SIMULATION OF SYSTEMS

Professor. Dr. S.M. Aqil Burney*
Department of Computer Science
University of Karachi

What is simulation ?

Simulation is usually understood as the process of generating reality. A dictionary meaning of " to simulate" is

" Feigrn,. . ., pretent to be, act like, wear the guise of, mimic, . . ., imitate conditions of (situation etc.)with model, for convenience or training…,."
(Ripley, 1987 & Morgan, 1984)

In its prevailing sense simulation is concerned with the representation of systems by suitably defined models and observing the operation of such models under particular set of conditions. Thus technically speaking, the following definition of simulation can be adopted.

Simulation is the numerical technique for conductiong experiments on digital computer, which involves logical and mathematical relationships that interact to describe the behavior and the structure of a complex real world system over extended period of time.

What is a system ?

A system is usually considered as a set of inter–related factors, which are described as entities activities and have properties or attributes. Processes that cause system changes are called **activities**. The state of a system is a description of all entities, attributes and the activities at any time. A system may have very large number of states. The variables or activities of a system are classified as endogenous and exogenous variables. Endogenous variables occur within the system, and exogenous variables are found outside the system boundary i.e those variables in the environment which affect the system. A system with exogenous variables is considered as open. A system with strict endogenous activities is called a closed system.

Example 1

In simulating the operation of a railway goods yard, the train and railway siding might be the entities, with the number of wagons per train and the contents of each wagon as attributes of the train. An attribute of a railway siding could be whether or not it is occupied by a train. Activites might be the loading, unloading and the departure or arrival of train. The arrival of a train would be an exogenous activity; and the unloading of that train in the goods yards would be an example of endogenous activity, as this operation occurs within the system boundary.

Example 2
A simulation of the growth of a cell is simulation of a closed system.

Deterministic and stochastic Systems

A system is deterministic if the variables of the system are deterministic or completely predictable and no such variable or activity displays any degree of randomness.

A **stochastic** process is a collection of random variables $\{ X_t , t \in T \}$.

If T is countable set $\{ X_t , t \in T \}$ is called discrete time stochastic process. If T is an uncountable subset of the set of real numbers $\{ X_t , t \in T \}$ is called continuous time stochastic process.

Models: Mathematical representation of a theory may be termed as model. A stochastic simulation is the simulation of a model that involve random variables and the aims of the simulation are closely connected to those of modelling (Ripley, 1987). There are two principal reasons for modelling a situation:

1. To summarize data
2. To predict observations

Types of Simulation Models.

We may distinguish between simulation models which are static, dynamic, deterministic or stochastic. A static simulation is a representation of system at a particular time.

Example 3

(Birthday Problem) Suppose that in a room of N individuals each of the 365 days of the year is equally likely to be some one's birthday. Using the theory of probability, it can be shown that contrary to intution , only 23 individuals need to be present for the chance of more than 50–50 that atleast 2 of them will have the same birthday! This problem can be solved using simulation. Clearly, all that is needed is to select N random integers from the set {1,2,3, . . . , 365} and examine to find if there is a match. Repeating this experiment a large number of times the probability of at least one match in a gathering of N individuals can be computed.

The following C subroutines *probab* and *birthday* can be used for this purpose.

```
double probab(int num)
{
      int IL = 10000;
      double sum =0;
      int match=0;
      double prob=0;
      for(int i=0; i<IL ; i++)
      {
            match = birthday(num);

            if(match==1)
```

```
		{
			sum = sum + 1;
		}
	}
	prob = sum / IL;
	return prob;
}

int birthday(int num)
{
	int days[365];
	double number;
	for(int i=0; i<365; i++)
	{
		days[i]=0;
	}
	for(int j=0; j<num;j++)
	{
		number = rand()%365 ;
		if(days[(int)number] == 1)
		{
			return 1;
		}
		else
		{
			days[(int)number] = 1;
		}
	}

	return 0;

}
```

The theoretical and simulated results are given in the following

Table 1 : Birthday Problem Results

N	**Theoretical**	**Simulated**
5	**.027**	**.026**
10	**.117**	**.123**
20	**.411**	**.418**
22	**.476**	**.480**
23	**.507**	**.511**
25	**.569**	**.569**

[Theoretical results can be computed using the formula (365/365)(364/365)(363/365). . .((365–(N–1))/365)]

A dynamic simulation model is the representation of a system as it evolves over time. A simulation model is said to be deterministic if it contains no random variables. For a given set of inputs (for illustration see example 4). On the other hand, a simulation model is stochastic if it contains one or more random variables. The output data for a stochastic model are themselves random and thus only estimate the true characteristic of the model. (see example 5).

Discrete – event simulation concerns modelling of a system as it evolves over time such that state variables change only at a countable number of point.

Continuous simulation concerns the modelling over time of a system by a representation in which state variables change involve one or more differential equations that give relationship for the rate of change of state variables with respect to time. If the differential equations are simple, they can be solved analytically. If analytical solution is not possible then numerical methods such as Runge–Kutta method can be used to solve the differential equation.

Before discussing the example to illustrate above mentioned simulation models, it is important to discuss how state changes are modelled over time? (or how time–flow might be handled while doing simulation?)

Time Handling

Some of the commonly used time handling techniques are **Time slicing** and **Next event** techniques. Time slicing involves updating and examining. The model is updated at time $t + \Delta t$ for changes occuring in the interval $(t, t+\Delta t)$. Decision must be taken about the length of time before simulation is carried out. For example, the activity levels at a seaport may need a time slice of a day or so whereas for a busy international airport the time slice may be a minute or less. Clearly if the time slice is too large then behaviour of the model will be coarse than real system.

Sometimes, an experimenter can not afford to waste time by adopting the time slicing techniques as a system may include slack periods of varying length. Thus it is preferable to use a variable time increment. In such a situation, the model is only examined and updated when it is known that a state change is due and such a technique is called **Next event** technique. For example, with software simulation, one can not usually afford to waste processing time by considering nonevent epochs and simulation clock is advanced in nonequal jumps from one event occuring epoch to the next such epoch.

Examples on Simulation Models.

Example 4

(Deterministic Simulation) (Pidd 1983) Consider a factory which manufactures only one product. Raw material is bought from external supplier and stored until required. Finished items are held in a warehouse. The operation of factory and its warehouse can be modelled as a set of equations as folllows. Let us define at time t :

R(t) = Raw material stored (units)
F(t) = finished goods stock (units)

B(t) = order backlog (units)

T(t) = target stock level for finished goods (units)

All variables defined above give quantities at the start of week t.

X(t, t+1)= weekly orders received from customers

M(t, t+1)= raw material supplied per week.

P(t, t+1)= production per week.

D(t, t+1)= amount dispatched to customers per week.

All variables defined above give quantities over week t to t+1
(i.e over the week t). The operation of the factory and its warehouse can be expressed as a set of equations given as follows:

Backlog and Stock Position

(1) B(t+1) = B(t)+X (t,t+1)–D(t,t+1)

(2) $T(t+1) = \frac{(m+1)}{m}$ (X(t,t+1)+X(t–1,t)+…+X (t–m+1,t–m+2))

(assuming that the company wishes to maintain m (suppose m=5)) weeks stock of finish items and hence the target level is m times the average of the last m–1 weeks)

(3) R (t+1)=R(t)+M(t,t+1)–P(t,t+1)

(4) F (t+1)=F (t) +P(t,t+1)–D(t,t+1)

Rates

(5) D (t,t+1)	=	B(t) if B(t)<F(t) F (t) otherwise
(6) M (t,t+1)	=	P(t–1,t)
(7) P (t,t+1)	=	T(t) – F(t) + D(t,t+1)
	=	R(t) if result exceeds R(t)
	=	0 if the result is negative

Given the initial values for the variables, it is possible to simulate this system to study how the system will respond to the order rate. Suppose that all is calm, and the factory has operated as follows for the last five weeks.

Target warehouse stock	=	250
Finished goods stock	=	250

Raw material stock	=	150
Production rate	=	50/week
Material supply rate	=	50/week
Order rate	=	50/week
Order backlog	=	50

Suppose the behaviour continues for the first week of the simulation but that during next week orders double due to the sales promotion. During the third week orders drops to zero as all demand returns of the previous week was satisfied. For the fourth week and the succeeding weeks, demand returns to an order rate of 50/week. What happens elsewhere in the system? A deterministic simulation will provide the answer to the above mentioned question. For this compute the following.

i) The values of the equations (1)–(4) at the start of week t.
ii) The values of the equations (5)–(7) i.e. the new values of the rates during the follwing week.
iii) Move simulation time to the start of the next week.

Next simulation should be presented in tabular form and plot production and demand rate to examine the performance of the system. Table 1 gives the results of the above mentioned simulation experiment.

From figure 1 it is clear that all is not well and there must be a better way of operating such a system and simulation permits us to check whether this the case. If a new policy { i.e. M(t, t+1) = 50 * (T(t)/F(t) and P (t,t+1) = R(t)/3) } is adopted then a revised dynamic response is obtained and the factory can run in a better way as in such situation production rate can satisfy weekly orders received (see figure 2).

Example 5

(Stochastic Simulation)

A multi –user computer system includes two disk units which, being mechanical, are prone to failure. If a disk unit fails in service, user lose their files (and their tempers) which need to be resorted. Restoration is achieved by copying on the disk back–up copies of the files held on magnetic tapes. This

Table 1 (a)

W	X(t,t+1)	B(t)	T(t)	F(t)	R(t)
1	50	50	250	250	150
2	100	100	313	250	150
3	0	0	250	313	33
4	50	50	250	313	200
5	50	50	250	263	200
6	50	50	188	250	163
7	50	50	250	200	200
8	50	50	250	250	100
9	50	50	250	250	150
10	50	50	250	250	150

Table 1(b)

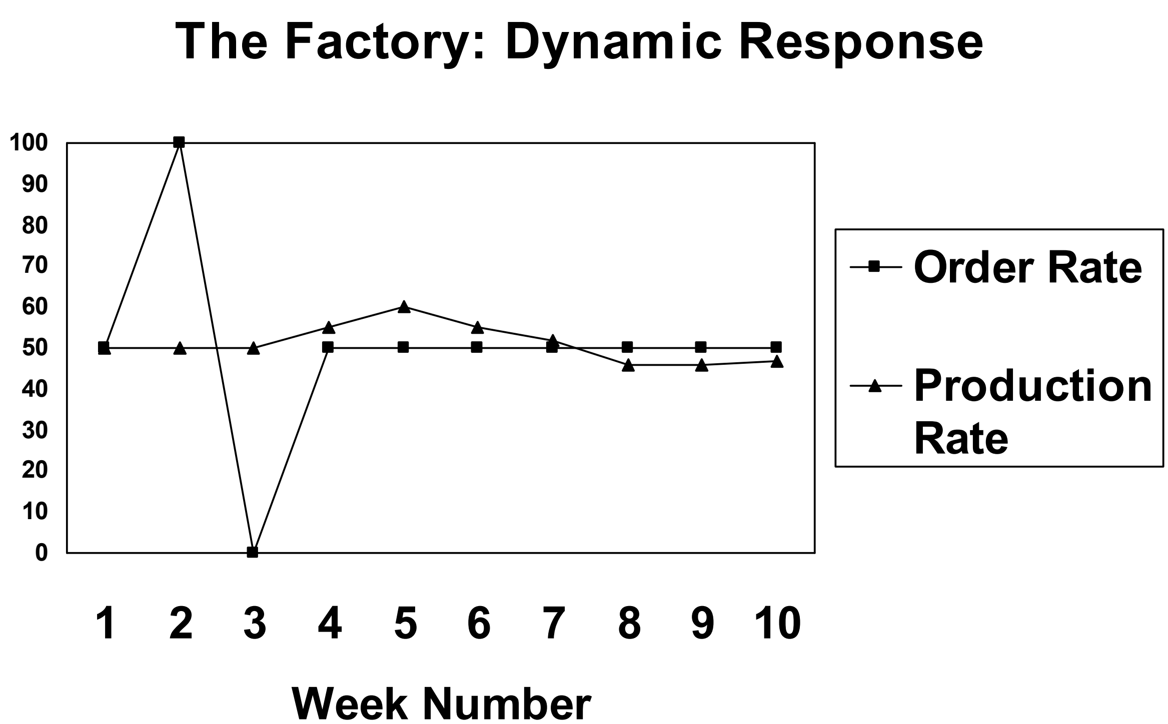
W	M(t,t+1)	P(t,t+1)	D(t,t+1)
1	50	50	50
2	50	50	50
3	50	163	100
4	163	0	0
5	0	0	50
6	0	38	50
7	38	0	50
8	0	100	50
9	100	50	50
10	50	50	50

Table 1(a) & Table 1(b) The Factory: The Time Deterministic Simulation.

Days since repair or maintenance	Probability of failure
1	0.06
2	0.14
3	0.22
4	0.28
5	0.22
6	0.08
> 6	0.00

Table 2: Probability of disk failure

restoration is inconvenient and so a new operating policy is being considered. At the moment, the disk units are repaired and restored as and when fail. The proposal is to introduce a joint repair system. Table 2.1 below shows the probability of a disk unit failing in the days following its last repair. That is, 6% of the units are expected to fail 1 day after repair or maintenance, 14% after 2 days, etc.



Under the current policy, it costs Rs. 50/disk to repair and restore a failed unit. The joint repair system would operate as follows. When either unit fails, the failed unit is repaired and restored at a cost of Rs. 50 per unit. If operational, the other unit will be cleaned at a cost of Rs. 25. Cleaning a disk places it in a state equivalent to having been just repaired and restored. Is the new joint repair system cost effective?

This question can be answered by a stochastic simulation which involves generation of realization of pattern of failure from the above mentioned failure distribution.

Some random number (from random table, see Morgan, 1984, pp. 250) as shown in table 2, the values range from 00 to 99, and any number in that range has an equal probability of appearing at any position in the table. If these random numbers are divided by 100, so that their range is 0.00 to 0.99. This corresponds to 3 days (see table 2).

36 15 88 60 36		02 68 35 83 38	
48 61 82 32 99		51 61 85 16 06	
29 76 40 88 52		92 61 75 39 90	
24 13 13 02 29		28 75 03 74 24	
84 84 07 72 77		57 05 08 97 00	
	63 02 90 50 69		01 68 17 31 51
	61 09 09 72 97		20 92 93 60 67
	88 29 70 63 55		00 09 91 99 04
	19 59 89 42 93		39 78 42 49 26
	57 07 75 25 85		80 47 32 79 76
95 03 88 22 41		51 20 33 37 58	
86 63 51 58 56		49 53 05 23 72	
75 05 63 80 09		43 31 24 82 55	
19 05 17 37 81		86 66 32 81 47	
50 95 02 88 59		97 52 77 41 21	

Figure 3. Some random numbers

Life associated with the random number 0.36. In general, each life may be associated with a range of random number as shown in **Table 3.**

Life (days)	Associated random numbers
1	0.00–0.05
2	0.06–0.19
3	0.20–0.41
4	0.42–0.69
5	0.70–0.91
6	0.92–0.99

Table 3. Random number linked to disk life

To carry out the simulation, a separate stream of numbers is used to represent the sequence of successive lives of the two units as in Table 4. These are then used in

Table 2.5 to compare the two policies. Thus, after 40 days operation, the results are as follows:

(a) **POLICY 1:** Separate repair

unit A : 11 repairs and restores

unit B : 11 repairs and restores

i.e. 22 repairs and restores at Rs. 50 each = Rs. 1100.00

(b) **POLICY 2** : Joint repair

Carried out on 11 occasions. Of these, 3 involved both units in a repair and restored ; 8 involved a single repair and restore with a clean up of the disk.

Table 4

	Unit A		**Unit B**	
	Random Number	Life	Random Number	Life
1.	0.36	3	0.02	1
2.	0.15	2	0.68	4
3.	0.88	5	0.35	3
4.	0.60	4	0.83	5
5.	0.36	3	0.38	4
6.	0.48	4	0.51	4
7.	0.61	4	0.61	4
8.	0.82	5	0.85	5
9.	0.32	3	0.16	2
10.	0.99	6	0.06	2
11.	0.29	3	0.92	6

(Next event simulation of two units)

Thus on the basis of a single simulation, the new policy costs Rs. 850.0 over a period of 40 days with the following details.

13 repairs and restores @ Rs. 50.0 = Rs. 650.00
8 clean ups @ Rs. 25.00 = Rs. 200.00

Therefore, a separate repair policy is not cost effective. If a set of random numbers are used the result will be different for both policies and the new policy might not be cheaper.

Table 5
(a)

Policy I : **Separate repair** **Cumulative lives**	
Unit A	Unit B
1. 3	1
2. 5	5
3. 10	8
4. 14	13
5. 17	17
6. 21	21
7. 25	25
8. 30	30
9. 33	32
10.39	34
11.42	40

(b)

Policy II : **Joint repair**	
Minimum Joint Life	Cumulative Joint Life
1	1
2	3
3	6
4	10
3	13
4	17
4	21
5	26
2	28
2	30
3	33

Table 5 replacement policies compared

Random numer generation.

The type of samples usually required from random number generators, are those with uniform distributions (simplest probability distribution is uniform or rectangular distribution in that the probability is uniform over the whole range of the variate). There are three type of methods for generating random numbers. They are:

i) manual methods (e.g rolling dice, spinning roulette wheels, etc).
ii) mechanical methods (i.e mechanise the manual methods)
iii) electronic methods (They work on the basis of converting by some suitable means, the noise created in electronic circuits into sequence random numbers, for detail see tocher (1963).
iv) Computer methods.(most common methods of obtaining random numbers is pseudo–random number generator). A sequence of pseudo–random numbers $\{U_i\}$is a deterministic sequence of numbers in [0,1] having the same relevant statistical properties as a sequence of random number, thus they are pseudo–random number.

The suitability of the various types of random number generator may be checked against the following six criteria which are desirable properties of random number sequences and their generators:

1) the random number should be uniformly distributed
2) each random number should be statistically independent of all other random number in the sequence
3) sequence should be reproducible
4) sequence should not repeat over a given period
5) the random number generator should be computationally fast
6) the software describing the generator should be as concise as possible.

Pseudorandom number generators generally satifsy these properties. Computer software pseudo–random number generators usually based on congruence or residue method. Lehmer (1951) worked on such generator and have the following general form:

$$X_i = aX_{i-1} + c(\text{mod } m) \quad \text{for } i = 0, \ldots .n$$

Where $\{X_i\}$ is the sequence of pseudo–random numbers, a,c,m are constants, X_o is seed or initially specified value for the sequence (mod m) means divide $aX_i + c$ by m use remainder as the result.

Example 6

For a = 3, c = 0, m=5, $X_o = 2$. The generator has the form

$$X_i = 3X_{i-1}(\text{mod } 5)$$

$$X_i = 6 \quad (\text{mod } 5) = 1$$

$$X_2 = 3 \quad (\text{mod } 5) = 3$$

$$X_3 = 9 \quad (\text{mod } 5) = 4$$

$$X_4 = 12 \quad (\text{mod } 5) = 2$$

$$X_5 = 6 \quad (\text{mod } 5) = 1$$

Notice two points from this set of pseudo–random numbers:

i) The generator produces values in the range 0, . . . , m–1. Therefore, to obtain uniformly distributed pseudo–random numbers in the interval [0,1], the values produced are modified s.t.

$$U_i = X_i / m$$

The sequence $\{U_i\}$ in the interval [0,1] are called the sequence of random numbers.

ii) value of X_i can not exceed m.

For the existence of random numbers see Neveu (1965) and for survey Knuth (1971) can be referred. For the computer generation of random numbers work of Ripley (1983, 1987) can be considered.

There are three types of pseudo–random number generators based on congruence generator, briefly described as follows.

The multiplicative congruence method.

This method is based upon the expression

$$X_{n+1} = a X_n \quad (\text{mod } m)$$

which is a special case of the general congruence expression (i.e. c=0). To obtain the maximum span for the generated sequences, one must carefully choose appropriate values of a and X_o. When using a binary computer, it may be shown that the maximum span is given by 2^{r-2} ,where r is the number of bits used to describe the modulus m. This span is obtained for the particular values of the constant a. The constant a should be of order $2^{r/2}$ (r is computer word length) to reduce the serial correlation (Chatifieldh between the pseudo–number).

The starting value X_o may be any positive odd number. This particular method is probably the most popular of the congruence methods, because of ease in programming and operating speed.

The additive congruence method.

This method is based upon the general recursive expression.

$$X_{n+1} = aX_n + c \ (\text{mod } m)$$

To achieve the full period or span of m pseudo–random numbers, the following are necessary conditions:

(a) the initial value X_o may be any positive number
(b) the multiplier constant a should be of the form
 $a = 2^{P+1}$ where $p \geq 2$
(c) the constant c must be an odd number.

Although this generator is capable of producing the span of pseudo–random numbers, tests indicate that correlation effects may influence the sequence. Thorough testing of particular generators is necessary to ascertain whether or not the statistical properties are suitable

Pseduo–random number tests.

It is important to remember that the sequence produced by pseudo–random number generators are only acceptable provided that they confirm to suitable tests for radomnees.

In some case, the pseudo–radnom number sequence may be required to possess particular properties. The existence of these properties may be adequately examined by the tests. Therefore, in such circumstances, one may have to design one or more suitable tests.

The generation of random variates

In simulation study of stochastic systems an obvious necessity is a suitable method of generating random samples of required random variables. The random variables may be discrete or continuous. We wish to sample a variate X from a distribution given by $P_r = P(x = r)$, r = 1,2,…. And $P_r = P(X \leq r)$. Then inversion technique can be used.

For a discrete distribution $F^{-}(u) = \min \{x \mid F(x) \geq u\} = i$ where $P_{i-1} < u \leq P_i$ so inversion amounts to searching a table of (P_i) for a suitable index i. Thus we have the following algorithm.

Algorithm

1. Generate U ~ U (0,1). Let i=1
2. While $P_i \leq U$ Do $i = i + 1$
3. Return x = i

For continuous random variable X with cumulative distribution function F(x) is replaced by U, where U is a uniform random variable over [0,1]. Hence any value of U may be transformed into a value of x, ; algebrically

$$\text{If } U = \int_{\alpha}^{x} f(t)\,dt \qquad \text{then } x = G(u)$$

where G(u) is known as the inverse cumulative function.

Example 6.

(Exponential Distribution)

$$f(x) = \left(\frac{1}{\lambda}\right)\exp\left(-\frac{x}{\lambda}\right) \qquad \text{for } 0 \leq x \leq \infty$$
$$= 0 \qquad \text{elsewhere.}$$

$$F(x) = 1 - \exp\left(-\frac{x}{\lambda}\right), \quad u = 1 - \exp\left(-\frac{x}{\lambda}\right)$$
$$x = -\lambda \log_e(1-u)$$

For continuous distrubution it is not always possible to integrate f(*x*). In such circumstances many other methods are available for detail see Ripley (1987) Morgan

(1984), Fishman (1973, 1978) and Atkinson and Pearse (1976). For normal distrubution Box–Muller (1958) method, Marsaglia's polar method and ratio–of–uniforms (Ripley 1983) are available.

Acknowledgement

The author is thankful to University of Karachi for providing the financial assistance to prepare this article.

Books on Modelling & Simulation

1. Sinelair, J.B. (2004). Simulation of Computer System and Computer Networks: A Process-Oriented Approach.
2. Banks, J., John S. Carson, Barry L. Nelson, David M. Nicol(2005). Discrete Event System Simulation. Pearson Education.
3. Michael pidd, Computer Simulation in Management Science. John Wiley & Sons.
4. B.D. Ripley (1987) Stochastic Simulation. John Wiley.
5. Morgan, Byron, J.T.(1984). Elements of Simulation. Chapman and Hall London.
6. Ross, Sheldon, M. Ross (2001). Simulation, 3rd ed. Academic Press, San diego.
7. Burney, S.M. Aqil (2003) Introduction to Stochastic Processes and their Applications in Insurance and Computer Science, University of Sindh, Jamshoro, Pakistan.
8. Knuth, D.E. (1999). The art of Computer Programming Vol. 2, Semi Numerical Algorithms. Addition-Wesley, Reading, M.A. 1999.
9. Oakshölt, Les (1997). Business Modeling and Simulation, Pitman Publishing, London.