Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

PRESENTING GPSS/H RESULTS WITH THE GRAPHICAL KERNEL SYSTEM (GKS)

Raphael Parambi
Aseem Chandawarkar
Department of Industrial Engineering & Operations Research
Virginia Polytechnic Institute & State University
Blacksburg, Virginia 24061

ABSTRACT

This paper discusses the interfacing of the latest version of GPSS, namely GPSS/H, and GKS, a graphics system which allows programs to support a wide variety of graphics devices. A computer program to convert a running GPSS/H program to a form compatible for graphical emulation has been developed. An animation program which takes as input the data generated by the modified GPSS/H program has also been developed. Thus, the GPSS programmer needs only to take care of correct modelling of the system under study and need not know anything at all about graphics and animation.

1. INTRODUCTION

The complexity of the interactions between various factors in contemporary industrial situations makes an objective analysis of an alternative a rather difficult task. Analytical models seldom come to the recue of the decision-maker owing to the oversimplifying assumptions that the models usually make. Simulation has, hence, emerged as a tool for mimicking real life situations on a digital computer and seeing the behaviour of a system under different conditions (Schmidt 1985).

Recent years have seen increasing use of simulation models for the analysis of manufacturing as well as other systems. The typical methods of displaying simulation output - a printed trace of the events occurring over time or a statistical summary are often inadequate (Larkin and Medeiros 1983) For simulation to be effective as a design tool, it needs to go beyond providing statistical summary reports. "All viable simulation languages contain the same type of a trace feature that allows the user to march along time and examine the performance of the system 'line by line'. In addition to being a tiring and time-consuming task, much of the inter-relationships and the proper view of the overall system is lost" (quoted from Keramati, Kelley-Sacks and Tonkay 1983). Graphics provides an effective means for displaying the results of simulation runs. Some of the benefits of graphics are:

1. Models with graphics have user confidence to a high degree, and hence the information generated is more likely to be acted upon.
2. Analysts understand the 'messages' of a simulation run much more clearly through animated graphics.
3. Graphics results in increased user involvement.
4. Graphics provides a further dimension to the verification of a simulation model (Hollocks 1984).

2. ON GPSS/H AND GKS

GPSS was chosen for the following reasons:

1. It is a very popular simulation language, the use of which facilitates the modeling of certain types of discrete-event simulations.
2. GPSS offers a rich set of semantics, yet is sparse in its syntax. No other language can be learnt so quickly and is so compact.
3. GPSS has the capability of interfacing with external routines written in Fortran or PL/1. This feature is very useful when reading data from or writing data to external files, thus making it possible for postprocessors to analyze the output of a simulation run (see Schriber 1985 and Bobillier, Kahan and Probst 1976).

A qualitative comparison of GPSS/H, SIMAN, SIMSCRIPT II.5, and SLAMII is given by Banks and Carson (1985). In a quantitative comparison of GPSS/H, SLAM and SIMSCRIPT based on a manufacturing jobshop problem, "GPSS/H was found to compile about 50 times faster than SLAM. GPSS/H executed about 3.8 times faster than SLAM" (quoted from Schriber 1985). Further, GPSS/H is an industry standard general purpose simulation language (Norman 1983).

GKS is a graphics system which allows programs to support a wide variety of graphics devices. It is defined independently of programming languages. GKS has been defined so that it is widely applicable as a graphic system in a range of different environments. In graphics today, there is a wide range of devices on the market for input and output. Depending on whether the application environment is simple or complex, an operator may be working on a single device or a number of devices. GKS caters to all such environments while allowing the application program to make best use of both the total environment available and the specific characteristics of each device (Duce, Gallop, Hopgood and Sutcliffe 1983).

3. GPSS/H - GKS INTERFACE

The use of graphics may be application-oriented or general purpose. The application-oriented approach typically involves building a pictorial model and mapping the results of the model into the pictorial layout. The problem with the application-oriented approach is that the user is restricted to one system or a class of systems. Since GPSS is well suited to the simulation of a wide range of problems like transportation, computers, health services, manpower planning, etc., the objective in constructing the animation program was to provide a general purpose representation of the state of a simulated system over time which would allow rapid determination of system capacity and utilization.

3.1 Working Principle

The input to the whole system is an executable GPSS program. This program is read by a Fortran program, CONVT, which modifies it to make it a graphics compatible program, GMOD. GMOD is run to create a datafile. This data file is read by the GKS program, ANIM, which graphically represents the system under study. A pictorial representation of the structure is shown in Figure 1.

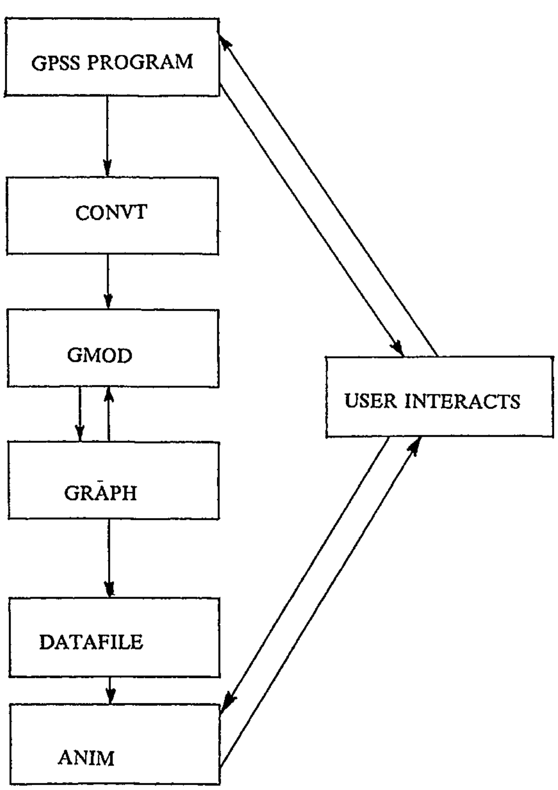

Figure 1. The Working Principle

3.2 Interfacing Methodolgy

Two important concepts in simulation using discrete-event scheduling are the concepts of "event" and "attributes". Events are things that occur at some point in time and change the state of some parameter in the system. Attributes are characteristics of events and entities. An example of an event is: "a machine just finished processing a part" and the attribute is "which machine finished, when it finished, etc.,"

GPSS blocks that could cause events were identified as: QUEUE and DEPART (associated with a queue), SEIZE and RELEASE (associated with a facility), ENTER and LEAVE (associated with a storage). QUEUE, SEIZE and ENTER are associated with incoming transactions and hence are assigned an attribute of +1, while the other three have a -1 since they are associated with outgoing transactions.

As mentioned before, GPSS/H has the capabilty to call external Fortran subroutines. The calling statement has the format

```
BCALL &name (a1, a2, ... , an)
```

where 'name' is the name of the external subroutine (in this case, GRAPH) and 'a1, a2,' etc. are arguments passed on by the GPSS/H program to the Fortran subroutine. The function of CONVT is thus to identify the above mentioned GPSS blocks and insert the appropriate BCALL statement after every occurence of such a block, thereby notifying the occurence of an event. To achieve this, CONVT reads the GPSS program line by line from a file. Every time one of the blocks is encountered the appropriate flag is set on to insert the relevant BCALL statement, if any, after the last line read. For example let us take the following program segment :

```
SEIZE MACH
ADVANCE 9
```

When the SEIZE statement is encountered the flag to print the BCALL statement related to SEIZE is set on. It takes the form

```
BCALL &GRAPH (C1,'MACH','    ','    ','
',200,1)
```

where GRAPH is the name of the external Fortran subroutine, C1 is the time the event took place, 'MACH' is name of the facility, '200' is the identification tag for a facility (100 for a queue and 300 for a storage) and '1' is an indication of an incoming transaction. GMOD, the modified GPSS source program, would now look like

```
SEIZE MACH

BCALL &GRAPH (C1,'MACH','    ','    ','
',200,1)

ADVANCE 9
```

An example of a small GPSS source program and its modified versions are shown in Figures 2 and 3.

```
        SIMULATE
BUFFE STORAGE    2
        GENERATE  3,,1,35
        QUEUE      WAIT
        SEIZE      MACH
      DEPART     WAIT
      ADVANCE    10
    ENTER      BUFFE
      RELEASE    MACH
      ADVANCE    9
        LEAVE      BUFFE
        TERMINATE 1
      START      6
      END
```

Figure 2. A Small GPSS Program

The GMOD program is responsible for creating the data file. During its execution, every time a BCALL statement is encountered, the external Fortran subroutine GRAPH is called. The function of GRAPH is to write the arguments received from GMOD into the data file. An example of the data file created by GMOD is shown in Figure 4. Each line of this file contains

1. The clock time,
2. The name of the queue/facility/storage,
3. An identification number, and
4. An associated +1 or -1.

This data file becomes the input to the graphics program ANIM.

The GKS graphics program ANIM is Fortran based and has in it a number of GKS call statements. Based on the name of the facility/storage/queue read, the appropriate subroutines are activated. A hardcopy of the graphical representation of the 1-machine / 1-storage / 1-queue by GKS is shown in Figures 5, 6 and 7. ANIM is capable of

1. Animating the build-up of queues chosen by the user. A total of eight queues can be displayed simultaneously.
2. Plot the number of jobs waiting for a facility as a function of time.
3. Draw a pie chart showing utilizations of facilities.
4. Display the status of facilities with respect to time.

```
        OPERCOL 60
        EXTERNAL &GRAPH
        LOAD     &GRAPH
        SIMULATE
BUFFE STORAGE    2
       GENERATE  3,,1,35
        QUEUE      WAIT
      BCALL &GRAPH(C1,'WAIT','    ','    ',100,1)
         SEIZE      MACH
      BCALL &GRAPH(C1,'MACH','    ','    ',200,1)
      DEPART     WAIT
      BCALL &GRAPH(C1,'WAIT','    ','    ',100,-1)
     ADVANCE    10
    ENTER      BUFFE
      BCALL &GRAPH(C1,'BUFF','E   ','    ',300,1)
      RELEASE    MACH
      BCALL &GRAPH(C1,'MACH','    ','    ',200,-1)
      ADVANCE    9
        LEAVE      BUFFE
      BCALL &GRAPH(C1,'BUFF','E   ','    ',300,-1)
        TERMINATE 1
      START      6
      END
```

Figure 3. The Modified GPSS Program

Graphics generators fall under two categories: realtime and postprocessing. In the former case, the animation is carried out during the simulation. The problem with this mode is that it is very time consuming and is not very flexible - if the user wishes to rerun the graphics program, the simulation model also has to be rerun which is a time-consuming and costly affair. The solution is postprocessing or running the graphics program independently of the simulation program. The postprocessor reads data from the data file created during the simulation run and uses this for repeated animation runs.

```
1  WAIT    100  1
1  MACH    200  1
1  WAIT    100 -1
4  WAIT    100  1
5  BUFFE   300  1
5  MACH    200 -1
5  MACH    200  1
5  WAIT    100 -1
7  WAIT    100  1
9  BUFFE   300  1
9  MACH    200 -1
9  MACH    200  1
9  WAIT    100 -1
10 WAIT    100  1
14 BUFFE   300 -1
14 BUFFE   300  1
14 MACH    200 -1
14 MACH    200  1
14 WAIT    100 -1
18 BUFFE   300 -1
18 BUFFE   300  1
18 MACH    200 -1
23 BUFFE   300 -1
27 BUFFE   300 -1
```

Figure 4. The Animation Data File

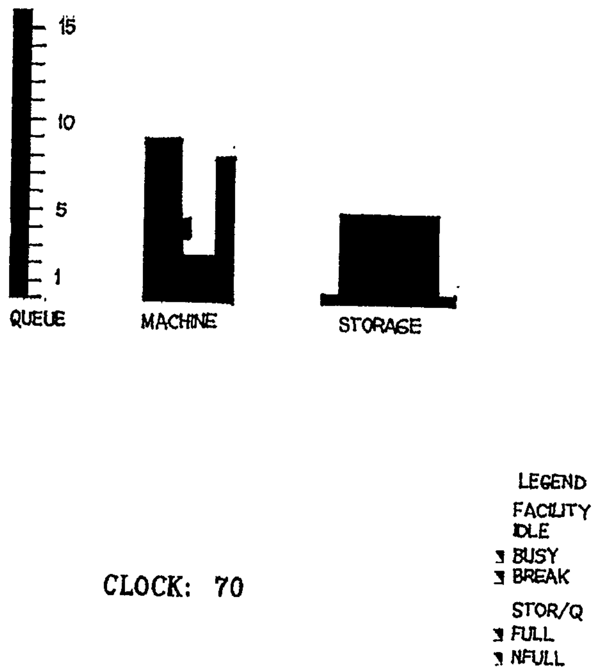

Figure 6. Animation Output - 2

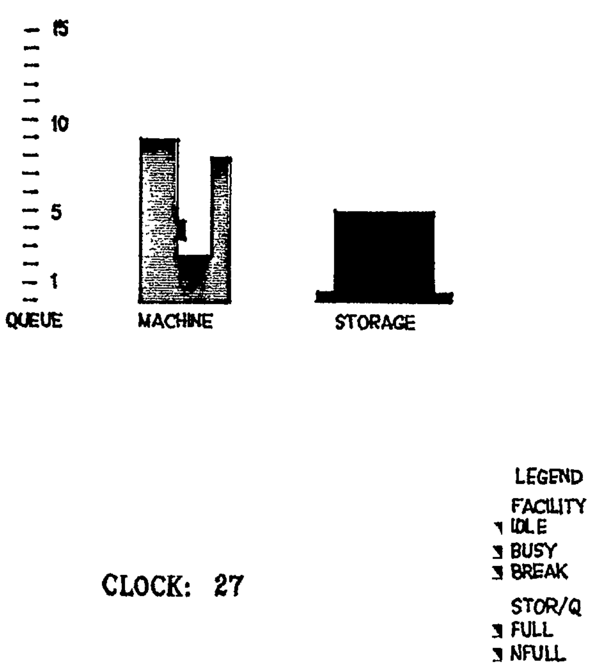

Figure 5. Animation Output - 1

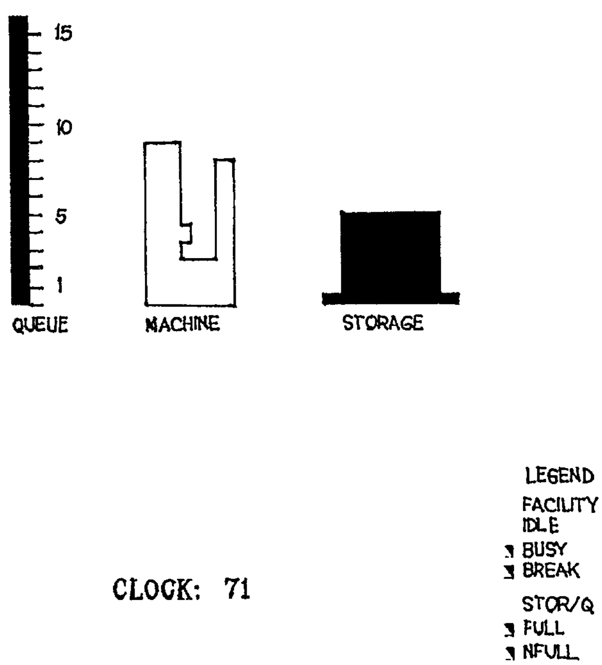

Figure 7. Animation Output - 3

Since GPSS/H is interactive, model input parameters can be changed and the user can view the system response immediately.

The entire operation was executed on an IBM 370 computer with the graphics displayed on a high resolution TEKTRONIX color graphics terminal. Since GPSS/H and GKS are available on the IBM PC there should be no problem whatsoever in implementing a PC version.

4. SUMMARY

The interactive features of GPSS/H and GKS provide necessary and sufficient capabilities for the presentation of simulation results. The build-up of a queue or its plot with respect to time proves very useful to get a clear picture of the transient behaviour of such entities. Since GKS can accept input from a number of input devices future research could be directed towards building the model of a system graphically and using the graphical model as the input to a program which translates it to a GPSS program.

REFERENCES

Banks, J. and Carson, J. S. II (1985). "Process-Interaction Simulation Languages", Simulation, Vol. 44, No. 5.

Bobillier, P. A., Kahan B. C. and Probst, A. R. (1976). Simulation with GPSS and GPSSV, Prentice-Hall, Englewood Cliffs, New Jersey.

Duce, D. A., Gallop, J. R., Hopgood, F. R. A. and Sutcliffe, D. C. , (1983). Introduction to The Graphical Kernel System (GKS), Academic Press

Hollocks, B. W. (1984). "Practical Benefits of Animated Graphics in Simulation", Proceedings of the 1984 Winter Simulation Conference, pp 323-328.

Keramati, B., Kelly-Sacks, C. M., and Tonkay, G. L. (1983). "Simulation and Animation of an Assembly System", Proceedings of the 1983 Winter Simulation Conference, pp 659-661.

Larkin, T. and Medeiros, D. J. (1983). "Animation of Output Applied to Manufacturing Capacity Analysis", Proceedings of the 1983 Winter Simulation Conference, pp 283-286.

Norman, V. B. (1983). "Simulation, Manufacturing and Graphics", Proceedings of the 1983 Winter Simulation Conference, pp 127-131.

Schmidt, J. W. (1985). "Introduction to Systems Analysis, Modeling and Simulation", Proceedings of the 1985 Winter Simulation Conference, pp 3-13.

Schriber, T.J. (1985). "Introduction to GPSS", Proceedings of the 1985 Winter Simulation Conference, pp 46-56.

AUTHORS' BIOGRAPHIES

RAPHAEL PARAMBI is a M.S. student in the Department of Industrial Engineering and Operations Research at Virginia Polytechnic Institute and State University. He received a B.E. in Production Engineering from the University of Madras (1983). His interests are in the areas of CAD/CAM, Graphics and Simulation. He is a student member of SME, IIE and ORSA.

Raphael Parambi
Department of Industrial
Engineering & Operations Research
Virginia Polytechnic Institute &
State University
Blacksburg, VA 24060, U.S.A.
(703) 953-1059

ASEEM CHANDAWARKAR is an Assistant Professor in the Department of Industrial Engineering and Operations Research at Virginia Polytechnic Institute and State University. He received a B.Tech. in Mechanical Engineering from the Indian Institute of Technology, Bombay (1979) and M.S. and Ph.D. degrees in Industrial Engineering from the Pennsylvania State University in 1984 and 1985 respectively. He worked as a Technical Director of a gem diamonds polishing company in Bombay from 1979 to 1982. His current research interests include CAM, scheduling and sequencing, inventory theory and linear programming. He is a member of IIE and ORSA.

Aseem Chandawarkar
Department of Industrial
Engineering & Operations Research
Virginia Polytechnic Institute &
State University
Blacksburg, VA 24060, U.S.A.
(703) 961-6978