Discrete Event simulation

Each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation can directly jump in time from one event to the next.

A more recent method is the **three-phased approach** to discrete event simulation (Pidd, 1998).
In this approach, the **first phase** is to jump to the next chronological event.
The **second phase** is to execute all events that unconditionally occur at that time (these are called B-events).
The **third phase** is to execute all events that conditionally occur at that time (these are called C-events).
The three phase approach is a refinement of the event-based approach in which simultaneous events are ordered so as to make the most efficient use of computer resources.

Components of Discrete-Event simulations

- State
- Clock - time ‘hops’ because events are instantaneous – the clock skips to the next event start time as the simulation proceeds.
- Event List - it lists events that are pending as a result of previously simulated event but have yet to be simulated themselves.
- RNG - The use of pseudo-random numbers as opposed to true random numbers is a benefit should a simulation need a rerun with exactly the same behaviour.
- Statistics - The simulation typically keeps track of the system's statistics, which quantify the aspects of interest. In a simulation model, performance metrics are not analytically derived from probability distributions, but rather as averages over replications, that is different runs of the model. Confidence intervals are usually constructed to help assess the quality of the output.
- Ending condition - theoretically a discrete-event simulation could run forever. So the simulation designer must decide when the simulation will end. Typical choices are “at time t” or “after processing n number of events” or, more generally, “when statistical measure X reaches the value x”.

Activity-Orientated Paradigm

Jobs arriving in random order
“We might break time into increments of size 0.001. At each time point, our code would look around at all the activities, e.g., currently-active job servicing, and check for the possible occurrence of events, e.g. completion of service. Our goal is to find the long-run average job wait time.”
Clearly, an activity-oriented simulation program is going to be very slow to execute. Most time increments will produce no state change to the system at all. Thus the activity checks will be wasted processor time. This is a big issue, because in general simulation code often needs a very long time to run.

Event-Orientated Paradigm

Instead of having time “creep along” so slowly, why not take a “shortcut” to the next event?
We could advance simulated time directly to the time of the next event:

```
if ServerBusy and NextArrivalTime < ServiceFinishedtime or
  not ServerBusy then
      SimTime = NextArrivalTime
else
  SimTime = ServiceFinishedtime
```

We store an event set, which is the set of all pending events.

Process-orientated Paradigm
Each simulation activity is modelled by a process.

If we were to simulate a queuing system as above, but using the process-oriented paradigm, we would have two threads, one simulating the arrivals and the other simulating the operation of the server. Those would be the application-specific threads (so NumActiveAppThreads = 2 in the code below), and we would also have a general thread to manage the event set.

There are two types of discrete simulation: discrete time models and discrete event models. Discrete time models (time-sliced) are those that split the simulation into fixed time intervals. At each interval, the state of the model is updated using functions that describe the interactions. Discrete event models (event-oriented) are those which maintain a queue of events scheduled to happen in order of time, each event representing the change of state of an element in the model.

Microsimulation
Ground transportation models can include all modes of roadway travel, including vehicles, trucks, buses, bicycles and pedestrians. In traditional road traffic models, aggregate representation of traffic is typically used where all vehicles of a particular group obey the same rules of behaviour; in micro-simulation, driver behaviour and network performance are included so that complete traffic problems (e.g. Intelligent transportation system, shockwaves) can be examined

Microsimulation models track individual vehicle movements on a second or sub-second basis. Microsimulation relies on random numbers to generate vehicles, select routing decisions, and determine behaviour.

Road traffic micro-simulation models are computer models where the movements of individual vehicles travelling around road networks are determined by using simple car following, lane changing and gap acceptance rules. They are becoming increasingly popular for the evaluation and development of road traffic management and control systems.

Microsimulation models usually produce two types of results: animated displays, and numerical output in text files. It is important to understand how the software has accumulated and summarised the numerical results to prevent incorrect interpretation. Animation can allow the analyst to quickly assess the performance, however it is limited to qualitative comparisons. The main indication of a problem that can be seen in an animation is the forming of persistent queues.

'Measures of Effectiveness' (MOEs)
The following MOEs are most common when analysing simulation models:

- 'VMT' (Vehicle Miles Traveled) is computed as a combination of the number of vehicles in the system and their distance they traveled.
- 'VHT' (Vehicle Hours of Travel) is computed as the product of the link volume and the link travel time, summed over all links.
- 'Mean system speed' is equal to VMT/VHT.
- 'Total system delay' is one of the most effective ways to evaluate different congestion relieving alternatives and it is usually the MOE that the travelling public notices. Delay can be calculated several ways. Some consider it to be only that delay which is above free flow conditions. Others include the baseline delay which occurs as a result of traffic control devices. Some even include acceleration and deceleration delay, while others include only stopped delay.

Other commonly reported metrics from traffic simulation tools include:

- Link road section speeds, flow, density, travel time, delay, stop time
- Intersection turning volumes, delay,
- Journey times
- loop detector records for speed, occupancy, headway, gap
- vehicle trajectories and speed vs. distance plots

Microscopic simulators model individual entities separately at a high level of detail, and are classed as discrete simulations. Each vehicle is tracked as it interacts with other vehicles and the environment. Interactions are usually governed by car-following and lane-changing logic. Rules and regulations are defined to control what can and cannot be done in the simulation, for example speed limits, rights of way, vehicle speed and acceleration. Traffic flow details usually associated with macroscopic simulation are the emergent properties of the microscopic simulation. Microscopic simulators can model traffic flow more realistically than macroscopic simulators, due to the extra detail added in modelling vehicles individually [16]. Microscopic simulators are widely used to evaluate new traffic control and management technologies as well as performing analysis of existing traffic operations.

A very simple form of microscopic simulation is cellular simulation, which involves modelling the road as a series of cells and moving the vehicles between cells based on vehicle parameters. This method can implement links using an array with length equal to the number of cells in the link. Cell length has to be determined and must be the same for all cells, which is a disadvantage because it assumes all vehicles occupy the same amount of space. When the simulation is run, each cell can be either empty or occupied by one vehicle. Vehicles are moved forwards by their speed and are restricted by vehicles in front. Links are connected to nodes and rules exist which determine where vehicles go when they reach a node. This method can be very efficient because of the simple array structure, but it lacks some realism. An even simpler approach is queue-based simulation, where vehicles always move at a set speed until they reach a queue at the end of each link.

Microscopic (Microsimulation) models

Models study individual elements of transportation systems, such as individual vehicle dynamics and individual traveler behaviour.

Car following models
Time-continuous models, all car-following models have in common that they are defined by ordinary differential equations describing the complete dynamics of the vehicles' positions x_α and velocities v_α. It is assumed that the input stimuli of the drivers are restricted to their own velocity v_α, the net distance (bumper-to-bumper distance) $s_\alpha = x_{\alpha-1} - x_\alpha - l_{\alpha-1}$ to the leading vehicle $\alpha - 1$ (where $l_{\alpha-1}$ denotes the vehicle length), and the velocity $v_{\alpha-1}$ of the leading vehicle. The equation of motion of each vehicle is characterised by an acceleration function that depends on those input stimuli:

$$\ddot{x}_\alpha(t) = \dot{v}_\alpha(t) = F(v_\alpha(t), s_\alpha(t), v_{\alpha-1}(t))$$

In general, the driving behaviour of a single driver-vehicle unit α might not merely depend on the immediate leader $\alpha - 1$ but on the n_a vehicles in front. The equation of motion in this more generalised form reads:

$$\dot{v}_\alpha(t) = f(x_\alpha(t), v_\alpha(t), x_{\alpha-1}(t), v_{\alpha-1}(t), \ldots, x_{\alpha-n_a}(t), v_{\alpha-n_a}(t))$$

Cellular Automaton
Cellular automaton (CA) models use integer variables to describe the dynamical properties of the system. The road is divided into sections of a certain length Δx and the time is discretised to steps of Δt. Each road section can either be occupied by a vehicle or empty and the dynamics are given by update rules of the form:

$$v_\alpha^{t+1} = f(s_\alpha^t, v_\alpha^t, v_{\alpha-1}^t, \ldots)$$
$$x_\alpha^{t+1} = x_\alpha^t + v_\alpha^{t+1}$$

(the simulation time t is measured in units of Δt and the vehicle positions x_α in units of Δx).

The time scale is typically given by the reaction time of a human driver, $\Delta t = 1\mathrm{s}$. With Δt fixed, the length of the road sections determines the granularity of the model. At a complete standstill, the average road length occupied by one vehicle is approximately 7.5 meters. Setting Δx to this value leads to a model where one vehicle always occupies exactly one section of the road and a velocity of 5 corresponds to $5\Delta x/\Delta t = 135\mathrm{km/h}$, which is then set to be the maximum velocity a driver wants to drive at. However, in such a model, the smallest possible acceleration would be $\Delta x/(\Delta t)^2 = 7.5\mathrm{m/s}^2$ which is unrealistic. Therefore, many modern CA models use a finer spatial discretisation, for example $\Delta x = 1.5\mathrm{m}$, leading to a smallest possible acceleration of $1.5\mathrm{m/s}^2$.
Although cellular automaton models lack the accuracy of the time-continuous car-following models, they still have the ability to reproduce a wide range of traffic phenomena. Due to the simplicity of the models, they are numerically very efficient and can be used to simulate large road networks in realtime or even faster.

Macroscopic

Models deal with aggregated characteristics of transportation elements, such as aggregated traffic flow dynamics and zonal-level travel demand analysis. Macroscopic simulators model the flow of traffic using high-level mathematical models often derived from fluid dynamics, thus they are continuous simulations. They treat every vehicle the same, and use input and output variables such as speed, flow and density. These simulators cannot differentiate between individual vehicles, and usually do not cater for different vehicle types. They lack the ability to model complex roadways, detailed traffic control features or different driver behaviours.

Mesoscopic

Models analyse transportation elements in small groups, within which elements are considered homogeneous.