Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

SIMULATION OF MANUFACTURING SYSTEMS

Averill M. Law
Michael G. McComas
Simulation Modeling and Analysis Company
University of Arizona
Tucson, Arizona

ABSTRACT

This paper discusses how simulation is used to design and analyze manufacturing or warehousing systems. Topics discussed include: manufacturing issues investigated by simulation, techniques for building valid and credible models, manufacturing simulation software, statistical considerations, and simulation pitfalls. A case study is included.

1. INTRODUCTION

In this paper we present an overview of the use of simulation in the design and analysis of manufacturing or warehousing systems. A more detailed discussion of simulation, in general, may be found in Law and Kelton or Banks and Carson (see "REFERENCES").

There has been a dramatic increase in the use of simulation in manufacturing during the past few years. Increased foreign competition in many industries has resulted in a greater emphasis on automation to improve productivity and quality and also to reduce cost. However, since automated systems are often considerably more complex, they typically can only be analyzed by a powerful tool like simulation. Reduced computing costs and improvements in simulation languages (which have reduced model development time) have also led to the increased use of simulation. Finally, the availability of graphical animation has resulted in a greater understanding and use of simulation by engineering managers.

1.1. Why (Simulation) Models are Necessary

It is often of interest to study a dynamic real-world system to learn something about its behavior. However, it is generally necessary to use a model to study the performance of the system, since experimentation with the system itself is disruptive, not cost-effective, or simply impossible (e.g., the manufacturing facility has not yet been build). For example, consider a manufacturing firm that is contemplating building a large extension onto one of its plants but it is not sure whether the potential gain in productivity would justify the construction cost. It certainly would not be cost-effective to build the extension and then remove it later if it does not work out. However, an appropriate model could shed some light on this question by allowing the operation of the plant to be studied as it currently exists and as it would be if the plant were expanded.

If the relationships which compose the model are simple enough, it may be possible to use mathematical methods (such as algebra, calculus, or probability theory) to obtain the exact answers to the questions of interest; this is called an analytic solution. As an example of an analytic solution, consider the single machine tool shown in Figure 1. Jobs (or work pieces) arrive to the machine tool and request processing. If the machine is idle when a job arrives, then processing of the job begins immediately. Otherwise, the job joins the end of a queue. When the machine finishes processing one job, it then begins processing the first job in the queue (if any).

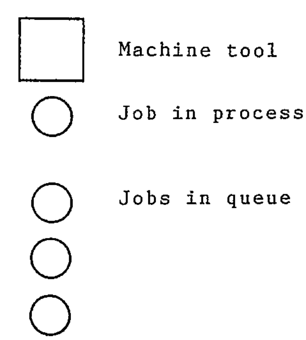

Figure 1: Single Machine Tool System

Let a be the rate at which jobs arrive to the machine (i.e., jobs per unit time) and let p be the rate at which the machine can process jobs. If we assume that a is less than p, then it can be shown under appropriate assumptions that the long-run average time a job spends in the system (in queue plus being processed), w, is given by

$$w = 1/(p-a).$$

Thus, this formula can be used to determine easily the average time in system for various legitimate values of a and p.

Unfortunately, however, most manufacturing systems are too complex to allow realistic models to be evaluated mathematically, and these models must be studied by means of simulation. In a simulation we use a computer to evaluate a model numerically over a time period of interest, and data are gathered to estimate the desired true characteristics (e.g., throughput) of the model.

We will concentrate our attention on a particular type of simulation, which is called discrete-event simulation. This type of simulation is generally stochastic in nature, which means that random samples from probability distributions are used to drive the model through time (see Section 5.1).

1.2. Intuitive Explanation of Simulation

In this section we intuitively explain the nature of simulation by showing how a simulation model of the single machine tool system (see Section 1.1) would evolve over time. Suppose that our goal in studying this system is to determine the average amount of time that a job spends in the system.

We begin with some definitions. An event is an instantaneous occurrence which changes the state of a system. The arrival of a new job and the end of processing of a job by the machine are each events for the machine tool system. The event list is an array giving the time of the next occurrence for each type of event. For the machine tool system, the first entry in the event list would give the time that the next job will arrive and the second (last) entry would give the time that the current job (if any) will complete processing and depart. The simulation clock is a variable which gives the current value of simulated time. (If a simulation is two hours into a desired run length of eight hours, then the value of the simula-clock would be 2, if the time unit is hours.) The system state are variables which describe the state of the system at a particular point in time. For our example, the state variables are the status of the machine (i.e., idle or busy), the number of jobs waiting in the queue, and the time of arrival for each job in the system. On the other hand, statistical counters are variables which contain information necessary to estimate the desired measure(s) of performance. The statistical counters necessary to estimate average job time in system are the number of jobs that have completed processing, and also the total time in system for all completed jobs. At the end of the simulation, we estimate the average time in system by dividing the total time in system for all completed jobs by the number of completed jobs.

We now define some notation to facilitate our explanation of simulation. Let

t_i = time of arrival of the ith job to arrive to the system ($t_0 = 0$)

$A_i = t_i - t_{i-1}$ = interarrival time between the (i-1)st and ith arrivals of jobs

P_i = time that the machine actually spends processing the ith job to arrive (exclusive of a job's delay in queue, if any)

W_i = time in system (wait) of the ith job to arrive

In the actual machine tool system, the A_i's and P_i's may be random variables (i.e., their values will not be known with certainty). However, for expository convenience we assume that the interarrival times and processing times are known and have the values

$$A_1 = 55,\ A_2 = 32,\ A_3 = 24,\ \ldots$$

$$S_1 = 43,\ S_2 = 36,\ \ldots$$

Thus, between 0 and the time when the first job arrives there are 55 time units (e.g., seconds or minutes), between the arrivals of the first and second jobs there are 32 time units, etc., and the processing time of the first job is 43 time units, etc.

Figure 2 gives a snapshot of the machine tool system itself and of a computer representation of the system at each of times 0, 55, 87, and 98. (The latter three times correspond to the occurrence of events.) Our discussion will focus on how the computer representation changes at each of the event times. In the first section of the figure, we show the computer representation of the simulation after the model has been initialized at time 0. Note that the status of the machine tool, the number of jobs in the queue, the simulation clock, and both statistical counters are initially set to zero. (We use 0 to represent a machine status of idle and 1 to represent a status of busy.) There is an array to store the times of arrival of all jobs in the system which is initially empty. For the event list, observe that the time of the next (first) job arrival (denoted by A) is set to 55 since $A_1 = 55$. Since no job is currently being processed, the time of the next departure (denoted by D) is set to the large positive number 10^{30} to guarantee that the next event which occurs is an arrival, as desired. Since $55 < 10^{30}$, the simulation clock is advanced to time 55 where an arrival of a job will occur. (In general, the simulation clock will be advanced from one event time to the next most imminent event time to the next most imminent event time, etc. Each time the clock is advanced, the simulation model is updated in accordance with the occurrence of the corresponding event.)

At time 55, the simulation model processes the arrival of a job. (The computer representation after all changes have been made at time 55 is shown in the second section of the figure.) Since the job arrives to find the machine idle (status equal to 0), it begins processing immediately and has no

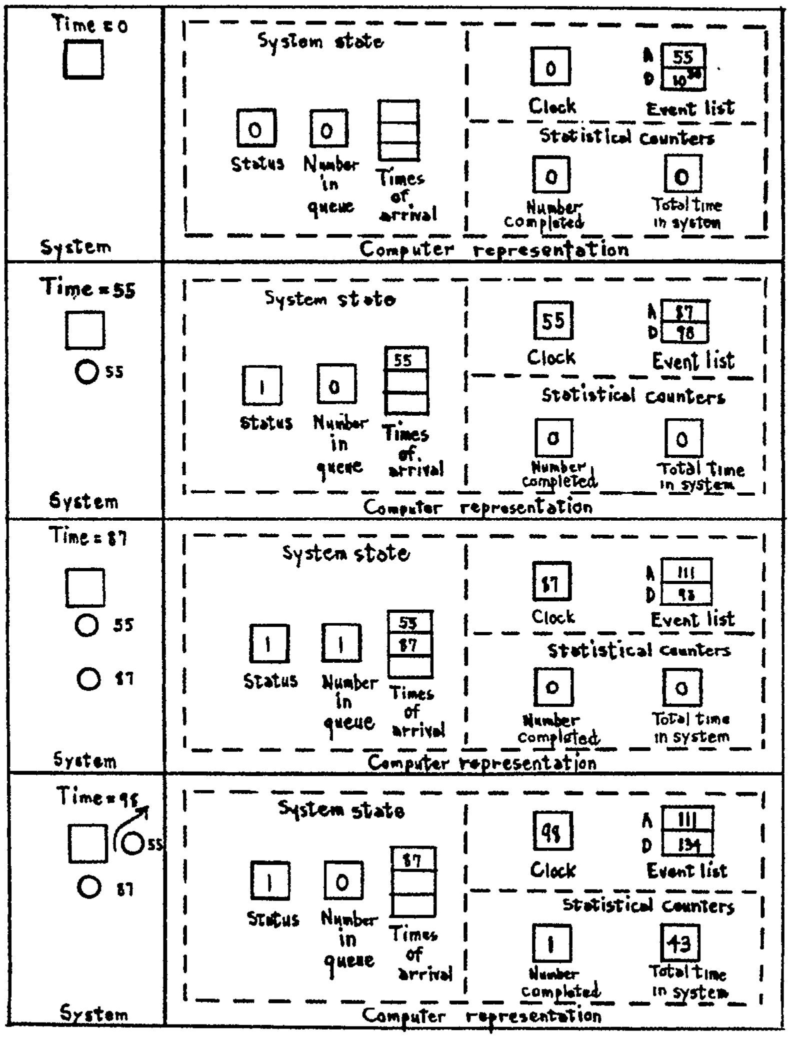

Figure 2: Snapshots of the System and of the Computer Representation of the System at Each of the Times 0, 55, 87, and 98

delay in queue. Note that the status of the machine has been set to 1, the time of arrival has been placed in the first location of the array, and the statistical counters are unchanged since the first job is just beginning processing. The time of the next arrival, 87, was determined by adding the second interarrival time, A_2 = 32, to the time of the first arrival, 55. Similarly, the time of the next departure (completion of processing), 98, was determined by adding the processing time of the first job, P_1 = 43, to the time that processing began, 55. Since 87<98, the simulation clock is advanced to time 87 where the arrival of another job will occur.

At time 87, the simulation model once again processes the arrival of a job. Since the arriving job finds the machine busy, it joins the queue, the number in queue is set to 1, and its time of arrival is placed in the second location of the array. The values of the statistical counters are unchanged since no job has completed processing. The time of the next arrival, 111, was determined by adding A_3 = 24 to the time of this arrival, 87, and the time of the next departure is unchanged. (The first job is still being processed at time 87.) Since 98<111, the simulation clock is advanced to time 98 where a departure (end of processing) will occur.

At time 98, the simulation model processes the departure of a job (the one which arrived at time 55) from the system. The time that this job spent in the system is computed as $W_1 = 98 - 55$, where 98 is the current value of the clock (the departure time) and 55 is the first entry in the array (the time of arrival). Then the statistical counters "number completed" and "total time in system" are updated to the values 1 and $43 = 0 + W_1$, respectively. Since there was a job in queue prior to this departure (number in queue was equal to 1), this job leaves the queue and begins being processed at time 98. The number in queue is reduced from 1 to 0 and the entries in the array are updated accordingly. The time of the next departure, 134, is computed by adding $P_2 = 36$ to the time the job (the one which arrived at time 87) is beginning service, 98, and the time of the next arrival is unchanged. Since $111<134$, the simulation clock is advanced to time 111 where an arrival will occur, etc.

The above procedure of advancing the simulation clock from one event time to the next event time is continued until some stopping rule is satisfied. At this time, an estimate of (expected) average time in system is obtained by dividing the total time in system by the number completed. For example, if the stopping rule is to run the simulation until 1000 completions have occurred, then the estimate of average time in system would be $(W_1 + W_2 + \dots + W_{1000})/1000$. An alternative stopping rule would be to run the simulation until a specified amount of simulated time has elapsed, say, 16 hours (the length of a work day consisting of two eight-hour shifts).

In the above explanation, we assumed that the values of the A_i's abd P_i's were simply given. However, in an actual simulation the A_i's and the P_i's would each have their own representative probability distributions. (Alternatively, for some manufacturing systems, processing times may be a constant.) The A_i's and P_i's are generated (using random numbers) from their corresponding probability distributions, as needed, during the course of the simulation. (See Section 5.1 for further discussion.)

1.3. The Steps in a Sound Simulation Study

There has been an unfortunate impression that simulation is just an exercise in computer programming, albeit a complicated one. Consequently, many simulation "studies" have been composed of heuristic model building, coding, and a single run of the program to obtain "the answers." This attitude, which neglects the important issues of how to develop a valid model and also of how to use a properly coded model to draw statistical inferences about the system of interest, has led to erroneous conclusions being drawn from many simulation studies.

In light of the above discussion, we present in Figure 3 the steps that will compose a typical, sound simulation study and the relationships between them. The number beside the symbol representing each step refers to the list of important comments for that step which is given below. Some studies may not necessarily contain all these steps and in the order stated; some studies may contain steps which are not depicted in the diagram. Furthermore, a simulation study is not a strictly sequential process.

1. Formulate the problem and plan the study
   - State the study's objectives clearly
   - Delineate the system designs to be studied (if possible)
   - Specify the criteria for comparing alternative system designs
   - Plan the study in terms of the number of people, the cost, and the time required for each aspect of the study
2. Collect data and define a model
   - Data should be collected on the system of interest (if it exists) to specify input parameters and probability distributions (e.g., a machine repair time distribution)
   - Data should be collected (if possible) on the performance of the system (e.g., throughput) to aid in validating the model
   - The level of model detail should be consistent with the study's objectives
3. Valid?
   - Involve people who are intimately familiar with the operations of the system (e.g., machine operators, industrial engineers, etc.) in the model building process
   - Analysts should interact with decision maker on a regular basis
4. Construct a computer program and verify
   - Decision must be made to use a general-purpose language (e.g., FORTRAN) or a simulation language (see Section 3)
   - Traces, structured walk-throughs, and animation should be used to debug (verify) the model
5. Make pilot runs
   - Used for validation purposes in Step 6
6. Valid?
   - Pilot runs can be used to test sensitivity of model's output to

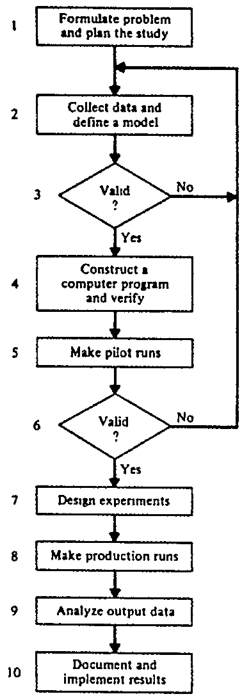

Figure 3: Steps in a Simulation Study

small changes in an input parameter

- o Compare output data from an existing system (from Step 2) to output data from pilot runs of the same system

7. Design experiments

- o Specify the system designs to be simulated
- o Specify the number of independent simulation runs for each alternative
- o Specify the length of each run
- o Specify the initial conditions for each simulation run (e.g., initial state of each machine, worker, etc.)

8. Make production runs

- o Simulation runs specified in Step 7 are made

9. Analyze output data

- o Estimate measures of performance for a particular system design
- o Determine best system design relative to some measure of performance

10. Document and implement results

- o Document model's assumptions as well as the computer program
- o Implement the results from the simulation study

2. OBJECTIVES IN SIMULATION OF MANUFACTURING SYSTEMS

In this section we discuss benefits of simulation, environments in which simulation is applied, manufacturing issues that simulation is used to address, and measures of performance in manufacturing simulation.

2.1. Benefits of Simulation in Manufacturing

The general benefit of simulation in manufacturing is that it allows a manager to obtain a system-wide view of the effect of changes on his or her manufacturing system (whether it exists or not). For example, the effect of adding an additional machine to a work station may be predictable by using simple queueing theory or back-of-the-envelope calculations; however, these techniques probably won't be adequate to determine what effect this change will have on the entire manufacturing system. Increasing the throughput at one work station might cause bottlenecks to develop at one or more other work stations.

Specific (potential) benefits of simulation in manufacturing are increased throughput, reduced in-process inventories, increased utilizations of machines and workers, increased on-time deliveries, and reduced capital requirements.

2.2. Manufacturing Environments in which Simulation is Applied

The following are three situations where simulation is applied in manufacturing:

i) New equipment and buildings are required (called "green fields")

- o An example is the building of the manufacturing facility for General Motors' new car, the Saturn

ii) New equipment is required in an old building

- o A new product will be produced in all or part of an existing building

iii) Upgrading of existing equipment or its operation

- o Concerned with producing the same product more efficiently
- o Changes may be to the equipment (e.g., introduction of a robot) or to operational procedures (e.g., scheduling rule employed)

2.3. Manufacturing Issues that Simulation is Used to Address

Below we list a number of manufacturing issues that simulation is used to address. These issues are broken into three general categories.

i) The need for and the quantity of equipment and personnel

- number and type of machines for a particular objective
- number, type, and physical arrangement of carts, conveyors, and other support equipment (e.g., pallets and fixtures)
- location and size of inventory buffers
- evaluation of a change in product mix (impact of new products)
- evaluation of the effect of a new piece of equipment on an existing manufacturing line
- evaluation of capital investments
- manpower requirements planning

ii) Performance evaluation

- throughput analysis
- makespan analysis
- bottleneck analysis

iii) Evaluation of operational procedures

- production scheduling (i.e., evaluating proposed policies for loading and sequencing machines)
- evaluation of policies for component part or raw material inventory levels
- evaluation of control strategies (e.g., for an automated guided vehicle system or a flexible manufacturing system)
- reliability analysis (e.g., effect of planned maintenance)
- evaluation of quality control policies

2.4. Measures of Performance in Manufacturing Simulation

The following is a list of measures of performance which are commonly used in manufacturing simulation studies:

- throughput (number of jobs produced per unit of time)
- time in system for jobs (makespan)
- times jobs spend in queues
- times that jobs spend being transported
- sizes of in-process inventories (WIP or queue sizes)
- utilizations of equipment and personnel (i.e., proportion of time busy)
- proportion of time that a machine is broken, blocked (i.e., unable to operate until current job is removed), starved (i.e., waiting for a job), or undergoing preventive maintenance
- proportion of jobs produced which must be reworked or scrapped
- return on investment for a new or modified manufacturing system (often given in terms of present value)
- payback period (time to earn back the money invested in a new or modified system)

3. SIMULATION LANGUAGES FOR MANUFACTURING

One of the major tasks in building a simulation model of a manufacturing system is converting a flowchart of a model of the system into an actual computer program. A simulation may use either a general-purpose language (e.g., FORTRAN or BASIC) or a simulation language for this purpose. The advantages of using a language like FORTRAN are that the language is probably already known by the analyst, is probably available on the analyst's computer, and that the required computer execution time may be less than for a model written in a simulation language since the program is tailor-made for the application. On the other hand, simulation models written in a general-purpose language tend to take a long time to develop since these languages are not particularly oriented toward simulation modeling.

Those simulation languages which are applicable to manufacturing problems may be further classified into two categories, general-purpose simulation languages and manufacturing-oriented languages/simulators. General-purpose simulation languages are useful for simulating a wide variety of systems (e.g., computer or military systems) in addition to manufacturing systems, but may contain certain features specifically for manufacturing. Examples of languages in this category are (in alphabetical order) GPSS (H, V, or PC), SEE WHY, SIMAN, SIMSCRIPT II.5, SLAM, and TESS. (Strictly speaking TESS is not a language, but rather a graphics/database interface.) These languages allow an analyst to develop a simulation model of a manufacturing system in less time than would generally be required when using a language like FORTRAN.

There are certain simulation software

packages which are designed specifically for simulating manufacturing-type problems, including AUTOMOD, MAP/1, SIMFACTORY, and WITNESS. The use of these packages may result in an additional reduction in programming time, since their modeling constructs are manufacturing oriented. The latter three packages in the above list (i.e., MAP/1, SIMFACTORY, and WITNESS) are actually simulators rather than languages, since a particular system within an available class of manufacturing systems is modeled by entering data rather than doing actual programming. In general, one would expect simulators to provide less modeling flexibility than languages.

The "hottest" feature which is currently available in many simulation languages is the capability for graphical animation of the simulation output. Important elements of a manufacturing system such as machines, workers, transporters, etc. are represented by icons on a graphics terminal or CRT. Every time there is a change in the state of the simulation, there is a corresponding change in the graphical representation. Thus an analyst or manager can graphically watch a manufacturing system change over time. Some animation software packages operate as post-processors, while others operate as the simulation executes. Among those which operate in real time, several packages allow an analyst to stop a simulation during execution, change parameters of the simulation, and then continue execution for this "new" system design. (This capability may exacerbate the problem of poor output data analyses.)

The following are some potential uses of animation:

i) Communicating the nature of a simulation model or its corresponding system to a manager

ii) Debugging a simulation computer program

iii) Showing a simulation model is <u>not</u> valid

iv) Training manufacturing personnel on the operation of a "new" system

v) Suggesting new control policies for material handling systems

On the other hand, some people may view animation as a substitute for a careful statistical analysis of the simulation output data (see Section 5.2). We strongly disagree with this attitude. For example, if a manufacturing system appears to operate properly during an animation of one hour of the system's operation, then this does not necessarily mean that the system is well defined. Machine breakdowns, which could result in system bottlenecks, may not have occurred during the animation.

4. DEVELOPING VALID AND CREDIBLE SIMULATION MODELS OF MANUFACTURING SYSTEMS

A simulation model is a surrogate for actually being able to experiment with a manufacturing system (see Section 1.1). Thus, an idealized goal in building a simulation model is for it to be accurate enough so that any conclusions drawn from the model would be the same as those derived from physically experimenting with the system (if this were possible). We will, however, not know for most simulation models whether this goal is realized. It is not necessary to have a one-to-one correspondence between each element of the actual system and each element of the simulation model. Indeed, a simulation model should be designed to meet a specified set of objectives, rather than to be universally valid.

We now briefly discuss what we feel are the most important ideas/techniques for deciding the appropriate level of model detail, for validating a simulation model, and for developing a model with high credibility. Note that if a model (or the modelers) is not credible, then it may never actually be used in the decision-making process by a manager even if the model is "valid." A list of these key ideas is as follows:

i) Define the issues to be investigated, the alternative system designs of interest, and the measures of performance for evaluation at the beginning of the study.

ii) Start with a simple model, which can later be embellished. This allows the analyst to get results to the manager/sponsor in a reasonable amount of time.

iii) Use "experts" and sensitivity analyses to help determine the level of model detail.

iv) Do not have more detail in the model than is necessary to address the issues of interest. On the other hand, a model must have enough detail to be credible.

v) If a similar existing manufacturing system exists, then talk to all important people associated with this system (e.g., machine operators, engineers, managers, vendors, etc.) and use this information to build the simulation model.

vi) Interact with the decision makers (or managers) throughout the course of the simulation, to help ensure that the model is both valid and understood.

vii) Perform a structured walk-through of the model's flowchart before an audience of all interested parties (see item v above), before the actual coding of the program begins.

viii) First, build a simulation model of a similar existing system (if possible). Compare the output data (e.g., throughput) from the model to output data from the actual existing system. If possible, perform this comparison by driving the model with actual shop floor input data (e.g., observed machine repair times).

5. STATISTICAL ISSUES IN MANUFACTURING SIMULATION

Since random samples from input probability distributions (e.g., a machine repair time distribution) are used to drive a simulation model through time, the output data from a simulation (e.g., daily throughputs) are also random samples from probability distributions. Therefore, it is important to model correctly the random inputs to a simulation model and also to design and analyze simulation experiments in a proper manner. These subjects are briefly discussed in this section.

5.1. Simulation Input Modeling

Most manufacturing systems contain one or more (input) sources of randomness (random variables). For example, interarrival times of jobs to a machine, processing times of jobs at a machine, machine running times before breakdown, machine repair times, and the outcomes of inspecting jobs (e.g., good, rework, or scrap) are possible examples of random variables in a manufacturing system. Furthermore, in order to model the system correctly, each random variable must be represented by an appropriate probability distribution in the simulation model. Procedures for selecting an appropriate probability distribution, which depend on whether system data are available or not, are discussed in Chapter 5 of Law and Kelton (see "REFERENCES").

Suppose that an appropriate probability distribution has been determined for each random variable in a simulation model. Then a random-number generator is used to generate random samples from these distributions as the simulation advances through time. (A random-number generator is typically a computer-based mechanism for generating a "random" value between 0 and 1, with each possible value being equally likely.) For example, suppose that the repair time for a broken machine is uniformly distributed on the interval [6, 10] minutes (i.e., each value between 6 and 10 minutes is equally likely). When the machine actually breaks down during the course of the simulation, a repair time is determined by generating a value from the random-number generator, multiplying this value by 4, and then adding 6.

We now present an example to illustrate the importance of correct input modeling for manufacturing systems. Suppose that a company is going to buy a new machine tool (see Section 1.1) from a vendor which claims that the machine will be down 10 percent of the time. However, the vendor has no data on how long the machine will operate before breaking down or on how long it will take to repair the machine. Historically, some analysts have accounted for random breakdowns by simply reducing the processing rate by 10 percent. We will see, however, that this can produce quite inaccurate results.

Suppose that the single machine tool system will actually operate in accordance with the following assumptions when installed by the purchasing company:

i) Jobs arrive with exponential interarrival times with a mean of 1.25 minutes.

ii) Processing times for a job at the machine are a constant 1 minute.

iii) The machine runs for an exponential amount of time with mean 540 minutes (9 hours) before breaking down.

iv) The repair time for the machine has a gamma distribution (shape parameter equal to 2) with mean 60 minutes (1 hour).

v) The machine is, thus, broken 10 percent of the time.

(The reader unfamiliar with exponential or gamma distributions should consult Chapter 5 of Law and Kelton.)

In column 1 of Table 1, we present results from five independent simulation runs (see Section 5.2) of length 160 hours (20 8-hour days) for the above system. In column 3 of the table are results from five simulation runs of length 160 hours for the machine tool system with no breakdowns, but with the processing (cycle) rate reduced from 1 job per minute to 0.9 job per minute. (This has sometimes been the approach of simulation practitioners.) Note first that the weekly throughput is almost identical for the two simulations. (For a system with no bottlenecks which is simulated for a long amount of time, the throughput for a 40-hour week must be equal to the arrival rate for a 40-hour week which is 1920.) On the other hand, note that such measures of performance as mean time in system for a job and maximum number of jobs in queue are vastly different for the two cases. Thus, the deterministic adjustment of the processing rate produces results which differ greatly from the correct results based on actual breakdowns of the machine.

In column 2 of Table 1 are results from five simulation runs of length 160 hours for the machine tool system with breakdowns, but with a mean running time of 54 minutes and a mean repair time of 6 minutes. (Thus, the machine is still broken 10 percent of the time.) Note that the mean time in system and the maximum number in queue are quite different for columns 1 and 2. Therefore, when explicitly accounting for breakdowns in a simulation model, it is also important to have an accurate assessment of mean running time and mean repair time for the actual system.

5.2. Design and Analysis of Simulation Experiments

One of the most common (and potentially dangerous) practices in simulating manufacturing systems is that of making only one run

Table 1: Simulation Results for the Single Machine Tool System

Measure of Performance	Breakdowns Mean = 540 min.	Breakdowns Mean = 54 min.	No breakdowns
Throughput per week	1908.8*	1913.8	1914.8
Mean time in system	35.1*	10.3	5.6
Maximum time in system	256.7#	76.1	39.1
Average number in queue	7.3*	7.3	3.6
Maximum number in queue	231#	67	35

All times are in minutes.

*Average over five runs.

#Maximum over five runs.

(or replication) of a stochastic simulation. For example, suppose that a manufacturing system operates sixteen hours a day and that we would like to estimate the mean (or expected) daily throughput or production. If we run the simulation only one time, then the value of the throughput from the simulation output is only one observation from a probability distribution whose mean is the desired expected daily throughput. (This is absolutely no different than trying to estimate the mean of a population in classical statistics with only one data point.) Furthermore, this single observed value of throughput may differ from the expected daily throughput by a large amount.

To emphasize the importance of the above point, consider a simple manufacturing system consisting of a machining center and an inspection station, as shown in Figure 4. Suppose that this system operates according to the following assumptions:

i) Jobs arrive to the machining center with exponential interarrival times with a mean of 1 minute.

ii) Processing times at the machining center are uniform on the interval [0.7, 0.8] minutes.

iii) After processing, jobs proceed to the inspection station where inspection times are uniform on the interval [0.8, 0.9] minutes.

iv) Ninety percent of the inspected parts are good and are sent to shipping; ten percent are bad and must be remachined.

v) The machining center is subject to randomly occurring breakdowns. In particular, a new (or freshly repaired) machining center will break down after an exponential amount of time with a mean of 6 hours.

vi) Repair times are uniformly distributed on the interval [8, 12] minutes.

In Table 2 we give selected results from five independent simulation runs of the manufacturing system (i.e., different random numbers are used for each run) each of length 16 hours. Note that results from different runs can be quite dissimilar. Thus, it is clear that one simulation run does not produce the "true answers" for the simulated system. If we want to estimate the expected daily throughput, then the average throughput across the replications (see the last row of the table) will be a better estimate, in general, than the observed throughput from only one run. Also, the maximum queue sizes in the last row of the table may be important in designing a manufacturing system because they are indicative of the amount of storage required for in-process inventory.

There is one additional issue related to the design and analysis of simulation experiments which we now discuss. When simulating manufacturing systems, we are often interested in the long-run behavior of the system, i.e., its behavior when operating in a "normal" manner. (In the above example, we were only interested in the behavior of the system over a 16-hour day.) On the other hand, simulations of manufacturing systems are often initialized in an empty and idle (or some other unrepresentative) state. This results in the output data (e.g., daily throughputs) from the "beginning" of the simulation not being represent-

Figure 4: Manufacturing System with Inspection Station

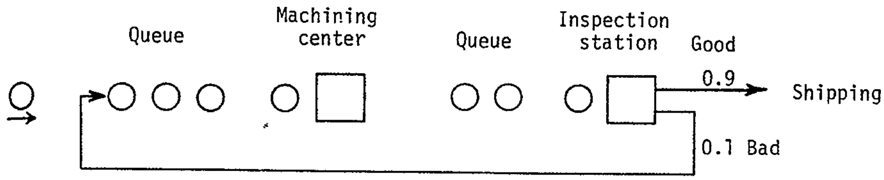

Figure 4: Manufacturing System with Inspection Station

Table 2: Simulation Results for Simple Manufacturing System with Breakdowns

Run	Throughput	Average time in system*	Maximum number in machine queue	Maximum number in inspector queue
1	972	19.0	21	33
2	922	7.6	19	12
3	963	20.6	20	53
4	930	6.4	17	9
5	896	7.1	23	12
Average or Maximum	936.6	12.1	23#	53#

*All times are in minutes.

#Maximum over five runs.

ative of the desired "normal" behavior of the system. Therefore, simulations are often run for a certain amount of time, the warm-up period, before the output data are actually used to estimate the desired measures of performance. A graphical approach for determining the length of the warm-up period is given in Welch.

6. SIMULATION ANALYSIS OF A MANUFACTURING SYSTEM

In this section we show, by means of an example, how simulation can be iteratively used to design a manufacturing system. However, due to space considerations we will not give a complete specification of the system.

A company is going to build a new manufacturing facility which will consist of a receiving/shipping (R/S) station and five work stations. The distances between the stations have been decided; however, one of the goals of the simulation study is to determine the number of machines needed in each work station.

Jobs arrive at the R/S station with exponential interarrival times having a mean of 1/15 hour. There are three types of jobs, with each type occurring with a specified probability. Furthermore, each type requires a specified number of tasks to be done, and each task must be done at a specified work station in a prescribed order. For example, a type 1 job begins at the R/S station, visits work stations 3, 1, 2, 5, and then leaves the system at the R/S station.

A job must be moved from one station to another by an automated guided vehicle (AGV), and one of the goals of the simulation study is to determine the required number of AGVs. When an AGV becomes available, it processes requests by jobs using a shortest-distance-first priority rule.

The machines in a particular work station are fed by a single first-in, first-out queue. Furthermore, the time to process a job at a particular machine has a gamma distribution (shape parameter equal to 2) whose mean depends on the job type and the work station to which the machine belongs. When a machine finishes processing a job, the job blocks that machine (i.e., the machine cannot process another job) until the job is removed by an AGV.

We will simulate the proposed system to determine how many machines are needed in each work station and also how many AGVs are needed to achieve an expected throughput of 120 jobs per 8-hour day (the maximum possible). Among those system designs which can achieve the desired expected throughput, the best system design will be chosen on the basis of such measures of performance as mean time in system of a job, maximum work station queue sizes, proportion of time transporters are busy, etc. For each system design, we will make 10 simulation runs of length 160 hours, with the first 24 hours of each run being a warm-up period (see Section 5.2).

We first simulated the system design consisting of 4, 2, 4, 3, and 2 machines in stations 1, 2, 3, 4, and 5, respectively, and 1 AGV; this system will be denoted by (4, 2, 4, 3, 2; 1). The results from these simulation runs are given in Table 3. Note that the average throughput per day is only 94.2, which is much less than the expected throughput of 120 for a well-defined system; it follows that this design must contain one or more bottlenecks. Since the average transporter utilization is so close to 1 and since the proportions of time that the machines are blocked are relatively large, we will add another AGV to the system.

The results from simulating system design (4, 2, 4, 3, 2; 2) are given in Table 4. Note that the average throughput is still somewhat less than 120, indicating that a bottleneck probably still exists. Since the statistics (e.g., maximum queue size) are very large for station 3, we will add another machine there.

The results from simulating system design (4, 2, 5, 3, 2; 2) are given in Table 5. The average throughput of 120.0 indicates that this system probably does not contain a bottleneck. Note also that the statistics for station 3 have been reduced considerably. To see the effect of adding additional machines (and to save space), we will now add machines to stations 1, 3, and 5.

The results from simulating system design (5, 2, 6, 3, 3; 2) are given in Table 6. The system is now somewhat balanced in the sense that the proportions of machine busy time and the maximum queue sizes are roughly equal for all stations. The choice between the latter two system designs depends on the performance requirements for the system and on the cost of machines. If floor space is very limited, then the last system design may be preferable since it results in smaller maximum queue sizes.

Table 3: Simulation Results for System Design (4, 2, 4, 3, 2; 1)

Station	1	2	3	4	5
Proportion machines busy	0.75	0.41	0.81	0.55	0.65
Proportion machines blocked	0.21	0.35	0.19	0.38	0.35
Average number in queue	10.6	0.9	214.2	4.9	63.3
Maximum number in queue	63	14	465	36	139

Average throughput: 94.2

Average transporter utilization: 0.996

Mean time in system: 20.4 hours

Table 4: Simulation Results for System Design (4, 2, 4, 3, 2; 2)

Station	1	2	3	4	5
Proportion machines busy	0.80	0.43	0.95	0.57	0.80
Proportion machines blocked	0.05	0.06	0.05	0.06	0.06
Average number in queue	3.0	0.2	67.5	0.4	4.1
Maximum number in queue	37	7	203	10	28

Average throughput: 114.5

Average transporter utilization: 0.70

Mean time in system: 5.9 hours

Table 5: Simulation Results for System Design (4, 2, 5, 3, 2; 2)

Station	1	2	3	4	5
Proportion machines busy	0.81	0.45	0.79	0.58	0.83
Proportion machines blocked	0.06	0.06	0.04	0.06	0.07
Average number in queue	3.8	0.2	1.9	0.5	4.8
Maximum number in queue	41	8	17	11	31

Average throughput: 120.0

Average transporter utilization: 0.71

Mean time in system: 1.6 hours

Table 6: Simulation Results for System Design (5, 2, 6, 3, 3; 2)

Station	1	2	3	4	5
Proportion machines busy	0.64	0.45	0.66	0.58	0.55
Proportion machines blocked	0.05	0.06	0.03	0.06	0.05
Average number in queue	0.5	0.3	0.5	0.5	0.4
Maximum number in queue	14	13	15	11	10

Average throughput: 120.0

Average transporter utilization: 0.70

Mean time in system: 1.0 hours

REFERENCES

Banks, J. and J. S. Carson, Discrete-Event System Simulation, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

Law, A. M. and W. D. Kelton, Simulation Modeling and Analysis, McGraw-Hill, New York, 1982.

Welch, P. D., "The Statistical Analysis of Simulation Results," in The Computer Performance Modeling Handbook, edited by S. S. Lavenberg, pp. 268-328, Academic Press, New York, 1983.

AUTHORS' BIOGRAPHIES

AVERILL M. LAW is President of Simulation Modeling and Analysis Company and Professor at the University of Arizona. His book Simulation Modeling and Analysis (coauthored with David Kelton) is widely used by industry and universities. He is also the author of three other books and numerous papers on simulation, operations research, and statistics. Dr. Law was the editor (and an author) of Industrial Engineering Magazine's recent series on the simulation of manufacturing systems. His seminars "Simulation of Manufacturing Systems" and "Simulation Modeling and Analysis" have been attended by more than 2000 people during the past ten years, and he has been a simulation consultant to more than 40 organizations. He received his Ph.D. in Industrial Engineering and Operations Research from the University of California at Berkeley.

Averill M. Law
Simulation Modeling and Analysis Company
P.O. Box 40996
Tucson, Arizona 85717
(602) 299-8441

MICHAEL G. McCOMAS is Vice President for Consulting Operations of Simulation Modeling and Analysis Company. He has considerable experience in building simulation models of manufacturing systems and in evaluating simulation software. He has an M.S. in Systems and Industrial Engineering from the University of Arizona.

Michael G. McComas
Simulation Modeling and Analysis Company
P.O. Box 40996
Tucson, Arizona 85717
(602) 299-8441