Simulation

Lecturer: Björn Landfeldt, Saeed Bastani

Department of Electrical and Information Technology

Bjorn.Landfeldt@eit.lth.se, Saeed.Bastani@eit.lth.se

Tuesday 21st March, 2017

Course Contents and Evaluation Criteria

- Lectures
- Labs (three)
- Home assignments (three, assessable)
- Home exam (if you need grade 4 or 5)

Important dates

- For home assignment 1: 5^{th} of May
- For home assignment 2: 26^{nd} of May
- For the project: 9^{th} of June
- For the home exam: 1^{st} of September

If you miss a deadline you can not get a higher grade than 3.

Literature

1. A free compendium available from the university library, see the home page
2. Material on the home page
   - Home assignments
   - Project
   - Slides from the lectures
   - Course program
   - etc.

Course content

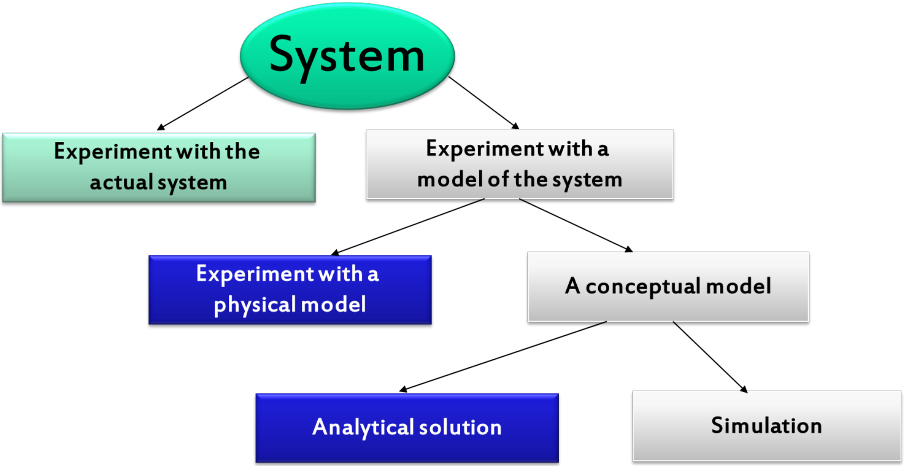
- Discrete event simulation (investigate system)
- Short intro to optimization
- Meta heuristics for simulated optimization

What is simulation?

- Simple synonym: imitation
- We are interested in studying a system

But, instead of experimenting with the system itself we experiment with a model of the system

How to study a system



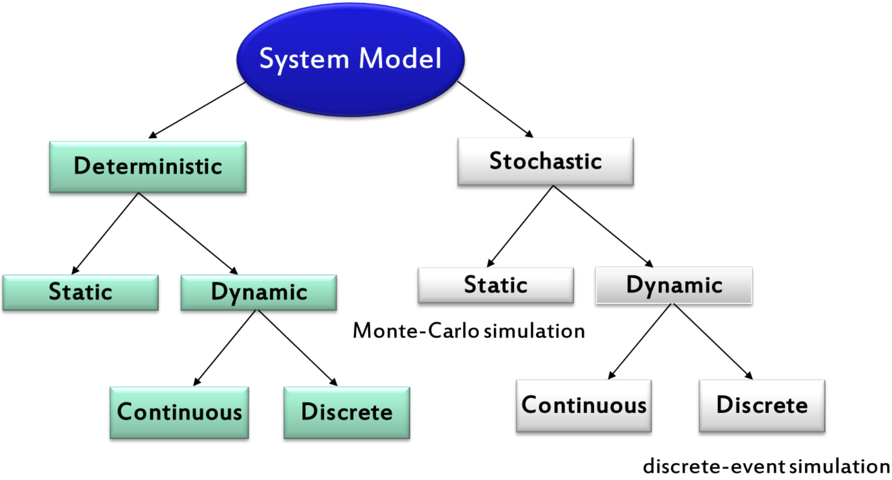
Why not experiment with the system itself?

1. It might be dangerous (control system in a nuclear power plant)
2. The system does not exist yet
3. It is expensive to experiment with the system
4. It is impossible to experiment with a system

What kind of systems do exist?

1. Continuous systems
   - Example: air pressure around an aeroplane
   - Are usually modelled by differential equations
2. Discrete systems
   - Example: systems described by queues
3. Hybrid systems
   - Example: a bouncing ball (continuous dynamics before a bounce and discrete dynamics after each bounce

Characterizing a model

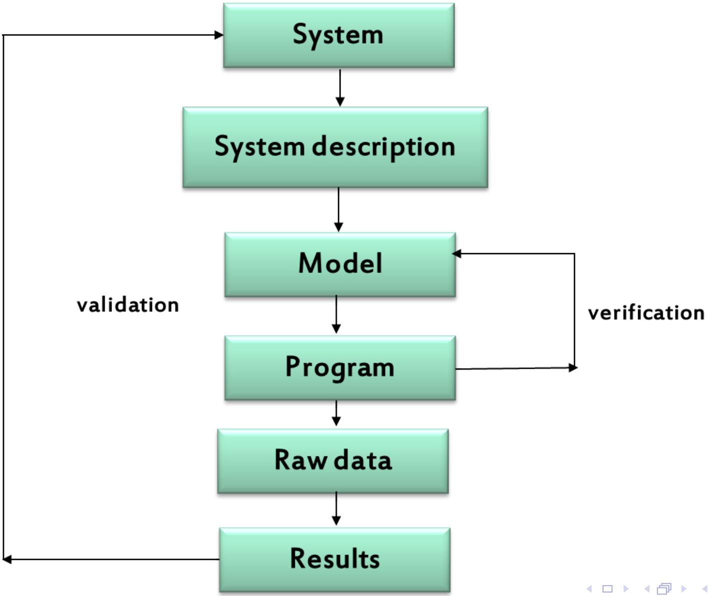


Different kinds of models

1. Continuous – discrete
2. Static – dynamic
3. Deterministic – stochastic

In this course our models will be discrete, dynamic and (usually) stochastic.

Steps in a simulation study



Application areas

1. Communication systems
2. Computer systems performance
3. Transportation
4. Manufacturing and material handling
5. Health systems
6. Public services
7. Military systems

Advantages of simulation

1. Makes it possible to predict impact of changes
2. Makes it possible to look at detailed behaviour
3. Can give a good understanding of a system
4. Can visualize a system
5. Find bottlenecks in a system
6. Gives a possibility to train a team

Disadvantages of simulation

1. Model building requires special training
2. Time consuming and expensive
3. Limitations of accuracy (approximations or rare events)

Modelling concepts

- A model is an abstract representation of a system
- A discrete model has
  - ⋆ State variables
  - ⋆ Events that change the state
  - ⋆ Rules that describes what shall happen at an event

Modelling concepts

- Entity – represents an object that requires definition
  - ⋆ Can move around (customers, airplanes etc.)
  - ⋆ Can be static (bank clerk, highway crossing etc)
- Attributes – entities have attributes, i.e. data connected to the attribute
- Resource – an entity that serves other entities

Modelling structures

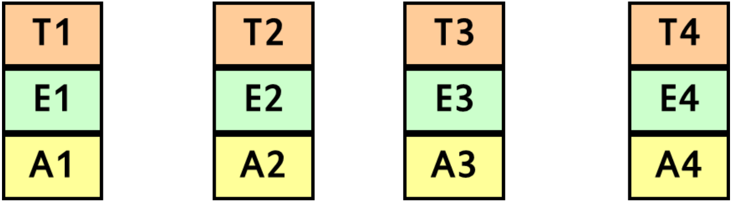
- Event-scheduling method
- Process-interaction method
- Other methods (later in the course)

Event-scheduling method

The following is needed:

1. A description of the state
2. The events that can occur
3. Rules describing what will happen if an event occurs

Keeps track of when events shall happen

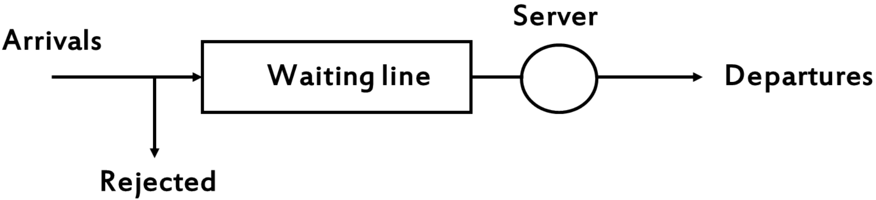


T_i = time when event E_i will take place
A_i = attributes to event i
The list is sorted: $T_1 < T_2 < T3 < T_4$ etc.

How a simulation run is done?

1. Extract the first element in the event list
2. Set Time = the time of the extracted event
3. Update the state of the system and insert new events if needed
4. If not finished, Go to 1

An example: a queuing system



It might be of interest to find

- Probability of rejection
- Mean (or variance) of time spent in system
- The mean number of customers in the system

Assume that we want to find the mean number of customers in the queue.

N = number of customers in the system

The appropriate state description depends on the results we look for.

Events that may take place

- Arrival
- Departure (when service is finished)
- Measurement (does not change the state)

What do we also need to know?

Assume the following:

- The service time distribution is exponential with mean 2
- The mean time between arrivals is exponential with mean 3
- The number of places in the waiting line is infinite

Rule at arrival

N := N + 1;
If N=1 then
    add departure to event list ;
Add a new arrival to event list;

When we add the departure and arrival we have to draw a random number (exponentially distributed)

Rule at departure

N := N - 1;
If N $>$ 0 then
add departure to event list ;

Rule at measurement

Write (N);
Add a new measurement to event list ;

When the simulation begins

System State		Event List	
Sim_Time	N	Event_Time	Event_Type
0	0	3	Arrival
		5	Measurement

Step 1

System State		Event List	
Sim_Time	N	Event_Time	Event_Type
3	1	4	Arrival
		5	Measurement
		9	Departure

Step 2

System State		Event List	
Sim_Time	N	Event_Time	Event_Type
4	2	5	Measurement
		9	Departure
		10	Arrival

Step 3

System State		Event List	
Sim_Time	N	Event_Time	Event_Type
5	2	9	Departure
		10	Arrival
		14	Measurement

Step 4

System State		Event List	
Sim_Time	N	Event_Time	Event_Type
9	1	10	Arrival
		12	Departure
		14	Measurement

Main Process

```
a := 3; ( mean time between arrivals = 3 )
s := 2; ( mean service time = 3)
m := 10; ( mean time between measurements = 10)
simulationLength := 1000;
No_in_queue := 0;
time := 0;
insert_event(measurement,Exp(m));
insert_event(arrival, Exp(a));
while time < simulationlength do
begin
  dummy := FirstInQueue(eventlist);
  time := dummy.eventtime;
  case dummy.eventkind of
    arrival: arrive;
    departure: depart;
    measurement: measure;
  end;
end;
end.
```

Other Processes

```
procedure arrive;
begin
   if No_in_queue = 0 then
      insert_event(departure,Exp(s));
   No_in_queue := No_in_queue + 1;
   insert_event(arrival, Exp(a));
end;
procedure depart;
begin
   No_in_queue := No_in_queue - 1;
   if No_in_queue > 0 then
      insert_event(departure, Exp(s));
end;
procedure measure;
begin
   write(utfil, No_in_queue);
   insert_event(measurement, Exp(m));
end;
```

First home assignment

You will get a Java program that simulates a simple queue.
You shall modify that program and do some investigations.

THE END