QUEUEING STRUCTURE MODELLING

Berth Eklundh

Dept. of Telecommunication Systems
Lund Institute of Technology
Lund, Sweden

ABSTRACT

A modelling scheme for simulation of queueing structures is proposed. After a general discussion about the special properties of queueing structures and how such usually are investigated, the scheme follows. The scheme makes use of three different concepts: Deterministic Analysis, Load Analysis and Balance Analysis. These concepts are explained and thereafter used in the modelling scheme. It is emphasised that complex models are difficult to handle and validate and that a simple model which comprises the important features of the "real" system is to prefer. The paper discusses which these features can be and how to find them.

1. Introduction

Complex technical systems are nowadays mostly designed and developed jointly by a large number of people. The work has to be organised in some way and one of the most commonly used approaches to this problem is to divide the system into modules. Each module is assigned a specific property or function and the design is thereby simplified.

This approach makes room for two observations: firstly, that there are few, if any, who have an overall view of the entire system, and, secondly, that the modularity could be used to enhance the possibilities to analyse the system with respect to its capacity.

The capacity of a system is usually expressed as waitingtimes, queue lengths, utilization and so forth. The capacity of a system is of interest since certain demands may be put on it and the system designer would like to know if the system can meet the demands. The measures can be estimated by analytical methods or by simulation. The analytical methods available are queueing theory, queueing network theory and similar methods which make use of a mathematical formalism to express the basic properties of the system and to derive interesting performance measures. The analytic methods do often need extensive simplifications to be applicable, and are consequently not always sufficiently accurate. They do on the other hand produce results relatively fast and with acceptable computational effort in most cases, and are therefore attractive when used in for instance dimensioning.

When analytic methods become difficult to use, most people resort to simulation, usually Discrete Event Simulation. Contrary to analytic methods, there are no formal limits to what can be described in a simulation. In reality, limits are always set, however, usually by the computer on which the program is supposed to be run or by the modeller's ability to handle a large program. As the technical systems we surround ourselves with get increasingly complex, attempts to analyse them will more often reach a stage at which the model gets difficult to handle. If such a limit is reached, it seems to be a reasonable requirement that the model should comprise the components of the "real" system that can be assumed to have impact on the performance measures we are interested in. Important questions are then which these components are and how to find them.

All modelling call for simplification. The simplifications that take place when analytic models are used are usually guided by the mathematical feasibility of a

Proceedings of the 1982
Winter Simulation Conference
Highland * Chao * Madrigal, Editors

82CH1844-0/82/0000-0245 $00.75

solution to the stated problem. Since the same limitations do not apply to a simulation approach, it becomes tempting to include as much as possible in the model. The lack of an overall view of a modular system makes it hard to judge the relative importance of the individual components, and introduces problems regarding information acquisition and component interaction.

Analytic models are usually prefered since they are less expensive to develop and run, and since they sometimes can provide insights that are hard to get from simulation models. The development of a simulation model is often a tedious process, and at a certain stage of the development it is almost impossible to change the model since so much time and effort already have been put into it. Verification and validation have to be extensive and may still not produce confidence in the model (GASS 77).

A simulation approach could therefore more readily be used if more of the modelling could be paperwork and if backtracking could be made at as early a stage as possible.

The following is an attempt to form some guidelines for modelling of queueing structures which perhaps can make simulation a tool which indeed can reveal system behaviour that is hard to grasp with analytic methods.

2. Some presuppositions

The measures of capacity mentioned in the introduction relate to structures in which some entities flow to receive service at different service centres. Such a structure can be called a queueing structure. The entities can be called jobs or customers, and it is assumed that these jobs appear somewhere outside the queueing structure and are let in through some gate to enter one of the service centres. Jobs are then routed among all or a part of the total number of service centres and do eventually leave the structure. The routing can be complex and dependent on the state of the structure or on other system properies.

A simulation implementation of such a structure is in this paper assumed to be based on a process approach, supported by for example the SIMULA language (BIRT 73, BIRT 81, FRAN 77). Each service centre is modelled as a process, and a job that enters the centre is, if necessary, queued and thereafter processed. The processing may be complex and a more or less elaborated scheduling may have to be modelled.

It is assumed that the simulation is a Discrete Event Simulation and that behaviour which is hard to model in detail is represented by some stochastic variable. The world outside the structure is modelled as a number of generators which in some manner, more or less sophisticated, create jobs and send them into the structure. Either the generation of jobs or the requests for service must have some degree of randomness, so that deterministic "simulations" are excluded.

3. Modelling issues

A simulation model is indeed, as the name indicates, a MODEL, i e it cannot - and ought not - be a perfect copy of the investigated system. The detailed behaviour of the "real" system has to be simplified and extraordinary performance overlooked. There is a risk however, that important behaviour is omitted, while unimportant details are modelled. Such things may happen for a number of reasons:

- The modelling decisions are based on incomplete knowledge of the system. Internal, unknown, behaviour may have great impact on the performance measures. The model is in such a case not sufficiently detailed.

- The degree of simplification for different model components do not agree. Unless the aim of the model is to study the intrinsic properties of some specific component, it seems to be unnecessary to model some components in detail while others are more crudely modelled. The model is in this case too detailed for some components and not sufficiently detailed for others.

- Some components may be insensitive to changes in the state of the simulation. A component may, for example, always delay a job equally long independent of the load, and may still have a complex internal behaviour which is time-consuming to simulate and difficult to model. Some of these components may be omitted, and the model simplified.

This list is far from complete, and a more thorough discussion can be found in (EKLU 82).

In order to minimize the existence of insufficient or overworked models in the above sense, three concepts are introduced:

Deterministic Analysis

Load Analysis

Balance Analysis

These concepts are explained below and employed in the subsequent section to form a modelling scheme.

Deterministic Analysis

Most things do, if we look into them, seem to have a cause. The behaviour of a system depends on the joint effects of its subsystems which in their turn depend on their subsystems and so on. But if we continue this division, we sooner or later find a level at which things appear to happen at random. At this level we cannot find the cause of individual behaviours. Things do either not lend themselves to be investigated or they do require more information than we can store and/or collect.

Random events can more or less adequately be expressed by stochastic variables and processes. A stochastic process does not behave exactly as the sequence of events it tries to describe, but can reproduce certain measures and does therefore look like the original process. A simulation is a type of stochastic process that uses a sometimes rather complex set of distributions from which the distances in time between different events are selected. These distributions have to be estimated and expressed either as continuous mathematical functions or as stepwise constant probability distribution functions (PDF). If time delays do not belong to a limited set of constant times, they have to be approximated, either by a continous function or by a limited set from which the times are selected. This approximation is, at least when the simulation is fairly detailed, possible to get sufficiently good.

But the distributions have to be estimated, and two different cases can thereby be observed:

a) The first case is when "reality" is not known in detail. (Reality can of course never be known in detail, but the level refered to here is where reality does not appear to behave randomly). Some part of the simulation does in this case illustrate actions that cannot or should not be a part of the simulation in the respect that they could or should be modelled. Typical examples can be found in actions whose length depend on human behaviour. How long people talk to eachother in telephone, how long it takes for someone to answer a question, etc. Times like these can in some cases be broken down into smaller pieces, but at the bottom we always find times that are extremly hard to estimate. We do, in cases like these, have to rely either on our own ability to estimate the times and how they vary, or on measurements. Whichever we find feasible, we have to fit some distribution to either our estimates or to the results of measurements. To fit a distribution to our own estimates is by no means simple, and does at any rate contain a great deal of arbitrariness. This has to be remembered when the rest of the system is modelled, since the ambiguity of one part is not compensated by the accuracy of antoher.

b) The second type is when we indeed can examine the behaviour of a system and break down complex series of events until deterministic times remain. This does not imply that we have perfect knowledge of the internal behaviour of some subsystem of the investigated system, only that the time it takes for it to perform its task is constant and known in advance. The level at which this happens may be very low, but it is formally possible to describe the times involved in the actions. Examples of this can be found in computer systems and in communication networks. Provided we know the length of all packets in a packet swithching network, the load (described as number of instructions to be executed in a particular node) and the priority structure, we are able to simulate the transport times of a specific packet. Such a simulation requires the packet to carry large amounts of data and the operating system of the switching computers to be described in detail, which would mean an extremely complex and heavy simulation. Simplifications are therefore made, but it is essential that these are not arbitrary, but performed in consistency with other system components.

Times of the second type could be exactly reproduced in a simulation provided that the necessary information is available when the time is to be selected. This is usually not the case however, and the constant times are therefore concatenated to form a distribution from which the times are selected. To reproduce the exact times would in most cases mean to describe the system in detail and perhaps to picture it completely.

Load Analysis

Jobs are routed among the service centres in a more or less complex manner. It is usually possible however, to estimate the fraction, of the total number of jobs that leaves a specific centre, which goes to some other specific centre. This fraction can be interpreted as a routing probability. The estimate will only be true on average, but it does not significantly differ from a measurement on the "real" system. The estimates of the routing probabilities can be used to estimate the arrival rate at each service centre. The arrival rate is that part of the total number of jobs emitted from the generators that reaches a specific centre. Each job type has to be treated separately since

these may differ what service requests are concerned. The load will be a number in the range (0,1), and is the product of the effective arrival rate and the mean service time.

Balance Analysis

The most relevant parameter in a simulation is time. Time is the concept around which the modelling turns, and it can be described differently accurate. A discrete event simulation centers, as the name indicates, on events, and these are measures of time. The time can be moved forward by many events occuring with small intervals or by few occuring with longer. The accuracy of a model is, generally speaking, related to the number of events used to describe an activity in the "real" system. An activity can either be described as a series of events occuring with smaller intervals (which perhaps are more easy to estimate the length of) or by one large that is drawn from a distribution illustrating the smaller subintervals.

A process is a structure which forever goes on in a loop and which within this loop has a number of welldefined states in which it stays some time before it goes on to the next. The time a process stays in a state is related to the time it takes for the "real" system to perform the tasks illustrated by that particular state. Depending on how states are identified and concatenated, differently many states, and relating delays, are described. A complex and multifacetted process can contain a large number of states while a more simple usually contains less. The individual processes of a process simulation are usually modelled and programmed almost independently. The intervals between consecutive events in different processes can therefore differ greatly, and some processes may loop several times before even a single event occurs in some other process. In a discrete event simulation, time is stepped forward from one point to the next, with an amount equal to the distance in time between to successive events as they occur if the system is viewed from above, where all processes are visible at the same time. Events are therefore related to different processes and their frequence depend on the intervals with which the individual processes create new events.

Different processes can therefore be said to reside on different levels of the system. At the lowest level fast processes, i e processes which have short intervals between successive events, rotate. From this level and upwards, processes have longer and longer intervals between events, and there is some process of the system that is the slowest, and which consequently will give rise to the least number of events in a simulation of given length. The time it takes to run a simulation is mainly dependent on the number of events that occurs during the run. Many processes at low levels will require a large number of events and the simulation will be very heavy to run.

If now the interesting properties of a simulation of the above kind are related to events occuring at a high level, measurements will be difficult to make. The studied quantities are random variables. We do therefore need a "large" number of samples to be able to make a statistical analysis of the result, and do consequently have to make a large number of measurements. Every event, which is of the type that makes it possible for us to performe a measurement, will involve a large number of events at lower levels and will therefore occur "seldom" in the simulation. If the level of detail of the processes at the lower levels could be reduced, less time would have to be spent before the required number of events at the higher level had occured and the simulation would be less expensive to run. The question is when this reduction can be performed without lessening the accuracy of the simulation.

4. A modelling scheme

A number of decisions can be identified in the development of a Discrete Event Simulation model based on a process approach:

- What is the aim of the model. Which performance measures are of interest?

- How much of the system context should be incorporated in the model, i e where should the border between the simulation and the outside world be set?

- Which processes should the model consist of?

- How detailed is it necessary to make the model in order to be able to derive pertinent information?

The existence of these questions is well known (see for example (ZEIG 76, ÖREN 81)), but there is a lack of guidelines to follow when answering them. The main reason for this is probably that simulation is applied to so many diverse kinds of problems that general rules are hard to find. What follows below is an attempt to form guidelines for the limited application of queueing structures.

Space does not permit argumentation of each step of the scheme, but the interested reader is refered to (EKLU 82) for a

detailed discussion.

Scheme

Step 1:

Identify interesting performance measures. Set an approximate border between the simulation and the outside world. Identify possible processes without mixing separate components and behaviour.

Step 2:

Perform a Deterministic Analysis as described in the previous section, and bring the investigation down to a level at which one of the two endpoints are reached.

This investigation will give the time scale of the individual processes, the average service time and the level at which stochastic variables have to be used.

Step 3:

Illustrate the outside world by identifying a number of generators which create jobs and enter them at some place of the network.

Step 4:

Steps 1, 2 and 3 make it possible to perform a Load Analysis as described above.

Step 5:

Exclude from the simulation model those centres that are not of specific interest and which are lightly loaded.

This point has to be discussed a bit further, since there are a few exceptions and since it may not be quite obvious.

A server which is lightly loaded will in most cases form no queue. The delay caused by the server will consequently consist almost entierly of the service time. The randomness of the service requests and the arrival pattern will undoubtly give rise to queues at times, but the question is if this will have any observable effect on the performance measures that are possible to measure in the simulation program. In most cases it will not. When in doubt or when the variance of service times and/or arrival intervals can be expected to be extremely high, keep the service centre in the model.

Another reason to keep the centre is if it is lightly loaded but has long service times compared with other centres. This can be the case if the centre has internal parallelism making it able to serve many jobs, each for a long time. This time will be almost constant due to the light load and could be added to the service time of the previous centre that the job has visited. If routing is complex, this may be unfeasible however.

Step 6:

Concatenate service times obtained in step 2 until a balanced model is obtained.

The speed of the individual processes will determine the required running time for statistically significant data to be produced. If the simulation is too detailed at the present stage, simplifications should be made at the lowest levels by making detailed models at low levels more crude. This can be done by letting the processes of one or more centres contain fewer scheduling statements.

Step 7:

Simulation models are used mainly because analytic models are too inaccurate or cannot illustrate the behaviour of the "real" system. It is therefore essential that the simulation model indeed comprises behaviour that is hard to model analytically. One way to get a list of such behaviour is to look at approximate analytic models where "difficult" behaviour is treated. A good overview can be found in (CHAN 78). The following is a list of tricky behaviour which should be considered for incorporation if present in the "real" system:

Dependent Service Times

Special order of service

Priorities

Overlapping service requests

Blocking (finite waiting room, always true in real systems)

Parallelism, forking and joining

Flow control and routing

Experience has shown that some of these properties are more important than for example a correct estimation of service time distributions (EKLU 81).

5. Obtainable results and conclusion

The reader may have objections against the rather crude modelling scheme proposed above. Objections do of course exist, since it always is possible to find cases for which a scheme of the above kind does not hold. But the scheme has to be related to what simulations actually can be used for. The most crusial steps of the scheme are

those that omit servers and concatenate events to speed up the simulation and create fewer events. These steps will also make the simulation less complex and therefore easier to handle, verify and validate.

Even lightly loaded servers do at times get long queues and a single measurement would yeild a queue length that substantially differed from the average value. But measuring such a value is a very unlikely event, and unlikely events are extremly hard to estimate in a simulation.

The goal of a simulation study is to tell something about the "real" world. An unlikely event in a simulation does not say much, if anything, about how likely such an unlikely event is in the "real" world, and are therefore not worth measuring. The only things a simulation can give information about are the quantities that have some degree of statistical significance. There are better means than simulation to investigate if an event ever can occur in "reality". That information is for example available already when the model is built and there is consequently no reason to write a program and make simulations to find it out.

Too large and detailed models are difficult to handle and it is likely that they contain unnecessary information as well as programming errors. The above scheme could be used when a simulation is needed with short notice and extensive runs have to be made to investigate a large number of situations, in which case a large and heavy model would be difficult to use.

6. Acknowledgement

This work has been supported by Ericsson (formly L M Ericsson), Sweden.

7. References

BIRT 73 Birtwhistle, G. M., Dahl, O-J., Myhrhaug, B. and Nygaard, K. "SIMULA BEGIN", Auerbach 1973.

BIRT 81 Birtwhistle, G. M. "The Design Decisions Behind Demos", Proceedings of the 1981 UKSC Conference on Computer Simulation, Westbury House 1981.

CHAN 78 Chandy, K. M. and Sauer, C. H. "Approximate Methods for Analyzing Queueing Network Models of Computer Systems", ACM Computing Surveys, September 1978.

EKLU 81 Eklundh, B. "Simulation of Large and/or Complex Systems; is it Worth-while?", Proceedings of the 1981 UKSC Conference on Computer Simulation, Westbury House 1981.

EKLU 82 Eklundh, B. "Simulation of Queueing Structures - A Feasibility Study", Ph. D. Thesis, Department of Telecommunication Systems, Lund Institute of Technology, Lund 1982.

FRAN 77 Franta, W. R. "The Process View of Simulation", North-Holland, New York 1977.

GASS 77 Gass, S. I. "Evaluation of Complex Models", Computing & Operations Research, Vol 4 1977, Pergamon Press.

ZEIG 76 Zeigler, B. P. "Theory of Modelling and Simulation", Wiley-Interscience, New York 1976.

ÖREN 81 Ören, T. I. "Concepts and Criteria to Assess Acceptability of Simulation Studies: A Framework of Reference", Communications of the ACM, April 1981.