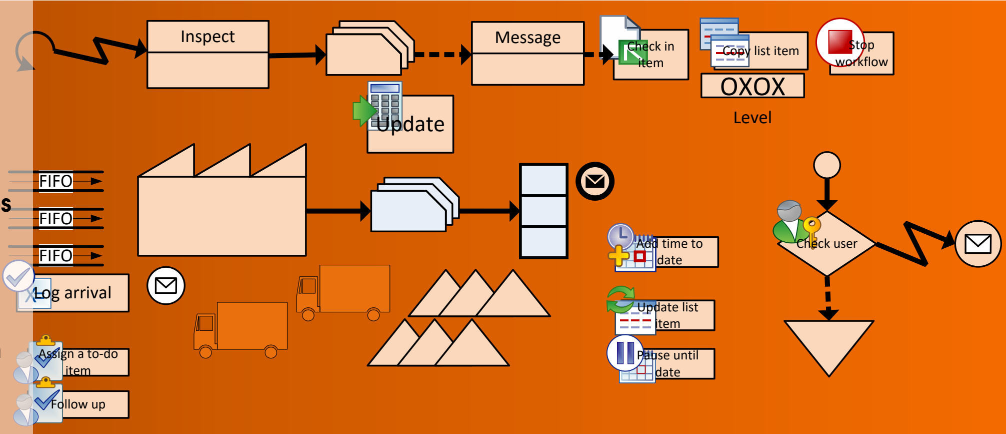
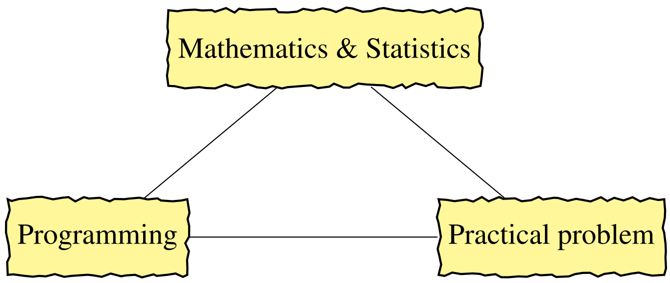
Basic concepts needed
Systems thinking
Modelling
Why simulate?
What is simulation
Typical application areas of simulation
Discrete and continuous random variables
Simulation in perspective
Advantages and drawbacks
Pitfalls of simulation studies
Other modelling tools
The discrete-event simulation mechanism
Simulation software packages
Animation in simulation



1. Introduction

Simulation is beautiful

SIMULATION is beautiful because it combines the application of some mathematical and statistical principles with computer programming to solve real-world problems. The following schematic shows the idea.



This set of short notes provides the essential content of the Simulation 442 module. The module content mainly covers the simulation of *discrete-event stochastic processes*. At the same time, special attention is paid to the basic statistical principles required for application in the simulation of discrete, stochastic processes. Using a simulation software package is a secondary aspect that is addressed, and *Tecnomatix Plant Simulation* will be introduced in this module. It is important to understand the statistical and other principles needed for discrete stochastic simulation – many people study the simulation software only, but this is insufficient to conduct a simulation study sensibly.

1. Overview, simulation in perspective, systems thinking, steps in a simulation study, concept models, basic statistics, demonstration and discussion of some models.
2. Modelling world views and imitating reality.
3. Analysis of input data for simulation models.
4. Analysis of output data for simulation models – terminating systems.
5. Analysis of output data for simulation models – non-terminating systems.
6. Techniques to select the best from a finite number of simulated scenarios.
7. Principles of simulation-optimisation.
8. Building computer models in Tecnomatix Plant Simulation followed by analysis of output data.

R Module survival hints: Study the notes, then do the tutorials with vigour. Always try to relate the tutorial questions to the theory in this book, and attend class. Attempt the tutorials during the week, and come to the tutorial classes to put the finishing touches to the tutorial submissions. This is an *applied* module.

1.1 Basic concepts needed

The following basic topics of Statistics should be well understood for the purpose of this module:

1. Sample space.
2. Probability and proportions.
3. Randomness, random numbers and random variables.
4. Distributions – Discrete (mass function) and continuous (density function).
5. Cumulative distribution functions.
6. Line graphs and histograms.
7. The mean, mode, median and variance of distributions.
8. The Poisson, exponential, Student t, normal and other distributions.
9. Parameters of distributions.
10. Hypothesis tests.
11. The Central Limit Theorem.
12. Statistical inference: point estimators and confidence intervals.

The student also needs a good understanding of queueing theory, which may require a revision of Chapter 20 in Winston (2003), while simulation is covered in Chapter 21 in Winston (2003). Further (not complete!) sources of simulation practice and theory are listed at the end of this book.

On completion of this module, the student should be sufficiently equipped to execute a simulation study with some senior guidance in practice. When some experience has been gained, the engineer should be able to conduct simulation studies independently. The primary aim is to explain the art and science of simulation analysis to the student, with applications developed and analysed with a specific simulation software package as the secondary aim. A thorough knowledge of descriptive and inferential statistics and hypothesis testing is recommended.

The significant simulation activities, which are input analysis, modelling, and output analysis, are explained. These can be subdivided into several chronological and concurrent steps, which are treated in detail. The steps are practically applied and illustrated with examples. The Tecnomatix Plant Simulation package is used in this module. The focus is mainly on stochastic, dynamic, discrete event simulation models. Before we look at simulation, we should briefly consider two fundamental topics: *Systems thinking* and *modelling*. Each topic can be studied independently, but we shall explore it briefly to understand the topic better.

1.2 Systems thinking

One of the most important skills of an industrial engineer is to be able to see the whole, as well as the hole (the doughnut principle). This is an art required for successful simulation – first to isolate a section of a whole for simulation purposes, and then to be able to describe this selected section at a sufficient level of detail (*resolution*). We must first understand what a system is, at least from a simulation point of view:

Definition 1. *A system is a* ***group*** *or* ***collection*** *of* ***interrelated objects*** *that* ***cooperate*** *to achieve one or more* ***objectives****.*

We could for example see the university as a system, and the objects could include *administration*, *finance*, *residences*, *maintenance* and the *academic* unit. This view comprises the bigger picture, but we can analyse for example the academic unit down to module codes, where we can have a lengthy discussion on the length of the module code, *i.e.* whether it must be five or six digits long.

Another example is a motorcar: The objects could be the *engine*, the *body*, the *support structure* and the *wheels*. They are interrelated/not interrelated in some ways, and these objects mentioned (usually) cooperate to transfer the system from one point to another. One or more human beings often accompany this system, so that they also get displaced. The motorcar could also be viewed as consisting of a *mechanical* object and an *electric/electronic* object. This viewpoint is also valid, which shows that there are many ways to view a system, and no view is correct or incorrect, but we must realise that some views are more applicable to simulation and/or the problem at hand than others.

More examples:

1. A manufacturing plant, comprising
    - people
    - machines
    - material
    - information
    - products

   It could also be viewed as consisting of
    - reception
    - processing
    - dispatch
2. A hospital with objects
    - theatres
    - wards
    - casualty
    - food preparation division
3. A law practice having objects
    - administration support
    - pending cases
    - completed cases
    - junior partners
    - senior partners

We can thus represent a system from different viewpoints. The industrial engineer must thus be able to identify the correct/applicable system for simulation purposes. Of course, what comprises a system for one person is only a subsystem or a super-system for someone else.

1.3 Modelling

To understand simulation, we also have to understand the idea of *modelling*. We may quote many dictionary definitions of modelling, but in simple terms we define modelling as follows:

Definition 2. *A model is the representation of an entity/object in a form other than the entity/object itself.*

We may thus model a house by making a simple drawing of it, as shown in Figure 1.1.

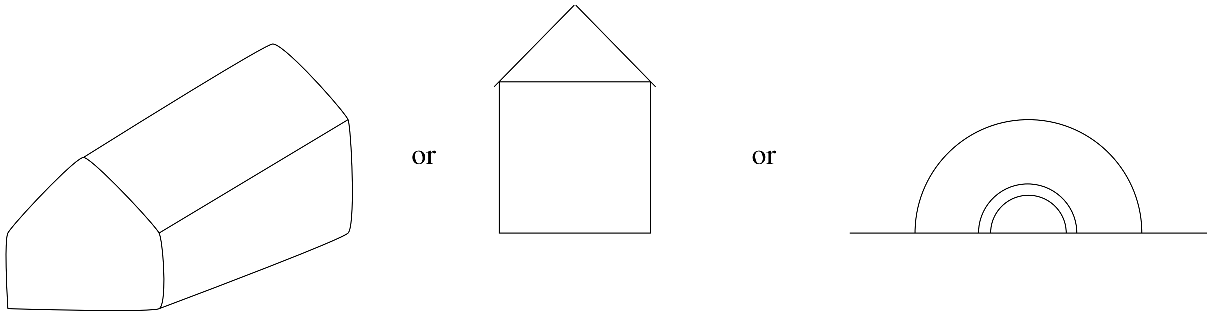


Figure 1.1: Schematics of house models

You would agree that these ‘models’ require some imagination to believe what they represent. A ‘better’ model could be one that we have built with a Lego set, which gives better views on layout and relative scales *etc*. Models like these are physical or iconic models, and are useful in many instances. This shows that a process of abstraction is present during modelling, and is also required by the person to whom the model is presented. We could also model other entities or systems with other approaches by following different viewpoints.

Typical viewpoints are
- Functional
- Logical
- Geographical
- Data flow.

Some types of models are
- Physical scale models: aeroplane models for wind tunnel tests, the human body for medical students
- Mathematical models: Linear programming, queueing theory, and economic models
- Social behaviour/psychological: Maslow’s five levels of human needs
- Ecological models: organism populations increase as a function of temperature.

Currently, four modelling paradigms are recognised. These are
1. Simulation, which is discussed in the next section.
2. System dynamics
3. Dynamic systems
4. Agent-based modelling, incorporating simulation.

These are summarized in Appendix B.

When we want to study a system, we can follow the map as defined by Law (2015), shown in Figure 1.2. We can experiment with the actual system, or develop a model of it, which we then experiment with. It is assumed that the behaviour of the model reflects the behaviour of the system it represents.

P “All models are false, but some models are useful". (George E.P. Box, famous statistician)

1.4 Why simulate?

We must ask ourselves “Why do we want to study a system?” A system study usually originates because of one or more problems that have been identified by those operating/managing the system. The manager may for example wish to increase throughput at a certain point in a process, and suppose they have several possible ways to achieve this, they will first have to do some analysis, starting with Figure 1 as guide. When a system is of such a complex nature that it

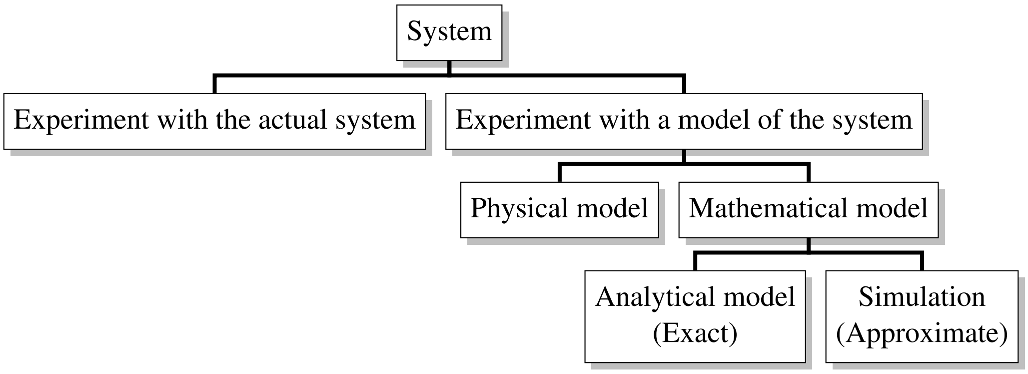


Figure 1.2: Ways to study a system (Law, 2015)

cannot be analysed analytically, simulation is strongly considered. In addition, if the system is of complex stochastic nature, then simulation is again indicated.

An example of a complex, stochastic process: Consider a simple queueing system with block arrivals from a finite population of size N, where the size of the block is a random variable X with a certain distribution. Assume that the system has four parallel service channels, and that there is only space for M entities in the system, while the queueing discipline is based on a general rule. It can be presented in Kendall-Lee notation (see Winston (2003)) as

$\mathrm{GI}^{[X]}|\mathrm{G}|4|\mathrm{GD}|M|N$.

Suppose we want to estimate the time an arrival spends in the common queue in front of the four service channels. There is no analytical solution to this queueing system, to the best of the author's knowledge. How practical is this system? Well, it represents the coal train arrival-service process at a South African coal terminal. Trains arrive with some variation of the time between arrivals, and each train consists of a different number of wagon sets (blocks). There are four service channels in the coal terminal that empty the wagons, and only 10 wagon sets can be hosted in the terminal, while the fleet of wagons is finite, *i.e.* 6 000 wagons. Each set consists of 50 wagons, so $N = 120$ in this case. The random variable X (the number of wagon sets in a train) could follow the distribution

$$f(x) = \begin{cases} 0.3 & x = 2 \\ 0.5 & x = 3 \\ 0.2 & x = 4. \end{cases}$$

A train thus consists of three wagon sets with a probability of 0.5. Since the sets are rotated (mine to coal terminal and back to mine) the population from which arrivals originate is finite. We see that a (practical) system can quickly become complex when we try to describe it analytically.

1.5 What is simulation

Simulation is a widely used term that is (over) utilised to cover virtually all perceptions on mock-ups of real/complex/non-existent entities. The industrial engineer views simulation as a computer model that yields the responses of a (usually) highly variable system, given a certain input data set. The economist simulates return on investment by applying different interest rates over different periods by way of example. A pilot exercising skills in a flight simulator is subjected to simulated situations.

The Handbook of Simulation (Banks, 1998) defines simulation as follows (p. 3):

Definition 3. *Simulation is the* ***imitation*** *of the* ***operation*** *of a* ***real-world process or system*** *over* ***time****.*

Simulation can thus be viewed as the experimentation with a model of a real-world system in order to study the behaviour of the model, given certain starting conditions. It is assumed that the behaviour of the model is a sufficient predictor of the real system's behaviour. This definition implies that we generally try to answer *what-if* questions with simulation, *i.e.* an existing or proposed system is modelled with a simulation software package and the behaviour of the system is studied when parameters are changed. Systems are thus analysed to find an acceptable solution for their operating parameters.

Simulation is very versatile, because it can be used in *manufacturing*, *agriculture*, *mining* and the *services* sector.

1.6 Typical application areas of simulation

The following is a list (incomplete) of some of the major fields where simulation is applied:

- Manufacturing (you guessed it!)
- Services (hospitals, business processes)
- Military
- Distributed systems
- Transport
- Computer and communication systems
- Physical sciences
- Aerospace systems
- Environmental, ecological studies
- Financial decision support systems
- Undersea systems
- Simulators
- Supply chains and logistics

1.7 Discrete and continuous random variables

A clear distinction should be made between *discrete* and *continuous* variables. A discrete variable represents countable things like number of products made, number of machine failures registered and number of hamburgers sold. The discrete variable can have finite or infinite limits on the integer numbers range. The continuous variables associates with time, distance, mass, and proportions like service level and utilisation. Discrete and continuous random variables consequently have different accompanying distributions. Some fundamental properties of a random variable, say X, are shown in Table 1.1.

R The reader should be aware of the various forms in which some probability density functions are presented – the Weibull p.d.f. has at least five forms, each presented within a certain context in the literature. The exponential distribution has the forms

$$f_X(x) = \lambda \exp^{-\lambda x}, \quad x > 0, \tag{1.1}$$

and

$$f_X(x) = \frac{1}{\beta} \exp^{-x/\beta}, \quad x > 0, \beta > 0 \tag{1.2}$$

and $\lambda = 1/\beta$. Thus, if $\lambda = 2$ arrivals/minute, then $\beta = 0.5$ minutes between arrivals.

Table 1.1: Discrete and continuous random variable properties

Discrete Probability *mass* function is $f_X(x)$.	**Continuous** Probability *density* function is $f_X(x)$.
$f_X(x) \geq 0, \quad x \in A$ (A is the definition range of X)	$f_X(x) \geq 0, \quad x \in \mathbb{R}$
$P(X = x) = f_X(x)$	$P(a < X < b) = \int_a^b f_X(x)\mathrm{d}x$
$\sum_{x \in A} f_X(x) = 1$	$\int_{-\infty}^{\infty} f_X(x)\mathrm{d}x = 1$
$F_X(x) = P(X \leq x) = \sum_{t \leq x} f_X(t)$	$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(t)\mathrm{d}t$

1.8 Simulation in perspective

The focus in this book is on *dynamic*, *stochastic*, *discrete event* simulation (DES) models. In dynamic, stochastic, discrete event simulation models, the state of the system being modelled changes at discrete and usually random-spaced points in time. A simulation model is characterised by specific features of the system that is being modelled. These features are *time dependency*, specific *characteristics of the variables* and the *simulation time increment*.

Time in this context means that the system or process being modelled *evolves over time*, for example the business processes of a retail shop: customers come and go, inventory is replenished, maintenance is being done, and more.

Deterministic variables assume exact values: there are 12 cashiers on duty in the retail shop, and the shop has 24 shelves.

A *stochastic variable* describes uncertainty, for example the number of customers that visit the shop per day. If the simulation time increment is *discrete*, it means that events are modelled that happen at discrete points in time, for example, when a customer arrives, it is considered an event. Other events are when the customer service starts and ends.

Figure 1.3 puts simulation in perspective within the mathematical modelling domain.

With this perspective in mind, one can consider the advantages and some drawbacks of simulation.

1.9 Advantages and drawbacks of simulation

Some advantages of simulation are:

- Analyse systems operations before they are implemented – avoiding and/or minimising cost
- Optimisation or sub-optimisation
- Tactical and strategic planning
- Changes to a system can be investigated without disrupting the operations of the system: New designs/acquisitions of resources can be investigated before the actual capital layout is made.
- Long (‘never ending’) processes are studied in a relatively short time.
- Critical parameters in a system can be identified and studied.
- Evaluation of alternatives.

It is true that simulation also has some drawbacks, some of which are:

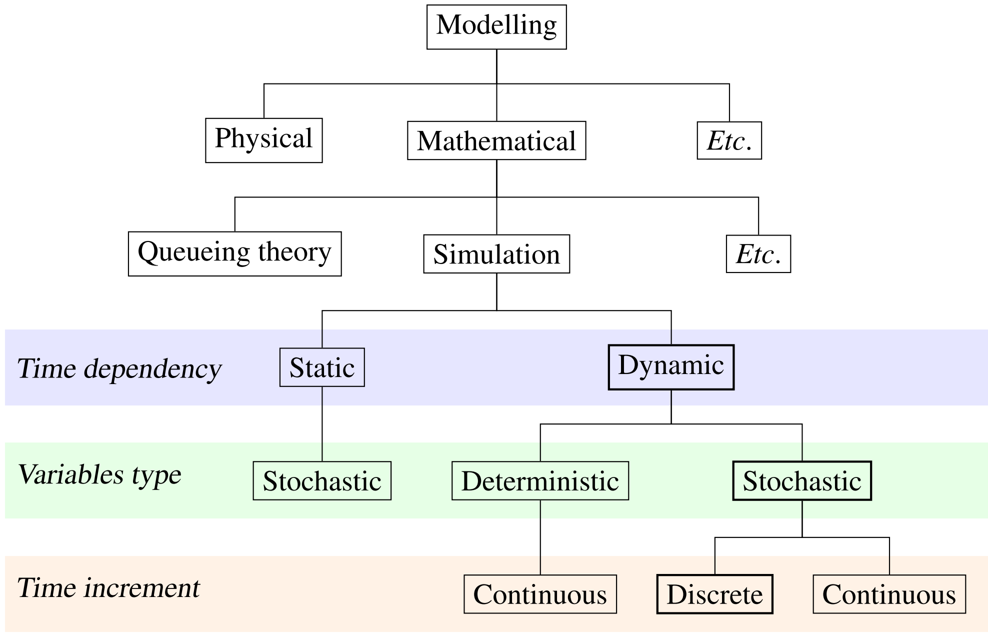


Figure 1.3: Simulation in perspective (Kruger and Du Preez (1988))

- Simulation studies can be *expensive* and *time-consuming* in many cases. (Software packages may become cheaper in some cases).
  - It is time-consuming – a simulation study takes time if properly executed.
  - A simulation study could be too costly for a given problem.
- It requires a certain level of expertise, although many software vendors claim that their products can be used by virtually anyone, not only bald professors with many years of experience. This is however the danger – anyone can use a simulation software package, but not everyone understands system dynamics and in many cases mathematical statistics. The software packages make analysis much easier today, but the author is still convinced that a person who understands the underlying concepts is required for simulation. A good simulation analyst needs good training and experience.
- Other tools may be more appropriate to analyse a particular problem, for example linear programming.
- The interpretation of simulation results requires a sound statistical background, regardless of what simulation software vendors tell you.

1.10 Pitfalls of simulation studies

Law (2015) discusses pitfalls to the successful completion of a simulation study. These are extremely important and some are quoted verbatim (Law (2015), pp. 71–72):

- "*Failure to have a well-defined set of objectives at the beginning of the simulation study.*
- *Misunderstanding of simulation by management.*
- *Treating a simulation study as if it were primarily an exercise in computer programming.*
- *Failure to have people with knowledge of simulation methodology and statistics on the modeling team.*
- *Failure to collect good system data.*
- *Misuse of animation.*

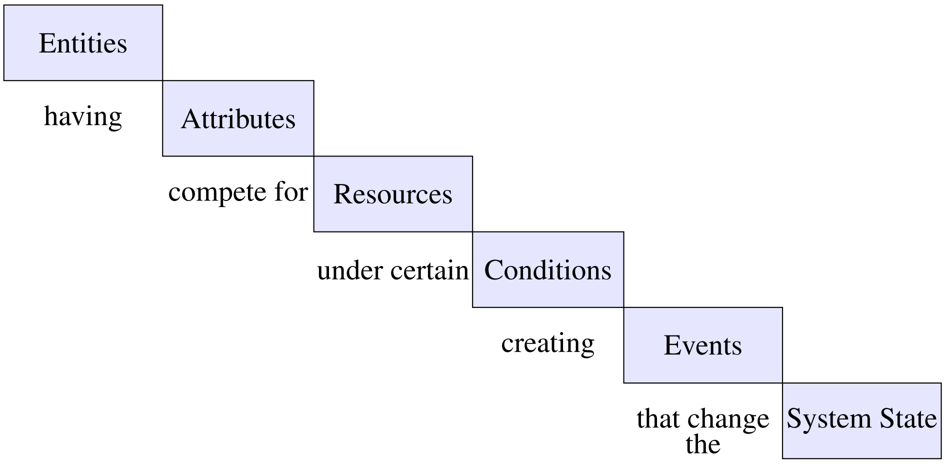


Figure 1.4: A typical simulation world view

- *Making a single replication of a particular system design and treating the output statistics as the "true answers".*
- *Comparing alternative system designs on the basis of one replication for each design."*

Also refer to Law (2003) and Sturrock (2018) for detailed discussions.

1.11 Other mathematical-based problem-solving alternatives

The industrial engineer is equipped with several problem-solving skills and tools. Although simulation is versatile and powerful (in the right hands), other mathematical-based problem solution tools can be used where appropriate, and some are:

- Queueing theory
- Linear programming
- Assignment algorithms
- Integer programming
- Graph theory
- Markov chains
- Stochastic inventory models
- Dynamic programming (deterministic & stochastic)

The problem must allow the tool to be used. For example, linear programming cannot be used if a problem is non-linear.

Next, the very important concept of the discrete-event mechanism is introduced.

1.12 The discrete-event simulation mechanism

We desire to model a complex real-world problem on a computer, which is fundamentally a discrete machine. How can that be done, especially in when a problem domain is continuous? To answer this, some important aspects regarding the *simulation mechanism* must be understood, namely the

World view: How is a real-world system conceptualized in a computer language? That is, what is the implicit view of the simulation software that we must follow to implement a real world system's behaviour? A typical world view is shown in Figure 1.4.

This is a model according to R. Shannon (1975), and it is the basic one we will often use in this course. The simulation modelling world view is a framework for defining a system in sufficient detail so that the behaviour of the system can be simulated (Pegden, R.E. Shannon, and R. Sadowski, 1995). A simulation model contains system state variables which must be changed over time, and the world view provides a set of rules for advancing simulation time and changing of state variables. Pegden, R.E. Shannon, and R. Sadowski (1995) describe three approaches to world views, these are: *event modelling*, *process modelling* and *object modelling*. The elements of Shannon's world view are discussed next. Also see Ingalls (2013) and White and Ingalls (2016).

Entities

Entities are objects that 'flow' through a process or system. They drive the business and are the reason for the existence of the process. Examples are:

1. Customers entering and leaving a bank or retail store.
2. Bottles in a bottling plant.
3. Containers in a ship yard.
4. Vehicles on an assembly line.

Consider a vehicle on an assembly line: it is modelled as an entity, but the subsystems making up the vehicle could also be modelled as entities. This means that several entities are merged into a final entity, so there is often more than one type of entity in a model.

Attributes

An attribute describes an entity, and is thus defines a property of the entity. It can be viewed as a local variable, as the attribute value flows with the entity through the process simulated. An entity can have attributes like colour, mass, volume, type and dimensions. Bottle entities in a bottling plant can be for example of *colour* green and *volume* 375 ml, while another type of bottle can have *colour* amber and *volume* 750 ml.

Resources

Entities compete for resources in the processes we typically simulate. Resources provide services to entities, and often transform entities. Resources can be manufacturing machines, cranes, people like receptionists, doctors, machine operators, managers, supervisors, and also moving units such as forklifts, trucks and carts.

Conditions

The conditions refer to the business rules built to be followed by a real-world process. Rules include

1. first-in-first-out when several entities compete for a resource
2. a task must be completed before an operator can go home at the end of a work day
3. priority tasks must first be processed by a resource.

When developing a simulation model, the analyst spends a lot of time understanding the business logic.

Events

In discrete-event simulation, events are 'holding points' in time. The simulation model stops at a scheduled point in time and make changes to entities, which consequently leads to model state changes. Events include arrival of entities in the model, departure of entities, an entity starting or completing its processing at a resource, a resource fails or is repaired, the simulation models starting and ending execution.

System state

The system state is described by a set of variables. The concept of system state variables can be explained as follows: suppose we observe the arrivals, processing and departure of customers at a single automated teller machine (ATM). What variables can we use to describe the process at the ATM? First, we draw a boundary around it, and we identify what 'drives' this system. In this case, it would be customers doing transactions at the ATM. So they enter the system by crossing the boundary drawn around it, and they exit the system by crossing the boundary again. Each customer also needs to occupy the ATM for some time, but this time is not known beforehand. Now we can say that 'the number of customers in the system' is a state variable, because it gives us a quantification of one property of the system. Our intuition tells us that if this variable has a high value, the system is 'busy', and if the number is small, the system is 'not busy'. Some other system state variables are

1. Number of customers waiting in the queue.
2. Number of customers served up to time T.
3. The utilisation of the ATM.
4. The average/minimum/maximum time a customer spends in the system.
5. The total amount of cash provided by the ATM at time = T.

A simple trick to identify system variables is to imagine that you sit back and observe the system, then take a photo of its operations. What do you count? What do you see the resources doing? Are they idle/busy/failed/on leave/maintained? The choice of system state variables depends on the analyst and the purpose of the simulation study. Usually, the performance measures, which are a subset of the system state variables, are the only state variables mentioned by the analyst.

Transaction-flow world view

The transaction-flow world view is explained by Schriber, Brunner, and Smith (2016). A very important principle to understand from this approach is that an entity proceeds through a model until it encounters a delay or is disposed of. Travelling from one point to another also implies a delay. Many simulation languages (*e.g.* Arena, Tecnomatix Plant Simulation) follow this approach. As explained by Schriber, Brunner, and Smith (2016), in simulation parallel real-world processes must be emulated on a serial processor, which means that two or more traffic units often have to be manipulated at one and the same time point. The traffic units are manipulated serially at those time points to achieve 'simultaneous' movement. The question now is in which order should the units be treated at a given time point, leading to logical complexities for both the simulation software developer and in many cases, the simulation analyst.

P Point to ponder: Can you suggest a better approach than the transaction-flow? How would you program it?

Entity control

To realise the flow of entities and maintain control over them, many simulation software packages use entity states and associated control structures. These support by implication Shannon's world view. The entity states and associated control structures are shown in Table 1.2, according to Schriber, Brunner, and Smith (2016).

Given the above, the focus area, namely *discrete-event simulation* (DES), can now be defined as

Definition 4. *Discrete-event simulation is one in which the state of a model changes only at a discrete, but possibly random, set of simulated time points, called event times (Schriber, Brunner, and Smith, 2016).*

Table 1.2: Entity states and control structures

Entity state	Control structure
Active: This is the entity the simulation currently deals with. It could be an entity that has just been created, or an entity that is placed in a queue, or at a resource. Once the entity has been advanced until it encounters a delay, or is disposed of, it is not the active entity anymore.	Not applicable, as no control structure is needed.
Ready: Entities in this state are all candidates to become the Active entity. The simulation software has built-in rules to decide in what order to serve the entities. When the simulation software 'stops' to serve entities in this state, it serves them all before it proceeds to the next event in the model.	Current events list (CEL).
Time-delayed: all entities that are served by resources are in this state, including entities being transported by vehicles or on conveyors.	Future events list (FEL).
Condition-delayed: Entities awaiting one or more conditions to become true are in this state. Typically, entities waiting to reach the front of a queue/buffer are kept in this state. Entities waiting for a merge to occur are also in this state.	Delayed list.
Dormant: The analyst must provide programming logic in the model to deal with special entities. Usually, entities follow flow logic through a model, and do so according to the preceding states, structures and Shannon's world view, but sometimes the analyst needs to take full control of entities.	User-managed list.

P Points to ponder: Current computers' processors are discrete machines. How would you simulate a continuous process on a discrete machine? Suppose you can execute the sample path on two processors, how would you manage the events then?

Sample path

The dynamic processes we study require us to consider a time-line on which events are superimposed. These events include entity arrivals in the process or system being modelled, entity departures, entities joining buffers or queues, and entities being served or processed by resources like human beings, machines, transporters or combinations of these. A sample path is used to understand the interactions and delays of entities when simulating a real process that evolves over time.

A simple sample path for three arriving entities at different points in time at a single server is shown in Figure 1.5. The length of the inter-arrival and processing times are arbitrarily chosen and the queue space is very large. The sample path shows events, server busy and idle times, and queue times.

One can apply some of the states and structures in Table 1.2 to the sample path in Figure 1.5. Assume it is an M|M|1FIFO|∞|∞ system that is simulated. So the inter-arrival times and service times follow exponential distributions. Assume the software to simulate this is called 'Simulation Package' or SP for short. The process for three arrivals is described in Table 1.3. The values of the time intervals are not explicitly shown, but we know they are exponentially distributed. Note that only one entity can be in the Active state at a time.

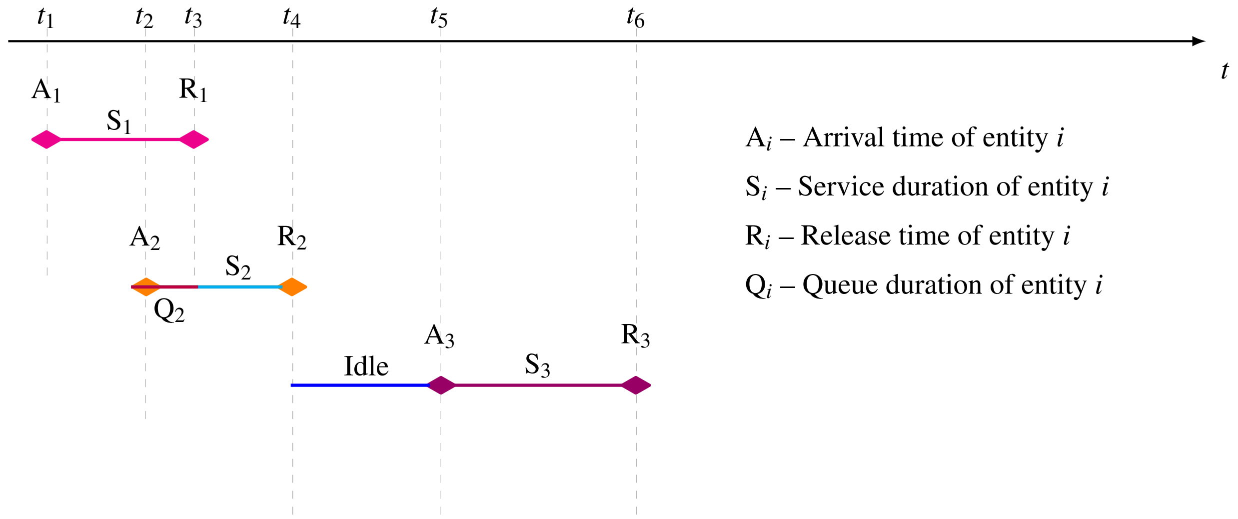


Figure 1.5: A simple sample path illustrated (single server, very large queue space)

Table 1.3: Example to illustrate the association of sample path and entity states and structures in Figure 1.5

At time	SP Action	State	Control structure
t_1	SP creates an arriving entity which is immediately in Active state, and since there are no entities in the system, the entity is assigned to the server. It is placed in the	Time-delayed state on the	Future Events list
t_2	SP creates a second arrival, which is immediately in Active state, and since there is an entity being served, this arrival is placed in a queue, which means it is placed in the	Condition-delayed state on the	Delayed list
t_3	The two entities both become SP takes the served entity, which is now in the The second entity on the Current events list is put in the The Active entity is assigned to the server and put in the	Ready and are placed on the Active state and disposes it Active state Time-delayed state on the	Current events list n/a n/a Future Events list
t_4	The service time of Entity 2 is over, and it is moved to the No other candidates currently exist in the model. The entity is moved to the	Ready state on the Active state and is disposed	Current events list n/a
t_5	A third arrival is created which is immediately in Active state, and since there are no entities in the system, the entity is assigned to the server. It is placed in the	Time-delayed state on the	Future Events list
t_6	The service time of Entity 3 is over, and it is moved to the No other candidates currently exist in the model. The entity is moved to the	Ready state on the Active state and is disposed	Current events list n/a
	There are no more entities to serve and the model terminates.		

1.13 Some simulation software packages

There are many simulation software packages available, including open source and proprietary. In this module, Tecnomatix Plant Simulation (TPS) of Siemens Digital Industries Software will be used. For a complete list, see `https://www.capterra.com/simulation-software/` and `https://en.wikipedia.org/wiki/List_of_computer_simulation_software` (viewed on 21 July 2023).

1.14 Animation in simulation

Animation is a feature that allows for a dynamic graphic representation of the simulation problem on the computer screen. This makes it easier to study the system's dynamics as one can see what happens while the model evolves over time. One danger is that one can get so involved with the animation that the simulation study is neglected. People who validate (see later) simulation models (usually managers) must therefore realise that a balance between studying the problem and representing it graphically must be maintained. Many hours can be spent on creating detailed animations, which usually contribute only marginally to the understanding of the problem under study.

Survival kit:

Construct your simulation foundation using the following:

1. Know the basic concepts of Statistics presented in Section 1.1.
2. Know the difference between *discrete* and *continuous* random variables.
3. Know the generic definition of simulation, and the definition of discrete-event simulation.
4. Know the entity states and associated structures and apply the knowledge through an example of a real-world process.
5. Apply the world-view shown in Figure 1.4.
6. List some advantages and some drawbacks of simulation.
7. List some simulation application areas.
8. Draw a simple sample path, similar to Figure 1.5. More complicated sample paths (more entities that are simultaneously present) should be examined.
9. Describe entity states and control structures similar to what is presented in Table 1.3.

P All the statistics in the world can't measure the warmth of a smile – Chris Hart