and a timing routine to remove event notices in sequence from the event list and execute its event routine. Event routines are written as regular Pascal routines, and parameters can be passed to the event routines. Functions return the current system time and a pointer

to the event notice of the current event. Tracing facilities allow each type of entity, resource and event to be named, and each entity and resource and event is uniquely identifiable through its name and number (for entities and resources) and its name and occurrence time (for events). When the trace is on, a message is created each time a major action occurs in the system, as for example, creating an entity or scheduling an event. Finally, data collection facilities include procedures to collect discrete and continuous data. Data on list membership (queue lengths), number of entities in the system and resource levels is automatically collected. Thus, SIMTOOLS is a full-featured simulation package with virtually all of the facilities of SIMSCRIPT II.5 or SIMULA. See Seila (1986, 1988) for a more detailed discussion of the features and capabilities of SIMTOOLS.

I will refer to Pascal, augmented by SIMTOOLS as an enhanced general-purpose language. SIMSCRIPT II.5 and SIMULA are also enhanced general-purpose languages since they have all of the features of a general-purpose language and in addition offer facilities for random variate generation, list processing and discrete-event simulation. The difference between these languages and SIMTOOLS is that the facilities for simulation are built-in as language commands; whereas SIMTOOLS provides these facilities in the form of additional procedures and functions in a library. A number of other packages for simulation with Modula-2, C, Ada and other languages are also available. (See L'Ecuyer and Giroux (1987) and Samuels and Spiegel (1987).)

Since this course is offered to students from such a diversity of academic areas, an attempt is made to teach a broader view of simulation than just discrete-event simulation. Topics such as random number and random variate generation, financial models, static models (not involving the evolution of a system over time), and input- and output-data analysis are discussed. Each student must complete three projects. The first involves writing and testing a function to generate random variates from a specific distribution, such as the gamma or beta distribution. The second project involves writing a simulation of a "static" model, running the simulation to generate data, and analyzing the data to evaluate alternative decisions. The important aspect of this project is that it uses random variates, but it does not involve the usual system representation facilities (entity, attribute, sets, events, etc.) associated with discrete event simulation. The third project is a discrete-event simulation which is a major part of the course. Each student is required to propose an individual project, develop the model, code the simulation program, verify and validate the model, make production runs, and evaluate alternative decisions using the model.

I have found that the use of a package such as SIMTOOLS provides the following advantages:

- Many other simulation packages (such as SLAM, SIMAN and GPSS) were developed primarily to model networks of queues, manufacturing systems and other specialized simulation applications. It is easier to use an enhanced general-purpose language to represent models that do not fall neatly into this category, such as corporate planning models, financial models, and others. This is particularly important when teaching simulation to business students and others from diverse backgrounds who may not be primarily interested in Industrial Engineering/Production Management applications.
- Students can gain a better understanding of simulation concepts because I can show them in the Pascal source code for SIMTOOLS how the event scheduling facilities and timing routine work. Since this is "built-in" to the code of languages such as SIMSCRIPT II.5 and SIMULA, the operations of the timing routine can appear mysterious to the student.
- When the student leaves school and begins working in the "real world," they will be better prepared to use whatever software is available to develop simulations. The use of a package such as SIMTOOLS makes it easier to separate simulation concepts from their implementation in the software. Thus, a student who uses a general-purpose language to develop simulations should be able to also do that in another environment, as well as to learn to use special-purpose simulation languages.
- I have found that SIMTOOLS provides better event tracing than many other systems. Good tracing is important for the student to understand how the simulation program works as well as to verify the correctness of the simulation.
- Students already know the basic language syntax. This relieves the instructor from the need to teach the language in addition to simulation concepts and simulation facilities in the language.
- Pascal has advanced features for dynamic memory allocation and data structuring which students may or may not have the opportunity to learn in the first or second course. Since SIMTOOLS uses these facilities, this provides an opportunity for students to utilize these capabilities and enhance their skill in using Pascal.
- A number of other "toolkits" are available for such things as screen management (windows, graphics), database management, processing input, etc. Thus, the tools are available at very reasonable cost for the student to develop stand-alone simulation decision support systems.

- Since SIMTOOLS does not provide a number of features, such as more general priority filing of entities into lists (sets), that are provided by SIMSCRIPT II.5, the resulting program generally requires less space and the code will run faster. This means that larger models can be developed and run on microcomputers.

This approach to teaching simulation has also had some disadvantages, among them:

- Some things, such as passing parameters to event routines and matching event routines to event notices, can be accomplished more easily and naturally in other languages. SIMSCRIPT II.5, for example, allows this as a natural part of the language. On the other hand, working with these aspects of the simulation program is very instructive for the student.
- Sometimes, runtime errors are difficult to diagnose. Since pointers are used, errors involving them can manifest themselves in rather obscure ways. Finding these errors can be confusing and time-consuming for the student.

The objective of the simulation courses that I teach is for the student to learn the basic concepts involved in developing models and writing a simulation program to produce data from the model. The types of models presented are much broader than just networks of queues, or just discrete event systems. I have found that the use of SIMTOOLS has enabled me to meet these objectives more easily than I can expect with special-purpose modelling languages, and the response from students has supported this conclusion.

REFERENCES

- L'Ecuyer, P. and Giroux, N. 1987. "A Process-Oriented Simulation Package Based on Modula-2." *Proceedings of the 1987 Winter Simulation Conference* (Atlanta, GA), IEEE, Piscataway, NJ, 165-174.
- Samuels, M. L. and Spiegel, J. R. 1987. "The Flexible Ada Simulation Tool." *Proceedings of the 1987 Winter Simulation Conference* (Atlanta, GA), IEEE, Piscataway, NJ, 175-184.
- Seila, A. F. 1986. "SIMTOOLS User's Manual." Technical Report. Department of Management Sciences, University of Georgia, Athens, GA.
- Seila, A. F. 1988. "SIMTOOLS: A Software Tool Kit for Discrete Event Simulation in Pascal." *Simulation* 50, No. 3, 93-99.

Andrew F. Seila is an Associate Professor of Management Sciences in the College of Business Administration at the University of Georgia. He received the B.S. degree in physics in 1970 and the Ph.D. degree in operations research in 1976, both from the University of North Carolina at Chapel Hill. Before joining the faculty at the University of Georgia, he was at Bell Laboratories in Holmdel, New

Jersey. His current research interests include all aspects of discrete event simulation, including model development and validation, and simulation output analysis, especially analysis of multivariate output data. He is a member of ORSA, TMS and ASA.

Andrew F. Seila
Department of Management Sciences
and Information Technology
297 Brooks Hall
The University of Georgia
Athens, Georgia 30602 USA
404-542-3586
BITNET: aseila@uscn