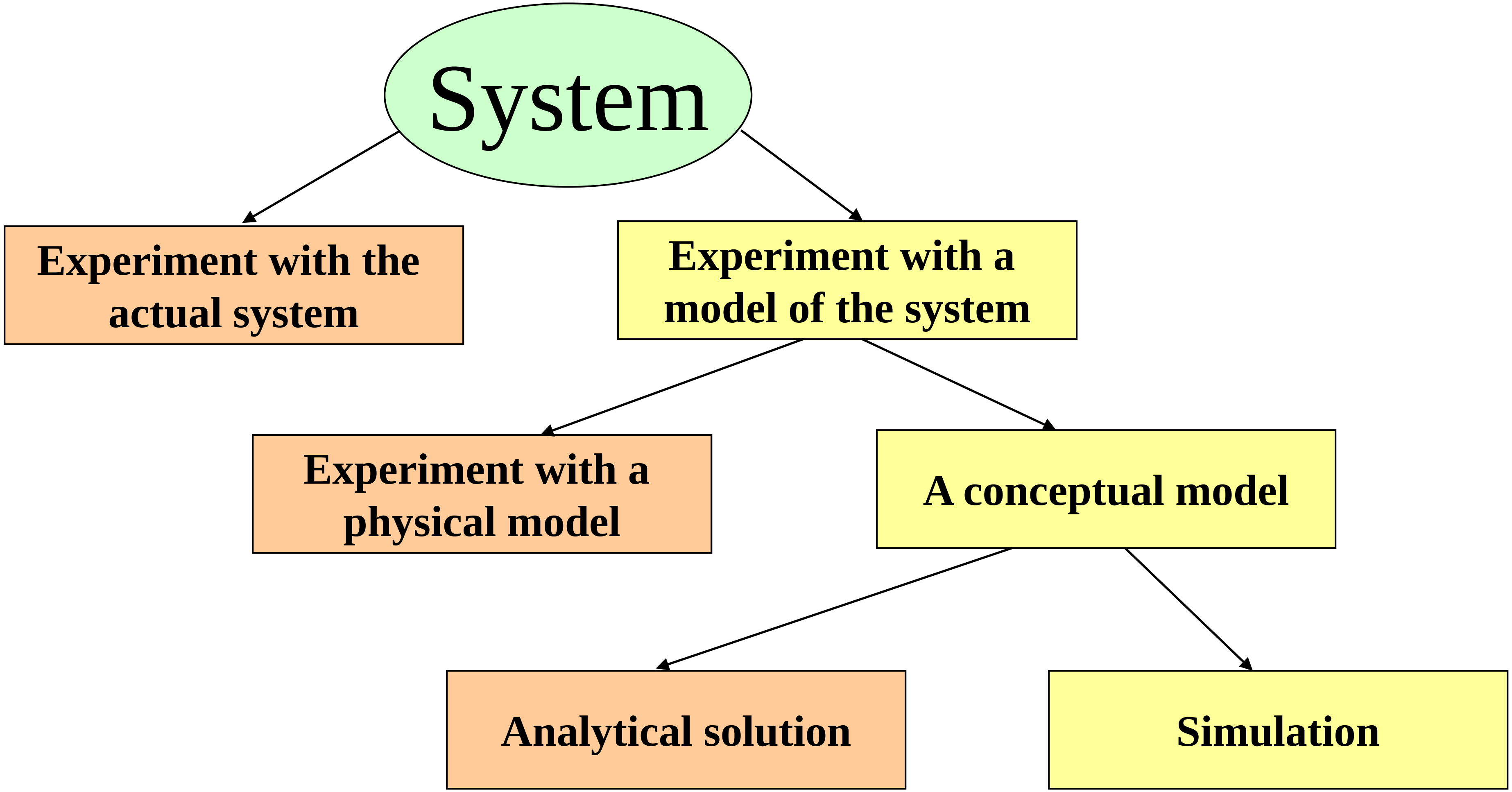
Simulation

LUND INSTITUTE
OF TECHNOLOGY
Lund University

What is simulation?

- Simple synonym: imitation
- We are interested in studying a system
- Instead of experimenting with the system itself we experiment with a model of the system

Ways to study a system



Why not experiment with the system itself?

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

Different kinds of systems

- Continuous systems
  - Examples: temperature in an engine, air pressure around an aeroplane etc
  - Are usually modelled by differential equations
- Discrete systems
  - Examples: systems described by queues
- Hybrid systems
  - Both continuous and discrete

Application areas

- Communication systems
- Computer systems performance
- Transportation
- Manufacturing and material handling
- Health systems
- Public services
- Military systems
- ,,,,,

Advantages of simulation

- Makes it possible to predict impact of changes
- Makes it possible to look at detailed behaviour
- Can give a good understanding of a system
- Can visualize a system
- Find bottlenecks in a system
- Gives a possibility to train a team

Disadvantages of simulation

- Model building requires special training
- Time consuming and expensive
- Limitations of accuracy (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

Two approaches to simulation

- Event-scheduling method
- Process-interaction method

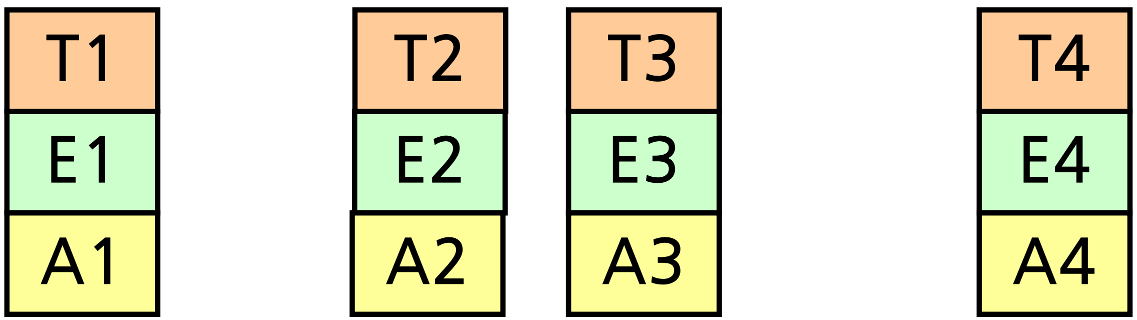
Event-scheduling method

The following is needed:

- A description of the state
- The events that can occur
- Rules describing what will happen if an event occurs

The event list

Keeps track of when events shall happen



T_i = time when event E_i will take place

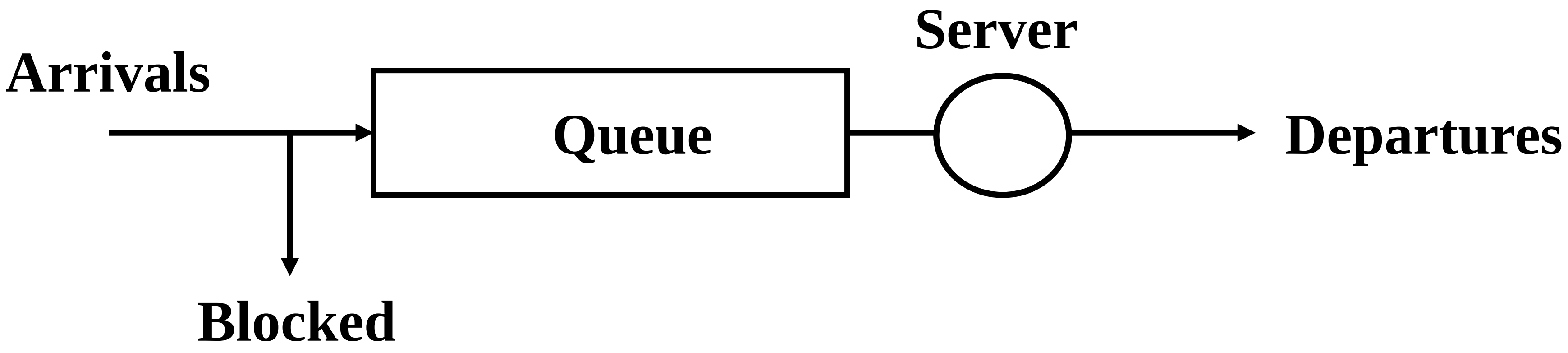
A_i = attributes to event I

The list is sorted: $T_1 < T_2 < T_3 < T_4$

How a simulation is performed

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 the

- **probability of rejection**
- **mean time spent in the system**
- **mean number of customers in the system**

The state description

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 desire

Events that may take place

- Arrival (new customer come)
- Departure (when service is ready)
- Measurement (does not change the state)

What we also need to know

Assume the following:

- The mean time between arrivals is exponentially distributed (*exp(a)*)
- The service time is exponentially distributed (*exp(s)*)
- The number of slots in the queue 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 an arrival event 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 is initiated

Time and state:

Time = 0

N = 0

Event list (*time*, *event*):

3 Arrival

5 Measurement

(One "arrival event" and one "measurement event" has to be put into the list to initiate the simulation.)

Step 1

Time and state:

Time = 3

N = 1

Event list:

4 Arrival

5 Measurement

9 Departure

Step 2

Time and state:

Time = 4

N = 2

Event list:

5 Measurement

9 Departure

10 Arrival

Step 3

Time and state:

Time = 5

N = 2

Event list:

9 Departure

10 Arrival

14 Measurement

Step 4

Time and state:

Time = 9

N = 1

Event list:

10 Arrival

12 Departure

14 Measurement

```
begin
   simulationlength := 1000;
   No_in_queue := 0;
   time := 0;
   insert_event(measurement, Random(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.
```

Detta är pseudokod

```
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, Random(m));
end;
```