1

Simulation

1.1. Introduction

Simulation aims to imitate the behavior of real systems on a computer equipped with appropriate software. It is generally used to analyze systems and make operational or resource policy decisions. Nowadays, simulation is a tool that is becoming increasingly popular, as computers and software become increasingly efficient (Chung 2003; Fleury et al. 2006; Altiok and Melamed 2007; Kelton et al. 2015; Polenghi et al. 2018).

The study of a system leads in most cases to modeling its functioning through the establishment of mathematical or logical relationships. It is possible to use mathematical methods if these relationships are simple enough. This solution is then called an analytical solution. However, systems in reality are most often too complex to be able to apply such an evaluation. We then resort to simulation in order to estimate the desired characteristics of the model.

Let us take the example of an industrial company which seeks to expand without being sure of the potential gain generated by this expansion. Indeed, it would not be profitable for the company to invest money for an extension, then later remove this extension, if the latter is deemed unprofitable. A simulation study could enlighten decision-makers on the consequences of this extension by simulating the operation of the factory before and after extension. We then achieve results without making physical changes in the factory.

Simulation is a process which therefore consists of designing a model of a real system, carrying out experiments on this model, interpreting the observations provided by the execution of the simulation of the model and formulating decisions relating to the system. The goal of this process can be to compare different configurations of the system studied or to evaluate different strategies for its control in order to optimize its performances.

The fields of application of simulation are numerous and varied. A non-exhaustive list of problems for which simulation has proven to be a useful and powerful tool includes:

– production flow systems (Addi Ait 2000):

- machining operations: simulations of machining operations may include processes involving manually or computer-controlled factory equipment for machining, turning, bending, cutting, and welding,

- assembly operations: assembly operations simulation can cover any type of assembly line or manufacturing operation that requires the assembly of multiple components into a single part,

– logistics flows and transport systems:

- material handling equipment: material handling simulations include analysis of cranes, forklifts and automated guided vehicles,

- warehousing: warehousing simulations may involve manual or automated storage and retrieval of raw materials or finished products;

– the production of services:

- hospitals and medical clinics: models of hospitals and medical clinics can be simulated to determine the number of rooms, nurses and doctors for a particular location,

- retail stores: retail stores may need to know how many checkouts to use,

- food or entertainment facilities: entertainment facilities, such as multi-theater movie theater complexes, may be interested in the number of ticket sellers, ticket checkers or concession stand attendants to employ,

- information technology: information technology models generally concern the number and type of networks or support resources to be made available,

- customer ordering systems: customer ordering systems may need to know how many customer order representatives are needed on duty.

1.2. Advantages of simulation

1.2.1. *Compressed time experimentation*

As the model is simulated on a computer, experimental simulations can be carried out in compressed time. This is a major advantage because some processes can take months or even years to complete. Long system processing times can make robust analysis difficult to achieve. With a computer model, the operation of long processes

can be simulated in seconds. Additionally, multiple repetitions of each simulation can easily be performed to increase the statistical reliability of the analysis (Kelton et al. 2015).

1.2.2. *Reduced analytical requirements*

Before the existence of digital simulation, we were forced to use other tools that were more analytically demanding. Even then, only simple systems involving probabilistic elements could be analyzed. More complex systems were strictly the domain of the mathematical researcher or operations research analyst. Furthermore, systems could only be analyzed with a static approach at a given point in time. On the other hand, the advent of simulation methodologies has made it possible to study systems dynamically in real time. Additionally, the development of simulation-specific software has eliminated many of the complicated basic calculations and programming requirements that might otherwise have been necessary. These reduced analytical requirements have made it possible to analyze many more different types of systems than before with a greater variety of experiments (Kelton et al. 2015; Polenghi et al. 2018).

1.2.3. *Easy to demonstrate models*

Most simulation software has the ability to dynamically animate the operation of the model. Animation is useful both for debugging the model and for verifying and demonstrating its operation. Animation-based debugging makes it easy to observe flaws in model logic. Using animation during a presentation can help establish the credibility of the model through the dynamic demonstration of how the system model handles different situations. Without the capability of animation, we would be limited to less effective textual and digital presentations (Kelton et al. 2015).

1.3. Disadvantages of simulation

Although simulation has many advantages, there are also some disadvantages. These drawbacks are not really associated directly with the modeling and analysis of a system, but rather with the expectations associated with simulation projects (Chung 2003; Fleury et al. 2006; Melamed and Rutgers 2007). These disadvantages are as follows:

– Under no circumstances can simulation give accurate results when the input data are inaccurate: no matter how good a model is, if its input data are not accurate, we cannot reasonably expect to obtain accurate output data. Unfortunately, the data collection phase is considered the most difficult part of the simulation process.

Nevertheless, it is common that little time is allocated to this phase. In many cases of simulations, historical data of questionable quality have been accepted in order to save input data collection time. Too often, the exact nature or conditions under which these data are collected are unknown.

– Simulation alone cannot solve problems: some managers may believe that carrying out a model simulation project is enough to solve the problem they are facing. However, simulation by itself cannot actually solve the problem. It can provide direction for potential solutions to resolve the problem and it is up to the manager to implement the proposed changes.

1.4. Concepts relating to simulation

1.4.1. *System concept*

Different definitions have been given to the word "system" in the field of simulation (Kelton et al. 2015). Here are some examples:

– a system is a set of components linked together;

– a system is an organized set of functional elements;

– a system is a combination of parts that coordinate to achieve a result so as to form a set;

– a system is a combination of people, machines, materials and information intended to satisfy a given objective.

All these definitions have in common a few key words which characterize the notion of system. Indeed, a system is characterized by ordered parts which compose it. Each of these parts has its own laws and a certain independence. These parts also have links or relationships between them. Furthermore, the whole thing, or the system, changes over time and is influenced by the environment in which it exists and which reacts on it. Finally, this set is most often subject to constraints and only exists to achieve a goal.

In summary, a system can be defined by the knowledge of its parts or its components, the laws specific to each component and the interactions which determine its purpose.

Example: in a production plant, the parts that make up the system could be the workforce and departments of purchasing and supply, inventory management, manufacturing and production scheduling, sales and administrative. Each of these services has its own operating laws and is partially independent of the others. But also, they interact with each other. Knowledge of these interactions and their

application to the system will significantly influence the goal to be achieved by the system such as profitability, investment, profit, etc.

1.4.2. *Model and modeling*

A model is a representation of a system whose goal is to explain and/or predict certain aspects of its behavior. This representation is more or less faithful, because, on the one hand, the model must be sufficiently complete in order to be able to answer the various questions that can be asked about the system it represents and, on the other hand, it must not be too complex to be easily manipulated (Kelton et al. 2015).

When modeling a system, any modeler must ensure that the boundaries of their model are clearly defined through the internal variables, inputs and necessary outputs. In addition, they must judge the level of detail to incorporate into the model.

The two methods of progressive refinement and submodel accumulation are commonly used for modeling. Progressive refinement proceeds through iterations towards increasingly increasing levels of complexity. This evolutionary process makes it possible to take into account the inadequacies of the model at a given stage and to improve it at the next stage. The accumulation of submodels is used when the system to be studied is large. In this case, the system is divided into distinct subsystems in order to control the complexity of the system. All problems can manifest themselves in the integration stage, which leads to often delicate interface problems.

There are several classifications of simulation models. However, the most used consists of classifying them according to three dimensions, which are as follows:

– Static or dynamic: static models have no interaction with time, which plays an essential role in dynamic models.

– Continuous or discrete: the state of the continuous system can change continuously over time, such as the level of a reservoir as water enters and exits. In contrast, state changes in a discrete model can only occur at specific times. This is the case of a manufacturing system where parts arrive and leave at specific times and where machines stop and start again at specific times. It is possible that the same model has both continuous and discrete elements of change. This model is then called a mixed continuous-discrete model; for example, the case of a system for extruding PVC evacuation tubes and cutting them into units of four meters in length.

– Deterministic or stochastic: models with constant inputs and variables are considered deterministic. On the other hand, stochastic models work with at least some random inputs.

– Terminating or non-terminating: terminating and non-terminating systems are distinguished by:

- initial starting conditions: terminating systems generally begin each time period without any influence from the previous period. Many systems that use client-type inputs are considered terminating-type systems. A bank, for example, does not let its customers stay overnight on its premises. This means that every new day or period of time, the system starts empty. The non-terminating system, on the other hand, can start with entries already present in the system from the previous period;

- the existence of a natural termination event: this may be the shutdown of the system at a specific time or the end of a period of activity which is of particular interest. Stores and restaurants that typically close at the end of the day are terminating-type systems. In fact, the closure constitutes a termination event and all customers are forced to leave the store. So, in each new period, the system starts empty. Airline ticket counters can also be modeled as terminating systems. In this case, the period of interest may be the time between the arrival of the plane and its departure. Thus, the final event could be the departure of the plane. The non-existence of natural stopping events means that the system can be classified as non-terminating. Many manufacturing systems operate continuously 24/7. The only time they stop is for annual maintenance. Other manufacturing systems shut down after one or two shifts per day. However, these systems keep work in progress, waiting between shutdowns. Hospitals do not normally close, although activity may be reduced overnight.

1.5. Components of a simulation model

1.5.1. *Entities*

The first type of component of a simulation model corresponds to entities. These entities are the dynamic objects of the simulation: they move for a while, and then are eliminated when they leave the system. As they move, they change state, affect and are affected by other entities as well as the state of the system, and ultimately affect system performance metrics. In many cases, particularly those involving service systems, the entity may be a customer. In other cases, entities may be objects. In a factory, entities are components waiting to be processed (Altiok and Melamed 2007; Kelton et al. 2015).

1.5.1.1. *Entity batches*

The number of entities arriving in the system at the same time is called the batch size. In some systems, the batch size is always one. In others, entities may arrive in

groups of different sizes. For example, families who go to the cinema arrive in batches. Batch sizes can be two, three, four or more.

1.5.1.2. *Entity inter-arrival times*

The time between batch arrivals is called the inter-arrival time. It does not matter whether the batch size is one or more. The previous batch may have included only one entity, while the next batch may have several. The inter-arrival time is also the reciprocal of the arrival rate. When collecting the arrival of entities, it is generally easier to collect the inter-arrival time of batches. However, some historical data may be in the form of arrival rates. The inter-arrival time is considered as input data that must be provided to the model (Addi Ait 2000).

1.5.2. *Attributes*

Attributes are common characteristics that are assigned to entities. However, they may have specific values which may differ from one entity to another. It is up to the modeler to determine which attributes entities need, name them, assign values to them, modify them if necessary and then use them when appropriate (Altiok and Melamed 2007; Kelton et al. 2015).

Even if the entity attribute has the same name, it can have as many different values as there are entities. Let us take the example of the "ArriveTime" attribute which relates to the arrival time of the entity. This attribute stores the arrival time of the entity in the simulation system. Thus, each entity will have a unique value in its "ArriveTime" attribute. Some entity attributes may have the same value. Indeed, in the case of airline passengers, the "PassengerType" attribute could contain a value corresponding to the type of passenger. So, a value of 1 in "PassengerType" could represent a first class passenger and a value of 2 could represent an economy class passenger. In the simulation model, this attribute would be used to prioritize passenger boarding. So, for example, 30% of the entities in a simulation could have a value of 1 for the "PassengerType" attribute, and the remaining 70% would have a value of 2.

1.5.3. *Variables*

Unlike the attribute, a variable does not relate to an entity but rather to a characteristic of the system. You can have many variables in a simulation model, but each one must be unique. Two types of variables can coexist in a simulation model. The first type concerns predefined variables built into the simulation software such as the number of entities in the queue, the number of busy servers, the current time of the simulation clock, etc. The second type relates to user-defined variables such

as average service time, travel time, etc. These variables are accessible to all entities, and many of them can be modified by any entity.

Variables are used for a wide variety of purposes. For example, the travel time between two stations in a model can be the same throughout the model; while a variable called "TransferTime" can be defined and set to the appropriate value, then used wherever it is necessary. Variables can also represent something that changes value during simulation, such as the number of parts in a certain area of the model. This variable will be incremented by one part when it enters this zone and decremented when a part leaves it (Altiok and Melamed 2007; Kelton et al. 2015).

1.5.4. *Resources*

Resources process or serve entities within the system. Here are some examples of resources:

– customer service representatives;

– hairdressers;

– counter agents;

– machines;

– ATMs.

In simple models, resources can have two states. They are then either free or busy. Resources are free when they are available for processing, but there are no waiting entities. Resources are busy when they are processing entities. In more complex models, resources may also be temporarily idle or down. Inactive resources are unavailable for scheduled work breaks, meals, preventive maintenance periods, etc. Down or failing resources correspond to broken machines or even inoperable equipment.

Resources take some processing time to serve entities, for example, the time to prepare an order and receive payment or to machine a part. Processing time is also frequently referred to as the processing time or service time. This time is considered input data that would normally be collected by observation or otherwise acquired (Altiok and Melamed 2007; Kelton et al. 2015).

1.5.5. *Waiting queues*

Entities typically wait in a queue until it is their turn to be processed. These queues are managed by priority rules. Queue priority means that the order of entities

in the queue can change depending on the priority scheme (Altiok and Melamed 2007; Kelton et al. 2015). There are many queue priority rules, including:

– first in first out (FIFO);

– last in first out (LIFO);

– shortest processing time (SPT);

– longest processing time (LPT);

– lowest value first (LVF);

– highest value first (HVF).

1.5.5.1. *FIFO, LIFO*

In most simple systems, entities are served based on a first-come, first-served rule (FIFO). In this rule, the first entity in the queue will be served before other arriving entities. This type of priority rule is most often encountered in service-type systems.

LIFO, another type of priority rule, is the direct opposite of FIFO. This means that whoever entered the queue last will be the first to be processed. The use of LIFO is nowhere near as common as that of FIFO. Most LIFO applications impose some sort of penalty where the least senior members of the queue face unwanted treatment.

1.5.5.2. *SPT, LPT*

Two other queue priority rules that we may encounter are SPT and LPT. The SPT priority rule can be used effectively when there is a service limit for the system. In this case, customers who have the shortest processing time are sent to the front of the queue and are processed first. The LPT priority rule means that customers with the most complex transactions are processed first. This priority rule is much less common in service systems. It is only encountered when the longer processing time presents a much greater economic benefit to the system.

1.5.5.3. *LVF, HVF*

LVF priorities are often used to model commercial passenger transportation systems. Passengers can be classified into first, second and third classes. The class value can then be 1, 2, and 3. The priority in the LVF queue makes the first class at the front of the queue, the second class in the middle and the third class at the end. Second-class passengers can only receive service if there are no first class customers in the queue. Similarly, third-class passengers can only be served if there are no first- or second-class passengers in the queue. Every time a first-class passenger

arrives in the system, they are automatically placed at the front of the queue. Similarly, any second-class passenger will automatically enter the queue behind any first-class passenger, but before any third-class passenger.

The final queue priority method is the HVF priority scheme. Here, each customer is assigned a value which rearranges the position of customers in the queue in descending order of that value. This type of priority in a queue can be used when some people are regular customers. In this situation, the service system may want to prioritize these customers.

It is possible that none of the queue priority or ranking criteria rules adequately model the type of queue priority that the actual system uses. In the case where the real system uses a complicated queue priority rule, most simulation languages offer the ability to program all the necessary calculations. These rules are generally called user-defined rules.

1.5.6. *Statistical accumulators and performance measures*

Users of simulation software are primarily interested in the performance of the system model. To ensure this, they must calculate some sort of output measure to possibly compare it to other alternative forms of the model (Altiok and Melamed 2007; Kelton et al. 2015).

To collect output performance measures during simulation, we must keep track of various intermediate variables in statistical accumulators as the simulation progresses. All these accumulators should be initialized to 0. When any event happens during the simulation, the concerned accumulator should be updated appropriately. Simulation software supports most statistical accumulators that may be collected. In some situations, it is up to modelers to define their own accumulators.

Several performance measures are used in a simulation. The most commonly used of these are:

– the total time in the system;

– the waiting time;

– the number of entities in the queue;

– the resource utilization.

1.5.6.1. *Total time in the system*

This involves measuring the time the entity spends in the system. This time begins when the entity arrives in the system and ends when it leaves the system. It is more common to use the mean flow time (MFT) for all entities. The mathematical representation of this average time is as follows:

$$\mathrm{MFT} = \frac{\sum_{i=1}^{n} T_i}{n} \qquad [1.1]$$

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

1.5.6.2. *Waiting time*

Waiting time is also an observational measure. It is similar to overhead, but it only takes into account the time the entity spends in the queue. Some people do not favor waiting time, because it is the most unacceptable time. Indeed, many customers are partially unsatisfied when the service time begins, even though the service time itself may be long. The formula for mean waiting time (MWT) is:

$$\mathrm{MWT} = \frac{\sum_{i=1}^{n} W_i}{n} \qquad [1.2]$$

where W_i is the waiting time for an individual entity (queue arrival time – service start time) and n is the number of entities that are processed in the queue.

1.5.6.3. *Number of entities in queue*

The number of entities in the queue is a time-dependent statistic. The average number in queue is the average number of entities that we would expect to see in the queue at any time during the considered period. At any time, the queue actually contains a discrete number of entities. However, since the average number in the queue is an average value, it will usually result in a number that has a fractional value. For lightly loaded queues, it is actually possible for the average number in the queue to be less than 1. The formula for calculating the average number in queue (ANQ) is the following:

$$\mathrm{ANQ} = \frac{\int_0^T Q\,dt}{T} \qquad [1.3]$$

where Q is the number of entities in the queue for a given duration, dt is the duration that Q is observed and T is the total duration of the simulation .

1.5.6.4. *Resource utilization*

Resource utilization is also a time-dependent statistic. At any time, a resource can be either idle or busy. The inactive state corresponds to a resource utilization level of 0 and, naturally, the busy state corresponds to a resource utilization level of 1. The duration during which the resource is either at level 0 either at level 1 is a function of the entities that enter the system. The formula for the average resource utilization (Util) is

$$\text{Util} = \frac{\int_0^T \text{Bdt}}{T} \qquad [1.4]$$

where B is either 0 for idle or 1 for busy, dt is the duration that B is observed and T is the total duration of the simulation.

1.5.7. *Event, activity and process*

The simulated system passes from one state to another during the simulation. The entities then undergo operations and move through the system. Resources, for their part, execute operations and change state. Basically, the entire simulation process is event-driven. An event is something that occurs at an instant in the simulated time and can modify attributes, variables or statistical accumulators (Kelton et al. 2015).

In a service system, customers may queue for different time durations or have different services or quantities of goods to purchase. Therefore, the model must at least include the following events:

– customer arrivals in the processing area;

– processing customer requests;

– payments for goods.

At each event, the objects, entities and resources concerned engage in operations. These operations which are initiated or terminated by events are called activities. Any activity is therefore limited at its beginning and its end by an event.

The grouping of events according to the chronological order in which they occur is a process. This also includes activities. An activity is then triggered by a given event and stopped by another event. A process is therefore a sequence of events ordered in time and which delimit the durations of several activities (Figure 1.1).

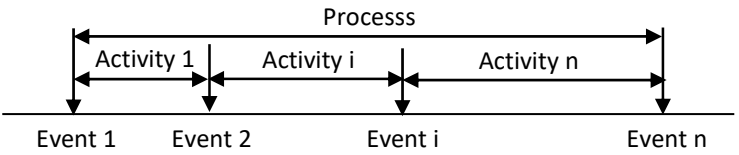

Figure 1.1. *Relationship between event, activity and process*

1.6. Basic principles of system representation

An organizational chart is an essential tool for defining the system. It then helps us to obtain a fundamental understanding of the logic of the system and graphically describes its main components and events. It also illustrates how input and output data play a role in the model.

1.6.1. *Standard flowchart symbols*

There are four basic symbols for flowchart processes which are as follows:

– oval;

– rectangle;

– inclined parallelogram;

– diamond.

The first guideline for using these symbols is that they should be arranged such that the sequence of processes flows down and to the right as much as possible. The second guideline is that a given symbol should have only one connection path to the other symbols. The only exception to this rule is the decision icon, which has two different exit paths. However, even with the decision icon, only one exit route can be taken at any given time. Finally, it is useful to try to keep the organization chart on as few sheets as possible.

1.6.1.1. *Start and stop oval*

The oval symbol is used to designate start and stop processes. The start symbol is the first symbol that must be present on the flowchart. The start symbol usually has only one output connector. The end symbol is normally the last symbol present on the flowchart. Even if the process has several possible paths, there should only be one stop or end symbol (Figure 1.2).

Figure 1.2. *Start and end symbols in a flowchart*

1.6.1.2. *Process rectangle*

The rectangle is used to designate general-purpose processes that are not specifically covered by one of the other flowchart symbols. The process rectangle is normally entered from the top or the left. Its exit can occur from the bottom or from the right side (Figure 1.3).

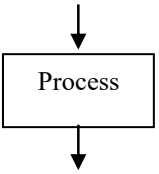

Figure 1.3. *Process symbol in a flowchart*

1.6.1.3. *Inclined input/output parallelogram*

The inclined parallelogram is used for processes that involve some sort of input or output. Entry into the inclined parallelogram is normally from the top or left side. Exit is from the bottom or right side. An example of an input process symbol in the flow diagram of a simulation program would be the creation or arrival of entities in the system (Figure 1.4).

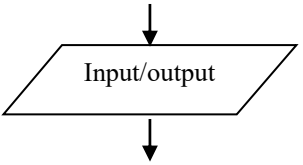

Figure 1.4. *Input/output symbol in a flowchart*

1.6.1.4. *Decision diamond*

The diamond is used to represent a decision in the logic of the organization. The diamond has one input connector but two output connectors. The single input connector should land on the top of the diamond symbol. The output connectors can

exit through one of the side vertices or through the bottom vertex. These output connectors should be specifically labeled as “true” or “false” or “yes” or “no”. As these responses are mutually exclusive, only one exit path can be taken at any given time (Figure 1.5).

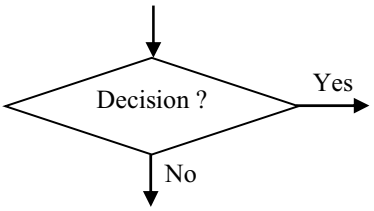

Figure 1.5. *Decision symbol in a flowchart*

1.6.2. *Flowchart example*

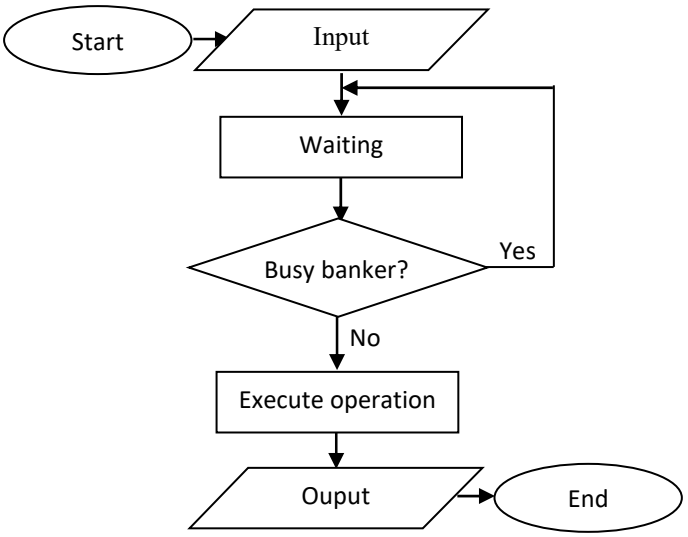

Figure 1.6. *Flowchart example*

Figure 1.6 is a single-queue, single-server system that represents the operation of a bank teller.

The system flow diagram begins with the starting oval. The following symbol is an inclined input/output parallelogram, which represents the customer entering the

system. Once customers enter the system, they are made to wait in the queue. Waiting in this queue is a process represented by a rectangle. The customer then determines whether the banker is busy or not. This operation is represented by a diamond. If the banker is free, the customer carries out the process of the operation represented by another rectangle. On the other hand, if the banker is busy, the customer remains in the queue. When the customer finally completes the operation, they exit the system. This operation is represented by another inclined parallelogram. The flow diagram then ends with an oval end symbol.

Notice how the flow of the process is oriented to the right and down as much as possible. The only exception where the flow is not downward is when the customer remains in the queue. It should also be noted that most symbols have only one input and one output. The only exceptions are starting with a single exit, stopping with a single entry, and decision. Although the decision has two exits, only one exit can be taken at a time. Either the banker is busy or they are not. Also note that when the banker is busy, the flow line does not return directly to the queue. Instead, the flow line intersects the other flow line that goes into the queue. This preserves the principle that the queue symbol can only be entered in one place.

1.7. Manual simulation

This is the same example of the bank teller discussed previously. The following inter-arrival and service times were observed in a single-server, single-queue system:

– inter-arrival times (minutes): 1, 2, 4, 2, 8, 2, 4, 3;

– service time (minutes): 2, 7, 4, 1, 3, 2, 1, 3.

Arrival number	Arriving time	Service start time	Service time	End of service time	Total time
1	1	1	2	3	2
2	3	3	7	10	7
3	7	10	4	14	7
4	9	14	1	15	6
5	17	17	3	20	3
6	19	20	2	22	3
7	23	–	–	–	–

Table 1.1. *Manual simulation of the example*

Let us try to calculate summary statistics for the average wait time, average total time and average utilization for a 20-minute simulation (Table 1.1).

The first event is the arrival of the first customer which occurs at t =1 minute. Since the queue is empty and the banker is inactive, this customer can immediately seize it to start the service. This means that the service time also starts at t = 1 minute.

The start of the service is the second event in the system. The service time for the first customer is two minutes; this means that the end of service occurs at t = 3 minutes. The end of this customer's service is the third event in the system. It should be noted that the total time this customer is in the system corresponds to the difference between their service end time at t = 3 minutes minus their arrival time at t = 1 minute which corresponds to a total duration of two minutes.

The fourth event that occurs is the arrival of the second customer at t = 3 minutes. The queue being empty and the banker being inactive, this customer goes directly to the counter. The second customer's service time is scheduled for seven minutes. This means that the final service time of the entity representing this customer is programmed at t = 10 minutes.

The inter-arrival time between the second and third customers is four minutes. This third customer therefore arrives at t = 7 minutes. Thus, the next event is not the end of the second customer's service, but the arrival of the third customer. However, as the banker is busy with the second customer, the third customer is put on hold at the first position in the queue.

The inter-arrival time between the third and fourth customers is two minutes. This means that the latter arrives at t = 9 minutes and enters the queue behind the third customer. There are now two customers in the queue.

Now, two other events could happen. The second customer's service time may end, or a fifth customer may arrive in the system. It turns out that the inter-arrival time of this fifth customer is two minutes. This means that the end of service for the second customer will be the following event at t = 10 minutes. It is then possible to calculate the overhead of the second customer in the same way as for the first. As the second customer did not wait in the queue before being served, their total time is the same as their service time, which is equal to seven minutes.

At t = 10 minutes, another event occurs. Indeed, as soon as the second customer has finished, the third customer immediately seizes the banker. With this customer's service now in progress, only the fourth customer is now on hold.

The third customer has a service time of four minutes. This means that its service time will be over at t = 14 minutes. And since the inter-arrival time of the fifth customer is eight minutes, the service time of the third customer will be finished before the arrival of the fifth customer, at t = 17 minutes. The overhead calculation for the third customer is slightly different from the first two. Indeed, in addition to their service time of four minutes, this customer waited in the queue for three minutes. In this case, the total time is therefore 14 – 7 = 7 minutes.

As the fourth customer was waiting in the queue when the third customer finished, he immediately leaves the queue and seizes the banker at t = 14 minutes. The queue is now empty.

The next event turns out to be the end of the fourth customer's service time. This customer has a short service time of only one minute. However, they also waited in the queue and their total time is 15 – 9 = 6 minutes.

The banker is now inactive at t =15 minutes, and the only event that can happen after that is another arrival. The next event is then the arrival of the fifth customer at t = 17 minutes. With no one in the queue, this customer immediately seizes the banker.

Subsequently, two different events could occur. The fifth customer's service time may end, or a new customer may arrive. The inter-arrival time of the sixth customer is two minutes, while the service time of the fifth customer is three minutes. This means that the next event is the arrival of the sixth customer at t = 19 minutes. As the banker is busy with the fifth customer, the sixth customer enters the queue.

The inter-arrival time of the seventh customer is four minutes for an arrival scheduled at t = 23 minutes, while the service of the fifth customer ends at t = 20 minutes. The next event is therefore the end of service for the fifth customer and the overhead can be calculated for this customer as 20 – 17 = 3 minutes. Additionally, as the sixth customer is waiting in the queue, they can immediately seize the banker at t = 20 minutes.

The next event can be either the arrival of the seventh customer or the end of service for the sixth customer. The inter-arrival time of the seventh customer is four minutes for an arrival at t = 23 minutes. The service time for the sixth customer is two minutes. This means that the next event would be the end of service for this customer at t = 22 minutes.

In fact, in the model, it was specified to calculate the summary statistics based on a simulation time of 20 minutes. This means that the arrival of the seventh customer

at t = 23 minutes as well as the end of service of the sixth customer at t = 22 minutes must not be taken into account.

1.7.1. *Calculation of average total time*

To calculate the average total time in the system, simply add up all the individual total times in the system and divide them by the number of entities processed during the simulation. This number of entities is equal to 5:

$$\text{MFT} = \frac{2+7+7+6+3}{5} = 5 \text{ minutes} \quad [1.5]$$

For all entities that have been processed by the system, this average total time is five minutes.

1.7.2. *Calculation of average waiting time*

Customers 1 and 2 arrive respectively at t = 1 and t = 3 minutes with a banker available immediately. No waiting can therefore be considered.

When customer 3 arrives at t = 7 minutes, they must wait in the queue until customer 4 arrives at t = 9 minutes. This means that for two minutes between seven and nine minutes, there was a customer in the queue:

1 customer in line × (9 – 7) minutes = 2 customers.minute

The next relevant event is the end of customer 2's service at t = 10 minutes. This means that both customers 3 and 4 waited in the queue for the period between nine and 10 minutes:

2 customers in queue × (10 – 9) minutes = 2 customers.minute

At t = 10 minutes, there is only customer 4 left in the queue. The next relevant event is the end of service of customer 3 at t = 14 minutes. This means that customer 4 waited alone in the queue between 10 and 14 minutes:

1 customer in queue × (14 – 10) minutes = 4 customers.minute

At t = 14 minutes, there is no one left in the queue. The next event is the arrival of customer 5 at t = 17 minutes. As the queue is empty and the banker is inactive, customer 5 does not have to wait. The next event that occurs is the arrival of customer 6 at t = 19 minutes. The banker being busy, this customer enters the queue. The next event is the end of customer 5's service time at t = 20 minutes, which is

also the end time of the simulation. This event causes client 6 to leave the queue. This was then the only customer in the queue for one minute:

1 customer in queue × (20 – 19) minutes = 1 customer.minute

The average time in queue (AWT), can now be calculated by adding the periods that entities waited in queue and dividing by the simulation time:

$$\text{AWT} = \frac{2+2+4+1}{20} = 0.45 \text{ minute} \quad [1.6]$$

This means that the average number of customers in the queue at any given time is almost half of a customer. Obviously, this is a mathematical representation, as it is impossible to see half of a customer. However, this indicates that the system is not very busy if there is only one "half a person" in the queue at any given time. The graphical representation of the time average of the number of customers in the queue is shown in Figure 1.7.

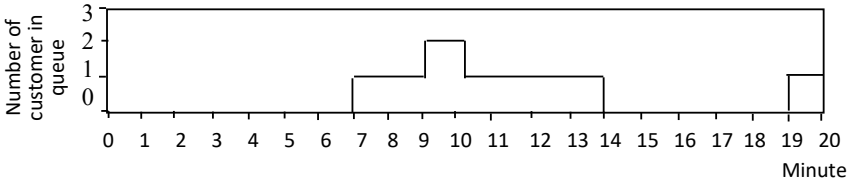

Figure 1.7. *Number of customers in queue*

1.7.3. *Calculation of average resource utilization*

Average resource utilization calculations are rather easier to perform because the single resource can only be idle or busy (Figure 1.8).

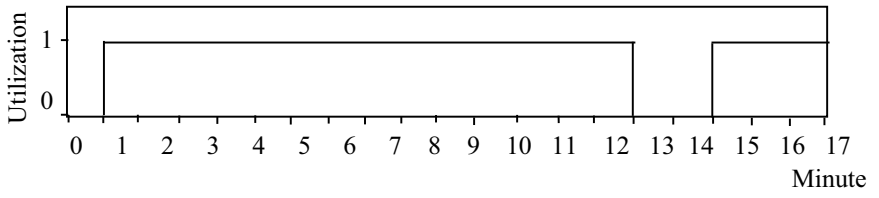

Figure 1.8. *Resource utilization*

Relevant resource utilization events start at t = 1 minute. This is the moment when customer 1 arrives and finds the queue empty and the banker idle. Thus, between zero and one minute, the utilization rate is 0:

0 resource busy × (1 – 0) minute = 0 resource.minute

The resource remains busy until customer 1's service time ends. This happens at t = 3 minutes. This area is calculated by:

1 resource busy × (3 – 1) minute = 2 resources.minute

At t = 3 minutes, customer 2 arrives and the banker is seized again until t = 10 minutes:

1 resource busy × (10– 3) minutes = 7 resources.minute

At the end of customer 2's service time, customer 3 is already waiting in the queue. Customer 3 immediately enters the resource for its four-minute service time, up to t = 14 minutes in the simulation:

1 resource busy × (14 – 10) minutes = 4 resources.minute

At the end of customer 3's service time, customer 4 waits in line. Customer 4 immediately grabs the resource for its one minute service time, up to t = 15 minutes in the simulation:

1 resource busy × (15 – 14) minutes = 1 resource.minute

At t =15 minutes, there are no more customers waiting in line. The next customer arrives at t = 17 minutes:

0 resource busy × (17–15) minutes = 0 resource.minute

At t = 17 minutes, customer 5 arrives and immediately grabs the resource and uses it for three minutes, until t = 20 minutes in the simulation:

1 resource busy × (20 – 17) minutes = 3 resources.minute

The average utilization of the single resource, "Util", can be calculated as follows:

$$\text{Util} = \frac{2+7+4+1+3}{20} = 0.85 \qquad [1.7]$$

The average resource utilization rate is 0.85. This resource is therefore used for 85% of the time it is available. Such an average can be considered high, which reflects the good exploitation of the resource.