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

AN APPROACH TO USING A DATA BASE MANAGEMENT SYSTEM IN MANUFACTURING SYSTEM SIMULATION

Hui Jun Han and Kenneth A. Fegley
University of Pennsylvania
School of Engineering and Applied Science
Department of Systems
Philadelphia, Pennsylvania

ABSTRACT

The organization and management of the data in a computer simulation of a discrete event system can be a significant task. This task can be facilitated by linking a data base management system with the computer language used for the simulation. This paper presents such a linkage using dBASE III for the data base management system and TURBO-PASCAL for the computer language. dBASE provides the means to file system model parameters, simulation input data and simulation output data. Since dBASE III has a capacity of about 10^9 records per data file, the user can store a large amount of experimental data.

Data for several models can be stored and conveniently retrieved, processed and compared via dBASE independently of the TURBO-PASCAL simulation model. The retrieved and processed data can be displayed in graphical or tabular form.

The combined use of the data base management system and a computer language can significantly facilitate many discrete event simulation projects.

This paper presents a manufacturing system simulation model as an example.

1. INTRODUCTION

Many simulation languages have been developed for facilitating the modeling of a system, but the problem of organizing and manipulating the data to be used in the simulation has received less attention.

Simulation projects are typically statistical experiments designed to yield output parameters and confidence limits for these output parameters. To get a good estimate of one or more parameters may require a large number of replications of the experiment and the collection of much data. The retrieval and manipulation of data are conveniently handled by a data base management system. The model itself may require parameters that must be estimated within a preset confidence limit.

In discussing "what-if" analysis, it has been said that "almost all engineering design has to do with the comparative analysis of alternative designs and operating scenarios".[1] If the parameters for alternative model designs are stored via a data base management system the comparative analysis of the alternative designs is considerably facilitated. The comparative evaluation of several simulation runs from the same model is similarly facilitated by having all of the output data from the several runs stored under the control of a data base management system.

In discrete event simulation the average values of system output variables can frequently be used to adequately evaluate the suitability of the system being simulated. In other cases the dynamic behavior of the system is critical and continuous monitoring of system variables is required to assure that no values exceed limits.

Typically the input and output files associated with a simulation program are formatted for or by the particular language or simulation package being used. Hence the files are not usually shared readily with another program nor can the data in the files be readily processed independently of the simulation program.

While a data file produced directly from a simulation program and a data file that was established via a data base management system may contain identical information, there are some distinct advantages to the latter file. The advantages include:

<1> A data base management system (DBMS) allows the data to be accessed flexibly and efficiently.

<2> A DBMS provides for convenient data manipulation and processing that is independent of the simulation program.

<3> The DBMS can act as an interpreter for a high-level programming language, allowing the user to specify what must be done, with only little attention to detailed algorithms.

But what should be the structure of the data base management system that is used to facilitate simulation? While the question has been previously considered[2], the data base management systems used have been special purpose systems, each connected with some special purpose simulation language. Examples include INDECS by Nof and Wilson[3], GPLAN[4] data base system used in the INS simulation language and the SDL5[5] data base system incorporated into the simulation language SLAM by C. R. Standridge and A. A. B. Pritsker.

To provide a flexible, general-purpose simulation package, we have chosen a general-purpose data base system - dBASE III - and a high-level language - TURBO-PASCAL. dBASE provides file management capabilities, a report generator and application development tools that permit the user to customize the DBMS to individual needs. TURBO PASCAL is a structured language that compiles quickly and is easy to use. It includes a convenient editor compatible with Word Star and provides graphics in some versions. Hence the total package includes the most-used DBMS and the high-level computer language that appears to be replacing FORTRAN as the most-used language for scientific programs.

Even though dBASE III is a useful and flexible tool, some effort is required to use it in a simulation environment. An interface is needed between the data base management system (dBASE III) and the simulation model (written in TURBO-PASCAL). This interface links the dBASE system and the simulation model and will assist in the interchange of data between the database and the model. In this paper we address the questions of how to input data to the simulation model and of how to output data from the simulation program to form a data base.

THE LINKING METHOD

The linkage between dBASE III and the TURBO-PASCAL simulation model must accomodate the data that is to be transferred across the interface.

Data handling requirements for a simulation study may include:

1) Storing the output data from the simulation runs;
2) Retrieving the output data and using application programs to obtain a histogram or a time plot of system status.
3) Analyzing the data directly within a data base management system to obtain estimated values for parameters, confidence intervals and probability density distributions.

To facilitate the transfer of data from the simulation program to a file under the control of the data base management system (dBASE III) we have written two programs, as follows:

1) A program to write the data into a temporary storage file.
2) A dBASE III application program which can append the data records in the temporary storage file to a dBASE III data file.

Although the data transfer from the TURBO-PASCAL simulation program to a dBASE III data file can be executed directly under the Disc

Operating System, the time required would take too long if the frequency of data transfer is high. By using temporary storage, the transfer of data to the dBASE III files can be initiated far less frequently and the data transfer time greatly reduced.

The temporary file should be structured in the same manner as the data files in the data base management system. That is, the data files in the temporary file should be arrayed in rows and columns in the same manner as for the files of the data base management system. The data can then be read from the temporary file by the application programs written in TURBO-PASCAL. The same data, after being read by dBASE III and placed in its filing system, can be analyzed and massaged by dBASE III programs.

The temporary file to which the TURBO-PASCAL simulation program writes should be a text file since a general file in PASCAL is a binary file and cannot be read under the DOS.

USING THE TEMPORARY FILE

The temporary data file can be structured to contain all of the recorded data for a series of simulation runs, with each run identified by an ordinal number. Then to retrieve data to draw a time plot of the status of the simulated system or a histogram of a system variable, the ordinal number denoting the sequential number for the desired simulation run is entered.

The application program that is to generate the time plot of the status of the system or the histogram of a system variable may be written in TURBO-PASCAL or in some other language. If the application program is to be written in TURBO-PASCAL, storing the data in a general Pascal file (versus a text file) has a handling speed advantage in that there is no need to convert the data to text form and then reconvert it to binary form. The disadvantage of this approach is that since a temporary file is to be created to permit conversion to a dBASE III file, the use of a general (binary) Pascal file is redundant and uses additional file space. TURBO-PASCAL has good graphics which can readily be utilized whether the user selects general Pascal files, text files or the combination of both file types.

USING THE dBASE FILE.

The dBASE file containing the output data of the simulation runs is easily accessed from dBASE. Hence the application program to construct a time plot of the status of the simulated system or a histogram of a system variable can conveniently be made a part of dBASE.

In addition to the time plots and histograms, it is frequently desired to perform statistical analyses of simulation output data. Statistical analyses of interest include:

1) Distribution Function of Input Variables. Data may have been collected to determine the interarrival time between entities, to determine the time required to service the entities at several workstations, etc. To conveniently represent these data in the simulation it is desired to characterize the data as coming from a Normal, Exponential, or Uniform distribution or from some other well-recognized distribution. Having selected a probability distribution that appears plausible, hypothesis testing is generally used to verify whether or not the selected distribution is suitable. To test the hypothesis that the data does match the selected distribution in a satisfactory manner, a "goodness of fit" test is made. One common test is the chi-square test. This test statistic expression is:

$$\chi_0^2 = \sum_{i=1}^{k} (F_i - E_i)^2/E_i$$

where:

F_i is the observed frequency in the i^{th} interval

k is the number of intervals

E_i is the expected frequency in the i^{th} interval, computed from the hypothesized probability distribution.

The hypothesis is accepted if $\chi_0^2 < \chi^2_{\alpha,k-m-1}$

where: α is called significance level

m is the number of estimated parameters

$\chi^2_{\alpha,r}$ is determined from table of critical values of the chi-square distribution.

2) Estimating the Confidence Interval of the Simulation Output, Such as Average Waiting Time and Average Queue Length.

Suppose that x is an estimated parameter which is a normally distributed random variable with unknown mean u and variance σ^2, respectively. Then a 100(1-α) percent confidence interval MEAN is given by

$$\bar{x} - t_{\alpha/2,n-1} s/\sqrt{n} <= u < \bar{x} + t_{\alpha/2,n-1} s/\sqrt{n}$$

where

$$\bar{x} = \sum_{i=1}^{n} x_i/n$$

$$s = \left(\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2/n\right)/(n-1)$$

and $t_{\alpha/2,n-1}$ is the $\alpha/_2$ percentage point of the t distribution with n-1 degrees of freedom.

3) Comparing the Simulation Output for Two Simulation Models or Comparing a Simulation Model with its Actual Situation.

Assume we have two independent normal random variables, where the first is x_1 with mean u_1 and variance σ^2, and the second is x_2 with mean u_2 and variance σ_2^2. u_1, u_2, σ_1^2, and σ_2^2 are unknown. If $\sigma_1^2 = \sigma_2^2 = \sigma^2$ then comparing the difference of the means;

$$\bar{x}_1 - \bar{x}_2 + t_{\alpha/2,n_1+n_2-2} sp \sqrt{(1/n_1)+(1/n_2)}$$

$$>= u_1 - u_2 > = \bar{x}_1 - \bar{x}_2 + t_{\alpha/2,n_1+n_2-2} sp \sqrt{(1/n_1)+1/n_2)}$$

where $sp = \sqrt{((n_1-1)s_1^2 + (n_2-1)s_2^2)/(n_1+n_2-2)}$

and where s_1^2 and s_2^2 are sample variances.

Calculations such as the one just given are simple but require much data handling. There are many operation commands in the dBASE III which can be used to analyze the statistics, such as counting, summarizing and averaging certain fields of selected records. For example, in calculating the confidence interval $\bar{x} \pm t_{\alpha/2,n-1} s/\sqrt{n}$, the $\bar{x}$ and $\sum_{i=1}^{n} x_i$ can be calculated automatically from AVERAGE and SUM. For $\sum_{i=1}^{n} x_i^2$ we need to move the record pointer to calculate x_i^2 then use SUM to get $\sum_{i=1}^{n} x_i^2$. All of the values in the confidence interval expression can be calculated easily except $t_{\alpha/2,n-1}$. Since $t_{\alpha/2,n-1}$ is often obtained from a distribution table, we set an interactive mode for entering the $t_{\alpha/2,n-1}$ value. A program for calculating the confidence interval of the average waiting time, TAV, is shown in Program 1 below.

PROGRAM CAL.PRG

```
* Calculate confidence interval of TAV
set  talk  off
set  echo  off
?
?
use dsout
input"WORKSTATION NO. - ? " to ok
?"ok',ok
count for numbers=ok to n
?"n",n
input "(N-1) FREEDOM DEGREES DISTRIBUTION
  ?" to tn
?"tn",tn
i=1
s1=0
s2=0
do while i<n+1
   goto (i-1)*7+ok
   s2=s2+tav*tav
   sum for number=ok to s1
   i=i+1
enddo
mean1=s1/n
sn1=sqrt((s2-s1*s1/n)/(n-1)/n)
conf1t=mean1-tn*sn1
conf2t=mean1+tn*sn1
?
?
? "LOWER CONFIDENCE INTERVAL=",conf1t
? "UPPER CONFIDENCE INTERVAL=",conf2t
return
```

PROGRAM 1

As another example of the use of dBASE III for statistical analysis, consider Program 2 shown below for the chi-square "goodness of fit" test to estimate the probability distribution. The main task in this case is counting the observed frequency of a large number of datum. Since there is a conditional COUNT command (COUNT FOR) in the dBASE, the observed frequency F_i is easy to obtain.

PROGRAM TES.PRG

```
*Calculate the XSUM for hypothesis test
set talk off
set echo off
?
?
use coll
i=1
?input "KINDS=?" to kinds
do while i < kinds+1
   ? input" SYMBOL=? " to r
   count for symbol=r to f
   . . . . . . . . . .
   (calculation of a theoretical frequency e)
   . . . . . . . . . .
   xsum-xsum+((f-e)*(f-e)/e)
   i=i+1
enddo
?"XSUM",xsum
return
```

PROGRAM 2

After a xsum (x_o^2) has been obtained, we can judge whether the probability distribution hypothesis can be accepted or rejected.

The simulation analysis is simple and fast in dBASE III, and large data arrays are easily handled.

The linking structure between a simulation model, some application programs and the data base system dBASE III is shown in Fig. 1.

SIMULATING A MANUFACTURING SYSTEM

An example will be presented to illustrate the approach for using a data base system in conjunction with a simulation program. Although this example is for modelling and simulating a simple manufacturing system, it has the generality to illustrate the approach for linking dBASE III and a simulation model written in a high-level language, especially in handling simulation input and output.

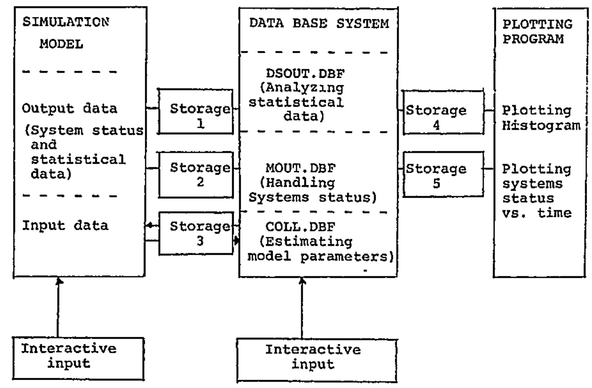

Fig. 1 The Linking Structure

1. The Simulation Problem.

The manufacturing system to be simulated consists of four kinds of facilities as follows:

1) A resource -- to provide workpiece flow.
2) Workstations -- such as machines, robots, inspection facilities, etc.
3) A conveyor - to move the workpieces between workstations.
4) Storage areas -- one storage area to provide queueing space in front of each workstation.

A layout of the facilities is shown in Fig. 2. It provides for up to seven workstations.

The objective of studying this system is to compare the performance of the system under various conditions. Different conditions which may be selected include:

- Layout -- The number of workstations and the arrangement can be selected;
- Probability distribution of the interarrival interval;
- Probability distribution of processing time for each workstation;

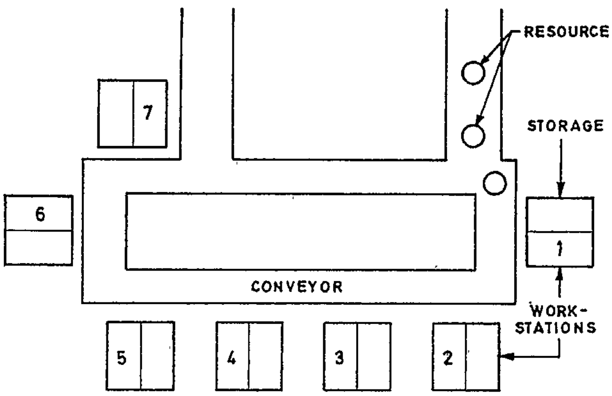

Figure 2

•Ratio of rework workpieces;
•The speed of the conveyor.

The simulation outputs include:

From the simulation model:
- A tabulation of average waiting time, average queue length and utilization of each workstation, and each queue length at the end of the simulation.
- Plots of queue length versus time for each workstation.

From the data base:
- An estimate of the confidence interval for average waiting time, and for average queue length.
- An estimate of model parameters, such as the interarrival interval probability distribution.
- A comparison of two model outputs; that is, an estimate of the difference between the mean confidence intervals. This feature also can be used in comparing actual values with a simulation model output.

2. THE STRUCTURE OF THE SIMULATION MODEL.

There are two main parts in the simulation model. One is called "READY." It allows the user to:
- •Design and select -- A layout graph is shown on the screen. The user can select the desired number of workstations. A graphical animation of the selected arrangement is displayed.
- •Enter model parameters including arrival interval probability distribution, conveyor speed and the distance between each workstation, etc.

The following probability distributions are provided in the program:
- •Constant
- •Uniform distribution
- •Erlang distribution
- •Gauss distribution

Other distributions may be readily added.

Input data are interactively entered by the user, and stored in the storage 3, or are taken from the data base.

The other main part of the simulation model is called "SIMMAIN." It contains numerous PASCAL procedures, including SIMUMAG, GENER, WORK, QUEUE, CLEAR and TRANS.

SIMUMAG manages the simulation events and advances the simulation clock;

GENER generates entitles (jobs) according to the specified arrival interval probability distribution;

QUEUE simulates the queue process, collecting the relative statistical data and adding a "WORK" event to the "event" list;

WORK is used to check the state of a workstation and the corresponding queue to determine if the job can be processed. If the job can be processed this procedure simulates the work process, collects the relative statistical data and placing a "CLEAR" event in the event list. Otherwise the "WORK" event is not executed;

CLEAR is used to clear the workstation, fill in the "event list" with a "WORK" event and collect the relative statistical data.

TRANS separates the rejected jobs and products according to a given ratio.

The appendix gives PASCAL programs for the procedures SIMUMAG, GENER, QUEUE and WORK.

This manufacturing system is a multiple server and single channel queue system. To simplify the simulation a single server model is used to represent all servers. The approach of repeatedly using one server is permissible since each server has its own waiting line. The result is the same as considering all of the multiple servers at the same time. No comparison has been made to determine the relative time required for the simulation when using a single server to represent all servers versus using multiple servers simultaneously.

The service time of each workstation may be different from all others. Each of these service times can be selected from the four distributions used for the interarrival time. All of the input data including the simulation time are stored in storage 3. The queue rule is first come first serve (FCFS). User generated PASCAL procedures can provide other queue rules.

The structure of the simulation model is shown in Fig. 3.

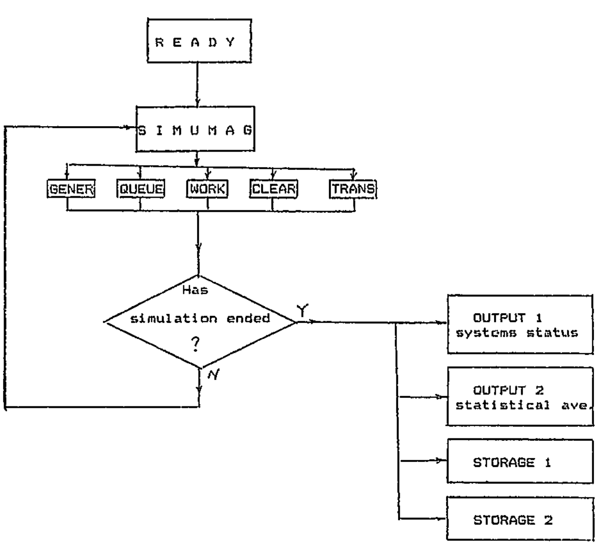

Figure 3

3. AN EXAMPLE

Let us consider a particular layout having three workstations. Workpiece flow is provided by the resource. A layout with three active workstations, Nos. 1, 3 and 7, is shown in Fig. 4. The interarrival interval of the workpieces is selected to be a constant 2 minutes and the conveyor speed is 5 feet/minute. The distance between workstations is specified.

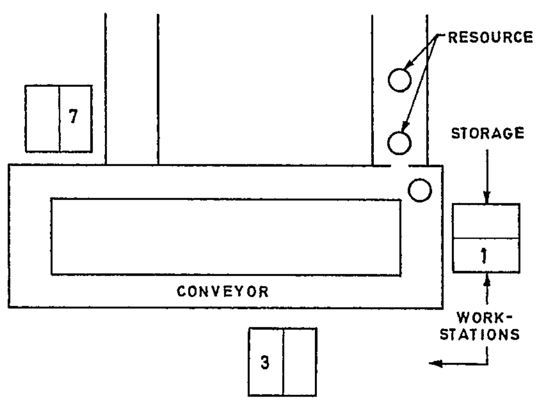

Figure 4

The processing time for each workstation is:

Workstation No.	Prob. Distrib.	Parameter 1	Parameter 2
1	Gauss	mean=2mins.	variance=0.5
3	Gauss	mean=3mins.	variance=1
7	Gauss	mean=4mins.	variance=2

The total simulation time is selected to be 40 minutes. The number of repetitive simulation runs under the same conditions is 10. There are six types of outputs available, as follows:

1) A report of the queue length versus time for all of the ten simulation runs can be obtained from dBASE III.
2) Queue length versus time can be plotted for any of the three active workstations for any of the runs. Figure 5 shows the result for a particular run.
3) A report of the statistical average outputs which include the average waiting time, average queue length and utilization for each selected workstation.
4) A set of histograms for the average waiting time and average queue length can be plotted. The data are taken from storage 3.

5) The confidence interval of the average waiting time and average queue length, estimated in dBASE III, are shown in Table 1.

6) The number of workpieces which have been completed.

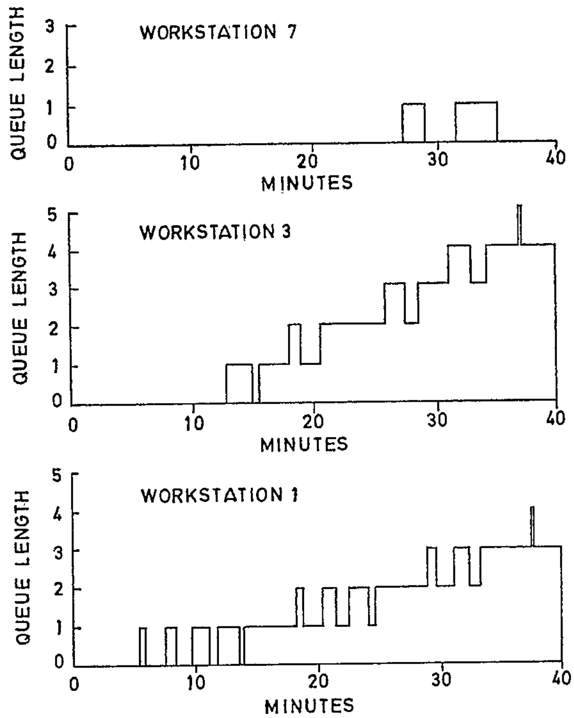

Figure 5

CONFIDENCE INTERVAL FOR AVERAGE QUEUE LENGTH AND AVERAGE WAITING TIME (TEST TIMES=10)

No. 1

α	$t_{\alpha/2,9}$		for average queue length	for average waiting time
0.5	1.38	lower	2.07	1.77
		upper	2.33	1.97
0.1	1.83	lower	2.03	1.74
		upper	2.38	2.01
0.005	2.26	lower	1.99	1.71
		upper	2.42	2.04

No. 3

α	$t_{\alpha/2,9}$		for average queue length	for average waiting time
0.5	1.3830	lower	2.94	1.54
		upper	3.32	1.73
0.1	1.8331	lower	2.88	1.51
		upper	3.38	1.76
0.005	2.2622	lower	2.82	1.48
		upper	3.43	1.79

No. 7

α	$t_{\alpha/2,9}$		for average queue length	for average waiting time
0.5	1.3830	lower	1.07	0.24
		upper	2.05	0.43
0.1	1.8331	lower	0.92	0.21
		upper	2.21	0.46
0.005	2.2622	lower	0.77	0.18
		upper	2.36	0.49

Table 1

CONCLUSIONS

The combined use of a data base management system, such as dBASE III, and a general purpose computer language, such as TURBO-PASCAL, is widely applicable to simulation tasks. The storage, retrieval and analysis of data is facilitated by incorporating the data base management system in the simulation program.

REFERENCES

1. John F. Cassidy, T. Z. Chu, Mike Kutcher, Stanely B. Gershwin and Yu-chi Ho, "Re search Needs in Manufacturing Systems", IEEE Control System Magazine, August 1985, p. 11.

2. C. R. Stanridge and A. A. B. Pritsker, "Using Data Base Capabilities in Simulation", in Progress in Modelling and Simulation, Edited by Francois E. Cellier, Academic Press, 1982, pp. 347-365.

3. S. Y. Nof and R. C. Wilson, "INDECS: General Conveyorized Facilities Description and Simulation", Proceedings, 1976 Winter Simulation Conference.

4. P. E. Joseph and S. D. Roberts, "Utility of Data-Base Management to Analyze the Output from Complex Simulations", Proceedings, 1977 Winter Simulation Conference.

5. C. D. Pegden and A. A. A. B. Pritsker, "The Many Interfaces of Slam", in Progress in Modelling and Simulation, Edited by Francois E. Cellier, Academic Press, 1982.

APPENDIX

PROCEDURE SIMUMAG

```
procedure simumag(var adr,lel:integer;var t,n:real;
                  var ev:evtype);
begin
  lev:=0;
  t:=n+1;
  for i:=1 to lel do
  begin
    if (ev[i,1]>0) and (ev[i,1]<t) then
    begin
      t:=ev[i,1];
      lev:=i;
    end;
  end;
  if lev>0 then
  begin
```

```
      adr:=round(ev[lev,2]);
      ev[lev,1]:=0;
      ev[lev,2]:=0;
      for i:=lev to lel do
      begin
        ev[i,1]:=ev[i+1,1];
        ev[i,2]:=ev[i+1,2];
      end;
      lel:=lel-1;
    end;
  end;
```

PROCEDURE GENER

```
procedure gener(var t,wat,cdt,mean1,mean2,sigma : real;
                var lel,lxac,xacp,distrb1,ntxc,ke,ua1,ub1:integer;
                var ev :evtype; var xac :xactype);
 var dt:real;
    x,y:integer;
 begin
   max:=10000;
   for i:=1 to xacp do
   begin
     if xac[i,1]=0 then
     begin
       lxac:=i;
     end;
   end;
   if lxac<max then
    begin
     ntxc:=ntxc+1;
     if ntxc<max then
      begin
        xac[lxac,1]:=ntxc;
        xac[lxac,2]:=t;
        xac[lxac,6]:=2;
        xacp:=lxac+1;
      end;
    end
    else
     halt;
    if distrb1>0 then
     begin
        case distrb1 of
        1:begin
           erlang(mean1,dt1,ke);
           dt:=dt1;
          end;
        2:begin
           gauss(mean2,sigma,dt2);
           dt:=dt2;
          end;
        3: dt:=ua1+(ub1-ua1)*random;
       end;
    end
    else
      dt:=cdt;
  lev:=lel+1;
  ev[lev,1]:=t+wat;
  ev[lev,2]:=2;
  ev[lev+1,1]:=t+dt;
  ev[lev+1,2]:=1;
  lel:=lev+1;
end;
```

PROCEDURE QUEUE

```
 procedure queue(var lel,lxac,lxac1,m,num:integer; var t:real; var
ev:evtype;
                var xac:xactype; var hql:hqltype;var
ne1,ne2,cun:wstype;
             var qs:qstype);
 begin
         qs[m]:=qs[m]+hql[m,2]*(t-hql[m,1]);
         hql[m,2]:=hql[m,2]+1;
         if num>1 then
         begin
           lxac1:=lxac
         end
         else
         begin
            lxac1:=lxac1+1
         end;
         xac[lxac1,5]:=hql[m,2];
         ne1[m]:=ne1[m]+1;
         aqt[m]:=qs[m]/(ne1[m]+ne2[m]);
         hql[m,1]:=t;
         lev:=lel+1;
         ev[lev,1]:=t;
         ev[lev,2]:=3;
         lel:=lev;
         if num>1 then
         begin
           lxac:=lxac+1;
         end;
 end;
```

PROCEDURE WORK

```
procedure work(var lel,distrb2,xacp,m,num,ke1,ua2,ub2:integer;
               var t,cwt,mean3,mean4,sigma1,cldt:real;
               var ev:evtype;var xac:xactype;var hql:hqltype;
               var qs,aql,aqt,uti:qstype;var ws,ne1,ne2,cun:wstype;
               var xt:xttype;var yl:yltype);
 var dt:real;
     k:integer;
 begin
   i:=m;
   if (ws[i]=0) and (hql[i,2]>0) then
   begin
          ws[i]:=1;
          ne2[i]:=ne2[i]+1;
          if distrb2>0 then
            begin
              case distrb2 of
              1:begin
                 erlang(mean3,dt1,ke1);
                 dt:=dt1;
                end;
              2:begin
                 gauss(mean4,sigma1,dt2);
                 dt:=dt2;
                end;
              3: dt:=ua2+(ub2-ua2)*random;
              end;
            end
          else
            dt:=cwt;
          if num=1 then
          begin
            k:=xacp-1
          end
          else
          begin
            k:=xn[num];
          end;
          for j:=1 to k do
          begin
              xac[j,5]:=xac[j,5]-1.0;
              if round(xac[j,5])=0 then
              begin
                cun[i]:=j;
                 xac[j,3]:=t;
                 xac[j,4]:=t+dt;
               end;
           end;
          qs[i]:=qs[i]+hql[i,2]*(t-hql[i,1]);
          aqt[i]:=qs[i]/(ne1[i]+ne2[i]);
          hql[i,2]:=hql[i,2]-1;
          hql[i,1]:=t;
          lev:=lel+1;
          ev[lev,1]:=t+dt;
          ev[lev,2]:=4;
          lel:=lev;
          cldt:=dt;
          if t+dt>=n+1 then
          begin
            ocupt[i]:=ocupt[i]+(n+1-t);
            uti[i]:=ocupt[i]/n;
          end;
   end;
   if m>m1 then
   begin
    j1:=0;
   end;
   m1:=m;
   j1:=j1+1;
   xt[i,j1]:=t;
   yl[i,j1]:=round(hql[i,2]);
 end;
^Z
```

Hui Jun Han is Associate Professor, Systems Engineering Research Institute, Shanghai Jiao Tong University. Professor Han is the author of three books published by the Shanghai Jiao Tong University Press. Her most recent book is System Simulation. During the 1985-86 academic year she was a visiting scholar at the University of Pennsylvania.

Kenneth A. Fegley is Professor and Chair, Department of Systems, University of Pennsylvania. He has been on the staff at the University of Pennsylvania since 1947. Dr. Fegley is a member of Sigma Xi, Tau Beta Pi, Eta Kappa Nu, a Fellow of the Institute of Electrical and Electronic Engineers and a Registered Professional Engineer in the Commonwealth of Pennsylvania.