Topic 8

Simulation and Queueing Theory

Contents

8.1 An Introduction to Simulation

Simulation enables the study of, and experimentation with, the interactions of a complex system (or a subsystem thereof). Informational, organisational, and environmental changes can be simulated and the changes to the model's behaviour can be observed. Knowledge gained by studying the model can be of great value, and may suggest improvement to the system under investigation. Changing simulation inputs can help understanding how certain components of the system interact. Simulation is, among others, used to experiment with new designs or policies prior to implementation, and may save huge amounts of money. Simulations are also used to verify analytical solutions.

According to the online version of the Encyclopedia Britannica, 'a [computer] simulation uses a mathematical description, or model, of a real system in the form of a computer program. This model is composed of equations that duplicate the functional relationships within the real system.'

Following Pegden, Shannon, and Sadowski (1995) we list some of the advantages and disadvantages of simulations:

- New policies, operating procedures, decision rules, information flows, organisational procedures, etc. can be tested without disrupting ongoing operations of the real system.
- New hardware designs, physical layouts, transportations systems, etc. can be tested without committing resources.
- Hypothesis about how and why certain phenomena occur can be tested for feasibility.
- Time can be compressed or expanded for a speed up or slow down of phenomena under investigation.
- Insight can be obtained about the interaction of parameters.
- Insight can be obtained about the importance of variables.
- Bottleneck analysis can be performed.
- A simulation can help in understanding how a system works, opposed to how individuals think the system works.
- Questions like 'what if' can be answered.

In addition, concise replications of simulation runs are possible which is sometimes important at early stages of a simulation project. Finally, people who are observed do often behave differently compared to times when they are not observed. This 'Hawthorne effect' is obviously not present in a computer simulation.

There are also some disadvantages in using simulations:

- Model building requires special training, and designing complex simulations is an art form. In practice, two models designed by different competent people will usually differ considerably, although there will be similarities.
- Simulation results may be difficult to interpret.
- Simulation modeling and analysis can be time consuming and expensive.
- Simulation is sometimes used where analytical models are available and even preferable.

There are further limitations to those listed by Pegden, Shannon, and Sadowski (1995). Simulation is an experimental problem solving technique. As said before, when available analytical (or 'close') solutions should be preferred since they are more accurate. To put this last statement differently, solutions found by simulations often tend to be suboptimal. The solution may be the best found after running several simulations with different parameters, but this is in general no guarantee that the solution is indeed the best available (which can usually only be proven using analytical methods). Simulations also come with a price tag. Developing a good simulation is a complex task usually performed by highly skilled people. In addition, substantial computing power may be required to run a large simulation process. None of this comes for free. In fact, even finding the right people to develop the simulation can be difficult, since those people need next to solid statistical and programming skills also deep insight into the problem or environment for which the simulation is to be developed. Finally, due to the lack of analytical methods it is often difficult to detect or recognise errors within the design or implementation of simulations.

The following are a couple of potential applications of simulation related to computer science, information technology, and electrical engineering (in the broadest sense):

- Material handling system design for semiconductor manufacturing
- Assembly operations
- Distributed models for computer integrated manufacturing
- Inventory cost model for 'just-in-time' production
- Heterogeneous networks
- Evaluating large-scale computer system performance
- Client/server system architecture

The list above suggests the following guidelines when to use simulations and models:

- real systems may not (yet) exist, may be expensive, time consuming to create, or hazardous;
- experimentation with the real system may be expensive, dangerous, or likely to cause serious disruption;

- there may be the need to study past or future behaviour of the system, for example in slow motion;
- analytical modelling of the system may be impossible;
- even if analytical modelling is possible there may not be a simple or practical solution available;
- validation of the model and the results is possible;
- models can be tuned at any desired accuracy;
- only incomplete information about the real system is available.

As said before, analytical methods should be preferred over simulation techniques provided that those analytical methods are available. Thus we can formulate the following decision procedure:

- Formulate problem
- Check for existing models. If such models (whether analytical or simulation models) are available then solve the model and analyse the solution.
- If there are no models relevant to the problem then one should first look for an analytical model. An alternative might also be to enumerate all possible alternatives and use evaluation procedures to select the optimal alternative. Only when these options fail should one proceed to developing a simulation model, use the model and analyse the data gained from the simulation.

8.2 Models

A model is any simplified representation of an object or a system. Models are used for analysing, understanding, or explaining an object or a system. S.E. Elnaghraby (The Role of Modeling in I.E. design, J. Industrial Engineering 1968) defines the role and purpose of models as follows: Models are

- an aid of thought;
- an aid to communication;
- a purpose of training and instruction;
- a tool of prediction;
- an aid to experimentation.

Models can be classified according to their nature, which can be physical, symbolic (mathematical, numerical), or procedural (simulations). Physical models often have high costs. Symbolic models, though often cheap, are difficult to communicate to people with less mathematical training. Procedural models offer a good balance between costs and use since they are easy to communicate.

Models can be further classified according to various characteristics. The following are only some characteristics often mentioned in the literature:

- static versus dynamic;
- aggregate versus detailed;
- computer versus human;
- continuous versus discrete;
- deterministic versus stochastic.

Most mathematical models contain certain parameters that describe how the model works or what the input or output of the model are. We distinguish between exogeneous and endogeneous parameters (or variables). Exogeneous parameters represent external parameters and may be seen as input parameters. An example are parameters in the input probability distribution. Endogeneous parameters of a model are those that are predicted by the model, i.e., the status or output variables.

8.3 Steps in Simulations

Designing a simulation consists at least of the following steps:

- Setting objectives/Overall design
  - Problem formulation
  - Setting objectives and overall project plan
- Model building
  - Model conceptualisations
  - Data collection (for input)
  - Model translation
  - Verification of the (computer) model
  - Validation (accurate description of the real world)
- Running the model
  - Experimental design
  - Production runs and analysis
- Documentation and reporting

The last steps are also often summarised as strategic planning, and include experimentation, interpretation of the results and the final implementation of the findings. The important parts of the simulation steps are contained in Figure 8.1:

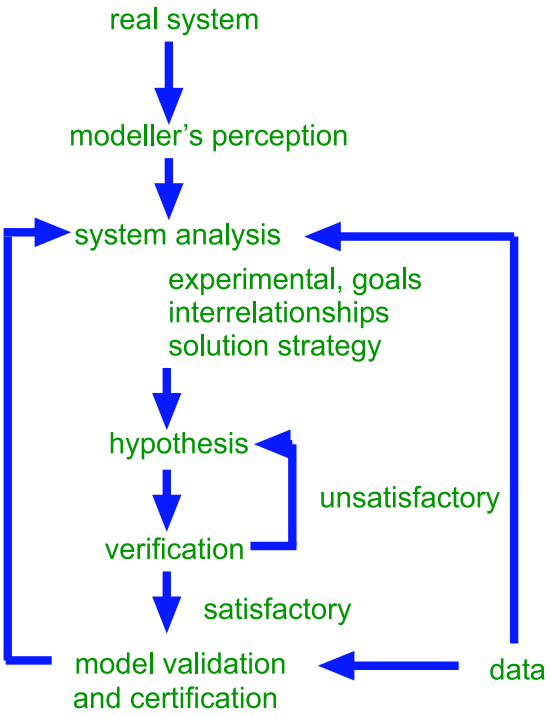


Figure 8.1:

Some of these steps will be discussed in detail below.

8.3.1 Problem Formulation

The problem formulation is maybe the hardest part in designing a simulation. Typical obstacles are the following:

- The problem formulation given to the engineer is often vague. In many cases this is just the statement that there seems to be a problem somewhere.
- Related to the previous point, the problem may not yet be correctly identified, let alone that there is a clear and concise problem formulation.
- The system for which a simulation is to be designed may not be well understood. This may be either due to the complexity of the system, or due to the engineer not being acquainted with the system.

As a result most simulations start with an orientation study which aims to get a firm understanding of the system in question. Also, objectives and alternatives are continuously generated through the study and may be subject to change according to the insights already gained.

8.3.2 Model Building

Model building is an art. Indeed, good model builders are equally engineers and artists. As a result, models built by different groups for the same purpose will usually look quite different. Such models will have many features in common, but there will also be substantial differences due to the different perception of the real world. The following is a minimal list of requirements for any good model:

- The model should model and show the operations of the system in question.
- The model should provide a solution to a real world problem.
- The model should be for the benefit of those who asked for the design of the model.

At the model building stage we can reiterate some of the common problems: Good models are expensive and time consuming. This applies both to the human factor, i.e., to the people involved, as well to computing power. Models are almost always imprecise since models are almost never a complete picture of the real world. Models may appear to be accurate where in fact they aren't. The converse is also true, where models may be accurate where they don't appear to be. Another common problem is that with the design of a (mathematical) model we will associate numbers to certain data. This may suggest a greater degree of validity to the model or its outcome than is actually justified.

8.3.3 Input Modelling

Once the model is designed data has to be collected and parameters have to be estimated. This is where all the statistical methods are applied:

- Input data has to be collected for the various input processes.
- Possible input distributions have to be selected to represent those input processes. The choice is between theoretical distributions or empirical distributions. Goodness-to-fit hypothesis testing is used to determine whether collected input data matches a given theoretical distribution.
- If the model is used to predict effects of new procedures then there may not be any data to collect. In this case theoretical distributions have to be chosen to model various input processes.
- Parameters of the distributions have to be estimated using either point estimators or interval estimators.

Let us state again the main use of the most important probability distributions:

- **Binomial distribution.** This distribution models the number of successes in n successive trials, where the trials are independent from each other and there is a common success probability p.
- **Poisson distribution.** This distribution is used to model arrival processes in fixed time intervals where arrivals are independent. Thus this distribution can be used to model arrivals of customers at a counter, or arrivals of printer jobs at the printer queue.
- **Exponential distribution.** The exponential distribution models time between independent events. A typical use of this distribution is the time *between* arrivals at a counter. If arrivals are modeled by the Poisson distribution then inter-arrivals are modeled by the exponential distribution.
- **Normal distribution.** The normal distribution is the most common distribution. It often models errors or processing time due to its symmetric character.

- **Weibull distribution.** Finally, the Weibull distribution is often used to model time-to-failure of systems or system components.

Once the choice is made for a particular distribution then the parameters have to be estimated. The following is a list of parameters for some of the more common distributions and the common point estimators for their parameters. For more complete tables see for example Banks, p. 370.

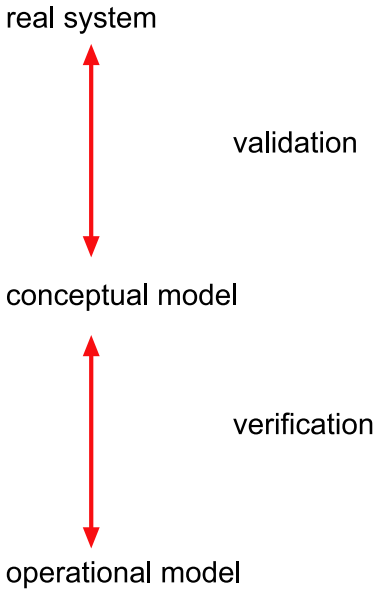
Distribution	**Parameters**	**Suggested Estimator**
Poisson	λ	$\widehat{\lambda} = \overline{x}$
Exponential	β	$\widehat{\beta} = \frac{1}{\overline{x}}$
Uniform on (0, b)	b	$\hat{b} = \frac{n+1}{n} \max x$
Normal	μ,σ^2	$\widehat{\mu} = \overline{x}$, $\widehat{\sigma^2} = s^2$

If no input data is available then parameters have to be estimated. The parameters can then be based on engineered data or on expert opinion. The physical nature of the process or natural conventions may impose limitations on the choice of the parameter.

Some of the parameters of the simulation may also need starting or initial values. Those may come again from observed data or from expert opinion.

8.3.4 Verification and Validation

Verification and validation are two important steps in design. Verification refers to whether a model behaves as intended. Validation refers to the agreement between a model and the real world. Verification and validation are also seen as answers to the questions 'are we building the model right?' (verification) and 'are we building the right model?' (validation). Thus validation relates the real world with the conceptual model, while verification relates the conceptual model with the operational model:



Validation aims to establish confidence into the model, i.e., confidence that the model is indeed a true picture of the real world. A model that is not validated is only of academic

interest and of hardly any use in decision making or analysing a real-world situation. Since validation is part of the art no prescribed mechanism exists for it. However, basic guidelines are the following:

- Results (say outcomes of the model) must be reasonable. This is often established with test data where the correct outcome is known beforehand.
- Testing of assumptions.
- Testing of input-output transformations, i.e., does a certain variation of the input parameters result in reasonable changes of the output results.

Validation is usually achieved through common sense and logic, by taking advantage of knowledge and insight available, by empirical testing, by paying attention to details, by debugging of the program, by input-output analysis and comparison with real-life data, and by checking predictions. From this list it becomes clear that next to data and results we also validate concepts, methodology, and inferences.

Calibration of the model may be seen as part of the validation process. Every model is only a partial representation of the real world. Calibration is the fine tuning of the validated model. It describes how accurate or inaccurate the model is. Calibration is the term used for the fine tuning of parameters of a system (like input distributions, parameters of the distributions, etc.). To be able to calibrate the major part of the model has be be validated.

A test often used in validating models is the Turing test. Real reports are mixed with fake reports based on model predictions. If experienced people cannot tell apart the reports then this suggests that the model in question is an accurate picture of the real world.

Verification is the formal process whereby it is checked that the implementation of the model is an accurate representation of the conceptual model. Typical verification techniques include the following:

- code is checked by someone else;
- use of flow charts or other visual aids in the design of the implementation;
- examination of output and comparison with real life;
- careful debugging;
- thorough documentation;
- examination of subsystem and subsystem testing.

Again there are several aspects of the verification process. We verify the structure of the program, the algorithm, but also the robustness of the code.

8.3.5 Output Analysis

The output analysis is often of stochastic nature. It can be performance measurements or estimations of performance. The output analysis can also result in confidence intervals for certain output parameters with specified confidence.

The analysis has to be done carefully, since it is easy to assign numbers too much validity and forget that they come from a model.

8.4 Queueing Systems

Most simulations contain queues as part of the model. Queueing theory refers to the mathematical models used to simulate these queues.

Calling populations are often assumed to be 'infinite' if the real population is large. This simplifies the model. For infinite populations the *arrival rate* is not affected by the number of people that left the population and entered the queue.

A *queueing system* consists of at least a server, a waiting line, and a calling population. Examples are

System	Customer/Population	Server
reception desk	people	receptionist
garage	cars	mechanics
warehouse	pallets	crane
computer	jobs	CPU, discs or tapes
telephone	calls	exchange

Typical examples with 'infinite' populations are

- people being served at an ATM,
- cars served (serviced) in a garage,
- printer jobs at a local network,
- requests to a web server

However, the one company helicopter that can be serviced is definitely a finite population.

In many queueing systems there is a limit to the length of the queue:

- A buffer holding requests to a CPU may have a fixed length.
- The buffer of a printer usually has a fixed length.
- A lane of a drive-in restaurant may only hold 10 cars.

This is *not* to be confused with customer behavior, where customers will not queue if there are already ten customers waiting in line to be served. Thus, the *system capacity* is a real constraint of the system, and an important parameter in a simulation.

The arrival process is characterised by the time intervals between customer arrivals. Such arrivals may be *scheduled*, or occur at random times, in which case they are characterised by a probability distribution. For example, if arrival of customers is

modelled by a Poisson process with mean λ (arrivals per time unit), then inter-arrival time is modeled by an exponential distribution with mean λ^{-1} (time units). The expected number of arrivals in t time intervals is thus λ. The Poisson process is often used to model arrival of customers.

Next to scheduled and random arrivals there is also the possibility that there is *always* one customer in the queue. This happens for example if the customer represents raw material, and the production process guarantees that such material is always present.

Queue behaviour refers to the actions customers take while arriving or waiting in the queue. Customers may

- balk (leave, if the queue is too long),
- renege (leave, if the queue proceeds too slowly), or
- jockey (move between lines according to what the customer thinks is the shortest/fastest line).

Discipline refers to the logical order in which customers are served. Often used models are

- FIFO (first-in-first-out), LIFO (last-in-first-out);
- SIRO (service in random order);
- SPT (shortest processing time first);
- PR (service according to priority).

Example

Problem:

As a first example we look at a simple queue model which has an infinite population, one queue, and one server.

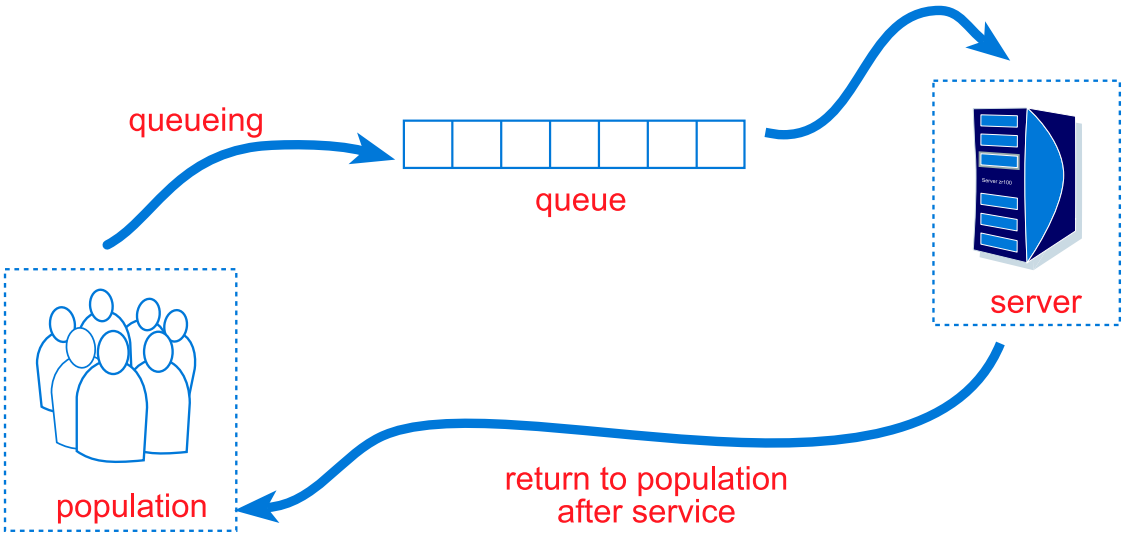


Figure 8.2: One server one queue model

Solution:

Suppose inter-arrival times are determined by rolling a die. If the numbers 6, 1, 4, 3, 6, 5 are rolled this means that the customers arrive at times

0,6,7,11,14,20,25

as in the following table:

Customer	Interarrival time	Arrival time on clock
1	–	0
2	6	6
3	1	7
4	4	11
5	3	14
6	6	20
7	5	25
8	3	28

Suppose also that the service time for each customer is

2, 3, 1, 1, 1, 1, 2

time units respectively, then we can extend our table to include service begin and time to get served to get the following table representing the time spent by customers in the system:

Customer	Arrival time	Service begin	Service time	Service ends
1	0	0	2	2
2	6	6	3	9
3	7	9	1	10
4	11	11	1	12
5	14	14	1	15
6	20	20	1	21
7	25	25	2	27

Note that, for example, customer 3 has to wait for 2 time units before being served since the server is still busy serving customer 2. After serving customer 3 (at time 10) the server is idle for 1 time unit until customer 4 arrives, which is then served immediately.

8.4.1 Worked Example

On the following pages we will go through a single queue system in some detail. We consider a server model with inter-arrival times between 1 and 8 time units. The probability of each time interval of length 1 is 1/8. We will use random numbers to generate customer arrivals as follows:

Time betw. arrivals	probability	cum. prob.	random digits
1	0.125	0.125	$001 - 125$
2	0.125	0.250	$126 - 250$
3	0.125	0.375	$251 - 375$
4	0.125	0.500	$376 - 500$
5	0.125	0.625	$501 - 625$
6	0.125	0.750	$626 - 750$
7	0.125	0.875	$751 - 875$
8	0.125	1.000	$876 - 000$

Thus, if we generate random numbers uniformly between 0.000 and 0.999 and if the random number is 0.371 then this number represents the time interval 3 between inter-arrivals. (If you draw the cumulative distribution and use the inverse transform technique then it is exactly the table above you arrive at.)

We do the same for the service time which is between 1 and 6 time units and arrive at the following table:

Time betw. arrivals	probability	cum. prob.	random digits
1	0.10	0.10	$01 - 10$
2	0.20	0.30	$11 - 30$
3	0.30	0.60	$31 - 60$
4	0.25	0.85	$61 - 85$
5	0.10	0.95	$86 - 95$
6	0.05	1.00	$96 - 00$

To run a simulation we will generate now random digits to simulate customer arrivals and service time. The tables below were generated using the C++ random number generator. There are also random numbers available in books of statistical tables which can be used.

Interarrival time

customer	random digits	time	customer	random digits	time
1	-	-	11	535	5
2	629	6	12	127	2
3	093	1	13	606	5
4	261	3	14	879	8
5	202	2	15	375	4
6	922	8	16	897	8
7	361	3	17	293	3
8	194	2	18	784	7
9	503	5	19	610	5
10	632	6	20	061	1

Service time

customer	random digits	time	customer	random digits	time
1	55	3	11	47	3
2	65	4	12	69	4
3	37	3	13	45	3
4	25	2	14	47	3
5	44	3	15	30	2
6	78	4	16	68	4
7	84	4	17	31	3
8	49	3	18	20	2
9	68	4	19	92	5
10	66	4	20	49	3

We now put the data together and have a look how the 20 customers proceed through the system (see below for the complete table).

- Customer 1 arrives at time 0 and has 3 time units service time, leaving the system at time unit 3. In total the customer spent 3 time units in the system, and the server was not idle.
- Customer 2 arrives 6 time units after customer 1, thus arrives at time unit 6. Since the server is idle the customer will be served immediately, leaving the system at time unit 10 since the service time was 4 time units. The server was idle for 3 time units (from time unit 3 until 6, after finishing serving customer 2 at time unit 3 and waiting for the next customer to arrive at time unit 6).
- The rest of the table is generated similarly.

Customer	Time since last arrival	Arrival time	Service time	Start service	Waiting (queue)	End service	Time spent in system	Idle time
1	0	0	3	0	0	3	3	0
2	6	6	4	6	0	10	4	3
3	1	7	3	10	3	13	6	0
4	3	10	2	13	3	15	5	0
5	2	12	3	15	3	18	6	0
6	8	20	4	20	0	24	4	2
7	3	223	4	24	1	28	5	0
8	2	25	3	28	3	31	6	0
9	5	30	4	31	1	35	5	0
10	6	36	4	36	0	40	4	1

Customer	Time since last arrival	Arrival time	Service time	Start service	Waiting (queue)	End service	Time spent in system	Idle time
11	5	41	3	41	0	44	3	1
12	2	43	4	44	1	48	5	0
13	5	48	3	48	0	51	3	0
14	8	56	3	56	0	59	3	5
15	4	60	2	60	0	62	2	1
16	8	68	4	68	0	72	4	6
17	3	71	3	72	1	75	4	0
18	7	78	2	78	0	80	2	3
19	5	83	5	83	0	88	5	3
20	1	84	3	88	4	91	7	0
			66		20		86	25

Let us summarise some of the data:

1. The average waiting time per customer is 1 time unit.

$$\text{average waiting time} = \frac{\text{total waiting time}}{\text{number of customers}} = \frac{20}{20} = 1$$

2. The probability for a customer to wait in the queue is 45%.

$$\text{Probability (waiting)} = \frac{\text{total number of waiting customers}}{\text{number of customers}} = \frac{9}{20} = 0.45$$

3. The proportion of idle time of the server is 27%.

$$\text{Probability (server idle)} = \frac{\text{idle time}}{\text{total run time}} = \frac{25}{91} \approx 0.27$$

4. The average service time is 3.3 time units.

$$\text{average service time} = \frac{\text{total service time}}{\text{number of customers}} = \frac{66}{20} = 3.3$$

Note that the expected service time is slightly lower,

$$E(\text{service time}) = 1(0.1) + 2(0.2) + 3(0.3) + 4(0.25) + 5(0.1) + 6(0.05) = 3.2$$

5. The average time between arrivals is 4.42 time units.

$$\text{average interarrival time} = \frac{84}{19} \approx 0.27$$

The expected time between arrivals is $E(A) = \frac{1+8}{2} = 4.5$ time units.

6. The average waiting time for customers that have to wait is 2.22 time units.

$$\text{average waiting time} = \frac{\text{total waiting time}}{\text{number of waiting customers}} = \frac{20}{9} \approx 2.22$$

7. The average time spent by a customer in the system is 4.3 time units.

$$\text{average time spent in system} = \frac{\text{total time spent}}{\text{number of customers}} = \frac{86}{20} = 4.3$$

Alternatively, the average time spent in the system is the average waiting time plus the average server time, which is 1 + 3.3 = 4.3 .

8.4.2 Equations for a Single Server

There are some useful formulae that can be derived for the standard queueing situation with one server. It is assumed items arrive for service according to a Poisson distribution with mean λ. Assume also FIFO, no balking and no limit on queue size. The service time is taken as following the negative exponential distribution, with mean service rate μ.

This is sometimes summarised as a M/M/1/∞ system.

(the two "M"s are the mean values above, "1" is 1 server and "∞" is because there is no limit on queue size.)

In the following equations, the "system" refers to both waiting in the queue and being served.

Average number in system,

$\overline{S} = \frac{\lambda}{\mu - \lambda}$

Average number in queue,

$\overline{q} = \overline{S} - \frac{\lambda}{\mu} = \frac{\lambda^2}{\mu(\mu-\lambda)}$

Average queueing,

$\overline{W} = \overline{S} \times \frac{1}{\mu} = \frac{\lambda}{\mu(\mu-\lambda)}$

Average time in system,

$\overline{t} = \overline{W} + \frac{1}{\mu} = \frac{1}{\mu - \lambda}$

In addition:

Probability of no customer in system,

$P(0) = 1 - \frac{\lambda}{\mu}$

(it is required that $\lambda/\mu < 1$)

Average number in the queue when it is not empty,

$\overline{q} = \frac{\mu}{\mu - \lambda}$

Example

Problem:

Application forms arrive at a clerk's desk for processing on average every 4 minutes. Arrivals are assumed to follow Poisson distribution. The service rate is, on average, 20 per hour. Calculate the various queue characteristics.

Solution:

Arrivals: One form arrives every 4 minutes, so there are 15 in an hour. This means that λ = 15.

Service: μ = 20

$\overline{S} = \frac{\lambda}{\mu - \lambda} = \frac{15}{20 - 15}$ = 3 forms

$\overline{q} = \overline{S} - \frac{\lambda}{\mu} = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{15^2}{20(20 - 15)}$ = 2.25 forms

$\overline{W} = \overline{S} \times \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{15}{20(20 - 15)}$ = 0.15 hours = 9 minutes

$\overline{t} = \frac{1}{\mu - \lambda} = \frac{1}{20 - 15}$ = 0.2 hours = 12 minutes

---

8.4.3 A Two-Server Model

We consider a second simulation. Here two servers serve one queue. We assume that a customer is served by the server that is available. If both servers are available then the customer is assigned randomly to one of the servers. The servers take different times to serve a customer (these could be mechanics, one being more senior and faster than the other, or network servers, where one model is older than the other). Below are the distribution functions for the two servers $A$and B, and the distribution for the arrival time between customers.

Inter-arrival time:

Inter - arrival time	Prob.	Cum. prob.	Random digits
1	0.25	0.25	$01 - 25$
2	0.40	0.65	$26 - 65$
3	0.20	0.85	$66 - 85$
4	0.15	1.00	$86 - 00$

Server A:

Service time	Prob.	Cum. prob.	Random digits
1	0.10	0.10	$01-10$
2	0.30	0.40	$11-40$
3	0.25	0.65	$41-65$
4	0.20	0.85	$66-85$
5	0.15	1.00	$86-00$

Server B:

Service time	Prob.	Cum. prob.	Random digits
2	0.35	0.35	$01-35$
3	0.27	0.62	$36-62$
4	0.20	0.82	$63-82$
5	0.18	1.00	$83-00$

The following three tables contain a run of this simulation dealing with 29 customers.

- The first column shows the customer number.
- The second column shows two random digits used to determine the inter-arrival time (column 3) and the clock time at arrival (column 4).
- The next column contains again two random digits, this time for determining the service time. Note that since server A and B have different service distribution we cannot determine how long a customer will spend being serviced until we have determined which server will serve the customer.
- The next 6 columns show the data for both servers, which are time when service starts, service time for the particular customer (determined using the two distributions above and the random digits of column 5), and the time when service finishes.
- In some cases (like for customers 3, 7, 8, 9, and 10) both servers are idle and can serve the customers. In those cases a random number was used to determine which server will serve the customer. Thus, in those cases the name of the server in column 6 was determined randomly.

Customer	Random digits (arrival)	Time betw. arrivals	Clock time of arrival	Random digits (service)	Server	A start	A service time	A end	B start	B service timer	B end	Time in queue
1	-	-	0	75	B				0	4	4	0
2	56	2	2	29	A	2	2	4				0
3	27	2	4	9	B				4	2	6	0
4	60	2	6	62	A	6	3	9				0
5	18	1	7	79	B							0
6	2	1	8	37	A	9	2	11	7	4	11	1
7	92	4	12	28	A							0
8	73	3	15	61	B	12	2	14	15	3	18	0
9	93	4	19	19	A	19	2	21				0

Figure 8.3: Simulation (a)

Customer	Random digits (arrival)	Time betw. arrivals	Clock time of arrival	Random digits (service)	Server	A start	A service time	A end	B start	B service timer	B end	Time in queue
11	36	2	24	10	A	24	1	25				0
12	32	2	26	6	A	26	1	27				0
13	55	2	28	47	B				28	3	31	0
14	56	2	30	93	A	30	5	35				0
15	45	2	32	25	B				32	2	34	0
16	74	3	35	43	A	35	3	38				0
17	78	3	38	68	B				38	4	42	0
18	43	2	40	14	A	40	2	42				0
19	20	1	41	68	B				42	4	46	1

Figure 8.4: Simulation (b)

Customer	Random digits (arrival)	Time betw. arrivals	Clock time of arrival	Random digits (service)	Server	A start	A service time	A end	B start	B service timer	B end	Time in queue
21	50	2	45	48	B				46	3	49	1
22	17	1	46	27	A	46	2	48				0
23	26	2	48	10	A	48	1	49				0
24	15	1	49	94	A	49	5	54				0
25	9	1	50	17	B				50	2	52	0
26	11	1	51	59	B				54	3	57	3
27	76	3	54	73	A	54	4	58				0
28	56	2	56	39	B				57	3	60	1
29	81	3	59	29	A	59	2	61				0

Figure 8.5: Simulation (c)

- Both server work about the same amount of time (A 40 time units, B 40 time units).
- Server A is 63% of the time busy, server B also 63%.
- Only 5 customers had to wait, and the total waiting time is very low with 7 time units. The average waiting time for those who had to wait was just

$$\frac{7}{5} = 1.4$$

  time units.
- The system looks well-balanced. One server would not be able to do the job, and, unless the costs for waiting are very high, a third server is not justified.

8.5 Conclusion

The design of a simulation is usually a large project and thus the rules of project management apply. Project management suggests a number of simple models that describe the process of running a project.

The simplest model is the waterfall model, which suggests a linear execution of various stages of the project, which could here be

- problem analysis;
- model conceptualisation;
- validation;
- input modeling;
- model building;
- verification;
- simulation runs;
- output analysis;
- documentation.

However, due to its linear nature the waterfall model is usually too simple to be employed for real projects. In fact, the waterfall model suggests that each stage has to be completed before the next stage can start, and the model does not allow to go back an correct any errors that were made.

More complex models have been suggested, most notably the so-called spiral model. The common characteristics of these models are that they allow for stages to be revisited and that they allow stages to be started before the preceding stage is completed. A typical example why this is needed is calibration. Calibration happens after the first simulation runs when the output of those runs is compared with real life data. Observed discrepancies are then used to do a fine tuning of the input parameters and the validation process is entered again.

Thus the list of stages above gives only the direction where a project is currently heading to, but it is allowed to return to earlier stages to modify the design, to correct errors, or to calibrate parameters.

8.6 Summary

- Simulation enables the study of, and experimentation with, the interactions of complex systems. Often the knowledge gained cannot be obtained by studying the real-life system.
- The design of simulations can be a highly complex task and, in fact, may be seen as art.
- Important stages in the design of a simulation are Overall Design (including the problem formulation), Model Building (conceptualisations, input modeling, model translation, verification and validation), Output Analysis, and Documentation.
- The problem formulation is the hardest part in the design of a simulation since the problem may not be correctly identified and the system may not be well understood.
- Finding the right model is the real part of the art.
- Input modeling is concerned with collecting data to determine input distributions and parameters for those distributions. Sampling, hypothesis testing, and estimation is often used in input modeling.
- Verification is about relating the conceptual model to the operational model (are we building the model right?), validation is about the relationship between the real system and the conceptual model (are we building the right model?).
- Output analysis is the statistical evaluation of the data generated by the model so that the model can be used in decision making.
- Queues are often the simplest components of a simulation. A queueing model consists in the simplest case of a population, the queue, and a server.
- To define a queueing model one needs to specify queue behaviour, which are the actions by the customers waiting in the queue. Customers could balk, renege, or jockey.
- Queue discipline refers to the logical order in which customers are served. Common models are FIFO (first in first out), LIFO (last in first out), SIRO (service in random order), SPT (shortest processing time first), or PR (service according to priority).
- More complex queueing models may have several queues, several servers, and complex queue behaviour and discipline.

Glossary

calibration

Calibration is the term used for the fine tuning of parameters of a system (like input distributions, parameters of the distributions, etc.).

endogeneous parameter

Endogeneous parameters of a model are those that are predicted by the model, i.e., the status or output variables.

exogeneous parameter

Exogeneous parameters represent external parameters and may be seen as input parameters. An example are parameters in the input probability distribution.

model

A model is any simplified representation of an object or a system.

queue behaviour

Queue behaviour refers to the actions customers take while arriving or waiting in the queue.

queue discipline

Discipline refers to the logical order in which customers are served.

queueing theory

Queueing theory refers to the mathematical models used to simulate these queues.

queueing theory equations

Average number in system, $\overline{S} = \frac{\lambda}{\mu - \lambda}$

Average number in queue, $\overline{q} = \overline{S} - \frac{\lambda}{\mu} = \frac{\lambda^2}{\mu(\mu-\lambda)}$

Average queueing, $\overline{W} = \overline{S} \times \frac{1}{\mu} = \frac{\lambda}{\mu(\mu-\lambda)}$

Average time in system, $\overline{t} = \overline{W} + \frac{1}{\mu} = \frac{1}{\mu-\lambda}$

In addition:

Probability of no customer in system, $P(0) = 1 - \frac{\lambda}{\mu}$

(it is required that $\lambda/\mu < 1$)

Average number in the queue when it is not empty, $\overline{q} = \frac{\mu}{\mu-\lambda}$

simulation

According to the online version of the Encyclopedia Britannica, 'a [computer] simulation uses a mathematical description, or model, of a real system in the form of a computer program. This model is composed of equations that duplicate the functional relationships within the real system.'

Turing test

Real reports are mixed with fake reports based on model predictions. If experienced people cannot tell apart the reports then this suggests that the model in question is an accurate picture of the real world.

validation

Validation refers to the agreement between a model and the real world.

verification

Verification refers to whether a model behaves as intended.