Mathematical
Modelling and
Simulation

Discrete Event Simulation and Modelling with Event Graphs

General

Modelling Approach/ Representation Form		Model Type
Event Graphs	$\xrightarrow[\text{to}]{\text{leads}}$	Discrete Event Simulation Model

General

Mathematical Modelling and Simulation

Modelling Approach/ Representation Form		Model Type
Event Graphs	leads to →	Discrete Event Simulation Model

Compare:

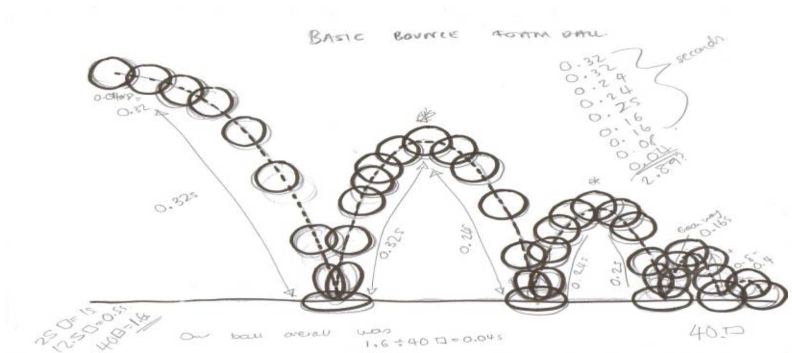
System Dynamics or Lagrange Formalism	leads to →	Differential Equation Model

Motivation

Mathematical Modelling and Simulation

- Simulation of systems that change their states only at so called „events“

- (Simulation of systems that can be approximated as such)



Discrete Event Simulation
Fundamental Concept

Mathematical Modelling and Simulation

Two fundamental components of a discrete event simulation (DES) model

State Variables

„Observables“ or the model. Used to generate the simulation output

Events:

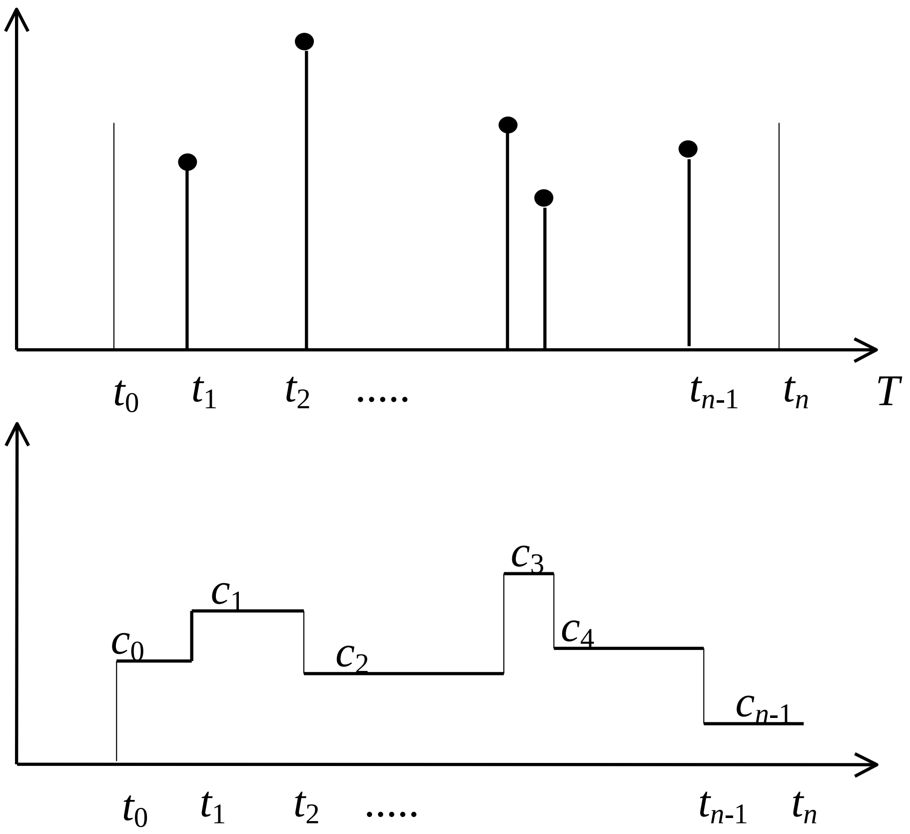
Cause state variables to change and schedule/cancle future events

Discrete Event Simulation
Fundamental Concept

Mathematical Modelling and Simulation

- Events

- States piecewise constant



Discrete Event Simulation
Fundamental Concept

Mathematical Modelling and Simulation

Events are scheduled using

Event Notices.

Every event notice contains two pieces of information:

- What (type of) event is being scheduled, and
- the (simulated) time at which the event is planned to occur

The

Event List

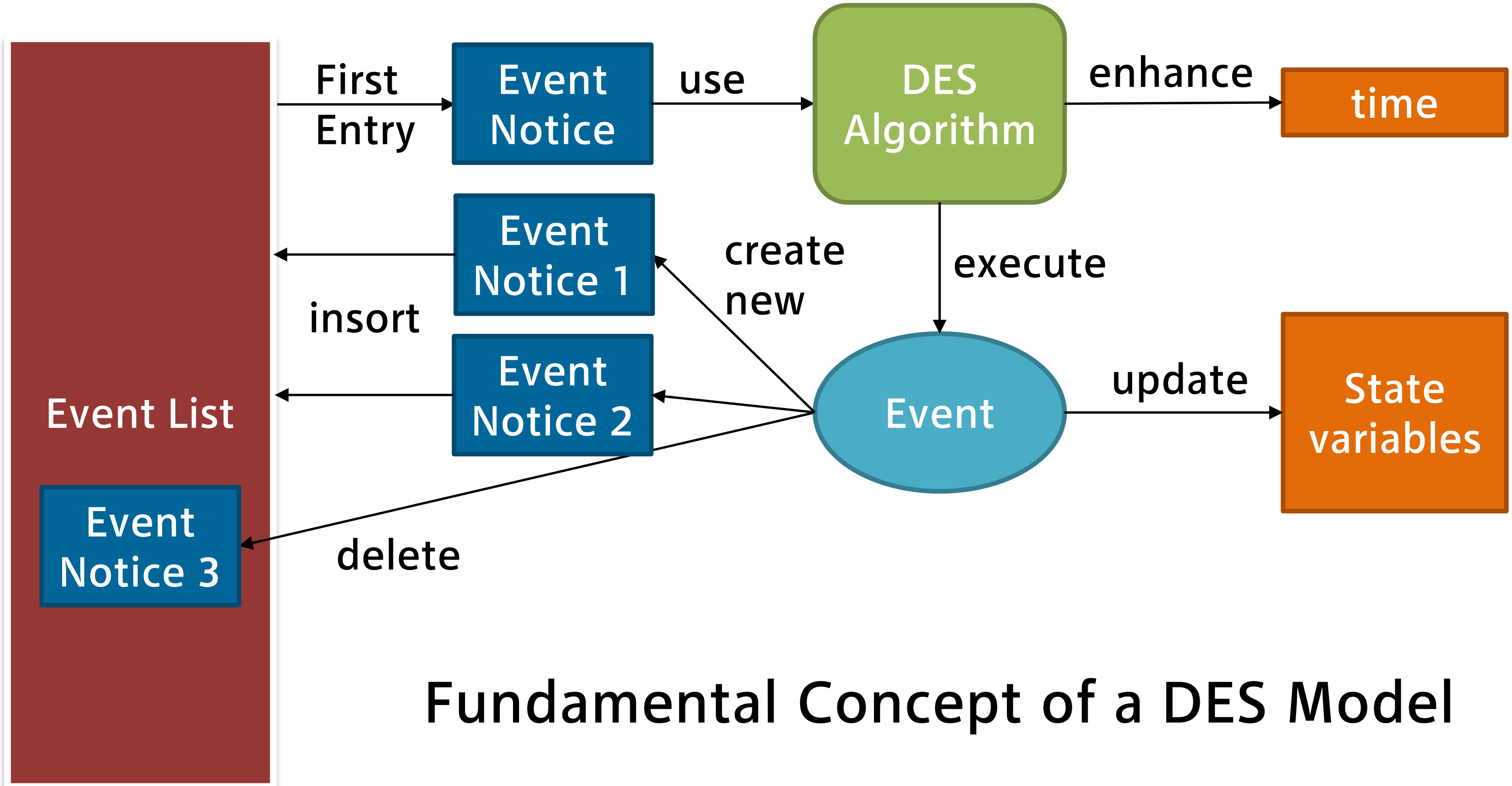
keeps the event notices in order by ranking them based on the lowest scheduled time.

The events list is managed by basic

Discrete Event Algorithm

that controls the flow of time in the simulated world of the model

Discrete Event Simulation
Fundamental Concept



Fundamental Concept of a DES Model

How to formalise DES Models

EVENT GRAPHS

Event Graphs General

- Concept introduced by Lee Schruben in 1983
- Sometimes called „Simulation Graphs“
- Graphical representation of a DES model which can directly be fed to Event Graphs simulators, e.g. SIGMA (Compare with System Dynamics and AnyLogic)
- Very general – for most applications, more specialised concepts / simulators are used

Event Graph Formalism

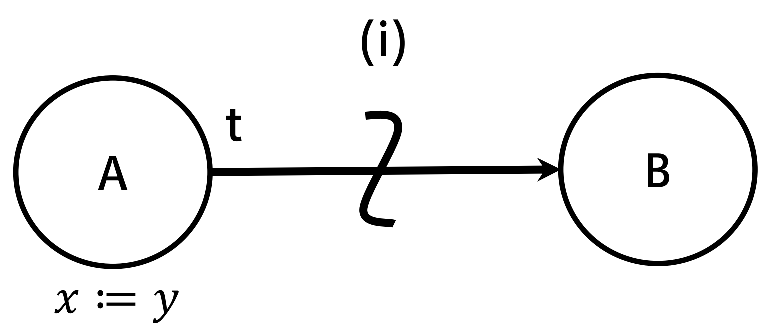
Mathematical Modelling and Simulation

The occurrence of an event with type A

- causes state variable x to change its state to y

causes an event with type B

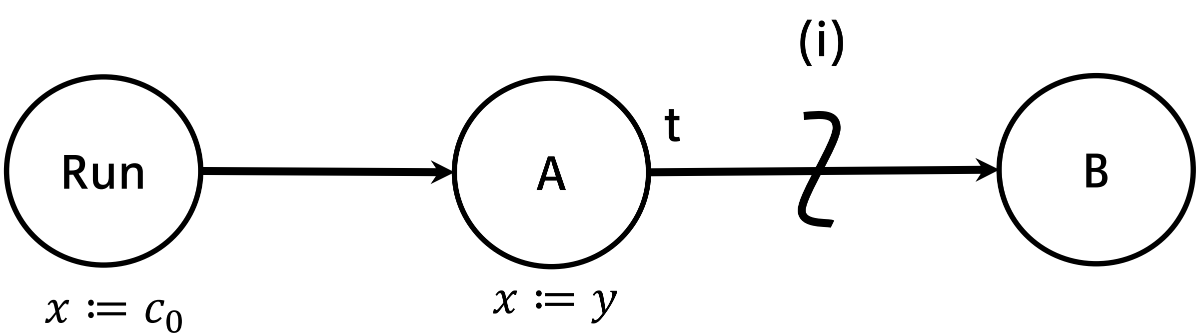
- to be scheduled after a time delay of t,
- providing condition (i) is true, after the state transitions for Event A have been performed



Event Graph Formalism

Mathematical Modelling and Simulation

- As the event-list is empty at the beginning of the simulation, a designated initial event needs to be given.
- Usually this event is labelled with „Run“



Example: Difference Equation

Mathematical Modelling and Simulation

- Goal: model the sequence

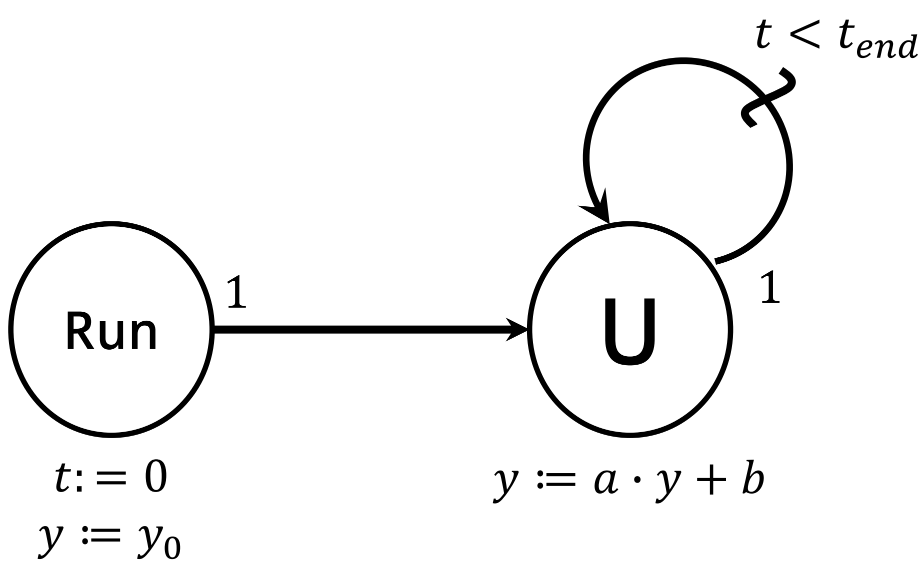
$$y(k+1) = ay(k) + b,$$
$$k = 0, \dots, t_{end}, \qquad y(0) = y_0$$

using the Event Graph formalism

Example: Difference Equation

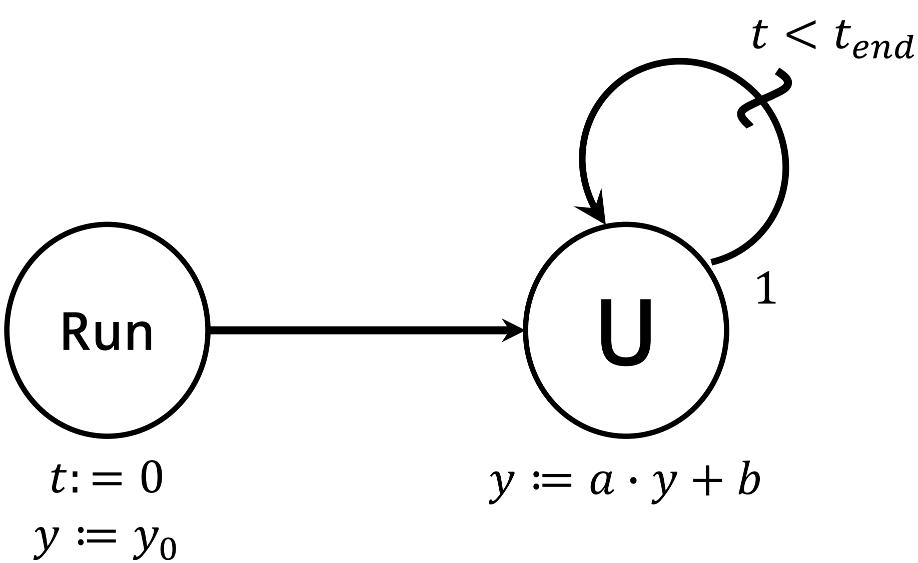
TU WIEN Mathematical Modelling and Simulation

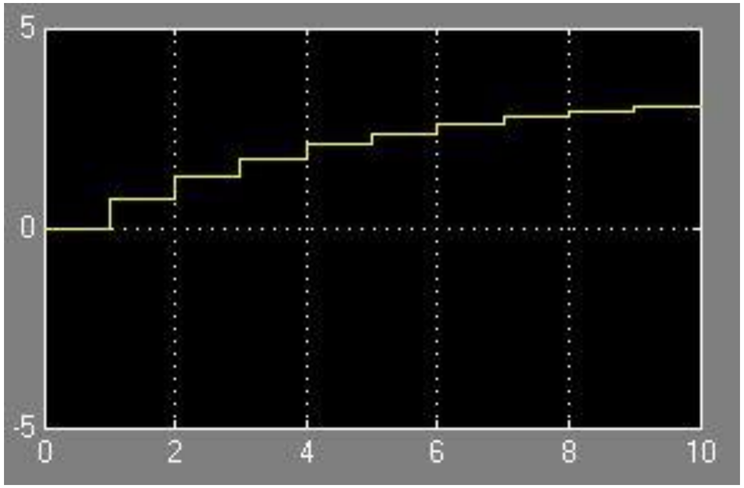
- Goal: model the sequence
$$y(k+1) = ay(k) + b,$$
$$k = 0, \dots, t_{end}, \qquad y(0) = y_0$$
using the Event Graph formalism

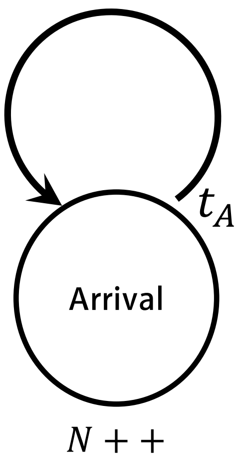


Example: Difference Equation

- Goal: model the sequence
$$y(k+1) = ay(k) + b,$$
$$k = 0, \dots, t_{end}, \qquad y(0) = y_0$$
using the Event Graph formalism



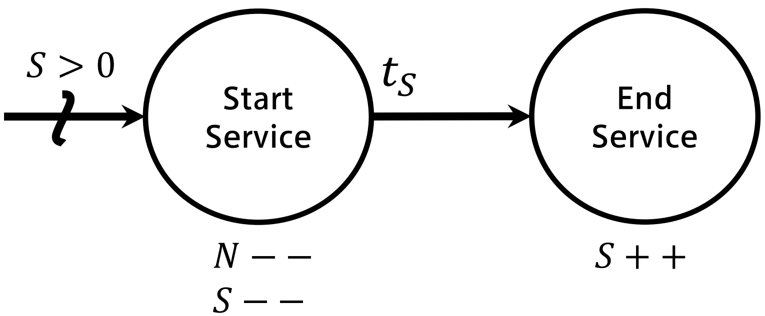




Arrival Process:

- Used to generate „entities" coming from outside the system boundaries
- Usually changes increases a cumulative state variable by one. This variable is usually called a **queue**
- Sequence of interarrival times t_A that can be
  - constant, a
  - deterministic sequence, or a
  - sequence of random variables

Classical Elements



Service Process:

- Used to treat „entities“ coming from, e.g. an arrival process
- If available ($S > 0$), takes an element from the queue
- Sequence of service times t_S that can be
  - constant, a
  - deterministic sequence, or a
  - sequence of random variables

Multiple Server Queue

- Customers arrive to a service facility according to an arrival process and are served by one of k servers.
- Customers arriving to find all servers busy wait in a single queue and are served in order of their arrival.
- Parameters:

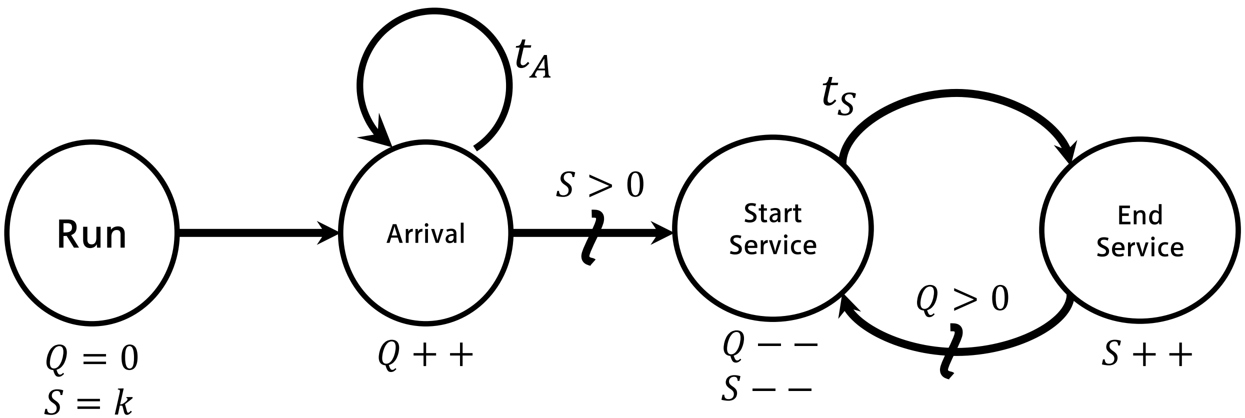
  t_A = interarrival times
  t_s = service times
  k = total number of servers

- State Variables:

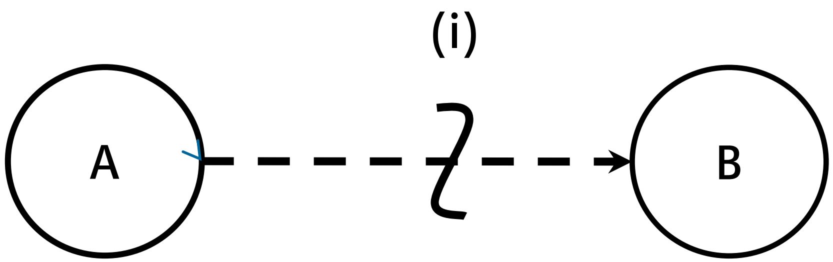
  $Q :=$ # of customers in queue
  $S =$ # of available servers

Multiple Server Queue

- Customers arrive to a service facility according to an arrival process and are served by one of k servers order of their arrival.
- Parameters:
  t_A = interarrival times
  t_s = service times
  k = total number of servers
- State Variables:
  Q := # of customers in queue
  S = # of available servers



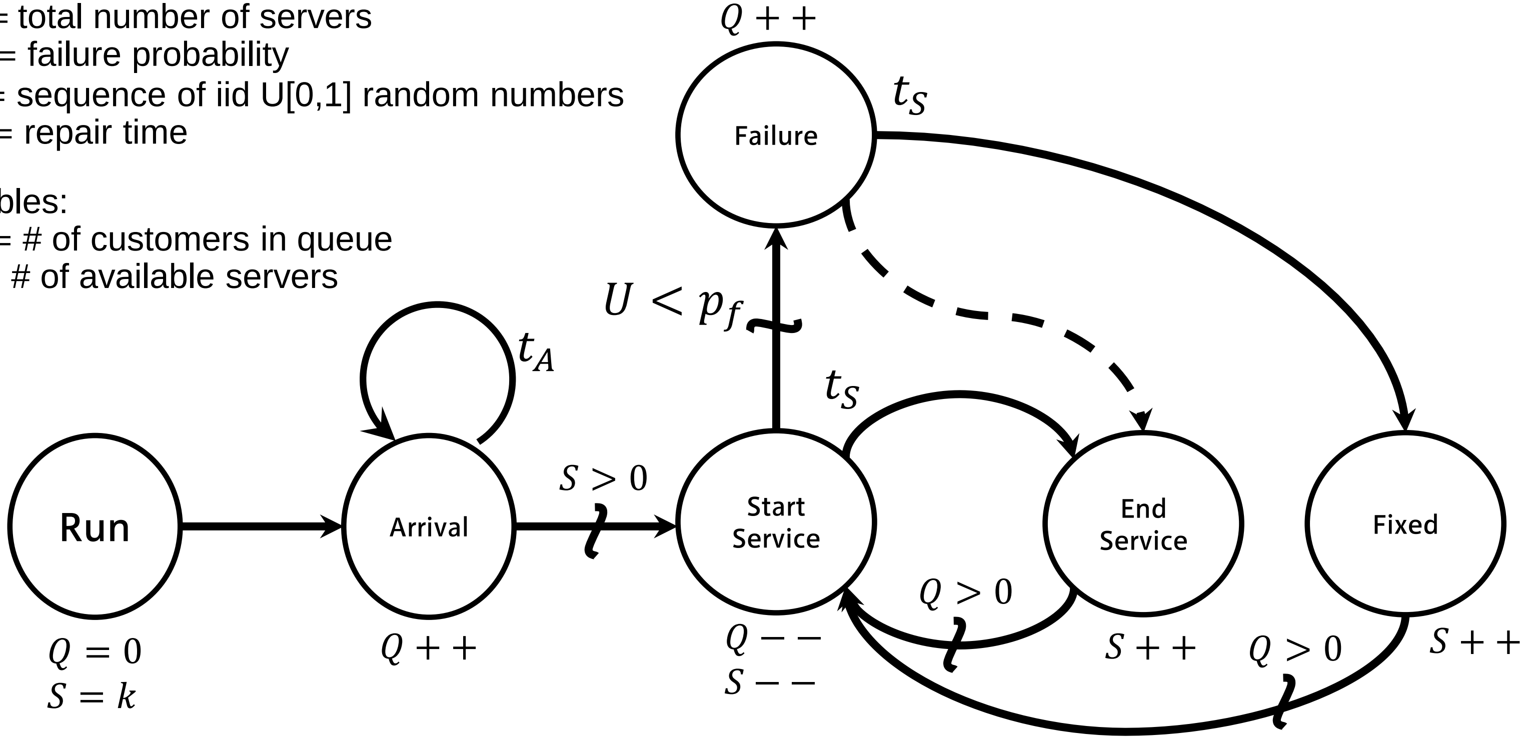
Cancelling Edge



- the inverse operation of the scheduling edge
- whenever event with tyoe A occurs, then if condition (i) is true, the first occurrence of an event with type B is removed from the event list
- if event B is not scheduled to occur, then nothing happens.
- if there are multiple occurrences, only the first is removed.

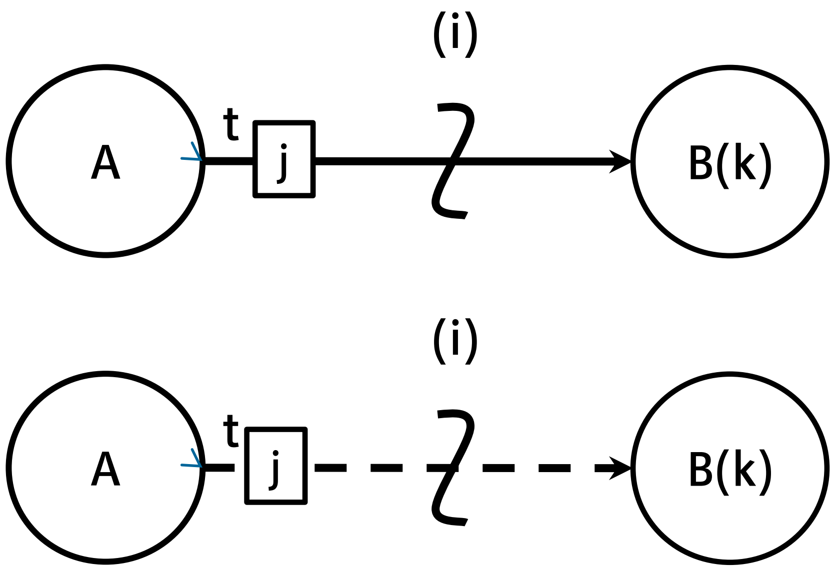
Multiple Server Queue with Failure

- Customers arrive to a service facility according to an arrival process and are served by one of k servers order of their arrival.
- With certain failure probability the server breaks while serving
- Parameters:
  - t_A = interarrival times
  - t_s = service times
  - k = total number of servers
  - p_f = failure probability
  - U = sequence of iid U[0,1] random numbers
  - t_R = repair time
- State Variables:
  - Q := # of customers in queue
  - S = # of available servers



Parameterization of Events

Scheduling edge with parameter: When A occurs then, if (i) is true, B is scheduled after t time units. When B occurs, its parameter k will be set to the value given by the expression j (j is calculated when A occurs).



Tandem Server Queue

- Customers processed by one workstation consisting of a multiple-server queue.
- Upon completion of service at the first workstation, a customer proceeds with probability p to a second workstation or departs the system with probability (1- p)..
- Parameters:

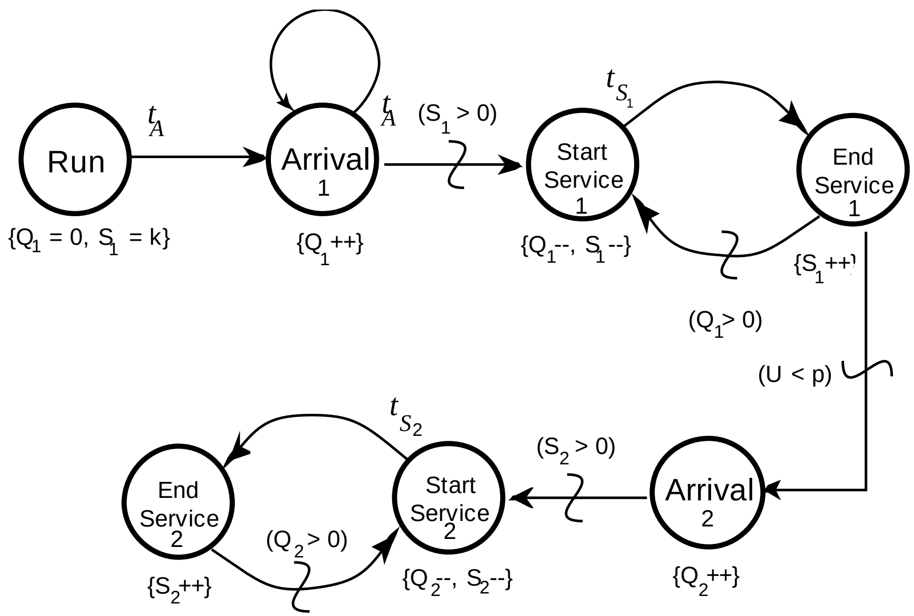
  t_{A_i} = interarrival times at WS i
  t_{s_i} = service times at WS i
  k_i = total number of servers at WS i
  p = probability to proceed from 1 to 2
  U = sequence of iid U(0,1) random numbers
- State Variables:

  $Q_i :=$ # of customers in queue at WS i
  $S_i =$ # of available servers at WS i

Tandem Server Queue

Mathematical Modelling and Simulation

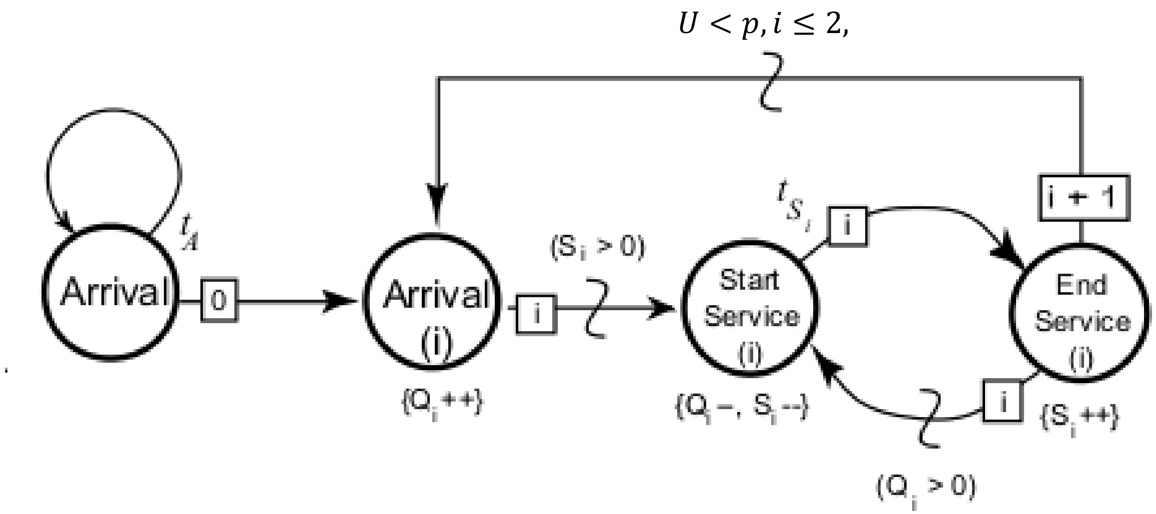
- Customers processed by one workstation consisting of a multiple-server queue.
- Upon completion of service at the first workstation, a customer proceeds with probability p to a second workstation or departs the system with probability (1- p)..



Tandem Server Queue

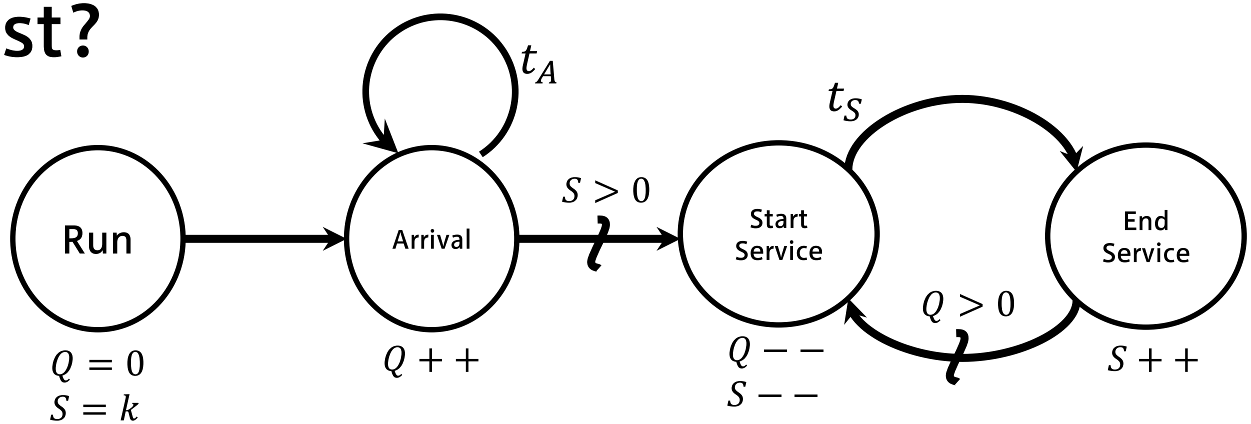
TU WIEN
Mathematical Modelling and Simulation

- Customers processed by one workstation consisting of a multiple-server queue.
- Upon completion of service at the first workstation, a customer proceeds with probability p to a second workstation or departs the system with probability (1- p)..



Case Study:

- What happens, when executing a Multiple Server Queue model with deterministic service and arrival times?
- Event Notices?
- Event List?



Event Notices and Parameters

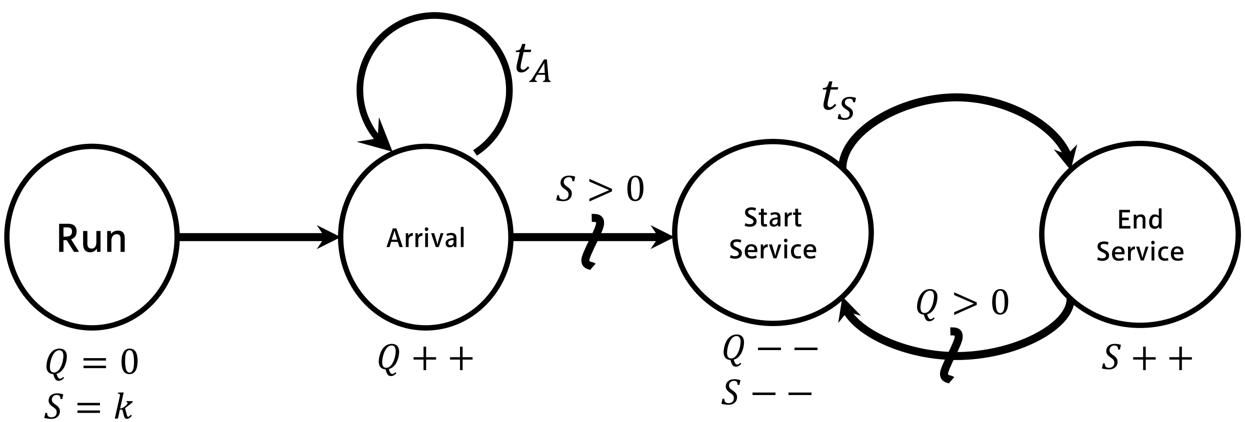
Mathematical Modelling and Simulation

```
DISCRETE start
 server = 2;  queue = 0
 SCHEDULE arrival .AT. t+0.
END ! of start

DISCRETE arrival
 queue = queue + 1;  t_arrival = 1
 SCHEDULE arrival .AT. t+tarr
 IF server .GE. 0 SCHEDULE start_service at t+0.
END ! of arrival
```

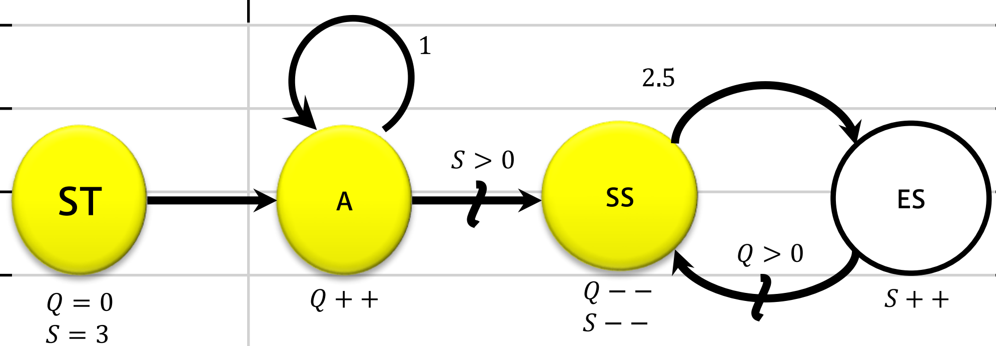
```
DISCRETE start_service
 queue  = queue – 1;  server = server -1
 t_service = 2.5
 SCHEDULE end_service .AT. t+t_service
END ! of start_service

DISCRETE end_service
 server = server + 1
 IF queue .GE. 0 SCHEDULE start_service at t+0.
END ! of end_service
```



Event List Multiple Server Queue

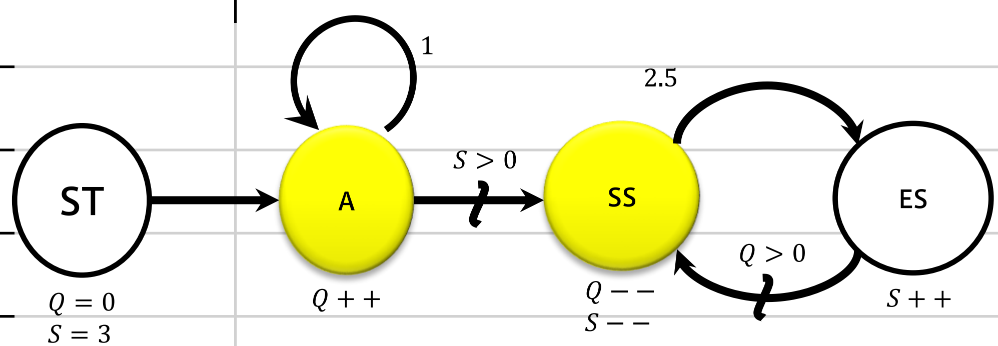
time	event	action	schedule
0	ST	Q=0; S=2;	A at t+0=0
0	A	Q=Q+1=1	A at t+1=1; SS at t+0=0
0	SS	Q=Q-1=0; S=S-1=1	ES at t+2.5=2.5
1	A		
2.5	ES		



Event List Multiple Server Queue

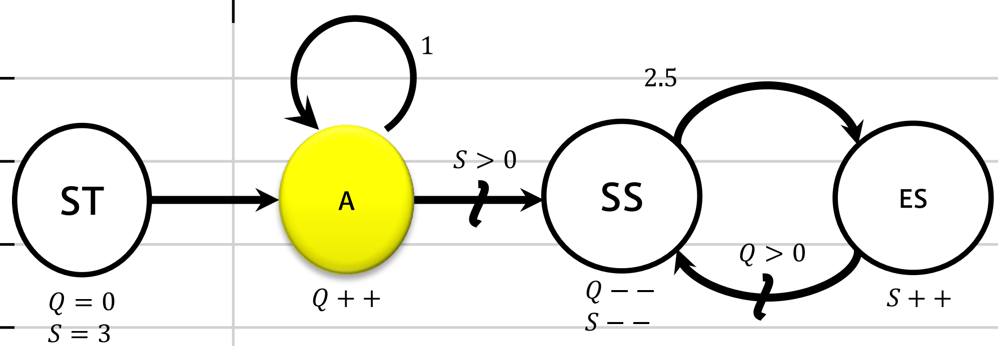
Mathematical Modelling and Simulation

time	event	action	schedule
0	ST	Q=0; S=2;	A at t+0
0	A	Q=Q+1=1	A at t+1=1; SS at t+0=0
0	SS	Q=Q-1=0; S=S-1=1	ES at t+2.5=2.5
1	A	Q=Q+1=1	A at t+1=2; SS at t+0=1
1	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=3.5
2.5	ES		
2	A		
3.5	ES		



Event List Multiple Server Queue

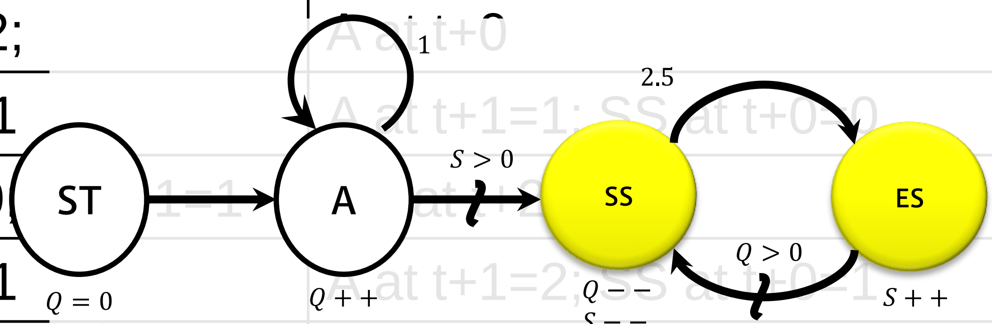
time	event	action	schedule
0	ST	Q=0; S=2;	A at t+0
0	A	Q=Q+1=1	A at t+1=1; SS at t+0=0
0	SS	Q=Q-1=0; S=S-1=1	ES at t+2.5=2.5
1	A	Q=Q+1=1	A at t+1=2; SS at t+0=1
1	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=3.5
2	A	Q=Q+1=1	A at t+1=3; (SS condition not true)
2.5	ES		
3.5	ES		
3	A		



Event List Multiple Server Queue

TU WIEN
Mathematical Modelling and Simulation

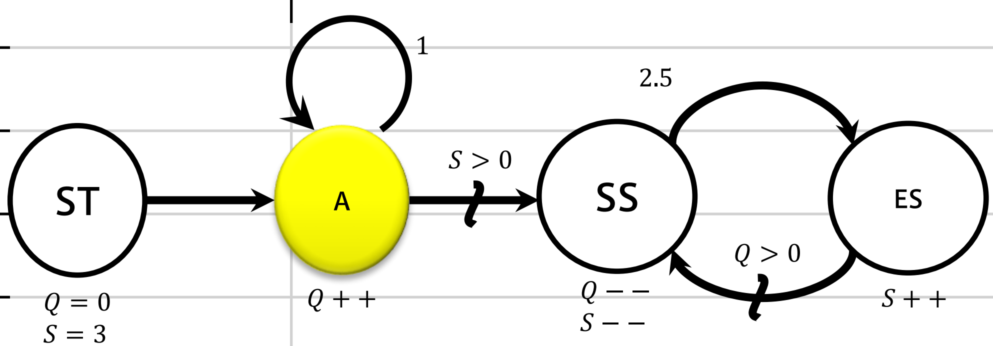
time	event	action	schedule
0	ST	Q=0; S=2;	A at t+0
0	A	Q=Q+1=1	A at t+1=1; SS at t+0=0
0	SS	Q=Q-1=0; … 1=1	… at t+2 …
1	A	Q=Q+1=1	A at t+1=2; SS at t+0=1
1	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=3.5
2	A	Q=Q+1=1	A at t+1=3; (SS condition not true)
2.5	ES	S=S+1=1;	SS at t+0=2.5
2.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=5
3	A		
3.5	ES		
5	ES		



Event List Multiple Server Queue

TU WIEN Mathematical Modelling and Simulation

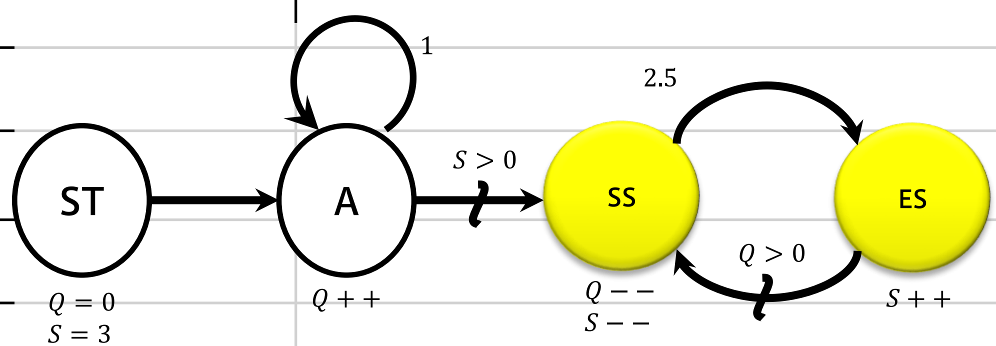
time	event	action	schedule
2.5	ES	S=S+1=1;	SS at t+0=2.5
2.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=5
3	A	Q=Q+1=1	A at t+1=4; (SS condition not true)
3.5	ES		
5	ES		
4	A		



Event List Multiple Server Queue

Mathematical Modelling and Simulation

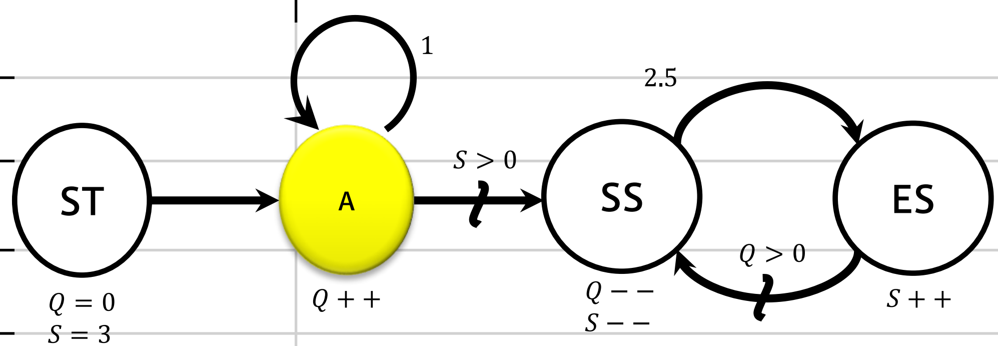
time	event	action	schedule
2.5	ES	S=S+1=1;	SS at t+0=2.5
2.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=5
3	A	Q=Q+1=1	A at t+1=4; (SS condition not true)
3.5	ES	S=S+1=1	SS at t+0=3.5
3.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=6
4	A		
5	ES		
6	ES		



Event List Multiple Server Queue

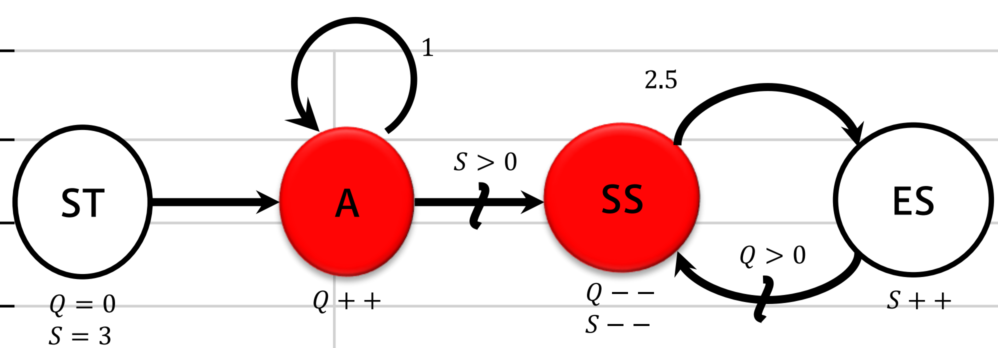
TU WIEN
Mathematical Modelling and Simulation

time	event	action	schedule
3.5	ES	S=S+1=1	SS at t+0=3.5
3.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=6
4	A	Q=Q+1=1;	A at t+1=5; (SS condition not true)
5	ES		
5	A		
6	ES		



Event List Multiple Server Queue

time	event	action	schedule
3.5	ES	S=S+1=1	SS at t+0=3.5
3.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=6
4	A	Q=Q+1=1;	A at t+1=5; (SS condition not true)
5	ES	S=S+1=1;	SS at t+0=5
5	*SS*	simultaneous events – ordering problems	
5	A		
6	ES		



Event List Multiple Server Queue

Mathematical Modelling and Simulation

time	event	action	schedule
3.5	ES	S=S+1=1	SS at t+0=3.5
3.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=6
4	A	Q=Q+1=1;	A at t+1=5; (SS condition not true)
5	ES	S=S+1=1;	SS at t+0=5
5	***SS***	Q=Q-1=0; S=S-1=0	ES at t+2.5=7.5
5	A	Q=Q+1=1;	A at t+1=6; (SS condition not true)
6	ES		
7.5	ES		
6	A		

Which one should occur first?
Does it matter?

Event List Multiple Server Queue

Mathematical Modelling and Simulation

time	event	action	schedule
3.5	ES	S=S+1=1	SS at t+0=3.5
3.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=6
4	A	Q=Q+1=1;	A at t+1=5; (SS condition not true)
5	ES	S=S+1=1;	SS at t+0=5
5	A	Q=Q+1=2;	A at t+1=6; SS at t+0=5
5	***SS***	Q=Q-1=1; S=S-1=0	ES at t+2.5=7.5
6	ES		
7.5	ES		
6	A		
5	SS		

Which one should occur first?
Does it matter?

Event List Multiple Server Queue

time	event	action	schedule
3.5	ES	S=S+1=1	SS at t+0=3.5
3.5	SS	Q=Q-1=0; S=S-1=0	ES at t+2.5=6
4	A	Q=Q+1=1;	A at t+1=5; (SS condition not true)
5	ES	S=S+1=1;	SS at t+0=5
5	A	Q=Q+1=2;	A at t+1=6; SS at t+0=5
5	SS	Q=Q-1=1; S=S-1=0	ES at t+2.5=7.5
5	SS	Q=Q-1=0; **S=S-1=-1**	ES at t+2.5=7.5
6	ES		
6	A		
7.5	ES		

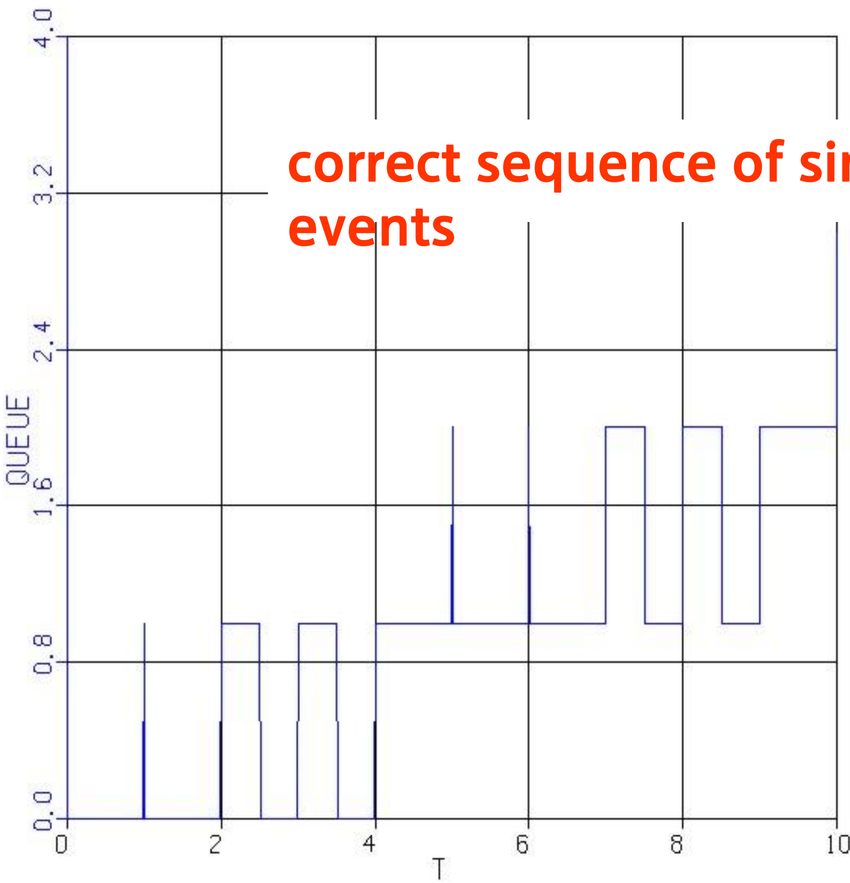
WRONG ORDER, WRONG RESULTS

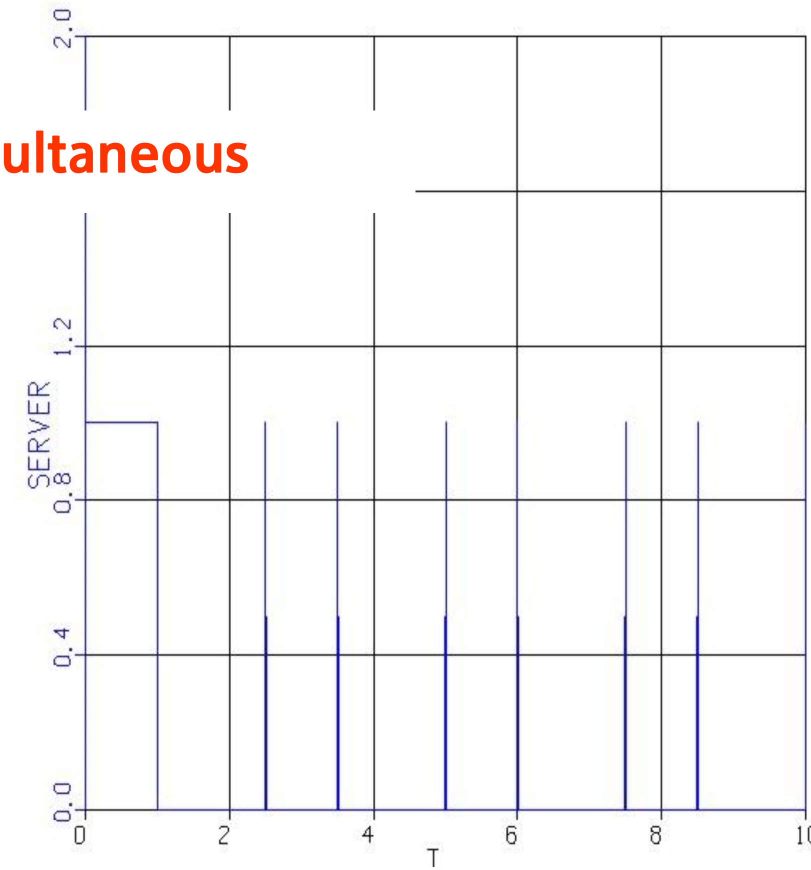
Simultaneous Events

- Simultaneous events occur when more than one event is schedule to occur the exactly the same time.
- In some cases the order of execution of the events is irrelevant, but in other cases certain permutations of the order of occurrence impact the outcome dramatically, often leading to invalid state trajectories and inadmissible values of state variables.
- Event Graph methodology provides the capability of prioritizing scheduling edges, so that simultaneous occurrences of the scheduled event always occur before other scheduled events.
- Although these edge priorities are typically not indicated on the graph itself, all software implementations of Event Graph methodology support edge prioritization.

Simulation Multiple Server Queue

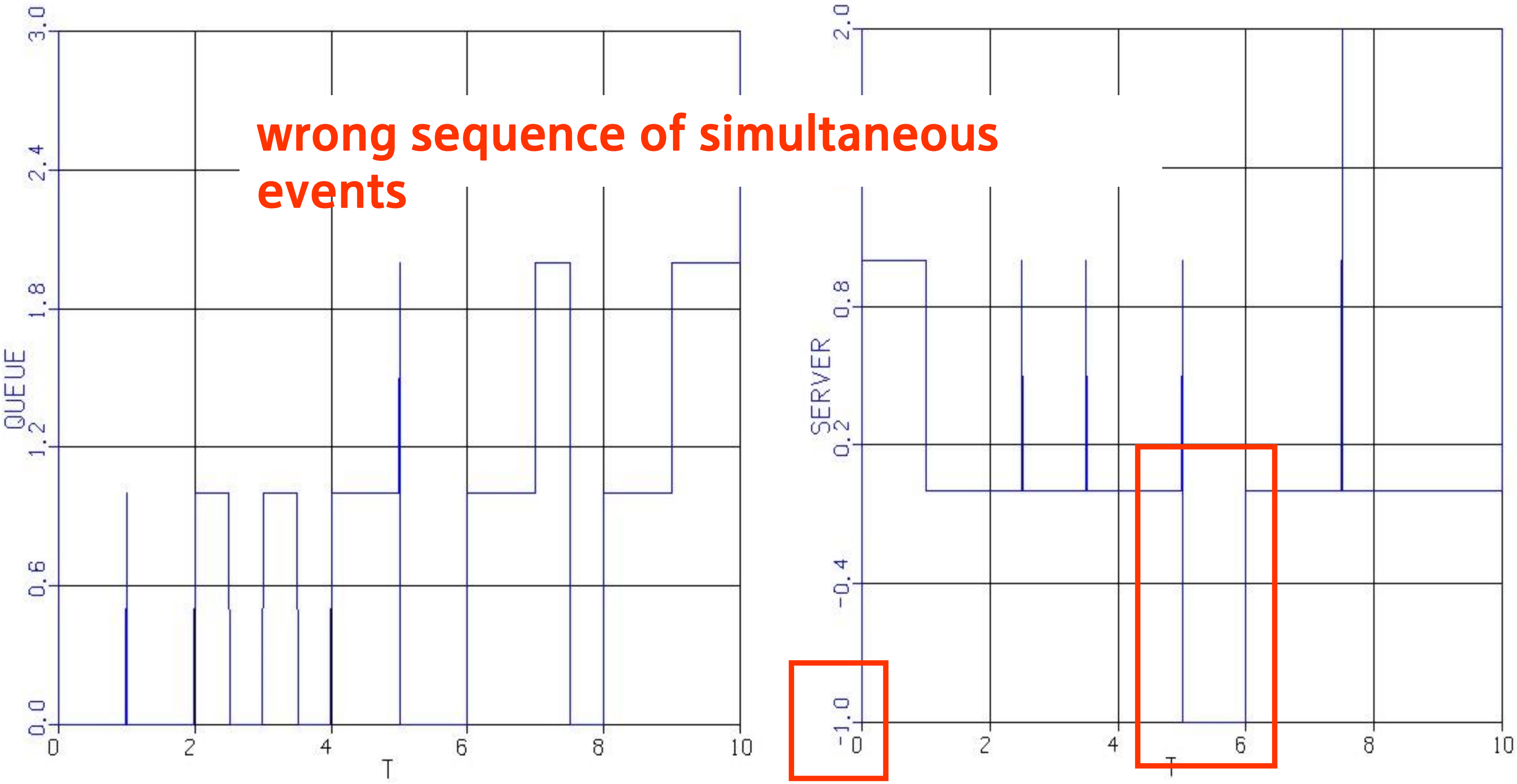
correct sequence of simultaneous events





t_arrival = 1,   t_service = 2.5,   max_server = 2

Simulation Multiple Server Queue



t_arrival = 1,   t_service = 2.5,   max_server = 2

ANALYSIS OF QUEUING MODELS

Terminology

Mathematical Modelling and Simulation

- Abbreviation of Queues:

Arrival Time	Service Time	Servers
Determinisitic D	Determinisitic D	One 1
Markovian M	Markovian M	Multiple m
General G	General G	

$\Rightarrow$ Possible combinations:
D/D/1, M/D/m, G/D/m, M/M/m, ...

Terminology

Mathematical Modelling and Simulation

- „Deterministic“: t is Constant
- „Markovian“: Distribution of t is memoryless.
  I.e. Exponentially distributed $t \sim E(\lambda)$
  $\Rightarrow$ times become a Markov-process
- „General“ : Distribution of t is arbitrary (positive)

- Deterministic Queues (D/D/1, D/D/m):

$$\frac{servicetime}{servers} > arrivaltime$$
$$\Rightarrow \color{red}{unstable}$$

$$\frac{servicetime}{servers} \leq arrivaltime$$
$$\Rightarrow \color{green}{stable}$$

- Stochastic Queues (M/M/1,G/M/m,...):

$$\frac{E(servicetime)}{servers} \geq E(arrivaltime)$$
$$\Rightarrow unstable$$

$$\frac{E(servicetime)}{servers} < E(arrivaltime)$$
$$\Rightarrow stable$$

Analysis of Queues

- Notation
  - Y_k – time elapsed between (k-1)th and k-th arrival

    $E(Y_k) = \frac{1}{\lambda}$ … average interarrival time

    (λ is the average arrival rate)

  - Z_k – k-th customer service time

    $E(Z_k) = \frac{1}{\mu}$ … average service time

    (μ is the average service rate)

  - W_k – k-th customer waiting time
  - $X(t)$ – average queue length

- Customer system time

  $S_k = W_k + Z_k$, the time k-th customer spends in the system

  $E(W_k) = W$ ... average waiting time

  $E(S_k) = T$ ... average system time, $T = W + \frac{1}{\mu}$

- Little's law
  - $\overline{N}$ ... average number of customers in the system

$$\overline{N} = \lambda T$$

  - special cases

    $\overline{X(t)} = \overline{N}_q = \lambda W$ ... average number of customers in the queue

    $\overline{N}_s = \frac{\lambda}{\mu}$ ... average no. of customers in service

Results M/M/1 queues:

- Average waiting time in the queue

$$W = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\rho}{(1-\rho)\mu}, \qquad \rho = \frac{\lambda}{\mu}$$

- Average length of the queue

$$\overline{X(t)} = \bar{N}_q = \lambda W = \frac{\rho^2}{1-\rho}$$

- Average system time of customers

$$T = W + \frac{1}{\mu} = \frac{1}{\mu - \lambda} = \frac{1}{(1-\rho)\mu}$$

- Average number of customers in the system

$$\bar{N} = \lambda T = \frac{\rho}{1-\rho}$$

Results M/G/1 queues:

- Exponential distribution of interarrival times
- Service times are mutually independent and distributed arbitrarily with parameters

$E(Z_k) = \frac{1}{\mu}$ in $var(Z_k) = \sigma^2$, we define also $\rho = \frac{\lambda}{\mu}$

- Average queue length

$$\overline{X(t)} = \bar{N}_q = \frac{\rho^2}{2(1-\rho)}(1 + \mu^2\sigma^2)$$

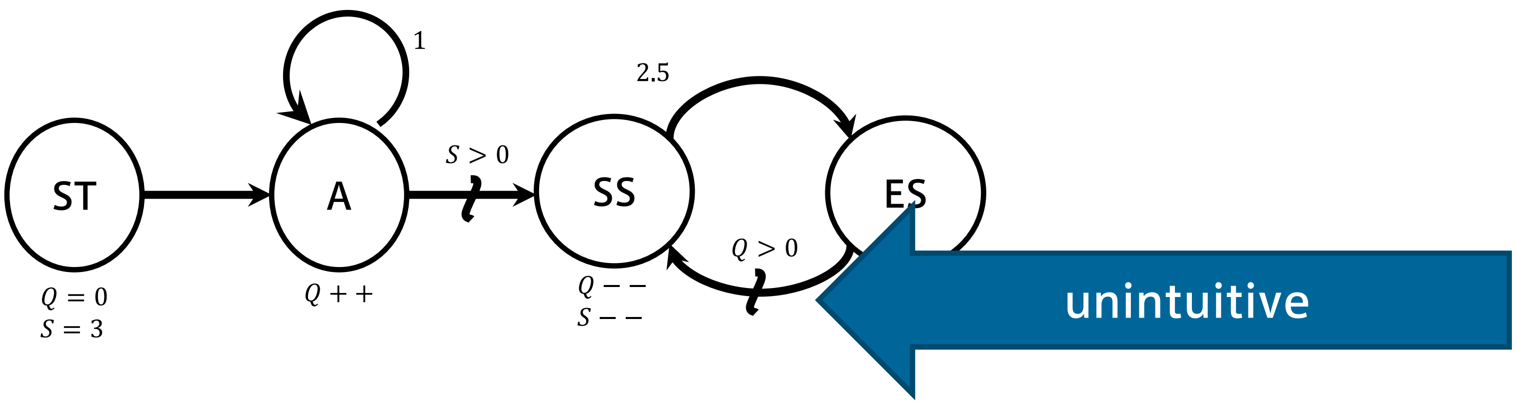
- Average number of customers in the system

$$\bar{N} = \bar{N}_q + \rho = \frac{\rho}{1-\rho} - \frac{\rho^2}{2(1-\rho)}(1 - \mu^2\sigma^2)$$

OTHER SIMULATION ENVIRONMENTS

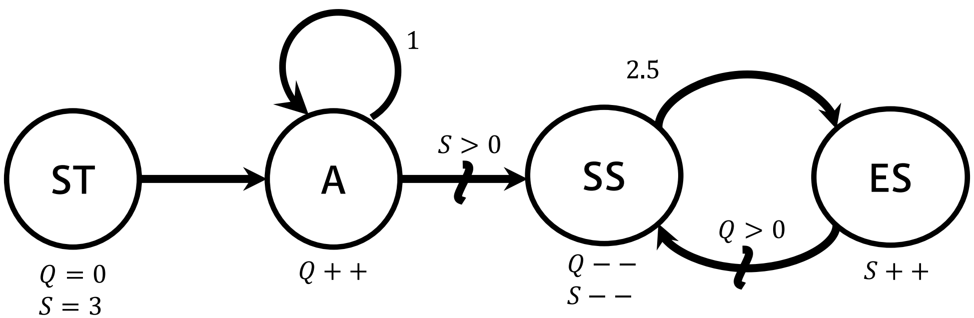
Other Simulation Environments

- Most DES models are based on entities being processed in a sysrem
- Therefore they use very similar process structures
- Event Graph description sometimes unnecessary general and unintuitive

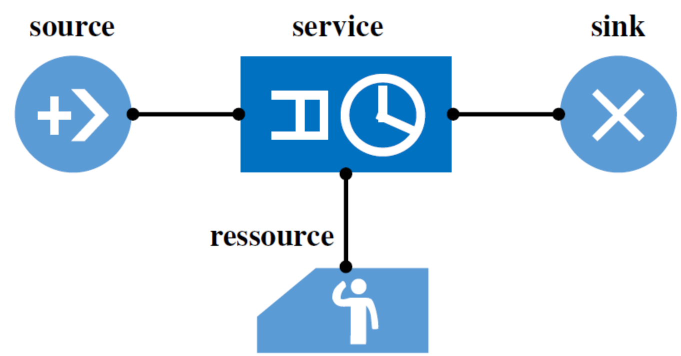


Other Simulation Environments

- DES Simulators for simulation of processes usually use a more intuitive description

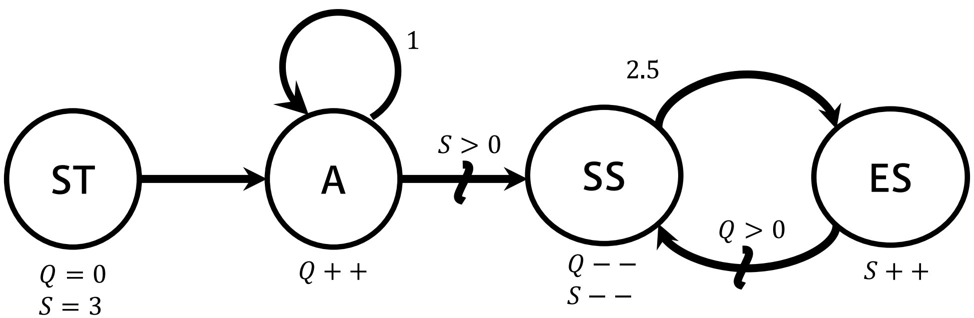


DES Modeling in AnyLogic

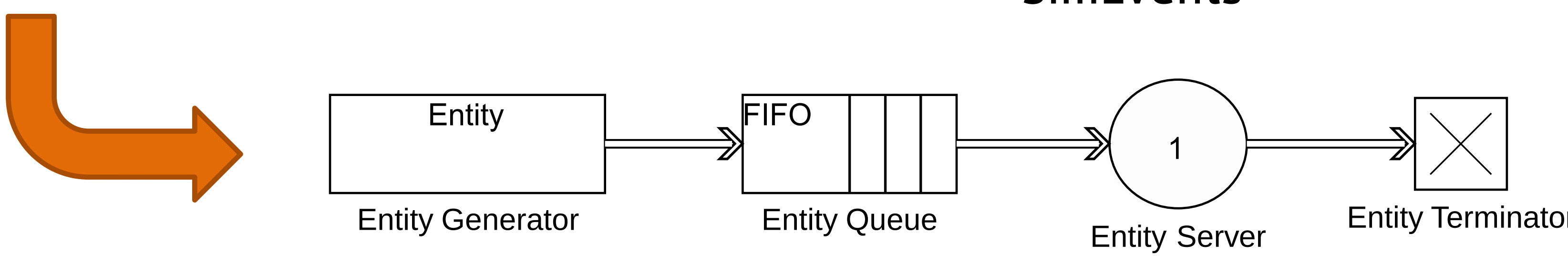


Other Simulation Environments

- DES Simulators for simulation of processes usually use a more intuitive description



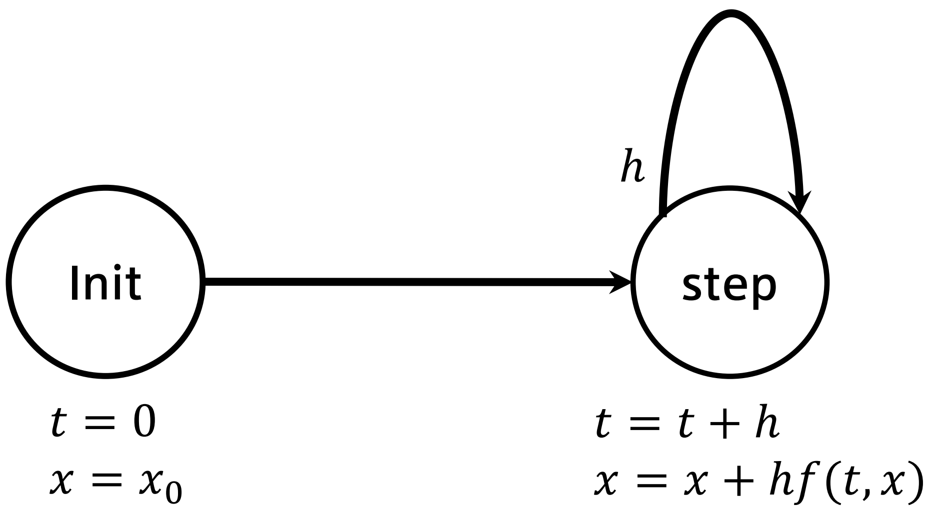
DES Modeling in SimEvents



EVENT GRAPHS BEYOND ENTITIES

Beyond Queues

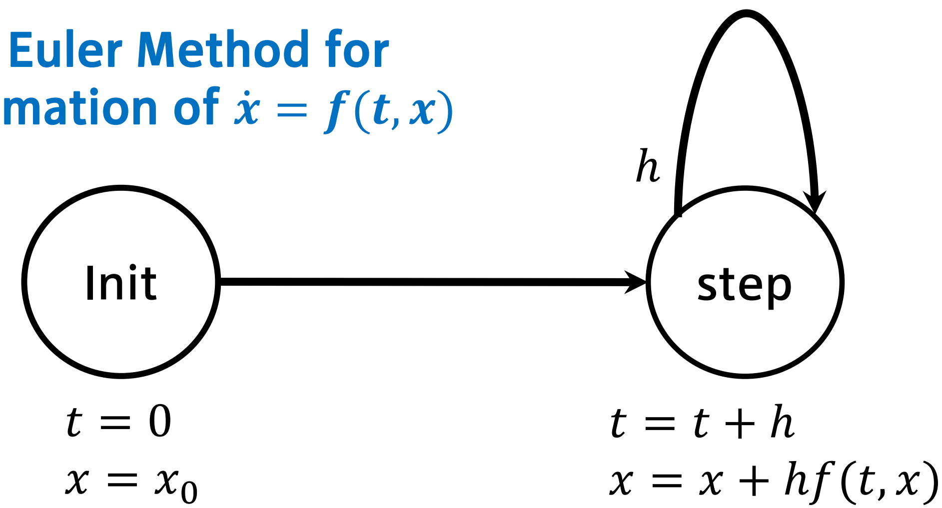
- DES / Event Graphs not only interesting for queuing systems.



Beyond Queues

- DES / Event Graphs not only interesting for queuing systems.

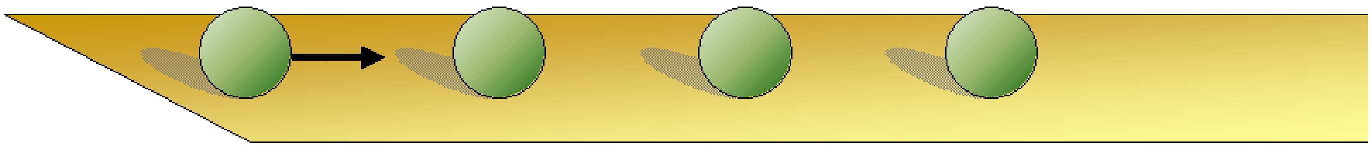
Explicit Euler Method for approximation of $\dot{x} = f(t, x)$



Beyond Queues

- DES / Event Graphs not only interesting for queuing systems.

Case Study 1: Collision of Spheres



Beyond Queues

- DES / Event Graphs not only interesting for queuing systems.