1

INTRODUCTION

Unit Structure :

1.0   Objectives

1.1   Introduction

1.2   What is simulation?

   1.2.1 A simple simulation model

   1.2.2 How simulation works?

   1.2.3 Advantages

   1.2.4 Limitations

   1.2.5 Why is simulation used?

   1.2.6 What can be simulated?

   1.2.7 Types of simulation

   1.2.8 Examples of simulation

   1.2.9 Purpose of simulation

1.3   Need of simulation

1.4   Time to simulate Inside simulation software

1.5   Modelling the progress of time

   1.5.1 Time slicing approach

   1.5.2 Discrete event simulation approach (three-phase method)

   1.5.3 Continuous time simulation approach

1.6   Modelling variability

   1.6.1 Modelling unforeseen variation

   1.6.2 Arbitrary numbers

   1.6.3 Relating the variability in a simulation to random numbers

   1.6.4 Modelling time variability

   1.6.5 Selecting samples from normal statistical distributions

   1.6.6 Random numbers produced by a computer

   1.6.7 Modelling expected variation

Summary

List of References

Unit End Exercises

1.0 OBJECTIVES

The objectives of this chapter are as follows:

- To understand what is simulation and how it works

- To acquaint with the significance and purpose of simulation

- To understand what is there inside the simulation software, how it works and the process of simulation

1.1 INTRODUCTION

The facilities needed in a new terminal building are being planned by airport management. Important choices must be made regarding, among other things, the quantity of security check positions, the size of the baggage handling system, the number of departure gates, and the number of check-in desks allocated to each airline. Additionally, it is necessary to decide how many employees to hire and what shifts they should work. These choices must be wisely chosen because the whole investment is in the tens of millions. How can the airport administration decide how many resources are needed in each location?

Building the terminal and hoping it works is one idea. With so much at risk, this seems exceedingly dangerous. Relying on intuitions, perhaps based on prior experience with designing and managing airport terminals, would only be marginally better. Even though a spreadsheet or a few calculations on paper might be helpful, they probably won't be able to tackle the whole intricacy of the scenario.

A simulation of the intended airport terminal is perhaps a far more successful strategy. This might serve as a model for the movement of travellers and their luggage through each of the crucial phases from arrival to departure and serve as the foundation for designing airport amenities. Indeed, many businesses employ simulation models to design new facilities and enhance those already in use. Financial services corporations simulate their call centres, transportation companies simulate their delivery networks, and manufacturing companies simulate their production lines. There are numerous instances of simulation in action.

Three simulation-related questions are addressed in this chapter:

1] What is a simulation and why it is essential?

2] Why would a company decide to create and apply a simulation model?

3] When is simulation useful?

1.2 WHAT IS SIMULATION?

A simulation is a replication of the evolving dynamics of a process or system in the actual world. Even though simulation could theoretically still

be performed "by hand," it now indirectly almost always calls on the use of a computer to fabricate a false history of a system in order to make assumptions about its characteristics and functioning.

By creating a simulation model, which typically takes the form of a collection of assumptions about how the system functions, the behaviour of the system is analysed. Once created, a simulation model can be applied to a number of projects, such as:

- Examine how the system behaves in a variety of situations. Additionally known as "what-if" analysis, this is;

- Before making changes, the system can be simulated to determine how they will affect the real world;

- Simulation can be used to direct system building when it is still in the design stage, or as the system is being created.

Computer simulation has been employed in many different fields, including management science, manufacturing, healthcare, transportation, and the military.

1.2.1 A simple simulation model

Let's say we've decided to build a donut store, but we're not sure how many staff members to hire to serve customers. The real-world system whose behaviour we seek to understand is the operations of our tiny shop. Only a simulation model can give us information because the shop is not yet open.

We might certainly create models of varying complexity, but let's assume for the moment that we are content with a straightforward model that has the following components:

- customers who enter our store at a specific rate;

- staff (of a number to be input) who service customers for a certain amount of time.

Implicitly, we are presuming a limitless supply of doughnuts and completely ignoring the quantity of donuts we have in stock. Of course, to provide a more accurate description of the system, we might also want to integrate this component in a more complex simulation model.

1.2.2 How simulation works?

Simulation creates a visual mock-up of a process using user-friendly simulation software. To effectively represent the real-world process, this visual simulation should incorporate information about timings, rules, resources, and limitations.

This can be applied to a variety of situations; for instance, you could model a supermarket and the typical consumer movements as business picks up. This can help with judgments on the need for more staff, the design of the shop floor, and the supply chain.

Another illustration would be a manufacturing setting where several line segments may be simulated to evaluate how their processes interact with one another. This can give a general idea of how the entire system will operate, which can be used to develop creative ways to boost performance.

1.2.3 Advantages

A variety of benefits can be attained by using simulation, including:

1. **Lower Risk to Your Money**

    Simulated experiments are less expensive than actual ones. Testing theories of actual systems may incur expenditures such as those related to switching to an unproven procedure, hiring personnel, or even purchasing new equipment. Through simulation, you may test hypotheses and steer clear of costly errors in the real world.

2. **Testing Repeatedly**

    A simulation enables you to repeatedly test various hypotheses and inventions under the same conditions. This implies that you can rigorously evaluate and contrast several views without deviating.

3. **Investigate Long-Term Effects**

    By precisely simulating the effects of years of use in a matter of seconds, a simulation that allows you to look into the future can be produced. The ability to see both immediate and long-term effects enable you to confidently make smart financial choices today that will pay off for years to come.

4. **Acquire Understanding for Process Improvement**

    The advantages of simulation are not just realised at the project's conclusion. By putting various theories to the test, improvements can be incorporated throughout the entire process.

5. **Examine Unexpected Events**

    A simulation can be used to evaluate unpredictable occurrences such an unanticipated staff absence or supply chain problems.

6. **Examine Non-Normal Distributions**

    Instead of needing to repeat only the predetermined parameters, a simulation can accommodate for varying and non-standard distributions. For instance, when replicating a supermarket, you can enter several consumer categories who will navigate the store at various paces. An elderly couple or a mother making a weekly shop with two kids in tow will go through the store differently than a young businesswoman picking up a lunch. An accurate representation of the real world can be achieved through simulation by accounting for such changing factors.

7. Promotes In-Depth Thinking Problems can be solved even during the simulation design and parameter selection phase. It is possible to develop ideas or innovations by carefully considering a process or procedure without even using the final simulation.

8. Increase Stakeholder Support

   Additionally, a visual simulation can aid in increasing stakeholders' and partners' buy-in. Visualizing the outcomes of any process modifications and how they were accomplished would increase attention from potential customers and may even allow for a sales pitch based on simulation.

1.2.4 Limitations

Even while simulation has a lot of benefits, it still has significant drawbacks as compared to other related methods and tools, such digital twins.

A digital twin is an extension of simulation that adds real-time feedback and a data stream between the virtual simulation and a real-world asset (or set of real-world assets). A digital twin is an actual object, as opposed to a simulation, which is a theoretical construct.

As a result, simulations cannot be used to evaluate actual real-world problems as they arise.

1.2.5 Why is simulation used?

Simulation is used to assess the impact of new procedures, process modifications, and equipment investment. Engineers can compare various solutions and designs using simulation to evaluate the performance of an existing system or forecast the performance of a planned system.

Instead of testing hypotheses and adjustments in the real world, which might be expensive, simulation is employed. System cycle times, throughput under various loads, resource utilisation, bottlenecks and choke points, storage demands, personnel needs, and the efficiency of scheduling and control systems are among variables that simulation may quantify.

1.2.6 What can be simulated?

Any process or system with an event flow can be emulated. In principle, you can simulate a process if you can depict it in a flowchart. However, simulation works best when it is used with systems or pieces of machinery that undergo continuous change, have changing parameters, or receive unpredictable inputs. For instance, the supermarket we mentioned before has unpredictable and variable elements because of client usage patterns, demands, and stock levels.

Simulating dynamic systems that are complicated and subject to change might provide insights that are challenging to obtain through other techniques.

While managing processes, procedures, and assets can benefit from simulation, Swedish philosopher Nick Bostrom expanded on the idea in his 2003 paper, "Are You Living in a Computer Simulation?" He contends that by incorporating artificial awareness into simulations, it is possible to obfuscate the distinction between reality and simulation, making it challenging to determine whether you are actually living in reality or a simulation. This simulation theory contends that, should you realise that your perceived reality wasn't actually "real," the simulation might change your memories to once again keep you blissfully unconscious that you aren't a genuine person in the real world!

1.2.7 Types of simulation

Three broad categories of simulation can be distinguished, as follows:

1] Discrete event simulation

Modelling a system's evolution over time, for instance;

- industrial processes (stamping, turning, milling)
- traffic study (roads, networks, queues)

2] Dynamic simulation

A system's progression through space being modelled, for instance;

- kinematics of machines
- ergonomics for people
- aerodynamic analysis
- digital prototyping

3] Process simulation

Modelling, for examples, the physical interactions between two or more systems;

- Listing of products used in use
- Modelling of products during production
- Weather prediction

1.2.8 Examples of simulation

There are numerous simulation examples in business, entertainment, education, and other fields. Here are a few noteworthy instances:

1] Automotive

Simulation makes it possible to imitate a real vehicle's features in a digital setting so that the user can experience driving a real vehicle. It is possible to simulate various situations so that the driver has an entirely immersive experience. These kinds of driving simulators can aid in the training of both inexperienced and seasoned drivers, providing a way to impart driving techniques that can lower maintenance and fuel expenses and guarantee the safety of the drivers themselves.

2] Biomechanics

In order to understand the function of anatomical structures in humans or other animals and develop medical treatments and equipment, biomechanics simulation can be used to build models of those structures. Additionally, biomechanics simulation can be used to evaluate joint stresses, mimic surgical procedures, and research athletic performance. Another illustration is neuromechanical simulation, which combines biomechanics and neural network simulation to test theories in a virtual setting.

3] Urban and city planning

In addition to testing how current urban areas might change as a result of policy decisions, simulation can be used to develop new cities and urban ecosystems. This incorporates, among other possible models, the city's infrastructure and traffic flow.

4] Designing the Digital Lifecycle

In addition to examining the lifecycle of the finished product, simulations can help with product design by enabling digital prototyping and testing to develop better performing goods with a shorter time-to-market.

5] Disaster Planning

In order to aid in disaster preparedness, simulations can simulate emergency conditions. This includes planning for reactions to situations like terrorism, pandemics, and natural catastrophes. Responses can be monitored and evaluated through the simulation, exposing potential issues and places where responders may need extra training, as well as ensuring that any mistakes are made in a safe environment before any real-life catastrophe.

6] Finance and Economics

Simulations are useful in macroeconomics, finance, and economics. For instance, historical data can be used as a stand-in for the real economy to evaluate a mathematical model of the economy. This can be used to evaluate budgets, trade balances, unemployment rates, and inflation. Simulators can also be used to evaluate financial models or to imitate the stock market in other contexts. Simulated methods for settlement of securities are also used by banks.

7] Technical Systems

Engineering systems frequently employ simulation to mimic the actions and functions of tools, workflows, and procedures. For process design or improvement, engineering simulations can mix mathematical models with computer-assisted simulation.

8] Layout

This model, also known as a digital human model or anthropometric virtual depiction of the human, can be used in simulation to analyse virtual products and work settings (DHM). In simulated scenarios, these DHMs are capable of simulating human performance and capabilities. Applications for this kind of simulation include waste collection, video games, assembly lines, and disaster management.

9] Aircraft Simulation

For years, new pilots have been trained in a secure setting using flight simulators. This not only makes it possible to evaluate pilots safely, but it also makes it possible to test instrument malfunctions and other issues without endangering the pilot, the teacher, or the aircraft. In addition to saving fuel and other expenses in comparison to actual flight time, it is also simple to repeat the same events, such as approaching a runway to land, under various conditions.

10] Simulation of Marine Craft

It is feasible to mimic operating on a ship or submarine in a manner similar to flight simulation. Simulators may resemble the bridge, the engine room, the cargo handling bay, the communications, or remotely operated vehicles. These are employed by colleges, navies, and training facilities.

11] Applications in the Military

Military simulations, also known as "war games," can be used to test out military strategies in a virtual setting utilising computer models. These are employed by governments and military organisations all over the world and can also involve social and political concerns.

12] Network systems

These simulations have been used to test novel algorithms and protocols before they are deployed in operational systems for networks and distributed systems. Applications for these include the Internet of Things, smart cities, and content delivery networks.

13] Project management

The usage of project management simulation is possible for training and analysis reasons. Simulation is routinely carried out using software tools, whether for manager training or analysing the results of various decisions.

14] Robotics

Robotics simulations are used to replicate circumstances that could be difficult or expensive to recreate and test in the real world. The outcomes of these experiments can subsequently be evaluated and applied to actual robots.

15] Manufacturing Systems

It is possible to evaluate manufacturing procedures, assembly delays, machine setup, and other factors by simulating production systems using techniques like discrete event simulation.

16] Sales

To assess the flow of transactions and client orders as well as expenses, labour times, and more, sales can be simulated.

17] Space and satellites

To prepare space shuttle engineers for launch operations, the Kennedy Space Center employed simulation. People would interact with a mock shuttle and ground support equipment in this scenario. Tests for satellite navigation also involve simulation.

18] Sports

In order to simulate sporting events and anticipate their results as well as the performance of specific athletes, sport statistics are frequently used. In addition to being utilised for fantasy sports leagues, sports simulation can also be used to forecast the results of matches and other events. Additionally, biomechanics models can be used to improve training, gauge levels of fatigue and how they affect performance, and more.

19] Weather

In order to forecast extreme weather events like hurricanes or cyclones, weather forecasting uses simulations based on historical data.

1.2.9 Purpose of simulation

For the following reasons, several categories of systems are addressed to simulation modelling and analysis

1] Getting a Better Understanding of How a System Works

Without a dynamic model, it might be challenging to comprehend how some systems function and interact with one another. To put it another way, it might not be possible to analyse the system by stopping it or by looking at its parts separately. Try to comprehend how manufacturing process bottlenecks emerge as a classic illustration of this

2] Creating Resource and Operating Policies

You might already use a system that you comprehend and would like to enhance. This can be accomplished primarily in two ways: by altering operating or resource regulations. Different scheduling priority for work orders could be a result of modifications to

operational procedures. Staffing numbers or break scheduling could change in resource policies.

3] Trying out new ideas

A simulation model can help give you an idea of how well the suggested system will work if it doesn't already exist or if you're thinking about buying new systems. When compared to the capital expense required to establish any large manufacturing process, the cost of designing a new system might be quite low. It is possible to assess the consequences of various equipment costs and levels. Additionally, using a simulation model prior to installation can help optimise the equipment's configuration.

Many businesses now demand that before making a purchase, sellers of material-handling equipment create a simulation of their suggested systems. The claims made by the different suppliers are assessed using the simulation model. The simulation model is useful even after installation. In the event that the deployed system does not perform as promised, the corporation can use the simulation model to help discover issues.

4] Information gathering without impacting with the system itself

Perhaps the only technique for testing out non-distortable systems is to use simulation models. Some systems are so vital or delicate that it is impossible to change the resource or operational policies in order to examine them. The security checkpoint at an airport would be a prime illustration of this kind of technology. Operating policy or resource level experiments would have a significant impact on the system's operational capabilities or security efficacy.

1.3 NEED OF SIMULATION

To adapt to changes in industry requirements, to optimise operations, and to estimate the possible effects of such improvements, simulation is necessary in logistics.

When experimenting on the actual system is costly, risky, or likely to create a substantial disturbance, it is helpful (e.g. transport systems, nuclear reactor and airline systems).

When it is impossible to model a system mathematically, it could also be an option. Although there are various mathematical analysis techniques, some of them are so complicated that simulation might offer a more straightforward answer. Computer networks, weather forecasting, and oil drilling are a few fields where simulation may be preferred to mathematical modelling.

More advanced technology, such simulation that can deal with the inherent volatility of real-world logistics systems, are needed for logistics operations management.

1.4 TIME TO SIMULATE

The concept of time in a simulation is a variable kept by the simulation programme and is not directly tied to the real-time that it takes to perform a simulation (as measured by a wall clock or the computer's own clock).

Consider the example of NetSim simulator. The virtual clock in NetSim keeps track of virtual time. Virtual time is a positive real number that begins at zero.

To clearly separate it from real (wall-clock) time, this virtual time is referred to as simulation time. As a discrete event simulator (DES), NetSim breaks down the model's evolution throughout the course of the simulation into discrete events where change might occur. Time only moves between occurrences; it is not a continuous process. This means that simulation time can only advance between events, not during them. In actuality, the simulation time is always the same as the moment the current event takes place. As a result, it is possible to think of simulation time as a variable that "jumps" to follow the time allotted for each new occurrence.

When the Simulation time is set to 10 seconds, users wonder if NetSim will operate for that amount of time. and the response is that if the network scenario is really large and has a high traffic load, the simulation may take longer than 10 seconds (Wall clock). Small networks with light traffic volumes could need substantially less time (wall clock). The capabilities of the system also affect how long it takes to perform the simulation. The simulations will execute more quickly on a PC with a more powerful processor and more RAM.

Due to the fact that NetSim's "Emulation mode" involves the transport of real packets through the virtual network, simulation time and the wall clock will be precisely synchronised when used.

INSIDE SIMULATION SOFTWARE

Most of the time, simulation models for operational systems are created using specialised software rather than by hand-coding. Modern simulation software is so powerful that using a programming language is rarely necessary. However, one risk of adopting packaged software is that the user has little knowledge of the fundamentals of the underlying method. Simulation is not a regular activity, in contrast to the majority of the software we use (such as spreadsheets), which merely automates routine processes and facilitates their performance on a bigger scale. This makes the risk considerably larger.

The basics of the simulation technique are described in this section to help readers comprehend what is contained in simulation software. In short, the software consists of two essential components: time progression modelling and variability modelling. All dynamic simulations contain the first, while most simulations contain the second. In fact, by modelling the variability in the first element and the interconnection and complexity in the latter two, a simulation is able to accurately represent the unpredictability,

11

interconnectedness, and complexity in an operational system. First, a description of modelling time progression is given, then a discussion of modelling variability.

1.5 MODELING THE PROGRESS OF TIME

There are various ways to model the passage of time. Here, three distinct strategies are described. To grasp the fundamentals of the simulation methodology, the time-slicing method is given first. Then, discrete-event simulation followed by continuous time simulation is explained.

1.5.1 Time slicing approach

The time-slicing strategy, which adopts a constant time-step (Δt), is the most basic way to model the passage of time. The simplest way to convey this is through an example. Calls to a call centre arrive every three minutes, are routed to one of two operators, and are handled by that operator in five minutes (Figure 1). The inter-arrival time and the service time are now believed to be constant.

Customer arrivals

Time: 3 mins

Operator 1

Time: 5 mins
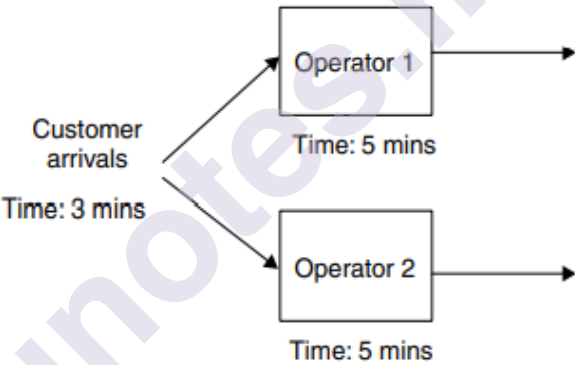
Operator 2

Time: 5 mins

Figure 1: Time-Slicing Approach: Simple Telephone Call Centre Simulation Technique

Table 1 displays a simulation of a 24-minute call centre shift with the timer Δt set to one minute. The time left till a call arrives is displayed in column two. The amount of time left till a client service is finished is displayed in columns three and four. Each operator's total number of calls completed is determined.

A time-slicing simulation may be put up for this scenario quite easily. For more complicated circumstances, the same method may be applied, but the table would quickly grow enormous and possibly become impossible to operate by hand. Larger-scale simulations might be achievable by creating a flow chart defining the order of the steps and incorporating it into a computer programme. A spreadsheet can also be used to quickly model the time-slicing strategy.

The time-slicing strategy has two key drawbacks. It is firstly incredibly ineffective. Since the system state does not change for a large portion of the time steps, many computations are not required. Only the times that a call

arrives, when an operator answers it, and when the operator ends the call are of importance in Table 1. There is a total of 22 such points as compared to the 72 (24x3) estimates made in Table 1's calculations. The bigger the simulation gets, the more likely it is that this issue will get worse.

A second issue is figuring out what Δt is worth. Even while a one-minute time-step for the aforementioned example seems simple, in most simulations, activity durations cannot be counted in full numbers. Additionally, there is frequently a large range of activity times inside a model, ranging from perhaps seconds (or less) to hours, days, weeks, or more. These two problems are both addressed by the discrete-event simulation method.

Table 1: Time-Slicing Approach: Simple Telephone Call Centre Simulation Technique

Time	Call arrival	Operator 1	Operator 2
0	3		
1	2		
2	1		
3	3	5	
4	2	4	
5	1	3	
6	3	2	5
7	2	1	4
8	1		3
9	3	5	2
10	2	4	1
11	1	3	
12	3	2	5
13	2	1	4
14	1		3
15	3	5	2
16	2	4	1
17	1	3	
18	3	2	5
19	2	1	4
20	1		3
21	3	5	2
22	2	4	1
23	1	3	
24	3	2	5
Completed calls		3	3

1.5.2 Discrete event simulation approach (three-phase method)

By modelling systems as undergoing separate state changes at different periods in time, discrete event simulation (DES) analyses the dynamic behaviour of systems. Variables like the number of consumers in the system or the size of a waiting list can be used to represent the state of the system. Every time an event happens, all or part of the variables receive new values. Events mark the transition of an entity into a new state or the allocation of resources. An example of an event is the beginning of an operation.

The N, Q, and S state variables in a DES of a single server queue system are tracked in Fig. 2's bar chart at various periods in time. N stands for the quantity of users entering the system, Q for the length of the single queue, and S for the server state (0-busy, 1-idle).

Discrete state changes at any point of time
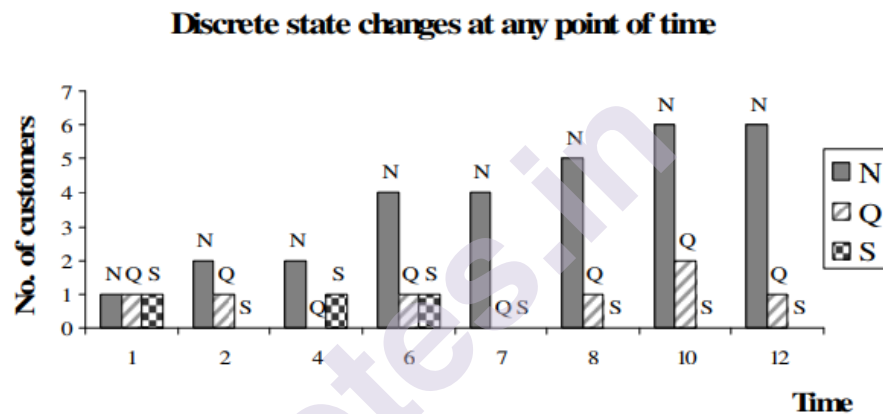
Figure 2: State variables discretely changing at specific times

DES is typically viewed as being separate from continuous simulation, which simulates how a system's status changes over a continuous period of time. But keep in mind that since digital computers are discrete state machines, continuous time is really just a collection of tiny, discrete changes. The mathematical models utilised by these two groups to track the system's states are different. Also keep in mind that this difference is fuzzier in situations where state variables, for instance, change continuously but we can only read their values at specific periods in time.

According to the random variability of the systems, DES and continuous simulations are also divided into deterministic and stochastic simulations. Given that the majority of simulation models have both deterministic and stochastic elements, these two groupings are not mutually exclusive. For instance, in the same scenario, arrivals can change at random while a machine's processing speed might be fixed.

1.5.3 Continuous time simulation approach

A simulation type called continuous-time simulation continuously monitors the target system's condition. In continuous-time simulation, differential equations are frequently used to simulate systems. These differential equations describe the temporal evolution of the system's state.

Computer-based continuous-time simulations are rarely actually continuous-time simulations. Digital computers, on the other hand, approximate continuous-time simulation. This is due to two factors:

1] Since real numbers cannot be counted, they cannot be accurately represented.

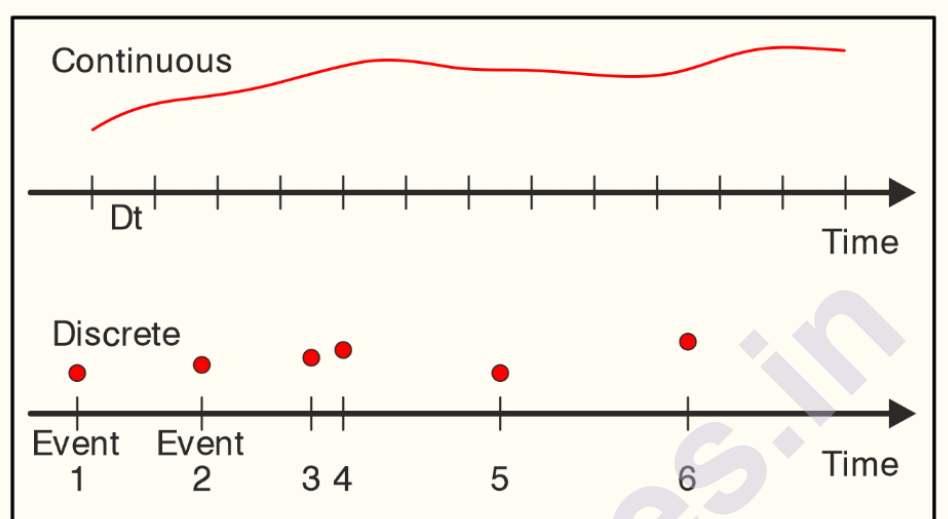2] Differential equations can only be somewhat solved using a discretization technique.

Figure 3: Continual and Discrete Simulation: Updated State Over Simulated Time

Due to this, digital computers use a technique known as fixed-increment time progression to execute continuous-time simulations by discretizing (dividing) time into small time steps. Variable-increment time progression can be used in some complex situations.

Systems that are anticipated to evolve constantly throughout time should be modelled using continuous-time simulation rather than discrete simulations. Continuous-time simulation can be used to simulate a variety of phenomena and procedures, such as: Cities, the environment, ecosystems, electricity and power grids, flight dynamics, hydraulics, intelligent complex adaptive systems, and temperature and humidity are some of the topics covered.

Continuous-time simulation can also be used to calculate the likelihood of various events, such as the likelihood that an industrial equipment would break down or that a piece of programme code will fall short of quality standards. Continuous-time simulation can also be utilised for tasks like system control management or generative design.

1.6 MODELING VARIABILITY

After discussing time progression modelling, the focus now shifts to modelling variability, the second essential component of simulation. In this

regard, modelling unanticipated variability poses the primary issue, which is why a large portion of the discussion that follows is centred on it.

1.6.1 Modelling unforeseen variation

The call centre simulation has not yet incorporated any unanticipated variability or other factors of variability. It is unreasonable to anticipate that the amount of time callers spend with the operators and router will be reduced. Additionally, calls won't come in at predetermined intervals in a ratio of exactly X:Y consumers. How can a simulation accurately depict such unpredictability?

Take the ratio of clients from X and Y as an example to begin responding to this question. It is standard practise to model a single arrival event and to determine the call type as a customer comes rather than modelling the arrival of various clients as distinct events. This eliminates the need for numerous B-events, one for each type of consumer, if there are numerous customer types. We'll assume for the time being that a consumer arrives exactly every three minutes.

Tossing a coin each time a consumer enters the model is an easy approach to determine the call type. A head might stand in for an X client, and a tail for a Y client. This strategy has the drawback of assuming an equal distribution of clients from X and Y. (unless the coin is biased). What if just 40% of your consumers are type Y and 60% are type X? You may illustrate this by taking 10 pieces of paper and writing X and Y on six of them and four of them, respectively. Every time a customer enters the model, a piece of paper from a hat containing the pieces of paper could be drawn to reveal the type of customer. To keep the customer ratio at 60:40, it's crucial that the paper be changed each time.

Although the second method would allow for the modelling of various customer ratios, it is only appropriate for hand simulations since computers cannot pull paper out of a hat! A similar theory is applied in computer simulation and is based on the usage of random numbers.

1.6.2 Arbitrary numbers

A series of integers that occur in a random order are known as random numbers. They are presented as real (with decimal places) numbers on a scale of 0 to 1, or as integer (whole) numbers on a scale of, instance, 0 to 9 or 0 to 99. You could create an integer random number series from 0 to 99 by putting 100 pieces of paper with numbers on them into a hat and drawing numbers out of it. Each time, new pieces of paper are added. The top hat approach is used in this situation.

This method produces random numbers that have two crucial characteristics:

1]   Uniform: Any number can appear at any place in the sequence with the same chance;

2]   Independent: Once a number is picked, the likelihood that it will be chosen again or that another number will be chosen is unaffected.

Due to regular paper replacement, these characteristics are preserved.

There is a list of random numbers in Table 2. You may purchase books with these tables (RAND Corporation 1955) and use spreadsheet functions (like Excel's "RAND" function) to build tables of random integers. Such tables could be saved for a simulation to use, but this would be a very inefficient use of computer memory. Therefore, it is more common to produce the random numbers as needed.

Table 2: Scale of Integer Random Numbers, 0-99

93	43	08	21	61	40	88	36	10	09
34	47	17	99	81	54	44	37	12	97
02	22	48	12	45	00	24	38	43	41
78	71	51	66	19	07	83	29	51	30
82	19	46	05	24	50	09	78	17	64
41	44	39	90	81	22	56	79	25	24
54	32	60	60	32	30	42	50	93	86
23	23	64	16	56	61	21	09	72	36
09	06	82	14	81	05	40	37	55	33
66	86	57	85	63	69	47	56	86	08
27	24	31	05	15	43	45	23	62	03
19	36	86	85	43	17	99	74	72	63
22	00	88	14	84	56	89	95	05	94
87	43	20	07	35	41	51	10	11	31
66	00	05	46	23	22	22	25	21	70
43	28	43	18	66	86	42	91	55	48
28	20	62	82	06	82	79	60	73	67
77	78	43	27	54	89	22	02	78	35
72	67	13	42	46	33	27	66	34	24
06	70	58	78	07	89	71	75	03	60

1.6.3 Relating the variability in a simulation to random numbers

To sample the kind of incoming call, utilise the random integers in Table 2.14. They can be connected to the client type in the following ways so that 60% of the random numbers are related to type X calls and 40% are related to type Y calls:

Random numbers	Customer type
00–59	X
60–99	Y

The first customer to arrive would be a type Y (93), the second a type X (43) and so on, reading across the top row of random digits. The first 10 clients are listed in the following order: Y (93), X (43), X (08), X (21), Y (61), X (40), Y (88), X (36), X (10), X (09). It should be noted that for this sequence, the X:Y customer ratio is 7:3. There won't always be five heads

and five tails when a coin is tossed ten times. In reality, it's feasible that there will be 10 of either at the most. This is due to the random nature of the process rather than a biased coin. It is assumed that the heads-to-tails ratio will be exactly 1:1 after a large number of tosses. In the same way, it is not anticipated that utilising random numbers to categorise clients into types X or Y over a small number of arrivals will result in an exact ratio of 6 (X): 4. (Y). But over a large number of arrivals, the ratio will essentially be reached.

1.6.4 Modelling time variability

For modelling proportions, the method previously provided is helpful. A slight modification to the approach must be made in order to model activity times (or other continuous real variables). The simplest way to convey this is through an example.

Calls have always been thought to occur at regular intervals up until this point. The period between call arrivals is probably going to vary somewhat, and this is plainly unrealistic. Figure 4 depicts a frequency distribution for the call center's inter-arrival time for calls. Although calls arriving simultaneously have an average inter-arrival of zero, actual inter-arrival delays might range from zero to seven minutes.
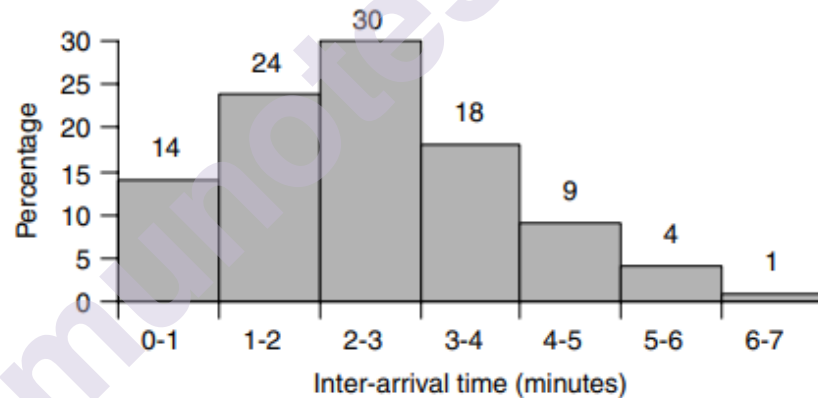
Figure 4: Distribution of Frequencies for Call Inter-Arrival Time.

In a manner similar to how it was done for the fraction of client kinds above, random numbers can be related to the frequencies in Figure 4. (Table 3). The appropriate ratio of inter-arrival times in each range can be found in this way. However, this merely provides the range that the inter-arrival time falls into. A second random number might be chosen, divided by 100, and added to the lower end of the range to get the actual inter-arrival time.

Table 3: Inter-Arrival Times and Random Numbers Have a Relationship

Random numbers	Inter-arrival time (minutes)
00–13	0–1
14–37	1–2
38–67	2–3
68–85	3–4
86–94	4–5
95–98	5–6
99	6–7

Table 4 provides the inter-arrival time for the first 10 calls as an example. Rows six and eleven of Table 2 are used as the source of random numbers. To achieve total independence, distinct rows are used for both these samples and the sampling of customer type. The frequency of samples in each range deviates significantly from the distribution shown in Figure 4, and the mean time for the 10 samples is just 2.38 minutes. The mean and shape of the sampled distribution wouldn't resemble the original data until many samples had been obtained.

Table 4: Inter-Arrival Time of the First Ten Calls: Random Number Sampling.

Customer	First random number (row 6)	Inter-arrival time range	Second random number (row 11)	Inter-arrival time (minutes)
1	41	2–3	27	2.27
2	44	2–3	24	2.24
3	39	2–3	31	2.31
4	90	4–5	05	4.05
5	81	3–4	15	3.15
6	22	1–2	43	1.43
7	56	2–3	45	2.45
8	79	3–4	23	3.23
9	25	1–2	62	1.62
10	24	1–2	03	1.03
Mean				2.38

The variability is produced using a predetermined set of random numbers. As a result, the same set of random numbers can be used repeatedly to generate the series of occurrences, in this case the inter-arrival times. In this instance, sampling the arrival times would entail always starting in rows 6 and 11. This method offers the advantage of being able to regulate the experimental circumstances by allowing experiments with a simulation model to be performed as often as necessary under the same conditions.

1.6.5 Selecting samples from normal statistical distributions

Samples were taken in the preceding subsection by connecting random integers to an empirical distribution. Samples from common statistical distributions, such the normal distribution, are frequently used. The

19

sampling idea is pretty similar to the one mentioned earlier. Consider a normal distribution, as depicted in Figure 5, with a mean of 5 and a standard deviation of 1. The random number chosen is assumed to be a proportion of the area under the curve in order to sample a value from this distribution. The sample value is the location on the x-axis where the area under the curve equals that %, starting from the left end of the distribution. The sample would be drawn from the location where 30% of the area under the curve is found, for example, if the random number is 30. This results in a sample value of 4.48 in the example, as illustrated in Figure 5.
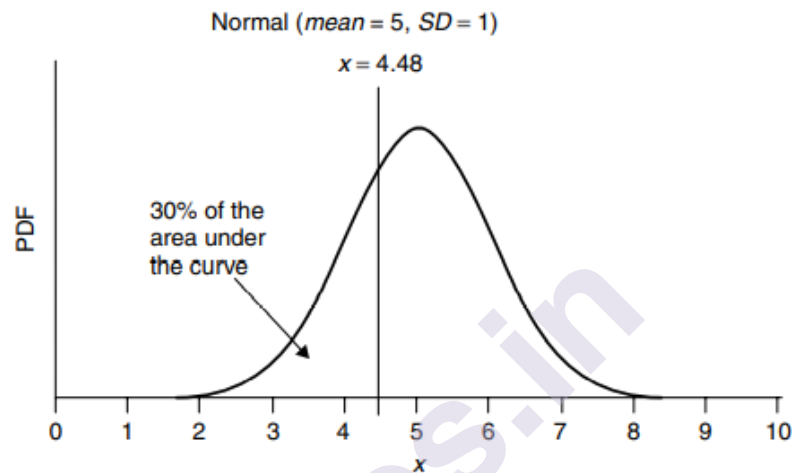
Figure 5: Using a Normal Distribution for Sampling

Thinking in terms of locating the region under a curve is really challenging. Therefore, samples are taken using the cumulative distribution function rather than directly from a distribution's probability density function (PDF), as in Figure 5. (CDF). This details the percentage of the curve's area under the provided x value. The cumulative distribution function for the normal distribution depicted in Figure 5 is shown in Figure 6. The sample value of x can be found by locating the cumulative distribution function's intersection with a random number (let's say 30, for example).
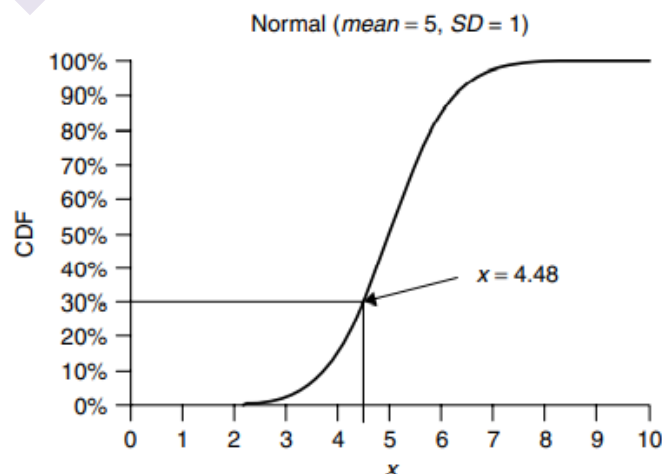
Figure 6: Sampling from a Normal Distribution's Cumulative Distribution Function

Despite the simplicity of the theory, direct or numerical integration of the distribution's probability density function is necessary in order to sample from such distributions (to obtain its cumulative distribution function). Thanks to functionalities offered by simulation software programmes, it is now possible to obtain samples from a variety of practical statistical distributions without having to consult the underlying theory.

1.6.6 Random numbers produced by a computer

During a run, large-scale simulation models could need dozens or even millions of random numbers. It is plainly impracticable to manually generate that many numbers, say, using the top hat method. To hold so many numbers, a large amount of computer memory is also necessary. It is more typical for the computer to provide the random numbers as needed to overcome this issue.

Computers are not good at generating random numbers because their behaviour is not by nature random. However, there exist algorithms that appear to generate random numbers, even though the outcomes are 100% foreseeable! The next number in the sequence can always be predicted (using the technique), but when a stream of the numbers is examined, they exhibit the uniformity and independence necessary for randomness. Therefore, the random numbers produced in this are known as pseudo-random numbers.

The following is a straightforward yet often employed procedure for producing random numbers:

$$X_{i+1} = a\,X_i + c \;(\text{mod } m)$$

where:

X_i: A stream of arbitrary integer values on the range (0, m 1)

a: Multiplier constant

c: additive constant.

m: modulus; mod m refers to taking the remaining amount after dividing by m.

A beginning value for X (X_0), sometimes known as the "seed," is chosen along with values for each of the constants. The X_i can be divided by m if the random numbers must be on a scale from 0 to 1, which is more typical for computer-generated numbers.

The procedure is shown in Table 5 with the parameters $X_0 = 8$, a = 4, c = 0, and m = 25. The greatest number produced by this is always one less than the value of m, and it ranges from 0 to 24. After I = 9, you'll notice that the stream repeats itself. This is a common issue with this approach, so it is important to carefully choose the constant values to make sure the cycle is long enough to prevent repetition during simulation runs. Normally, far

21

bigger numbers, at least of m, are required to ensure the cycle is very long; nevertheless, this example is just illustrative.

Table 5: Algorithm-Based Random Number Generation

i	X_i	$4X_i$
0	8	32
1	7	28
2	3	12
3	12	48
4	23	92
5	17	68
6	18	72
7	22	88
8	13	52
9	2	8
10	8	32
11	7	28
12	3	12
13	12	48
14	23	92
15	17	68

The random numbers that are created can be completely controlled using this type of strategy. The same stream of random integers can be produced repeatedly by utilising the same seed and constants. This provides the same level of control over the experimental settings as using a random number table (Table 2). Modifying the model's parameters, a separate random number seed (X0) must be chosen in order to increase diversity. Different streams of pseudo-random numbers are generated by altering the seed. Pseudorandom number streams are the name given to these streams.

1.6.7 Modelling expected variation

The discussion that has just been had is centred on modelling unexpected fluctuation. Randomness is not necessary for predictable variability; all that is needed is a way to predict when a variation (or event) will happen. An example of a piece of data is the time when an operator begins or ends their shift. The simulation's B and C phases can then be used to carry out the event as usual.

SUMMARY

Industry uses simulations for a variety of purposes to save time and money while testing theories and concepts before putting them into practise. Simulations still have a lot of uses, even though related techniques like digital twin may offer extra advantages owing to the two-way information flow this permits.

For many enterprises and organisations, simulation is a useful tool for testing theories, evaluating procedural performance, or figuring out the lifecycle of an asset.

Based on the characteristics of operational systems and the benefits of simulation, the rationale for its use is examined. The latter explains why simulation is frequently preferred to other possible improvement strategies. Additionally, simulation's drawbacks are noted. Finally, a list of some typical simulation modelling application fields is provided. It also explains how a simulation model functions by demonstrating the modelling of time progression and variability.

The fundamental problems with modelling variability are also covered in this module. The use of random numbers and the ways of connecting these numbers to empirical and statistical distributions to create samples are fundamental to the modelling of unpredictable variability. There is also some discussion of the modelling of predictable variability and the creation of random numbers by computers.

LIST OF REFERENCES

1]   Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004

2]   The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013

3]   Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003

4]   Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003

UNIT END EXERCISES

1]   Define simulation and illustrate it with an example.

2]   Explain the process of simulation and state its advantages and disadvantages.

3]   What are the different types of simulations? State its need.

4]   Illustrate different examples of simulation.

5]   What is the purpose of simulation?

6]   Write a note on time to simulate.

7]   Explain the time slicing approach of simulation.

8]   Write a note on discrete event simulation approach.

9]   Write a detailed note on continuous time simulation approach.

10]   Explain the concept of modelling variability.

❊❊❊❊❊❊❊

2

CONCEPTUAL MODELLING

Unit Structure :

2.0 OBJECTIVES

The objectives of this model are

- To acquaint with the concepts of conceptual modelling

- To get familiar with the requirements and communication of conceptual model

- Different frameworks associated with conceptual modelling

- Adoption of methods for model simplification

2.1 INTRODUCTION TO CONCEPTUAL MODELLING

A simulation of a restaurant serving fast food take many different natures. The easy version is having service desks and queues. However, the model might be enlarged to incorporate the dining room, kitchen, raw material supply, drive-through, parking lot, and other areas. For example, the service desks can be developed as a set period of time. The process might be more precisely described as a sequence of sub-steps and process interruptions and failures could be developed. This conceptual modelling requires the modeller to decide on the right scope and level of detail to model.

2.2 DEFINING CONCEPTUAL MODEL

By defining four terms, Zeigler (1976) clarifies what a conceptual model is. The original model is which a simulation model is meant to mimic. The framework is the constrained set of conditions where the original system is observed; that is we can say that the genuine system is not fully understood. The fundamental model has the capacity to capture the entire behaviour of the real system. This model cannot be fully understood because of how intricate it is. The system's components are grouped together and the interconnections are made simpler in the lumped model. The modeller is completely aware of this model's structure. The conceptual model and the empirical model are equivalent in our understanding.

However, this description only really conveys the idea of concise summary of the actual model. A conceptual model can be described in more detail as follows:

"The conceptual model describes the goals, inputs, outputs, content, assumptions, and simplifications of the simulation model that will be created without referring specifically to any software".

The key components of this definition are two. It is independence from the program in where simulation produced is clearly mentioned in the first sentence. In a perfect world, the software would be chosen based on how well it comprehends. Since the globe is not perfect, it frequently happens that the conceptual model is created using the modeler's programme. In fact, there is interaction between the computer and conceptual model, with

ongoing revisions, because the activities in simulation research are carried out iteratively.

The definition also includes a list of following essential elements:

1] Objectives: The model's intended use.

2] Inputs: Also referred to as experimental factors, are components to enhance or better understand the real world.

3] Outputs: Summarize the outcomes of simulated iterations.

4] Content: the elements that the model represents and how they relate to one another.

5] Assumptions that are made either when there are uncertainties or when there are convictions about the modelled real world.

6] The model has been simplified to speed up model creation and application.

The features of assumptions and simplifications are distinguished. Assumptions are techniques to include doubts and preconceptions about reality in the model. Simplicities can be used to make the model less complex. As a result, simplicity is a side effect of the aim to build simple models, whereas assumptions are a side effect of having restricted information or presumptions.

Two dimensions should be used to describe the model's content (Robinson, 1994):

- The model's domain, also known as the model constrain or the size of the original model.

- The extent of detail is the amount of information that should be provided for each part of the system.

The goal is to specify the foundation upon which the simulation (computer model) is created. In essence, it is a program description. There is a desire for many modellers to begin coding the computer model right away. However, if the conceptual model is not developed with sufficient care, it may not accomplish the desired results and, in the worst-case scenario, may need to be totally overwrite and time consuming.

2.3 REQUIREMENTS OF THE CONCEPTUAL MODEL

A list of needs can be beneficial when creating a system. The model is developed in such a way that it satisfies these specifications. What therefore are the prerequisites for a successful model? This is first addressed by outlining 4 key needs, and then the general necessity to keep the model as straightforward as feasible is covered.

2.3.1 Four criteria for conceptual models

According to Willemain (1994), an effective model must have the following characteristics: validity, usability, client value, practicality, and aptness for the customer's problem. A good model must meet 11 performance requirements, according to Brooks and Tobias (1996). Based on these lists, it is suggested that a conceptual model must meet the following four criteria: validity, credibility, utility, and feasibility.

A valid model is if it can accurately represent the situation at hand. However, as a model with no numerical output has limited use for accuracy, validity might be characterised more accurately as:

"The modeller's belief that it will result in a system that is efficient enough for the task taken".

The difficulty to predict efficiency lies at heart of this idea. In keeping with the majority of definitions of validity, it also upholds the idea that a model is created with a specific goal in mind.

Credibility is also measured through client's viewpoint apart from modeller. Thus, following criteria determine credibility:

"The clients' expectation that the system developed is precise enough for task considered".

Utility is described as follows:

"A belief shared by the modeller and the clients that the system developed will be used as a decision-assistance tool".

Utility is viewed as a shared understanding of the model's usefulness, in contrast to the notions of validity and credibility that are particular to the modeller and the customers. The definition of utility presented here shifts the focus from merely determining accuracy. A variety of system might be created in any situation, and each one might be realistic enough for the task at hand. As a result, each of these models would be reliable and accurate. A implemented model, even one that is vast and laborious but nonetheless accurate enough, might only be of limited use.

Feasibility is the last condition and is defined as follows:

"The modeler's and the clients' belief that the system can be transformed into a computer model".

A model could be unworkable for a variety of reasons. For example, it's impossible to produce the suggested system within the necessary timeframe, the information requirements capability be too oppressive, or the real system may not be well enough understood for implemented system. Whatever the case, it's critical of being capable to convert into a computer model.

Finally, it should be noted that these four ideas are not incompatible with one another. The correctness of a model as seen by the modeller and the clients is likely to be strongly connected. A useless model is also not a beneficial model.

2.3.2 Maintain a basic model

The necessity to prevent the creation of an excessively complex model unites all of the objectives outlined above. To accomplish the goals of the simulation study, the system must be as easy. Simple models have several benefits. As we know the systems structure, they can be constructed more quickly, are much reliable, needs less information, execute more quickly, and simple to induce findings. These benefits are lost as complexity rises.

Good modelling practise revolves around keeping models basic. This is not to say that sophisticated models should never be created; in some cases, they are essential to achieving the study's goals. However, there is a propensity to attempt to represent every component that might accomplish the goals with a great deal low work.

Robinson (1994) uses graph in Figure 1 to illustrate the necessity for simplicity. This demonstrates the increase in model accuracy that comes with difficulty. It demonstrates the common 80-20 rule, which states only 20% of the complexity results in 80% of the accuracy (point x). In addition, the benefits of adding complexity decrease over time. Since it is impossible to accurately represent every component of reality in a model, there can be no model that is 100% correct. In fact, it is suggested that going overboard with complexity may result in a model that is less accurate since the data and knowledge needed to support the intricacy being modelled are not readily available. For example, it is improbable that we could precisely model every individual's behaviour in restaurant scenario and beyond certain guidelines result in lower outcome. This representation is helpful for demonstrating the necessity for simplicity even though it is not based on real data.
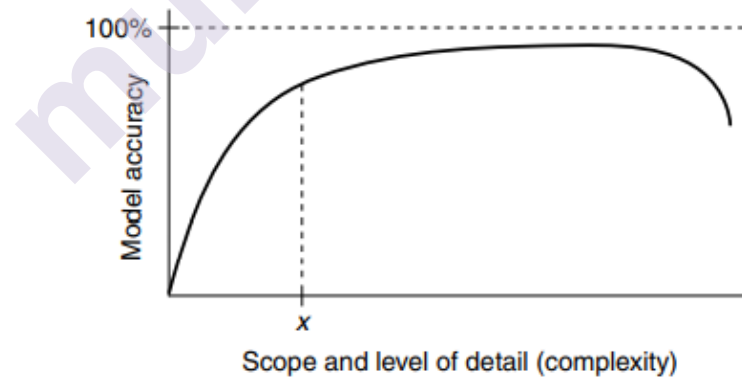

Figure 1: Complexity and Accuracy of Simulation Models

Ward (1989) draws a helpful difference between constructive simplicity (CS) and transparency (T) in a study on simplicity. CS is feature of the system, whereas T is a feature of the client (how proper they comprehend the system). When creating a model, the modeller must take transparency into account in addition to simplicity. Transparency is a client attribute, hence it depends on the client's expertise and understanding. Thus, the model must be created with the demands of the specific client in mind.

2.4 COMMUNICATING THE CONCEPTUAL MODEL

It is crucial that the modeller and clients have a mutual sharing of information and its architecture to assess if it satisfies the 4 requirements outlined above. As a result, a system for disseminating the conceptual model is required. This is one of a project specification's functions.

2.4.1 Project specifications for simulation

A project specification should outline the results of conceptual modelling as well as the administration of the simulation research. In fact, the key method for validating the conceptual model is the project specification. Additionally, it offers a point of reference for creating and validating the computer model, carrying out pertinent tests, and evaluating the effectiveness of the simulation research.

The bulk, if not all, of the following should be covered in the specification, depending on the project's specifics and the rapport between the client and modeller:

- Background information on the issue

- The simulation study's goals

- Expected advantages

- The conceptual model: assumptions and simplifications, inputs, outputs, content (scope and amount of detail)

- Experimentation: Potential Scenarios

- Data requirements: data necessary, when necessary, collection responsibilities

- Timeline and objectives

- Approximate price

A documentation that is distributed to each party engaged in the simulation research often serves as the specification. To track that the paper is overlooked and output is gathered, it is recommended to make it somewhat brief—probably no more than 10 pages. Naturally, at times when a more detailed specification is necessary, just as there are conditions when a quick discussion is sufficient. Each of this is dependent on the model's size, complexity, and level of formality.

The modeller must get feedback in order to assess the validity, credibility, usefulness, and practicality of the suggested system. Discussions about project's administration, such as data collecting, timelines, and expenses, should also be included. It could be helpful to formally depict the system requirements to the team members to get rapid input to help with this

process. All comments should be handled properly. If the conceptual model is questioned, the modeller should either explain rationale of the system or update it in response to the input. This is especially important when assumptions and oversimplifications are called into question. They should choose whether to modify the system, defend the considered scenario, or simplify data. What is better depends on how much flexibility vs a reason induces the model's validity, credibility, utility, and practicability.

It should not be expected that the specification would remain unchanged once model coding begins because the process is iterative. It must anticipate the considerations that will alter for four key reasons:

- The original specification contained omissions

- Alterations in the outside environment

- An improved level of client knowledge of simulation

- The creation and application of a simulation model to identify new issues.

The primary source of change ought to be constrained by effective conceptual modelling, communication, and feedback. Inevitably, the real world undergoes changes. For instance, a production system may alter in design on a little or large scale (for example, by adding a machine). The final 2 modification causes are advantageous features to be promoted.

There needs to be a system in place for dealing with changes because things change. The specification quickly goes out of date and there is no validation if the model is just changed as needed, without any effective reporting. It is helpful to have a "characteristic modification application" that is utilised each time a modification is proposed in order to keep track of changes. To make sure everyone is aware of and in favour of the change, this can be distributed.

Of course, it might be impossible to finish the model development and experimentation if system is constantly varying. Therefore, it is beneficial to arrive to a consensus that the standard has been unchanged. If any modification are recorded, but model is not changed unless the change is especially substantial. Following the conclusion of the simulation and reporting of the findings, the simulation procedure may be repeated with the logged changes factored into the model. Whether this is required depends on whether the changes are deemed significant enough to justify additional modelling.

2.4.2 Representing the conceptual model

Having a way to represent the system content is crucial for the project specification. The four most popular representational techniques are as follows:

1] Component list

This gives a catalogue of elements with a brief explanation of each detail contained. The single server queue is exemplified in Table 1. Even though this method is fairly straightforward, it lacks a graphical presentation of the system making it challenging to record sophisticated rationale and the process description.

Table 1: Catalogue of Elements for a Single Server Queue

Component	Detail
Customers	Time between arrivals (distribution)
Queue	Capacity
Service desk	Service time (distribution)

2] Process flow diagram

According to this method, a system is shown as a process map, illustrating every system part and offering brief summary of the model's specifics. A queue may be depicted as a circle and a procedure as a box. Its representation is shown in Figure 2. The part specifics are indicated in brackets in diagram.

This method is quite straightforward, and the graphic representation helps to illustrate how the process moves along. This strategy is advantageous because numerous simulation software makes use of a comparable format. More complicated logic, however, is still challenging to understand.
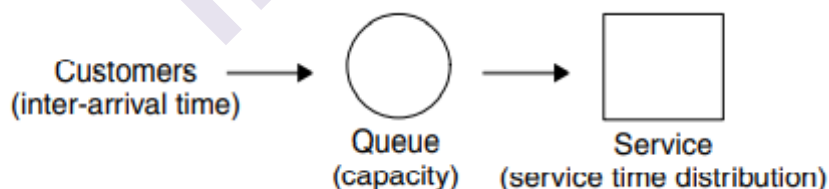
Figure 2: Flow representation of process

3] Logic flow diagram

They use symbols to convey system rationale. Figure 3 depicts an illustration. The user is probably already familiar with the vocabulary, and the illustrations are effective at illustrating logic. However, the process flow is not always evident, and for models of any realistic scale, these diagrams can often grow huge, complex, and burdensome.
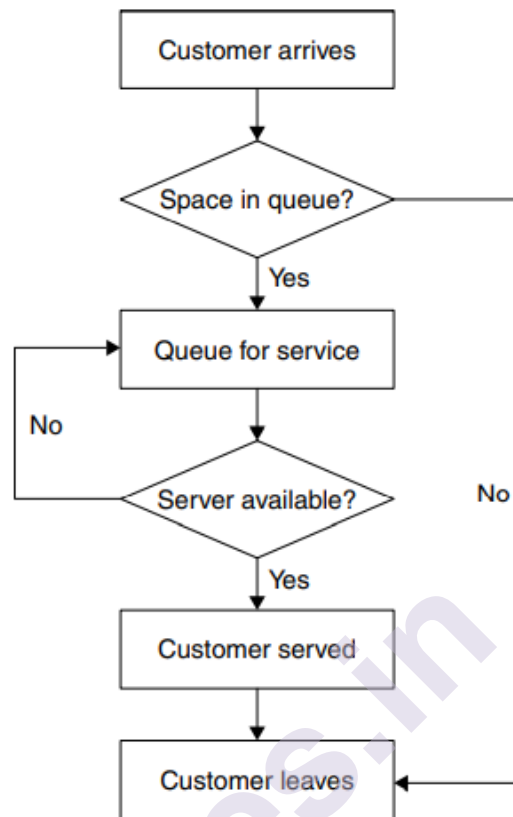
Figure 3: Logic flow representation

4]   Activity cycle diagram

Discrete-event simulation models are represented using activity cycle diagrams (Hills 1971). An illustration of the single server queue is shown in Figure 4. In dead states depicted by circles an object waits for anything to happen. Rectangles indicate the active states, which are where an item is being acted upon. This typically involves processing the item for a period of time before moving on to the subsequent step. Active and dead states typically alternate. In order to build a full activity cycle, in which consumers originate from and return to "outside," Figure 4 includes a dead state of the model.

Activity cycle diagrams are a hybrid of logic flow diagrams and process flow diagrams that partially define a model's logic while also providing a visual depiction. For models with large scales, however, they can easily become exceedingly complex. They have mostly been used as a foundation for programming simulation models because they offer a straightforward way to identify the events in a simulation. As a result, if a simulation package is being used, they are probably less useful.
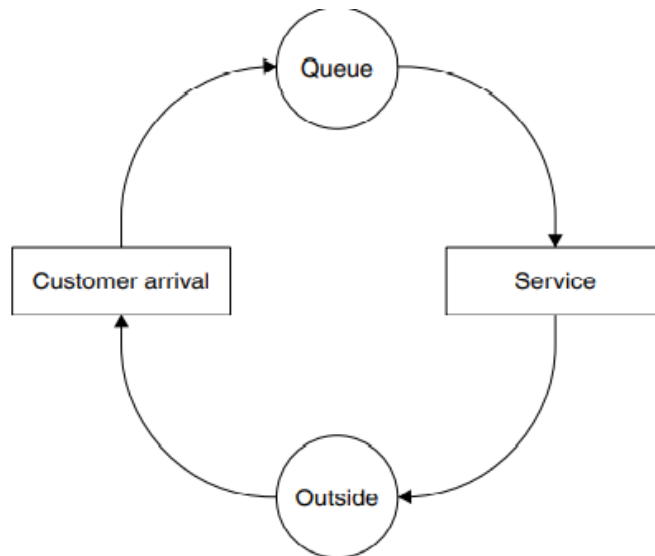
Figure 4: Activity cycle representation

DEVELOPING THE CONCEPTUAL MODEL

2.5 INTRODUCTION

The concept and prerequisites for a conceptual model were covered in detail in the previous section, which served as an introduction to the fundamental ideas. The query of system development was left unanswered. This chapter's topic is that. There are two ways that the question is addressed. First, a conceptual model development framework is provided. Second, various model simplification techniques are described. This first viewpoint assumes that the modeller is working with a blank piece of paper. According to the second viewpoint, the modeller already has a system architecture and it search for methods for enhancement.

2.6 A FRAMEWORK FOR CONCEPTUAL MODELLING

A representation for modelling can be found in Figure 5. A modeller can learn how to create a conceptual model by using this framework, which serves that goal. Four essential components make up the framework:

• Improve your knowledge of the situation at hand

• Establish the modelling goals

• Inputs and outputs for conceptual model design

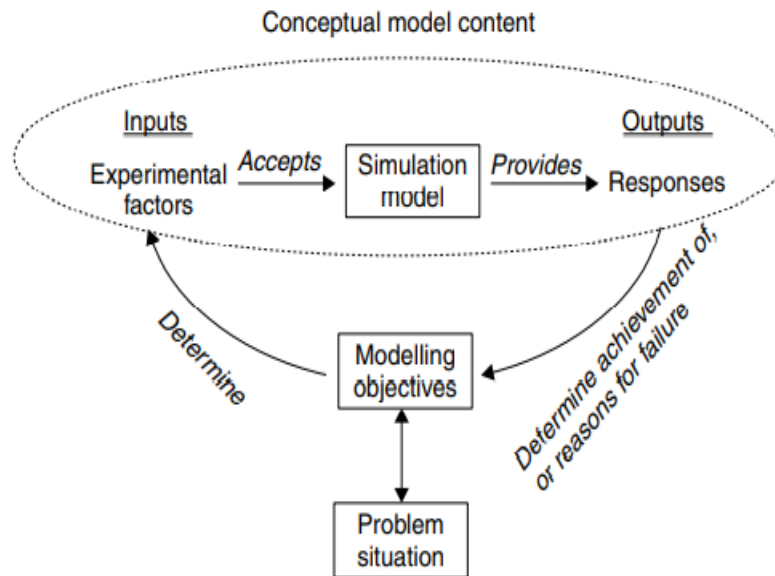• Create the conceptual model and its contents

Figure 5: Conceptual modelling framework

A set of modelling goals are chosen after gaining a grasp of the problem situation. Then, using these objectives as a guide, the conceptual model is generated by first specifying input-output followed by the description of model's actual information. Details of components are provided as follows:

2.6.1 Acquiring knowledge of the underlying situation

In order to develop a system that accurately represents the situation, the modeller must obviously gain a thorough knowledge of system difficulty. The method used in this procedure mostly depends on how well clients comprehend and can articulate the difficult circumstance.

Mostly, the clients offer such description, for example, by outlining how the model operational behaviour that is at the root of the issue. One concern is that the clients could not fully comprehend the relationships between causes and effects that exist in the problem situation. For instance, it was believed that the support function was overloaded (cause) in a description of a telephone helpline resulting in an inadequate level of customer service (effect). However, the effect is accurately detected (the study was indeed conducted for this purpose), it turned out that adding human resources had little to no effect on improving the service. There is a necessity of business process modification.

Clients almost definitely have diverse worldviews, which presents another challenge for the modeller. It appears that varied accounts of how the maintenance engineers approach their tasks as of interviewees exist. This is expected, when working with systems because whims of human behaviour and capability of making decision affect the system's performance.

It soon becomes clear that while the modeller's responsibility appears to be to listen to clients in order to comprehend task, the modeller actually needs to take a stand. To achieve this understanding, it's essential to use the

appropriate stimuli and communicate with the appropriate individuals. In order to encourage fresh perspectives on the problematic issue, they should also be open to suggesting alternate task formulations. Such conversations may take place in person during meetings and workshops, over the phone or over email, for example.

Discussion and meticulous note-taking should be sufficient when customers have understanding of the scenario. Additionally, it's critical that the modeller verify their comprehension by giving the clients details of the problematic issue. More formal issue structuring techniques, such as soft systems methodology, cognitive mapping, and causal loop diagrams, may be helpful if the clients have less understanding of the scenario.

Areas where there is a lack of awareness of the operational behaviour exist during task understanding phase. Assumptions must be made regarding these areas. This ought to be noted in the project specification and documented. As a simulation study develops, new areas of limited information are continually discovered for the reasons listed below. This indicates that fresh hypotheses must be developed before they can be included in the project specification.

The issue at hand should not be viewed as static, nor should our understanding of it. Both will evolve as the simulation research goes on, with the simulation itself serving as one of the driving forces behind this evolution. The knowledge needed to create a simulation model serves as a focal point for elaborating on and deepening our knowledge of the model developed.

Like previously said, a fast-food restaurant serves as an example to illustrate the conceptual modelling framework. The issue at the restaurant is detailed in Table 2.

Table 2: Problem situation of a fast-food restaurant scenario

A fast-food restaurant is experiencing problems with one of the branches in its network. Customers regularly complain about the length of time they have to queue at the service counters. It is apparent that this is not the result of shortages in food, but a shortage of service personnel.

2.6.2 Establishing the modelling goals

The modelling method revolves around the objectives. They serve as a method for determining the nature of the model, a point of comparison for system deployment, a manual a performance indicator to assess the effectiveness.

Since a system consist of little inherent information unless utilised to support to take decision, a modelling study's goal is not to create a model from scratch. If it were, after the model was created, the goal would have been achieved, and the study would be finished. This process leads logically to system development that are not used or are actively seeking a task to resolve. Of course, there is a special case too. A generic model of a hospital

emergency room, for instance, might be created with the intention of selling the model to various hospitals. The creation of a model is the overarching goal of the initial modelling effort. However, the model's creators must have had a goal in mind, such as figuring out resource needs, when creating the model. In fact, it appears that this method is used to construct several military models. A model is created, after which a use is found out. Here, anytime a new purpose is discovered, the model needs to be evaluated.

What standard of performance is required, secondly? It is insufficient to just mention that the goal is to increase throughput. How big of an increase in throughput is necessary? Every time it is feasible, operational goals for each should be found. This could be articulated as simple goals or as the desire to maximise or decrease some measure.

What limitations must the clients (or modeller) operate under, in the end? Often, there is a finite amount of money or strategies at hand to accomplish the goals. For instance, the clients might be unwilling to explore equipment purchases and open to considering adjustments in development formulation in order to increase efficacy.

For the same reasons that their comprehension of the problem scenario may be imperfect and must acknowledged to provide defined goals. Additionally, if a client has never participated in a simulation study before, they should be a constrained and probably incorrect notion of what system can achieve. It is crucial that the modeller be open to recommend new outcomes, as well as to rephrase and scrap the ones suggested. Additionally, the modeller should inform the users by outlining the potential benefits of simulation. One way to do this is for the modeller to show one or more models of problems that are comparable to yours and to describe the modelling effort that went into them. The clients will learn more about how simulation can and cannot help in this way. The modeller should understand as much about the problem situation as the clients should about simulation and its capabilities in the objective context.

The goals may alter and are not at all constant. Additionally, when customers' perceptions of simulation's potential develop, which is inevitable, so will their needs and expectations. Due to the increased iteration between modelling stages, changes in project objectives have an impact on the model's design, experimentation, and final results. This explains why Figure 5's "problem circumstance" and "modelling objectives" are connected by a two-way arrow. Table 3 provides the modelling goal for the fast-food restaurant scenario.

Table 3: Modelling objectives of fast-food restaurant scenarios

The number of service staff required during each period of the day to ensure that 95% of customers queue for less than 3 minutes for service. Due to space constraints, a maximum of six service staff can be employed at any one time.

2.6.3 The inputs and outcomes of conceptual model design

The inputs and outputs of the model, which are shown in Figure 5 as the experimental elements and reactions, are what are focused on in the initial stage of conceptual model design rather than the model's specifics. Starting by thinking about these is significantly simpler than thinking about the model's content. In fact, the transition from the modelling goals to the experimental elements should be pretty simple. In essence, these are the methods that are suggested for achieving the goals.

These methods may be stated in the objectives itself, such as "to boost throughput by altering the production schedule," or "to get a 10% enhancement in service by designing personnel agenda." The modeller must contribute as well using his or her understanding. When taken as a whole, this could result in a long list of factors.

In the actual world, clients would frequently have influence over the experimental elements, but there are occasions when it can be beneficial to experiment with variables. A deeper comprehension of the actual system can be attained by conducting experiments with such variables. After all, this is a major advantage of simulation.

The range that the experimental factors are to be adjusted over should be determined whenever possible. Discussions between the modeller and the clients can help achieve this. What is the largest size that could or would be taken into consideration if a storage area's size, what is the bare minimum and maximum that can be employed if employees shift count is under investigation? After that, the model is developed to support varied incoming information.

The procedure for entering data for the experimental factors should also be covered. This could be done directly in the model code, using a menu system, a information file, or a firmware. This is mostly dependent on the model's intended users and their experience with firmware. This choice correlates to the overarching the goals that were previously established.

Similar to this, finding the replies that the model must produce shouldn't be too difficult. There are two goals for the responses. The first step is to determine whether the goals have been accomplished. For instance, a system must address if efficacy increases by a specific amount. The replies' second objective is to identify the causes of the failure to meet the goals. This can necessitate statistics on machine and resource utilisation. One must be capable to detect the flaws along with the solutions to overcome it.

Another thing to think about is how the data is presented, such as graphical data or numerical data (mean, maximum, minimum, standard deviation). Close collaboration between system and customers, each with skills, should be used to identify appropriate responses and reporting techniques. The kind of reports produced depends on the model's requirements for visual and interactive features, which were discussed above in the section on the project's overall objectives. The essential experimental variables and outcomes for the restaurant case are shown in Table 4.

37

Table 4: Experimental factors and responses of fast-food restaurant scenarios

Experimental Factors
- Staff rosters (total number of staff at each hour of the day)

Responses (to determine achievement of objectives)
- Percentage of customers queuing for less than 3 minutes

Responses (to identify reasons for failure to meet objectives)
- Histogram of waiting time for each customer in the queues, mean, standard deviation, minimum and maximum
- Time-series of mean queue size by hour
- Staff utilization (cumulative percentage)

This changes as the project advances, just like with all other components of the modelling process. For instance, it can be found that altering the business procedure is more successful at improving customer service than altering the staffing rosters. As the experiment develops, it may become necessary to review on the amount of study to comprehend the limitations. The problem scenario or the objectives varies with result of experimental conditions and responses.

2.6.4 Conceptual model designing: model content

Assuming simulation is chosen as the best strategy, the first step in creating system information to acknowledge that the system needs to be capable of accepting the real time cases and supplying the necessary solutions. Here the model's necessary components are based on the experimental conditions and responses. It is instantly clear that the model must include staff rosters in its example. Then, the model must deliver the pertinent reports, such as waiting time. As a result, the queues must be included in the model.

The modeller must next determine the crucial linkages between these and the other elements of the real world after determining incoming and the target scenarios. The only connections that should be considered for inclusion in the model are those that are deemed crucial for accurately generating the information. It is probably helpful to consider the scope first.

2.6.5 Data's significance in conceptual modelling

In an ideal scenario, when precise data for every step to access is system creation. Of course, the world is not ideal. The proposed conceptual model is problematic since not all data are readily accessible or even collectable, and sometimes it is difficult to get sufficient data. The modeller is then left with two choices. Firstly, changing system so that the requirement for problematic content is eliminated. Another option is to steadfastly handling facts in other ways while resisting changing the conceptual model. In reality, the modeller most likely combines the two strategies. As a result, it establishes necessary information, whereas the design is influenced by the accessible or collected data. As a result, the modeller must alternate between taking the model's design into consideration and the availability of data, which increases the level of modification necessary in the process.

2.7 METHODS OF MODEL SIMPLIFICATION

Reducing the scope and level of detail in a model simplification entails one of two methods:

- eliminating parts and connections that don't significantly affect the model's accuracy, or by:

- preserving a satisfactory level of model accuracy while showing parts and relationships more simply.

This can be done either by looking for areas that can be simplified while conceptual modelling, or after the conceptual model is finished and beyond, as during model coding. The basic goal of simplification is to boost a model's utility without seriously compromising its validity or credibility. In general, simplicity allows for quicker model creation and application. If the initial model design is judged unworkable, for instance because the relevant data are not available, simplification could be required.

2.7.1 Combination of model elements

A method for lowering the level of detail is to aggregate model elements. Here, two particular strategies are described: black-box modelling and entity grouping.

- Black-box modelling

A portion of a process is modelled as a time delay in black-box modelling. Model entities that stand in for components, people, information, and other things enter the black box and exit later. With this method, you may simulate anything, from a collection of machines or service counters to a whole manufacturing or service operation. I have built a model of a whole manufacturing supply chain as a collection of interconnected plants, each portrayed as a black-box.

The strategy is demonstrated in Figure 6. The time at which an entity X_i is expected to exit the black box, t_i, is determined as it enters. The entity exits the box at time t_i in the simulation. Of course, a sample from a distribution can be used to determine how long an entity spends in the box. By adjusting the values of t_i for each entity in the box, the method can also be expanded to take stoppages, shifts, and entity re-sequencing (for example, re-work).
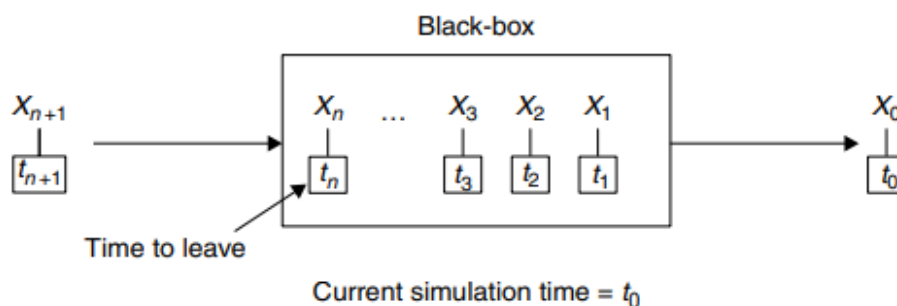

Figure 6: Black-box modelling

• Entity grouping

A simulation entity can represent a set of items instead of modelling each item as it moves through a system. This is especially helpful when a system is processing a large number of things quickly, such during the confectionery wrapping process, where hundreds of chocolate bars are wrapped per minute. The number of events per minute that would result from modelling each chocolate bar separately would be hundreds, which would slow down the simulation run-speed. In this situation, having an entity represent, let's say, 100 chocolate bars, is advantageous.

The method is easily adaptable to modelling scenarios where an entity's number of things it represents changes as it passes through the model. For instance, at a checkpoint, a specific amount of chocolate bars is rejected (or consumed!). This can be modelled by having the quantity of chocolate bars the entity represents as a property. As the entity navigates the model, the attribute value can then be modified.

2.7.2 Holding out parts and details

Some components may not always need to be included in simulations because leaving them out has little impact on the model's correctness. The scope has been reduced in this way.

If it can be assumed that a resource is always, or almost always, available to carry out that task, then the resource is not necessary to be represented in order for the process to occur. In this instance, modelling the process is all that is required. For instance, it is not necessary to explicitly represent an operator on a production line who is committed to a task.

A highly particular example of model simplification that is motivated by the availability of pertinent data is the modelling of machine repairs. It is vital to obtain information on real repair times if the resources needed for repair (often maintenance personnel and maybe some equipment) are to be clearly modelled. Many businesses, however, only track the whole amount of time a machine is down, including the time it takes for resources to be made available. Resources shouldn't be explicitly included in the simulation if downtime data are being modelled, as this would amount to double counting.

A model may not include certain elements that are thought to have minimal bearing on the model's correctness. The modelling of shift patterns is one instance. These must only be modelled if

• Different departments adhere to various shifts.

• Between shifts, there are differences in labor availability, process speed, or process guidelines.

• Outside of shifts, activities like machine repair continue.

• In order to give the simulation legitimacy, shifts must be simulated.

Otherwise, there is no need to simulate the downtime between shifts.

2.7.3 Using random variables to replace components

It might be conceivable to describe a component or group of components as a collection of random variables drawn from several distributions rather than modelling them in detail. Forklift trucks, autonomous guided vehicles, big freight vehicles, and trains, for example, might be complicated to represent. Allowance must be made for breakdowns, punctures, traffic jams, weather conditions, turnaround times, and driver shifts, depending on the situation.

I was tasked with simulating the distribution of items between two locations as part of a model that represented two sites. After spending some time comprehending the delivery process's intricacy and all of its potential pitfalls, it became clear that a complicated model would be necessary for an accurate portrayal. The answer was to find out how many deliveries are made daily and what the average moving time is. The total number of deliveries every day, as well as the timings of the departure and arrival, might potentially be represented as three random variables. The creation of this model and its manipulation during experimentation were significantly easier.

2.7.4 Excluding irregular occurrences

Some incidents only occasionally have an impact on an operations system. Only every two years, a warehouse crane could experience a failure. Major disasters do not frequently affect hospitals. In order to evaluate the operations system under real-world operating circumstances, it is usually advisable to rule out the chance of such events happening during a simulation run.

By executing specific runs in which the event is forced on the model (for example, a crane breakdown or a flow of patients into an emergency department), the influence of such events may always be explored.

2.7.5 Reduce the number of rules

In simulation models, rules are used to establish routes, processing times, schedules, resource allocation, and other factors. By minimizing the number of rules, a model can be made simpler while yet being sufficiently accurate. When it comes to route choices for automatically guided vehicles, for instance, 80% of situations are frequently covered by 20% of the rule set. Decision-making is necessary to determine whether modelling the remaining 80% of the rule set is worthwhile for a marginal increase in model accuracy.

The representation of human interaction with an operational system is a unique challenge in simulation modelling. For instance, it can be quite challenging to predict how people would act when waiting in line for a service. How does one choose which line to get in at the grocery store? When does a person choose to skip one line in favor of another? When might a person opt to get out of a line? What situations would lead someone to choose not to join a line? It is nearly impossible to create a set of rules

that work for everyone in all circumstances because such decisions depend on the individual. As a result, standard procedure calls for the employment of a set of criteria that are simplified. For example, clients may opt to join the shortest queue or refrain from doing so if there are more than five persons in it.

To completely ignore the rule set is an extreme but nonetheless effective strategy. The simulation in the service system scenario above could only possibly assume that people join the shortest queue in terms of queuing behavior. This would imply that long lines would form if there is an imbalance between the service rate and arrival rate. Although implausible, this gives the model's user useful information, such as the fact that the system is unbalanced and that customers are likely to leave unless the service rate can be raised.

2.7.6 Dividing up models

It can be advantageous to divide the model into two or more portions rather than creating a single, huge model. To do this, it is easy to divide the models so that model A's output serves as model B's input, as shown in Figure 7. Data about the model's output, including output time and any entity properties, can be sent to a data file as model A executes. In order to recreate the entities in model B at the proper time, model B is then run and the data is read.

Splitting models has the benefit of making each model run more quickly. Assuming the three-phase method is being used, it is also extremely likely that a single run of all the sub-models is quicker than a single run of a combined model due to less processing at the C-phase. This is because each sub-model contains fewer conditional events. Every time an event happens anywhere in a combined model, every C-event would have to be checked, creating a lot of redundant work. Having separate modelers work on each model in parallel can speed up development time, which is another benefit of dividing models.
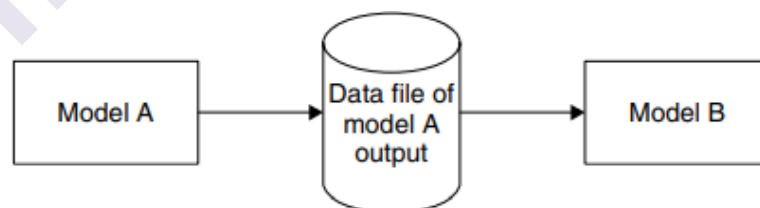

Figure 7: Concept of splitting (dividing up) the models

When there is feedback between the models, separating the models is less effective. For instance, it is not possible to prevent model A from producing an entity if model B is unable to accept it because the first buffer is full, even though in fact this is what would happen. Therefore, it is better to split models at a point with little to no feedback, like where there is a big buffer.

To improve run-speed, there is a lot of interest in running simulations concurrently on different computers. Running split models simultaneously

should make it easy to model feedback effects and circumvent the problem mentioned above. However, there are already a number of challenges to using parallel computing for simulation, not the least of which is creating effective systems for synchronizing the models as they run.

2.7.7 What defines a good simplification?

Even while model simplifications are advantageous, the accuracy of the simulation can be significantly impacted by a bad choice of simplification or oversimplifying a model. A successful simplification is one that benefits from quicker model building and run-speed (utility), while still keeping a high enough level of accuracy (validity). How does a modeler assess the merits of a simplification? There are two major strategies.

The first is to decide if a simplification is likely to have a major impact on model correctness using your best judgement. Discussions between the modeler, client, and other members of the simulation project team should be used to decide this. The project specification serves as a helpful forum for describing and debating the merits of suggested simplifications. Of course, using this method does not guarantee whether or not a simplification is necessary. Before using a certain simplification, it may be helpful to consult with an expert modeler who has a lot of experience with model simplifications.

The second strategy involves prototyping the simplification in the computer model and testing it. Two computer models—one with and one without the simplification—are created by the modeler. The impact on accuracy can then be determined by contrasting the two models' outputs. Naturally, this gives much more assurance regarding the necessity of a simplification, but the benefit of a quicker model creation is lost.

Along with preserving an adequate level of truth (accuracy), a successful simplification should also maintain credibility. A model's trustworthiness can be damaged by oversimplification, which can make it less transparent. Consider the application of black-box modelling. The intricacies of the representation are not transparent, even though a black-box may offer a sufficiently accurate representation of a portion of an operating system. This might be adequate for certain clients, but for others, it might be required to offer a more thorough portrayal in order to give the model legitimacy. In order to ensure the model's credibility, it is occasionally necessary to incorporate a wider scope and more specific degree of detail than is necessary to ensure the model's correctness. An inadequate simplification is one that makes a client doubt the validity of a model. In fact, there are times when it's required to invert the idea of simplification and actually make the model more complex (in terms of its breadth and amount of information) in order to meet the demand for believability.

2.8 SUMMARY

Virtually without a doubt, conceptual modelling is the most crucial component of a simulation study. The success of the simulation study as a

whole depends on the construction of an acceptable model. Unfortunately, conceptual modelling is also the component of simulation modelling that is least well-known. In order to address the problem, this chapter defines a conceptual model and explains its needs, including validity, credibility, utility, and feasibility. It is crucial to create a model that is as straightforward as feasible while still being able to achieve the study's goals. Additionally discussed are ways to represent the conceptual model and how to use a project specification to communicate it.

The topic of creating conceptual models is also covered from two angles: first, by offering a framework for conceptual modelling that enables a modeler to create a conceptual model from scratch; and second, by outlining various techniques for streamlining an existing conceptual model. The framework is demonstrated using a fast-food restaurant as an example.

2.9 LIST OF REFERENCES

1]   Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004

2]   The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013

3]   Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003

4]   Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003

2.10 UNIT END EXERCISES

1]   Define and explain the conceptual model.

2]   What are the requirements of conceptual model?

3]   Write a detailed note on communicating the conceptual model.

4]   Illustrate project specifications for simulation.

5]   State and explain various ways to represent the conceptual model.

6]   Explain the framework for conceptual models.

7]   Write a note on different process involved in the formation of conceptual model framework.

8]   State and explain methods for model simplification

❄❄❄❄❄❄❄

3

DATA COLLECTION AND ANALYSIS

Unit Structure :

3.0 OBJECTIVES

After going through this chapter, students will able to learn

- To understand what are data requirements

- To deal with and collect various types of data

- To represent data using various statistical distributions

- To obtain accurate simulation results

- To identify number of runs and replicaitons

3.1 INTRODUCTION

The initial step in simulation is gathering data and for the same determine what data is essential for building the model. The initial focus should be on defining the overall process flow for more detailed information. The information can then be added gradually as it is available . This detailed approach of data collection enables to build efficient model in the entire process.

One needs to be careful in this data collection because if the data that are used to design and design the model are erroneous then the results from the model will also be erroneous. In this chapter a number of problems with respect to the collection and analysis of data are discussed. Initially the focus is on identifying the data requirements. Later it turns to obtaining the data and in precise how inaccurate data and data that are not available should be dealt with.

3.2 DATA REQUIREMENT AND OBTAINING OF DATA

Data plays an important role in simulation . Data can be qualitatiative (in orm of numbers like age, salary etc.) or quantitative (in terms of some quality like colour of hair, quality of rice etc.). while studying simulation quantitative data is of more use as compared to qualitative data. In this chapter, the word data refers to both quantitative and qualitative data, even though that much of the discussion focuses on the collection and particularly analysis of quantitative data.

It is also important to note that data and information are two different concepts. Information is usually perceived as data with interpretation, in other words data that has been investigated for some purpose. A simulation modeller may be provided with raw data or data that have been interpreted in some manner .

In simulation, as in any modelling exercise, data requirements can be split into three types as follows :

1. Preliminary or contextual data : To develop a thorough understanding of the problem some data needs to be accessible for some processing capability. Here in this phase massive data collection should be avoided, since the data are only required for developing an understanding and are usually not needed for detailed analysis. These data are essential for conceptual modelling process as they help in advancement of conceptual modelling.

2. Data required for developing model: while we move from the conceptual model to a computer model different kind of data are required, for example, customer arrival patterns and descriptions of customer types. It may be essential to carry out a detailed collection process to acquire these data. These data are directly identified from the conceptual model.

3. Data required for validation : It is important to guarantee that each part of the model, as well as the model as a whole, represents the real world system with adequate accuracy. Assuming the existence of real world system , the apparent way to do this is to compare the model results with data from the real system.

    Once data has identified according to requirements the next step is to "obtain data". Some data are easily available while some needs to be collected . As discussed earlier there are three types of data of which:

- Type 1 data is <u>available easily</u> either because they are known or have been collected previously. For example, data may have been collected on service times and arrival rates in a bank for a survey of staffing levels.

- Type 3 data <u>need to be collected</u>. Data that fall into this category include service times, arrival patterns, machine failure rates and repair times etc. While collecting this type of data it is imperative to ensure that the data obtained are both accurate and in the correct format.

- Type 3 <u>data are not available and cannot be collected</u>. These often happens because the real world system does not yet exist, making it impossible to observe it in operation. Another factor is Time availability both in terms of person-time and elapsed time available to collect meaningful data. For example, data are not available on the repair time for a machine.

- There are two main ways of handling type 3 data.

1. To estimate the data from various sources and

2. To treat the data as an experimental factor rather than a fixed parameter. Instead of asking what the data are, the issue is twisted and the question asked is : what do the data need to be? But these can only be applied when there is some control over the data in question.

Data accuracy, is the essential standard of data quality, **refers to the consistency of data with reality.** Even if the data is vaialable or collected one must ensure that it is accurate. The source of the data should be investigated. If the data are too inaccurate for the simulation model, then an alternative source could be required. If alternative is not available, then expert judgement and analysis might be used to determine the more likely values of the data.

In addition to accuracy, data needs to be in the right format for the simulation. The modeller must know the format of the data that are being supplied or collected and also ensure that these are appropriate for the simulation model. In case they are not, then the data should be considered as inaccurate and improvement of the data should be carried out or find an alternative source. The last resort is to treat the data as type 3.

3.3 REPRESENTING DATA

At the heart of simulation modelling lies modelling variability, especially unpredictable (or random) variability. While designing the simulation, the modeller must determine how variability that is present in each part of the model can be represented appropriately.

Three options are available for the same as follows :

1. Traces : A trace is a stream of data that describes a sequence of events. It holds data about the time at which the events occur. It may also hold additional data about the events such as the type of part to be processed (part arrival event) or the nature of the fault(event of machine brekadown). As simulation runs ,it reads the trace and the events are recreated in the model as described by the trace. The data are typically held in a data file or a spreadsheet.

2. Empirical distributions : An empirical distribution is one for which each possible event is assigned a probability derived from experimental observation. It is assumed that the events are independent and the sum of the probabilities is 1. An empirical distribution may represent either a continuous or a discrete. distribution. An empirical distribution shows the frequency with which data values, or ranges of data values, occur and are represented by histograms or frequency charts and are usually based on historic data.

3. Statistical distributions : They are defined by some mathematical function or probability density function (PDF). There are many standard statistical distributions(continuous and discrete) available to the simulation modeller. The best known is the normal distribution that is specified by two parameters: mean (its location) and standard deviation (its spread).

Some of the most useful distributions are divided into three types:

a. Continuous distributions: for sampling data that can take any value across a range for example height of students in a class.

    Some of commonly used continuous distributions are : normal (Figure), negative exponential (Figure) and Erlang (Figure) distributions.

b. Discrete distributions: for sampling data that can take only specific values across a range, for example number of students in a class. Some of commonly used discrete distributions are : Binomila(that describes the number of successes, or failures, in a specified number of trials) and Poisson (used to represent the number of events that occur in an interval of time, for example , total customer arrivals in an hour in a bank)

c. Approximate distributions: used in the absence of data. The simplest form of approximate distribution is the uniform distribution, which can either be discrete or continuous. Another example is triangular distribution.

    Having discussed three types of variability next question is "Which of the three approaches for modelling unpredictable variability should be preferred?"

Each type has its own advantages and disadvantages.

Type	Advantage	Disadvantage
Traces	• Represent historic events in the real system exactly as they occurred. • Help to improve the reliability of a mode.	• need for the real system to exist and for the necessary data to have been collected from that system.
Empirical Distribution	• does not use up large quantities of computer memory.	• difficult to perform sensitivity analysis
Statistical Distribution	• limited use of computer memory • sensitivity analysis can be performed easily	• least transparent approach for the clients, potentially reducing the credibility of the model

A fourth option for modelling unpredictable variability, known as bootstrapping, also is of some interest. Instead of fitting a distribution to the data or summarize the data in an empirical distribution, data are simply re-sampled at random with replacement from the original trace.It is useful when there is only a small sample of data available.

Dependency, or correlation in input data can also be an issue in representing unpredictable variability. To handle such dependencies conditional probability can be used. Another problem that occurs when distributions change over time, stated to as non-stationary input data.One of the simplest examples is modelling arrival times.

3.4 SELECTING STATISTICAL DISTRIBUTIONS

After data has been collected , in order to proceed further a modeler must decide which statistical distributions are most suitable for the model that is being created.

The Distributions needed to model can be chosen by studying the existing properties of the process or can be fitted using empirical data.

1. **Selection of distribution from existing properties of the process:**

    When a process id being modelled it is often possible to choose a model using the properties of process. For example if we have to model arrival time of customers in a café then this arrival is random in nature for a given time interval so a negative exponential distribution can be used. To signify service times for example the mount of time a customer needs to get the service in café can be modelled suitably using Erlang, gamma and lognormal distributions.

If time between arrival is being modelled, then it is sensible to assume a Weibull distribution.

The benefit of using properties to select a distribution is that only the parameters of the distribution need to be defined and that there is no reuirement of collecting complete data. For example, for the negative exponential distribution, just the mean inter-arrival time is needed instead of inter-arrival time between every customer.this method is also useful when data cannot be obtained directly.

2. **Selection of distribution from empirical data :**

In case data is available from experimentation it is feasible to fit a statistical distribution to those data. This process consists of three stages:

o   Select a statistical distribution.

o   Determine the parameters.

o   Test the goodness-of-fit i.e determine how good the distribution fits the data.

Also care has to be taken that a sequence of distributions should tried with different parameters instead of relying on a single distribution . In other words, there should be a number of iterations through these phases.

3.5 OBTAINING ACCURATE RESULTS

In earlier sections we saw requirements of data, planning and building of model , selection of appropriate distributions for the given data. In this section now we proceed with acquiring accurate results and process needed for the same.

3.5.1 Introduction

There are two significant issues in experimentation of simulation.

I To guarantee that the on the basis of model performance accurate results are obtained from the simulation model. We will do in this section

II To confirm that the search for a better understanding and improvements is achieved as efficiently and effectively as possible.This is referred to as searching the solution space and will be done in next section .

In this section , before we discuss precise methods for finding accurate results on performance of model, the nature of simulation models and simulation output are described.

This is significant because it affects the approaches that need to be taken to obtaining accurate results. The primary issues in obtaining accurate results are then explained i.e

- Dealing with initialization bias and obtaining sufficient output data.

- Various methods for dealing with these two issues which are described (warm-up period, setting initial conditions, multiple replications and long runs).

These methods are demonstrated by applying them to the output data from a model of a computer user help desk. The main emphasis is on practical issues, which in some cases are supported by the use of statistical methods.

3.5.2 The nature of simulation models and simulation output

THE NATURE OF SIMULTION MODELS AND SIMULATION OUTPUT

Depending upon the accurate results that are obtained from a model depends the nature of a simulation model and its output .We assume that that the

simulation output is stochastic, for the purposes of this discussion ,that is, the model comprises random events. Eventhough possible to have a simulation model that does not contain any random events, it is not common exercise. It is to be noted that for such deterministic models, some of the concerns described below still needs consideration.

1   Terminating and non-terminating simulations :

A simulation model can be categorized as one of two types: terminating and non-terminating.

For a *terminating* simulation there is a regular end point that defines the length of a run. The end point can be defined in a number of ways, for instance:

- The model reaches an empty condition, e.g. a bank that closes at the end of a day.

- The completion of the time period under search, e.g. the end of the busy lunch period in a supermarket.

- The completion of a trace of input data, e.g. the completion of a production schedule.

    Whereas , a *non-terminating* simulation does not have a natural end point.

- An example is a model of a production facility that aims to determine its throughput capability.

    There is no precise reason as to why a simulation experiment should terminate other than the model user interrupting the run. For non-terminating models the length of a simulation run needs to be determined by the model user.

2   Transient output:

In maximum cases the output from a terminating simulation is *transient*. Transient output means that the distribution of the output is

constantly changing. For aexample ,in a simulation of a bank. One of the responses of interest is the number of customers served in each hour of the day. In addition ,for any time period the number of customers served is unlikely to be same on any given day. This is purely as a result of the random variation in the system.

The distribution of customers served in the hour 11:00–12:00 on any day could be between about 60 and about 100. Over many days, the mean number of customers served between 11:00 and 12:00 will be about 80. Similarly, for each hour of the day there is a distribution of the number of customers served. Because the output data are transient, the distribution varies for each hour of the day.

3   Steady-state output

For non-terminating simulations the output often reaches a *steady state*. Steady state means that the output is varying according to some fixed distribution (the steady-state distribution). Consider the example of a simulation of a production facility. The throughput varies from day-to-day owing to breakdowns, conversions and other interruptions. In the long run, however, the throughput capability (the mean throughput level) remains constant. In steady state the level of inconsistency about that mean also remains constant since the steady-state distribution is constant.

Even though steady-state output is defined with respect to non-terminating simulations, it is possible that a terminating simulation may also reach a steady state, mostly if the termination point infers a long run-length. In the Similar manner, the output from a non-terminating simulation may not reach a steady state.

4   Other types of output

Transient and steady-state are not the only types of output that occur from simulation models. A third type, *steady-state cycle* is also identified by Law and Kelton (2000) . Consider simulation of a production facility working two shifts. The night shift has lesser operators and hence works at a slower rate, in this situation the throughput recorded by the simulation cycles

between two steady states. A similar effect might occur in a 24-hour service operation such as a call handling centre for the emergency services. The rate at which calls are received varies according to the time of the day. Hence , a simulation response there is change in calls handled as the day progresses. Assuming that the call pattern is similar on each day, the simulation output will cycle through the same series of steady states. The pattern may be more complex, with a day-of-week effect as well. In this case there are two cycles overlapped on one another, a daily and a weekly cycle.

Steady-state-cycle output can be distributed simply by lengthening the observation interval in the time-series to the length of the longest

cycle. Instead of recording hourly throughput or throughput by shift in the production example above, the data could be recorded daily. Due to this ,the cycles are subsumed into the longer observation interval and the output analysis can be performed as for steady-state output.

Robinson *et al*. (2002)also define a fourth type of output i.e *shifting steady-state*. In certain models the output shifts from one steady state to another as time advances.For example, this may be due to changes in product type, number of staff or operating practice, assuming that each of these affects the output response. Unlike the steady-state cycle described above, these shifts do not necessarily occur in a regular or even predictable pattern. A heuristic method was also described by the authors for detecting such output behaviour their analysis.

5   Determining the nature of the simulation output

Genarally, the output from terminating simulations is transient and that from nonterminating simulations is steady-state (possibly with a cycle or shifts). However, this is not always the case. Some additional investigation is desirable before determining on the nature of the output.

1   The input data should be examined. Do they change during a simulation run? For example, the customer arrival rate might change as the simulation run progresses. If they do not change, then it is probable that the model output is steady-state. On the other hand if the data change, and the model is terminating, then this is revealing of transient output. If the model is non-terminating, and the data change according to a regular cycle, then this suggests the output is steady-state-cycle.

2   The output data should be explored, mainly by inspecting time-series.If the output is steady-state, then the time-series should disclose a typical initial transientand then steady-state pattern. However, if the output is transient,then the time-series should not resolve, Steady-state cycle output should be reasonably easy to detect, assuming that multiple cycles are not overlaid. Shifts in steady state are not so willingly identified by straightforward inspection and need more detailed analysis (Robinson *et al*. 2002).

3.5.3 Issues In Obtaining Accurate Simulation Results

Let us understand the main difference between model performance and real system performance before discussing the issues in obtaining accurate simulation results.In this sectionwe will discuss about obtaining accurate data on the performance of the model.

The main aim of simulation output analysis is to obtain an accurate estimate of average (mean) performance, although measures of variability are also important .There are two primary issues in guaranteeing the accuracy of the estimates obtained from a simulation model.

- The first is the elimination of any initialization bias, and

- The second is confirming that enough output data have been obtained from the simulation to acquire an accurate estimate of performance.

Both issues, can lead to results if not properly addressed, that are biased and ambiguous.

1 Initialization bias: warm-up and initial conditions

The first issue, the removal of initialization bias, is applicable to non-terminating simulations and sometimes needs to be addressed for terminating simulations. Many terminating simulations start from, and return to, an empty condition.

For example , most service operations open and close each day with no customers present. However, there are, situations where the empty condition is not a realistic starting point.

If a week's production schedule is run through a simulation model, it would be wrong to assume that there is no work-in-progress on Monday morning. We may want to model the lunch period in a bank and it would be inappropriate to ignore the customers who are present at the beginning of this period.

There are two ways of handling initialization bias.

- The first is to run the model for a warm-up period. Basically this involves running the model until it reaches a realistic condition (steady-state for a non-terminating simulation) and only collecting results from the model after this point.
- The second approach is to set initial conditions in the model. Instead of running the model until it is in a realistic condition, the model is placed in a realistic condition at the start of the run. This often means placing work-in-progress into the model at the beginning of a run, for example, customers or parts.
- A third option is to use a blend of initial conditions and warm-up.

### 2 Obtaining sufficient output data: long runs and multiple replications

The second issue, ensuring that enough output data have been obtained from the simulation,can be solved in two ways.

- The first is to perform a single long run with the model.This is only an option for a non-terminating simulation, unless perchance the termination point for a terminating simulation is sufficiently far off to collect enough output data.
- In general, for terminating simulations the only option is to use the second approach,performing multiple replications.

A replication is a run of a simulation model that uses specified streams of random numbers, which in turn cause a specific sequence of random events. By changing the random number streams another replication is performed in which the sequence of random events that occur during the model run changes, as do the results obtained. By performing multiple replications and taking the mean of the results, a better estimate of model performance is gained. Performing multiple replications is equivalent to taking multiple samples in statistics. Meanwhile, performing one long run is equivalent to taking one large sample.Multiple replications as discussed above, is generally the only approach available for obtaining sufficient output data from terminating simulations. Meanwhile, for non-terminating simulations the model user can use either long runs or multiple replications.

3.5.4 An Example Model: Computer User Help Desk

In order to describe the methods for dealing with initialization bias and ensuring that sufficient output data are obtained, it is useful to refer to an example. Figure 1 shows a time-series of output from a simulation of a computer user help desk. This model has been used for a simulation study of a real life help desk.
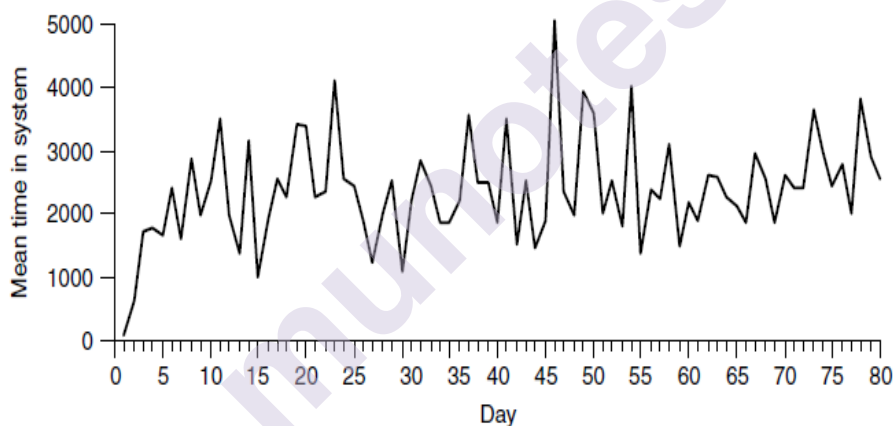
Figure 1 : Time-Series of Mean Time in the System for User Help Desk Example.

The desk receives email enquiries and telephone call from computer users. The enquiries are received, logged and whenever possible dealt with by the staff on the help desk. A high proportion of the calls require technical expertise or a site visit and so they are passed on to the technical team. Because of a backlog of work it may take hours or even days to bring an

enquiry to completion. The time-series in Figure 1 shows the mean time that enquiries being completed on each day have spent in the system (in minutes); this is just one of a number of output statistics that may be of interest. It is evident that there is a lot of variability in the time it takes to complete enquiries.The model is non-terminating and review of the time-series intensely suggests that the output is steady-state. The input data do

not change during the simulation run, lending further weight to this conclusion. There appears to be some initialization bias since at least

the first two observations are low. This is expected as the initial condition of the model is unrealistic; there are no enquiries in the system.

3.5.5 Dealing with Initialization Bias

This section defines two methods for dealing with initialization bias: a warm-up period and setting initial conditions. The third option of using mixed initial conditions and warm-up discussed, as well .The advantages and disadvantages of the different methods are also discussed

1   Determining the warm-up period :

If a warm-up period is to be employed, the main question is what should be the length of the warm-up period? The simple answer is that the warm-up period should be long enough to ensure the model is in a realistic condition. For a non-terminating simulation this normally means that the initial transient has passed and the model output is in steady state. The trouble in this answer lies in finding whether the model is in a realistic condition.

A variety of methods have been projected for identifying and determining the warm-up period and initialization bias. These can be classified into five types.

- *Graphical methods*: which involve the visual inspection of time-series of the output data.

- *Heuristics approaches*: which apply simple rules with few underlying assumptions.

- *Statistical methods*: which depend upon the principles of statistics for determining the warmup period.

- *Initialization bias tests*: that identify whether there is any initialization bias in the data. Rigorously these are not methods for identifying the warm-up period, but they can be used in combination with warm-up methods to determine whether they are working effectively.

- *Hybrid methods*: these involve a combination of graphical or heuristic methods with an initialization bias test.

For those that wish to investigate these approaches further, a list of methods and references is provided in Table 1. All of these methods unfortunately, have limitations and there is no one method that can be recommended for all circumstances. Main problems that occur with these methods are overestimating or underestimating the length of the initial transient, relying on very restrictive assumptions and using highly complex statistical procedures.

Category	Method	Reference
Graphical methods	Time-series inspection	See below
	Ensemble average plots	Banks et al. (2001)
	Cumulative mean rule	Gordon (1969)
	Deleting the cumulative mean rule	Banks et al. (2001)
	CUSUM plots	Nelson (1992)
	Welch's method	Welch (1983)
	Variance plots	Gordon (1969)
	Statistical process control	Robinson (2002)
Heuristics approaches	Schriber's rule	Pawlikowski (1990)
	Conway rule	Gafarian et al. (1978)
	Modified Conway rule	Gafarian et al. (1978)
	Crossing of the mean rule	Fishman (1973)
	Autocorrelation estimator rule	Fishman (1971)
	Marginal confidence rule	White (1997)
	Goodness of fit	Pawlikowski (1990)
	Relaxation heuristics	Pawlikowski (1990)
	MSER and MSER-5	White and Spratt (2000)
Statistical methods	Kelton and Law regression method	Kelton and Law (1983)
	Randomization tests	Yucesan (1993)
Initialization bias tests	Schruben's maximum test	Schruben (1982)
	Schruben's modified test	Nelson (1992)
	Optimal test	Schruben et al. (1983)
	Rank test	Vassilacopoulos (1989)
	The new maximum test	Goldsman et al. (1994)
	Batch means test	Goldsman et al. (1994)
	Area test	Goldsman et al. (1994)
Hybrid methods	Pawlikowski's sequential method	Pawlikowski (1990)

Table 1 : Methods for Determining the Warm-up Period (Robinson 2002).

Two fairly straightforward graphical methods which are commonly used are described here : Time-series inspection and Welch's method.

1 Time-series inspection :

The simplest method for identifying the warm-up period is to inspect a time-series of the simulation output, that is, the key response(s) of the simulation model .The problem with reviewing a time-series of a single run, is that the data can be very noisy and hence difficult to spot any initialization bias. Consequently, if a series of replications are run and the mean averages of those replications for each period are plotted on a time-series it would be better. At least five replications should be performed, while more may be required for very noisy data. The more replications the more the time-series will be smoothed as outliers are subsumed into the calculation of the mean for each period.

To determine the warm-up period, the point at which the output appears to settle into a steady state should be identified. That is, the point at which the data are neither consistently higher or lower than their ''normal'' level and where there is no apparent upward or downward trend in the data. particularly noisy, subtle patterns in the data may go unnoticed. Time-series

output that looks as if in steady state, but more detailed analysis reveals a trend in the data and that the warm-up period needs to be somewhat longer than originally thought.

2   Welch's method

Welch (1983) suggests a method that is based on the calculation and plotting of moving averages.This involves the following steps:

- T o obtain time-series of the output data,perform a series of replications (at least five) .
- Calculate the mean of the output data across the replications for each period (*Yi*).
- Calculate a moving average based on a window size *w* (start with *w* = 5).
- Plot the moving average on a time-series.
- Are the data smooth? If not, increase the size of the window (*w)* and return to the previous two steps.
- Identify the warm-up period as the point where the time-series becomes flat.

The moving averages are calculated using the following formula:

$$\overline{Y}_i(w) = \begin{cases} \dfrac{\sum\limits_{s=-(i-1)}^{i-1} \overline{Y}_{i+s}}{2i-1} & \text{if } i = 1, \ldots, w \\[2em] \dfrac{\sum\limits_{s=-w}^{w} \overline{Y}_{i+s}}{2w+1} & \text{if } i = w+1, \ldots, m-w \end{cases}$$

where:

$\overline{Y}_i(w)$ = moving average of window size w

$\overline{Y}_i$ = time-series of output data (mean of the replications)

i = period number

m = number of periods in the simulation run

While using the Welch's method the intention should be to select the smallest window size that gives a sensibly smooth line. Eventhough selecting a larger window size will give a smoother line, it also tends to give a more conservative (longer) estimate of the warm-up period.It is also recommended that the value of *w* should be no more than a quarter of the total observations in the original time-series. If more observations are required, the simulation model should be run for longer.

Even if Welch's method requires the calculation of moving averages, it is relatively simple to use and also has the advantage that the calculation of the moving average smooths out the noise in the data and helps to give a clearer picture of the initial transient.

However, it is still a subjective method and the conclusion made depends on the experience of the user. A specific difficulty is in determining whether the line is 'smooth' and what is the appropriate window size. Lastly, since the method is based on cumulative statistics (moving averages), according to some it is conservative and tends to overestimate the warm-up period.

Some additional issues related to both method discussed above needs to be stated.

- First,while generating the time-series data the length of the simulation run should be muchgreater than the anticipated warm-up period. Also oone should be sure that the output data have established into a steady state beyond the warm-up period that is identified.

- Secondly, when a model has more than one key response (as defined by the conceptual model, the initial transient should be investigated for each one. The responses may settle to a steady state at different times in the simulation run.

The warm-up period should be selected based on the response that takes longest to settle. Each of these should be investigated for initialization bias.

An alternative to using a warm-up period is to set the initial conditions of the model. There are two ways in which appropriate initial conditions can be identified.

- The first is to observe the real system. In some cases, data on the current state of the real system can be downloaded directly from automatic monitoring systems . Clearly this approach can only be used if the real system exists.

- The second approach is to run the simulation model for a warm-up period and record the status of the model, using this to define the initial condition of the model for future runs.

Besides defining the work-in-progress in a model, initial conditions can be set for the activities that take place in the model. It is often not worth the additional effort in data collection and model coding to include initial conditions for activities. Athird area for which initial conditions can be set is for equipment stoppages (breakdowns and changeovers). If specific initial conditions are not set then the default would effectively be to assume that all activities have just completed a stoppage at the start of the simulation run.

In certain cases it is useful to use a combination of a warm-up period and initial conditions. The aim here is to reduce the length of the warm-up period required.

3.5.6 Selecting The Number Of Replications And Run-Length

In the following section we will describe the methods for finding the number of replications that should be executed with a model and for selecting an appropriate run-length for a long run.

In both the cases the aim is to guarantee that sufficient output data have been obtained from the simulation in order to estimate the model performance with sufficient accuracy.

Along with discussion of multiple replications there is also a brief explanation of variance reduction. At the ens we also discuss the relative merits of using multiple replications and long runs.

1 Performing multiple replications :

A replication is a run of a simulation that uses specific streams of random numbers. Multiple replications are performed by changing the streams of random numbers that are referenced and re-running the simulation. The aim is to produce multiple samples in order to obtain a better estimate of mean performance. How many replications need to be performed? Three approaches to answering this question are: a rule of thumb, a graphical method and a confidence interval method.

A rule of thumb

Law and McComas (1990) recommend that at least three to five replications are performed.

This simple rule of thumb is useful because it makes clear that model users should not rely on the results from a single replication. It however ,does not, take into account the characteristics of a model's output. Models with output data that vary normally require more replications than models with a more stable output. The two methods below address this issue by inspecting the output data from a model.

Graphical method

A simple graphical approach is to plot the cumulative mean of the output data from a series of replications. It is recommended that at least 10 replications are performed initially. As more replications are performed the graph should become a flat line (minimal variability and no upward or downward trend).

The number of replications required is defined by the point at which the line becomes flat. Performing more replications away from this point will only give a marginal improvement in the estimate of the mean value.

If the line does not become flat, then more replications are needed.

Confidence interval method

A confidence interval is a statistical method which shows accurately the mean average of a value is being estimated. The narrower the interval the more accurate the estimate is considered to be. In general, the more sample data that are included in the interval, the narrower it becomes. When applying confidence intervals to simulation output, more replications (samples) are performed until the interval becomes sufficiently narrow to satisfy the model user (and the clients).

When analysing simulation output data a confidence interval is calculated as follows:

$$CI = \overline{X} \pm t_{n-1,\alpha/2}\frac{S}{\sqrt{n}}$$

where:

$\overline{X}$ = mean of the output data from the replications

S = standard deviation of the output data from the replications (see equation belc

n = number of replications

$t_{n-1,\alpha/2}$ = value from Student's t-distribution with $n-1$ degree of freedom and a significance level of $\alpha/2$

The formula for the standard deviation is:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}}$$

Where X_i = result from replication i.

Often a significance level (α) of 5% is selected. This gives a 95% probability that the value of the true mean (obtained if the model is run for an infinite period) lies within the confidence interval (this is known as a 95% confidence interval). On the contrary, it implies that there is a 5% likelihood that the mean does not lie in the interval.

Because the confidence interval provides an upper and a lower limit the significance level is divided by two ($\alpha/2$). So for a 5% significance level, values at 2.5% significance are selected from the Student's t-distribution.

An alternative method for determining the number of replications required is to rearrange the confidence interval formula above so that n (the number of replications) is on the left-hand side, as follows:

$$n = \left(\frac{100St_{n-1,\alpha/2}}{d\overline{X}}\right)^2$$

where:

d = the percentage deviation of the confidence interval about the mean

By performing some initial replications (say five to ten) to estimate S and $\overline{X}$, the number of replications required to achieve a specified percentage deviation (d) can be determined. The accuracy of this method depends, of course, on the accuracy with which S and $\overline{X}$ are estimated from the initial replications.

For both the graphical and confidence interval method it is important to obtain output data from more replications than are required in order to be sure that the cumulative mean line has flattened and that the confidence interval remains narrow. If there is more than one key response ,then the number of replications should be selected on the basis of the response that requires the most replications. Since the graphical and confidence interval

methods use the output data from the model to draw a conclusion about the number of replications required, they are preferred to the rule of thumb.

The confidence interval approach effectively builds on the graphical method by not only enabling an inspection of the cumulative mean line, but also providing a measure of accuracy. As a result, eventhough it requires some more complex calculations, the recommended approach is to use confidence intervals.

2 Variance reduction (antithetic variates)

One of the aim of variance reduction is to obtain an accurate estimate of model performance while reducing the number of replications required. Many methods have been proposed for variance reduction . In practice, however, it would appear that only two methods are used frequently:

- Antithetic variates.

- Common random numbers

Antithetic variates, proposed by Tocher (1963), are the inverse of the random numbers normally generated by a pseudo random number stream. A pseudo random number stream $\{u1, u2, u3, . . .\}$ is inverted to become the stream $\{1 - u1, 1 - u2, 1 - u3, . . .\}$. If samples are taken from a normal distribution, the use of antithetic variates would have the effect of changing the sample given by the original variate to be on the equal and opposite side

of the mean of the normal distribution. In effect, the samples from the original replication are reversed in the second (antithetic) replication. The mean result from the two replications (original and antithetic) gives a better estimate of model performance than from two completely independent replications.

Although the use of antithetic variates is appealing, some words of caution are :

- First, the reversal effect occurs because the normal distribution is symmetrical. If the distribution is not symmetrical, the effect is less marked.

- Second, simulation models normally consist of many random vents that interact in a complex fashion. Therefore, it is difficult to predict the effect of inverting the random streams and certainly it cannot be guaranteed that an equal and opposite result will be obtained. Law and Kelton (2000) suggest that the use of antithetic variates may actually increase the variance in some circumstances, meaning that more replications are required to obtain a good estimate of model performance.

- A third issue is that, although the use of antithetic variates may enable the mean performance of the model to be estimated from fewer replications, the approach by nature restricts the variance in the results. The results cannot, therefore, be used fully to understand the likely spread of model performance (e.g. the standard deviation).

In practice, it is probably sensible to test the effect of using a mix of original and antithetic variates. If it reduces the number of replications required for a particular model then we can continue using the approach. If it does not, then all that is lost is the time taken to test the idea.

3 Performing a single long run

Instead of using multiple replications, if a single long run is to be performed an appropriate length of run needs to be determined. Robinson (1995) describes a graphical method for determining the run-length of a single long run with the aim of ensuring that the results are sufficiently accurate.

Initially, three replications are performed with the model. These should be run for longer than the anticipated run-length. An initial estimate could be made using Banks *et al.*'s (2001) rule of thumb that the run-length should be at least 10 times the length of the warm-up period .Time-series data are generated for the key output data and then cumulative means are calculated for each of the replications. The cumulative meansare plotted on a graph. As the run-length increases, it is expected that the cumulative means of the three replications will converge. If the replications were run for an infinite

period, they would produce exactly the same result! The level of convergence is calculated as follows:

$$C_i = \frac{Max(\overline{Y}_{i1}, \overline{Y}_{i2}, \overline{Y}_{i3}) - Min(\overline{Y}_{i1}, \overline{Y}_{i2}, \overline{Y}_{i3})}{Min(\overline{Y}_{i1}, \overline{Y}_{i2}, \overline{Y}_{i3})}$$

where:

C_i = convergence at period i
$\overline{Y}_{ij}$ = cumulative mean of output data at period i for replication j

The run-length is selected as the point where the convergence is seen as acceptable. This might be at a level of less than 5%. Because of variations in the output data the convergence may temporarily increase with a longer run, particularly when there are only a few observations. It is important, therefore, that the convergence value is not only within an acceptable level, but that it is also fairly steady at the selected run-length. If an acceptable and steady value is not obtained with the output data generated, the run-length should be increased.

It is also recommended that histograms are drawn and compared for the output data from each of the replications. If the model run is sufficiently long the distribution of the output data, as well as the mean, should be reasonably similar.

4 Multiple replications versus long runs

For terminating simulations we need to perform multiple replications. For

non-terminating simulations, such as the user help desk model, there is an option. The question is whether it is better to perform multiple replications or long runs.The advantage of performing multiple replications is that confidence intervals can easily be calculated, and they are an important

measure of accuracy for simulation results. The disadvantage of multiple replications is that if there is a warm-up period, it needs to be run for every replication that is performed. This wastes valuable experimentation time.

On the other hand, with long runs, the warm-up period is only run once for each experimental scenario. This saves time. Another advantage of long runs is that the results probably appear more intuitive to the model user and the clients, since most operations run week-on-week and they are not constantly returned to the same starting state as with multiple replications. It is not easy, however, to calculate confidence intervals from a single time-series, since the data are likely to be correlated. The choice of which approach to use depends upon their relative merits within the context of the simulation study.

3.6 SUMMARY

In this chapter , we discussed a series of decisions that need to be taken when performing simulation experiments. These are as follows:

- Determine the nature of the simulation model: terminating or non-terminating.

- Determine the nature of the simulation model output: transient or steady-state (steadystate cycle, shifting steady-state).

- Determine how to deal with initialization bias: warm-up period, initial conditions or mixed warm-up and initial conditions. This is an issue for both terminating and non-terminatingsimulations.

- Determine the amount of output data required: multiple replications or long runs.For terminating simulations the only option is to perform multiple replications. Either approach can be used for non-terminating models.

3.7 EXERCISE

For the following simulation models identify the expected type of model (terminating or non-terminating) and the nature of the simulation output (transient, steady-state, steady-state cycle).

a) A model of a refrigerator manufacturing plant that aims to determine plant throughput.

b) A model of a chemical plant that tests the production schedule for the next week.

c) A model of a supermarket checkout that aims to determine customer service levels over a typical day.

d) A model of a supermarket checkout that aims to determine customer service levels during a busy period.

e) A model of a hospital emergency unit that aims to determine service levels for patients.

3.8 REFERENCES

- Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004.

- The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013.

- Agent Based Modeling and Simulation, Taylor S, 2014.

- Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003.

- Object Oriented Simulation: A Modeling and Programming Perspective, Garrido, José M, 2009.

- Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003.

- Process Control: Modeling, Design and Simulation, Wayne Bequette W, Prentice Hall of India, 2003.

❄❄❄❄❄❄

4

SEARCHING THE SOLUTION SPACE

Unit Structure :

4.0 OBJECTIVES

The main objective of this chapter is to discourse procedures for searching the solution space .There is a discussion on how the results from a simulation experiment should be analysed. This provides an important foundation for being able to compare alternative scenarios when searching the solution space. The discussion then moves on to cover three key areas in relation to searching the solution space:

- The comparison of results from two or more different scenarios.

- Methods for searching the solution space, covering informal methods, experimental design, metamodelling and optimization.

- Sensitivity analysis.

We will also concepts of verification and validation methods for model testing. The discussion is split into four parts.

- First, the terms verification and validation are defined, and various forms of verification and validation are described and set in the context of the process of performing a simulation study.

- There is then a discussion on the difficulties that are encountered when trying to perform verification and validation.

- Thirdly, some useful verification and validation methods are described.

- Finally,there is a brief discussion on independent verification and validation.

4.1 INTRODUCTION

In this chapter we move on to discuss the selection and comparison of alternative scenarios in experimentation. This comprises of a search for a solution to the real world problem being addressed by the simulation study. It means   finding the best scenario or the one which will satisfy the requirements of clients'. This entire procedure is designated as searching the solution space. The solution space is defined as the total range of conditions under which the model might be run. Two precise terms are used during the chapter.

- *level* for an experimental factor. For experimental factors quantitative in nature (e.g. cycle times, arrival rates) the level is the value of the factor and for qualitative factors (e.g. rules) the level is inferred as an option.
- *scenario*, is a run of the simulation under a specific set of conditions, that is, levels set for experimental factors. A scenario can be thought of as a specific factor/level combination. By changing the level of one or more experimental factors, the scenario is changed.

In the process of determining how well the model performance it reflects the real world issue for verification and validation.Indeed verification and validation intends to determine the accuracy with which the model predicts the performance of the real system.

4.2 NATURE OF SIMULATION EXPERIMENT

There are various forms of Simulation experiments .In this section we discuss two forms :

- Interactive and batch experimentation : this describes the means by which the simulation runs are performed

- Comparing alternatives and search experimentation:this describes the means by which the scenarios for experimentation are determined.

1   **Interactive and batch experimentation :**

Interactive experimentation comprises inspecting the simulation run and making changes to the model to see the effect. For example, while watching the simulation, the model user might notice a bottleneck in one area of the model. The capacity of that area could be increased (e.g. faster cycle, more machines) and the model run continued to see the effect of such a change. The aim here is to cultivate an understanding of the model (and so the real system), its key problem areas and recognize possible solutions. Such an approach is very useful for facilitating group decision-making.

2   **Batch experiments** are achieved by setting the experimental factors and leaving the model to run for a predefined run-length (or to a specific event) and for a set number of replications. This needs no communication from the model user and so the display is usually switched off. This also improves the run-speed of the model. The aim here is to run the simulation for sufficient time in order to acquire statistically significant results. The run-length (including a warm-up period) and the number of replications are determined using methods such as those described in Chapter 3.

There are a restricted number of scenarios to be compared when comparing alternatives. These scenarios are frequently known at the start of the simulation study, for example, there may be three alternative factory layouts. On other occasions the scenarios arise as the simulation study progresses. The number of scenarios (the solution space) is often small, although there are occasions when a large number exist.

In *search experimentation* there are no predefined scenarios. As an alternative, one or more experimental factors are varied till a target or optimal level is reached. For instance,the aim might be to reach a target throughput or to achieve an optimum level of customer service by balancing the cost of resources with the cost of lost custom. For this type of

experimentation there either needs to be a clearly defined target, normally expressed in the objectives of the project, or a well defined function (e.g. cost or profit) to be optimized.

4.3 ANALYSIS OF RESULTS FROM A SINGLE SCENARIO

In order to decide the performance of the model, Simulation experiments are achieved. This is measured in terms of values of the responses. For each response two measures are generally of interest:

➢   The average (or point estimate) : Mean is commonly used to measure the average level of a response.It would be possible to obtain an exact value of the mean for each response if a simulation could be run for an infinite amount of time. Since this is not practically possible, we must depend on upon simulation runs which provide a sample of

results. Since simulation experiments provide only a sample of output data it is important that a confidence interval for each mean is stated. A confidence interval delivers information on the range within which the population mean (obtained from an infinite run-length) is expected to lie. It is, therefore, the primary method for reporting the mean in simulation studies. It would be beneficial to be able to construct a confidence interval for the output data since single long runs have a number of advantages over performing multiple replications. Various methods have been proposed for accomplishing this:

- Batch means method

- Overlapping batch means method

- Regenerative method

- Standardized time-series method

- Spectral estimation method

- Autoregressive method

We will discuss only batch method here.

In the batch mean method, only one simulation run is executed. After deleting the warm up period, the remainder of the run is divided into k batches, with each batch average representing a single observation In the batch means method the time-series of output data ($Y_1, Y_2, \ldots, Y_n$) is divided into k batches of length b, such that the mean of each batch is calculated as follows:

$$\overline{Y_i}(b) = \frac{1}{b} \sum_{j=1}^{b} Y_{(i-1)b+j}$$

where:

$$\overline{Y_i}(b) = \text{batch means of length } b$$

The batches can be assumed to be independent of each other if the batch size is sufficiently large. In this case the confidence interval can be constructed in usual manner as :

$$CI = \overline{X} \pm t_{k-1,\alpha/2} \frac{S}{\sqrt{k}}$$

where:

$S = \text{standard deviation of the batch means}$
$\overline{X} = \text{mean of the individual data}$
$t_{k-1,\alpha/2} = \text{value from Student's } t\text{-distribution with } k-1 \text{ degree of freedom and a}$
   $\text{significance level of } \alpha/2$

The primary issue with this method is determining the batch size. Numerous methods have been proposed, but not any seems to be satisfactory.

- Schmeiser (1982) proposes that the time-series should not be split into more than 30 batches. He found that the accuracy of the confidence interval does not improve greatly by having more batches. He also recommends that there should be no fewer than 10 batches, since this also affects the accuracy of the interval.

- Fishman (1978) recommends a process based on the von Neumann (1941) test for correlation. The batch size is doubled until the null hypothesis that there is no correlation in the batch means is accepted. An advantage of the von Neumann test is that it can be functional to small sample sizes (as few as $k = 8$ batches).

- Beyond Schmeiser and Fishman, many other measures have been proposed for determining the batch size. For instance, Banks *et al*. (2001) propose a four-step method and Hoover and Perry (1990) outline an approach that uses the runs test to check for independence.

  Alternative measure of average performance is the median and quantiles estimation. In other words the level of performance can be achieved with a given probability. The median is simply the 0.5 quantile, and the upper and lower quartiles the 0.25 and 0.75 quantiles respectively

- ➢ The variability : An average does not provide a complete depiction of model performance. We may have instances where two different data have the same mean (and indeed mode), but the variability is much greater in the one data set. Majorly we prefer a lower levelof variability since it is easier to match resources to the levels of demand. Apart from creating histograms of the output data, useful measures of variability are the minimum, maximum and standard deviation. One needs to take care of outliers when stating the minimum and maximum, else these measures may be ambiguous. For a median, quartiles and more generally quantiles provide a measure of variability. Time-series plots are also significant, since they display the form of variability over time.

4.4 COMPARING ALTERNATIVES

When matching alternative situations the model user must be capable to determine whether one alternative is improved than another. Following factors need to be considered:

➢ What is the standard deviation of the mean daily throughput for the two situations?

➢ How many replications (or batches) were used to generate the results?

  If the data have been generated from only a few replications and there is a lot of variation in the results, this gives little confidence that the difference is significant. However, if many replications have been

executed and the standard deviation is low, there can be more confidence that the difference is real.

A realistic method would be to deliberate all three factors

- the size of the difference,

- the standard deviation and

- the number of replications

Then make a decision whether the difference in the results is significant. A more difficult approach depend on on developing confidence intervals for the difference between the results.

We will now discuss how to compare scenarious .Following are some cases:

1 **Comparison of two scenarios**

Supposing that common random numbers are being used in the model a confidence interval for the difference between the results from two scenarios can be calculated as follows :

$$CI = \overline{D} \pm t_{n-1,\alpha/2} \frac{S_D}{\sqrt{n}}$$

$$\overline{D} = \frac{\sum_{j=1}^{n} (X_{1j} - X_{2j})}{n}$$

$$S_D = \sqrt{\frac{\sum_{j=1}^{n} (X_{1j} - X_{2j} - \overline{D})^2}{n-1}}$$

where:

$\overline{D}$ = mean difference between scenario 1 (X_1) and scenario 2 (X_2)
X_{1j} = result from scenario 1 and replication j
X_{2j} = result from scenario 2 and replication j
S_D = standard deviation of the differences
n = number of replications performed (same for both scenarios)
$t_{n-1,\alpha/2}$ = value from Student's t-distribution with $n-1$ degree of freedom and a significance level of $\alpha/2$

This formula is essentially the same as the confidence interval formula given earlier in Section except that it uses a single set of values the rather than the difference between two sets of values .The subsequent confidence interval can lead to one of three outcomes as follows :

a) The confidence interval is completely to the left of zero. It can be concluded, with the specified level of confidence (normally 95%), that the result for scenario 1 is less than the result for scenario 2.

b) The confidence interval includes zero. It can be concluded, with the specified level of confidence (normally 95%), that the result for scenario 1 is not significantly different from the result for scenario 2.

c) The confidence interval is completely to the right of zero. It can be concluded, with the specified level of confidence (normally 95%), that the result for scenario 1 is greater than the result for scenario 2.

In order to identify the statistical significance of a difference in the results from two scenarios a paired-*t* confidence interval can be used.

2   Comparison of many scenarios :

To enable more than two scenarios to be compared at once,the paired-*t* confidence interval can be extended by use of the Bonferroni inequality. This states that if we wish to make *c* confidence interval statements with an overall significance level of *α*, the individual confidence intervals should be formed with a significance level of *α*/*c*.

For example, if 10 confidence intervals are to be formed for comparison and an overall significance level of 10% (90% confidence) is required, each confidence interval should be calculated with a significance level of 1% (99% confidence).

3   Choosing the best scenario(s) :

Apart from comparing scenarios, a group of scenarios are of much interest. At the simplest level this can be achieved by inspecting the mean results for each scenario. Also , merely comparing point estimates does not take explain the standard deviation of the results or the number of replications accomplished. Hence it is better to refer to confidence intervals.

Beyond comparing means and using confidence intervals for differences, there are statistical methods for choosing the best scenario known as ranking and selection methods.

In this section our discussion is centred on identifying the statistically best scenario. The model user and clients need to discuss the practical issues surrounding the decision to determine whether the statistically best scenario is indeed the best decision.

4.5 SEARCH EXPERIMENTATION

Since there is the prospective to have many scenarios (factor/level combinations) in search experimentation, very often it is not possible to simulate every single scenario in the time available in order to determine which meet the target required or provide the optimum result. Subsequently, procedures need to be established for improving the efficiency of the experimentation process. There are three methods for accomplishing this:

1   *Experimental Design*: identify the experimental factors that are most likely to lead to significant improvements, thereby reducing the total factor/level combinations to be analysed.

2   *Metamodels*: fitting a model to the simulation output (a model of a model). Because the fitted model runs much faster than the simulation, many more factor/level combinations can be investigated

3   *Optimization*: performing an efficient search of the factor/level combinations, trying to identify the optimum combination There is much written on these approaches and each provides a fertile area for continued research.

Though these methods are described as separate topics, it must be recalled that they overlap with one another and that the approaches can be used in combination during experimentation. Before discussing the formal methods to carry out search investigation let s discuss some informal approaches.

4.5.1 Informal Approaches to Search Experimentation

Many simulation model users do not have the essential skills and simulation software usually does not provide ample support for search experimentation, with the exception of optimization .In such cases , informal approaches can be quite operative and they also have the advantage that the model user is closely involved with the selection of scenarios.

Following are some informal approaches to search experimentation.These are classified under three headings that relate closely to those listed above:

1   Identifying important experimental factors (similar to experimental design):
There are three ways in which the importance of an experimental factor can be identified:

   a)   *Data Analysis*: by analysing the data in a model it is sometimes possible to draw conclusions about the likely impact of a change to an experimental factor.
   b)   *Expert Knowledge*: subject matter experts, for example, operations staff, often have a good understanding of the system and the factors that are likely to have greatest impact.
   c)   *Preliminary Experimentation*: varying the levels of experimental factors and executing the model to see the effect. If used with caution, Interactive experimentation may be helpful in this respect, although it is important to perform batch experiments to test fully the effect of a change to an experimental factor.
   An advantage of Data analysis and expert knowledge as compared preliminary experimentation is that they require less time. However, Preliminary experimentation, delivers a more detailed means for inspecting the effect of a change to an experimental factor.

2   Developing an understanding of the solution space (similar to metamodelling)
It is often possible to form an opinion by simulating a limited number of scenarios (factor/level combinations) as to the likely outcome of other scenarios without having to run the simulation .It may be likely to identify those scenarios that are possible to yield the anticipated result and those that are unlikely to do so. Through this process the model user forms an understanding of the solution space.

3   Searching factor/level combinations efficiently (similar to optimization)

The model user should recognize factor changes that have the utmost influence in improving the simulation result or moving the simulation result towards the wanted objective. The user can then ponder on continuing to change those experimental factors in the direction of the improvement. For instance, if the addition of service personnel leads to the greatest improvement, then further service personnel could be added.

An additional apprehension is recognizing a scenario that appears optimal when an extensive search would disclose a better result. This can only be handled by jumping to quite different factor/level combinations to understand if there is a substantial enhancement.

4.5.2 Experimental Design

Experimental design acts as a method of recognizing important experimental factors, that .These are those factors to which changes are most likely to produce the desired result. It is a formal method for carrying out the preliminary experimentation As such,experimental design can be valuable, mainly in the early stages of experimentation, for recognizing scenarios that should be simulated.

2k factorial designs

One approach to experimental design is to adopt a 2*k* factorial design, where *k* is the number of experimental factors as described by Law and Kelton (2000)

If k number of variables/factors are considered to determine/screen the important ones, the total number of treatment combinations for a k number of factors can be calculated as in Equation given below :

2x 2 x ----- x 2 = 2^k

Therefore, this method is known as the *2^K design of experiments.*

The main advantage of this methd is it helps you to screen out factors strongly influencing your response variable from those which are not.

However there are some restrictions in using 2^k factorial designs as follows:

1. If interaction effects exist then the interpretation of the main effects becomes more difficult.

2. It is precarious to extrapolate the findings of a 2^k factorial design outside the range of the levels used in the simulation runs.

3. Because the approach interpolates between results, a linear model is assumed effectively.

Some other methods of experimental design are :

*   *Fractional factorial designs* which are applied when there are too many factors to enable full experimentation with every factor/level combination.

- *Analysis of variance* (ANOVA) provides a more rigorous means for identifying the effect of changes to factors. It involves a series of hypothesis tests in which it is determined whether changes to the experimental factors have an effect on the response.

4.5.3 Metamodelling

As the name indicates a metamodel is a model of a model, or a model of the simulation output. Because the metamodel is usually an analytical model it runs much faster than the simulation. It is, therefore, probable to investigate many more scenarios with a metamodel than with the simulation itself. The disadvantage is that the metamodel is an approximation of the simulation output and so the results it provides are not precise. There is also the overhead of creating the metamodel.

Once the metamodel is fitted it is used for continued experimentation in place of the simulation. In doing so it must be recollected that the metamodel is only an approximation.It should only be used to identify candidate scenarios, which will be then executed in the full simulation model. It is also unsafe to extrapolate results from the metamodel outside the range of the factor/level combinations used to create the model.

4.5.4 Optimization

The main aim in simulation optimization is to find the combination of factor/level that gives the finest value for a response, that is the maximum or minimum value. The problem is similar to standard mathematical optimization methods. There is some objective function to be optimized, typically, cost, profit or customer service. Then there is a set of decision variables that can be changed; in simulation these are the experimental factors. Finally, there are a series of constraints within which the decision variables can be changed; this is expressed in terms of the range within which the experimental factors can be altered.

The difference between two optimizations is that there is no algorithm for guaranteeing an optimum solution. One standard approach to use heuristic

search methods .The difficulty,though, is that a heuristic search requires the simulation to be run, which makes it a time consuming approach.

Many simulation software dealers deliver optimization packages for their software.The majority of these use heuristic search approaches and in particular a set of methods known as meta-heuristics, like simulated annealing, genetic algorithms and tabu search (Reeves 1995; Debuse *et al*. 1999).

But none of the optimization packages can assure that an optimum solution will be found.

4.6 SENSITIVITY ANALYSIS

Sensitivity analysis determines how different values of an independent variable affect a particular dependent variable under a given set of

assumptions. This model is also referred to as a what-if or simulation analysis.

In other words, sensitivity analyses study how various sources of uncertainty in a mathematical model contribute to the model's overall uncertainty. In sensitivity analysis the consequences of changes in model inputs are assessed. In this framework model inputs are inferred more generally than just experimental factors and include all model data. The sensitivity analysis is as shown in figure :
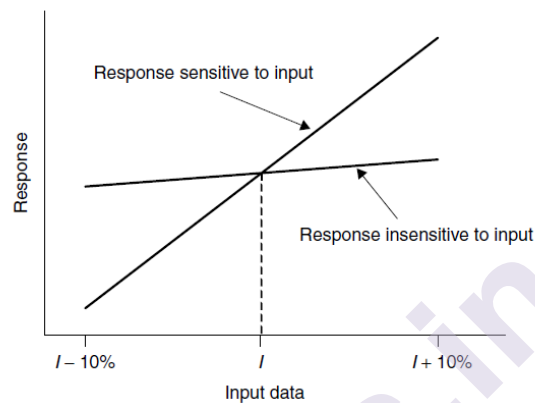
Figure 1 : Sensitivity Analysis

The input (I) is varied, the simulation run and the effect on the response is measured. If there is a significant shift in the response (the gradient is steep), then the response is sensitive to the change in the input. If there is little change (the gradient is shallow), then the response is insensitive to the change.

Sensitivity analysis is beneficial in three main areas:

- Assessing the effect of uncertainties in the data.

- Understanding how variations to the experimental factors affect the responses.

- Assessing the robustness of the solution.

The key method to performing sensitivity analysis is to vary the model inputs, run the simulation and record the change in the responses. This can be a very time consuming process, particularly if there are many model inputs. Hence , sensitivity analysis should be limited to a few key inputs, which might be identified as those about which there is greatest uncertainty and which it is believed have the greatest impact on the response. In addition to this, experimental design and metamodelling methods can be beneficial in helping to achieve and speed up sensitivity analysis (Kleijnen 1998; Noordegraaf *et al*. 2003). Perturbation analysis tries to predict the sensitivity of the results from a single run of a simulation model (Glasserman 1991). Because the simulation does not have to be run repeatedly this should save time.

4.7 VERIFICATION, VALIDATION AND CONFIDENCE

Verification and validation of computer simulation models is carried out during the development of a simulation model with the final goal of producing an accurate and credible model.Verification and validation intends to determine the accuracy with which the model predicts the performance of the real system.

4.7.1 Definition

Verification is the process of guaranteeing that the model design (conceptual model) has been converted into a computer model with sufficient accuracy (Davis 1992). On the other hand ,Validation is the procedure of ensuring that the model is sufficiently accurate for the purpose at hand (Carson 1986).

There are two main concepts in validation:

- *sufficient accuracy* and

- models that are constructed for a specific *purpose*.

The main aim in verification and validation is to guarantee that the model is adequately accurate. Additionally, this accuracy is with mention to the

purpose for which the model is to be used. As a result, the purpose, or objectives, of a model should be known before it can be validated. This purpose may have been determined at the commencement of the simulation study, being expressed through the objectives or it may be an alternative use for an existing model.

Concepts of validity and accuracy are related but are separate.While accuracy is measured on a scale of zero to 100%,validity is a binary decision.
Various types of validation can be defined as follows:

- *Conceptual Model Validation*: Finding that the content, assumptions and simplifications of the proposed model are sufficiently accurate for the purpose at hand.

- *Data Validation*: determining that the contextual data and the data required for model realization and validation are sufficiently accurate for the purpose at hand.

- *White-Box Validation*: determining that the essential parts of the computer model signify the corresponding real world elements with sufficient accuracy for the purpose at hand.

- *Black-Box Validation*: determining that the overall model represents the real world with sufficient accuracy for the purpose at hand. This is an overall, or macro, check of the model's operation.

- *Experimentation Validation*: determining that the experimental procedures accepted are providing results that are satisfactorily accurate for the purpose at hand.

- *Solution Validation*: determining that the results obtained from the model of the proposed solution are sufficiently accurate for the purpose at hand. Similar to black-box validation in that it involves a comparison with the real world. The difference is that only compares the final model of the proposed solution to the implemented solution.

  verification and validation is a continuous process that is executed throughout the life-cycle of a simulation study. Just as modelling is an iterative process, so too is verification and validation.

- At an primary stage in project of simulation a conceptual model is established. Here the model should be validated.

- As the project evolves the conceptual model is possibly revised as the understanding of the problem and the modelling requirements change.

- As a result, the conceptual model also needs to be revalidated.

- -While the conceptual model is being transformed into a computer model, the constituent parts of the model should be continuously verified. Similarly, the details of the model should be checked against the real world throughout model coding (white-box validation). Black-box validation requires a completed model, since it makes little sense to compare the overall model against the real world until it is complete.

- The identification of model errors and constant changes to the conceptual model demands model revisions and therefore further black-box validation.

4.7.2 The Dfficulties Of Verification And Validation

Before we discusse precise methods of verification and validation it is important to know that there are a number of problems that result when we try to validate a model.

- There is no such thing as general validity

  o A model is only validated with respect to its purpose. It cannot be expected that a model that is valid for one purpose is also valid for another.

- There may be no real world to compare against

  - Considerable validation needs a comparison of the model to the real system. There is no real world to use for comparison. Even if the model is of an existing system, its purpose is to investigate alternative operating practices, for which again no real world exists. The model may be shown to be valid when it is representing the existing operation, but this does not guarantee that it is valid once it represents some change to the system.

- Which real world?

  - o Different people have different interpretations of the real world, as described by Weltanschauung or world views by Checkland (1981). An employee in a bank may see the bank as a means for earning money, while a customer may see it as a means for safely storingmoney, or as a means for borrowing money. This presents a problem when validating models. If people have different world views,which interpretation(s) should be used for developing and validating a model? A model that is valid to one person may not be valid to another.

- Often the real world data are inaccurate

  - o Validation often involves a comparison of some facet of the model, for instance throughput,against real world data. The model is run under the same conditions as the real world to see if it performs in a similar manner. There are two difficulties that arise with this procedure.

- First, the real world data may not be accurate. Indeed, the purpose of data validation is to determine the accuracy of the data that are being used. If the data are not accurate,however, this creates problems in determining whether a model's results are correct.

- Secondly, even if ''accurate'' real world data do exist, it must be remembered that these are only a sample, which in itself creates inaccuracy. For instance, data may have been collected on the throughput of a production facility over a 10-week period.

- There is not enough time to verify and validate everything

  - o There is purely not enough time to verify and validate every aspect of a model (Balci 1997). This is a problem that affects both verification and validation.The modeller's job is to ensure that as much of the model is verified and validated as possible, both in terms of the model details (conceptual model validity, verification, whitebox validation and data validation), the overall validity (black-box valid-ation) and the experimental procedures (experimentation validation).

- Confidence not validity

  - Though, in theory, a model is either valid or it is not, proving this in practice is a very different matter. It is not possible to prove that a model is valid. Instead, it is only imaginable to think in terms of the confidence that can be placed in a model. The process of verification and validation is not one of trying to demonstrate that the model is correct, but is in fact a process of trying to prove that the model is incorrect.

4.7.3 The Methods uf Verification and Validation

Following are some useful methods of verification and validation existing to simulation modellers.

1 Conceptual model validation

For validating a conceptual model there are no official methods. The project specification is the major means available for determining what confidence should be placed in the model. The specification should be distributed among those who have a detailed knowledge of the system and feedback must be obtained on whether the model is appropriate.It is also useful that the modeller and the clients together assess the assumptions and simplifications for the level of confidence that can be placed in them and their probable impact on the accuracy of the model. Those assumptions and simplifications about which there is little confidence, and which it is believed have a high impact, need to be addressed. One approach is to eliminate them by altering the model or investigating the real system further. Iit is not possible to remove them, sensitivity analysis can be performed later in the project to determine their impact.

2 Data validation

obviously data are possible source of inaccuracy in a simulation model and can move a model from being sufficiently accurate to being invalid. Effort

should be made to ensure that the data are as accurate as possible. The data should be analysed for inconsistencies and any cause for concern investigated.

3 Verification and white-box validation

Although verification and white-box validation are conceptually different, they are treated together here because they are both performed continuously throughout model coding.

- Verification ensures that the model is true to the conceptual model, while white-box validation ensures that the content of the model is true to the real world (in this way it is an indirect form of conceptual model validation).

- Verification can be performed by the modeller alone,comparing the computer model with the conceptual model description. In the meantime, whitebox validation requires the involvement of those knowledgeable about the real world system.

- Whereas verification can be performed almost continuously during model coding,white-box validation is performed less frequently since it requires the involvement of more than just the modeller.

    Various aspects of the model should be checked during model coding:

- Timings, e.g. cycle times, repair times and travel times.

- Control of elements, e.g. breakdown frequency and shift patterns.

- Control of flows, e.g. routing.

- Control logic, e.g. scheduling and stock replenishment.

- Distribution sampling, e.g. the samples obtained from an empirical distribution.

Three methods of verification and white-box validation are :

- *Checking the code*

    The modeller needs to read through the code to ensure that the right data and logic have been entered. We can get someone else to read the code, or to explain the code to someone else as a second check. If no modelling experts are available, then most simulation software vendors offer a help-desk service with which specific areas of code could be discussed.

- *Visual checks*

    The visual display of the model proves to be a powerful support for verification and validation.By running the model and watching how each element behaves both the logic of the model and the behaviour against the real world can be checked. Various ideas in this approach are :

- Stepping through the model event by event.

- Stopping the model, predicting what will happen next, running the model on and checking what happens.

- Interactively setting up conditions to force certain events to take place.

- Creating extreme conditions, such as a very high arrival rate, to determine whether the model behaves as expected

- Isolating areas of the model so it runs faster, reducing the time to perform thorough verification and validation.

- Explaining the model as it runs to those knowledgeable about the real system in order to gain their opinion.

- Tracing the progress of an item through the model.

- *Inspecting output reports*

By reviewing the reports from a simulation run,comparison can be done of the actual and expected results. In verification and white-box validation of interest is the performance of the individual elements. Graphical reports of samples from input distributions, for instance, machine repair times, are an aid in checking that they are being modelled correctly. A report which may be of some use is a 'trace' of a simulation run.

4 Black-box validation

In black-box validation the complete behaviour of the model is considered. There are two broad approaches to performing this form of validation.

- The first is to compare the simulation model to the real world.
- The other is to make a comparison with another model.

The second approach is particularly useful when there are no real world data to compare against.

5 Experimentation validation

Guaranteeing the accuracy of simulation experiments requires consideration to the issues like initial transient effects, run-length, the number of replications and sensitivity analysis. Also, the search of the solution space should be sufficient to obtain an adequate understanding and identify appropriate solutions.

6 Solution validation

The goal of all modelling and verification and validation is to try and assure the validity of the final solution. Once implemented, it should be possible to validate the implemented solution against the model's results. This is similar in concept to the comparisons with the real world performed in black-box validation, except that the comparison is between the final model of the proposed solution and the implemented solution.

4.7.4 Independent Verification, Validation

Independent verification and validation (IV&V) or verification, validation and accreditation (VV&A) includes an independent third party whose aim is to determine whether a model is suitable for a particular use.

Gass (1983) defines model assessment (or evaluation) as ''a process by which interested parties (who were not involved in a model's origins, development and implementation) can determine, with some level of confidence, whether or not the model's results can be used in decision-making''. He considers that model valuation is essential in three circumstances:

- When the decision-makers are far removed from the process of developing the model.
- When the model is to be applied to a new set of situations other than that originally planned.
- Even if the decision-makers work closely with the analysts during model development, it is unlikely that they have the necessary knowledge and skills to evaluate the model.

Typically, independent verification and validation is only carried out for large-scale military and public policy models, probably because the costs of the process are prohibitive for most manufacturing and service sector projects which tend to be smaller in scale (Cochran *et al*.1995). In 1977, Gass suggests that in selecting a model for major evaluation it should have involved an expenditure of over $250,000 and more than five person-years

of effort (Gass 1977). Even for large-scale models, independent verification and validation is not always common practice (Arthur and Nance 1996).

A complete range of procedures for independently assessing simulation models have been planned over the years by several authors .Most of this criterias involve model verification and validation, although other factors such as documentation and training are also considered to be important. For example Gass and Joel (1981) use seven criteria:

- Model definition

- Model structure

- Model data

- Computer model verification

- Model validation

- Model usability

- Model pedigree

For each of the criteria either a subjective score is given (e.g. on a scale of 1–5) or a set of qualitative statements is made. Where subjective scores are given, then some overall score can be calculated, possibly taking into account the importance of each criteria (Gass and Joel 1981). The overall score indicates the level of confidence that can be placed in the model for its intended purpose. Balci *et al*. (2002) describe software that aids the evaluation process.

4.8 SUMMARY

In this chapter we discussed how simulation experiments are performed and how the results should be reported. Methods for comparing alternative scenarios are described and approaches for searching the solution space are discussed. Some of the key areas and methods that are identified are The nature of simulation experimentation,The analysis of results,Comparison of alternatives,Informal search experimentation,Formal search experimentation and Sensitivity analysis .The experimental methods described focus on finding a solution and on the statistical significance of the results. The proper use of experimental methods, if used together with these wider considerations, must only enhance this process.

Model verification and validation is concerned with creating enough confidence in a model for the results to be accepted. This is done by trying to prove that the model is incorrect. The more tests that are performed in which it cannot be proved that the model is incorrect, the more confidence in the model is increased. For verification and validation the general rule is: the more testing the better.

Lastly, the modeller should recollect that the approval of a simulation study and its results does not depend exclusively on the validity of the model. Verification and validation assures (content) quality in the sense that the model conforms to the clients' technical requirements for a model and a set of results that are sufficiently accurate

4.9 EXERCISE

1. Compare and contrast the difficulties that might be encountered in validating a simulation model of:

   a) An existing manufacturing plant

   b) A unique construction project

2. Carry out some verification and validation tests with the bank model.

3. Explain nature of simulation

4. Explain verificationand validation methods

5. Write a note on sensitivity analysis.

4.10 REFERENCES

- Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004.

- The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013.

- Agent Based Modeling and Simulation, Taylor S, 2014.

- Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003.

- Object Oriented Simulation: A Modeling and Programming Perspective, Garrido, José M, 2009.

- Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003.

- Process Control: Modeling, Design and Simulation, Wayne Bequette W, Prentice Hall of India, 2003.

❄❄❄❄❄❄❄

5

MODELING AND SIMULATION MODELLING: PART 1

Unit Structure :

5.1   Introduction:

5.2   Objectives:

5.3   Types of models

5.4   Application of simulation modeling

5.5   Level of abstraction

5.6   Simulation Modeling Methods

5.7   System Dynamics

5.8   Discrete Event Modeling

5.9   Conclusion

5.10 Practice Question

5.11 References

5.1 INTRODUCTION:

Simulation in general is to pretend that one deals with a real thing while really working with an imitation. In operations research the imitation is a computer model of the simulated reality. A flight simulator on a PC is also a computer model of some aspects of the flight: it shows on the screen the controls and what the "pilot" (the youngster who operates it) is supposed to see from the "cockpit" (his armchair).

☐   Gain greater understanding of a process

☐   Identify problem areas or bottlenecks in processes

☐   Evaluate effect of systems or process changes such as demand, resources, supply, and constraints

☐   Identify actions needed upstream or downstream relative to a given operation, organization, or activity to either improve or mitigate processes or events

☐   Evaluate impact of changes in policy prior to implementation

Why to use models? To fly a simulator is safer and cheaper than the real airplane. For precisely this reason, models are used in industry commerce and military: it is very costly, dangerous and often impossible to make experiments with real systems. Provided that models are adequate descriptions of reality (they are valid), experimenting with them can save money, suffering and even time.

By *method* in simulation modeling, we mean a general framework for mapping a real-world system to its model. A method suggests a type of language, or "terms and conditions" for model building. The choice of method should be based on the system being modeled and the purpose of the modeling – though often it is most heavily influenced by the background or available tool set of the modeler. At a higher level, it is desirable that the engine supports:

- A large number of concurrent activities, including their dynamic creation and destruction.

- Correct handling of multiple instantaneous events, in particular deterministic and random execution. This is important for synchronous models.

- Networks and communication.

- 2D, 3D, and geographical space, and space-related functionality.

5.2 OBJECTIVES:

☐ To understand different modelling techniques used in simulation

☐ Comparative study of different models

☐ To learn important aspects of model building

5.3 TYPES OF MODELS: ANALYTICAL VS. SIMULATION MODELING

There are many different types of models and associated **modeling languages** to address different aspects of a system and different types of systems. Since different models serve different **purposes**, a classification of models can be useful for selecting the right type of model for the intended purpose and **scope**.

An **analytical model** describes mathematical relationships, such as differential equations that support quantifiable analysis about the system parameters. Analytical models can be further classified into dynamic and static models. Dynamic models describe the time-varying state of a system, whereas static models perform computations that do not represent the time-varying state of a system. A dynamic model may represent the performance of a system, such as the aircraft position, velocity, acceleration, and fuel consumption over time. A static model may represent the mass properties estimate or **reliability** prediction of a system or component.

The term **simulation**, or more specifically **computer simulation**, refers to a method for implementing a model over time. The computer simulation includes the analytical model which is represented in executable code, the **input** conditions and other input data, and the computing infrastructure. The computing infrastructure includes the computational engine needed to execute the model, as well as input and **output** devices. The great variety

of approaches to computer simulation is apparent from the choices that the designer of a computer simulation must make, which include:

- stochastic or deterministic;

- steady-state or dynamic;

- continuous or discrete; and

- local or distributed.

Other classifications of a simulation may depend on the type of model that is being simulated. One example is an agent-based simulation that simulates the interaction among autonomous agents to predict **complex emergent** behavior (Barry 2009). There are many other types of models that could be used to further classify simulations. In general, simulations provide a means for analyzing complex dynamic behavior of systems, software, hardware, people, and physical phenomena.

Simulations are often integrated with the actual hardware, software, and operators of the system to evaluate how actual components and users of the system perform in a simulated **environment**. Within the United States defense community, it is common to refer to simulations as live, virtual, or constructive, where live simulation refers to live operators operating real systems, virtual simulation refers to live operators operating simulated systems, and constructive simulations refers to simulated operators operating with simulated systems. The virtual and constructive simulations may also include actual system hardware and software in the loop as well as stimulus from a real systems environment.

5.4 APPLICATION OF SIMULATION MODELING

Modelling & Simulation can be applied to the following areas − Military applications, training & support, designing semiconductors, telecommunications, civil engineering designs & presentations, and E-business models.

Additionally, it is used to study the internal structure of a complex system such as the biological system. It is used while optimizing the system design such as routing algorithm, assembly line, etc. It is used to test new designs and policies. It is used to verify analytic solutions. Some of the area of application is as shown below:

1.   Modelling in Engineering and Sciences

2.   Applications of Simulation

3.   Modelling and Simulation tools

4.   Big Data Simulation and IoT

5.   High Performance Computing and Network Simulation

6.   Artificial Intelligence and Machine Learning

7. Choosing drilling projects for oil and natural gas

8. Evaluating environmental impacts of a new highway or industrial plant

9. Setting stock levels to meet fluctuating demand at retail stores

10. Forecasting sales and production requirements for a new drug

11. Planning aircraft sorties and ship movements in the military

12. Planning for retirement, given expenses and investment performance

13. Deciding on reservations and overbooking policies for an airline

14. Selecting projects with uncertain payoffs in capital budgetin

5.5 LEVEL OF ABSTRACTION

Abstraction level of complex simulation models such as large manufacturing systems is always a critical factor in simulation projects. It not only helps define boundaries of a simulation model but also defines the complexity and resource requirements for the model. Many a times a simple looking model grows into a complex model because of incorrect choices in abstraction level. Developing the model in stages or steps of abstraction is sometimes a favored approach. In this paper we study and analyze 'why' and 'how' these choices in abstraction level of a simulation model at various stages in a project's life cycle results in answering the objective function more precisely. Selection of abstraction levels should be based on project objectives and should not purely depend on the system itself.
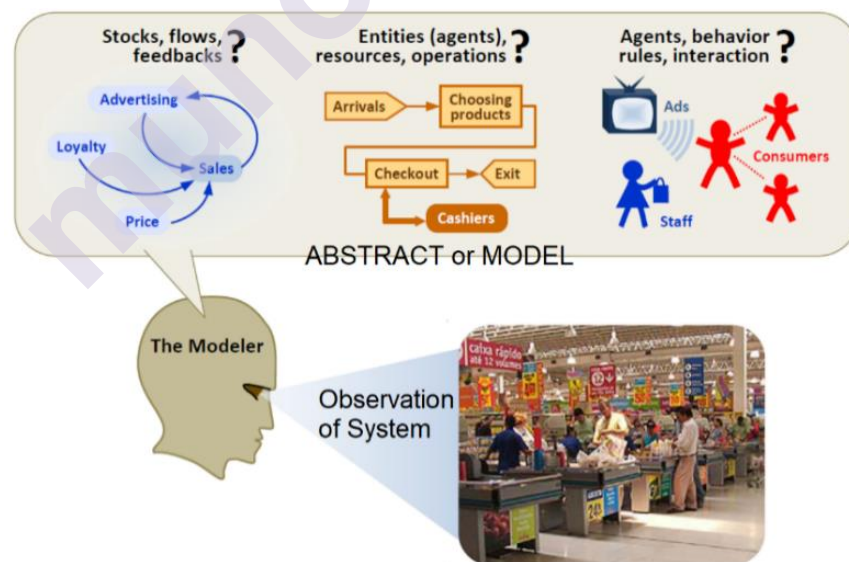
The level of abstraction of a model **determines the amount of information that is contained in the model**. The quantity of information in a model decreases with the lowering levels of abstraction. Thus a „low level abstraction" model contains more information than a „high level abstraction" model. The significance of abstraction is further amplified by pressures on both time and costs of projects and does not allow the use of a „safe" abstraction level that would have more than required details in the

model. It hence becomes vital to model at the highest possible abstraction level that does not compromise in any way on the accuracy of outputs or ensuing decisions. That said, there is a need for models and modelers to be flexible with abstraction levels during projects.

- Selection of abstraction levels should be based on project objectives and should not purely depend on the system itself

- Modelers should note adopt a safe-bet approach of putting more detail into the model than what is required. Models should be built at the highest level of abstraction possible without compromising accuracy.

- Analysis of high-level model results and TOC based tools can help in identify areas for adding model fidelity

- Changes to model efficiencies should be considered right from the conceptual phases when moving to low abstraction level models

5.6 SIMULATION MODELING METHODS

A system can be classified into the following categories.

- **Discrete-Event Simulation Model** − In this model, the state variable values change only at some discrete points in time where the events occur. Events will only occur at the defined activity time and delays.

- **Stochastic vs. Deterministic Systems** − Stochastic systems are not affected by randomness and their output is not a random variable, whereas deterministic systems are affected by randomness and their output is a random variable.

- **Static vs. Dynamic Simulation** − Static simulation include models which are not affected with time. For example: Monte Carlo Model. Dynamic Simulation include models which are affected with time.

- **Discrete vs. Continuous Systems** − Discrete system is affected by the state variable changes at a discrete point of time. Its behavior is depicted in the following graphical representation.
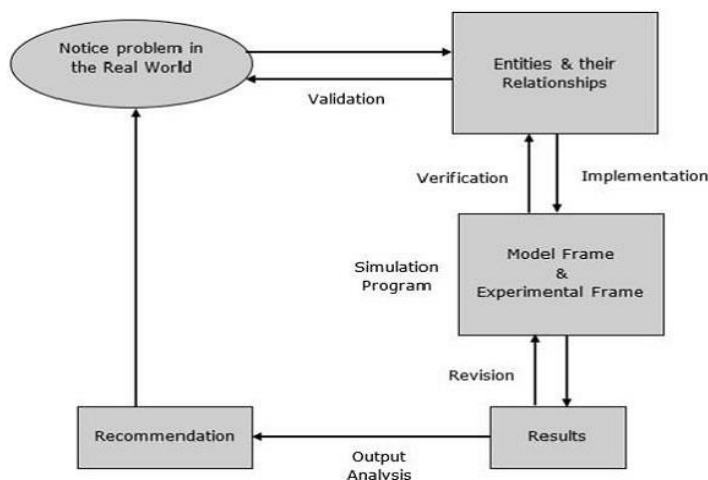
Step 1 − Examine the problem. In this stage, we must understand the problem and choose its classification accordingly, such as deterministic or stochastic.

Step 2 − Design a model. In this stage, we have to perform the following simple tasks which help us design a model −

• Collect data as per the system behavior and future requirements.

• Analyze the system features, its assumptions and necessary actions to be taken to make the model successful.

• Determine the variable names, functions, its units, relationships, and their applications used in the model.

• Solve the model using a suitable technique and verify the result using verification methods. Next, validate the result.

• Prepare a report which includes results, interpretations, conclusion, and suggestions.

Step 3 − Provide recommendations after completing the entire process related to the model. It includes investment, resources, algorithms, techniques, etc.

5.7 SYSTEM DYNAMICS

System dynamics is a method created in the mid-1950s by MIT Professor Jay Forrester, whose original background was in science and engineering. Forrester's idea was to use the laws of physics, in particular the laws of electrical circuits, to describe and investigate the dynamics of economic and, later on, social systems. The principles and the modeling language of system dynamics were formed in the 1950s and early 1960s, and remain unchanged today.

System dynamics is a method of studying dynamic systems. It suggests that you should:

• Take an endogenous point of view. Model the system as a causally closed structure that itself defines its behavior.

• Discover the feedback loops (circular causality) in the system. Feedback loops are the heart of system dynamics.

• Identify stocks (accumulations) and the flows that affect them. Stocks are the memory of the system, and sources of disequilibrium.

• See things from a certain perspective. Consider individual events and decisions as "surface phenomena that ride on an underlying tide of system structure and behavior." Take a continuous view where events and decisions are blurred.

To understand the essence of system dynamics, consider a shop with a counterman serving the shop's clients. The more people come to the shop per hour, the longer the queue grows. You can build a discrete event model

that will give you the length of the queue as a function of the clients' arrival rate and the service time. However, in a real shop, as the queue grows longer, some clients may decide not to join the queue, and instead leave the shop. Others may decide to leave the queue after having waited longer than they expected to. In other words, the length of the queue feeds back to inhibit the rate of queue growth. The results of the "straightforward" model (sometimes called open-loop models in the system dynamics community), will not be valid unless it addresses these circular causal dependencies. One of the key advantages of the system dynamics approach is to readily and elegantly identify such feedback loops and include them into the model.

Consider a company that starts selling a new consumer product. The addressable market has a known size, which does not change over time. Consumers are sensitive to both advertising and word of mouth. The product has an unlimited lifetime and does not need replacement or repeated purchases. A consumer needs only one product. We are to forecast the sales dynamics.

We will start with identifying the key variables in our model, and will iteratively draw *causal loop diagrams*. In a causal loop diagram, variables are connected by arrows showing the causal influences among them, with important feedback loops explicitly identified. In our system, one of the variables is obviously **Sales** – the number of people who bought our product per time unit, e.g. per week. The number of **Potential Clients** will be the other variable. The bigger the market, the greater the sales; therefore, we can draw a causal dependency from **PotentialClients** to **Sales** with positive polarity. On the other hand, as potential clients buy the product, they stop being potential clients, so there is another influence from **Sales** back to **PotentialClients**, this time with negative polarity. The feedback loop we have just created is a *negative*, or *balancing feedback loop*: it works for reaching a certain goal. In our case, we ultimately will sell the product to all potential clients, and both variables will become zero.

A, B, C: Causal loop diagrams
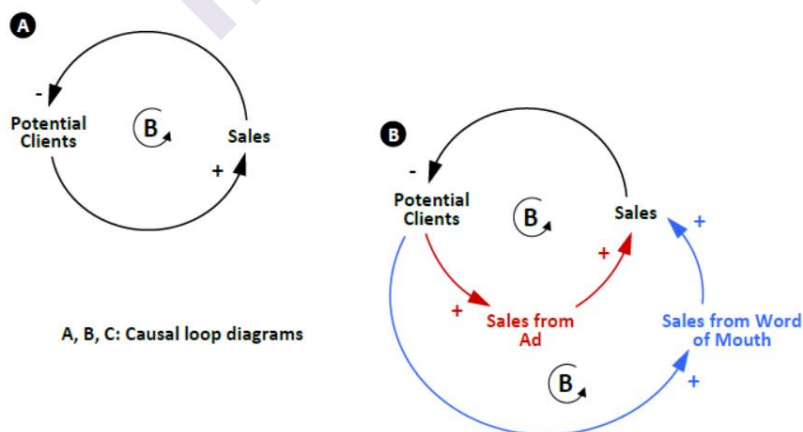
What determines the sales rate? According to our assumptions, consumers are sensitive to ads and to what other consumers say. So, we will distinguish between sales from advertising, and sales from word of mouth. We

introduce two new variables and create two balancing loops instead of one, see Figure B.
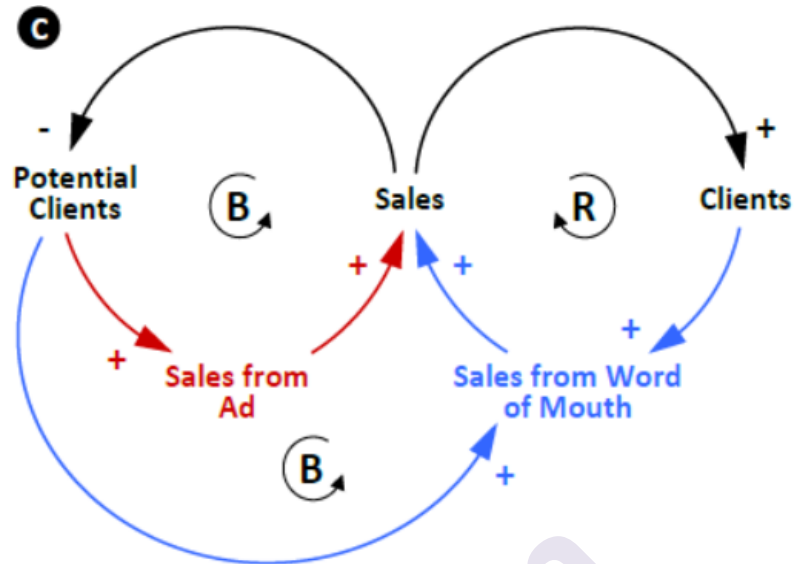
The **SalesFromWordofMouth** depend on the number of (hopefully happy) owners of our product – our **Clients**. The number of clients grows with **Sales**. We draw another feedback loop, this time *positive*, or *reinforcing,* see Figure C.
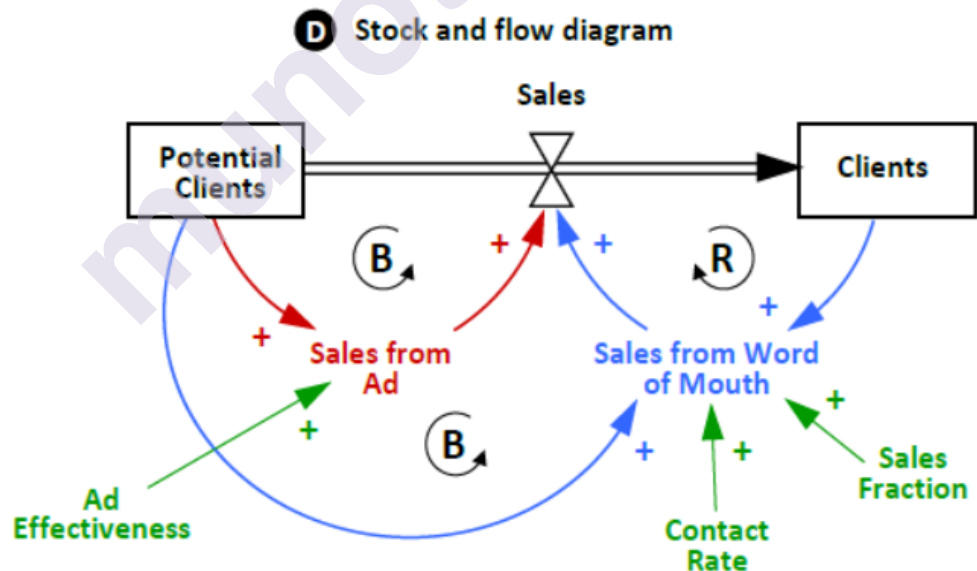
While the causal loop diagram we have drawn shows variable interdependencies and feedbacks, it misses the clear mathematical interpretation, and therefore cannot be simulated directly. One of the things that we need to do on our way to the mathematical model is to identify *stocks and flows* among the variables in our system. Stocks are accumulations, and characterize the state of the system. Flows are the rates at which these system states change. Units of measure can help identify stocks and flows. Stocks are usually quantities such as people, inventory,

money, and knowledge. Flows are measured in the same units per time period; for example, clients per month, or dollars per year.

In our model, the stocks are **PotentialClients** and **Clients**, and the flow between them is **Sales**. We can now draw a *stock and flow diagram* and write equations for our model. The diagram is shown in Figure 2.2, D. The equations behind that diagram are:

$$\frac{d(PotentialClients)}{dt} = -Sales$$

$$\frac{d(Clients)}{dt} = Sales$$

$$Sales = SalesFromAd + SalesFromWordofMouth$$

The first two equations are *differential equations*. They define how the stock values change over time. For example, the number of **Clients** grows at the **Sales** rate. The third equation tells that the sales rate consists of two sources, and those sources are independent. The equations for those sources, however, are not clear from the causal loop diagram, and we need to make more assumptions in order to define them.

Underlying mathematics and simulation engine

Mathematically, a system dynamics model is a system of coupled, nonlinear, first-order differential equations

$$\frac{d(X)}{dt} = F(X,P)$$

where is a vector of stocks, is a set of parameters, and is a nonlinear vector-valued function. Simulation of system dynamics models is done with *numerical methods* that partition simulated time into discrete intervals of length *dt* and step the system through time one *dt* at a time. *X P F*

While numerical methods may be very sophisticated in the modeling tools used by natural scientists and engineers (especially the ones with the adaptive variable time step), the numerical methods used in system dynamics are simple, fixed-step methods: Euler and Runge-Kutta. In addition to differential equations, the simulation engine must be able to solve algebraic equations that appear in the models with *algebraic loops*.

Unlike discrete event and agent-based models, system dynamics models are deterministic, unless stochastic elements are explicitly inserted into them.

5.8 DISCRETE EVENT MODELING

Discrete event modeling is almost as old as system dynamics. In October 1961, IBM engineer Geoffrey Gordon introduced the first version of GPSS (General Purpose Simulation System, originally Gordon's Programmable Simulation System), which is considered to be the first method of software implementation of discrete event modeling. These days, discrete event modeling is supported by a large number of software tools, including modern versions of GPSS itself.

The idea of discrete event modeling method is this: the modeler considers the system being modeled as a process, i.e. a sequence of operations being performed across entities.

The operations include delays, service by various resources, choosing the process branch, splitting, combining, and some others. Since entities compete for resources and can be delayed, queues are present in virtually any discrete event model. The model is specified graphically as a process flowchart, where blocks represent operations (there are textual languages as well, but they are in the minority). The flowchart usually begins with "source" blocks that generate entities and inject them into the process, and ends with "sink" blocks that remove entities from the model. This type of diagram is familiar to the business world as a process diagram and is ubiquitous in describing their process steps. This familiarity is one of the reasons why discrete event modeling has been the most successful method in penetrating the business community.

The *entities* (originally in GPSS they were called *transactions*) that are flowing through the process flowchart, are actually *agents. Agents* may represent clients, patients, phone calls, documents (physical and electronic), parts, products, pallets, computer transactions, vehicles, tasks, projects, and ideas.

Resources represent various staff, doctors, operators, workers, servers, CPUs, computer memory, equipment, and transport.

Service times, as well as agent arrival times, are usually stochastic, drawn from a probability distribution. Therefore, discrete event models are stochastic themselves. This means that a model must be run for a certain time, and/or needs a certain number of replications, before it produces a meaningful output.

The typical output expected from a discrete event model is:

- Utilization of resources,
- Time spent in the system or its part by an agent,
- Waiting times,
- Queue lengths,
- System throughput,
- Bottlenecks,
- Cost of the agent processing and its structure.

Consider a bank with an ATM inside. The process in the bank is described as follows:

- ☐☐ On average, 45 clients per hour enter the bank.

- Having entered the bank, half of the clients go to the ATM, and the other half go straight to the cashiers.

- Usage of the ATM has a minimum duration of 1 minute, a maximum of 4 minutes, and a most likely duration of 2 minutes.

- ☐☐ Service with a cashier takes a minimum of 3 minutes and a maximum of 20 minutes, with a most likely duration of 5 minutes.

- After using the ATM, 30% of the clients go to the cashiers. The others exit the bank.

- There are 5 cashiers in the bank, and there is a single shared queue for all the cashiers.

- After being served by a cashier, clients exit the bank.

We need to find out the:

- Utilization of cashiers,

- Average queue lengths, both to the ATM and to the cashiers, and the

- Distribution of time spent by a customer in the bank.

With this problem definition, building a discrete event model is more or less a straightforward task. Clients obviously are agents, and the cashiers are resources. The flowchart of the bank is shown in Figure 2.4. The block **ClientsArrive** generates clients at the rate of 0.75 per minute (45 per hour). Having appeared in the model, 50% of the clients go to the cashiers, and 50% to the ATM. The usage of the ATM is modeled by the **Delay** block **ServiceAtATM**, preceded by the **Queue** block. Service at cashiers is modeled by a pair of blocks: **Service** with triangularly distributed service time and **ResourcePool Cashiers** with capacity 5. The flowchart ends with the **Sink** block **ClientsLeave**.

The output data is generated as the model is running. Statistics are collected at the blocks, as well as by the agents while they move through the process flowchart. The data (for example, the cashiers' utilization) can be observed on-the-fly via inspect windows and is stored in the model execution logs, which can be accessed after the model run is finished. Each agent (client) measures time spent in the bank by making a timestamp at the entry and then comparing it with the current time at the exit.

The mathematics behind discrete event simulation are based on *discrete time.* The model clock is advanced only when something significant happens in the model –namely, when an agent starts or finishes an operation. Any change in the model is associated with those events; continuous changes are approximated by instantaneous ones.

Example:

Most business processes can be described as a sequence of separate discrete events. For example, a truck arrives at a warehouse, goes to an unloading

gate, unloads, and then departs. To simulate this, discrete-event simulation is often chosen.

Using discrete-event simulation modeling, the movement of a train from point A to point B is modeled with two events, namely a departure and an arrival. The actual movement of the train would be modeled as a time delay between the departure and arrival events. These events and movement between them can be smoothly animated.

Discrete-event simulation focuses on the processes in a system at a medium level of abstraction. Typically, specific physical details, such as car geometry or train acceleration, are not represented. Discrete-event simulation modeling is widely used in the manufacturing, logistics, and healthcare fields.

Example:

Three callers problem

Problem Definition:

Two lines services three callers. Each caller makes calls that are exponentially distributed in length, with mean $1/^1$. If both lines are in service by two callers and the third one requests service, the third caller will be blocked. A caller whose previous attempt to make a call was successful has an exponentially distributed time before attempting the next call, with rate $,$. A caller whose previous call attempt was blocked is impatient and tries to call again at twice that rate $(2,)$, also according to exponential distribution. The callers make their calls independent of one another.

Define the following six states:

0   no calls in progress, 3 callers idle
1   1 call in progress, 2 callers idle
2   2 calls in progress, 1 caller idle
3   2 calls in progress, 1 caller impatient
4   1 call in progress, 1 caller impatient
5   0 calls in progress, 1 caller impatient

The state transition diagram is
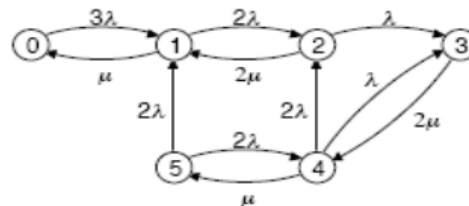
The rate generator matrix is

$$
\underline{\underline{Q}} =
\begin{bmatrix}
-3\lambda & 3\lambda & 0 & 0 & 0 & 0 \\
\mu & -\mu-2\lambda & 2\lambda & 0 & 0 & 0 \\
0 & 2\mu & -2\mu-\lambda & \lambda & 0 & 0 \\
0 & 0 & 0 & -2\mu & 2\mu & 0 \\
0 & 0 & 2\lambda & \lambda & -\mu-3\lambda & \mu \\
0 & 2\lambda & 0 & 0 & 2\lambda & -4\lambda
\end{bmatrix}
$$

5.9 CONCLUSION

Simulation and modelling is used to representing the system and its environment, the simulation must provide efficient computational methods for solving the equations. Simulations may be required to operate in real time, particularly if there is an operator in the loop. Other simulations may be required to operate much faster than real time and perform thousands of simulation runs to provide statistically valid simulation results. Several computational and other simulation methods are described in *Simulation Modeling and Analysis*.

5.10 PRACTICE QUESTION

1. Why to use models in system study?
2. List the different types of Models.
3. Explain the area of application of Simulation.
4. What do you mean by level of abstraction in simulation? Explain.
5. Explain the Simulation Modelling Methods.
6. Write a short note on following classification of:
   • Discrete-Event Simulation Model.
   • Stochastic vs. Deterministic Systems.
   • Static vs. Dynamic Simulation.
   • Discrete vs. Continuous Systems
7. Write a short note on system dynamic method of modeling using suitable example.
8. Demonstrate the use of discrete event simulation method.

5.11 REFERENCES

☐ Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004.

☐ The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013.

☐ Agent Based Modeling and Simulation, Taylor S, 2014.

☐ Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003.

☐ Object Oriented Simulation: A Modeling and Programming Perspective, Garrido, José M, 2009.

☐ Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003.

☐ Process Control: Modeling, Design and Simulation, Wayne Bequette W, Prentice Hall of India, 2003.

❊❊❊❊❊❊❊

6

MODELING AND SIMULATION MODELLING: PART 2

Unit Structure :

6.1 INTRODUCTION

Agent-based models (ABM) are used to analyze the influence of the behavior of individual agents (farmers, households, consumers) on the emergent properties of the system (e.g., landscape) in a dynamic way. Agent-based modeling is a powerful simulation modeling technique that has seen a number of applications in the last few years, including applications to real-world business problems. After the basic principles of agent-based simulation are briefly introduced, its four areas of application are discussed by using real-world applications: flow simulation, organizational simulation, market simulation, and diffusion simulation.

6.2 OBJECTIVES:

- To understand the working of Agent Based Models(ABM)

- To study the factors influencing the Model creation in ABM

- To study Dynamic System

6.3 AGENT BASED MODELING

6.3.1 Introduction to Agent

Agent-based modeling is a powerful simulation modeling technique that has seen a number of applications in the last few years, including applications to real-world business problems. After the basic principles of agent-based simulation are briefly introduced, its four areas of application are discussed by using real-world applications: flow simulation, organizational simulation, market simulation, and diffusion simulation.

The purpose of agent-based models is to explain system-level properties by the behaviour of interrelated individuals. An animal for instance is not aware of any birth-rates at population-level. Actually, it will successfully give birth to offspring, if it finds a mating partner at due time, if there is enough food around the nest for feeding, etc. The behaviour of a population thus 'emerges' in a self-organised manner from the behaviour of individual animals during their life cycles and the interplay of these individuals with other individuals. System-level properties like population size, birth- and death rates are the result rather than the input. Especially in the Life Sciences, agent-based modelling has gained momentum for its close resemblance of living systems, where the core units of an ecosystem are individual animals and plants.

For each category, one or several business applications are described and analyzed.

Agent-based modeling is a more recent modeling method than system dynamics or discrete event modeling. Until the early 2000s, agent-based modeling was pretty much an academic topic. The adoption of agent-based

modeling by simulation practitioners started in 2002-2003. It was triggered by:

- Desire to get a deeper insight into systems that are not well-captured by traditional modeling approaches;

- Advances in modeling technology coming from computer science, namely object-oriented modeling, UML, and statecharts ;

- Rapid growth of the availability of CPU power and memory (agent-based models are more demanding of both, compared to system dynamics and discrete event models).

6.3.2 Agent-based modeling

Agent-based modeling suggests to the modeler yet another way of looking at the system.

- You may not know how the system as a whole behaves, what are the key variables and dependencies between them, or simply don't see that there is a process flow, but you may have some insight into how the objects in the system behave individually. Therefore, you can start building the model from the bottom up by identifying those objects (agents) and defining their behaviors.

- Sometimes, you can connect the agents to each other and let them interact; other times, you can put them in an environment, which may have its own dynamics. The global behavior of the system then emerges out of many (tens, hundreds, thousands, even millions) concurrent individual behaviors.

There are no standard languages for agent-based modeling. The structure of an agent-based model is created using graphical editors or scripts, depending on the software. The behavior of agents is specified in many different ways. Frequently, the agent has a notion of state, and its actions and reactions depend on its state. In such cases, behavior is best defined with statecharts. Sometimes, behavior is defined in the form of rules executed upon special events. In many cases, the internal dynamics of the agent can be best captured using system dynamics or discrete event approach. In these cases, we can put a stock and flow diagram or a process flowchart inside an agent. Similarly, processes outside of agents and the dynamics of the environment where they live are often naturally modeled using traditional methods. We find that a large percentage of agent-based models, therefore, are multi-method models.

6.3.3 Time in agent based models

Time is sometimes handled as 'just another dimension'. For many cases this pragmatic view is a useful way to think about time. Analogous to the spatial dimensions, the choice of scale and data model types are of decisive importance. However, time has some peculiarities that we need to think about and explicitly address during model design. Unlike space, time has a direction and it is only one-dimensional. This makes the

adequate order of update routines and scheduling of processes an important factor in the model design process. Finally, time often exhibits a cyclic nature that is given by the day / night rhythm and seasonal changes. Therefore, the state of the system a year ago may be more relevant in predicting the upcoming change, than the state two months ago.

There are four aspects in the representation of time that are important from a modelling perspective:

- Temporal data models

- Temporal scale

- Update routines (synchronous / asynchronous)

- Scheduling (Process timing)

Continuous phenomenon discretised into hourly time steps
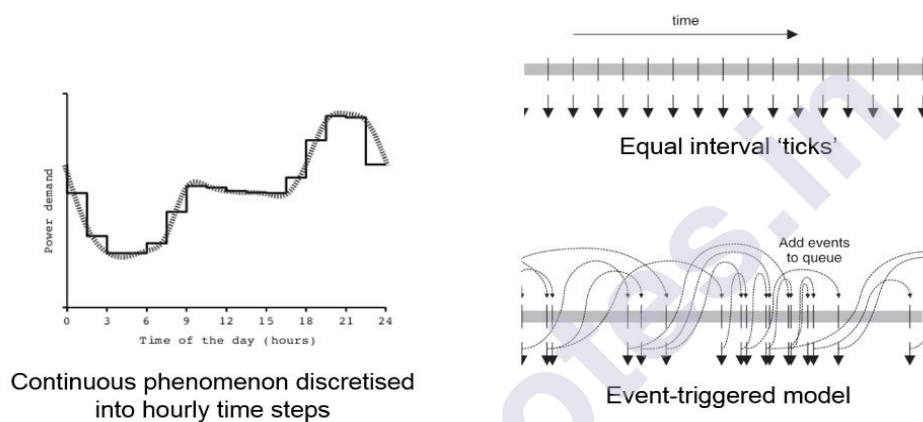
Time is a continuous phenomenon. Nevertheless, to represent time in a digital model, time needs to be discretised. Strictly speaking, even equation-based simulation models need to be discretised for computation. However, the actual parameter value can be computed for any point in time and we can think of these models as continuous models in the context of simulation modelling. In rule-based models (cellular automata and agent-based models) time is conceptualised in discrete time steps.

6.3.4 Space in agent based models

To design a simulation model, we need to think about how we want to model spatial features. On the one side this depends on the phenomenon, we are interested in. On the other side, aspects of performance need to be considered. As GIScientists, we are familiar with the concepts of how to represent geographic space in computer models. From the specific perspective of simulation modelling, four aspects are of particular importance:

- Spatial data models (vector, raster, graphs)

- Scale(s)

- Neighbourhood: Moore and more

- Boundaries: finite, infinite, toroidal

6.3.5 Discrete space

A discrete space is one in which the state variable(s) change only at a discrete set of points in time. The bank is an example of a discrete system: The state variable, the number of customers in the bank, changes only when a customer arrives or when the service provided a customer is completed Eg. how the number of customers changes only at discrete points in time.

Discrete temporal phenomena are termed 'events'. Events can happen regularly, e.g. each morning the sun rises and thus triggers multiple processes: flowers open, birds become active, etc. Such events are well represented in classical cellular automata and agent-based models, where time steps usually are assumed to be regular. However, events are often irregularly paced, e.g. natural catastrophes. Such irregularity of events can be addressed by a fine resolution of discrete steps, where there is only a certain probability that an event happens. For systems that are strongly governed by irregular events, it is probably more adequate to apply an event-based approach, where events trigger the placement of later events in a queue.

6.3.6 Continuous space movement in continuous space

A continuous system is one in which the state variable(s) change continuously over time. An example is the head of water behind a dam. During and for some time after a rain storm, water flows into the lake behind the dam. Water is drawn from the dam for flood control and to make electricity. Evaporation also decreases the water level. Eg. how the state variable head of water behind the dam changes for this continuous system.

Continuous temporal phenomena describe an ongoing process, like temperature change or tree growth. System dynamics are the classical approach to model such phenomena. If we have a strong interest in how the process operates in space, a cellular automaton approach is more adequate. In this case, the continuous process needs to be broken down into time steps that are small enough to adequately represent the dynamic behaviour of the modelled process.

6.3.7 Communication between agents

Interaction is a key aspect in ABM. There is a plethora of definitions for the concept of agent and most of them emphasize the fact that this kind of entity should be able to interact with their environment and with other entities in order to solve problems or simply reach their goals according to coordination, cooperation or competition schemes. The essence of an ABM is the fact that the global system dynamics emerges from the local behaviors and interactions among its composing parts. Strictly speaking, for some kind of ABM the global dynamics is just the sum of local behaviors and interactions, so we cannot always speak of emergent behavior when we talk about ABM. However the assumptions that underlie the design of an interaction model (or the choice of an existing one for the design and implementation of a specific application) are so important that they have a deep impact on the definition of agents themselves (e.g. an interpreter of a

specific language, a perceiver of signals). Therefore it is almost an obvious consequence that interaction mechanisms have a huge impact on the modeling, design and development of applications based on a specific kind of ABM, which in turn is based on a particular interaction model. It is thus not a surprise that a significant part of the research that was carried out in the agent area was focused on this aspect.
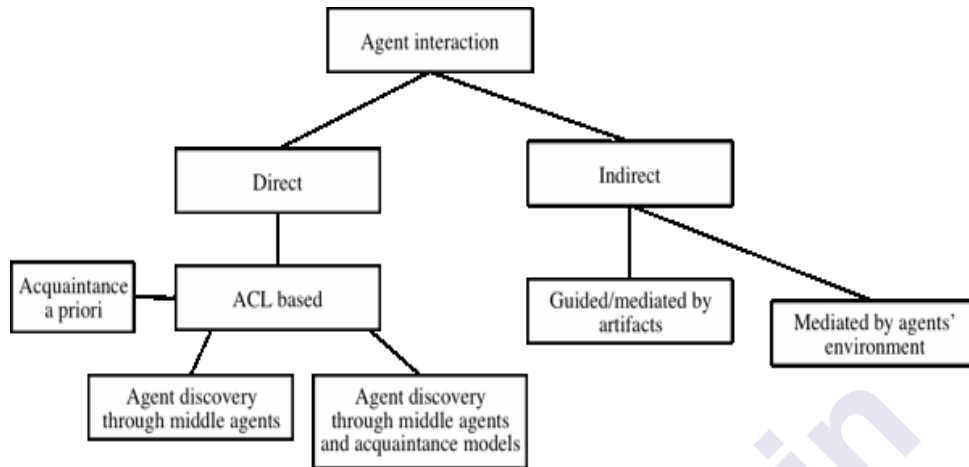
6.3.10 Condition triggered events and transition in agents

Events (low-level constructs that allow to schedule one-time or recurrent action) and statechart transitions are frequent elements of agent behavior. Among other trigger types, both can be triggered by a condition – a Boolean expression. If the model contains dynamic variables, all conditions of events and statechart transitions are evaluated on each integration step, which ensures the event or transition will occur exactly when the (continuously changing) condition turns true. Here event waits on the Money stock to fall below zero. Event and statechart can be located on the same level with the SD, or in a different active object.

A transition is enabled if its source state is active and if any additional enabling condition is true. Only an enabled transition can be executed. Executing a transition changes the active state from the source state of the transition to its destination state. In addition, some actions may be performed.
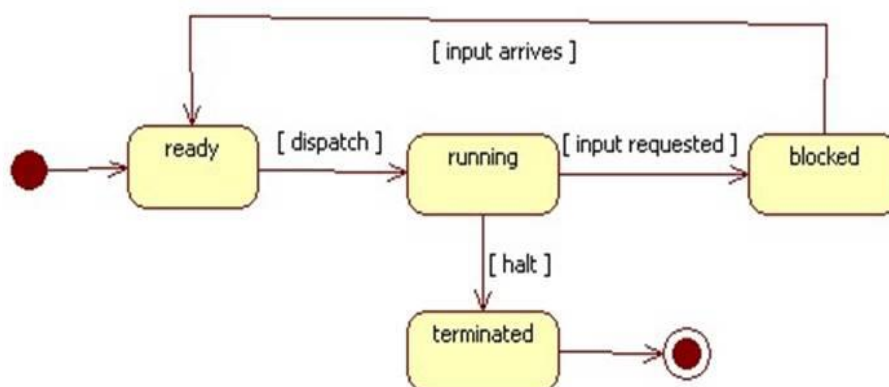
6.4 BUILDING AGENTS BASED MODELS:

Macal and North suggest considering the following aspects when you design an agent-based model:

1. Specific problem to be solved by the ABM

2. Design of agents and their static/dynamic attributes

3. Design of an environment and the way agents interact with it

4. Design of agents' behaviors

5. Design of agents' mutual interactions

6. Availability of data

7. Method of model validation

It is important to keep in mind that just building an arbitrary ABM and obtaining results by simulation wouldn't produce any scientifically meaningful conclusion. In order for an ABM to be scientifically meaningful, it has to be built and used in either of the following two complementary approaches:

A. Build an ABM using model assumptions that are derived from empirically observed phenomena, and then produce previously unknown collective behaviors by simulation.

B. Build an ABM using hypothetical model assumptions, and then reproduce empirically observed collective phenomena by simulation.

Described in one page, the process of building an agent-based model includes answering the following questions:

1. Which objects in the real system are important? These will be the agents.

2. Are there any persistent (or partially persistent) relationships between the real objects? Establish the corresponding links between the agents.

3. Is space important? If yes, choose the space model (GIS, continuous, discrete) and place the agents in the space. If the agents are mobile, set speeds, paths, etc.

4. Identify the important events in the agents' life. These events may be triggered from outside, or they may be internal events caused by the agent's own dynamics.

5. Define the agents' behavior:

    5.1. Does the agent just react to the external events? Use message handling and function calls.

    5.2. Does the agent have a notion of state? Use a statechart.

    5.3. Does the agent have internal timing? Use events or timeout transitions.

5.4. Is there any process inside the agent? Draw a process flowchart.

5.5. Are there any continuous-time dynamics? Create a stock and flow diagram inside the agent.

6. Do agents communicate? Use message sequence diagrams to design communication/timing patterns.

7. What information does the agent keep? This will be the memory, or state information, of the agent. Use variables and statechart states.

8. Is there any information, and/or dynamics, external to all agents and shared by all agents? If yes, there will be a global part of the model (the term "environment" is sometimes used instead).

9. What output are you looking for? Define the statistics collection at both the individual and aggregate levels.

6.4.1 The problem statement,

The assumptions we make about the market are similar to ones of the classical models of product/innovation diffusion, e.g. of Bass model with discards and replacements. We will however consider two competing products instead of one.

- There are two alternative products A and B manufactured by different (and competing) companies. The products are equivalent, i.e. can replace each other. The product prices are equal and therefore do not matter.

- Consumers (there are Total Population = 1000 of them) initially are not using any products but all are potentially interested (are potential users). • Consumers are sensitive to advertizing and to word of mouth.

- Advertizing generates the demand for a product among the potential users. Advertizing Effectiveness = 0.011 is the percent of potential users that become ready to buy a particular product (A or B) during a day. Both companies do advertizing.

- Consumers contact each other. A consumer contacts on average a Contact Rate = 5 other people per day.

- During those contacts the users of products may influence potential users. If a user of e.g. A contacts a potential user, the latter will want to buy A with probability Adoption Fraction = 0.015, same for B.

- Any product discards in Discard Time = uniform(17,23) days and generates the immediate need to buy a replacement of the same brand.

- If a person wants to buy e.g. A, but A is not available for Maximum Waiting Time = 2 days, he becomes ready to buy anything that is available (A or B), same for B.

  Each company (A and B) has its own supply chain that delivers products to the end consumers. The supply chains are very simple and work as follows:

105

- The product can be purchased by a consumer only from the retailer stock, initially holding a certain amount (Initial Retailer Stock = 100) of product.

- The product is manufactured by a producer. A producer makes Production Rate products per day, and this rate may vary, e.g. it can be adjusted according to the demand (which is known to the producer)

- The finished products are delivered to the retailer within Delivery Time = 2 days.

The output of the model should include the market shares for A and B, the demand (i.e. the number of people who want to buy while the product(s) are not available) and the inventory levels in the supply chains.

6.4.2 Phases of modeling,

One of the main objectives of ABM is to test, by experimental means, the hypothesised mechanisms that bring about the macroscopic phenomenon the researcher is interested in explaining.

In ABM these mechanisms are translated as the model microspecifications, that is to say, the set of behavioural and simple rules that specify how the agents behave and react to their local environment (which includes, of course, other agents). Once the population of agents and the environment are defined, the researcher can implement the microspecifications and run the computer simulation in order to evaluate whether these rules bring about or 'generate' the macro phenomenon of interest, over the simulated time.

In order to simplify the presentation, we have identified three major stages:

1) Specification and formalisation;

2) Modelling, verification and experimentation; and

3) Calibration and validation.

The first stage involves translating the theoretical hypothesis that explains the social process of interest, usually expressed in natural languages, into formal languages, using logics or mathematics. The second stage includes the modelling itself, in which the researcher builds and verifies the model by experimental means. The third step includes the calibration of the model with empirical data and the consequent validation of it using appropriate statistical tests.
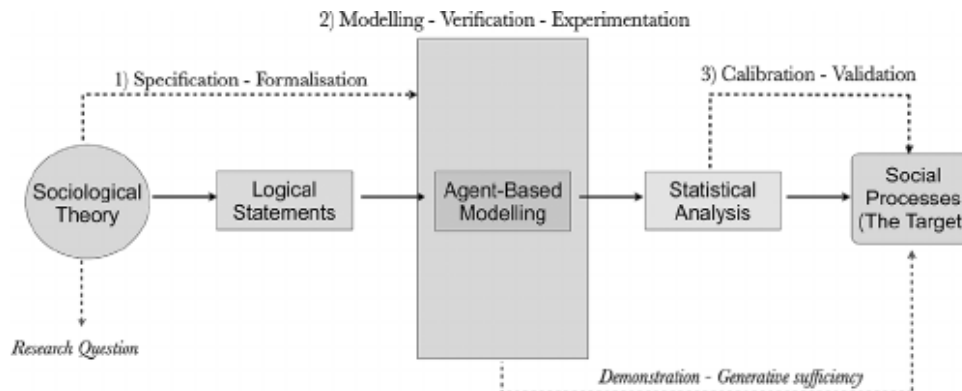
6.4.3 Assumptions,

Assumptions can be informed by data or theory, and outcomes at both the individual and population levels can be compared with data statistically. ABM allows enormous flexibility in assumptions, and agents can be modeled at any level (or multiple levels) of scale. One of the main advantages of agent-based simulation is the flexibility in model design that it allows. Basically, that means that the agent model can be formulated with every level of detail the modeler want it to contain. Pedestrian simulation may be based on reactive agents as well as agents with full spatial cognition and more. An simulation of an ant colony may integrate detailed predator models or just contain a probability for an ant to be killed outside the ant-hill. one aspect that makes a good model is that it just contains necessary assumptions. Every detail incorporated into a model means increasing the number of assumptions that have to be justified and explained. The problem is that everything is in principle possible and the decision about necessary level of detail is not easy to answer. Modelers may fall in love with their model enriching it step by step without stopping at the appropriate level. Nevertheless, every assumption – every decision about a model detail – has to be documented and justified, why this part is elaborated in this particular way.

6.4.4 3-D animation.

3D Modeling is a three-dimensional representation of objects, animals, machines, and humans. In animation production, all the settings and characters are composed of a 3D model. Modeling is an initial step in producing an animation that is done in an animation studio. This is because characters and settings need to be completed before they can be rigged and animated. Also, settings need to be located in their places to specify the very last layout and composition of shots. 3D modeling is used in different kinds of fields from engineering, game industry, film and animation, business advertising to architecture, and special effects. 3D modeling software lets the designer design 3D models of what he has in mind. Some of them provide you with tools to elaborate additional details to your model. The highest used ones are listed here: 3Ds Max, Zbrush, Cinema 4D, Blender and others

6.5 DYNAMICS SYSTEMS:

System Dynamics is a computer-aided approach for strategy and policy design.

The main goal is to help people **make better decisions when confronted with complex, dynamic systems**. The approach provides methods and tools to model and analyzes dynamic systems. Model results can be used to communicate essential findings to help **everyone understand the system's behavior.**

It uses **simulation modeling** based on feedback systems theory that complements systems thinking approaches. It applies to dynamic problems arising in complex social, managerial, economic, or ecological systems. It can be applied to social, managerial, economic, ecological, and physiological systems. As the complexity of our world increases, we need holistic approaches to tackle the problems we encounter in this complex and developing world. Missing the holistic view could lead us to struggle with the symptoms of a larger problem arising from the structure of the system. Hence, System Dynamics approach provide us tools and methods to understand the complex systems.

6.5.1 Stock and flow diagrams,

A system is a set of interrelating, interconnected parts or elements that, together, generate some distinct outcome or behavior over time. In dynamical systems modeling, the behavior that the system exhibits over time is called it's dynamic. Stocks and flows are the basic building blocks of system dynamics models. Jay Forrester originally referred to them as "levels" (for stocks) and "rates" (for flows). A stock variable is measured at one specific time and represents a quantity existing at that point in time (say, December 31, 2004), which may have accumulated in the past. A flow variable is measured over an interval of time. Therefore a flow would be measured per unit of time (say a year).
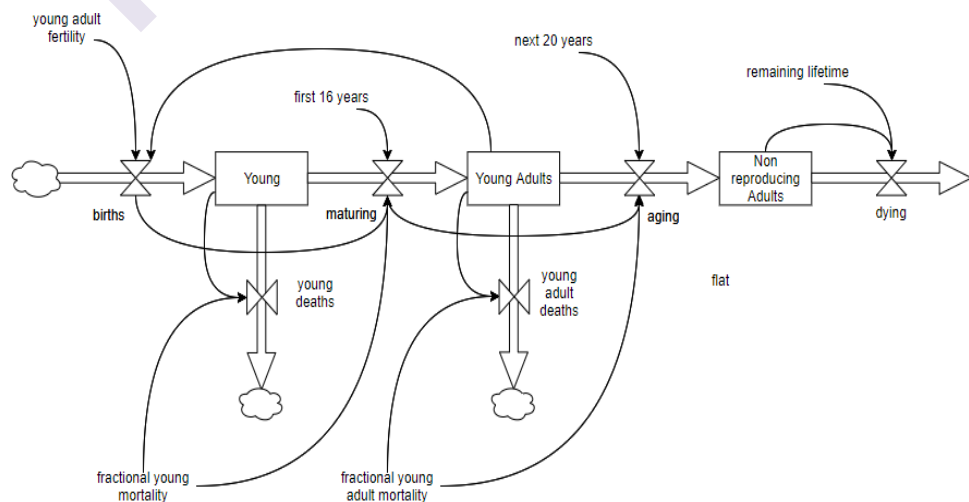
6.5.2 Examples of stock and flow diagrams.

Economics, business, accounting, and related fields often distinguish between quantities that are **stocks** and those that are **flows**. These differ in their units of measurement. A *stock* is measured at one specific time, and represents a quantity existing at that point in time (say, December 31, 2004), which may have accumulated in the past. A *flow* variable is measured over an interval of time. Therefore, a flow would be measured *per unit of time* (say a year). Flow is roughly analogous to rate or speed in this sense.

For example, U.S. nominal gross domestic product refers to a total number of dollars spent over a time period, such as a year. Therefore, it is a flow variable, and has units of dollars/year. In contrast, the U.S. nominal capital stock is the total value, in dollars, of equipment, buildings, and other real productive assets in the U.S. economy, and has units of dollars. The diagram provides an intuitive illustration of how the *stock* of capital currently available is increased by the *flow* of new investment and depleted by the *flow* of depreciation.

Elements of Stock Flow Diagram

Name	Symbol	Description
Stock		· A **stock** is accumulated over time by inflows and/or depleted by outflows. Stocks can only be changed via flows. · Mathematically a stock can be seen as an accumulation or integration of flows over time – with outflows subtracting from the stock. · Stocks typically have a certain value at each moment of time. · Example: the number of population at a certain moment.
Flow		· A **flow** changes a stock over time. Usually, we can clearly distinguish inflows (adding to the stock) and outflows (subtracting from the stock). · Flows typically are measured over a certain interval of time · Example: the number of births over a day or month.

Flow Rate		· It is always expressed per some unit time: If these flow into/out of a stock that keeps track of things of type X · e.g. Incidence Rates is calculated by accumulating people over a year, revenue is $/Time, water flow is liters/minute
Cloud		Cloud represents either: · Source of the flow – when the flow is originated from outside the model · Sink of the flow – when the flow sinks Both sources and sinks are assumed to have infinite capacity and do not impose any limitations on the flows. Clouds are drawn as part of the flow element, in case the flow does not flows in/out of some stock.
Link	⟶	· Link is used to define a dependency between elements of a stock and flow diagram · If some element A is mentioned in the equation or initial value of element B, you should first connect these elements with a link going from A to B and only then type the expression in the properties of B.

Stocks. A stock represents a part of a system whose value at any given instant in time depends on the system's past behavior. The value of the stocks at a particular instant in time cannot simply be determined by measuring the value of the other parts of the system at that instant in time – the only way you can calculate it is by measuring how it changes at every instant and adding up all these changes.

This sounds more complicated than it is, so let us look at a simple example: driving a car along the motorway. Say you start driving at 8:00 AM and you want to know how far you have driven at 10:00 AM. We know that the only factor that determines this is the speed you were driving at. But it is not enough to just know your current speed at 10:00 AM, you actually need to know exactly how fast you were driving at every instant in time between 8:00 AM and 10:00 AM to calculate this. In this example, the distance you have driven is a stock – if you look at the dashboard in your car, you will most likely find a representation of this stock on your car's dashboard: the mileage counter (odometer). On diagrams, stocks are represented by rectangles.

Flows. Flows represent the rate at which the stock is changing at any given instant, they either flow into a stock (causing it to increase) or flow out of a stock (causing it to decrease).

To continue our example above, the car's velocity at any particular instant is a flow that flows into the mileage counter stock. It is important to note here that the distinction between stock and flow is not absolute – from the point of view of the mileage counter the velocity is a flow. But the velocity itself most likely also changes and depends on the acceleration and deceleration. So, even though we can determine the current velocity almost instantaneously (this is done by the speedometer), we again cannot explain why the velocity is at its current level without knowing the system's past behavior. On diagrams, flows are represented by small valves attached to flow pipes that lead into or out of stocks.

Converters. Converters either represent parts at the boundary of the system (i.e. parts whose value is not determined by the behavior of the system itself) or they represent parts of a system whose value can be derived from other parts of the system at any time through some computational procedure.

To continue our motorway example, we could assume that acceleration and deceleration are determined by outside circumstances (e.g. such as the positions of the accelerator and brake). In this case, we would model both the accelerator and brake positions as converters. On diagrams, converters are represented by small circles.

Connectors. Much like in causal loop diagrams the connectors of a system show how the parts of a system influence each other. Stocks can only be influenced by flows (i.e. there can be no connector that connects into a stock), flows can be influenced by stocks, other flows, and by converters. Converters either are not influenced at all (i.e. they are at the systems' boundary) or are influenced by stocks, flows and other converters.

Source/Sink. Sources and sinks are stocks that lie outside of the model's boundary – they are used to show that a stock is flowing from a source or into a sink that lies outside of the model's boundary. On diagrams, sources and sinks are represented by small clouds.

The notation used in stock and flow diagrams was originated by Jay Forrester in his book "Industrial Dynamics". It was based on a hydraulic metaphor: the flow of water into and out of reservoirs. Hence the names of these elements and their visualization.

The key feature of a stock and flow diagram is that each construct can be precisely specified using a mathematical formalism – viewed from a mathematical perspective, such fully specified stock and flow models are just a way of visualizing a corresponding set of integral equations.
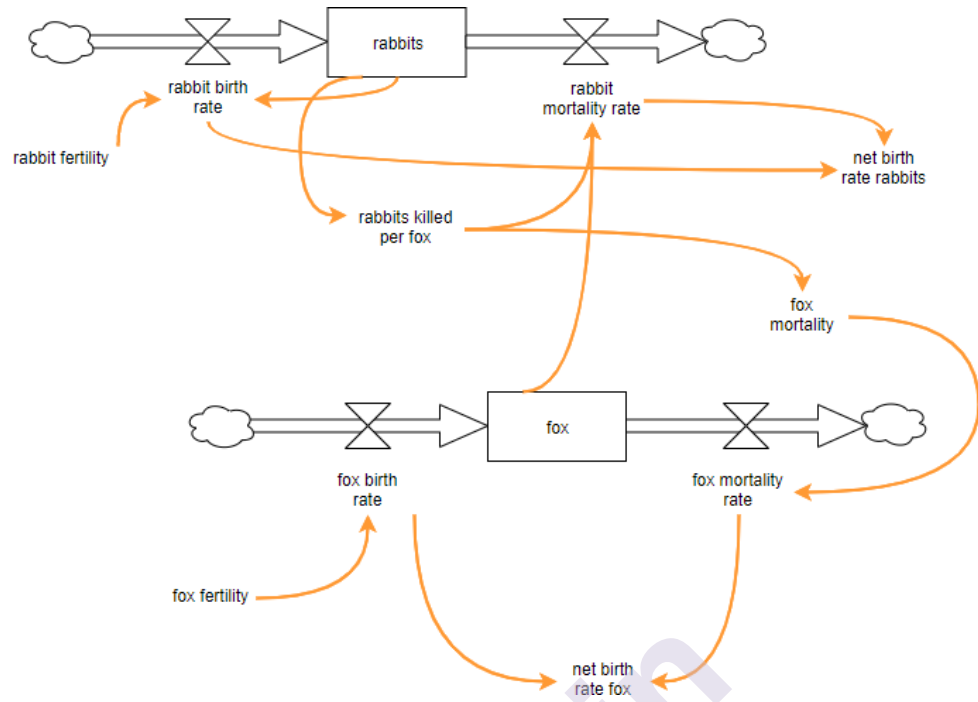
6.6 MULTI-METHOD MODELING:

The idea of multimethod modeling is simple: to seamlessly integrate different methods of modeling and simulation to overcome the drawbacks of individual approaches and get the most from each one. Combining different methods leads to efficient and manageable models without using workarounds.

There are three major methodologies used to build dynamic business simulation models: system dynamics, discrete event modeling, and agent based modeling.

The system dynamics method assumes a high abstraction level and is primarily used for strategic level problems, such as market adoption rates and social process dependency.

Discrete event modeling is mainly used at operational and tactical levels, like manufacturing processes and equipment investment evaluation.

Agent-based models are used at all levels, with the agents possibly being any active entity. Example applications include supply chain optimization and epidemiology.

Depending on the simulation project goals, the available data, and the nature of the system being modeled, different problems may call for different methods. Also, sometimes it is not clear at the beginning of the project which abstraction level and which method should be used. The modeler may start with, say, a highly abstract system dynamics model and switch later on to a more detailed discrete event model. Or, if the system is heterogeneous, the different components may be best described by using different methods. For example in the model of a supply chain that delivers goods to a consumer market the market may be described in system dynamics terms,

the retailers, distributors, and producers may be modeled as agents, and the operations inside those supply chain components – as process flowcharts.

6.**6.1 Architecture,**

The number of possible multi-method model architectures is infinite, and many are used in practice. Popular examples are shown in the Figure 1. In this section we briefly discuss the problems where these architectures may be useful.
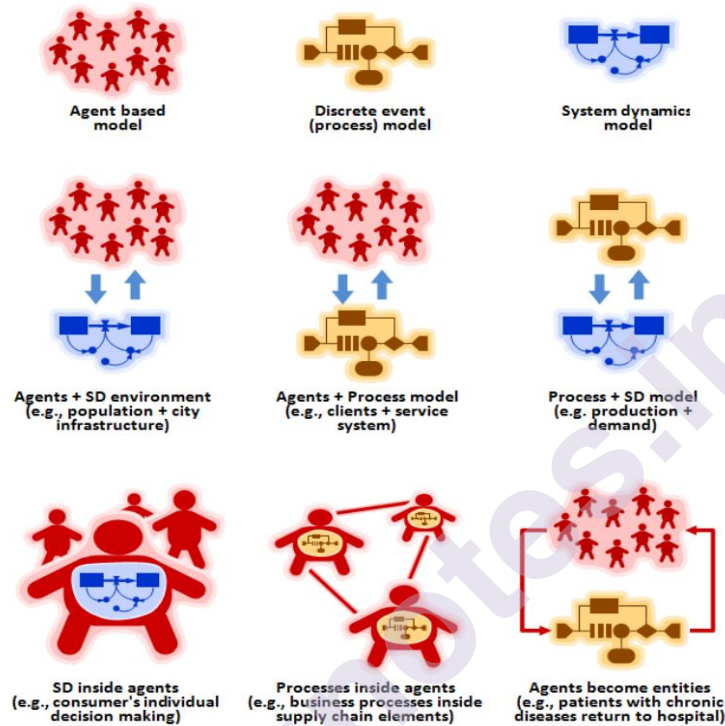
Agents in an SD environment. Think of a demographic model of a city. People work, go to school, own or rent homes, have families, and so on. Different neighborhoods have different levels of comfort, including infrastructure and ecology, cost of housing, and jobs. People may choose whether to stay or move to a different part of the city, or move out of the city altogether. People are modeled as agents. The dynamics of the city neighborhoods may be modeled in system dynamics way, for example, the home prices and the overall attractiveness of the neighborhood may depend on crowding, and so on. In such a model agents' decisions depend on the values of the system dynamics variables, and agents, in turn, affect other variables. The same architecture is used to model the interaction of public policies (SD) with people (agents). Examples: a government effort to reduce the number of insurgents in the society; policies related to drug users or alcoholics.

Agents interacting with a process model. Think of a business where the service system is one of the essential components. It may be a call center, a set of offices, a Web server, or an IT infrastructure. As the client base grows, the system load increases. Clients who have different profiles and histories use the system in different ways, and their future behavior depends on the
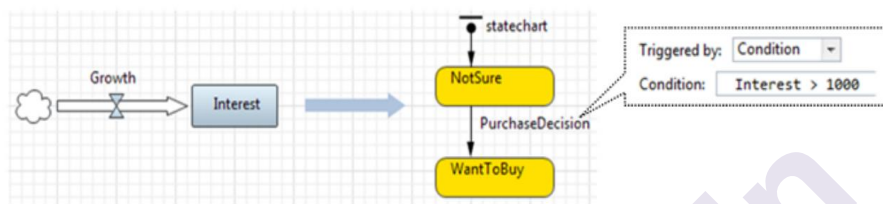
response. For example, low-quality service may lead to repeated requests, and, as a result, frustrated clients may stop being clients. The service system is naturally modeled in a discrete event style as a process flowchart where requests are the entities and operators, tellers, specialists, and servers are the resources. The clients who interact with the system are the agents who have individual usage patterns. Note that in such a model the agents can be created directly from the company CRM database and acquire the properties of the real clients. This also applies to the modeling of the company's HR dynamics. You can create an agent for every real employee of the company and place them in the SD environment that describes the company's integral characteristics (the first architecture type). A process model linked to a system dynamics model. The SD aspect can be used to model the change in the external conditions for an established and ongoing process: demand variation, raw material pricing, skill level, productivity, and other properties of the people who are part of the process. The same architecture may be used to model manufacturing processes where part of the process is best described by continuous time equations – for example, tanks and pipes, or a large number of small pieces that are better modeled as quantities rather than as individual entities. Typically, however, the rates (time derivatives of stocks) in such systems are piecewise constants, so simulation can be done analytically, without invoking numerical methods. System dynamics inside agents. Think of a consumer market model where consumers are modeled individually as agents, and the dynamics of consumer decision making is modeled using the system dynamics approach. Stocks may represent the consumer perception of products, individual awareness, knowledge, experience, and so on. Communication between the consumers is modeled as discrete events of information exchange. A larger-scale example is interaction of organizations (agents) whose internal dynamics are modeled as stock and flow diagrams. Processes inside agents. This is widely used in supply chain modeling. Manufacturing and business processes, as well as the internal logistics of suppliers, producers, distributors and retailers are modeled using process flowcharts. Each element of the supply chain is at the same time an agent. Experience, memory, supplier choice, emerging network structures, orders and shipments are modeled at the agent level. Agents temporarily act as entities in a process. Consider patients with chronic diseases who periodically need to receive treatment in a hospital (sometimes planned, sometimes because of acute phases). During treatment, the patients are modeled as entities in the process. After discharge from the hospital, they do not disappear from the model, but continue to exist as agents with their diseases continuing to progress until they are admitted to the hospital again. The event of admission and the type of treatment needed depend on the agent's condition. The treatment type and timeliness affect the future disease dynamics. There are models where each entity is at the same time an agent exhibiting individual dynamics that continue while the entity is in the process, but are outside the process logic – for example, the sudden deterioration of a patient in a hospital.
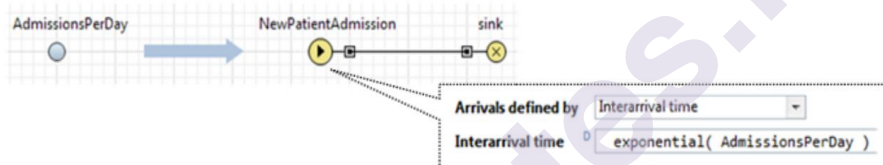
6.6.2 Technical aspects of combining modeling methods,

Important feature of the modeling language we are using is that all model elements of all methods, be they SD variables, statechart states, entities, process blocks, exist in the "same namespace": any element is accessible from any other element by name (and, sometimes, "path" – the prefix describing the location of the element in the model hierarchy). The following examples are all taken from the real projects and purged of all unnecessary details. This set, of course, does not cover everything, but it does give a good overview of how you can build interfaces between different methods.

The system dynamics model is a set of continuously changing variables. All other elements in the model work in discrete time (where any changes are associated with events). SD itself does not generate any events, so it cannot actively make an impact on agents, process flowcharts, or other discrete time constructs. The only way for the SD part of the model to impact a discrete element is to let that element watch on a condition over SD variables, or to use SD variables when making a decision. The Figure 2 shows some possible constructs. In the Figure 2 case A an SD variable triggers a statechart transition. Events (low-level constructs that allow scheduling a one-time or recurrent action) and statechart transitions are frequent elements of agent behavior. Among other types of triggers, both can be triggered by a condition – a Boolean expression the model contains dynamic variables, all conditions of events and statechart transitions are evaluated at each integration step, which ensures that the event or transition will occur exactly when the (continuously changing) condition becomes true. In the figure the statechart is waiting for the Interest stock to rise higher than a given threshold value. In the Figure 2 case B the flowchart source block NewPatientAdmissions generates new entities at the rate defined by the dynamic variable AdmissionsPerDay, which may be a part of a stock and flow diagram.

6.6.3 Examples.

Consumer market and supply chain: We will model the supply chain and sales of a new product in a consumer market in the absence of competition.

115

The supply chain will include the delivery of the raw product to the production facility, production, and the stock of the finished products. The QR inventory policy will be used. Consumers are initially unaware of the product; advertizing and word of mouth will drive the purchase decisions. The product has a limited lifetime, and 100% of users will be willing to buy a new product to replace the old one. The full version of the model is available at RunTheModel.com. We will use discrete event methodology to model the supply chain, and system dynamics methodology, namely, a slightly modified Bass diffusion model (Bass 1969), to model the market. We will link the two models through the purchase events.

Discrete event model of a supply chain
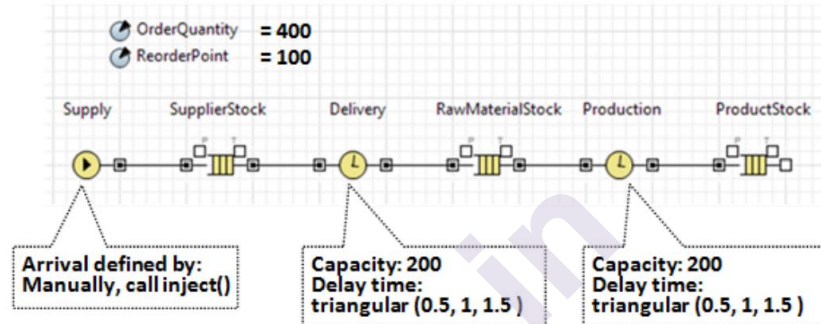
System dynamics model of the new product diffusion in the market
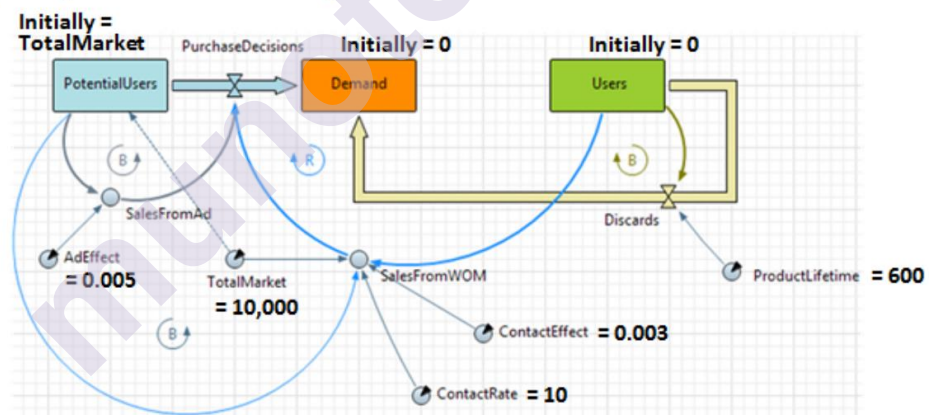
PurchaseDecisions = SalesFromAd + SalesFromWOM

SalesFromAd = PotentialUsers * AdEffect

$$\text{SalesFromWOM} = \text{Users} * \text{ContactRate} * \frac{\text{PotentialUsers}}{\text{TotalMarket}} * \text{ContactEffect}$$

Discards = Users / ProductLifetime

The supply chain flowchart (top of the Figure 5) includes three stocks: the supplier stock of raw material, the stock of raw material at the production site, and the stock of finished products at the same location. Delivery and production are modeled by the two Delay objects with limited capacity. The Supply block of the flowchart is not generating any entities unless explicitly asked to do so (the inventory policy is not yet present at this stage). To load the supply chain with some initial product quantity we will add this Startup code: Supply.inject(OrderQuantity);. If we run this model, at the beginning

of the simulation, four hundred items of the product are produced and accumulate in the ProductStock.

How do we link the supply chain and the market? We want to achieve the following: • If there is at least one product item in stock and there is at least one client who wants to buy it, the product item should be removed from the ProductStock queue, the value of Demand should be decremented, and the value of Users should be incremented.

6.7 CONCLUSION

When developing a discrete event model of a supply chain, IT infrastructure, or a contact center, the modeler would typically ask the client to provide the arrival rates of the orders, transactions, or phone calls. He would then be happy to get some constant values, periodical patterns, or trends, and treat arrival rates as variables exogenous to the model. In reality, however, those variables are outputs of another dynamic system, such as a market, a user base. Moreover, that other system can, in turn, be affected by the system being modelled. For example, the supply chain cycle time, which depends on the order rate, can affect the satisfaction level of the clients, which impacts repeated orders and, through the word of mouth, new orders from other customers. The choice of the model boundary therefore is very important.

6.8 PRACTICE QUESTION

1.   Write a short note on Agent Based modeling

2.   Explain the Time dimension in agent based models

3.   Explain the Space dimension in agent based models

4.   What do you mean by discrete space in ABM? Explain.

5.   Explain the Communication process between agents in ABM.

6.   How agents are created and destroyed dynamically?

7.   Explain the process Building agents based models:

8.   Explain using suitable example the following stages in building ABM

    a.   The problem statement,

    b.   Phases of modeling,

9.   Explain the role of Assumptions in ABM.

10.   How 3-D animation helps in modeling and simulation.

11.   Write a short note on Dynamics Systems:

12.   Explain different symbols used Stock and flow diagrams,

13.   Give a suitable Examples of stock and flow diagrams.

14.   Write a short note on Multi-method modeling.

15.   Explain the Architecture of Multi-method modeling.

16.   Explain the Technical aspects of combining modeling methods.

6.9 REFERENCES

☐ Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004.

☐ The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013.

☐ Agent Based Modeling and Simulation, Taylor S, 2014.

☐ Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003.

☐ Object Oriented Simulation: A Modeling and Programming Perspective, Garrido, José M, 2009.

☐ Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003.

☐ Process Control: Modeling, Design and Simulation, Wayne Bequette W, Prentice Hall of India, 2003.

❊❊❊❊❊❊

7

DESIGN AND BEHAVIOR OF MODELS

Unit Structure :

7.1 INTRODUCTION:

Simulation involves the development of descriptive computer models of a system and exercising those models to predict the operational performance of the underlying system being modeled. Systems that change with time, such as a gas station where cars come and go (called dynamic systems) and involve randomness. Nobody can guess at exactly which time the next car should arrive at the station, are good candidates for simulation. Modeling complex dynamic systems theoretically need too many simplifications and the emerging models may not be therefore valid.

7.2 OBJECTIVES:

- To understand Modeling tools that permit specialized device characterization or custom model development

- To understand and Design knowledge and thorough understanding of simulation objectives and expectations of system behaviour.

- To learn and design a simulator that is robust and reliable that can always be accepted with confidence that they reflect the state of the system.

7.3 DESIGNING STATE-BASED BEHAVIOR:

Deterministic refers to the uniqueness of the computation. A deterministic model will always produce the same output from a given starting condition or initial state.

A dog is always **asleep** or **awake**. The dog can't be asleep and awake at the same time, and it's impossible for the dog to be neither asleep nor awake. There's only these two states, a precisely limited, *finite* number of states. How the dog goes between **asleep** and **awake** is through *transitions*, which are symbolised by an arrow pointing from one state to the next state in the process's sequence. A transition is caused by an *event* that results in the change of state. Transitions are labelled with their events.

7.3.1 Statecharts

A statechart is a visual construct that enables you to define event- and time-driven behavior of various objects (agents). Statecharts are very helpful in simulation modeling. They are used a lot in agentbased models, and also work well with process and system dynamics models.

Statecharts consist of "states" and "transitions". A state can be considered as a "concentrated history" of the agent and also as a set of reactions to external events that determine the agent's future. The reactions in a particular state are defined by transitions exiting that state. Each transition has a "Trigger", such as a message arrival, a condition, a timeout, or the agent arrival to the destination. When a transition is taken ("fired") the state may change, and a new set of reactions may become active. State transition is atomic and instantaneous. Arbitrary actions can be associated with transitions and with entering and exiting states.

Statecharts define internal states, reactions to external events, and the corresponding state transitions of a particular agent: a person, a physical device, an organization, a project, etc. The (simple) states of the statechart are alternative: at any given moment in time the statechart is in exactly one simple state.
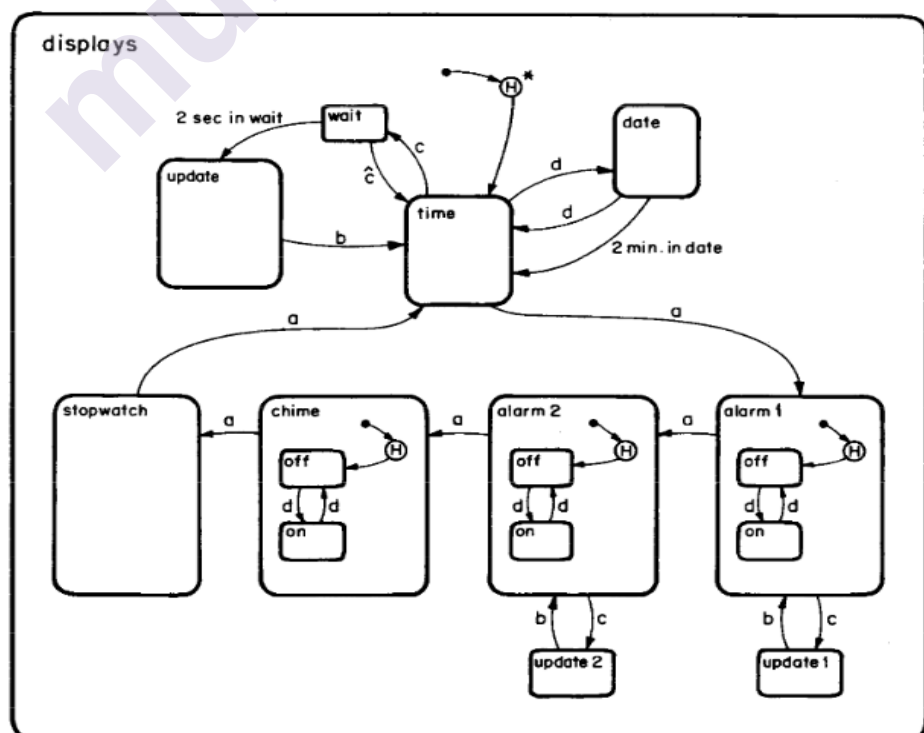
Statecharts constitute a visual formalism for describing states and transitions in a modular fashion, enabling clustering, orthogonality (i.e., concurrency) and refinement, and encouraging 'zoom' capabilities for moving easily back and forth between levels of abstraction.

A basic fragment of such a description is a state transition, which takes the general form "when event E occurs in state A, if condition C is true at the time, the system transfers to state B"

7.3.2 State transitions,

Transitions and events are *deterministic*. Deterministic means that each transition and event always points to the same next state, and always produces the same result from their given starting condition, every time the process is run. Dogs never **wake up** to become **asleep** or **fall asleep** to become **awake**.
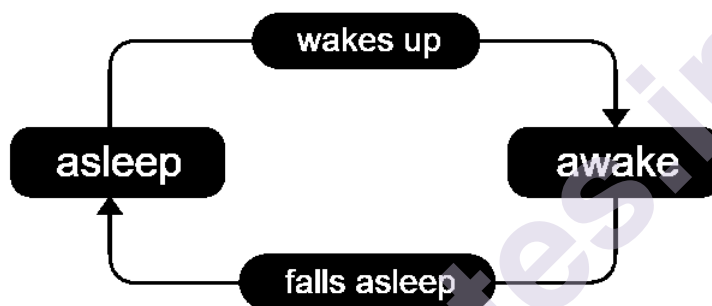
This tiny dog process, with its two finite states and two transitions is a *Finite State Machine.* A state machine is used to describe the behavior of something. The machine describes the thing's states and the transitions between those states. It's a Finite State Machine because it has a finite number of states.

7.3.3 Viewing and debugging Statecharts at runtime,

At model runtime values of parameters and variables are displayed under their names.

While debugging a statechart, you can perform the following actions:

• Set breakpoints.

• Highlight statechart execution.

• Single-step through a statechart.

• Watch statechart data.

7.3.4 Statecharts for dynamic objects.

The *dynamic model* describes the internal behavior of a system. *Statechart diagrams* describe the states of an individual object and the possible transitions between states

As in general, the individuals can appear in the model at different ages (e.g. as a result of immigration), the statechart may initialize in either of the states, depending on the age. (Note that the statechart entry point points not to a state directly but to a decision point with four branches.) The transitions between the life phases are triggered by stochastic timeouts. For example, the transition from Adult to MidAge happens when the person is around 49, which is modeled by the timeout normal(49,5) – age(), where normal(49,5) is a normally distributed age with mean 49 and standard deviation 5, and age() is the age the person became Adult (remember that the timeout expressions are evaluated at the moment the statechart gets into the transition source state, in this case Adult). The Adult life phase is further decomposed to describe family-related behavior. The decisions in this section of the statechart (to have family or not, how long to wait before the first kid, how many kids to have, etc.) are also stochastic and may depend on the gender, the level of education, the cultural norms, etc. If this statechart is inside an agent in an agent-based model, the act of childbearing may result in a new agent added to the model, who may inherit the characteristics of the parents. The event of death then may delete the agent from the model.
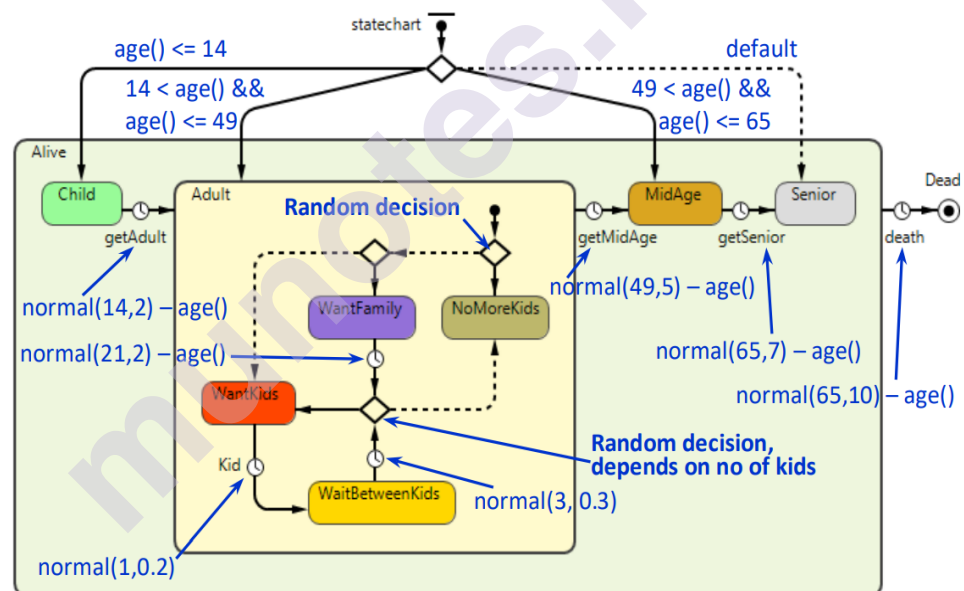
Statechart for life phases

7.4 DISCRETE EVENTS AND EVENT MODEL OBJECT:

Discrete event simulation (DES) is a method used to model real world systems that can be decomposed into a set of logically separate processes that autonomously progress through time. Each event occurs on a specific process, and is assigned a logical time (a timestamp). The result of this event can be an outcome passed to one or more other processes. The content of the outcome may result in the generation of new events to be processed at some specified future logical time. The underlying statistical paradigm that supports DES is based in queuing theory.

Discrete event modeling is almost as old as system dynamics. In October 1961, IBM engineer Geoffrey Gordon introduced the first version of GPSS (General Purpose Simulation System, originally Gordon's Programmable Simulation System), which is considered to be the first method of software implementation of discrete event modeling. These days, discrete event modeling is supported by a large number of software tools, including modern versions of GPSS itself.

The typical output expected from a discrete event model is:

• Utilization of resources,

• Time spent in the system or its part by an agent,

• Waiting times,

• Queue lengths,

• System throughput,

• Bottlenecks,

• Cost of the agent processing and its structure.

The simple example consists of a cashier serving arriving customers, one at a time. Customers queue if the cashier is not available (serving another customer). Here, the state of the system consists of the state of the queue and that of the cashier. The queueing discipline is First In First Out (FIFO) and individual customers are assumed not to have any distinguishing features (such as age, or number of items bought). Thus, it is meaningful to model the state of the queue by means of the queue length, a natural number. The cashier can be in either the Idle or the Busy state. The dynamics of the system is determined by:

• The arrival pattern of customers characterized by their Inter Arrival Time (IAT) distribution,

• The time required by the cashier to serve a customer characterized by the Service Time (ST) distribution,

• The logical sequence of customers progressing through the system under different conditions (queue empty/not empty, cashier Busy/Idle).

Physical View

123
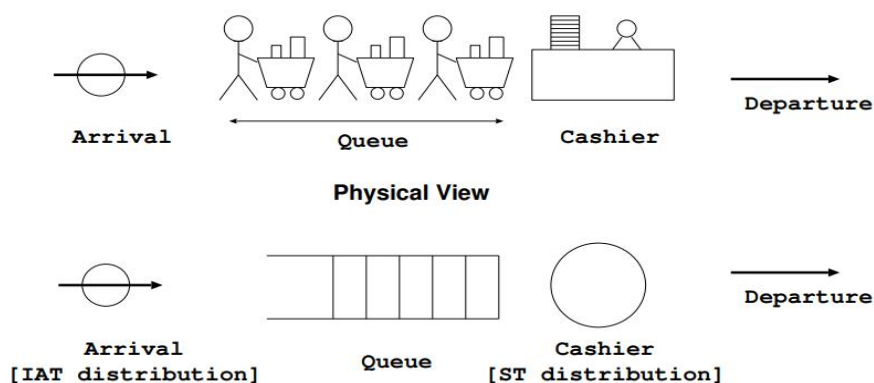
7.4.1 Discrete event,

Discrete-event is the intended change to simulate systems where events occur at specific, separable instances in time. The defining characteristics of discrete-event simulation are as follows:

- Events occurring at specific points in time. As mentioned earlier, DES is used to model processes or systems that change at identifiable time instances. DES does not track system state continuously.

- Emphasis on events. Changes in state and events are the focus of DES, hence why DES is often referred to as "event-driven."

- Time skipping (next-event time advance). Typically, DES does not consider the system's state between events, instead "jumping" to subsequent events as the simulation progresses. Time skipping reduces the complexity and resource intensity of discrete-event simulations.

- Heavy use of queueing theory. Queueing theory is the cornerstone of many discrete-event simulations, defining how resource-constrained processing is executed. Although queueing theory is not always used in DES, it can be used whenever the arrival and the service of requests are the features of interest.

7.4.2 Event-the simplest low level model object,

An event is the specification of a significant occurrence that has a location in time and space. Anything that happens is modeled as an event in UML. In the context of state machines, an event is an occurrence of a stimulus that can trigger a state transition four kinds of events – signals, calls, the passing of time, and a change in state.

Most business processes can be described as a sequence of separate discrete events. For example, a truck arrives at a warehouse, goes to an unloading gate, unloads, and then departs. To simulate this, discrete-event simulation is often chosen.

7.4.3 Dynamic events, and Exchanging data with external world.

Using discrete-event simulation modeling, the movement of a train from point A to point B is modeled with two events, namely a departure and an arrival. The actual movement of the train would be modeled as a time delay between the departure and arrival events. These events and movement between them can be smoothly animated.

Discrete-event simulation focuses on the processes in a system at a medium level of abstraction. Typically, specific physical details, such as car geometry or train acceleration, are not represented. Discrete-event simulation modeling is widely used in the manufacturing, logistics, and healthcare fields.

An event in the context of state machines is an occurrence of a stimulus that can trigger a state transition.

7.5 DESIGNING INTERACTIVE MODELS: USING CONTROLS, DYNAMIC PROPERTIES OF CONTROLS, 3D ANIMATION.

Willemain (1994) lists five qualities of an effective model: validity, usability, value to client, feasibility and aptness for clients' problem. Meanwhile, Brooks and Tobias (1996) identify 11 performance criteria for a good model. Based on these lists, here it is proposed that there are four main requirements of a conceptual model: validity, credibility, utility and feasibility.

Here are three examples:

1. A natural scientist may be interested in a system of wolves and sheep, where the number of wolves changes with a constant birth rate and a death rate that is inversely proportional to the number of sheep, and the number of sheep changes with a constant birth rate and a death rate that is directly proportional to the number of wolves. The scientist would like to know the following: Do the number of wolves and the number of sheep stabilize in the long run, and if so to what values? Or do they vary cyclically, and if so with what period and phase?

2. A computer scientist may be interested in a system of jobs that circulate in a network of servers (e.g., CPU's and I/O devices). The computer scientist would like to know whether a particular server is a ''bottleneck'', i.e., in the long run, is that server always busy while the other servers are mostly idle.

3. A classical system example is a queuing system with a single server. Here, customers arrive with certain service requirements, get served in some order, say first-comefirst-served, and depart when their service is completed. Note that a customer who arrives when the server is busy has to wait (in a queue). For this system, we would like to determine the average waiting time for customers, the average number of customers in the system, the fraction of time the server is busy, etc.

A Simple Example: Building a simulation of gas station with a single pump served by a single service man. Assume that arrival of cars as well their service times are random. At first identify the:

1. states: number of cars waiting for service and number of cars served at any moment
2. events: arrival of cars, start of service, end of service
3. entities: these are the cars
4. queue: the queue of cars in front of the pump, waiting for service
5. random realizations: inter-arrival times, service times
6. distributions: we shall assume exponential distributions for both the inter-arrival time and service time.

7.6 RANDOMNESS IN MODELS:

Uncertainty is an essential part of our everyday lives. Here's a simple example: imagine you own a café that serves hot beverages and delicious pastry to around 100 customers per day. The café is open from 9 am to 7 pm. However, the first guests could arrive at 10 one day and 9:50 the next day, it could be two customers or a dozen entering at the same time. When we build simulation models, we want them to reflect the real world as closely as possible. For that, you would need to include randomness in your simulation.

In most real-life situations, the arrival process and the service process occur in a random fashion. Even though the processes may be random, it does not mean that you cannot describe or model the randomness. To have any hope of simulating the situation, you must be able to model the randomness. One of the ways to model this randomness is to describe the phenomenon as a random variable governed by a particular probability distribution. For example, if the arrivals to the bank occur according to a Poisson process, then from probability theory it is known that the distribution of inter-arrival times is an exponential distribution. In general, information about how the customers arrive must be secured either through direct observation of the system or by using historical data. If neither source of information is available, then some plausible assumptions must be made to describe the random process by a probability model.

Probability distributions:

The first choice is to develop a probability model given the data. The second choice is to try to drive the simulation directly from the historical data. The latter approach is not recommended. First of all, it is extremely unlikely that the captured data will be in a directly usable form. Secondly, it is even more unlikely that the data will be able to adequately represent all the modeling scenarios that you will need through the course of experimenting with the model. For example, suppose that you only have 1 day's worth of arrival data, but you need to simulate a month's worth of system operation. If you simply re-drive your simulation using the 1 day's worth of data, you are not simulating different days! It is much more advisable to develop probability models either from historical data or from data that you capture in developing your model.

Once a probability model has been developed, statistical theory provides the means for obtaining random samples based on the use of uniformly distributed random numbers on the interval (0,1). These random samples are then used to map the future occurrence of an event on the time scale. For example, if the inter-arrival time is exponential then a random sample drawn from that distribution would represent the time interval until the occurrence of the next arrival.

Sources of randomness in the model:

There are two types of models: stochastic and deterministic.

A deterministic model doesn't have internal randomness. It runs with the same set of input parameters and gives the same output results. For example, if 10,000 individuals each have a 95% chance of surviving one year, we can be reasonably sure that 9,500 of them will survive.

A stochastic model, on the other hand, does have internal sources of randomness. So, each run (even with the same parameters) may give a different output.
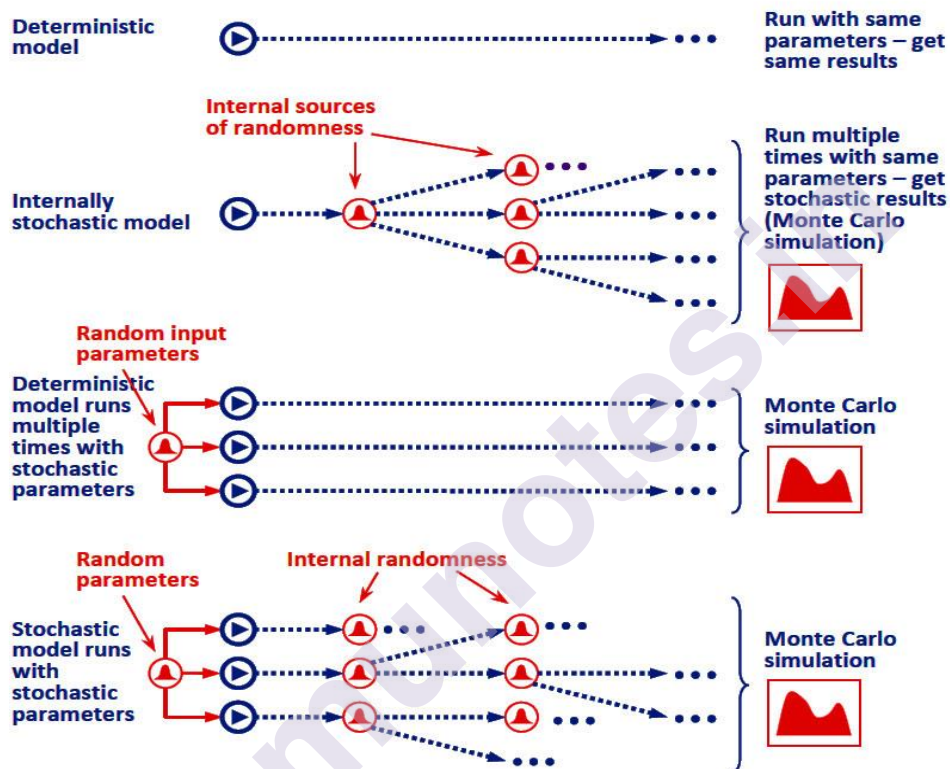
Randomness in system dynamics model:

System Dynamics models often incorporate random components, in two ways:

• Internal: the system itself is stochastic (e.g. parts failures, random variations in sales, Poisson arrivals, etc.

• External: All the usual Monte-Carlo explorations of uncertainty from either internal randomness or via replacing constant-but-unknown parameters with probability distributions as a form of sensitivity analysis.

There is also a kind of probabilistic flavor to the deterministic simulations in System Dynamics. If one has a stochastic linear differential equation with deterministic coefficients and Gaussian exogenous inputs, it is easy to prove that all the state variables have time-varying Gaussian

127

densities. Further, the time-trajectories of the means of those Gaussian process can be computed immediately by the deterministic linear differential equation which is just the original stochastic equations, with all random inputs replaced by their mean trajectories. In System Dynamics, this concept, rigorous in the linear case, is extended informally to the nonlinear case as an approximation. That is, the deterministic solution of a System Dynamics model is often taken as an approximation of what would be concluded about the mean of a Monte-Carlo exploration. Of course it is only an approximate notion, and it gives no information at all about the variances of the stochastic variables.

A third kind of randomness in System Dynamics models is also a bit informal: delays, which might be naturally modeled as stochastic, are modeled as deterministic but distributed. For example, if procurement orders are received on average 6 months later, with randomness of an unspecified nature, a typical System Dynamics model would represent the procurement delay as a deterministic subsystem, usually a first- or third-order exponential delay. That is the output of the delay, in response to a pulse input, is a first- or third-order Erlang shape. These exponential delays often do a good job of matching data taken from high-volume stochastic processes.

Random number generators:

A random number generator is a critical component in modern cryptographic systems, communication systems, statistical simulation systems and any scientific area incorporating Monte Carlo methods and may other systems. Random number generators can be classified in three classes; true random number generators, pseudo random number generators (PRNG) and hybrid random number generators. seudo random number generators are deterministic processes which generate a series of outputs from an initial seed state.
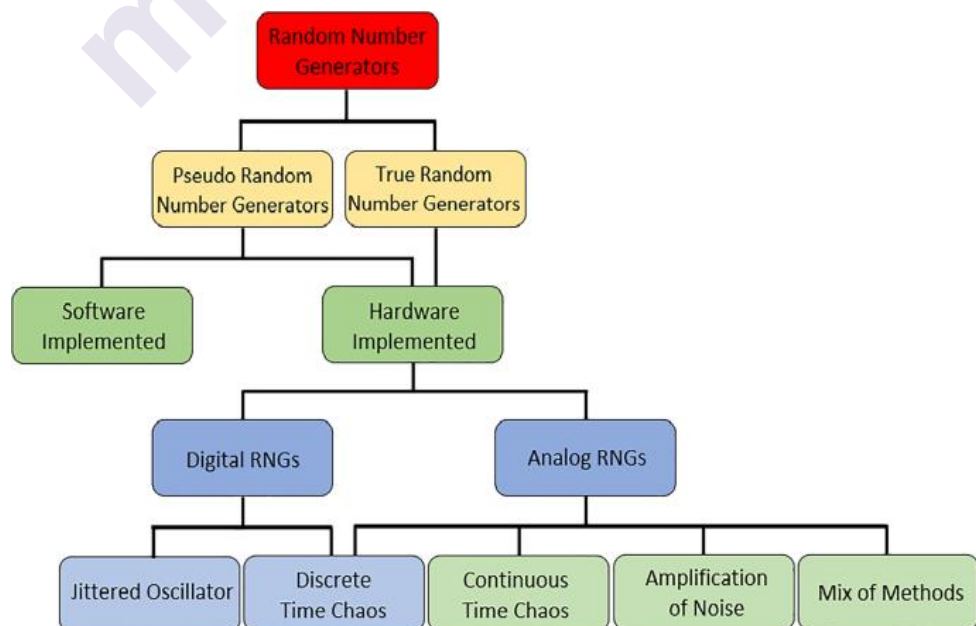
The Linear Congruential Generator

This generator produces a series of pseudorandom numbers. Given an initial seed X0 and integer parameters a as the multiplier, b as the increment, and m as the modulus, the generator is defined by the linear relation:

$Xn \equiv (aXn\text{-}1 + b)$mod m. Or using more programming friendly syntax:

$Xn = (a * Xn\text{-}1 + b) \% m$.

Each of these members have to satisfy the following conditions:

m > 0 (the modulus is positive),

0 < a < m (the multiplier is positive but less than the modulus),

0 ≤ b < m (the increment is non negative but less than the modulus), and

0 ≤ X0 < m (the seed is non negative but less than the modulus).

```
function H(x · y):
    f = 2^x mod p
    b = 0
    for i = 0 to |x|:
        b = b XOR (x[i] AND y[i])
    return f · y · b
```

7.7 MODEL TIME, DATE AND CALENDAR VIRTUAL AND REAL TIME:

Time is the central axis in the dynamic simulation models we are building. The models are full of various references to time: delays, arrival times, service times, rates, timeouts, schedules, dates, velocities, etc. This section explains what model time is and how the user can work with it.

The model time, date and calendar:

Model time is the virtual (simulated) time maintained by the simulation engine. The model time has nothing to do with the real time or the computer clock.

The model time takes Java double type values (real numbers with double precision). The model clock is advanced in steps: while the engine is executing a discrete event model, the model time jumps from one event to another; if a continuous-time model is being executed, the time steps are typically smaller and have equal size.

To establish the correspondence between the model time and real world time where the system being modeled lives, we need to define the time units.

The type of time unit depends on the time scale of the activities you are modeling. For example, if you are

modeling a call center where the call durations are measured in seconds or minutes, you may set the time units to seconds or minutes. If you are modeling a supply chain, where manufacturing and shipping times are measured in days, days would be the right choice. The expression triangular(10, 12, 15) used, e.g., in the Delay object, means a minimum of 10 days, a maximum of 15 days and most likely value of 12 days if day is the time unit.

1. Select the experiment and open its **Model time** property page.
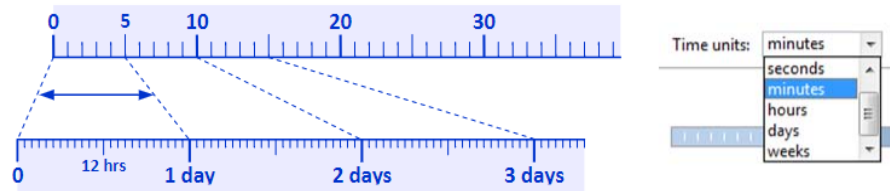
Setting the time units

- long getTimeUnit() – returns the current time unit, namely the number of milliseconds in one time unit. If you are using one of the standard units, a constant from the Table above is returned.
- setTimeUnit(long tu) – sets the time unit to a given number of milliseconds (tu). If you are setting a standard time unit, you can use a constant from the table.

milliseconds	TIME_UNIT_MILLISECOND	1
seconds	TIME_UNIT_SECOND	1000
minutes	TIME_UNIT_MINUTE	60*1000
hours	TIME_UNIT_HOUR	60*60*1000
days	TIME_UNIT_DAY	24*60*60*1000
weeks	TIME_UNIT_WEEK	7*24*60*60*1000

- double millisecond() – returns the value of a one-millisecond time interval.

- double second() – returns the value of a one-second time interval.

For example, if the time unit is hours, minute() will return 0.0166, and week() will return 168.0. Thus, instead of remembering what the current time unit is and writing 48 or 5./60, you can simply write 2*day() and 5*minute(). You can also combine different units in one expression: 3 * hour() + 20 * minute()

Date and calendar

To use calendar in the model you need to tie the start point of the simulation to a particular date. This is also done on the experiment's Model time property page.
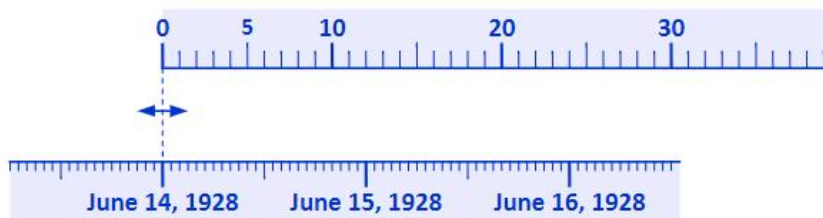
130

Setting the start and stop dates

To set the simulation start date

1.   Select the experiment and open its Model time property page.

2.   Check the CheckBox "Use calendar"

3.   Use the Start date control to set the start date.

Bydefault, the date is set to the date when the model was created.

If STOP is set to STOP at time, the end date of the simulation will be:

$$Start\ date + Time\ unit * (Stop\ time - Start\ time)$$

The date in AnyLogic is stored in the form of the Java class Date.

Date is composed of the year, month, day of month, hour of the day, minute, second and millisecond.

To find out the current date, you should call:

•   Date date() – returns the current model date. A number of functions return particular components of the current date (and all those functions also have the form with parameter <function name>(Date date), in which case they return the component of a given, not current, date):

•   int getYear() – r eturns the year of the current date.

•   int getMonth() – retJANUARY, FEBRUARurns the month of the current date: one of the constants

•   int getDayoffMonth(Y, MARCH, ... int getDayO) – returns the day of the month of the current date: 1, 2, ...
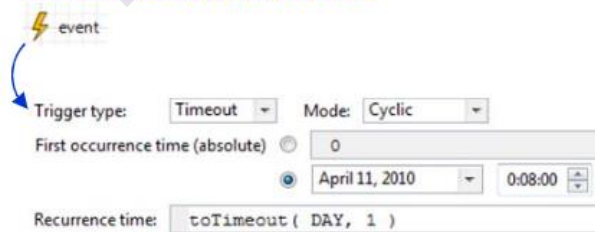
131

- int getDayOfWeek() – returns the day of the week of the current date: one of the constants SUNDAY, MONDAY, ...

- int getHourOhour fDay() – returns the hour of the day of the current date in 24-format: for 10:20 PM, will return 22.

- int getHour() – dafor returns the hour of the y of the current date in 12-hour format: 10:20 PM, will return 10.

- int getAmPm() –PM otherwise returns the constant AM if the current date is before noon.

- int getMinute() – re turns the second within the minute of the current date.

- int getSecond() – returns the minute within the hour of the current date.

- int getMillisecond() – returns the millisecond within the second of the current date.

Consider a model of a processing center that operates from 9 AM to 6 PM on weekdays.
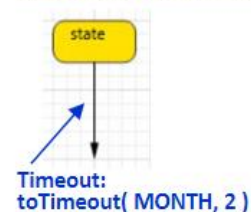
The following function returns true if the center is currently open and falseotherwise: boolean isOpen()

{

int dayofweek = getDayOfWeek();

if(dayofweek == SUNDAY || dayofweek == SATURDAY)

return false;

int hourofday = getHourOfDay();

//will be in 24-hour format   return hourofday >= 9 && hourofday < 18;

}

Event that occurs every day at 8 AM

⚡ event

Trigger type: Timeout ▾   Mode: Cyclic ▾

First occurrence time (absolute) ○   0

                              ◉   April 11, 2010 ▾   0:08:00 ▴▾

Recurrence time: toTimeout (DAY, 1)

Statechart spends in the state exactly two months

state

Timeout:
toTimeout(MONTH, 2)

Source generates an entity at exactly the same time of the day every two weeks

source

Arrivals defined by   ○ Rate   ◉ Interarrival time

Interarrival time   toTimeout (WEEK, 2)

Entities are delayed here for 2.5 years

delay

Delay time:
toTimeout(YEAR, 2.5)

**Using toTimeout() function in events, statecharts and Enterprise Library objects**

Virtual and real-time execution modes.

AnyLogic can execute the simulation model in two modes, virtualtime and real time on a given scale. Virtual time is the "natural" execution mode when the simulation engine executes the model as fast as possible. The model time progresses unevenly and not continuously relative to real time; see the Figure. In discrete event models, the model clock may instantly jump to the next event or may stall at one point while several simultaneous events are being executed. The model execution rate may appear more continuous if the model contains continuous-time dynamics (as in system dynamics models): in that case, the model is driven by the numeric solver, which makes small time steps that are more or less even. The computational complexity of events and equations obviously affects the speed of the execution of the model. The virtual time mode is used when simulation performance is important and animation of the model dynamics is not needed, in particular in optimization, sensitivity analysis, parameter variation, Monte Carlo and other experiments where the model is run multiple times. System dynamics modelers also use the virtual time mode as they are typically interested more in the output graphs of the simulation than in the simulation process itself. Event that occurs everyday at 8AM State chart spends in the state exactly two months Timeout: to Timeout (MONTH,2) Source generates an entity at exactly the same time of the day every two weeks Entities are delayed here for 2.5 years Delay time: to Timeout (YEAR,2.5)

Virtual time ("natural") execution mode
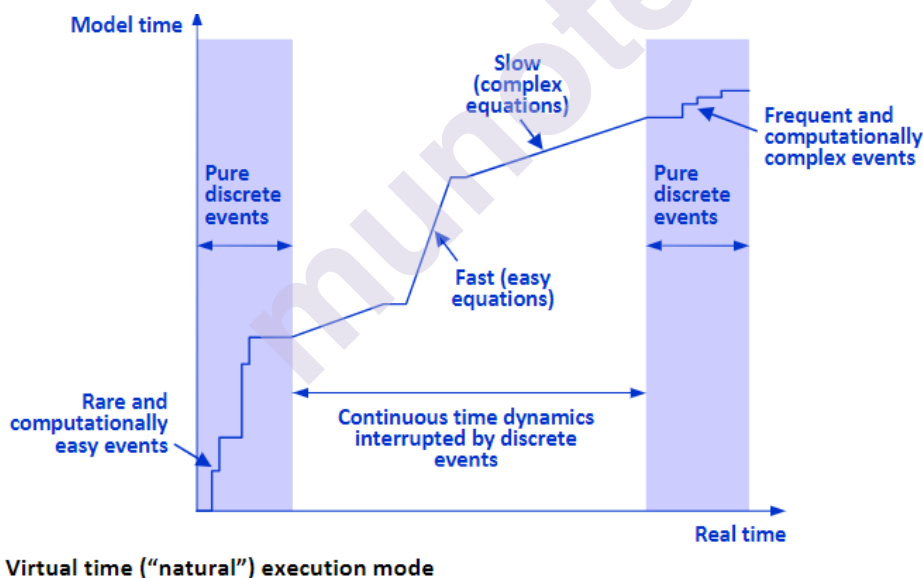
Virtualtime("natural")execution mode : In the scale to real-time mode, the engine tries to keep to a given scale, say 10 model time units (e.g., 10 simulated weeks) per 1 real second. If the model's computational complexity is not too high, the engine will periodically put itself in the "sleep" state and wait for the correct real time to execute the next event or make the next step in the numeric calculations. Sometimes, though, the engine is unable to keep a given time scale because of too-frequent or too-complex events or because of a large system of equations and/or a too-small time step. Then the engine will work as fast as possible until it finds the next

opportunity to maintain the real-time scale. Thus, the only thing the model can guarantee with respect to the real time is that the model execution will never go faster than requested;
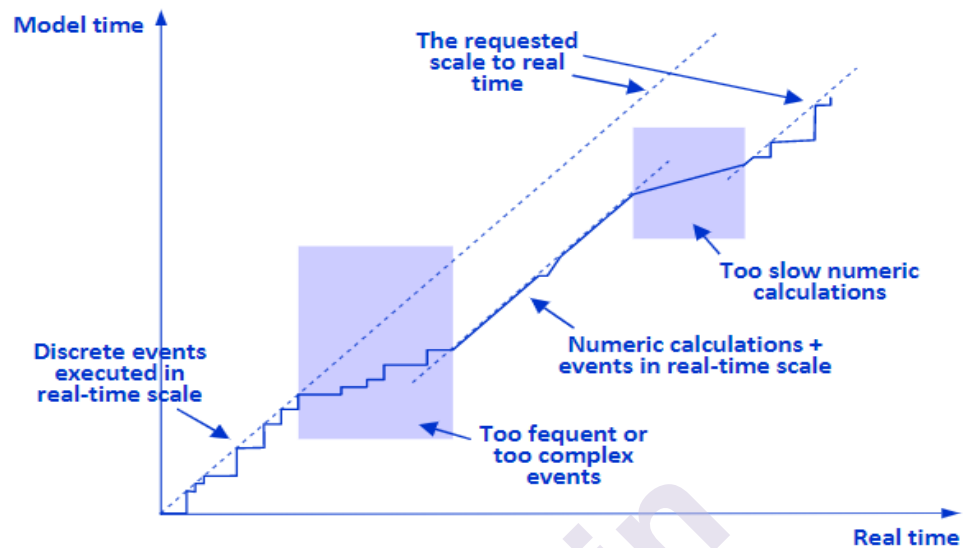
7.8 CONCLUSION

A discrete-event simulation (DES) models the operation of a system as a (discrete) sequence of events in time. Each event occurs at a particular instant in time and marks a change of state in the system.[1] Between consecutive events, no change in the system is assumed to occur; thus the simulation time can directly jump to the occurrence time of the next event, which is called next-event time progression.

In addition to next-event time progression, there is also an alternative approach, called incremental time progression, where time is broken up into small time slices and the system state is updated according to the set of events/activities happening in the time slice. Because not every time slice has to be simulated, a next-event time simulation can typically run faster than a corresponding incremental time simulation.

Both forms of DES contrast with continuous simulation in which the system state is changed continuously over time on the basis of a set of differential equations defining the rates of change of state variables.

7.9 PRACTICE QUESTION

1) Define the following with respect to System Simulation and Modeling:

    a. State

    b. Statecharts

    c. State Transitions

    d. Events

2) Write a short note on State charts.

3) Design a state chart to represent Car Ignition System

4) Describe the events in tea vending machine System

5) How to view and debug Statecharts at runtime? Explain.

6) How to design Statecharts for dynamic objects? Explain using suitable example.

7) Write a short note one discrete events and Event model object.

8) Explain the discreet events associated with queuing system.

9) Write a note on Randomness in system dynamics model.

10) Explain the different procedures of generating random numbers.

11) Explain the Linear Congruential Generator method. Support your answer with suitable example.

12) How to Model time, date and calendar Virtual and real time? Explain.

7.10 REFERENCES

☐ Simulation: The Practice of Model Development and Use by Stewart Robinson, John Wiley and Sons, Ltd, 2004.

☐ The Big Book of Simulation Modeling: Multi Method Modeling by Andrei Borshchev, 2013.

☐ Agent Based Modeling and Simulation, Taylor S, 2014.

☐ Simulation Modeling Handbook: A Practical Approach, Christopher A. Chung, 2003.

☐ Object Oriented Simulation: A Modeling and Programming Perspective, Garrido, José M, 2009.

☐ Simulation, Modeling and Analysis, Averill M Law and W. David Kelton, "Tata McGraw Hill, Third Edition, 2003.

☐ Process Control: Modeling, Design and Simulation, Wayne Bequette W, Prentice Hall of India, 2003.

❄❄❄❄❄❄