Developments in Business Simulation & Experiential Exercises, Volume 12, 1985

A SIMULATION MODEL TO CALCULATE SESSION TIME FOR RUNNING SIMULATION MODELS IN A SHARED RESOURCE ENVIRONMENT

F. Paul Fuhs, Virginia Commonwealth University

ABSTRACT

This paper demonstrates how a simulation model can be developed to estimate the average time to run a computer simulation with a large number of users in a computer equipment environment where some of the equipment is dedicated to individual users during their session and other equipment is shared by the users. We explain the rationale for building such a simulation, develop the analytical basis upon which the simulation is built and present the details of the simulation. In essence, we advocate the use of one simulation model to insure the successful use of another simulation model. In this paper we refer to the simulation used by the participants in the lab as the “object simulation” and the simulation used to estimate the average session time as the “estimator simulation.”

INTRODUCTION

SCENE: You assign a simulation exercise to a class of two hundred students to be completed within one week. The purpose of the simulation is to lay an experiential basis for the lectures scheduled for the follow- trig week. The computer lab has a number of CRT terminals and students share one or more common printers. Since the lab is used for many different classes, the number of hours per week of lab time must be allocated. You have been allocated the number of hours you requested for the simulation exercise. Halfway through the simulation week you realize that all of the participants will not be able to complete the simulation model on time. You have underestimated the amount of time it will take on the average for a student to complete the simulation.

You art faced with a catch-22 dilemma. Additional computer facilities cannot be acquired in time because of budgetary cycles. You can not easily request additional hours during that week because of resultant interference with other professors lab work in a tightly schedule lab environment. On the other hand, you cannot extend the completion date for the simulation. To do so would interfere with the lecture sequence for the entire course. You are left with canceling the simulation and suffering the resultant student bad will. Many of those students who did not get a chance to run the simulation will feel cheated. Many of those who completed the simulation will feel they wasted their time when you cancel it. Again, catch-22. For a simulation exercise to be successful it must be run to completion by all participants. To insure adequate allotted lab time for a simulation targeted for a large group of participants, one must first be able t calculate the average time to completion per participant, namely, the session time.

ESTIMATING SESSION TIME

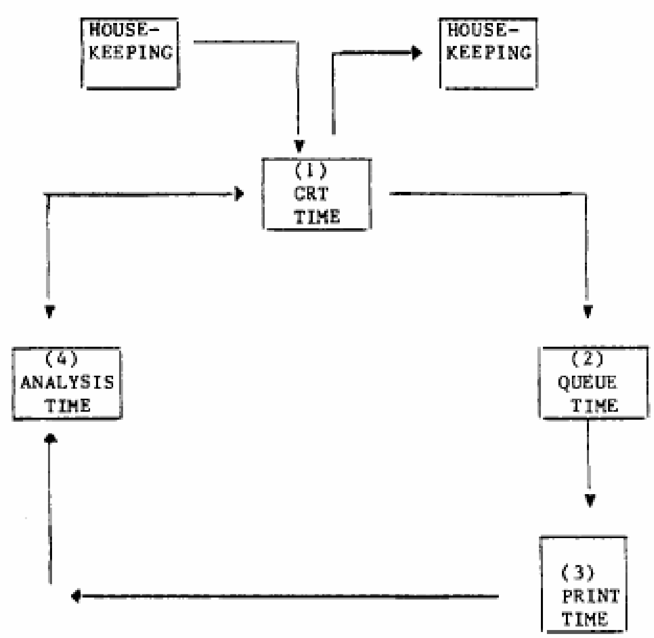
The problem with estimating average session time can be trivial or highly complex. It is trivial if the entire simulation can be experimentally completed by a statistically adequate number of users in an environment similar to that which will be used for the entire population of users. However, this situation is not always feasible.

The post-mortem analysis of such tragedies as mentioned above often points to an unexpected source of underestimation of session time; namely, the use of shared computer lab equipment by the participants. Such equipment produces bottlenecks that in turn cause extended queues and the resultant waiting times increase average simulation session time. Frequently a computer will execute the instructions of a simulation in a few seconds. However, the participants are found waiting for many minutes to use a common resource as a printer or plotter. Accumulated wait time does not present a problem when the shared resource is only needed at the end of the simulation, since such use can be viewed as “offline” with respect to the simulation. Accumulated wait time does increase session time when the participants hold (monopolize) other resources while waiting to use the common resource. Without empirical waiting time data it is not trivial to estimate such expected wait times and this calculation is the subject of this paper.

Figure 1 illustrates the steps in running many types of simulations. The shared resource in the figure is a printer, but any shared resource can be substituted without loss of generality. The figure shows that after logging onto the system and performing other housekeeping tasks a user runs part of the model, acquires printout, analyzes the printout, returns to the CRT. runs more of the simulation, and continues to cycle through steps 2 to 4 and back to step l until completion. A relatively significant portion of the time may be involved in waiting for printout. Note that the user The CRT is idle every time the user is waiting for printout. The users CRT cannot be used by another person until the current holder has completed the simulation. Therefore not only does session time include these waiting periods, but any estimate of CRT utilization must also include these wait times.

FIGURE 1

GENERAL MODEL OF SESSION TIME FOR ONE USER



From figure 1 we see that the session time is composed of housekeeping operations (H) at the beginning and end of the object simulation and a number of iterations (cycles). Each cycle includes time at the CRT; time waiting in a queue for hardcopy, (Q); time for hardcopy to be printed, (P); and time to analyze the hardcopy (A).

$$\text{Session Time} = H + \sum_{i=1}^{C} (CRT_i + Q_i + P_i + A_i) \qquad (1)$$

where C is the number of cycles to complete the simulation.

Initial housekeeping operations, (H), include logging onto the computer or booting up the operating system and starting the simulation. Termination housekeeping operations, that are part of (H), include operations needed to end a session at a CRT. The time required for housekeeping operations is not difficult to estimate.

The CRT time per cycle, (CRT_i), is defined as the time spent sitting at the CRT and, for computational convenience, includes the time to walk from the CRT to a printer station. CRT_i includes for the ith cycle the time to enter data and to review softcopy as well as CPU time. CRT_i can be estimated knowing average typing speed, the number of typed characters per cycle and the location of the printer station In relation to the CRTs. The number of characters typed is determined by analyzing the demanded input to the object simulation.

The print time per cycle, (p_i), can be calculated knowing the printer speed and the number of lines of printout per cycle. We include in Pi the time spent in tearing *off* the output from the printer and the time to walk to a table to begin analyzing the output. Some minor adjustment is needed in estimating due to the printing of blank lines. When micros are used, single directional versus bi-directional printing must be considered as well as single versus double pass printing.

The amount of time to analyze the hardcopy, A_1, is the time spent by the user in cycle i for determining the appropriate data content and form for the start of the next cycle at the CRT. Again, for computational convenience we also include here the time needed for the user to walk from the study area back to his/her CRT. A_1 can be determined In the following manner even without running either the object simulation or the estimator simulation. Select an appropriate sample of users. Give each user a representative printout for each cycle to analyze. For each cycle calculate the average amount of time for a user to decide upon input values. Add in the walk time multiplied by the number of cycles.

Q_i, the average wait time per cycle [or printout, is the amount of time extending from when the user is standing in front of the printer awaiting hardcopy to the time of the printing of the first line of output. Q_i, is the most complex value of session time to determine.

In the absence of empirical data, these are two approaches to the solution of closed loop systems like that shown in figure 1. The first is analytical modeling using Queuing Theory and the second is Computer Simulation. A purely analytical approach to this problem is difficult since the closed loop system includes two types of servers; multiple servers (users) and a single server (the shared resource). The problem becomes intractable when one realizes that the amount of service (printing) may be different for each cycle and the service for any user must be serial, that is, the amount of printing for cycle i must be completed before that of cycle i + i.

The Wait time determination problem is better suited to a computer simulation solution. We begin by discussing some of the known relationships within session time.

Session time and wait time are positively correlated with increased print load per session. A relevant ratio is the total amount of Print time/Session time. Initially, this ratio can be expressed as:

$$\text{Print Load} = P \;/\; \Big(\sum_{i=1}^{C} CRT_i + P_i + A_i \Big) \qquad (2)$$

This ratio is a relative measure of print load on the session. Later, when the total wait time per session, Q, has been determined through computer simulation, a more accurate measurement of the print load becomes:

$$\text{Print Load} = P \;/\; \Big(Q + \Big(\sum_{i=1}^{C} CRT_i + P_i + A_i \Big)\Big) \qquad (3)$$

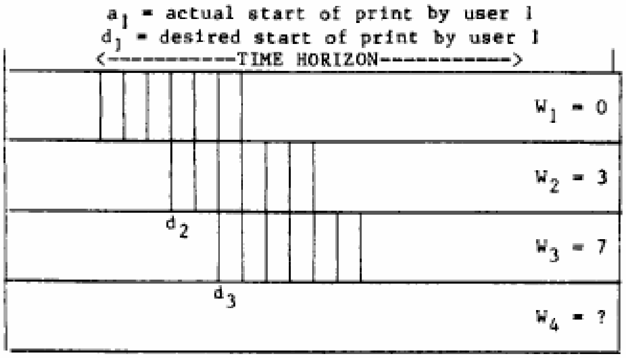
For a constant amount of printout and constant number of concurrent users, the session time is a function of accumulated wait time. The longer the accumulated wait time, the longer the session time. However, the wait time conversely is a function of the session time. The longer the session time the smaller the probability that two users will desire printout at the same time. Therefore, the average session time represents a steady state, an equilibrium in which increased wait time causes increased session time which in turn results in decreased wait time.

THE ESTIMATOR SIMULATION MODEL

Values for all of the variables comprising session time in equation I can be specified as input to the estimator simulation model with the exception of Q, the accumulated wait time. Average wait time per cycle, Qi, is the same for any cycle under the assumption that the users start the object simulation at random times. For any user the wait time during any cycle is a random variable. Figure 2 illustrates a single random occurrence of users simultaneously desiring hardcopy. Each row in the diagram represents a time horizon for one user. The grouped blocks within each row represent units of time. Each group of blocks represent the beginning, duration, and end of printing for a user for one cycle as if each user were not in contention with others for the shared resource, that is, if there were no queuing delays. Let the size of a block during cycle i be s_i The beginning of each block represents the time at which user u desires printout to begin. We designate this specific point in time d_u, D_u is synonymous with the point in time at which user u approaches the printer to see whether his/her output is printing. The point in time when printing actually occurs, au may be different from d_u In figure 2 when we consider users 1, 2, and 3, only d_1 - a_1. For user l the desired and the actual times are the same.

FIGURE 2

FOUR USERS SIMULTANEOUSLY DESIRING HARDCOPY



For clarity the rows (users) in figure 2 are shown sorted according to increasing du along the time horizon. This ordering also aids in the derivation of average wait time per user for any given random occurrence along the time horizon as will be shown later in this paper. In figure 2 the wait time for user I is zero, since that user is first. The wait time for user 2 is 3 time units; the overlap between user I and user 2. The wait time for user 3 is 7 time units; I rime unit from user I and 6 time units from user 2. The wait time for user 4 depends upon where that user is on the time horizon. A wait exists only if the desired point in time for printing is less than the sum of the starting time of user 3's printing plus the printing time of user 3 for cycle i, that is, s_3 In figure 2 a wait exists for user 4 if

$$d_4 < a_3 + s_{3,i} \qquad (4)$$

Equation 4 can be generalized.

$$d_u < a_{u-1} + s_{u-1,i} \qquad (5)$$

In figure 2, a_3 is calculated from the following recursive relationships.

$$a_1 = d_1 \qquad (6)$$

$$a_2 = a_1 + s_{1,i} \qquad (7)$$

$$a_3 = a_2 + s_{2,i} \qquad (8)$$

Equation 8 can be generalized to give the actual starting time to print for a given user.

$$a_u = a_{u-1} + s_{u-1,i} \qquad (9)$$

From the example in figure 2 we derive for one observation the wait time for any specific user in the queue.

$$W_u = \sum_{u=1}^{U-1} s_{u,i} - (d_u - a_1) \qquad (10)$$

Rearranging terms,

$$W_u = a_1 - d_u + \sum_{u=1}^{U-1} s_{u,i} \qquad (11)$$

Since wait time can not be negative,

$$W_u >= 0$$

For one observation the average wait time per user is derived.

$$W_n = (\sum_{u=1}^{U} W_u)/U \qquad (12)$$

The average wait time per user over a set of observations is where C is the number of cycles in the object simulation.

$$W = (\sum_{n=1}^{N} W_n)/N \qquad (13)$$

where N is the number of observations in the set.

The total wait time for a session becomes

$$Q = W * C \qquad (14)$$

The task is to generate adequate samples of random occurrences of queued printing, given all session time values except wait time and for each occurrence to calculate the average wait time for those users in the queue.

Structure of the Evaluator Simulation

Although this model could be more elegantly implemented using a true simulation language like SIMSCRIPT or GPSS, the structure of the model is developed here using ordinary matrices. In this way the model can be implemented easier by more people using more commonly known computer languages.

The simulation requires three matrices. Figure 3 illustrates these matrices. The first is the Event Matrix. Each row in this matrix represents one step of one cycle of session time. Column 2 contains data on the amount of time it takes to complete a step. Notice that values for the queue steps are left undefined and will remain undefined in this matrix for the duration of the simulation. The simulation determines individual values for the queue steps as each user cycles through the simulated events.

FIGURE 3 (Part 1)
STRUCTURE OF ESTIMATOR SIMULATION

EVENT MATRIX

EVENT	AMOUNT OF TIME FOR EVENT	
HOUSE-KEEPING	2	
CRT	1	<--- CYCLE 1 (start)
PRINT QUEUE	?	CYCLE 1
PRINT	2	CYCLE 1
ANALYZE	4	<--- CYCLE 1 (end)
CRT	2	
PRINT QUEUE	?	
PRINT	3	
HOUSE-KEEPING	2	

me second matrix is the Recorded Time matrix. Each row contains information on a single user. The row subscripts designate the users. This matrix holds the changes in events and the time of these changes as the simulation progresses through time.

Column 1 contains the event designation and matches some row subscript in the Event matrix.
Column 2 contains the point along the time horizon at which that event began.
Column 3 contains the point along the time horizon at which that event ended.
Column 4 contains accumulated elapsed time.
Column 5 contains the accumulated wait time.

FIGURE 3 (Part 2)
STRUCTURE OF ESTIMATOR SIMULATION

RECORDED TIME MATRIX
(At Start of Simulation)

EVENT #	START TIME OF EVENT	END TIME OF EVENT	TOTAL ELAPSED TIME	TOTAL WAIT TIME	
2	0	1			User 1
4	0	2			User 2
5	0	4			User 3
2	0	1			User 4

The third matrix is the Queue Matrix.

Column 1 contains the user identification number matching a row subscript in the Recorded Time matrix.
Column 2 contains the point in time at which the user will desire hardcopy, d_u.
Column 3 contains the calculated wait time for the user during the current cycle.
Column 4 contains the point in time at which printing actually occurs for the user during the current cycle, a_u.

QUEUE MATRIX
(When all users arrive at their next queue)

USER	d_u	WAIT TIME	a_u	
1	1	0	1	(processed members are rows 1 and 2 only. see Case 2)
4	1	2	3	
3	6	*	*	(* = not calculated)
2	8	*	*	

PROCESSING OF THE EVALUATOR SIMULATION

Any simulation model in which events march through time must consider simulation initialization, continuance, and the length of the simulation run.

Initialization: This part of the simulation must take into account transient states, the effect on the simulation results of the starting conditions of the simulation. Techniques have already been published for handling these problems [1]. Suffice it to say that in this simulation one needs to run the simulation a number of times with different randomly chosen starting steps for each user.

Continuance: Since we are assuming the running of the object simulation under constrained computer resources, in the absence of information to the contrary, the Estimator simulation will start another user session at the immediate termination of any user session. In this way an adequate sample of users through time can be generated.

Length of the Simulation: Emshoff succinctly answers the question of how long a simulation involving events in time should last when he states:

> “We say that a system has reached stable or steady-state conditions when successive observations of the system’s performance are statistically indistinguishable.” [2]

Procedures

The Estimator simulation begins with the selection of a random starting event for each CRT station. The starting events are selected from the Event matrix. Each selected event is entered into the Recorded Time matrix and each user will progress sequentially through the Event matrix as the simulation progresses. At the start of the simulation each user in the Recorded Time matrix is driven forward in time only until all users desire hardcopy. This is accomplished by repeatedly incrementing by I column I of the Recorded Time matrix, finding the amount of time to complete the event time from the Event matrix, and updating columns 3 and 4 of the Recorded Time matrix. Then information from the Recorded Time matrix is moved to columns I and 2 of the Queue matrix. The Queue matrix is sorted. Once sorted, the users in the Queue matrix are referenced by their row number in the Queue matrix, not their actual column 1 identification number. The true identity of the user is restored later when data from the Queue matrix is transferred back to the Recorded Time matrix.

The users in the Queue matrix are examined to detect which of the following cases have occurred.

Case I: All users except user I exist in a single queue. This case can be determined by applying equation 5 sequentially to each user in the Queue matrix, calculating for each user the wait time and resultant point in time at which printing will begin for that user. When case I occurs, the Recorded Time matrix is driven forward in time until each user again desires hardcopy.

Case 2: All users 2 through k are in one queue, irrespective of whether users k+2 through U are in one or more different queues, where k lies in the range of 2 to U-i. For example, users 2, 3, and 4 are in one queue and users 6 and 7 are in the next queue further along the time horizon. Note that here user 5 is not considered as being queued. This case identifies those users who are members of the first queue along the time horizon. Wait time for the members of this first queue must be calculated and the members of this queue must then be advanced in simulated time before any member of the second queue is considered. The reason for this is that it is possible for members of the first queue to exit that queue, cycle through the steps and again enter as members of the second queue. Again, equation 5 is applied to determine the kth user.

Case 3: User 2 is not queued for output as user I is printing. This situation presents another example of the possibility that a user may finish printing and cycle forward to become part of a queue further down the time horizon. After equation S Is applied to detect this case, the simulation advances only user 1 along the time horizon until user I again desires hardcopy.

For cases 1 and 2 equation 11 is applied to determine average wait time for the current cycle of each user. At the end of the simulation the average wait time in the queue, W, is calculated using equation 13. The equation ion 14 is used to calculate the total wait time per session, Q. From that, equation I gives the average session time. Optionally, equation 3 may then be used to calculate the Print Load ratio. Lastly, knowing the actual estimated session time per user, one ran calculate whether the given computer facilities will support an object simulation within a given time period for the large number of users.

FLEXIBILITY OF TILE ESTIMATOR SIMULATION

The estimator simulation can be used for many different object simulations through definition of the hvent Matrix. Events can be added, cycles added, and lapsed time per event can be updated as empirical data is updated. Lastly, the estimator model can be expanded to include a specific probability distribution for Ai, the time to analyze hardcopy at each cycle.

REFERENCES

[1] Shannon, Robert E., System Simulation: The Art and Science, (Prentice-Hall, 1975) 180ff.

[2] Emshoff, James R., Design and Use of Computer Simulation Models, (MacMillan Co., 1970), 190.