Application of Run Time Control to a Multi-Objective, User Oriented Simulation System

David Roggendorff, Dale Rowe

General Electric Company
Huntsville, Alabama

ABSTRACT

The application of simulation as a system analysis tool is growing. One reason for this growth is that user-oriented simulation systems are being developed so that end users can interface directly with simulation systems without support from simulation "experts." Although this is a healthy trend, end users of simulations tend to accept the results of their simulations without regard to confidence levels, stability conditions, or steady state attainment. It is the authors' experience that users tend to terminate simulations before the system has reached steady state in order to conserve computer time. This paper describes a system for automatic run time control of a discrete event simulation. A variety of run time control techniques were identified. Autocorrelation was selected on the bases of simplicity and efficiency and applied to a "user-oriented" simulation system, "Data System Dynamic Simulator (DSDS)." A brief description of DSDS is provided along with results of a sample simulation employing automatic run time control of a DSDS implemented simulation.

INTRODUCTION

User-oriented simulation systems have placed the power of simulation analysis in the hands of the people with the problem. Although this has greatly increased the used of simulation as a system evaluation tool, it has led to several problems as well. Frequently, end users tend to terminate runs before steady state statistics are reached and accept the results without checking for steady state. For this reason, user-oriented simulation systems require automatic run time control built into the system to help insure that incorrect conclusions are not reached from bad data or that run times are not excessively long. This paper describes various techniques for run time control, the selection of autocorrelation as the particular technique for implementation and describes the implementation into the Data System Dynamic Simulator (DSDS) [1] along with results of a sample simulation.

DSDS is a user-oriented simulation system for performing simulation analyses of data processing and communication systems. It was designed to be used by data/computer system engineers to evaluate system throughput, processing delays, timing, control and sizing of data and information systems. Automatic run time control has been implemented on DSDS. It can be enforced on any user selectable set of transit time statistics to establish when steady state is reached. In the example described, run time is controlled on the basis of "message" transit time from source to sink for two different message paths. Termination is automatically executed when the transit time statistics for both paths reach steady state.

ALTERNATE RUN TIME CONTROL METHODS

Determination of how long a stochastic system must run to obtain sufficient sample size to insure steady state can be accomplished in two ways, 1) prior to and independent of model operation, and 2) during the operation of the model. In this paper, we address the second method since end users are usually unfamiliar with performing the analysis required to determine run time on an a priori basis.

Two approaches to automatic stopping rules were considered as suggested by Shannon [2]:

1. Run the simulation in two stages. First, run a sample of size n. Use the results to estimate n* by one of the methods previously described. If $n^* < n$, the run is over. Otherwise, extend the run by $n^* - n$.

2. Use sequential sampling by specifying a minimum n, after which time a subroutine calculates the sample

CH1437-3/79/0455-0463$00.75

1979 Winter Simulation Conference

standard deviation, s. Then compare the quantity

$$\frac{(s)t_{1-\alpha,n-1}}{\sqrt{n}}$$

with d, and the simulation stops when the quantity is $\leq d$ for the first time. It is usually advantageous to do the calculations after every y output rather than after each datum point.

Because the processing of the run time calculation is smaller, the second type of automatic stopping rule was selected.

Similar dichotomies were established and decisions selected for considerations such as regeneration cycles, confidence intervals, confidence levels and the t statistic and methods such as replication, batch means, spectrum analysis, autoregressive representation, and autocorrelation. See [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. These selections were based on the guidelines of having good efficiency and not requiring the user to have a significant knowledge of statistics.

Autocorrelation was selected using the guidelines and considerations stated above.

Autocorrelation indicates those situations where future value of the transit time will be directly influenced by the present value [2]. Fishman [6] and others have shown that required sample sizes are very sensitive to the amount of autocorrelation present within each run.

The mean, variance, t statistic, and autocorrelation are used to calculate the minimum sample size [2].

The observed mean is the sum of each transit time divided by the number of observations in the sample:

$$m = \sum_{i=1}^{n_s} \frac{X_i}{n_s} \qquad (1)$$

where, m = observed mean of the transit times,

X_i = the ith observed transit time, and

n_s = the current number of samples.

The observed variance is the sum of each squared transit time less the number of observations multiplied by the squared observed mean, and divided by one less than the number of observations:

$$s^2 = \frac{\sum_{i=1}^{n_s} X_i^2 - n_s m^2}{n_s - 1} \qquad (2)$$

where, s^2 = observed variance.

Fishman [7] reports that Starr [15] has shown that s^2 for variance erodes a desired confidence level of .95 to .929 at worst. Therefore, we use s^2, the observed variance.

The autocorrelation for each phase lag of the transit times is the sum of the products of the observed displacements divided by the product of the observed variance and one less than the number of observations:

$$\rho_{p,x} = \sum_{i=1}^{n_s-p} \frac{(X_i - m)\ (X_{i+p} - m)}{s^2(n_s - 1)} \qquad (3)$$

where, $\rho_{p,x}$ = autocorrelation of phase lag p of the transit times X, and

p = phase lag on which the autocorrelation is calculated.

Frequently, the autocorrelations at higher phase lags are not significant. Each phase lag is tested for significance using equation (4) below.

The autocorrelation is significantly greater than zero if the students t statistic [2] for the desired confidence level is less than the product of the absolute value of the autocorrelation and the square root of the quotient of the number of observations less the phase lag, less two, and the complement of the autocorrelation squared:

$$t_{\alpha/2} < \left|\rho_{p,x}\right| \sqrt{\frac{n_s - p - 2}{1 - \rho_{p,x}^2}} \qquad (4)$$

$\rho_{p,x}$ is significantly greater than zero.

Otherwise, $\rho_{p,x}$ is not significantly greater than zero.

where, $t_{\alpha/2}$ = the student's t statistic for confidence level specified. See Appendix C-V of Shannon [2].

The confidence interval may be defined as a specific value or as a fraction of the observed mean. We provide the capability for both methods by calculating the confidence interval half-width as a sum of specific value in seconds and a fraction of the observed mean:

$$d = d_s + f \cdot m \quad (5)$$

where, d = confidence interval,

d_s = specified specific value of confidence interval, and

f = fraction specified.

Each autocorrelation through the highest significant lag is used to calculate the minimum sample size by the product of the square of the t statistic, the observed variance and the sum of one and twice the sum of the product of the complement of the quotient of the phase lag and one more than the highest significant phase lag times the autocorrelation, all divided by the confidence interval squared [9]:

$$n = \frac{t^2_{\alpha/2} \; s^2 \left\{ 1 + 2 \left[\sum_{p=1}^{p_m} \left(1 - \frac{p}{p_m + 1} \right) \rho_{p,x} \right] \right\}}{d^2} \quad (6)$$

where, n = the calculated minimum sample size, and

p_m = the highest significant phase lag.

IMPLEMENTATION IN DSDS

Description of DSDS

The Data System Dynamic Simulator is a user oriented simulation tool designed for performing comprehensive design analyses and trade studies of large data processing and communication systems. The system was designed to reduce cost and time of performing simulation analyses of alternate data system configurations. The system simulates timing, control and sizing characteristics of data system elements.

The concept used in DSDS--piecing together real data system elements to form complete data systems --can also be used with DSDS models of these elements. DSDS is built around a fundamental modeling entity called a Data System Element Model (DSEM). A DSEM is a mathematical representation (i.e., model) of the functions performed by real elements typically found in a data system (i.e., multiplexer, tape recorder, sensor, software task, central processor, etc.). DSEM's are interconnected via parametrically assigned output vectors to form a Data System Model. Internal parameters are assigned to particularize the function(s) performed by a DSEM. Each DSEM is automatically "instrumented" to collect the data required for the user selectable system report generators. All parameters required to interconnect DSEM's, assign values to DSEM functional parameters, initialize the system, control simulation run time and select the desired output reports are supplied using free form NAMELIST input on the batch DSDS system or by appropriate input when using the interactive DSDS system. The system model can be modified or reconfigured via user input parameters without modification of the system source code. This minimizes the amount of time required for verification when a data system model is changed.

Data System Modeling Using DSDS

A data system is modeled by selecting, interconnecting, and parameterizing Data System Element Models. There are 130 DSEM's which permit modeling of:

Data Generation	-	sensor, experiment, data generator
Data Transfer	-	transmitter, receiver, communication link
Data Storage	-	tape recorder, disc, RAM
Data Switching	-	signal switching unit, mux/dmux
Data Conditioning	-	A/D, D/A, decoder, format conversion unit
Data Output	-	line printer, interactive console, encoder
Operations	-	production, shift scheduler, personnel scheduler
Distribution	-	mail service, user
Computers	-	CPU, software, teletype, computer interface unit
Logic	-	and, or, inverter, flip flops
High Level Elements	-	spacecraft, ground station, OCC, NASCOM

The steps involved in creating a Data System Model are:

DSEM Selection	-	The proper DSEM's to represent the data system are selected.
DSEM Interconnection	-	The data and control outputs of each DSEM are "wired" to the appropriate inputs.
Parameter Selection	-	Each DSEM has parameters which control data conversion, and processing delays (either fixed or statistical). Default values are supplied for all parameters. The simulation user can either use the default values or override any parameter to particularize his model.
Simulation Control	-	Parameters to control simulation duration and starting date and time are specified.
Report Selection	-	The desired system performance reports are selected for output.

The Device Utilization Model (DUM), one of six DSDS system core models, provides nine output reports which are tailored to data system analysis. The type of data supplied by the DUM report generators is shown in Table 1, "Device Utilization Model Statistics Reported for Each Element and for the Total System."

TABLE 1

DEVICE UTILIZATION MODEL STATISTICS REPORTED FOR EACH ELEMENT AND FOR THE TOTAL SYSTEM

STATISTIC	MEASURED FOR		DESCRIPTION
	ELEMENT	SYSTEM	
Throughput	Yes	Yes	Provides a measure of the average rate at which data passes out of the system and through each element's data output pins.
Transit Time	Yes	Yes	Provides a measure of the time required for data to pass through an element, from point-to-point, or from its origination point to a user.
Utilization	Yes	No	Provides a measure of the percent of time each element is utilized.
Q-Statistics	Yes	N/A	Provides the following measures for Time in Q: Mean, σ, Min, Max, Final Value, Total through the Q.
Storage Statistics	Yes	N/A	Provides the normal Q-statistics plus storage capacity and Mean, σ, Min, Max, and Final Value for percent of capacity used.
Lost Data	Yes	Yes	Measures the percent of data lost for each element and a history of the messages lost for the system.
Throughput Timeline	Yes	Yes	Measures the throughput as f (time) for any element or system output.

σ = Standard Deviation

Method of Entering Data

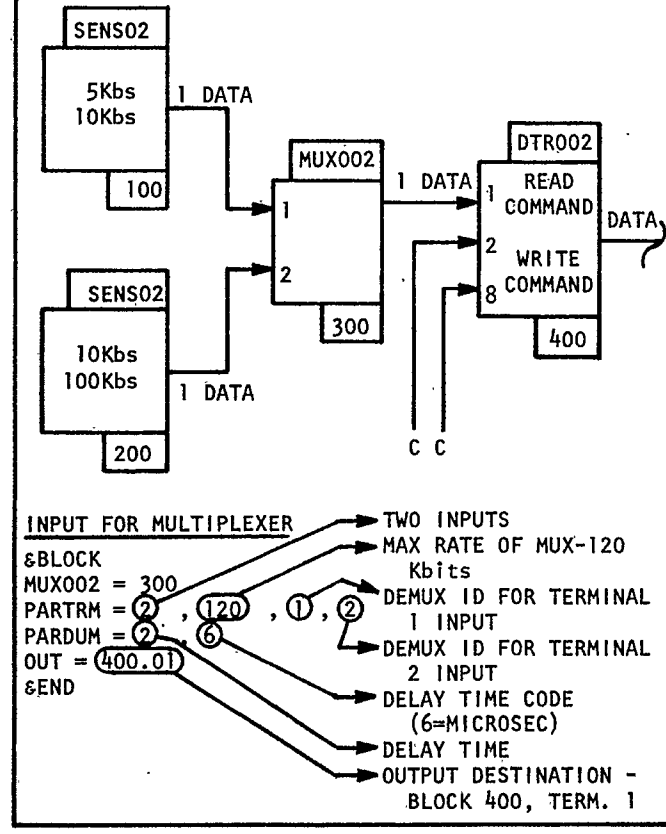
Figure 1, "Partial Model Diagram," shows a simple model block diagram of a portion of a data system. The model shows two sensors, connected to a multiplexer which in turn feeds a digital tape recorder.

To enter the model description of this abbreviated model, the user simply defines the characteristic parameters of each block and the inter-block connections. For example, to specify a multiplexer having a built-in delay of 5 microseconds, and whose output feeds the Digital Tape Recorder shown in Figure 1 as block 400, the following cards are keypunched for the batch mode.

INPUT	EXPLANATION OF INPUT
&BLOCK	Flags start of a new DSEM Block Description.
MUX002=300	A level 2 model of a multiplexer is to be defined as Block 300.
PARTRM=2, 120, 1, 2	Sets parametric values as defined in Figure 1.
PARDUM=2, 6	
OUT=400.01	Output is directed to Block 400, Input Terminal 1.
&END	Flags end of DSEM Block Description.

FIGURE 1

PARTIAL MODEL DIAGRAM



Once a Data System Model has been generated and input, it is run under control of the DSDS executive. The executive reads the simulation user's input description, checks for input errors and sets up the necessary arrays for execution of the simulation. Data flow within a DSDS system model is achieved by creation of events which are passed from DSEM to DSEM. The events are passed through a time ordered EVENT FILE under control of the DSDS executive. When an event arrives at a DSEM, it initiates the process or activity modeled by that DSEM. At the end of the process activity, an event is created and passed to the next DSEM via the EVENT FILE. In each DSEM, whenever an event arrives or leaves, statistical data required by the report generators is recorded.

The first part of a three part example taken from the DSDS Training Manual is shown on the following pages. It illustrates the method of inputting the parametric data to describe the model and several of the output reports from Device Utilization Model (DUM).

Incorporating Automatic Run Time Control Into DSDS

In order to implement run time control into DSDS, a method of allowing the user to select 1) which transit time statistics should reach steady state before simulation shutdown, 2) the desired level of confidence at shutdown, and 3) the accuracy (confidence interval) necessary to achieve the desired level of confidence. The method developed had to be consistent with DSDS modeling methodology.
modeling methodology.

The user specified which transit time statistics he desired to have reach a steady state by placing THRUST and THRUOT blocks into the system model such that messages of interest passed through 1) a THRUST block, 2) one or more system element blocks to 3) a THRUOT block as illustrated in Figure 2, "Placement of THRUST and THRUOT Blocks."

In the simple implementation described above, as each message or message transaction passes through a THRUOT block, the minimum sample size is calculated in accordance with user supplied parameters. The message transit time from the THRUST block to the THRUOT block is used as the observed parameter. These parameters and sample values are listed in Table 2, "THRUOT Run Stopping Parameters."

The stopping count (Parameter 1) is set equal to the number of THRUOT's to have their stopping rules satisfied before the simulation is to be stopped. As each THRUOT is satisfied, it sends an event to the "Shutdown THRUOT" which decrements the stopping count and sends an event to DSDS to stop the simulation when the stopping count goes to zero. The user must connect the second output pin of each THRUOT to the second input pin, the Shutdown THRUOT.

The minimum sample size (Parameter 1) is set to specify the number of messages to pass thru the THRUOT before any calculations are performed. The user is admonished to set this value high enough to include many regeneration cycles to avoid simulation startup biases [7].

FIGURE 2

PLACEMENT OF THRUST AND THRUOT BLOCKS

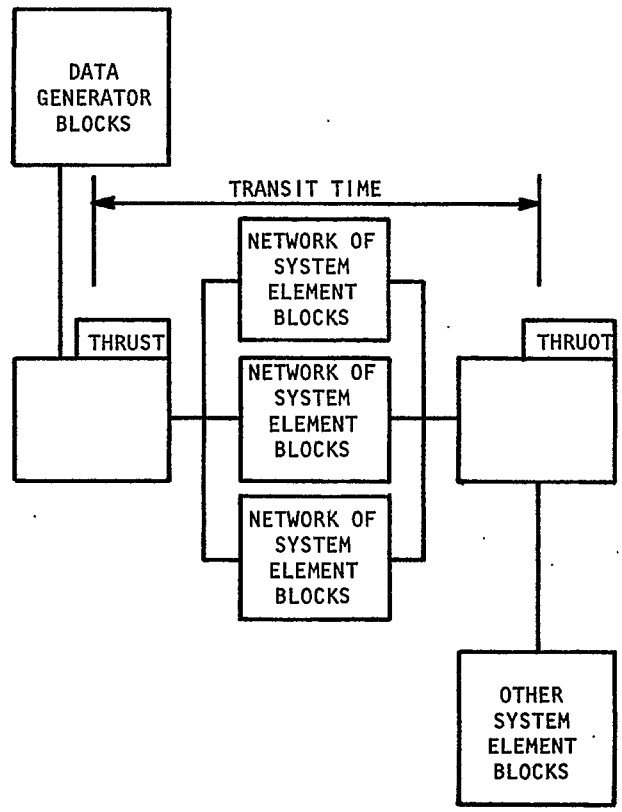


TABLE 2

THRUOT RUN STOPPING PARAMETERS

NO.	DEFAULT VALUE	USE
1	1	Number of THRUOT blocks to have their minimum sample size satisfied before stopping the simulation.
2	25	Minimum sample.
3	1	Sample increment to next calculation.
4	0	Absolute value of maximum underestimate (E) of the mean transit time.
5	.1	Fraction of observed mean to be the maximum underestimate (E) of mean transit time.
6	.95	Probability that the Real Mean Transit Time is no higher than the Estimated Mean Transit Time (m) plus the Maximum Underestimate (E).

The sample increment (Parameter 3) may be set higher when the total model is small enough to make the minimum sample size calculation processing significant.

The absolute delta error (Parameter 4) may be set to the value, in seconds, of the confidence interval half-width.

The fractional delta error (Parameter 5) may be set to the fraction of the observed mean that is to be the confidence interval half-width.

The confidence level (Parameter 6) is used to select the student's t statistic to be used as a parameter of the minimum sample size calculation.

Each THRUOT has several arrays to store data while calculating the minimum sample size. The pilot DSDS system modification used in this study retains the current 300 transit times, with the assumption that only nice distributions will occur.

The calculation of the minimum sample size involves equations (1-6) found and explained in the previous section of this paper, the current transit time values, the number of current transit times, and the total number of transit times.

Sample Simulation Description

The simulation model selected to demonstrate the multi-objective run time control is a portion of a model used to evaluate the relative performance of six minicomputers. It was a timed test involving the performance of text editing from eleven interactive terminals (KVDT) while concurrently copying data files, executing two simultaneous application programs, and performing a COBOL compilation.

The primary measure of performance in this test was the response time for interactive text editing. The purpose of the other tasks was to provide a simulated environment to consume and compete for resources. The number of transactions completed in the applications programs, the number of records converted, and the operating system overhead provided secondary measures of performance from the simulation.

The model is depicted in Figure 3, "Hardware and Software Modeled." Each task competes for scheduling of common resources. The task models are depicted in Figure 4, "Operating System Overhead - Task 1, Memory Swap - Task 2, Interactive Software Development - Tasks 3-13."

FIGURE 3

HARDWARE AND SOFTWARE MODELED

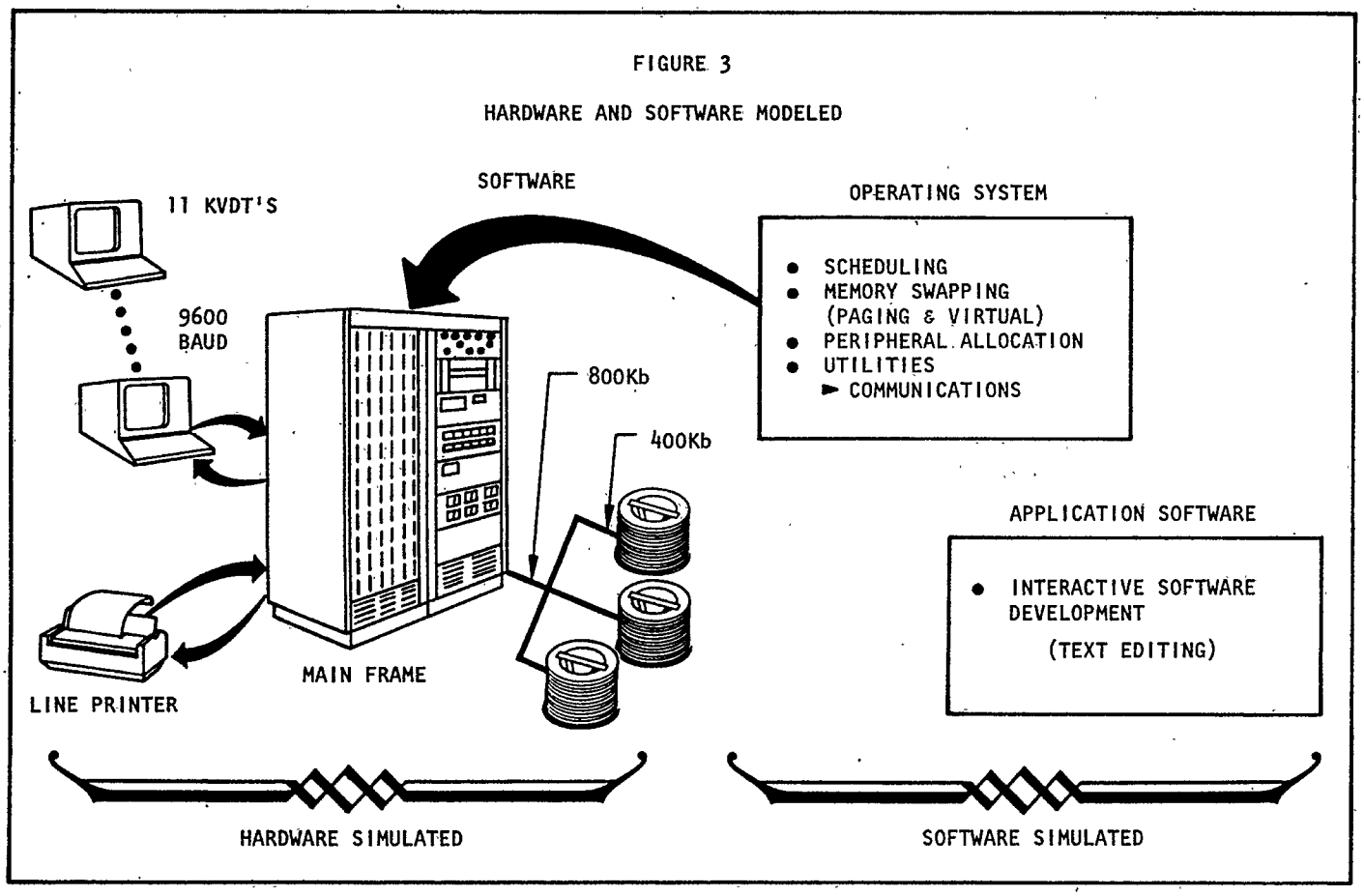
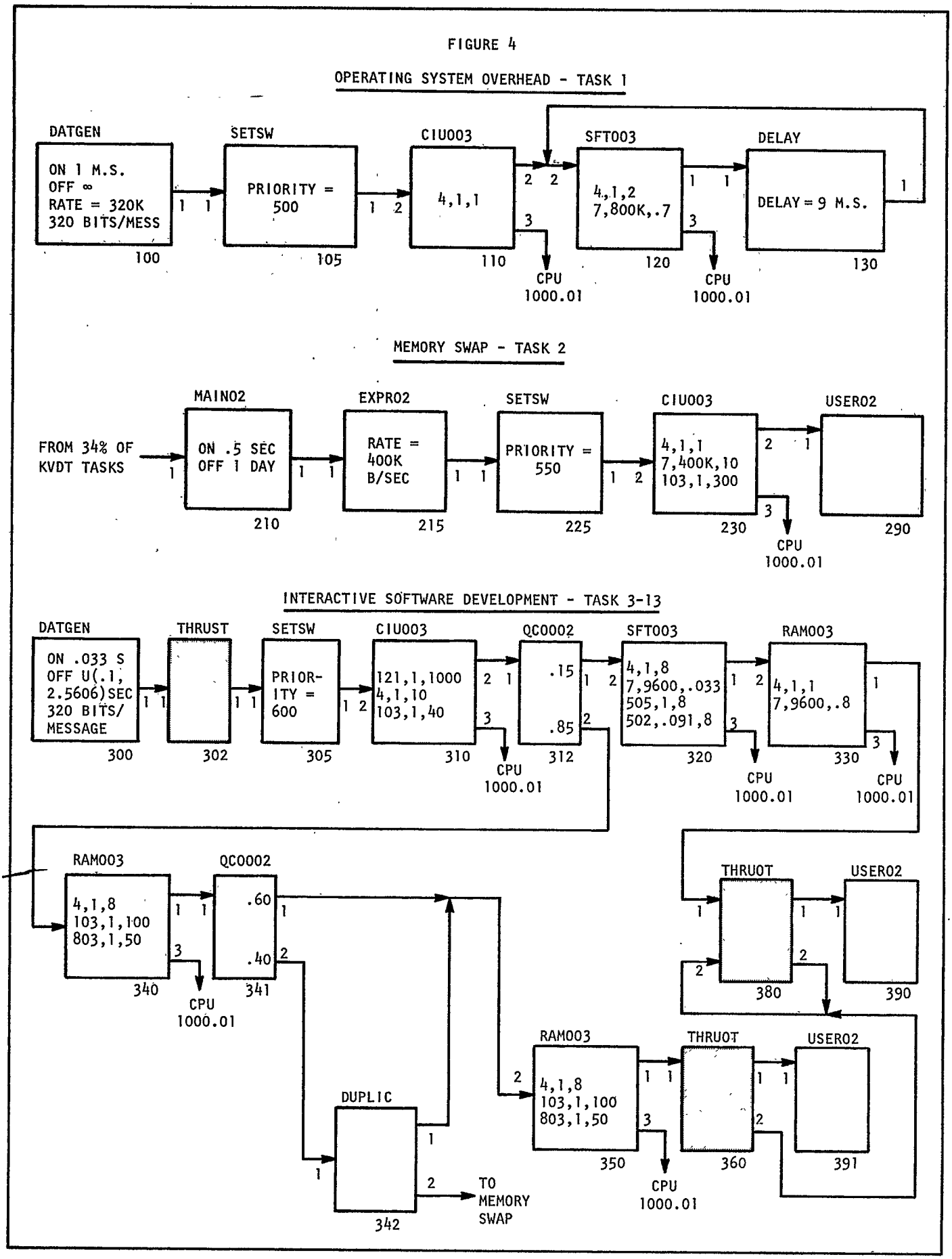


FIGURE 4


OPERATING SYSTEM OVERHEAD - TASK 1
DATGEN
ON 1 M.S.
OFF ∞
RATE = 320K
320 BITS/MESS
100
SETSW
PRIORITY = 500
105
CIU003
4,1,1
110
SFT003
4,1,2
7,800K,.7
120
DELAY
DELAY = 9 M.S.
130
CPU
1000.01
CPU
1000.01
MEMORY SWAP - TASK 2
FROM 34% OF KVDT TASKS
MAIN02
ON .5 SEC
OFF 1 DAY
210
EXPR02
RATE = 400K B/SEC
215
SETSW
PRIORITY = 550
225
CIU003
4,1,1
7,400K,10
103,1,300
230
USER02
290
CPU
1000.01
INTERACTIVE SOFTWARE DEVELOPMENT - TASK 3-13
DATGEN
ON .033 S
OFF U(.1, 2.5606)SEC
320 BITS/ MESSAGE
300
THRUST
302
SETSW
PRIOR-ITY = 600
305
CIU003
121,1,1000
4,1,10
103,1,40
310
QC0002
.15
.85
312
SFT003
4,1,8
7,9600,.033
505,1,8
502,.091,8
320
RAM003
4,1,1
7,9600,.8
330
CPU
1000.01
CPU
1000.01
CPU
1000.01
RAM003
4,1,8
103,1,100
803,1,50
340
QC0002
.60
.40
341
CPU
1000.01
DUPLIC
342
TO MEMORY SWAP
THRUOT
380
USER02
390
RAM003
4,1,8
103,1,100
803,1,50
350
THRUOT
360
USER02
391
CPU
1000.01

Operating system overhead was modeled as a single task with only one activity. This was the highest priority task.

The overhead encountered from the need to swap memory space is modeled in Task 2.

Interactive Software Development is the principal activity modeled. The model was accelerated 11 times to represent the concurrent use of 11 interactive terminals. Individual activities were modeled when an obvious distinction occurred between principally CPU and principally I/O functions. The stochastic nature of software development was modeled to account for the differing task magnitudes when the editing is a continuation of present work versus the location of new work. A portion of the time called for activation of the memory swap task. These stochastic processes were implemented by Monte Carlo techniques.

For this demonstration, three blocks were added to Tasks 3 through 13. The quick response transit time is the time for a message transaction to go from the THRUST block 302 to the THRUOT block 380. The memory swap response time is the message transaction transit time from the same THRUST block to THRUOT block 360.

The triplets inside the other blocks represent the 1) resource code, 2) quantity, and 3) time in milliseconds used each time a message transaction is passed through the block. These resources are allocated by task priority when available and released upon completion according to the time specified by the activity block and size of the data transaction.

AUTOMATIC RUNTIME CONTROL RESULTS

The simulation model used to demonstrate the autocorrelation stopping was taken from a model that was run with a sample size of approximately 125. This demonstration run shows that the confidence for the mean transit time would be low for such a run length.

The first 25 samples of the transit time for the slow response are listed in Table 3, "Transaction Response Times with Memory Swap." These represent the times for the transactions to pass through blocks 305, 310, 312, 340, 341, (possibly 342), and 350, as seen in Figure 4, above. Only blocks 310, 340, and 350 use computer resources and add to the transit time.

After 150 samples of transit time, the autocorrelation was calculated through 25 lags as shown in Table 4, "Statistics for Each Autocorrelation." The confidence level used was 0.9975 to test for the autocorrelation as significantly greater than zero.

TABLE 3

TRANSACTION RESPONSE TIMES WITH MEMORY SWAP

NO	TRANSIT TIME	NO	TRANSIT TIME
1	1.0775208	14	0.0889031
2	1.1023910	15	2.4294815
3	0.1212915	16	1.0852468
4	0.0653685	17	0.4727072
5	1.7270277	18	3.2372255
6	0.9717969	19	2.5588303
7	0.1182494	20	0.9973438
8	1.1147590	21	0.9852446
9	1.0227640	22	1.1002066
10	0.1184105	23	1.0947452
11	1.1027319	24	1.0810652
12	0.5738835	25	1.2800381
13	1.1795151		

TABLE 4

STATISTICS FOR EACH AUTOCORRELATION

LAG	AUTOCOR	T CALC	SIGNIFICANT
1	0.5133018	7.2516823	YES
2	0.2566107	3.2080603	YES
3	0.0346254	0.4171956	NO
4	-0.0679502	0.8172914	NO
5	-0.0082339	0.0984659	NO
6	0.0542844	0.6478277	NO
7	0.0187602	0.2228047	NO
8	-0.0865894	1.0284016	NO
9	-0.2633605	3.2185984	YES
10	-0.2124351	2.5538888	NO
11	-0.1000855	1.1773818	NO
12	0.0176943	0.2063810	NO
13	0.1171401	1.3704791	NO
14	0.0812145	0.9432418	NO
15	-0.0693859	0.8021307	NO
16	-0.1230346	1.4243813	NO
17	-0.1078472	1.2416091	NO
18	-0.0164907	0.1880486	NO
19	0.0725486	0.8261706	NO
20	0.0395759	0.4481015	NO
21	-0.0535792	0.6046759	NO
22	-0.1944492	2.2251596	NO
23	-0.1446191	1.6340687	NO
24	0.0055363	0.0616508	NO
25	0.1059437	1.1816220	NO

The simulation run through 225 samples is shown in Table 5, "Statistics for Selected Samples." Focusing on the statistics after sample number 150, all correlations of lags 1 through 9 was calculated using equation 6 to calculate the minimum sample size with autocorrelation. It is to be noticed that the negative autocorrelations tend to lower

TABLE 5

STATISTICS FOR SELECTED SAMPLE

AFTER SAMPLE NO.	MEAN	VARIANCE	MINIMUM WITHOUT AUTOCORRELATION	MAXIMUM LAG	MINIMUM WITH AUTOCORRELATION
25	1.07	0.61	419	0	419
50	1.22	0.93	493	0	493
75	1.31	0.84	385	0	385
100	1.25	0.89	449	1	724
125	1.29	0.94	445	22	189
150	1.37	1.18	491	9	934
175	1.42	1.15	451	2	1209
200	1.41	1.12	442	2	1108
225	1.38	1.08	447	2	1097

the result. After samples number 175, 200, 225, only positive autocorrelations are significant, resulting in larger minimum sample size calculations.

These data indicate that the minimum sample size using autocorrelation would be about twice the number found using only the observed mean and variance.

When comparing two similar, but significantly different systems, uniform criteria should be used not only in the modeled elements, but the length of the simulation runs. This is easier to attain using an automatic run-time rule such as described above.

BIBLIOGRAPHY

1. Rowe, D.W., and Hooper, J.W., "Data Systems Dynamic Simulation - A Total System for Data System Design Assessments and Trade Studies," Eleventh Annual Simulation Symposium, 1978, Tampa, FL. Tutorial for DSDS.

2. Shannon, Robert E., Systems Simulation: The Art and Science, Prentice-Hall, 1975, pp 180-197.

3. Andrews, Richard W., and Schriber, Thomas J., "Interactive Analysis of Output from GPSS-Based Simulations," pp 266-273, 1978 Winter Simulation Conference.

4. Bandemer, H., and Stoyan, D., "Optimal Sample Size with a priori Information about Certain Parameters," Biometriche Zeitschrift, Vol. 13, 1971, pp 376-382.

5. Crane, Michael A., and Lemoine, Austin J., "An Introduction to the Regenerative Method for Simulation Analysis," Control Analysis Corporation, October 1976.

6. Fishman, George S., "Estimating Sample Size in Computing Simulation Experiments," Management Science, Vol. 18, No. 1, September 1971, pp 21-38.

7. Fishman, George S., "Achieving Specified Accuracy in Simulation Output Analysis," University of North Carolina, December 1974.

8. Fishman, George S., "Grouping Observations in Digital Simulation," Management Science, Vol. 24, No. 5, January 1978, pp 510-521.

9. Geisler, M.A., "The Sizes of Simulation Samples Required to Compute Certain Inventory Characteristics with State Precision and Confidence," Memorandum RM-3242-RP, The RAND Corporation, Santa Monica, California, October 1962.

10. Kropp, Dean H., Carlson, Robert C., and Jucker, James V., "Use of Both Optimization and Simulation Models to Analyze Complex Systems," pp 194-201, 1978 Winter Simulation Conference.

11. Law, Averill M. and Carson, John S., "A Sequential Procedure for Determining the Length of a Steady-State Simulation," University of Wisconsin, April 1977. This is based on the method of batch means. It appears quite efficient, but requires a more skilled user.

12. Mize, Joe H., and Cox, J. Grady, Essentials of Simulation, 1968, "Estimation" Chapter 7, Confidence Intervals, Sample Size, Improving Efficiency.

13. Robinson, David W., "Determination of Run Length in Simulations of Stable Stochastic Systems," Control Analysis Corporation, February 1976.

14. Seila, Andrew F., "Quantile Estimation Methods for Discrete Event Simulations of Regenerative Systems," University of North Carolina, June 1976.

15. Starr, N., "The Performance of a Sequential Procedure for the Fixed Width Interval Estimation of the Mean," Ann. Math. Stat. Vol. 37, No. 1, February 1966, pp 36-50.