Università degli Studi di Messina
DIPARTIMENTO DI INGEGNERIA

Dependable Computing Modeling and Simulation

Discrete event simulation

Master degree in

Engineering in Computer Science

Simulation

- Software program emulating how a system works
  - Existing systems
  - Systems in designing phase
- System behavior is described by its **state**

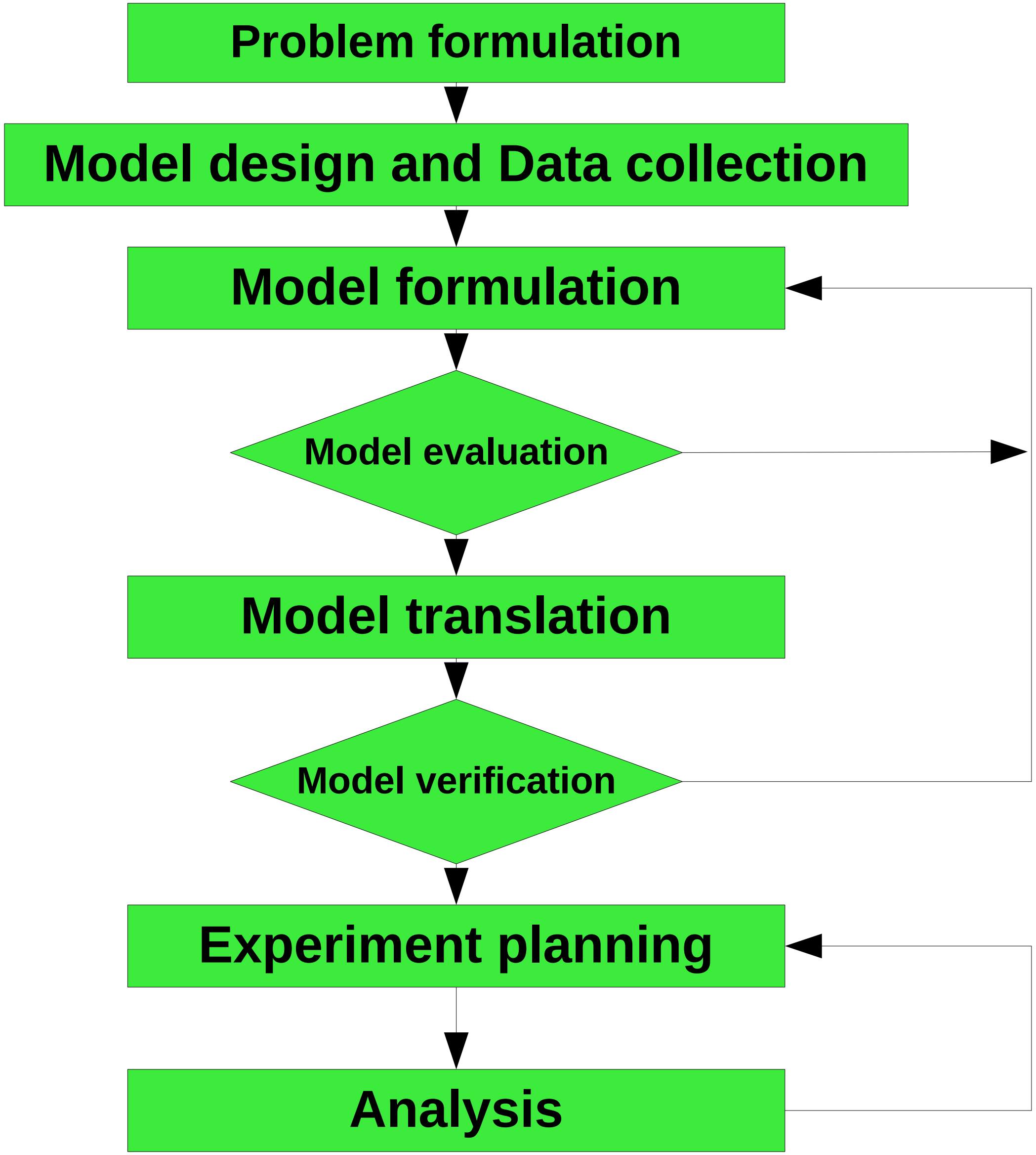
Applications

- System components interactions
- Evaluation about how the system behavior changes when subject to modifications
- Performances in different functional modes (design phase)
- Analytical models validation

Typical phases

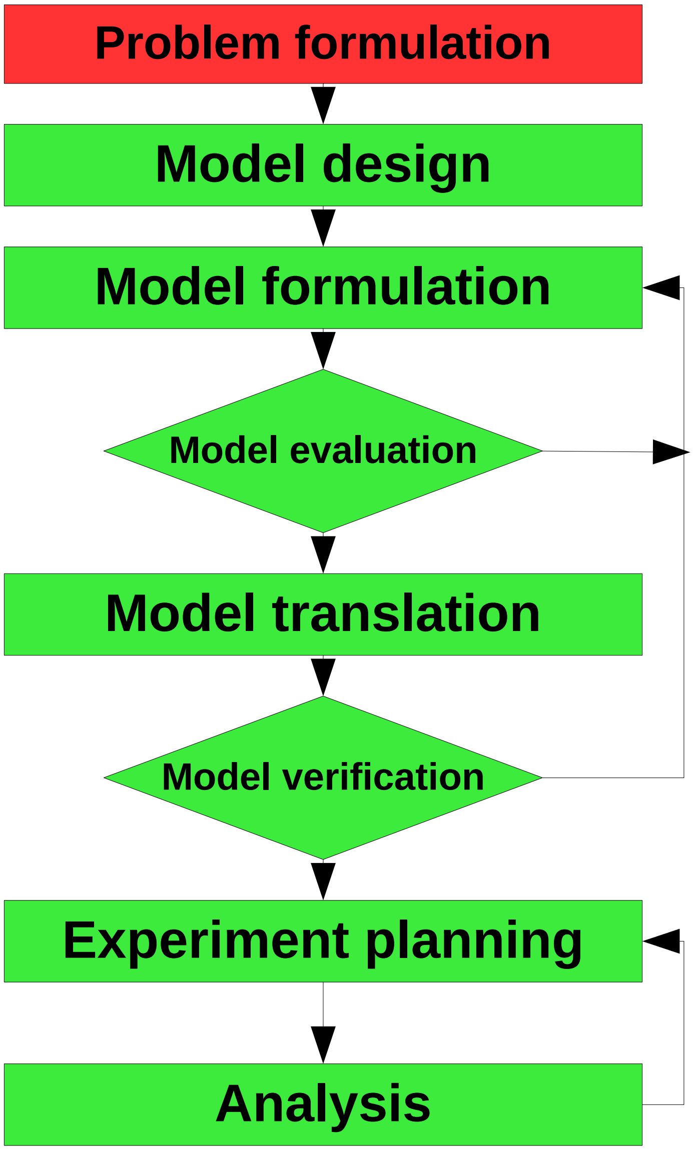
1) Problem formulation
2) Data collection
3) Model formulation
4) Model evaluation
5) Model translation
6) Model verification
7) Experiments planning
8) Analysis

Phases diagram



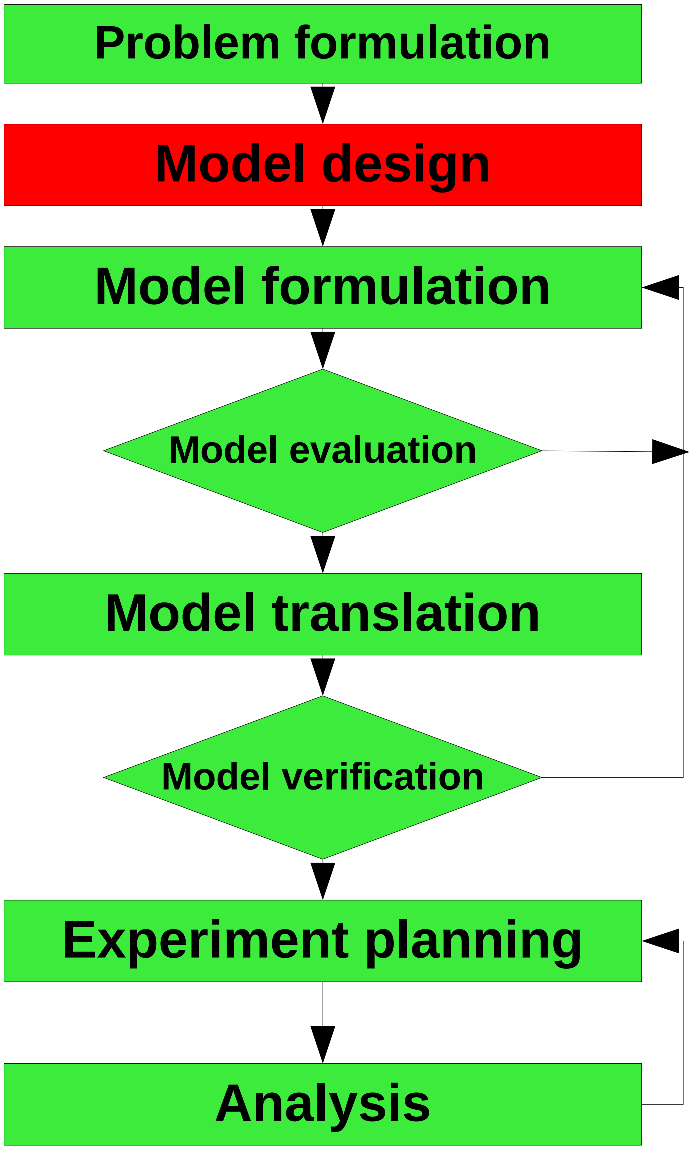
Problem formulation

- Goals
  - performances
  - input/output relations
  - optimization
  - comparison
- Measures of interest
- Input parameters



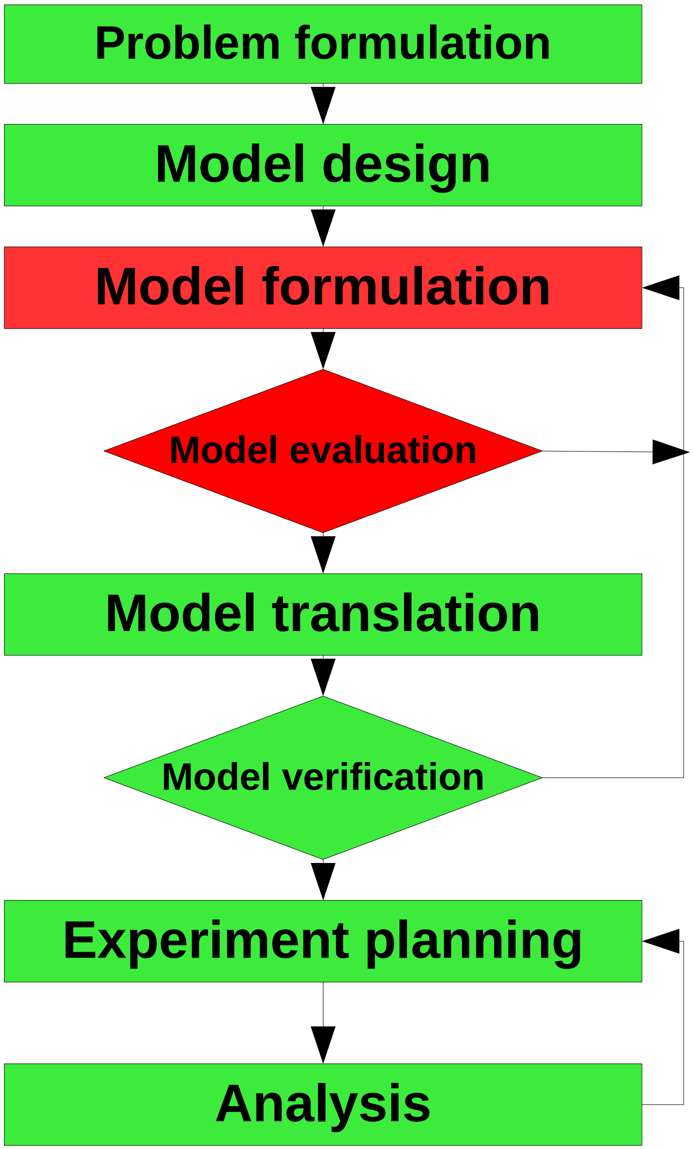
Model design and data collection

- Choice on what to represent
- Inputs
  - Service times
  - Inter-arrival times
- Experimental data
- Realistic hypothesis



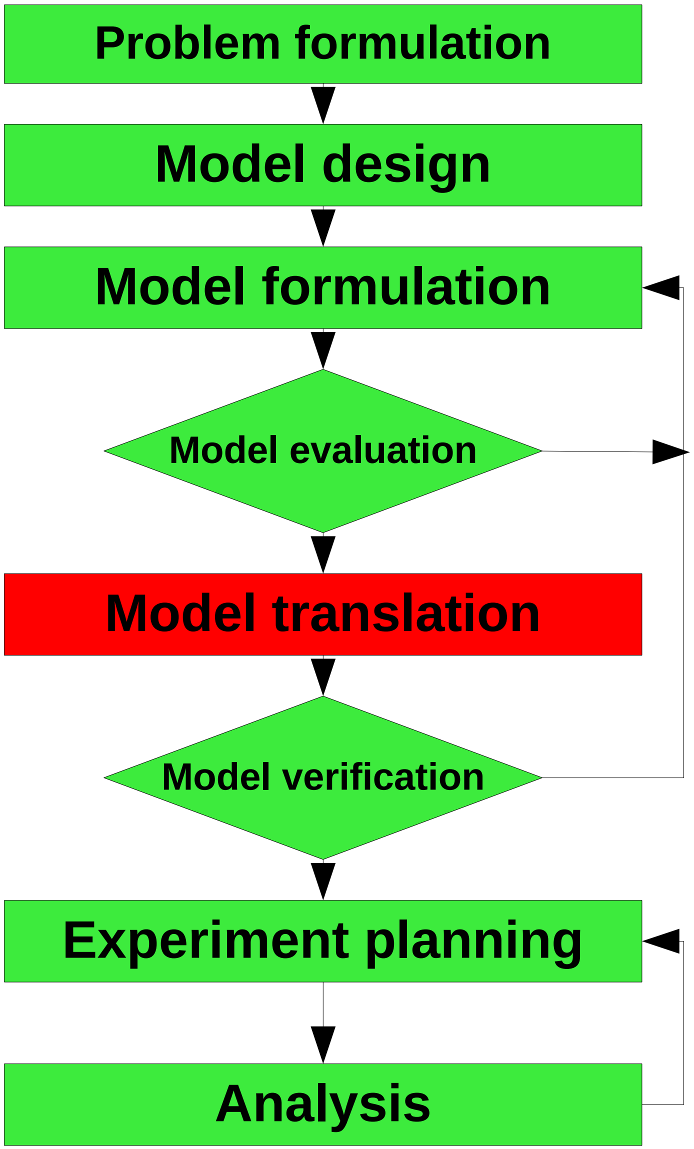
Model formulation and evaluation

- Model simplification
- Refinement
- Abstraction of main characteristics



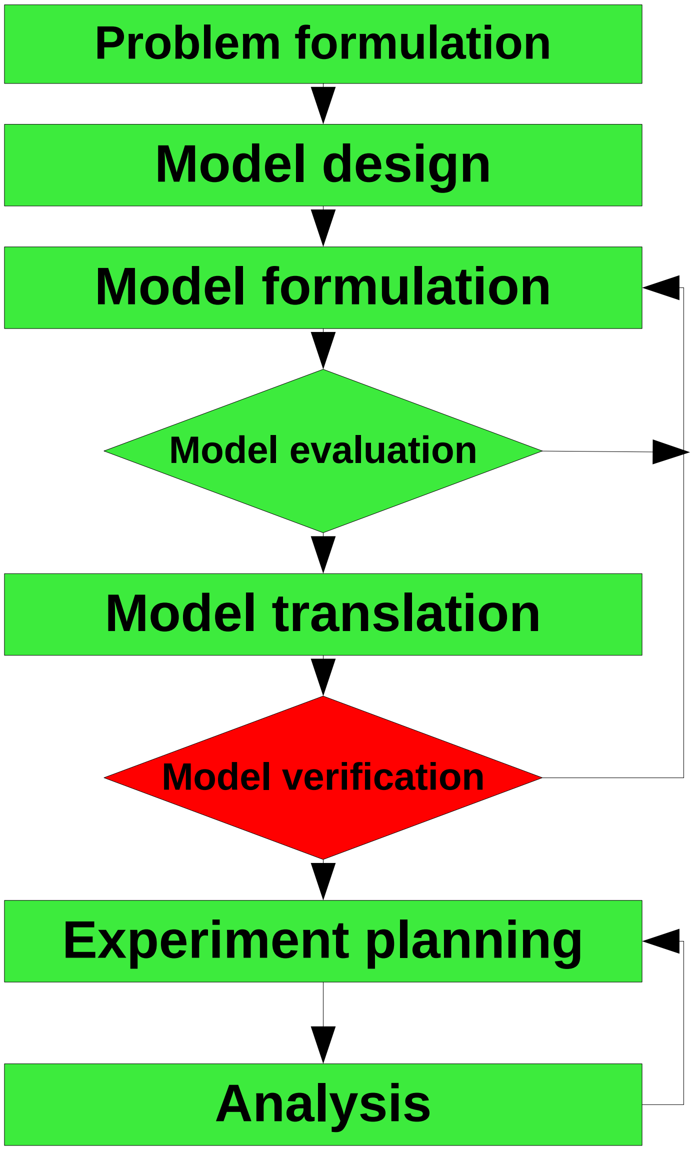
Model translation

- Programming language
  - General purpose (SIMSCRIPT, MODSIM, …)
  - Simulation oriented



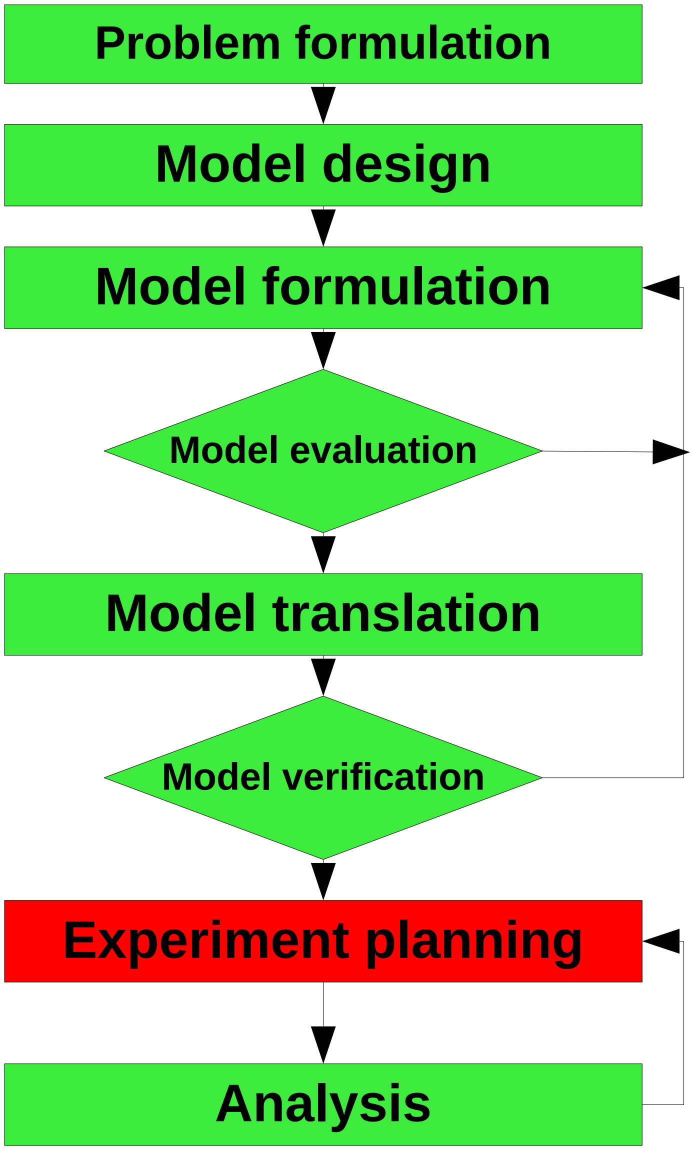
Model verification

- Correctness
  - Logic structure
  - Input and output interface modules



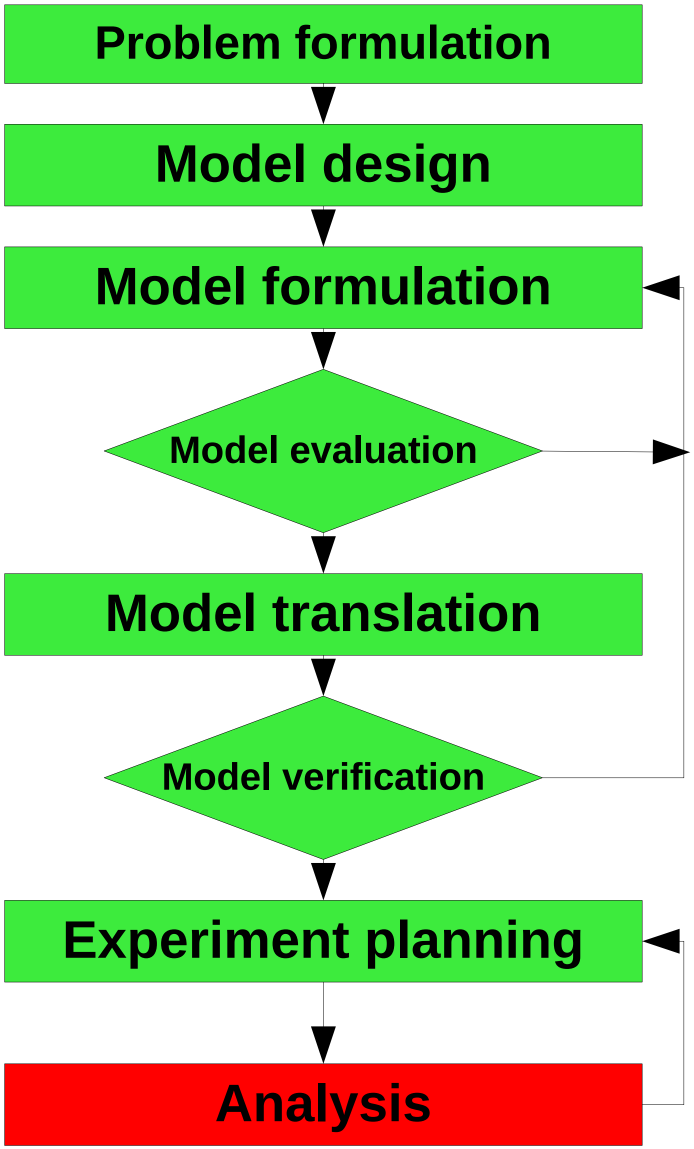
Experiment planning

- In this phase should be choosen
  - Initialization period
  - Simulation time
  - Number of simulations



Result Analysis

- Results are collected
- Goodness of results
  - More simulations can be needed



Some definitions

- *System* : set of entities interacting to achieve a goal
- *Model*: abstract representation of logical system connections; it describes the system through state, activities and their attributes

Some definitions (contd.)

- *System state*: the set of variables that identify the system at any time instant
- *Entity*: each system component we want to represent
- *Attributes*: characteristic of an entity
- *Events*: a fact able to change system state
  - conditioned (or dependent) event
  - primary (or independent) event

Some definitions (contd.)

- *Event calendar*: list of the happening events, ordered over the time
- *Activity*: a well identified time period
- *Delay*: an undefined time period (the length is unknown)
- *Clock*: a variable accounting for the ***simulated*** time

Events

- System state changes in a discrete way
- Type of events must be identified
- *Longitudinal* analysis
  - entities evolve in parallel
  - different objects in movement

Events

- Given an event, it influences the system state and attributes of entities
- Based on the attribute values at a given simulated time, future events are identified

Discrete event simulation

- A sequence of *system snapshots* over the time (simulated)
- A snapshot is constituted by:
  - System state
  - Active activities (a list)
  - Entity states
  - Cumulative statistics and counters
- Final state

Events

- When an event happens
  - The clock advances
  - Future events list is modified (at least one event is removed)
- System state updating
  - Data for statistics
  - Entity set
  - System state

Notation

- τ: simulation time
- x_i: system state at i-th step
- e: generic event
- $\delta(x_i,e)$: transition function state

Initialization

- Future event calendar is set with the happening events at time 0 (primary events)
- Implementation
  - Chained list ordered according to the scheduling time (the time when an event happens)

Future event list

- It is an important data structure
  - It is always accessed by read and insert operations
  - efficiency
- The head is removed (next event)
- Sets of event are modified

Simulation running

Time instant τ_k, states x_k

1) Get (e_k, τ_k) by the future event list
2) $\tau = \tau + \tau_k$ (time advances in a discrete way)
3) $x_{k+1} = \delta(x_k, e_k)$
   - Update statistics
   - impossible events in x_{k+1} are removed
   - New possible events, due to the happening of e_k, are added to the future events list
4) Sort the list

An example: a queuing system

- Costumers arrive at random inter-arrival time (Poissonian process)
- If the server is busy, costumer will wait into the queue
- When the service completes, costumer under service leaves the system
- The server can be into one of two states: *busy* and *free*

Basic elements

- *System state*: number of costumers in the queue, server state
- *Entities*: server, costumers, queue
  - Attributes?
- *Events*: costumer arrival, costumer departure, service starting (dependent event), simulation end
- *Activities*: inter-arrival and service times
- *Delays*: costumer waiting times

Event: *costumer arrival*

- Actions
  - next costumer arrival is planned (primary event)
  - free server
    - Server state changes to busy and the end of service event is planned
  - busy server
    - number of queued client increases

Event: *costumer departure*

- Actions
  - the server becomes free
  - if the queue is not empty
    - the number of queued costumers decreases
    - server becomes busy
    - a departure is planned

Event: *simulation end*

- Action
  - none (system state remains the same)

- Simulation ends and all the statistics are computed

Simulation run

- Simulation time?
  - Time to serve the first five costumers
    - Simulation end event
  - Unknown
- Measures
  - Expected waiting time
  - Server utilization
  - Expected queue length

Expected waiting time

- Each client has a different waiting time (θ_i) in the queue
- N costumers
- A possible estimation is

$$\hat{\theta} = \frac{1}{N} \sum_{k=1}^{N} \theta_k$$

Utilization

- $T_N = \mathcal{T}_f - \mathcal{T}_0$
- *T(i)*: overall time during which *i* costumers are inside the system

$$\hat{v} = \frac{1}{T_N} \sum_{i=1}^{\infty} T(i) = 1 - \frac{T(0)}{T_N}$$

Expected queue length

- $p_N(i)$: probability that the queue length is i

$$\bar{x} = \sum_{i=1}^{\infty} i\, p_N(i)$$

$$\hat{x} = \sum_{i=1}^{\infty} i\, \hat{p}_N(i) = \frac{1}{T_N} \sum_{i=0}^{\infty} i\, T(i)$$

Involved variables

- Computation of expected waiting time:

  $\theta_i, \quad 1 \leq i \leq N$ => list of floats/doubles

- Computation of utilization:

  $T(0)$ => a float/double

- Computation of expected queue length:

  $T(i), \quad 1 \leq i \leq N$ => list of floats/doubles

  – Optimization by using an accumulator

- Number of customers => an integer N

Events: occurrences

Costumer	Inter-arrival times	Arrival times
1		0
2	2	2
3	4	6
4	1	7
5	1	8

Costumers	Service time
1	3
2	2
3	3
4	4
5	2

Calendar

Simulated time	Event type	Client number
0	arrival	1
2	arrival	2
3	depature	1
5	depature	2
6	arrival	3
7	arrival	4
8	arrival	5
9	depature	3
13	depature	4
15	depature	5

Simulation run

- $\mathcal{T}$=0: simulation starts
- Server state: busy
- Queue length: 1
- Future events
  - $\mathcal{T}$=2 : arrival (2)
  - $\mathcal{T}$=3 : departure (1)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 0

Simulation run

- $\mathcal{T}$=2: costumer 2 arrives
- Server state: busy
- Queue length: 2
- Future events
  - $\mathcal{T}$=3 : departure (1)
  - $\mathcal{T}$=6 : arrival (3)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 0
  - T(1) = 2

Simulation run

- $\mathcal{T}$=3: client 1 departs
- Server state: busy
- Queue length: 1
- Future events
  - $\mathcal{T}$=5 : departure (2)
  - $\mathcal{T}$=6 : arrival (3)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 1
  - θ_1 = 3
  - T(1) = 2
  - T(2) = 1

Simulation run

- $\mathcal{T}$=5: client 2 departs
- Server state: free
- Queue length: 0
- Future events
  - $\mathcal{T}$=6 : arrival (3)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 2
  - $\theta_1 = 3$
  - $\theta_2 = 3$
  - T(1) = 4
  - T(2) = 1

Simulation run

- $\mathcal{T}$=6: client 3 arrives
- Server state: busy
- Queue length: 1
- Future events
  - $\mathcal{T}$=7 : arrival (4)
  - $\mathcal{T}$=9 : departure (3)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 2
  - $\theta_1 = 3$
  - $\theta_2 = 3$
  - T(0) = 1
  - T(1) = 4
  - T(2) = 1

Simulation run

- $\mathcal{T}$=7: client 4 arrives
- Server state: busy
- Queue length: 2
- Future events
  - $\mathcal{T}$=8 : arrival (5)
  - $\mathcal{T}$=9 : departure (3)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 2
  - $\theta_1 = 3$
  - $\theta_2 = 3$
  - T(0) = 1
  - T(1) = 5
  - T(2) = 1

Simulation run

- $\mathcal{T}$=8: client 5 arrives
- Server state: busy
- Queue length: 3
- Future events
  - $\mathcal{T}$=9 : departure (3)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 2
  - $\theta_1 = 3$
  - $\theta_2 = 3$
  - T(0) = 1
  - T(1) = 5
  - T(2) = 2

Simulation run

- $\mathcal{T}$=9: client 3 departs
- Server state: busy
- Queue length: 2
- Future events
  - $\mathcal{T}$=13 : departure (4)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 3
  - $\theta_1 = 3$
  - $\theta_2 = 3$
  - $\theta_3 = 3$
  - T(0) = 1
  - T(1) = 5
  - T(2) = 2
  - T(3) = 1

Simulation run

- $\mathcal{T}$=13: client 4 departs
- Server state: busy
- Queue length: 1
- Future events
  - $\mathcal{T}$=15 : departure (5)
  - $\mathcal{T}=\mathcal{T}_f$: simulation end

- Statistics
  - N = 4
  - $\theta_1 = 3$
  - $\theta_2 = 3$
  - $\theta_3 = 3$
  - $\theta_4 = 6$
  - T(0) = 1
  - T(1) = 5
  - T(2) = 6
  - T(3) = 1

Simulation run

- τ=15: client 5 departs
- Server state: free
- Queue length: 0
- Future events

- Statistics
  - N = 5
  - θ_1 = 3
  - θ_2 = 3
  - θ_3 = 3
  - θ_4 = 6
  - θ_5 = 7
  - T(0) = 1
  - T(1) = 7
  - T(2) = 6
  - T(3) = 1

Measures

- Simulation ends at time 15

$$\hat{\theta} = \frac{3+3+3+6+7}{5} = 4{,}4$$

$$\hat{v} = 1 - \frac{T(0)}{15} = 1 - \frac{1}{15} \simeq 0.933$$

$$\hat{x} = \frac{1}{15} \sum_{i=0}^{\infty} i\,T(i) = \frac{(0\times 1)+(1\times 7)+(2\times 6)+(3\times 1)}{15} \simeq 1{,}467$$