Proceedings of the 1984
Winter Simulation Conference
S. Sheppard, U. Pooch, D. Pegden (eds.)

Discrete Event Simulation in Pascal with SIMTOOLS

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

INTRODUCTION

In recent years a great deal of research effort has been devoted to improving simulation software. Products currently available for discrete event simulation include GPSS/H, SIMSCRIPT II.5, SIMULA, SLAM, SIMAN, SIMPAS, PASSIM, and ASSE, to name a few. Some of these products (SIMSCRIPT II.5 and SIMPAS, for example) use process-oriented approaches. All of them assume an entity-attribute-set basis for describing the model to be simulated.

SIMSCRIPT II.5 and SIMULA are general-purpose simulation programming languages. Alternatively, GPSS/H, SIMAN, SLAM, Micro NET, INTERACTIVE and ASSE are "packages" (I won't debate whether the terminology "language" is appropriate or not) that were developed primarily for simulating queueing networks, such as might be found in manufacturing or computer systems. SIMPAS is a preprocessor for a Pascal program that converts "simulation" statements into Pascal code for compilation and execution.

In this presentation we will review the facilities of a collection of data structures and routines for discrete event simulation in Pascal, called SIMTOOLS. This package was developed for use in teaching discrete event simulation and for developing simulations for testing statistical methods in discrete event simulation.

In developing SIMTOOLS, five principles were used:

1. Data structures should be as simple as possible.
2. Procedures and functions should be simple and descriptive.
3. The internal mechanics of the simulation program should be transparent to the user.
4. The source code of event routines should be easy to read and understand.
5. Standard Pascal should be used as much as possible.

SIMTOOLS implements the familiar entity-attribute-set representation in which a system's static structure consists of a collection of entities which have attributes and may belong to sets. In addition, a special type of entity, called a resource, is implemented. SIMTOOLS provides a fundamental set of facilities for creating and deleting entities, inserting them into and removing them from lists, and managing resources.

The dynamic structure implemented by SIMTOOLS is the event view, in which the state of the system changes only at points in time which mark the occurrence of events. The package contains facilities for scheduling and cancelling events, returning the current system time, running the simulation (a timing routine), and stopping the simulation. Some features were added which do not appear to be part of other simulation packages. In SIMTOOLS, one can "enable" and "disable" events, so that, for example, a "disabled" event that is scheduled will not occur. In addition, the timing routine was constructed so that the simulation can be run either until the event list is empty or a specific length of time has elapsed.

Other features of SIMTOOLS include automatic event tracing, which can be turned on and off, and automatic collection of list and resource statistics.

This package has been used successfully in teaching introductory simulation to advanced undergraduates and beginning graduate students. This success appears to come primarily from the fact that the student can see all parts of the simulation, including the timing routine, and that the source code is highly understandable.

Currently, SIMTOOLS is being extended to include the process view of simulation. The new package, SIMTOOLS/P, which contains SIMTOOLS as a subset, facilitates development of queueing models and other production systems.