Simulation:

System Engineering & Management Tool

John Crocker

Published by MIRCE Science Limited

Editor: Dr J.Knezevic, President MIRCE Akademy

Preface

I was first introduced to the concept of simulation, in particular discrete event simulation, around 1968 by the late Professor Kenneth D. (Toch) Tocher of the United Steels Company (now part of Corus). I remember thinking at the time that it was something I vowed I would never get involved with as I had pretensions of becoming a pure mathematician when I graduated and this was about as far removed from pure mathematics as any branch of the subject could be. At about the same time, I had also made up my mind that if I ever got married, my wife would have a degree and be a non-smoker. My career aspirations were very abruptly brought down to earth in 1970 when I was referred in Applied Mathematics and would have to re-sit these exams the following year if I was to leave university with any qualifications.

With no degree to my name, my job prospects were somewhat limited but I had the great fortune to be offered a job (on the condition that I graduated in 1971) in the Operational Research Group within the recently re-nationalised British Steel Corporation, headed by none other than Professor Tocher. Over the next four years I learnt a great deal about Operational Research, in general, and simulation modelling in particular using GSP (general simulation programme) that was developed by Tocher, et al based on Algol and designed to run on an Elliot 503 computer. In the army, there is a saying, "If it moves salute it, if it doesn't whitewash it". Well, in the OR group we had a similar saying, "If it moves simulate it, if it doesn't linear program it". By the way, GSP eventually became "Witness" and almost certainly provided the inspiration behind many of today's simulation languages.

Four years later, my wife who, needless to say smoked and did not have a degree, and I got married. Shortly after I took up a post within Rolls-Royce's OR Group where I became involved with forecasting and logistics engineering.

It was actually 1994 before I found out that what I had been doing since 1975 was called Logistics Engineering. In that year a whole new world opened before me when I met Dr Jezdimir "Mirce" Knezevic and was accepted by the M.I.R.C.E Centre to undertake a Master of Science degree in Logistics Engineering. I graduated in 1997, this time with out being referred in anything and even went on two years later to take my doctorate which I received in 2001.

At BSC, I had learnt about simulation modelling and had been involved in a number of models but it was not until I took over a model called ORACLE (Operational Research AirCraft Logistics Evaluator) in 1975 that I appreciated just how useful purpose built simulation languages could be. This model was written in FORTRAN, not FORTRAN 90/95, FORTRAN 77 or even FORTRAN 66 but FORTRAN IV (g21). This was in the days when variable names could not exceed 6 alphanumeric characters, arrays had to be defined statically and there was no such thing as a text variable. The program was written on punched cards with probably no more than a dozen comments, which I suspect someone had randomly inserted into the boxes of cards as they appeared to bear no connection to any of the code within their vicinity. In 1990, when ORACLE was finally replaced by MEAROS (Modular Engine Arisings, Repair and Overhaul Simulation) I was still waiting for the documentation that was supposed to accompany ORACLE – I guess it got lost in the post. ORACLE had what was for me, a unique feature; it used part of the memory occupied by the program as an overflow area if any of the arrays went out of bounds. This made error diagnostics rather interesting because if one decided to add a few write statements to find out the values of key variables at certain times in the simulation, this changed the part of the program that became corrupted with the result that the run would invariably either fail at a different time for a different reason or, perversely run to completion apparently without error.

The above is an attempt to explain why I feel qualified to burden the world's libraries with yet another book on simulation. It is not meant to be indicative of the contents. This book is not intended to be historical (or even, alas, hysterical). Its primary aim is to share with Engineers, general, but Logistics Engineers, Systems Operational Scientists and Spares Forecasters in particular, some of the potential benefits of simulation modelling. To this aim, I make no claim to it being either academic or rigorous, if that is what you are after then I would recommend you read one of the books listed in the bibliography, such as George Fishman's *Discrete-Event Simulation* or Byron Morgan's *Elements of Simulation* or, if you can find a copy, Kenneth Tocher's *The Art of Simulation.*

I hope there are not too many errors in the text but if you do find any, I would be most grateful if you could email them to me at John.Crocker@DS-S.com so that I may correct them in the next version.

John Crocker, 2005

Acknowledgements

I would like to thank my wife Frances and daughters Elizabeth and Louise for leaving me alone long enough to allow me write this book. I would also like to thank all of my colleagues over the years who have helped me get to where I am today even if at times (far too numerous to mention) it was not where I wanted to go. But most of all, I must say a massive thank you to Jezdimir Knezevic who opened my eyes onto a whole new world and who has stuck by me providing much needed support and encouragement.

INTRODUCTION

If you believe that the number of spares required is the time the system will be operational multiplied by the failure rate then please put this book back on the shelf where you found it. If however you think it might be more complicated than that and you are interested in finding out how simulation may help you then this could be the book you have been looking for.

Many of the world's warehouses are piled high with components that are quietly obeying the third law of thermodynamics by releasing much of the energy that went into their manufacture as they revert back to the various metallic oxides from which they were derived. It is extremely likely that a very large percentage of these components will never be used for their intended purpose; to replace similar components that have failed in service. Except for a very small number of notable cases, the sizes of these stocks of "spares" will have been calculated using little more than the formula quoted in the opening paragraph.

Since the beginning of the second half of the 20th Century almost all Defence organisations have based their spares policy on the assumption that components only ever need to be replaced if they have failed due to external, non-age-related causes, that the systems containing them will be restored to an as-good-as-new condition if they are repaired (by the replacement of failed components) and that somehow, the reliability of these systems will, if anything, improve with age, like a good vintage wine! The heads of these same organisations, at the same time, however are happy to have their expensive cars serviced regularly and invariably replace them after one, two or, at most three years because they do not want to suffer from high depreciation or pay expensive maintenance bills and, of course, they cannot possibly afford to be inconvenienced by a breakdown. I should add that this phenomenon is by no means restricted to the Defence sector; the same applies to railways, airlines, power generation, process and manufacturing industries and any other operators of high-value, repairable assets.

There are many reasons for this, most of which are beyond the scope of this book. However, there are two that could be considered as falling within its boundaries. The first is Drenick's Theorem[1] and the second is that to use anything other than the exponential probability distribution makes the mathematics extremely complicated. Although Drenick's theorem is mathematically rigorous, it is very commonly assumed to apply to all components of a system rather than to the system itself.

Simulation is one of the few branches of mathematics that does not use theorems, equations, symbols and Greek letters. In Chapter 5 you will find that the basic building blocks are not variables and symbols but entities, events, processes and resources. An entity passes through time (its life) going from one state to another. At its simplest, this could be from busy to idle to busy to, … If we want to get a little more realistic or complicated then we may consider what conditions allow it to change, for example, an

[1] Drenick (1960) proved, that as the number of modes of failure for a system increases indefinitely, the times between failures for such a system operating under steady-state conditions, becomes exponentially distributed asymptotically. In other words, the more components there are in the system to fail and the longer it is operated (at the same rate) in the same environment then the less influence the age of the system will have on the time between failures.

aircraft may only become busy (take-off) if it has a full crew, the passengers and their baggage has been loaded, it has sufficient fuel, a flight plan has been filed, the weather reports are favourable, it has been allocated a runway, Air Traffic Control has given it permission and it has a "landing slot" at its destination. All of these could be modelled as "resources" or they may be regarded as different entities or, if they were not considered relevant to what we wished to model, they simply could be ignored. Suppose the manufacturer of aircraft engines developed the model and their interest was primarily in wanting to know what maintenance and support would be needed over the life of the aircraft. The main drivers of the need for maintenance of aircraft engines are "Engine Flying Hours", EFH, and "Stress Cycles", SC. For airliners it is a reasonable approximation to assume each flight (stage length) accumulates one stress cycle so a non-stop flight from London to Los Angeles, say, would accumulate 10 EFH but only 1 SC. If the flight is cancelled due to the pilot not turning up, bad weather or because take-off has been delayed to the extent that the expected landing time would be after the destination airport's night noise curfew then we can reasonably assume that the engines were not started and therefore did not accumulate any flying time or stress cycles.

Although simulation may not use mathematical symbols, it is nonetheless based on a comprehensive set of proven theorems. I have not included these, hence the title, but I have included Chapters on some of the more fundamental aspects of simulation such as generating and testing pseudo-random numbers – Chapter 2 – and sampling from known statistical distributions – Chapter 3. I would recommend that you skip these chapters, at least on the first reading, returning to them later if you need to understand more about these things before feeling confident to use the technique.

Chapter 4 deals primarily with what many authors refer to as Monte Carlo (MC) simulation. Unfortunately, there are no clear-cut definitions of what constitutes an MC simulation and what a discrete-event (DE) simulation. Rather than try to make a distinction, I have chosen to consider models that can be accommodated relatively easily, without the need for macros, using spreadsheets in this chapter and have left more complicated scenarios to the final two chapters. In general, this restricts the options to those involving only one or possibly two types of component at a time and ignoring the need for or interactions with other resources. Although this may sound rather restricting, there is actually quite a lot that can be done within these constraints. Suppose, for example, you have a component that can fail in one of two ways, such a component might be the inner-tube of a bicycle tyre which can either puncture or perish. We would expect the punctures to take no account of the age of the tube – the sharp-pointed object is unlikely to know (or care about) the age of your tube when it decides to put itself in your path. By contrast, we would be rather unhappy if we found the brand new tube just purchased from the cycle shop is perished when we take it out of the box; we would expect it to last quite some time, several years, maybe, before this should happen. Thus we would expect failures due to punctures to be described adequately using the exponential distribution with its constant failure rate but would expect the failures due to perishing to be age-related, for example, a Weibull with a shape parameter $\beta > 3$. The question that we might want to address is "At what age should we stop patching punctures and replace the tube?".

If your main interest is in how simulation can help you in your line of work then I would recommend you skip Chapter 5 and go straight to Chapter 6 referring back to 5 if you need to understand more about some of the terms used. Chapter 5 describes the main building blocks so, instead of equations with variables and operators (e.g. +, -, x) we have entities, resources, queues, events and processes. Basically, entities "move" from one event to another through time possibly dwelling in queues en route. Processes are little more than a sequence of events all acting on the same entity. These building blocks help us define the crucial elements of the system in which we are interested. With many of the more recently developed simulation languages/packages, the user does not even need to know about these basic units but can simply build models using icons. This makes it much easier and quicker to develop models but it does come at a price, which is the understanding of what is going on in the background.

Chapter 6 describes how an integrated, holistic model might look. It describes many of the characteristics that such a model will need particularly if the system can be regarded as multi-indentured existing in a multi-echelon environment. If we are interested in what really drives the reliability of the system – i.e. the low level components – and how these may be recovered then it is necessary to model the whole system down to this level of detail particularly if the failure of one component may affect the replacement of others or the availability of resources. A model similar to that described in this chapter has been in service since 1969 with a mid-life up-date (rewrite into Simscript II.5) in 1990.
…………………………………………….

CHAPTER 1: THE CONCEPT OF SIMULATION

> *The first duty in life is to be as artificial as possible. What the second is, no one has yet discovered.*
>
> (Oscar Wilde, 1894)

Would you like to know how likely your plane, boat or train will depart on time, how many mechanics you will need to ensure it does, what spares will be required, how much your direct maintenance will cost or how you might reduce the in-service support costs?

Simulation cannot answer any of these questions! But, neither can any other mathematical or probabilistic method.

Simulation provides two major advantages over other types of mathematical model. The first is that neither the modeller nor the users need to understand the complex equations that are needed to describe the system. The second is that if the model is written properly, it is very easy to change the rules and assumptions and see what happens. In many cases, it is impractical or undesirable to try out different scenarios in the real world. With systems such as fleets of vehicles operating over 30 years, the operators, maintainers and other service providers cannot afford to wait to see if their policies were effective. They also do not usually have enough different fleets to be able to try different solutions simultaneously but they do desire to stay in business, which means they have somehow to make sure their costs are less than their revenue.

Many real world problems involving man-made systems involve various levels of uncertainty. Where the past behaviour of such systems can only be described by using statistics, future behaviour can only be predicted using probability theory. If the probabilities of some of the events are believed to be a function of time then simulation is likely to provide the only practical method for modelling the operational behaviour of the system and hence predicting the future.

Simulation has been around for many years, possibly centuries, however it really started life in World War II and then became a practical method for "solving" complex problems in the post-war years with the advent of Operational Research and electronic computers. The nuclear physicists working in Los Alamos on the Manhattan Project to develop a bomb which could release the levels of energy intimated in Einstein's famous equation $E = mc^2$ gave the code word "Monte Carlo" [simulation] to a method which could be used to evaluate the multi-dimensional, intractable integrals describing the phase state of atoms bombarded by neutrons.

Simulation falls into a branch of mathematics that is generally referred to as *Numerical Methods*. It cannot be used as a form of proof, in a mathematically rigorous way. It can often suggest relationships from which we can derive "conjectures" but to prove these we must resort to more rigorous methods but that is outside the remit of this text.

This monograph will list and explain some of the types of problem that can be "solved" using simulation. It is not meant to be a mathematical treatise on simulation methods and, as such, there are no theorems or proofs but that does not mean the method is not rigorous in a mathematical sense, merely which it was decided that this was not necessary or appropriate. There are numerous books on simulation ranging from Tocher's 1962 book "The Art of Simulation" to more recent ones by George Fishman such as "Discrete-Event Simulation" published in 2001.
…………………………………………………

CHAPTER 2: PSEUDO-RANDOM NUMBERS

The generation of random numbers is too important to be left to chance.
(Robert R Coveyou)

A key feature of simulation is the use of random numbers to decide when events will happen and what the outcomes of these events will be. There are a number of possible ways of generating these random numbers.

One way would be to toss a coin n times recording a "tail" as a "0" and a head as a "1". Assuming neither the coin nor the method of tossing it are biased then the resulting n-bit binary number would be a number on the range $[0, 2^n]$ so, if we then divide this number by 2^n the result would be a random number (on the range [0,1]). This method is clearly rather tedious especially when you consider you not only have to enter the number into the computer but that you may need several million such numbers during the course of a simulation.

Any mechanical procedure will inevitably suffer from the same problem - that it is tedious to use and very labour intensive. An alternative might be to feed in a table of random numbers - the sort that could be found at the back of majority of Statistics books. This has the advantage that the numbers have been tested for randomness and that you will not have to generate them. Unfortunately you will still need to type them into the model which is likely to take you several weeks or even months although you would only need to do it once.

Since we are likely to need such a large number, it would be very much more convenient, and practical, if the method of generating these random numbers could be done by the computer from within the program itself (or, more particularly, as a pre-programmed function). For the method to be of practical benefit, it needs to be fast and, somewhat contradictory, reproducible.

………………………………..

CHAPTER 3: Generating Random Samples from Distributions

We've got to stand on our heads, as men of intellect should.
R. Austin Freeman, The Red Thumb Mark

Although we can make decisions, (e.g. which team bats first, who faces the sun, whether we head north, east, south or west, etc.), using uniformly distributed random numbers, most simulations will be required to do more complex tasks. Typically, we will want to decide when events will happen based on some probability distribution.

Before we actually look at how to write a simulation that will make use of such distributions, we will first look at how we can use random numbers to sample from them, or, at least, from some of the more useful and commonly used distributions.

In mathematical terms, if a probability distribution has a cumulative density function $F(x)$ $(-\infty < x < \infty)$ or $F(t)$ $(0 < t < \infty)$ such that it has an inverse function (F^{-1}) then we can use a uniformly distributed random number to sample values of x or t which will tend to be distributed according to the original distribution (as the sample size tends to infinity).

In less formal terms, we can use pseudo random numbers to sample from theoretical probability distributions provided that for every value of p $(0 \le p \le 1)$ there is one, and only one, value of x (or t). Note that it does not matter if a range of values for p maps to a single value of x or t. What is not allowed is for a specific value of p to map into 2 or more values of x or t, because we would not know how to decide which value of x or t to use.

……………………………………………….

CHAPTER 4: Simulation Using Spreadsheets

The scientists working on the Manhattan Project at Los Alamos during World War II gave the code name *Monte Carlo simulation* to a method of evaluating a particular group of integrals of functions that were intractable – i.e. they could not be solved analytically. The method makes use of the central limit theorem and random sampling so is somewhat more closely linked to gambling and gaming casinos than to the Rally for which Monte Carlo is also famous. It was recognised that, although the mean cannot be determined precisely, it is possible to determine, with increasing confidence, that it lies between two limits as the size of the sample (from which it has been estimated) increases indefinitely. Monte Carlo Simulation generally refers to using random sampling to solve specific types of integrals, however there is no universally accepted definition and to avoid confusion, the term will not be used again.
……………………………………………….

CHAPTER 5: Discrete-Event-Based Simulation

"Forty-two!" yelled Loonquawl, "Is that all you've got to show for seven and a half million years' work?"

"I checked it very thoroughly," said [Deep Thought] the computer, "and that quite definitely is the answer. I think the problem, to be quite honest with you, is that you've never actually known what the question is."

Douglas Adams (1979)[1]

Discrete-event-based simulation (which I will simply refer to in future as *simulation*) which allows us to define the operation of a system through time by considering it as a sequence of events each of which changes the state of the system in some way. In general, we are likely to be more interested in how different entities and activities interact rather than trying to solve a mathematical problem, *per se*. It is a technique that has been developed by and for Operational Researchers, although it is by no means restricted to this discipline and has found its way into many areas. In this chapter, we will explore the technique and look at how it can be used to help us achieve systems operational effectiveness.

To start with, we will look at the basic building blocks that this technique requires: events, activities, queues, lists, time sets, processes, resources, tiebreaks and conflicts. The last two are really not building blocks but are two important considerations that need to be taken into account when designing simulations.

A number of specialist computing languages have been written and developed over the past 40 years. Although it is not essential to use such a language, it is certainly worth the investment if you expect to be writing several models or developing one or more particularly complex models. We will investigate the use of BASIC for some simpler scenarios and then go on to look at SIMSCRIPT II.5[2] when it starts to get too difficult to use BASIC.

BASIC is by no means a perfect language for simulation – indeed ALGOL, FORTRAN and PL/1 are all vastly superior (from the programmer's point of view) but, with the advent of PC's and Windows, they are becoming increasingly more difficult to obtain. Conversely, BASIC, which was almost never used when I started my career, has become ubiquitous.

SIMSCRIPT II.5 (hereafter *Simscript*) is certainly not the only simulation language and I am in no position to claim it is in any way the best. The only reason I have used it in this Monograph, is that it is the language with which I am most familiar, having worked with it for the past 20 years, or so. Simscript was developed by the RAND Corporation in the early 1960's and was largely based on FORTRAN. It has moved on a long way since those days but, as someone once said, it is still possible to write bad FORTRAN in Simscript if that is what turns you on!
…………………………………………………

CHAPTER 6: ARISINGS FORECASTING

Imitation is the highest form of flattery
Charles Caleb Colton 1780-1832

In the preceding chapters we have looked, in some detail, at the theory of simulation. We considered stochastic processes, pseudo-random numbers, sampling from different distributions, Monte Carlo simulation and discrete-event based simulation. Now we will concentrate on a particular application in an attempt to show how the theory can be applied to a practical problem.

Before you get too excited, however, it is worth pointing out that the following is intended only as a guide to how to write an arisings forecast. It will attempt to explain why I believe simulation is the best tool for the purpose and, in the process, we will look at the factors that need to be considered.

6.4 Maintenance

Once we have identified most of the components that require maintenance we can leave the "arisings" section and enter the "maintenance environment". At this stage we do not necessarily know the identities of all of the rejected components because in the arisings section we only identified the "minimum echelon capable" and that was done before we had checked for secondary lifing.

[2] Simscript II.5 is a trade mark of CACI Products , 3333 North Torrey Pines, La Jolla, San Diego, California, USA.

In a multi-indenture, multi-echelon environment we will need to first consider where we can perform the first maintenance task. In the case of an engine, which has already been removed from the aircraft, the first task will be to strip/disassemble the engine in order to remove those modules that have either been rejected or contain parts that have been rejected.

At this stage, it will very much depend on how detailed you want to get. We could, for example, decide that all stripping (and subsequent re-building) will be done at the base, 2nd line or "I" level. If this is the case then all we need to do is to move the engine from the squadron, 1st line or "O" level to the base and then go to the next stage. If, on the other hand, our "base" can only perform a limited number of strips (at the same time) or can only strip to a certain depth or is required to only do a certain percentage of the number of engine strips then we will have to consider these constraints.

If the base has a capacity constraint then we can model this quite simply by defining the limit in terms of (Simscript) *resources*. We would then request one of these resources and wait until it becomes free. If it is further complicated by a limit on the number of engines that can be held at the base awaiting stripping then we may need to model this scenario as a queue and check its length. If there is no space left then the engine would have to be sent to another site.

6.5 Supply Environment

The "supply environment" is responsible for storing serviceable components (stock) and for moving components between the various sites either to have them recovered or to meet the requirements of their "parents". As with the other parts of the model, the supply environment can range from the very simplistic to the extremely complex.
…………………………………………………….

CHAPTER 7: Miscellaneous Topics

"What is the use of a book," thought Alice,
"without pictures or conversations?"
Lewis Carroll 1832-1898

7.1 Visual Simulation

Before the advent of electronic computers, if someone wanted to investigate how a complex plant (e.g. iron and steel works, railway system or telephone exchange) they basically had two choices. They could build the plant and watch how it worked in practice or they could construct some form of 2 or 3-dimensional model and "play games" with it.

Obviously, the latter was the cheaper option if it was thought likely that major design changes might be necessary. The problem was that it could be very tedious and labour intensive to simulate a realistic scenario. Someone would need to sample arrival times but before the unit actually "arrived" it would be necessary to check every other unit in the system to see whether they could end their current event and/or start a new one.

This meant someone would have to keep track of what resources were in use, and sample service and transit times, etc. Needless to say the whole thing got very complicated very quickly as the number of units and the complexity of the system grew.

It was therefore not surprising that Operational Research Groups within these organisations were quick to recognise the potential benefits of the electronic computer. Unfortunately, what they could visualise using a physical model was very much more difficult to do using a computer. In the early days of simulation, the idea of producing pictures of what was happening during the simulation was somewhat thwarted by the fact that the early computers were not connected to a VDU (visual display unit). Input was usually by (80 column Holerith) cards or by (5 or 8-hole) paper tape. Output was usually via a line printer onto fan-fold, "music score" paper. If you wanted to follow what was going on at any given time, it was necessary to include "print" statements at the start and end of every event, say. You would then have to trawl through often thousands of lines of output usually heavily coded to minimise the amount and to avoid the need to try to handle characters, as most languages did not support alpha/character/text variables.

With the advent of the PC (personal computer) and workstations, it was, at last, possible to produce pictures although in the early days the possibilities were very limited. Today, however, the situation has changed drastically to the point where many (televised) advertisements and cartoons are produced using computer graphics. Indeed, the computer games that are so popular are really a form of visual, interactive simulations. Having said that, it is not these that I want to discuss in this chapter.

One of the most common uses of discrete, event-based simulation modelling is to study the behaviour of queues. We may be interested in how many spare engines an airline should hold at its base, how many beds should be reserved for patients arriving via the Accident and Emergency Centre or how many checkouts a supermarket should have. In each case with have the concept of a "customer" arriving (usually at some random time or interval) and the concept of someone or something serving or servicing that customer again either over a fixed or variable length of time. Typically, the customer will move through the system from one server to another possibly requiring resources, equipment or facilities *en route*. Servers may be responsible for several tasks that may, or may not, have a predefined priority. As queues build up, customers may choose alternative routes or leave the system altogether (without completing their transactions).

For relatively simple systems, in which the numbers of customers, servers, resources and equipment are small (at any given time), it may be possible to show, on one screen, the whole operation. The arrival of a customer may be shown as a (cartoon) picture of a person entering the "door". They then move to the end of the appropriate queue to join others waiting for the same service/server. As the server finishes one task, the satisfied customer moves on to the next activity and the server (momentarily) becomes idle. The next customer then moves forward to the server (who now becomes busy). This is repeated for the duration of the working day or the length of time the user wishes to simulate.

The user/observer can watch how queues build up and dissipate. He/She can see how much of the time the servers are busy and how many customers can be "processed".

However, unless the system clock is made to run slowly or the system is very complex, the movements of the customers and servers (etc) will be far too quick for anyone to really see what is going on. (Note this did not used to be a problem with the early PC's.)

…………………………………………….

7.2 Optimisation

Normally simulation and optimisation do not make "happy bedfellows". One of the problems with trying to use simulation to optimise is that the cost function will inevitably be a random variable (if the simulation model is stochastic). This means that its value will almost certainly be different each time the model is run with different random numbers. Thus when we compare the cost function values from two scenarios we cannot be certain that A is truly less than B as it could be that we have "picked" a low value from the [unknown] cost function distribution for the first scenario and a high one for the second. If we ran both scenarios again with different PRN, we may or may not see a reversal. Only by running both scenarios a number of times could we get enough data to estimate the type and parameters of the cost function distribution. This is, however, likely to be costly in computing time.

The second problem with simulation is that it is likely to be used to model a complex, real-world problem. These seldom have a single objective or single-valued cost function. If we consider a commercial airline, their primary aim is likely to be to maximise their profits but they will also have to work within a number of constraints. Firstly, they have to work to a given timetable so they need a high level of "dispatch reliability", i.e. a minimum number of flights delayed or cancelled. They will want a high seat occupancy rate, which means achieving a high level of customer satisfaction so they will need to have a good safety record and provide a good-value-for-money service. They will need to be able to demonstrate that once the aircraft has left the ground, there is a very high probability that its next landing will be at the planned destination on or ahead of schedule. Failure to meet some of these criteria may make the airline less attractive but it will certainly not be easy to determine exactly how much this will cost them (in lost revenue or possible compensation payments).

Another major difficulty with stochastic models is that the parameters of the probability distributions are unlikely to be known accurately but will usually have been estimated based on past data (using statistics). Ironically, the least reliable/accurate [parameter] estimates are most probably those of the reliability of the various components being modelled. There is a classic paradox with reliability estimation: the more reliable a component the fewer failures will be observed so the less data there is from which to estimate the reliability distribution parameters and hence the less reliable those estimates will be. At the same time, if a component is particularly unreliable, especially if this might compromise system safety, then the more likely the component will be modified so much of the data collected will not be relevant to the version of the component in service.

One of the primary reasons for modelling systems is to understand, manage and ultimately control those systems. This is rather different from weather forecasting because, as yet, we are unable to have a significant influence on the weather we can expect over the short period of the forecast. We may have a major, albeit detrimental, effect on climate through our burning of fossil fuels and production of "greenhouse" gases but that is quite a different matter. With systems operational effectiveness, the model can and will be used to change the way the system is maintained and supported, possibly even the way it is operated. By modelling different maintenance and support policies we can determine the likely effects on both the in-service costs and system availability over the life of the system.

………………………………………………..

7.3 Benchmarking

As mentioned at the head of this chapter, one of the reasons for using simulation is to help improve the management and control of the system being modelled. The problem is that if we change the rules by which the system is maintained and/or supported then we have no way of knowing whether our actions were better or worse than had we done something else or even nothing.

Given that the forecasts and optimisation are based on statistical data, i.e. using probability distributions, the "accuracy" of the model can be checked without the need to use real data or wait until the system has been retired to check what actually happened against what was predicted. By using reasonably realistic, but fictitious data we can run a forecast for a short term of 1-2 years to generate a sample of times to failure and maintenance task times. These can then be fed back into the model and a full optimisation performed. This will identify the "best" maintenance and support policies. Now, using the original "unknown" parameters together with these recommended policies, the model can be used to "determine" the "true" future. This can then be compared with the optimised results (using the estimated parameters). By repeating this exercise a [statistically] large number of times, a picture of how well the model performs against "reality" can be obtained.

The first part of the exercise is to sample the parameters. The sampling will be done from known distributions (of possible times-to-failure parameters) but in such a way that the user cannot know what the sampled values are. These sampled values need only be to the right order of magnitude so that in the first year or two of operation there are a reasonable number of failures from all of the main reliability drivers. If there is too few, then the accuracy of the estimates of these parameters will be poor but if there are too many, the accuracy could be considered to be too good and the results accruing therefore unrealistic. If there is insufficient data after the first year then the base case can be run on for a second, third or even fourth year until sufficient data has been gathered. The model will need to output the ages at the times of each failure and, at the end of the year, the ages of the components that have not yet failed.

In the case of the transit, strip, rebuild, repair and recondition times, these can either be sampled or they can be assumed. Almost all forecasts currently run use constant values for these times and are usually based on the quoted times. In practice, the actual times will vary from the quoted values and will, almost certainly depend on the number of resources available, the capacity of the repair and overhaul facilities and the throughput of these. Eventually, it is intended to model these facilities in more detail both to improve the forecasts but more importantly to help them improve the services they provide.

By using the same parameters for the maintenance and support, the comparisons between the "actual" and the forecasts will be more representative of the accuracy of the arisings forecasts. It will remove much of the variance that may be caused by the variance in these parameters. Further trials can be carried out with sampled values of these times or even of the parameters of the distributions of these times from which estimates of the contribution of these factors to the overall accuracy can be estimated.

This exercise may sound a little incestuous, however, there is no reason why the "actual" could not be generated using one model and the predictions using a different model. To get a reasonable idea of whether one model is truly better than another, each of the two models should be used to generate a large number of "actuals" which the other model should then attempt to predict. It should be noted that if a model does not have the capability to simulate a particular action or type of behaviour then the modeller has, consciously or unconsciously decided that the accuracy of the forecasts is independent of this action or type of behaviour.

Referring back to the submarine model in Chapter 2, we might have decided not to differentiate between night and day or have considered the depth of the submarine was unimportant. If all searches were to be carried out in daylight then clearly there would no need to include the probabilities of night time detection. If however, this was not the case, then we would need to modify the probabilities of detection during daylight to take account of the night flying. Similarly, we could replace the three probabilities of detection based on the depth of the submarine by a single probability weighted by the amount of time the submarine is expected (or thought) to spend at each depth. In this particular example, it should make very little difference to the final results because we are not looking at different behaviour based on the depth only the different probabilities of detection. In practice, a diesel-electric submarine generally has a much slower speed when submerged and without the aid of a snorkel can only stay submerged for a limited amount of time. Visibility would also be affected by weather conditions as, indeed, would flying so there could be periods when the aircraft would be grounded or would need to fly the legs much closer together to have a chance of seeing the enemy. It is also possible that lookouts on the submarine might spot or even hear it the aircraft and hence take avoiding action.

Any model, whether it is mathematical or simulation, must be a simplification of reality. The art of the modeller is in deciding what needs to be included and what can be omitted, what needs to be modelled in minute detail and what can be dealt with by a simple probability. If the model is to have a future, however, it is important for the modeller to frequently talk to the users to suggest what they could get from the model as well as to listen to their needs. Modellers can misunderstand the users' requirements possibly because the user does not always know what they are himself/herself. Users equally can have a misconception of what the model can and does provide – very often they will ask for outputs because they believe these are easy to produce rather than because they are what they really need.

In general, it is better to start with a simple model and add in complexity as and when it is needed. Although it may be possible to remove unwanted complexity, it is generally a great deal more difficult than to add it plus one is always tempted to think it might come in useful at some time in the future (when the users have appreciated how much effort it took to include, how clever it is or even when they "know what they are doing). There is a tendency for modellers to believe they understand the systems being modelled better than the people using them!
…………………………………………………………………………..

BIBLIOGRAPHY

The art of simulation (Electrical engineering series)
K. D Tocher

Elements of Simulation
Byron J.T. Morgan

Simulation Modeling and Analysis (Industrial Engineering and Management Science Series)
Averill Law, W. David Kelton (Hardcover)

Discrete-event simulation: from the pioneers to the present, what next?
Journal of the Operational Research Society, **56.6** Pp 619-629

Handbook of Simulation : Principles, Methodology, Advances, Applications, and Practice
Jerry Banks (Editor)

Discrete-Event System Simulation, Fourth Edition
Jerry Banks, et al

Introduction to Simulation and Risk Analysis
James R. Evan, David Louis Olson

Discrete Event Simulation for Performance Evaluation of Systems
Feldman

Concepts and Methods in Discrete Event Digital Simulation
George S. Fishman

Discrete-Event Simulation
George S. Fishman

Principles of Discrete Event Simulation (Wiley Series on Systems Engineering & Analysis)
George Samuel Fishman

Art and Techniques of Simulation (Quantitative Literacy)
Mrudulla Gnanadesikan, et al

Simulation Modeling Methods: To Reduce Risks and Increase Performance (CD-ROM included)
H. James Harrington, et al (Hardcover)

Discrete Event Simulation and Operations Research
Philip Heidelberger

Simulation With Arena
W. David Kelton, et al

A Guide to Monte Carlo Simulations in Statistical Physics
David Landau, Kurt Binder

Discrete-Event Simulation : A First Course
Larry H Leemis, Stephen K. Park

Simulation Techniques for Discrete Event Systems (Cambridge Computer Science Texts)
I. Mitrani

A history of discrete event simulation programming languages (TR)
Richard E Nance

Stochastic Modeling : Analysis & Simulation
Barry L. Nelson;

Introduction to Simulation Using Siman
Claude Dennis Pegden, et al

Discrete Event Simulation: A Practical Approach
Udo W. Pooch, James A. Wall

Computer Simulation Applications: Discrete-Event Simulation for Synthesis and Analysis of Complex Systems.
Julian Reitman

Simulation
Sheldon M. Ross

A comparative study of some discrete event simulation languages (Norwegian Computing Center/SIMULA information publication)
Antti Virjo

Simulation Modeling Using @RISK: Updated for Version 4
Wayne L. Winston (Paperback)

Theory of Modeling and Simulation
Bernard P. Zeigler, et al (Hardcover)

[1] Douglas Adams, *The hitch-Hikers Guide to the Galaxy*, 1979, Pan