1 Introduction

"Where do we begin?"

1.1 Introduction

The objective of this chapter is to provide the simulation practitioner with some basic information about simulation modeling and analysis. Experienced practitioners using the handbook for reference purposes are advised to bypass this chapter and proceed to the appropriate chapter. Practitioners who have never received training in simulation or whose training is dated are strongly recommended to work through not only the examples but also the sample problems at the end of the chapter.

The chapter includes:

- An introduction to simulation modeling and analysis
- Other types of simulation

- Purposes of simulation
- Advantages and disadvantages of simulation
- Famous simulation quotes
- Basic simulation concepts
- A comprehensive example of a manual simulation

1.2 Simulation Modeling and Analysis

Simulation modeling and analysis is the process of creating and experimenting with a computerized mathematical model of a physical system. For the purposes of this handbook, a system is defined as a collection of interacting components that receives input and provides output for some purpose. Included within this field are traditional simulation and training simulators. In general, the distinction is as follows. Traditional simulation is used for analyzing systems and making operating or resource policy decisions. Training simulators are used for training users to make better decisions or improving individual process performance. A short section is included at the end of the handbook that discusses the basic nature of simulators.

Conceptos de simulación, sistema, simulación tradicional, simuladores de entrenamiento

The vast majority of this handbook concentrates on the field of simulation versus simulators. Although many different types of systems can be simulated, the majority of the systems that we discuss in this handbook are manufacturing, service, or transportation related.

Existen muchos tipos de sistemas que pueden ser simulados, en este libro se discuten en su mayoría aquellos relacionados con la manufactura, servicios o transportación.

Examples of manufacturing systems include:

- Machining operations
- Assembly operations
- Materials-handling equipment
- Warehousing

Machining operation simulations can include processes involving either manually or computer numerically controlled factory equipment for machining, turning, bending, cutting, welding, and fabricating. Assembly operations can cover any type of assembly line or manufacturing operation that requires the assembly of multiple components into a single piece of work. Material-handling simulations have included analysis of cranes, forklifts, and automatically guided vehicles. Warehousing simulations have involved the manual or automated storage and retrieval of raw materials or finished goods.

Casos específicos de ejemplos que pueden simularse para los sistemas de manufactura.

Examples of service systems include:

- Hospitals and medical clinics
- Retail stores
- Food or entertainment facilities
- Information technology
- Customer order systems

Hospital and medical clinic models can be simulated to determine the number of rooms, nurses, and physicians for a particular location. Retail stores may need to know how many checkout locations to utilize. Entertainment facilities such as multitheater movie complexes may be interested in how many ticket sellers, ticket checkers, or concession stand clerks to employ. Information technology models typically involve how many and what type of network or support resources to have available. Customer order systems may need to know how many customer order representatives are needed to be on duty.

Examples of transportation systems include:

- Airport operations
- Port shipping operations
- Train and bus transportation
- Distribution and logistics

Airport operations simulations have been performed on airport security checkpoints, check-in counters, and gate assignments. Port shipping operations can include how many cranes and trucks are needed to offload transportation ships. Train and bus transportation can include analysis involving routes. Distribution and logistics studies have included the analysis of shipping center design and location.

1.3 Other Types of Simulation Models

The types of simulation models previously discussed are not the only types of simulation model that the practitioner may encounter or have a need for. Another type of computer simulation model is the computer simulator. Though the distinction between simulation models and computer simulators may differ somewhat among practitioners, the following discussion may help differentiate these two types of simulation.

Otros tipos de simulación son: Modelos de simulación y los simuladores de computadoras.

So far, the types of simulation models that we have discussed have been models of actual or proposed systems. Models of the systems are normally created with different resource or operating policies that have been previously determined to be of interest. After the simulation runs, the output measures of performance are compared between or among the models. Thus, the ultimate use of the models is to make resource or operating policy decisions concerning the system.

Los modelos de simulación antes descritos corresponden a modelos actuales o propuestos de sistemas. Los modelos de sistemas generalmente se crean con recursos y políticas de operación distintas (según se requiera). Después de ejecutar la simulación, las medidas de las salidas se comparan entre modelos. Por tanto, el uso final de los modelos es tomar recursos o decisiones de operación que conciernan al sistema.

Simulators are also models of existing or proposed systems. In contrast to simulation models, resource and operating policy decisions are not made beforehand. These types of decisions are actually made during the simulation run. Thus, the output measures are observed not only at the end of the run but, more importantly, during the simulation run. The practitioner or user can see the effects of executing different resource and operating policy decisions in real time. Thus, the purpose of the simulator is not to make a decision but to expose the users to the system and to train them on how to make decisions. These types of simulators are often referred to as training simulators.

En los simuladores de computadoras las decisiones se toman durante la ejecución. Por tanto, las medidas de salida se observan no solo al final de las pruebas, sino durante la ejecución de la simulación. Es posible ver los efectos de diferentes recursos y decisiones de operación aplicadas en tiempo real. Su propósito no es tomar una decisión, sino el exponer al usuario para entrenarlo en la toma de decisiones. A estos tipos de simuladores se les conoce como simuladores de entrenamiento.

Although the principal focus of this book is on developing and analyzing simulation models, a short section has been included on simulators in Chapter 12, "Training Simulators."

1.4 Purposes of Simulation

The simulation modeling and analysis of different types of systems are conducted for the purposes of (Pedgen et al., 1995):

- Gaining insight into the operation of a system
- Developing operating or resource policies to improve system performance
- Testing new concepts and/or systems before implementation
- Gaining information without disturbing the actual system

Propósitos de la simulación:
El modelado de simulación y el análisis de diferentes tipos de sistemas se lleva a cabo para los siguientes propósitos:
1. Obtener conocimiento de la operación interna del sistema.
2. Desarrollar políticas de operación y recursos para mejorar el rendimiento del sistema.
3. Probar nuevos conceptos y/o sistemas antes de la implementación.
4. Obtener información sin interrumpir el sistema actual.

1.4.1 Gaining Insight into the Operation of a System

Some systems are so complex that it is difficult to understand the operation of and interactions within the system without a dynamic model. In other words, it may be impossible to study the system by stopping it or by examining individual components in isolation. A typical example of this would be to try to understand how manufacturing process bottlenecks occur.

Algunos sistemas son tan complejos que es difícil entender su operación y las interacciones dentro de él cuando no se tiene un modelo dinámico. Es casi imposible estudiar un sistema al detenerlo o al examinar sus componentes de forma individual. Un ejemplo puede ser tratar de entender cómo ocurren los procesos de cuello de botella en la manufactura...

1.4.2 Developing Operating and Resource Policies

You may also have an existing system that you understand but wish to improve. Two fundamental ways of doing this are to change operating or resource policies. Changes in operating policies could include different scheduling priorities for work orders. Changes in resource policies could include staffing levels or break scheduling.

Es posible tener un sistema que se conoce y se desea mejorar. Hay dos caminos fundamentales para llevarlo a cabo: cambiando la operación o las políticas de los recursos. Los cambios en las políticas de operación incluyen el esquematizar prioridades para ordenes de trabajo. Los cambios en las políticas incluyen niveles de dotación de personal y programación de breaks.

1.4.3 Testing New Concepts

Probar conceptos
Si un sistema no existe aún, o si estás pensando en comprar uno, un modelo de simulación puede ayudar a tener una idea de que tan bien se comportará el nuevo sistema.
El costo de modelar un nuevo sistema puede ser muy barato respecto al capital involucrado en instalar cualquier proceso de manufactura.

If a system does not yet exist, or you are considering purchasing new systems, a simulation model can help give you an idea how well the proposed system will perform. The cost of modeling a new system can be very small in comparison to the capital investment involved in installing any significant manufacturing process. The effects of different levels and expenses of equipment can be evaluated. In addition, the use of a simulation model before implementation can help refine the configuration of the chosen equipment.

Currently, a number of companies require vendors of material-handling equipment to develop a simulation of their proposed systems before purchase. The simulation model is used to evaluate the various vendors' claims. Even after the installation, the simulation model can be useful. The company can use the simulation model to help identify problems should the installed system not operate as promised.

Actualmente, varias compañías requieren de sus vendedores de materiales para el manejo de equipos, que desarrollen una simulación de sus sistemas propuestos antes de la compra.
Los modelos de simulación permiten evaluar muchas de las reclamaciones del cliente.

1.4.4 Gaining Information without Disturbing the Actual System

Obtener info. sin interrumpir el sistema actual:
La simulación es el único método disponible de experimentación de sistemas que no pueden ser interrumpidos.
Algunos sistemas son tan críticos o sensibles que no es posible hacer algun

Simulation models are possibly the only method available for experimentation with systems that cannot be disturbed. Some systems are so critical or sensitive that it is not possible to make any types of operating or resource policy changes to analyze the system. The classical example of this type of system would be the security checkpoint at a commercial airport. Conducting operating policy or resource level experimentation would have serious impact on the operational capability or security effectiveness of the system.

1.5 Advantages to Simulation

In addition to the capabilities previously described, simulation modeling has specific benefits. These include:

- Experimentation in compressed time
- Reduced analytic requirements
- Easily demonstrated models

Ventajas de la experimentación:
i} Experimentación en tiempo reducido.
ii) Reducción de requerimientos analíticos.
iii) Modelos fácilmente demostrables.

1.5.1 Experimentation in Compressed Time

Because the model is simulated on a computer, experimental simulation runs may be made in compressed time. This is a major advantage because some processes may take months or even years to complete. Lengthy system processing times may make robust analysis difficult or even impossible to perform. With a computer model, the operation and interaction of lengthy processes can be simulated in seconds. This also means that multiple replications of each simulation run can easily be run to increase the statistical reliability of the analysis. Thus, systems that were previously impossible to analyze robustly can now be studied.

1.5.2 Reduced Analytic Requirements

Before the existence of computer simulation, practitioners were forced to use other, more analytically demanding tools. Even then, only simple systems that involved probabilistic elements could be analyzed by the average practitioner. More complex systems were strictly the domain of the mathematician or operations research analyst. In addition, systems could be analyzed only with a static approach at a given point in time. In contrast, the advent of simulation methodologies has allowed practitioners to study systems dynamically in real time during simulation runs. Furthermore, the development of simulation-specific software packages has helped insulate practitioners from many of the complicated background calculations and programming requirements that might otherwise be needed. These reduced analytic requirements have provided more practitioners, with a wider variety of backgrounds, with the opportunity to analyze many more different types of systems than was previously possible.

1.5.3 Easily Demonstrated Models

Most simulation-specific software packages possess the capability of dynamically animating the model operation. Animation is useful both for debugging the model and also for demonstrating how the model works. Animation-based debugging allows the practitioner to observe flaws in the model logic easily. The use of an animation during a presentation can help establish model credibility. Animation can also be used to describe the operation and interaction of the system processes simultaneously. This includes dynamically demonstrating how the system model handles different situations. Without the capability of animation, practitioners would be limited to less effective textually and numerically based presentations.

1.6 Disadvantages to Simulation

Aunque la simulación tiene muchas ventajas, hay también desventajas de las cuales hay que tener cuidado. Las desventajas no están asociadas directamente con el modelado y el análisis de un sistema sino con las expectativas asociadas con los proyectos de simulación.
Estas desventajas incluyen lo sig:
1. La simulación no da resultados precisos cuando los datos de entrada son imprecisos.
2. La simulación no proporciona respuestas rápidas a problemas complejos.
3. La simulación no puede resolver problemas por sí misma.

Although simulation has many advantages, there are also some disadvantages of which the simulation practitioner should be aware. These disadvantages are not really directly associated with the modeling and analysis of a system but rather with the expectations associated with simulation projects. These disadvantages include the following:

- Simulation cannot give accurate results when the input data are inaccurate.
- Simulation cannot provide easy answers to complex problems.
- Simulation cannot solve problems by itself.

1.6.1 Simulation Cannot Give Accurate Results When the Input Data Are Inaccurate

No importa que tan bueno sea el modelo desarrollado, si el modelo no tiene datos de entrada precisos, no puede esperarse el obtener resultados de salida adecuados. Desafortunadamente, la recolección de datos es considerada la más difícil parte del proceso de simulación.
Este problema se incrementa con el hecho de que los practicantes probablemente prefieren desarrollar un modelo de simulación más que recolectar datos mundanos.
Muchos son atraídos a aceptar datos históricos de dudosa calidad para ahorrarse tiempos de recolección de datos de entrada.

The first statement can be paraphrased as "garbage in, garbage out." No matter how good a model is developed, if the model does not have accurate input data, the practitioner cannot reasonably expect to obtain accurate output data. Unfortunately, data collection is considered the most difficult part of the simulation process. Despite this common knowledge, it is typical for too little time to be allocated for this process. This problem is aggravated by the fact that many practitioners probably prefer to develop a simulation model rather than collect mundane data.

Many simulation practitioners are lured into accepting historical data of dubious quality in order to save input data collection time. All too often the exact nature or the conditions under which these data were collected is unknown. In more than one case, the use of externally collected historical data has been the foundation of an unsuccessful simulation project.

1.6.2 Simulation Cannot Provide Easy Answers to Complex Problems

Some analysts may believe that a simulation analysis will provide simple answers to complex problems. In fact, it is more likely that complex answers are required for complex problems. If the system analyzed has many components and interactions, the best alternative operating or resource policy is likely to consider each element of the system. It is possible to make simplifying assumptions for the purpose of developing a reasonable model in a reasonable amount of time. However, if critical elements of the system are ignored, then any operating or resource policy is likely to be less effective.

Si el sistema analizado tiene muchos componentes e interacciones, la mejor alternativa es considerar cada elemento del sistema.
Es posible hacer suposiciones para desarrollar modelos en cantidades de tiempo razonables.
Sin embargo, si se ignoran partes criticas del sistema, entonces cualquier operación o recurso será menos efectivo.

1.6.3 Simulation Alone Cannot Solve Problems

Some managers, on the other hand, may believe that conducting a simulation model and analysis project will solve the problem. Simulation by itself does not actually solve the problem. It provides the management with potential solutions to solve the problem. It is up to the responsible management individuals to actually implement the proposed changes. For this reason, it is to the advantage of the practitioner to

La simulación en sí misma no resuelve problemas. Proporciona un medio con soluciones potenciales para resolver el problema. Es la responsabilidad del director el implementar los cambios propuestos.
Es por ello que se requiere mantener al director y a todos los involucrados en el proyecto al día.

keep the manager or customer stakeholders as involved in the project as much as possible. The practitioner should strive to have the stakeholders develop a sense of ownership in the simulation process. All too often, potential solutions are developed but are never or only poorly implemented because of organizational inertia or political considerations.

1.7 Other Considerations

In addition to the advantages and disadvantages to simulation modeling and analysis previously discussed, the practitioner should be aware of some other serious considerations when embarking on a project. These considerations may influence the practitioner's decision whether or not to undertake the project alone or even to undertake the project at all. These include the following:

- Simulation model building can require specialized training.
- Simulation modeling and analysis can be costly.
- Simulation results involve many statistics.

Existen otras consideraciones a tomar en cuenta, tales que pueden influenciar en las decisiones del practicante:
1. La construcción del modelo puede requerir entrenamiento especializado.
2. El modelado y la simulación pueden ser costos.
3. Los resultados de la simulación involucran muchas estadísticas.

1.7.1 Simulation Model Building Can Require Specialized Training

In the past, simulation modeling used to be extremely difficult to perform. In the days before graphic displays, all modeling was performed with arcane simulation languages with a text editor. It was necessary to create the source code, compile, link, and run the program. If any comma, colon, or period was out of place, the practitioner received a slew of compiling errors. Thus, without strong computer programming skills it was difficult to build successfully anything other than the simplest model. Fortunately for us, the advent of the powerful multimedia personal computer has brought simulation modeling more into the realm of the practitioner. The arcane simulation programming languages have given way to reasonably easy-to-use graphic interfaces. However, the overall simulation modeling and analysis process can still be complex. Many accomplished simulation analysts do have engineering, computer science, mathematics, or operations research degrees with specific coursework in simulation modeling and analysis.

Se requieren habilidades en programación de computadoras.

1.7.2 Simulation Modeling and Analysis Can Be Very Costly

There is no question that the development of a complex simulation model can be very time consuming and hence costly. Even if the practitioner is proficient with a given simulation software package, a complex system still will require a proportionally larger amount of time for data collection, modeling building, and analysis. Some models have a way of initially appearing to be relatively simple. However, once the practitioner actually begins modeling, he or she may realize that the system is far more complex than it originally appeared. Although many simplifying modeling assumptions may be made to reduce the amount of development time, the assumptions may also be so extreme as to render the model invalid. So, just as with the collection of input data, there is a limit to how much time may be saved by cutting corners with the model development process.

No hay duda de que desarrollar un modelo de simulación complejo puede consumir mucho tiempo y ser costoso.

1.7.3 Simulation Results Involve Many Statistics

Finally, simulation results are usually in the form of summary statistics. For this reason, simulation results may be difficult for individuals without any statistical knowledge to interpret. It is assumed that any practitioner utilizing this handbook have at least a limited knowledge of statistics. Some of the simulation-specific types of statistics presented in this handbook may not be familiar at all, even to statistically proficient practitioners. Although a few of the statistical techniques presented in this handbook may be new to many practitioners, a step-by-step format has been incorporated to minimize potential difficulties.

Finalmente los resultados de la simulación están en forma de resumen estadístico. Por esta razon, los resultados de simulación puede ser difíciles para individuos sin ningún conocimiento estadístico para interpretarlos.

1.8 Famous Simulation Quotes

Now that we have some familiarity with simulation, it is time to introduce a few famous simulation quotations that have been recorded over the years:

- "You cannot study a system by stopping it."
- "Run it again."
- "Is that some kind of game you're playing?"

1.8.1 "You Cannot Study a System by Stopping It"

The first quotation is believed to have originated from the science fiction series *Dune* by Frank Herbert. It was in reference to a particular problem that the futuristic scientists were attempting to analyze. If the system was stopped long enough to analyze it, it would alter the nature of the system. Thus, the only way to study it properly was while it was in motion. Computer simulation allows the model of the system to be in motion. Any output measures of performance are acquired while the system is in operation. This quotation also focuses on computer simulation's ability to study an actual system while it is still in operation.

1.8.2 "Run It Again"

The second quotation is from an episode of *StarTrek: The Next Generation.* In the episode, the starship Enterprise was damaged and trapped inside a debris field. The chief engineering officer Geordi LaForge developed a plan to extricate the Enterprise from the debris field. The difficulty with the plan was that if it failed, it would deplete all of the Enterprise's energy resources and the fate of the Enterprise would be in question. The Enterprise's Captain Jean Luc Picard instructed Commander LaForge to run a simulation of his proposed plan. The simulation run indicated that Commander LaForge's plan would successfully remove the Enterprise from the debris field. However, Captain Picard, having some familiarity with simulation, ordered Commander LaForge to "run it [the simulation] again." The result of the second simulation run was that the Enterprise would not be able to remove itself successfully from the debris field. These conflicting simulation results convinced Captain Picard to seek an alternate plan for the Enterprise.

This example emphasizes that probabilistic systems cannot be analyzed with a single simulation run. The innate variability of probabilistic systems results in corresponding variability in any system output. Thus, in most of the systems that simulation is used to analyze, a single simple simulation run is not sufficient for making serious resource or operating policy decisions.

1.8.3 "Is That Some Kind of Game You Are Playing?"

The last quotation was directed toward one of us during work on a project that involved modeling and analyzing a multitheater movie complex. While in the process of animating this system another individual happened to pass by. The animation illustrated customers purchasing tickets, having their tickets collected, going to the concession area, and finally entering the theater. On observing the animation, the individual asked if the computer model was actually some kind of game. After recovering from his initial shock, the practitioner replied that it was actually a highly complex mathematical model based on probability distributions and statistics. Apparently unimpressed, the visitor shrugged his shoulders and left the practitioner to complete the model.

This example illustrates the danger that uneducated or uninformed individuals will fail to appreciate the complexity behind a sophisticated simulation model. Many of these types of individuals have the capacity to comment only on the animation aspect of the simulation model. Unfortunately, many of these individuals are the same ones whom you must convince that your model is useful for making significant resource and operating policy decisions. Not only must your model be mathematically and logically correct, it must also look convincing to the uninformed.

1.9 Basic Simulation Concepts

Virtually all simulation modeling that the practitioner is likely to be involved in will be implemented in a simulation specific-software package such as ARENA, AutoMod, Simscript, or SimPak. Although these simulation packages facilitate and speed the development time for a simulation model, they may insulate the practitioner from an understanding of the basic concepts of simulation modeling. The following section can introduce or refamiliarize the practitioner with these concepts before he or she proceeds with the remainder of the handbook. In this section we discuss:

- Basic simulation model components
- Simulation event lists
- Measures of performance statistics

1.9.1 Basic Simulation Model Components

For demonstration purposes, consider the simplest possible system that may be of interest to the practitioner. Examples of this simple type of system would include, but not be limited to:

- A customer service center with one representative
- A barber shop with one barber
- A mortgage loan officer in a bank
- A piece of computer-controlled machine in a factory
- An ATM machine

Each of these simple systems consists of three types of major components:

- Entities
- Queues
- Resources

The relationships among these components are illustrated in Figure 1.1.

Componentes básicos de un modelo de simulación. Los ejemplos de estos tipos de sistemas incluyen y no están limitados a:
1. Centro de servicios al cliente con un representante.
2. Una peluquería con un peluquero.
3. Una oficina de préstamos hipotecarios en un banco.
4. Una maquina para controlar piezas de computadora en una fabrica.
5. Una máquina ATM.
Cada uno de estos sistemas contiene tres tipos de componentes principales:
* Entidades.
* Colas.
* Recursos.
La relación entre los componentes de muestra en la figura 1.1.

1.9.1.1 Entities

The first type of component is an entity: something that changes the state of the system. In many cases, particularly those involving service systems, the entity may be a person. In the customer service center, the entities are the customers. Entities do not necessarily have to be people; they can also be objects. The entities that the mortgage loan officer deals with are loan applications. Similarly, in the factory example, the entities are components waiting to be machined.

Entidades: algo que cambia de estado al sistema. En muchos casos, particularmente en aquellos que involucran sistemas de servicios, las entidades pueden ser personas.
En el centro de servicios al cliente, las entidades son los clientes.
Las entidades no necesariamente tienen que ser personas, también pueden ser objetos.
Las entidades en la oficina de préstamos hipotecarios son las solicitudes de préstamos.
Similarmente, en el ejemplo de la fábrica, las entidades son los componentes que esperan ser producidos.

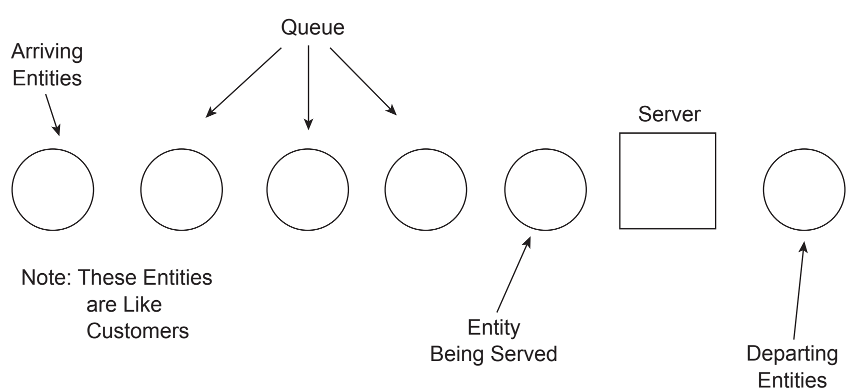

FIGURE 1.1 Basic simulation model components.

1.9.1.1.1 Entity Batches

The number of entities that arrive in the system at the same given time is known as the batch size. In some systems, the batch size is always one. In others, the entities may arrive in groups of different sizes. Examples of batch arrivals are families going to a movie theater. The batch sizes may be two, three, four, or more.

Es el número de entidades que llegan al sistema al mismo tiempo. En algunos sistemas el batch size es de uno. En otros, las entidades pueden llegar en grupos de tamaños distintos. Ejemplos de llegadas de batch son familias yendo a un cine. El batch size puede ser dos, tres, etc.

1.9.1.1.2 Entity Interarrival Times

Cantidad de tiempo entre llegadas de batches(lotes). No importa si el batch size es uno o más. Nos interesa solo el intervalo desde que el último lote llegó hasta que llega el actual.

The amount of time between batch arrivals is known as the interarrival time. It does not matter whether the normal batch size is one or more. We are interested only in the interval from when the last batch arrived to when the current batch arrives. The previous batch may have had only one entity, whereas the next batch has more than one. Interarrival time is also the reciprocal of the arrival rate. In collecting entity arrival data it is usually easier to collect the batch interarrival time. However, some historical data may be in arrival rate format. Interarrival time is considered input data that the practitioner would have to provide for the model.

1.9.1.1.3 Entity Attributes

Entities may also possess attributes. These are variables that have values unique to each entity in the system. Even though the entity attribute will have the same name, there could be as many different values as there are entities. An example of an attribute of this type involves the entity's arrival time. Each entity's attribute ARRTIME would store the simulation system time that the entity arrived in the system. So, unless a batch of entities arrived at the same time, each entity would have a unique value in its attribute ARRTIME. Some entity attributes may have the same value. In the case of airline passengers, the attribute PASSTYPE could hold a value corresponding to the type of passenger the entity represents. A value of 1 in PASSTYPE could represent a first-class passenger, and a value of 2 could represent a coach-class passenger. Thus, 20% of the entities in a simulation might have a value of 1 for PASSTYPE, and the remaining 80% would have a value of 2 for PASSTYPE. In the actual model, the attribute PASSTYPE would be used to prioritize the servicing and loading of passengers.

Simulation programs may also utilize global variables. Global variables are not to be confused with entity attributes. These variables differ from entity attributes in that each global variable can maintain only one value at a given time. A typical use of a global variable in a simulation program is the variable that keeps track of the simulation run time.

Las entidades poseen atributos.
Un ejemplo de un atributo involucra el tiempo de llegada de una entidad. Cada atributo de entidad ARRTIME puede almacenar el tiempo en el que la entidad llegó al sistema. Así que aunque un lote de entidades llegue al mismo tiempo, cada entidad tendrá un valor único en ese atributo ARRTIME.
Así que los atributos de una entidad pueden tener el mismo nombre.
En el caso del pasajero de una aerolinea, el atributo PASSTYPE puede contener un valor que corresponde al tipo de pasajero que la entidad representa.
Los programas de simulación también podrán usar variables globales, tales que no deben confundirse con atributos de la entidad.

1.9.1.2 Queues

Colas:
El segundo mayor componente que un sistema posee son las colas.
Las colas son el término de simulación para las líneas. Las entidades generalmente esperan en una cola hasta que sea su turno para ser procesadas.
Los sistemas simples usan generalmente FIFO.
Una vez que las entidades se encuentran en la cola, no pueden salirse hasta que reciban servicio.

The second major type of components that simple systems possess is queues. Queues are the simulation term for lines. Entities generally wait in a queue until it is their turn to be processed. Simple systems generally use first-in-first-out (FIFO) queue priorities. Another characteristic of simple systems is that once customers enter the system, they must enter the queue. Furthermore, once entities enter the queue, they cannot depart before receiving service. We explore different variations of queue priorities and queue behavior in Chapter 4, "System Definition."

1.9.1.3 Resources

The third component that simple systems contain is resources. Resources process or serve the entities that are in the queue. Examples of resources are:

- Customer service representatives
- Barbers
- Loan officers
- Factory machines
- ATMs

El tercer componente de un sistema simple son los recursos. Los recursos procesan o sirven a las entidades que se encuentran en la cola. Ejemplos de recursos son:
* Representantes de servicios al cliente.
* Peluquero.
* Los oficiales de créditos.
* Máquinas de la fábrica.
* ATMs.

In simple models, resources can be either idle or busy. Resources are idle when they are available for processing, but there are no more entities waiting in the queue. Resources are busy when they are

En modelos simples los recursos pueden estar ocupados u ociosos. Los recursos están ociosos cuando están disponibles para procesar, pero no hay más entidades esperando en la cola.

processing entities. In more complex models, resources may also be temporarily inactive or failed. Inactive resources are unavailable because of:

- Scheduled work breaks
- Meals
- Vacations
- Preventive maintenance periods

Los recursos están ocupados cuando están procesando entidades. En modelos más complejos, los recursos pueden estar temporalmente inactivos o haber fallado.
Los recursos se encuentran inactivos debido a:
* Breaks programados.
* Comidas.
* Vacaciones.
* Periodos de mantenimiento preventivo.

Failed resources would correspond to:

- Broken machines
- Inoperative equipment

Los recursos fallan debido a:
* Máquinas descompuestas.
* Equipo inoperable.

Resources take a certain amount of processing time to service the entities, for example, the time to total an order and receive payment, process a loan, or machine a part. The processing time is also frequently referred to as processing delay time or service time. Processing time is considered input data that the practitioner would normally have to collect by observation or otherwise acquire.

Los recursos toman ciertas cantidades de tiempo para servir a las entidades, por ejemplo, el tiempo total para ordenar y recibir una orden, procesar un préstamo,...
El tiempo de procesamiento se refiere frecuentemente como a un retraso en el tiempo o tiempo de servicio.
El tiempo de procesamiento se considera como un dato de entrada que el practicante debe recolecta a través de la observación o de otra forma adquirirlo.

1.9.2 The Simulation Event List

The simulation event list is a means of keeping track of the different things that occur during a simulation run (Law and Kelton, 2000). Anything that occurs during the simulation run that can affect the state of the system is defined as an event. Typical events in a simple simulation include entity arrivals to the queue, the beginning of service times for entities, and the ending of service times for entities. These events change the state of the system because they can increase or decrease the number of entities in the system or queue or change the state of the resources between idle and busy.

La lista de eventos de simulación:
Es una lista para llevar el control de las distintas cosas que ocurren durante la simulación.
Cada cosa que ocurre en la simulación que puede afectar el estado del sistema se define como un evento. Los eventos típicos en un sistema de simulación simple incluyen llegadas de entidades a la cola, el inicio del tiempo de servicio de entidades, el final del tiempo de servicio de entidades. Estos eventos cambian el estado del sistema debido a que pueden incrementar o decrementar el número de entidades en el sistema o cola, o cambiar el estado de los recursos entre ocioso y ocupado.

The event list is controlled by advances in the simulation clock. In our basic simulation model, the simulation clock advances in discrete jumps to each event on the event list. This type of model is called a discrete event simulation. In more sophisticated models, the simulation clock may operate continuously. This type of model is usually associated with processes involving fluid or material that could be modeled as fluids. These types of models involve continuous event simulation. Systems that require continuous event simulation are usually significantly more difficult to model because they involve the use of differential equations. It is also possible to model a system that involves both discrete and continuous components. An example of this would be a refinery that fills tanker trucks. The refinery tanks that store liquid would require continuous simulation, while the individual tanker trucks would need to be modeled discretely.

La lista de eventos es controlada por el reloj de simulación. En un modelo básico de simulación, el reloj avanza en saltos discretos hacia cada evento de la lista de eventos. A este tipo de simulación se le llama simulación de eventos discretos.
En modelos más sofisticados, el reloj puede operar de forma continua. A este tipo de modelo se le asocia con el proceso de fluidos. Involucran eventos de simulación continuos. Estos sistemas son más dificiles de modelar debido a que involucran el uso de ecuaciones diferenciales.

Regardless of whether the model is discrete, continuous, or combined, the simulation event list is extremely important to the practitioner. In even our very simple simulation model, many different events can occur simultaneously. For example, entities may arrive at any give time, or a service period may end at any given time. This means that one moment an entity may arrive, and a second entity may arrive before the first entity receives service. Similarly, the first entity may arrive, receive processing, and depart before the second entity arrives. The arrival, service start, and service end processes can take on an infinite number of possible sequences. Without a formal means of keeping track of these events, the output measures of performance of the system would become hopelessly complicated. This is, in fact, the reason it is virtually essential to implement any simulation on a computer system.

1.9.3 Measures of Performance Statistics

We are almost always interested in how well the actual system and hence the system model performs. In order to ascertain this we will need to calculate some sort of output measure to eventually compare with other alternative forms of the model. Output measures of performance can be either observational or time dependent. Observational performance measures are based on the number of entities observed

Medidas de rendimiento estadísticas:
Las medidas de salida pueden provenir de la observación o ser dependientes del tiempo.

going through the process. Conversely, time-dependent measures are based on the length of time the statistics are collected. There are four commonly utilized measures of performance (Kelton et al., 2002):

1. System time
2. Queue time
3. Time average number in queue
4. Utilization

Las medidas que se basan en el tiempo en el que los datos son recolectados. Hay 4 medias utilizadas comúnmente para el rendimiento:
1. Tiempo del sistema.
2. Tiempo de la cola.
3. Tiempo promedio de datos en la cola.
4. Utilización.

1.9.3.1 System Time

System time is an observational output measure. It is the total amount of time that the entity spends in the system. System time begins when the entity arrives in the system and enters the queue. It ends when the entity's service time is complete and it exits the system. The average system time for all of the entities is of most importance to the practitioner. The mathematical representation of the average system time is

$$\text{Average System Time} = \frac{\sum_{i=1}^{n} T_i}{n}$$

where T_i = the system time for an individual entity (arrival time – departure time) and n = the number of entities that are processed through the system.

Tiempo del sistema.
Es una medida observacional. Es la cantidad total de tiempo que la entidad gasta en el sistema. El sistema inicia cuando la entidad llega y entra a la cola. Termina cuando el servicio de la entidad se ha completado y sale del sistema. El tiempo de sistema promedio para todas las entidades es el más importante para los practicantes.
La representación matemática del tiempo promedio es:
Ti = tiempo del sistema para una entidad individual (tiempo llegada-tiempo salida)
n = número de entidades que son procesadas en el sistema.

1.9.3.2 Queue Time

Queue time is also an observational measure. It is similar to system time, except it accounts only for the time that an entity spends in the queue. Queue time is preferred by some practitioners because they suspect that the most objectionable time period, at least in customer-oriented service processes, is the waiting time in the queue. Many customers are at least partially satisfied when their service times begin, even though the service time itself may be lengthy. The formula for queue time is

$$\text{Average Queue Time} = \frac{\sum_{i=1}^{n} D_i}{n}$$

where D_i = the queue time for an individual entity (queue arrival time – service begin time) and N = *the* number of entities that are processed through the queue.

Tiempo de cola:
También es una medida observacional. Similar a la del tiempo del sistema, excepto que cuenta solo el tiempo que la entidad gasta en la cola. Es preferido por algunos practicantes debido a que es el más objetable periodo de tiempo, al menos en procesos de servicios al cliente. Muchos clientes están menos satisfechos si dicho tiempo se extiende.
La formula para calcularlo es:
Donde Di = tiempo de cola para una entidad individual(tiempo de llegada a la cola - tiempo en que se inició el servicio)
N = el número de entidades que son procesadas en la cola.

1.9.3.3 Time-Average Number in Queue

The time-average number in queue is a time-dependent statistic. As a time-dependent statistic, the time-average number in queue is not directly a function of the number of entities that have been processed through the queue. It is rather the average number of entities that you could expect to see in the queue at any given time during the period of interest. At any given time the queue will actually have a discrete number of entities. However, because the time-average number in queue is an average value, it will usually yield a number that also has a fractional value. For lightly loaded queues it is actually possible for the time average number in queue to be less than 1. The formula for calculating the time-average number in queue is

$$\text{Time Average Number in Q} = \frac{\int_0^T Q\,dt}{T}$$

Tiempo promedio de datos en la cola:
Es una estadística dependiente del tiempo.
Es el número promedio de entidades que se esperan ver en la cola en cualquier tiempo durante el periodo de interés. En cualquier momento dado la cola podrá tener un número de entidades discretas. Sin embargo, debido a el el tiempo promedio de datos en la cola es un valor promedio, puede llevarnos a un valor que es fraccional.

Donde Q = el número en la cola por un periodo de tiempo dado.
dt = la longitud de tiempo en el que Q se observó.
T = longitud total de tiempo de simulación

where

Q = the number in the queue for a given length of time.

dt = the length of time that Q is observed.

T = the total length of time for the simulation.

Because the equation for the time-average number in queue is time dependent, further explanation is warranted. The equation essentially calculates the total entity-time in the queue that is observed during the simulation run divided by the total simulation run time. In an entity customer simulation, this would correspond to the customer waiting time by all of the customers that were waiting in line. Each period of time is calculated by multiplying the number of customers waiting in line for the amount of time that number of customers waited in line. A change in the number of customers waiting in line triggers the beginning of a new period of calculation. At the end, all of the periods with customer-minutes are totaled and divided by the length of the simulation.

Manual calculations of the time-average number in queue are best handled by drawing a two-axis graph of the system. The vertical axis records the number of entities in the queue. The horizontal axis records the simulation time. A line is drawn at the number of entities in the queue for the length of time that number of entities is in the queue. The entity time is calculated by calculating the area of each box, which is the number of entities waiting multiplied by the ending time for that number of entities in the queue minus the starting time for that number of entities in the queue. The total area is calculated by summing all of the individual areas. The time average number in queue is then calculated by dividing the total area by the length of the simulation run. This method is illustrated in Figure 1.2.

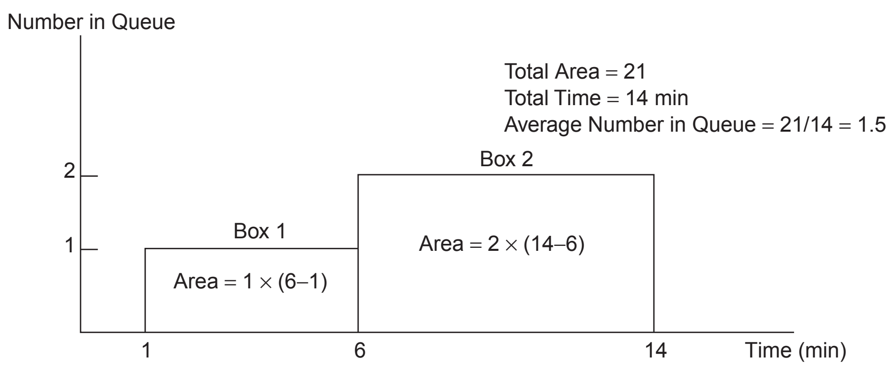

FIGURE 1.2 Resource utilization calculations.

1.9.3.4 Resource Utilization

Resource utilization is also a time-dependent statistic. At any given time a single resource can be either idle or busy. The idle state corresponds to a resource utilization level of 0. Naturally, the busy state corresponds to a resource utilization level of 1. The length of time that the resource is either at a 0 level or a 1 level is a function of the entities that come into the system. The formula for average resource utilization is

$$\text{Average Resource Utilization} = \frac{\int_0^T B\,dt}{T}$$

where

B = either 0 for idle or 1 for busy.

dt = the length of time that B is observed.

T = the total length of time for the simulation.

As with the time-average number in queue, we are summing the length of time that the resource is either busy or idle and then dividing by the total time of the simulation run. The average utilization rate can be calculated using a bar chart in the same manner as for the time-average number in queue. The only difference is that the vertical axis of the graph can take only a value of either 0 or 1.

1.9.3.5 Manual Simulation Example

The following interarrival and service times were observed in a single-server, single-queue system:

Interarrival times (min)	1, 4, 2, 1, 8, 2, 4, 3
Service times (min)	2, 5, 4, 1, 3, 2, 1, 3

Calculate summary statistics for the time-average number in queue, average system time, and average utilization based on 20 min.

1.9.3.6 Example Solution

It is best to begin by organizing our data in a chart with headings for the number arrival, arrival time, begin-service time, end-service time, and total system time. We can then populate the chart with our input data:

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2

The first event is the first arrival, which occurs at 1 min after the system starts. Because there is no one else in the queue, and the server is idle, the customer can immediately seize the resource and begin the service time. This means that the service time also begins 1 min after the system starts. The service start is the second system event. The service time for the first customer was 2 min. This means that the service end occurs at 3 min. The service end is the third system event. Note also that the total time that the customer was in the system is the service end or departure time of 3 min minus the arrival time of 1 min.

The fourth event that occurs is the arrival of the second entity. Because there is no one in the queue, and the server is idle, this entity also goes directly to the server.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5		

The second entity is scheduled to have a service time of 5 min. This means that the second entity's service time would end at 10 min. However, the interarrival time between the second and third entities is 2 min. This results in the arrival of the third entity at 7 min. Thus, the next event is not the end of the service of the second entity but the arrival in the system of the third entity. Because the server is busy with the second entity, the third entity takes the first position in the queue.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5		
3	7			

At this point two different things could occur. The service time end for entity 2 could occur at 10 min, or another entity could arrive in the system before 10 min. As it turns out, the interarrival time between the third and fourth entities is 1 min. This means that the fourth entity arrives at 8 min. This is before the service time ends for the second entity. The fourth entity enters the queue behind the third entity. There are now two entities in the queue.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5		
3	7			
4	8			

Now two other events could occur. The service time for entity 2 could end at 10 min, or the fifth entity could arrive in the system. As it turns out, the interarrival time for the fifth entity is 8 min. This means that the service end for the second entity will be the next event.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10		
4	8			

We can calculate the system time for the second entity in the same manner as the first entity. Because the second entity did not wait in the queue before being serviced, its system time is the same as its service time. At 10 min a second event occurs. Because entity 3 was waiting in the queue, as soon as entity 2 was finished, entity 3 immediately seized the resource at 10 min. With entity 3 now being served, only entity 4 is now waiting in the queue.

Entity 3 has a service time of 4 min. This means that its service time will be completed at 14 min. Because the interarrival time for entity 5 was 8 min, the service time for entity 3 will be completed before entity 5 arrives at 16 min.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10	14	7
4	8	14		

The system time calculation for the third entity is slightly different. Entity 3 actually waited in the queue for 3 min. So in this case, the system time is 14 min minus 7 min, for a system time of 7 min. Because entity 4 was waiting in the queue when entity 3 finished, it immediately left the queue and seized the server. The queue now has no entities.

The next event turns out to be the end of the service time for entity 4. This is because entity four has a short service time of only 1 min. Entity 4 also waited in the queue. The system time for entity four is 15 min minus 8 or 7 min.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10	14	7
4	8	14	15	7

Because the server is now idle at 15 min, the only event that can occur next is an arrival. The next event is the arrival of entity 5 at 16 min. No one is in the queue, so entity 5 immediately seizes the resource at 16 min.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10	14	7
4	8	14	15	7
5	16	16		

At 16 min, two different events could occur. The service time for entity 5 could end, or entity 6 could arrive. The interarrival time for entity 6 is 2 min. The service time for entity 5 is 3 min. This means that the next event is the arrival of entity 6 at 18 min. The server is busy with entity 5, so entity 6 enters the queue.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10	14	7
4	8	14	15	7
5	16	16		
6	18			

The interarrival time for entity 7 is 4 min for an arrival at 22 min, whereas the service end for entity 6 is 19 min. Because no more entities can arrive before 22 min, the next event is the service end for entity 6. The system time can be calculated for entity 5 as 3 min. Because entity 6 is waiting in the queue, it can immediately seize the resource at 19 min.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10	14	7
4	8	14	15	7
5	16	16	19	3
6	18	19		

The next event can be either the arrival of entity 7 or the service end for entity 6. The interarrival time for entity 7 is 4 min for an arrival at 22 min. The service time of entity 6 is 2 min. This would make the next event be the end of service for entity 6 at 21 min.

We actually specified in the model that we wanted to calculate the summary statistics based on 20 min. This means that we do not need to worry about the arrival of entity 7 at 22 min. We also do not need to worry about the end of service for entity 6 at 21 min.

Arrival num.	Arrival time	Begin sv. time	End sv. time	System time
1	1	1	3	2
2	5	5	10	5
3	7	10	14	7
4	8	14	15	7
5	16	16	19	3
6	18	19	21	
7	22			

Our event list is now complete, and we can turn our attention to calculating the output measures of performance. The easiest measure of performance to calculate is the system time of the entities. At time

20, only five different customers have exited the system. We do not need to be concerned about any entities still in the system when we are calculating an observational type of measure such as system time. We can calculate the average system time by summing all of the individual system times and dividing by 5. Thus, the average system time for all of the entities that were processed through the system was 4.8 min.

$$\text{Average System Time} = \frac{2+5+7+7+3}{5} = 4.8$$

To calculate the time-dependent statistics, time-average number in queue and average resource utilization, it is best to resort to the use of a chart. However, we also include a textual description of the relevant events.

1.9.3.6.1 Time Average Number in Queue Calculations

For entity 1 at 1 min and entity 2 at 5 min, service is available immediately. When entity 3 arrives at 7 min, it must wait in line. Entity 3 waits in line until entity 4 arrives at 8 min. This means that for 1 min between 7 and 8 min there was one entity in line:

$$1 \text{ entity in the queue} \times (8 - 7) \text{ min} = 1 \text{ entity-min}$$

The next relevant event is the end of service of entity 2 at 10 min. This means that two entities, entities 3 and 4, waited in the queue for the period between 8 and 10 min:

$$2 \text{ entities in the queue} \times (10 - 8) \text{ min} = 4 \text{ entity-min}$$

At 10 min there is only one entity, entity 4, in the queue. The next relevant event is the service end of entity 3 at 14 min. This means that entity 4 waited in the queue alone between 10 and 14 min:

$$1 \text{ entity in the queue} \times (14 - 10) \text{ min} = 4 \text{ entity-min}$$

At 14 min there is no one waiting in the queue. The next event is the arrival of entity 5 at 16 min. Because there is no one in the queue, and the server is idle, entity 5 does not have any queue time. The next event that occurs is the arrival of entity 6 at 18 min. The server is busy, so entity 6 enters the queue. The next relevant event is the end of service time for entity 5 at 19 min. This event causes entity 6 to leave the queue. Entity 6 was the single entity in the queue for 1 min:

$$1 \text{ entity in the queue} \times (19 - 18) \text{ min} = 1 \text{ entity-min}$$

When entity 6 leaves the queue, there are no other entities in the queue. The next entity does not arrive until 22 min. This means that for the remainder of the simulation, up to 20 min, the queue is 0.

The average time in queue can now be calculated by summing the periods of time that entities were waiting in the queue and dividing by the simulation time:

$$\text{Time Average Number in Q} = \frac{1+4+4+1}{20} = 0.50$$

This means that the average number of entities in the queue at any given time would be one half of an entity. Obviously, this is a mathematical representation because it is impossible to see half of an entity. It does indicate that the system is not very stressed if there is only half a person waiting in the queue at any given moment. The graphic representation of the time average number in queue is illustrated in Figure 1.3.

1.9.3.6.2 Average Resource Utilization Calculations

The average utilization calculations are somewhat easier to compute because the single resource can be only idle or busy (see Figure 1.4). The relevant resource utilization events begin at 1 min. This is when

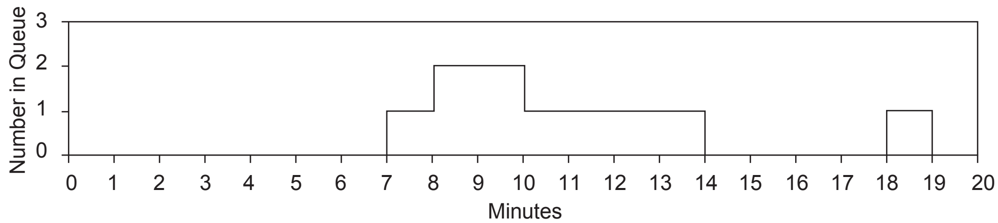

FIGURE 1.3 Average number in queue calculations.

entity 1 arrives and finds the queue empty and the server idle. So between 0 and 1 min, the utilization rate is 0:

$$0 \text{ resources busy} \times (1 - 0) \text{ min}$$

The resource remains busy until the end of the service time for entity 1. This occurs at 3 min. This area is calculated by:

$$1 \text{ resource busy} \times (3 - 1) \text{ min} = 2 \text{ resource-min}$$

At 3 min, there are no entities waiting in the queue. The next resource event occurs at 5 min, when entity 2 arrives to an empty queue and an idle server:

$$0 \text{ resources busy} \times (5 - 3) \text{ min} = 0 \text{ resource-min}$$

The server stays busy with entity 2 until the end of entity 2's service time at 10 min:

$$1 \text{ resource busy} \times (10 - 5) \text{ min} = 5 \text{ resource-min}$$

At the end of entity 2's service time, entity 3 is already waiting in the queue. Entity 3 immediately seizes the resource for its service time of 4 min, until 14 min into the simulation:

$$1 \text{ resource busy} \times (14 - 10) \text{ min} = 4 \text{ resource-min}$$

At the end of entity 3's service time, entity 4 is waiting in line. Entity 4 immediately seizes the resource for its service time of 1 min, until 15 min into the simulation:

$$1 \text{ resource busy} \times (15 - 14) \text{ min} = 1 \text{ resource-min}$$

At 15 min, there are no entities waiting in the queue. The next entity arrives at 16 min:

$$0 \text{ resources busy} \times (16 - 15) \text{ min} = 0 \text{ resource-min}$$

At 16 min, entity 5 arrives and immediately seizes the resource. Entity 5 uses the resource for 3 min, until 19 min into the simulation:

$$1 \text{ resource busy} \times (19 - 16) \text{ min} = 3 \text{ resource-min}$$

At 19 min, entity 6 is already waiting in the queue. It immediately seizes the resource at 19 min and keeps it past our cutoff time of 20 min. The average utilization of the single resource can be calculated to be an 80% utilization rate:

$$\text{Average Resource Utilization} = \frac{2+5+4+1+3+1}{20} = 0.80$$

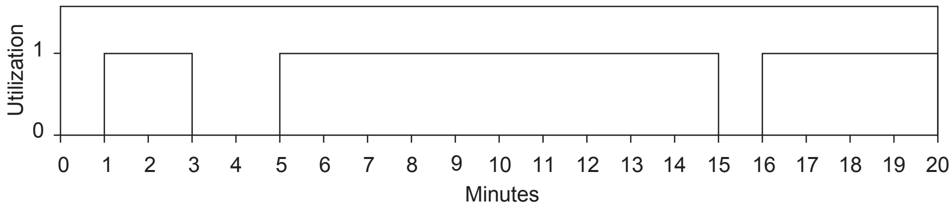

FIGURE 1.4 Average utilization.

It should be obvious to the practitioner that event lists involving anything more than a single queue and single server system run for more than a short period of time can become quite complex. The individual calculations themselves are not particularly cumbersome; however, there are a large number of events that could be happening nearly simultaneously. The consequences of a single sequencing or calculation error anywhere in the list could completely skew the output measures of performance. The advantages of utilizing a simulation-specific software package that automatically handles the event list should be readily apparent at this point.

1.10 Additional Basic Simulation Issues

Occasionally practitioners have difficulty understanding how even simple simulation models and event lists work. Sometimes this is because the practitioner has had significant experience in general-purpose programming languages. This is particularly noticeable when the practitioner's programming skills were developed using procedural or structured programming with languages such as FORTRAN, and the difficulty results from the experienced programmer's tendency to think about the program from an external viewpoint. Practitioners who become involved in simulation modeling need to look at the simulation model from the viewpoint of the entity. For example, the entity arrives in the system, the entity enters the queue, the entity seizes the resource, the entity releases the resource, and the entity is disposed of. As the practitioner develops more sophisticated and larger models, this issue becomes increasingly critical.

1.11 Summary

The objective of this chapter was to provide the novice and rusty practitioner with a beginning point from which to proceed with the simulation project. In this chapter we discussed the purposes of simulation, the advantages and disadvantages of simulation, and basic simulation concepts. In addition, we provided a few humorous quotations with what we hope was an educational message.

The section on basic simulation concepts included a step-by-step manual simulation model to provide experience with handling the event list and calculating summary statistics. Although the small example may have little actual operational value, it clearly demonstrates the complexity of handling the event list and calculating even a few summary statistics. Fortunately, most simulation projects will use a simulation software package that will completely insulate the practitioner from manually maintaining the event list or calculating any summary statistics. The practitioner can always refer to this chapter when questions arise with the reams of statistics that can be automatically generated by most simulation software packages.

Chapter Problems

1. What is the difference in use between traditional simulation models and training simulators?

2. Name a few typical manufacturing applications that can be modeled and analyzed using simulation.

3. Name a few typical service applications that can be modeled and analyzed using simulation.

4. Why might some systems be able to be analyzed only by using simulation?

5. What components does a simple system such as a one-chair barbershop contain?

6. What is the difference between observational and time-dependent data?

7. What does time-average number in queue mean?

8. What is the difference between an entity attribute and a global variable?

9. What is the difference between a discrete event simulation model and a continuous event simulation model?

10. Generate a manual event list for customers arriving at a single-queue, single-server system. Calculate system time, average number in queue, and resource utilization based on the system for 18 min.

    Interarrival times in minutes for 10 arrivals: 2, 1, 3, 1, 3, 2, 4, 2, 1, 1

    Service times in minutes for 10 arrivals: 2, 3, 1, 3, 2, 2, 1, 3, 2, 2

References

Kelton, D.W., Sadowski, R.P., and Sadowski, D.A. (2002), *Simulation with Arena,* 2nd ed., McGraw-Hill, New York.

Law, A.M. and Kelton, D.W. (2000), *Simulation Modeling and Analysis,* 3rd ed., McGraw-Hill, New York.

2

Problem Formulation

"We solved the wrong problem!"

2.1 Introduction

The first step in conducting a significant simulation project is to ensure that adequate attention has been directed toward understanding what is to be accomplished by performing the study. These activities are known as the problem formulation process and consist of:

- A formal problem statement
- Orientation of the system
- Establishment of specific project objectives

The importance of directing adequate attention toward the problem formulation process cannot be overemphasized. During this period, the simulation practitioner can firmly establish the practicality of using simulation to analyze the system. It may turn out that the system is deterministic or otherwise will not lend itself to simulation analysis-type approaches. If either of these situations exists, the practitioners can save themselves a significant amount of work and embarrassment early in the process. In the more favorable event that the system lends itself to simulation analysis, the practitioner should begin by developing a problem statement.

2.2 Formal Problem Statement

The problem statement component of the problem formulation step should provide both the practitioner and the potential audience with a clearly understandable high-level justification for the simulation.

Although each situation will be unique, problem statements commonly include text related to: