Investigación Operativa

Operations Research

comillas.edu

COMILLAS UNIVERSIDAD PONTIFICIA ICAI ICADE CIHS

Discrete Event Simulation Modeling

Departamento de Organización Industrial

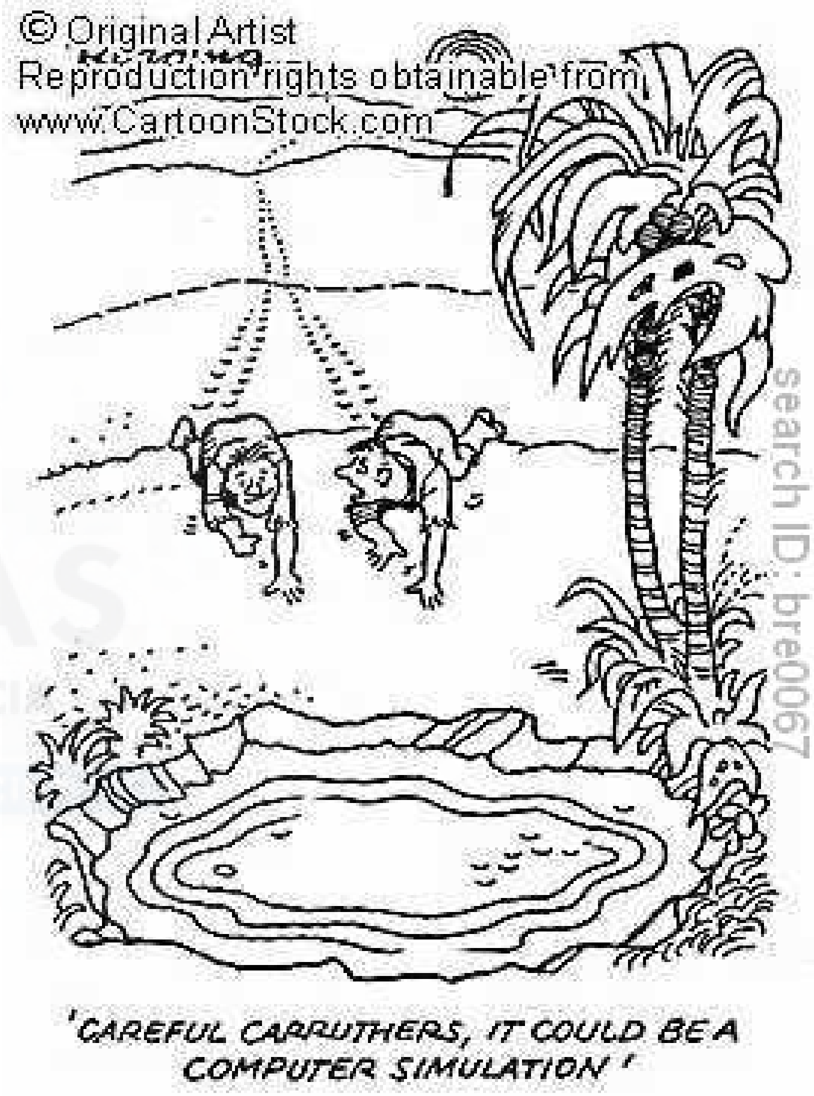
September 1, 2023

Contents

1

Discrete Event Simulation

A bit of humor - simulation as an imitation of reality



Discrete Event Simulation

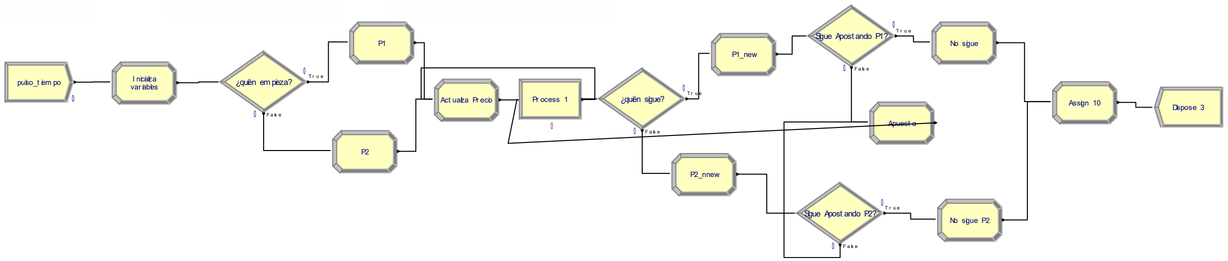
- *Simulation*: imitates the operations of a real-world facility or process, usually via computer
  - What's being simulated is the *system*
  - A system could be *discrete* (State variables change instantaneously at separated points in time, e.g., when the customer arrives/leaves) or *continuous* (state variables change continuously as a function of time, e.g., airplane flight with state variables like position or speed).
  - The logical and mathematical assumptions about how the system works form a *model* of the system
  - If the model structure is simple enough, could use mathematical methods to get exact information on questions of interest — *analytical solution*
  - But most *complex systems require models that are also complex*. Must be studied via simulation — evaluate model numerically and collect data to estimate model characteristics

Discrete Event Simulation: Analytical Technique vs. Simulation

- Queueing Theory (QT) sets mathematical models to estimate the steady-state performance of waiting lines for different types of queueing systems

- Advantages QT vs. Simulation
  - More accurate when QT can be used
  - Much less computational effort
  - Initial system modeling may be studied with QT

- Disadvantages QT vs. Simulation
  - Need strong queueing modeling assumptions to use analytical formulas
  - Many real systems (that can be simulated) are not queueing systems



Discrete Event Simulation: Advantages, disadvantages, and pitfalls of simulation

- Advantages:
  - Simulation allows excellent flexibility in modeling complex systems so that simulation models can be highly valid
  - Easy to compare alternatives
  - Control experimental conditions
  - Can study systems with a very long timeframe
- Disadvantages:
  - Stochastic simulations produce only estimates – with noise
  - Simulation models can be expensive to develop
  - Simulations usually produce large volumes of output – need to summarize, statistical analysis done appropriately
- Implementation pitfalls:
  - Failure to identify objectives clearly upfront
  - Inappropriate level of detail (both ways)
  - Inadequate design and analysis of simulation experiments
  - Inadequate education and training

Discrete Event Simulation: Application areas

- Some (not all) application areas
  - Designing and analyzing manufacturing systems
    - Example: Manufacturing company considering extending its plant. Build it and see if it works out. Simulate current and expanded operations — could also investigate many other issues along the way quickly and cheaply
  - Evaluating military weapons systems or their logistics requirements
  - Determining hardware requirements for communication networks
  - Determining hardware and software requirements for a computer system
  - Designing transportation systems — airports, freeways, ports, and subways
  - Evaluating designs for service organizations such as call centers, fast-food restaurants, hospitals, and post offices
  - Determining ordering policies for an inventory system
  - Analyzing financial or economic systems

Discrete Event Simulation: Basic Concepts

- *System*: A collection of entities (people, parts, messages, machines, servers, etc.) that act and interact together toward some end (Schmidt and Taylor, 1970).

- *Entities*: Objects that compose a simulation model. Elements that go through the model.

- *Attributes:* Data values that characterize entities.

- *State of a system*: Collection of *variables* and their values necessary to describe the system.

- *Resources*: Elements demanded by entities (machines, personnel, etc.)

Discrete Event Simulation: Basic Concepts - Example

- *System*: Model that considers the daily operation of a bank branch.

- *Entities*: Customers that enter the bank branch.
- *Attributes (of each entity)*:
  - Time of arrival/departure of each customer
  - Number of activities each customer must carry out in the bank
  - Cash each customer withdraws
- *Resources*: Tellers

- *System Variables*:
  - Number of busy tellers
  - Total number of customers in the bank branch
  - Number of people in the queue

Example of Entities, Attributes and Variables

System: Warehouse

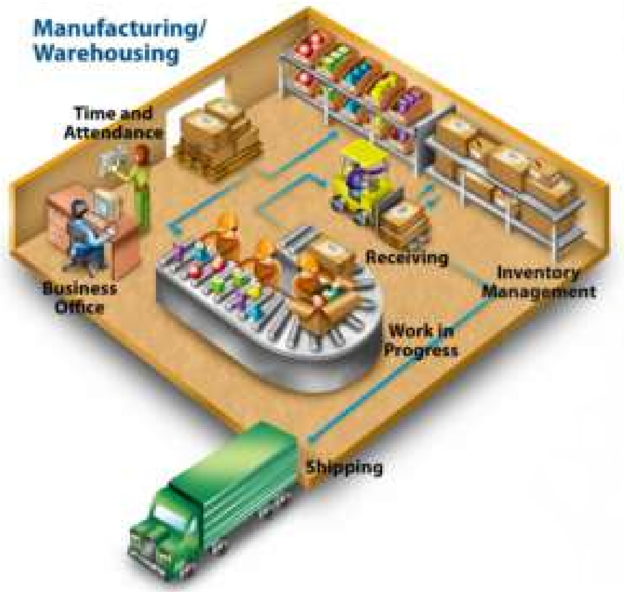
Entity: Part
Attributes: Units, Reference, Weight,…

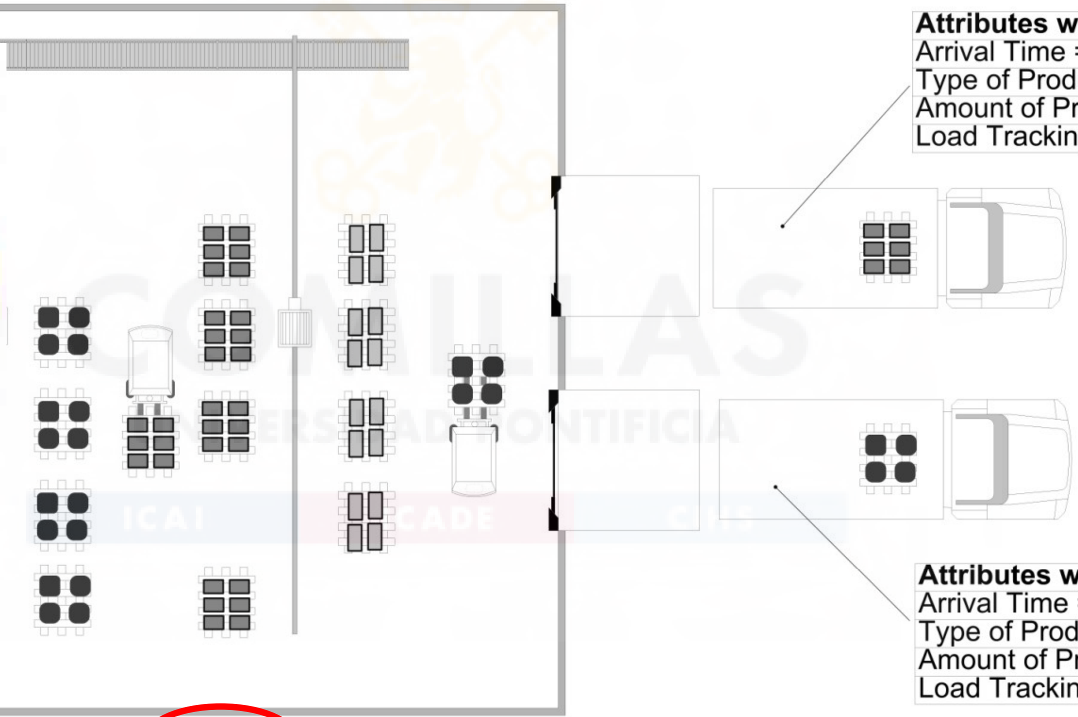
Entity: Truck

Examples of Delivery Truck Attributes
Arrival Time
Type of Product
Amount of Product
Load Tracking Number

Loading Area

Unloading Area





Attributes with values
Arrival Time = 10:45
Type of Product = A
Amount of Product = 6
Load Tracking Number = 0684432

Attributes with values
Arrival Time = 10:00
Type of Product = B
Amount of Product = 4
Load Tracking Number = 0687922

Examples of Global Variables
Number of Trucks Loading = 0
Number of Trucks Unloading = 2
Number of Busy Forklifts = 2
Number of Busy Operators = 2
Amount of Product A in Storage = 54
Amount of Product B in Storage = 20
Amount of Product C in Storage = 16

Attributes can be thought of
as variables attached to entities

Discrete Event Simulation: More Concepts

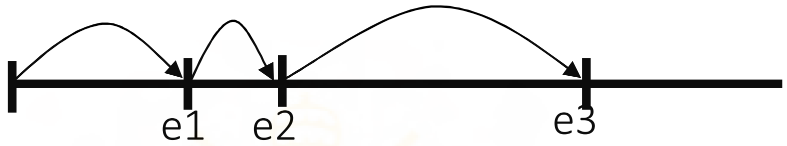
- *Discrete event simulation*: Modeling a system as it evolves, and state variables only change instantaneously at separated points in time
  - State changes at only a *countable* number of points in time
  - These points in time are when *events* occur

- *Event*: Instantaneous occurrence that may change the state of the system

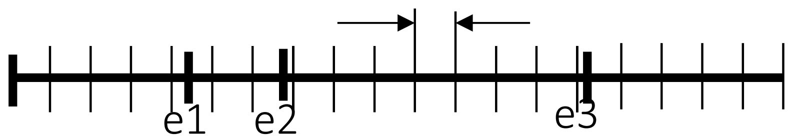
- Simulation clock: Variable that keeps the current value of (simulated) time in the model
  - Must decide on and be consistent about time units
  - Usually, no relation between simulated time and (real) time needed to run a model on a computer

Discrete Event Simulation: Time-Advance Mechanisms

- Two approaches for time advance:
  - *Next-event time advance* (usually used) … described in detail below



    - Determine times of occurrence of future events - update the *event list*
    - Clock "jumps" from one event time to the next (until the stopping criterion is met) and doesn't "exist" for times between successive events; periods of inactivity are ignored
  - *Fixed-increment time advance* (seldom used: *virtual simulation*)
    - Generally, introduces some amount of modeling error in terms of when events *should* occur vs. *do* occur



2

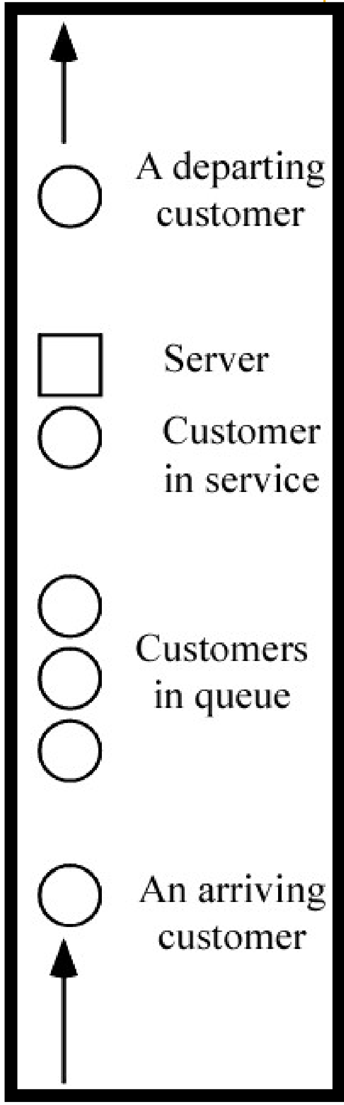
Example of a Simulation Model

Example of a Simulation Model: Problem Statement

Single-Server Queueing Model

Data

- Assume interarrival times are independent and identically distributed (IID) random variables
- Assume service times are IID and are independent of interarrival times
- Queue discipline is FIFO (First In, First Out)
- Starts empty and idle at time 0
- The first customer arrives after an interarrival time, not at time 0
- Stopping rule: When the n-th customer has completed the delay in the queue (i.e., *enters* the service) … n will be specified as input



Example of a Simulation Model: Time Inputs

- Next-event time advance for the single-server queue is composed by:

  t_i = time of arrival of i-th customer ($t_0 = 0$)

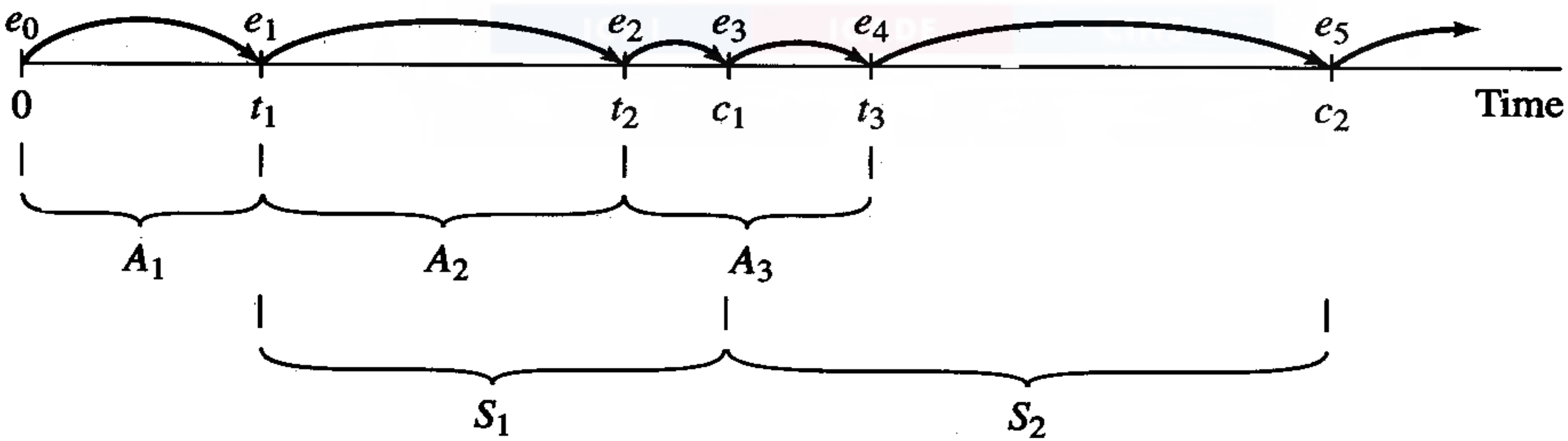
  $A_i = ti - t_{i-1}$ = interarrival time between $(i-1)$-th and i-th customers (usually assumed to be a random variable from some probability distribution)

  S_i = service-time requirement of i-th customer (another random variable)

  D_i = delay in queue of i-th customer

  $C_i = ti + Di + Si$ = time i-th customer completes service and departs

  e_j = time of occurrence of the j-th event (of any type), j = 1, 2, 3, …



Example of a Simulation Model: Outputs

- Outputs to be estimated:
  - *Expected average delay in the queue* (excluding service time) of the n customers completing their delays
    - Why "expected?"
  - *Expected average number of customers in queue* (excluding any in service)
    - A *continuous-time average*
    - Area under $Q(t)$ = queue length at time t, divided by $T(n)$ = time simulation ends
  - *Expected utilization (proportion of busy time) of the server*
    - Another *continuous-time average*
    - Area under $B(t)$ = server-busy function (1 if busy, 0 if idle at time t), divided by $T(n)$ …
  - Many others are possible (maximum, minimum, time or number in system, proportions, quantiles, variances …)
- Important: *Discrete-time* vs. *continuous-time* statistics

Example of a Simulation Model: Simulation by Hand

- Simulation by "hand":
  - Display system, state variables, clock, event list, and statistical counters … all *after* execution of each event
  - Use the lists below of interarrival and service times to "drive" the simulation
  - Stop when number of delays hits n = 6 (6th customer starts being served), compute output performance measures

- Given (for now) interarrival times (all times are in minutes):

  0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, …

- Given service times:

  2.0, 0.7, 0.2, 1.1, 3.7, 0.6, …

- $\boldsymbol{n}$ = 6 delays in queue desired

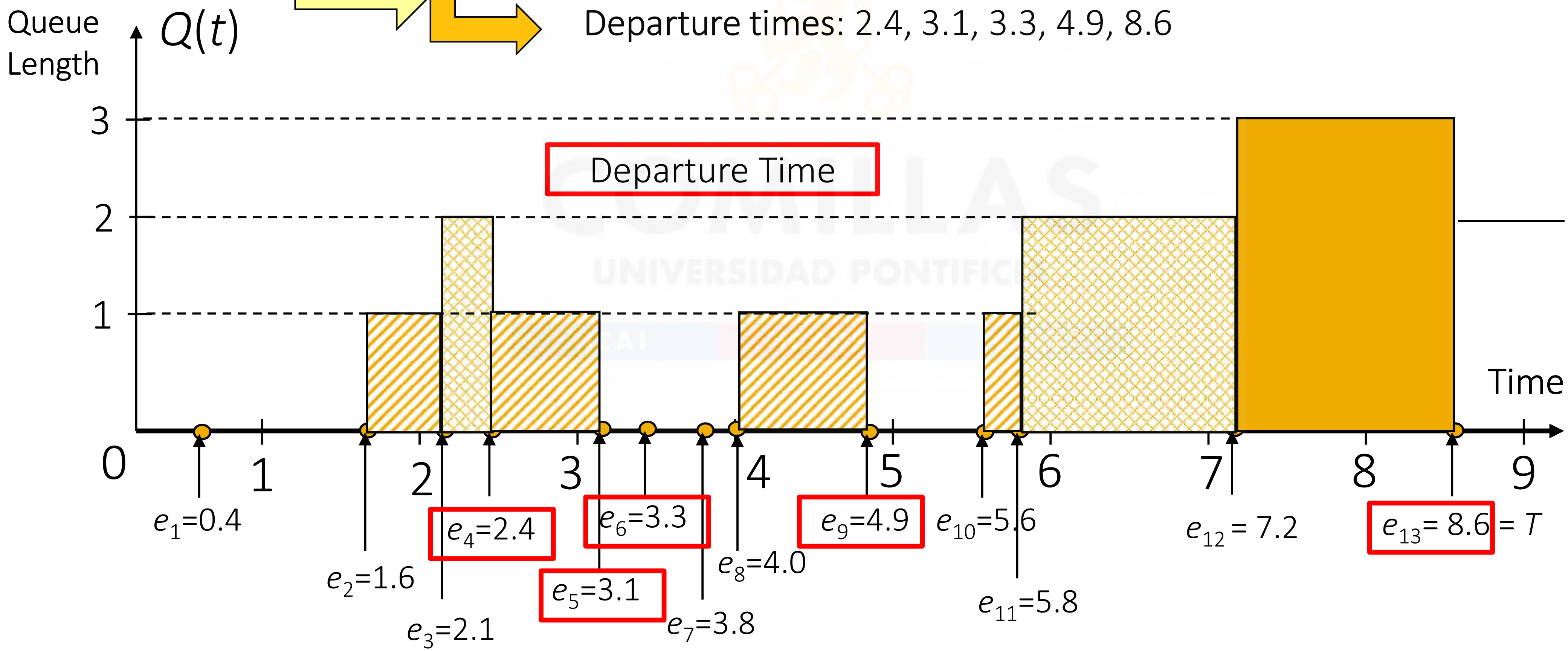
Example of a Simulation Model: Simulation by Hand (Cont'd)

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, …

Arrival times: 0.4, 1.6, 2.1, 3.8, 4.0, 5.6, 5.8, 7.2

Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6,…

Departure times: 2.4, 3.1, 3.3, 4.9, 8.6



3

Simulation Software

Simulation Software:
Simulation Packages vs. Programming languages

- Packages
  - Software in high-level programming code very specialized in simulation modeling
  - Examples: ARENA, SIMIO, SimEvents, WITNESS, AUTOMOD, GPSS, etc.

- Programming languages
  - General programming languages that can be used for simulation modeling
  - Examples: C++, Visual Basic, etc.

Simulation Software:
Simulation Packages vs. Programming languages (Cont'd)

- Advantages of using packages
  - Natural framework for simulation modeling
  - Usually make it easier to modify models
  - Better error detection for simulation-specific errors

- Pitfalls of using packages
  - Require a specific package learning process
  - Higher execution time
  - Less modeling flexibility
  - More expensive software

- Advantages of using programming languages
  - More widely known, available, and more modeling flexibility.
  - Usually executes faster … if well written
  - Software cost is usually lower

Simulation Software: Desirable features of packages

STATISTICAL CAPABILITIES

- Adequate **random-number generator** (RNG) for basic U(0,1) variables
  - Statistical properties, cycle length, adequate streams, and substreams
  - RNG seeds should have good defaults and be fixed – not dependent on the clock
- Comprehensive list of input probability distributions
  - Continuous, discrete, empirical
- Ability to make independent replications
- Confidence-interval computation for output performance measures
- Warm-up
- Experimental design
- Optimum-seeking

Simulation Software: Common elements

- Common elements of simulation packages:
  - Entities: Elements that go through the model
  - Attributes: Specific characteristics for each entity (type, size, etc.)
  - Resources: Elements demanded by entities (machines, personnel, etc.)
  - Queues: Waiting lines in front of resources

  - Examples:

	Entities	Attributes	Resources
• Manufacture	Products	Due date	Machines
• Communication	Calls	Destination	Phone operators
• Airports	Planes	Capacity	Runways
• Bank branch	Customer	Waiting time	Tellers

Simulation Software: Commercial packages

- ARENA
  - General purpose package
  - Includes basic and advanced modeling modules
  - Graphical model design
  - 2D and 3D Animation
  - Includes transport components (conveyors, trucks, and AGVs)
  - Visual Basic Applications

https://www.rockwellautomation.com/es-mx/products/software/arena-simulation.html

4

Analyzing Simulation Output

Contents – Analyzing Simulation Output

i. Types of Statistical Variables

ii. Types of Simulation concerning Output Analysis

iii. Analysis of Finite-Horizon Simulations

iv. Analysis of Infinite-Horizon Simulations

4i

Types of Statistical Variables

Types of Statistical Variables: Definitions

- A simulation experiment occurs when the modeler sets the input parameters to the model and runs the simulation

  - Events occur, and the simulation model evolves over time

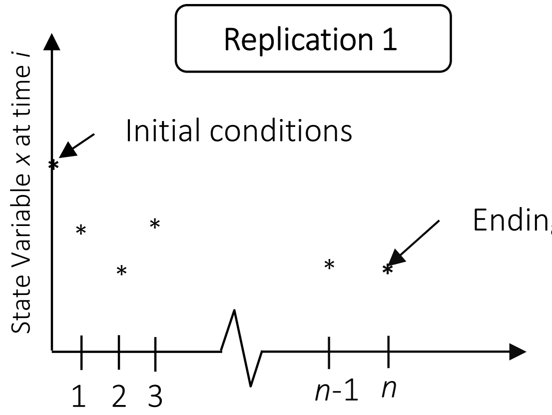
  - Statistical quantities are computed, and at the end of the simulation those quantities are summarized in output reports

- A replication is the generation of one sample path, representing the evolution of the system from its initial conditions to its ending conditions

  - If the simulation experiment has multiple replications within an experiment, each replication represents a different sample path (with the same initial conditions and input parameter settings)

  - The underlying random numbers within each replication can be made to be independent

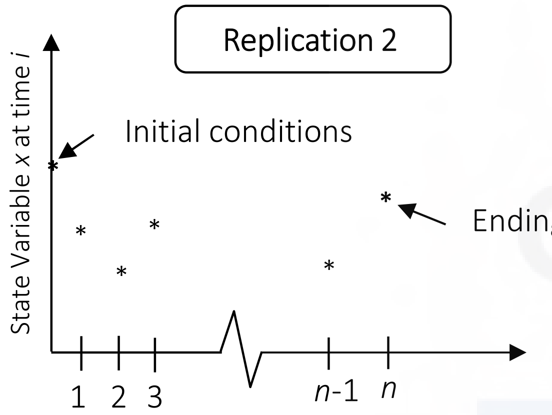
Types of Statistical Variables: Within/Across Replication Stats



Within Replication Statistics

$$\bar{X}_1 = \frac{1}{n}\sum_{i=1}^{n} X_{1,i}$$

The statistical quantities collected within a replication



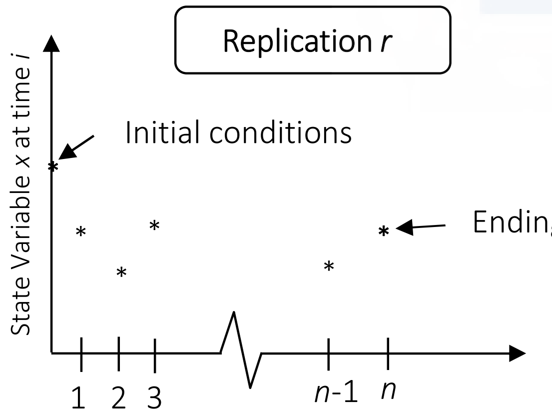
$$\bar{X}_2 = \frac{1}{n}\sum_{i=1}^{n} X_{2,i}$$

Across Replication Statistics

The statistical quantities collected across the replications

$$\bar{Y} = \frac{1}{2}\sum_{j=1}^{2} \bar{X}_j$$

The statistical properties of both types are different and require different methods of analysis



$$\bar{X}_r = \frac{1}{n}\sum_{i=1}^{n} X_{r,i}$$

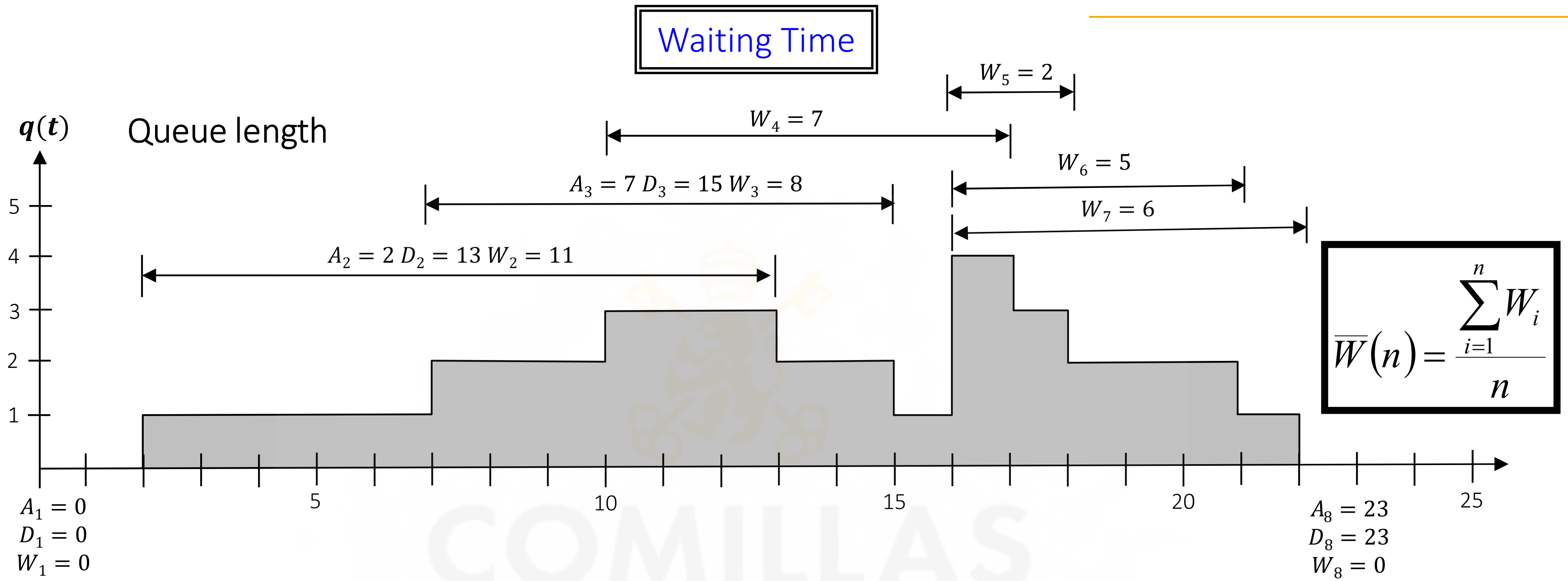
$$\bar{Y} = \frac{1}{r}\sum_{j=1}^{r} \bar{X}_j$$

Types of Statistical Variables: Observation- vs Time-based

- Within replication statistics, there are two primary types of statistical data:

  - Observation-based data:
    * Sequence of equally weighted data values that do not persist over time
    * *Example*: Average waiting time for a queue
  - Time-based data:
    * This type of data is associated with the duration or interval of time that an object is in a particular state
    * It is observed by marking the time that the object is in a particular state
    * They represent a sequence of values that persist over some specified amount of time with that value being weighted by the amount of time over which the value persists
    * *Example*: Average number of people in the queue

Types of Statistical Variables: Example (I)



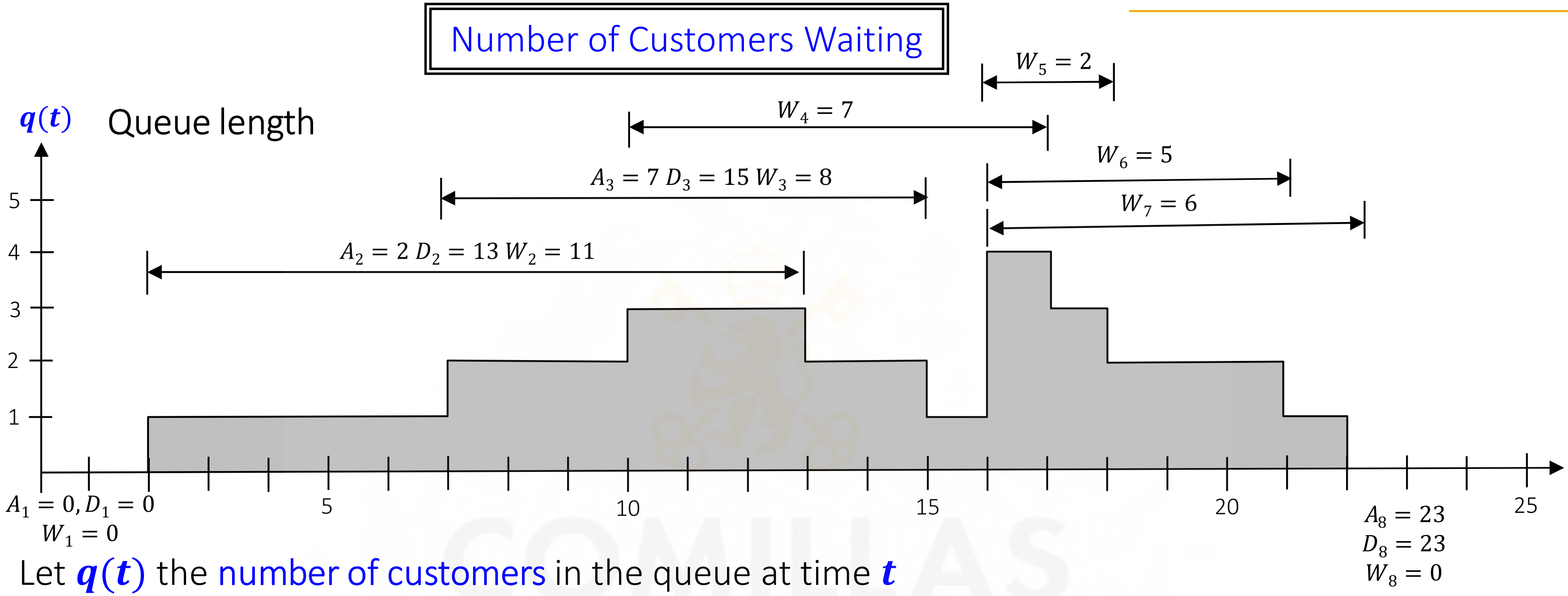
- Let A_i the time that the i-th customer enters the queue
- Let D_i the time that the i-th customer exits the queue
- Let $W_i = Di - Ai$ the time that the i-th customer spends in the queue

○ The W_i is an observation-based data and, once observed, the value never changes again concerning time

○ Observational data are often associated with an entity moving through states of the simulation model. The observation is available when the entity enters and subsequently exits the state

○ Other statistical quantities, such as minimum, maximum, and variance, can also be computed

Types of Statistical Variables: Example (II)



Let $\boldsymbol{q(t)}$ the number of customers in the queue at time $\boldsymbol{t}$

The $\boldsymbol{q(t)}$ is a time-based function whose value changes concerning time

However, its value takes constant values during intervals of time corresponding to when the queue has a certain number of customers

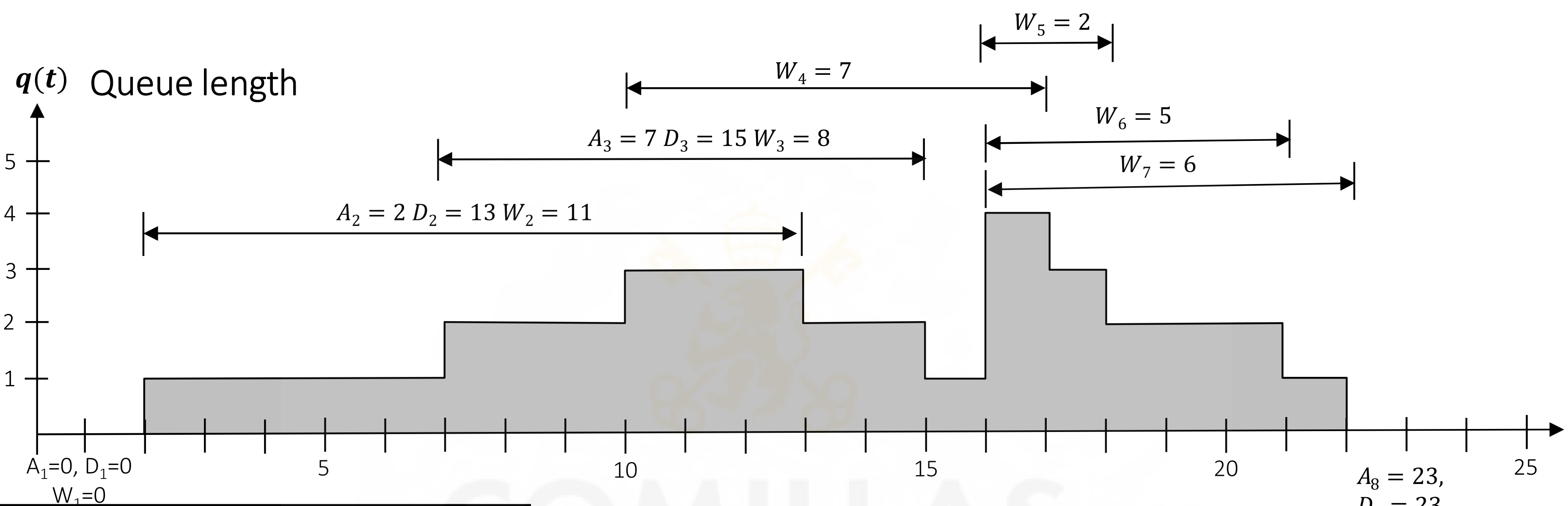
To estimate the percentage of time that the variable takes a particular value, $\boldsymbol{p_i}$, is computed as the proportion of time that the queue has $\boldsymbol{i}$ customers during the simulation time

$$\overline{L}_q(n) = \frac{\int_{t_0}^{t_n} q(t)dt}{t_n - t_0} = \sum_{k=1}^{n} \frac{q_k(t_k - t_{k-1})}{t_n - t_0}$$

$$p_i = \frac{T_i}{t_n - t_0}$$

Where $\boldsymbol{T_i}$ is the time where the queue has i length

Types of Statistical Variables: Example (III)



$$p_0 = \frac{T_0}{T} = \frac{(2-0)+(25-22)}{25} = 0.2$$

$$p_1 = \frac{T_1}{T} = \frac{(7-2)+(16-15)+(22-21)}{25} = 0.28$$

$$p_2 = 0.32;\ p_3 = 0.16;\ p_4 = 0.04$$

$$\sum_{i=0}^{4} p_i = 1$$

Average waiting time

$$\overline{W}(8) = \frac{\sum_{i=1}^{n} W_i}{n} = \frac{0+11+8+7+2+5+6+0}{8} = \frac{39}{8} = 4.875$$

Average number of customers waiting

$$\overline{L}_q(n) = \frac{0(2-0)+1(7-2)+2(10-7)+3(13-10)+2(15-13)+1(16-15)}{25} + \frac{4(17-16)+3(18-17)+2(21-18)+1(22-21)+0(25-22)}{25} = \frac{39}{25} = 1.56$$

Example in Arena
How to obtain these variables of interest in ARENA?

LOTRExample.doe

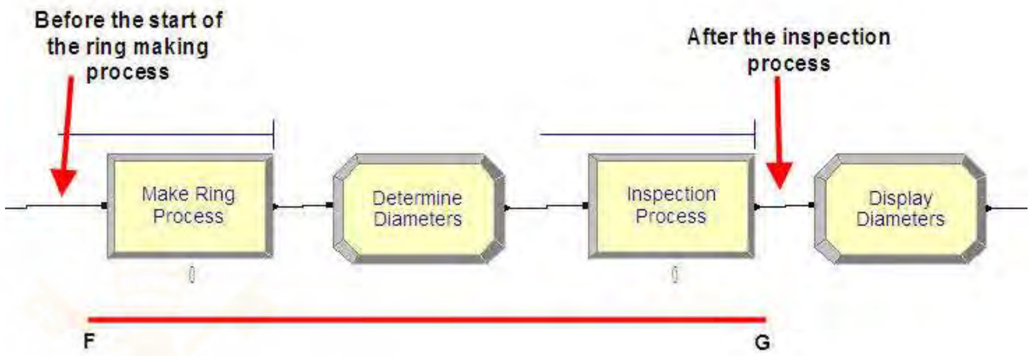
In the LOTR Ring Maker model, there are two random variables (in the "Ring Processing" part of the model) of interest whose mean value and confidence interval should be computed:

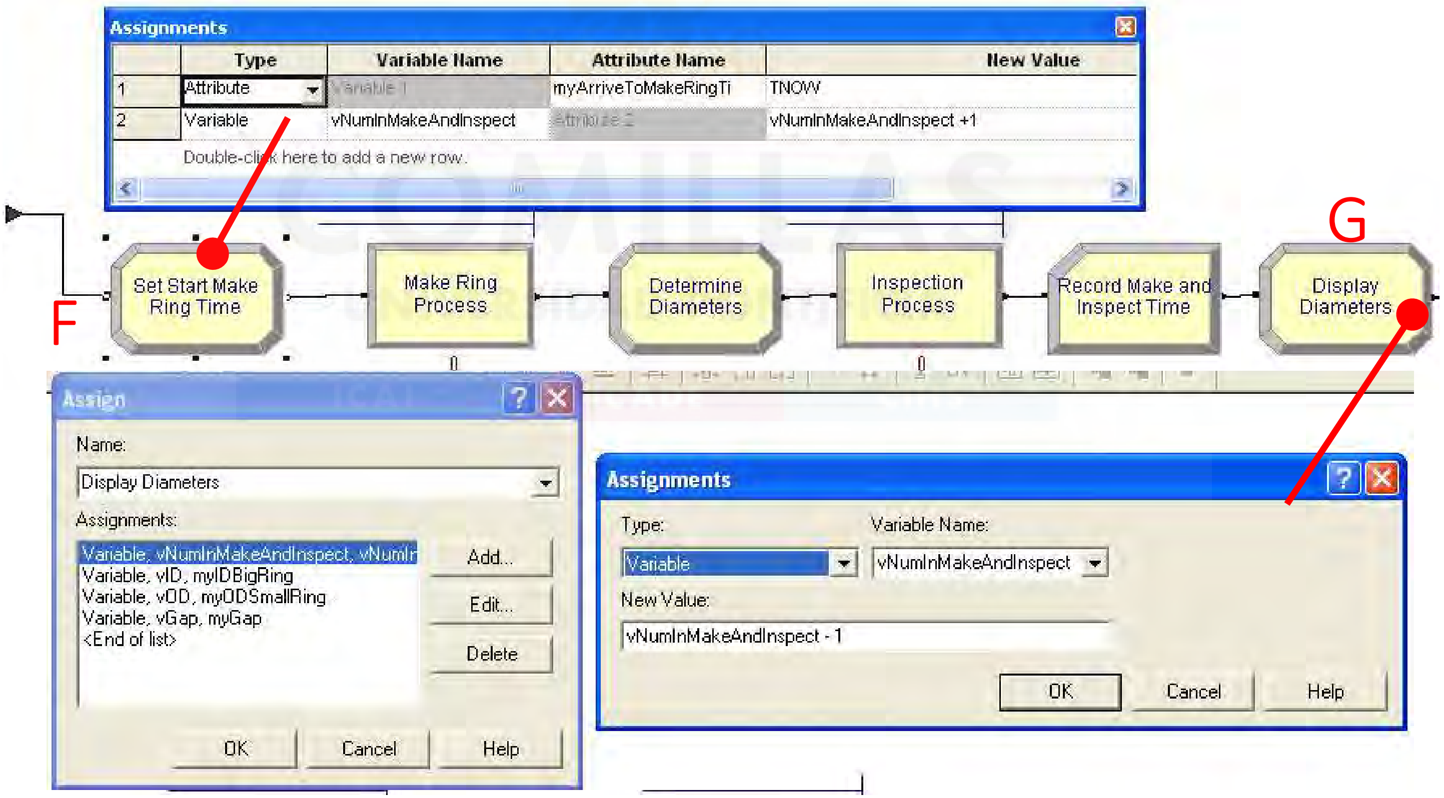
A) The average number of pairs of rings in both the ring-making and the ring inspection process (joint average)

B) The average time that it takes for a pair of rings to go through the ring-making and the ring-inspection processes. A 95% confidence interval for the mean time within ± 20 min.

Example in Arena
How to obtain variable of interest A) in ARENA

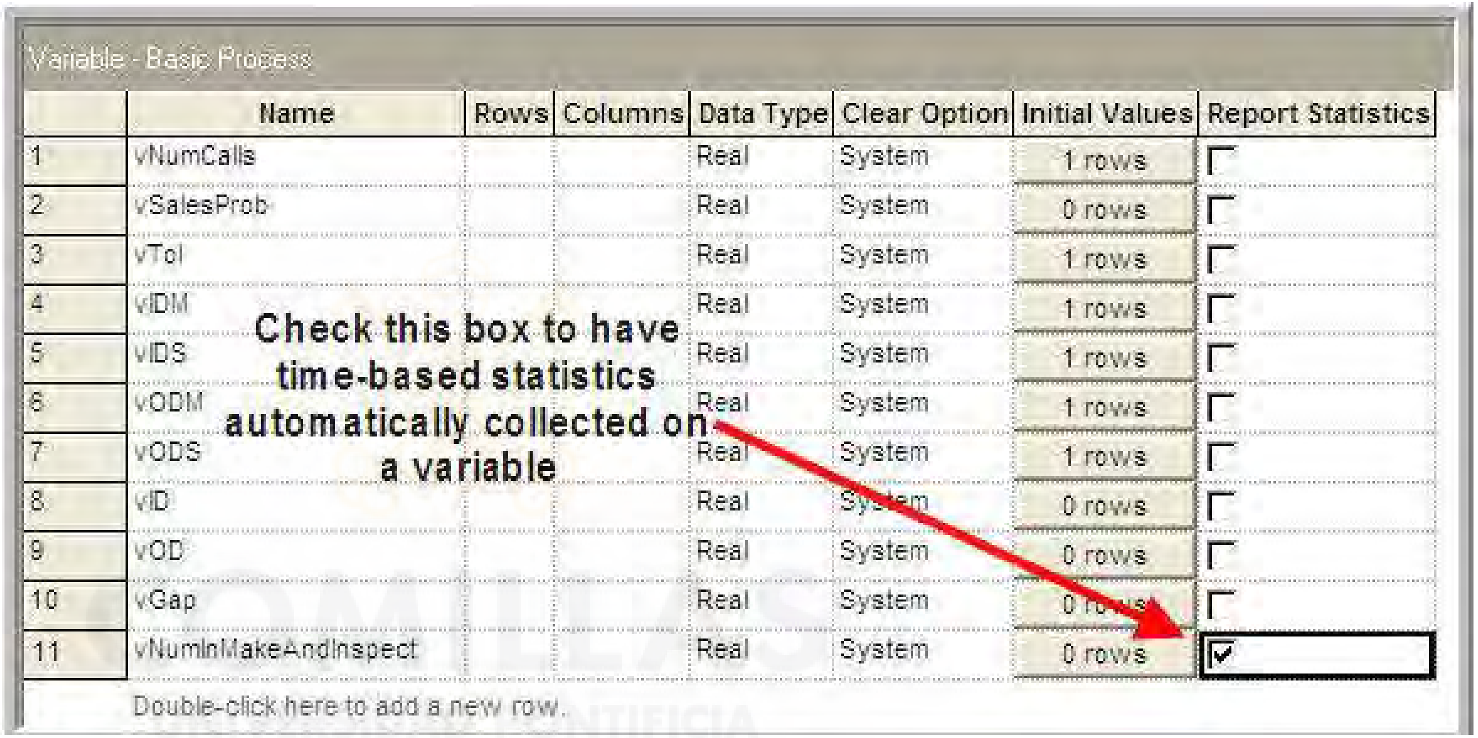
To obtain the average WIP of rings between F and G process points:
Time-based variable
vNumInMakeAndInspect





Example in Arena
How to obtain variable of interest A) in ARENA

How to collect statistics on a Time-based variable?



Variable - Basic Process

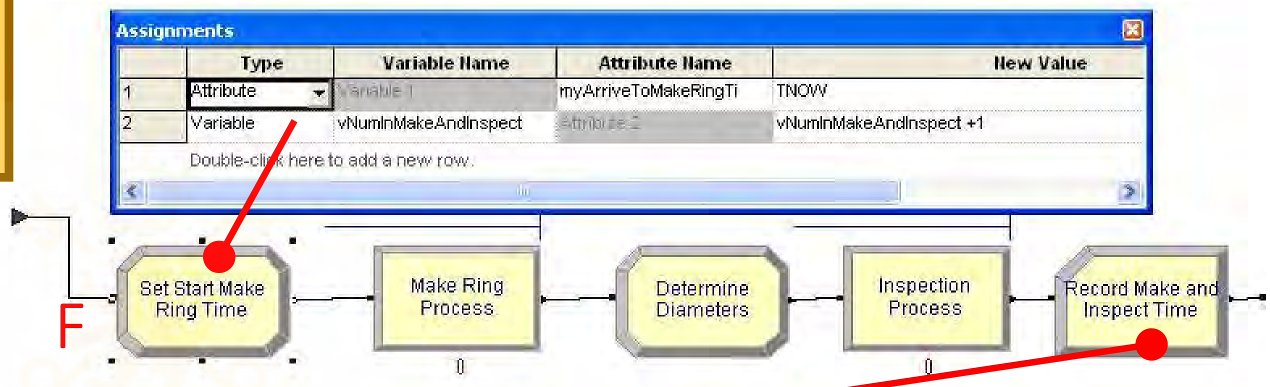
	Name	Rows	Columns	Data Type	Clear Option	Initial Values	Report Statistics
1	vNumCalls			Real	System	1 rows	☐
2	vSalesProb			Real	System	0 rows	☐
3	vTol			Real	System	1 rows	☐
4	vIDM			Real	System	1 rows	☐
5	vIDS			Real	System	1 rows	☐
6	vODM			Real	System	1 rows	☐
7	vODS			Real	System	1 rows	☐
8	vID			Real	System	0 rows	☐
9	vOD			Real	System	0 rows	☐
10	vGap			Real	System	0 rows	☐
11	vNumInMakeAndInspect			Real	System	0 rows	☑

Double-click here to add a new row.

The average number of pairs of rings in both the ring-making and the ring-inspection process

Example in Arena
How to obtain variable of interest B) in ARENA

To collect observation-based statistics, it is necessary to place RECORD modules at appropriate locations in the model

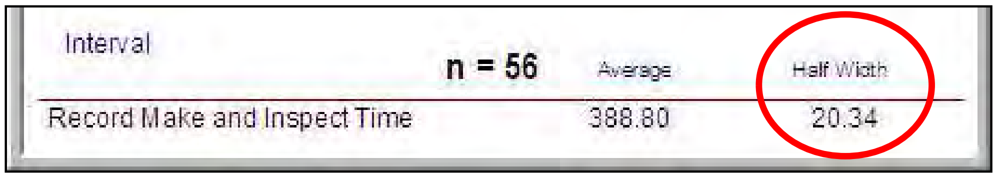


When the entity passes the point F, there is an assignment:
myArriveToMakeRingTime =TNOW

When the entity passes the RECORD module (of Type: Time Interval):
The module records the difference between the current time, TNOW, and the specified attribute (*myArriveToMakeRingTime)*

This represents the elapsed time interval for the entity to move from F to the RECORD.

After sampling the required number of replications of the new average time obtained follows:



Example in Arena
How to obtain variables of interest in ARENA

There are five RECORD types:

Count: A counter is defined, and it will be incremented or decremented by the value specified

Entity Statistics: This option will record entity statistics

Time Interval: The difference between the current time, TNOW, and the specified attribute

Time Between: The difference between the current time, TNOW, and the last time an entity passed through this RECORD module

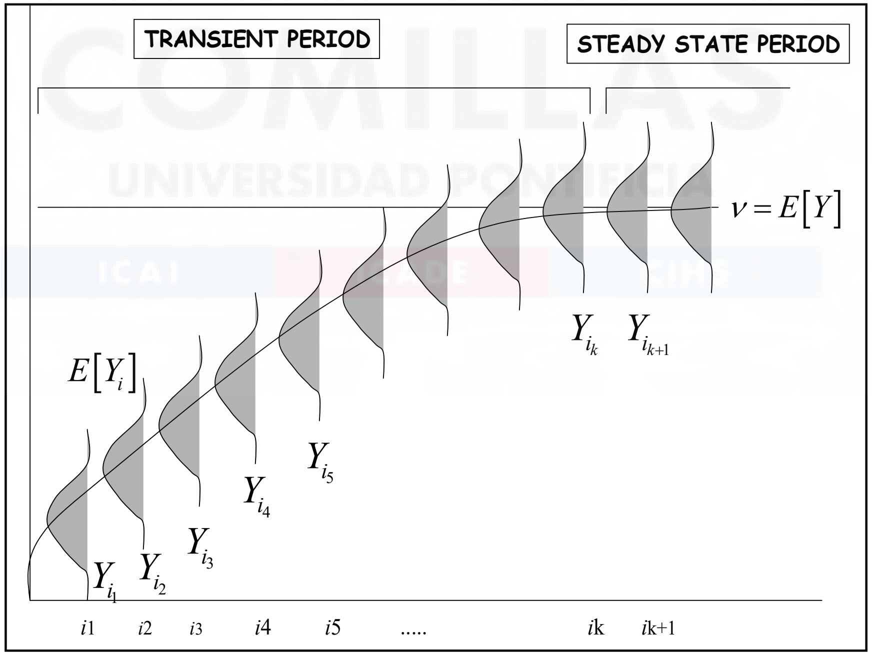
Expression: It allows statistics to be collected on a general expression when the entity passes the module

4ii

Types of Simulation concerning Output Analysis

Types of Simulation concerning Output Analysis

- When planning the experimental analysis, it is helpful to think of simulations as consisting of two main categories related to the period over which a decision needs to be made:
  - Finite horizon: A well-defined ending time or condition can be specified (*terminating* simulations). They are focused on the *transient* period.
  - Infinite horizon: There is no well-defined ending time or condition. The planning period is over the life of the system. They are also called *steady-state* simulations.



Types of Simulation concerning Output Analysis

- Examples of Finite horizon: Its ending time of each replication is specified or a random variable (for example, when no entities in the model)
  - Bank: doors open at 9 a.m. and close at 5 p.m.
  - Military battle: Simulate till force strength reaches a critical value
  - Filling a customer order: Simulate the production of the first 100 products

- Examples of Infinite horizon: There is no natural ending point. However, a finite replication length must be specified
  - A factory in which you are interested in measuring the steady-state throughput
  - A hospital emergency room that is open 24 hours a day, 7 days a week
  - A telecommunications system that is always operational

- Infinite-horizon simulations often model situations that involve modeling of non-stationary processes. It is often possible to find a period where the non-stationary behavior repeats (steady state cyclical estimation, Law (2007))

- Of the two types of simulations, Finite-Horizon simulations are easier to analyze

4iii

Analysis of Finite-Horizon Simulations

Analysis of Finite-Horizon Simulations

• The analysis of the system will require some replications to obtain an output result for decision-making with a confidence interval half-width

Arena computes a 95% confidence interval
and reports the half-width value for that interval

• To get that interval size, it is necessary to estimate in advance the sample size, i.e., the number of replications of the experiment

• **For example**: Estimating $E[W_r]$ it is possible to be 95% confident that the average of the random variable is the true value within ± 2 minutes

• In practice, this half-width ratio method is used to estimate the adequate number of replications n to achieve the desired half-width h:

- Set an initial pilot run with n_0 replications and compute the half-width h_0 (corresponding to this initial run). Then n is approximated by

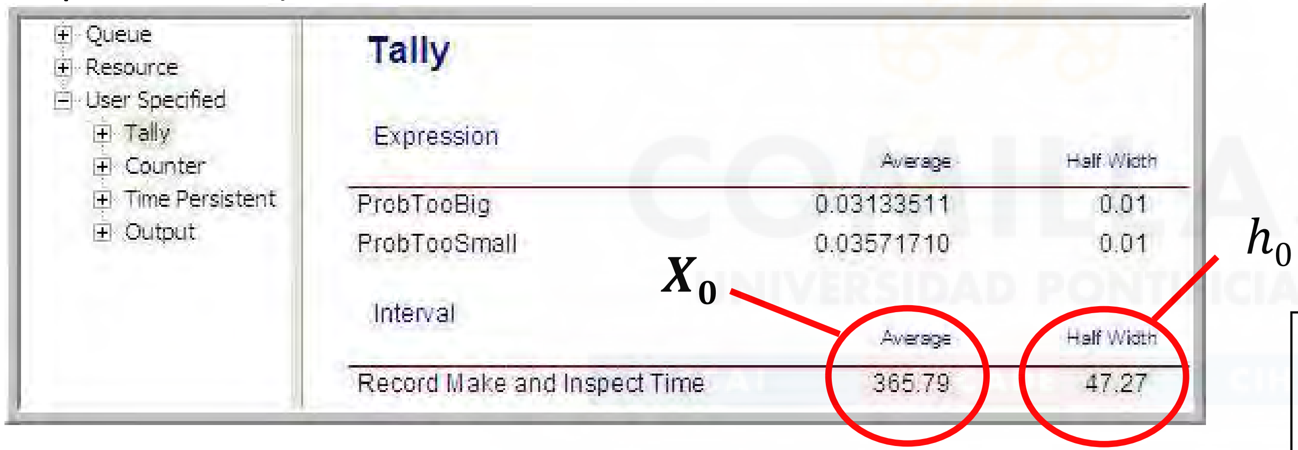
Quadratic behavior

$$n \approx n_0 \left(\frac{h_0}{h} \right)^2$$

Example in Arena
How to estimate the number of replications n

- How to estimate the number of replications n to ensure a 95% confidence interval with an error bound of ± 20 min (i.e., half width of 20 minutes) for variable B.

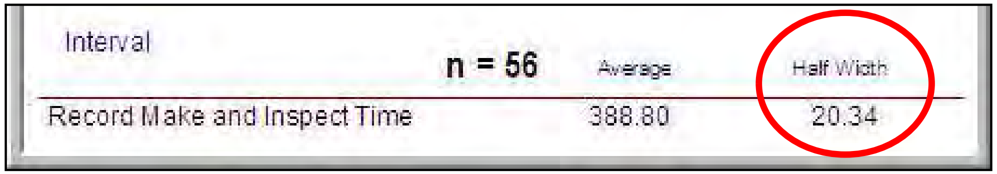
- Run the model with the initial number of replications n_0 (we choose this number, e.g., n_0 =10) to obtain the corresponding initial half-width h_0 (Run>Setup>Number of Replications)



Number of replications to get 20 min accuracy using half-width ratio method

$$n \approx n_0 \left(\frac{h_0}{h} \right)^2 = 10 \left(\frac{47.27}{20} \right)^2 = 56$$

After sampling the required number of replications, the new average time obtained as follows:



i. Types of Statistical Variables
ii. Types of Simulation concerning Output Analysis
iii. Analysis of Finite-Horizon Simulations
iv. Analysis of Infinite-Horizon Simulations

4iv

Analysis of Infinite-Horizon Simulations

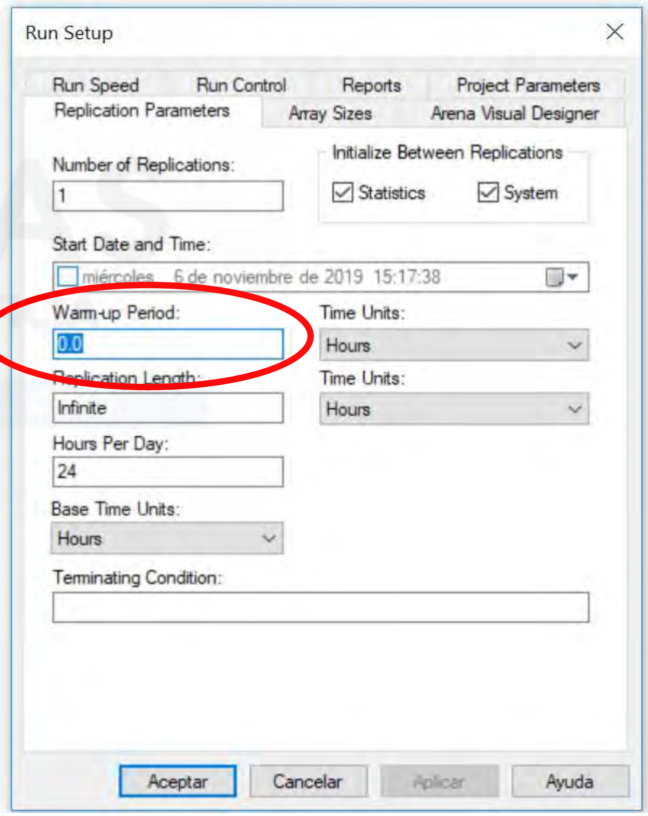
Analysis of Infinite-Horizon Simulations

- The initial conditions of the system impact model results and statistics and can distort the actual value of those statistics.

- Analyzing the effect of initial conditions mitigates the nonstationary aspect of within-replication data for infinite-horizon experiments.

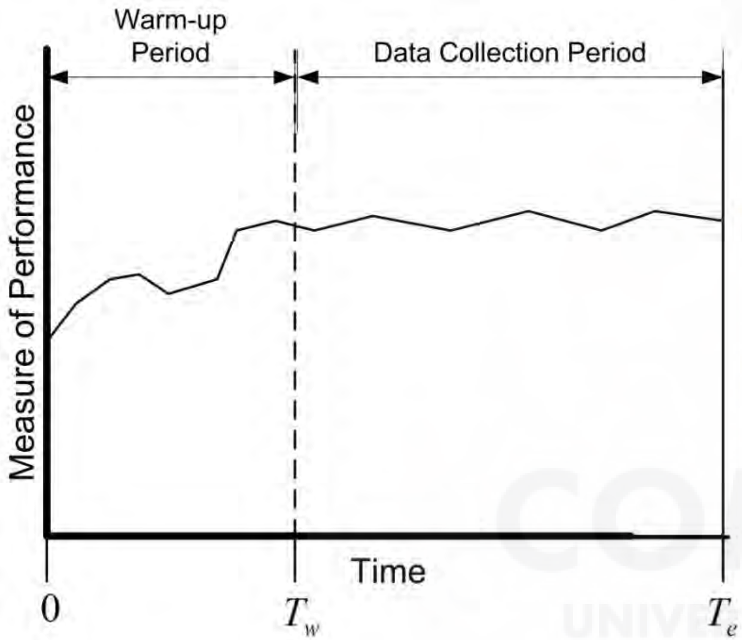
- A warm-up time (the time it takes the system to reach a steady state) must be determined within the simulation tool. In ARENA: (Run>Setup>Warm-up Period)

- Specifying a warm-up period in Arena causes to schedule a warm-up event for time T_w. At that time, all the accumulated statistical counters are cleared so that the net effect is that statistics are only collected over the period from T_w to T_e



Analysis of Infinite-Horizon Simulations

Assessing the Effect of Initial Conditions



- Let $F(x/I)$ be the conditional cumulative distribution function of X where I represents the initial conditions to start the simulation at time 0

- If $F(x/I) \to F(x)$ when $t \to \infty$ for all initial conditions, then $F(x)$ is called the steady-state distribution of the output process

- Unless the system is initialized using the steady-state distribution, there is no way to observe the steady-state distribution directly

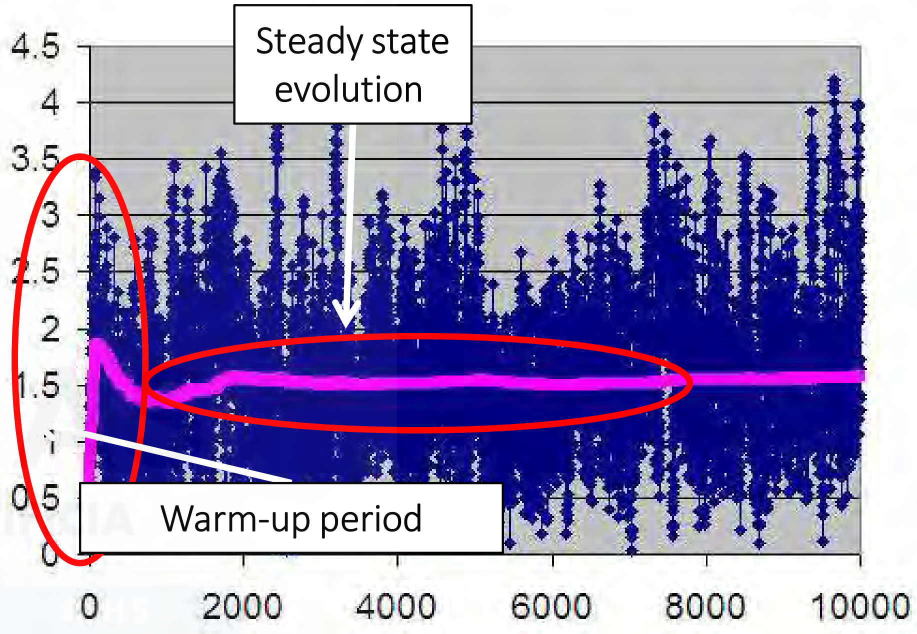
- It should be decided on how long to run the simulations and how to handle the effect of the initial conditions on the estimates

- Rule of thumb: the simulation length should be at least 10 times the warm-up period

Analysis of Infinite-Horizon Simulations

Warm-up period (cont.):

- To determine the warm-up period, a visual method proposed by Welch (1983) consists of:



1. Make R replications
2. Let Y_{ri} be the i-th observation within replication r
3. Compute the averages across replications
4. Plot $\overline{Y_{\blacksquare i}}$ for each i
5. Apply smoothing techniques to this plot
6. Visually assess where the plots start to converge

$$\overline{Y}_{\bullet i} = \frac{\sum_{r=1}^{R} Y_{ri}}{R}$$

Example: On sheet *10Replications* at Excel file *MM1-QueueingSimulation.xls*

Pedro Sánchez (Pedro.Sanchez@comillas.edu)
Andrés Ramos (Andres.Ramos@comillas.edu)
Eugenio Sánchez (Eugenio.Sanchez@comillas.edu)
Sonja Wogrin(Sonja.Wogrin@comillas.edu)