OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Introduction to Simulation

Discrete Event Simulation

Motivation & Contents

Motivation

- Discrete–event simulation is very common
- It is used to analyse & optimise many types of system
- There is lots of professional software available
- We need to understand the basics of how it works

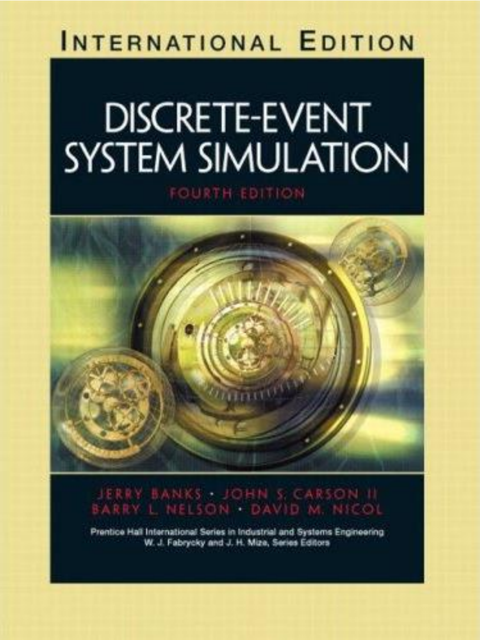
Contents

- Models and some modelling terminology
- How a discrete–event simulation works
- The classic example – the queue in the bank
- Example of a discrete–event simulation

Background Reading

Relevant sections of the book:

- 1.5
- 1.6
- 1.11
- 2.1
- 3.1
- 3.2



Simulation

A definition of simulation:

1. Imitation of the operation of a real-world system
2. Generation of an artificial history of a system
3. Observation of the artificial history

Simulation is performed using a *model*

Model

A model:

- is a set of assumptions about the operation of the system

These assumptions can be:

- algorithmic (a sequence of steps)
- mathematical (a set of equations)
- logical (a set of conditions)

This model is "run" in order to study the real system

Abstraction

Models are built using *Abstraction*

There are two types of abstraction:

- Simplification
- Idealisation

Simplification means leaving out unnecessary details

- Represent only selected attributes of objects
- Leave out unimportant effects

Idealisation means replacing real things by concepts

- Replace a set of measurements by a mathematical function
- Replace an object by theoretical abstractions

Abstraction

Examples in a Solar System model:

- Simplification: Leave out the asteroids
- Idealisation: Replace the earth by its centre of gravity

Examples in a restaurant model:

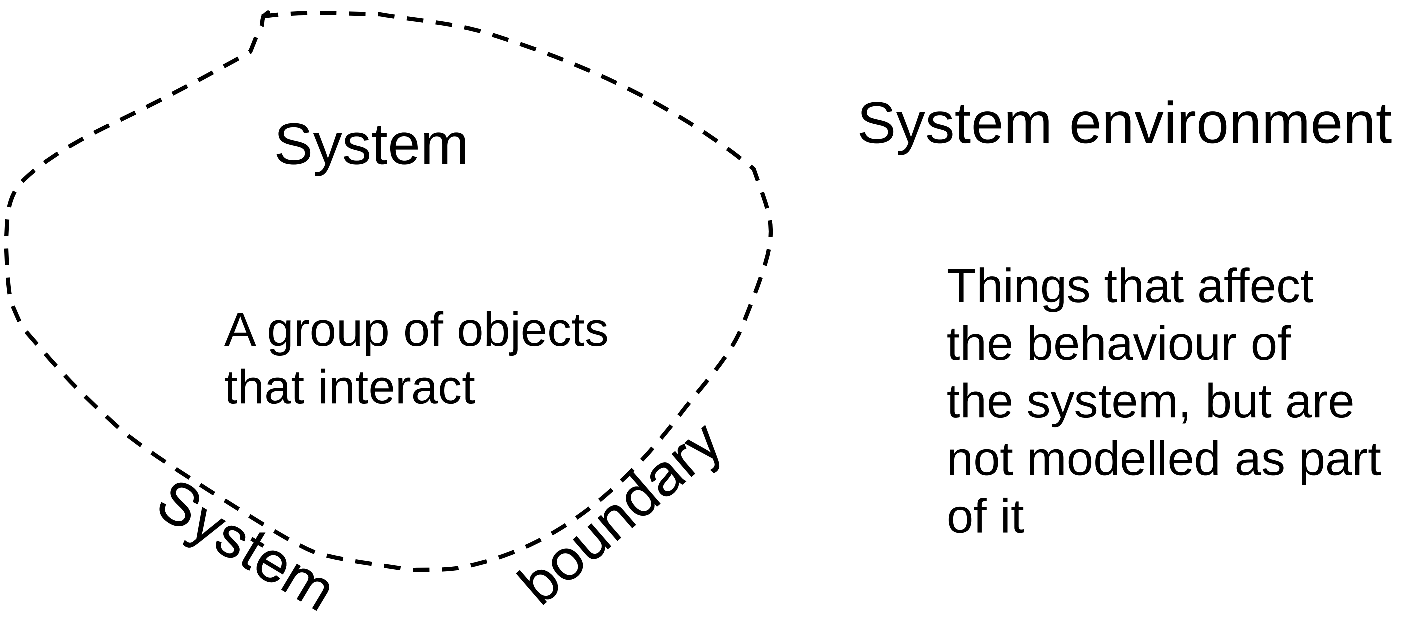
- Simplification: Leave out customer gender and age
- Idealisation: Replace inter-arrival times by their probability distribution

These examples demonstrate:

- The decision whether (and how) to abstract is not trivial!

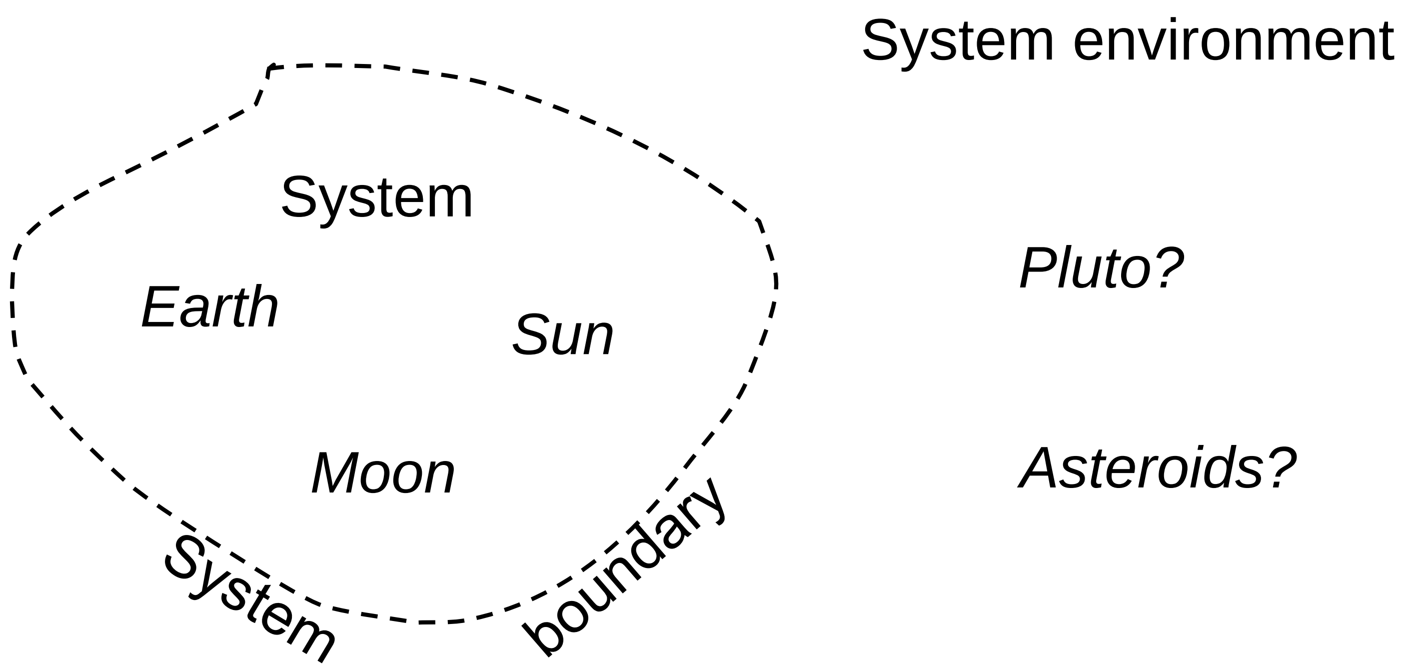
Systems

Placing the system boundary is the first difficult task in modelling



Systems

Example: Modelling the Earth's orbit:



Entities & Attributes

An entity is an object of interest in the system

customer *manager* *cashier*

An attribute is a (relevant) property of an entity

account balance *gender* *skills*

Attributes are part of the system, i.e. state variables

Activities & Delays

An activity is a duration of known length

check balance *drink coffee* *serve customer*

Activities form part of the model specification

A delay is a duration of unknown length

Waiting time in queue *Time until 10 customers have been served*

Delays form part of the simulation result

State & State Variables

The (system) state is a description which is

- complete
- minimal

It is the information needed to continue the simulation

A state variable is a variable which is part of the state

length of queue (0, 1, 2, ...)	*current activity of manager (sleeping, coffee break, ...)*	*Time of next customer arrival (T+13s)*

Events

An event is an occurrence which

- is instantaneous
- changes the state of the system

Customer arrives from outside

Manager wakes up

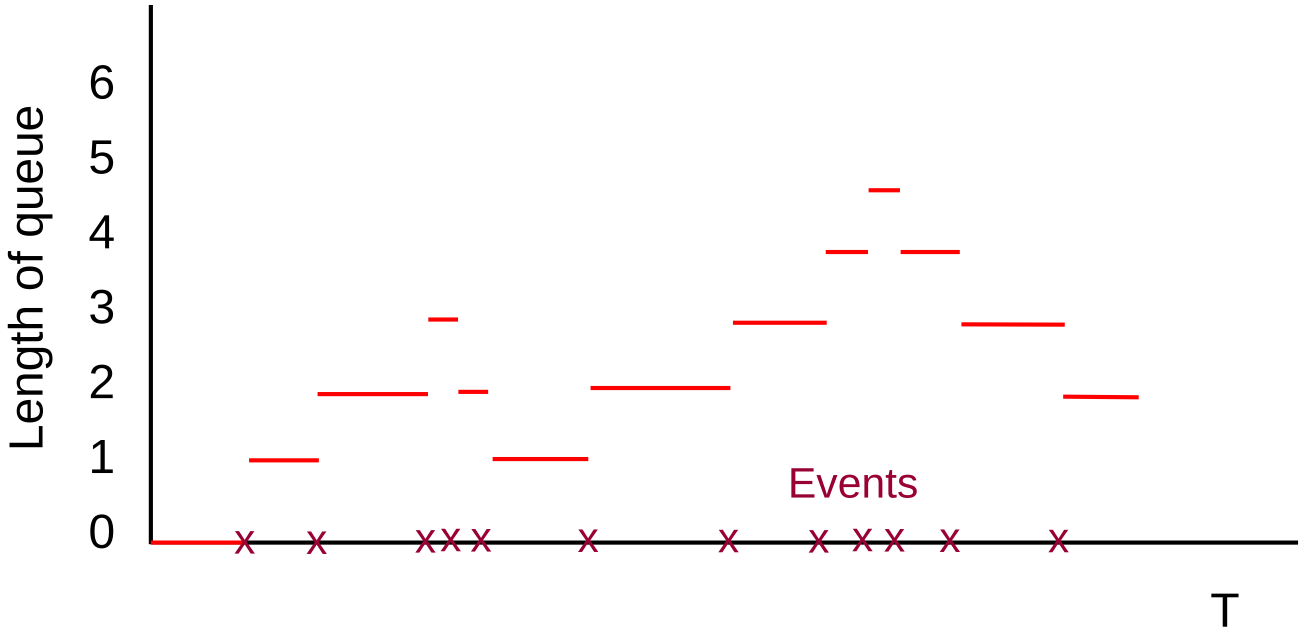
Cashier completes service

Activities are begun and ended by events

The fact that events take no time has important consequences!

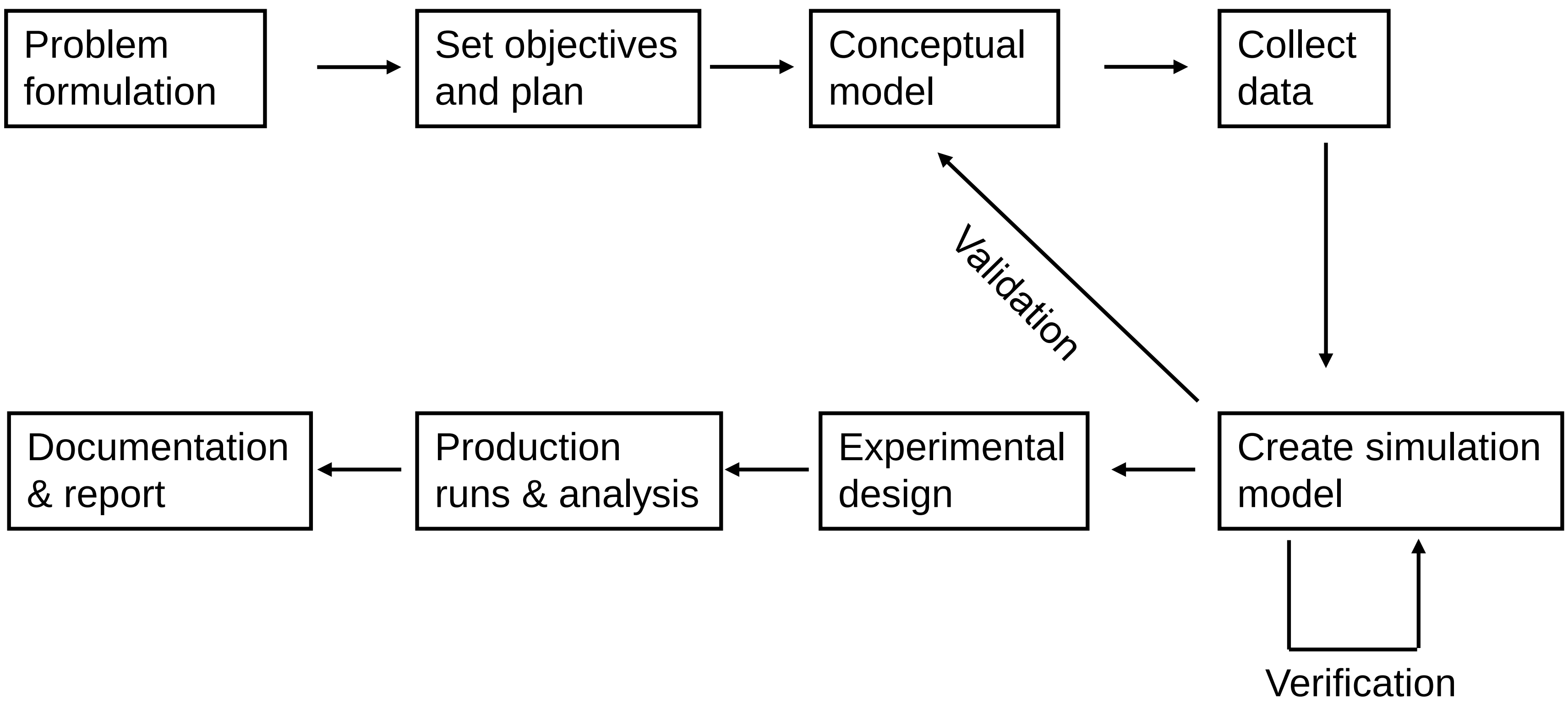
Discrete Time Behaviour

A typical (discrete time) simulation result:



The solution changes at discrete times (only!)

Steps in a Simulation Study



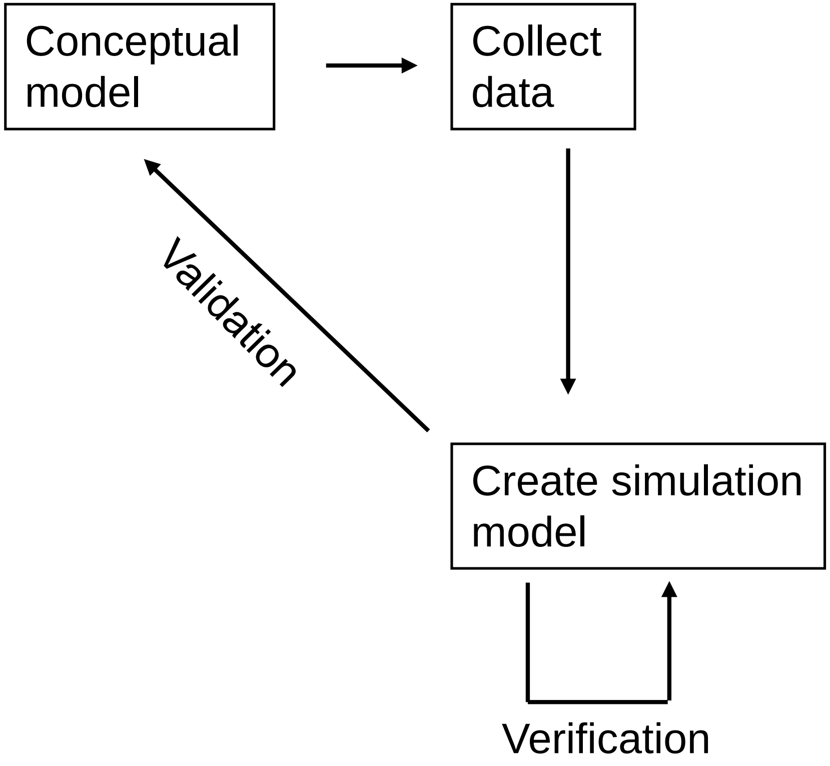
Validation & Verification

Validation means creating a model that accurately represents reality

"building the correct model"

Verification means writing a program that corresponds to its specification

"building the model correctly"



Model Specification

Discrete-event modelling asks the following questions:

- What events are possible?
- How does each event affect the system state?
- What activities are there?
- How long do activities last?
- Which events begin and end each activity?
- What conditions (if any) must hold for each activity?
- How are delays defined?
- How must the simulation be initialised?

The Sims – Almost normal Family Life

What events are possible?

- The school calls to report damaged property
- The sons mood changes from heartbroken to in love

What activities are there?

- The father being employed or unemployed
- The time between two calls from school

How long do activities last?

- A period of unemployment lasts Uni(5,15) days
- The time until the school calls next is Exp(1/30.5)

Which events begin and end each activity?

- Finding a job starts a period of employment
- Falling in love ends a period of being heartbroken

Star Trek – USS Enterprise in Danger

What events are possible?

- An antimatter particle hits the shields
- The USS Saratoga arrives

What activities are there?

- One treatment of one crew member
- Repairing the engine

How long do activities last?

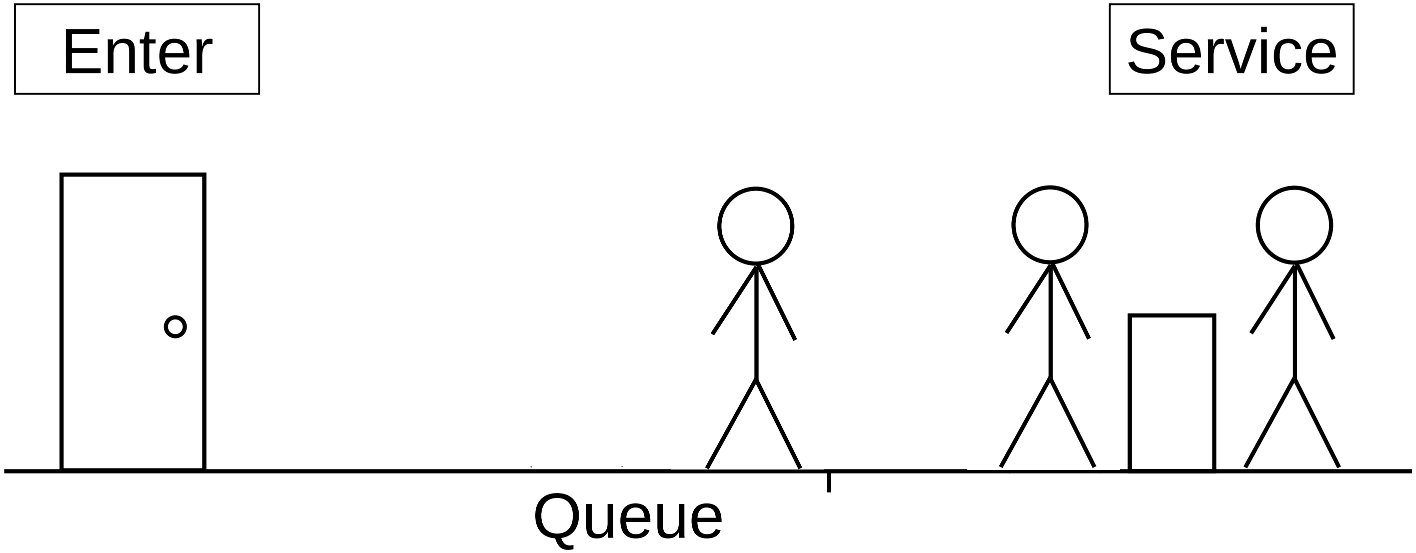
- Repair blocks take 22 minutes each
- Total repair time is estimated at Norm(150,20)

Which events begin and end each activity?

- The arrival of the USS Saratoga ends the simulation
- The end of a repair slot begins medical treatment

A Simulation Classic

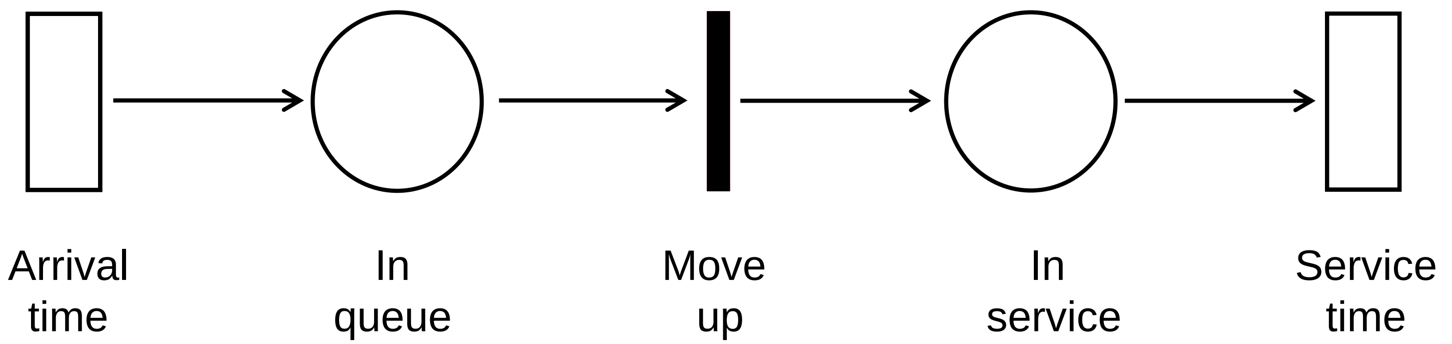
The single-server queue:



A Simulation Classic

There are many methods for modelling such systems

One common class of model: "Stochastic Petri Net"



We will look at SPNs later in the semester

A Simulation Classic

One possible problem formulation:

- *"Customers have to wait too long in my bank"*

A typical objective:

- Determine the effect of an additional cashier on the mean queue length

Data needed:

- Inter-arrival times of customers
- Service times

Primary & Conditional Events

Primary event:

- An event whose occurrence is scheduled at a certain time

Arrivals of customers

- Primary events can be predicted in the short term

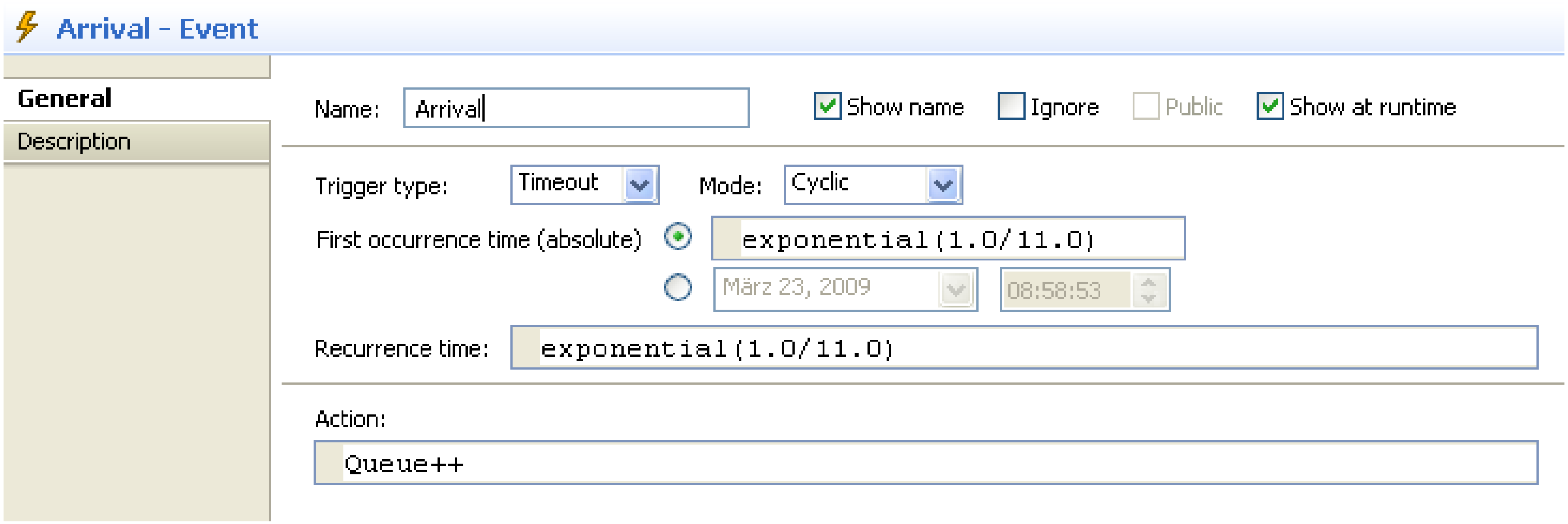
Conditional event:

- An event which is triggered by a certain condition becoming true

Customer moving from queue to service

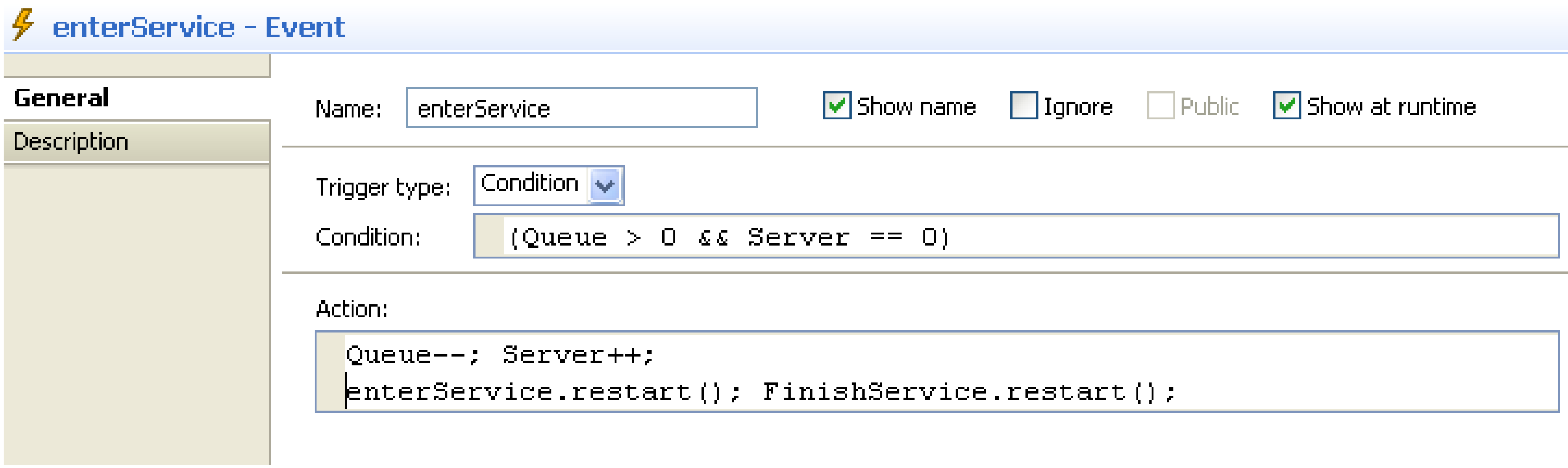
Primary & Conditional Events

Primary event:



Primary & Conditional Events

Conditional event:



Event Notice & Event List

Event notice:

- A data record specifying an event

The event notice must contain all the information necessary to execute the event (in particular the time it is scheduled to occur)

Future event list:

- A list of event notices for future events

The event list is the main data structure in a discrete–event simulator

The Event List

The future event list (FEL) …

- controls the simulation
- contains all future events that are already scheduled
- is ordered by increasing time of event notice
- contains only primary events

Example FEL for some simulation time $t \leq t_1$:

$$(t_1, \text{Event}_1) \longrightarrow (t_2, \text{Event}_2) \longrightarrow (t_3, \text{Event}_3) \longrightarrow (t_4, \text{Event}_4)$$

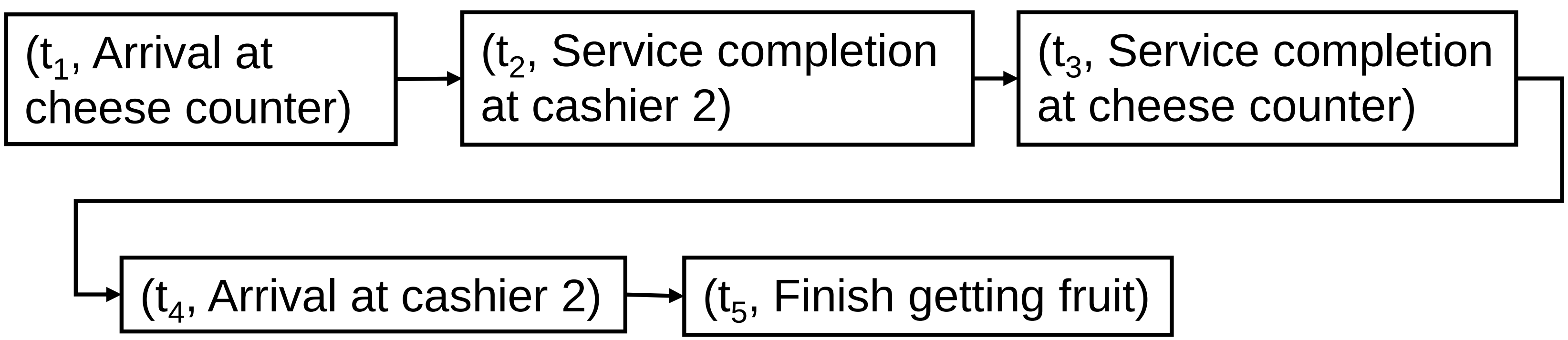
$$t_1 \leq t_2 \leq t_3 \leq t_4$$

Event List

Example: Simulation of a supermarket:

Some state variables:

- # people in fruit & salad section
- # people in cheese counter queue
- # people at cashier queue 2
- Time at which cashier 2 will finish processing customer



Event List

Operations on the FEL:

- Insert an event into FEL (at appropriate position!)
- Remove first event from FEL for processing
- Delete an event from the FEL

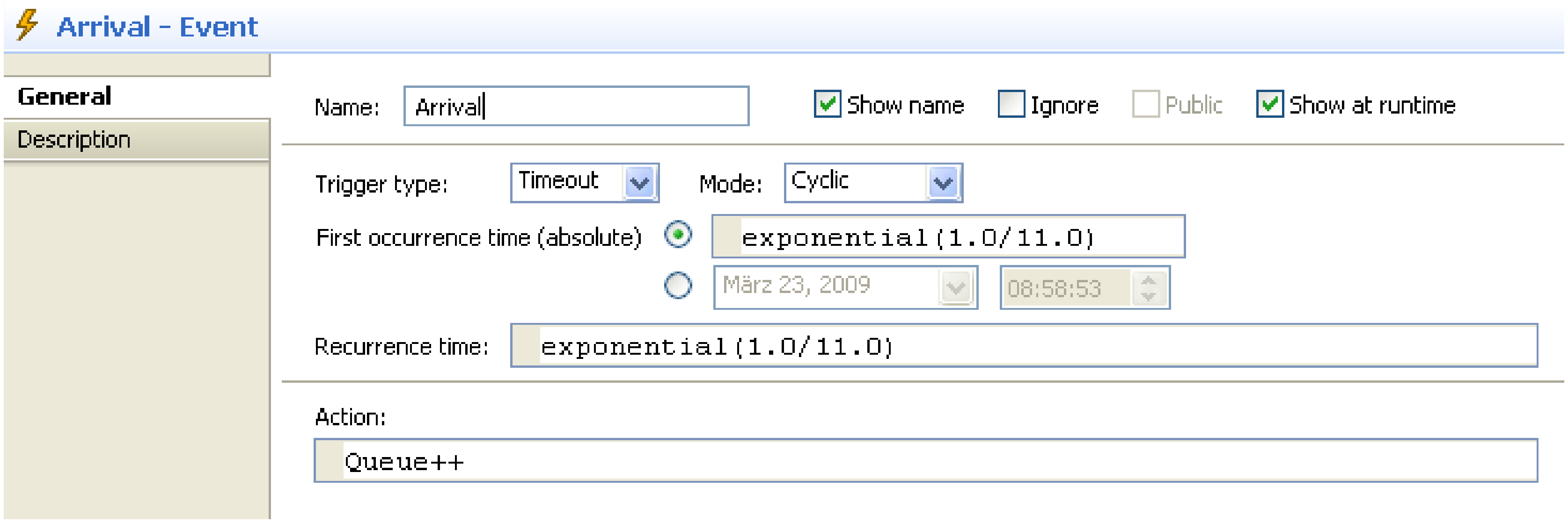
The FEL is thus usually stored as a linked list

The simulator spends a lot of time processing the FEL

- Efficiency is thus very important!

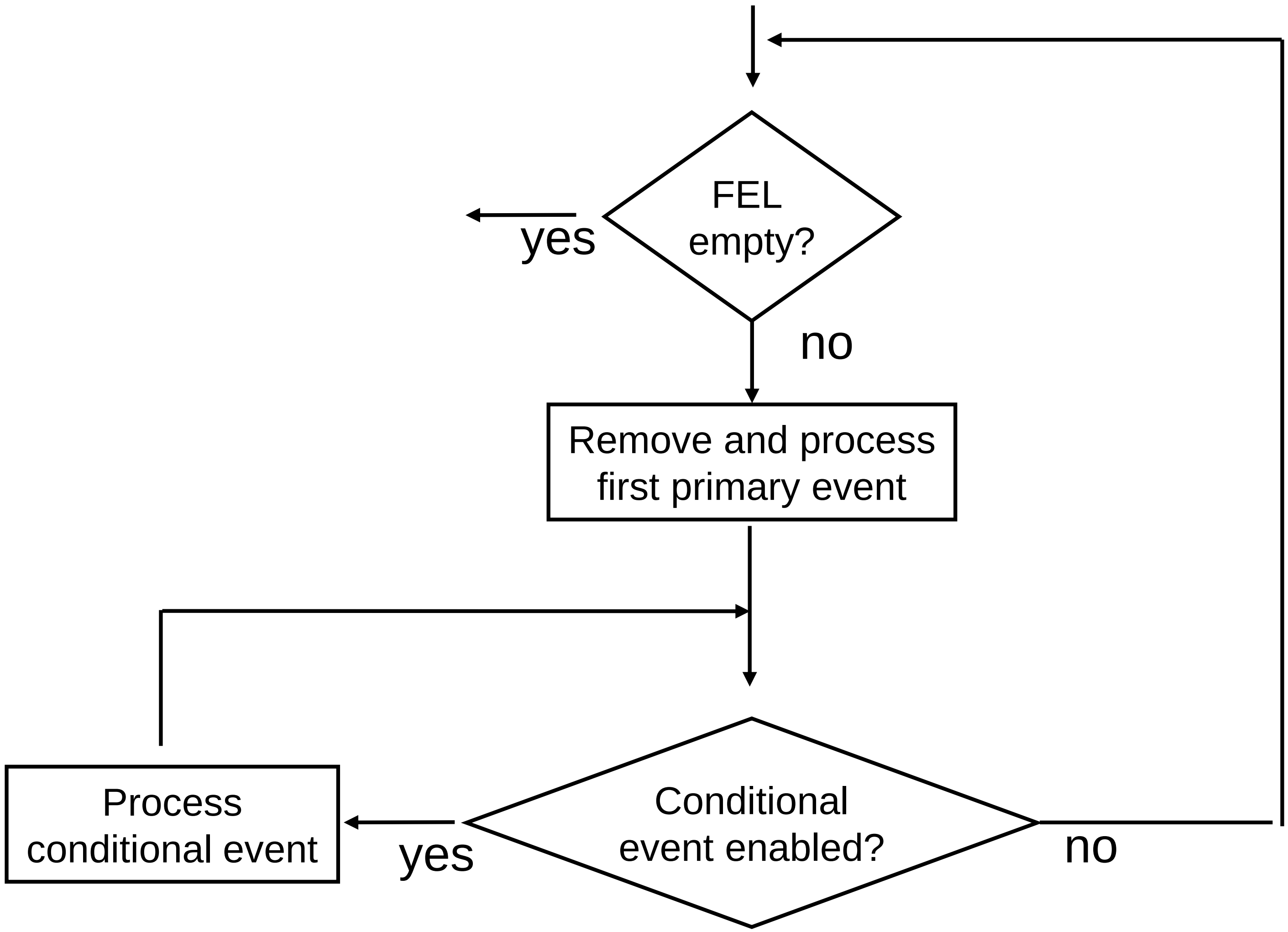
Event List

Example: Event "Customer arrives" in the bank model:



Executing this event at time T inserts the event
($T + Exp(1/11)$, Customer arrives) into the FEL

Simulation Algorithm



Simulation Algorithm

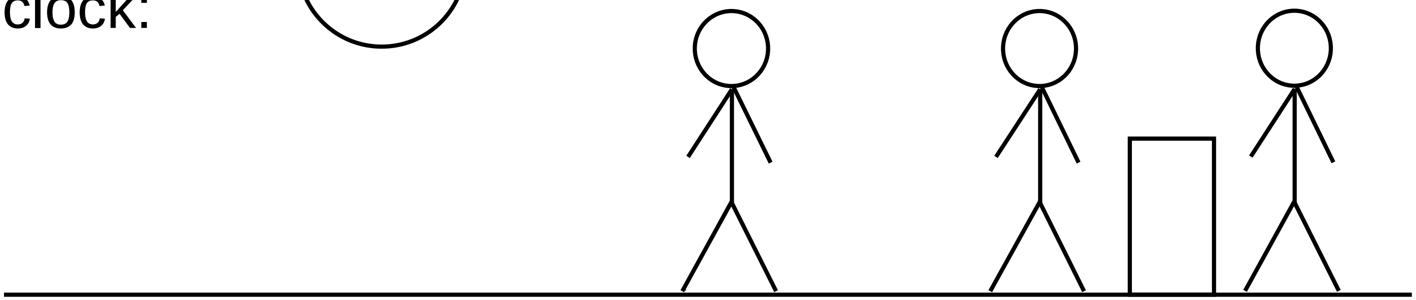
Remove and process first primary event:

- Remove first primary event (t, E) from FEL
- Advance simulation time to t
- Update state variables according to E
- Insert new events into FEL according to E
- Compute statistics

Every discrete–event simulator works like this
(even if the programming model looks different!)

Simulation Algorithm

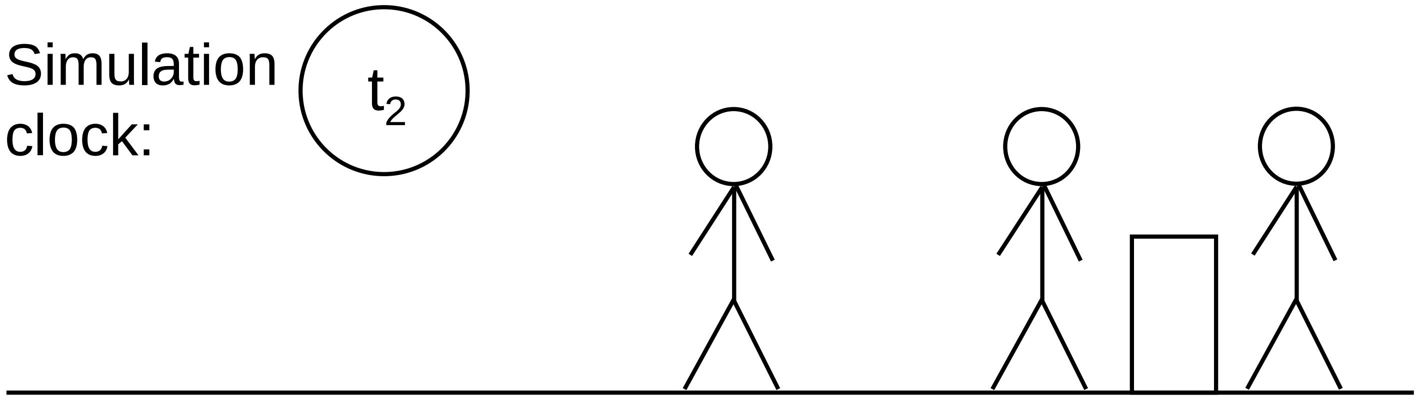
Simulation clock: t_2



(t_4, Arrival) → (t_3, Service complete)

State & State Variables

The system state for the bank simulation at time = t_2



- Time = t_2
- # Customers in queue = 1
- # Customers in service = 1
- Time until next arrival = t_4
- Time until service completion = t_3

Timing

Sometimes, activities last for an exact amount of time:

- The clock cycle in a computer
- Each phase of a traffic light
- A certain operation in a manufacturing line

Such exact times are called deterministic

These are easy to deal with in a simulation

Timing

Usually, activities last for random amounts of time:

- Inter–arrival times at bank
- Service times at bank
- Time to failure for a machine
- Time that a user program runs

Such times are called random or stochastic

Random Variables

If any quantity in a simulation is random, then ...

- ... the simulation result must also be random

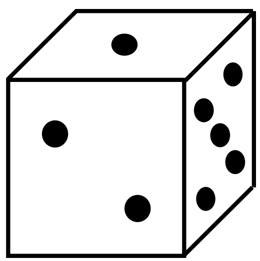
This can make things complicated:

- The simulator will need to use random variables
- We will need to calculate some statistics

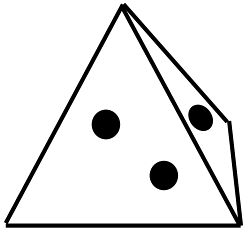
This will be the subject of later lectures

Random Variables

The best-known random variable:



Discrete and uniformly distributed on [1,6]



Discrete and uniformly distributed on [1,4]

Simulating the Bank by Hand

Simulation clock: 15

Arrival interval	Customer arrives	Begin service	Service duration	Service complete
5	5	5	2	7
1	6	7	4	11
3	9	11	3	14
3	12	14	1	15

Computing Statistics

Average waiting time for a customer: (0+1+2+2)/4=1.25

Arrival interval	Customer arrives		Begin service	Service duration	Service complete
5	5	←0→	5	2	7
1	6	←1→	7	4	11
3	9	←2→	11	3	14
3	12	←2→	14	1	15

Computing Statistics

P(customer has to wait): =3/4 =0.75

Arrival interval	Customer arrives		Begin service	Service duration	Service complete
5	5		5	2	7
1	6	←W→	7	4	11
3	9	←W→	11	3	14
3	12	←W→	14	1	15

Computing Statistics

P(Server busy): 10/15=0.66

Arrival interval	Customer arrives	Begin service	Service duration	Service complete
5	5	5	2	7
1	6	7	4	11
3	9	11	3	14
3	12	14	1	15

Computing Statistics

Average queue length: =(1*1+2*1+2*1)/15=0.33

Arrival interval	Customer arrives	Begin service	Service duration	Service complete
5	5	0→ 5	2	7
1	0→ 6	1→ 7	4	11
3	0→ 9	1→ 11	3	14
3	0→ 12	1→ 14	1	0→ 15

Learning Goals

Questions from this lecture:

- What are the two types of abstraction? Give an example of each.
- What are system and system environment?
- Explain the following terms: entity, delay, activity, event, attribute and give examples of each
- What is meant by the state of a system?
- What is the difference between validation and verification?
- How are primary and conditional events defined?
- How does a simulator use the future event list?
- Describe the basic discrete event simulation algoirthm