INTRODUCTION TO SIMULATION

Pau Fonseca i Casas pau@fib.upc.edu

An ad hoc model classification

$$\text{MODELS}\begin{cases}\text{STATIC}\begin{cases}\text{CONTINUOUS} \Rightarrow \ \text{Min } f(x) \qquad f(x): R^n \to R \\ \qquad\qquad g(x) \le b, g(x): R^n \to R^m, b \in R^m \\ \qquad\qquad\qquad x \in X \subseteq R^n \\ \text{DISCRETE} \qquad x \in Z_+^n \quad (\text{or } x \in \{0,1\}^n)\end{cases} \\ \text{DYNAMIC}\begin{cases}\text{CONTINUOUS} \to \begin{cases}\text{Differential Equations} \\ \text{Difference Equations}\end{cases} \\ \text{DISCRETE} \to \mathit{DESS}\end{cases}\end{cases}$$

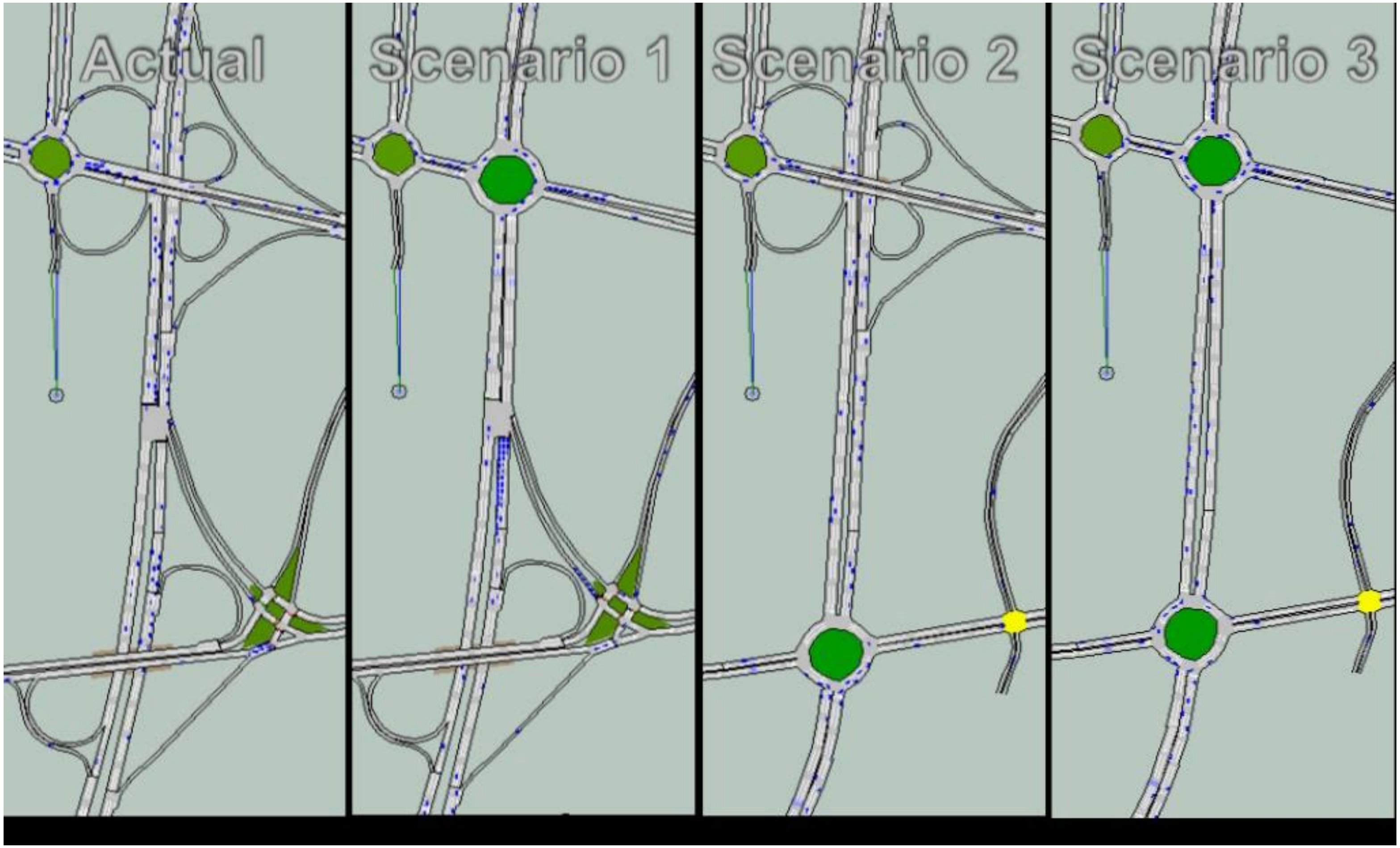
Main concepts

- ENTITY→ An object (component) of interest in a system
- ATTRIBUTE → A property (or characteristic) of an entity
- ACTIVITY → Any process that causes changes in a system
  - ENDOGENOUS ACTIVITIES → Occur within the system
  - EXOGENOUS ACTIVITIES → Occur outside the system, in the System Environment, and affect the system
- STATE OF THE SYSTEM → Description of all the entities, attributes and activities as they exist at one point in time
- SYSTEM PROGRESS → Determined by the changes in the state of the system over time

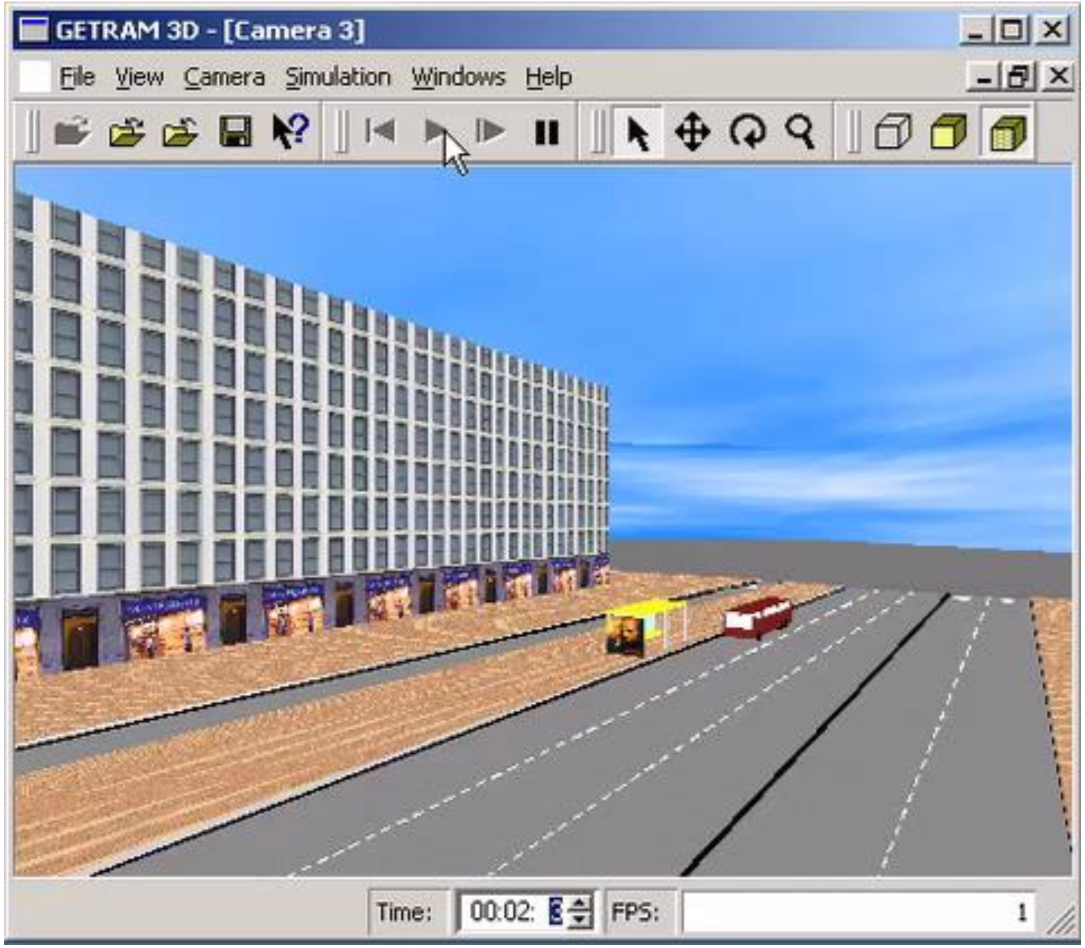
CONTINUOUS vs DISCRETE SYSTEMS

- CONTINUOUS SYSTEMS $\rightarrow$ Changes in system's state are smooth
  - A description of a continuous system will be in the form of continuous equations showing how the system attributes change with time
  - For numerical convenience quite often time is discretized (e.g. economic systems)
- DISCRETE SYSTEMS $\rightarrow$ Changes in system's state are discontinuous
  - A description of a discrete system is concerned with the events producing changes in the state of the system

COMPARING ALTERNATIVE SCENARIOS

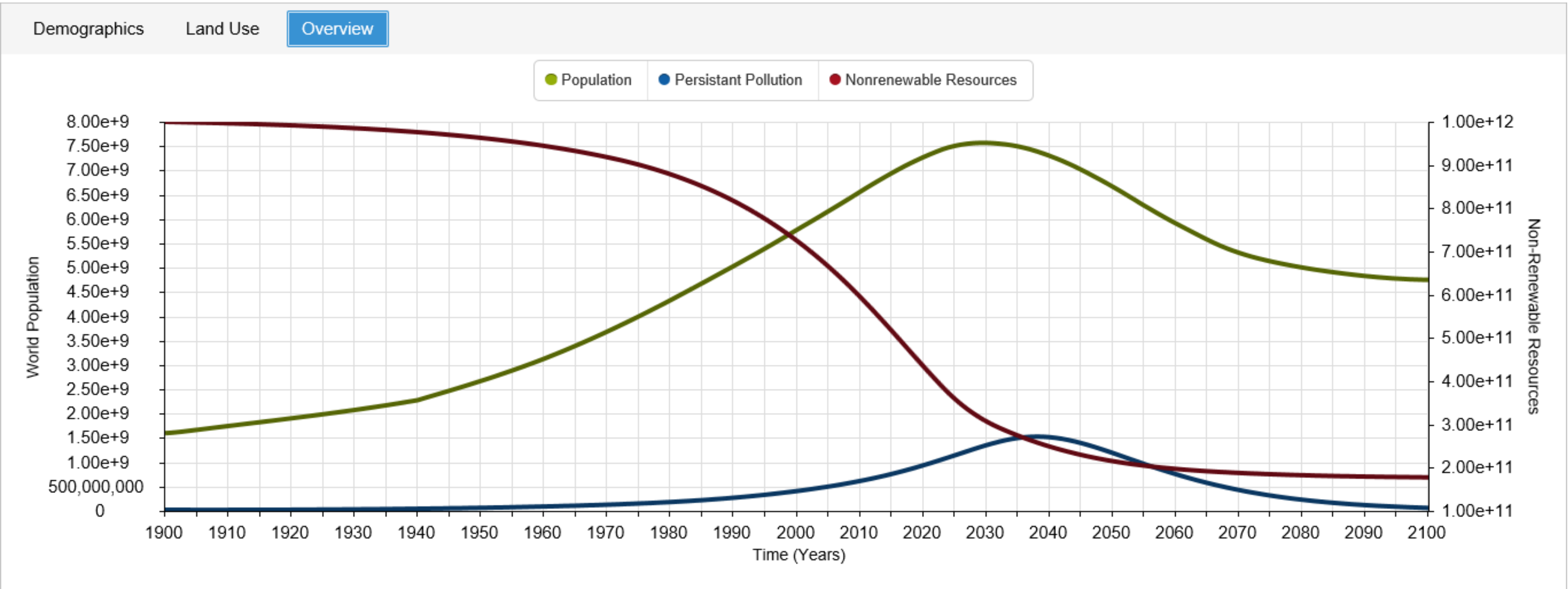


TRAMWAY PRIORITY IN PARIS



Continuous systems

- World3
- https://insightmaker.com/insight/17417/Clone-of-The-World3-Model-A-Detailed-World-Forecaster



Methodological aspects of modeling

- **Formal representations** in terms of logical and/or quantitative relationships imply
  - Establish formal relationship between entity attributes and logical and mathematical variables
  - Formulate **modeling hypothesis** about how the system works (system behavior):
    - How the attributes (variables) change with time as a consequence of the activities
  - **Translate** the modeling hypothesis in terms of **quantitative** (e.g. equations) and/or **logical** (e.g. rules) relationships between variables (attributes)

Methodological aspects of modeling

- The methodological approach is based on the relationship SYSTEM ↔ MODEL
- If the systems of interest are Dynamic Systems evolving with time:
  - System's time evolution ↔ state changes over time
  - System state at time t: determined by the values of the state variables at time t
  - A state variable is one of the set of variables that are used to describe the "state" of a dynamical system. The state of a system describes enough about the system to determine its future behavior.

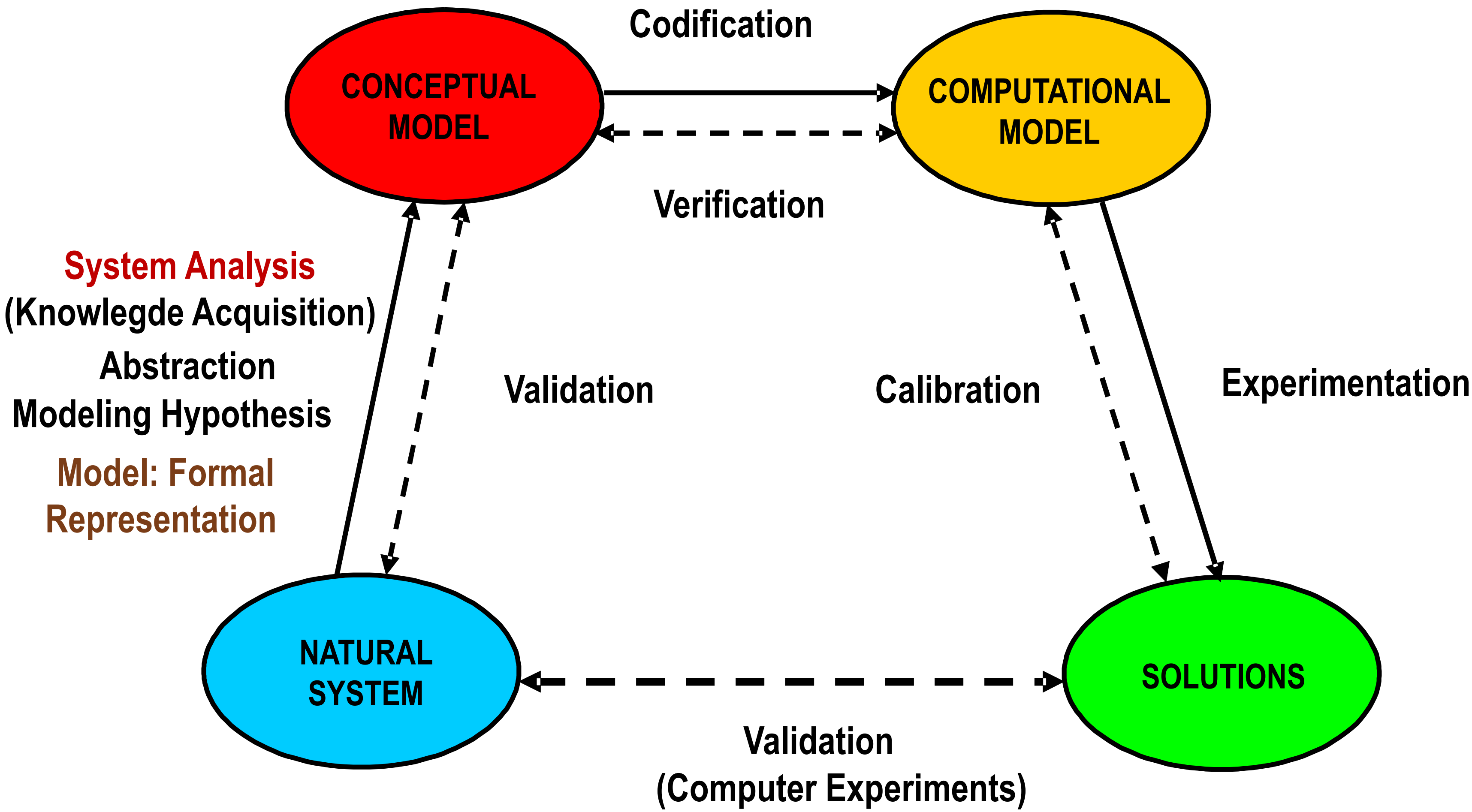
Methodological aspects of modeling

- Objective: describe and analyze the time evolution of the system in terms of the time evolution of its representation by a model.
- Description of the system's state time evolution
  - Description of time changes of the state variables as a function of the activities.
  - Description of model's state changes over time.
- SIMULATION: A numerical technique to track the time changes of the states of a system by means of tracking (usually in a computer) the time changes of the corresponding state variables of a valid mathematical model representing it.

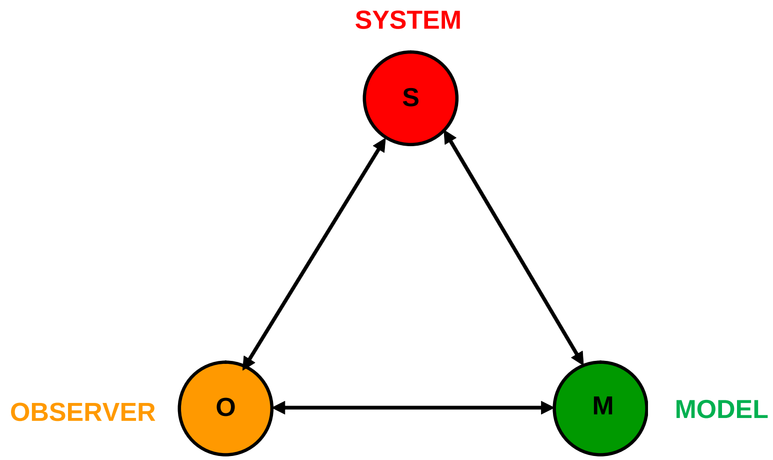
SIMULATION steps

- SYSTEM'S ANALYSIS
  - Identify system entities
  - Identify relevant attributes in terms of model's objectives
  - Knowledge acquisition: learn how the system works
  - Collect data and empirical evidence
- MODEL BUILDING (Conceptual)
  - Formulate modeling hypothesis after the knowledge acquired
  - Translate attributes in terms of variables
  - Formalize modeling hypothesis in terms of equations and logical relationships
- VALIDITY CHECKING: Verify that the model built corresponds to the modeled system and responds to study objectives
- ALGORITHMIC DEVELOPMENT (Computational model)
- MODEL VALIDATION
- PUTTING MODEL SOLUTIONS TO WORK IN PRACTICE

The model building process



Systems, observers, models



- An object M is a model of a system S if it can provide valid answers to the questions of an observer O on the system S (Minsky)

SIMULATION

- **TECHNIQUE** FOR **IMITATING** IN A COMPUTER THE OPERATIONS OF REAL WORLD **SYSTEMS** [AS THEY **EVOLVE WITH TIME**], THROUGH **MODELS** REPRESENTING THEM IN A REALISTIC WAY

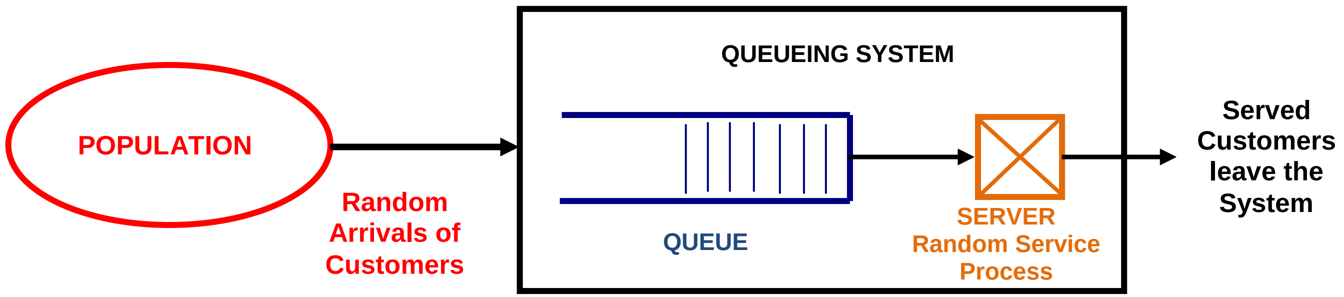
TWO MAIN CONCEPTS

- System
  - A collection of entities, characterized by attributes, that act and interact together toward the accomplishment of some logical end
- Model
  - **Formal** representation of a system

Event Scheduling approach

System characterization

- Example: Queueing Systems
- System Components:
  - Customer's Population: Size, arrival process
  - Queue: length, operational rules (FIFO, LIFO, random, priorities….)
  - Service System: Number of servers, service process

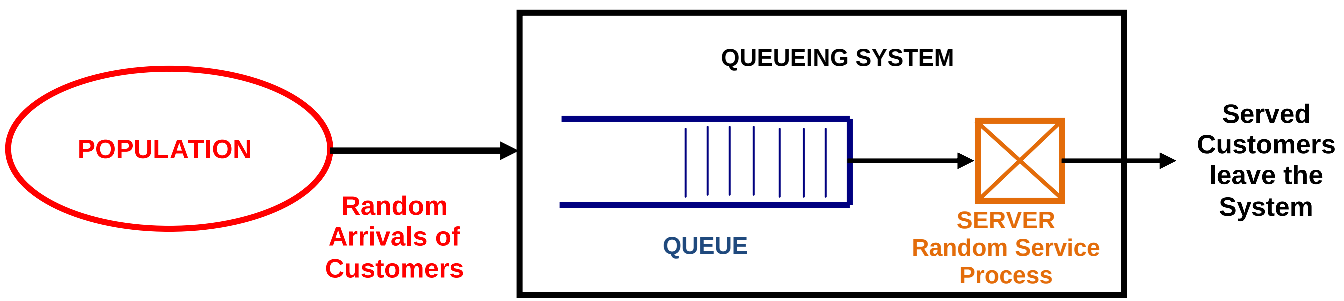


System's evolution over time

- The System is in state s at time t
- The System's State: is defined by the values of the state variables at time t
- The Modeler defines the mapping:
  - entities' attributes ⇔ variables
- The Modeler defines the subset of relevant variables which define the system's state.
- Tracking the system's evolution over time
  - Tracking the State changes over time.
  - How state variables change their values over time.

QUEUING SYSTEMS:
states and state evolution over time

- System's State at time t: defined by the value of the state variable N(t) = number of clients in the system at time t
- State changes: consequence of the occurrence of two independent random events:
  - Arrivals of new customers at time t
  - Departures of served customers at time t



Event Scheduling approach to system simulation

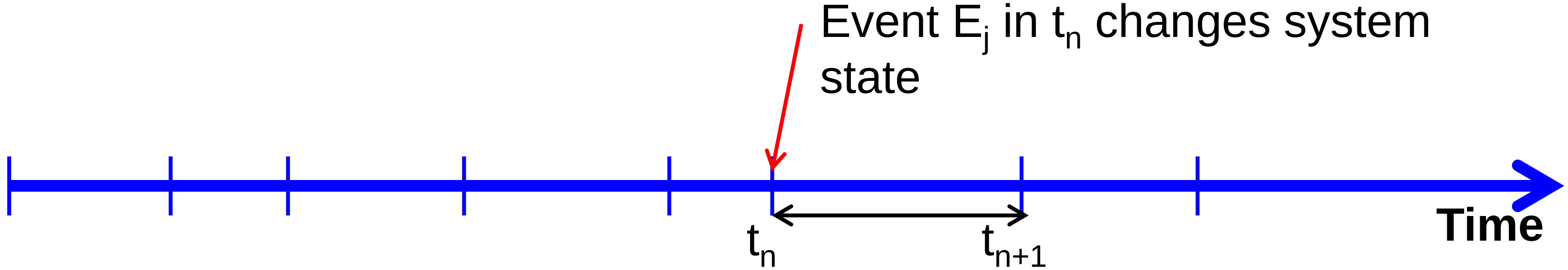
- Systems are modeled in terms of their **states** at each point in time based on:
  - Entities and attributes, activities and **events**
  - Entities that:
    - Pass through the system
    - Represent system resources
  - Activities and events that cause system state to change
- Discrete event models are appropriate for those systems for which changes in system state occur only at discrete points in time

Time evolution in simulation

ASYNCHRONOUS IN EVENT SCHEDULING

OCCURRENCE OF EVENTS CHANGE SYSTEM STATE

System in state S_k at time t_n
Event E_j in t_n changes system state

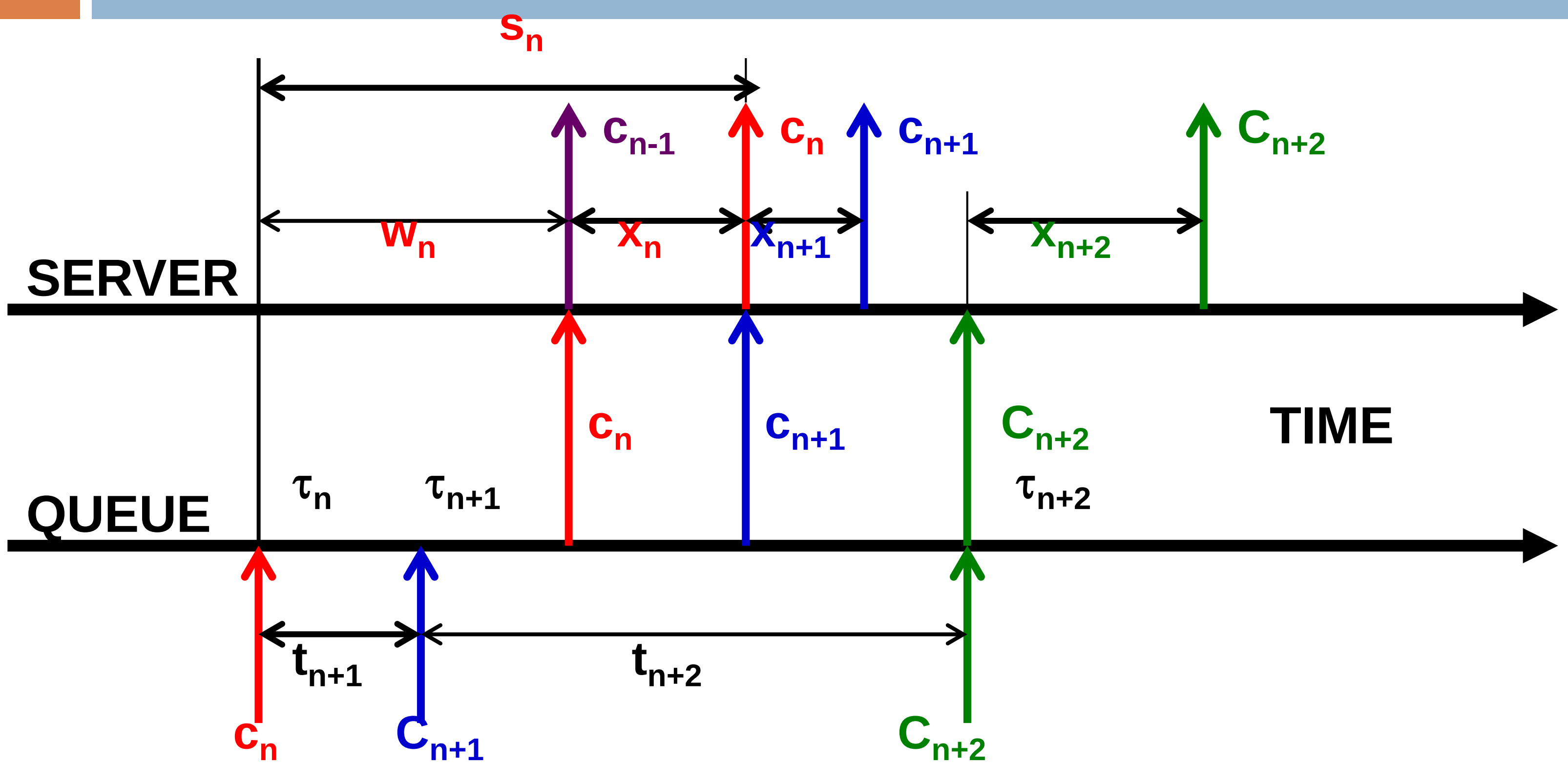


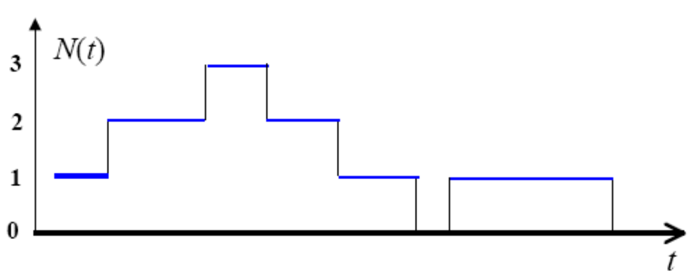
$$\tau_{n+1} = t_{n+1} - t_n$$

Time interval between successive events

- Events occur randomly $\Rightarrow$ τ is a random variable
- Which random variable?
- How to identify the randomness and the type of randomness (probability distribution that "explains" τ)
- How to generate sequences of events according with the type of randomness identified? $\Rightarrow$ Generating samples of random distributions

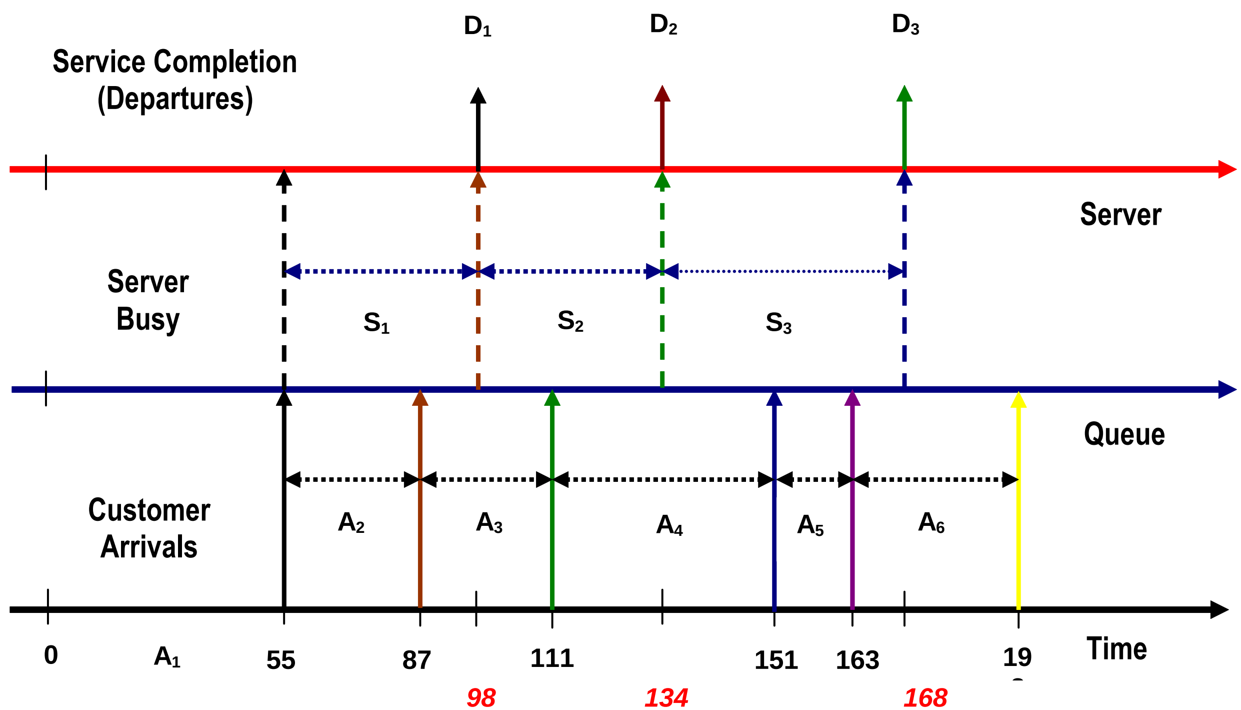
A time diagram for event-scheduling





- C_n n-th client
- τ_n arrival time of client n-th
- $t_{n+1} = \tau_{n+1} - \tau_n$ interarrival time between clients n and n+1
- x_n service time for client n
- w_n waiting time of client in the queue
- s_n total time of client n in the system = $w_n + x_n$

"Event scheduling" approach for a queuing system



Interarrival Times
A_1= 55, A_2= 32, A_3= 24, A_4= 40, A_5 = 12, A_6 = 29, ….

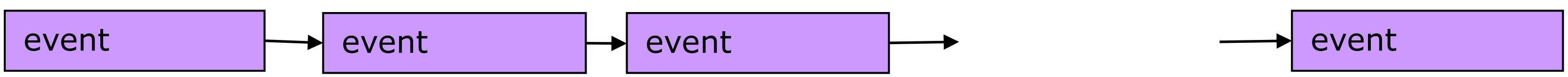
Service Times
S_1 = 43, S_2 = 36, S_3= 34, ….

Where come these numbers from?

Event

- Kind of event
  - Depends on the model definition.
  - Exit event, enter event for a MM1 queue.
  - The Kind of event allows to define the procedure that the simulation engine must run when the event is executed.
- Creation time
  - Shows the time when the event enters in the simulation system.
- Running time
  - Shows when the simulation engine must run the event.
- Priority
  - Priority must be taken in consideration only if two ore more) events have the same *run time*.

Event list

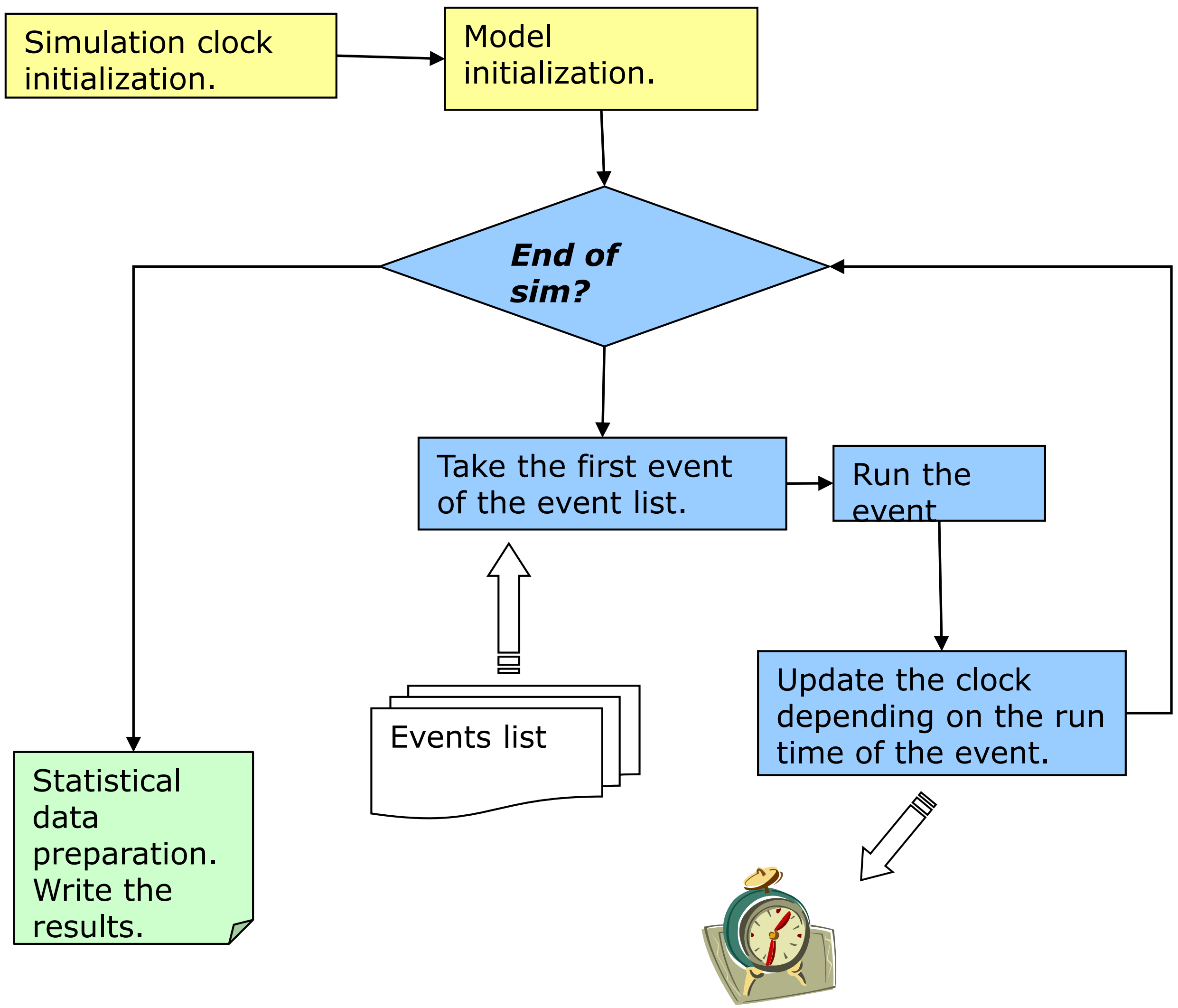


event
<u>struct</u>
{
    Execution time: real
    Priority: enter
    Kind: enter
}
<u>End struct</u>

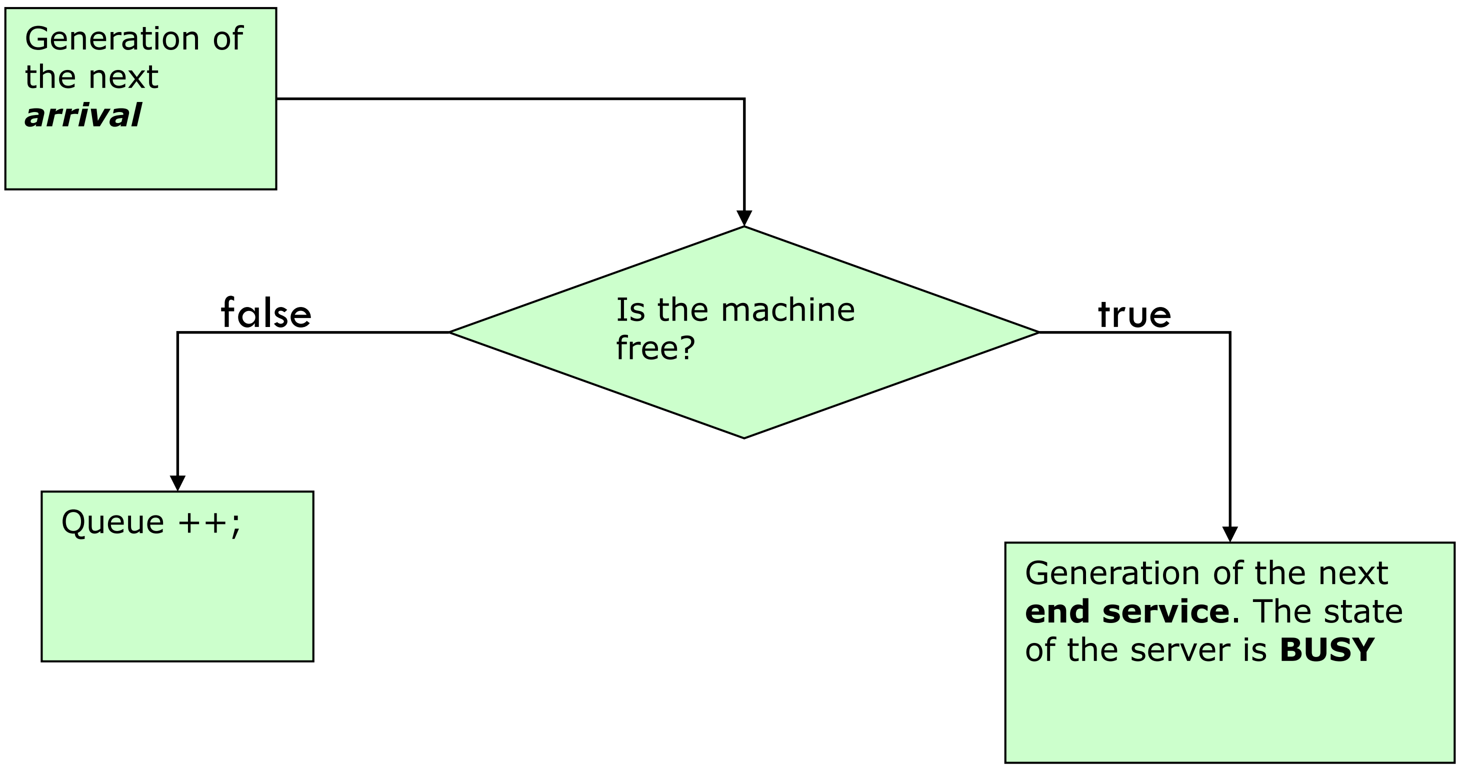
Event scheduling algorithm

Generalizing the algorithm for any system.

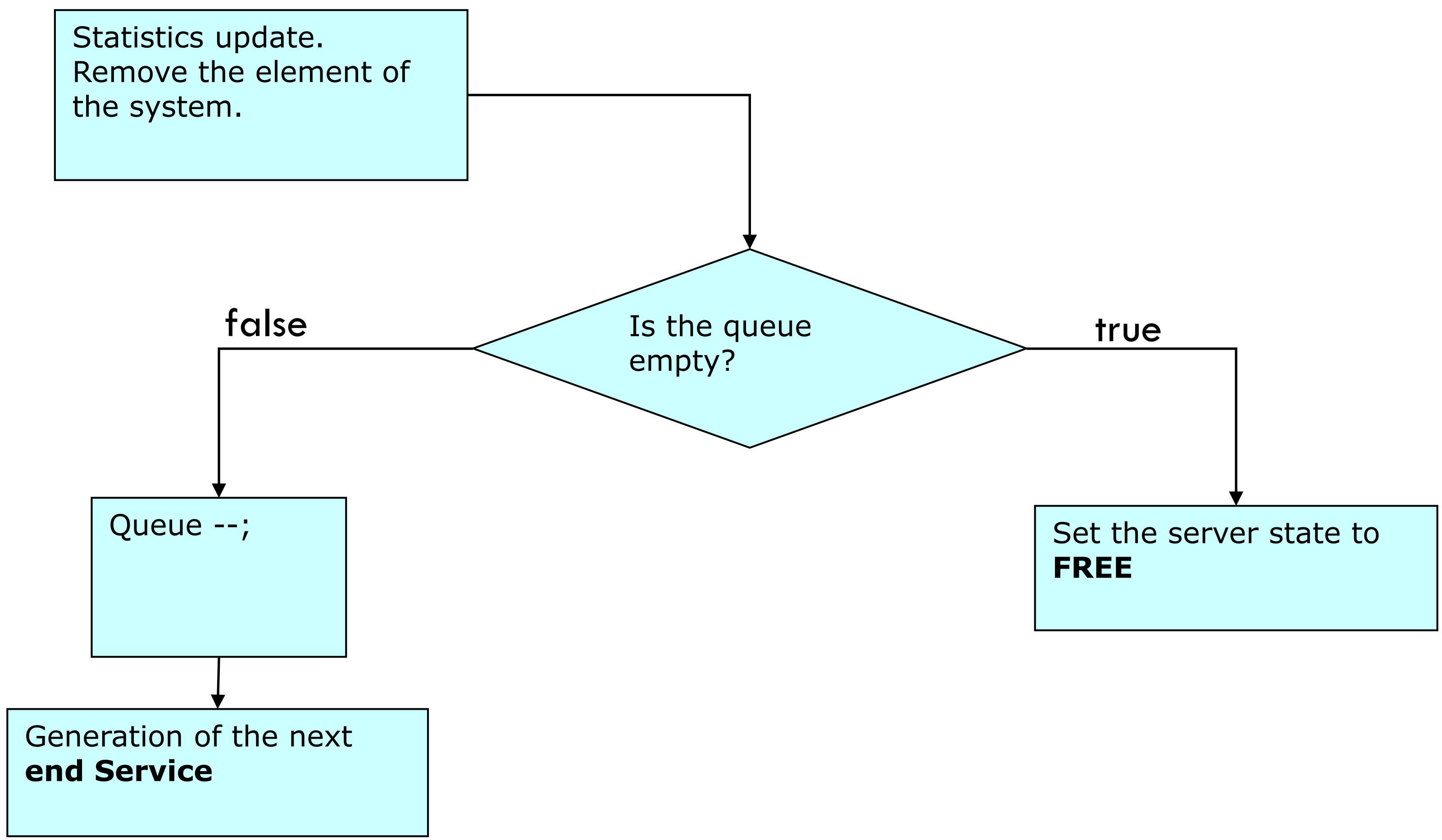
Event Scheduling algorithm



Event Scheduling: Arrival event



Event Scheduling: Exit event



Pseudo code for es simulation of a queue X/Y/s

```
Program X_Y_s;
        {Initialization of state variables};
        {Initialization of the Event List}
        {Select first event from the list};
        Time =Time of the first event;
While Time < Simulation Horizon do
        If  Event = 'Arrival' Then
                {Procedure Arrival Event}
        Otherwise
                {Procedure Departure Event}
        End If;
                {Collect Statistics};
                {Select next Event from Event List};
End While;
        {Print Results}
End Program
```

Procedure Arrival Event

{Procedure Arrival Event}

Time = Time Next Arrival;

If There is any server free Then

Select free server $\rightarrow$s;

Mark server s busy;

{Generate departure (End of Service)}

Otherwise

Add client to the queue

End If;

{Generate Next Arrival}

Procedure Departure Event

{Procedure Departure Event}

Time = Time next Departure;

If Queue Empty Then

Mark server as free;

Otherwise

Select server free → s;

Mark server s as busy;

Decrease queue by one unit;

{Generate departure (End of service)}

End if;

Event Scheduling Example

Trace

ES: Events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0

ES: Events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0
	0	0,1509	1E+12	0	0	0	0

ES: Events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0
	0	0,1509	1E+12	0	0	0	0
1	0,1509	0,5778	0,93940	1	0	1	0

ES: Events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0
	0	0,1509	1E+12	0	0	0	0
1	0,1509	0,5778	0,93940	1	0	1	0
2	0,5778	1,4772	0,93940	1	1	1	0

ES: Events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0
	0	0,1509	1E+12	0	0	0	0
1	0,15099	0,5778	0,9394	1	0	1	0
2	0,57788	1,4772	0,9394	1	1	1	0
3	0,93940	1,4772	3,5225	1	0	0	1

ES: Events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0
	0	0,1509	1E+12	0	0	0	0
1	0,1509	0,5778	0,9394	1	0	1	0
2	0,5778	1,4772	0,9394	1	1	1	0
3	0,9394	1,4772	3,5225	1	0	0	1
4	1,4772	1,5657	3,5225	1	1	1	0

Event scheduling Trace example 1

Using this data

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0

Next arrival	**Next exit**
1,6933	1,8840
4,0012	4,3038
5,2509	5,6282
5,5315	6,5012
5,6327	7,0477
6,0014	

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,693356	1E+12				
1	1,693356	4,001288	1,884081404	1	0	1	0
2	1,884081	4,001288	1E+12	0	0	0	1
3	4,001288	5,250927	4,303805741	1	0	1	0

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1
3	4,0012	5,2509	4,3038	1	0	1	0
4	4,3038	5,2509	1E+12	0	0	0	1

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1
3	4,0012	5,2509	4,3038	1	0	1	0
4	4,3038	5,2509	1E+12	0	0	0	1
5	5,2509	5,5315	5,6282	1	0	1	0

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1
3	4,0012	5,2509	4,3038	1	0	1	0
4	4,3038	5,2509	1E+12	0	0	0	1
5	5,2509	5,5315	5,6282	1	0	1	0
6	5,5315	5,6327	5,6282	1	1	1	0
7	5,6282	5,6327	6,5012	1	0	0	1
8	5,6327	6,0014	6,5012	1	1	1	0
9	6,0014	7,3736	6,5012	1	2	1	0
10	6,5012	7,3736	7,0477	1	1	0	1
11	7,0477			1	0	0	1

Event Scheduling trace exemple 2

We supose this data set (priority for the arrivals)

Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?
0	0	1	-	0	0	0	0

Element	Arrival time	Service time
1	1	1
2	2	2
3	2.5	2
4	3	1
5	6	2

Event Scheduling trace

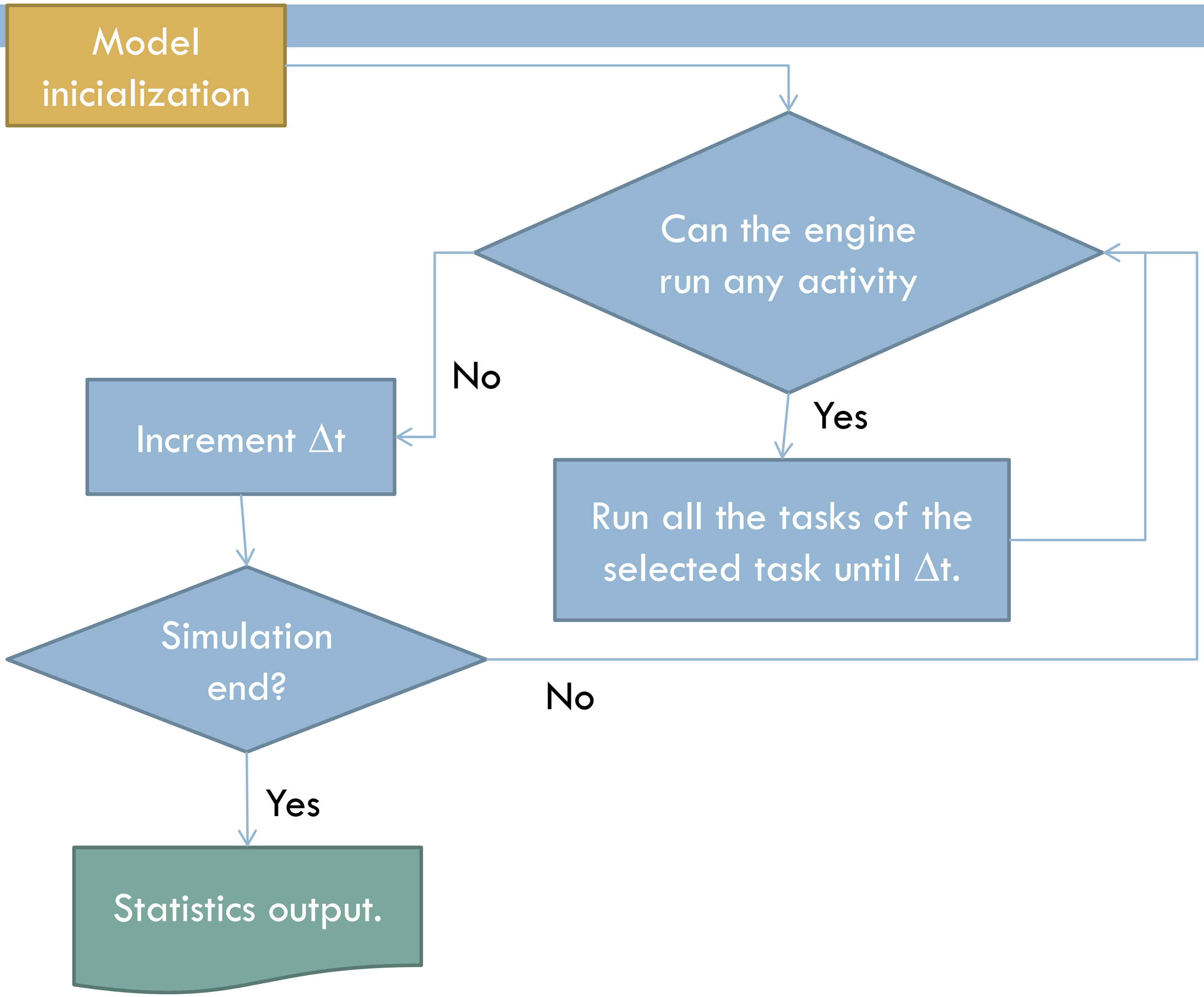
Id	Time	Next Arrival	Next exit	Server state	Queue long	Is Arrival?	Is Exit?
0	0	1	-	0	0	0	0
1	1	2	2	1	0	1	0
2	2	2.5	2	1	1	1	0
3	2	2.5	4	1	0	0	1
4	2.5	3	4	1	1	1	0
5	3	6	4	1	2	1	0
6	4	6	6	1	1	0	1
7	6	-	6	1	2	1	0
8	6	-	7	1	1	0	1
9	7	-	9	1	0	0	1
10	9	-	-	0	0	0	1

Activity Scanning

Activity scanning

1. Analyze if the simulation engine can run some activity, this depends on the conditions of each activity, and run it until Δt.
2. When the simulation engine cannot run more activities increment the clock with Δt.

AS: simulation engine



Activity scanning Example

Trace

Activity scanning events evolution

- Using $\Delta t=1$. run the simulation until time = 6.

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0

Next arrival	Next exit
1,6933	1,8840
4,0012	4,3038
5,2509	5,6282
5,5315	6,5012
5,6327	
6,0014	

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1
4	2		4,0012	1E+12	0	0	0	0

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1
4	2		4,0012	1E+12	0	0	0	0
5	3		4,0012	1E+12	0	0	0	0

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1
4	2		4,0012	1E+12	0	0	0	0
5	3		4,0012	1E+12	0	0	0	0
6	4		4,0012	1E+12	0	0	0	0

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,69335	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1
4	2		4,0012	1E+12	0	0	0	0
5	3		4,0012	1E+12	0	0	0	0
6	4		4,0012	1E+12	0	0	0	0
7	5	4,0012	5,2509	4,3038	1	0	1	0

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1
4	2		4,0012	1E+12	0	0	0	0
5	3		4,0012	1E+12	0	0	0	0
6	4		4,0012	1E+12	0	0	0	0
7	5	4,0012	5,2509	4,3038	1	0	1	0
8	5	4,3038	5,2509	1E+12	0	0	0	1

Activity scanning events evolution

Id	Time	Event Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
1	1		1,6933	1E+12	0	0	0	0
2	2	1,6933	4,0012	1,8840	1	0	1	0
3	2	1,8840	4,0012	1E+12	0	0	0	1
4	2		4,0012	1E+12	0	0	0	0
5	3		4,0012	1E+12	0	0	0	0
6	4		4,0012	1E+12	0	0	0	0
7	5	4,0012	5,2509	4,3038	1	0	1	0
8	5	4,3038	5,2509	1E+12	0	0	0	1
9	6	5,2509	5,5315	5,6282	1	0	1	0
10	6	5,5315	5,6327	5,6282	1	1	1	0
11	6	5,6282	5,6327	6,5012	1	0	0	1
12	6	5,6327	6,0014	6,5012	1	1	1	0

Output comparison

Compare the previous trace wit one obtained using Event Scheduling

Event scheduling events evolution

Using this data

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
0	0	0	0	0	0	0	0

Next arrival	Next exit
1,6933	1,8840
4,0012	4,3038
5,2509	5,6282
5,5315	6,5012
5,6327	7,0477
6,0014	

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,693356	1E+12				
1	1,693356	4,001288	1,884081404	1	0	1	0
2	1,884081	4,001288	1E+12	0	0	0	1
3	4,001288	5,250927	4,303805741	1	0	1	0

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1
3	4,0012	5,2509	4,3038	1	0	1	0
4	4,3038	5,2509	1E+12	0	0	0	1

Event scheduling events evolution

Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1
3	4,0012	5,2509	4,3038	1	0	1	0
4	4,3038	5,2509	1E+12	0	0	0	1
5	5,2509	5,5315	5,6282	1	0	1	0

Event scheduling events evolution

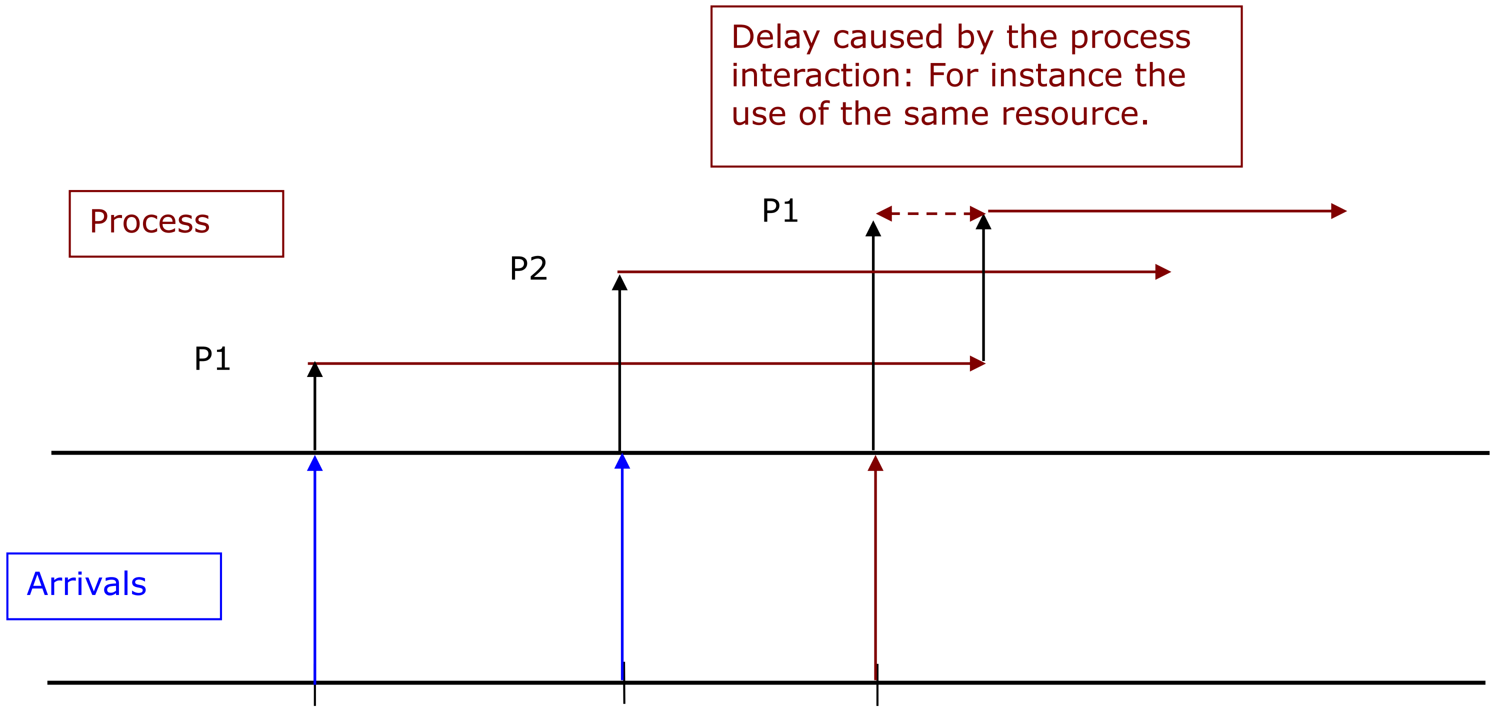
Id	Time	Next arrival	Next exit	Server state	Queue	Arrive	Exit
		1,6933	1E+12				
1	1,6933	4,0012	1,8840	1	0	1	0
2	1,8840	4,0012	1E+12	0	0	0	1
3	4,0012	5,2509	4,3038	1	0	1	0
4	4,3038	5,2509	1E+12	0	0	0	1
5	5,2509	5,5315	5,6282	1	0	1	0
6	5,5315	5,6327	5,6282	1	1	1	0
7	5,6282	5,6327	6,5012	1	0	0	1
8	5,6327	6,0014	6,5012	1	1	1	0
9	6,0014	7,3736	6,5012	1	2	1	0
10	6,5012	7,3736	7,0477	1	1	0	1
11	7,0477			1	0	0	1

Process interacion

Process interaction

- Two different process typologies, P1 and P2:
  - P1 in the usual process of a G|G|1 system. The entity that arrives to the system needs the services of the server.
  - The second process, P2, represents the process where the entities do no require the services of a server, however the entities suffers a delays.

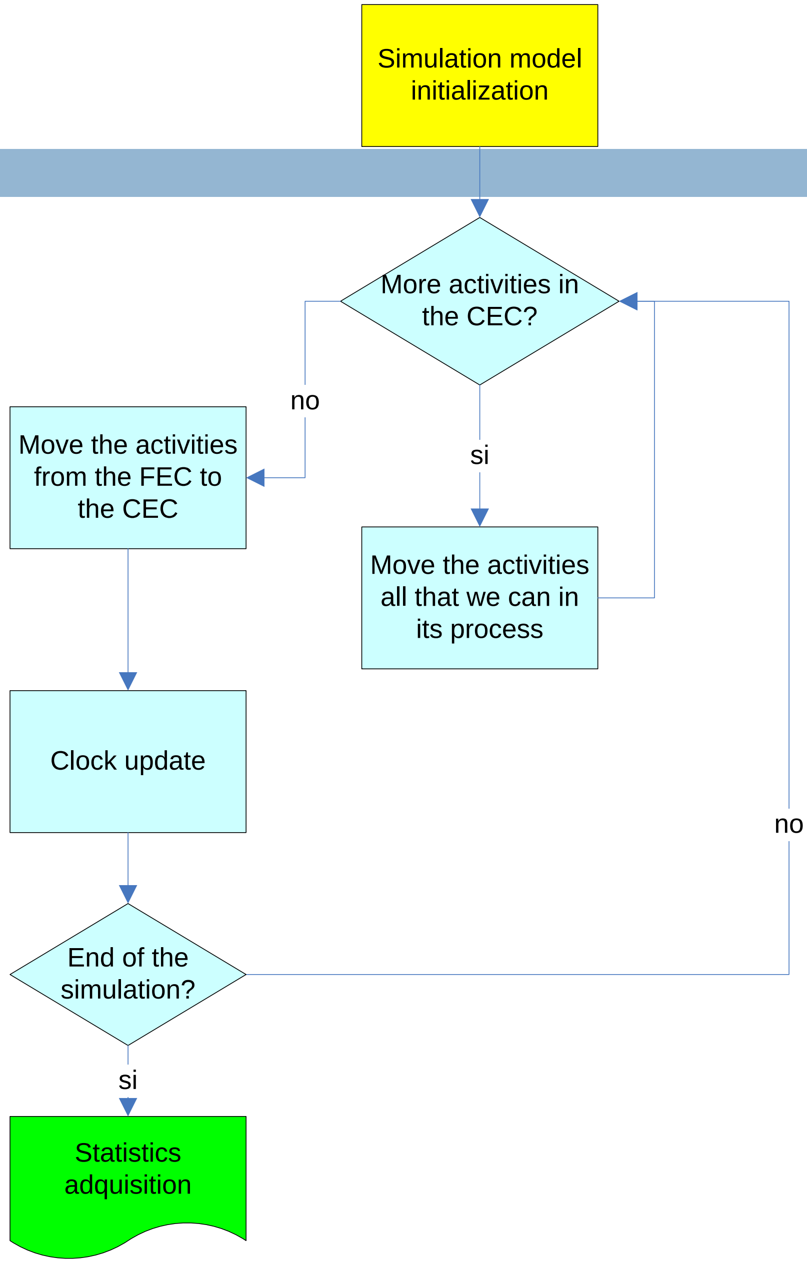
PI: chronogram



PI: Event list

- To simplify usually two list of activities are used. The activities that must be processed in the actual time, and the activities that must be processed in the future.
- The structure, however is quite similar to the structure shown in the Event Scheduling paradigm. Is important to remark the strong relation between the entity and the process linked to each entity.

PI:



Simulation model initialization
More activities in the CEC?
no
si
Move the activities from the FEC to the CEC
Move the activities all that we can in its process
Clock update
no
End of the simulation?
si
Statistics adquisition

Process interaction Example

Trace

Example (data)

- Interval between generations:
  - (2,2,4,4)
- We only generate 4 entities.

1. Enter 3 ± 1 minutes
2. Start Store
3. Entering Lathe
4. Leaving Store
5. Turning 3
6. Exit Lathe
7. Exit System

Pas	Temps	CEC	FEC
1	Inici	-	-
2	0	-	(1,-,1,2)

Example (event chains)

Step	Time	CEC	FEC	Comments
1	Inici	-	-	
2	0	-	(1,-,1,2)	First Xact.
3	2	(1,-,1,Now)	-	Xact from FEC to CEC.
4	2	-	(2,-,1,4) (1,5,6,5)	Moving the Xact 1 all that we can, entering in 5 (*advance*). Generatio of the second Xact.
5	4	(2,-,1,Now)	(1,5,6,5)	Xact from FEC to CEC.
6	4	(2,2,3,Now)	(1,5,6,5) (3,-,1,8)	Moving the Xact 2 all that we can, entering the 2 (*seize). Generation of the third* Xact.

Example (event chains)

Step	Time	CEC	FEC	Comments
7	5	(2,2,3, now) (1,5,6, now)	(3,-,1,8)	Xact from FEC to CEC.
8	5	-	(3,-,1,8) (2,5,6,8)	Moving the Xact 1 all that we can, leaving the system. Moving the Xact 2 all that we can, entering the 5 (*advance*).
9	8	(3,-,1,now) (2,5,6, now)	-	Xact from FEC to CEC.
10	8	-	(3,5,6,11) (4,-,1,12)	Moving the Xact 2 all that we can, leaving the system. Moving the Xact 3 all that we can, entering the *5(advance*). Programming the next arrival.

Example (event chains)

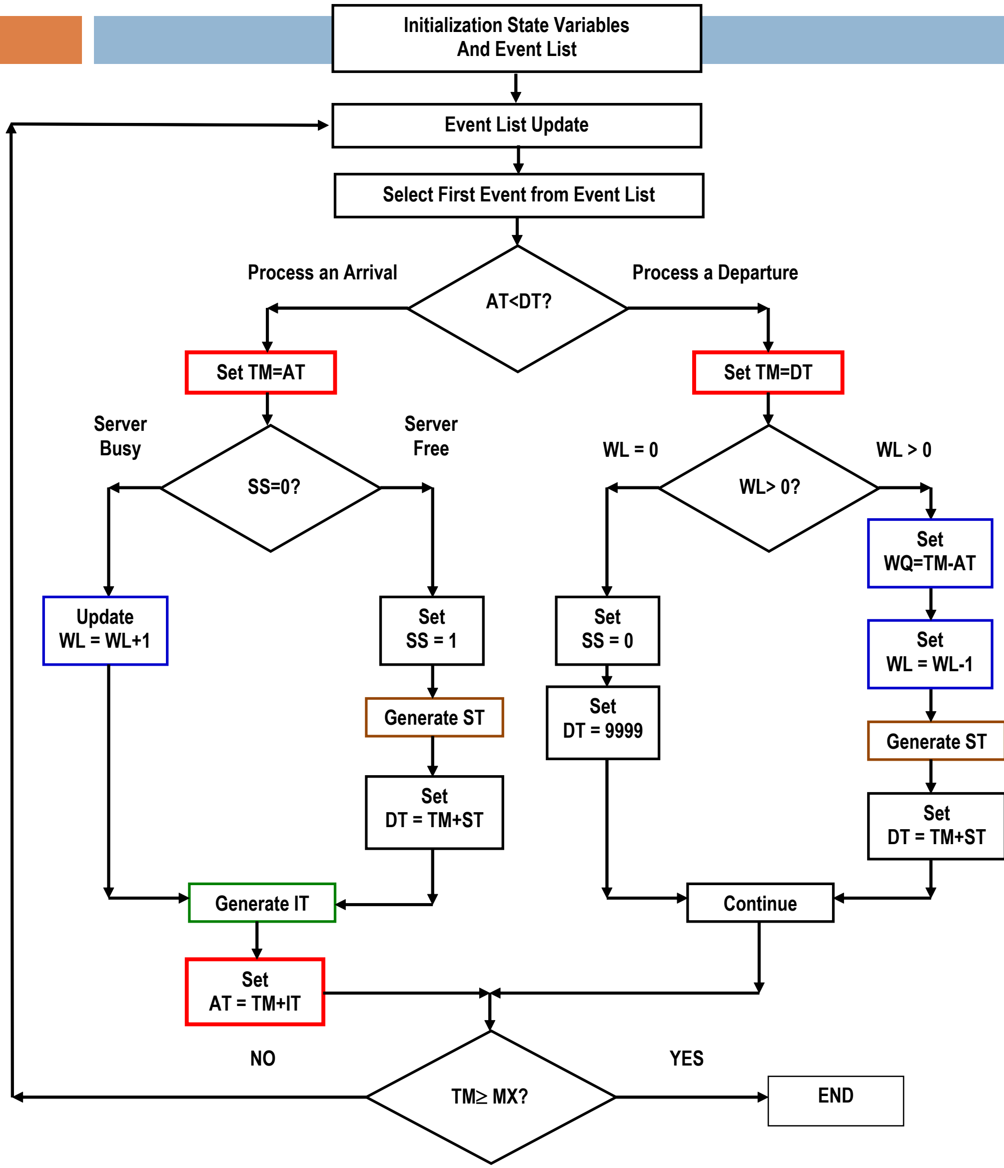
Step	Time	CEC	FEC	Coments
11	11	(3,5,6,Now)	(4,-,1,12)	Xact from FEC a CEC.
12	11	-	(4,-,1,12)	Moving the Xact 3 all than we can, leaves the system.
13	12	(4,-,1,Now)	-	Xact from FEC a CEC.
14	12	-	(4,5,6,15)	Moving the Xact 4 all that we can, entering the 5 bloc (*advance*).
15	15	(4,5,6,Now)	-	Xact from FEC to CEC.
16	15	-	-	Moving the Xact 4 all that we can, leave the system.

An approximation to EVENT SCHEDULING through a QUEUING model

Definition of auxiliary variables

- TM = SIMULATION CLOCK
- ST = SERVICE TIME
- IT = INTERARRIVAL TIME
- AT = ARRIVAL TIME NEXT ARRIVAL AT = TM + IT
- DT = DEPARTURE TIME NEXT DEPARTURE DT = TM + ST
- SS = SERVER STATUS (1= busy, 0 = free)
- WL = QUEUE LENGHT
- WQ = QUEUE WAITING TIME
- MX = LENGTH OF THE SIMULATION HORIZON

M/M/1 queue system



CONCEPTUAL MODEL FOR THE EVENT SCHEDULING SIMULATION ENGINE OF A M/M/1 QUEUE SYSTEM

TM = SIMULATION CLOCK
ST = SERVICE TIME
IT = INTERARRIVAL TIME
AT = ARRIVAL TIME NEXT ARRIVAL AT = TM + IT
DT = DEPARTURE TIME NEXT DEPARTURE DT = TM + ST
SS = SERVER STATUS (1= busy, 0 = free)
WL = QUEUE LENGHT
WQ = QUEUE WAITING TIME
MX = LENGTH OF THE SIMULATION HORIZON

The loading dock example

- Consider a loading dock that has only one facility to load/unload trucks . When a truck arrives, it begins unloading immediately if the dock is free, otherwise it waits in the queue until the dock is free. Trucks arrive in a Poisson fashion at a mean arrival rate of $\lambda = 2$ per hour, while unloading is modeled as a random variable exponentially distributed with a service rate of $\mu = 1.25$ per hour.

Excel event scheduling simulation of the loading dock

		SYSTEM VARIABLES							EVENT LIST	
Random Number Arrivals	Random Number Service Times	EVENT	TM	IT	ST	SS	WL	WQ	AT=TM+IT	DT=TM+ST
*	*	**Initialization**	0	0	0	0	0	0	0	9999
0,08458145	0,11747803	1st Arrival	0	74,101209	102,792189	1	0	*	74,101209	102,792189
0,11320622	*	2nd Arrival	74,101209	65,3563238	*	1	1	*	139,457533	102,792189
	0,96334676	1st Departure	102,792189	*	1,79240894	1	0	0	139,457533	104,584598
		2nd Departure	104,584598	*	*	0	0	28,6909798	139,457533	9999
0,82995016	0,38388135	3rd Arrival	139,457533	5,59168881	45,956244	1	0	0	145,049222	185,413777
0,4386371		4th Arrival	145,049222	24,722486	*	1	1	*	169,771708	185,413777
0,29370974		5th Arrival	169,771708	36,7548983	*	1	2	*	206,526606	185,413777
	0,8844339	3rd Departure	185,413777	*	5,89475975	1	1	40,3645552	206,526606	191,308537
	0,06369117	4th Departure	191,308537	*	132,178046	1	0	21,5368289	206,526606	323,486582
0,04063897		6th Arrival	206,526606	96,0908339	*	1	1	*	302,61744	323,486582
0,55158488		7th Arrival	302,61744	17,8487862	*	1	2	*	320,466226	323,486582
0,86872049		8th Arrival	320,466226	4,22201547	*	1	3	*	324,688242	323,486582
	0,5669783	5th Departure	323,486582	*	27,2368436	1	2	116,959976	324,688242	350,723426
0,77824225		27th Arrival	779,05714	7,52152283	*	1	9	*	786,578663	853,570473
0,51206971		28th Arrival	786,578663	20,0788355	*	1	10	*	806,657499	853,570473
0,79755138		29th Arrival	806,657499	6,7862707	*	1	11	*	813,443769	853,570473
0,66887433		30th Arrival	813,443769	12,0647726	*	1	12	*	825,508542	853,570473
0,65338755		31st Arrival	825,508542	12,7675451	*	1	13	*	838,276087	853,570473
0,3206131		32nd Arrival	838,276087	34,1256051	*	1	14	*	872,401692	853,570473
	0,54679759	18th Departure	853,570473	*	28,976476	1	13	213,14413	872,401692	882,546949
0,60783443		33rd Arrival	872,401692	14,9355825	*	1	14	*	887,337275	882,546949
	0,07730083	19th Departure	882,546949	*	122,882431	1	13	208,933816	887,337275	1005,42938

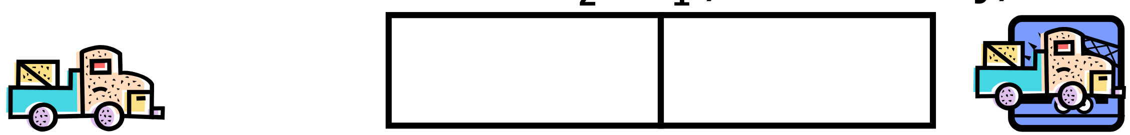
The loading dock example avoiding unlimited queue

- Consider a change in the loading dock limiting the room to two places for trucks to wait. If a truck is at the dock being unloaded and two trucks are waiting, all other arriving trucks go to other loading docks. When a truck arrives, it begins unloading immediately if the dock is free, otherwise it waits in the queue until the dock is free or is turned away if the queue is full. Trucks arrive in a Poisson fashion at a mean arrival rate of $\lambda = 2$ per hour, while unloading is modeled as a random variable exponentially distributed with a service rate of $\mu = 1.25$ per hour.

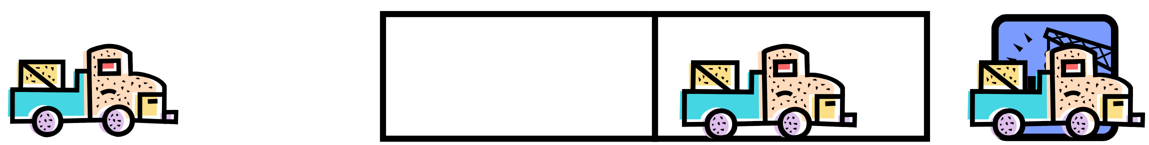
A truck arrives at time t_1, will be served until τ_1 , system empty, server free

The truck seizes the server

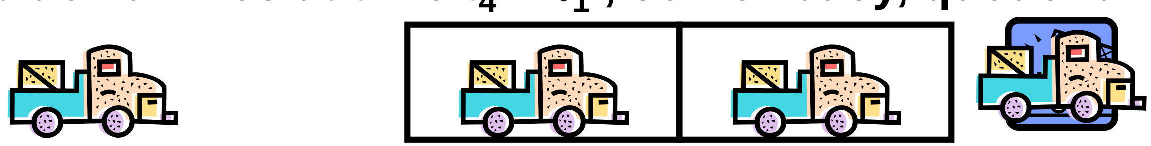
Next truck arrives at time $t_2 < \tau_1$, server busy, enters queue



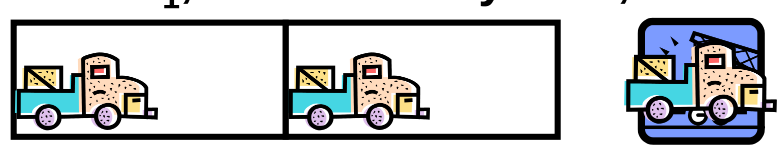
A third truck arrives at time $t_3 < \tau_1$, server busy, enters queue



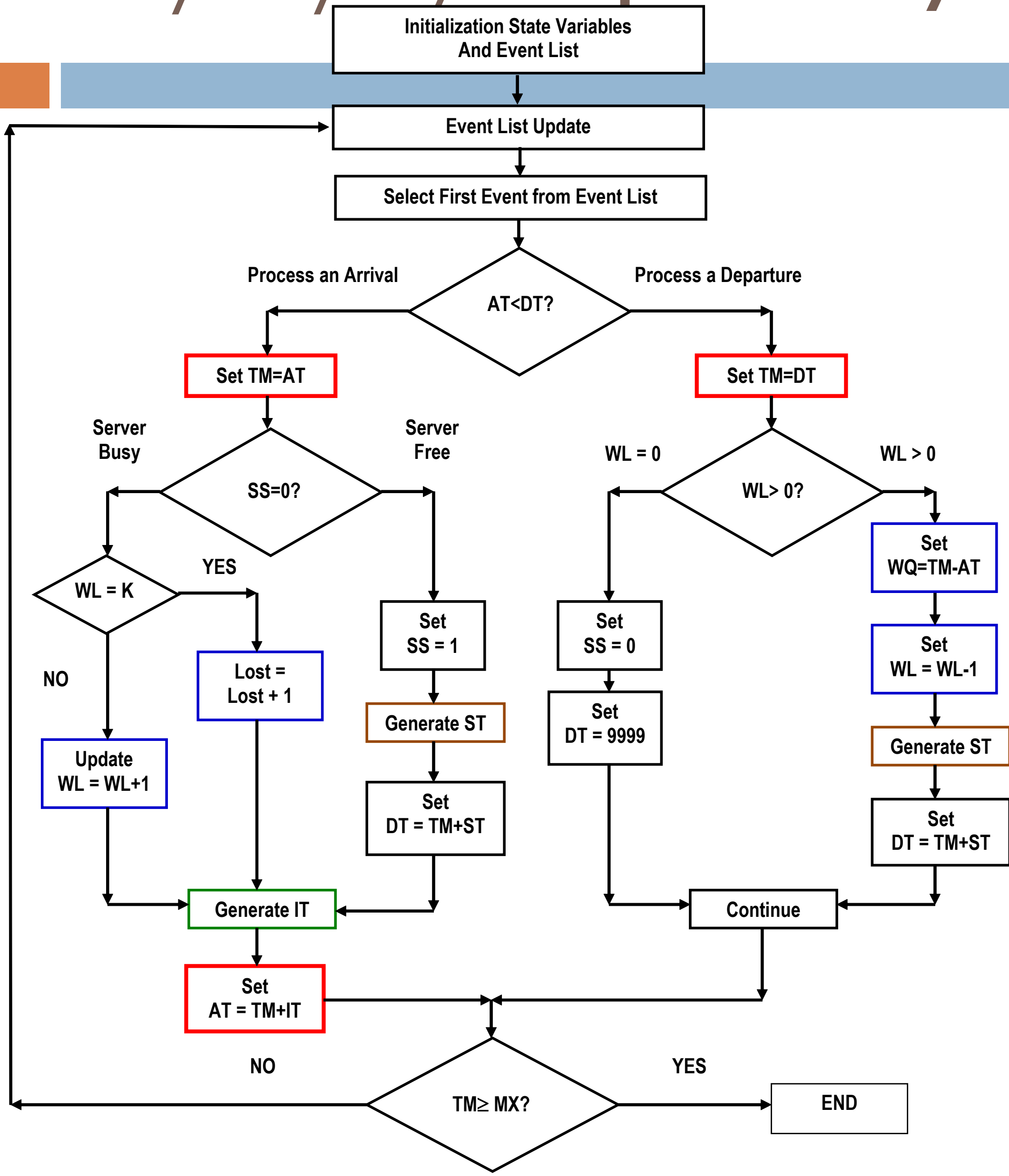
A Fourth truck arrives at time $t_4 < \tau_1$, server busy, queue full, departs not served



The first truck unloaded at time τ_1, leaves the system, server free, second truck seizes server



M/M/1/3 queue system



CONCEPTUAL MODEL FOR THE EVENT SCHEDULING SIMULATION OF A M/M/1/K QUEUE SYSTEM

TM = SIMULATION CLOCK
ST = SERVICE TIME
IT = INTERARRIVAL TIME
AT = ARRIVAL TIME NEXT ARRIVAL AT = TM + IT
DT = DEPARTURE TIME NEXT DEPARTURE DT = TM + ST
SS = SERVER STATUS (1= busy, 0 = free)
WL = QUEUE LENGHT
WQ = QUEUE WAITING TIME
MX = LENGTH OF THE SIMULATION HORIZON

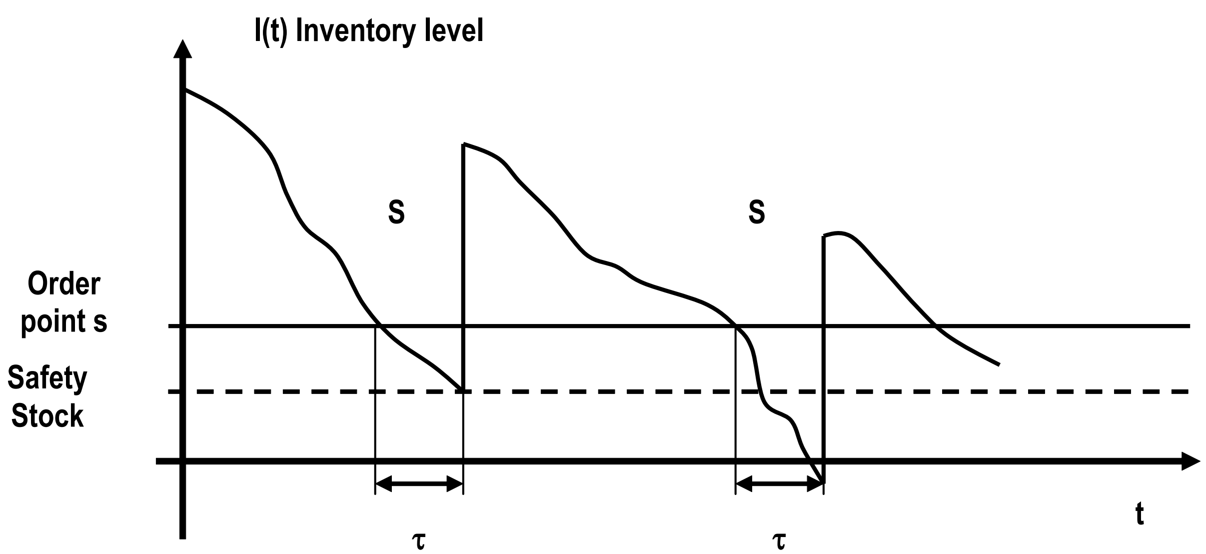
Simulating M/M/1/3 queue system

		SYSTEM VARIABLES							EVENT LIST						
															0,08458145
Random Number Arrivals	Random Number Service Times	EVENT	TM	IT	ST	SS	WL	WQ	AT=TM+IT	DT=TM+ST		λ = 2 trucks/h	avg-it=30 min	30,00	0,11747803
*	*	Initialization	0	0	0	0	0	0	0	9999		μ = 1,25 trucks/h	avg-st = 48 min	48,00	0,11320622
0,08458145	0,117478031	1st Arrival	0	74,101209	102,792189	1	0	*	74,101209	102,792189					0,96334676
0,113206225	*	2nd Arrival	74,101209	65,3563238	*	1	1	*	139,457533	102,792189					0,82995016
	0,963346756	1st Departure	102,792189	*	1,79240894	1	0	0	139,457533	104,584598					0,38388135
		2nd Departure	104,584598	*	*	0	0	28,6909798	139,457533	9999					0,4386371
0,829950161	0,383881353	3rd Arrival	139,457533	5,59168881	45,956244	1	0	0	145,049222	185,413777					0,29370974
0,438637096		4th Arrival	145,049222	24,722486	*	1	1	*	169,771708	185,413777					0,8844339
0,293709741		5th Arrival	169,771708	36,7548983	*	1	2	*	206,526606	185,413777					0,06369117
	0,884433904	3rd Departure	185,413777	*	5,89475975	1	1	40,3645552	206,526606	191,308537					0,04063897
	0,063691174	4th Departure	191,308537	*	132,178046	1	0	21,5368289	206,526606	323,486582					0,55158488
0,040638971		6th Arrival	206,526606	96,0908339	*	1	1	*	302,61744	323,486582					0,86872049
0,551584882		7th Arrival	302,61744	17,8487862	*	1	2	*	320,466226	323,486582					0,5669783
0,868720492		8th Arrival	320,466226	4,22201547	*	1	3>2	*	324,688242	323,486582	The truck leaves the system without being served				0,0801374
	0,566978303	5th Departure	323,486582	*	27,2368436	1	1	116,959976	324,688242	350,723426					0,91190624
0,080137396		9th Arrival	324,688242	75,7203801	*	1	2	*	400,408622	350,723426					0,22119177
	0,911906237	6th Departure	350,723426	*	4,42646904	1	1	30,2571999	400,408622	355,149895					0,67407083
	0,221191769	7th Departure	355,149895	*	72,4188107	1	0	30,4616534	400,408622	427,568706					0,80855459
0,674070832		10th Arrival	400,408622	11,8326024	*	1	1	*	412,241224	427,568706					0,68465343
0,808554587		11th Arrival	412,241224	6,37521258	*	1	2	*	418,616437	427,568706					0,42390011
0,684653434		12th Arrival	418,616437	11,3652751	*	1	3>2	*	429,981712	427,568706	The truck leaves the system without being served				0,13578017
	0,423900106	8th Departure	427,568706	*	41,1963576	1	1	27,160084	429,981712	468,765063					0,79403135
0,135780173		13th Arrival	429,981712	59,9015422	*	1	2	*	489,883254	468,765063					0,25515927
	0,794031348	9th Departure	468,765063	*	11,0703522	1	1	56,5238391	489,883254	479,835415					0,16884783
	0,25515927	10th Departure	479,835415	*	65,5616324	1	0	61,2189788	489,883254	545,397048					0,76021492
0,168847835		14th Arrival	489,883254	53,3627207	*	1	1	*	543,245975	545,397048					0,79561849
0,760214919		15th Arrival	543,245975	8,22462294	*	1	0	*	551,470598	545,397048					0,33512386
	0,795618491	11th Departure	545,397048	*	10,9745035	1	1	55,5137939	551,470598	556,371551					0,64334889
0,335123856		16th Arrival	551,470598	32,7976529	*	1	2	*	584,268251	556,371551					0,7129241

“EVENT SCHEDULING” SIMULATION OF AN INVENTORY SYSTEM

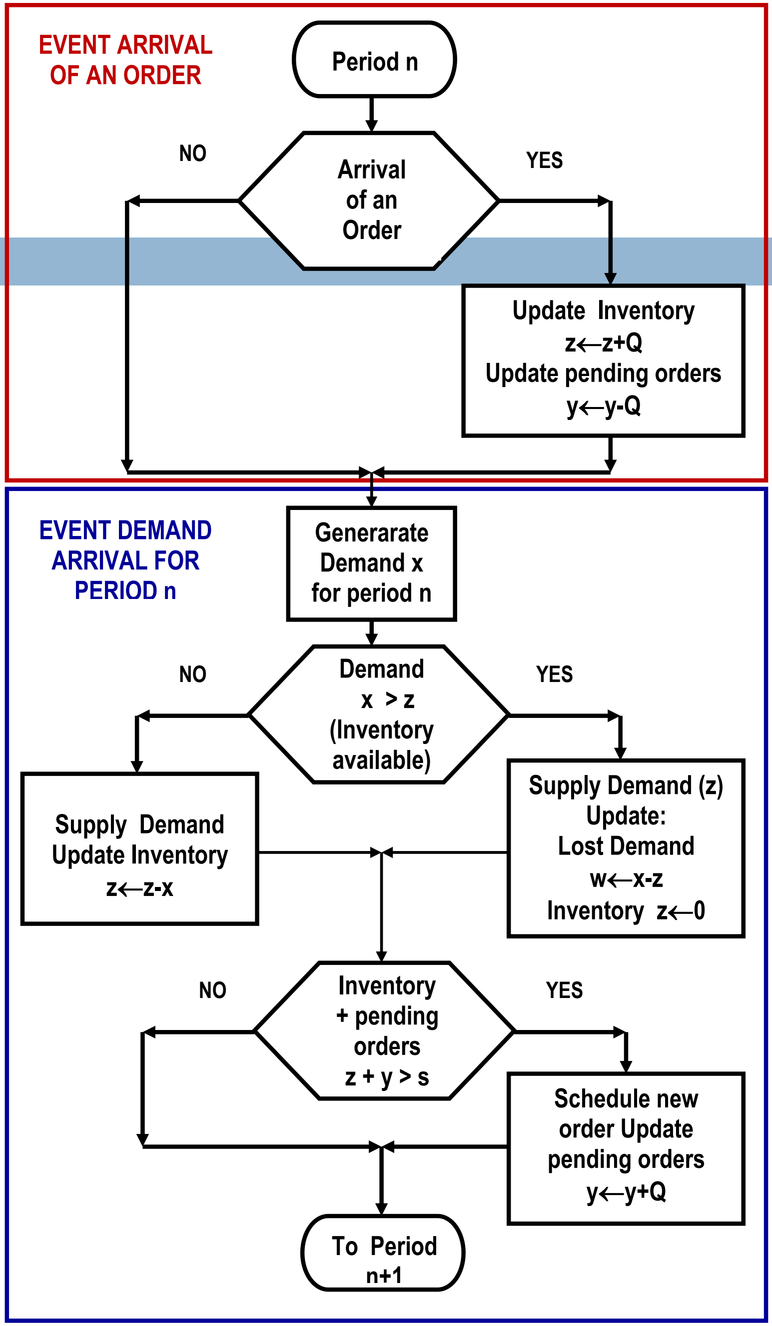
A SIMPLIFIED SYSTEM'S DESCRIPTION

- The demand D of a given product over a given time period (i.e. one week) is a random variable with probability function PD(d)=P{D=d} (probability of a demand equal to d units of the product in the time period considered)in the discrete case and, and $\varphi D(\xi)$ in the continuous case.
- S = is the size of the ordering lot, S may have a variable size or a fixed size S=Q, (ECONOMIC ORDER QUANTITY).
- s = ORDERING POINT, each time the inventory level reaches the value s a new lot is ordered. General Policy (s,S), if s=k, fixed, policy (k,Q))
- τ = LEAD TIME for servicing the ordered lot (may also be a random variable)



Conceptualization

- Z: inventory
- Q: orders
- Y: pending orders
- X: demand
- W: lost demand
- S: safety level



Excel for the Event Scheduling simulation of an inventory system

Q* = 60, r* = 48

Demand	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Frecuency	0.26	0.14	0.12	0.10	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01	0.01	0.01

Lead Time	7	8	9	10	11	12	13	14
Frecuency	0.07	0.12	0.18	0.25	0.20	0.10	0.05	0.03

						Orders			
Period	Invent. z	Order arrrivals	Demand x	Invent. z	Lost Demand w	Done	Lead Time	Arrival Period	y
Initial	-	-	-	60	-	-	-	-	0
1	60		3	57					0
2	57		0	57					0
3	57		0	57					0
4	57		1	56					0
5	56		6	50					0
6	50		1	49					0
7	49		2	47		60	8	15	60
8	47		5	42					60
9	42		0	42					60
10	42		4	38					60
11	38		5	33					60
12	33		4	29					60
13	29		4	25					60
14	25		10	15					60
15	15	60	3	72					0

USING MODELS TO SUPPORT QUANTITATIVE DECISIONS

Statistical calculations from simulation

$$\text{Average Waiting Time} = \frac{\text{Total time customers wait in queue}}{\text{Total number of customers}} = \frac{\sum_{i=1}^{N} w_i}{N}$$

$$\text{Average Time Customer in System} = \frac{\text{Total time customers spend in system}}{\text{Total number of customers}} =$$

$$\frac{\sum_{n=1}^{N} s_n}{N} = \frac{\sum_{n=1}^{N} \left(x_n + w_n\right)}{N}$$

$$\text{Average Queue Length} = \frac{\sum_{n=1}^{N} q_n}{N}$$

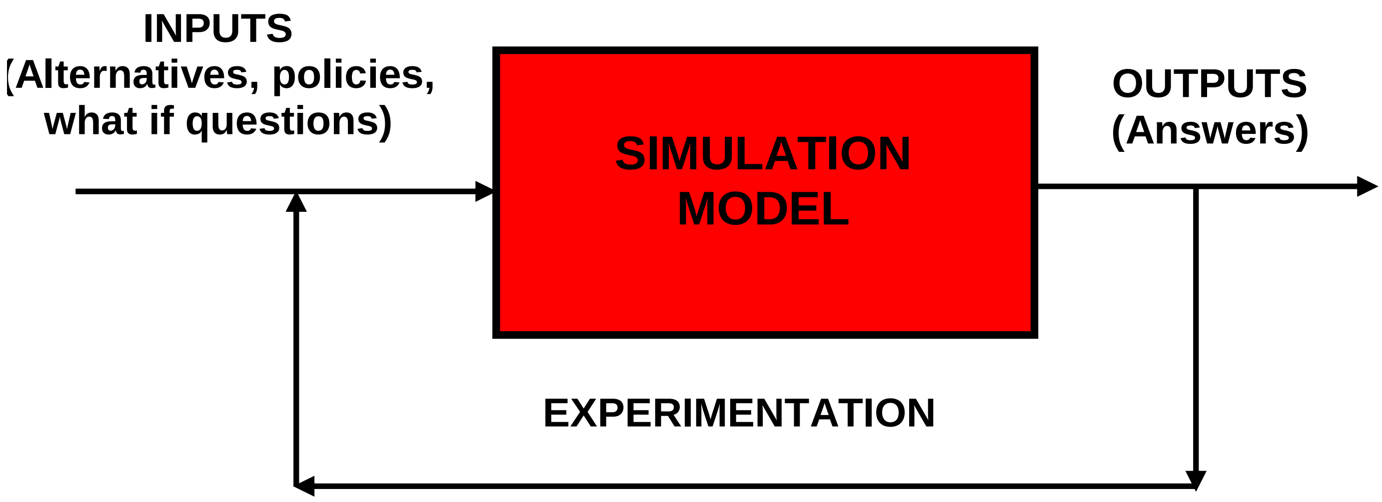
Simulation results

- Average queue length 1.0487 trucks
- Average waiting time in queue 66.4391 minutes
- 33 Trucks arrived
- 19 were unloaded
- 12 left the system without being served
- 2 were in queue awaiting for service

The nature of simulation

- Simulation can be seen as an alternative to analytical models consisting of a technique that imitates on a computer the operation of a real-world system as it evolves over time
- Simulation may be seen as a sampling experiment on the real system through its model

Scheme of the experimental process of simulation



The simulation model can be thought in terms of a computer laboratory to replace experiments with the physical system by experiments with the model to draw valid answers to questions concerning the design of the system, decisions on alternative policies, and so on.

Valid models?

All models are WRONG…

… but some are useful

G. Box

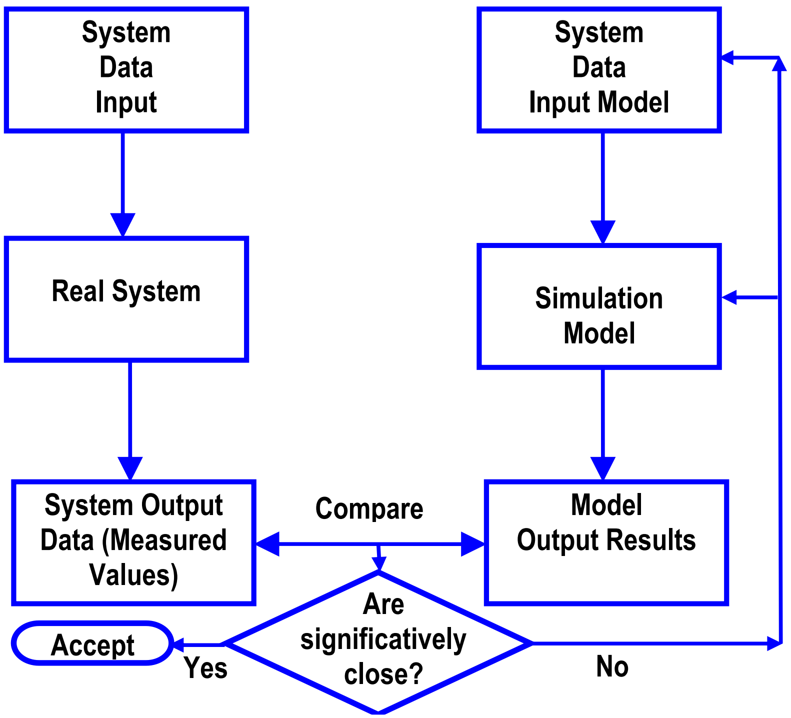
What makes a model useful?

Validation: a key contribution to make models useful

- What is validation?
- Do model predictions faithfully represent reality?
- Quantification:
- (•) P{ |"reality" – simulation prediction| < d } > a
- d= tolerable difference = how close
- a= level of assurance = how certain
- Issues:
  - We are implicitly assuming that "reality" is defined in terms of what we observe: traffic data
  - What is needed to compute P?
  - How to set d and a…. Affected by data availability its accuracy and randomness

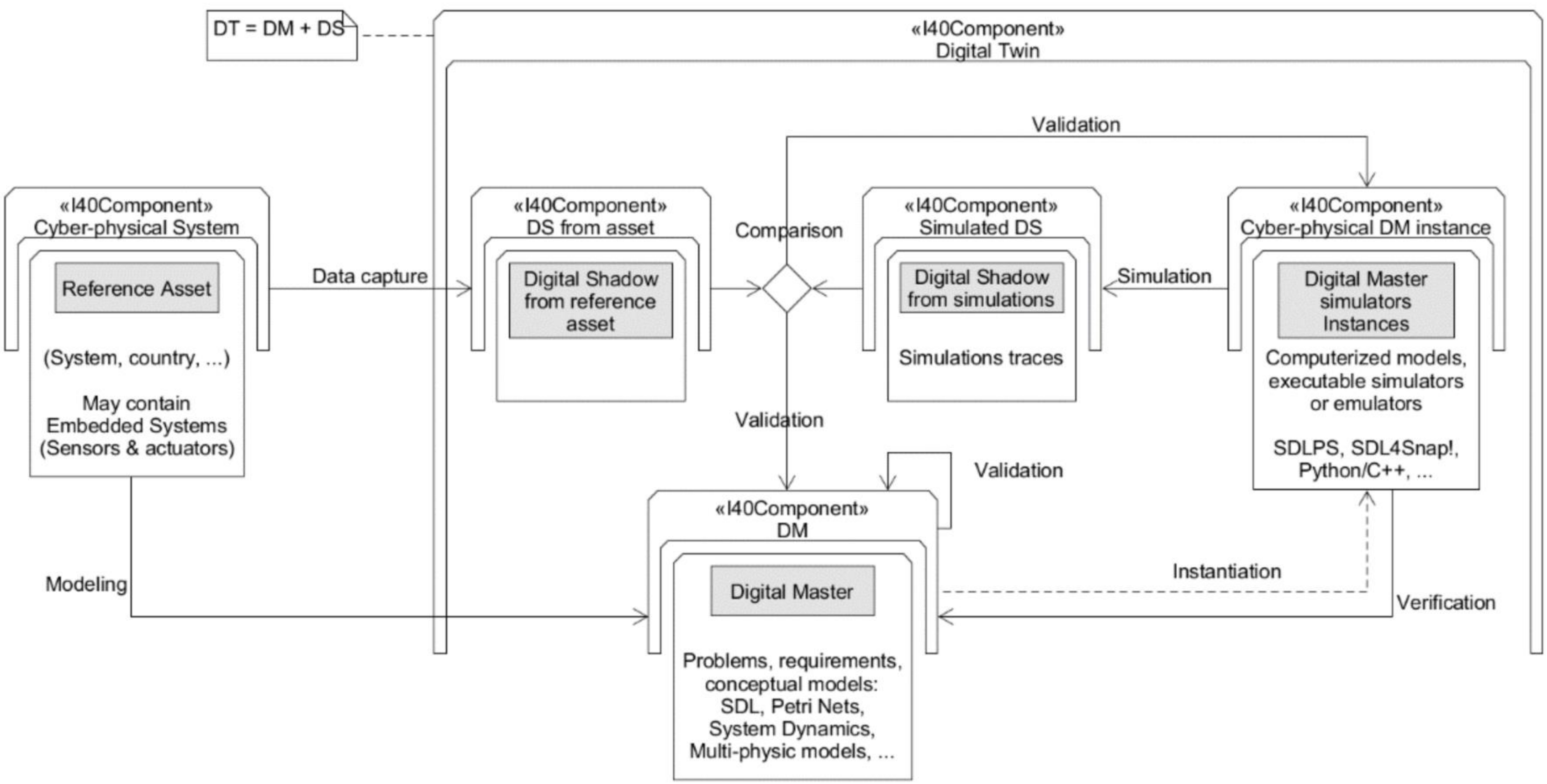
Nagui M. Rouphail & Jerry Sacks, Workshop on Modeling Trends, Sitges, June 2003

Basic methodological approach to V&V of simulation models



- System Data Input
  - Observable: measurements of system variables (i.e. arrival times, service times) affected by errors
- System Data Input Model
  - Exact or good approximations: probability distributions of input data
- Simulation Model
  - Calibrated, that is supplied with the right parameter values
- System Output Data
  - Observable
- Model Output Results
- Are they significantly close?
  - How d and a values are set up?
  - Which statistical techniques are more appropriate?

In the context of Industry 4.0 and 5.0



Fonseca i Casas, P.; Garcia i Subirana, J.; García i Carrasco, V.; Pi i Palomés, X. SARS-CoV-2 Spread Forecast Dynamic Model Validation through Digital Twin Approach, Catalonia Case Study. *Mathematics* **2021**, *9*, 1660. https://doi.org/10.3390/math9141660

Results for the theoretical M/M/1/K model

$$P_n = \left(\frac{1-\rho}{1-\rho^{K+1}}\right)\rho^n; n = 0,1,2\ldots K \qquad \rho = \frac{\lambda}{\mu} \rightarrow \rho = \frac{2}{1.25} = 1.6, K = 3$$

$$P_0 = 0.1734417; P_1 = 0.2168021; P_2 = 0.27100271; P_3 = 0.3387533$$

$$1 - P_0 = 0.8265583$$

$$L = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} \rightarrow L = 1.77506775$$

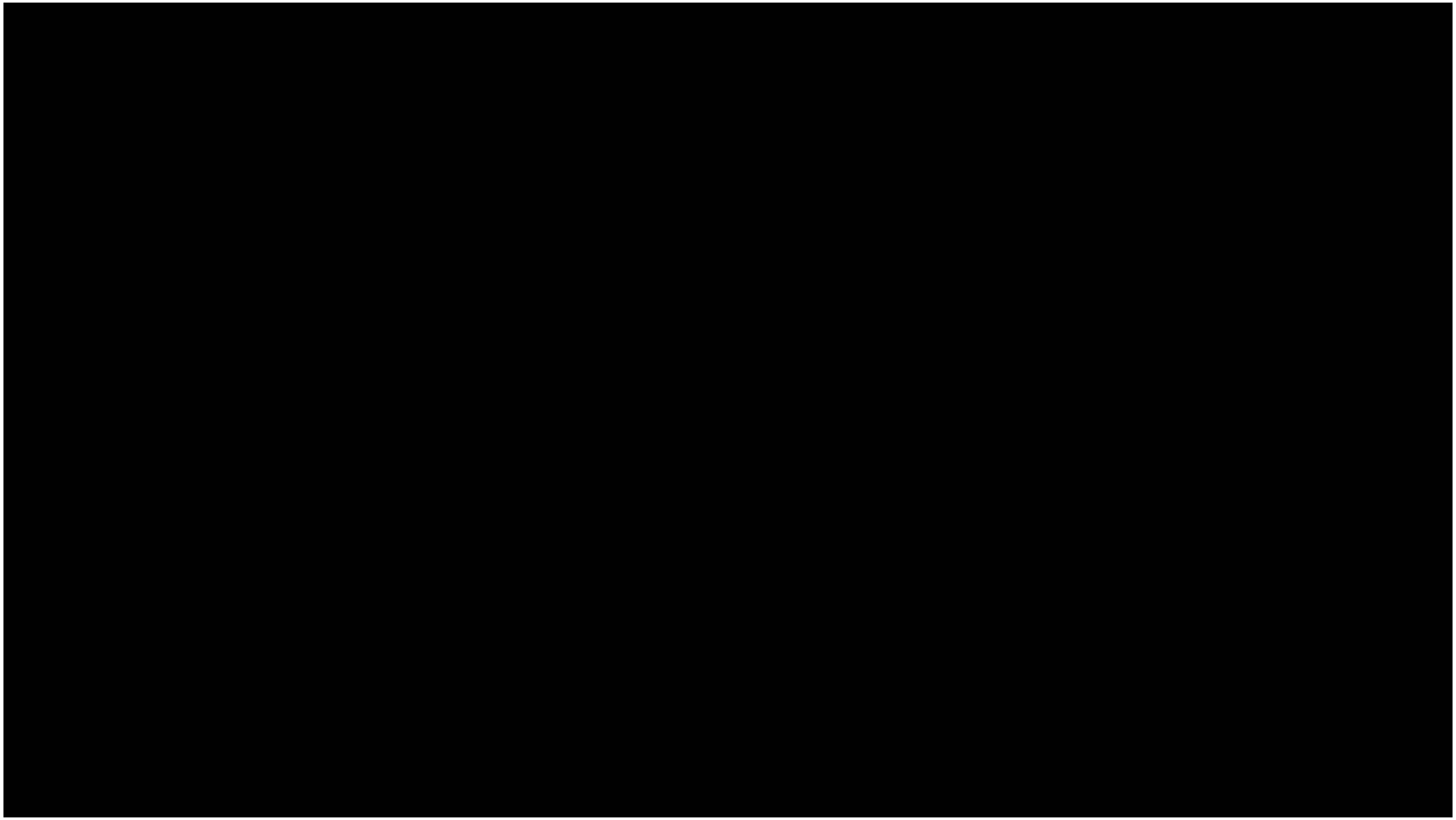
$$L_q = L - (1 - P_0) \rightarrow L_q = 0.9485094$$

$$\overline{\lambda} = \lambda(1 - P_K) = 1.322493 \rightarrow W_q = \frac{L_q}{\overline{\lambda}} = 43.032786$$

$$\overline{W} = W_q + \frac{1}{\mu} = 91.032786 \text{ minutes}$$

Observe the differences with the simulation results: How can be explained?

Multi Agent Systems



Environmental simulation