COMPUTER GRAPHICS FOR SIMULATION PROBLEM-SOLVING

Thomas E. Bell
The RAND Corporation
Santa Monica, California

Abstract

Interactive computer graphics was used while simulating a computer system. The graphics capabilities were extremely valuable for sifting through the large volume of simulation output in order to discover anomalous behavior characteristics. Hard copy options were available, and they were used extensively.

I. INTRODUCTION

Interactive computer graphics is an excellent technique for analyzing computer simulations. Graphical displays present detailed results more satisfactorily than tabular or statistical techniques, and man-computer interaction allows rapid searches through the masses of detailed data from simulations.

Many possibilities exist for implementing such systems. New hardware devices remove limitations on interactive techniques nearly as fast as they are noted, so the number of display methods is limited only by a designers' imagination. The challenge for the graphics designer and the simulation analyst is determining which techniques aid humans most in solving simulation problems.

Earlier investigations with GAPSS* (Graphical Analysis Procedures for System Simulation) indicated that some techniques dominated others in solving the most common types of simulation problems. The ten problem types considered were regarded as equally important. In the present study the superior display types were used as aids in solving real problems that arose in simulating an advanced computer-graphics system. Certain types of problems occured frequently, but new uses of the graphical-analysis facilities enabled the user to solve them.

*GAPSS (Graphical Analysis Procedures for System Simulation) consists of data collection and graphical presentation packages for use with an IBM/360 Model 40 or above, an IBM 2250 graphical-display unit, and a RAND Tablet (see [1] and [2]).

II. DISPLAY ROUTINES

Simulation and analysis are separated in the GAPSS approach. First, a simulation written in the GPSS* language is run and period-by-period results are stored on disk. The results are analyzed graphically in a second separate step. This allows the analyst to view the data in many ways without waiting for repeat simulations.

GPSS HELP blocks collect period-by-period data immediately after the clock has been advanced. Non-zero values of many attributes are written onto disk; all attributes not recorded are assumed equal to zero. Data is collected on the following variables and is referred to by the indicated mnemonics:

Mnemonic	Variable
C	Current value of relative clock
CH	Current contents of User chain
F	Current transaction SEIZing or PREEMPTing a facility
Q	Current number in Queue
S	Current contents of Storage
TB	Current average of Table
W	Current number in Block
X	Current value of Fullword Savevalue
XH	Current value of Halfword Savevalue

This data-collection method increases run times on a PCP system by a factor of two to four. However, the overhead need be incurred only when data are of interest; collection can be initiated and terminated as desired.

Results are presented on an IBM 2250 graphic-display unit that is attached to a dedicated IBM 360/40. A RAND Tablet (see Fig. 1) is used for all human inputs. The tablet remains horizontal as shown; the position of the pen appears on the screen when the operator writes on the tablet. The operator designates which display he desires through this medium.

Fig. 1 - Graphic Display Station

Previous investigation[1] indicated that three of the six displays tested were the most useful. (A display presenting individual states of the system sequentially proved inferior to those displays that indicate many states over time; therefore, no period-by-period display was included in this study.) The three display types are shown in Figs. 2, 3, and 4 and will be referred to as the Statistics display, the Variable Graph display, and the Gantt Chart display respectively.

---

*Simulations are run in GPSS with no restrictions, except that priority class 127 is reserved and 1282_{10} bytes are used by three HELP routines. Though presently programmed for GPSS Parameter=B, they can be altered for Parameter=A or C.

STATISTICS

The Statistics display (Fig. 2) provides the standard deviation, the average value, the minimum value, and the maximum value, as attained during the indicated simulation period for any chosen variables (as many as 14). The meaning of these statistics is straightforward for most variable types. However, the interpretation is less obvious for Facilities (TYPE F).

SUMMARY STATISTICS

COMPUTE FROM __1600_ TO __2000_

VARIABLE TYPE	VARIABLE NO.	VALUES MAXIMUM	VALUES MINIMUM	VALUES AVERAGE	VALUES STANDARD DEVIATION
_S	__1	30.000	0.000	20.982	12.865
_S	__3	19.000	2.000	12.615	7.667
_F	_15	1.000	1.000	1.000	0.000
_F	_16	1.000	0.000	0.252	0.434
_F	_17	1.000	0.000	0.987	0.111
_F	_18	1.000	1.000	1.000	0.000
_F	_19	1.000	1.000	1.000	0.000
_F	__5	0.000	0.000	0.000	0.000
_F	__4	1.000	0.000	0.982	0.131
_F	__1	1.000	0.000	0.030	0.171
_X	_16	3.000	1.000	1.425	0.504
_X	_12	2.000	0.000	0.742	0.680
--	---				
--	---				

STATISTICS	GANTT CHART	VARIABLE GRAPH

Fig. 2 - Statistics Display

If a Facility is idle, its "value" is assumed zero; if it is SEIZEd or PREEMPTed, a value of one is assumed. The maximum value can be used to determine whether a Facility has been in use; if so, the maximum value will be one. Similarly, the minimum value will indicate if the Facility has ever been idle; if so, the minimum value will be zero. The average value (otherwise known as the utilization) tells the portion of time the facility is in use. The standard deviation is not particularly useful for examining a Facility.

VARIABLE GRAPH

The Variable Graph display (Fig. 3) presents graphs of four variables over the desired time period. The analyst must indicate the TYPE and NUMBER as well as the maximum and minimum values to be displayed. The meanings are obvious for graphs of all TYPEs, except the Facilities. For the purpose of this display a Facility may take on values of zero (idle), one (SEIZEd but not PREEMPTed), or two PREEMPTed).

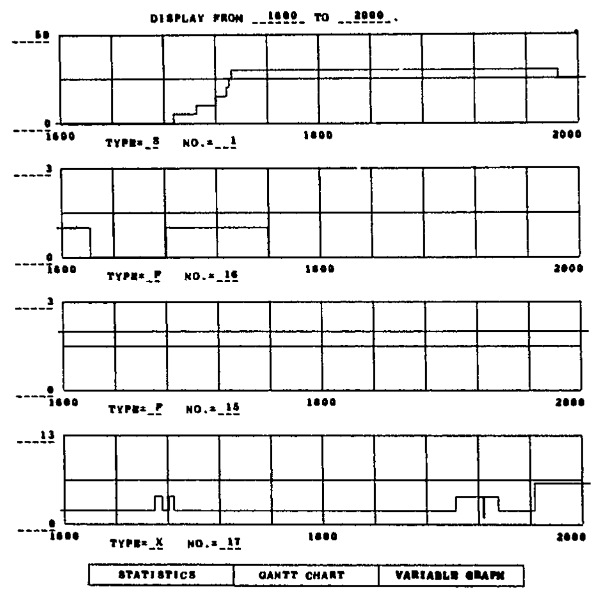

Fig. 3 - Variable Graph Display

GANTT CHART

The top portion of the Gantt Chart display (Fig. 4) is a single-variable graph. The bottom portion is the Gantt Chart. A bar is drawn to indicate the particular time a Facility is in use by a transaction, and the transaction's number is given immediately above the bar, e.g., Fig. 4 indicates that Facility 16 was used by transaction 25 between time 1681 and time 1760.

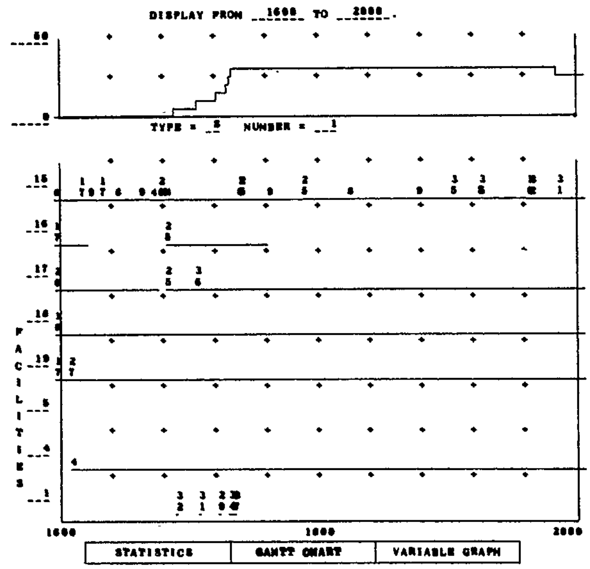

Fig. 4 - Gantt Chart Display

The analyst can change time limits and/or variables by writing in the new values with the RAND Tablet. He can also switch from one display to any other by pressing the pen in the box giving the name of the desired display. Each change to a new display will be referred to as a "display switch," and each change to a new set of time limits or variables will be called a "data change." Each display switch and data change was recorded during display usage and subsequently analyzed.

The analyst can also obtain a permanent copy of the display. He can press one of the 2250 function keys to dump the 2250 buffer on disk. This recorded data is later sent to a SC-4060 hardcopy device for printing. The computer made no record of such requests, but the hardcopy was dated and numbered so that records of hardcopy requests could be made later.

III. THE SIMULATION

The system being simulated is RAND's next-generation graphics system--the Video Graphics System. The simulation's objective is to provide system designers with information about such system characteristics as: 1) the effect of various scheduling algorithms for system resources, 2) the degree of degradation caused by various input devices, 3) the load conditions that may cause slow response to user requests, and 4) the effect of short machine down times.

The simulation is written in GPSS and consists presently of approximately 1000 blocks. GPSS was chosen as the simulation language for two reasons: 1) The graphical-display tools, designed for use with GPSS, increase modeling efficiency; 2) No system description existed, so a language was needed that permitted modeling to begin while learning about the system took place. GPSS is particularly useful in this regard because it uses a predefined structure.

The simulation accounts for all synchronization signals, interrupts, and changes in control in the central computer. This computer is an IBM 1800 that employs cycle-stealing for I/O; the effect of this cycle-stealing on CPU operations is modeled. The base time of the simulation is 50 μsec (about the time required for the 1800 to execute 10 instructions). Investigations of system response are usually performed by running 5 sec of simulated time (about 15 min of real time on a 360/40). In most cases, initialization occurs after about 100 ms of simulated time.

The structure of the modeled system undoubtedly influenced the use of displays. The following description is provided only to convey the general structure of the model; it does not describe the real system or the model with detailed accuracy.

The Video Graphics System is being developed to provide low cost, high capability graphics to many users. Each user sits at a terminal consisting of a television-type screen, a keyboard, a control box, and such other optional input devices as a RAND Tablet or a light pen (see Fig. 5).

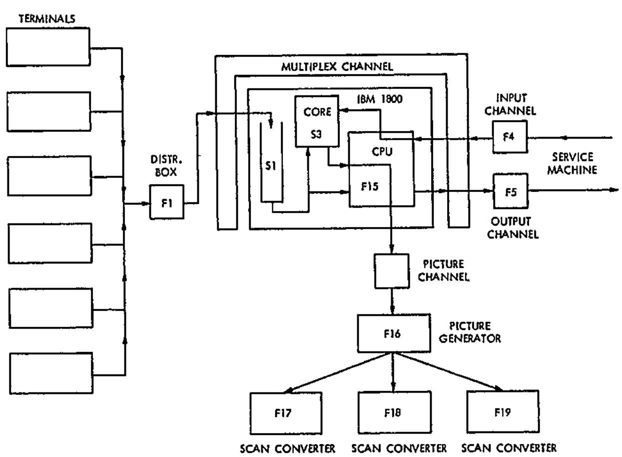

Fig. 5 - Video Graphics System

His inputs go to a distribution system (designated Facility 1 in the simulation) and then to the IBM 1800 CPU (designated Facility 15), which serves mainly as a switching center. Inputs go directly into a finite-length queue (designated Storage 1) and are eventually removed for processing. They can be sent to any of several service machines, mainly system 360 computers, through output channels (e.g., Facility 5) or used for direct feedback to the user. The service machines can also send input to the 1800 through input channels (e.g., Facility 4). These service machine inputs are mainly pictures to be displayed on terminal screens.

This system uses one picture generator, which serves a number of users. The picture generator (Facility 16 in the simulation) converts the digital instructions from the 1800 into analog signals corresponding to each line that will appear on a picture. The signals control an electron beam in one of three modified vidicon tubes (Facilities 17-19) where pictures are created. The scan converter (vidicon tube) then creates a raster image of the painted picture as an FM signal, which is stored on a disk and which continually refreshes the terminal's image. Since painting a picture takes much less time than scanning it, the picture generator can finish painting one scan converter and begin painting on the next while the first is scanning. After painting the second picture, it can begin on the third while the first and second are still being scanned.

The entire 1800 software, developed at RAND, uses no general operating system. The system is interrupt-driven with each type of interrupt assigned an appropriate priority. Processing an interrupt involves executing a routine that processes all inputs at that priority level. In many cases the processing routine enters a high priority level so that essentially new interrupts are masked. When traffic is heavy, input messages tend to be processed in platoon fashion; a number stack up at one priority level and then are processed as a single batch.

IV. RESULTS

The primary objective of the simulation effort was to aid system designers--not to investigate the use of graphics as an aid to simulation problem-solving. During the first two months of simulation effort, no data on graphics usage were collected; such data would have had no application at that time. Unanticipated uses of graphics occurred so data were collected during the following two-month period. The results below were based on quantative and qualitative data obtained during that period.

PREVALENT PROBLEM TYPES

Ten problem types, which were identified in previous work,* are:

1) Cyclical pattern's presence or absence;
2) Particular ordering of items;
3) Queue length;
4) Initialization of system;
5) Change in processing rate;
6) Change in fraction of transactions taking a particular route;
7) Change in input rate;
8) Sensitivity of system to random number stream;
9) Particular occurrence at one time;
10) Anomaly search.

*Pg. 158 of [1]

Table 1

INCIDENCE OF PROBLEM TYPE

Problem Type	Portion of Time
Particular ordering of items	40%
Queue length	25%
Anomaly search	15%
Particular occurrence at one time	10%
Sensitivity of system to random number stream	5%
Initialization of system	5%

Some problem types not noted in Table 1 failed to occur. The rest were solved by using the standard, printed GPSS output without graphics.

INCIDENCE OF DISPLAY USE

The time spent using different displays was not evenly divided over the possible alternatives. Table 2 gives the number of times each display page was requested, the number of data changes in each display, and the total time spent viewing each display.

Table 2

INCIDENCE OF DISPLAY USE

Display	No. of Times	Data Changes	Total Time
Statistics	0	0	0
Variable Graph	4	14	1326
Gantt Chart	26	77	15232

Although the Statistics display was never used, it probably would have been in the absence of the standard GPSS output. Statistics in the standard ouput are easily available to the analyst, are well formatted, and are very comprehensive. Also, they can be examined while the Gantt

Chart or Variable Graph display is on the screen. The most efficient procedure was to examine the printed output and then to view the graphical displays, if the answer was not obvious. Since the statistics had already been examined, the Statistics display was not useful. The special ability of computing statistics over arbitrary time periods was not needed, as indicated in the previous study*.

Problems involving queue length were extremely prevalent (25 percent). The incidence of Variable Graph display usage was lower (8 percent), even though this display was commonly used with this problem type. The 1800 was interrupt-driven, and priorities of individual transactions determined the state the system would take. As a result, a queue's level was not particularly meaningful unless combined with information showing which transaction had control of the CPU; this was given in the Gantt Chart display, but not in the Variable Graph display. The particular characteristics of the simulated system evidently had a strong influence on display usage.

The Statistics display's lack of utility is probably characteristic of any simulation in which printed statistics are available. However, heavy usage of the Gantt Chart, relative to the Variable Graph, is dependent on the particular system being simulated; a simulation of a non-computer system could give opposite results.

*Pg. 64-66 of [1]

DATA CHANGES REQUESTED

Table 3 presents the incidence of types of data changes made in the Gantt Chart and Variable Graph displays. A "V" change occurs when a requested variable or scaling factor on a variable graph is altered (this includes the variable graph on the Gantt Chart display). When the analyst requests a different set of Facilities on the Gantt Chart display, a "G" change has occurred. A "Time" change is recorded when the requested time limits are altered.

Table 3
DATA CHANGE REQUESTS**

Display	Type of Data Change V	G	Time
Variable Graph	6	0	6
Gantt Chart	31	17	73
Total	37	17	79

The low incidence of Gantt Chart change is easily explained; eight Facilities can be examined concurrently on the Gantt Chart display, but only one variable graph is available. Therefore, higher probability exists that a Facility will already be displayed on the Gantt Chart than that a variable will be present on the Variable Graph.

The high incidence of time changes is caused by a different phenomenon. The display area on the IBM 2250 is only ten in. wide; the characters on it must be at least large enough to identify. These hard facts limit the simulation time that

**In some cases two or three types of changes were made at once.

can be displayed while retaining separation of characters. In Fig. 4 a relatively short time span is displayed (400 time units--20 ms), but character separation is poor for Facility 15 and non-existent for Facility 1. Expanding the displayed period reduces separation further and makes the display unreadable. The alternative is to look at a series of displays--each showing a later time frame. The time between presentations of subsequent data sets varies between 15 sec and 70 sec, so relationships between presentations are often missed. Although this alternative has drawbacks, it has been adopted and accounts for the large number of time changes.

HARDCOPY

Hardcopy images of displays were made often during the model's development and analysis. Some of the images were made to communicate potential problems in system design to the designers; they proved to be effective for this purpose, and would have been even more useful had the design project included more than the 10-15 people involved. (In that case, hardcopy could have been mailed to project personnel residing in remote locations.)

Another use of hardcopy was maintaining records of specific situations for later comparison. An alternative to keeping hardcopy might have been to retain all raw data so that the display could have been re-created later. However, this was not efficient because time limits, and facilities requested, for example, would have to be recalled for input to the display. Experience showed that great difficulty was encountered when trying to re-create an old display from memory; noting input data on paper might help, but hard copy images did this and gave the display too.

Time at the 2250 was definitely not a free good in this study; interactive graphics would undoutedly have been used more heavily had equipment been more available. Occasionally hardcopy was made simply because analysis of a display was not completed, but the allowed time at the console was over. Had turn-around time for the hardcopy been a few minutes, this approach might have found application. However, turn-around time was about 24 hours, and analysis was always completed using other techniques before the hardcopy was available.

Many hardcopy prints were made for subsequent analysis by the modeler or to overcome the problem of the narrow time frame possible on the 2250. Hardcopy images were pasted together to make long strip charts. These could then be analyzed to trace detailed interactions through time. This eliminated the time lag between viewing sequential sets of data, so relationships could be readily identified. As model development progressed, complexities increased, and the use of strip charts increased too. Hardcopy was made of 46 percent of all displays, and toward the end of development, more than half the hardcopy was generated for use in strip charts. Many time changes noted in the previous subsection (Data Changes Requested) occurred while the user was collecting hardcopy for the strip charts.

INCREASED EFFICIENCY THROUGH GRAPHICS

Increased simulation problem-solving efficiency was anticipated by using the interactive graphics capability; the potential decrease in effort required was one reason for choosing GPSS as the simulation language. The expectation about graphics' utility seemed to be justified.

Quantitative estimates of graphics' effects on problem-solving time can be little more than rough guesses since the total time for developing the simulation model without the use of graphics is unknown. However, some figures will be given since even rough estimates are often more useful than no estimates at all.

Graphics seemed to increase efficiency in two areas: model definition and code debugging. Early results of simulation runs were shown continuously to the system designers for their comments. These results were graphical, so the designers could easily give immediate feedback about the validity of the model. Many incorrect ideas about the real system were detected early. This probably reduced model definition and coding time by about one-third; irrevalent code was not written and programming traps were avoided. Since a third of the code was not generated, the number of bugs to identify and solve were reduced by one-third.

Certain parts of the simulation code involved interactions throughout the entire program. Debugging these parts was very difficult and was aided significantly by the use of graphics. In one case, a bug caused by a complex interaction was identified in approximately half a day with the use of graphics. Several days later the same bug occurred, but graphics were unavailable in this instance. Though the problem area had been identified previously, two days of analyzing repeated computer runs were needed to realize that the same interaction was at fault. Possibilities for this 75 percent savings would probably occur during about half the total debugging time without graphics, so the total savings in debugging time was about one-third.

In the absence of graphics, model definition and coding would probably require about the same amount of time as debugging. Using this estimate* savings amounted to about one-half the expected time to perform the work without the presence of computer graphics. Though this figure is only an approximation, its magnitude seems to justify the use of GPSS and the graphics tools.

V. DISCUSSION

This discussion is based on data from one analyst working on one simulation; the analyst was also the developer of the graphics tools. Though the points listed below reflect only one data point, that point comes from a real-world application of interactive computer graphics to simulation problem-solving. The relevancy of these points in another type of simulation or with a different analyst is not known.

*Similar estimates are given in [7]

1) The primary uses of the available displays were:
   a) following the particular ordering of transactions
   b) monitoring queue lengths
   c) identifying anomalous situations.

   These may be the major applications in most simulations.
2) Statistical presentations beyond the standard GPSS output were not needed; a special Statistics display was superfluous.
3) Hardcopy capability, which was an integral part of the interactive computer graphics analysis system, was used to communicate discoveries about model behavior, to record interesting situations for later study, and to expand the capability of displays in unanticipated ways.
4) An option would have been useful in the interactive computer-graphics analysis system allowing strip charts to be made automatically. An interactive display large enough to present as much data would be unwieldy and costly.
5) Use of the interactive computer-graphics tools seemed to reduce modeling efforts by approximately one-half.

REFERENCES

1. Bell, T. E., "Graphical Analysis Procedures for Simulation and Scheduling," Unpublished Doctoral Dissertation, University of California, Los Angeles, 1968.

2. Bell, T. E., "GAPSS (Graphical Analysis Procedures for System Simulation)," The RAND Corporation, P-3977, January 1967.

3. Brown, G. D., and C. H. Bush, "The Integrated Graphics System for the IBM 2250," The RAND Corporation, RM-5531-ARPA, October 1968.

4. Brown, G. D., C. H. Bush, and R. A. Berman, "The Integrated Graphics System for the SC-4060. I User's Manual," The RAND Corporation, RM-5660-PR, December 1968.

5. Davis, M. R., and T. O. Ellis, "The RAND Tablet: A Man-Machine Graphical Communication Device," The RAND Corporation, RM-4122-ARPA, August 1964.

6. Haverty, J. P., "GRAIL/GPSS: Graphic On-Line Modeling," The RAND Corporation, P-3838, June 1968.

7. Haverty, J. P., "The Role of Programming Languages in Command and Control: An Interim Report," The RAND Corporation, RM-3293-PR, September 1962.

BIOGRAPHY

Thomas E. Bell was born in Phoenix, Arizona on December 24, 1940. He received the B.S. degree in Applied Physics from the University of California at Los Angeles in 1963, MBA degree in Production Management in 1964, and the Ph.D. in Operations Management in 1968 from UCLA.

His work experience includes electronics engineering, university teaching, and operation research investigations. He has been at The RAND Corporation since 1967 and is presently studying the modeling and analysis of large computing systems.