The Reduction of a Discrete Event Simulation to a Markov Chain

Andrew S. Noetzel
Applied Mathematics Department
Brookhaven National Laboratory
and
Elaine J. Weyuker
The Courant Institute of Mathematical Sciences,
New York University
and
Software Research Group, Sperry Univac Corp.

ABSTRACT

An event-driven, time-based stochastic simulation model may be modified in a straightforward manner to satisfy the axioms of a discrete-state, continuous-time Markov chain. The method requires casting the probability distribution of each random variable representing a time interval of the process into the form of a series of exponentially-distributed stages. An algorithm for this transformation has been developed. The technique is demonstrated via a simple example of a computer system simulation. The Markov chain representation of the simulation can be solved numerically, providing an independent verification of the logic of the simulation.

1. INTRODUCTION

Many problems can be approached by either simulation modelling or analytic modelling. Each method has its advantages. For simulation, there is precision, and the power to describe complex processes. An analytic model requires simplicity in its assumptions, and yields clarity and mathematical certainty of the demonstrated relationships.

For each approach there is a highly developed discipline. Modelers therefore tend to be of one type or the other, and modelling efforts that make use of both techniques are rare. Yet the advantages of the analytical techniques should be available to simulation modelers. In many cases, a simulation model is an elaboration and a refinement of a developed analytic model. It includes second and third order effects — parameters deemed not to bear significantly on the essential analytical results. At the least, the analytic model can be used to validate the more complex simulation model. If the results of the two modelling efforts are not comparable, an inquiry into the source of the disparity is well advised.

In the following sections, we explore the relationship between a time-based stochastic simulation, and a well-known analytic model — the discrete-state, continuous-parameter Markov process (or Markov chain). We offer a general description of each model, and discuss their essential differences. The, using a simple computer system model as an example, we show the structure of the simulator, and how it may be reduced to the Markov chain. The technique involves the decomposition of general probability distributions. When the simulation model is made to conform to the assumptions of the Markov chain its results can be compared with the numerical equilibrium solutions of the Markov chain. Thus, an independent verification of the logic of the simulator is obtained. In principle, as long as all of the simulation random variables that are not discrete are time intervals of the process, the reduction to the Markov chain conditions can be attained without reducing the power of the simulation. However, several practical constraints will limit the applicability of the technique. In practice, several different homomorphic mappings from the simulation model to the Markov process may be realized, allowing validation of the simulation in various modes of operation.

In the next sections, we outline the form of the event-driven simulation, then review the definition of the Markov chain. We then present the elementary computer system model, and discuss the transformation from simulation to Markov chain.

2. OUTLINE OF THE STOCHASTIC SIMULATION

We consider event-driven stochastic simulations of the following general form. Processing, or operation of the model, takes place at discrete points in simulated time, represented by the clock variable T. The events that are simulated at these discrete times are scheduled on the Future Events chain (FE) which is a list of (time, event) pairs, ordered by time. The main simulator loop removes the first item (t_i, e_i) on FE, updates T to the event time t_i, and then processes the model for the particular event type e_i. Generally, during the processing of event type e_i, it is possible to schedule its

*The submitted manuscript has been authored under contract EY-76-C-02-0016 with the U. S. Department of Energy.

CH1437-3/79/0391-0397$00.75 © 1979 IEEE
1979 Winter Simulation Conference

consequences, in the form of events e_j, e_k (including, possibly, e_i), to occur at a future time. The time of the future event e_j is generally determined by sampling the probability distribution of the random variable T_j. Then the entry $(T+T_j, e_j)$ is inserted in its proper location on FE. [Ref. (9), pg. 152 ff.]

We distinguish between the model state at clock time T, and the state of the simulation. The model state M is simply the vector of all the variables that describe the model under simulation, after those variables have been updated by the processing at clock time T. The simulation state S is M and FE. The fact that the process being modeled is approaching a particular future event cannot be determined by observing M, but can be seen in the more general simulation state S.

3. THE MARKOV CHAIN

The discrete state, continuous-parameter Markov process consists of a set of model states $\{M_1, M_2, \ldots\}$, and, for each pair of model states (M_i, M_j) a rate ρ_{ij} of transition from M_i to M_j. [Ref. (3,4).]

The transition rate ρ_{ij} is an instantaneous conditional probability rate: given that the model is in state M_i at time t, the probability that it will enter state M_j during the interval $(t, t+\Delta y)$, for small Δt, is $\rho_{ij}\Delta t$. Let the random variable T_i represent the time the process will remain in state M_i, given that it is in M_i. As a consequence of the assumption of constant conditional transition probability rates, T_i will have the exponential probability density function

$$f_i(t) = \rho_i e^{-\rho_i t} ,$$

where $\rho_i = \Sigma_j \rho_{ij}$, the summation being over all model states $M_j, j \neq i$. Furthermore, when the transition out of M_i is taken, the model enters state M_j with probability ρ_{ij}/ρ_i.

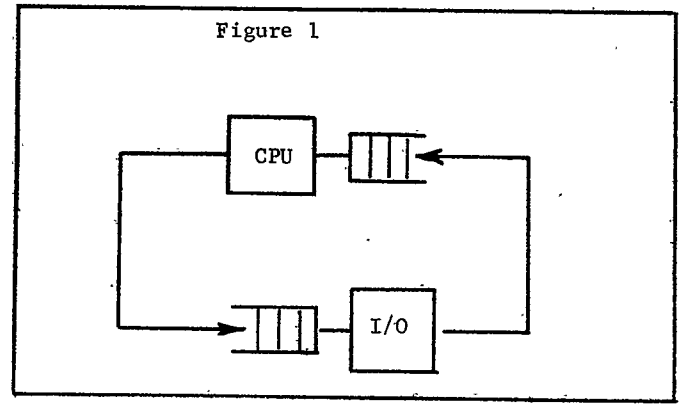
The Markov chain model includes no concept of a future events chain. At any time T, the model state M_i must embody all that needs to be known of the history of the process, so that the future of the model is determined (stochastically) only by the fixed transition probability rates ρ_{ij}.

We wish to consider the changes necessary to the simulation model in order that it satisfy the assumptions of the Markov chain. Rather than consider the matter in the abstract (which will be ambiguous, given the vagueness of the concept of the model state), we provide a simple example of a simulation, and refer to it throughout.

4. A SIMULATION EXAMPLE

As an elementary example, consider the simulation of the computer system model shown in Figure 1, consisting of a CPU and a single I/O device. We assume that the jobs in the system will be able to overlap CPU and I/O processings. Upon gaining access to the CPU, a job computes for an

Figure 1



interval T_c and then requires an I/O operation. Then while the I/O operation takes time T_i, CPU processing continues for time T_o. Initially, T_o is overlapped with T_i: when both the I/O and overlap period are complete, the computation period T_c may begin again. T_c, T_i and T_o are independent random variables. If only a single job is in the system, these timing variables fully determine the system operation, as shown in Figure 2.

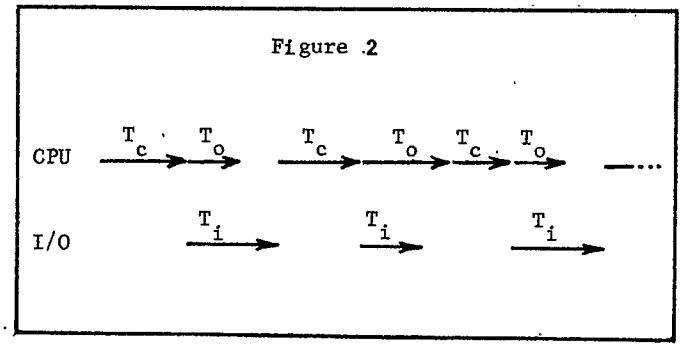
We propose to simulate the system with the following simple scheduling rules: both CPU and I/O device are allocated on a first-come, first-served basis, and are held until they can no longer be used. (Note that a job may hold the CPU indefinitely under this rule, if $T_o > T_i$ always.)

We assume further that T_c, T_i and T_o have probability density functions f_c, f_i, and f_o, which are identical for all jobs.

A natural data structure for the model state is a pair of ordered lists of job numbers — the CPU queue and the I/O queue. If either queue is empty, the corresponding device is idle; otherwise, the first job on the queue is using the device. This model state is used to describe the simulation process. Later, it will be seen that for the analysis of the simulation a more refined specification is necessary.

The simulation state consists of the model state and the future events chain. There are three event types, each identified with the end of a processing period. Event c, at the end of a compute period, indicates a request for an I/O operation, and the beginning of the overlap period.

Figure 2



Event o, at the end of the overlap period, indicates that further CPU processing must wait until completion of the I/O operation. Event i, at the end of the I/O period, indicates that a new compute period may begin, if, or as soon as, the overlap period is done.

A few steps of the simulation are shown in Figure 3, for the case when only two jobs are in the modelled system. Each event is subscripted by the number of the associated job. The model states are labled (CPUQ/IOQ). The leftmost job on each queue occupies the device. Thus, at $T=t_0$ $M=(12/-)$, indicating that job one is on the CPU, job two is waiting for the CPU, and the I/O device is idle.

Note that the events are placed on the future events chain only when their position in absolute simulation time can be conveniently calculated. For example, at time t_4 job two requests an I/O operation, which will require sampling the distributions for T_o and T_i. The time of the o_2 event is immediately computed as $t_4 + T_o$ and added to FE. But the I/O operation may not begin immediately because the I/O device is unavailable. Therefore, the computation of T_i and the addition of the i_2 event to FE, is postponed until the i_1 event.

A flow chart for the simulation system is shown in Figure 4. At the beginning of the main loop, the next event (t, e_{job}) is taken off FE. The clock time T is updated to the event time t, and the index j becomes the job number for the event. Then the subroutine determined by the event type is called to perform the specific queueing actions.

The State Diagram Representation of the Simulation

In order to express the simulation system in a form similar to that of the Markov chain, we may list the model states and show the transitions among them that correspond to the various possible events. Attempting this for the simulation example, we find that from model state alone (in the form chosen), it cannot be known which future events are actually possible. From state (12/-), for example, event c is possible only if job one is in a compute period, and the o event is possible only if it is in an overlap period. The distinction was not necessary in the model state chosen for the simulation, because the appropriate termination event was in every case posted on FE.

For consistancy with the Markov chain form, M must more fully express the instantaneous state of the model. It is therefore modified as follows. When job j is on the CPU, it will be designated in M as either j_o or j_c depending on whether it is in the compute or overlap period. (Because of the FCFS CPU scheduling, the designation is not necessary for the remaining jobs on the CPU queue.)

The model state transition diagram can now be drawn. An example, for the case of two jobs in the system, is shown in Figure 5.

The state diagram representation does not completely describe the simulation. It shows possible transitions, but does not show how the simulator determines which transition from a given model state M is actually taken. In the simulation, the transition events would have been placed on FE by sampling the distributions for T_c, T_o and T_i. It is essential to note that this sampling would generally have taken place in states occupied prior to the arrival at M, at the event corresponding to the beginning of each period. But without the future event chain, the sampling for the next event after each state M must take place at M, and this sampling must be done with conditional probability distributions.

For example, referring back to Figure 3, it is seen that at time t_2 when the model enters M=(2/1), the time of event c_2 can be computed by evaluating the random variable T_c. But if the time of event i_1 were to be calculated from M at t_2 one would have to use the conditional distribution for T_i given that time $t_e = t_2 - t_1$ has expired. As seen in Figure 6, the evaluation of the conditional random variable is in general a quite different computation from the original random variable. It is therefore seen to be possible to describe the

Figure 3

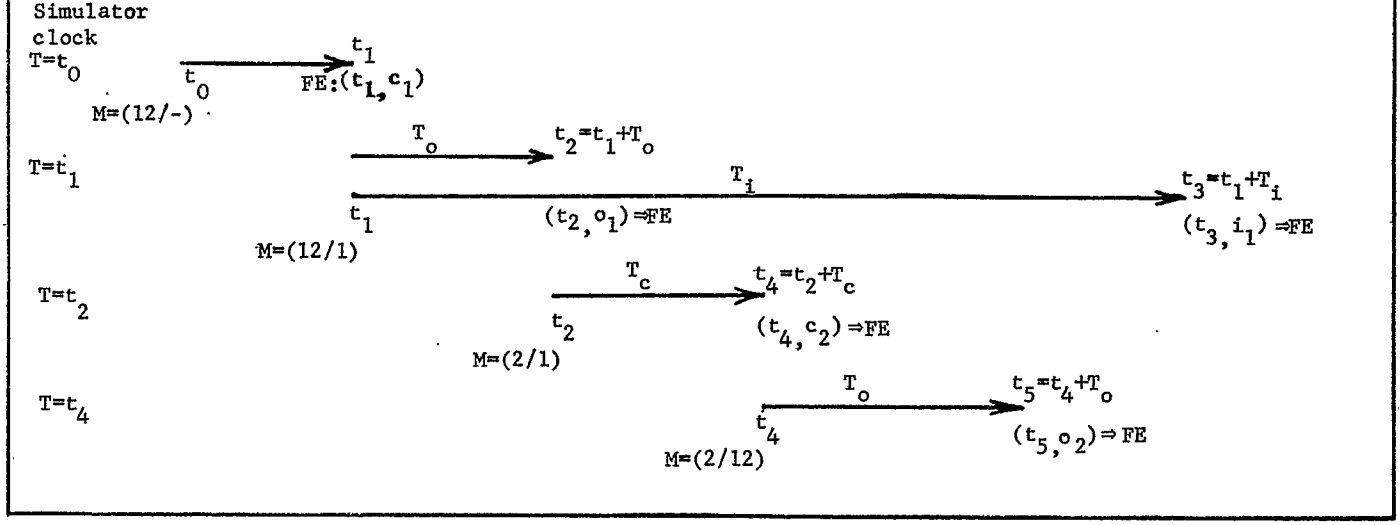


Figure 4

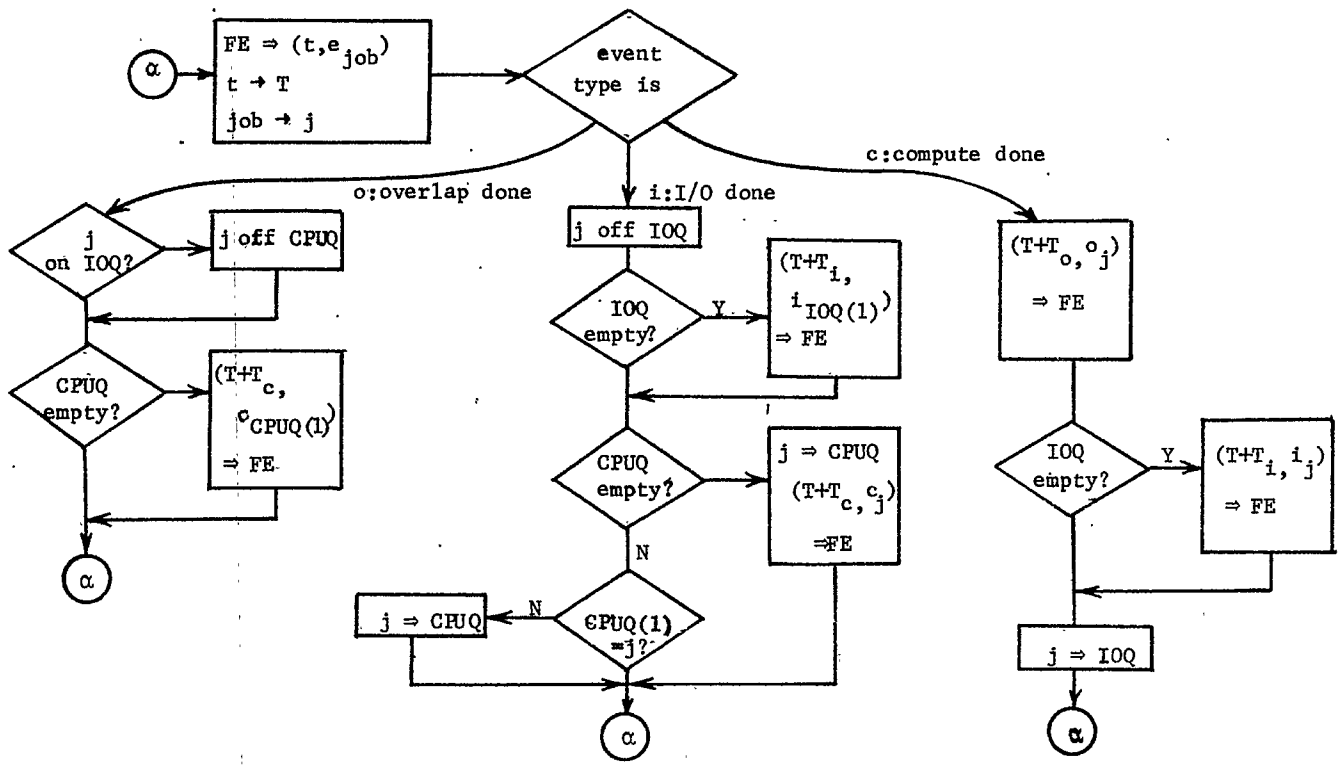


Figure 5

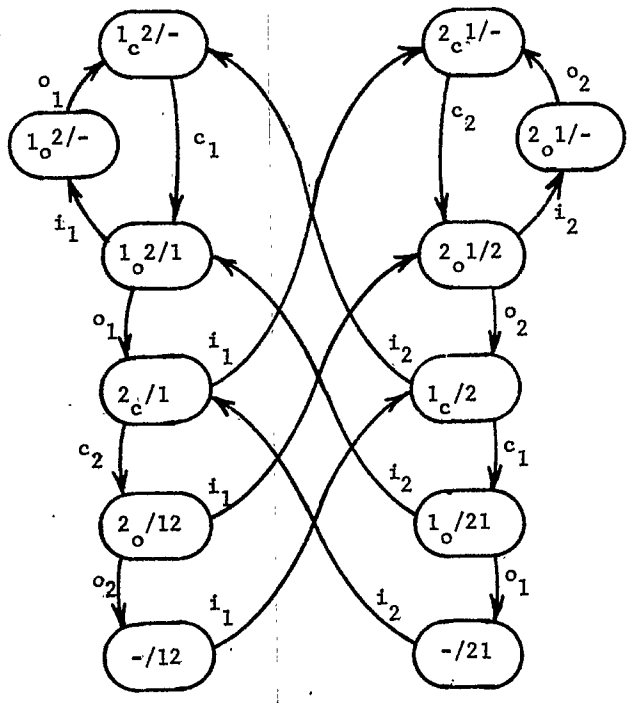
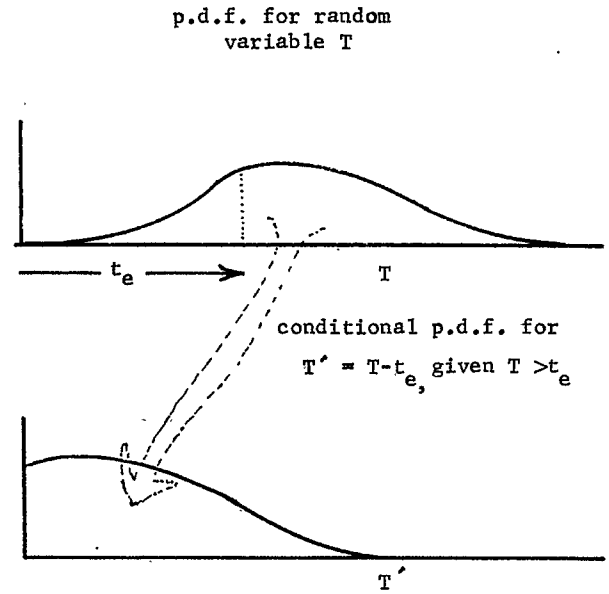


Figure 6



simulation without FE, but at the expense of both the additional computation for the conditional random variables, and the necessity to record the time elapsed since the occurrence of certain past events. In other words, the future events chain can be eliminated if it is replaced by a 'past events chain'.

Eliminating the Future Events Chain

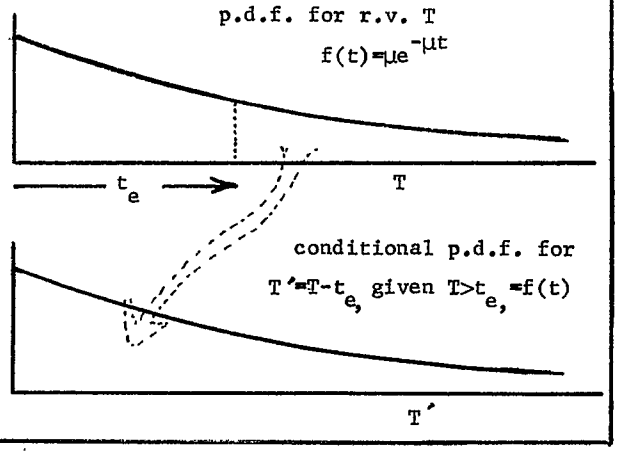
We now consider the special case that all of the times T_c, T_o, and T_i are exponentially distributed random variables. The exponential distribution has the well-known 'memoryless' property: if T is exponentially distributed, the conditional distribution of random variable $T' = T - t_e$ given that $T > t_e$ is exactly that of T. (Example shown in Figure 7). Accordingly, it will no longer be necessary to know how long an operation has been in progress in order to compute the random time of its termination. At each state in Figure 5, one may sample the exponential distribution for each possible future event, in order to determine which will occur first. The transition to the next state is made in processing that event. The times until the remaining future events are again determined by sampling their distributions — it is not necessary to remember the results of the previous sampling.

With the assumption of exponentially distributed intervals, the state diagram of Figure 5 becomes a Markov chain. The labels on the transitions must now be taken to signify not only the event on the transition, but the probability rate of the event. Thus if $M_j = (1_c 2/-)$ and $M_k = (1_o 2/1)$, then $\rho_{jk} = c$.

5. REDUCING GENERAL DISTRIBUTIONS TO EXPONENTIAL DISTRIBUTIONS

If the distributions of the random variables occurring in the simulation are not exponentially distributed it is still possible to achieve a Markov Chain representation of the process. The technique requires an expansion in partial fractions of the Laplace transform of the nonexponential probability distribution. In [(2)], Cox showed that a rational distribution (a function whose Laplace transform is a ratio of polynomials) can be expressed as series of exponentially distributed stages. A nonrational function can be approximated arbitrarily closely by a rational function.

Figure 7



Suppose a nonexponential probability density function f(t) has a Laplace transform

$$f^*(s) = \frac{P(s)}{Q(s)} \qquad (1)$$

where P and Q are polynomials in S, of degree m and n, respectively, with $m \leq n$. Then $f^*(s)$ can be written

$$f^*(s) = \sum_{i=1}^{n} a_1 \ldots a_{i-1}(1-a_i) \prod_{j=1}^{i} \frac{\mu_j}{s+\mu_j} \qquad (2)$$

where $a_1, \ldots, a_n$ are probabilities, and $\mu_1, \ldots, \mu_n$ are roots of Q(s).

The form $\frac{\mu_j}{s+\mu_j}$ is recognised as the Laplace transform of an $s+\mu_j$ exponential density function, and the product of such forms as the probability density function of the sum of exponentially distributed independent random variables. The general random variable is therefore seen to be the sum of up to n exponentially distributed stages, as shown in Figure 8. The conditional probability rate of leaving stage i is μ_i. Upon completing stage i, the next stage is entered with probability a_i and the entire nonexponentially distributed period is completed with probability $1-a_i$.

We show a simple example of the use of the partial fraction expansion to convert a nonexponential distribution into a series of exponential stages. Suppose T_c is assumed to have the hyperexponential probability density function

$$f_c(t) = 2(1+e^{-3t})e^{-3t} \qquad (3)$$

The Laplace transform of this function is

$$f_c^*(s) = \frac{4s+18}{s^2+9s+18}$$

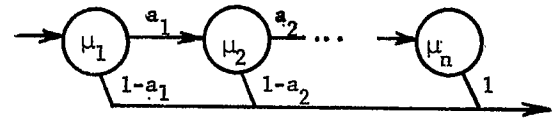
which can be expressed, through the partial fraction expansion (2), as

$$f_c^*(s) = \frac{2}{3}\frac{6}{6+s} + \frac{1}{3}\frac{6}{6+s}\frac{3}{3+s} \quad . \qquad (4)$$

The distribution takes the form of Figure 8, with two exponential stages, with probability rates $\mu_1=6$, $\mu_2=3$, and $a_1=1/3$. (The hyperexponential distribution has a more natural representation as two parallel exponential stages. However, it also serves as a simple example of the standard form.)

The state representation of the entire simulation, as shown in Figure 5, is a Markov chain representation of the simulation only if each of the random variables T_c, T_o and T_i has an exponential distribution. But if T_c has the two-stage distribution (3), the state diagram must be

Figure 8



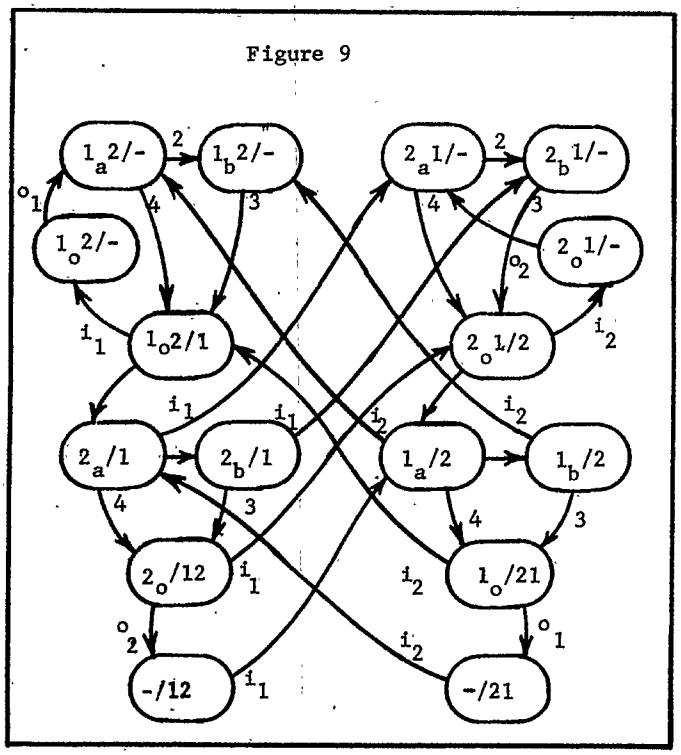
expanded to show states for each of the two stages of T_c wherever the random variable T_c is in effect. These are the states for which the job using the CPU is in the compute phase. We therefore designate T_a and T_b to be the two stages of T_c and replace each state for which j_c is on the CPU, with states in which j_a and j_b are on the CPU.

The resulting state diagram, shown in Figure 9, is a Markov chain representation for the simulation, in which T_c has the probability density function given by (3). State $(1_c 2/-)$ is replaced by $(1_a 2/-)$ and $(1_b 2/-)$. The probability rate of leaving $(1_a 2/-)$ is $\mu_1=6$. Upon completing stage T_a, the process continues to state $(1_b 2/-)$ with probability $a_1=1/3$, and enters state $(1_o 2/1)$ with probability 2/3.

We note that for this example, the conversion of one random variable from an exponential to a two-stage distribution has increased the number of states from twelve to sixteen.

The reduction of a general probability distribution to a series of exponential stages can be done without actually obtaining the Laplace transform of the distribution. In (1), Bux and Herzog present an algorithm for obtaining the parameters of the exponential stages for an approximation, to any desired degree of accuracy, of any general distribution. With this method, the first and second moments of the approximate distribution (the exponential stages) may be exactly that of the required distribution, and, by increasing the number of stages, the approximate distribution may be brough arbitrarily close to the general distribution, throughout the range of the random variable.

Figure 9



6. USE OF THE MARKOV CHAIN IN VALIDATION

The reduction of each general probability distribution to a series of exponential stages entails an increase in the number of model states. For each independent time random variable that is in effect throughout the simulation period, the number of model states is multiplied by the number of stages required. The increase in the number of model states limits the practical application of the technique.

However, the equilibrium balance equations of the Markov chain are a system of linear equations. The numerical solution of such systems has received much attention. It is possible to solve systems of linear equations when the number of variables is in the tens of thousands. The practical limit of the expansion of the state space is not quickly reached.

It is not proposed that the Markov chain reduction should replace the simulation, for the simulation technique will always be capable of greater complexity than the theoretical model. Rather, it should be possible to design the simulator to be capable of reduced mode configurations compatible with the assumptions of the Markov chain. A reduced mode configuration would assign exponential distributions to a subset of the random variables representing time intervals of the process (even if such distributions do not represent realistic cases) and would assign exponential-stage distributions, algoritmically constructed from the numerical values of the actual distributions, to other random variables. Other discrete parameters of the simulations must be selected to limit the total number of states. The logic of the simulation system may be validated by comparing the results of the simulation in the reduced mode with the numerical solutions of the equilibrium equations of the equivalent Markov chain. Although each such run will be only a partial test of the validity of the simulation, it will result from an independent computation, and hence will be free of the errors that other (integrated)validation efforts may carry over from the simulator itself.

REFERENCES

1. Bux, W. and Herzog, U. The Phase Concept: Approximation of Measured Data and Performance Analysis. Computer Performance, Chandy & Reiser (eds.) North-Holland, NY 1977.

2. Cox, D. R. A Use of Complex Probabilities in the Theory of Stochastic Processes. Proc. Cambridge Philosophical Society 51, 313-319 (1955).

3. Feller, W. An Introduction to Probability Theory and Its Applications. Wiley, NY 1968.

4. Kleinrock, L. Queueing Systems, Volume I: Theory. Wiley, NY 1975.

5. Kobayashi, H. Modeling and Analysis. Addison Wesley, NY 1978

6. Noetzel, A. S. and Herring, L. Experience with Trace-Driven Modeling. Proceedings of the Symposium on the Simulation of Computer Systems, N.B.S., Boulder Colorado, August, 1976.

7. Shannon, R. E. Systems Simulation, the Art and Science. Prentice-Hall, 1975.

8. Weyuker, E. J. Modifications of the Program Schema Model. J. Computer and Systems Science, 18,3 June 1979.

9. Zeigler, B. P. Theory of Modeling and Simulation. Wiley & Sons, NY 1976.