INTERACTIVE GRAPHICS AND DISCRETE EVENT SIMULATION LANGUAGES

Julian Reitman
Norden Systems, Inc.
Norwalk, Connecticut 06856

ABSTRACT: Graphical presentations show interrelationships that are obscure in tabular listings. Experience, since 1966, with interactive display devices and simulation languages indicates the benefits of clearer presentation of results, more effective user training, less debugging effort, easier setting up of input data, and reduced elapsed time to return meaningful results. The additional effort to include the graphical presentation is small in the overall scope of model development and use.

However, the early promise of widespread use of graphics with discrete event simulation languages has not materialized despite the greatly increased use of the languages. Among the possible reasons for this are: high cost of interactive terminals, general tightening of the economy with adverse effects on high cost projects, continued widespread use of HOL's other than simulation languages, and graphics packages that are different for each terminal and each simulation language.

1. INTRODUCTION

Experienced simulation practitioners have discovered that simulation is the method of last resort. The problem must have resisted other approaches. Therefore simulation is not used on simple problems. Instead, it is reserved for complicated, difficult, interrelated ones. Under these circumstances, cost becomes a very significant item. The need to reduce the cost of system simulation, at a time when computer costs were much higher than they are today, inspired the coupling of interactive graphics with simulation languages as soon as the graphics hardware became generally available.

2. INTERACTIVE GRAPHICS

One early effort started in 1966 when Hunter and Reitman (1968a) coupled GPSS and interactive graphics using the IBM 2250 graphics display terminal. A need for interactive capability occurred when results were obtained after a series of simulation runs which required over an hour of dedicated machine time. During model development, it was discovered that different input data could cause the model to loop. Under the HASP operating system, then in use, there was no output until run completion. Since, the 2250 was in the system as an interactive element, the model could be stopped at an arbitrary point, and after human intervention, it could either continue to the next arbitrary check point or be terminated. Obviously, results were achieved more quickly and with less total effort. The interactive graphics features were further described by Reitman et al (1970b) and Katzke and Reitman (1972c). These techniques were applied to numerous models by Reitman (1971d) and by Preston and Reitman (1976e). In an attempt to reduce the cost of the graphics terminal some of the same programs were modified in 1973 to use the Tektronix 4012 terminal.

Another early project, also using GPSS and the 2250, was Bell's (1968a, 1969b) Graphical Analysis Procedures for System Simulation. The goal was to learn how computer-graphics could aid people in their analyses of simulated systems. The procedure was to run a simulation and store period-by-period results. A second step permited the resultant data to be analyzed graphically. The RAND tablet was used for all human inputs to control the display.

TH0079-4/80/0000-0125$00.75 © 1980 IEEE.
Simulation with Discrete Models: A State-of-the-Art View
T.I. Ören, C.M. Shub, P.F. Roth (eds.)

A third early project, again using GPSS and the 2250, was Stephenson's (1968) Graphical Simulation with GPSS. This need was derived from a requirement to speed the check-out of new models. The Saturn V prelaunch model must be reprogrammed each time the countdown sequence changes. Two weeks of coding and three weeks of check-out were required to get the new model into production. Using graphics to debug on-line, the improved efficiency reduced the check-out time to one week.

A common aspect of these efforts was the development of a set of graphics macros for GPSS. These macros draw lines between starting and ending coordinates from stored data. The internal GPSS structure proved to be the key factor when it came to adding interactive capability. In GPSS the addresses and starting locations are known. The data for a complex simulation may be distributed in many locations representing queues, facilities, storages and matrices. Moreover, it is convenient to locate data needed for graphs, since, the base addresses are known. Similarly, it is simple to resolve an external interrupt generated by the user. The block address is known for the interrupting transaction. Therefore, an interrupt to the current events chain makes it easy to allow the simulation to resume, to end, or to allow the user to change data.

One significant nongraphics enhancement to GPSS accommodated the large amount of data in these simulations. This allowed data matrices to be stored and manipulated on large random access devices and enabled a permanent disk-resident library to store the descriptive data about system elements and scenarios.

Similar considerations resulted in Joline (1971) adding graphics to SIMSCRIPT for computer-drawn motion pictures to visualize and validate airport simulations. A factor that limited simulation for airport planning was insufficient user confidence. As the selection of airport design alternatives involves millions of dollars, the airport planning team must justify their selection in terms that can be readily understood by all. A communication problem may exist between the airport planners and simulation analysts. Computer drawn motion pictures provided a solution to the above problems with added visibility of model operations so that airport analysts determine that there is a real approximation to the way the real airport would operate.

3. USER ENVIRONMENT IMPROVEMENTS

There are many stages in the development of a fully interactive graphics environment. While the efforts in GPSS have reasonably well reached that goal, the efforts in SIMSCRIPT and GASP have made some progress toward that goal. West (1979) has developed SIMGRAPH a set of printer-plot subroutines for SIMSCRIPT II.5. The interactive version of GASP IV has been developed by Fox and Pritsker(1976). In the severe memory constraints of a minicomputer, the interactivity was limited to the presentation of system status without the graphics.

The advantages of pictorial graphics become obvious when the user needs to observe intermediate results as the model is running. The model may be interrupted to get key sets of output statistics for validity checks. An example of the graphics utility, taken from a transportation model, occurs when the symbol for a vehicle appears in an unexpected location. The model logic handles most cases properly, but a particular set of circumstances causes an error. A failsoft error may allow the model to continue with the statistical inference appearing to confirm that reasonable results have been obtained. The pictorial display may be of great utility. Instead of statistical inference, there may be seen an actual deviation from the intended logic. The model is halted, the data are analyzed and the process of isolating the bug begin.

4. SURVEY RESULTS

During the summer of 1980, a survey was undertaken of individuals active in large scale modeling and graphics to provide a modest review of the current state-of-the-art with regard to interactive graphics and simulation languages. The results may be consolidated into the following:

- There is widespread use of interactive graphics terminals. However, the graphics subroutines are not common among terminals. As is to be expected, the most frequent language for the graphics subroutines is FORTRAN. Therefore, the incentive is to use one language for the entire effort - frequently FORTRAN. This is in contrast to the effort required to set up graphics routines for specific terminals and simulation languages.

- Large scale model efforts use simulation languages and the common Higher Order Languages, FORTRAN, PL/1, etc.. Where interactive graphics is a large part of the effort, such as in military war game models, the use of one of the approved standard languages, FORTRAN, has dictated the approach. In some cases, there are plans to make a transition to SIMSCRIPT in the next few years. Another element in military applications is the expected influence of the new Departmant of Defense language, Ada.

- There is a strong desire to retain machine independence and model portability. At this time, there is still a preference for FORTRAN to achieve this goal.

- There are, at present, numerous military and commercial groups on the verge of significant discrete simulation graphics activity. Once the selection of terminals becomes less software dependent, then there will be the broad growth in interactive graphics that was started in the late 1960's. The needs that were determined then are still valid, especially since the scale of simulation activites has increased.

- The choice of simulation language has not been affected by the intended extension to interactive graphics. Those using GPSS, SIMSCRIPT, and GASP all expect to continue using the same language. Some of the FORTRAN users hope to switch. The SIMULA community was not sampled, since it is not prevalent in the United States.

5. SUMMARY

The lack of progress after the bright begining with interactive graphics is probably mostly a function of the high cost of the early terminals. With hardware costs high, the effort to circumvent this with statistical analysis of batch results seemed the most promising avenue of research. When terminals became cheaper, there were so many to choose from, all with different software, that the effort to add capability to one language did not yet seem worth while. Now there appear to be stabilizing forces. Terminals, software, and languages are getting ready for a new burst of activity. Furthermore, for both commmercial and military users displays have proven necessary to handle their complex models. The obvious next step into the future is for these capabilities to be tied into a real-time system for the combined simulation and command and control of on-line systems.

REFERENCES

Bell, T.E. (1968a), "GAPSS Graphical Analysis Procedures for System Simulation", Second Conference on Applications of Simulation, pp. 145-6.

Bell, T.E. (1969b), "Compter Graphics for Simulation Problem Solving", Third Conference on Applications of Simulation, pp. 47-56.

Hunter, S. and J. Reitman (1968a), "GPSS/360 - NORDEN, A Partial Conversational GPSS", in Second Conference on Applications of Simulation, pp. 147-150.

Fox, M.A. and A.A.B. Pritsker (1976), "Interactive Simulation with GASP IV on a Minicomputer", Simuletter/VI/3.

Joline, E.S. (1971), "Applications of Computer-Drawn Motion Pictures to Visulization and Validation of Airport Simulations", Winter Simulation Conference, pp. 304-16.

Katzke, J. and J. Reitman (1972e), "Approaching a Universal GPSS", in Progress in Simulation, Volume 2, Gorden and Breach, New York, N. Y.

Preston, F.S. and J. Reitman (1976f), "Tactical Engagemant Simulation and Graphics", ORSA Conference.

Reitman, J., D. Ingerman, J. Katzke, J. Shapiro, K. Simon, and B. Smith (1970c), "A Complete Interactive Simulation Environment, GPSS/360 - Norden", in Fourth Conference on Applications of Simulation, pp. 260-70.

Reitman, J. (1971d), Computer Simulation Applications, Wiley-Interscience, New York, N. Y., 413 p.

Stephenson, M.E. (1968), "Graphic Simulation with GPSS", Second Conference on Applications of Simulation, pp. 151-5.

West, J. (1979), "SIMGRAPH A Graphics Package for SIMSCRIPT II.5", CACI, Inc..