NGI101x - 4.4A - Discrete Modeling I

Hi, my name is Alexander Verbraeck, full professor of systems and simulation at Delft University of Technology.

Today I'm going to cover the topic of discrete event simulation with you. Topics I'm going to focus on are:

What is a discrete event simulation?

Where does it fit historically?

How does it differ from other types of simulation, such as agent based simulation and continuous simulation?

What are the steps in a simulation study and what are the important aspects of simulation for infrastructure studies?

According to Shannon, 1975, simulation is a process of designing a model of a concrete system and conducting experiments with this model in order to understand the behavior of the system and/or to evaluate various strategies for the operation of the system.

In this definition, we see a couple of very important words. One is: it is a process to conduct experiments. Experimentation is *key* to simulation and especially to discrete event simulation.

Secondly, we need to use the simulation to understand the behavior of a system or to evaluate strategies. That means also we can use it for design. In that sense, simulation is a 'what if study'. We change parameters to the simulation model and look at the effect on the output of the simulation.

Why do we use discrete event simulation? One is that it is an instrument to evaluate alternative systems designs, to compare alternative solutions, to predict system performance.

Often, it is used for infrastructure problems and logistical problems, especially when we have limited capacity resources. For instance, in energy networks, transportation systems and logistics systems. Sometimes, we also use it for more advanced simulation, such as sensitivity analysis or optimization.

If we look historically, discrete event simulation started already pretty early as one of the modelling techniques. Round the 1950-ties, the first discrete event simulation models were built. In that particular time, especially for military simulations, transportation simulations and energy simulations.

Later, continuous simulations and agent based simulations joint the set of simulation studies that can be executed for infrastructures.

There are a couple of similarities and differences between discrete event simulation and continuous simulation.

One is that we very much look at the state of the system over time but in this particular case we do it at discrete moments. It's not that the system's state changes continuously over time, but more that the system state changes at discrete events, concurring at an instant.

In continuous models states, and the state variables are a continuous function of time. This means, that and the state variables have a value for each point in time and the variables change at every instant of time. This is in sharp contrast with discrete-event models where the state is a piecewise constant function over time. So not continuous, but constant. Only at discrete instances, the function changes its value. And we can see that, for instance, at the instance of time 17, time 27, time 70 and around time 95.

The values change exactly at that point in time and don't change in between. It also makes the discrete event simulation models pretty fast because the state only has to be changed and these point in time. This makes it very useful for cueing systems, for resource systems, for transportation and logistics but also for control systems. For all these systems we only have to focus on events: starts and ends of processes rather than the evolution of processes themselves. So how do we build these types of simulations?

The simulations are very much built based on models. Models for the current situation *and* models for potential future situations.

The models have to be identified based on states and information we can get from the infrastructure system itself. We build a model, diagnose the problem, test whether the model is corresponding to the real system (validation), build alternative models to evaluate them for solutions. And hopefully we find solutions we can implement in a new situation in the actual system.

The steps we take for this are called conceptualization, specification, verification and validation, experimentation and analysis. These steps are often carried out in a more iterative manner. That means that we go back to earlier steps that we have not yet carried out, or to earlier steps that we have already carried out.

Often, we use a traditional water full model for this. This is also will be seen many textbooks. First conceptualization, then specifications, then data collection and verification, etc.

More and more, iterative modelling becomes normal in discrete event simulation. This means that we build a model, we go to specification, we go to data collection, we find in the data collection that maybe we need to make some changes and then we go back to one of the earlier phases to adapt the model.

However, an incremental model is lately seen as the best way to build discrete-event simulation models. It means that we start small. We start with a tiny model that models the core of the problem or of the infrastructure we are looking at. Step by step, we extend to model in a number of steps where we go through all the phases of the model cycle again. It means that we analyze, diagnose, conceptualize specified data collection etc. a couple of times after another. Starting small and extending it step by step to a bigger model

Especially for larger infrastructure models this is an excellent way to do it because we don't have to make the model all at once leading to all kinds of bug finding and mistakes.

So what are these different phases? Let's start with the conceptualization.

Conceptualization, for discrete event simulation modeling, is especially aimed at demarcation of the system. It means that we find the system boundaries and we determine what goes into the system, when it goes into the system and what belongs to the system and what belongs to the environment? We also establish the components, the objects the entities that are in the model and the processes that we see in the model. Sometimes, we also look at a time based aspects. The conceptualization is often carried out in some kind of diagramming technique that makes it clear what the models look like. This is, for example, a model of parts of the infrastructure of a large airport and with this we can look how we built up the model from smaller pieces. We can also look at the flow of the model, for instance in a process model, such as a flow diagram. In this case aimed at queuing theory and inventory theory.

The specification, the next step in the model phase, is aimed at leading to a working model that one can experiment with. The first step is reduction of the model, also seen as one of the most difficult steps. Reduction means that we leave out everything that's not needed to complete the model and to run a successful simulation study. We specify the model, we gather data and we build the model.

An example how to build a model is shown here in one of the simulation modelling packages called SIMIO that's available and often used in university surroundings.

We built a model based on a number of standard building blocks that are available on the left hand side of the screen in a so-called library. With these building blocks, we very quickly built up the model connected by, in this case path, and we have a generation of entities, a server that serve the entities and a sink in which the entities again disappear. The model has both two dimensional and three dimensional representations and one can see that in one minute one builds a queuing model that would normally take, if you would have to program it, a much, much longer time.

One of the interesting things is also that the model of course generates a lot of output. To give an example, you can see from this very small model that quite a lot of output is generated for later analysis and it's this analysis that so important also for discrete event simulation because the models are aimed at generating this particular output.

What kind of data do we need to fill these kind of models? Especially generators and sinks at the sources and ends of the model are very important.

Means that we have to know when entities enter the model and when an entities leave the model.

We also have to know how long processes take and when resources are available. We can gather it in many, many different ways. For instance, expert opinions, measurements, or historical sources like databases.

One of the steps that's extremely key in discrete event simulation modelling is verification and validation. We need to know that our model is a valid representation of reality and there are different ways to test that. *Verification* looks at the translation of our conceptual model into a simulation model. *Validation* looks at the relationship between our simulation model and reality.

There are different ways to validate models. One is to test hypotheses using statistics, where we compare output from our simulation model and output from reality under similar circumstances. But we can also look at expert opinions about our simulation, for instance by having experts look at our animations. There are many relationships and many types of validation that we can carry out. And in principle every step that we take in the simulation modelling study has to be followed by a validation or a verification step where we test the correctness and the accuracy of the model that we just created and to look whether the model is still really fit for purpose for what we try to accomplish with the model. When the model is considered to be valid and verified, we want to look at experiments of course. Because, if we look back at the definition of Shannon, the model is to be experimented with. We want to look at the behavior of the model as a representation of our system under all kinds of different circumstances.

But, the question of course is: how do we exactly do that? Do we really need to test the system only with one particular run or would it be statistically more valid if we use multiple runs? Would it be better to test model for longer time or a shorter time?

This is exactly what experiment specification is about. We determine the run controlled conditions under which the model is experimented with and thereby we can establish a statistical validity of our model. When we have done a number of experiments, of course we want to compare alternatives. We want to do statistical analyses and we might look at bottlenecks or do a sensitivity analysis on the stability of the results for some of the input variables.

This is what we do in analysis and diagnosis, where we use the model to calculate results that are statistically valid and from which we can draw conclusions about our infrastructure models. An example is to export model results to, for instance SSPSS, or one of the statistical packages and analyze it in detail.

With this we can test hypotheses about the differences or the similarities between different model runs. If we look at infrastructure simulation, discrete event simulation is very much suited because we can represent model components one to one and, in the form of entities or building blocks in our simulation model. We use hierarchy very often to build a model more bottom up. That means that all kinds of components, as we have seen in the example from SIMIO, can be dragged into our model - using drag and drop interface- from which we can build more complex model or even build more complex components that can then again be used in model again.

We use libraries of components, sometimes specifically aimed at a particular type of infrastructure. Infrastructure simulations are very often aimed at capacity usage. How much capacities there are available, for instance on the transportation network and energy network, and how much do we use? The types of simulations that have been built for infrastructure simulation and for discrete event simulation show lots of similarities and therefore discrete-event simulation is used very often for infrastructure testing. The animation can help in building, debugging and presenting the model.

Conclusions. Discrete-event simulation very much focuses on *events.* Events are times when the stage changes. The state is continuous between those different events. And this leads to very fast execution of the models. That means that we can build quite complex models that still have very acceptable runtimes. And that means that we can model our infrastructure in quite some detail. We have also looked at the model cycle where we focused on incremental building and building blocks. But we also will see later, in some of the examples for discrete event simulation in one of the next presentations, is that we focus very much on hierarchy, where we can build up models in a hierarchical fashion and that means that the modeler always keeps overview of the model itself.

Modeling is quite data intensive. This means that we have to very carefully look at the boundaries of our system and conceptualization and the reduction our system in the specification to make sure that we have the right input for the model and we can collect the data.

Thank you very much for your attention.