Proceedings of the 1985 Winter Simulation Conference
D. Gantz, G. Blais, S. Solomon (eds.)

A PANEL ON COMBINED MODELING

A. Alan B. Pritsker
Pritsker & Associates, Inc.
P.O. Box 2413
West Lafayette, Indiana 47906

A simulation language provides an organizational structure to build models that simulate the dynamic performance of systems on a digital computer. The organizational structure allows the variables that define system performance to be described by equations and logical conditions. System descriptions written in terms of mathematical equations are referred to as continuous models. System descriptions written in terms of entity flow or the logical conditions that cause state changes are referred to as discrete models. A simulation language supports the building of continuous or discrete models and, in some languages, combined continuous discrete models.

The organizational structure of a simulation language specifies where and how system equations are written, the procedures for modeling the flow of entities through a process, and the method for defining the logical conditions that affect system variables. Within the defined organizational structure, the time advancement function required to study the dynamics of systems performance is formalized. Furthermore, specific user-written models to obtain system performance updates and system performance changes are identified.

The two main aspects of a simulation language are the method for specifying system status and the procedure for advancing time to update system status. In continuous and discrete simulation models, the system variable description and the time advance procedures are different. In continuous models, a system description is in terms of state variables as modeled by equations. Events occur when state variables cross thresholds. State variable updating is obtained by evaluating the system equations in a step-evaluate-step fashion.

In contrast to this, discrete models including networks describe the status of the system in terms of the number of entities residing in the system and the values of attributes associated with entities. Changes to the status of the system variables only occur with the addition or deletion of entities or when attribute values are recomputed. In discrete simulation, such changes can only occur at event times.

To combine continuous and discrete models, the concept of time advance, status changes and event processing must be integrated. To do this a conceptual change was required which is reflected in the redefinition of an event "as a point in time beyond which the status of a system cannot be projected with certainty."(11,12). This definition allows state variables to change continuously without an event occurring as long as the equations defining the state variables do not change. At the same time, it allows events that cause discrete jumps in continuous variables to be included in the model. It also allows events to occur where a decision is made not to change the status of the system.

For combined simulation models, it is useful to describe events in terms of the mechanism by which they occur. Those that occur at specified projected points in time are referred to as time-events. They are commonly thought of in conjunction with process or discrete event simulation. As previously discussed, those that occur when a system variable reaches a particular threshold are called state-events. Unlike time-events, they are not scheduled to occur in the future but occur when state variables are found to satisfy prescribed conditions.

In a combined simulation model, the behavior is determined by calculating the values of the state variables at the end of steps and making the changes to the values of attributes or to the number of entities in existence at event times. The step size is determined by the executive function so that events can only occur at the end of a step. The executive also considers accuracy requirements, minimum and maximum user prescribed step size values, and output reporting requirements when determining the appropriate size for the next step.

At the end of each time step, the state variables are evaluated to determine if the conditions prescribing a state-event have occurred. If a state-event was passed, that is, a state variable crossed a threshold by more than an allowed tolerance, the step size was too large and is reduced. If a state-event occurs, the model status needs to be updated according to the event logic. As stated above, the executive function sets the step size so that no time-event will occur within a step. This is accomplished by adjusting the size of the step so that the end of the step occurs at the time-event instant.

Time-events are scheduled happenings and are placed on the event calendar. The event calendar maintains events in chronological order based on the time the event is to occur. Two attributes of a time-event are its time of occurrence and its event type. In addition, if the time-event is for an entity, the attributes of that entity must be associated with the time-event. When a time-event occurs, it can change system status in four ways: (1) by altering the value of state variables, (2) by changing the values of attributes of current entities in the system, (3) by altering relationships that exist among entities or state variables, and (4) by changing the number of entities present. A combined simulation language provides methods for accomplishing each type of change when an event occurs.

In this panel, a discussion of the concepts of combined simulation and the trend of research on combined simulation will be discussed. Joining me on the panel are Professor Francois E. Cellier of the

University of Arizona and Professor Donald G. Golden of Cleveland State University. The types of questions that will be discussed by the panel are listed below:

* Should one look at combined modeling as discrete perturbations to continuous processes or as entities of discrete simulations having some attributes that vary continuously?
* Is the educational background of an individual the main determinant of his modeling approach?
* What problems are best suited to a combined approach? To a discrete approach? To a continuous approach?
* Describe some of the applied models that have been developed using a combined approach.
* How were problems/systems modeled before combined languages were introduced?
* Should problems be classified according to an appropriate modeling and analysis method?

References

1. Cellier, F. and A.E. Blitz, GASP V; A Universal Simulation Package, Proceedings of the IFAC Conference, August, 1976.

2. Cellier, F.E., B.P. Zeigler, M.S. Elzas, G.J. Klir, and T.I. Oren, Combined Continuous/Discrete System Simulation Languages --- Usefulness, Experiences and Future Development, Proceedings of the Conference Modelling and Simulation Methodology, Rehovot, Israel, August 14-18, 1978.

3. Cellier, F.E., G. Savastano, G.C. Vansteenkiste, L. Dekker, and A.P. Bongulielmi, The COSY Simulation Language", Proceedings of the 9th IMACS Congress on Simulation of Systems, Sorrento, Italy, September 24-28, 1979.

4. Cellier, F.E., "Combined Continuous/Discrete System Simulation by Use of Digital Computers: Techniques and Tools", Dissertation ETH No. 6483, 1979.

5. Cellier, F.E. and A.A.B. Pritsker, Teaching Continuous Simulation Using GASP, Simulation, Vol. 34, No. 4, April 1980, pp. 137-139.

6. Cellier, F.E.(ed.) , Progress in Modelling & Simulation, Academic Press, New York, NY, 1982.

7. Fahrland, D.A., Combined Discrete Event/Continuous Systems Simulation, Simulation, 14, 1970, pp. 61-72.

8. Golden, D.G., and J.D. Schoeffler, "GSL-A Combined Continuous and Discrete Simulation Language," Simulation, Vol. 20, 1973, pp. 1-8.

9. Pegden, C.D., and A.A.B. Pritsker, SLAM: Simulation Language for Alternative Modeling, Simulation, Vol. 33, No.5, 1979, pp. 145-157.

10. Pritsker, A.A.B., Applications of SLAM, Proceedings, 1981 AIIE Spring Conference, Detroit, Michigan.

11. Pritsker, A.A.B., and N.R. Hurst, GASP IV: A Combined Continuous-Discrete FORTRAN Based Simulation Language, Simulation, Vol. 21, 1973, pp. 65-70.

12. Pritsker, A.A.B., Introduction to Simulation and SLAM II, Second Edition, Halsted Press (John Wiley & Sons) New York, and Systems Publishing Corp., West Lafayette, Indiana, 1984.

13. Pritsker, A.A.B., The GASP IV Simulation Language, Wiley, New York, 1974.

14. Pritsker, A.A.B. and R.E. Young, Simulation with GASP PL/I, Wiley, New York, 1975.

15. Sigal, C.E., and A.A.B. Pritsker, SMOOTH: A Combined Continuous-Discrete Network Simulation Language, Simulation, Vol.22, March 1974, pp. 65-73.

16. Washam, W.B., and A.A.B. Pritsker, Putting Process Interaction Capability into GASP IV, presented at the ORSA/TIMS Joint National Meeting, Philadelphia, 1976.