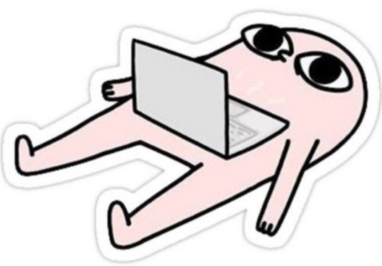
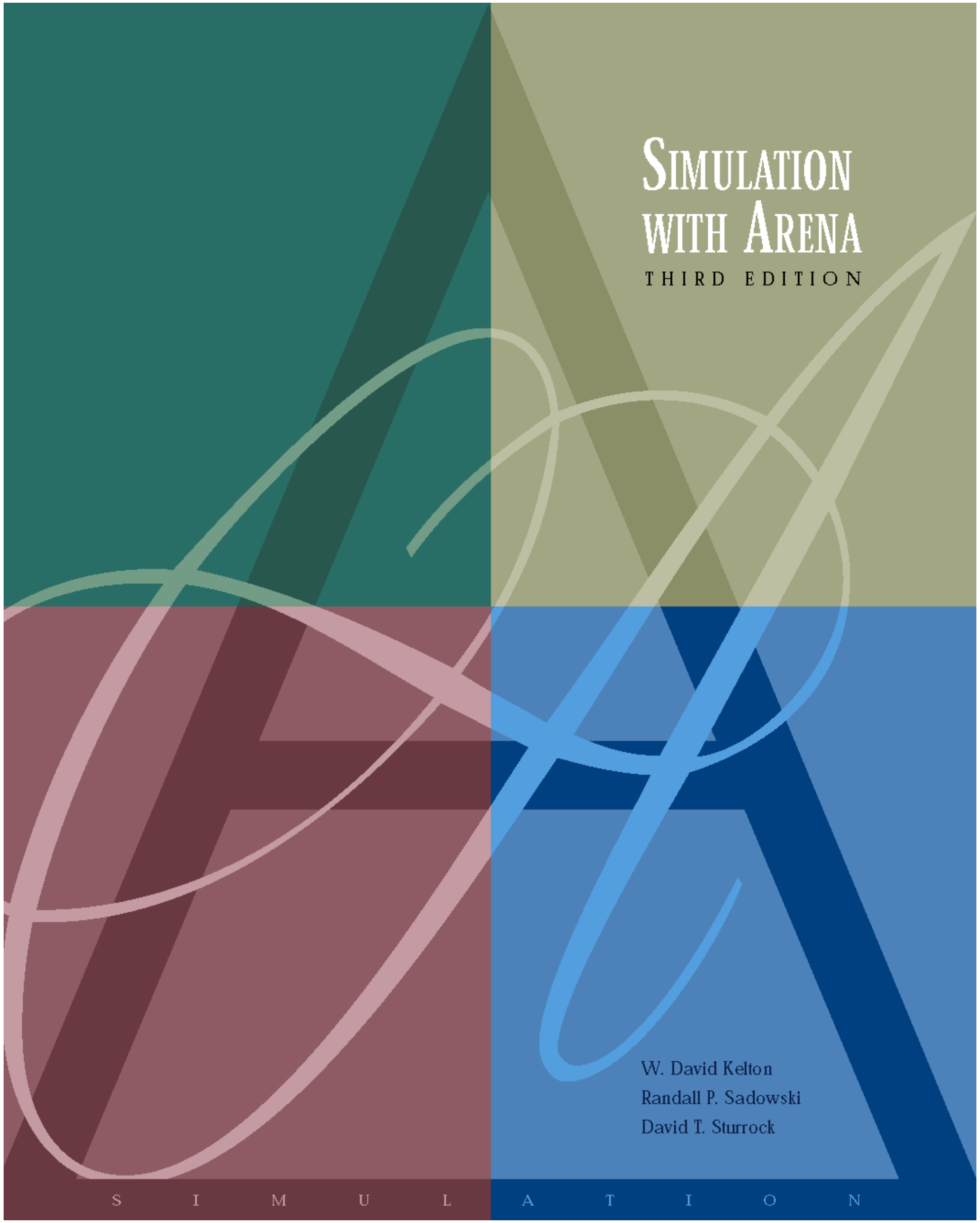
Simulation Slides With Notes

Dr Mahmoud Barghash
(1st Semester 2024/2025)
Notes are written by Nada Ababneh





Fundamental Simulation Concepts

Chapter 2

* This course in general is about discrete event simulation

Last revision June 7, 2003

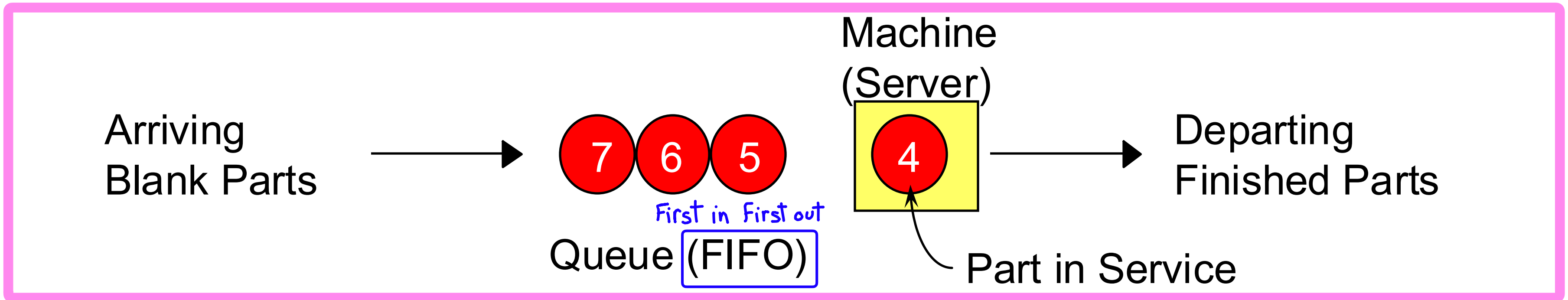
first to download (ARENA) write in Google:

What We'll Do ...

Download Arena Rockwell

- **Underlying ideas, methods, and issues in simulation**
- **Software-independent (setting up for Arena)**
- **Centered around an example of a simple processing system**
  - Decompose the problem
  - Terminology
  - Simulation by hand
  - Some basic statistical issues
  - Overview of a simulation study

The System:
A Simple Processing System



- **General intent:**
  - Estimate expected production
  - Waiting time in queue, queue length, proportion of time machine is busy
- **Time units**
  - Can use different units in different places … must declare
  - Be careful to check the units when specifying inputs
  - Declare *base time units* for internal calculations, outputs
  - Be reasonable (interpretation, roundoff error)

Model Specifics

- **Initially (time 0) empty and idle**
- **Base time units: minutes**
- **Input data (assume given for now …), in minutes:**

(Data Collected)

الوقت بين قدوم الشخص و قدوم الشخص يلي بعده

Departure

Departure time

commulative ال

Part Number	Arrival Time	Interarrival Time	Service Time
1	0.00	1.73	2.90
2	1.73	1.35	1.76
3	3.08	0.71	3.39
4	3.79	0.62	4.52
5	4.41	14.28	4.46
6	18.69	0.70	4.36
7	19.39	15.52	2.07
8	34.91	3.15	3.36
9	38.06	1.76	2.37
10	39.82	1.00	5.38
11	40.82	.	.
.	.	.	.
.	.	.	.

- **Stop when 20 minutes of (simulated) time have passed**

The End

Goals of the Study: Output Performance Measures

- ***Total production*** **of parts over the run (*P*)**
- ***Average waiting time*** **of parts in queue:**

$$\frac{\sum_{i=1}^{N} WQ_i}{N}$$

N = no. of parts completing queue wait
WQ_i = waiting time in queue of ith part
Know: $\boxed{WQ_1 = 0 \text{ (why?)}}$ أول واحد بالعادة ما بيستنى (دغري بيدخل على الProcess)
$N \geq 1$ (why?)

- ***Maximum waiting time*** **of parts in queue:**

$$\max_{i=1,\ldots,N} WQ_i$$

Goals of the Study: Output Performance Measures (cont'd.)

- ***Time-average number of parts in queue***:

$$\frac{\int_0^{20} Q(t)\,dt}{20}$$

$Q(t)$ = number of parts in queue at time t

- ***Maximum number of parts in queue***: $\max\limits_{0 \le t \le 20} Q(t)$
- ***Average*** and ***maximum total time in system*** of parts (a.k.a. ***cycle time***):

$$\frac{\sum_{i=1}^{P} TS_i}{P}, \quad \max_{i=1,\ldots,P} TS_i$$

TS_i = time in system of part i

Goals of the Study: Output Performance Measures (cont'd.)

- ***Utilization*** **of the machine (proportion of time busy)**

$$\frac{\int_0^{20} B(t)\,dt}{20}, \quad B(t) = \begin{cases} 1 & \text{if the machine is busy at time } t \\ 0 & \text{if the machine is idle at time } t \end{cases}$$

- **Many others possible (information overload?)**

Analysis Options

- **Educated guessing**
  - Average interarrival time = 4.08 minutes
  - Average service time = 3.46 minutes
  - So (on average) parts are being processed faster than they arrive
    - System has a chance of operating in a stable way in the long run, i.e., might not "explode"
    - If all interarrivals and service times were exactly at their mean, there would never be a queue
    - But the data clearly exhibit variability, so a queue could form
  - If we'd had average interarrival < average service time, and this persisted, then queue would explode
  - Truth — between these extremes
  - Guessing has its limits …

Analysis Options (cont'd.)

- **Queueing theory**
  - Requires additional assumptions about the model
  - Popular, simple model: *M/M/1 queue*
    - Interarrival times ~ exponential
    - Service times ~ exponential, indep. of interarrivals
    - Must have E(service) < E(interarrival)
    - Steady-state (long-run, forever)
    - Exact analytic results; e.g., average waiting time in queue is

$$\frac{\mu_S^2}{\mu_A - \mu_S}, \qquad \begin{aligned} \mu_A &= \text{E(interarrival time)} \\ \mu_S &= \text{E(service time)} \end{aligned}$$

  - Problems: validity, estimating means, time frame
  - Often useful as first-cut approximation

Mechanistic Simulation

- **Individual operations (arrivals, service times) will occur exactly as in reality**
- **Movements, changes occur at the right "time," in the right order**
- **Different pieces interact**
- **Install "observers" to get output performance measures**
- **Concrete, "brute-force" analysis approach**
- **Nothing mysterious or subtle**
  - But a lot of details, bookkeeping
  - Simulation software keeps track of things for you

Pieces of a Simulation Model

- ***Entities*** الاشياء يلي بتتحرك جوا ال system
  - "Players" that move around, change status, affect and are affected by other entities
  - *Dynamic objects* — get created, move around, leave (maybe)
  - Usually represent "real" things
    - Our model: entities are the parts
  - Can have "fake" entities for modeling "tricks"
    - Breakdown demon, break angel
      - Though Arena has built-in ways to model these examples directly
  - Usually have multiple *realizations* floating around
  - Can have different types of entities concurrently
  - Usually, identifying the types of entities is the first thing to do in building a model

Pieces of a Simulation Model (cont'd.)

- ***Attributes***

الشيء الذي يختلف من Entity
ءالى Entity

  - Characteristic of all entities: describe, differentiate
  - All entities have same attribute "slots" but different values for different entities, for example:
    - Time of arrival
    - Due date
    - Priority
    - Color
  - Attribute value tied to a specific entity
  - Like "local" (to entities) variables
  - Some automatic in Arena, some you define

Pieces of a Simulation Model (cont'd.)

- (Global) ***Variables*** كل ال entities جوابهم نفس الاشي
  - Reflects a characteristic of the whole model, not of specific entities
  - Used for many different kinds of things
    - Travel time between all station pairs
    - Number of parts in system
    - Simulation clock (built-in Arena variable)
  - Name, value of which there's only one copy for the whole model
  - Not tied to entities
  - Entities can access, change variables
  - Writing on the wall (rewriteable)
  - Some built-in by Arena, you can define others

Pieces of a Simulation Model (cont'd.)

- ***Resources***
  - What entities compete for
    - People
    - Equipment
    - Space
  - Entity *seizes* a resource, uses it, *releases* it
  - Think of a *resource being assigned to an entity*, rather than an entity "belonging to" a resource
  - "A" resource can have several *units* of capacity
    - Seats at a table in a restaurant
    - Identical ticketing agents at an airline counter
  - Number of units of resource can be changed during the simulation

Pieces of a Simulation Model (cont'd.)

- ***Queues***
  - Place for entities to wait when they can't move on (maybe since the resource they want to seize is not available)
  - Have names, often tied to a corresponding resource
  - Can have a finite capacity to model limited space — have to model what to do if an entity shows up to a queue that's already full
  - Usually watch the length of a queue, waiting time in it

Pieces of a Simulation Model (cont'd.)

- ***Statistical accumulators***
  - Variables that "watch" what's happening
  - Depend on output performance measures desired
  - "Passive" in model — don't participate, just watch
  - Many are automatic in Arena, but some you may have to set up and maintain during the simulation
  - At end of simulation, used to compute final output performance measures

Pieces of a Simulation Model (cont'd.)

- **Statistical accumulators for the simple processing system**
  - Number of parts produced so far
  - Total of the waiting times spent in queue so far
  - No. of parts that have gone through the queue
  - Max time in queue we've seen so far
  - Total of times spent in system
  - Max time in system we've seen so far
  - Area so far under queue-length curve $Q(t)$
  - Max of $Q(t)$ so far
  - Area so far under server-busy curve $B(t)$

Simulation Dynamics: The Event-Scheduling "World View"

- **Identify characteristic *events***
- **Decide on *logic* for each type of event to**
  - Effect *state changes* for each event type
  - Observe statistics
  - Update times of future events (maybe of this type, other types)
- **Keep a simulation *clock*, future *event calendar***
- ***Jump* from one event to the next, process, observe statistics, update event calendar**
- **Must specify an appropriate *stopping rule***
- **Usually done with general-purpose programming language (C, FORTRAN, etc.)**

Events for the Simple Processing System

- ***Arrival*** **of a new part to the system**
  - Update time-persistent statistical accumulators (from last event to now)
    - Area under $Q(t)$
    - Max of $Q(t)$
    - Area under $B(t)$
  - "Mark" arriving part with current time (use later)
  - If machine is idle:
    - Start processing (schedule departure), Make machine busy, Tally waiting time in queue (0)
  - Else (machine is busy):
    - Put part at end of queue, increase queue-length variable
  - Schedule the next arrival event

Events for the Simple Processing System (cont'd.)

- ***Departure*** **(when a service is completed)**
  - Increment number-produced stat accumulator
  - Compute & tally time in system (now - time of arrival)
  - Update time-persistent statistics (as in arrival event)
  - If queue is non-empty:
    - Take first part out of queue, compute & tally its waiting time in queue, begin service (schedule departure event)
  - Else (queue is empty):
    - Make the machine idle (Note: there will be no departure event scheduled on the future events calendar, which is as desired)

Events for the Simple Processing System (cont'd.)

- ***The End***
  - Update time-persistent statistics (to end of the simulation)
  - Compute final output performance measures using current (= final) values of statistical accumulators
- **After each event, the event calendar's top record is removed to see what time it is, what to do**
- **Also must initialize everything**

Some Additional Specifics for the Simple Processing System

- **Simulation clock variable (internal in Arena)**
- **Event calendar: list of event *records*:**
  - [Entity No., Event Time, Event Type]
  - Keep *ranked* in increasing order on Event Time
  - Next event always in top record
  - Initially, schedule first Arrival, The End (Dep.?)
- **State variables: describe current status**
  - Server status $B(t)$ = 1 for busy, 0 for idle
  - Number of customers in queue $Q(t)$
  - Times of arrival of each customer now in queue (a list of random length)

Output ⟶ اما continuous / او discrete

* Note :-

اذا كان ال output عبارة عن Assumption of time (يعني يتغير مع الوقت) ← Continuous

و يوجد تكامل

مثل (عدد ال Parts in Que) و (عدد ال Parts in system)

اذا كان ال output يختلف حسب ال Entity ← Discrete

(Not Related to time)

(مثلاً لما أشوف كل Entity كم انتظرت وقت وكل وحدة جوابها مختلف عن الثانية وهكذا)

* طبيعة اسئلة ال (Hand simulation)

- ارسم ال curve
- ارسم جزء من ال curve و اعمل حسابات
- مرسوم جزء و احنا نكمل الجزء الضايل ب ال curve
- Average waiting time in Que
- ⋮

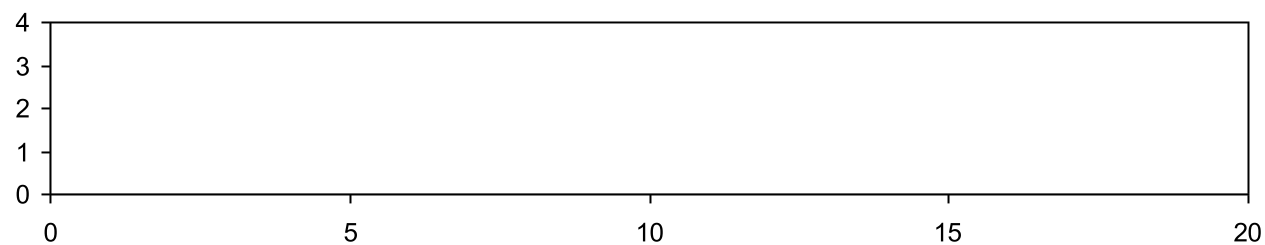
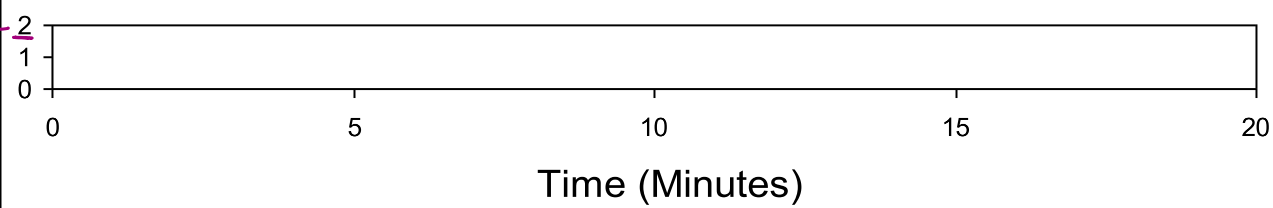
Simulation by Hand

- Manually track state variables, statistical accumulators
- Use "given" interarrival, service times
- Keep track of event calendar
- "Lurch" clock from one event to the next
- Will omit times in system, "max" computations here (see text for complete details)

جاي سؤال على ال
Hand Simulation
في الميد و الفاينل

Simulation by Hand: Setup

<table>
<tr><td>System</td><td>Clock</td><td>B(t)
the variable related to the worker or machine
either → 1 (working) / 0 (Not working)</td><td>Q(t) continuous
Que Size at any time</td><td colspan="2">Arrival times of custs. in queue</td><td colspan="2">Event calendar</td></tr>
<tr><td>Number of completed waiting times in queue</td><td colspan="2">Total of waiting times in queue</td><td colspan="3">Area under Q(t)</td><td colspan="2">Area under B(t)</td></tr>
<tr><td>Q(t) graph

B(t) graph
if two machines are busy
Either → 1 / 0
(Shows if the machine is working or Not)</td><td colspan="7"></td></tr>
<tr><td>Interarrival times
الوقت بين entity و entity</td><td colspan="7">1.73, 1.35, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times
متى بصير Departure</td><td colspan="7">2.90, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

*Hand Simulation Starts at time Zero

(Service time) نفسه ال
و بنزل فيه Commulative

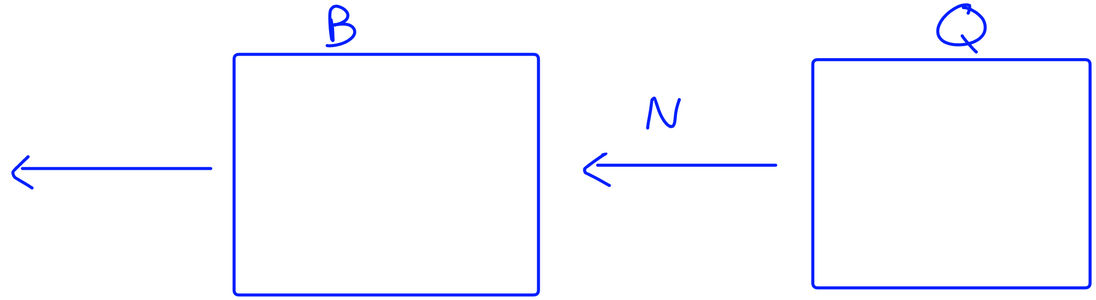
the first Arrival for entity one is zero
إلا اذا حكى السؤال اشي ثاني

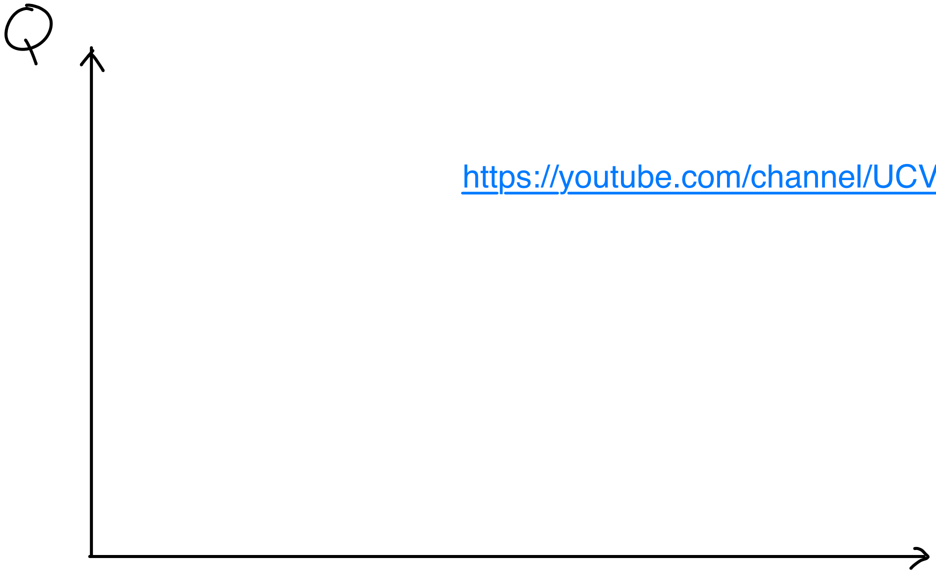
بطرح وبحط قيمة مطلقة

Entity#	Arrival	Departure	Waiting time in Que (W_Q)	Total time in system (T_s)
1	0	2.9	0	0 − 2.9 = 2.9
2	zero + 1.73 = 1.73	2.9 + 1.76 = 4.66	1.73 − 2.9 = بهمل السالب 1.17	1.73 − 4.66 = 2.93
3	1.73 + 1.35 = 3.08	4.66 + 3.39 = 8.05	3.08 − 4.66 = 1.58	3.08 − 8.05 = 4.97
4	3.08 + 0.71 = 3.79	12.57	4.26	8.78
5	3.79 + 0.62 = 4.41	17.03	8.16	12.62
6	18.64	21.39	0 لأنه الجواب (positive) لم يحصل انتظار	
7	19.39			
8	34.41			
9				

$W_{Q(n)} =$ Arrival time (n) − Departure time (n−1)

$T_{s(n)} =$ Arrival time (n) − Departure time (n)





https://youtube.com/channel/UCVVK0WemNQzrzUsQQZk--BQ?si=rPkGVPSun8fi_aCG

شرح لـ Hand Simulation

اي اشي بدخل جوا ال Process (B) بحسبله departure

ما بحسب dep الثالثة الا لما تدخل ال Process (B)

Just Finished Event			Variables		Attributes (arrival Times) When the last entity enters the:		Statistical Accumulators (P) لما تطلع من									Next Event Calendar		
Entity No	Time	Event Type	Q(t)	B(t)	In Queue	In Service	N	P	Total WQ (بحسبه لل entity لما تطلع من (Q))	WQ* Max Waiting Time	Total Time in System ΣT_s	T_s^*	Integrate $\int Q(t)\,dt$	Q^*	$\int B(t)\,dt$	Entity No (متى بصير Departure و متى بصير Arrival للـ entity يلي بعدها)	Time	Type
—	0	Entity	0	0	0	0	0	0	0	0	0		0	0	0	1 —	اقل وقت 0 20	Arr end
1	0	Arr	0	1			1	0	0	0	0		0	0	0	1 2	2.9 1.73	dep arr end
2	1.73	arr	1	1			0	0	0	0	0		0	1	1.73	1 3 —	20 2.9 3.08 20	dep arr end دائماً
1	2.9	dep	1	1			2	1	0+1.17 1.17	Max(1,2) 1.17	2.9	مين طلع بال P 2.9	1.17	1	2.9	2 3	4.66 (3.08)	dep arr
3	3.08	arr	1	1			2	1	1.17	1.17	2.9	2.9	1.17	1	3.08	2 4	4.66 (3.79)	dep arr
4	3.79	arr	2	1			2	1	1.17	1.17	2.9	2.9	1.88	2	3.79	2 5	4.66 (4.41)	dep arr
5	4.41	arr	3	1			2	1	1.17	1.17	2.9	2.9	3.12	3	4.41	2 6	(4.66) 18.69	dep arr
2	4.66	dep	2	1			3	2	for 1,2,3 ⊕ 2.75	Max 1,2,3 1.58	1,2 ⊕ 5.83	Max 1,2 2.93	3.87	3	4.66	3 6	(8.05) 18.69	dep arr
3	8.05	dep	1	1			4	3	7.01	4.26	10.8	4.97	10.65	3	8.05	4 6	(12.57) 18.69	dep arr
4	12.57	dep	0	1			5	4	15.17	8.16	19.58	8.78	15.17	3	12.57	5 6	(17.03) 18.69	dep arr
5	17.03	dep	0	0			5	5	15.17	8.16	32.2	12.62	15.17	3	17.03	6 —	(18.69) 20	arr end
6	18.69	arr	0	1			6	5	15.17	8.16	32.2	12.62	15.17	3	17.03	6 7 —	21.39 (19.39) 20	dep ar end
7	19.39	arr	1	1			6	5	15.17	8.16	32.2	12.62	15.17	3	17.73	6 8 —	21.39 34.91 (20)	dep arr end
—	20	END	1	1			6	5	15.17	8.16	32.2	12.62	15.78	3	18.34			

At time zero
the first product Arrives
Person

Simulation by Hand: t = 0.00, Initialize

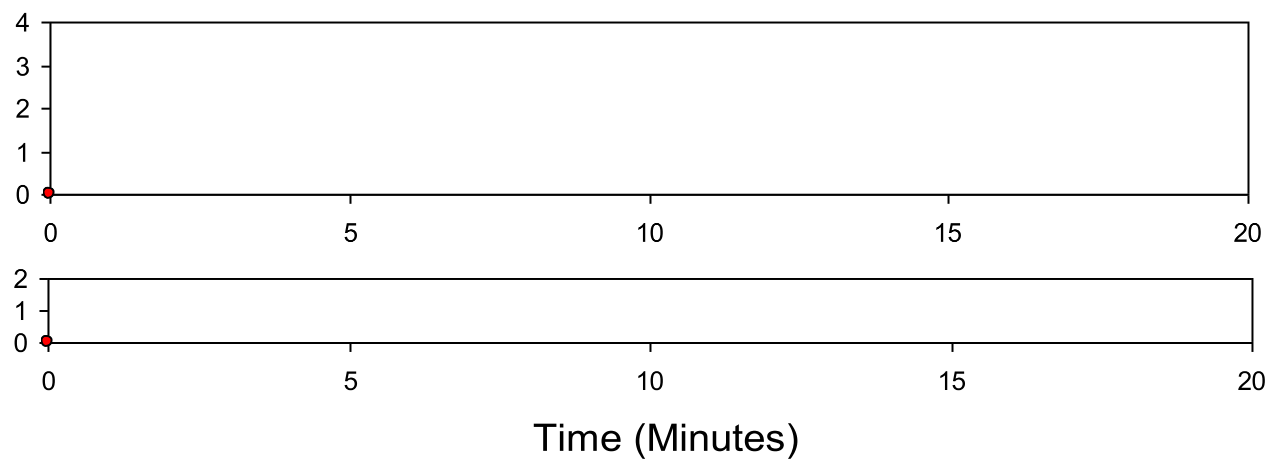
* مجرد ما اجى الـ time zero بوقف عالباب
و بسأله
when will the next person arrive
⇐ الجواب (1.73) اذاً بضيف على الـ Calender

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	0.00	0	0	<empty>	[1, 0.00, Arr] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
0	0.00	0.00	0.00

$Q(t)$ graph

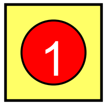
$B(t)$ graph



Interarrival times	1.73, 1.35, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	2.90, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand: *t* = 0.00, Arrival of Part 1

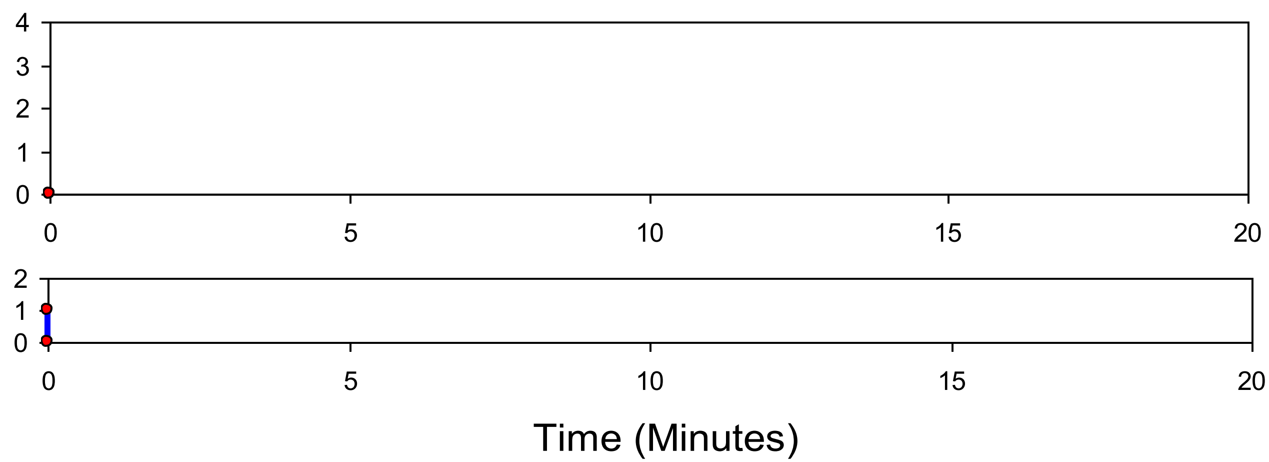
move directly to the closest time

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	0.00	1	0	<empty>	[2, 1.73, Arr] [1, 2.90, Dep] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
1	0.00	0.00	0.00

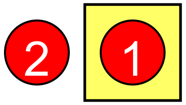
$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, 1.35, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand:
t = 1.73, Arrival of Part 2

<table>
<tr><td colspan="2">System
</td><td>Clock

1.73</td><td>$B(t)$

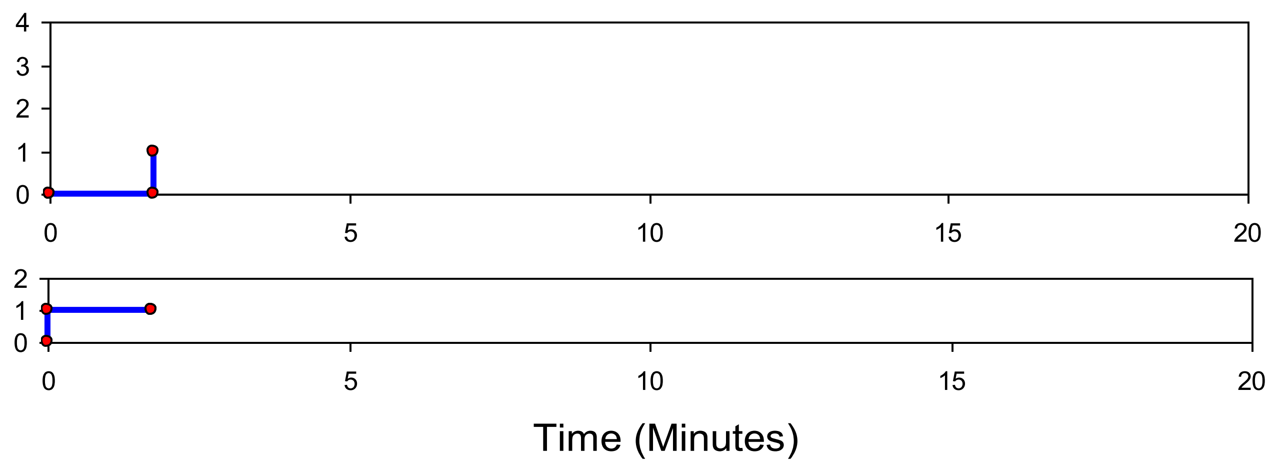
1</td><td colspan="2">$Q(t)$

1</td><td>Arrival times of custs. in queue
(1.73)</td><td>Event calendar
[1, 2.90, Dep]
[3, 3.08, Arr]
[–, 20.00, End]</td></tr>
<tr><td colspan="2">Number of completed waiting times in queue
1</td><td colspan="3">Total of waiting times in queue

0.00</td><td colspan="2">Area under $Q(t)$

0.00</td><td>Area under $B(t)$

1.73</td></tr>
<tr><td colspan="2">$Q(t)$ graph

$B(t)$ graph</td><td colspan="6"></td></tr>
<tr><td colspan="2">Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td colspan="2">Service times</td><td colspan="6">~~2.90~~, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand: t = 2.90, Departure of Part 1

<table>
<tr><td>System
</td><td>Clock

2.90</td><td>$B(t)$

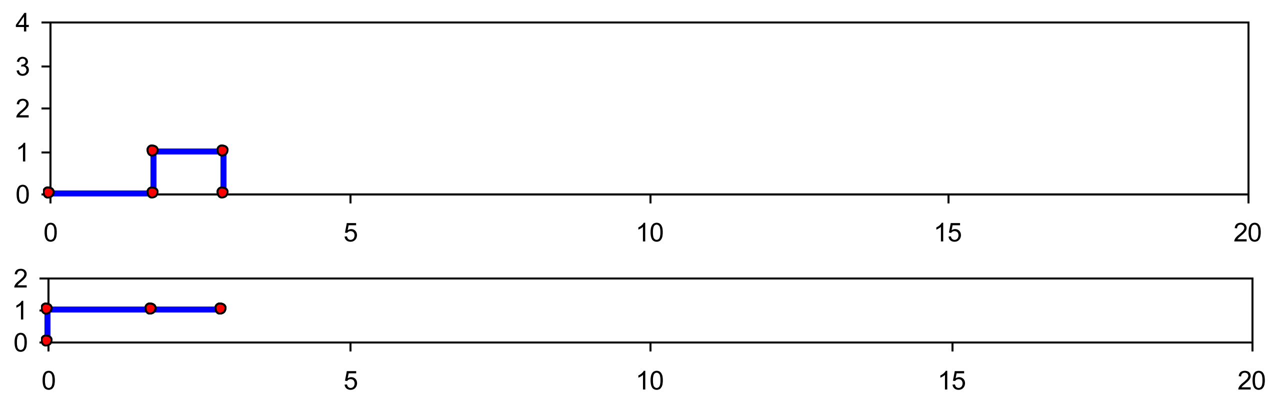
1</td><td>$Q(t)$

0
continuous</td><td>Arrival times of custs. in queue
<empty></td><td>Event calendar
[3, 3.08, Arr]
[2, 4.66, Dep]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
2</td><td colspan="2">Total of waiting times in queue

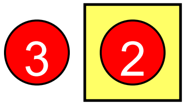
1.17</td><td colspan="2">Area under $Q(t)$

1.17</td><td>Area under $B(t)$

2.90</td></tr>
<tr><td>$Q(t)$ graph

$B(t)$ graph</td><td colspan="5">Time (Minutes)</td></tr>
<tr><td>Interarrival times</td><td colspan="5">~~1.73~~, ~~1.35~~, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="5">~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

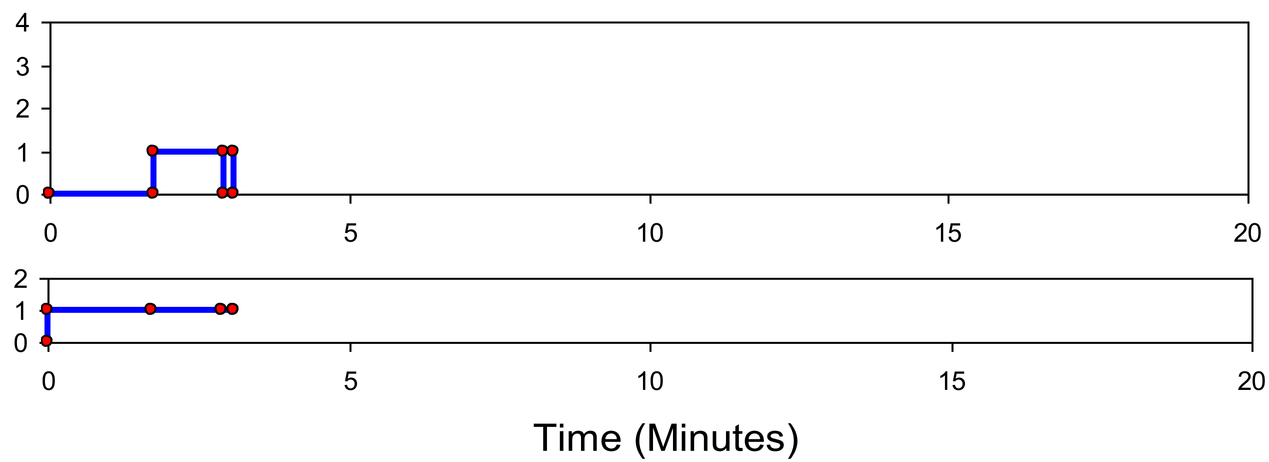
Simulation by Hand:
t = 3.08, Arrival of Part 3

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	3.08	1	1	(3.08)	[4, 3.79, Arr] [2, 4.66, Dep] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
2	1.17	1.17	3.08

$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

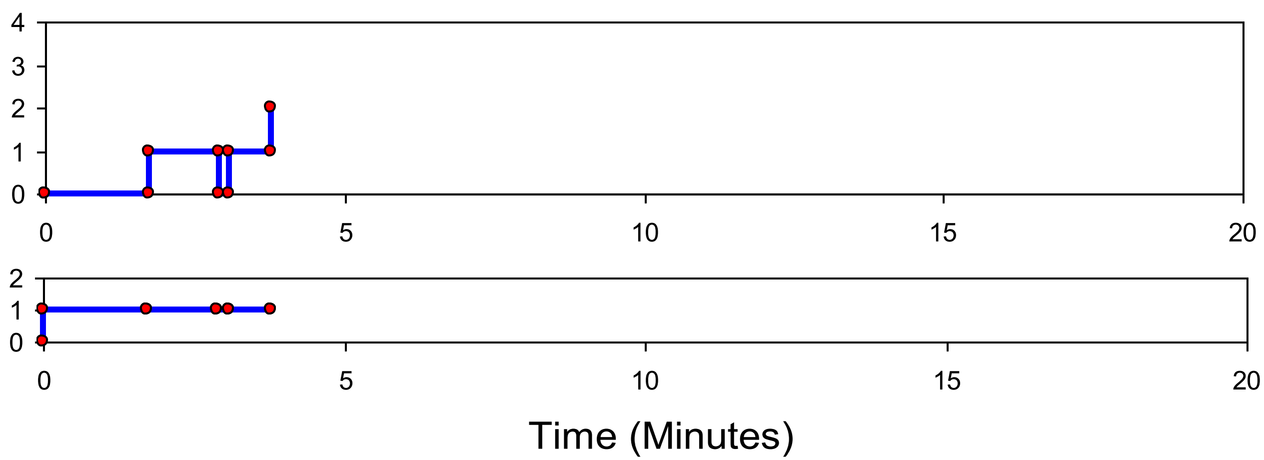
Simulation by Hand: *t* = 3.79, Arrival of Part 4

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
4 3 2	3.79	1	2	(3.79, 3.08)	[5, 4.41, Arr] [2, 4.66, Dep] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
2	1.17	1.88	3.79

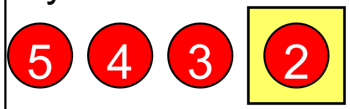
$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand:
t = 4.41, Arrival of Part 5

<table>
<tr><td>System
</td><td>Clock

4.41</td><td>$B(t)$

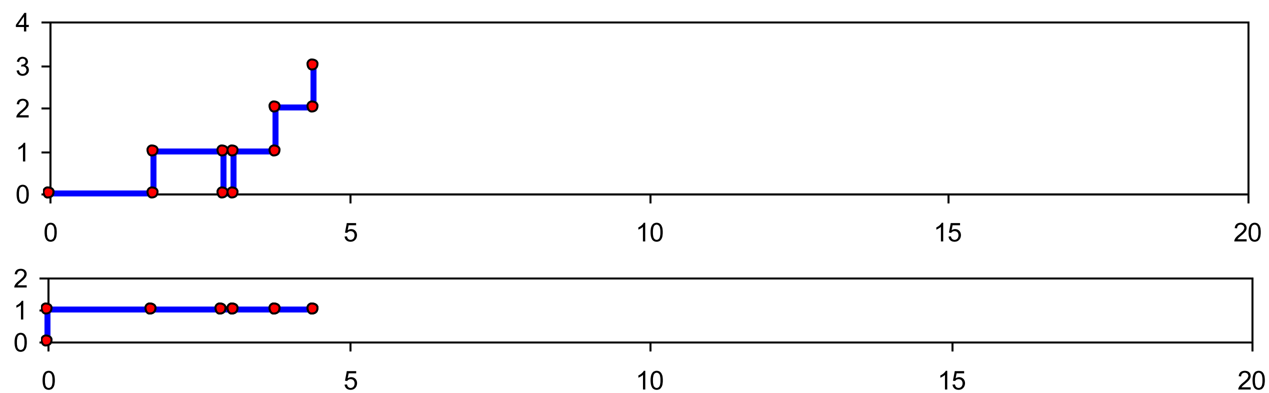
1</td><td colspan="2">$Q(t)$

3</td><td>Arrival times of custs. in queue
(4.41, 3.79, 3.08)</td><td>Event calendar
[2, 4.66, Dep]
[6, 18.69, Arr]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
2</td><td colspan="3">Total of waiting times in queue

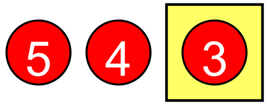
1.17</td><td colspan="2">Area under $Q(t)$

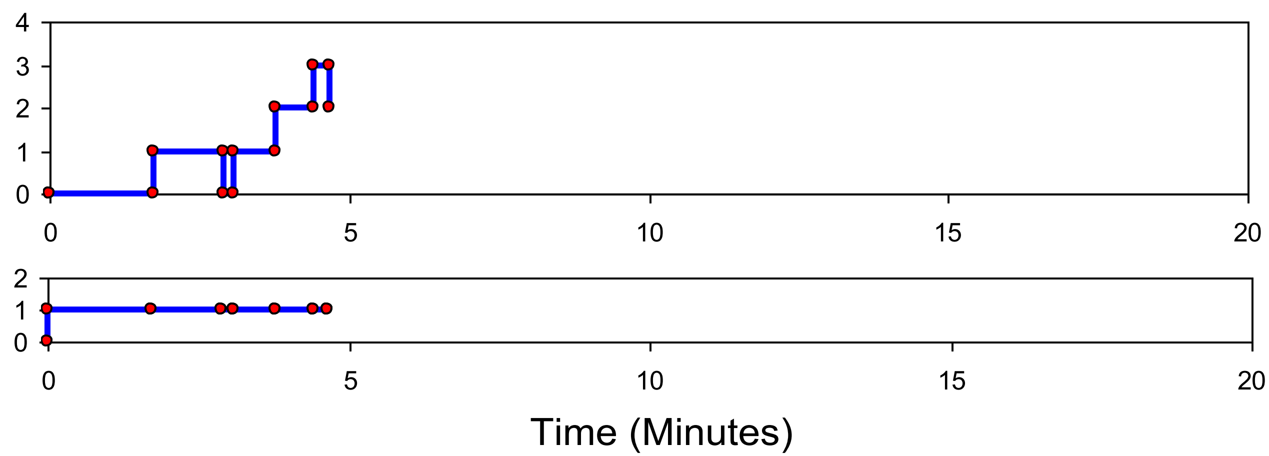
3.12</td><td>Area under $B(t)$

4.41</td></tr>
<tr><td>$Q(t)$ graph

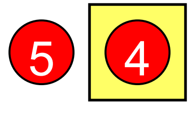
$B(t)$ graph</td><td colspan="6"></td></tr>
<tr><td>Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="6">~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand:
t = 4.66, Departure of Part 2

<table>
<tr><td colspan="2">System
</td><td>Clock
4.66</td><td>$B(t)$
1</td><td colspan="2">$Q(t)$
2</td><td>Arrival times of custs. in queue
(4.41, 3.79)</td><td>Event calendar
[3, 8.05, Dep]
[6, 18.69, Arr]
[–, 20.00, End]</td></tr>
<tr><td colspan="2">Number of completed waiting times in queue
3</td><td colspan="3">Total of waiting times in queue
2.75</td><td colspan="2">Area under $Q(t)$
3.87</td><td>Area under $B(t)$
4.66</td></tr>
<tr><td colspan="2">$Q(t)$ graph

$B(t)$ graph</td><td colspan="6"></td></tr>
<tr><td colspan="2">Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td colspan="2">Service times</td><td colspan="6">~~2.90~~, ~~1.76~~, ~~3.39~~, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

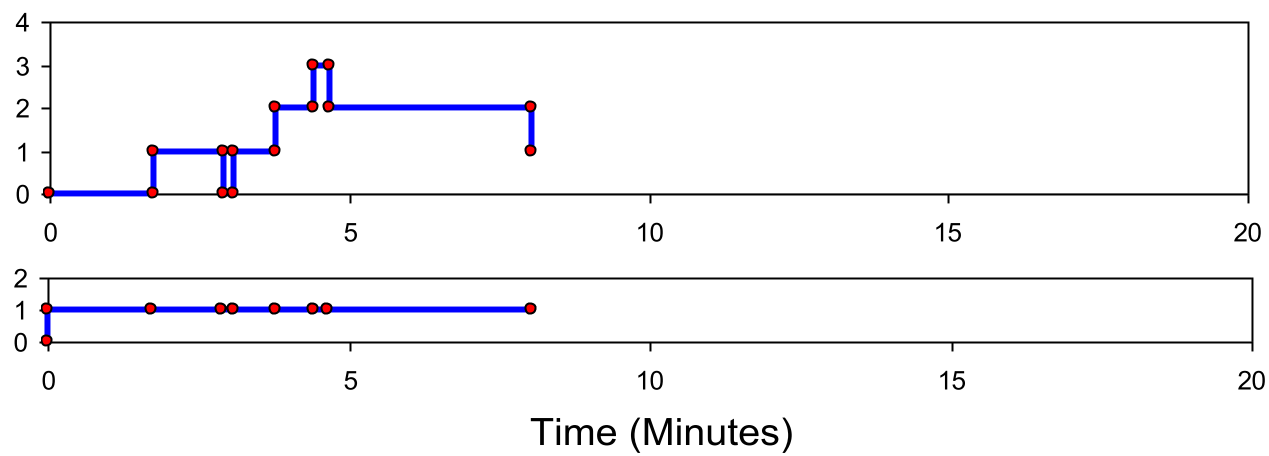
Simulation by Hand:
t = 8.05, Departure of Part 3

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	8.05	1	1	(4.41)	[4, 12.57, Dep] [6, 18.69, Arr] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
4	7.01	10.65	8.05

$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand: t = 12.57, Departure of Part 4

<table>
<tr><td>System
5</td><td>Clock

12.57</td><td>B(t)

1</td><td>Q(t)

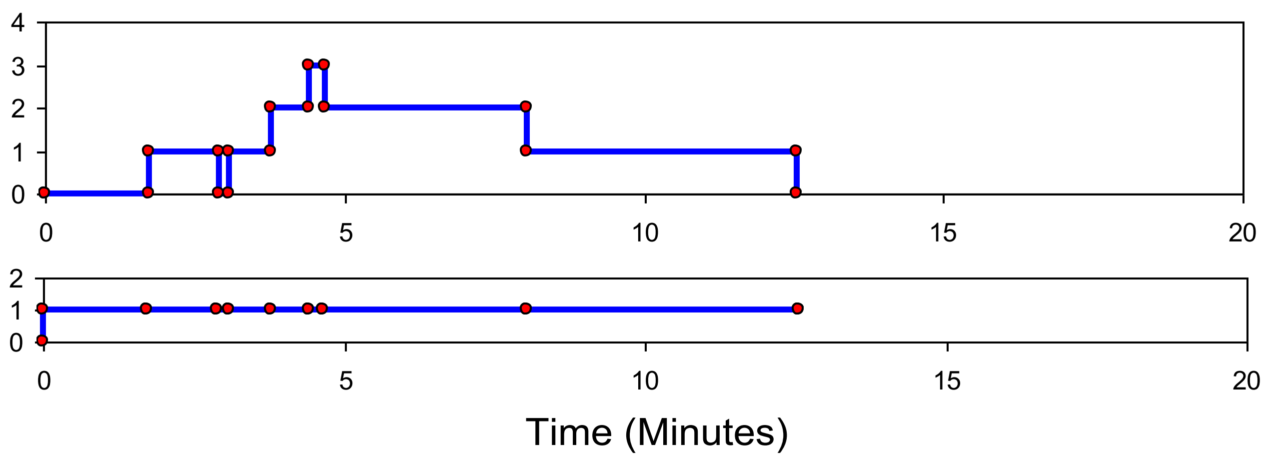
0</td><td colspan="2">Arrival times of custs. in queue

()</td><td>Event calendar
[5, 17.03, Dep]
[6, 18.69, Arr]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
5</td><td colspan="3">Total of waiting times in queue

15.17</td><td colspan="2">Area under Q(t)

15.17</td><td>Area under B(t)

12.57</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="6">
</td></tr>
<tr><td>Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="6">~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

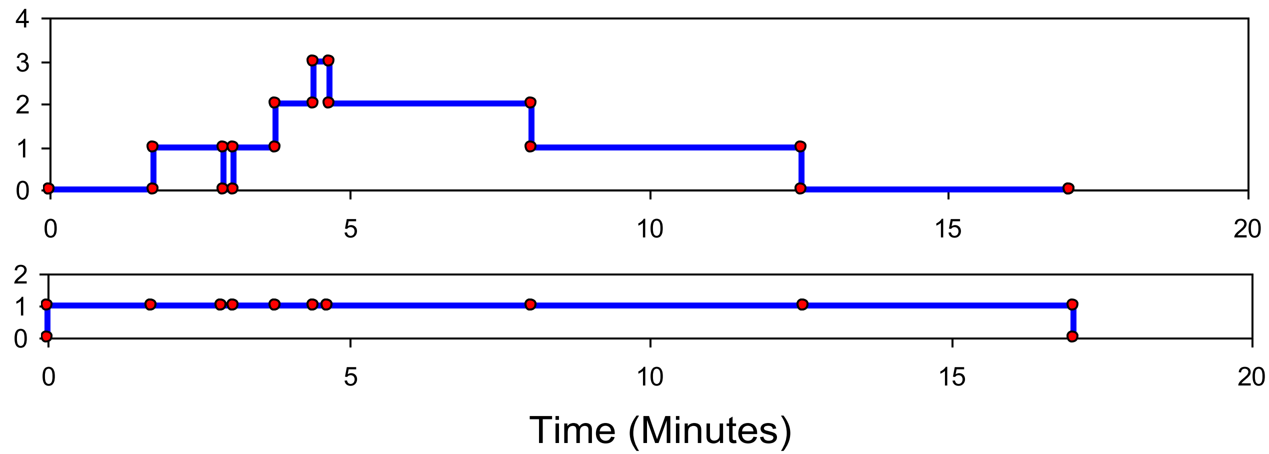
Simulation by Hand: t = 17.03, Departure of Part 5

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	17.03	0	0	()	[6, 18.69, Arr] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
5	15.17	15.17	17.03

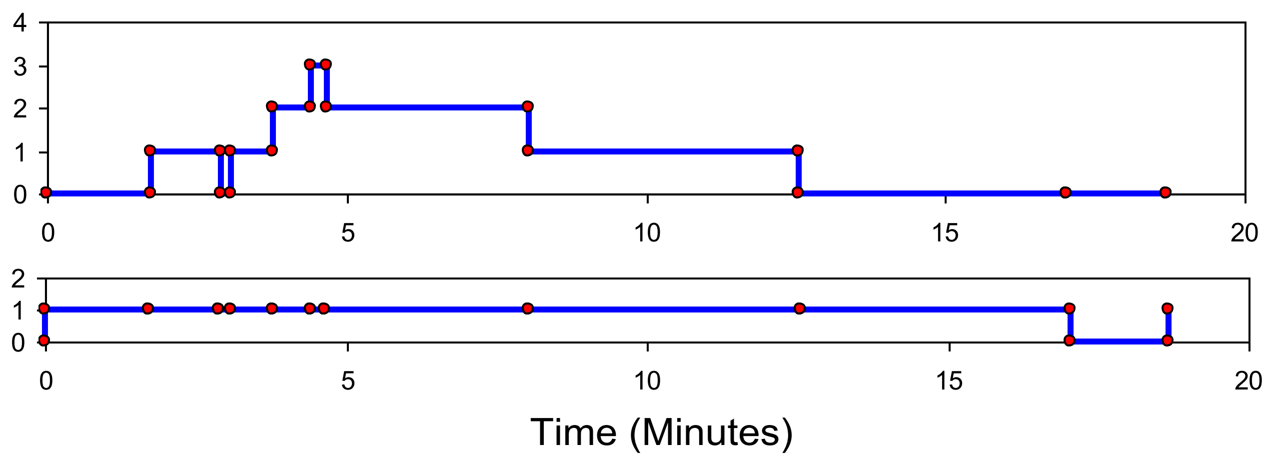
$Q(t)$ graph

$B(t)$ graph

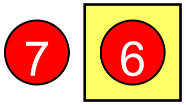


Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, 4.36, 2.07, 3.36, 2.37, 5.38, ...

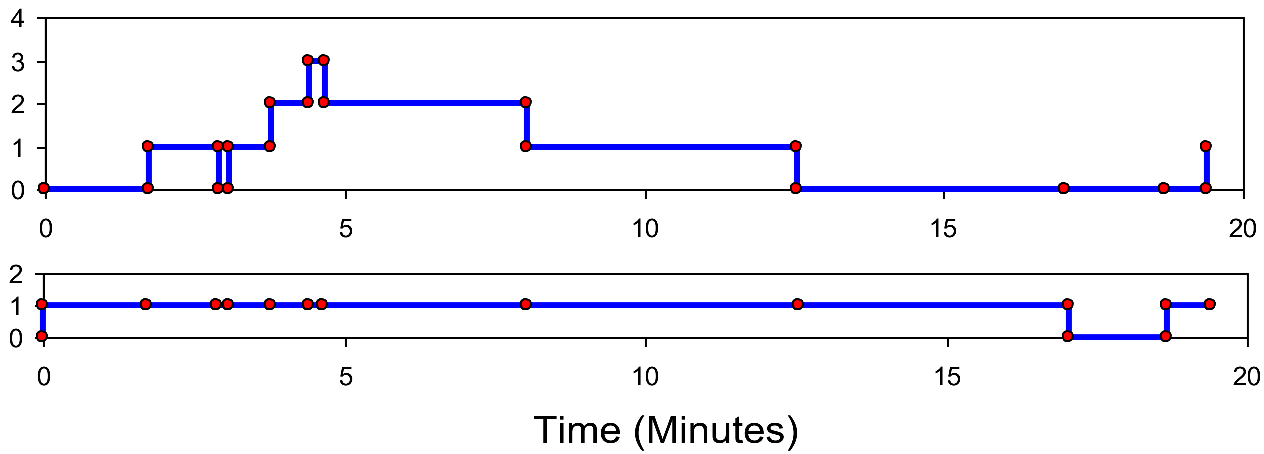
Simulation by Hand:
t = 18.69, Arrival of Part 6

System: 6	Clock 18.69	$B(t)$ 1	$Q(t)$ 0	Arrival times of custs. in queue ()	Event calendar [7, 19.39, Arr] [–, 20.00, End] [6, 23.05, Dep]
Number of completed waiting times in queue 6	Total of waiting times in queue 15.17		Area under $Q(t)$ 15.17		Area under $B(t)$ 17.03
$Q(t)$ graph $B(t)$ graph					
Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, ~~0.70~~, 15.52, 3.15, 1.76, 1.00, ...				
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, ~~4.36~~, 2.07, 3.36, 2.37, 5.38, ...				

Simulation by Hand: *t* = 19.39, Arrival of Part 7

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	19.39	1	1	(19.39)	[–, 20.00, End] [6, 23.05, Dep] [8, 34.91, Arr]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
6	15.17	15.17	17.73

$Q(t)$ graph $B(t)$ graph	
Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, ~~0.70~~, ~~15.52~~, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, ~~4.36~~, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand: *t* = 20.00, The End

(Continuos Distribution)

لأني عم ارسمها مع الزمن

و بستخدم المساحة تحت المنحنى

<table>
<tr><td>System
7 6</td><td>Clock

20.00</td><td>B(t)

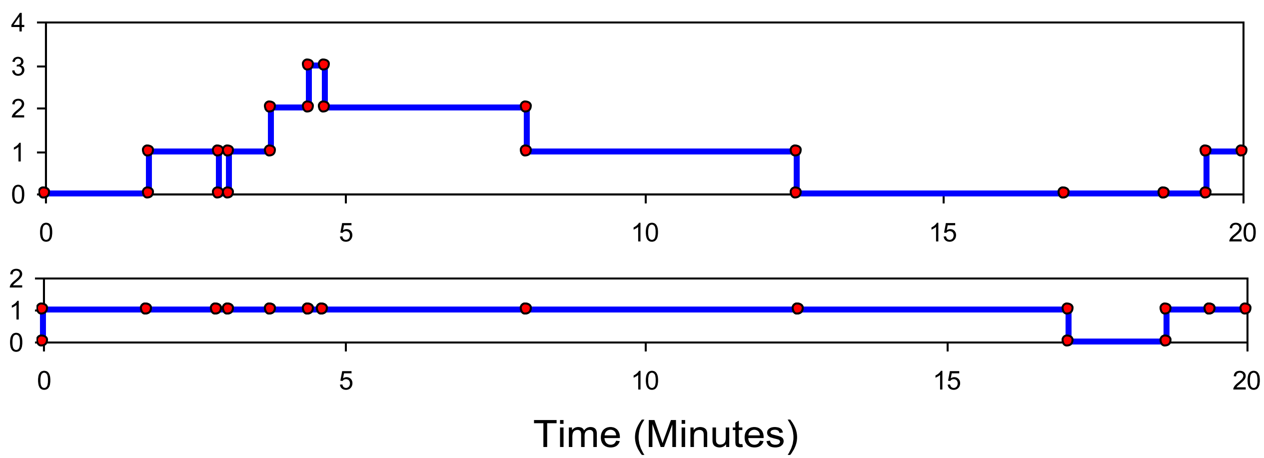
1</td><td>Q(t)

1</td><td>Arrival times of custs. in queue
(19.39)</td><td>Event calendar
[6, 23.05, Dep]
[8, 34.91, Arr]</td></tr>
<tr><td>Number of completed waiting times in queue
6</td><td colspan="2">Total of waiting times in queue

15.17</td><td colspan="2">Area under Q(t)

15.78</td><td>Area under B(t)

18.34</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="5">
</td></tr>
<tr><td>Interarrival times</td><td colspan="5">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, ~~0.70~~, ~~15.52~~, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="5">~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, ~~4.36~~, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Watch Lecture 10_21_2024 (Quiz Solution in Arena)

Simulation by Hand: Finishing Up

- **Average waiting time in queue:**

$$\frac{\text{Total of times in queue}}{\text{No. of times in queue}} = \frac{15.17}{6} = 2.53 \text{ minutes per part}$$

- **Time-average number in queue:**

$$\frac{\text{Area under } Q(t) \text{ curve}}{\text{Final clock value}} = \frac{15.78}{20} = 0.79 \text{ part}$$

- **Utilization of drill press:**

17.03x1 + 1.031 x1

$$\frac{\text{Area under } B(t) \text{ curve}}{\text{Final clock value}} = \frac{18.34}{20} = 0.92 \text{ (dimensionless)}$$

p80

Complete Record of the Hand Simulation

Just-Finished Event			Variables		Attributes		Statistical Accumulators									Event Calendar
Entity No.	Time t	Event Type	$Q(t)$	$B(t)$	Arrival Times: (In Queue)	In Service	P	N	ΣWQ	WQ^*	ΣTS	TS^*	$\int Q$	Q^*	$\int B$	[Entity No., Time, Type]
–	0.00	Init	0	0	( )	–	0	0	0.00	0.00	0.00	0.00	0.00	0	0.00	[1, 0.00, Arr] [–, 20.00, End]
1	0.00	Arr	0	1	( )	0.00	0	1	0.00	0.00	0.00	0.00	0.00	0	0.00	[2, 1.73, Arr] [1, 2.90, Dep] [–, 20.00, End]
2	1.73	Arr	1	1	(1.73)	0.00	0	1	0.00	0.00	0.00	0.00	0.00	1	1.73	[1, 2.90, Dep] [3, 3.08, Arr] [–, 20.00 End]
1	2.90	Dep	0	1	( )	1.73	1	2	1.17	1.17	2.90	2.90	1.17	1	2.90	[3, 3.08, Arr] [2, 4.66, Dep] [–, 20.00, End]
3	3.08	Arr	1	1	(3.08)	1.73	1	2	1.17	1.17	2.90	2.90	1.17	1	3.08	[4, 3.79, Arr] [2, 4.66, Dep] [–, 20.00, End]
4	3.79	Arr	2	1	(3.79, 3.08)	1.73	1	2	1.17	1.17	2.90	2.90	1.88	2	3.79	[5, 4.41, Arr] [2, 4.66, Dep] [–, 20.00, End]
5	4.41	Arr	3	1	(4.41, 3.79, 3.08)	1.73	1	2	1.17	1.17	2.90	2.90	3.12	3	4.41	[2, 4.66, Dep] [6, 18.69, Arr] [–, 20.00, End]
2	4.66	Dep	2	1	(4.41, 3.79)	3.08	2	3	2.75	1.58	5.83	2.93	3.87	3	4.66	[3, 8.05, Dep] [6, 18.69, Arr] [–, 20.00, End]
3	8.05	Dep	1	1	(4.41)	3.79	3	4	7.01	4.26	10.80	4.97	10.65	3	8.05	[4, 12.57, Dep] [6, 18.69, Arr] [–, 20.00, End]
4	12.57	Dep	0	1	( )	4.41	4	5	15.17	8.16	19.58	8.78	15.17	3	12.57	[5, 17.03, Dep] [6, 18.69, Arr] [–, 20.00, End]
5	17.03	Dep	0	0	( )	–	5	5	15.17	8.16	32.20	12.62	15.17	3	17.03	[6, 18.69, Arr] [–, 20.00 End]
6	18.69	Arr	0	1	( )	18.69	5	6	15.17	8.16	32.20	12.62	15.17	3	17.03	[7, 19.39, Arr] [–, 20.00, End] [6, 23.05, Dep]
7	19.39	Arr	1	1	(19.39)	18.69	5	6	15.17	8.16	32.20	12.62	15.17	3	17.73	[–, 20.00, End] [6, 23.05, Dep] [8, 34.91, Arr]
–	20.00	End	1	1	(19.39)	18.69	5	6	15.17	8.16	32.20	12.62	15.78	3	18.34	[6, 23.05, Dep] [8, 34.91, Arr]

$\div 6 =$

Event-Scheduling Logic via Programming

- **Clearly well suited to standard programming language**
- **Often use "utility" libraries for:**
  - List processing
  - Random-number generation
  - Random-variate generation
  - Statistics collection
  - Event-list and clock management
  - Summary and output
- **Main program ties it together, executes events in order**

Simulation Dynamics: The Process-Interaction World View

- **Identify characteristic *entities* in the system**
- **Multiple copies of entities co-exist, interact, compete**
- **"Code" is non-procedural**
- **Tell a "story" about what happens to a "typical" entity**
- **May have many types of entities, "fake" entities for things like machine breakdowns**
- **Usually requires special simulation software**
  - Underneath, still executed as event-scheduling
- **The view normally taken by Arena**
  - Arena translates your model description into a program in the SIMAN simulation language for execution

Randomness in Simulation

- The above was just one "replication" — a sample of size one (not worth much)
- Made a total of five replications:

Performance Measure	Replication 1	2	3	4	5	Sample Avg.	Sample Std. Dev.	95% Half Width
Total production	5	3	6	2	3	3.80	1.64	2.04
Average waiting time in queue	2.53	1.19	1.03	1.62	0.00	1.27	0.92	1.14
Maximum waiting time in queue	8.16	3.56	2.97	3.24	0.00	3.59*	2.93*	3.63*
Average total time in system	6.44	5.10	4.16	6.71	4.26	5.33	1.19	1.48
Maximum total time in system	12.62	6.63	6.27	7.71	4.96	7.64*	2.95*	3.67*
Time-average number of parts in queue	0.79	0.18	0.36	0.16	0.05	0.31	0.29	0.36
Maximum number of parts in queue	3	1	2	1	1	1.60*	0.89*	1.11*
Drill-press utilization	0.92	0.59	0.90	0.51	0.70	0.72	0.18	0.23

Note substantial variability across replications

- Confidence intervals for expected values:
  - In general, $\left[\bar{X} \pm t_{n-1,1-\alpha/2}\, s/\sqrt{n}\right]$
  - For expected total production, $3.80 \pm (2.776)(1.64/\sqrt{5})$

    3.80 ± 2.04

stat 2 من

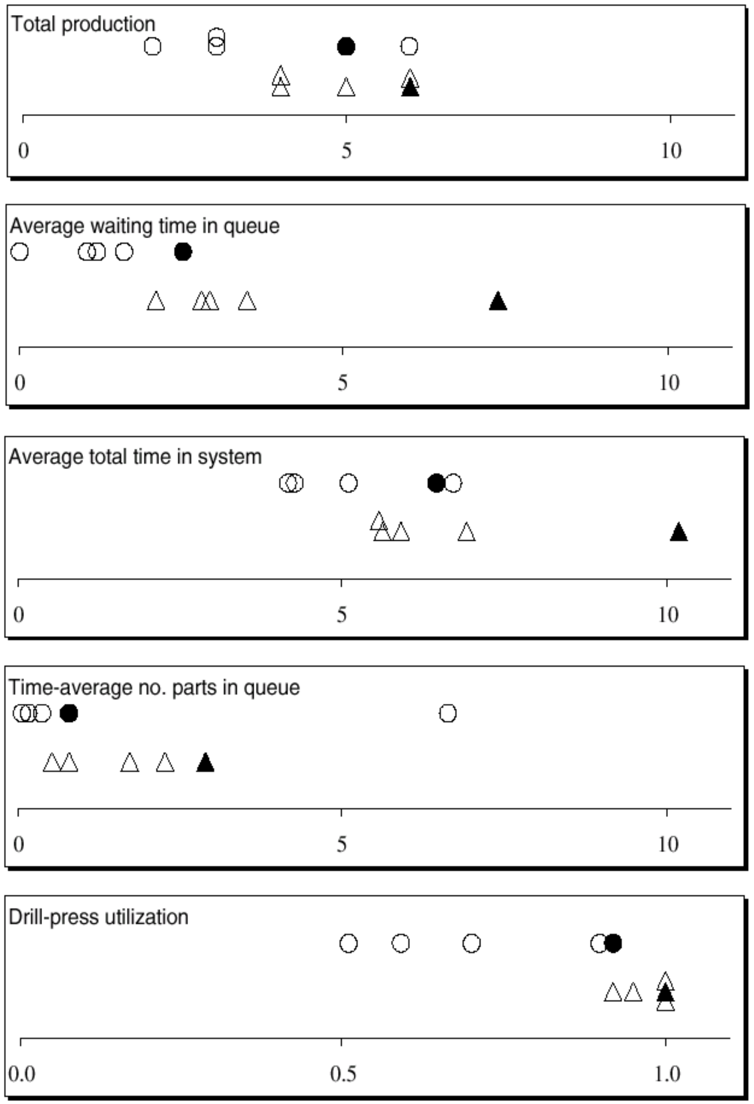
Point estimate

interval estimate

Comparing Alternatives

- **Usually, simulation is used for more than just a single model "configuration"**
- **Often want to compare alternatives, select or search for the best (via some criterion)**
- **Simple processing system: What would happen if the arrival rate were to double?**
  - Cut interarrival times in half
  - Rerun the model for double-time arrivals
  - Make five replications

Results: Original vs. Double-Time Arrivals

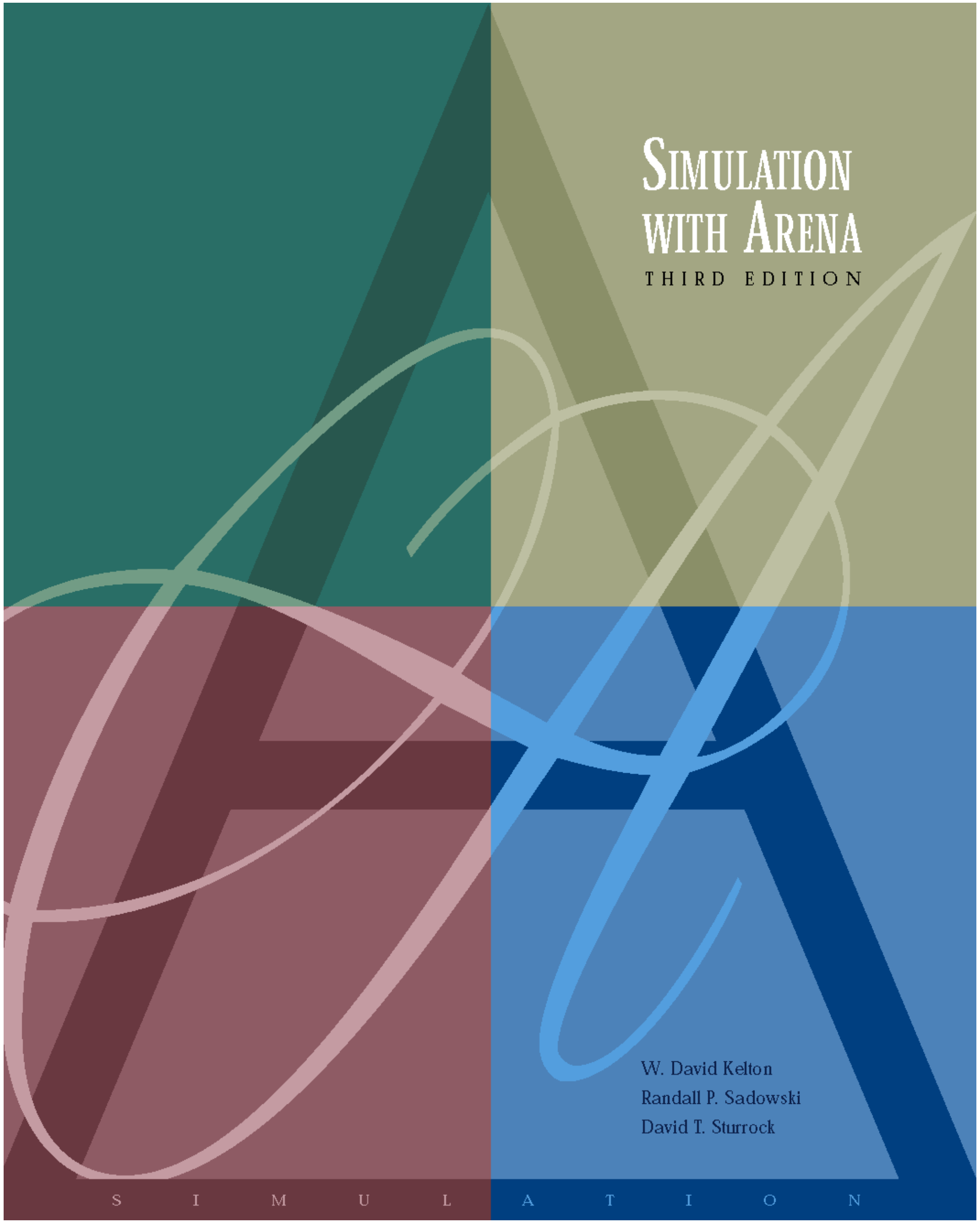


- Original – circles
- Double-time – triangles
- Replication 1 – filled in
- Replications 2-5 – hollow
- Note variability
- Danger of making decisions based on one (first) replication
- Hard to see if there are really differences
- Need: Statistical analysis of simulation output data

Overview of a Simulation Study

- Understand the system
- Be clear about the goals
- Formulate the model representation
- Translate into modeling software
- Verify "program"
- Validate model
- Design experiments
- Make runs
- Analyze, get insight, document results

Statistical Test → هو أضعف نوع



A Guided Tour Through Arena

Chapter 3

Self Reading

Last revision June 7, 2003

What We'll Do ...

- **Start Arena**
- **Load, explore, run an existing model**
  - Basically the same model as for the hand simulation in Chapter 2
  - Browse dialogs and menus
  - Run the model
  - Look at results
- **Construct the same model from scratch**
- **Tour menus, toolbars, drawing, printing**
- **Help system**
- **Options for running and control**

Behavior of Arena

- **Arena is a true Windows application**
  - Appearance, operation, functions, are standard
  - Interoperability with other software (MS Office, CAD)
  - Interact, communicate with other software (Chapter 10)
- **Assume you already know basics of Windows:**
  - Disks, files, folders, paths
  - Mousing, keyboarding
  - Resizing, moving, maximizing, minimizing windows
  - Menu operations
  - Ctrl, Alt, Shift keys
  - Cut, copy, paste
  - Filling out dialog fields

Starting Up

- **Installing Arena – Appendix E**
- **Locate icon or shortcut; double-click**
  - Or, *Start* > *Programs* > *Rockwell Software* > *Arena 7.0* > *Arena 7.01*
  - Licensed vs. Academic, Evaluation mode
- **See File, View, Tools, Help menus**
  - Other menus present if a model file is open
- **Toolbars with buttons**
  - Unless a model file is open, only New model file, Open model file, Template Attach/Detach, Context Help (click it, then click on buttons or menu items)
- **Tooltips – roll over toolbar buttons for names**
- **Quitting Arena: *File* > *Exit* or Alt+F4 or top right**  **button**

Opening an Existing Model

- ***File > Open*** **… or**  **button**
  - Navigate to desired disk/directory
  - *Click* > *Open* or double-click **`Model 03-01.doe`**
  - Book models: Rockwell Software\Arena 7.0\Book Examples
  - More examples: Rockwell Software\Arena 7.0\Examples
- **Model window (usually on right side of Arena window)**
  - Where model is built
  - Resize, maximize, minimize, scroll/pan, zoom
  - Can have multiple model windows open at once
- **Cut, Copy, Paste within Arena, and between Arena and other applications (when sensible)**

Why the **`.doe`** *default extension to Arena model filenames?*

Flowchart and Spreadsheet Views

- **Model window split into two views**
  - *Flowchart* view
    - Graphics
    - Process flowchart
    - Animation, drawing
    - Edit things by double-clicking on them, get into a dialog
  - *Spreadsheet* view
    - Displays model data directly
    - Can edit, add, delete data in spreadsheet view
    - Displays all similar kinds of modeling elements at once
  - Many model parameters can be edited in either view
  - Horizontal splitter bar to apportion the two views
  - *View > Split Screen* (or push ) to see both flowchart and spreadsheet views (otherwise, get just flowchart view)

Project Bar

- **Usually down the left edge of Arena window**
- **Hosts panels with modeling building blocks:** ***modules***
    - Both flowchart and spreadsheet modules
- **Displays one panel at a time**
    - Switch to different panels via horizontal buttons
    - Panels for Basic Process, Reports (after running), Navigate (to different views within a model or to different hierarchical submodels), … others can be attached (Template Attach button) for different modeling levels, specialties
- **Usually docked to left edge but can move, float**
- **Hide it via *View > Project Bar* or its own small**

Status Bar

- **At very bottom of Arena window**
- **Displays various information sensitive to status**
  - Coordinates of cursor in “worldspace”
  - Simulation clock value, replication number being executed, number of replications to be done, when simulation is running
- **Hide by clearing (unchecking) *View > Status Bar***

Moving Around, Up, Down in Flowchart View of Model Window

- **Underlying *world space* for model**
  - (x, y) coordinates, arbitrary units (thousands in each direction)
- ***Pan* with scroll bars, arrow keys**
- ***Zoom* in (down): or + key**
- **Zoom out (up): or – key**
- **See all at min altitude: or * key**
- ***Named views***
  - Save a pan/zoom view for different parts of model
  - Assign a *Hot key* (case-sensitive)
  - Access via *View > Named Views …* or ? key
- **Display *grid* ( ), *snap* to grid ( ) toggles**

To navigate via keyboard, the model window must be active ... click in it.

Modules

- **Basic building blocks of a simulation model**
- **Two basic types: *flowchart* and *data***
- **Different types of modules for different actions, specifications**
- **"Blank" modules are on the Project Bar**
  - To add a flowchart module to your model, drag it from the Project Bar into the flowchart view of the model window
    - Can have many instances of the same kind of flowchart module in your model
  - To use a data module, select it (single-click) in the Project Bar and edit in the spreadsheet view of the model window
    - Only one instance of each kind of data module in your model, but it can have many entries (rows) in the spreadsheet view
    - Can edit via dialog – double-click on number in leftmost column

Flowchart Modules

- **Describe dynamic processes**
  - Nodes/places through which entities flow
  - Typically connected to each other in some way
- **Basic Process panel flowchart module types:**
  - Create, Dispose, Process, Decide, Batch, Separate, Assign, Record
- **Other panels – many other kinds**
- **Shape like flowcharting (also use colors for hints)**
- **Two ways to edit**
  - Double-click to open up, then fill out dialogs
  - Select (single-click) a module type in model or Project Bar, get all modules of that type in the spreadsheet view

Data Modules

- **Set values, conditions, etc. for whole model**
  - No entity flow, no connections
- **Basic Process panel data module types:**
  - Entity, Queue, Resource, Variable, Schedule, Set
- **Other panels – many other kinds**
- **Icons in Project Bar look like little spreadsheets**
- **To use a data module, select it (single-click) in the Project Bar, edit in spreadsheet view**
  - Can edit via dialog – double-click in leftmost column
  - Double-click where indicated to add new row
  - Right-click on row, column to do different things
- **Only one instance of each kind of data module in a model**
  - But each one can have many entries (rows)

Relations Among Modules

- **Flowchart and data modules are related via names for objects**
  - Queues, Resources, Entity types, Variables … others
- **Arena keeps internal lists of different kinds of names**
  - Presents existing lists to you where appropriate
  - Helps you remember names, protects you from typos
- **All names you make up in a model must be unique across the model, even across different types of modules**

Internal Model Documentation

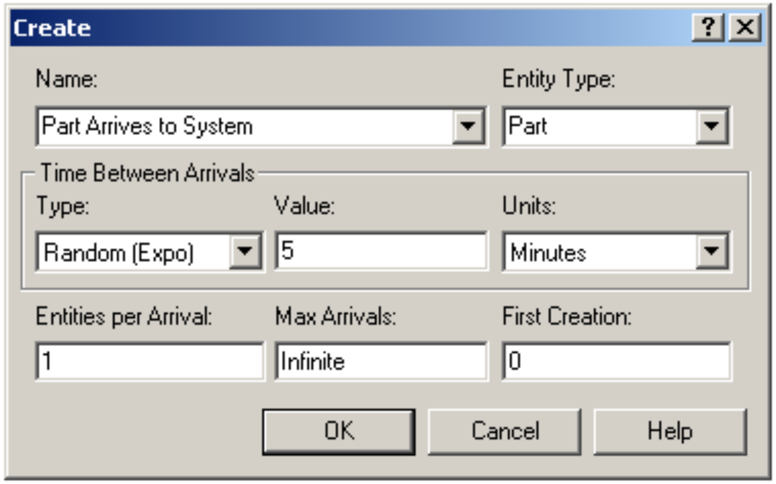
- ***Data Tips*** **on modules, graphics – hover mouse over object to see**
  - Default part – generic info on object (name, type)
  - User-defined part – right-click on object, select Properties, enter text under Property Description
  - Toggle display of Data tips via *View > Data Tips*
- ***Project Description*** **–** ***Run > Setup > Project Parameters,*** **enter text under Project Description**
- ***Model Documentation Report*** **–** ***Tools > Model Documentation Report***
  - Generates HTML file with model details (can choose which kinds of details to include)

Browsing Through Model 3-1

- **Open `Model 03-01.mod` (in Book Examples folder)**
- **Three flowchart modules (Create, Process, Dispose)**
- **Entries in three data modules (Entity, Queue, Resource)**
- **Animation objects**
  - Resource animation
  - Two plots
  - Some (passive) labels, artwork

The *Create* Flowchart Module

- **"Birth" node for entities**
- **Gave this instance of the Create-type module the Name `Part Arrives to System`**
  - If we had other Create modules (we don't) they'd all have different Names
- **Double-click on module to open property dialog:**



The *Create* Flowchart Module (cont'd.)

- **Name – for module (type it in, overriding default)**
- **Entity Type – enter a descriptive name**
  - Can have multiple Entity Types with distinct names
- **Time Between Arrivals area**
  - Specify nature of the time separating consecutive arrivals
  - Type – pull-down list with several options
  - Value – depends on Type … for Random (Expo) is mean
  - Units – time units for Value
- **Entities per Arrival – constant, random variable, very general "Expression" (more later …)**
- **Max Arrivals – choke off arrivals (from this source) after this many entities**
- **First Creation – time of first arrival (need not be 0)**

Editing Flowchart Modules in the Spreadsheet View

- **Alternative to dialog for each instance of a module type**
- **See all instances of a module type at once**
  - Convenient for seeing, editing lots of things at once
- **Selecting a module in either flowchart or spreadsheet view also selects it in the other view**
- **Click, double-click fields to view, edit**
- **Right-click in row to Edit via Dialog, define a user Data Tip (via Properties)**
- **Right-click in expression fields to get Expression Builder for help in constructing complex expressions with Arena variables (more later …)**

The *Entity* Data Module

- **A data module, so edit in spreadsheet view only**
- **View, edit aspects of different Types of entities in your model (we have just one Type, `Part`)**
- **Pull-down lists activated as you select fields**
- **Our only edit – Initial Picture for animation**
  - We picked **`Picture.Blue Ball`** from the default list

The *Process* Flowchart Module

- **Represents the machine, including the resource, queue, and entity delay time (processing)**
- **Enter Name – `Drilling Center`**
- **Type – picked `Standard` to define logic here rather than in a submodel (more later …)**
- **Report Statistics check box at bottom**
  - To get utilizations, queue lengths, queue waiting times, etc.

The *Process* Flowchart Module (cont'd.)

- **Logic area – what happens to entities here**
  - Action
    - *Seize Delay Release* – entity Seizes some number of units of a Resource (maybe after a wait in queue), Delay itself there for the processing time, then Release the units of the Resource it had Seized – we chose this option
    - Could just *Delay* entity (red traffic light) – no Resources or queueing
    - Could also *Seize Delay* (no Release … Release downstream)
    - Could also *Delay Release* (if Resource had been Seized upstream)
    - Priority for seizing – lower numbers mean higher priority
    - Different Action choices could allow stringing together several Process modules for modeling flexibility
  - Resources – define Resource(s) to be seized, released
    - Double-click on row to open subdialog
    - Define Resource Name, Quantity of units to be Seized/Released here
    - Several Resources present (Add) – entities must first Seize all

The *Process* Flowchart Module (cont'd.)

- **Delay Type – choice of probability distributions, constant or general Expression (more later …)**
- **Units – time units for the delay (*don't ignore*)**
- **Allocation – how to "charge" delay in costing (more later …)**
- **Prompts on next line – change depending on choice of Delay Type – specify numerical parameters involved**
- **Can also edit in spreadsheet view**
  - Subdialogs (e.g., Resource here) become secondary spreadsheets that pop up, must be closed

The *Resource* Data Module

- **Defining the `Drilling Center Resource` in the Process module automatically creates an entry (row) for it in the Resource data module**
- **Can edit it here for more options**
  - Type – could vary capacity via a Schedule instead of having a fixed Capacity
    - Would define the Schedule via Schedule data module … more later
  - Failures – could cause resource to fail according to some pattern
    - Define this pattern via Failure data module (Advanced Process panel) … more later

The *Queue* Data Module

- **Specify aspects of the queues in the model**
  - We only have one, named **`Drilling Center.Queue`** (the default name given the Process name)
- **Type – specifies *queue discipline* or ranking rule**
  - If Lowest or Highest Attribute Value, then another field appears where you specify which attribute
- **Shared – it this queue will be shared among several resources (more later …)**
- **Report Statistics – check to get automatic collection and reporting of queue length, time in queue**

Animating Resources and Queues

- **Got queue animation automatically by specifying a Seize in the Process module**
  - Entity pictures (blue balls) will line up here in animation
- **Don’t get Resource animation automatically**
  - To add it, use Resource button in Animate toolbar … get Resource Picture Placement dialog
    - Identifier – link to Resource name in pull-down list
    - Specify different pictures for Idle, Busy state
      - For pre-defined artwork, Open a picture library (.plb filename extension)
      - Scroll up/down on right, select (single-click) a picture on right, select Idle or Busy state on left, then << to copy the picture
  - To edit later, double-click on picture in flowchart view

The *Dispose* Flowchart Module

- **Represents entities leaving model boundaries**
- **Name the module**
- **Decide on Record Entity Statistics (average and maximum time in system of entities exiting here, costing information)**

Check boxes for statistics collection and reporting:

- Most are checked (turned on) by default
- Little or no modeling effort to say yes to these
- However, in some models this can slow execution markedly
- Moral – if you have speed problems, clear these if you don’t care about them

Connecting Flowchart Modules

- **Establishes the (fixed) sequence of flowchart modules through which entities will flow**
- **To make a connection**
  - Click Connect button (or *Object > Connect*)
  - Cursor changes to cross hairs
  - Click on exit point ▶ from source module, then entry point ■ on destination module
  - Make intermediate clicks for non-straight line in segments
- **Object menu toggles**
  - Auto-Connect – automatically connect entry point of newly placed module from exit point of selected module
  - Smart Connect – force segments to horizontal/vertical
  - Animate Connectors – show entities moving along connectors (zero time for statistics collection)

Dynamic Plots

- **Trace variables, queues as simulation runs – a kind of "data animation"**
- **Disappear after run is ended (to keep, must save data, postprocess via Output Analyzer — later)**
- **Plot button from Animate toolbar… "Add" for**
  - Expression to plot (help via Expression Builder … later)
  - Min/Max *y*-axis values (initially guesses, maybe revise)
    - Arena can do this automatically and dynamically in Plot dialog
  - Number of "corners" to show (# History Points) at a time
  - Stepped option (for piecewise-constant curves)
  - Colors
- **In Plot dialog – Time Range (*x* axis), cosmetics, automatic scaling options**
- **Drop plot in via crosshairs (resize, move later)**

Dressing Things Up

- **Add drawing objects from Draw toolbar**
  - Similar to other drawing, CAD packages
  - Object-oriented drawing tools (layers, etc.), not just a painting tool
- **Add Text to annotate things**
  - Control font, size, color, orientation

Setting the Run Conditions

- ***Run > Setup* menu dialog – five tabs**
  - Project Parameters – Title, your name, output statistics, Project Description
  - Replication Parameters
    - Number of Replications
    - Initialization options Between Replications
    - Start Date/Time to associate with start of simulation
    - Warm-up Period (when statistics are cleared)
    - Length of Replication (and Time Units)
    - Base Time Units (output measures, internal computations)
    - Hours per "Day" (convenience for 16-hour days, etc.)
    - Terminating Condition (complex stopping rules)
  - Other tabs for animation speed, run conditions, reporting

Terminating your simulation:
- You must specify – part of modeling
- Arena has no default termination
- If you don't specify termination, Arena will usually keep running forever

Running It

- **Plain-vanilla run: Click ▶ from Standard toolbar (like audio/video players)**
  - First time or after changes: *Check*
  - Enters *run mode* — can move around but not edit
  - Speed up (> on keyboard) or slow down (<) animation display
  - When done, asked if you want to see summary reports
  - Click ■ to get out of run mode (*can't edit until you do*)
  - Can *pause* run with ❚❚ or Esc key
- **Other run control, viewing, checking options**

Viewing the Reports

- **Click Yes in the Arena box at the end of the run**
  - Opens up a new reports window (separate from model window) inside the Arena window
  - Project Bar shows Reports panel, with different reports (each one would be a new window)
  - Remember to close all reports windows before future runs
- **Default installation shows Category Overview report – summarizes many things about the run**
  - Reports have “page” to browse ( and )
  - Also, “table contents” tree at left for quick jumps via , 
- **Times are in Base Time Units for the model**

Viewing the Reports – Examples

- **Entity → Time → Total Time → Part:**
  - Avg. time in system was 6.4397 min., max was 12.6185
- **Resource → Usage → Instantaneous Utilization → Drill Press:**
  - Utilization was 0.9171 (busy 91.71% of the time)
- **Process → Other → Number In → Drilling Center:**
  - During the run, 7 parts entered the `Drilling Center`
- **Process → Other → Number Out → Drilling Center:**
  - 5 entities left the `Drilling Center` (so were produced)
- **Entity → Time → Wait Time → Part:**
  - Avg. wait time in all queues was 3.0340 min. (counts only entities that left the *system*, but Queue → Time → Waiting Time → Drilling Center.Queue counts all entities that left *this queue*, so can differ)
- **Entity → Other → Wip → Part:**
  - Average Work in Process was 1.7060, max WIP was 4

Types of Statistics Reported

- **Many output statistics are one of three types:**
  - *Tally* – avg., max, min of a discrete list of numbers
    - Used for discrete-time output processes like waiting times in queue, total times in system
  - *Time-persistent* – time-average, max, min of a plot of something where the *x*-axis is continuous time
    - Used for continuous-time output processes like queue lengths, WIP, server-busy functions (for utilizations)
  - *Counter* – accumulated sums of something, usually just nose counts of how many times something happened
    - Often used to count entities passing through a point in the model

More on Reports and their Files

- **Reports we just saw – based on a MS Access database that Arena writes as it runs**
  - Can be saved and viewed later
  - Viewing within Arena done via Crystal Reports to query the Access database
- **Arena also produces a plain-text summary report (.out filename extension)**
  - Was in previous versions of Arena, underlying SIMAN simulation language
  - Fairly cryptic, but gives quick view of lots of output data
- **Get multiple reports for multiple replications**
- **“Half Width” columns – for confidence intervals on outputs in long-run simulations … more later**

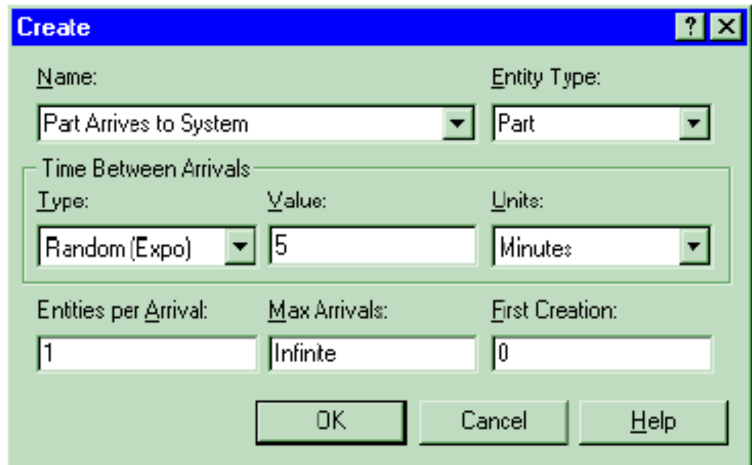
Build It Yourself

- **Build the same model from scratch – see book for more detail**
- **Handy user-interface tricks:**
  - Right-click in an empty spot in flowchart view – small box of options, including Repeat Last Action … useful in repetitive editing like placing lots of the same module type
  - Ctrl+D or Ins key – duplicates whatever's selected in flowchart view, offsetting it a bit … usually drag elsewhere and edit
- **Open a new (blank) model window – name it, save it, maybe maximize it**
- **Attach modeling panels you'll need to Project Bar if they're not there**

Build It Yourself (cont'd.)

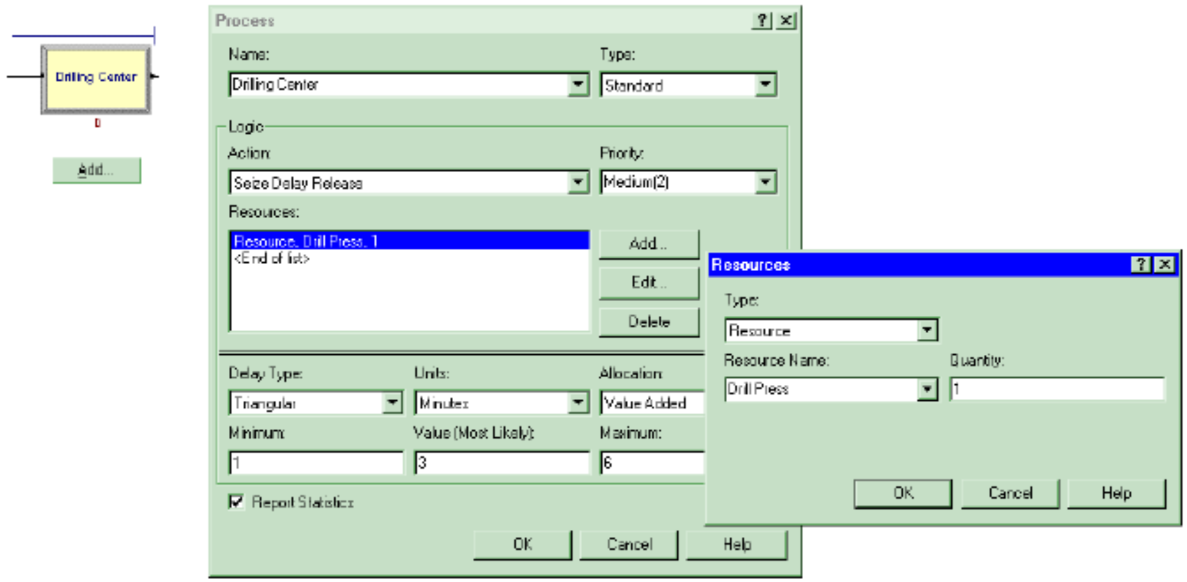
- **Place and connect flowchart modules**
- **Edit flowchart and data modules as needed**
  - Experiment with Expression Builder – right-click in expression field
- **Add plots, animation, artwork**
- **Add named views (? key or *View > Named Views*)**
  - Name, hot key (case-sensitive) – useful in big models
- **Edit *Run > Setup* dialog**
- **"Displays" in text**
  - Compact way of saying what needs to be done in a dialog
  - Omits Arena defaults
  - Shows completed dialogs

Display for the Create Module



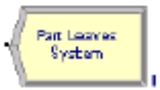
Name	`Part Arrives to System`
Entity Type	`Part`
Time Between Arrivals area	
Type	`Random (Expo)`
Value	`5`
Units	`Minutes`

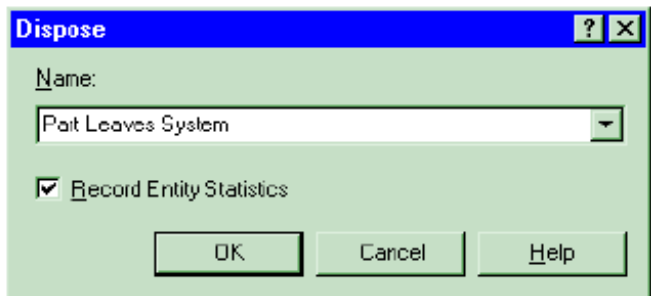
Display for the Process Module



Name	Drilling Center
Action	Seize Delay Release
Resources (secondary dialog via Add button)	
Type	Resource
Resource Name	Drill Press
Quantity	1
Delay Type	Triangular
Units	Minutes
Minimum	1
Value	3
Maximum	6

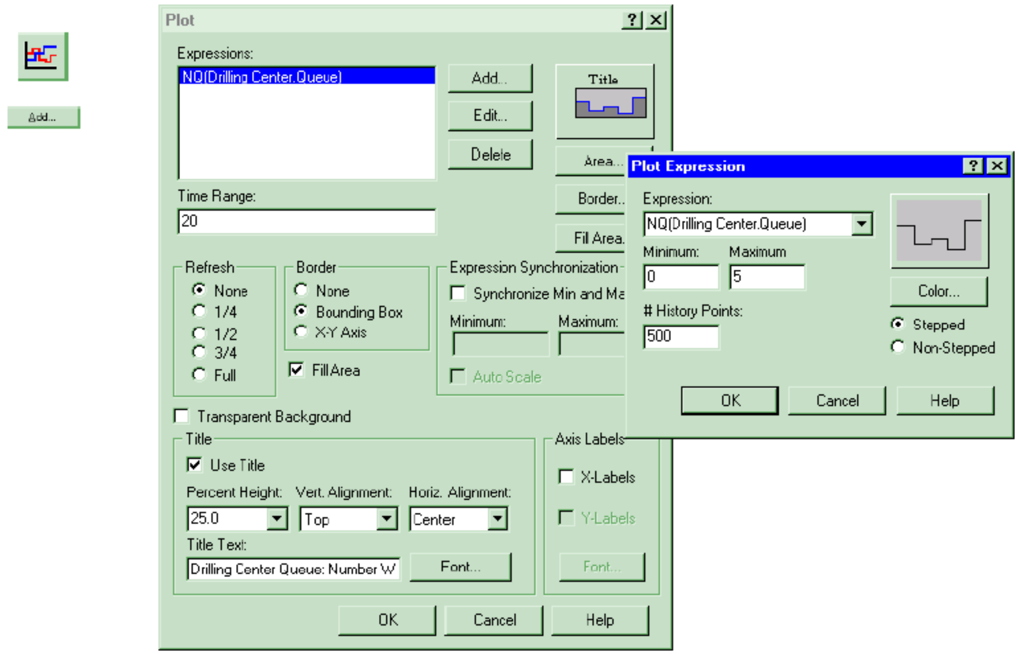
Display for the Dispose Module





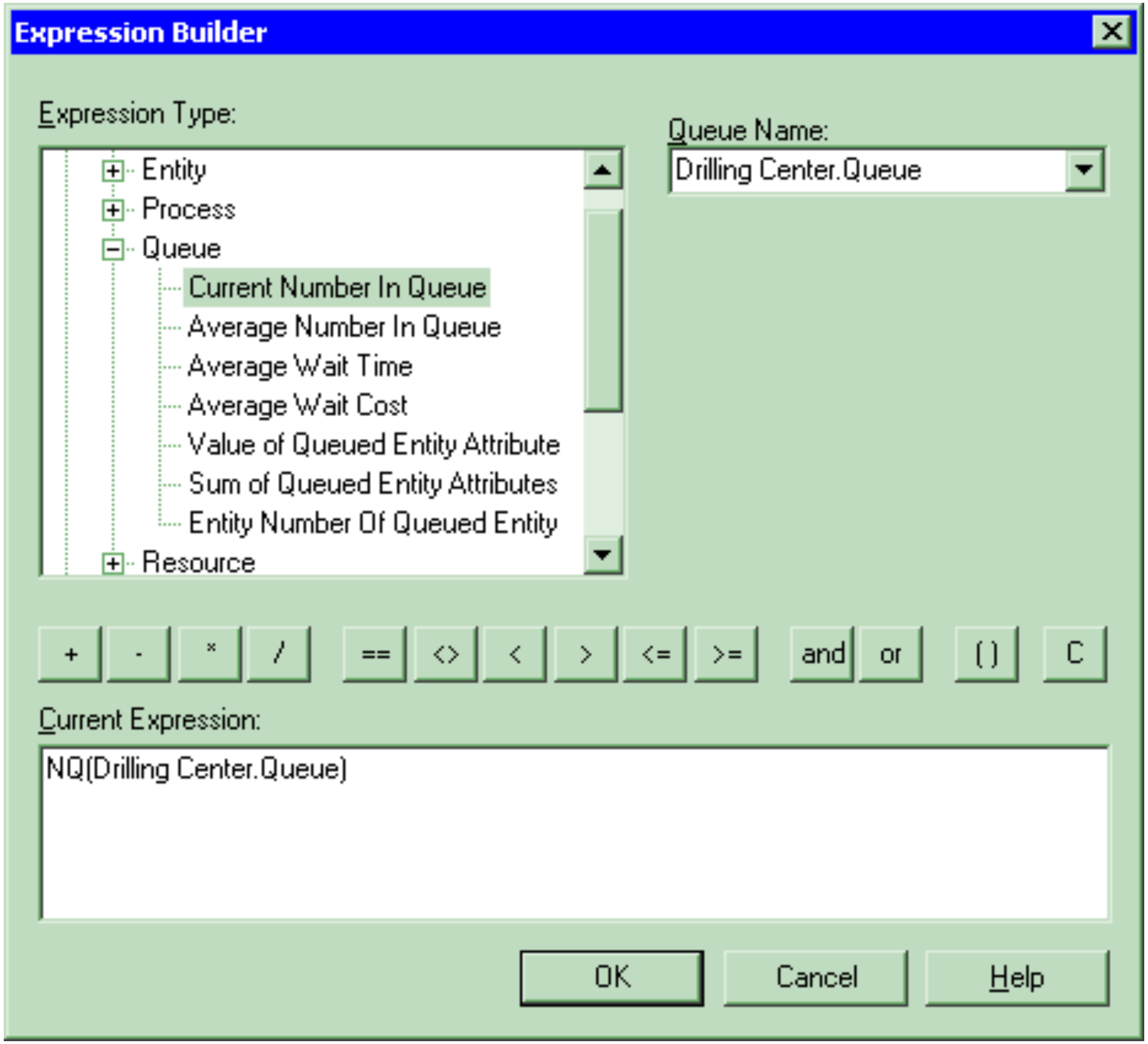
Name	Part Leaves System

Display for the Queue-Length Plot



Plot Expressions (secondary dialog via Add button)	
Expression	`NQ(Drilling Center.Queue)`
Maximum	5
Color	*black*
Plot	
Time Range	20
X-Labels	*clear* (i.e., *uncheck*)
Title - Use Title	*select*
Horiz. Alignment:	*Center*
Title Text:	`Drilling Center Queue: Number Waiting`

Expression Builder for Queue-Length-Plot Expression



More on Menus – File Menu

- Model-file management
- Template attach/detach
- DXF import (from CAD packages), Visio import
- Color palettes
- Printing
- E-mail open model file
- Recent models
- Exit from Arena

Edit Menu

- **Undo/Redo**
- **Cut/Copy/Paste**
- **Paste Link (create OLE link)**
- **Duplicate, Delete selection**
- **Select/Deselect All**
- **Entity Pictures – change content, definition of pictures presented in Entity data module**
- **Calendar Schedules – specify complex time patterns in hierarchies (weeks are made of days, which are made of shifts, etc.), exceptions (holidays), view composite net effect**

Edit Menu (cont'd.)

- **Find – searches all modules and animation objects for a text string … useful for finding wrong names, typos after an error message from Arena**
- **Properties – display internal Arena object properties**
- **Links – to link to other files (spreadsheets, sounds, etc.)**
- **Insert New Object – from other applications (e.g., graphics)**
- **Object – edit object imported from another application**

View Menu

- **Zooming – discussed before**
- **Zoom Factor – step size when zooming**
- **Views – canned Arena views of flowchart view**
- **Named Views – define, change, use views**
- **Grid/Snap/Grid Settings – control grid, snapping**
- **Page breaks – shows page breaks if printed**
- **Data Tips – toggles display of Data Tips**
- **Layers – which objects show up in which mode**
- **Split Screen – toggle for viewing both flowchart and spreadsheet views, or just one of them**
- **Toolbars – decide which toolbars show up**
- **Project/Status Bar – toggle to show up or not**

Tools Menu

- **Separate applications for modeling, analysis**
  - Arena Symbol Factory – large collection of graphics in categories, use to create graphical symbols for animation
  - Input Analyzer – fit probability distributions for input, using field-collected data
  - Process Analyzer – run, compare many "scenarios" at once
    - Also Output Analyzer … not on menus ... start from Start menu
- **Special "editions" of Arena (FactoryAnalyzer, Contact Center) – depends on licensing**
- **Model Documentation Report – generate HTML file with many details of this model**

Tools Menu (cont'd.)

- **Import/Export model to/from Database – bring in, save model details to Excel or Access**
- **OptQuest for Arena – separate application that "takes over" running of the model to search for an optimal scenario**
- **Macro – create Visual Basic macros (mini programs), VB editor ... more in Chapter 10**
- **Options – control many aspects of how Arena works, looks**

Arrange Menu

- For modeling, graphics objects – first select object(s)
- Bring object to Front, Send it to Back — for “stacking” effects
- Group, Ungroup
- Flip around Vertical, Horizontal line
- Rotate object (90° clockwise)
- Align objects on top, bottom, left, or right edges
- Distribute objects evenly (horizontally, vertically)
- Flowchart Alignment – arrange flowchart modules (horizontally, vertically)
- Snap to Grid the selected object(s)
- Change Snap Point on object that gets snapped

Object Menu

- Connect tool – changes cursor to cross hairs
- Auto-Connect newly placed module to selected module – toggle on/off
- Smart Connect – new connections in horizontal/vertical segments rather than one diagonal segment – toggle on/off
- Animate Connectors – to show entities moving (at infinite speed for statistics collection)
- Submodel – define and manage hierarchical submodels (see Chapter 5)

Run Menu

- Setup – control model run conditions
- Entries to run, check, pause, step through
- Alternatives to watch execution, view results (or errors)
- Control how run goes and is displayed
- Most capabilities on Run or Run Interaction Toolbar, and will be described a bit later in detail
- Access the "code" in the underlying SIMAN simulation language

Window Menu

- Cascade, Tile multiple open model windows
- Arrange Icons for any minimized model windows
- Use system Background Color — use Windows colors rather than Arena settings
- List of open model windows

Help Menu

- One of several ways to get into Help system
- Arena Help Topics – TOC, Index, Search
- What's This? – adds ? to cursor, then click on things for brief description
- Release notes – recent changes, requirements
- Arena SMART Files – subject-based index to many small (but complete) models that illustrate specific modeling techniques

Help Menu (cont'd.)

- **List of attached modeling panels – select to get Help on that one**
- **Product Manuals – detailed documents on Arena components**
- **Web links to product support (must be online ...)**
- **Product support/training**
- **Copy protection information for commercial, research, and lab versions**
- **About Arena... – version number, etc.**

More on Toolbars

- **Collections of buttons for "frequent" operations**
  - Most are duplication of menu entries
  - Standard, Draw, Animate, Integration, View, Arrange, Run Interaction, Record Macro, Animate Transfer, Professional
- ***View > Toolbars*** **(or right-click in a toolbar area) to decide which ones show up, which to hide**
- **Toolbars can be torn off ("floating" palettes), or "docked" to an edge of screen**
- **Arena remembers your Toolbar configuration for next time**
- ***View > Toolbars > Customize*** **to alter how toolbars and buttons are displayed**
- **See text for run-through description of toolbars and buttons**

More on Drawing

- Draw via toolbar buttons only (no menus):

- Line, Polyline (hold Shift key for 45^0), Arc, Bézier Curve
- Box, Polygon, Ellipse (fill, line, shade)
- Text (font, size, style)
- Colors for Lines, Fill, Text, Window Background
- Line Style, Fill Pattern
- By far best way to learn: just play around

Printing

- **Print all or parts of flowchart view of active model window – supports color**
- **Usual Print, Print Preview, Print Setup (File menu)**
- **Could consume many pages … also prints named views separately**
  - Print Preview, select only what you want for printing
- ***View > Page Breaks* to show how pages will break**
- **Alternative to printing directly from Arena: PrintScreen key — sends screen to clipboard, paste into another application**
  - Alt+PrintScreen — sends only active window to clipboard
  - Could first pass through a paint application to crop, etc.

Help!

- Extensive, comprehensive online system – including complete (electronic) manuals
- Interlinked via hypertext for cross referencing
- Multiple entry points, including Help menu (described above), links to websites
-  button for context-sensitive help
- 

button in most dialogs
- 

button (What's This?) for info on items
- Tooltips – roll over things, get little sticky note
- Examples folder inside Arena 7.0 folder
- SMARTs library – small models illustrating points – subject index via *Help > Arena Smart Files*

More on Running Models

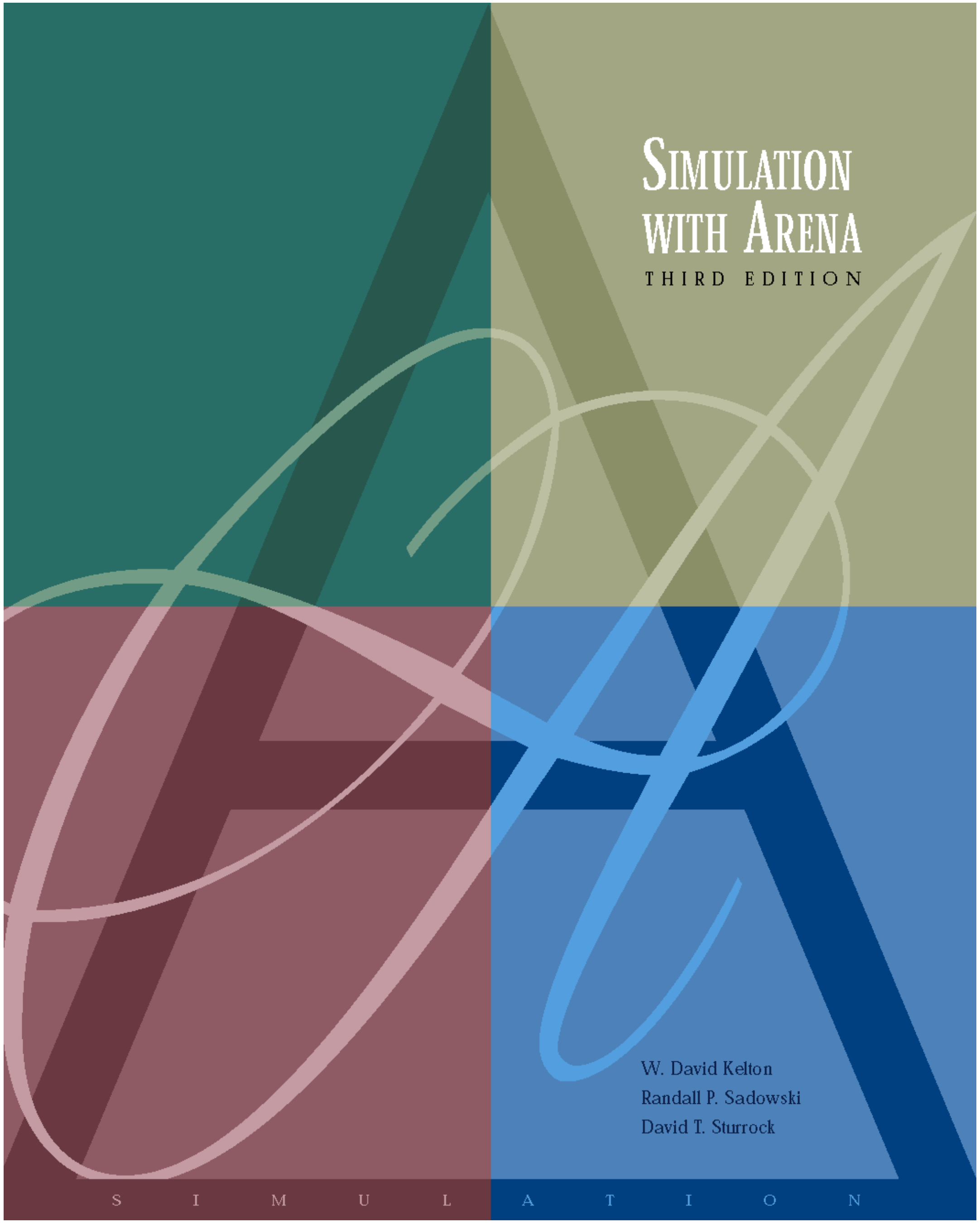
- Run Menu; Standard & Run Interaction toolbars
- ***Run > Setup*** – many options to control the run
  - These are attached to the model, and are not global
- ***Run > Go***  – run simulation "normally" (depends on selections from *Run > Run Control* and *Tools > Options > Run Control*)
- ***Run > Step***  – one "step" at a time (verify, debug)
- ***Run > Fast-Forward***  – disable animation (faster)
- ***Run > Pause***  (or Esc key) – freeze run, resume with Go
- ***Run > Start Over***  – go back to beginning of simulation

More on Running Models (cont'd.)

- ***Run > End*** – get out of run mode
- ***Run > Check Model*** – like compiling
- ***Run > Review Errors*** – for most recent Check
- ***Run > Run Control > Command*** – bring up interactive command-line window to control run
- ***Run > Run Control > Break*** – set times, conditions to interrupt for checks, illustration
- ***Run > Run Control > Watch*** – bring up a window to watch a variable or expression during run

More on Running Models (cont'd.)

- ***Run > Run Control > Break on Module***  – set/clear break when an entity enters or resumes activity on a module
- ***Run > Run Control > Highlight Active Module*** – highlight the flowchart module being executed
- ***Run > Run Control > Batch Run (No Animation)*** – run model with no animation … this is even faster than Fast-Forward … usually used for "production runs" for statistical analysis
- ***Run > SIMAN*** – view or write the model (.mod) and experiment (.exp) files for the underlying SIMAN model



Modeling Basic Operations and Inputs

Chapter 4

Last revision June 7, 2003

What We'll Do ...

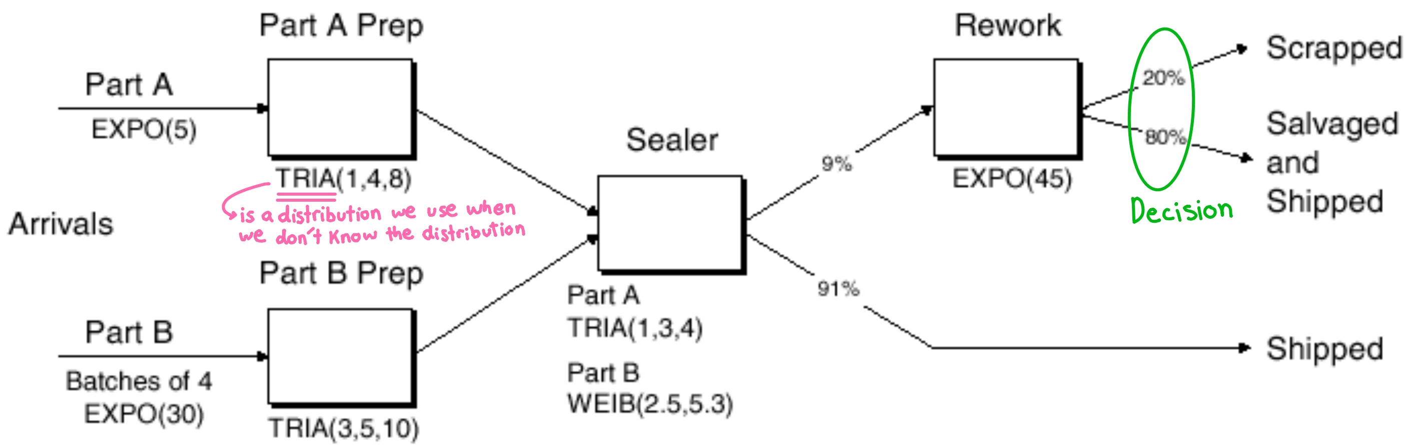
- **Model 4-1: Electronic assembly/test system**
  - Modeling approaches
  - New Arena modules (Decide, Record)
- **Model 4-2: Enhanced electronic assembly/test**
  - Resource Schedules, States, and Failures
  - Frequency outputs
  - More on utilizations
- **Model 4-3: Enhancing the animation**
  - Queues, Entity Pictures, Resource Pictures
  - Adding Plots and Variables

What We'll Do ... (cont'd.)

- **Model 4-4: Adding entity travel times**
  - Modify with Stations, Transfers, Routes, animation of entity movement
- **Input analysis**
  - Specifying input distributions, parameters
  - Deterministic vs. random input
  - Collecting and using data
  - Fitting input distributions via the Input Analyzer
  - No data?
  - Nonstationary arrival processes
  - Multivariate and correlated input data

Electronic Assembly/Test System (Model 4-1)

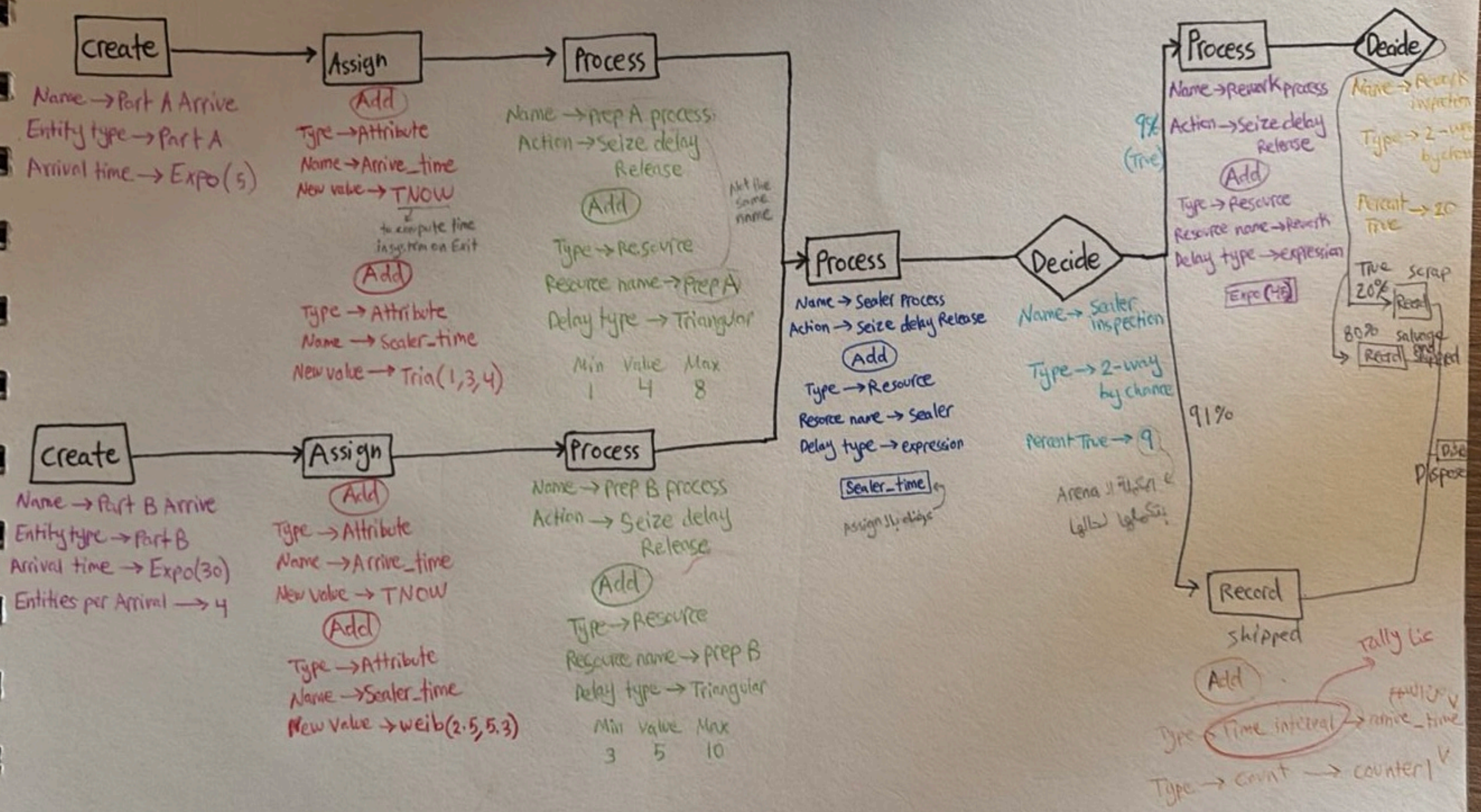
Show the arena simulation for the model described?



- Produce two different sealed elect. units (A, B)
- Arriving parts: cast metal cases machined to accept the electronic parts

the triangular distribution is a part from the possibility theory

- Part A, Part B – separate prep areas
- Both go to Sealer for assembly, testing – then to Shipping (out) if OK, or else to Rework
- Rework – Salvaged (and Shipped), or Scrapped



Create
Name → Part A Arrive
Entity type → Part A
Arrival time → Expo(5)
Assign
Add
Type → Attribute
Name → Arrive_time
New value → TNOW
to compute time in system on Exit
Add
Type → Attribute
Name → Sealer_time
New value → Tria(1,3,4)
Process
Name → prep A process
Action → Seize delay Release
Add
Type → Resource
Resource name → Prep A
Delay type → Triangular
Min Value Max
1 4 8
Not the same name
Create
Name → Part B Arrive
Entity type → Part B
Arrival time → Expo(30)
Entities per Arrival → 4
Assign
Add
Type → Attribute
Name → Arrive_time
New Value → TNOW
Add
Type → Attribute
Name → Sealer_time
New Value → weib(2.5,5.3)
Process
Name → Prep B process
Action → Seize delay Release
Add
Type → Resource
Resource name → prep B
Delay type → Triangular
Min Value Max
3 5 10
Process
Name → Sealer Process
Action → Seize delay Release
Add
Type → Resource
Resource name → Sealer
Delay type → expression
Sealer_time
Decide
Name → Sealer inspection
Type → 2-way by chance
Percent True → 9
9% (True)
91%
Process
Name → Rework process
Action → Seize delay Release
Add
Type → Resource
Resource name → Rework
Delay type → expression
Expo(45)
Decide
Name → Rework inspection
Type → 2-way by chance
Percent True → 20
True 20%
Record
scrap
80%
Record
salvaged and shipped
Dispose
Record
shipped
Add
Type → Time interval → arrive_time
Tally
Type → count → counter1

Part A

- **Interarrivals: expo (5) minutes**
- **From arrival point, proceed immediately to Part A Prep area**
  - Process = (machine + deburr + clean) ~ tria (1,4,8) minutes
- **Go immediately to Sealer**
  - Process = (assemble + test) ~ tria (1,3,4) min.
  - 91% pass, go to Shipped; Else go to Rework
- **Rework: (re-process + testing) ~ expo (45)**
  - 80% pass, go to Salvaged; Else go to Scrapped

Part B

entities per arrival

- **Interarrivals: *batches* of 4, expo (30) min.**
- **Upon arrival, batch separates into 4 individual parts**
- **From arrival point, proceed immediately to Part B Prep area**
  - Process = (machine + deburr +clean) ~ tria (3,5,10)
- **Go to Sealer**
  - Process = (assemble + test) ~ weib (2.5, 5.3) min. , *different* from Part A, though at same station

    Attribute
  - 91% pass, go to Shipped; Else go to Rework
- **Rework: (re-process + test) = expo (45) min.**
  - 80% pass, go to Salvaged; Else go to Scrapped

Run Conditions, Output

- **Start empty & idle, run for four 8-hour shifts (1,920 minutes)**
- **Collect statistics for each work area on**
  - Resource utilization
  - Number in queue
  - Time in queue

→ They are collected Automatically in Arena Simulation

- **For each exit point (Shipped, Salvaged, Scrapped), collect total time in system (a.k.a. cycle time)**

Time in system → لأنه يختلف من Entity إلى Entity
يعني اشي Attribute
و مجرد ما دخل ال system لازم اعمل
Assign → Attribute → TNOW

Developing a Modeling Approach

- **Define pieces of model, modules, data structures, control logic**
- **Appropriate level of detail – judgment call**
- **Often multiple ways to model, represent logic**
- **This model:**
  - Entities are the individual parts (two types)
  - Separate Create modules for two part types
  - Separate Process modules for each Prep area
  - Process modules for Sealer and Rework, each followed by a Decide module (2-way by Chance)
  - تم إنقاذه تم إلغاؤه
    Depart modules for Shipped, Salvaged, Scrapped
  - Attribute **`Sealer Time`** assigned after Creates in Assign modules (parts have *different* times at *the* Sealer)
  - Record modules just before Departs for time in system

Building the Model

- **New model window**
- **Attach Basic Process panel (if needed)**
- **Place modules**
  - Create (x 2) A B
  - Assign (x 2)
  - Process (x 4)
  - Decide (x 2)
  - Record (x 3)
  - Dispose (x 3)
- **Right click — repeat last action (place module)**
- **Auto-Connect, or manually connect via** 

Alternate strategy – place one module at a time, fill it out completely

الـ (Entity) هي الشيء الذي يتحرك

Part A Create Module

لازم يكون اسم مستقل
(مونفس الـ Entity ولانفس الـ Resource)

الـ (Capacity) لها علاقة بالـ Resource
مثلاً عدد الناس يلي بشتغلو
عدد الـ Machines
و هكذا

- Name: **`Part A Arrive`**
- **Entity Type:** **`Part A`**
- **Time Between Arrivals**
  - Type: **`Random (Expo)`**
    - Pull-down list with options
  - Value: **5**
  - Units: **`Minutes`**
    - Pull-down list with options
- **Default what's not mentioned above**

Once these entries are made, they are placed on the list for names of that type (Module Name, Entity Type, etc.) and will appear on future pull-down lists for that type of name.

Part B Create Module

- **Name: `Part B Arrive`**
- **Entity Type: `Part B`**
- **Time Between Arrivals**
  - Type: **`Random (Expo)`**
    - Pull-down list with options
  - Value: **`30`**
  - Units: **`Minutes`**
    - Pull-down list with options
- **Entities per Arrival: `4`**

How is the batch size modelled?

from create → Entities per arrival → (4)

Part A Attributes Assign Module

- **Name:** `Assign Part A Sealer and Arrive Time`
- **Add button:**
  - Type: `Attribute`
  - Attribute Name: `Sealer Time`
  - New Value: `TRIA(1, 3, 4)`

يجب أن يكون
Pre Assigned

- **Add button:**
  - Type: `Attribute`
  - Attribute Name: `Arrive Time`
  - New Value: `TNOW` (to compute time in system on exit)

TNOW is the internal Arena variable name for the simulation clock; see *Help > Arena Help > Contents >*
Variables, Functions, and Distributions > Date and Time Variables

Part B Attributes Assign Module

- **Name:** **`Assign Part B Sealer and Arrive Time`**
- **Add button:**
  - Type: **`Attribute`**
  - Attribute Name: **`Sealer Time`**
  - New Value: **`WEIB(2.5, 5.3)`**
- **Add button:**
  - Type: **`Attribute`**
  - Attribute Name: **`Arrive Time`**
  - New Value: **`TNOW`**

Names for things in Arena
– Default names usually suggested
– Names placed on appropriate pull-down lists for future reference
– All names in a model must be unique (even across different kinds of objects)

Process Module *Actions*

- **Delay**

  بستخدمه اذا ما عندي معلومات كافية عن ال Resources فبعتبر (No Resource) و هي من أضعف أنواع البرمجة

  Entity just sits here for the specified time; no Resource involved, so multiple entities could be undergoing this Delay simultaneously

- **Seize Delay**

  ترتيب ال seize مهم
  the most scarced should be at the end

  (مثلاً الدكتور بيجي بعد حجز غرفة العملية و المريض)

  Entity must first Seize the specified number of units of a Resource (possibility for Queueing if they're not available), then undergoes the Delay … assume that the entity will Release the Resource units at another downstream module

- **Seize Delay Release**

  اخلي العامل يشتغل اشي ثاني بعد ما يخلص

  Like Seize Delay, but entity releases Resource units after Delay (*what we want in this model*)

- **Delay Release**

  Assumes entity had already Seized Resource units at another upstream module, now Delays and Releases Resource units

Prep A Process Module

- **Name: `Prep A Process`**
- **Action: `Seize Delay Release`**
- **Resources subdialog (Add button):**
  - Type: **`Resource`** (a pull-down option)
  - Resource Name: **`Prep A`**
  - Quantity: `1` (default)
- **Delay Type: `Triangular`**
- **Units: `Minutes`**
- **Minimum: `1`**
- **Value (Most Likely): `4`**
- **Maximum: `8`**

ما بصير نسميهم
نفس الاسم

If several Resources were named (Add button), entity would have to Seize them all before the Delay could start.

Prep B Process Module

- **Name:** `Prep B Process`
- **Action:** `Seize Delay Release`
- **Resources subdialog (Add button):**
  - Type: `Resource` (a pull-down option)
  - Resource Name: `Prep B`
  - Quantity: `1` (default)
- **Delay Type:** `Triangular`
- **Units:** `Minutes`
- **Minimum:** `3`
- **Value (Most Likely):** `5`
- **Maximum:** `10`

Sealer Process Module

- **Name: `Sealer Process`**
- **Action: `Seize Delay Release`**
- **Resources subdialog (Add button):**
  - Type: **`Resource`** (a pull-down option)
  - Resource Name: **`Sealer`**
  - Quantity: **`1`** (default)
- **Delay Type: `Expression`**
- **Units: `Minutes`**
- **Expression: `Sealer Time`**

عرفناه بالا Assign

Recall – **`Sealer Time`** attribute was defined upstream for both Parts A and B … now its value is being used … allows for different distributions for A and B.

Sealer Inspection-Result *Decide* Module

- **Decide module provides branch points**
  - *By Condition* (entity Attributes, global Variables)
  - *By Chance* (multi-sided, possibly-biased hypercoin flip)
- **Name: `Failed Sealer Inspection`**

* مثلاً اذا كانو ٣ بالDecide
بحط تنين و التالتة ما بحطها لأنها (compliment)

- **Type: `2-way by Chance` (default)**
- **Percent True: `9`**
- **Different exit points for True, False results – connect appropriately downstream**

> − Note it's *percent* true, not *probability* of true … so "9" means probability of 0.09
> − We arbitrarily decided "true" meant part failed inspection … could have reversed (but would change numerical results ... why? ... does this upset you? ... why?)
> − This is a rich, deep, versatile module … explore its Help button

Rework Process Module

- **Name:** `Rework Process`
- **Action:** `Seize Delay Release`
- **Resources subdialog (Add button):**
  - Type: **`Resource`** (a pull-down option)
  - Resource Name: **`Rework`**
  - Quantity: **`1`** (default)
- **Delay Type:** `Expression`
- **Units:** `Minutes`
- **Expression:** `EXPO(45)`

كما نلاحظ لازم
يكونو مختلفين عن بعض

Had to use the general **`Expression`** choice for Delay Type since what we want (**`EXPO`**) is not directly on the Delay Type pull-down list.

Rework Inspection-Result Decide Module

- **Name:** `Failed Rework Inspection`
- **Type:** `2-way by Chance` **(default)**
- **Percent True:** `20`

We arbitrarily decided "true" meant part failed inspection

Record Modules

- **Arena collects and reports many output statistics by default, but sometimes not all you want**
- **We want time in system (average, max) of parts sorted out by their exit point (Shipped, Salvaged, Scrapped)**
  - It's this sorting that Arena doesn't do by default ... it would automatically sort *by Entity Type* if we had Entities checked in Run > Setup > Project Parameters (which we don't)
- **Record module can be placed in the flowchart to collect and report various kinds of statistics from within the model run as entities pass through it**
- **For Tally-type output performance measures (see Chapter 3)**

Shipped Parts *Record* Module

- **Name:** `Record Shipped Parts`
- **Type:** `Time Interval`
  - This option records the length of time that elapsed up to now (`TNOW`) from when an entity attribute was marked with a time “stamp” upstream ... Attribute Name is below ...
  - There are several other options for Type … explore via Record module’s Help button!
- **Attribute Name:** `Arrive Time`
  - Was defined upstream as the clock value in the Assign modules instantly after each entity was Created
- **Tally Name:** `Record Shipped Parts`
  - Determines the label in the reports

Other two Record modules – just like this except for Name and Tally Name.

Describe how to record time in system? And the parts produced?

a

in Assign we Add an Attribute and we name it (Arrivetime) → new value (TNow)

and then in the Record → add then choose time interval ☑ and Attribute name → Arrivetime

assigned previously

b And the parts produced → in the Record → Add → Type (count) → counter Name (counter 1)

Dispose Modules

- **Three separate exit points for three separate part disposition (Shipped, Salvaged, Scrapped)**
- **Could have directed all three exit types to a single Dispose module**
  - But having separate ones produces animation counts of the three dispositions
- **Also, having separate Dispose modules allows for differentially checking the boxes to Record Entity Statistics**
  - Produces flow statistics separated by entity type (*if* Entities Statistics Collection is checked in *Run > Setup > Project Parameters*), *not* by final disposition of part … so we *did* need our Record modules and Arrive Time attribute

What are the replications for? How to add them?

Run > Setup for Run Control

They are basically for the number of repetition in the system to not let the model run forever because the will affect the results

we can add them from → Run tab → setup → Replication parameters → number, Length

- **Without this, model would run forever – no defaults for termination rule**
  - That's part of modeling, and generally affects results!
- **Project Parameters tab:**
  - Fill in Project Title, Analyst Name
  - Defaults for Statistics Collection, but we cleared the check box for Entities – not needed for what we want (we installed our own Record modules), and would slow execution
- **Replication Parameters tab:**
  - Replication length: **`32`**, accept **`Hours`** default for Time Units
  - Base Time Units: **`Minutes`** for inputs without Time Units option, internal arithmetic, and units on output reports

Different Part A, B Entity Pictures

- **Entity data module (just single-click on it in Project Bar, edit via spreadsheet only)**
- **Row for each Entity Type (`Part A`, `Part B`)**
- **Pull down Initial Picture pull-down menu, select different pictures for each Entity Type**
  - *Edit > Entity Pictures* to see, change the list of pictures that's presented here … more later

Describe how change the entity picture before run and during the replicate?

* How can I change the picture of a part or Entity

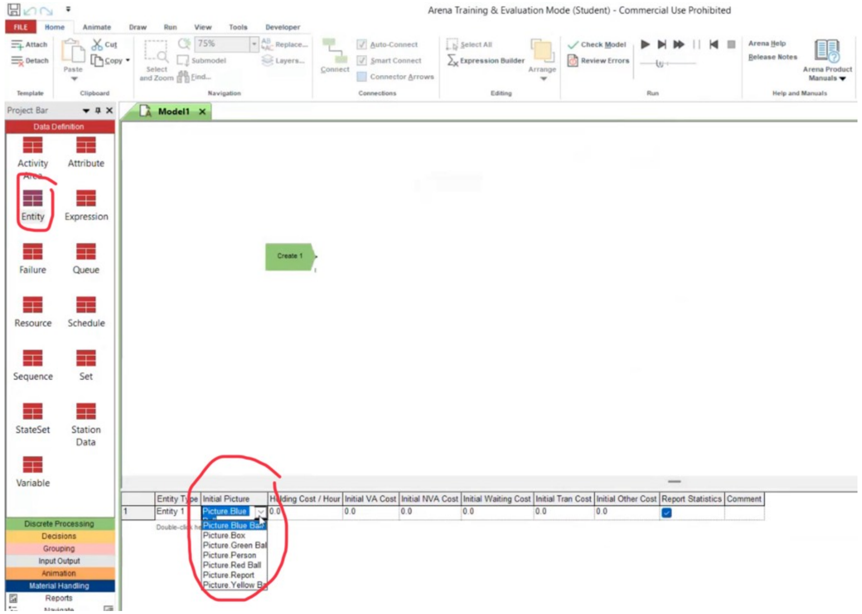
→ Arena → Data Definition → Entity → in Table down → initial picture

Scroll down بالسهم

و هكذا نغير صورة ال Entity

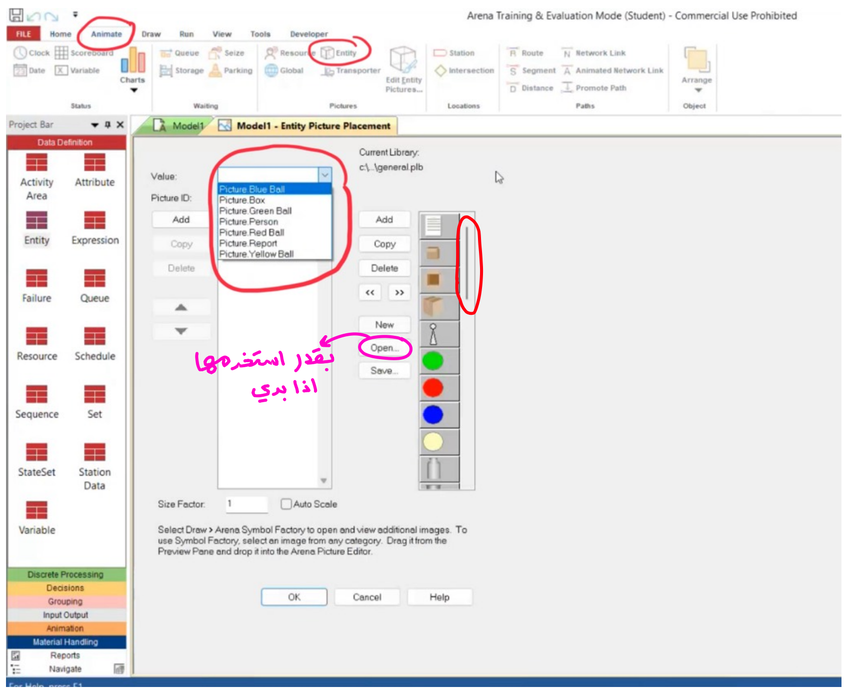
OR

→ Arena → Animate → Entity → scroll down



Method ①

أو



Method 2

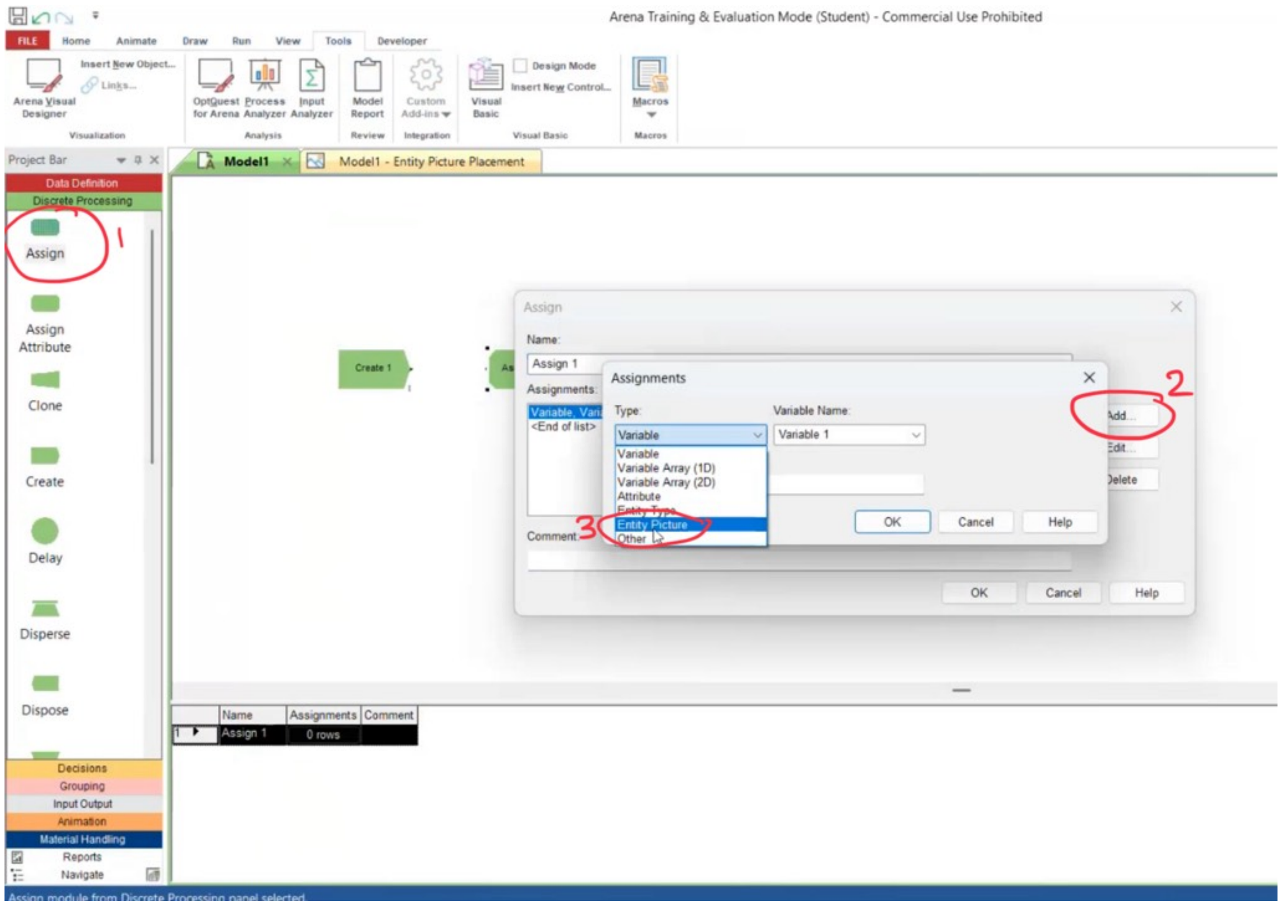
→ Arena

→ Discrete processing

→ Assign

→ Add (type Entity Picture)

→ scroll down then choose Entity Picture



What does this symbol represent

Running the Model

Describe the job of the different buttons?

- **Check ✓ (if desired)**
  - Find button to help find errors
- **Go ▶ (will automatically pre-Check if needed)**
  - Some graphics don't show during run … will return when you End your run … control via *View > Layers*
  - Status Bar shows run progress – replication number, simulation time, simulation status
- **Animation speed – increase (>), decrease (<)**
- **Pause (❚❚) or Esc key; ▶ to resume**
- ***Run > Step* (▶❚) to debug**
- ***Run > Fast-Forward* (▶▶) to turn off animation**
  - *Run > Run Control > Batch Run (No Animation)* is fastest

capacity schedule → هي متعلقة بال Resource

Schedule → Capacity
Schedule → Arrival

Failure two types → time
Failure two types → count

Value Duration

حسب شو يلي حطيته بال time units

is always per hour

Viewing the Results

- **Counters during animation for modules**
  - Create, Dispose, Decide – incremented when entity leaves
  - Process – number of entities currently in the module
- **Asked at end if you want to see reports**
  - What you get depends on *Run* > *Setup* > *Project Parameters*
    - *Looks like* the Rework area is a bottleneck ... more later
  - Navigate through report with browsing arrows, tree at left
  - Tally, Time-Persistent, and Counter statistics
  - Avg, Min, Max, and 95% Confidence Interval half-widths
    - Confidence intervals are for steady-state expectations … more later
    - May not be produced if run is not long enough for reliable stats
- **Generally difficult/unreliable to draw conclusions from just one run … more later**

Model 4-2: The Enhanced Electronic Assembly and Test System

- **A Story**
  - Original model shown to production manager
  - Pointed out that this is only the first shift of a two-shift day — on second shift there are two operators at Rework (the bottleneck station) … 16-hour days
  - Pointed out that the Sealer fails sometimes
    - Uptimes ~ exponential, mean 2 hours
    - Repair times ~ exponential, mean 4 minutes
  - Wants to buy racks to hold rework queue
    - A rack holds 10 parts
    - How many racks should be bought?
  - Run for 10 days
- **Need:** ***Resource Schedules***, ***Resource States***, ***Resource Failures***

Modify the model to include the following ?

Given the report make the decision on the number of racks?

Change Run Conditions

- Redefine a "day" to be 16 hours – ***Run > Setup > Replication Parameters***

Run Tab → Setup → Replication parametes → Hours per day → 16

- Change Replication Length to 10 (of these) days

Run for 10 days → Run Tab → Setup → Replication parametes → Replication Length → 10 days

How can you make the above settings?

مثلاً ممكن يجي سؤال شو معنى ignore بالResource

Describe how to model the lunch hour without failres?

Schedules

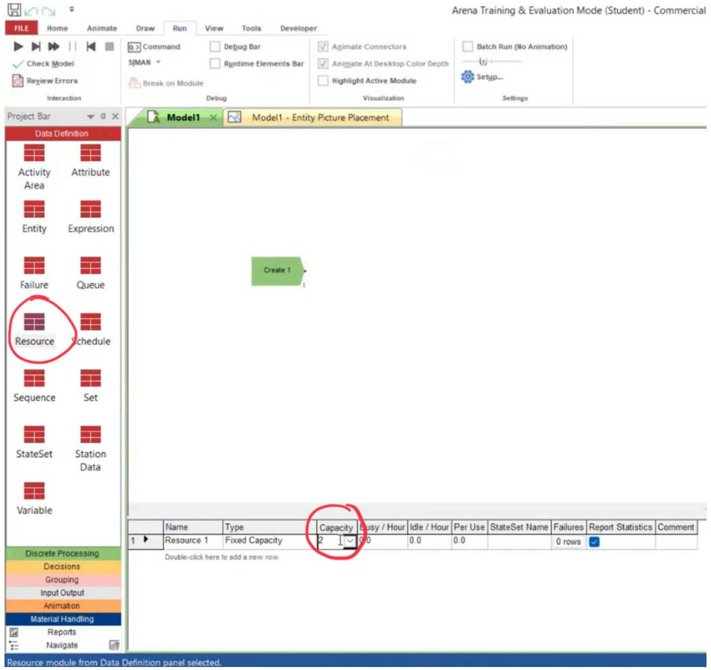
Describe how to model workers available for work fro 8-12 and 1 till 4?

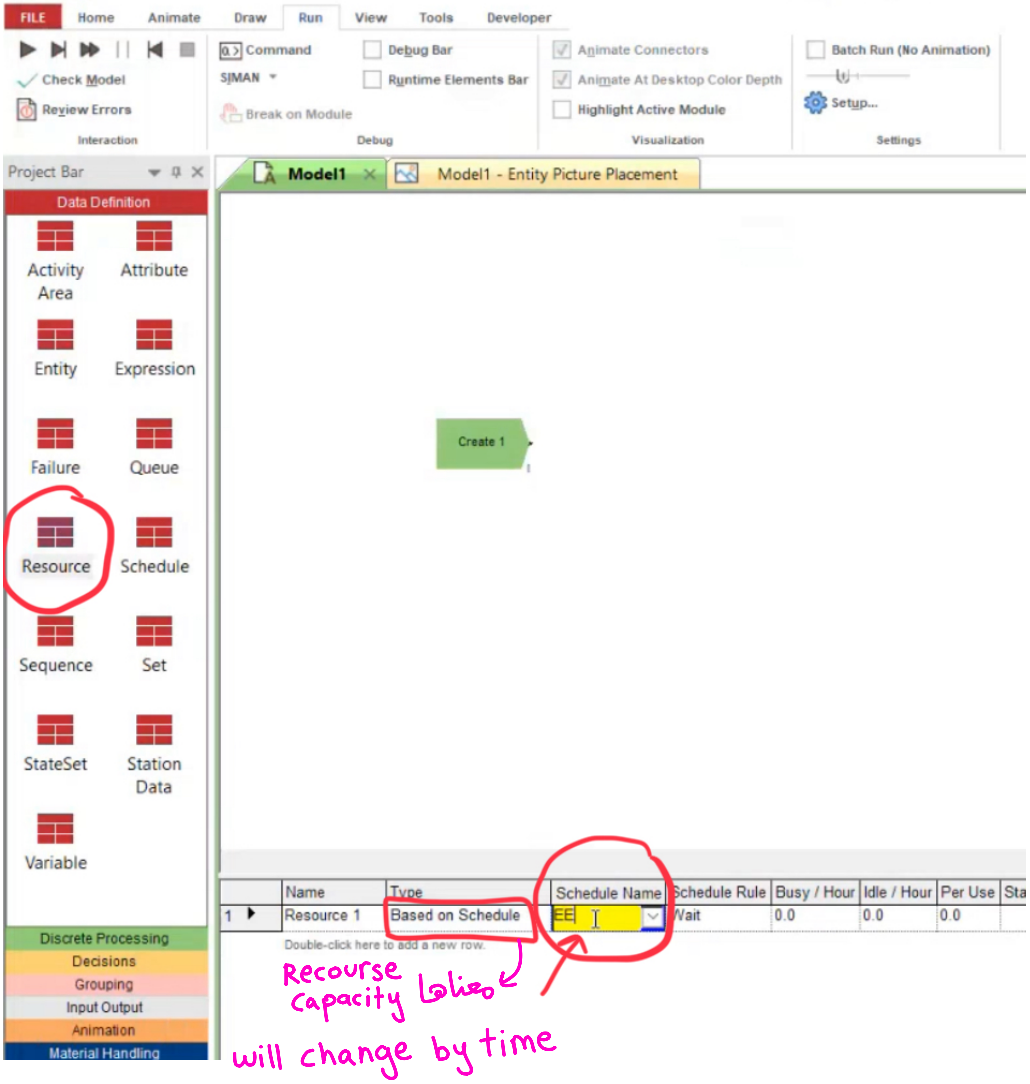
- **Vary Capacity (number of units) of a resource over time**
- **In Resource Data module (spreadsheet view)**
  - For Rework Resource, change Type from **`Fixed Capacity`** to **`Based on Schedule`**
  - Two new columns – Schedule Name and Schedule Rule
  - Type in a schedule name (**`Rework Schedule`**)
  - Select a Schedule Rule – details of capacity decrease if the Resource is allocated to an entity

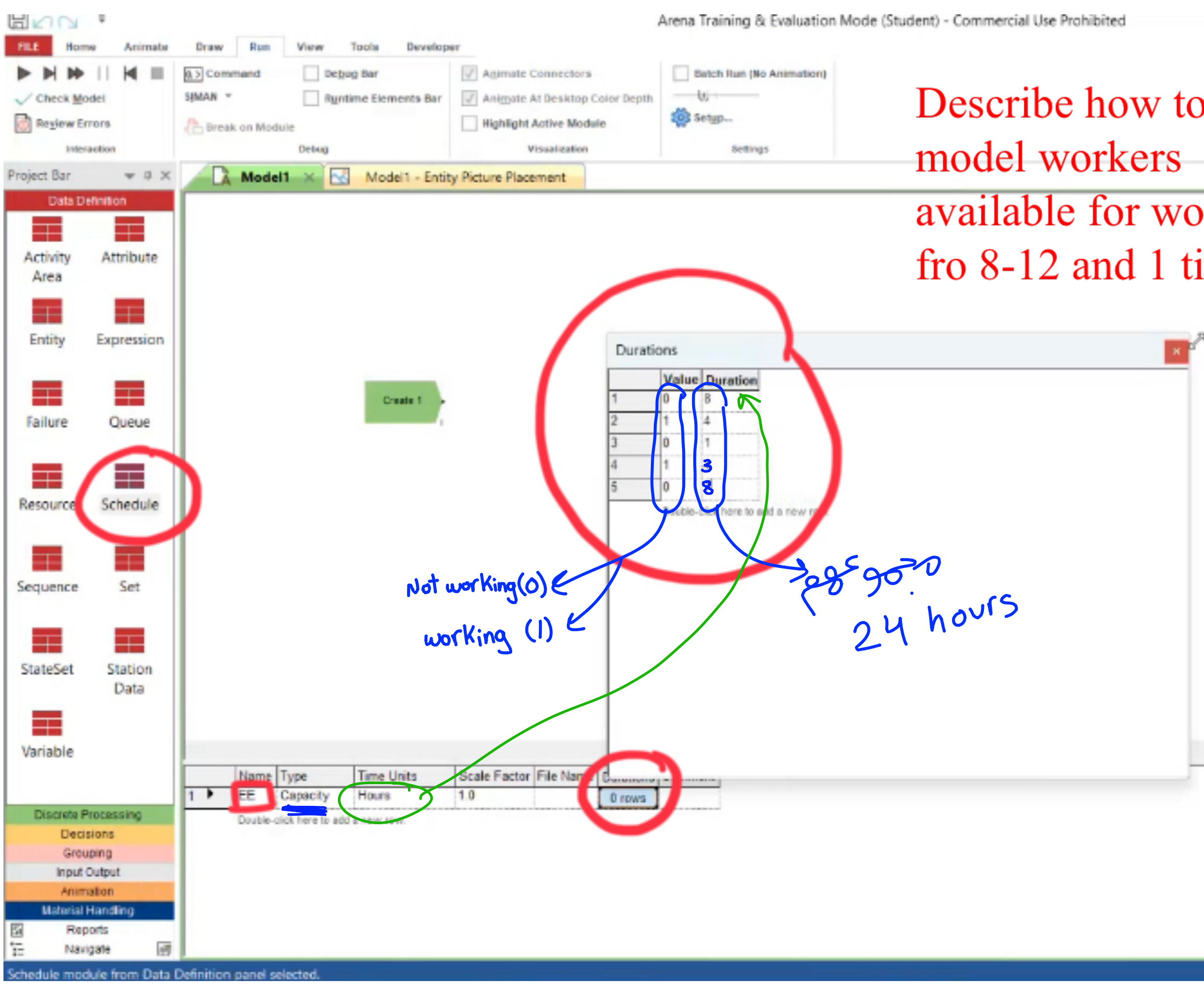
Describe and illustrate the schedule rules?

  - *Ignore* – Capacity goes down immediately for stat collection, but work goes on until finished … “break” could be shorter or gone
  - *Wait* – Capacity decrease waits until entity releases Resource, and “break” will be full but maybe start/end late
  - *Preempt* – Processing is interrupted, resumed at end of “break”

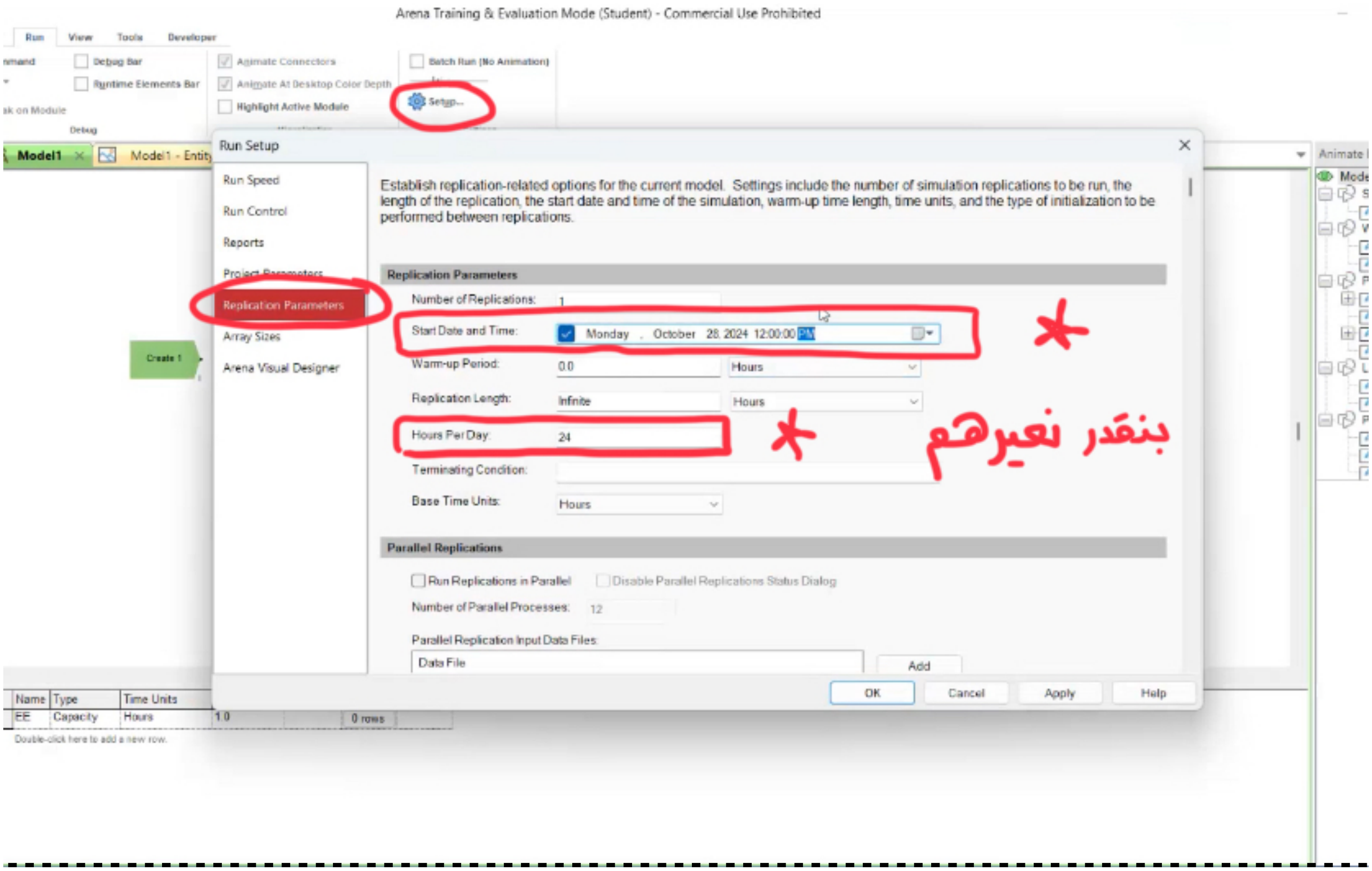
* How can I modify the (Capacity) of the (Resources)

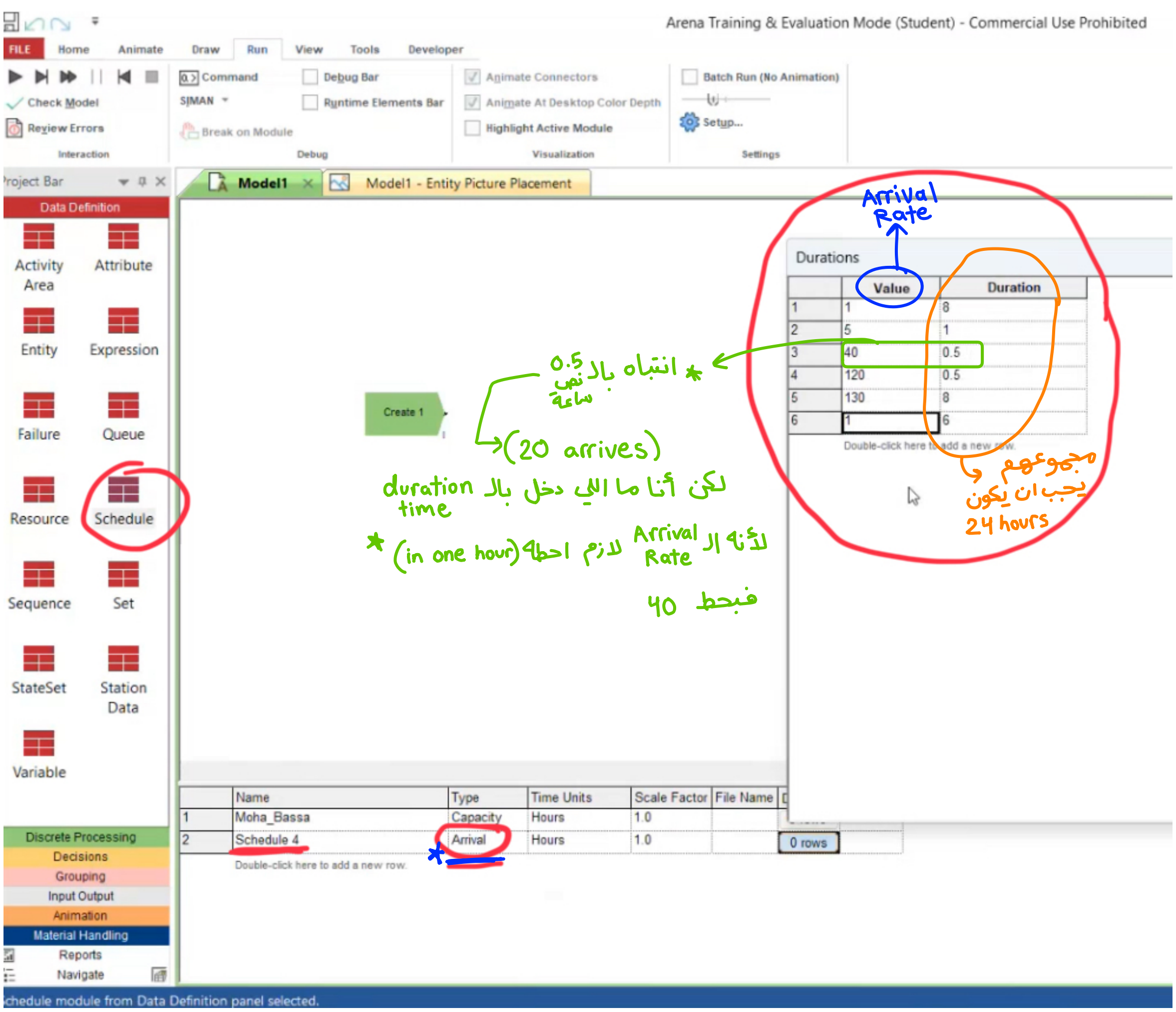


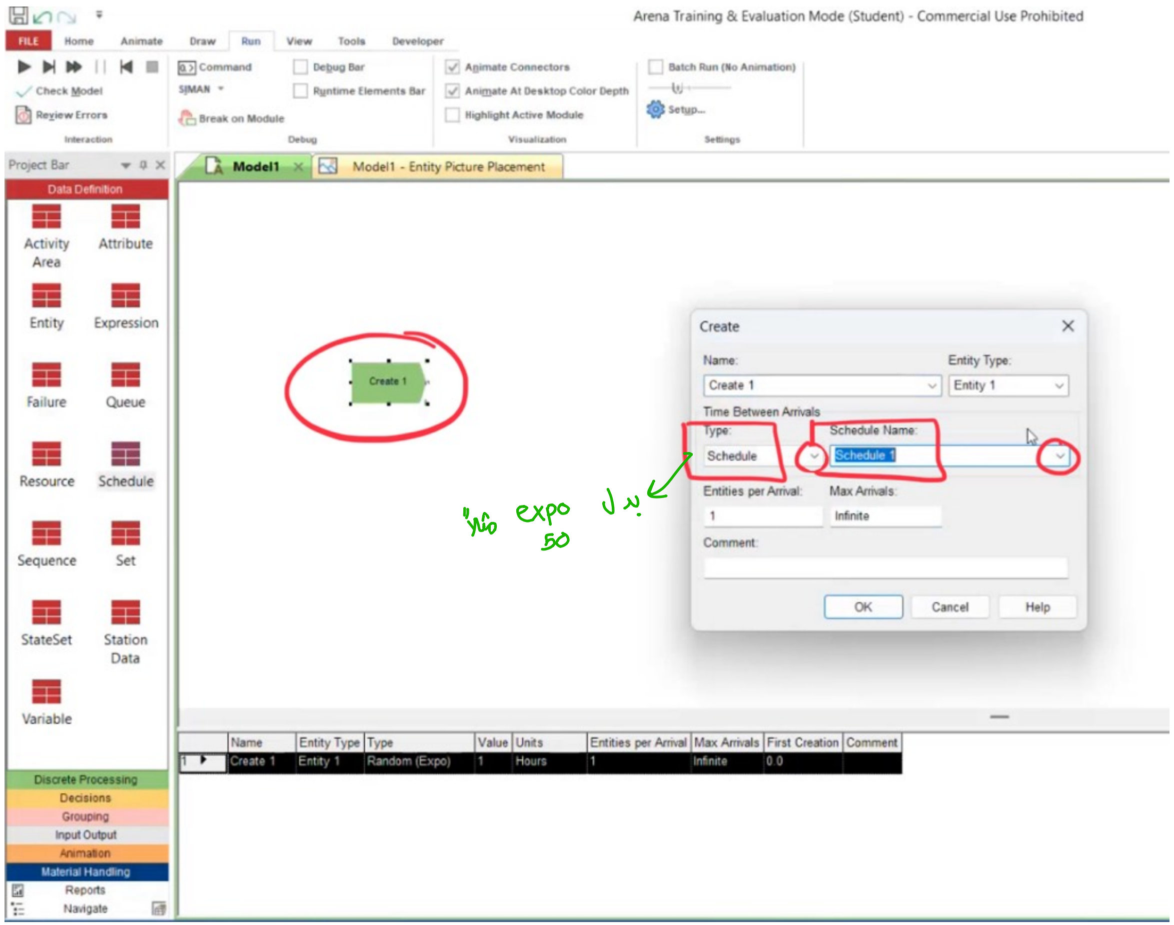




Describe how to model workers available for work fro 8-12 and 1 till 4?


Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
Run Setup
Run Speed
Run Control
Reports
Project Parameters
Replication Parameters
Array Sizes
Arena Visual Designer
Establish replication-related options for the current model. Settings include the number of simulation replications to be run, the length of the replication, the start date and time of the simulation, warm-up time length, time units, and the type of initialization to be performed between replications.
Replication Parameters
Number of Replications: 1
Start Date and Time: Monday , October 28, 2024 12:00:00 PM
Warm-up Period: 0.0 Hours
Replication Length: Infinite Hours
Hours Per Day: 24
Terminating Condition:
Base Time Units: Hours
بنقدر نغيرهم
Parallel Replications
Run Replications in Parallel
Disable Parallel Replications Status Dialog
Number of Parallel Processes: 12
Parallel Replication Input Data Files
Data File
Add
OK
Cancel
Apply
Help


Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
Data Definition
Activity Area
Attribute
Entity
Expression
Failure
Queue
Resource
Schedule
Sequence
Set
StateSet
Station Data
Variable
Model1
Model1 - Entity Picture Placement
Create 1
Durations
Arrival Rate
Value
Duration
1 1 8
2 5 1
3 40 0.5
4 120 0.5
5 130 8
6 1 6
Double-click here to add a new row.
0.5 انتباه بال
نص ساعة
(20 arrives)
لكن أنا ما الي دخل بال duration time
لأنو ال Arrival Rate لازم احطه (in one hour)
فبحط 40
مجموعهم يجب ان يكون 24 hours
Name Type Time Units Scale Factor File Name
1 Moha_Bassa Capacity Hours 1.0
2 Schedule 4 Arrival Hours 1.0
0 rows
Schedule module from Data Definition panel selected.


Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
FILE Home Animate Draw Run View Tools Developer
Model1 - Entity Picture Placement
Project Bar
Data Definition
Activity Area
Attribute
Entity
Expression
Failure
Queue
Resource
Schedule
Sequence
Set
StateSet
Station Data
Variable
Create 1
Create
Name:
Create 1
Entity Type:
Entity 1
Time Between Arrivals
Type:
Schedule
Schedule Name:
Schedule 1
Entities per Arrival:
1
Max Arrivals:
Infinite
Comment:
OK
Cancel
Help
بدل expo مثلاً 50
Name | Entity Type | Type | Value | Units | Entities per Arrival | Max Arrivals | First Creation | Comment
Create 1 | Entity 1 | Random (Expo) | 1 | Hours | 1 | Infinite | 0.0

Schedules (cont'd.)

Capacity

Arrival

- **Define the actual Schedule the Resource will follow – Schedule data module (spreadsheet)**
  - Row already there since we defined **`Rework Schedule`** in Resource
  - Format Type is Duration for entries based on time past simulation beginning
  - Type is Capacity, for Resource schedule (more later on Arrival Type)
  - Click in Durations column, get Graphical Schedule Editor
    - *X*-axis is time, *Y*-axis is Resource Capacity
    - Click and drag to define the graph
    - Options button to control axis scaling, time slots in editor, whether schedule loops or stays at a final level forever
    - Can use Graphical Schedule Editor only if time durations are integers, and there are no Variables or Expressions involved

Schedules (cont'd.)

- Alternatively, right-click in the row, select Edit via Dialog
  - Enter schedule Name
  - Enter pairs for Capacity, Duration … as many pairs as needed
    - If all durations are specified, schedule repeats forever
    - If any duration is empty, it defaults to infinity
  - Can involve Variables, Expressions
- Another alternative – right-click in the row, select Edit via Spreadsheet
  - Enter capacity Value, Duration pairs

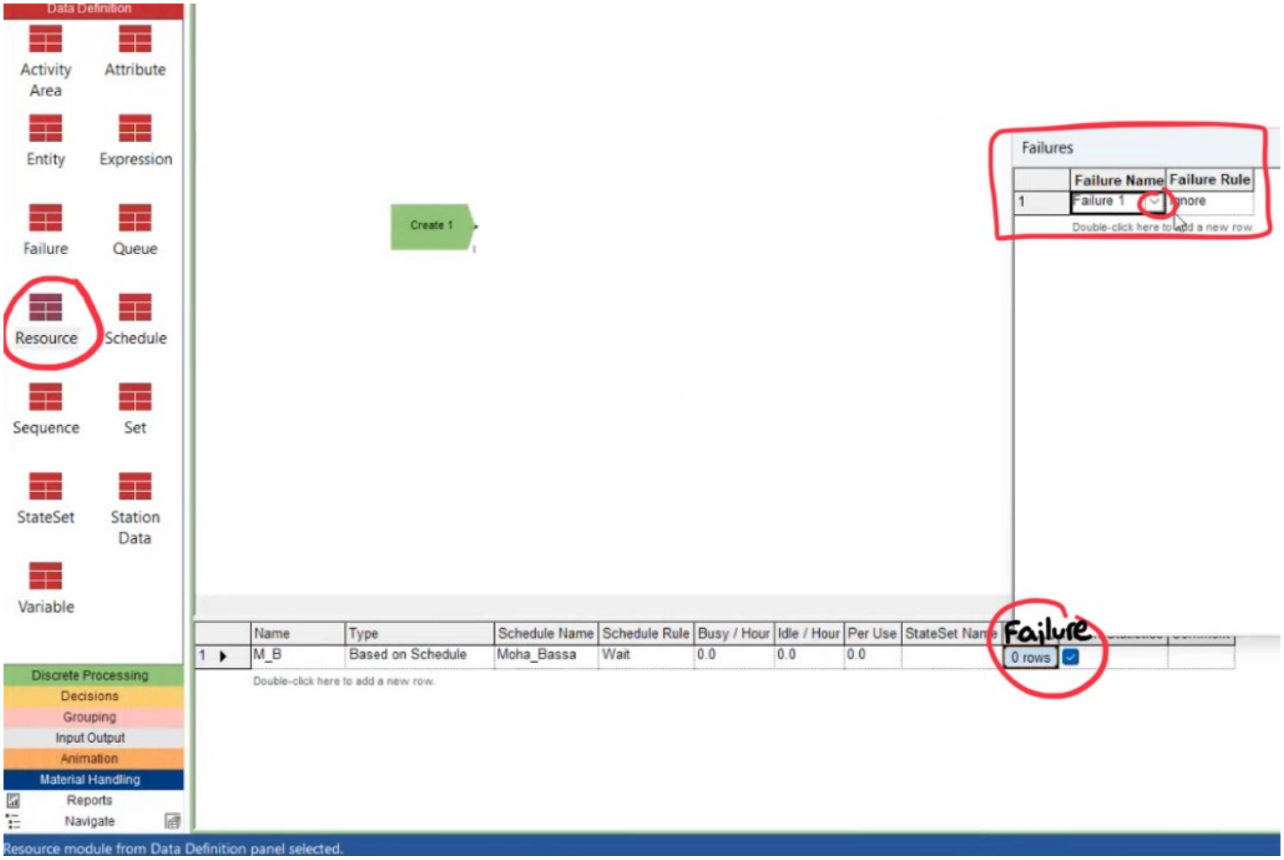
Resource Failures

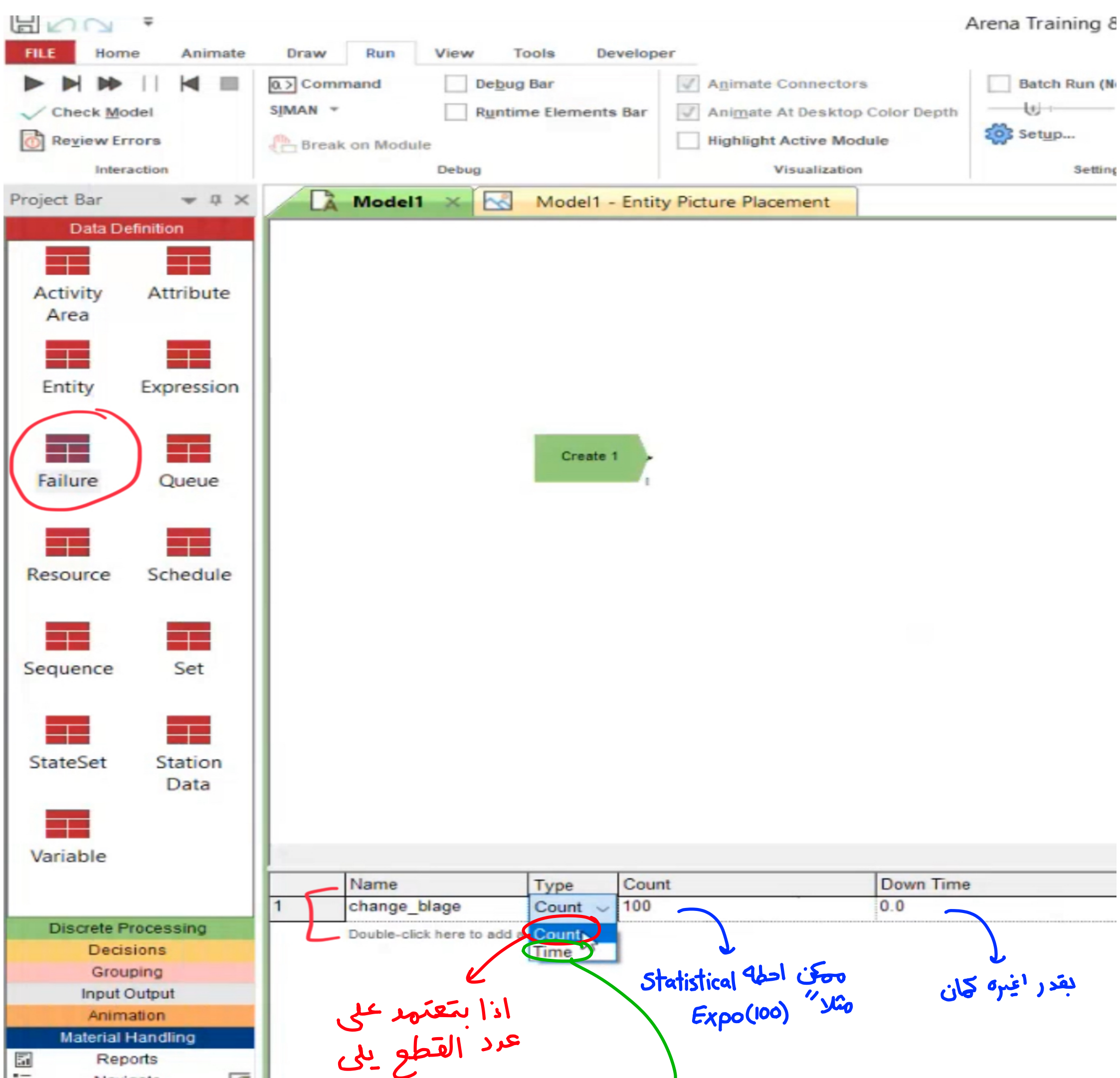
type → count based, time based

- Usually used to model unplanned, random downtimes
- Can start definition in Resource or Failure module (Advanced Process panel) … we'll start in Failure
- Attach Advanced Process panel if needed, single-click on Failure, get spreadsheet view
- To create new Failure, double-click – add new row
- Name the Failure
- Type – Time-based, Count-based (we'll do Time)
- Specify Up, Down Time, with Units

Resource Failures (cont'd.)

- **Attach this Failure to the correct Resource**
  - Resource module, Failures column, Sealer row – click
  - Get pop-up Failures window, pick Failure Name **`Sealer Failure`** from pull-down list
  - Choose Failure Rule from **`Wait`**, **`Ignore`**, **`Preempt`** (as in Schedules)
- **Can have multiple Failures (separate names)**
- **Can re-use defined Failures for multiple Resources (operate independently)**





اذا بتعتمد على عدد القطع يلي بتمر قبل ما يصير Failiure مثلاً

ممكن احطها Statistical مثلاً Expo(100)

بقدر اغيره كمان

بقدر استخدمها لـ Maintenance و احط الوقت كل كم لازم يصير

Describe how to collect queue size in categories of 10?

Frequencies

from (Input − Output)

Describe how to measure time in queue and time in system and time prior to rework?

Tally

- **Record time-persistent occurrence frequency of variable, expression, or resource state**
  - Use here to record % of time rework queue is of length 0, (0, 10], (10, 20], … to give info on number of racks needed
- **Statistic data module (Advanced Process panel)**
  - Five Types of statistics, of which Frequencies is one
  - Specify Name (**`Rework Queue Stats`**), Frequency Type (**`Value`**)
  - Specify Expression to track and categorize
    - Right-click in field to get to Expression Builder
  - Report Label (**`Rework Queue Stats`**)
  - Pop-up secondary spreadsheet for Categories (browse file)

Describe the how to get the average, min and max for the queue size?

يجب فهم التسميات و عدم الخربطة بينهم

- **Tally** → anything related to time

  time

  تجميع الأشياء
  as a function of entity

  تجميع معلومات Entity
  by Entity

  مثل :- time in Que
  time in system

  ★ ٥٥٠

- **(time persistance)** → الأشياء التي تختلف حسب الوقت

  anything not related to time

  is a function of time
  and not a function of entity

  و يهمني هنا ال Average size أو ال Maximum size

  و هي تأتي من ال Variables
  و ال Que Size
  و ال utilization
  و عدد القطع الموجودة بال system حالياً
  و ال work in process
  و هكذا

- **counters** → are a special case of (time persistant)

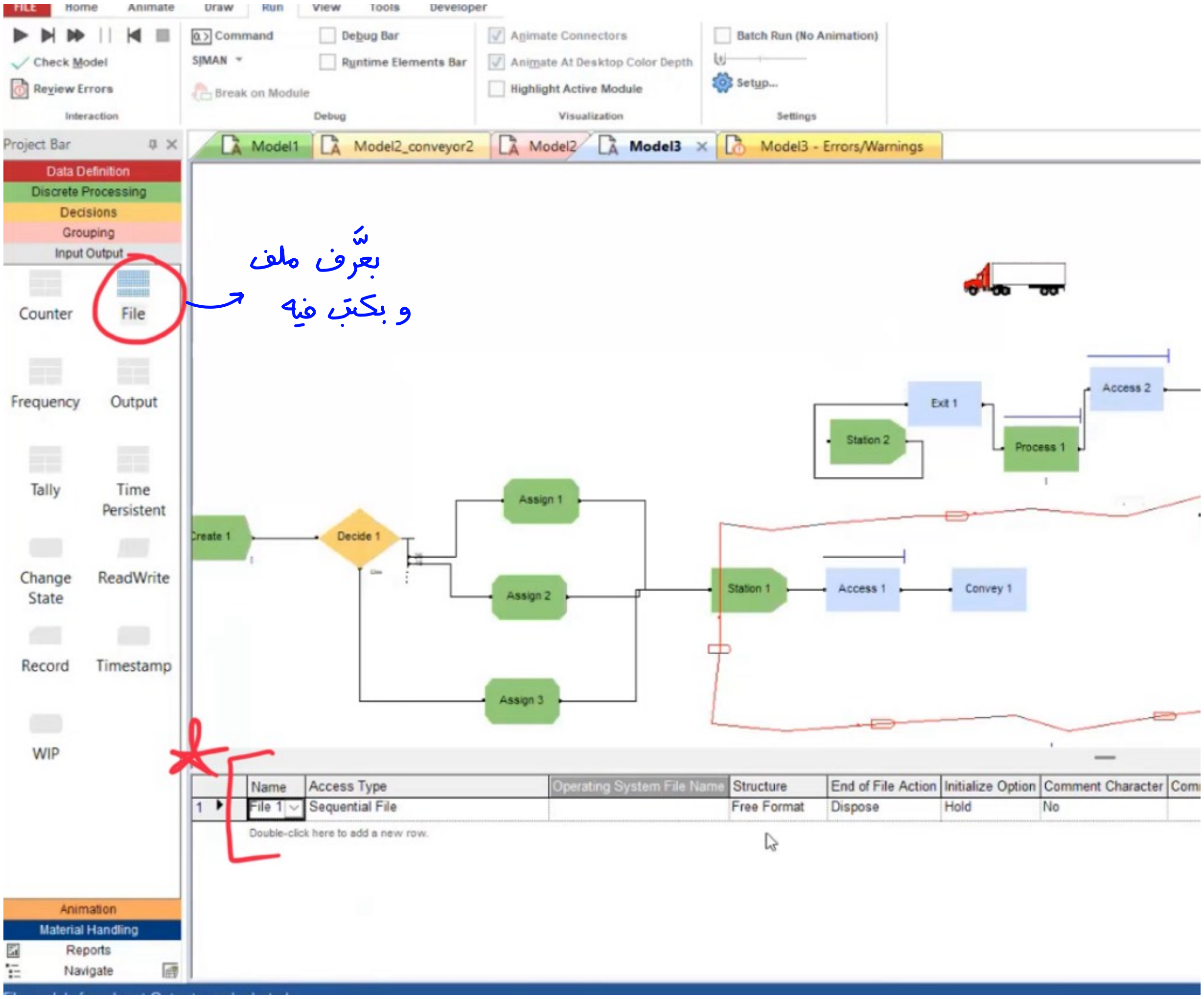
  و يهمني ال Last count
  و هي تهتم بالأعداد

Frequencies → يوجد فيها تبويبات
categories

0 rows → double click

يقدر اعبيهم اما → constant / Range

الفرق بين Frequency و time Persistant	
اذا بحتاج ادخل بموضوع التبويبات و احط تسميات مختلفة	لتجميع المعلومات العادية بدون تبويبات


بعرّف ملف
و بكتب فيه
Counter
File
Frequency
Output
Tally
Time Persistent
Change State
ReadWrite
Record
Timestamp
WIP
Name
Access Type
Operating System File Name
Structure
End of File Action
Initialize Option
Comment Character
File 1
Sequential File
Free Format
Dispose
Hold
No
Double-click here to add a new row.


بقدر اسجل أو اسحب معلومات من هنا
drag it
ReadWrite 1
ReadWrite
Name:
Type:
Arena File Name:
Read from File
File 1
Overriding File Format:
Assignments:
<End of list>
Add...
Edit...
Delete
Comment:
OK
Cancel
Help

Frequencies (cont'd.)

- **Add another Frequency (in Statistic module) to give a finer description of the Sealer states**
  - Will produce statistics on proportion of time Sealer is in each of its *three* possible states – Busy, Idle, and Failed
- **Frequencies are not part of default Category Overview report – open Frequencies report from Project Bar (get a separate window for them)**

Results of Model 4-2

- **Differ from those of Model 4-1 since this is a longer run, modeling assumptions are different**
  - All of which causes underlying random-number stream to be used differently (Chapter 12)
- **Prep A/B didn't change (other than run length and random variation) … need statistical analysis of simulation output (Chapters 6, 7, 12)**
- مكتظ
  Sealer is more congested (it now fails)
- **Rework is less congested (50% higher staffing)**
- يكفي
  Frequencies report suggests one rack suffices about 95% of the time, two racks all the time
  - See text for discussion of Standard, Restricted Percents

الاستخدامات

Utilizations – Some Fine Points

ممكن يكون
واحد أو صفر

- **Two utilizations reported for each Resource**
  - الاستخدام الفوري
    Instantaneous Utilization is the time-average of the ratio of the number of units that are busy to the number of units that are scheduled
    - By definition, counts periods when zero units are scheduled as zero-utilization periods
  - *Scheduled Utilization* is the average number busy divided by the average number available
    - No division-by-zero problem, assuming there were ever any units of the Resource scheduled at all (if not, it shouldn't be in the model)
- **Identical for fixed-capacity Resource**
- **Can differ for Resources on a variable Schedule**
  - If Resource capacity varies among several different positive values, it's better to use Scheduled Utilization
  - See text for discussion of issues and even finer points

Model 4-3: Enhancing the Animation

- **Get "Spartan" generic default animation for some things (queues, connector-animation movement)**
  - Usually sufficient for verification, validation
- **Often want to customize, enhance it a bit**
  - More realism, impact
- **Can pull animation away from model logic in model window**
  - Useful for big models, complex animation
  - Set up Named Views for model logic, animation, or close-ups of parts of animation
- **Animation objects are connected to model logic**
  - Identifiers, physical location (Shift-drag to decouple)

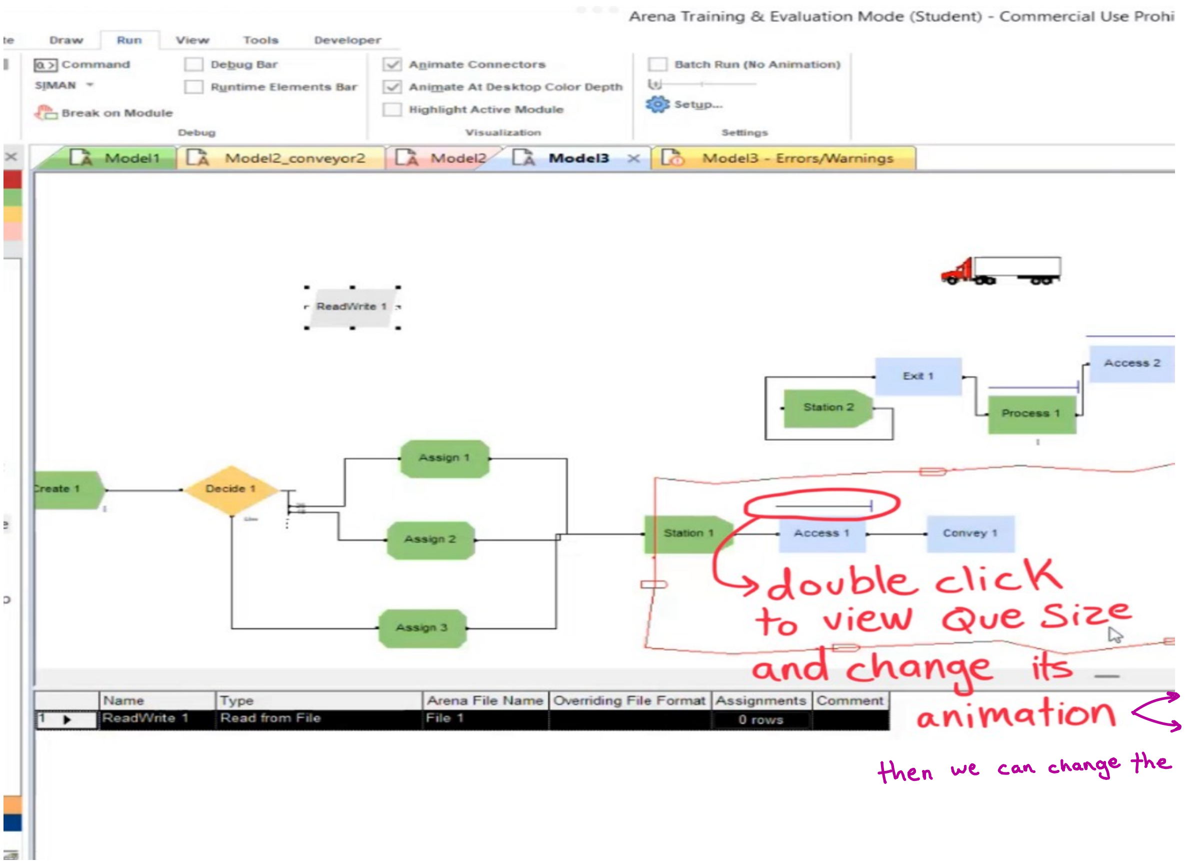
Changing Animation Queues

- **Lengthen (click, drag, maybe hold shift) to "hold" more entities**
  - Simulation logic, results still OK even if animated queue overflows
- **Rotate to re-orient for realism**
- **Change the "form" of the queue from *Line* (the default) to *Point* — fixed places for entities**
  - Double-click on the queue
  - Select Type to be Point
  - Click Points… button
  - Successively click Add for points, then OK
  - Drag them around on screen
  - *Check* Rotate box to show entities turning

Describe how to animate the queue , resource and entity in arena?

Animation for Que Size

بالنسبة لل Que size

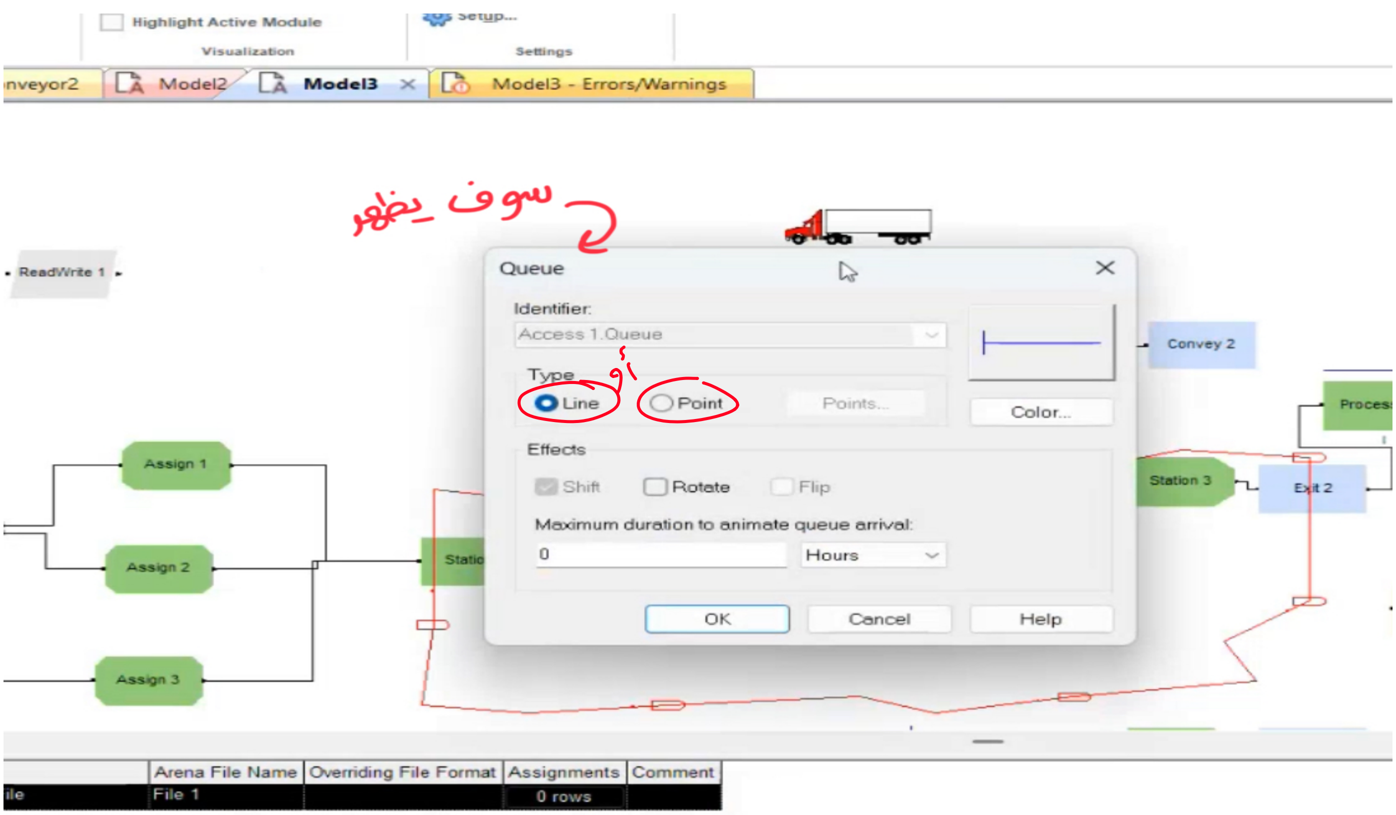


double click to view Que Size and change its animation

Point
Line

then we can change the Rotation

سوف يظهر



Changing the Entity Pictures

- **Earlier – used Entity data module to assign different Initial Pictures to different Entity Types**
- **Now – customize the list, or alter the pictures in it**
  - *Edit > Entity Pictures* Animate → Entity Edit picture
  - Left column – names, pictures currently on the list
  - Right column –picture libraries (.plb filename extension)
  - Add a hand-drawn picture – Add button on left, name it in Value field at top, double-click on blank depressed button, then artwork (or paste in a copied graphics image)
    - New name won't appear in Entity data module until you type it there
  - Edit an existing picture – double-click, artwork
  - Copy a picture over from picture library

Adding Resource Pictures

- **Animate a Resource – Resource button in animate toolbar – get Resource Picture Placement window**
- **Left column – default pictures for different Resource states**
  - Attach logically to a Resource by Identifier pull-down list
  - Double-click to edit the artwork by hand, or paste in previously copied graphics images
  - Seize area – where seizing entity will "reside"
  - Multiple seize areas for multi-capacity Resources
- **Right column – picture libraries (.plb files) – can copy over to selected (depressed) state pictures**
- **Accept window, cross hairs, click to place**
  - Resize, reposition later

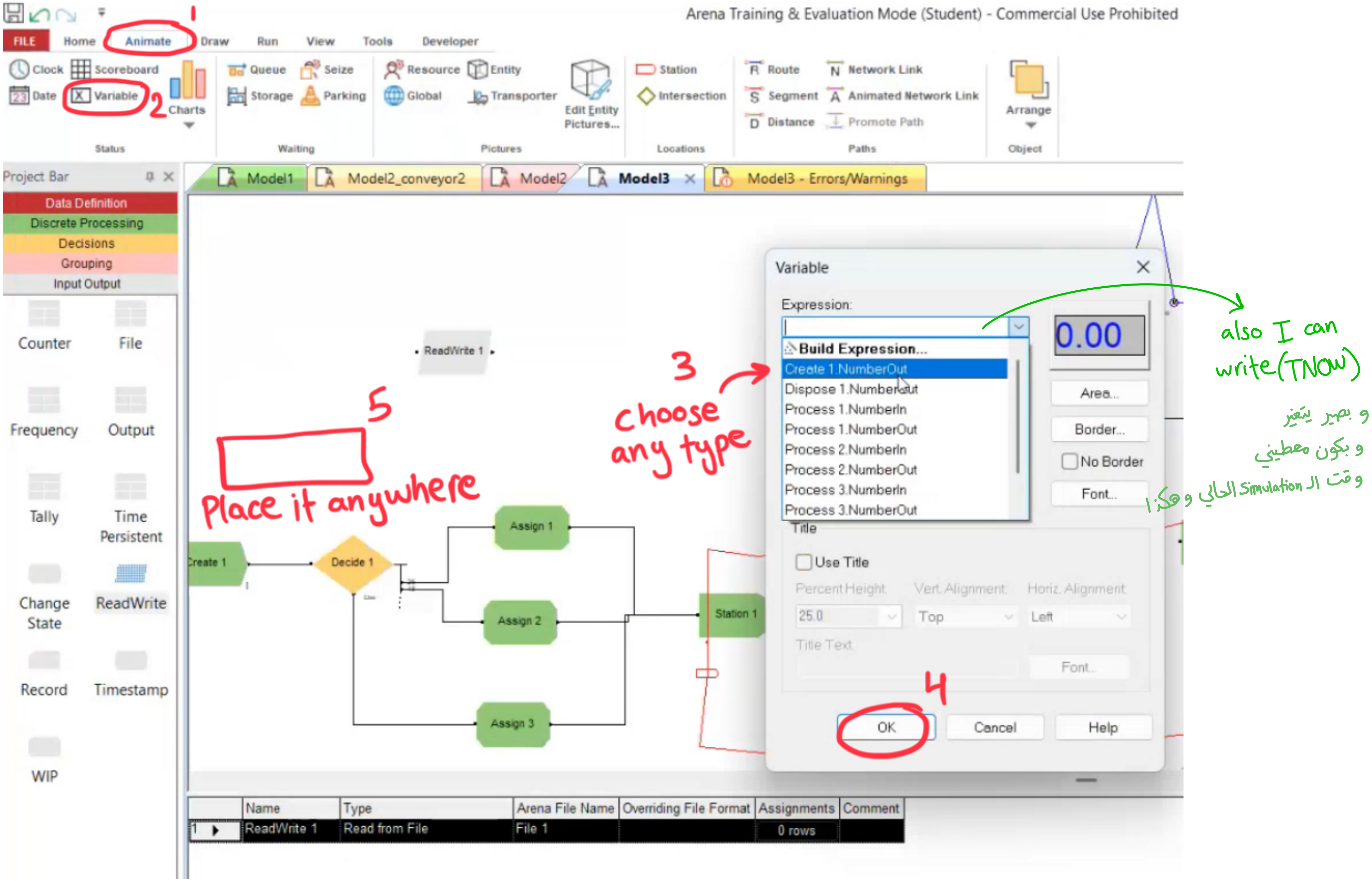
Adding Variables and Plots

- **Variable animation – just show a value of something as a number, watch it change**
  - Variable object 

from Animate toolbar
  - Double-click, specify Expression to be shown (Expression Builder), and cosmetics
  - Resize, reposition later
- **Dynamic animated plots – discussed in Chapter 3**
- **Other animation objects from Animate toolbar**
  - Clock (TNOW), variety of formats
  - Level (thermometer) animation
  - Others discussed later

Describe how to add an animation for TNOW variable in arena?

Adding Variables and Plots



Model 4-4: The Electronic Assembly and Test System with Part Transfers

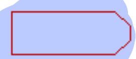
- **Generalize Model 4-3**

ليس فوريا

- **All part transfers now take 2 minutes (not instant) … want to model and animate**
  - Includes:
    - Arriving parts to prep areas
    - Departing parts to appropriate exit
    - All internal part transfers
  - Regardless of distance ... will fix this (unrealistic) assumption in Chapter 8

Describe the station marker?

New Arena Constructs

- ***Station*** **– location where some process occurs**
  - Arrivals, manufacturing cells, departures
  - Each Station given a unique name
  - Can serve as an entry point for a section of model logic
  - *Station marker*  represents a logical station in the flowchart/animation
- ***Station Transfer*** **– entities move between Stations without direct connection**
  - Several different types – we'll use *Routes* here, which allow for positive transfer time, but no other delays like "room" on the transitway or transporters
  - *Route paths* represent Routes in the flowchart/animation

Adding the Route Logic – From Arrival

is a sequence that is related to the stations

- **Stations and Station Transfers affect both the model logic and the animation**
- **Start with Model 4-3 … change to Model 4-4**
- **For incoming parts (A and B) delete connection from Assign modules to "Prep" Process modules**
  - Replace with Station/Route module pairs Station process Route
    - Station module (Advanced Transfer panel) – define entity's current location
      - Module Name vs. Station Name
    - Route module (Advanced Transfer panel) – send entity out
      - Route Time, Destination Station
  - No direct connections exiting from the Route modules – Route module's Destination Station Name defines that

Adding the Remaining Route Logic

- **Add Station modules for entry to each Prep area**
  - Station names are `Prep A Station`, `Prep B Station`, and are the destination stations for Routes after arrivals
- **Process modules for Prep A, Prep B unchanged**
- **After prep, entities connected to Route module to send to next station (sealer)**
  - Don't need a separate Station module for outgoing side
- **Similar changes for rest of model**
  - Station modules for incoming parts into sealer, rework, each of three Record modules (entity exit points)
  - Route modules for outgoing parts out of sealer inspection, rework inspection (two for each Decide module – pass/fail)
- **Could run model now, get correct results … but no animation of transfers …**

Why Not Just Add Delays?

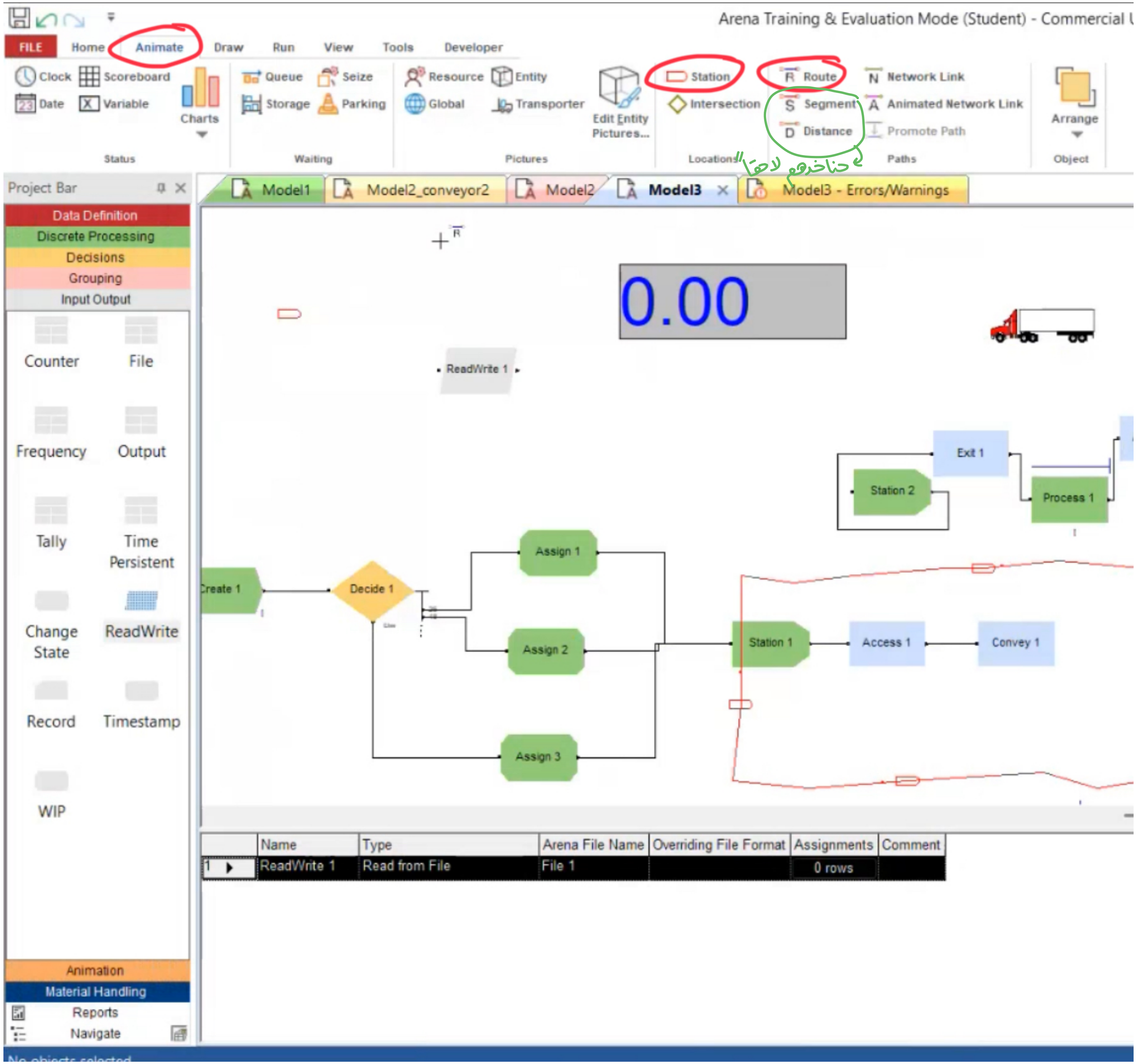
- **Simpler approach than the above to getting the two-minute transfer times:**
  - Insert a Process module with Action = Delay for 2 minutes on each relevant connection
  - Alternatively, use Delay module from Advanced Process panel
- **Actually this *would* work fine from modeling, numerical-output viewpoints**
- **But it would not allow animation of part transfers, so we'll proceed with the Stations and Routes**

Altering the Animation – Station Markers, Routes

- **Add animation for Stations and Routes**
- **Station button , Animate Transfer toolbar**
  - Attach Identifier to it from pull-down list of station names
  - Get cross hairs, place (click) marker in animation
  - Can place several station markers for the same logical station (to represent incoming, outgoing sides)
  - Can drag station markers around later
- **Route button from Animate Transfer toolbar**
  - Options for appearance of entities as they travel the route
  - Get cross hairs; click in origin, destination Station Markers
    - Intermediate clicks for corners along the route
  - Can drag around endpoints, corners later

Describe how to add route animation?

Altering the Animation – Station Markers, Routes



Altering the Animation – Entity Pictures

- **Part B arrivals are in batches of four parts/batch**
  - But constant travel time to Prep B implies they travel "on top of each other" so it looks like just one part B
  - Try – change Route time from `2` to `EXPO(2)`, see separation along the route
- **Create illusion to animate the batch**
  - Assign module just after **`Part B Arrive`**
  - Add assignment of Entity Picture to **`Picture.Batch B`**
  - *Edit > Entity Pictures* to draw the new picture
    - Copy **`Picture.Part B`** and rename it **`Picture.Batch B`**
    - Double-click on picture, use Picture Editor to get four circles
  - When batch arrives to Prep B, change to single circle
    - Add Assign module after **`Prep B Arrival Station`**

Input Analysis: Specifying Model Parameters, Distributions

- ***Structural*** **modeling: what we've done so far**
  - Logical aspects — entities, resources, paths, etc.
- ***Quantitative*** **modeling**
  - Numerical, distributional specifications
  - Like structural modeling, need to observe system's operation, take data if possible

Deterministic vs. Random Inputs

- ***Deterministic*: nonrandom, fixed values**
  - Number of units of a resource
  - Entity transfer time (?)
  - Interarrival, processing times (?)

Describe how to make the intearrival time deterministic or random? Same for process time?

- ***Random* (a.k.a. *stochastic*): model as a distribution, "draw" or "generate" values from to drive simulation**

Random number generation is (systematic)

we use (Replicates)

  - Transfer, Interarrival, Processing times
  - What distribution? What distributional parameters?
  - Causes simulation output to be random, too
- **Don't just assume randomness away — validity**

Collecting Data

- **Generally hard, expensive, frustrating, boring**
  - System might not exist
  - Data available on the wrong things — might have to change model according to what's available
  - Incomplete, "dirty" data
  - Too much data (!)
- **Sensitivity of outputs to uncertainty in inputs**
- **Match model detail to quality of data**
- **Cost — should be budgeted in project**
- **Capture variability in data — model validity**
- **Garbage In, Garbage Out (GIGO)**

Using Data: Alternatives and Issues

- **Use data "directly" in simulation**
  - Read actual observed values to drive the model inputs (interarrivals, service times, part types, …)
    - Arena ReadWrite module ... see Model 10-2
  - All values will be "legal" and realistic
  - But can never go outside your observed data
  - May not have enough data for long or many runs
  - Computationally slow (reading disk files)
- **Or, fit probability distribution to data**
  - "Draw" or "generate" synthetic observations from this distribution to drive the model inputs
  - We've done it this way so far
  - Can go beyond observed data (good and bad)
  - May not get a good "fit" to data — validity?

Fitting Distributions to Data with the Arena Input Analyzer

- **Assume:**
  - Have sample data: Independent and Identically Distributed (IID) list of observed values from the actual physical system
  - Want to select or fit a probability distribution for use in generating inputs for the simulation model

Describe how to fit a distribution to an interarrival data?

- **Arena Input Analyzer**
  - Separate application, also accessible via Tools menu in Arena
  - Fits distributions, gives valid Arena expression for generation to paste directly into simulation model

Fitting Distributions to Data with the Arena Input Analyzer (cont'd.)

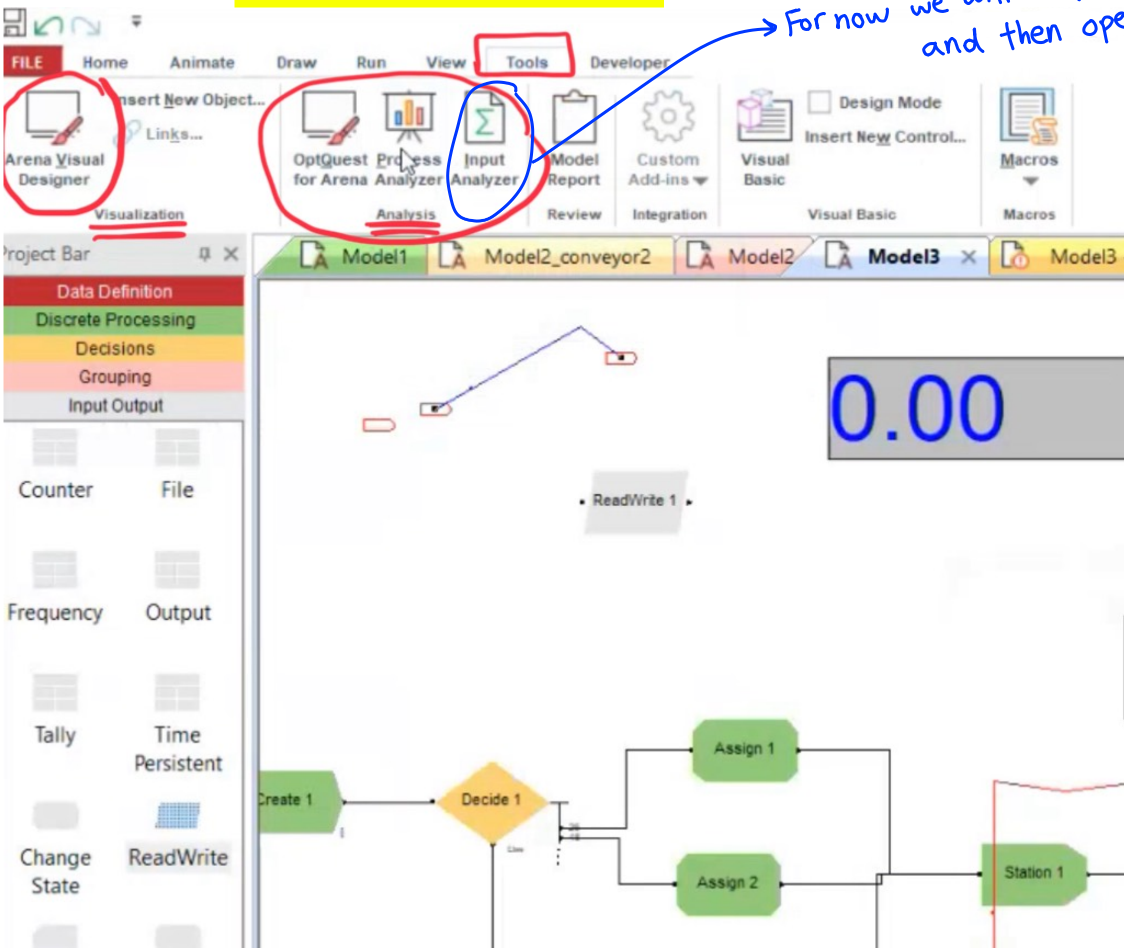
- **Fitting = deciding on distribution form (exponential, gamma, empirical, etc.) and estimating its parameters**
  - Several different methods (Maximum likelihood, moment matching, least squares, …)
  - Assess goodness of fit via hypothesis tests
    - H_0: fitted distribution adequately represents the data
    - Get *p* value for test (small = poor fit) What is the empirical distribution?
- **Fitted "theoretical" vs. empirical distribution**
- **Continuous vs. discrete data, distribution**
- **"Best" fit from among several distributions**

Data Files for the Input Analyzer

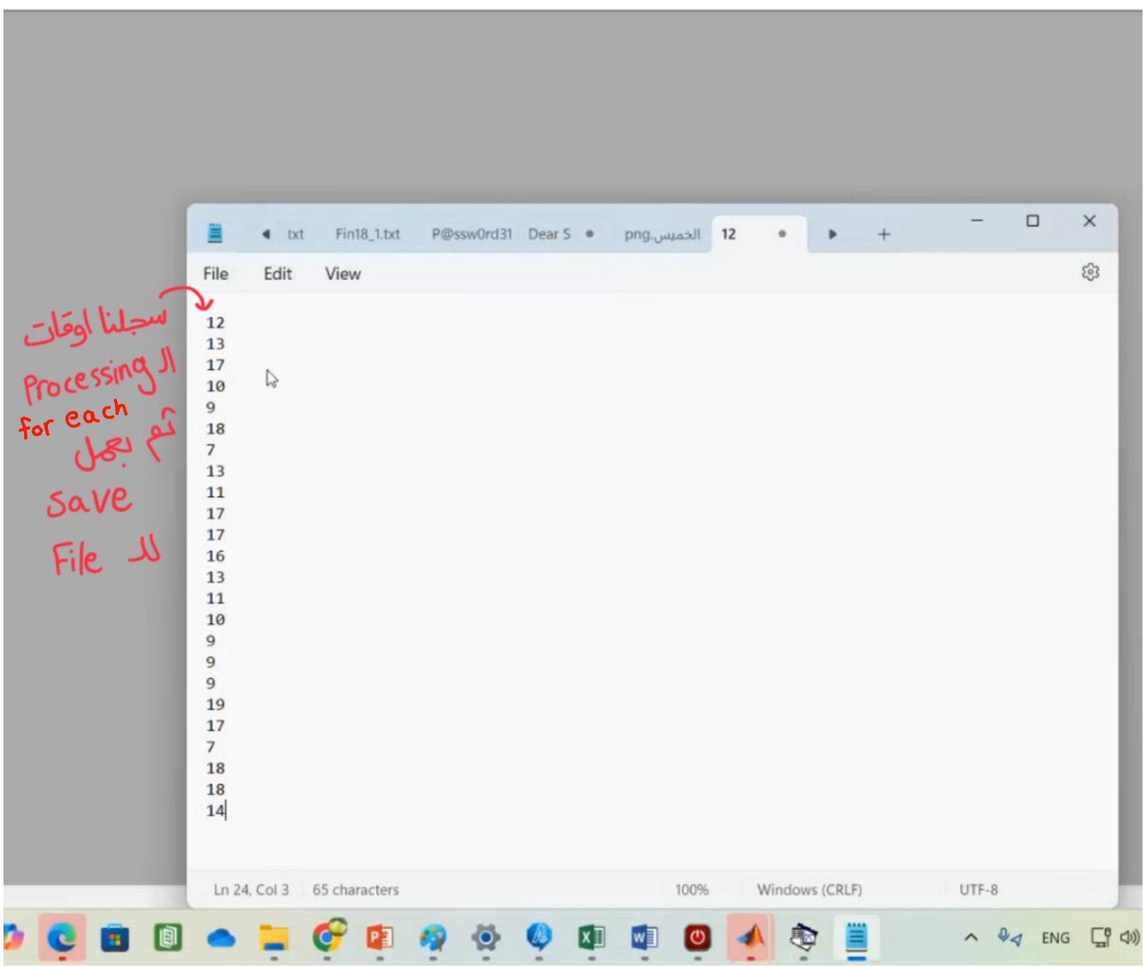
- **Create the data file (editor, word processor, spreadsheet, ...)**
  - Must be plain ASCII text (save as text or export)
  - Data values separated by white space (blanks, tabs, linefeeds)
  - Otherwise free format
- **Open data file from within Input Analyzer**
  - *File > New* or 
  - *File > Data File > Use Existing* or 
  - Get histogram, basic summary of data
  - To see data file: *Window > Input Data*
- **Can generate "fake" data file to play around**
  - *File > Data File > Generate New*

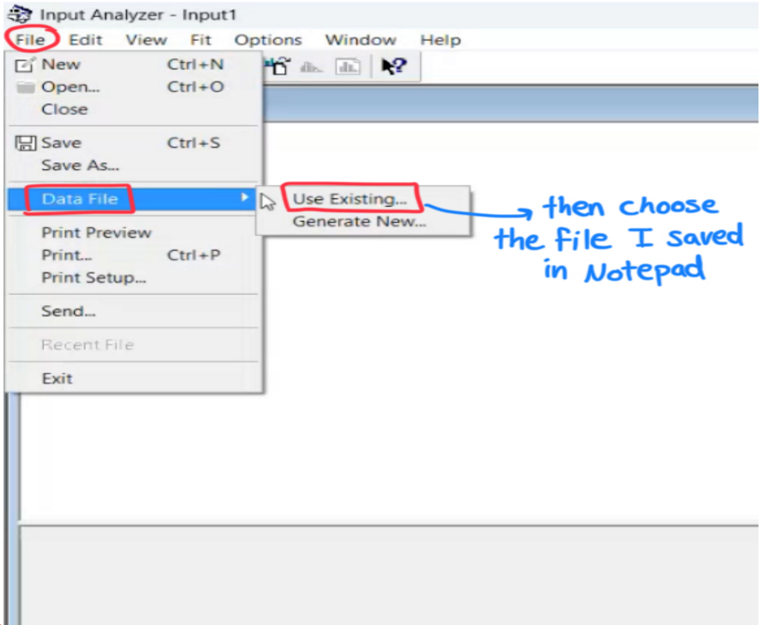
Fitting Distributions to Data
with the Arena Input Analyzer

For now we will use (Input Analyzer)
and then open [Notepad]

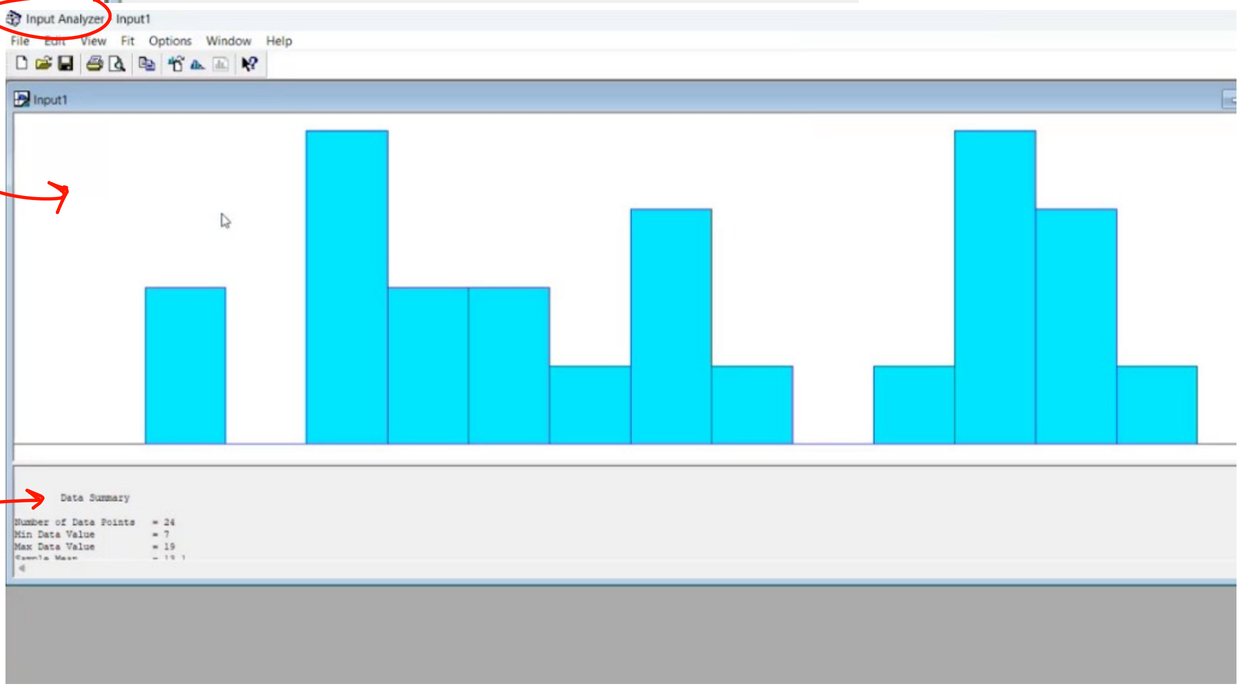


سجلنا اوقات
ال Processing
for each
ثم بعمل
save
لل File





then choose the file I saved in Notepad

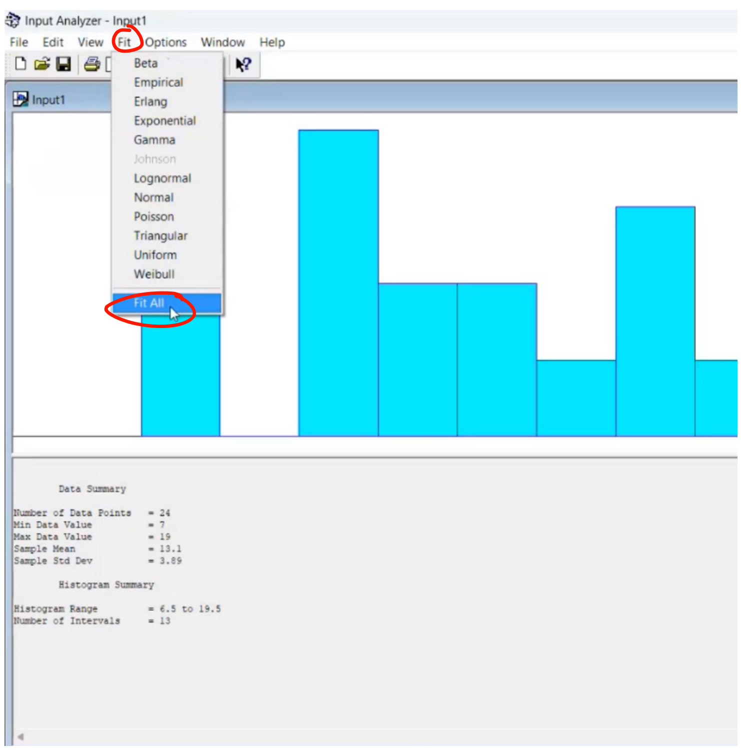


يعطيني هاي النتيجة

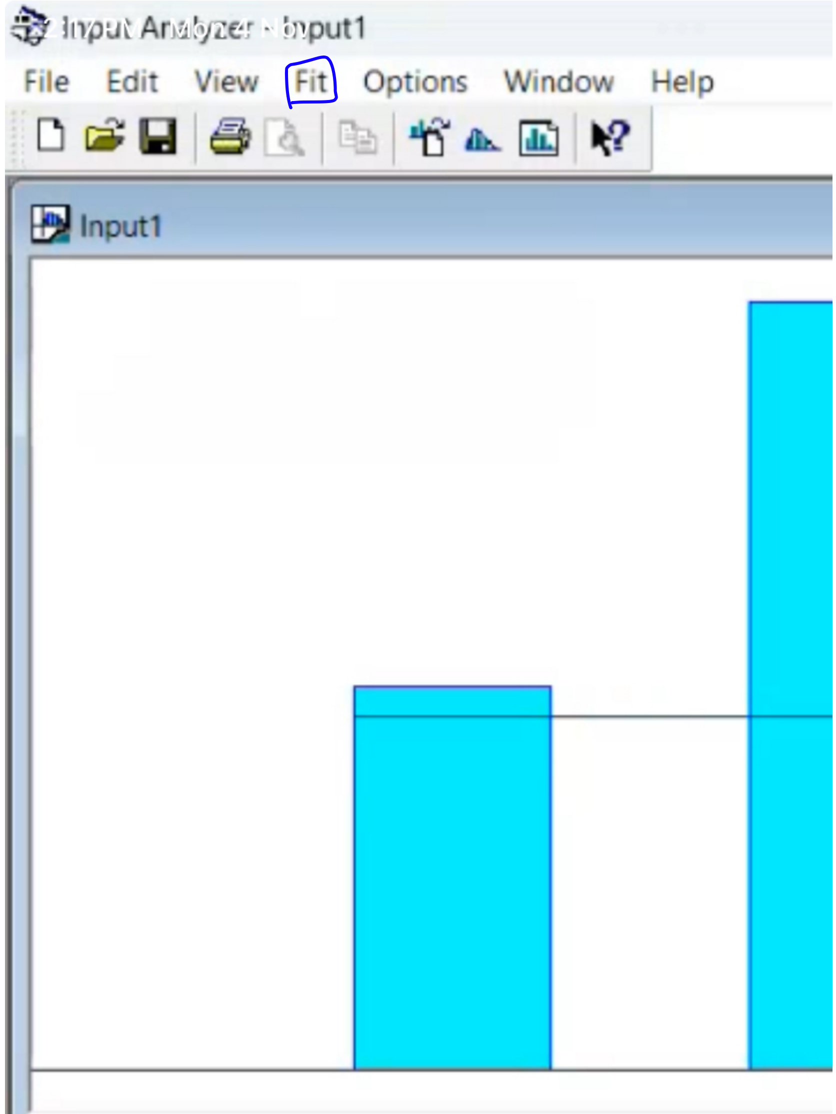
يوجد معلومات احصائية

then I choose

Fit → fit All



After I made (Fit All)



Distribution Summary

Distribution: Uniform → the best distribution for you
Expression: UNIF(6.5, 19.5)
Square Error: 0.037660

وبقدر من فوق اغيره
بال (Fit)

Chi Square Test
Number of intervals = 4
Degrees of freedom = 3
Test Statistic = 1.97
Corresponding p-value = 0.587

Data Summary

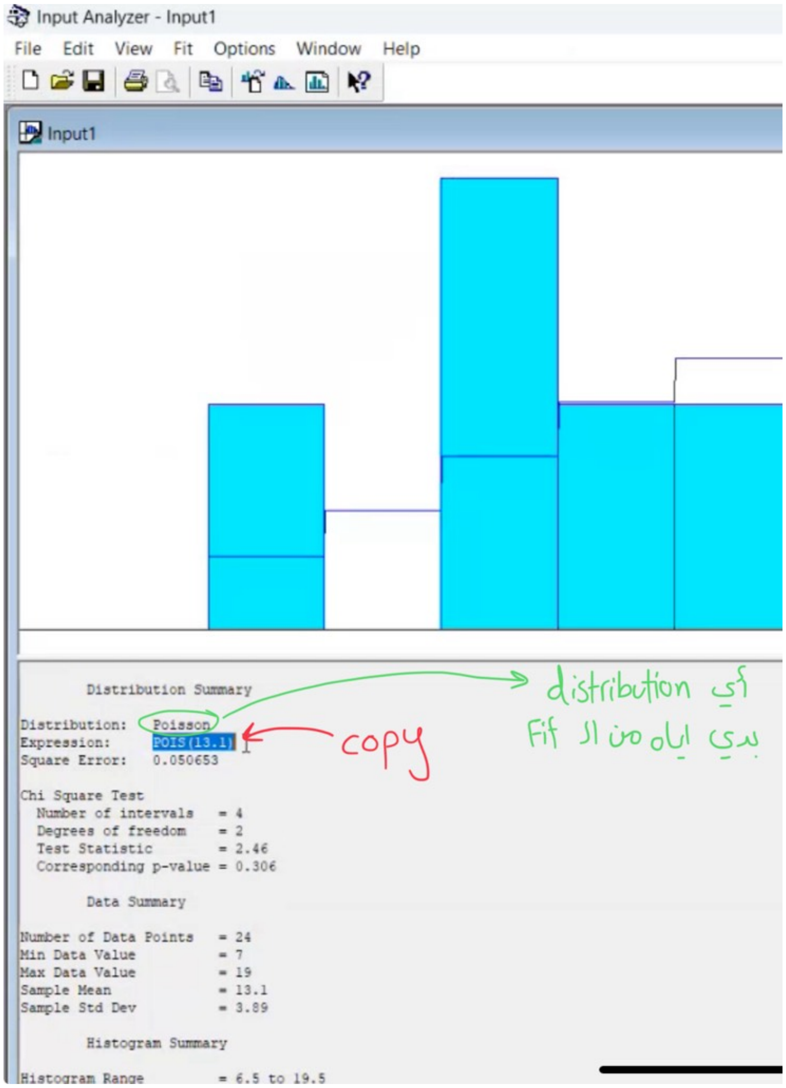
Number of Data Points = 24
Min Data Value = 7
Max Data Value = 19
Sample Mean = 13.1
Sample Std Dev = 3.89

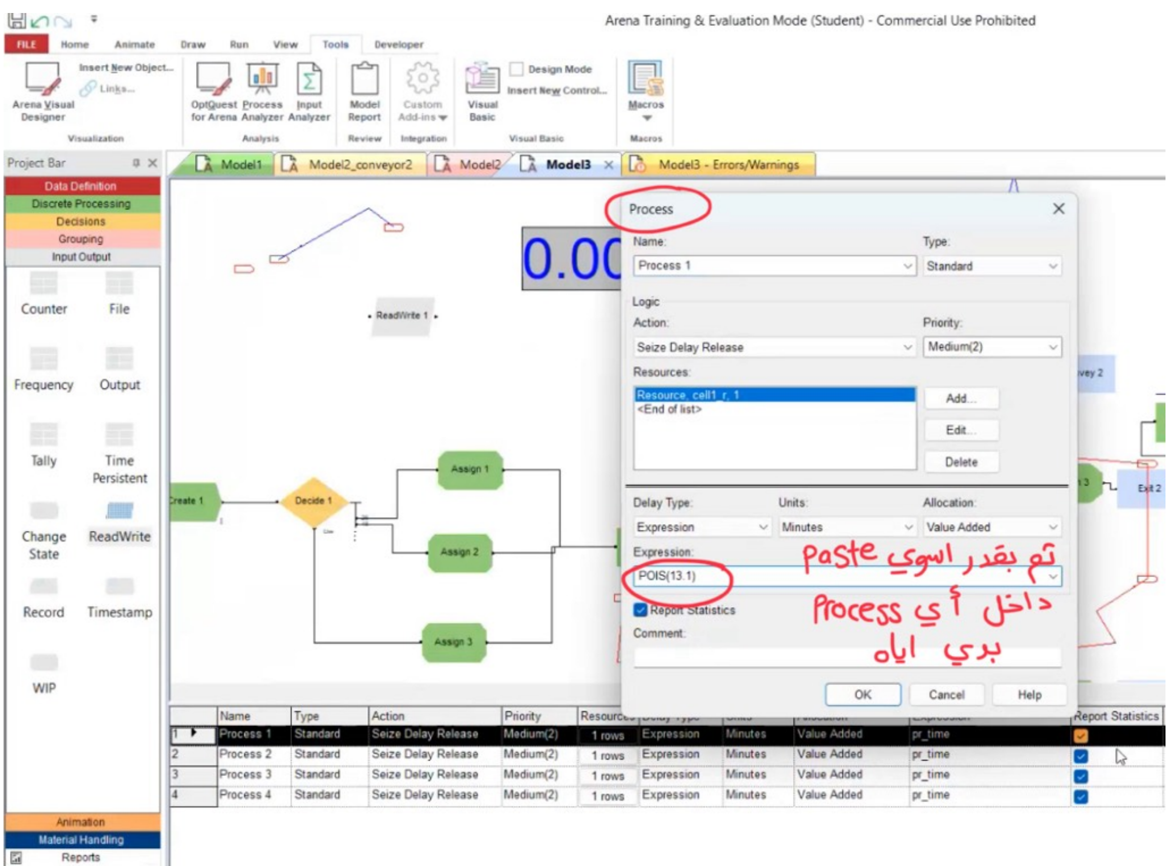
Histogram Summary

Histogram Range = 6.5 to 19.5
Number of Intervals = 13

Gamma Distribution → سيء

Empirical Distribution → بسوي Fit لل data
و ال Square Error بكون صفر


Input Analyzer - Input1
File Edit View Fit Options Window Help
Input1
Distribution Summary
Distribution: Poisson
Expression: POIS(13.1)
Square Error: 0.050653
Chi Square Test
Number of intervals = 4
Degrees of freedom = 2
Test Statistic = 2.46
Corresponding p-value = 0.306
Data Summary
Number of Data Points = 24
Min Data Value = 7
Max Data Value = 19
Sample Mean = 13.1
Sample Std Dev = 3.89
Histogram Summary
Histogram Range = 6.5 to 19.5
copy
أي distribution
بدي اياه من ال Fit


Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
Process
Name: Process 1
Type: Standard
Logic
Action: Seize Delay Release
Priority: Medium(2)
Resources:
Resource, cell1_r, 1
<End of list>
Add...
Edit...
Delete
Delay Type: Expression
Units: Minutes
Allocation: Value Added
Expression:
POIS(13.1)
Report Statistics
Comment:
OK
Cancel
Help
ثم بقدر اسوي paste
داخل أي Process
بدي اياه

The Fit Menu

- **Fits distributions, does goodness-of-fit tests**
- **Fit a specific distribution form**
  - Plots density over histogram for visual "test"
  - Gives exact expression to Copy and Paste (Ctrl+C, Ctrl+V) over into simulation model
  - May include "offset" depending on distribution
  - Gives results of goodness-of-fit tests
    - Chi square, Kolmogorov-Smirnov tests
    - Most important part: *p-value*, always between 0 and 1:

      Probability of getting a data set that's more inconsistent with the fitted distribution than the data set you actually have, if the the fitted distribution is truly "the truth"

      "Small" *p* (< 0.05 or so): poor fit (try again or give up)

The Fit Menu (cont'd.)

- **Fit all of Arena's (theoretical) distributions at once**
  - *Fit > Fit All* or 
  - Returns the *minimum square-error* distribution
    - Square error = sum of squared discrepancies between histogram frequencies and fitted-distribution frequencies
    - Can depend on histogram intervals chosen: different intervals can lead to different "best" distribution
  - Could still be a poor fit, though (check *p* value)
  - To see all distributions, ranked: *Window > Fit All Summary* or 

The Fit Menu (cont'd.)

- **"Fit" Empirical distribution (continuous or discrete): *Fit > Empirical***
  - Can interpret results as a Discrete or Continuous distribution
    - Discrete: get pairs (*Cumulative* Probability, Value)
    - Continuous: Arena will linearly interpolate *within* the data range according to these pairs (so you can never generate values outside the range, which might be good or bad)
  - Empirical distribution can be used when "theoretical" distributions fit poorly, or intentionally

Some Issues in Fitting Input Distributions

- **Not an exact science — no “right” answer**
- **Consider theoretical vs. empirical**
- **Consider range of distribution**
  - Infinite both ways (e.g., normal)
  - Positive (e.g., exponential, gamma)
  - Bounded (e.g., beta, uniform)
- **Consider ease of parameter manipulation to affect means, variances**
- **Simulation model sensitivity analysis**
- **Outliers, multimodal data**
  - Maybe split data set (see textbook for details)

No Data?

كل ما يقل تجميع ال statistical data
كلما قل مستوى الدقة لل simulation

- **Happens more often than you'd like**
- **No good solution; some (bad) options:**
  - Interview "experts"
    - Min, Max: Uniform
    - Avg., % error or absolute error: Uniform
    - Min, Mode, Max: Triangular

      Mode can be different from Mean — allows asymmetry

اذا ما عندي أي Data بقدر ألجأ لل Triangular distribution

  - Interarrivals — independent, stationary
    - Exponential— still need some value for mean
  - Number of "random" events in an interval: Poisson
  - Sum of independent "pieces": normal
  - Product of independent "pieces": lognormal

Cautions on Using Normal Distributions

- **Probably most familiar distribution – normal "bell curve" used widely in statistical inference**
- **But it has infinite tails in both directions … in particular, has an infinite left tail so can always (theoretically) generate negative values**
  - Many simulation input quantities (e.g., time durations) must be positive to make sense – Arena truncates negatives to 0
- **If mean μ is big relative to standard deviation σ, then P(negative) value is small … one in a million**
- **But in simulation, *one in a million can happen***
- **Moral – avoid normal distribution as input model**

بالعادة بضيف بعد ال Normal distribution "مباشرة Assign statment

انه اذا الرقم طلع اكبر من حد معين ← يحط حد اقصى لل distribution

و اذا طلع اقل من حد معين ← يحط الحد الأدنى لل distribution

Nonstationary Arrival Processes

- **External events (often arrivals) whose rate varies over time**
  - Lunchtime at fast-food restaurants
  - Rush-hour traffic in cities
  - Telephone call centers
  - Seasonal demands for a manufactured product

Describe how to model non-stationary interarrival times? Example peak rush hour?

- **It can be critical to model this nonstationarity for model validity**
  - Ignoring peaks, valleys can mask important behavior
  - Can miss rush hours, etc.
- **Good model:** ***Nonstationary Poisson process***

Nonstationary Arrival Processes (cont'd.)

- **Two issues:**
  - How to specify/estimate the *rate function*
  - How to generate from it properly during the simulation
- **Several ways to estimate rate function — we'll just do the *piecewise-constant* method**
  - Divide time frame of simulation into subintervals of time over which you think rate is fairly flat
  - Compute observed rate within each subinterval
  - In Arena, must convert to expected number of arrivals *per hour* on subintervals of time that need not be of one-hour length
    - Want expected 45 arrivals in a half hour; specify rate = 90 *per hour*
- **Example: Model 5-2 in Chapter 5**

ال Schedule لل Arrival يلي سميناها Piecewise constant

هنا بنفترض مثلاً انو ال Arrival time من ال 8 إلى 8 وربع ← have the same values

و بين ال 8 وربع إلى 8 ونص ← the same values

⋮

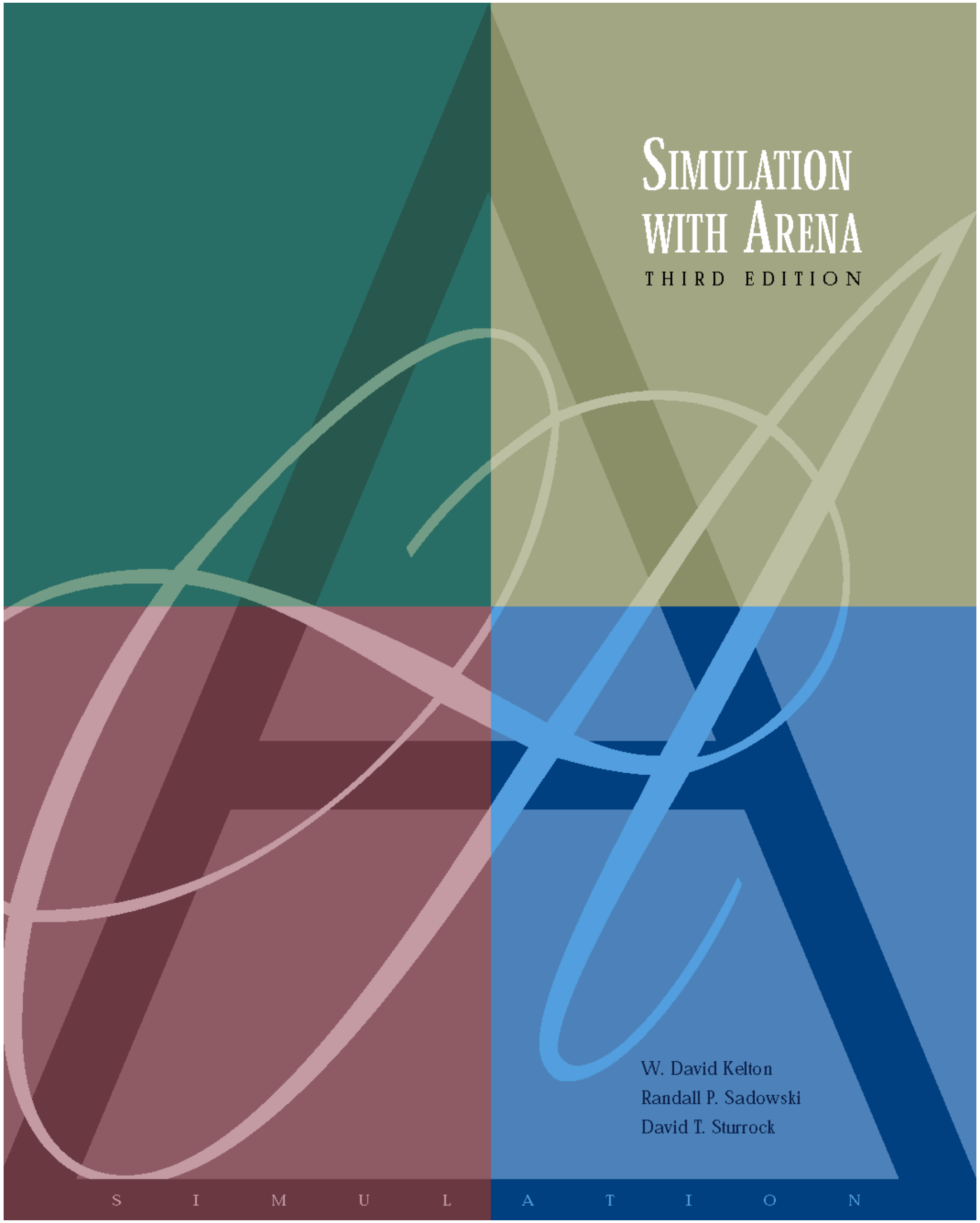
وهكذا

Multivariate and Correlated Input Data

ممكن تكون ال Data ← correlated و هنا لازم نحاول نتخلص من ال correlation

- **Usually we assume that all generated random observations across a simulation are independent (though from possibly different distributions)**
- **Sometimes this isn't true:**
  - A "difficult" part requires long processing in both the Prep and Sealer operations
  - This is positive correlation
- **Ignoring such relations can invalidate model**
- **See textbook for ideas, references**

الأفضل تكون ال Data Totally Random or Representative of the model



Modeling Detailed Operations

Chapter 5

Last revision August 25, 2003

What We'll Do ...

- **Explore lower-level modeling constructs**
- **Model 5-1: Automotive maintenance/repair shop**
  - Multi-way decisions, Sets, Variables, Expressions, Submodels, Duplicating entities, Holding entities
  - Terminating vs. steady-state simulations
- **Debugging**
- **Model 5-2: Enhanced automotive shop model**
  - Sets and Resource logic, nonstationary arrival process
- **Model 5-3: An (*s*, *S*) inventory simulation**
  - Built exclusively via SIMAN Blocks, Elements

Model 5-1: An Automotive Maintenance and Repair Shop

Arena in the Lab ← راح نحله على ال

interarrival time ال create بتحتاج ال

محل ميكانيك

- **New three-bay suburban satellite facility**
- **Appointments taken up to three days in advance**
- **No appointments taken for current day**

create يعني

Entity per Arrival

- **Number of calls per day ~ Poisson with mean 25 calls/day**

هنا ٣ اماكن تصليح وكل واحد بيشتغل ٨ ساعات

Decide by chance

  - 55% of calls schedule appointment for the next day
  - 30% for two days out
  - 15% for three days out

time between Arrivals of calls ✱ المفروض يعطيني السؤال

we either go for entities per arrival or Duplicate an entity 5 times "مثلاً"

- **If no appointment is available on the desired day, there's a 90% chance that the caller will want to schedule for the following day**
  - Other 10% just go away

*this is a very complex problem, Low probability to come in exam

I have 25 calls per day (No specific time) So I will Assume they come in the morning

An Automotive Repair and Maintenance Shop (cont'd.)

- **80% leave vehicle for service, come back later**
  - Other 20% wait at the shop
- **Wait customers given approximate wait time**
  - Estimated service time (Book Time) + Allowance (1 hour)
- **Limit of five wait customers per day** نستقبل 5 customers في اليوم الواحد
- **Book Time ~ 44 + 90*BETA(2, 3) minutes**

  min
- **Actual service time ~ GAMM(Book Time/1.05, 1.05) minutes**

  gamma
- **Maximum of 24 hours (Book Time) scheduled per day (three bays, eight hours each)** → ٣ اماكن تصليح وكل واحد بداوم ٨ ساعات

اذا بلش بالسيارة بستمر بالشغل عليها لحد ما يخلصها

Costs, Revenues

- **Capital and labor cost = $45/hour/bay**
  - 40-hour work week
- **Revenue from customer = $78/hour (charged for Book Time, not actual time)**
- **Overtime costs $120/hour**
  - Maximum of three hours/day/bay
- **If service can't be completed that day, vehicle is held over night and customer gets free loaner car**
  - Loaner car costs the shop $35/day
- **If service isn't started that day, customer takes car home and returns next day – no cost to shop**

System Performance

الأشياء يلي لازم نحسبهم

- **Daily profit**
- **Daily Book Time**
- **Daily actual service time**
- **Daily overtime**
- **Daily number of wait appointments not completed on time**

New Modeling Issues

- **Service System – not manufacturing**
  - But can use same modeling capabilities
- **Multiple-way decisions** (Decide) here based on if-Statement
  - Appointment entities go to one of five wait queues based on day of appointment
  - Capability available in Decide module

is a variable
Array → هي مجموعة من الارقام
it always needs an index
is an Attribute

- **Sets** is a very important concept
  sets $\xrightarrow{\text{is an}}$ Array
  - Groups of similar objects
  - Reference by a common set name and index into the set (subscript)
    - Can also be referenced via their original name
    - Arena sets are *ordered*
  - Service bays will be modeled as a set of resources
    - Also use sets for entity pictures and appointment queues
  - Define via Sets data module

* Array of objects ⟹ sets
(expressions)

النوع الثاني
باﻟ Arrays

⟸ انو يكون رقم السطر بيعتمد على نوع الـ Entity

يتغير حسب الـ Entity اذاً هو Attribute

هنا مثلاً

Entity type one → index one
سطر

Rework Entities → index 4
سطر

وهكذا

⋮

So the index that comes with the Array must be manipulated by Entity → A or B or C
⋮
وهكذا

Array (Attribute)

Here
index → Attribute

يجب أن يكون PreAssigned
قبل الوصول إلى مكان الاستعمال

ممكن مثلاً يكون
Set of Resources

→ Ahmad
→ Omar
⋮

are Resources

New Modeling Issues (cont'd.)

- **Variables and Expressions**
  - Can be referenced in model by name
  - Can be one- or two-dimensional arrays, indexed by one or two integers
  - User-defined Variables

    ثابتين و ما الهم علاقة بالـEntities

    ممكن استخدمه كوسيلة لتخزين المعلومات أو نقل المعلومات (transfer data)
    - Store some numerical value (not a formula)
    - Can be initialized in Variable data module
    - Can be used, reassigned during the simulation run by any entity
  - User-defined Expressions
    - A name defined by a mathematical expression
    - This name can be referenced anywhere in the model
    - Can use constants, Variables, Attributes, system state variables, values from distribution – connected via mathematical operations

      Can use Expression Builder to help define
    - Defined in Expression data module (Advanced Process panel)

Animation Requirements

- **No entity movement**
- **Can still display queues**
- **Appointments**
  - Number
  - Type
  - Book Time hours per day
- **Bay resources**

System or Simulation Type

- ***Terminating***

  ال system الذي بدايته
  و نهايته واحدة
  (بنفس ال Condition)

  - Known starting and stopping conditions – part of model
  - Time frame is known (and finite)

  ال Simulation بيلش من الصفر
  مثلاً: بعتبر انه المستشفى ما فيو ولا مريض و بيدأ ال Simulation

- ***Steady-State***

  لما يكون ال system
  ما بيفضى

  - Initial conditions are not always well defined
  - No defined stopping condition (theoretically infinite)
  - Interested in system response over the long run
- **Automotive repair model**

  (Warm up period)
  الفترة الزمنية يلي ما بجمع فيها data
  Arena → setting → Replicate

  - Starting and ending times known each day
    - Some vehicles can be held over from one day to the next
    - But we'll simulate for 20 days (approximately a working month)
  - Treat the system as terminating

Arena Modeling Panels

- **Basic Process panel**
  - Highest level of modeling
- **Advanced Process panel**
  - More detailed (and different) modeling capabilities
- **Advanced Transfer panel**
  - Material-handling, entity-movement capabilities
- **Blocks, Elements panels**
  - Lowest level of modeling capabilities – the underlying SIMAN simulation language itself
  - Other panels are created using modules from these panels
  - Occasionally needed, but not very often

Building the Model

- **Defining the Data**
- **Submodel Creation**
- **Divide model in sections or submodels**
  - Generate Appointment Calls
  - Make Appointment
  - Service Activity
  - Update Performance Variables
  - Control Logic
- **We'll discuss each of these in turn**

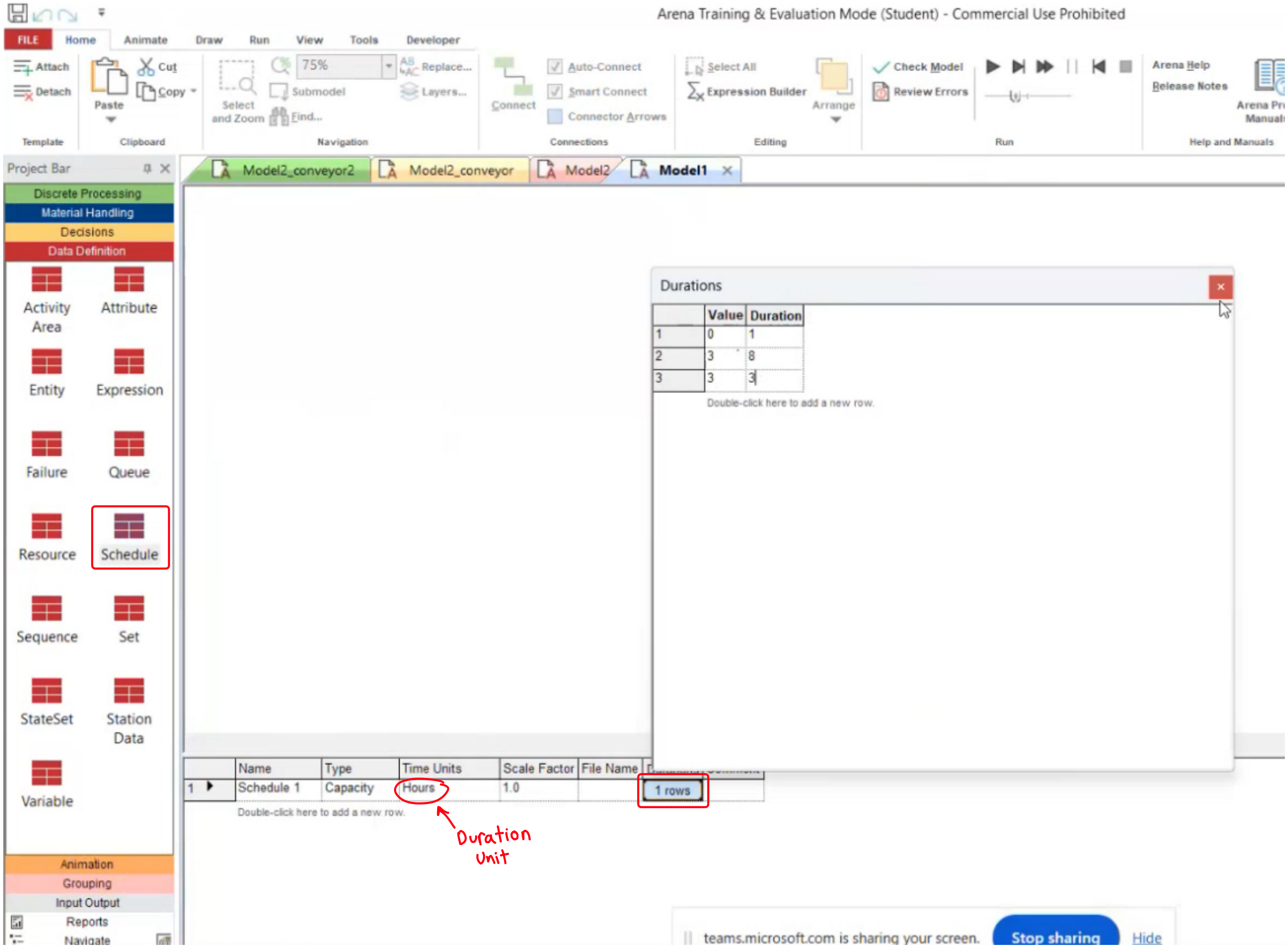
بدنا نعمل Schedule لل Resources

Schedule Data

→ 12 hour schedules

- **Schedules**
  - Enter into Schedule data module
  - 3 schedules required, one for each of the service bays

    But they're identical so we need define only one, and use it for all three bays
  - 12-hour schedule
    - One-hour down time at the beginning (take appointments, etc.)
    - Eight-hour working day
    - Three-hour overtime
  - Use Graphical schedule editor → بطل موجود على ال Arena
  - Use Edit via Dialog (or Edit via Spreadsheet) if you need trailing zeros in the capacity to fill out the cycling time window, or if you need variables in the schedule

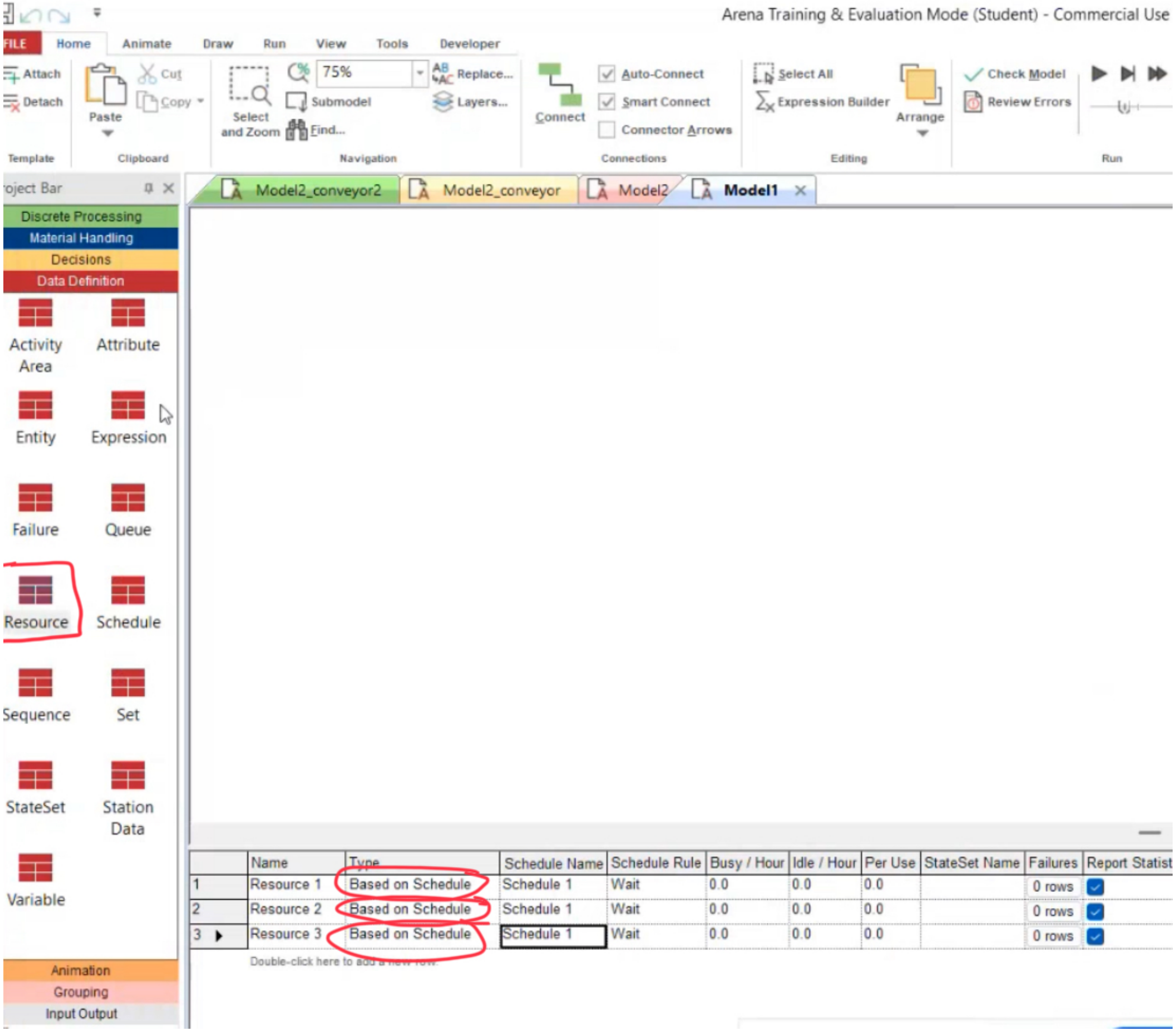

Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Attach
Detach
Template
Paste
Cut
Copy
Clipboard
Select and Zoom
75%
Submodel
Find...
Replace...
Layers...
Navigation
Connect
Auto-Connect
Smart Connect
Connector Arrows
Connections
Select All
Expression Builder
Arrange
Editing
Check Model
Review Errors
Run
Arena Help
Release Notes
Help and Manuals
Project Bar
Discrete Processing
Material Handling
Decisions
Data Definition
Activity Area
Attribute
Entity
Expression
Failure
Queue
Resource
Schedule
Sequence
Set
StateSet
Station Data
Variable
Animation
Grouping
Input Output
Reports
Navigate
Model2_conveyor2
Model2_conveyor
Model2
Model1
Durations
Value
Duration
1 0 1
2 3 8
3 3 3
Double-click here to add a new row.
Name
Type
Time Units
Scale Factor
File Name
1 Schedule 1 Capacity Hours 1.0 1 rows
Double-click here to add a new row.
Duration Unit
teams.microsoft.com is sharing your screen.
Stop sharing
Hide

Resource Data

- **Define resources**

بنعرف ٣ Resorces و كلهم بسويهم Based on schedule

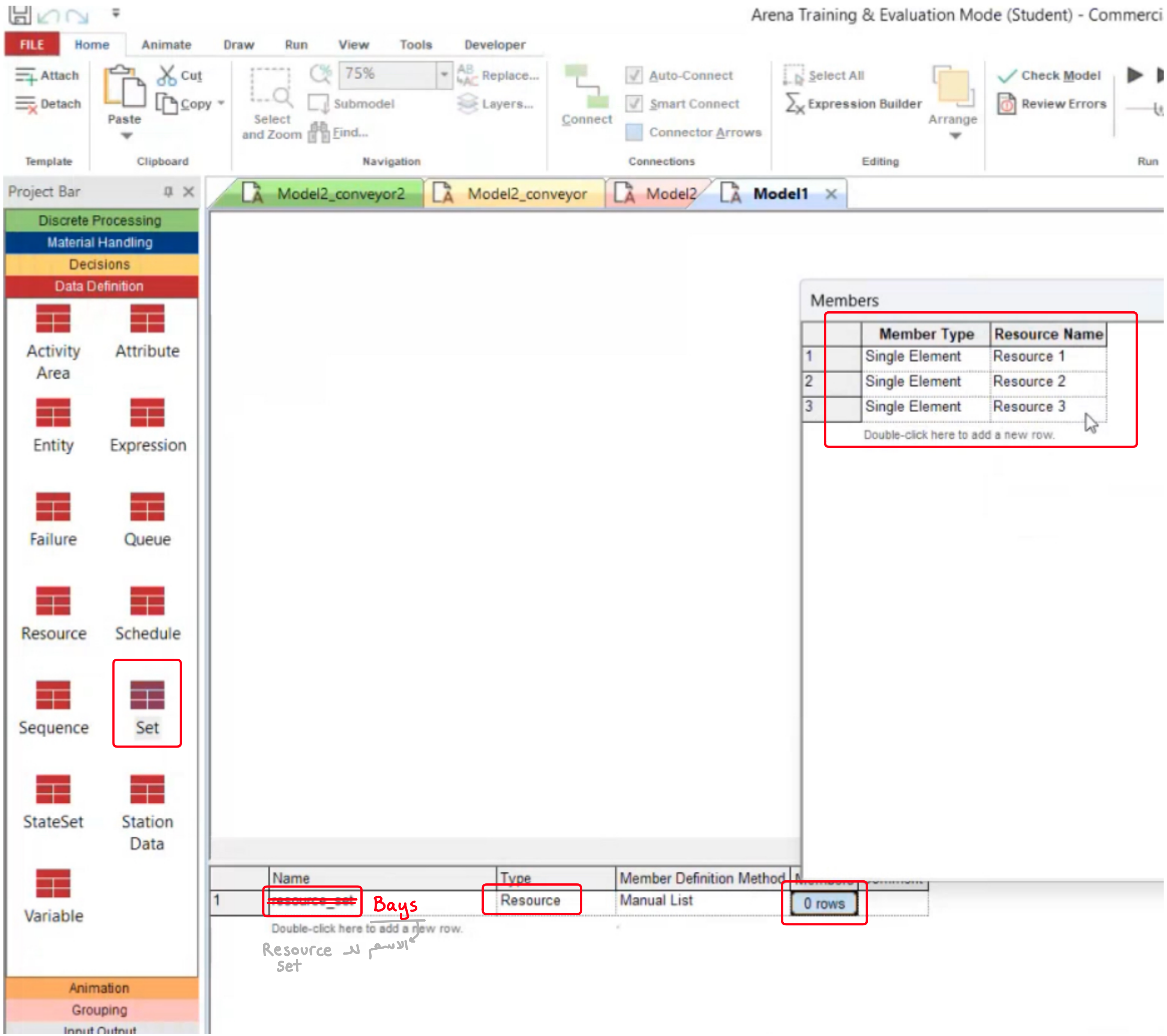
  - Use Resource data module
  - Three resources – one for each service bay
  - Enter Schedule Name
    - Use Preempt option for Schedule Rule to ensure all work stops at the end of the three-hour overtime period


Arena Training & Evaluation Mode (Student) - Commercial Use
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Attach
Detach
Template
Paste
Cut
Copy
Clipboard
Select and Zoom
75%
Submodel
Find...
Replace...
Layers...
Navigation
Connect
Auto-Connect
Smart Connect
Connector Arrows
Connections
Select All
Expression Builder
Arrange
Editing
Check Model
Review Errors
Run
Model2_conveyor2
Model2_conveyor
Model2
Model1
Discrete Processing
Material Handling
Decisions
Data Definition
Activity Area
Attribute
Entity
Expression
Failure
Queue
Resource
Schedule
Sequence
Set
StateSet
Station Data
Variable
Animation
Grouping
Input Output
Name
Type
Schedule Name
Schedule Rule
Busy / Hour
Idle / Hour
Per Use
StateSet Name
Failures
Report Statist
1 Resource 1 Based on Schedule Schedule 1 Wait 0.0 0.0 0.0 0 rows
2 Resource 2 Based on Schedule Schedule 1 Wait 0.0 0.0 0.0 0 rows
3 Resource 3 Based on Schedule Schedule 1 Wait 0.0 0.0 0.0 0 rows

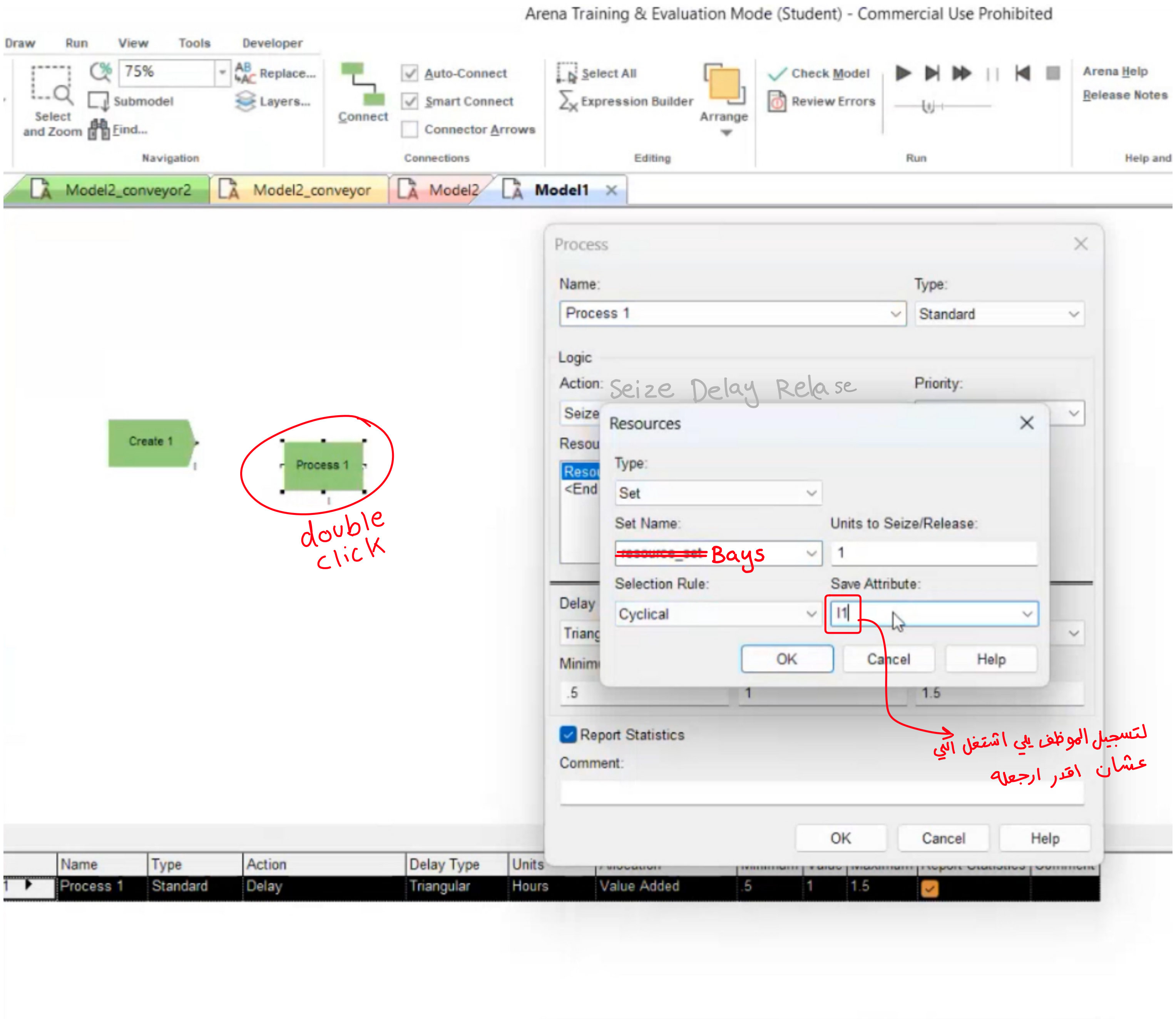
بقدر اسوي الـ Type ← Resource
وهيك بقدر ارجع لنفس الموظف
عشان يكمل شغله مثلاً

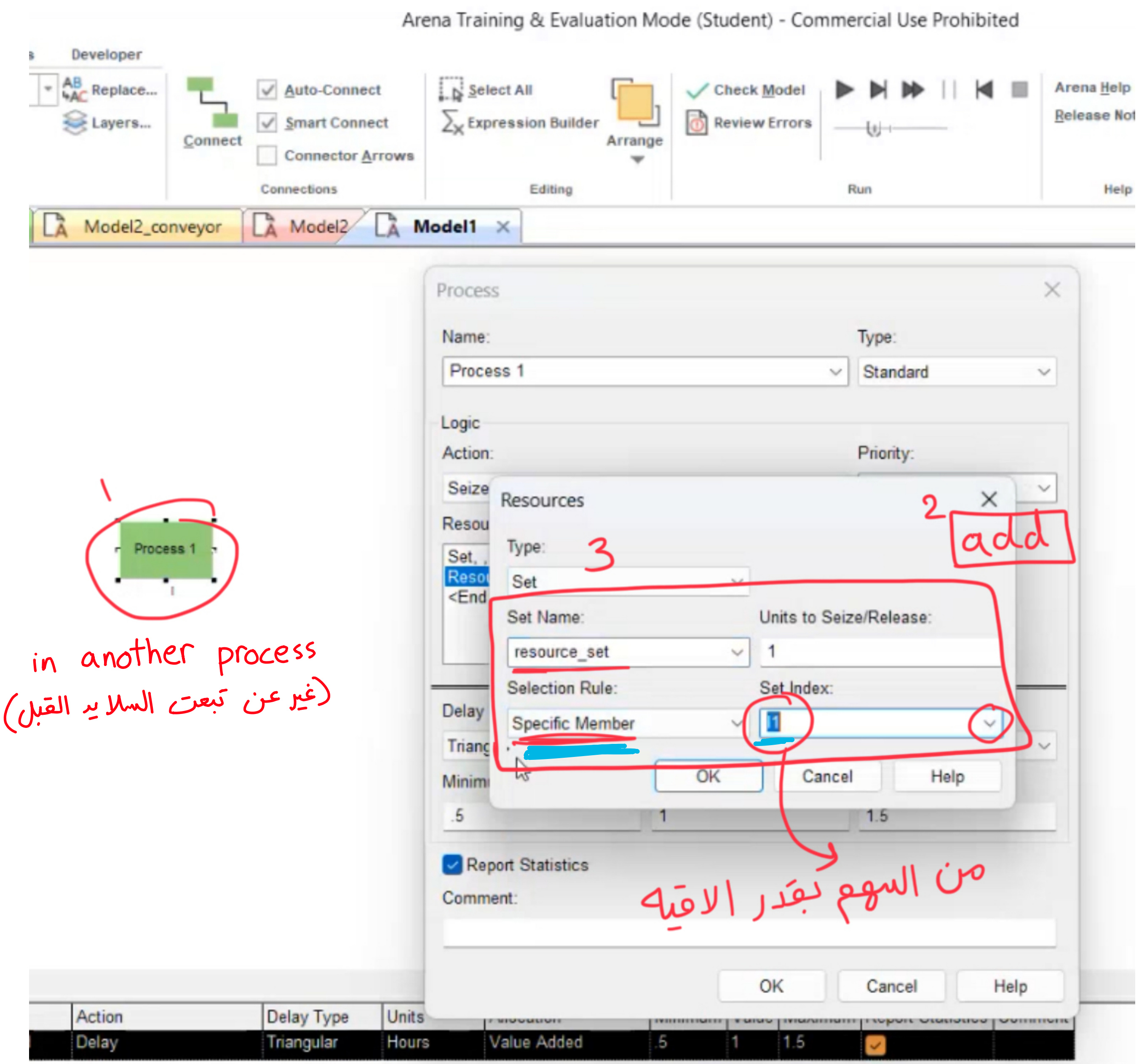
Sets Data

- **Use Set data module (Basic Process panel)**
- **Develop one Resource set**
  - Service bays – `Bays` مكان التصليح
- **Develop two entity picture sets**
  - `Customers`
  - `Vehicles`
- **Develop one queue set**
  - Use Advanced Set data module (Advanced Process panel)
  - First, define appointment queues – Queue data module
  - Then, define `Appointment Queues` set


Arena Training & Evaluation Mode (Student) - Commerci
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Attach
Detach
Template
Cut
Copy
Paste
Clipboard
Select and Zoom
75%
Submodel
Find...
Replace...
Layers...
Navigation
Connect
Auto-Connect
Smart Connect
Connector Arrows
Connections
Select All
Expression Builder
Arrange
Editing
Check Model
Review Errors
Run
Project Bar
Discrete Processing
Material Handling
Decisions
Data Definition
Activity Area
Attribute
Entity
Expression
Failure
Queue
Resource
Schedule
Sequence
Set
StateSet
Station Data
Variable
Animation
Grouping
Model2_conveyor2
Model2_conveyor
Model2
Model1
Members
Member Type
Resource Name
1 Single Element Resource 1
2 Single Element Resource 2
3 Single Element Resource 3
Double-click here to add a new row.
Name
Type
Member Definition Method
1 ~~resource_set~~ Bays Resource Manual List 0 rows
Double-click here to add a new row.
Resource set الاسم لـ

* اذا بدي بال process ارجع إلى نفس الموظف يلي اشتغل بالمرة الأولى مثلاً

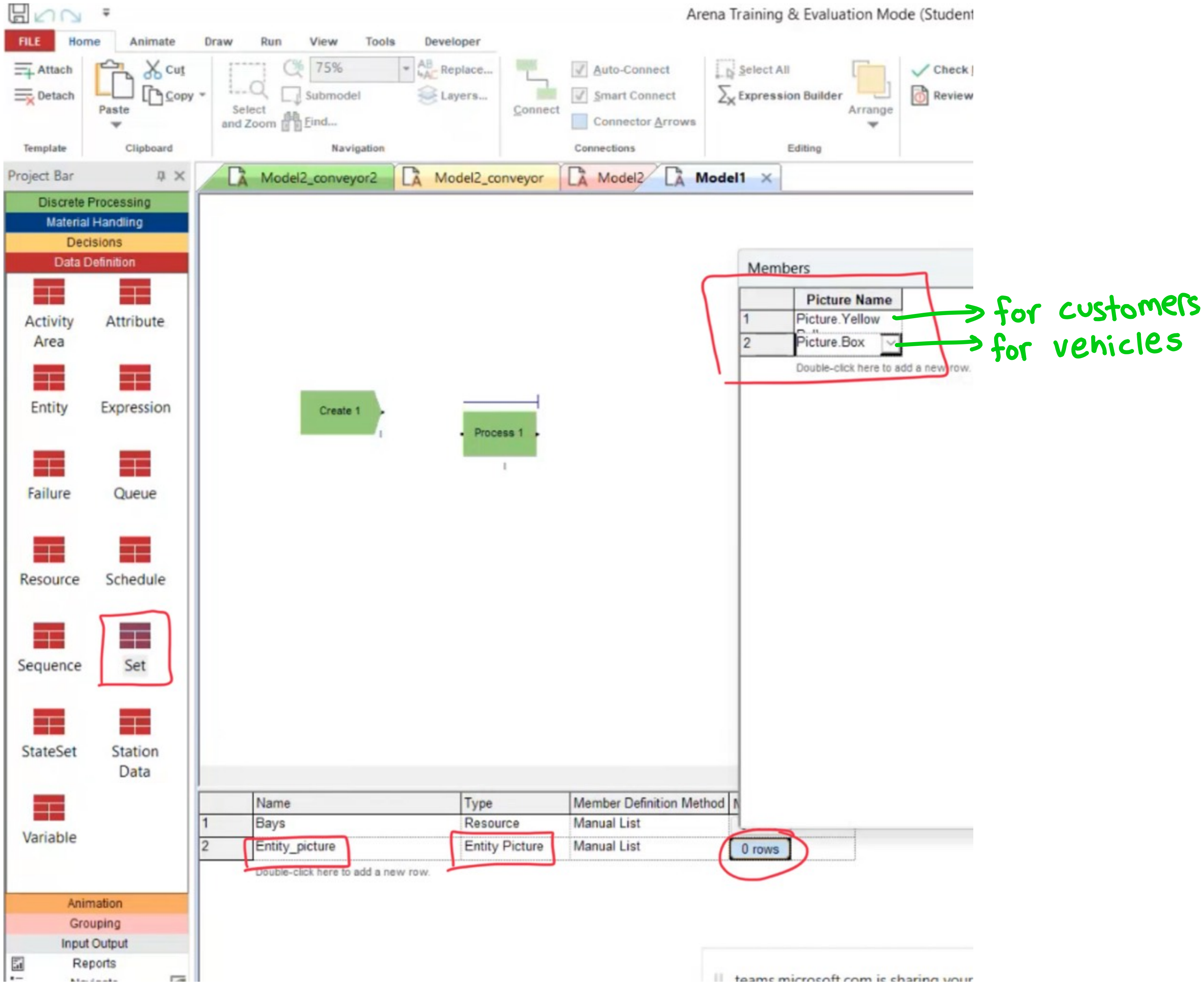




Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
Developer
Replace...
Layers...
Connect
Auto-Connect
Smart Connect
Connector Arrows
Connections
Select All
Expression Builder
Arrange
Editing
Check Model
Review Errors
Run
Arena Help
Release Not
Help
Model2_conveyor
Model2
Model1
Process
Name:
Process 1
Type:
Standard
Logic
Action:
Seize
Priority:
Resou
Set, .
Reso
<End
Resources
Type:
Set
Set Name:
resource_set
Units to Seize/Release:
1
Selection Rule:
Specific Member
Set Index:
1
OK
Cancel
Help
Delay
Triang
Minim
.5
1
1.5
Report Statistics
Comment:
OK
Cancel
Help
Action
Delay Type
Units
Delay
Triangular
Hours
Value Added
.5
1
1.5
1
Process 1
in another process
(غير عن تبعت السلايد القبل)
2
add
3
من السهم بقدر الاقيه

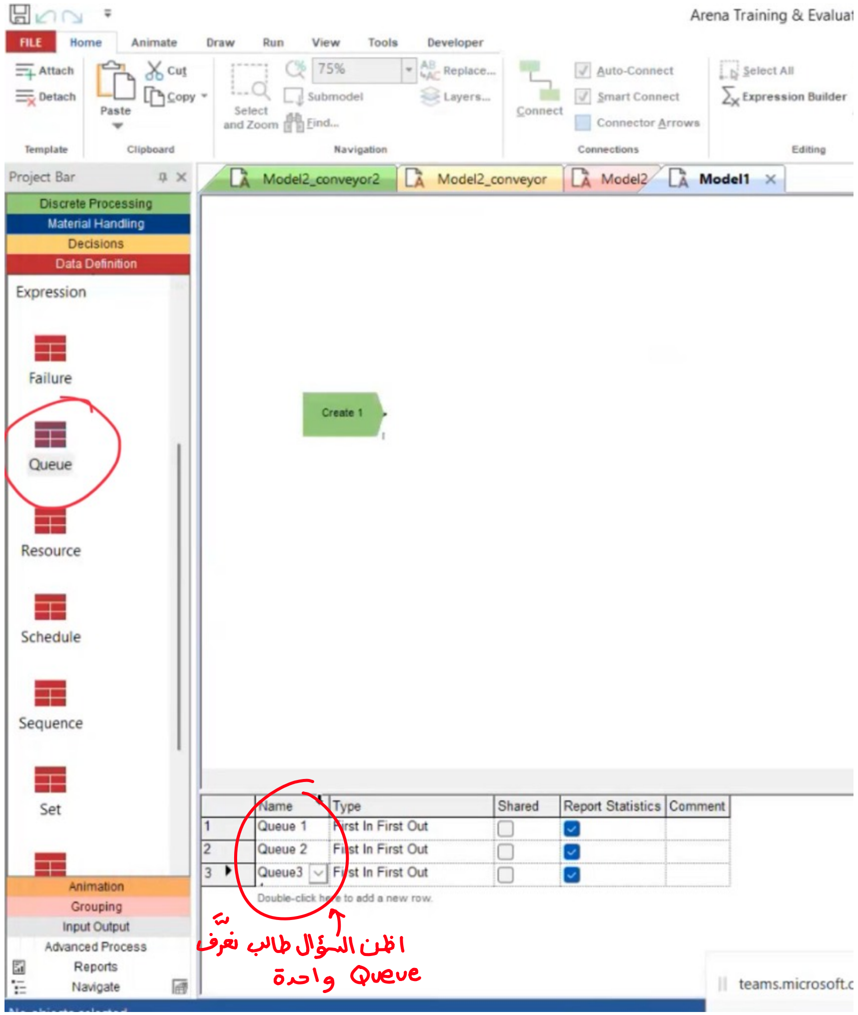
Develop two entity picture sets

- Customers
- Vehicles



- Develop one queue set
  - Use Advanced Set data module (Advanced Process panel)
  - First, define appointment queues – Queue data module
  - Then, define Appointment Queues set

1

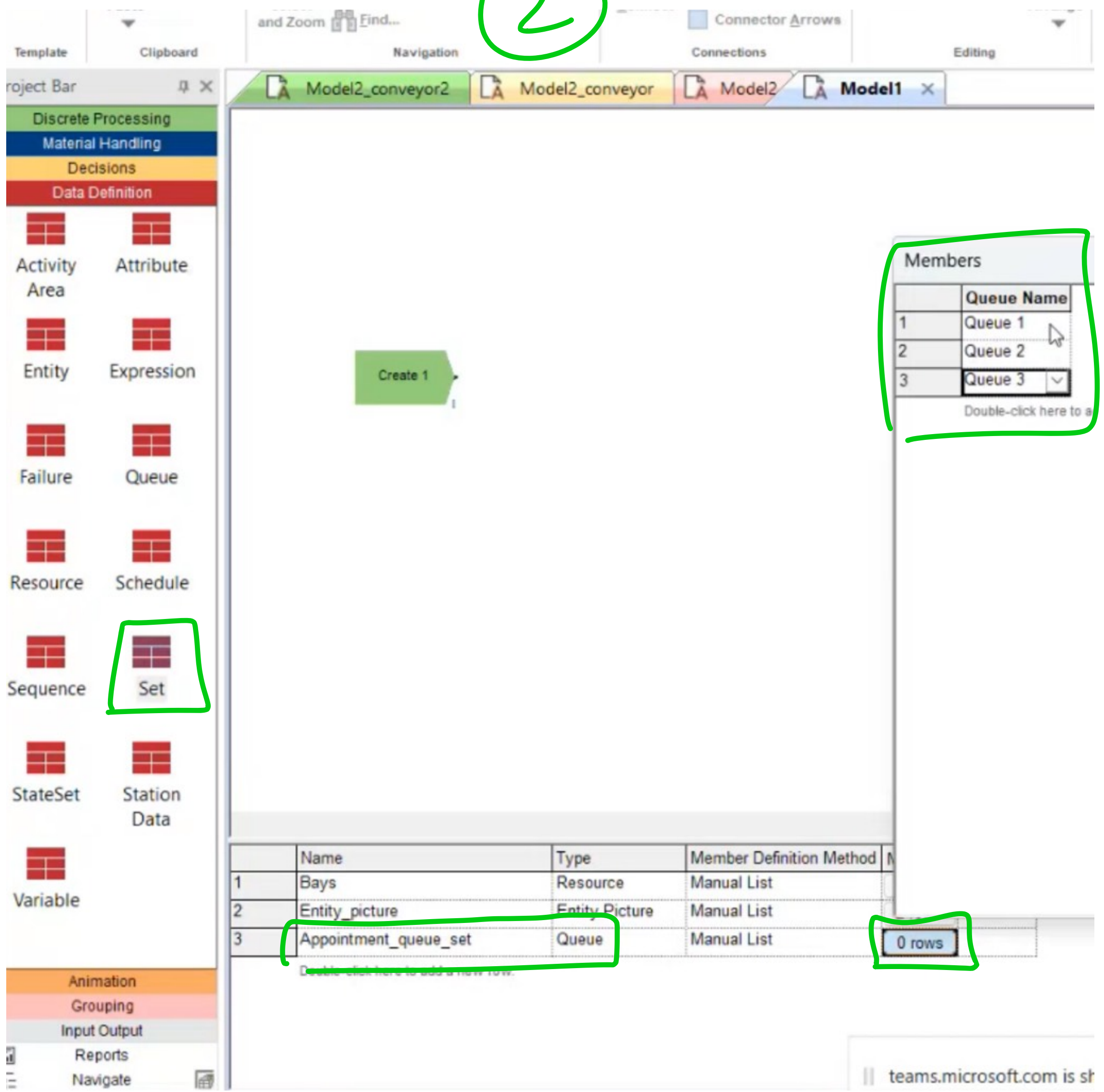


	Name	Type	Shared	Report Statistics	Comment
1	Queue 1	First In First Out		✓	
2	Queue 2	First In First Out		✓	
3	Queue3	First In First Out		✓	

Double-click here to add a new row.

اظن السؤال طالب نعرّف
واحدة Queue

2

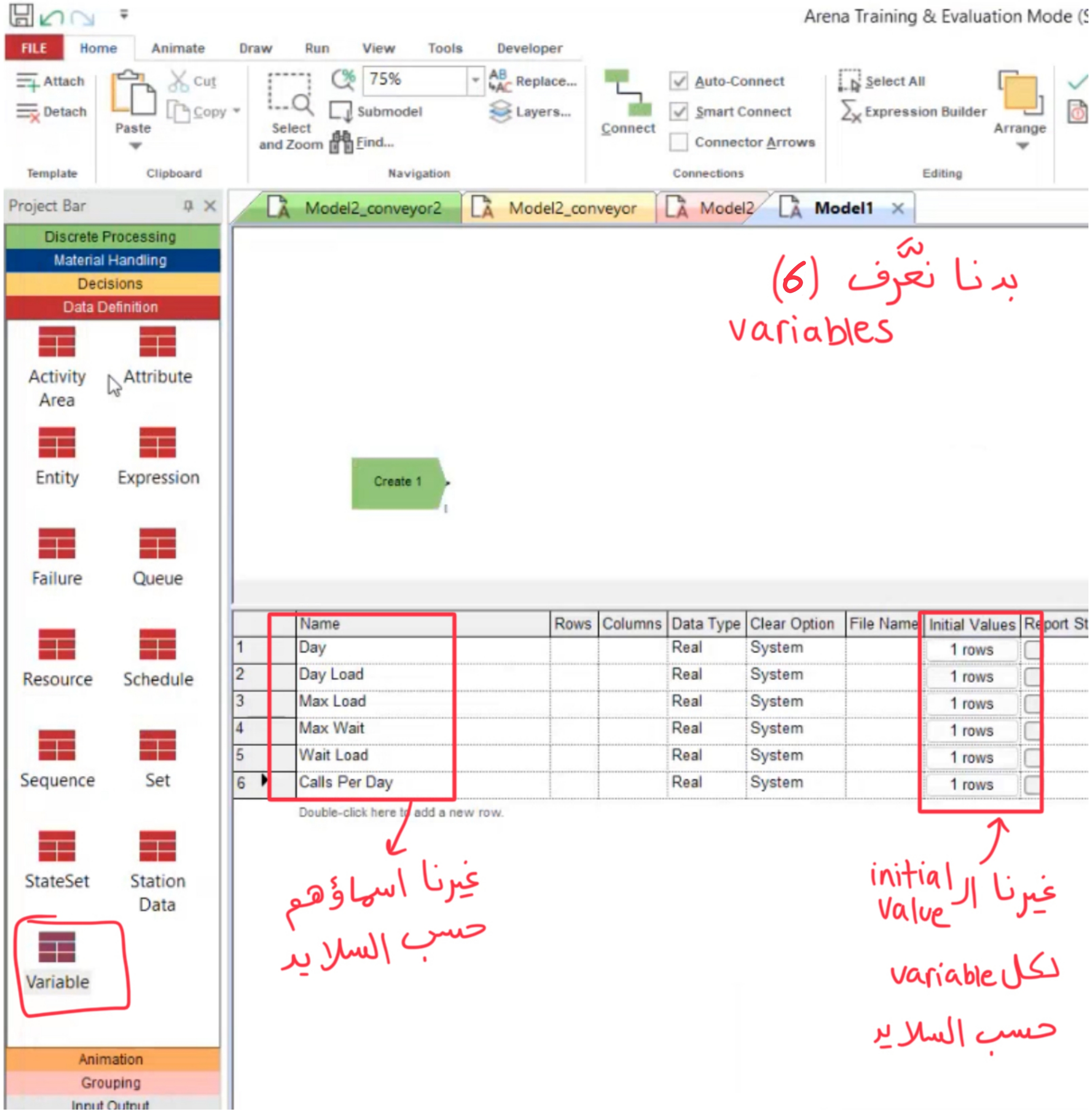


	Queue Name
1	Queue 1
2	Queue 2
3	Queue 3

	Name	Type	Member Definition Method
1	Bays	Resource	Manual List
2	Entity_picture	Entity Picture	Manual List
3	Appointment_queue_set	Queue	Manual List

Variable Data

- **Six control variables**
  - `Day` – current day
    - initialized to 4
  - `Day Load` – current number of appointment hours per day
    - arrayed variable, 5 rows
  - `Max Load` – maximum appointment hours per day
    - initialized to 24
  - `Max Wait` – maximum number of wait customers per day
    - initialized to 5
  - `Wait Load` – current number of wait jobs per day
    - arrayed variable, 5 rows
  - `Calls Per Day` – average number of calls per day
    - initialized to 25



بدنا نعرّف (6) variables

غيرنا أسماؤهم حسب السلايد

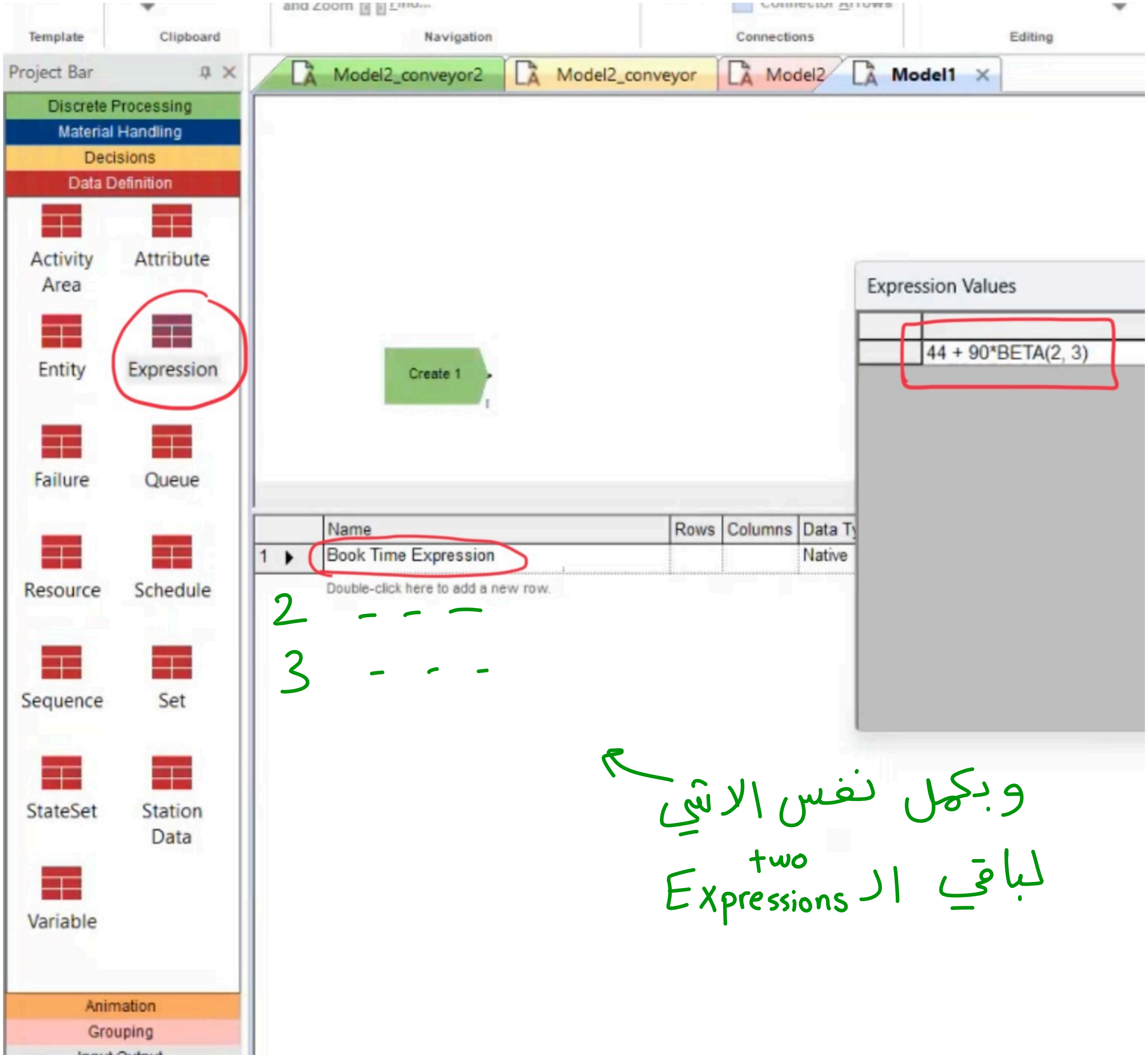
غيرنا ال initial value لكل variable حسب السلايد

Variable Data (cont'd.)

- **Five output variables**
  - `Day Book Time` – total book time hours processed
  - `Day Actual time` – total actual service time processed
  - `Day Overtime` – total overtime hours
  - `Day OT` – total number of vehicles held over night
  - `Day Wait Late` – total number of late wait jobs
- **Four control-logic variables**
  - `Work Day` – the current work day
  - `Day Start Time` – starting time of current day
  - `Day End Time` – ending time of current day
  - `Wait Allowance` – wait allowance time
    - Initialized to 1 (hour)

Expression Data

- **Three expressions**
  - `Book Time Expression` – returns the service book time (in hours) from Slide(4) → Book time expression is (~44+90*BETA(2,3)minutes
  - `Wait Priority Expression` – returns 1 for wait customer, 2 for leave customer
  - `Actual Service Time Expression` – returns the actual service time based on the Book Time



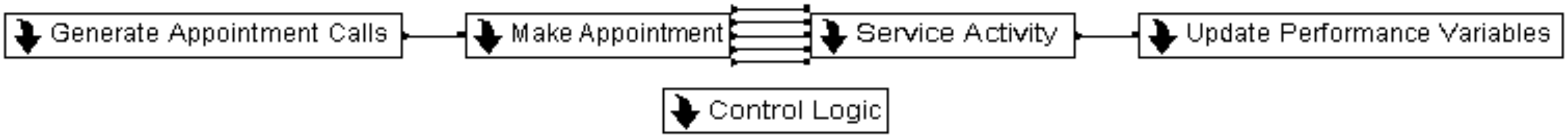
2 - - -
3 - - -

وبكمل نفس الاشي
لباقي ال two Expressions

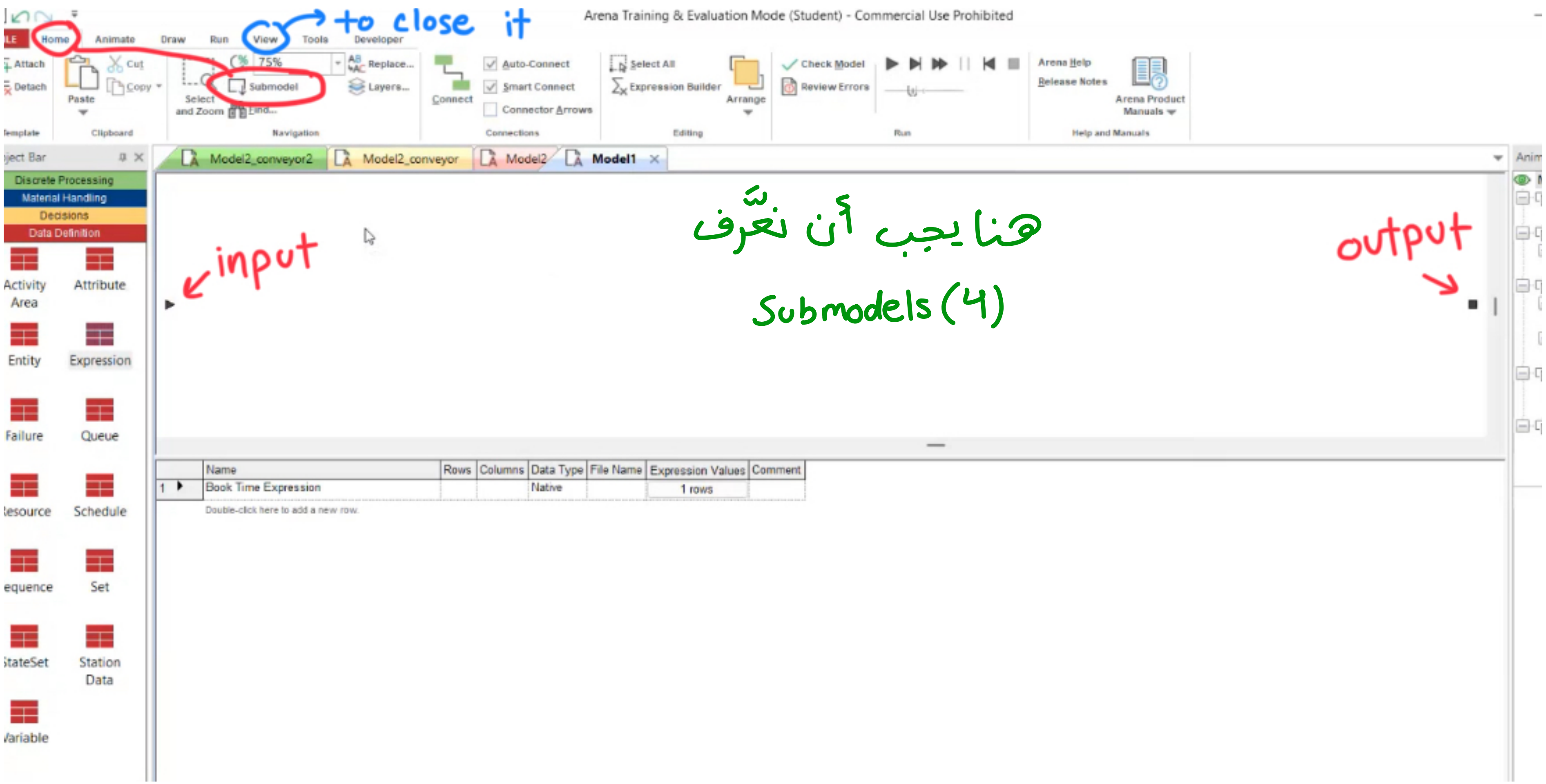
Output Data

- **Use Statistic data module**
- **Five output values**
  - `Daily Book Time` – daily average book time processed
  - `Daily Actual Time` – daily average actual time processed
  - `Daily Overtime` – daily average overtime
  - `Daily Late Wait Jobs` – daily average number of late wait jobs
  - `Daily Profit` – average daily profit
- **All are of type Output, so will be computed only at the end of the simulation**

Submodel Creation

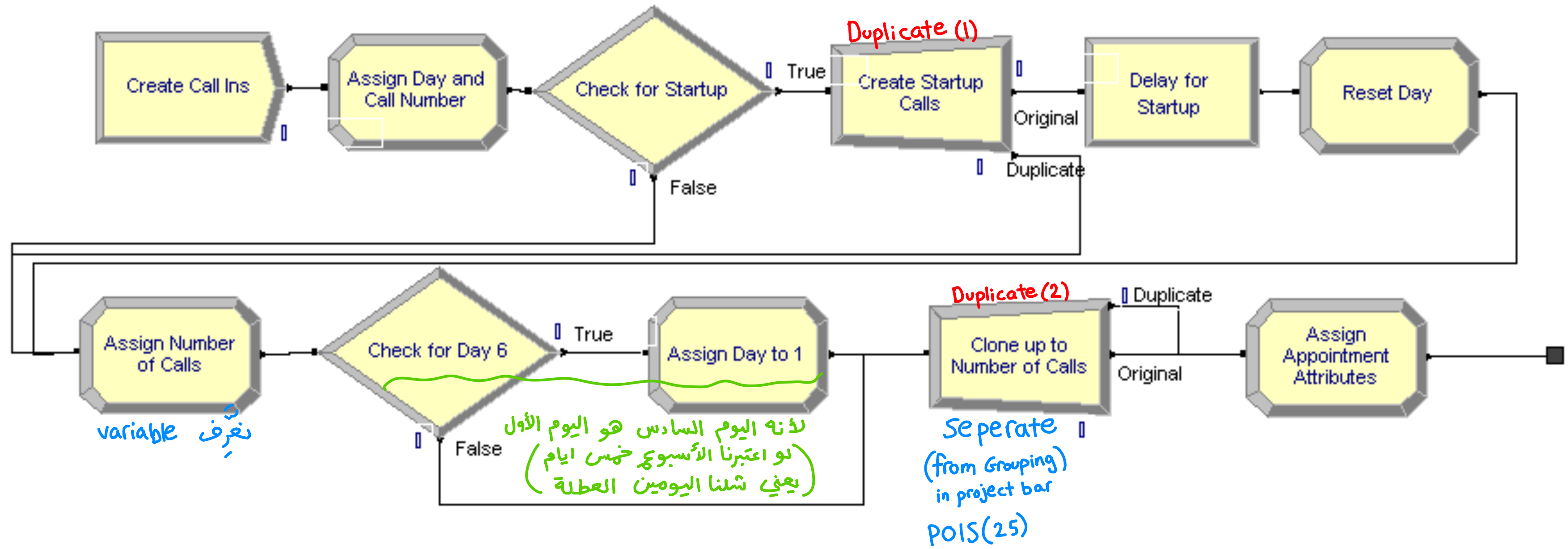


- ***Object/Submodel/Add Submodel*** **menu option (or ) to create a submodel … we'll use five submodels**
  - Define (right-click, then Properties)
    - Name
    - Number of entry, exit points (could be 0 if there's no flow interaction)
  - Move between submodels: Navigate panel, Named Views, or mouse
    - Double-click on a submodel to open it
    - When in a submodel, right-click in an empty place, then Close Submodel to go up



Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
Home
Animate
Draw
Run
View
Tools
Developer
to close it
Attach
Detach
Paste
Cut
Copy
Clipboard
Select and Zoom
75%
Submodel
Replace...
Layers...
Navigation
Connect
Auto-Connect
Smart Connect
Connector Arrows
Connections
Select All
Expression Builder
Arrange
Editing
Check Model
Review Errors
Run
Arena Help
Release Notes
Arena Product Manuals
Help and Manuals
Model2_conveyor2
Model2_conveyor
Model2
Model1
Discrete Processing
Material Handling
Decisions
Data Definition
Activity Area
Attribute
Entity
Expression
Failure
Queue
Schedule
Set
Station Data
input
هنا يجب أن نعرّف
Submodels (4)
output
Name
Rows
Columns
Data Type
File Name
Expression Values
Comment
Book Time Expression
Native
1 rows
Double-click here to add a new row.

Generate Appointment Calls Submodel



- **Generates the daily appointment calls**
- **No entry point – one exit point**
- **First day of simulation is different**
  - Generate three duplicates to make appointments for the initial three days

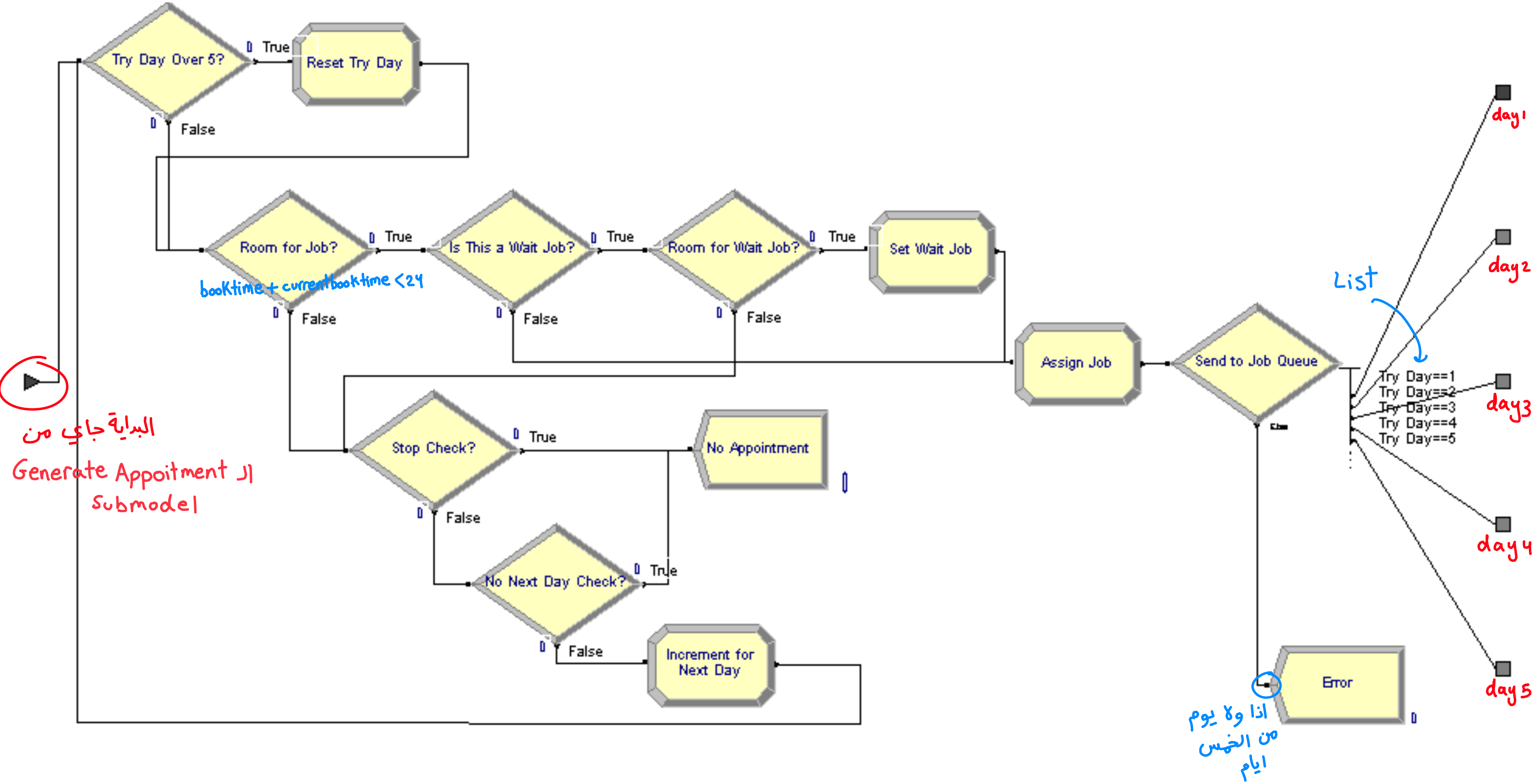
Submodel Logic

- **Create one entity each morning (at just > 0)**
- **Increment `Day` & set `Call Number` attribute to 1**
- **If start of replication (TNOW < 1)**
  - `Day` was initialized to 4 (Thurs.) so it's now 5 (Fri.)
  - Create three duplicate entities for next Mon., Tues., Wed. appointments
    - Send the three duplicates immediately to *** below to schedule initial 3 days out (Mon., Tues., Wed.)
    - Delay original entity (representing calls coming in today) a bit so duplicates will do the future scheduling first, and reset `Day` back to 1 for calls coming in on Monday
- *******
  - Assign number of calls (N) for this day
  - If `Day` is 6 (Saturday), set it back to 1 (Monday)
  - Create N - 1 duplicate entities
  - Assign appointment attributes

Generate Appointment Calls Submodel
(cont'd.)

- **Create a single entity each day (every 12 hours)**
- **Assign `Number of Calls` for this day**
- **Increment `Day` variable**
- **Create duplicates (`Number of Calls` – 1)**
  - Separate module for "cloning"
- **Assign call attributes (*after* cloning to make them distinct)**
  - `Day Inc` – first feasible appointment day out (1, 2, or 3)
  - `Priority` – wait (1) or leave (2)
  - `Book Time`
  - `Try Day` – first day of week customer wants appointment
  - `Entity.Picture` – from picture set `Customers`
- **Entities exiting this submodel are individual calls**

Make Appointment Submodel



Make Appointment Submodel (cont'd.)

- **One entry point**
  - From Generate Appointment Calls Submodel
- **Five exit points**
  - Corresponds to five appointment days
- **Arriving entity is a customer call wanting to make an appointment**
- **Contains checks to see if customer *can* make an appointment**

Make Appointment Submodel (cont'd.)

- **Check for valid `Try Day` value**
  - If not 1 through 5, subtract 5 from it
- **Check for room in schedule**
  - Is `Book Time + Day Load(Try Day) <= Max Load`?
  - If not, check for next day
- **Check for Wait job**
  - If this is a wait job, check for room
    - Is `Wait Load(Try Day) < Max Wait`?
    - If not, check for next day
    - Otherwise, increase `Wait Load` variable by 1

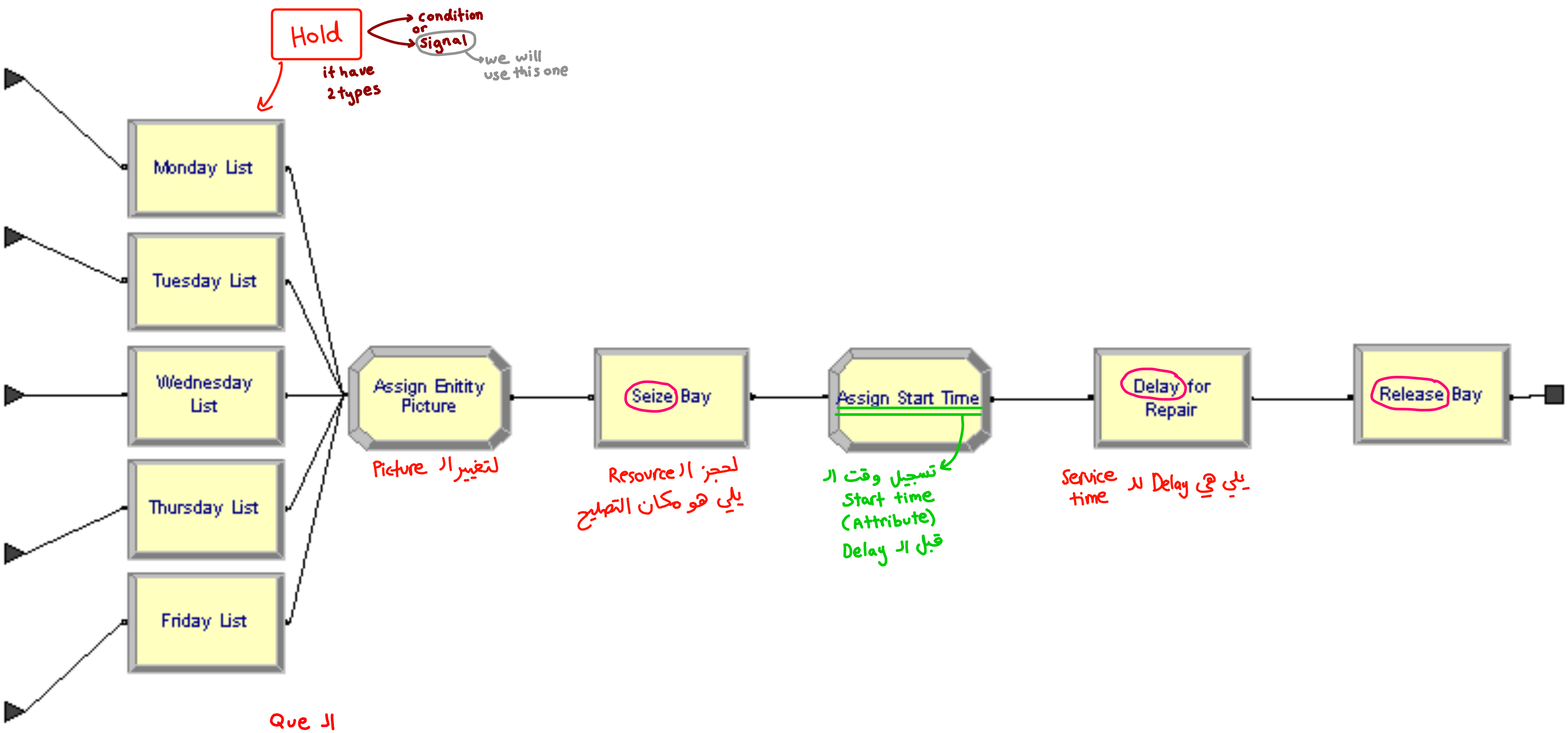
Make Appointment Submodel (cont'd.)

- **Assign Job**
  - Increase Variable `Day Load(Try Day)` by `Book Time`
- **Send to job queue**
  - Decide module based on appointment day
  - Send to one of five exit points
  - Error trap should never be taken
- **Appointment not possible – check for next day**
  - If `Day Inc` variable >= 3, no next day (dispose)
  - Decide module with 90% chance of next day
    - Increment `Day Inc` and `Try Day` variables and start over
    - Otherwise, dispose

Call → عبارة عن Entity

السيارة → Entity

Service Activity Submodel



The Hold → either ... or → connected to a different Signal

Service Activity Submodel (cont'd.)

- **Five entry points**
  - Corresponding to five appointment days
- **One exit point**
  - Service complete, send to Update Performance Variables submodel
- **Arriving entity is a customer appointment**
- **Holds entity until service day**
- **Actual vehicle service**

Service Activity Submodel (cont'd.)

- **Arriving entity enters customer appointment queue**
  - One of five Hold modules
  - Entities are released based on a signal sent from the Control Logic submodel
    - Signal value of variable Day is sent at the start of each work day
- **Assign new picture from picture set Vehicle**
  - Based on service type – attribute `Priority`
- **Assign current time to attribute `Arrive Time`**

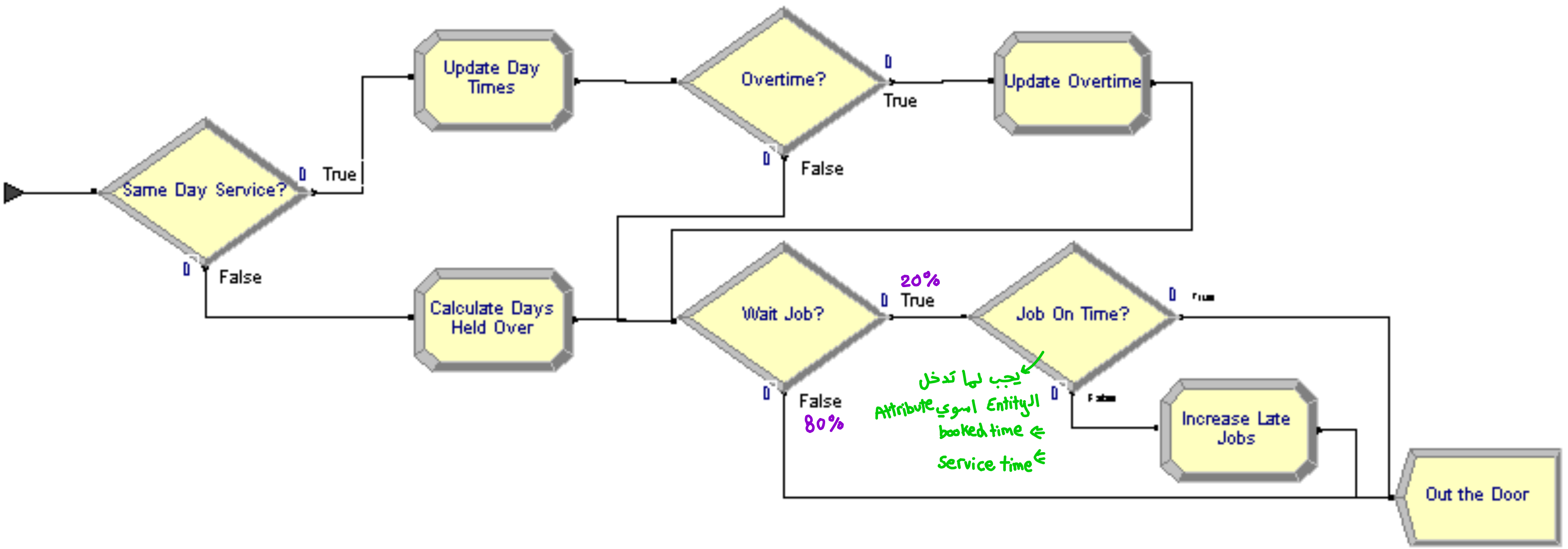
Service Activity Submodel (cont'd.)

- **Seize service bay**
  - From resource set `Bays`
- **Assign service start time to attribute `Start Time`**
- **Delay for actual service time**
- **Release service bay**
- **Exit submodel**

Update Performance Variables Submodel

Collecting the outputs

for Output



Update Performance Variables Submodel (cont'd.)

- **One entry point**
  - Completed service entity
- **No exit points**
- **Updates the performance variables**
- **Disposes of entity**

Update Performance Variables Submodel (cont'd.)

- **Check for same-day service – was job started on a previous day?**
  - Same day
    - Update `Day Book Time` and `Day Actual Time` variables
    - Check for overtime
      - Overtime – update `Day Overtime` variable
    - Check for wait job
      - If yes, check for wait job on time
        - » If yes, dispose; if no, update `Day Wait Late` variable
      - If no, dispose
  - Not same day
    - Calculate days held over – attribute `Days Held`
    - Update `Day Book Time`, `Day Actual Time`, `Day Overtime`, and `Total OT` variables
    - Repeat wait job check (above)

Control Logic Submodel

بسبب ال Hold

it is not a car
and
it is not a call
(it is a redundant (dummy) Entity)

الدليل انها ما مرت
بال Seize Delay
Release

- **No exit or entry points**
- **Causes current day's appointments to be released for service**
- **Updates control variables**

Control Logic Submodel (cont'd.)

ما كملنا شرح

- **Create control entity**
  - One every 12 hours – one each day
  - Offset first creation by 0.9999 hours (see text for details)
- **"Broadcast" Signal = current value of variable `Day`**
  - Causes the day's appointments to be released for service
- **Reset `Wait Load` and `Day Load` variables for current day to zero**
- **Update `Work Day`, `Day Start Time`, and `Day End Time` variables**
- **Dispose of control entity**

Finding and Fixing Model Errors

- **Arena picks up "simple" errors in Check phase, and leads you to them via Find and Edit buttons in Errors/Warnings windows**
  - Undefined variables, attributes, resources
  - Unconnected modules
  - Duplicate module names
  - Typos

Finding and Fixing Model Errors (cont'd.)

- **Other kinds of errors are more complex, can't be detected without trying to run**
  - Options on Run Interaction toolbar or on Run menu
  - Only mention capabilities here; see text for details
- **Run Controller**
  - Command-driven window to control, display details about model operation and underlying SIMAN code
- **Highlight active module**
  - Highlights the active module during the simulation run

Finding and Fixing Model Errors (cont'd.)

- **Layers**
  - Gives control over what you see during the simulation run
- **Break on Module; Break**
  - Stop run when entity hits a selected module, at a specific time, or when a selected entity is about to become active
- **Watch**
  - Select expressions to display in a window as model runs
- **Look at reports when model is running or paused**
  - Remember to close reports windows

Animating the Automotive Shop Model

- **No "normal" entity movement**
  - Entities jump from queue to queue to resource
- **Appointment and service-bay queues**
  - Queue button from Animate toolbar
- **Appointment queue variables**
  - Book Time hours
  - Number of appointments
  - Number of Wait appointments
- **Resource animation**
  - Resource button from Animate toolbar
  - Take pictures from libraries (.plb files), different states
- **Add various text annotations, boxes, etc.**

Animating the Automotive Shop Model

(cont'd.)

- **Created digital clock "by hand" (details in text)**
  - Could have used ready-made animated clock
    - Only because day is 12 hours
- **Current day of week**
  - Global variable
  - Use Global button from Animate toolbar
  - Trigger value based on variable Day
  - Text picture contains week day name
    - Monday through Friday

Model 5-2: Enhancing the Automotive Shop Model

- **New Elements**
  - Not all jobs can be done in all bays
    - All jobs can be done in Bays 2 & 3 (60%)
    - Bay 1 can only handle a subset (40%)
  - Not all customers arrive on time
    - 60% to 70% arrive on time (uniform distribution)
  - Some customers never show up

New Modeling Issues

- **Sets**
  - Overlapping Resource Sets
    - Set `Bays` contains Resources `Bay 1`, `Bay 2`, and `Bay 3`
    - Set `Bays 2 or 3` contains Resources `Bay 2` and `Bay 3`
- **Resource Logic**
  - Multiple queues requesting the same resource
    - Seize priority
    - Tie breaker is entity waiting the longest
    - Shared queues
    - See text for details

New Modeling Issues (cont'd.)

- **Nonstationary Poisson arrival process**
  - Arrivals occur one at a time and are independent of one another
  - Expected rate *varies* over time (would be constant for a stationary Poisson process)
  - Generation capability built into Create module
    - Type = Schedule
    - Specify arrival-rate function in Schedule model
    - Limited to piecewise-constant arrival-rate functions
  - Used for late customer arrivals

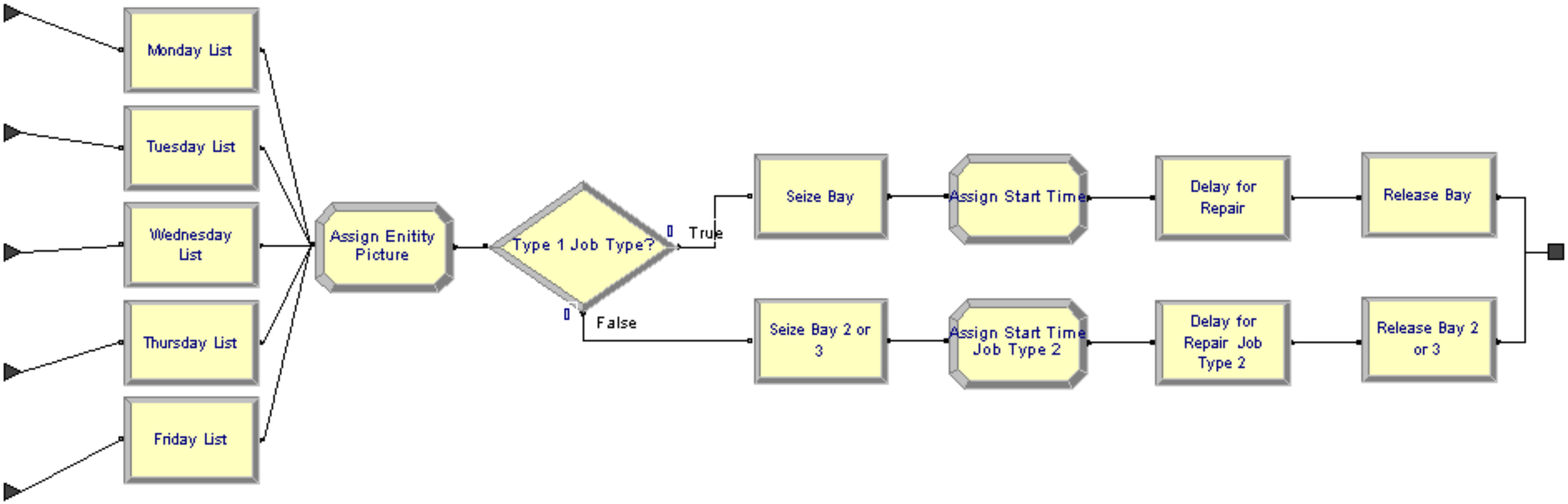
Changes to Model 5-1

- **Add/change logic to account for difference in service bay capabilities**
- **Add/change logic for new customer arrival process**

Service-Bay Logic Changes

- **Generate Appointment Calls submodel**
  - Add assignment for `Job Type` attribute
    - 1 if job can be serviced in any bay
    - 2 if only bays 1 or 2
    - Generate via DISC(0.60,1, 1.0,2)
- **Change picture set Vehicle**
  - Put yellow doors on new job types (see text for details)

Service-Bay Logic Changes (cont'd.)



- Modify logic in Service Activity submodel
- Assign new picture
  - Vehicle(Priority + (AINT(Job Type/2))*2)

Service-Bay Logic Changes (cont'd.)

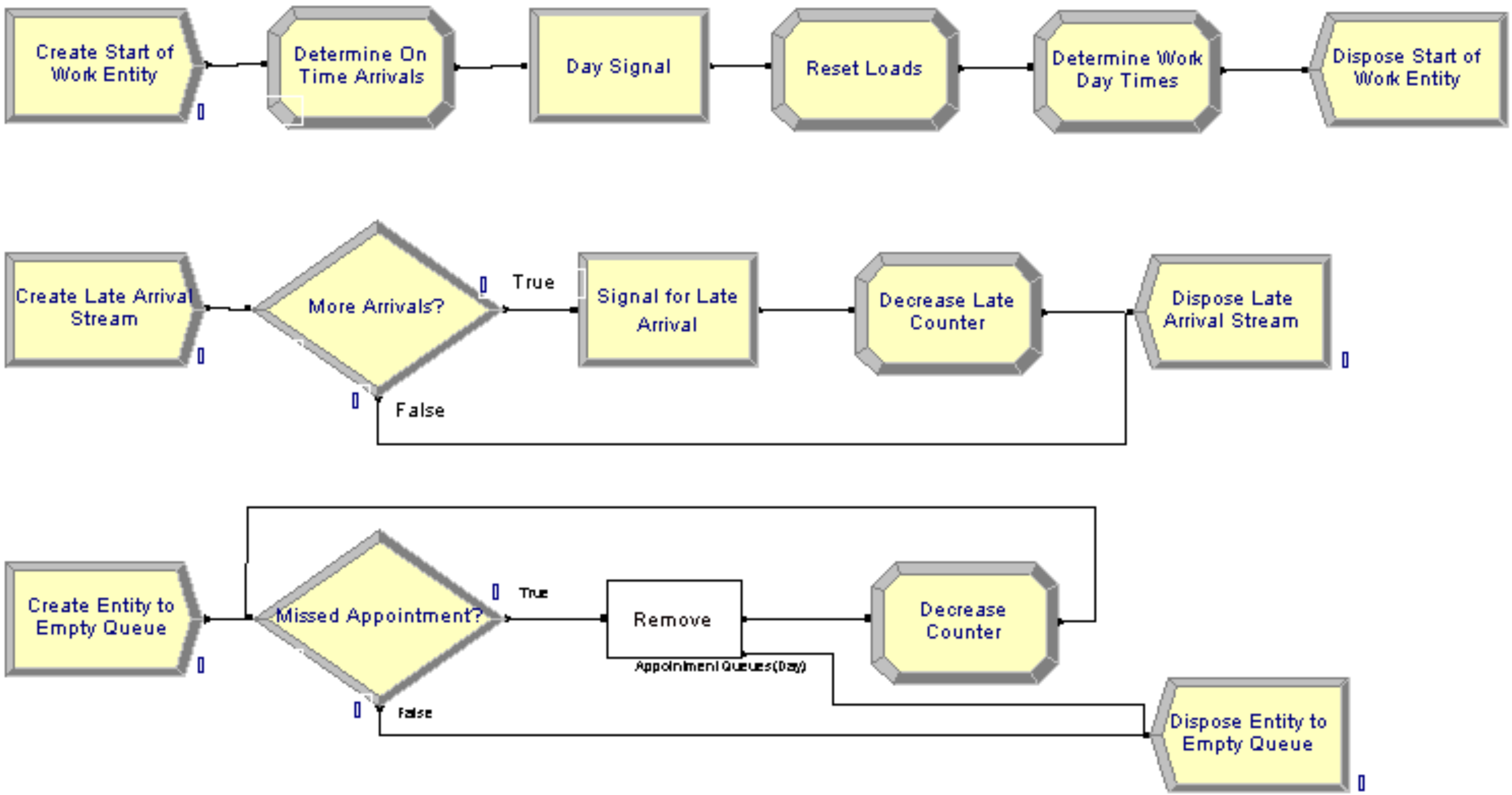
- **Insert Decide module**
  - Direct entity based on attribute `Job Type`
  - If `Job Type` = 1, send to existing logic
  - If `Job Type` = 2, send to new logic
    - New logic copy of existing logic
    - Change resource set to `Bays 2 or 3`
- **Change queue rankings**
  - Queue data module
  - Lowest Attribute Value First
    - Based on the attribute `Priority`

Service-Bay Logic Changes (cont'd.)

- **Change seize priorities**
  - Existing Logic: (Priority*10) +1
  - New logic: Priority*10
- **Resulting queue priorities**
  - Priority 1 and Job Type 1 with a Seize priority of 11
  - Priority 2 and Job Type 1 with a Seize priority of 21
  - Priority 1 and Job Type 2 with a Seize priority of 10
  - Priority 2 and Job Type 2 with a Seize priority of 20

Customer Arrival Process

- **Change Control Logic submodel**



Customer Arrival Process (cont'd.)

- **On-time customers**
  - Insert new Assign module
    - Variable `On Time`

      `ANINT( UNIF( 0.60 , 0.70) * NQ(Appointment Queues( Day) ) )`
    - Variable `Late`

      `NQ(Appointment Queues( Day) ) - On Time`
    - Variable `Total Late`

      `Total Late + Late`
  - Change Signal module
    - Set signal Limit to variable `On Time`
    - Only the on-time customers are released from appointment queue

Customer Arrival Process (cont'd.)

- **Late-arriving customers**
  - Arrive in first two hours after start of service
  - Not all customers show up for appointment
  - Data in 15-minute time periods (8 periods)
  - Use Create module with arrival Type of Schedule
- **Add new schedule**
  - `Late Arrival Schedule`
  - 48 time periods (12 hours)
  - First four and last 36 have zero arrival rate

Customer Arrival Process (cont'd.)

- **Late arrival logic**
  - Create late arrival entity
    - Start one hour into day
  - Check appointment queue
    - Entity in queue, signal with Limit of 1 to release customer
    - No entity in queue, dispose of entity
- **No-show customer logic**
  - Create control entity three hours into day
  - Check appointment queue
    - Entity in queue
      - Remove first entity and send back to check
    - No entity in queue
      - Dispose of entity

Model 5-3:
An (*s*, *S*) Inventory Simulation

- **Different kind of model – not queueing**
- **Will use modules from the Blocks and Elements panels exclusively – SIMAN simulation language**
  - Doing this mostly just to demonstrate this capability
  - Could be done with higher-level panels we've been using
- **Company carries a single discrete item (widgets) in inventory**
- **$I(t)$ = inventory level (an integer) at time t days past the beginning of the simulation; $I(0) = 60$**
- **Run simulation for 120 round-the-clock days**

Customer Demands Against Inventory

- **Customer interarrival times ~ exponential with mean 0.1 day (round the clock)**
  - First arrival not at time 0 but after an interarrival time past 0
- **Demand size is discrete random variable**
  - 1, 2, 3, 4 with respective probabilities 0.167, 0.333, 0.333, 0.167
- **If enough items are physically on hand in inventory to satisfy a demand, customer gets demand and leaves**
- **If demand > number of items on hand, customer gets whatever is there and the rest of the demand is backlogged ($I(t)$ becomes negative)**
  - If $I(t)$ was already negative, it just goes more negative

Inventory Review, Replenishment

- **"Take inventory" at beginning of each day**
  - So at exactly times 0, 1, 2, ..., 119 (not 120 ... see below)
  - Two managerially-chosen constant integers $s = 20$ and $S = 40$ (must have $s < S$ if we change these values)
  - If $I(t) \geq s$, do nothing until next inventory evaluation exactly one day later
  - If $I(t) < s$, order $S - I(t)$ items from supplier (order "up to" S)
  - Order does not arrive instantly from supplier, but after a *delivery lag* (a.k.a. *lead time*) ~ UNIF(0.5, 1.0) day, so sometime during the last half of the day of ordering
    - In the meantime, inventory level could fall further from additional demands, so inventory level will not necessarily pop up to S when the order arrives, but to something less than S

Cost Structure

- **Average ordering cost per day**
  - When an order is placed, incur a fixed cost of \$32, plus an incremental cost of \$3 per item ordered
  - If no order is placed at the beginning of a month, there's no ordering cost, not even the fixed cost
  - At end of simulation, divide total of ordering costs by 120
- **Average holding cost per day**
  - Whenever $\mathtt{I}(t) > 0$, incur \$1 per day per item on hand
  - Average holding cost = $\int_0^{120} 1 \times \max(\mathtt{I}(t), 0)\, dt \Big/ 120$
- **Average shortage cost per day**
  - Whenever $\mathtt{I}(t) < 0$, incur \$5 per day per item in backlog
  - Average shortage cost = $\int_0^{120} 5 \times \max(-\mathtt{I}(t), 0)\, dt \Big/ 120$

Cost Structure (cont'd.)

- **During periods when $I(t) = 0$ there's neither holding nor shortage cost incurred**
- **Overall performance measure**

  = Average *total* cost per day

  = sum of average ordering, holding, and shortage costs per day
- **Don't evaluate inventory at time 120**
  - We might order and incur an ordering cost then, but order will never arrive
  - We'll fudge this, but an Exercise asks you to do it right

Data Structure

- **Use Blocks, Elements panels exclusively**
  - Even for Variables, Expressions, Attributes, Entities, statistics collection, and run control
- **Variables Element (initialized, or default to 0 initially)**
  - `Inventory Level` = $I(t)$, changes during run, initialized to 60
  - `Little s` = s = 20
  - `Big S` = S = 40
  - `Total Ordering Cost` accumulator
  - `Setup Cost` = 32
  - `Incremental Cost` = 3
  - `Unit Holding Cost` = 1
  - `Unit Shortage Cost` = 5
  - `Days to Run` = 119.9999 (The Fudge)

Data Structure (cont'd.)

- **Expressions element**
  - Define `Interdemand Time`, `Demand Size`, `Evaluation Interval`, `Delivery Lag`
    - **Note *cumulative* probabilities in DISC function for `Demand Size`**
- **Attributes, Entities elements**
  - Just to define these objects
- **Project, Replicate elements**
  - Similar to *Run > Setup*
- **DStats element**
  - To ask for accumulation of integrals for total holding, shortage costs

Data Structure (cont'd.)

- **Outputs element**
  - Two entries, both of Data type "Output" so that they're executed only at end of run, and reported
  - Avg Ordering Cost computed
  - Avg Total Cost added up
    - OVALUE returns most recent value
    - DAVG returns time-persistent average

Logic for Customer Demands Against Inventory

- **Create block for arrival**
  - Entity Type is **`Customer`**
  - Uses **`Interdemand Time`** Expression
  - First Creation after an **`Interdemand Time`**
- **Assign block to decrement `Inventory Level` by a `Demand Size`**
  - **`Demand Size`** was defined as an Expression
  - Backlogging naturally happens
- **Dispose block for customer exit**
  - If backlogged, is accounted for automatically in the (simple) definition and tracking of **`Inventory Level`**

Inventory Evaluation

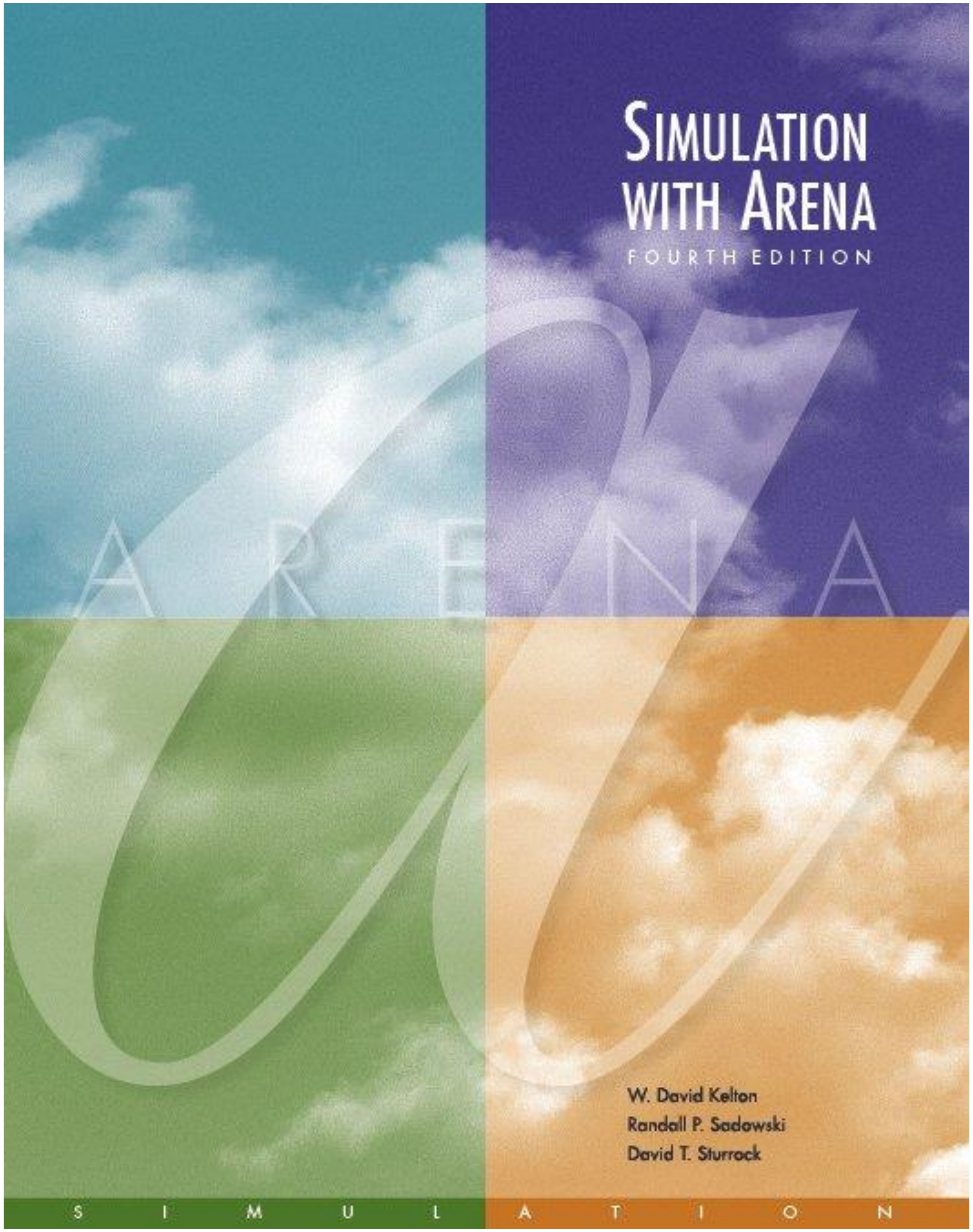
- **Create block for `Inventory Evaluator` entities**
  - First Creation at time 0 – evaluate inventory at start of run
  - Interval is **`Evaluation Interval`**, defined as Expression
- **Branch block – somewhat like Decide module**
  - To determine whether to place an order now
  - Add "branches," each evaluated as true or false
  - Clone of incoming entity sent out along each "true" branch, but at most Max Number of Branches will be sent out
    - So we set Max Number of Branches to `1` (default is ∞)
  - First branch of type "If" – if "true" we want to order
  - Second branch of type "Else" – if "true" it means that the first branch was "false" so we don't order – just Dispose

Placing an Order

- **If we exit the Branch block via the top "If" branch, it must be that $I(t) < s$ so we want to order up to S**
- **Assign block**
  - Define `Order Quantity` Attribute
    - Could have made this a Variable in this model with these parameters, but it's more general for it to be an Attribute ... why??
  - Increment `Total Ordering Cost` Variable
- **Delay block for `Delivery Lag`**
- **Assign block to increment `Inventory Level` by the `Order Quantity`**
- **Dispose block**

Animation

- **Plot with separate "in the black" and "in the red" curves**
  - If in backlog, red curve will be plotted in negative direction due to its Expression
- **Pair of Level ("thermometer") animations**
  - Fill Direction for "in the red" is Down



Modeling Detailed Operations

Chapter 5

Last revision August 20, 2006

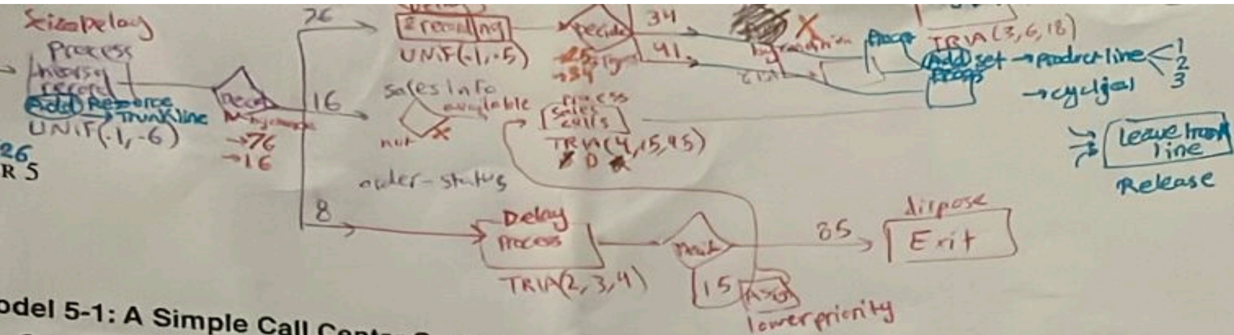
What We'll Do ...

- **Model 5-1: Simple call center**
  - Lower-level modeling, Advanced Process panel
  - Three-way decisions, Variables, Expressions, Storages
  - Blocks panel
  - Terminating vs. steady-state operation
  - Logical ("fake") entities
  - Terminating Condition in *Run* > *Setup*
- **Model 5-2: Enhanced call center**
  - Nonstationary Poisson arrival process
  - Sets – Resource, Counter
  - New Statistic data module Types
    - Counter, Time Persistent

What We'll Do ... (cont'd.)

- **Model 5-3: Enhanced call center with more output performance measures**
  - New Statistic data module Type
    - Output
  - Additional variable resources – look at staffing levels
- **Model 5-4: (*s*, *S*) inventory**
  - Not queueing
  - Choose to use low-level Blocks, Elements panels (SIMAN)
    - Can be done with higher-level panels

هاي الورقة أسهل لفهم ال call center



5.1 Model 5-1: A Simple Call Center System

Our generic call center system provides a central number in an organization that customers call for technical support, sales information, and order status. Incoming calls arrive with interarrival times being exponentially distributed with a mean of 0.857 minute. This central number feeds 26 trunk lines. If all 26 lines are in use, a caller gets a busy signal; hopefully, the caller will try again later, but for our model, he just goes away. An answered caller hears a recording describing three options: transfer to technical support, sales information, or order-status inquiry (76%, 16%, and 8%, respectively). The estimated time for this activity is UNIF(0.1, 0.6); all times are in minutes.

If the caller chooses technical support, a second recording requests which of three product types the caller is using, which requires UNIF(0.1, 0.5) minutes. The percentage of requests for product types 1, 2, and 3 are 25%, 34%, and 41%, respectively. If a qualified technical support person is available for the selected product type, the call is automatically routed to that person. If none are currently available, the customer is placed in an electronic queue where he is subjected to annoying rock music until a support person is available. The time for all technical support calls is estimated to be TRIA(3, 6, 18) minutes regardless of the product type. Upon completion of the call, the customer exits the system.

Sales calls are automatically routed to the sales staff. If a salesperson is not available, the caller is treated to soothing new-age space music (after all, we're hoping for a sale). Sales calls are estimated to be TRIA(4, 15, 45)—sales people tend to talk a lot more than technical support people! Upon completion of the call, the happy customer exits the system.

Callers requesting order-status information are automatically handled by the phone system, and there is no limit on the number the system can handle (except that there are only 26 trunk lines, which is itself a limit, since an ongoing order-status call occupies one of these lines). The estimated time for these transactions is TRIA(2, 3, 4) minutes, with 15% of these callers opting to speak to a real person after they have received their order status. These calls are routed to the sales staff, where they wait with a lower priority than sales calls. This means that if an order-status call is in a queue waiting for a salesperson and a new arriving sales call enters, the sales call will be given priority over the order-status call and answered first. These follow-up order-status calls are estimated to last TRIA(2, 3, 4) minutes. These callers then exit the system.

The call center hours are from 8 AM until 6 PM, with a small proportion of the staff on duty until 7 PM. Although the system closes to new calls at 6 PM, all calls that enter the system by that time are answered and served.

Over the course of a day there are eight technical support employees to answer technical support calls. Two are devoted to product Type 1 calls; three to product Type 2 calls; and three to product Type 3 calls. There are four sales employees to answer the sales calls and those order-status calls that opt to speak to a real person.

As a point of interest, we'll count the number of customer calls that are not able to get a trunk line and are thus rejected from entering the system (similar to balking in queueing systems, though balking usually means a decision on the customer's part

Definition → Resource → identify trunkline → Fixed capacity 26 mean of 20 could easily[1] return a value more than 31 for the time to the next arrival. This would result in no arrivals during the second period, when in fact there should be an expected value of 30 arrivals.[2] In general, using this simplistic method causes an incorrect decrease in the number of arrivals when going from one period to the next with an increase in the rate, or a decrease in the interarrival time. Going from one period to the next with a decrease in the rate will incorrectly increase the number of arrivals in the second period.

Nevertheless, it's important to be able to model and generate such arrival processes correctly since they seem to arise all the time, and ignoring the nonstationarity can create serious model-validity errors since the peaks and troughs can have significant impact on system performance. Fortunately, Arena has a built-in ability to generate nonstationary Poisson arrivals (and to do so correctly) in the Create module. The underlying method used is described in Section 12.3.

The second new concept is the need to model an entity arriving at a location or station and selecting from one of several similar (but not quite identical) objects. The most common situation is the selection of an available resource from a pool of resources. Let's assume you have three operators: Sean, Lynn, and Michael. Any one of these operators can perform the required task, and you would like to select any of the three, as long as one is currently available. The Sets data module in the Basic Process panel provides the basis for this functionality. Arena *sets* are groups of objects of the same type that can be referenced by a common name (the *set name*) and a *set index*. The objects that make up the set are referred to as *members* of the set. Members of a particular set must all be the same type of object, such as resources, queues, pictures, etc. You can collect almost any type of Arena objects into a set, depending on your modeling requirements. An object can also reside in more than one set. Let's assume in our `Operators` set that Lynn is also qualified as a setup person. Therefore, we might define a second resource set called `Setup` as Lynn and Doug (Doug's not an operator). Now, if an operator is required, we'd select from the set called `Operators`; if a setup person is required, we would select from the set called `Setup`. Lynn might be chosen via either case because she's a member of both sets. You can have as many sets as you want with as much or as little overlap as required.

For our call center, we'll need to use sets to model the technical support staff correctly. We also need to consider how to model the returned technical support calls. These are unique in that they must be returned by the same tech-support person who handled the original call, so we must have a way to track who handled the original

[1] With probability $e^{-31/20} = 0.21$, to be (almost) exact. Actually this figure is the *conditional* probability of no arrivals in the second period, *given* that there were arrivals in the first period and that the last of these was at time 29. This is not quite what we want, though; we want the *unconditional* probability of seeing no arrivals in the second period. It's possible to work this out, but it's complicated. However, it's easy to see that a lower bound on this probability is given by the probability that the first arrival after time 0, generated as exponential with mean 20 minutes, occurs after time 60—this is one way (not the only way) to have no arrivals in the second period, and has probability $e^{-60/20} = e^{-3} = 0.0498$. Thus, the incorrect method would give us at least a 5% chance of having no arrivals in the second period. Now, go back to the text, read the next sentence, and see the next footnote.

[2] The probability of no arrivals in the second period should be $e^{-60(1/2)} = 0.000000000000093576$.

5-4
مال Decide
واحد اقل من ٢
وواحد اقل من ٣

Table 5-2. Call Arrival Rates (Calls Per Hour)

Time	Rate	Time	Rate	Time	Rate	Time	Rate
8:00–8:30	20	10:30–11:00	75	1:00–1:30	110	3:30–4:00	90
8:30–9:00	35	11:00–11:30	75	1:30–2:00	95	4:00–4:30	70
9:00–9:30	45	11:30–12:00	90	2:00–2:30	105	4:30–5:00	65
9:30–10:00	50	12:00–12:30	95	2:30–3:00	90	5:00–5:30	45
10:00–10:30	70	12:30–1:00	105	3:00–3:30	85	5:30–6:00	30

Next let's take a look at our staffing. Although our initial model assumed constant staffing levels for our sales and technical-support areas, the staffing level actually varies over the day. It turns out that there are six salespeople with the staggered daily schedules summarized as (number of people @ time period in minutes): 1@60, 3@60, 4@90, 5@60, 6@60, 5@90, 6@90, 5@30, 3@60, and 2@60.

Our technical support employees work an 8-hour day with 30 minutes off for lunch (lunch is not included in the 8 hours). There are 11 technical support people whose work schedules are shown in Table 5-3. Charity and Noah are only qualified to handle calls for product Type 1; Tierney, Aidan, and Emma are only qualified to handle calls for product Type 2; Mya, Ian, and Christie are only qualified to handle calls for product Type 3. Molly is qualified to handle product Types 1 and 3, and Anna and Sammy are qualified to handle calls for all three product types.

The last bit of detail that we omitted from our initial model was that 4% of technical calls require further investigation after completion of the phone call. The questions raised by these callers are forwarded to another technical group, outside the boundaries

→ Arena Sets

1Γ → عدد ال Resources بیاد

mΓ → عدد ال Re

Table 5-3. Technical Support Schedules

Name	Product Lines	Time Period (30 minutes)																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Charity	1	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•					
Noah	1						•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•
Molly	1, 3			•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•			
Anna	1, 2, 3					•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	
Sammy	1, 2, 3				•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•		
Tierney	2	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•					
Aidan	2						•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•
Emma	2				•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•		
Mya	3	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•					
Ian	3						•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•
Christie	3				•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•		

Model 5-1: Simple Call Center Setup

- **One phone number for customers to call in to**

  Resource ← trunk line راح نعتبر ال

  - 26 trunk lines, one needed for each call (incoming or outgoing, talking or on hold)
  - Arriving call finding no free trunk lines gets busy signal, goes away
    - Count number of such rejected calls
- **Calls arrive with interarrivals ~ EXPO (0.857) min.**
  - First call arrives at time 0
- **Three incoming call types**
  - Initial recording to decide ~ UNIF (0.1, 0.6) min.
  - Tech support (76%), sales (16%), order status (8%)

Model 5-1: Simple Call Center Setup (cont'd.)

- **Tech-support calls**
  - For product type 1 (25%), 2 (34%), or 3 (41%)
  - Needs qualified tech-support person
    - Two for type 1, three for type 2, three for type 3
      No crossover to another type ... change in Model 5-2
    - Separate FIFO queues for each type
    - Conversation time ~ TRIA (3, 6, 18) min. for all types
  - Then leaves system
- **Sales calls**
  - All the same
  - Four sales staff, all the same
    - One FIFO queue feeding all sales staff
  - Conversation time ~ TRIA (4, 15, 45)
  - Then leaves system

Model 5-1: Simple Call Center Setup (cont'd.)

- **Order-status calls**
  - All the same
  - Handled automatically by phone system
    - No limit on number in process at a time, except for trunk-line limit
  - "Conversation" time ~ TRIA (2, 3, 4)
  - After "conversation," 15% of callers opt to talk to a person
    - Routed to sales staff
    - Sales calls have higher priority (non-preemptive)
- **Center receives calls 8am – 6pm**
  - Must terminate arrival process at 6pm
  - Operate past 6pm if necessary to "flush out" all calls

Model 5-1: Simple Call Center Setup (cont'd.)

- **Output performance measures**
  - Number of calls attempted, rejected, completed
  - By call type – total time in system
  - By resource – time on hold, number of calls on hold
  - Resource utilization – personnel, trunk lines
- **Terminating or steady-state**
  - Time frame of interest for each replication
    - Terminating – specific starting, stopping conditions (this model)
      - Stopping conditions could be of several forms – fixed time, count, condition (here)
    - Steady-state – output performance measures are a limit as simulated time $\rightarrow \infty$
    - Choice usually depends on intent of study, not on model logic

Model 5-1: Simple Call Center Modeling Panels

- **Basic Process**
  - Highest, fastest modeling level, usually the place to start
- **Advanced Process**
  - Smaller building elements, other functions, more detail
- **Advanced Transfer**
  - Entity movement, material handling
- **Blocks, Elements**
  - Lowest modeling level, SIMAN simulation *language*
  - Repeats some capabilities of higher-level panels
  - Some functions available only here
- **Other special-purpose panels**
  - License-dependent

Model 5-1: Simple Call Center Data Structure

Technical staff
Sales staff
order staff
Lis

- **Re-use data in several places**
  - Define once, global to whole model
  - Redefine once – modeling generality, user efficiency
- **Arena *Variables***
  - Store numbers (not formulas)
  - Define, initialize in Variable data module (Basic Process)
  - Can change during run (Assign module, other ways)
  - Scalar, 1-d array (vector), 2-d array (matrix)
- **Arena *Expressions* – generalize Variables**
  - Store formulas (as well as numbers)
    - Use math ops, numbers, random variates, Attributes, Variables, ...
  - Define in Expression data module (Advanced Process)
  - Scalar, 1-d array (vector), 2-d array (matrix)

Model 5-1: Simple Call Center Arrivals, Direct to Service

*لما يدخل واحد جديد
number in system = number in system + 1

*لما بده يطلع واحد من ال system
number in system = number in system − 1

- **Create attempted calls**
  - Entity type **`Incoming Call`**, change later
  - Max Arrivals = **`MaxCalls`**, Variable initialized to 999999
    - At 6pm (time 600 minutes) change this to 1 to cut off arrivals ... later
- **Entity data module**
  - **`Incoming Call`** Entity Type already there
  - For Initial Picture, select **`Picture.Black Ball`**

N(Q) → تقيس ال Current Number in Que

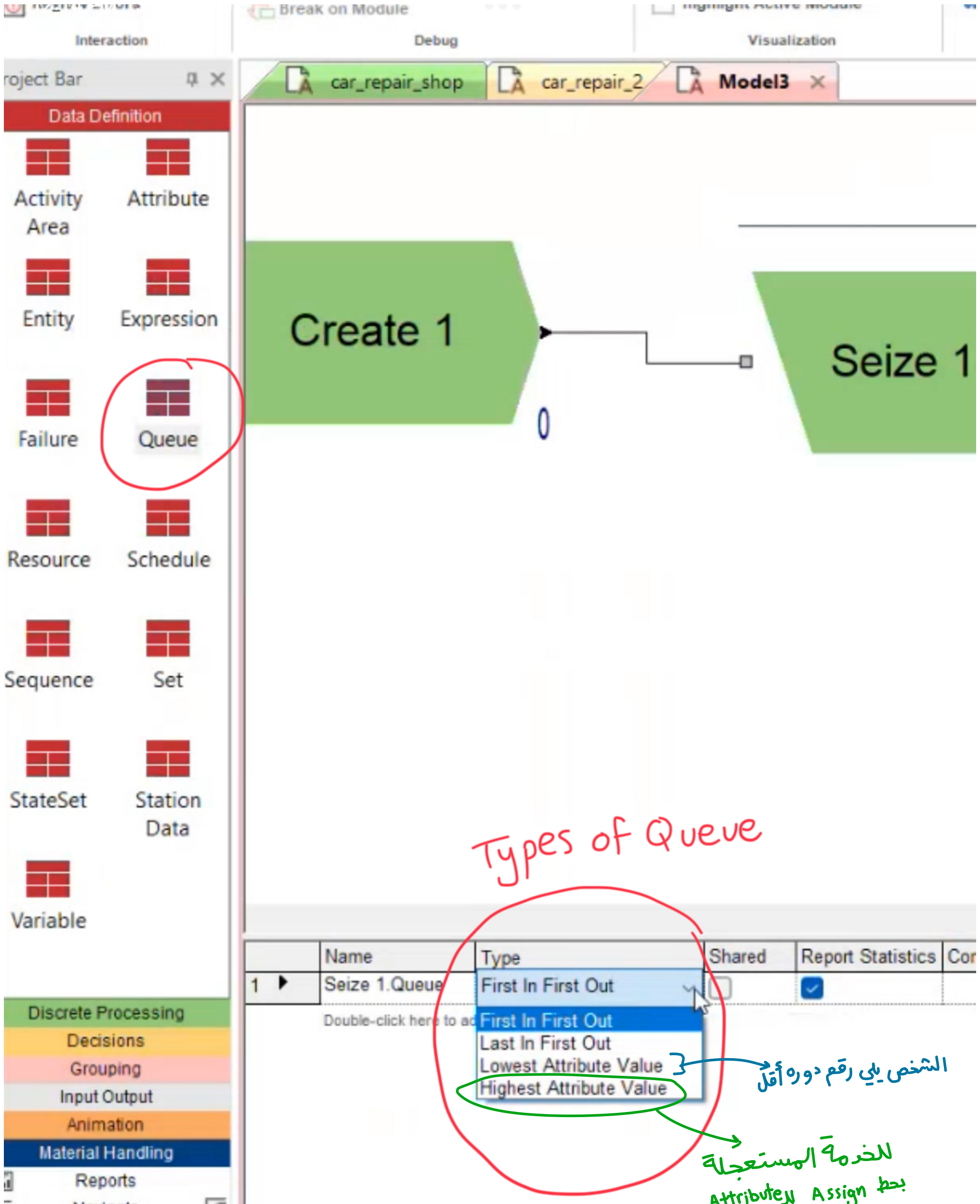
- **Record module for an attempted call**
  - Add 1 to Counter Name **`Attempted Calls`**
  - Results – Category Overview report, User Specified

More detailed description – mouse over modules, read Data Tips that pop up

Model 5-1: Simple Call Center Arrivals, Direct to Service (cont'd.)

- **Decide module – Trunk Line Available?**
  - Type = 2-way by Condition
    - Select (logical) Expression for "If"
      - NR() is number of units of that resource that are busy now
      - MR() is number of units of that resource that exist now
  - False – Record rejected call counter, Dispose
  - True:
    - Seize a unit of **`Trunk Line`** Resource – Release later
      - Resources data module for Trunk Line and other Resource levels
    - Increment Variable **`Total WIP`** for number of active calls
      - Used in stopping rule at or after 6pm to sense if system is empty
    - Store module to enable entity animation during next Delay module
      - Add Storage animation separately, identify with this logical storage by name
      - Storage data module – entry made there by Store module
    - Delay module to listen to initial recording, make selection
      - Could have used Process module, but this is simpler, faster
    - Unstore module to make entity animation disappear

Alternate strategy – Queue module from Blocks panel ... details in text



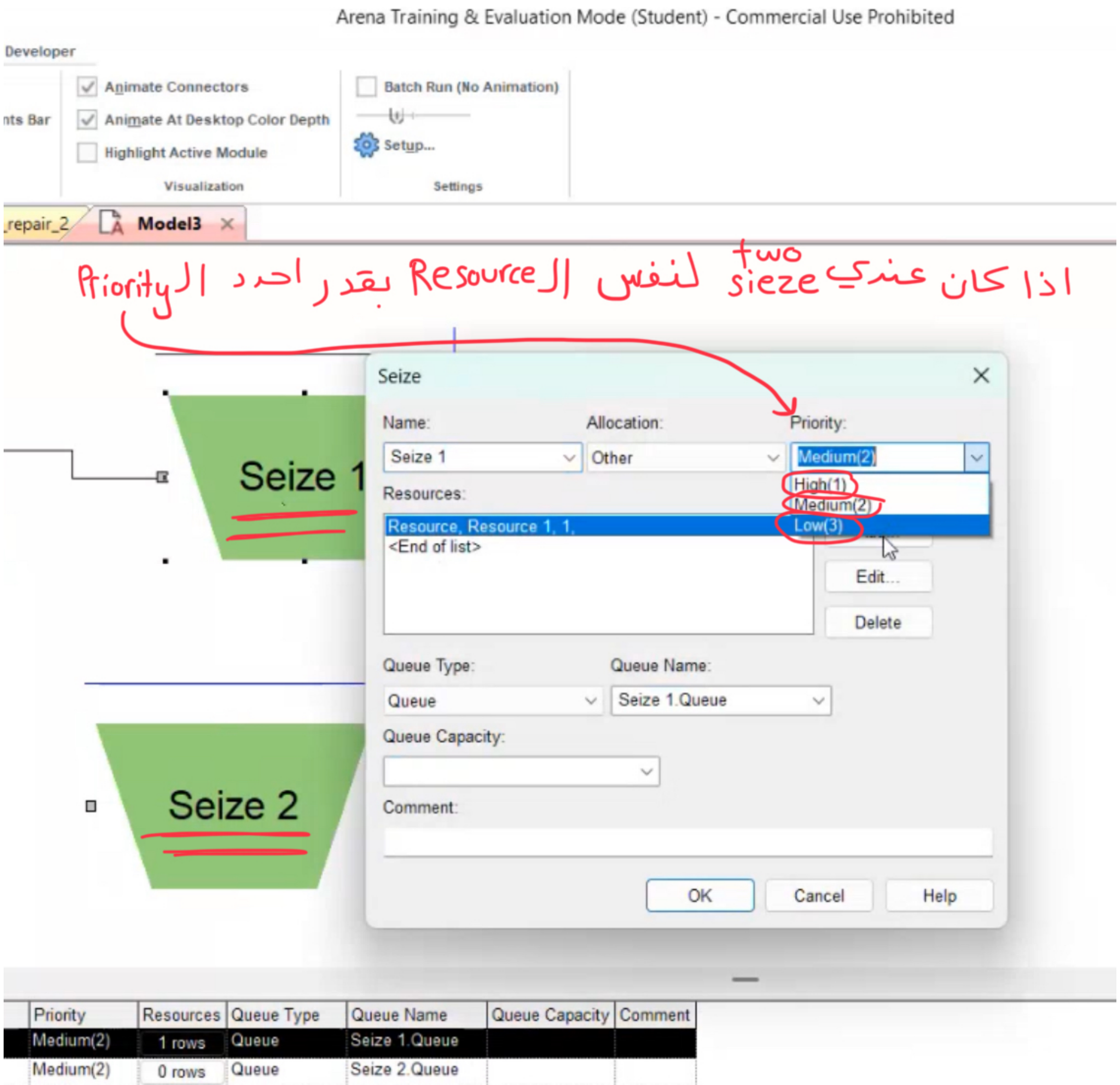
Types of Queue

الشخص يلي رقم دوره أقل

للخدمة المستعجلة

بحط Assign الـAttribute ← High

اذا كان عندك two sieze لنفس ال Resource بقدر احدد ال Priority



Model 5-1: Simple Call Center Arrivals, Direct to Service (cont'd.)

- **Decide module – Determine Call Type**
  - Three-sided coin flip – Type = N-way by Chance
    - Add button for more sides of coin
    - Get new exit point for each Add, plus one for Else
    - Note that probabilities are entered as *percentages* (0-100, not 0-1)
    - Last entry is "else"
- **Direct call to one of tech support, sales, or order-status areas**

Backed each area with colored box
Alternative way to organize – Submodels

Model 5-1: Simple Call Center Tech-Support Calls

- **Assign module**
  - Change Entity Type for separating out in results
  - Change Entity Picture for animation
- **Store – Delay – Unstore for recording, product type selection**
- **Decide module for product type**
  - Different three-sided coin flip
  - Direct to appropriate Process module for that product type
- **Process modules for tech-support service**
  - Seize-Delay-Release
  - Seize a unit from appropriate multi-unit Resource
  - Use `Tech Time` defined in Expression data module
- **Proceed to system exit logic ... later**

Model 5-1: Simple Call Center Sales Calls

- **Assign module – change Entity Type, Picture**
- **Process module**
  - Seize-Delay-Release
  - Seize a unit of **`Sales`** Resource
- **Sales calls priority over order-status calls that seek a person?**
  - Queue data module, **`Process Sales Call.Queue`**
    - Type = Lowest Attribute Value

      Attribute Name = **`Sales Call Priority`**

      Undefined for sales calls, so has value 0 ... will set to 1 for order-status calls that seek a person, putting sales calls ahead in the queue
    - Shared queue (with order-status calls seeking a person)
- **Proceed to system-exit logic**

Not the only way to do this

Model 5-1: Simple Call Center Order-Status Calls

- **Assign module – change Entity Type, Picture**
- **Delay block (Blocks panel) for robo-chat**
  - Includes Store/Unstore logic – alternative to earlier method
    - No automatic entry in Storage data module, so must enter manually
- **Decide module**
  - No sales person required – go directly to system-exit logic
  - Sales person required:
    - Assign module – set `Sales Call Priority` Attribute to 1 so these will have lower priority than real sales calls
    - Seize module for a unit of `Sales` resource
      - Define Queue Name = `Process Sales Call.Queue` – shared with sales calls
      - Process module does not allow for specifying a shared queue, so can't use here
    - Delay for conversation with sales person
    - Release the unit of `Sales` resource
- **Proceed to system-exit logic**

Model 5-1: Simple Call Center System Exit

- **All calls of all types come here when finished**
- **Release module – release the unit of `Trunk Line` resource seized upstream**
- **Assign module – decrement `Total WIP` variable**
- **Record module – increment `Completed Calls` counter**
- **Dispose of call**

Model 5-1: Simple Call Center Arrival-Cutoff Logic

- **Used to "choke off" arrival stream at 6pm**
- **Create a single "logical" entity at time 600 min. (6pm)**
  - Overkill on making sure just one is created
    - Time Between Arrivals = 999999 min., Max Arrivals = 1
- **Assign module to set Variable `MaxCalls` to 1**
  - Recall use of `MaxCalls` for Max Arrivals in Create module for attempted calls
- **Dispose of this single logical entity**

Creative use of such "logical" (a.k.a. "fake") entities enhances modeling flexibility, power, detail

Model 5-1: Simple Call Center
Run > Setup

- **Replication Parameters tab (other tabs as usual)**
- **Base Time Units = Minutes**
- **Replication Length = Infinite (the default)**
- **Terminating Condition field:**

```
TNOW >= 600.0 && Total WIP == 0
```

TNOW	>= 600.0	&&	Total WIP	== 0
Arena clock Variable	*Greater than or equal to 600 minutes, (6pm)*	*Logical "and"*	*Variable we maintained in model*	*Equality test for zero*
	Base Time Units			
It's 6pm or later		*and*	*there are no calls in the system.*	

Could have used **NR(Trunk Line)** *instead of* **Total WIP**

Model 5-1: Simple Call Center Animation

- **Place three Storage animations**
  - Initial Recording Delay, Tech Call Recording Delay, Order Status Delay
  - Select proper Identifier in each from pull-down list
  - Graphic behaves like Queue animations
- **Four Queue animations**
  - Three tech-support call product types, sales
  - Came with four Process modules specifying Seize
- **Resource animations for three tech-support types, sales Resources**
  - Multi-unit Resource animations, as in Models 4-3, 4-4

Model 5-1: Simple Call Center Animation (cont'd.)

- **Variable animations for WIP at tech calls, sales**
  - For tech calls, Arena variable to animate is **`Process Product Type 1 Tech Call.WIP`**, etc. – pull-down list
  - For sales calls, must include order-status calls seeking a real person:

    `NR(Sales) + NQ(Process Sales Call.Queue)`
- **Plot number of trunk lines busy, `NR(Trunk Line)`**
- **Labeling, background boxes as in model logic**

Model 5-1: Simple Call Center Results *(one replication ... sample of size only one!!)*

- **Trunk-lines-busy plot**
  - Starts, ends at 0 – startup, termination logic
  - Capped at 26 during run
- **735 attempted calls (User Specified section)**
  - 644 completed, other 91 rejected
- **Sometimes see mixture of sales (green), order-status (blue) entities in sales queue**
- **Other "usual" outputs**
  - Times in system – separated out by call type
  - Queue lengths, times in queue – separated out by resource
  - Resource utilizations – normalized to [0, 1] by capacity

Model 5-2: Enhanced Call Center Changes

- **Incoming calls' arrival rate varies over day**
  - Probabilistic model – *Nonstationary Poisson process*
    - More in Section 12.3
  - Instead of a constant rate (= 1 / mean interarrival time), specify a rate *function*
    - Arena supports *piecewise-constant* rate function – "step" functions
      - Easy to specify, strong theoretical support
    - Rate-function specification:

In Arena, rates MUST be in arrivals per HOUR, regardless of base time units or time intervals

Caution – it's easy to generate this incorrectly ... see text for details

it is an hourly based

Table 5-2. Call Arrival Rates (Calls Per Hour)

Time	Rate		Time	Rate	Time	Rate	Time	Rate
8:00 - 8:30	20	40	10:30 - 11:00	75	1:00 - 1:30	110	3:30 - 4:00	90
8:30 - 9:00	35	70	11:00 - 11:30	75	1:30 - 2:00	95	4:00 - 4:30	70
9:00 - 9:30	45		11:30 - 12:00	90	2:00 - 2:30	105	4:30 - 5:00	65
9:30 - 10:00	50		12:00 - 12:30	95	2:30 - 3:00	90	5:00 - 5:30	45
10:00 - 10:30	70		12:30 - 1:00	105	3:00 - 3:30	85	5:30 - 6:00	30

Model 5-2: Enhanced Call Center Changes (cont'd.)

- **Sales-staff size varies over day**
  - Data in text, Schedule data module, `Sales Schedule`
- **Tech-support staff are partially cross-trained, work complicated schedule:**

Table 5-3. Technical Support Schedules

Name	Product Lines	Time Period (30 minutes)																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Charity	1	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●					
Noah	1						●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●
Molly	1, 3			●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●			
Anna	1, 2, 3					●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	
Sammy	1, 2, 3				●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●		
Tierney	2	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●					
Aidan	2						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●
Emma	2				●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●		
Shelley	3	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●					
Jenny	3						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●
Christie	3				●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●		

Will use Arena Sets concept to implement this cross training

Model 5-2: Enhanced Call Center Changes (cont'd.)

- **4% of tech-support calls cannot be handled during the call, need offline back-office research**
  - Original call ends, same original talk-time distribution, gives up its trunk line, but not counted (yet) as completed
  - Case sent to back office (outside model boundaries), takes EXPO (60) minutes to resolve
    - Offline research may be carried over night, completed on a later day
  - Answer goes back to *same* tech-support person who took original call, with higher priority than incoming calls, but still might have to queue for this person
  - This tech-support person requests a trunk line for outgoing call, higher priority than incoming calls, but still might have to queue, talks for TRIA (2 ,4 ,9) min., call is *now* completed
  - Track number of each product type after research is done

Model 5-2: Enhanced Call Center Data Structure

- **Resources, Schedules**
  - Resource, Schedule data modules
  - Trunk Line – fixed capacity at 26
  - Sales – on Schedule **`Sales Schedule`**
  - 11 individual tech-support people on individual schedules
    - Caution – must fill out each schedule to all 22 half-hour periods, with leading/trailing 0's if necessary ... use Edit via Dialog or Spreadsheet, not graphical schedule editor
    - **`Ignore`** option to avoid shifting back schedule over multiple days
    - Include costing data for people in Resource data module
  - Define nonstationary arrival-rate function in Schedule module – **`Arrival Schedule`**
    - Enter trailing 0's in Edit via Dialog or Spreadsheet, not graphical schedule editor

Model 5-2: Enhanced Call Center Data Structure (cont'd.)

- **Sets – collect same-type items together**
  - Set, Advanced Set data modules (Basic, Advanced Process panels, resp.)
    - Refer to items in set by original name, or index (subscript) in set
  - Resource set for each tech-support product type
    - Members are those tech-support resources qualified
    - Individual resources already defined – Resource data module
    - Overlapping membership – some resources in multiple sets
    - Sets are ordered – here, put most versatile tech-support people at bottom, to "save" them for other calls ... Preferred Order in Seize
    - Will Seize from a set in model
  - Counter set – one for each hour
    - Count number of rejected calls in each hour
    - Individual counters already defined – Statistic data module
    - Use results later to decide when to increase staffing

Model 5-2: Enhanced Call Center Modifying the Model

- **Call-arrivals, termination, Run > Setup**
  - Create module
    - Type = Schedule, Schedule Name = **`Arrival Schedule`**
  - Delete the entire arrival-cutoff section from Model 5-1
    - **`Arrival Schedule`** cuts off arrivals at 6pm, via 0 rate
  - Delete **`Total WIP`** variable used to terminate Model 5-1
    - Use built-in NR(**`Trunk Line`**) instead in Terminating Condition
    - Delete Assign modules used to manage **`Total WIP`**
  - Record module for rejected calls
    - Index into Counter Set **`Rejected Calls`** with index

      `AINT((TNOW/60) + 1)`

      which is 1 for first hour, 2 for second hour, etc. (AINT truncates decimals toward zero)

Model 5-2: Enhanced Call Center Modifying the Model (cont'd.)

- **Tech-support calls**
  - Same through `Determine Product Type` Decide
  - Add Assign modules for each product type thereafter
    - Entity Type to distinguish product type in reports
    - Entity Picture to distinguish product type in animation
    - Attribute `Tech Call Type` (1, 2, or 3 by product type) for routing
  - Process modules, Resources subdialogs
    - Type = Set
    - Set Name = Product 1, etc.
    - Selection Rule = Preferred Order, to select earlier entries in set first
      Recall – we put more versatile tech-support people lower in the set list
    - Save Attribute = `Tech Agent Index`
      Entity attribute, carried along, in case of back-office research to send back to this same tech-support person for return call

Model 5-2: Enhanced Call Center Modifying the Model (cont'd.)

- **Back office, returned tech-support calls – all new**
  - Entry via True branch (4%) in Decide module **`Backoffice Research and Return Call?`**
  - Release this call's trunk line – going offline now
  - Delay (with storage) for EXPO (60) back-office research
  - Increment **`Tech Return WIP(Tech Call Type)`**
    - 1-dim. Variable array – defined in Variable data module
    - **`Tech Call Type`** is 1, 2, or 3, assigned in earlier Assign module
  - Decide module **`Product Type?`** based on Entity Type
  - Seize the same tech-support person – higher priority
  - *Then* seize a trunk line (higher priority), make return call
  - Then release this trunk line, tech-support person
  - Decrement **`Tech Return WIP(Tech Call Type)`**
  - Send entity to final Record, *after* trunk-line release there

Model 5-2: Enhanced Call Center Modifying the Model (cont'd.)

- **Statistic data module**
  - Ten Counter-type statistics, discussed earlier
  - Four Time-Persistent statistics to track expressions
    - **Backoffice Research WIP** to track total number of cases in research, via NSTO(**Backoffice Research Storage**)
    - **Tech 1 Total Online WIP Stat**, etc., to track number of that product type in back office via Expression **Tech 1 Total Online WIP**, etc., defined in Expression data module as

Process Product Type 1 Tech Call.WIP + Tech Return WIP(1), etc.

- **No changes needed in sales-calls or order-status-calls section of Model 5-1**

Model 5-2: Enhanced Call Center Modifying the Model (cont'd.)

- **Animation**
  - Delete `Tech 1`, `Tech 2`, and `Tech 3` resource animations
  - Change variables in three tech-support WIP displays to track total number of tech-support calls of that type present
  - New back-office storage animation, variable animation for number present
  - A new queue for each tech-support product type for return calls waiting for service
  - Added a resource animation (from a .plb library) for each individual tech-support person
    - Grouped by product type, colors for capabilities
- **Results**
  - Most rejected calls in hours 5-8 ... increase staff then?

Model 5-3: Overall Call-Center Stats Setup

- **Develop an overall operational-cost measure**
  - Two cost categories – staffing/resource, and poor service
- **Also develop overall measure of service, % of calls rejected**
- **Also add options for increased staffing, improvement**
- **Make 5 replications, focus on weekly costs**
  - IID replications, so will not carry over back-office research

Model 5-3: Overall Call-Center Stats Staffing/Resource Costs

- **Resource data module – hourly costs for people**
  - $20/hr. for each sales staffer
  - $18/hr. - $20/hr. for each tech-support, depending on skill
  - These salary costs paid when on duty, busy or idle
  - Summing, get $12,820/week (details in text)
  - View all this existing staff as fixed

Model 5-3: Overall Call-Center Stats Staffing/Resource Costs (cont'd.)

- **Increase sales, tech-support staff noon-4pm**
  - Variable **`New Sales`** = number of new sales staff
    - $17/hr., 4 hrs./day, 5 days/week, so $340/week for each add'l. staff
    - Schedule data module to add capacity – edit via dialog or spreadsheet, not graphical editor
    - Resource (**`Sales`**) already exists in Resource data module
  - Variables **`New Tech 1`**, etc., and **`New Tech All`** for number of new tech-support people qualified as named
    - $16/hr. for each one-product staff, $18/hr. for each all-product staff
      - $320/week for each single-product staff, $360/week for each all-product staff
    - New entries in Resource data module
      - Larry, Moe, Curly, Hermann for 1, 2, 3, All, resp.
    - Schedule data module to add capacity – dialog or spreadsheet edit

Model 5-3: Overall Call-Center Stats Staffing/Resource Costs (cont'd.)

- **Maybe increase number of trunk lines beyond 26**
  - $98/week flat fee for each trunk line
- **Define Expression `New Res Cost` for all resource costs:**

```
New Sales*340
+ (New Tech 1 + New Tech 2 + New Tech 3)*320
+ New Tech All*360
+ 98*MR(Trunk Line)
```

  - This does not depend on simulation results, only on setup

Model 5-3: Overall Call-Center Stats Customer-Dissatisfaction Costs

- **Incur cost for caller wait on hold, past a threshold**
  - 3 min. for tech, 1 min. for sales, 2 min. for order-status
  - Beyond threshold, incur per-min. costs of $0.368 for tech, $0.818 for sales, $0.346 for order-status
    - In practice, such costs are difficult to estimate
  - Three new Assign modules (orange backing) accumulate "excess" (beyond threshold) wait times on hold
    - Tech support (other two are similar): Variable **`Excess Tech Wait Time`** increased by **`MAX(ENTITY.WAITTIME - 3, 0)`**

      ENTITY.WAITTIME is built-in Arena attribute holding all wait times (including in queues) so far ... luckily, there were none before the preceding Process module
  - At end, multiply excess wait times by per-min. costs, multiplied by 5 (to put on a weekly basis)

    5 × \$0.368 = \$1.84 for tech, 5 × \$0.818 = \$4.09 for sales,
    5 × \$0.346 = \$1.73 for order-status

Model 5-3: Overall Call-Center Stats Overall Output Performance Measures

- **Statistic data module, `Total Cost` entry**
  - Type = Output, computed only at end of replication

```
New Res Cost
+ Excess Sales Wait Time  * 4.09
+ Excess Status Wait Time * 1.73
+ Excess Tech Wait Time   * 1.84
+ 12820
```

- **Statistic data module, `Percent Rejected` entry**
  - Counter `Total Rejected Calls` accumulated in new Record module in call-arrival area (orange backing)
    - Already accumulating hour by hour, but this is total over the day
  - Type = Output

```
100 * NC(Total Rejected Calls) / NC(Attempted Calls)
```

    - NC is Arena function that returns the value of that counter

Model 5-3: Overall Call-Center Stats Replication Conditions

- **Run > Setup > Replication Parameters, Initialize Between Replications**
  - Statistics? System? Details in text
  - Default is both – only way to get truly IID replications
    - Destroys overnight tech-support research jobs, but to do otherwise would complicate model – so accept
- **Run > Setup > Project Parameters**
  - Turned off all but Costing Statistics Collection, for speed
    - Costing required to get ENTITY.WAITTIME

Model 5-3: Overall Call-Center Stats Results

- **Results from five replications**
  - Base Case – no additional staff, still 26 trunk lines

    `Total Cost` = $22,500.07

    `Percent Rejected` = 12.9%

    Average over 5 replications
    Conf. int. half-widths in output

  - Add 3 of each of five staff types, 3 more trunk lines

    `Total Cost` = $22,668.69

    `Percent Rejected` = 1.6%

    Is this better?

- **Use in Chapt. 6 for statistically valid experiments**
  - Statistical precision
  - Compare several alternatives, select best
  - Search for configuration that minimizes cost, subject to upper limit on percent rejected

Model 5-4: (*s*, *S*) Inventory Simulation Setup

- **Different kind of model – not queueing**
- **Use Blocks and Elements panels exclusively – SIMAN simulation language**
  - Mostly just to demonstrate this capability
  - Could be done with higher-level panels we've been using
- **Company carries a single discrete item (widgets) in inventory**
- **$I(t)$ = inventory level (an integer) at time t days past the beginning of the simulation; $I(0) = 60$**
- **Run simulation for 120 round-the-clock days**

Model 5-4: (*s*, *S*) Inventory Simulation Customer Demands Against Inventory

- **Customer interarrival times ~ EXPO (0.1) day (round the clock)**
  - First arrival not at time 0 but after an interarrival time past 0
- **Demand size is discrete random variable**
  - 1, 2, 3, 4 with respective probabilities 0.167, 0.333, 0.333, 0.167
- **If enough items are physically on hand in inventory to satisfy a demand, customer gets demand and leaves**
- **If demand > number of items on hand, customer gets whatever is there and the rest of the demand is backlogged ($I(t)$ becomes negative)**
  - If $I(t)$ was already negative, it just goes more negative

Model 5-4: (*s*, *S*) Inventory Simulation Inventory Review, Replenishment

- **"Take inventory" just past midnight each day**
  - So at exactly times 0, 1, 2, ..., 119 (not 120 ... see below)
  - Two managerially-chosen constant integers s = 20 and S = 40 (must have $s < S$ if we change these values)
  - If $\mathtt{I}(t) \geq s$, do nothing until next inventory evaluation exactly 24 hours later
  - If $\mathtt{I}(t) < s$, order $S - \mathtt{I}(t)$ items from supplier (order "up to" S)
  - Order does not arrive instantly from supplier, but after a *delivery lag* (a.k.a. *lead time*) ~ UNIF(0.5, 1.0) day, so sometime during the last half of the day of ordering
    - In the meantime, inventory level could fall further from additional demands, so inventory level will not necessarily pop up to S when the order arrives, but to something less than S

Model 5-4: (*s*, *S*) Inventory Simulation Cost Structure

- **Average ordering cost per day**
  - When an order is placed, incur a fixed cost of \$32, plus an incremental cost of \$3 per item ordered
  - If no order is placed at the beginning of a day, there's no ordering cost, not even the fixed cost
  - At end of simulation, divide total of ordering costs by 120
- **Average holding cost per day**
  - Whenever $I(t) > 0$, incur \$1 per day per item on hand
  - Average holding cost = $\int_0^{120} 1 \times \max(I(t), 0)\, dt \Big/ 120$
- **Average shortage cost per day**
  - Whenever $I(t) < 0$, incur \$5 per day per item in backlog
  - Average shortage cost = $\int_0^{120} 5 \times \max(-I(t), 0)\, dt \Big/ 120$

Model 5-4: (*s*, *S*) Inventory Simulation Cost Structure (cont'd.)

- **During periods when $I(t) = 0$ there's neither holding nor shortage cost incurred**
- **Overall performance measure**
  = Average *total* cost per day
  = sum of average ordering, holding, and shortage costs per day
- **Don't evaluate inventory at time 120**
  - We might order and incur an ordering cost then, but order will never arrive
  - We'll fudge this, but an Exercise asks you to do it right

Model 5-4: (*s*, *S*) Inventory Simulation Data Structure

- **Use Blocks, Elements panels exclusively**
  - Even for Variables, Expressions, Attributes, Entities, statistics collection, and run control
- **Variables Element (initialized, or default to 0 initially)**
  - `Inventory Level` = $I(t)$, changes during run, initialized to 60
  - `Little s` = s = 20
  - `Big S` = S = 40
  - `Total Ordering Cost` accumulator
  - `Setup Cost` = 32
  - `Incremental Cost` = 3
  - `Unit Holding Cost` = 1
  - `Unit Shortage Cost` = 5
  - `Days to Run` = 119.9999 (The Fudge)

Model 5-4: (*s*, *S*) Inventory Simulation Data Structure (cont'd.)

- **Expressions element**
  - Define `Interdemand Time`, `Demand Size`, `Evaluation Interval`, `Delivery Lag`
    - ***Cumulative* probabilities in DISC function for `Demand Size`**
- **Attributes, Entities elements**
  - Just to define these objects
- **Project, Replicate elements**
  - Similar to *Run > Setup*
- **DStats element**
  - Request accumulation of integrals for total holding, shortage costs

Model 5-4: (*s*, *S*) Inventory Simulation Data Structure (cont'd.)

- **Outputs element**
  - Two entries, both of Data type "Output" so that they're executed only at end of run, and reported
  - Avg Ordering Cost computed
  - Avg Total Cost added up
    - OVALUE returns most recent value
    - DAVG returns time-persistent average

Model 5-4: (*s*, *S*) Inventory Simulation Logic for Customer Demands

- **Create block for arrival**
  - Entity Type is `Customer`
  - Uses `Interdemand Time` Expression
  - First Creation after an `Interdemand Time`
- **Assign block to decrement `Inventory Level` by a `Demand Size`**
  - `Demand Size` was defined as an Expression
  - Backlogging naturally happens
- **Dispose block for customer exit**
  - If backlogged, is accounted for automatically in the (simple) definition and tracking of `Inventory Level`

Model 5-4: (*s*, *S*) Inventory Simulation Inventory Evaluation

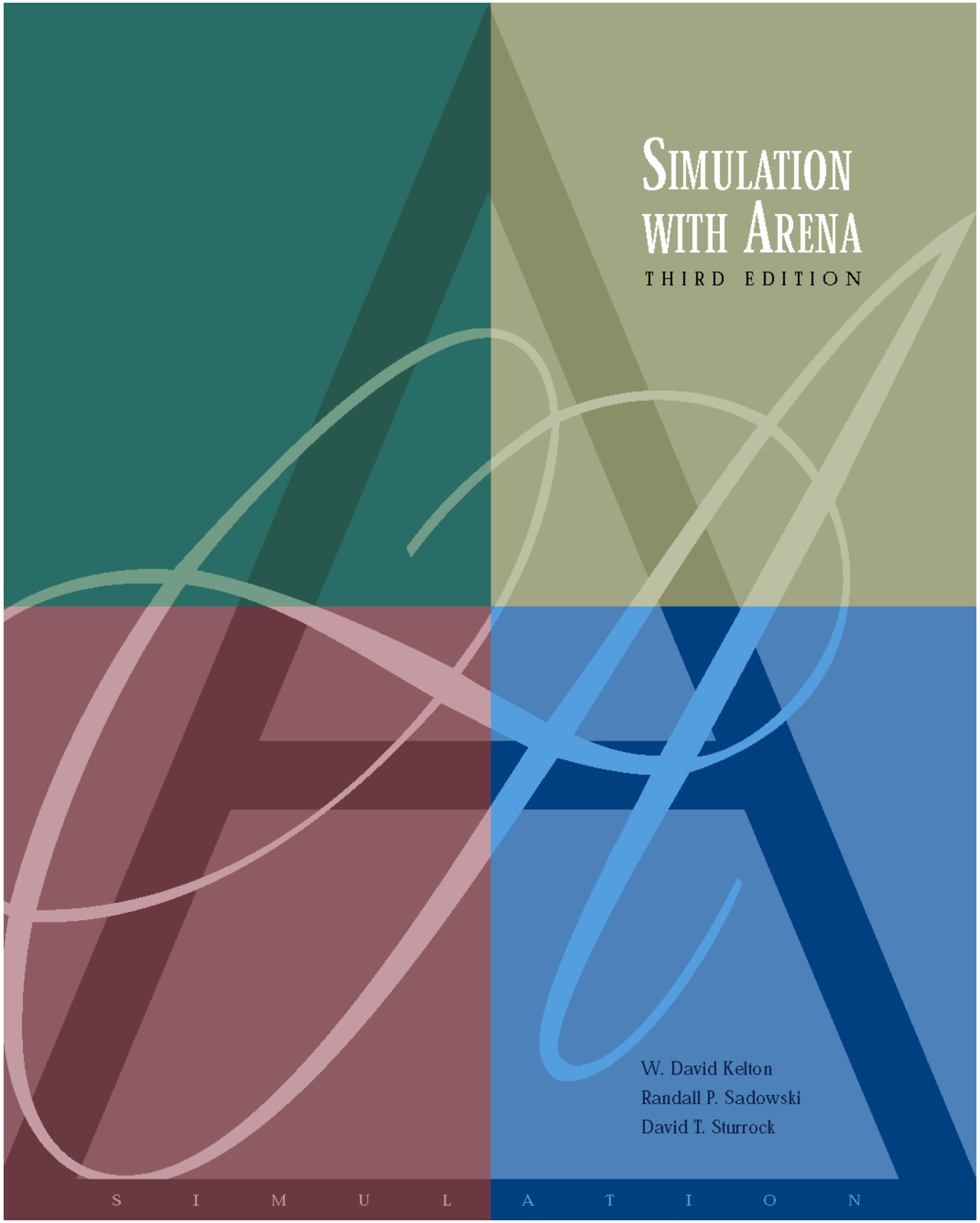
- **Create block for `Inventory Evaluator` entities**
  - First Creation at time 0 – evaluate inventory at start of run
  - Interval is **`Evaluation Interval`**, defined as Expression
- **Branch block – somewhat like Decide module**
  - To determine whether to place an order now
  - Add "branches," each evaluated as true or false
  - Clone of incoming entity sent out along each "true" branch, but at most Max Number of Branches will be sent out
    - So we set Max Number of Branches to `1` (default is ∞)
  - First branch of type "If" – if "true" we want to order
  - Second branch of type "Else" – if "true" it means that the first branch was "false" so we don't order – just Dispose

Model 5-4: (*s*, *S*) Inventory Simulation Placing an Order

- **If we exit the Branch block via the top "If" branch, it must be that $I(t) < s$ so we want to order up to S**
- **Assign block**
  - Define `Order Quantity` Attribute
    - Could have made this a Variable in this model with these parameters, but it's more general for it to be an Attribute ... why?
  - Increment `Total Ordering Cost` Variable
- **Delay block for `Delivery Lag`**
- **Assign block to increment `Inventory Level` by the `Order Quantity`**
- **Dispose block**

Model 5-4: (*s*, *S*) Inventory Simulation Animation

- **Plot separate "in the black" and "in the red" curves**
  - If in backlog, red curve will be plotted in negative direction due to its Expression
- **Pair of Level ("thermometer") animations**
  - Fill Direction for "in the red" is Down



Intermediate Modeling and Steady-State Statistical Analysis

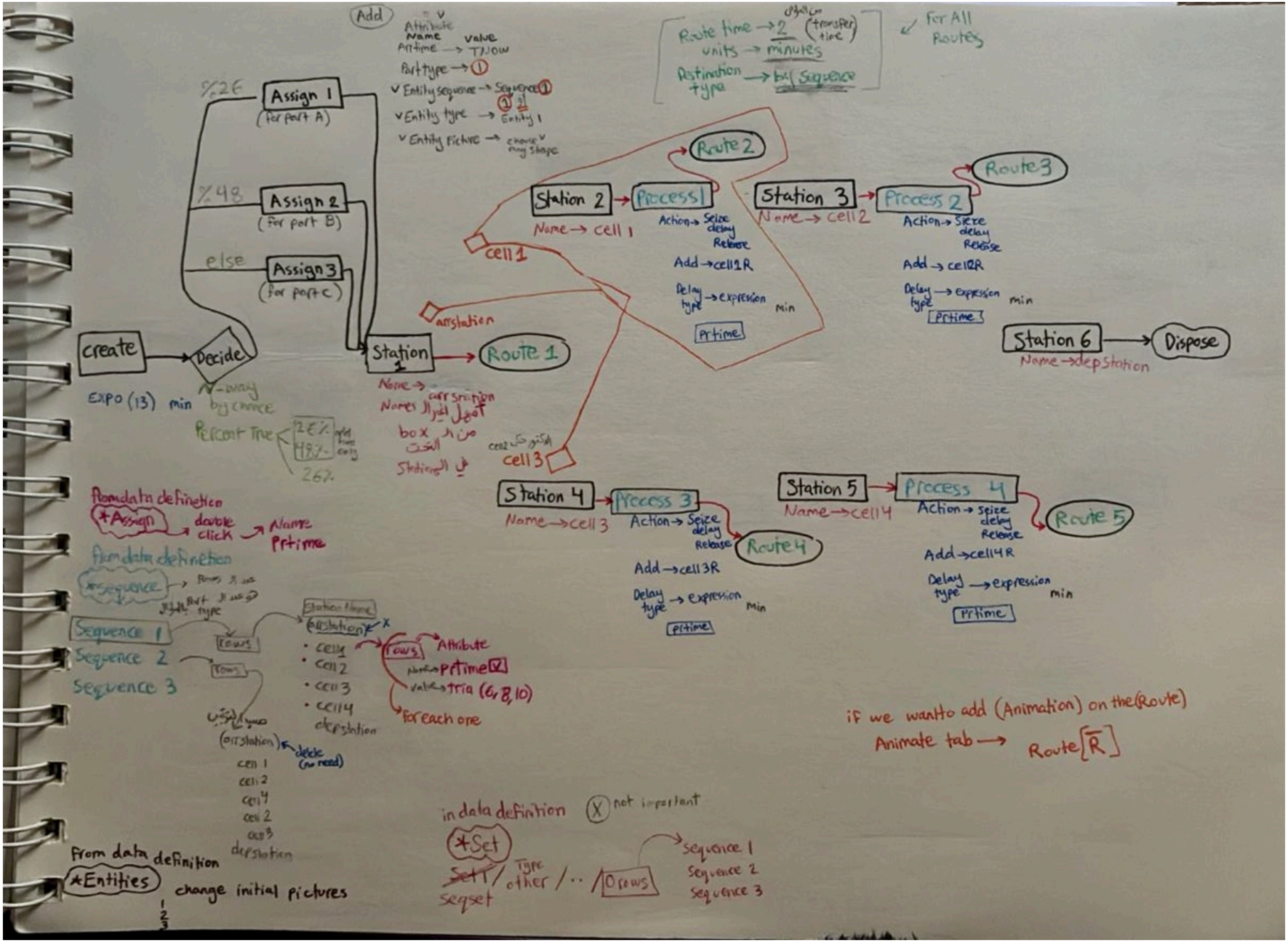
Chapter 7

Last revision June 9, 2003

What We'll Do ...

- **Model 7-1: A small manufacturing system**
  - Entity-dependent Sequences
  - Data requirements and availability
  - Verification (debugging)
- **Statistical analysis of steady-state simulations**
  - Warmup and run length
  - Truncated replications
  - Batching
  - Other methods and goals

Route

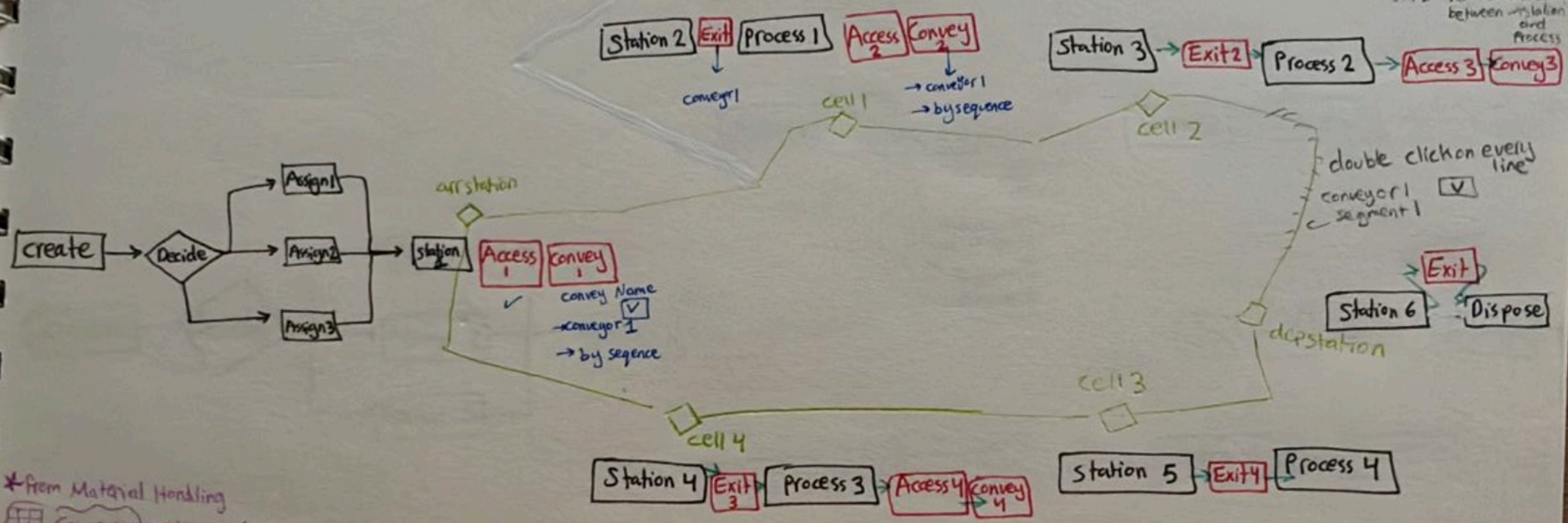


Add
Attribute
Name Value
Arrtime → TNOW
Part type → 1
Entity sequence → Sequence 1
Entity type → Entity 1
Entity Picture → choose any shape
Route time → 2 (transfer time)
units → minutes
Destination type → by Sequence
For All Routes
%26
Assign 1 (for part A)
%48
Assign 2 (for part B)
else
Assign 3 (for part C)
Create
EXPO (13) min
Decide
N-way by chance
Percent True
26%
48%
26%
Station 1
Route 1
Name →
Name
arrstation
box
cell 3
Station
Station 2
Name → cell 1
Process 1
Action → Seize delay Release
Add → cell 1 R
Delay type → expression min
Prtime
Route 2
cell 1
arrstation
Station 3
Name → cell 2
Process 2
Action → Seize delay Release
Add → cell 2 R
Delay type → expression min
Prtime
Route 3
Station 6
Name → depStation
Dispose
Station 4
Name → cell 3
Process 3
Action → Seize delay Release
Add → cell 3 R
Delay type → expression min
prtime
Route 4
Station 5
Name → cell 4
Process 4
Action → seize delay Release
Add → cell 4 R
Delay type → expression min
Prtime
Route 5
from data definition
*Assign
double click
Name
Prtime
from data definition
*Sequence
Rows
Part type
Sequence 1
Sequence 2
Sequence 3
rows
Station Name
arrstation
cell 1
cell 2
cell 3
cell 4
depstation
rows
Attribute
Prtime
Value → tria (6,8,10)
for each one
arrstation
delete (no need)
cell 1
cell 2
cell 4
cell 2
cell 3
depstation
From data definition
*Entities
change initial pictures
1
2
3
in data definition
X not important
*Set
Set
Type
other
Seqset
0 rows
sequence 1
sequence 2
sequence 3
if we want to add (Animation) on the (Route)
Animate tab →
Route [R]

conveyor

from the page before
we will learn → Conveyor
→ Transportation

step 1 remove Route
step 2 remove connector between station and process



*from Material Handling
Conveyor → double click (we can modify) units per minute

*from Material Handling
Segment → double click → Begining Station → 0 rows
arrstation

Next	Length
cell 1	24
cell 2	39
depstation	21
cell 4	24
cell 3	39
arrstation	21

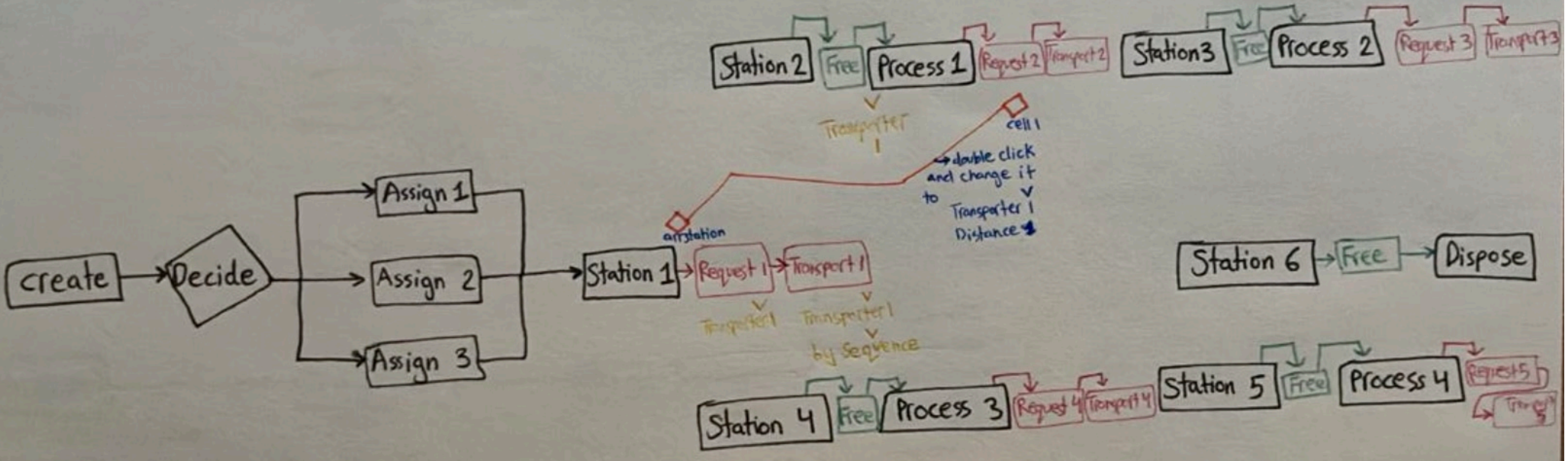
*To see the Animation
→ Animate → Segment → then draw lines and Stations

to see Resource Animation
→ Animate → Resource → اختار أي رسمة
ونكبس على وحدة
منهم و السهم الشمال

*for Transportation
Transporter → double click
Distance → double click → 0 rows

Transporters

هي نفس الـ Conveyors لكن الاختلاف
انه هنا بيقدر يتحرك من اي مكان



* From Material Handling

Transporter → double click → we can modify to per minute (but for here put → per hour, velocity → 120, # of units → 10)

* From Material Handling

Distance → double click → Stations, 0 Rows → double click

Beginning Station	Ending Station	Distance
arrstation	cell1	37
arrstation	cell2	74
cell 1	arrstation	155
cell 1	cell2	45
cell 1	cell3	92
cell 1	cell 4	129
⋮	⋮	⋮

The Total number of them is (25)

* From Material Handling

→ put Request, Transport, Free

* then For Animation :-

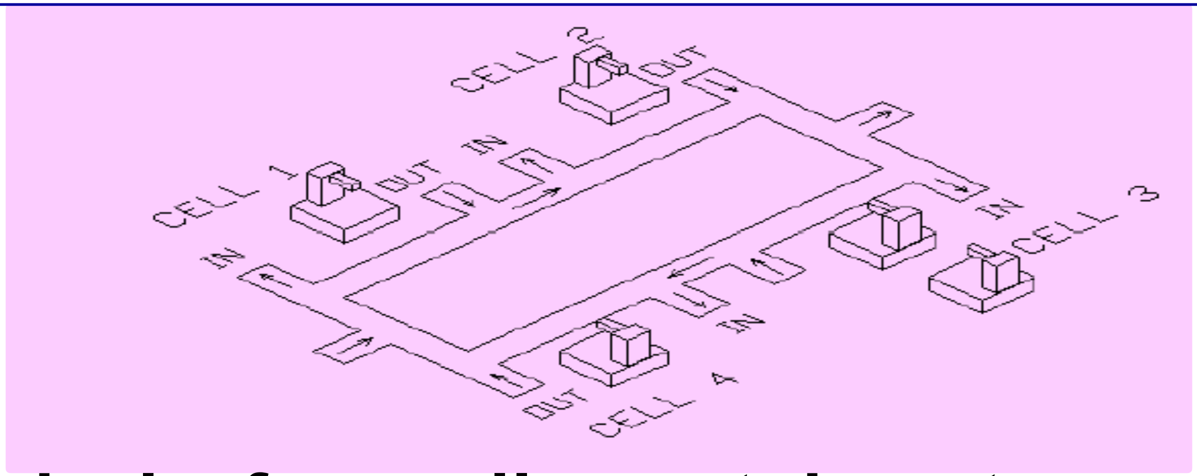
① → Animate → Distance → (نسوي الشبكة علويه)

② → Animate → Transporter → الشاحنة ثم السهم » → Identifier → Transporter 1 → then ok

From the Question :-

FROM \ TO	ORDER Release	Cell 1	Cell 2	Cell 3	Cell 4	Exit system
Order Release		37	74			
Cell 1	155		45	92	129	
Cell 2	118	139		55	147	
Cell 3	71	92	129		45	155
Cell 4	34	55	92	139		118
Exit system	100	121	158	37	74	

Model 7-1:
A Small Manufacturing System



- **Part arrivals, four cells, part departures**
- **Cells 1, 2, and 4: single machine each**
- **Cell 3: two machines — newer one 20% faster**
  - Need: way to model non-identical resource units
- **Circular layout of cells**
- **Parts enter at left, exit at right, travel only clockwise, all transfer times = 2 min. (realistic?)**

Route time

A Small Manufacturing System (cont'd.)

Decide

- **Three separate part types**
  - create واحدة
  - Interarrivals (all types merged) ~ expo(13) minutes
  - 26% type 1, 48% type 2, 26% type 3
- **Different part types follow different routes, have different (triangular) processing times:**

Part كل حيمشي من sequence مختلف

Part Type	Cell/Time	Cell/Time	Cell/Time	Cell/Time	Cell/Time
1	1 6, 8, 10	2 5, 8, 10	3 15, 20, 25	4 8, 12, 16	
2	1 11, 13, 15	2 4, 6, 8	4 15, 18, 21	2 6, 9, 12	3 27, 33, 39
3	2 7, 9, 11	1 7, 10, 13	3 18, 23, 28		

Parameters are for the slow machine at Cell 3.

- **Observe utilizations, time/number in queues, cycle times (times in system) by part type**
- **Run for 32 hours**

New Arena Concepts

- **Non-identical machines at Cell 3**
- **Different entity types follow different process plans**
  - Previous models – all entities went through same sequence of stations, maybe with Decides for branching
  - Now, need process plan with automatic routing by entity type – different *Sequence* assigned to each entity (like an attribute), and entity follows its own sequence
  - Won't use direct Connect or Routes … instead we tell entities departing from modules to follow their own Sequence

$\boxed{\text{Route}}$
Route time → 2
unit → min
Destination type → by Sequence

  - Arena internally keeps track of where entity is, where it will go next

The Modeling Approach

- **Usually there are many ways to build a (correct) Arena model**
  - And also many ways to do so incorrectly …
- **Important to think about data structures**
  - What data are available?
  - How will they be stored in the model?
- **For this model …**
  - Use Sequence for part transfer (described below)
  - As part of Sequence definition, can define Attributes
    - Do for processing times at all cells but Cell 1
  - Use an Expression for processing times at Cell 1
  - Use Variables for new-machine speedup at Cell 3, part transfer times

Sequence Data Module

- **Advanced Transfer panel**
- **Double-click for new row for each process plan**
  - Name for each Sequence
  - Open Steps column for subdialog
    - Define ordered sequence of Stations to be visited in the Sequence … must have Station Names already defined
    - Double-click to add a new Station to the bottom of the Sequence list; right-click to insert/delete a row
    - Name for each step
    - Possible Assignments of Attribute, Variable, Pictures, etc. at each station in the Sequence … this is done *before* transferring the entity to this step in the sequence
    - In this model, Attribute assignment used to attach `Process Time` Attribute to entity for the next Cell (except for Cell 1)

Sequence Data Module (cont'd.)

- **Assign Sequence Name to entities that follow it**
- **In Route modules, select Sequence as Destination Type (rather than Station)**
  - Departing entity looks in its own sequence to know where to go next
- **Arena tracks Sequence-following entities via automatic attributes**
  - Sequence name, NS (or Entity.Sequence)
  - Station (where entity is or is going to), M (or Entity.Station)
  - JobStep along the sequence, IS (or Entity.JobStep)
- **Normally, entity is assigned a Sequence, travels its route, then exits**
  - Can interrupt this sequence, jump forward/backward (tricky)
- **Remember to define the "exit" station**

Expression Data Module

- **Advanced Process panel**
- **Use for processing times at Cell 1**
  - Could have done in Sequences, as for other Cells … done this way mostly to illustrate its use
- **Three different part types at Cell 1, so use a vector-valued Expression with three rows**
  - Name for the expression, **`Cell 1 Times`**
  - Rows, **`3`**
  - Expression Values subdialog
    - Cell 1 processing times for the three part types
    - Order matters, since index is part type … will reference as **`Cell 1 Times(Part Index)`** in model

Variable Data Module

- **Basic Process panel**
- **`Factor` variable**
  - Speed factor at Cell 3 – need a two-row vector
    - Assume new (faster) machine is #1, old (slower) machine is #2
    - Set to `0.8` for index 1; set to `1.0` for index 2
- **`Transfer Time` variable**
  - Holds transfer-time constant of 2 minutes between stations
  - Just a scalar, not a vector or matrix
  - Used for model generality – if all transfer times changed, this makes it easy to implement this change
- **These are the Initial Values of variables … any entity can change them**
  - But they're constant in this model

Set Data Module

- **Basic Process panel**
- **Define three sets**
  - Resource set, **`Cell 3 Machines`**
    - For new and old machine (in that order) at Cell 3
    - Resource Names – could have already defined them in Resource data module, or can define them here
  - Entity Picture set, **`Part Pictures`**
    - To attach to entities once their part type is determined
    - Picture Names – could have already defined them elsewhere (*Edit > Entity Pictures*), or can define them here
  - Entity Type set, **`Entity Types`**
    - To attach to entities once their part type is determined
    - Entity Types – define them here

Advanced Set Data Module

- **On Advanced Process panel**
- **Needed since Set data module does not have "Other" category for Type**
  - Need to form a set of Sequences to attach the right one to arriving entities once their part type is determined
  - Define Name of set to be **`Part Sequences`**
  - Set Type is "Other"
  - Members subdialog – Add rows, type in names in "Other" column (have to remember or look up the Sequence names)

Run > Setup and *Edit > Entity Pictures*

- ***Run > Setup* Dialog**
  - Replication Parameters Tab
    - Replication Length = 32 Hours
    - 24 Hours/Day
    - Base Time Units = Minutes
- ***Edit > Entity Pictures***
  - Create three custom pictures – **`Picture.Part 1`**, **`Picture.Part 2`**, **`Picture.Part 3`**
  - Copy blue, red, and green ball pictures
  - Rename them
  - Picture Editor to put white numbers inside via Text object

Part Arrivals

- **Create module for arrival of one part**
  - One-at-a-time, Time Between Arrivals is exponential with mean 13 minutes
  - Don’t know the part type yet …
- **Assign module for part attributes**
  - **`Part Index`** = draw from DISC probability distribution
    - Pairs *cumulative* probability, value
  - **`Entity.Sequence`** = **`Part Sequences(Part Index)`**
    - **`Part Index`** attribute already assigned … order matters
    - Index into **`Part Sequences`** (Advanced) Set
  - **`Entity.Type`** = **`Entity Types(Part Index)`**
  - **`Entity.Picture`** = **`Part Picture(Part Index)`**

Release Arriving Entity into System

- **Use previously defined Sequences, assigned to entity via (Advanced) Set of Sequences**
- **Send arriving entity through a Station module to define its current station location**
  - Station Name = **`Order Release`**
  - Other five station names already defined via Sequences
- **Route module to start it on its way**
  - Route Time = **`Transfer Time`** (a Variable previously defined) Minutes
  - Destination Type = Sequential
    - Arena will direct this entity according to its own sequence
    - It just arrived so Arena initializes its JobStep attribute

Logic for Cell 1

- **Station module to define the station location**
  - Station Name = `Cell 1`, on pull-down list for stations since it was previously defined in Sequences
- **Cell 1 Process module**
  - Action = Seize Delay Release
  - Resources subdialog
    - Type = Resource (not Set … yet)
    - Resource Name = `Cell 1 Machine`, Quantity to seize = 1
  - Delay Type = Expression
    - Expression = `Cell 1 Times(Part Index)` Minutes, using the previously-defined Expression `Cell 1 Times`
- **Route module from Cell 1**
  - Destination Type = Sequential
  - Station already defined (on incoming side)

Logic for Cells 2 and 4

- **Incoming Station module – similar to Cell 1**
  - Except for names of Module and Station
- **Process module**
  - Action, Resources, Delay Type – similar to Cell 1
  - Expression for Delay time = **`Process Time`**
    - Attribute defined in Sequence module for each job type at this point in its sequence for Cells 2 and 4
    - Note that Part Type 2 visits Cell 2 twice in its sequence, with *different* delay-time distributions … this data structure is general enough to handle this
- **Outgoing Route module – similar to Cell 1**
  - Except for name of Module

Logic for Cell 3

- **Station, Route modules – similar to Cells 1, 2, 4**
- **Process module**
  - Action, Delay Type – similar to Cells 1, 2, 4
  - Resources subdialog
    - Type = Set, Set Name = **`Cell 3 Machines`**
    - Selection Rule for set = Cyclical

      Maybe Preferred Order would have been better???
    - Save Attribute = **`Machine Index`** (will be 1 or 2)
  - Expression for Delay time =

    `Process Time * Factor(Machine Index)`

    to multiply by 0.8 if entity gets the new machine (#1), using the preciously-defined vector variable **`Factor`**
  - See book for alternative (cute) expression that avoids the need for the vector variable **`Factor`**

Digression: Data Structures

- **Why an Expression for processing times at Cell 1 rather than entity Attribute assigned in Sequences as for the other cells?**
  - Frank answer: Just to show the use of Expression
  - Could easily have treated Cell 1 like the others
- **Conversely, could have used Expression for processing times at Cells 3 and 4**
  - But there would be a problem with Cell 2
    - Part 2 visits it twice with different processing-time distributions, so would have to indicate which visit somehow
  - Moreover, this is a very small model
- **Moral: Think carefully about data structure!**

Logic for Exiting the System

- **Station module to define this location**
  - Station Name = **`Exit System`**
- **Dispose module**
  - Record Entity Statistics box is checked
  - Will generate one of the outputs we want, cycle time (time in system) separated out by part type, since they map onto the entity types for this model
  - So don't need separate Record modules here to collect cycle times
- **Model would run at this point, give correct output results … but develop animation to show queues, resources, and movement …**

Animation

- **Pull animation away from logic, data modules**
- **Move, resize, reorient queues for realism**
- **Animate Routes (all movement possibilities)**
  - Thick "bundles" of routes — Shift key, Snap to Grid
  - Heed clockwise direction
  - Draw lines to define route "lanes"
- **Import, modify AutoCAD .dxf file for backdrop and resource pictures (see text)**
- **Fine-tune resource pictures**
  - Layers for seize point
- **In animation, note that entities travel at very different rates, pass each other … realistic???**

Verification

- System $\rightarrow$ Model $\rightarrow$ "Code"
- *Validation*: Is Model = System?
- *Verification*: Is "Code" = Model? (debugging)
- The Truth: Can probably never completely verify, especially for large models

Verification (cont'd.)

- **Some techniques to *attempt* verification**
  - Eliminate error messages (obviously)
  - Single entity release, Step through logic
    - Set Max Batches = 1 in Arrive
    - Replace part-type distribution with a constant
  - "Stress" model under extreme conditions
  - Performance estimation — like slide-rule decimal placement
  - Look at generated SIMAN .mod and .exp files
    - *Run > SIMAN > View*

Statistical Analysis of Output from Steady-State Simulations

- **Recall: Difference between terminating, steady-state simulations**
  - Which is appropriate depends on goal of study, and not so much on the model structure
    - Most models could be used for terminating or steady-state analysis
- **Now, assume steady-state is desired**
  - Be sure this is so, since running and analysis is a lot harder than for terminating simulations
- **Naturally, simulation run lengths can be long**
  - Opportunity for different internal computation order
  - Can change numerical results
  - Underscores need for statistical analysis of output

Warm Up and Run Length

- **Most models start *empty and idle***
  - *Empty*: No entities are present at time 0
  - *Idle*: All resources are idle at time 0
  - In a terminating simulation this is OK if realistic
  - In a steady-state simulation, though, this can bias the output for a while after startup
    - Bias can go either way
    - Usually downward (results are biased low) in queueing-type models that eventually get congested
    - Depending on model, parameters, and run length, the bias can be very severe

Warm Up and Run Length (cont'd.)

- **Remedies for initialization bias**
  - Better starting state, more typical of steady state
    - Throw some entities around the model
    - Can be inconvenient to do this in the model
    - How do you know how many to throw and where?

      This is what you're trying to estimate in the first place!
  - Make the run so long that bias is overwhelmed
    - Might work if initial bias is weak or dissipates quickly
  - Let model *warm up*, still starting empty and idle
    - *Run* > *Setup* > *Replication Parameters*: Warm-up Period

      Time units!
    - "Clears" all statistics at that point for summary report, any Outputs-type saved data from Statistic module of results across replications

Warm Up and Run Length (cont'd.)

- **Warm-up and run length times?**
  - Most practical idea: preliminary runs, plots
  - Simply "eyeball" them
  - Be careful about variability — make multiple replications, superimpose plots
  - Also, be careful to note "explosions"
- **Possibility – different Warm-up Periods for different output processes**
  - To be conservative, take the max
  - Must specify a single Warm-up Period for the whole model

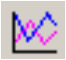
Warm Up and Run Length (cont'd.)

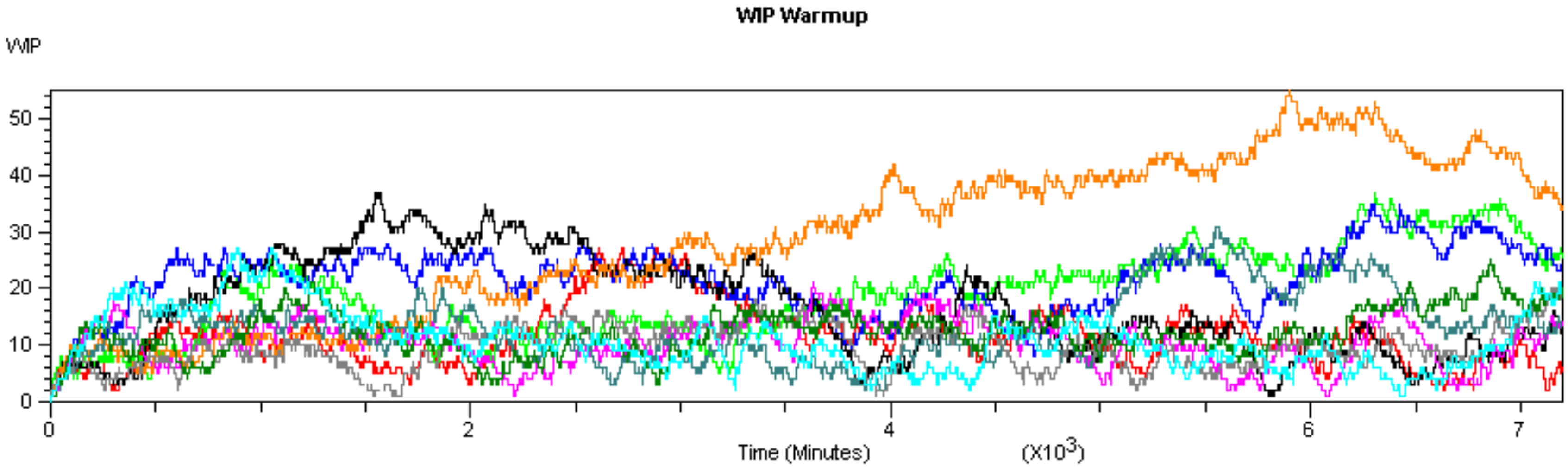
- **Create a single overall output performance measure for Model 7-1 … modify it into Model 7-2**
  - Measure is time-average total number of parts in system
  - Statistic module
    - Time-Persistent type, Name and Report Label **`Total WIP`**
    - Expression (via Expression Builder … details in book)

      `EntitiesWIP(Part 1) + EntitiesWIP(Part 2) + EntitiesWIP(Part 3)`
    - Output File **`Total WIP History.dat`** to save within-run data

      Animated plots disappear, can't overlay plots from multiple replications … will use Output Analyzer to plot the saved data
  - Speed up the run
    - Check *Run > Run Control > Batch Run (No Animation)*
    - Uncheck boxes in *Run > Setup > Project Parameters*, Dispose module
  - Lengthen Replications to 5 days, do 10 Replications

Warm Up and Run Length (cont'd.)

- **In Output Analyzer**
  - New data group, Add the file **`Total WIP History.dat`**
  - *Graph > Plot* or 
  - Add **`Total WIP History.dat`**, Replications = All, enter Title, axis labels



  - No apparent explosion
  - Warm-up about 2000 min.; round up to 2 days (2880 min.)

Truncated Replications

- **If you can identify appropriate warm-up and run-length times, just make replications as for terminating simulations**
  - Only difference: Specify Warm-up Period in *Run > Setup > Replication Parameters*
  - Proceed with confidence intervals, comparisons, all statistical analysis as in terminating case
- **Model 7-3: modify Model 7-2**
  - Warm-Up period = 2 Days
  - Stick with (total) replication length of 5 Days
  - Stick with 10 replications
  - Delete Output File in Statistic module

Truncated Replications (cont'd.)

- **Get cross-replications 95% confidence-interval Half Widths in Reports**
  - For average Total WIP, got 16.39 ± 6.51
  - Without the Warm-up, this was 15.35 ± 4.42
  - To sharpen the comparison of the effect of the Warm-up, did 100 (rather than 10) replications with and without it:
    - With Warm-up: 15.45 ± 1.18
    - Without Warm-up: 14.42 ± 0.86
  - Half Widths with Warm-up are larger since each replication is based on the last 3 days, not all 5 days
- **Smaller confidence intervals? Have a choice:**
  - More replications, same length
  - Same number of replications, each one longer
    - This might be the safer choice to guard against initialization bias

Batching in a Single Run

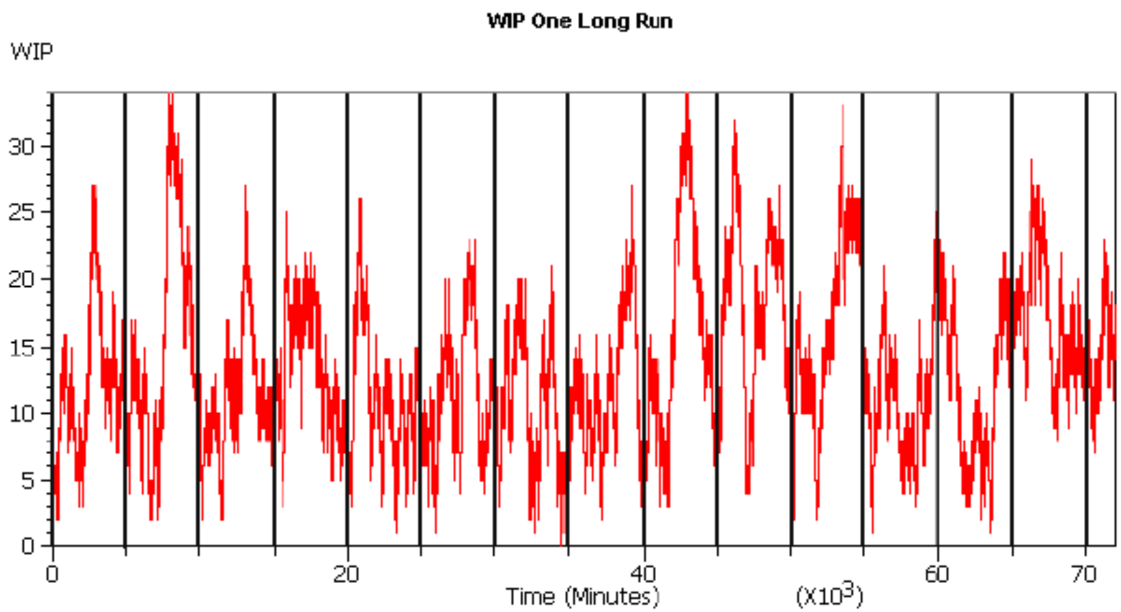
- **If model warms up very slowly, truncated replications can be costly**
  - Have to "pay" warm-up on each replication
- **Alternative: Just one *R E A L L Y* long run**
  - Only have to "pay" warm-up once
  - Problem: Have only one "replication" and you need more than that to form a variance estimate (the basic quantity needed for statistical analysis)
    - Big no-no: Use the individual points within the run as "data" for variance estimate
    - Usually correlated (not indep.), variance estimate biased

Batching in a Single Run (cont'd.)

- **Break each output record from the run into a few large *batches***
  - Tally (discrete-time) outputs: Observation-based
  - Time-Persistent (continuous-time): Time-based
- **Take averages over batches as "basic" statistics for estimation: *Batch means***
  - Tally outputs: Simple arithmetic averages
  - Time-Persistent: Continuous-time averages
- **Treat batch means as IID**
  - Key: batch size must be big enough for low correlation between successive batches (details in text)
  - Still might want to truncate (once, time-based)

Batching in a Single Run (cont'd.)

- **Modify Model 7-3 into Model 7-4**
  - One replication of 50 days (about the same effort as 10 replications of 5 days each)
  - A single 2-day Warm-up Period
  - Statistic module, save WIP data once again for plot



How to choose batch size?

Equivalently, how to choose the number of batches for a fixed run length?

Want batches big enough so that batch means appear uncorrelated.

Batching in a Single Run (cont'd.)

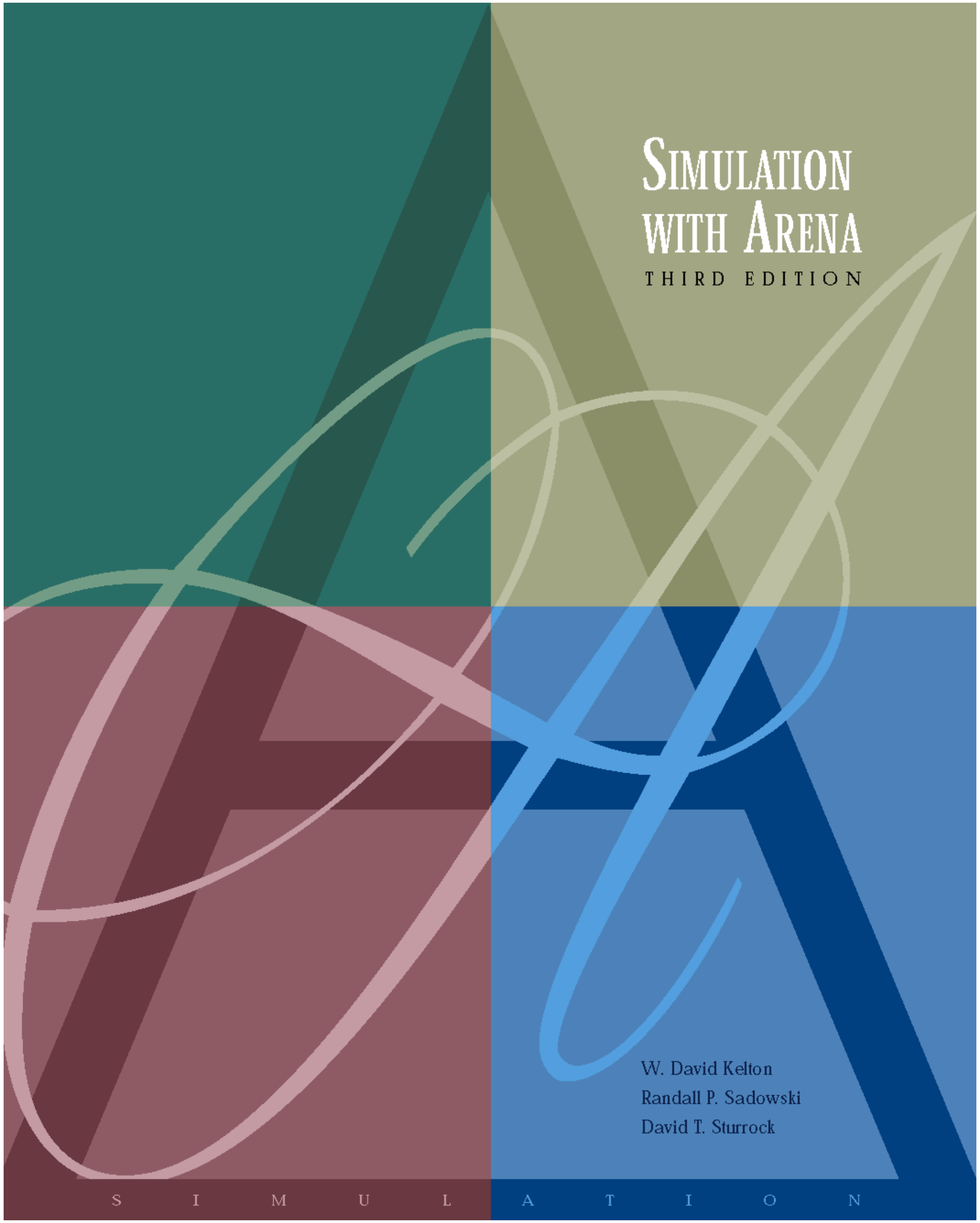
- **Arena automatically attempts to form 95% confidence intervals on steady-state output measures via batch means from within each single replication**
  - "Half Width" column in reports from one replication
    - In Category Overview report if you just have one replication
    - In Category by Replication report if you have multiple replications
  - Ignore if you're doing a terminating simulation
  - Uses internal rules for batch sizes (details in text)
  - Won't report anything if your run is not long enough
    - "(Insufficient)" if you don't have the minimum amount of data Arena requires even to form a c.i.
    - "(Correlated)" if you don't have enough data to form nearly-uncorrelated batch means, required to be safe

Batching in a Single Run (cont'd.)

- **Results from Model 7-4:**
  - Category Overview report, average total WIP: 13.64 ± 1.38
  - Half Width considerably smaller than for truncated replications (10 replications, 5 days each, 2-day Warm-ups)
    - Here we spend only a total of 2 days warming up, and with truncated replications we spent $10 \times 2 = 20$ days warming up
- **Can check batch-means half widths during run**
  - Arena variables THALF(Tally ID), DHALF(Dstat ID)
- **Can decide on your own batch sizes, form batch means and c.i.'s "by hand" with Output Analyzer**
  - Why? Use in statistical comparison procedures
  - More information in book

What To Do?

- **Several approaches, methods for steady-state statistical analysis … many more exist**
- **Opinion:**
  - Avoid steady-state simulation … look at goal of project
  - If you really *do* want steady-state
    - First try Warm-up, truncated replications
    - If model warms up slowly, making truncated replications inefficient, consider Arena's batch-means methods in a single long run with a single Warm-up Period at its beginning … can't use statistical methods in PAN or OptQuest, though
- **Other methods, goals – references in text**



Entity Transfer

Chapter 8

Last revision June 9, 2003

What We'll Do ...

- **Types of Entity Transfers**
- **Model 8-1: Resource-Constrained Transfers**
- **Models 8-2, 8-3: Transporters**
- **Conveyors**
  - Model 8-4: Non-accumulating
  - Model 8-5: Accumulating

Types of Entity Transfers So Far

- **Connect**
  - Zero-delay
  - Connection graphic vs. module Labels (no graphic)
- **Route**
  - Non-zero-delay — constant, r.v., expression
  - Stations, animated Routes
  - Fixed routes vs. entity-dependent Sequences
- **Connect and Route both assume:**
  - No limit on number in transit at a time
  - Entities have their own feet

New Types of Entity Transfers

- ***Resource-constrained* transfers**
  - Limit total number of entities in transit at a time
  - Entities still have their own feet
  - Telecommunications (number of packets), logistics (number of vehicles)
- **Material-handling devices**
  - *Transporters* – fork lifts, trucks, carts, wheelchairs
    - Usually place limits on numbers, capabilities of transporters
    - Like a Resource, except moveable
  - *Conveyors*
    - Belts, hook lines, escalators
    - Usually limit space on conveyor, speed
    - Non-accumulating vs. accumulating

Model 8-1: Small Manufacturing System with Resource-Constrained Transfers

- **Original system (Model 7-1)**
  - Assumed all transfer times = 2 minutes … keep (for now)
  - Parts have their own feet … keep (for now)
  - No limit on number of parts in transit at a time … dump
    - Now – no more than 2 parts can be in motion at a given time
    - If other parts are ready to go, they must wait until there's room to go
- **Model via existing constructs — think creatively**
  - Model "space" on the "road" as a Resource
  - Limit the number of Units of this Resource
  - Entity must Seize unit of "space" resource before beginning trip, Release it at end of trip

Two Ways to Model Resource-Constrained Transfers

- **Both use a new `Transfer` Resource representing space on the transitways**
  - Capacity set to **2** in Resource data module
- **Maybe the most obvious way (but won't do) …**
  - Before each Route module insert a Seize module to Seize one unit of **`Transfer`** (queue, priority details … see text)
  - After each Station module (except **`Order Release`**) insert a Release module to free up one unit of **`Transfer`**

Two Ways to Model Resource-Constrained Transfers (cont'd.)

- **Different way (will do, to illustrate new modules, set up for transporters and conveyors) …**
  - Replace Route modules with *Leave* modules (Advanced Transfer panel)
    - Transfer Out: Seize unit of `Transfer` resource before leaving station
      - Resource, Resource Set, particular member of a Resource Set
      - Can specify priorities
    - Also contains the Route operation
    - Get individual queues, with animation, for parts waiting to go
  - Replace Station modules with *Enter* modules (Advanced Transfer panel)
    - Defines the Station
    - Option of an unload Delay time (`0` for this model)
    - Transfer In: Release `Transfer` resource
- **Effect – slight increase in cycle times in system**

Transporter Concepts

- **Carts, fork lifts, trucks, wheelchairs, people, …**
- **When entity is ready to go somewhere, it needs to be "picked up" and moved**
- **Use *Transporters* — "moveable" resources**
- **Activities: *Request*, *Transport*, *Free***
  - Transporter Selection Rule: If > 1 transporter is available when Requesting
  - When freed and > 1 entity is waiting: Priorities, closest one
- **Two types of Transporters**
  - *Free-Path* (we'll do)
    - Travel time depends only on velocity, distance
    - Ignore "traffic jams" and their resulting delays
  - *Guided* (won't do)
    - AGVs, intersections, etc.

The Small Manufacturing System with Transporters

- **Have two carts to transport parts**
  - A cart can carry one part at a time
  - Carts move at 50 feet/minute
    - Will need to specify accurate distances between Stations
  - It takes 0.25 minute to load part on a cart, 0.25 minute to unload it from a cart
- **Modify Model 8-1 to Model 8-2**

The Small Manufacturing System with Transporters (cont'd.)

- **Create Transporter in Transporter data module (Advanced transfer panel)**
  - Name = **`Cart`**, Capacity = **`2`**, Velocity = **`50`**
  - Default the Distance Set (later), Units = Per Minute, Initial Positions
    - *Mind the units* – consistency here, in Distance Set (later)
- **Animation picture for `Cart` Transporter**
  - Transporter button , Animate Transfer toolbar
  - Identifier = **`Cart`**, pictures for Idle, Busy, Inactive states
  - Draw or copy from .plb picture libraries
  - Ride point (details in book)
  - Drop it anywhere in flowchart view (hidden during run)

The Small Manufacturing System with Transporters (cont'd.)

- **Request a `Cart` – modify existing Leave modules**
  - Delay = **`0.25 Minute`** for load time
  - Transfer Out = **`Request Transporter`**
  - Transporter Name = **`Cart`**
  - Selection Rule = **`Smallest Distance`**
    - Applies when > 1 transporter is available
    - Others: Cyclic, Random, Preferred Order, Largest Distance (???)
  - Save Attribute = **`Cart #`** (remember which cart … for later)
  - Connect Type = **`Transport`**
    - Move Time disappears … determined by Velocity, Distances (later)
  - Station Type = **`Sequence`**
- **Instead of Leave: Request-Delay-Transport**
  - More complex, more flexible – book has details, examples

The Small Manufacturing System with Transporters (cont'd.)

- **Free the Cart – modify existing Enter modules**
  - Delay = `0.25 Minute` for unload time
  - Transfer In = `Free Transporter`
  - Transporter Name = `Cart`
  - Unit Number = `Cart #` attribute of part entity
- **Instead of Enter: Station-Delay-Free**
  - More complex, more flexible – book has details, examples

Distances for Transporters

- **Define contents of Distance Set `Cart.Distance`**
- **Distances (in feet) moved by parts:**

	To				
	Cell 1	Cell 2	Cell 3	Cell 4	Exit System
Order Release	37	74			
Cell 1		45	92		
Cell 2	139		55	147	
Cell 3				45	155
Cell 4		92			118

Units!!

  - Blank cells: part movements that don't occur
- **Enter these data in Distance data module (Advanced Transfer panel)**
  - Name = `Cart.Distance`
  - Stations button, add 11 rows with Beginning Station, Ending Station, Distance for above data

Units!!

  - Direction is implied; could be asymmetric

Why are there 25 rows??

Animating Transporter Movement

- **Add distances to animation**
- **Delete all the old Route Path animation objects**
  - But leave the Station animations
- **Add animated transporter distances with Distance button**  **, Animate Transfer toolbar**
  - Dialog, placement similar to Route Paths
  - Identifier = **`Cart.Distance`**
  - Click in Beginning Station marker, intermediate clicks, Ending Station marker
  - Options for Rotate, Flip
  - Grid, Snap to help place animated transporter distances

Parking Areas for Transporters

- **Animate transporters when they're free**
- **Parking button**  **, Animate Transfer toolbar**
  - Like a Queue animation – Point vs. Line, Shift, Rotate
  - Cursor becomes cross hairs, click near lower left of Station marker to start, click for first Point or head of Line
  - More clicks for more Points (double-click to end), or second click to end Line
  - Want enough points/space for all transporters (2 here)
  - Repeat for all Stations where Transporters could be freed

More Distances — Empty Transporters

- **Above Distances incomplete — only for part movements along their sequences**
- **Transporters must also move when empty (*deadheading*)**
  - In general, $n(n - 1)$ distances need definition for network with n nodes
  - Some not possible — Order Release to Exit System
- **14 more distances to define in Distances data module (not grayed):**

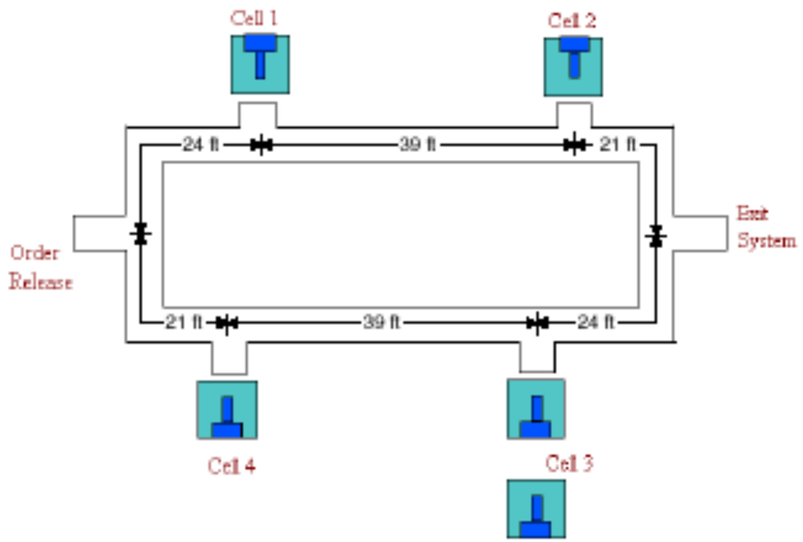
		To					
		Order Release	Cell 1	Cell 2	Cell 3	Cell 4	Exit System
From	Order Release		37	74			
	Cell 1	155		45	92	129	
	Cell 2	118	139		55	147	
	Cell 3	71	92	129		45	155
	Cell 4	34	55	92	139		118
	Exit System	100	121	158	37	74	

Model 8-3: Refining the Animation for Transporters

- **As it stands, `Part` Entities disappear from animation when waiting to be picked up by a `Cart` Transporter**
  - Model logic OK … get right answers … animation is flawed
- **Solution – *Storage* for entity to reside in, be animated, while it waits for something (here, a `Cart` Transporter)**
  - Can get statistics on numbers in Storages
- **But Storages not available with modules from Advanced Transfer panel**
  - Use lower-level SIMAN modules from Blocks panel … see book for specific details

Conveyors

- **Replace Transporters with a conveyor**
- **Loop conveyor to follow main path, clockwise**
- **Six entrance/exit points**
  - Load, Unload takes 0.25 minute
  - Each part is 4 feet per side, but want 6 feet of conveyor space for clearance on corners
- **Speed = 20 feet/minute** *Units!!*
- **Distances:**



Conveyor Concepts

- **Entity to be conveyed must wait for space**
- **Conveyor consists of *cells***
  - Equal size, constantly moving – think of a narrow escalator
- **Entities might require multiple contiguous cells**
- **Must define *cell size*; tradeoff involved:**
  - Small cells: accurate model but slow execution
  - Large cells: just the opposite!
- **Entities *Access* space, *Convey*, and *Exit***
- **Conveyor = series of linear *Segments***
  - Each segment starts and ends at a Station
  - Link to form loops, diverge points, converge points

Types of Conveyors

- **Both travel in a single, irreversible direction**
- ***Nonaccumulating*: belt, bucket line, escalator**
  - Spacing between entities on it doesn't change
  - Entire conveyor stops for entity Access/Exit if Load/Unload time is > 0
- ***Accumulating*: rollers, freeway**
  - Conveyor never stops moving
  - If entity on it stops to Exit, other entities behind it are blocked and bunch up (entities ahead of it keep moving)
  - When blockage ends, blocked entities go on but maybe not all at once (spacing requirements)

Model 8-4: Small Manufacturing System with Nonaccumulating Conveyors

- **Modify Model 8-1 (resource-constrained transfer)**
- **Define new Variables `Load Time` and `Unload Time`, each with initial value 0.25**
- **Delete all the Route Paths**
- **Define Conveyor via Conveyor data module, Advanced Transfer panel**
  - Conveyer = `Loop Conveyor`
  - Segment Name = `Loop Conveyor.Segment`
  - Type = Non-Accumulating
  - Velocity = `20` (feet), Units = Per Minute *Units!!*
  - Cell Size = `3` (feet) *Units!!*
  - Max Cells Occupied = `2` (cells per entity)

Leave, Enter Modules for Conveyor

- **Change each Leave module**
  - Delay = `Load Time`, Units = Minutes
  - Transfer Out = Access Conveyor
  - Conveyor Name = `Loop Conveyor`
  - # of Cells = `2`
  - Connect Type = Convey
- **Change each Enter module**
  - Delay = `Unload Time`, Units = Minutes
  - Transfer In = Exit Conveyor
  - Conveyor Name = `Loop Conveyor`

Conveyor Segments

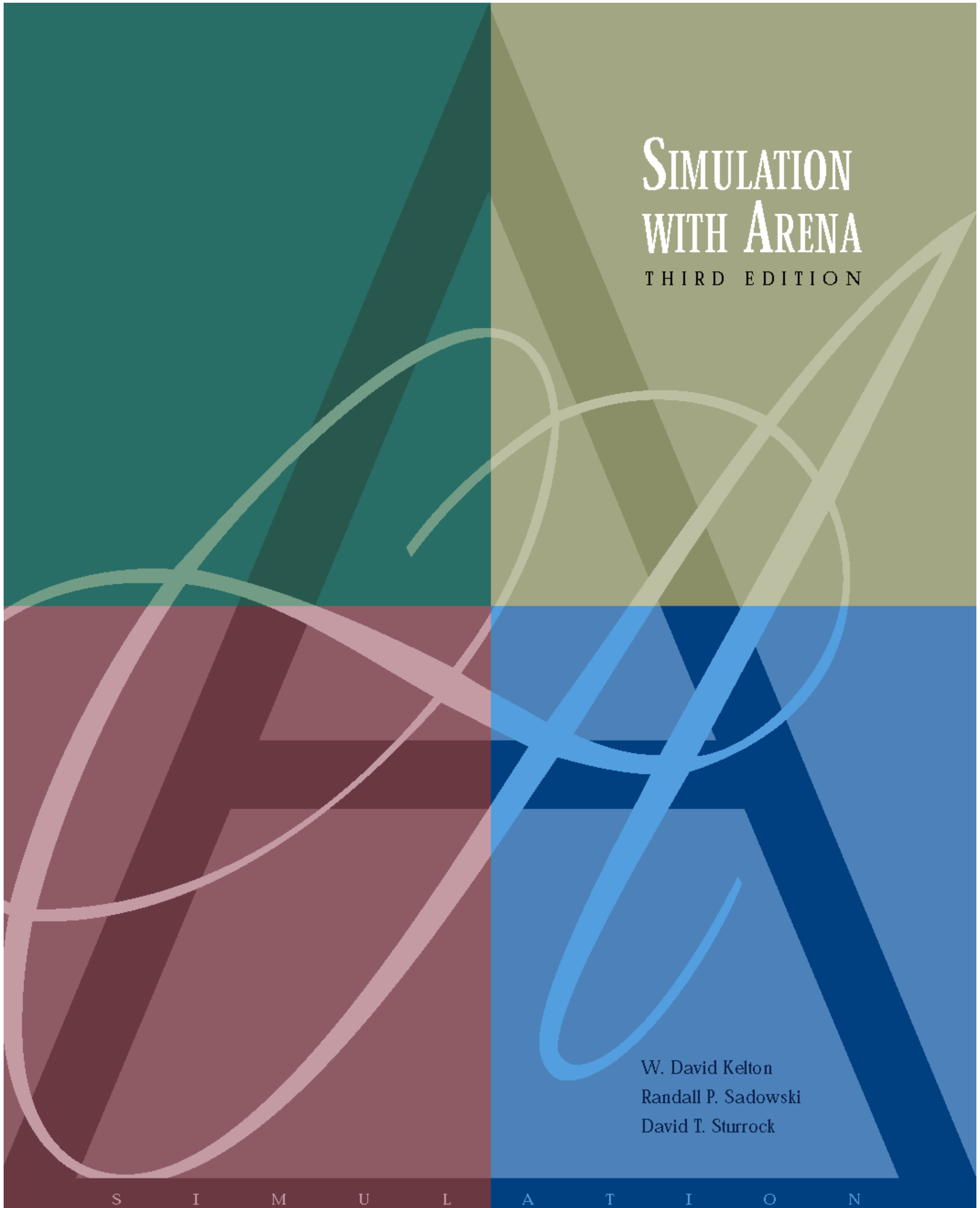
- **Define one-way lengths (in feet) of segments**
- **Segment data module, Advanced Transfer panel**
  - Name = **`Loop Conveyor.Segment`**
  - Beginning Station = **`Order Release`**
  - Next Stations button
    - Name Next Station in correct sequence
    - Give distance (in feet) to this next station
- **Segment animation**
  - Put Station markers in front of each Resource picture
  - Segment button  , Animate Transfer toolbar
  - Dialog, crosshairs, clicking just like Distances for Transporters
    - Except here, have to place only 6 Segment animations

Conveyor Statistics

- ***Run > Setup > Project Parameters* to check Conveyor Statistics**
- **Get percent of time blocked (stopped)**
- **Utilization statistic is average percent of space occupied on conveyor (not percent of time that a part was on the conveyor)**
- **To see conveyor stop (it's nonaccumulating) more clearly, change `Load Time` and `Unload Time` to much greater values than `0.25`**
  - Could do this during run with VBA (Chapter 10), or Run Controller – see text for details – but makes output statistics nearly impossible to interpret

Model 8-5: Change Conveyors to Accumulating

- **Conveyor module**
  - Change Conveyor Type to Accumulating
  - Accumulation Length = 4 (in feet), amount of space the accumulated parts need on the conveyor
- **Running, see very little accumulation in animation**
  - To see more, increase **`Load Time`** and **`Unload Time`**



Statistical Analysis of Output from Terminating Simulations

Chapter 6

Last revision June 8, 2003

What is required in this chapter is what are the software's associated with arena for output analysis, every software and what it its used for and
Important keywords in it such as controls scenarios, constraints objectives etc. every terminology

And how to used it step by step

Terminating System → يبدأ و ينتهي ال system بنفس ال initial conditions

Steady State Systems → Doesn't go to the initial condition it started with

warm up period → الفترة يلي ما بجمع فيها statistical output

What We'll Do ...

- Time frame of simulations
- Strategy for data collection and analysis
- Confidence intervals
  - point estimate
  - interval estimate
- Comparing two alternatives
- Comparing many alternatives via the Arena Process Analyzer (PAN)
- Searching for an optimal alternative with OptQuest

if we determined a Warm Up period it will be done before every Replication one Replicate يجب عدم اعتماد

Introduction

- **Random input leads to random output (RIRO)**
- **Run a simulation (once) — what does it mean?**
  - Was this run “typical” or not?
  - Variability from run to run (of the same model)?
- **Need statistical analysis of output data**
  - From a single model configuration
  - Compare two or more different configurations
  - Search for an optimal configuration

extremely important to determine the effect of randomness in general

- **Statistical analysis of output is often ignored**
  - This is a big mistake – no idea of precision of results
  - Not hard or time-consuming to do this – it just takes a little planning and thought, then some (cheap) computer time

في برنامج اسمه SPSS
ممكن برضه يعمل هاي الأشياء

Time Frame of Simulations

- ***Terminating*: Specific starting, stopping conditions** مثلاً أسبوع و بيتكرر
  - Run length will be well-defined (and finite)

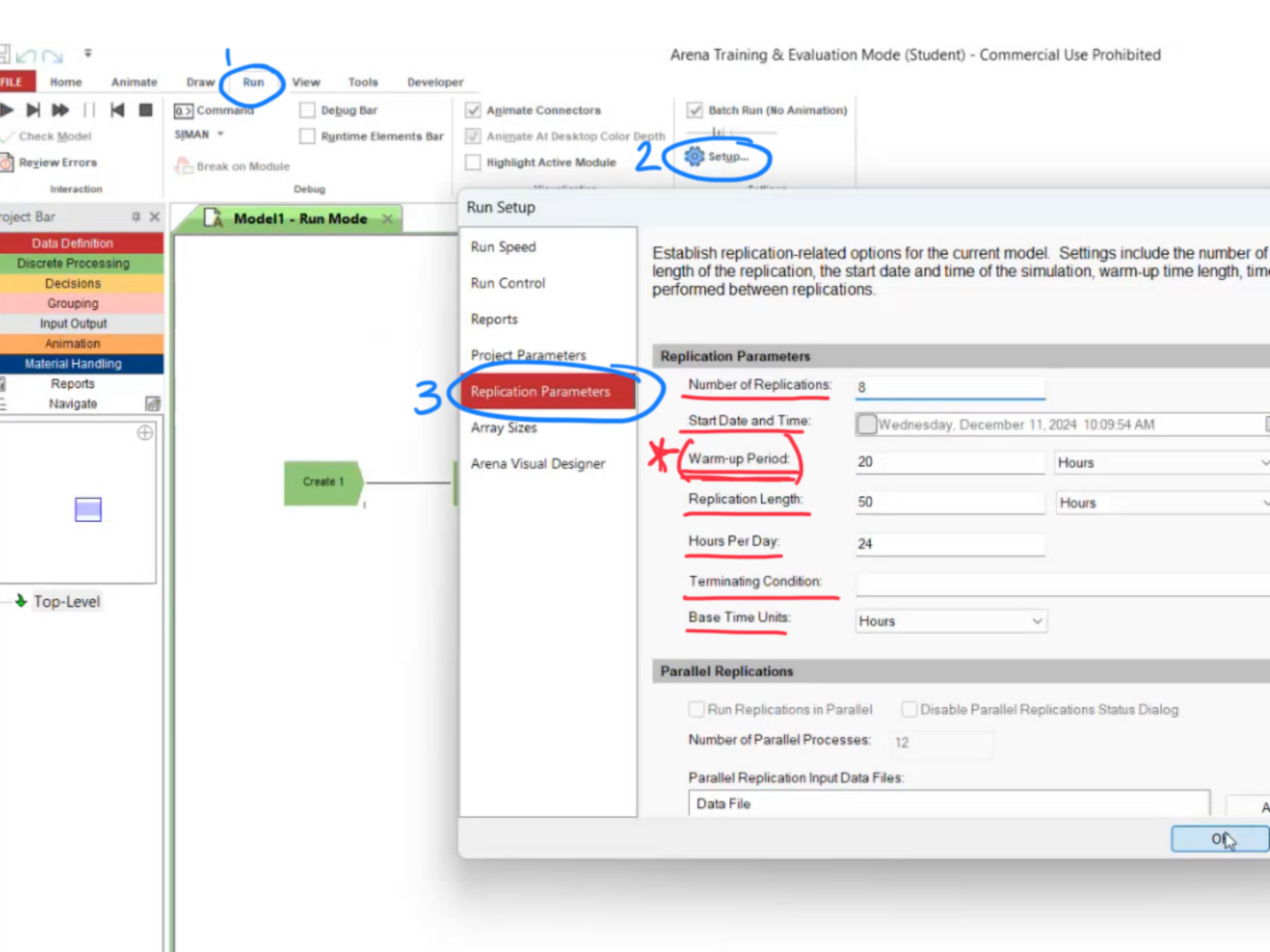
مثلاً اذا انا مهتمة بفترة (سنة)
بنفذ (سنة) وبضيف عليها
بالبداية مثلاً فترة ثلاث شهور
warm up period

- ***Steady-state*: Long-run (technically forever)**

عشان اتخلص منها بحط warm up period

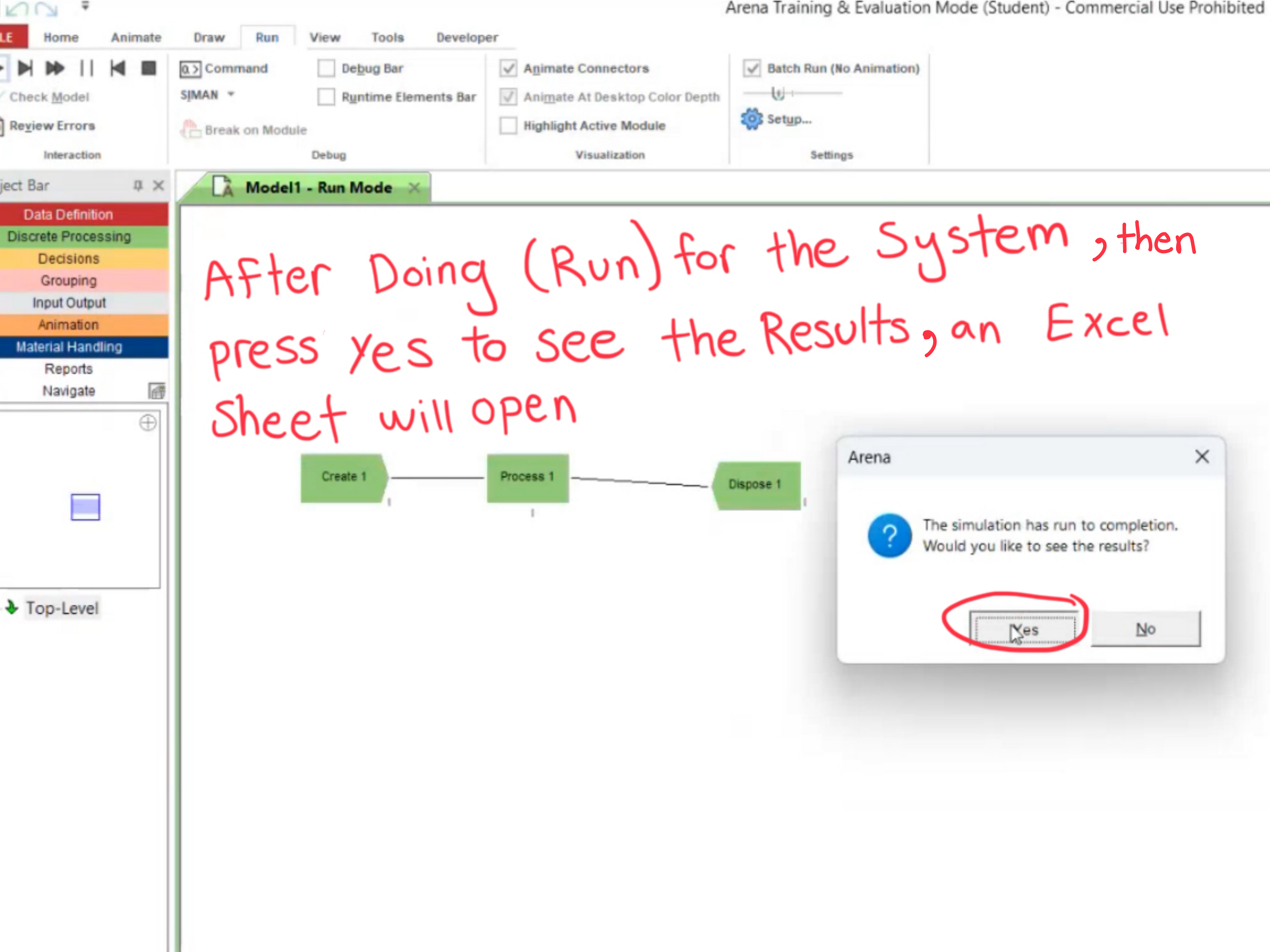
بسوي هيك لكل سنة عشان ما تصير ال Data Correlated

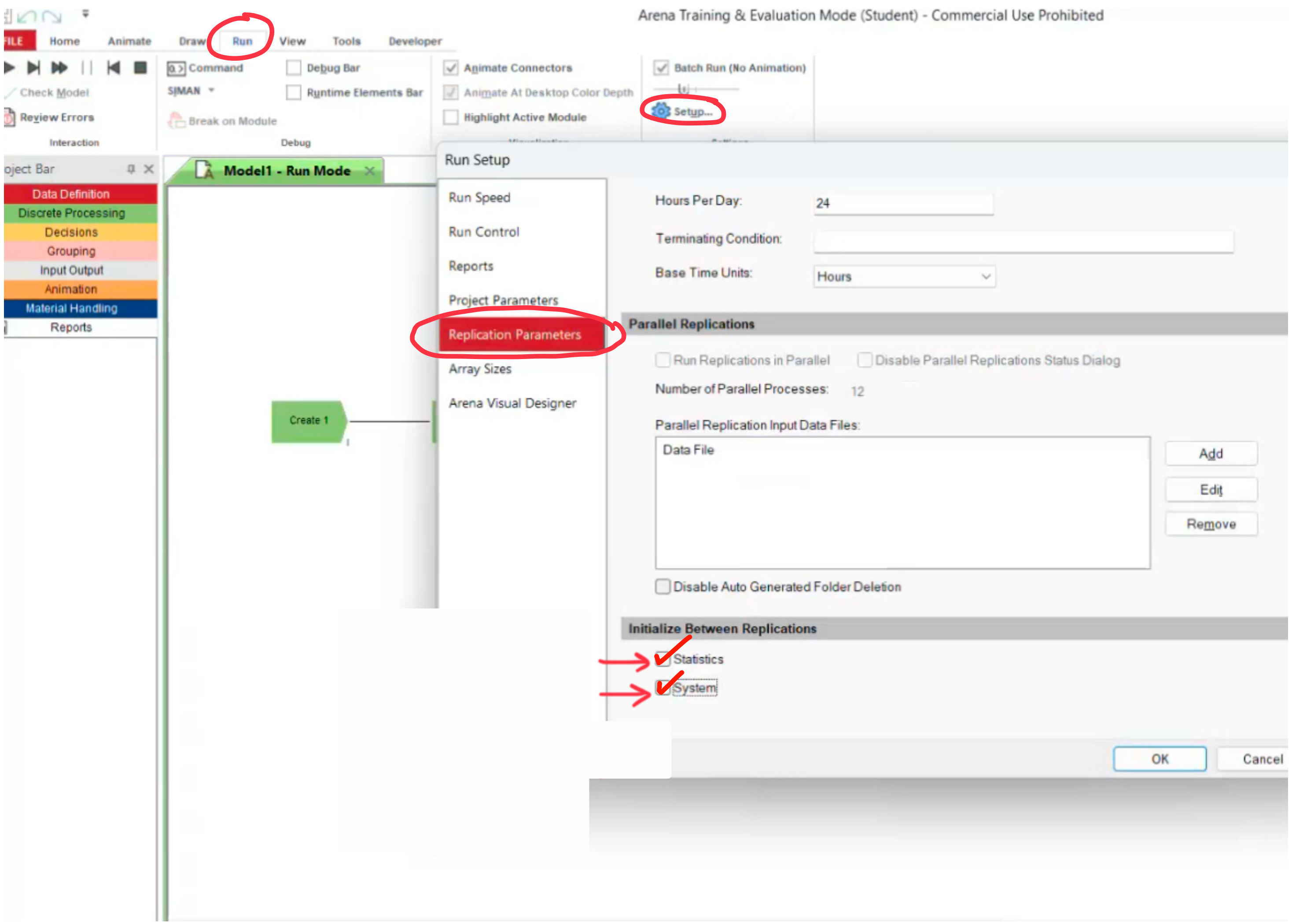
  - Theoretically, initial conditions don’t matter (but practically they usually do)
  - Not clear how to terminate a simulation run
- **This is really a question of intent of the study**
- **Has major impact on how output analysis is done**
- **Sometimes it’s not clear which is appropriate**
- **Here: Terminating (steady-state in Section 7.2)**



Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
FILE
Home
Animate
Draw
Run
View
Tools
Developer
1
Check Model
Review Errors
Interaction
Command
SIMAN
Break on Module
Debug Bar
Runtime Elements Bar
Debug
Animate Connectors
Animate At Desktop Color Depth
Highlight Active Module
Batch Run (No Animation)
2
Setup...
Project Bar
Data Definition
Discrete Processing
Decisions
Grouping
Input Output
Animation
Material Handling
Reports
Navigate
Top-Level
Model1 - Run Mode
Create 1
Run Setup
Run Speed
Run Control
Reports
Project Parameters
3
Replication Parameters
Array Sizes
Arena Visual Designer
Establish replication-related options for the current model. Settings include the number of
length of the replication, the start date and time of the simulation, warm-up time length, time
performed between replications.
Replication Parameters
Number of Replications: 8
Start Date and Time: Wednesday, December 11, 2024 10:09:54 AM
Warm-up Period: 20 Hours
Replication Length: 50 Hours
Hours Per Day: 24
Terminating Condition:
Base Time Units: Hours
Parallel Replications
Run Replications in Parallel
Disable Parallel Replications Status Dialog
Number of Parallel Processes: 12
Parallel Replication Input Data Files:
Data File
OK

After Doing (Run) for the System, then press yes to see the Results, an Excel Sheet will open

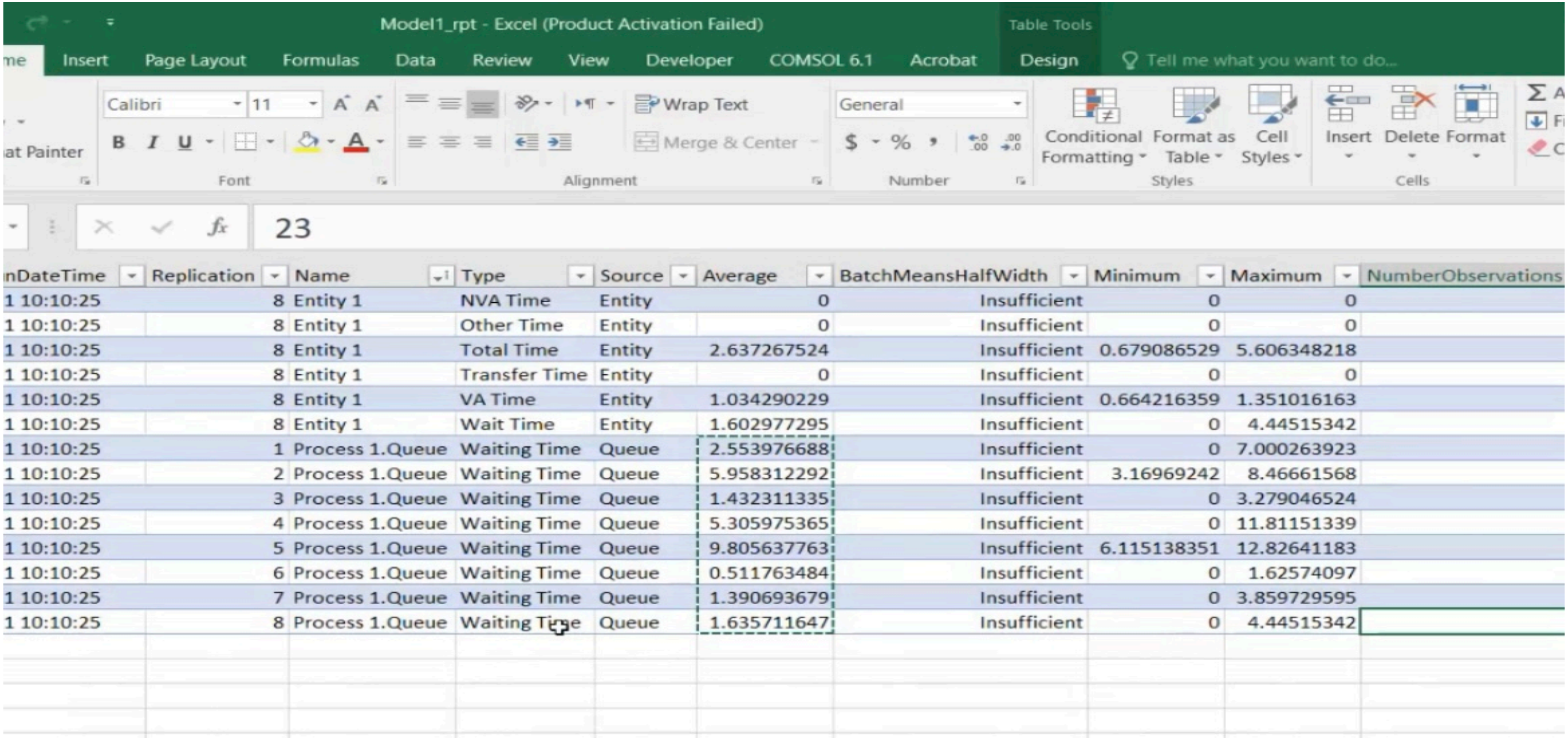



Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Check Model
Review Errors
Interaction
Command
SIMAN
Break on Module
Debug Bar
Runtime Elements Bar
Debug
Animate Connectors
Animate At Desktop Color Depth
Highlight Active Module
Batch Run (No Animation)
Setup...
oject Bar
Data Definition
Discrete Processing
Decisions
Grouping
Input Output
Animation
Material Handling
Reports
Model1 - Run Mode
Create 1
Run Setup
Run Speed
Run Control
Reports
Project Parameters
Replication Parameters
Array Sizes
Arena Visual Designer
Hours Per Day:
24
Terminating Condition:
Base Time Units:
Hours
Parallel Replications
Run Replications in Parallel
Disable Parallel Replications Status Dialog
Number of Parallel Processes:
12
Parallel Replication Input Data Files:
Data File
Add
Edit
Remove
Disable Auto Generated Folder Deletion
Initialize Between Replications
Statistics
System
OK
Cancel

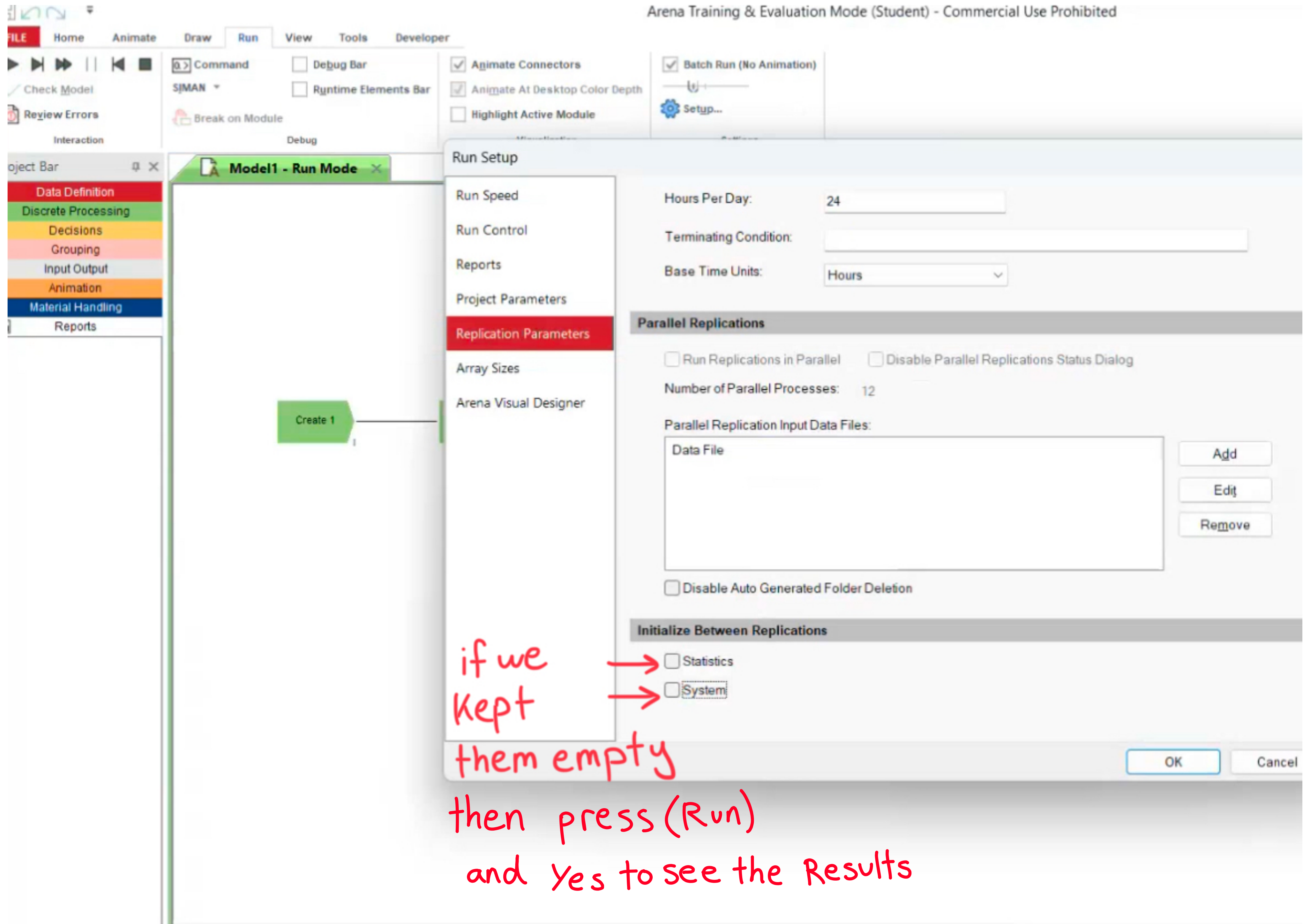
* لما عملنا initialization لل system

← في نهاية كل Replicate اذا ضل شوي من القطع بصفرهم و بنفذ كمان مرة

فأول Replicate وآخر Replicate ببلشو من الصفر



nDateTime	Replication	Name	Type	Source	Average	BatchMeansHalfWidth	Minimum	Maximum	NumberObservations
1 10:10:25	8	Entity 1	NVA Time	Entity	0	Insufficient	0	0	
1 10:10:25	8	Entity 1	Other Time	Entity	0	Insufficient	0	0	
1 10:10:25	8	Entity 1	Total Time	Entity	2.637267524	Insufficient	0.679086529	5.606348218	
1 10:10:25	8	Entity 1	Transfer Time	Entity	0	Insufficient	0	0	
1 10:10:25	8	Entity 1	VA Time	Entity	1.034290229	Insufficient	0.664216359	1.351016163	
1 10:10:25	8	Entity 1	Wait Time	Entity	1.602977295	Insufficient	0	4.44515342	
1 10:10:25	1	Process 1.Queue	Waiting Time	Queue	2.553976688	Insufficient	0	7.000263923	
1 10:10:25	2	Process 1.Queue	Waiting Time	Queue	5.958312292	Insufficient	3.16969242	8.46661568	
1 10:10:25	3	Process 1.Queue	Waiting Time	Queue	1.432311335	Insufficient	0	3.279046524	
1 10:10:25	4	Process 1.Queue	Waiting Time	Queue	5.305975365	Insufficient	0	11.81151339	
1 10:10:25	5	Process 1.Queue	Waiting Time	Queue	9.805637763	Insufficient	6.115138351	12.82641183	
1 10:10:25	6	Process 1.Queue	Waiting Time	Queue	0.511763484	Insufficient	0	1.62574097	
1 10:10:25	7	Process 1.Queue	Waiting Time	Queue	1.390693679	Insufficient	0	3.859729595	
1 10:10:25	8	Process 1.Queue	Waiting Time	Queue	1.635711647	Insufficient	0	4.44515342	



if we Kept them empty

then press (Run) and yes to see the Results

lication	Name	Type	Source	Average	BatchMea
5	Entity 1	Wait Time	Entity	5.273185856	
6	Entity 1	NVA Time	Entity	0	
6	Entity 1	Other Time	Entity	0	
6	Entity 1	Total Time	Entity	6.700185049	
6	Entity 1	Transfer Time	Entity	0	
6	Entity 1	VA Time	Entity	0.997524469	
6	Entity 1	Wait Time	Entity	5.70266058	
7	Entity 1	NVA Time	Entity	0	
7	Entity 1	Other Time	Entity	0	
7	Entity 1	Total Time	Entity	6.11504971	
7	Entity 1	Transfer Time	Entity	0	
7	Entity 1	VA Time	Entity	1.003975051	
7	Entity 1	Wait Time	Entity	5.111074658	
8	Entity 1	NVA Time	Entity	0	
8	Entity 1	Other Time	Entity	0	
8	Entity 1	Total Time	Entity	5.758379774	
8	Entity 1	Transfer Time	Entity	0	
8	Entity 1	VA Time	Entity	1.003694315	
8	Entity 1	Wait Time	Entity	4.754685459	
1	Process 1.Queue	Waiting Time	Queue	2.519805681	
2	Process 1.Queue	Waiting Time	Queue	3.321153383	
3	Process 1.Queue	Waiting Time	Queue	4.851126727	
4	Process 1.Queue	Waiting Time	Queue	4.779695775	
5	Process 1.Queue	Waiting Time	Queue	5.285710867	
6	Process 1.Queue	Waiting Time	Queue	5.697260533	
7	Process 1.Queue	Waiting Time	Queue	5.109791215	
8	Process 1.Queue	Waiting Time	Queue	4.747710241	

Strategy for Data Collection and Analysis

- **For terminating case, make IID replications**
  - *Run > Setup > Replication Parameters*: Number of Replications field
    ☑Statistics
    ☑System
  - Check both boxes for Initialize Between Replications
- **Separate results for each replication – Category by Replication report**
  - Model 5-2; Daily Profit, Daily Late Wait Jobs; 10 replications

Replication	Daily Profit	Daily Late Wait Jobs
1	$ 475.43	0.6500
2	525.17	0.6500
3	513.98	6.5500
4	389.42	0.6000
5	513.96	0.7000
6	401.20	1.0500
7	450.52	0.6500
8	388.71	0.9000
9	574.67	0.4000
10	565.81	0.2500

Strategy for Data Collection and Analysis (cont'd.)

- **Category Overview report will have some statistical-analysis results of the output across the replications**
- **How many replications?**
  - Trial and error (now)
  - Approximate number for acceptable precision (below)
  - Sequential sampling (Chapter 12)
- **Turn off animation altogether for max speed**
  - *Run > Run Control > Batch Run (No Animation)*

if we do initialize for the statistics
it will take every replicate alone

if we do initialize for the system
it will start from scratch

Simulation Replication Data

from Simulation setting

- **Project replication parameters**

  معتبر الشهر ٢٠ يوم
  و يلي هو عدد ايام العمل

  - *Run/Setup* dialog – Replication Parameters tab
  - 10 Replications of 20 twelve-hour days each

    one Replication for each month

  - Four options for Initialization Between Replications:
    - Initialize system (yes), initialize statistics (yes): selected
      - 10 independent, identical replications – vehicles not carried over at end of month
      - Reports for each month separately, and aggregated

      مهم
      chapter 6

    - Initialize system (yes), initialize statistics (no)
      - 10 independent, identical replications – vehicles not carried over at end of month
      - Cumulative summary reports (month 1, months 1-2, months 1-3, …, months 1-10)
    - Initialize system (no), initialize statistics (yes)
      - 10 continuous months – vehicles carried over at end of each month
      - Reports are by replication (month)
    - Initialize system (no), initialize statistics (no)
      - 10 continuous months – vehicles carried over at end of each month
      - Cumulative summary reports (month 1, months 1-2, months 1-3, …, months 1-10)

Confidence Intervals for Terminating Systems

- **Using formulas in Chapter 2, viewing the cross-replication summary outputs as the basic data:**

	Daily Profit	Daily Late Wait Jobs
Sample Mean	$ 479.89	0.6400
Sample Standard Deviation	70.17	0.0510
95% Confidence Interval Half Width	50.20	0.1616
Minimum Summary Output Value	388.71	0.2500
Maximum Summary Output Value	574.67	1.0500

- **Possibly most useful part – 95% confidence interval on expected values**
- **This information (except standard deviation) is in Category Overview report**
  - If > 1 replication specified, Arena uses cross-replication data as above
  - Other confidence levels, graphics – Output Analyzer

Half Width and Number of Replications

- **Prefer smaller confidence intervals — *precision***
- **Notation:** $n = \text{no. replications}$

  $\overline{X} = \text{sample mean}$

  $s = \text{sample standard deviation}$

  $t_{n-1,1-\alpha/2} = \text{critical value from } t \text{ tables}$
- **Confidence interval:** $\boxed{\overline{X} \pm t_{n-1,1-\alpha/2} \dfrac{s}{\sqrt{n}}}$
- **Half-width =** $t_{n-1,1-\alpha/2} \dfrac{s}{\sqrt{n}}$

  Want this to be "small," say $\leq h$ where h is prespecified
- **Can't control *t* or *s***
- **Must increase *n* — how much?**

Half Width and Number of Replications

(cont'd.)

- **Set half-width = *h*, solve for** $n = t^2_{n-1,1-\alpha/2} \dfrac{s^2}{h^2}$
- **Not really solved for *n* (*t*, *s* depend on *n*)**
- **Approximation:**
  - Replace *t* by *z*, corresponding normal critical value
  - Pretend that current *s* will hold for larger samples
  - Get $$\boxed{n \cong z^2_{1-\alpha/2} \frac{s^2}{h^2}}$$

    s **= sample standard deviation from "initial" number** n_0 **of replications**
- **Easier but different approximation:**

$$\boxed{n \cong n_0 \frac{h_0^2}{h^2}}$$

h_0 **= half width from "initial" number** n_0 **of replications**

n **grows quadratically as *h* decreases**

Half Width and Number of Replications

(cont'd.)

- **Application to automotive repair shop**
  - From initial 10 replications, 95% half-width on Daily Profit was ±$50.20 ... let's get this down to ±$20 or less
  - First formula: $n \cong 1.96^2(70.17^2/20^2) = 47.3$, so 48
  - Second formula: $n \cong 10(50.20^2/20^2) = 63.0$, so 63
  - Modified Model 5-2 into Model 6-1
    - Checked *Run > Run Control > Batch Run (No Animation)* for speed
    - In *Run > Setup > Replication Parameters*, changed Number of Replications to 100 (conservative based on above)
  - Got 492.63 ± 13.81, satisfying criterion

Interpretation of Confidence Intervals

مهم في حالات التداخل

- **Interval with random (data-dependent) endpoints that's supposed to have stated probability of containing, or covering, the expected valued**
  - "Target" expected value is a fixed, but unknown, number
  - Expected value = average of infinite number of replications
- **Not an interval that contains, say, 95% of the data**
  - That's a *prediction* interval … useful too, but different
- **Usual formulas assume normally-distributed data**
  - Never true in simulation
  - Might be approximately true if output is an average, rather than an extreme
  - Central limit theorem
  - Robustness, coverage, precision – see book (Model 6-2)

Comparing Two Alternatives

- **Usually compare alternative system scenarios, configurations, layouts, sensitivity analysis**
  - For now, just two alternatives ... more later
- **Model 6-3**
  - Model 6-1, but add file **`Daily Profit.dat`** to Statistic module, Output column, **`Daily Profit`** row
  - Saves this output statistic to this file for each replication
  - Two versions
    - *Base case* – all inputs as originally defined
    - *More-bookings case* – Change **`Max Load`** from 24 to 28 hours (allow more bookings per day ... increase utilization, profit? Maybe)

Comparing Two Alternatives (cont'd.)

- **Reasonable but not-quite-right idea**
  - Make confidence intervals on expected outputs from each alternative, see if they overlap
  - Base case:
    492.63 ± 13.81, or [478.82, 506.44]
  - More-bookings case:
    564.53 ± 22.59, or [541.94, 567.12]

    No Overlap
  - But this doesn't allow for a precise, efficient statistical conclusion

الغرض منه Statistical Analysis (التحليل الاحصائي)

Compare Means via the Output Analyzer

عشان افتحه بلاقيه عند فايل تنزيل الـArena وليس جاهز داخلها

- **Output Analyzer is a separate application that operates on .dat files produced by Arena**
  - Launch separately from Windows, not from Arena
- **To save output values (Expressions) of entries in Statistic data module (Type = Output) – enter filename.dat in Output File column**
  - Just did for Daily Profit, not Daily Late Wait Jobs
  - Will overwrite this file name next time … either change the name here or out in Windows before the next run
  - .dat files are binary … can only be read by Output Analyzer

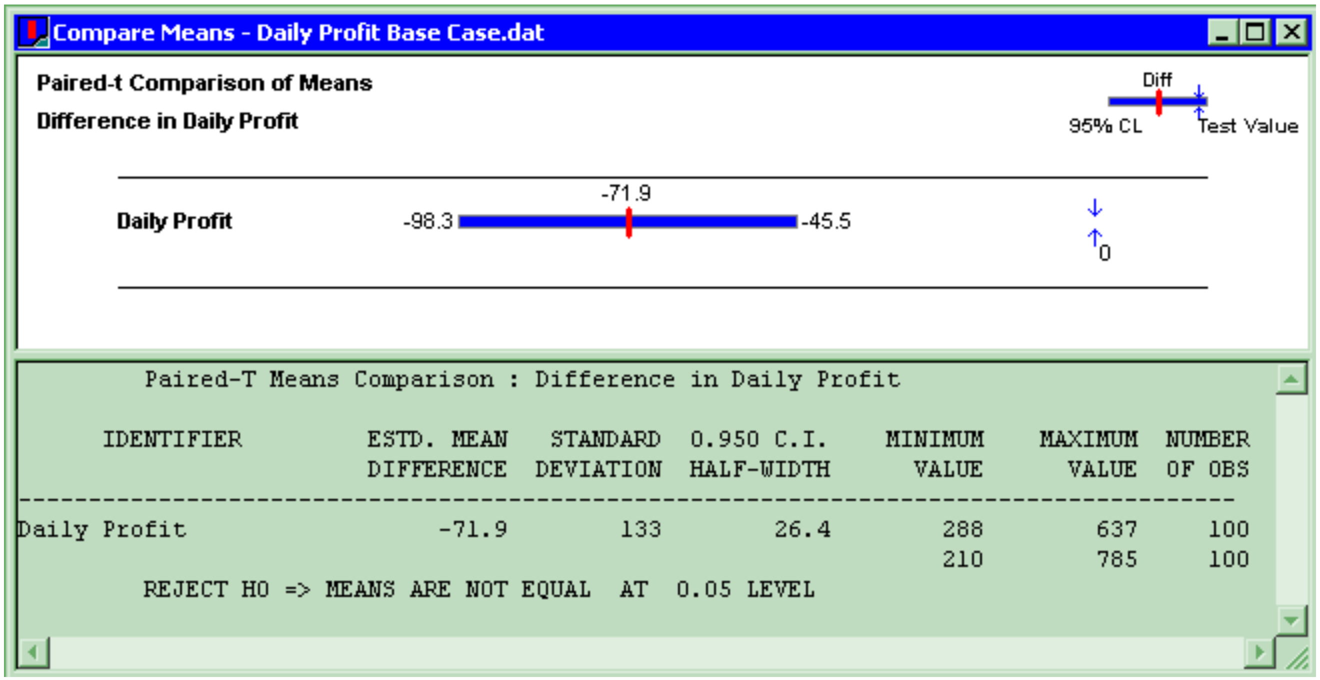
Compare Means via the Output Analyzer
(cont'd.)

- **Start Output Analyzer, open a new data group**
  - Basically, a list of .dat files of current interest
  - Can save data group for later use – .dgr file extension
  - Add button to select (Open) .dat files for the data group
- ***Analyze > Compare Means* menu option**
  - Add data files … "A" and "B" for the two alternatives
  - Select "Lumped" for Replications field
  - Title, confidence level, accept Paired-t Test, Scale Display

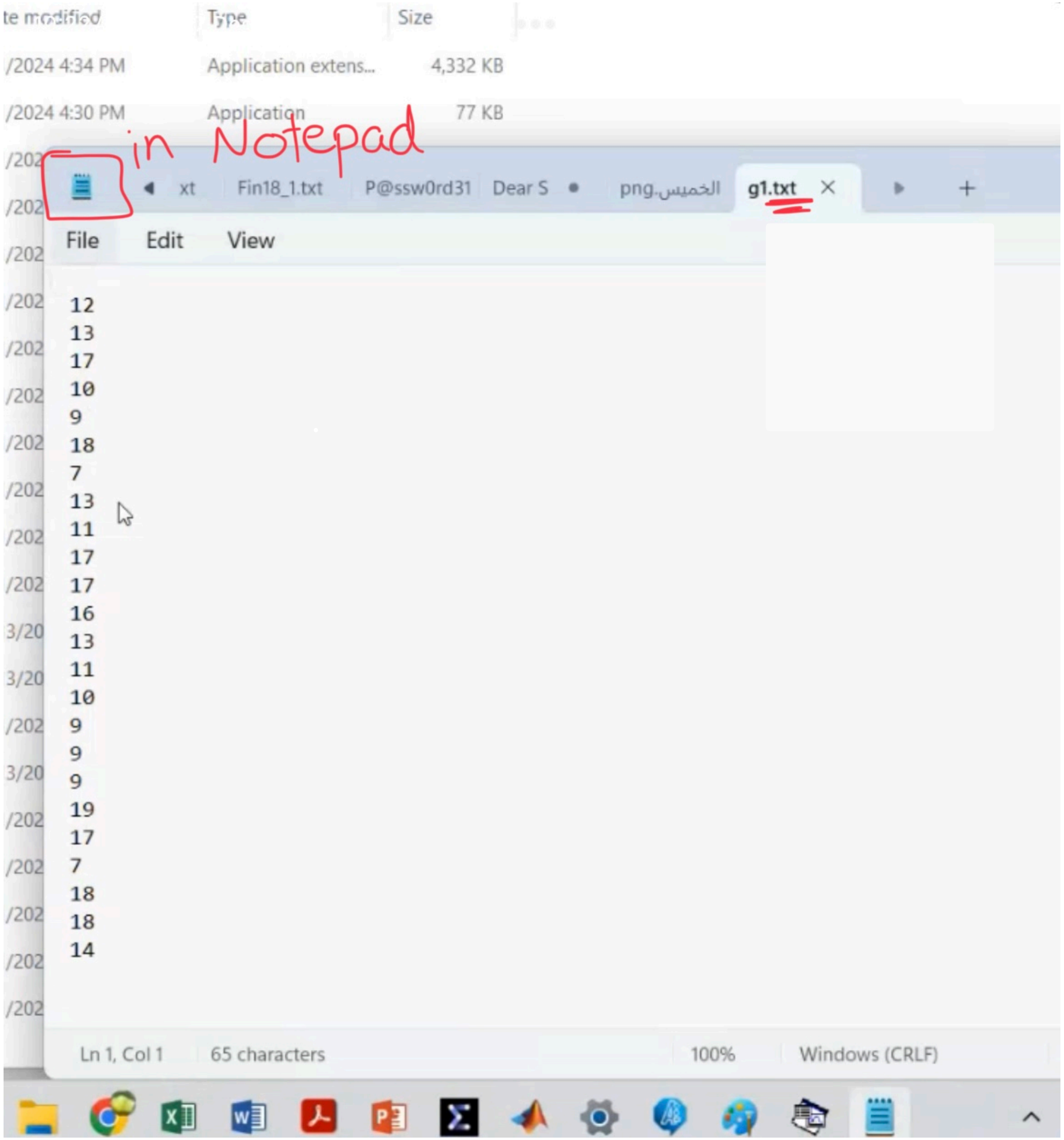
Compare Means via the Output Analyzer
(cont'd.)

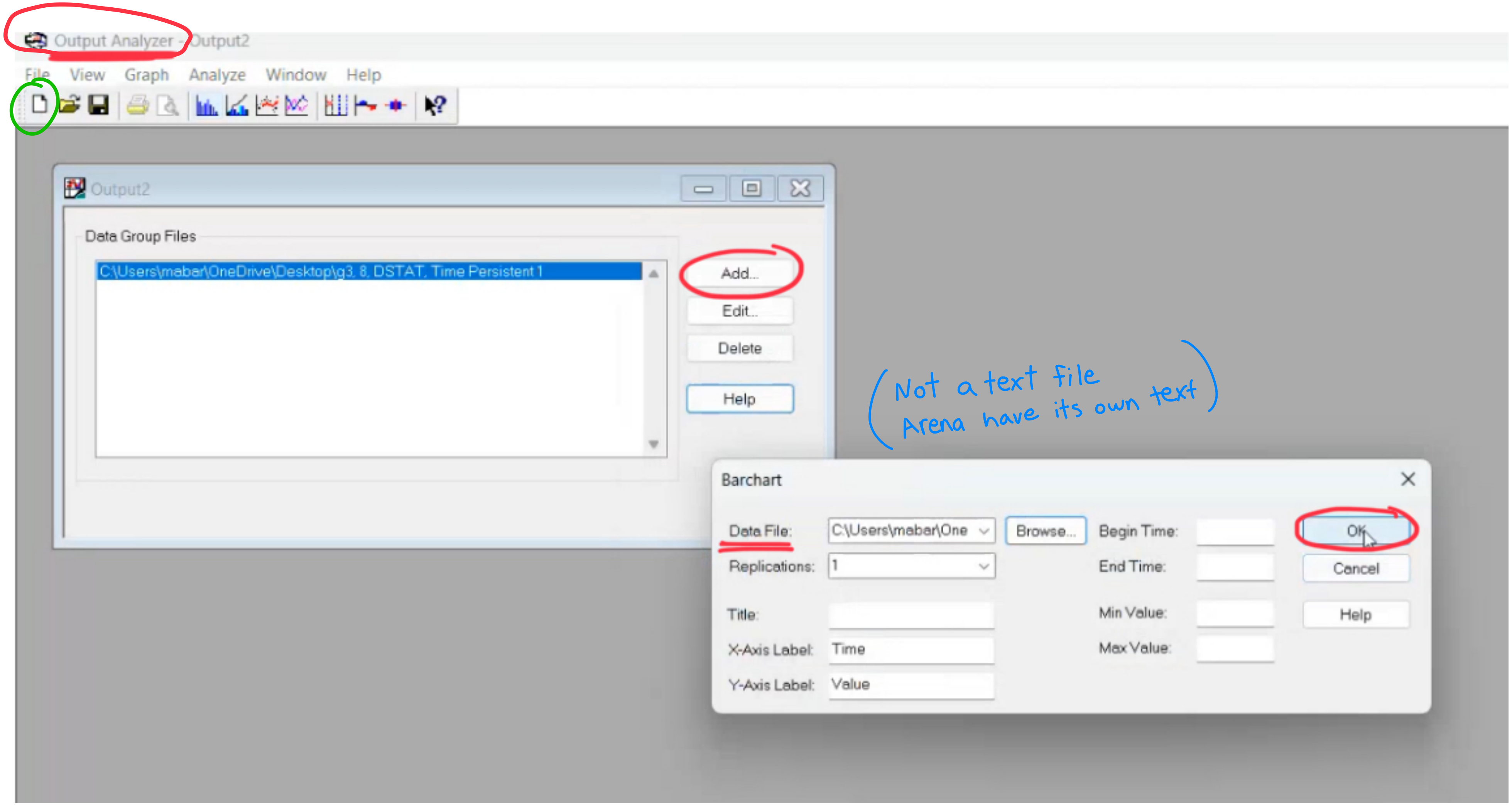
نادراً نستخدمه (برنامج spss أفضل)

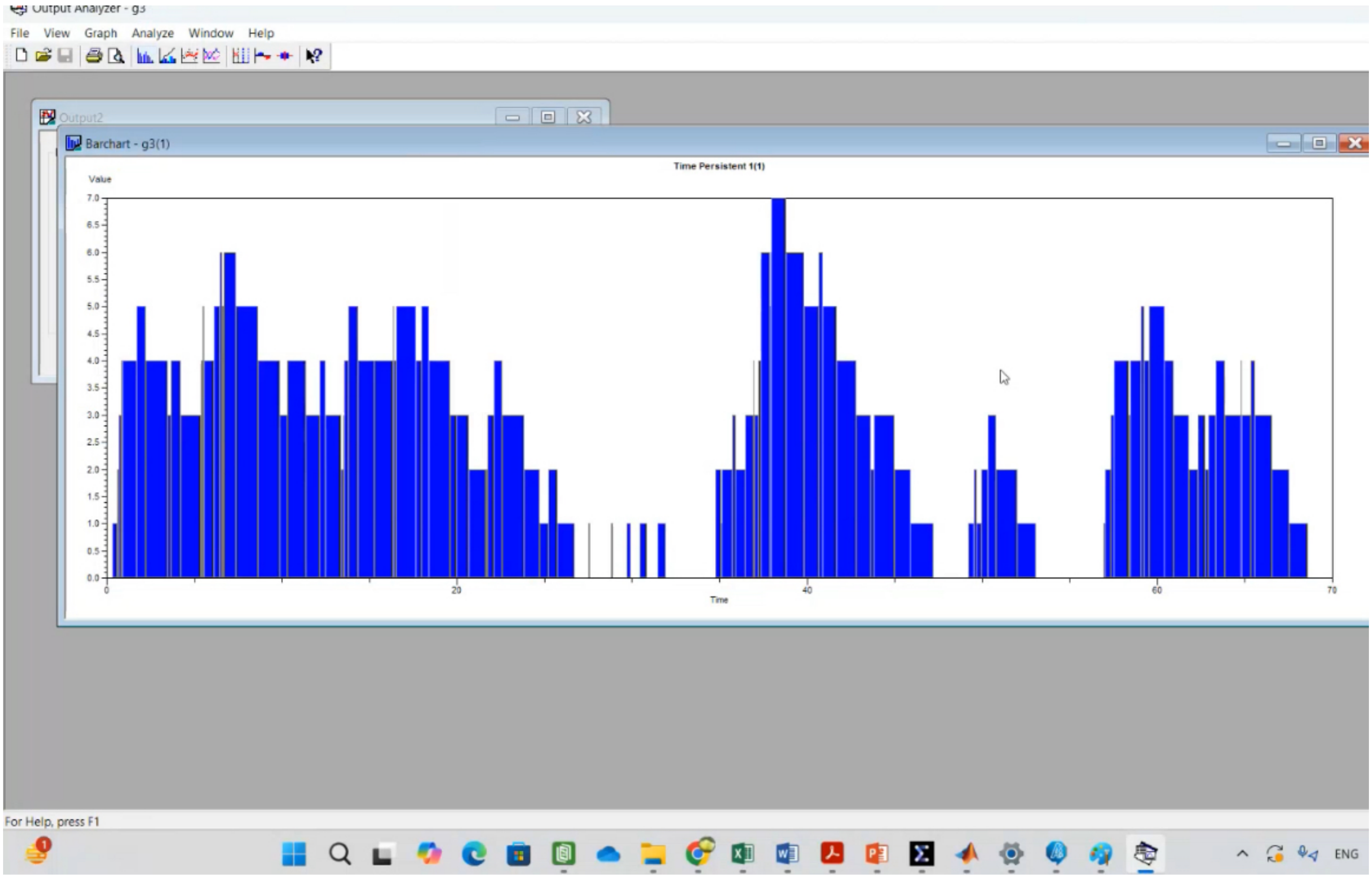
- **Results:**



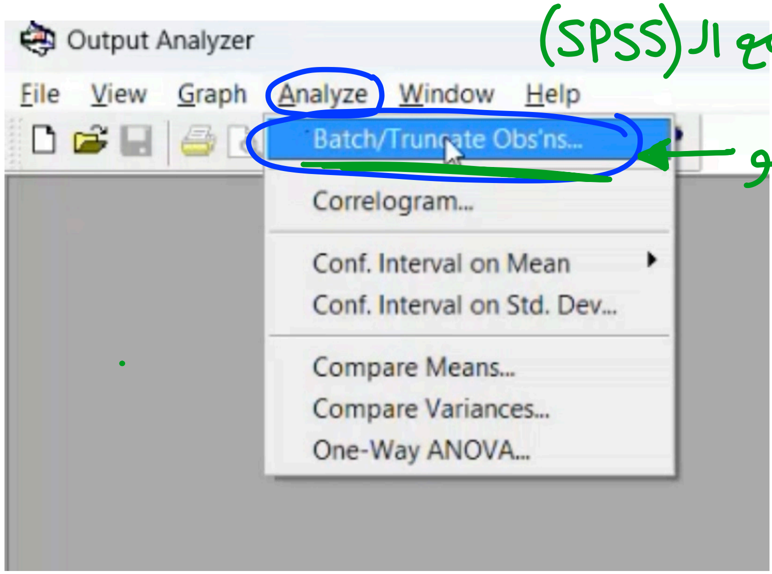
- **Confidence interval on difference misses 0, so conclude that there *is* a (statistically) significant difference**


te modified
Type
Size
/2024 4:34 PM
Application extens...
4,332 KB
/2024 4:30 PM
Application
77 KB
in Notepad
xt
Fin18_1.txt
P@ssw0rd31
Dear S
png.الخميس
g1.txt
File
Edit
View
12
13
17
10
9
18
7
13
11
17
17
16
13
11
10
9
9
9
19
17
7
18
18
14
Ln 1, Col 1
65 characters
100%
Windows (CRLF)


Output Analyzer - Output2
File View Graph Analyze Window Help
Output2
Data Group Files
C:\Users\mabar\OneDrive\Desktop\g3, 8, DSTAT, Time Persistent 1
Add...
Edit...
Delete
Help
Not a text file
Arena have its own text
Barchart
Data File: C:\Users\mabar\One
Browse...
Begin Time:
End Time:
OK
Cancel
Replications: 1
Title:
X-Axis Label: Time
Y-Axis Label: Value
Min Value:
Max Value:
Help

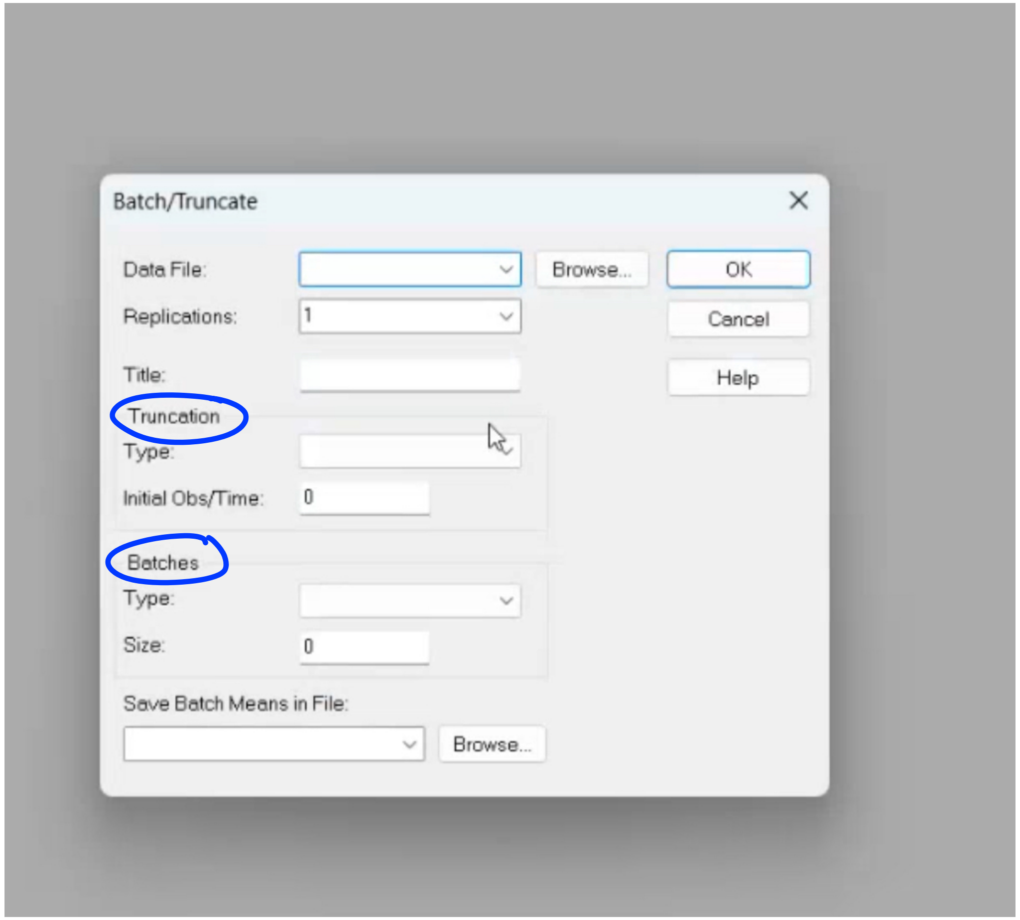

Output Analyzer - g3
File View Graph Analyze Window Help
Output2
Barchart - g3(1)
Time Persistent 1(1)
Value
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
70
Time
For Help, press F1
ENG

يتشابه ال (output Analyzer) مع ال (SPSS)



والفرق الحقيقي هو

if we have an extremely long run we don't need to do initialize for statistics

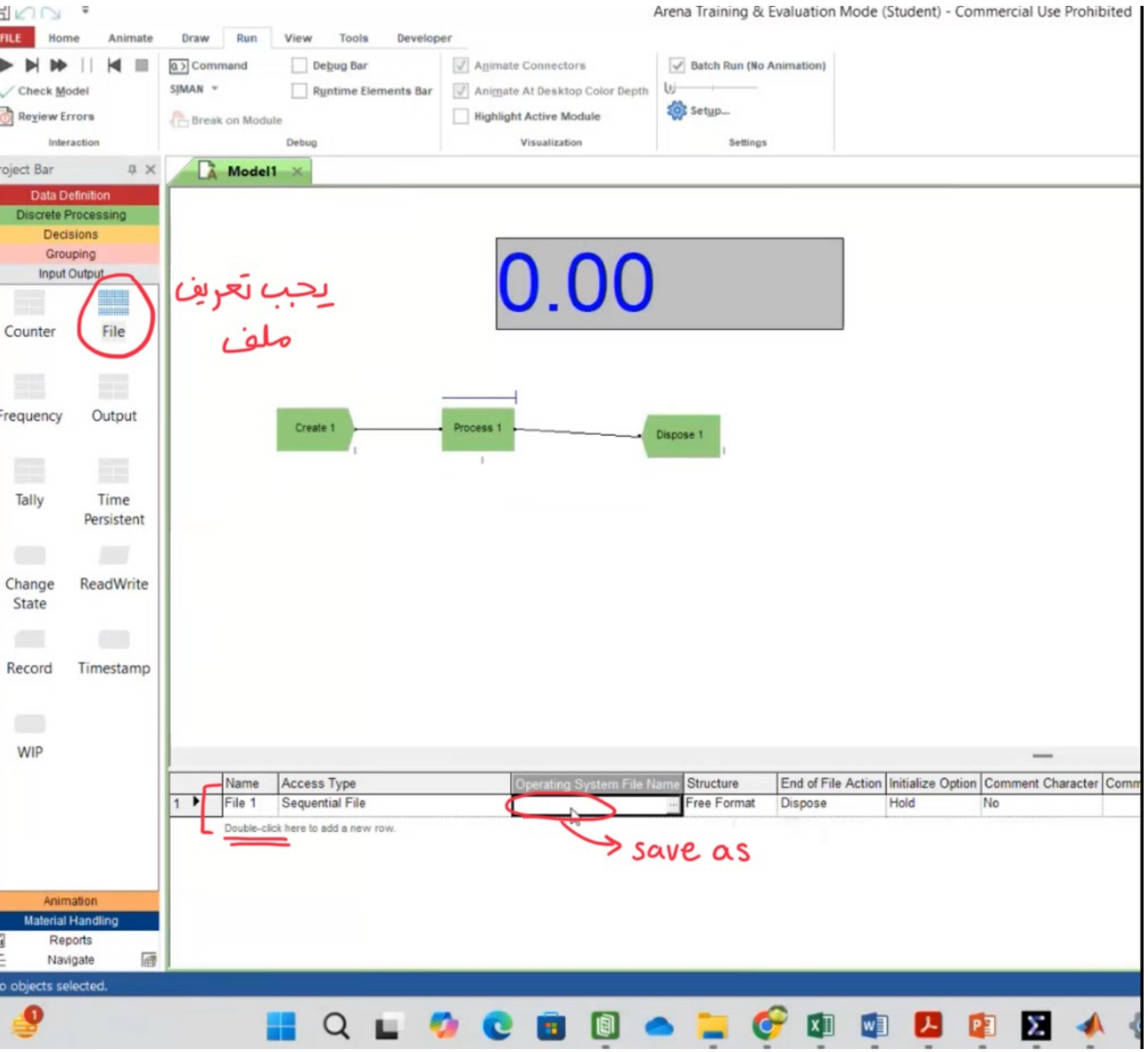


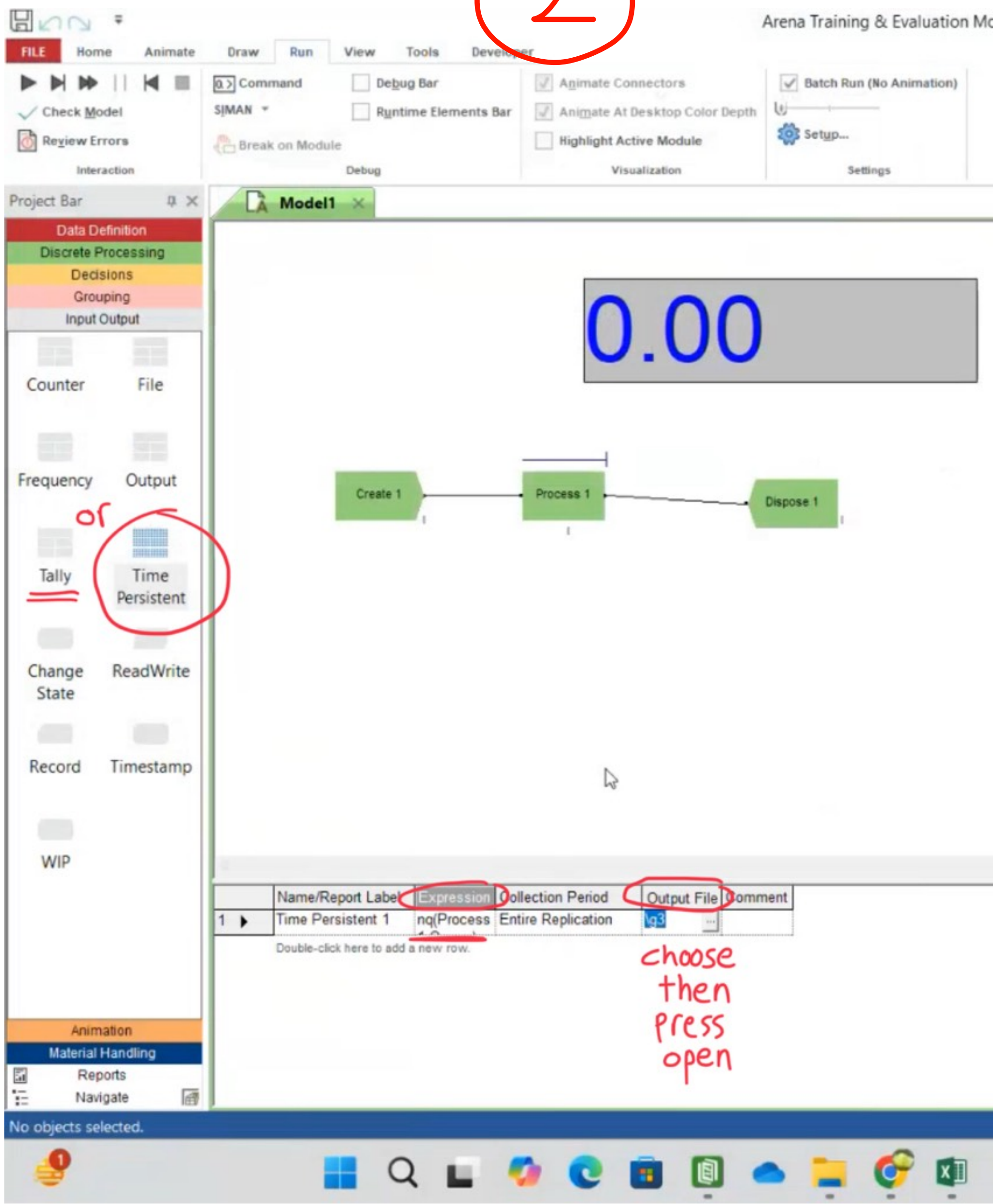
بنقدر نلغي ال warm up period

و في فترة الTruncate Period

و ال batch period

1 اشي ثاني مو تابع للي فوق 2





Evaluating Many Alternatives with the Process Analyzer (PAN)

used to compare Alternatives

- **With (many) more than two alternatives to compare, two problems are**
  - Simple mechanics of making many parameter changes, making many runs, keeping track of many output files
  - Statistical methods for drawing reliable and useful conclusions
- **Process Analyzer (PAN) addresses these**
- **PAN operates on program (.p) files – produced when .doe file is run (or just checked)**
- **Start PAN from Arena (*Tools > Process Analyzer*) or via Windows**
- **PAN runs on its own, separate from Arena**

كل واحد ال Scenarios عنده خيارات مختلفة

PAN Scenarios

is for evaluating many Alternatives

- **A *scenario* in PAN is a combination of:**
  - A program (.p) file
  - Set of input *controls* that you choose
    - Chosen from Variables and Resource capacities – think ahead
    - You fill in specific numerical values
  - Set of output *responses* that you choose
    - Chosen from automatic Arena outputs or your own Variables
    - Values initially empty … to be filled in after run(s)
  - To create a new scenario in PAN, double-click where indicated, get Scenario Properties dialog
    - Specify Name, Tool Tip Text, .p file, controls, responses
    - Values of controls initially as in the model, but *you can change them in PAN* – this is the real utility of PAN
    - Duplicate (right-click, Duplicate) scenarios, then edit for a new one
  - Think of a scenario as a row

Decision Analysis

PAN Projects and Runs

- **A *project* in PAN is a collection of scenarios**
  - Program files can be the same .p file, or .p files from different model .doe files
  - Controls, responses can be the same or differ across scenarios in a project – usually will be mostly the same
  - Think of a project as a collection of scenario rows – a table
  - Can save as a PAN (.pan extension) file
- **Select scenarios in project to run (maybe all)**
- **PAN runs selected models with specified controls**
- **PAN fills in output-response values in table**
  - Equivalent to setting up, running them all "by hand" but much easier, faster, less error-prone

Model 6-4 for PAN Experiments

- **Same as Model 6-3 except remove Output File entry in Statistic module**
  - PAN will keep track of outputs itself, so this is faster
- **Controls – set up a formal 2^3 factorial experiment**

Control (*factor*)	"–" Level	"+" Level
`Max Load`	20	40
`Max Wait`	1	7
`Wait Allowance`	0.5	2.0

2^3 = 8 Scenarios

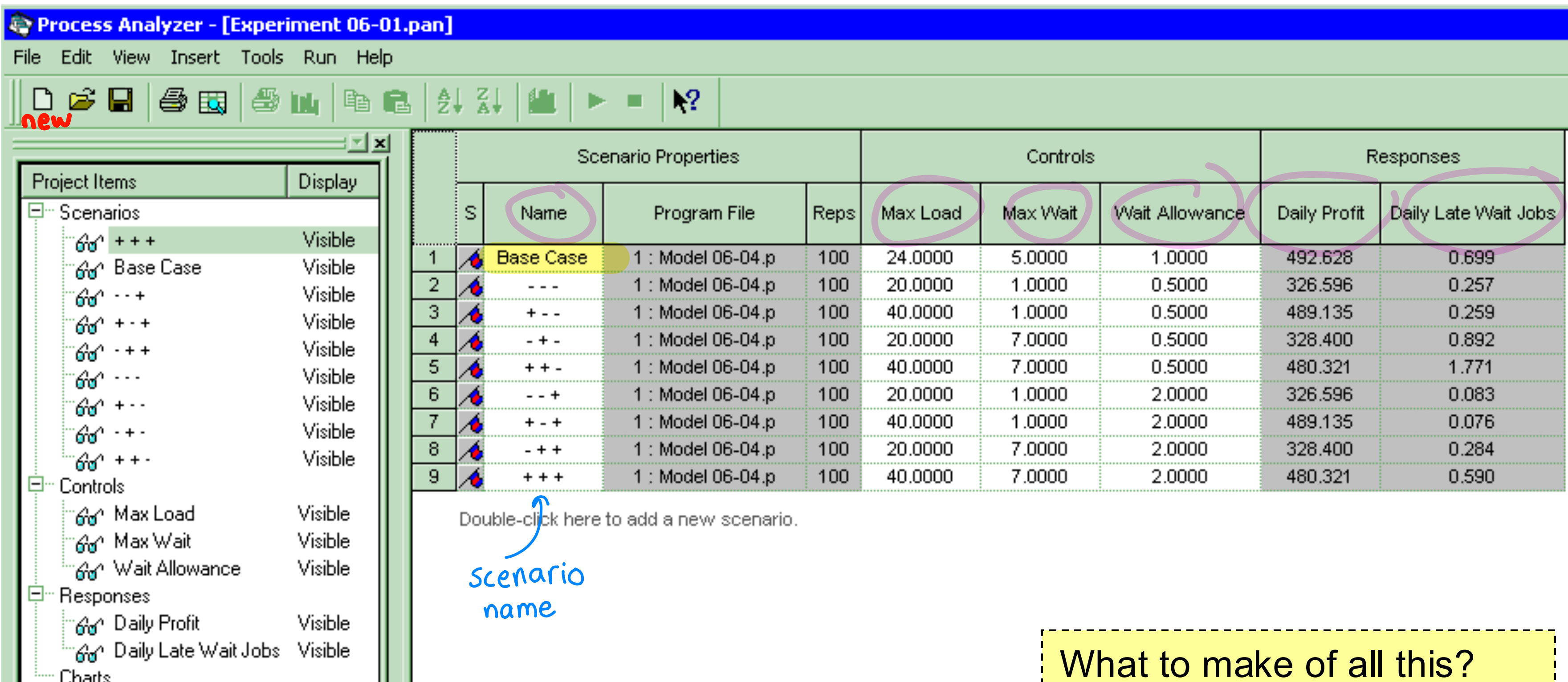
Also do Base Case

- **Responses**
  - `Daily Profit`
  - `Daily Late Wait Jobs`

Not *required* to do a designed experiment with PAN, but this is more informative than haphazard fooling around

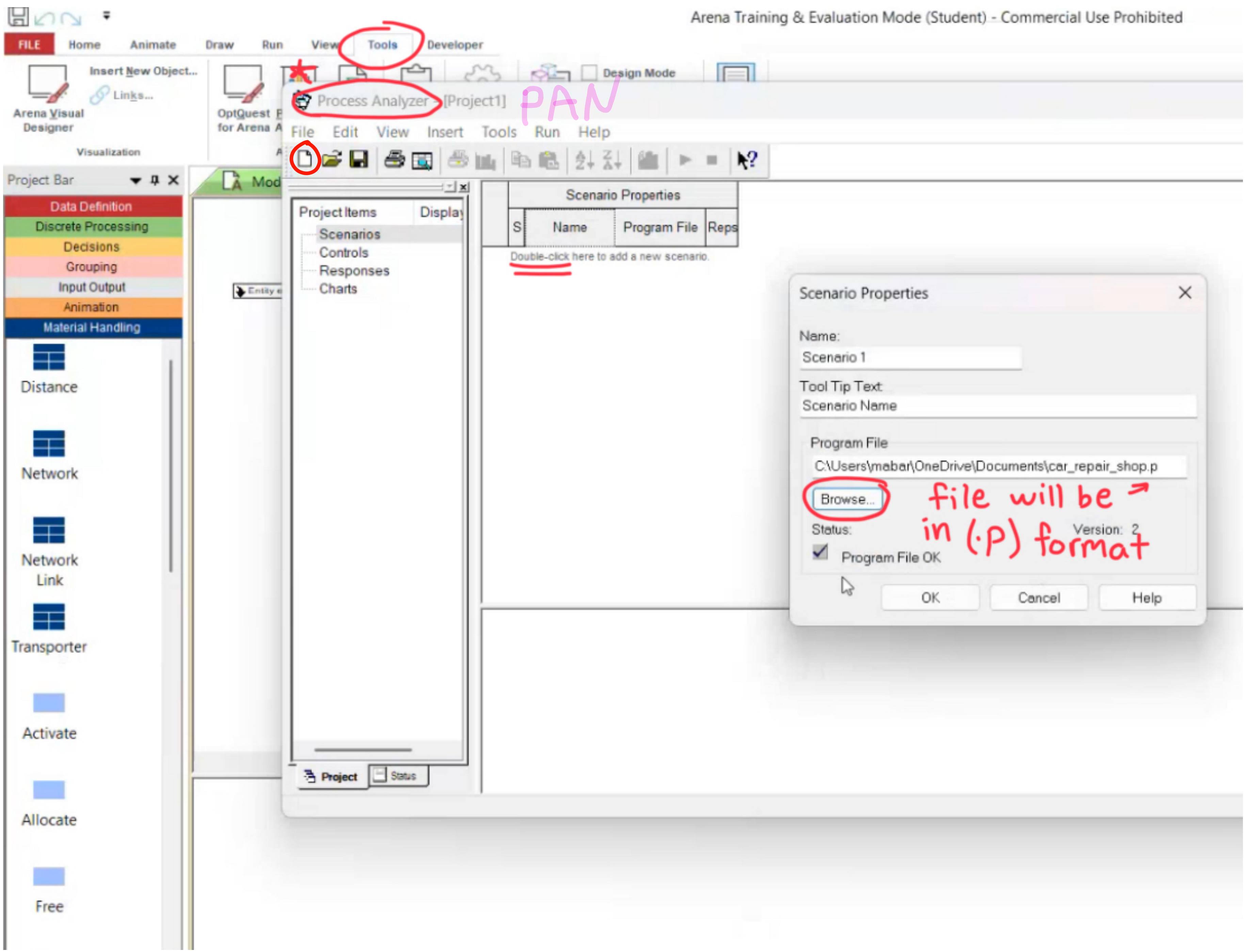
Running Model 6-4 with PAN

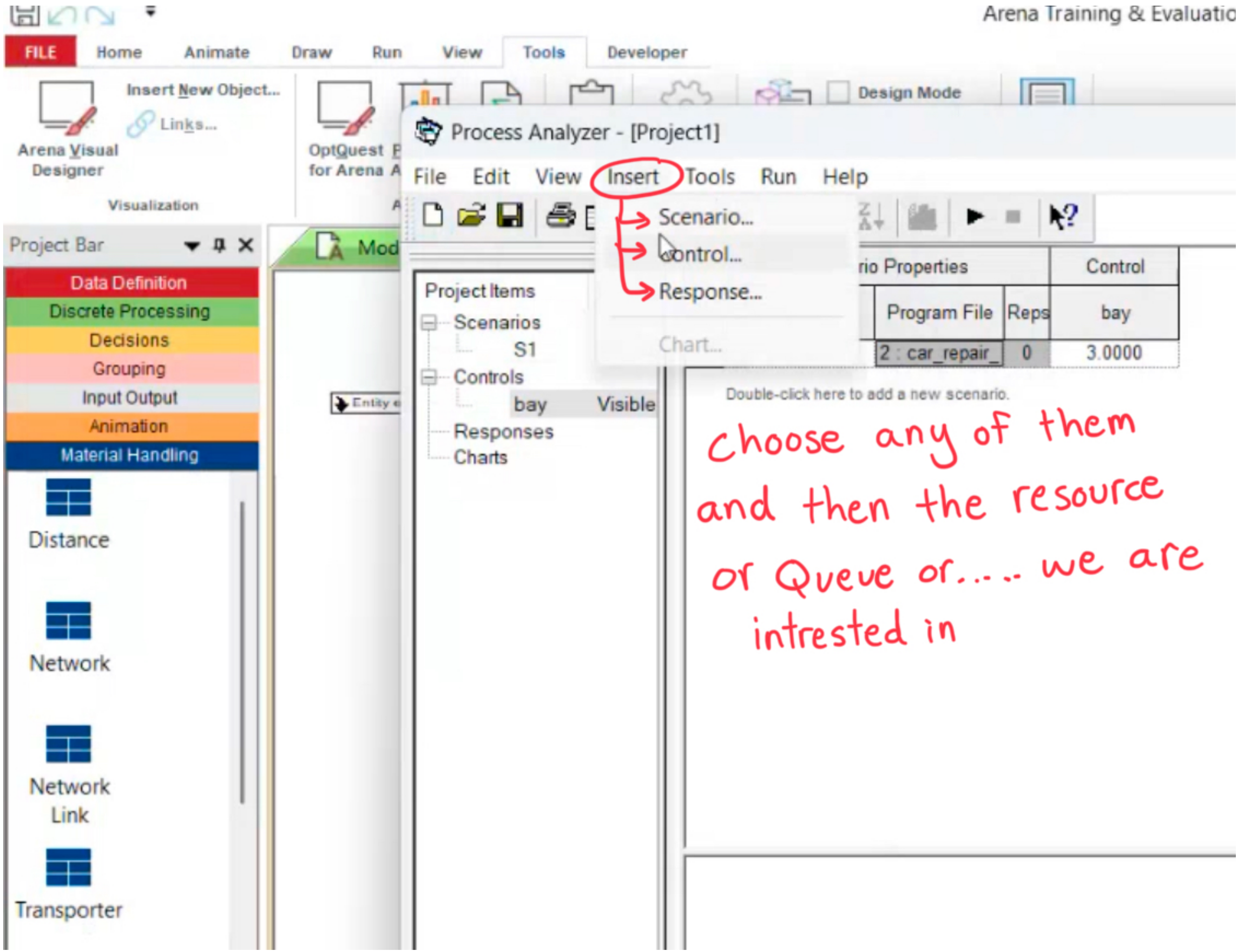
- **Scenarios**
  - Select all to run (click on left of row, Ctrl-Click or Shift-Click for more)
  - To execute, ▶ or *Run* > *Go* or F5

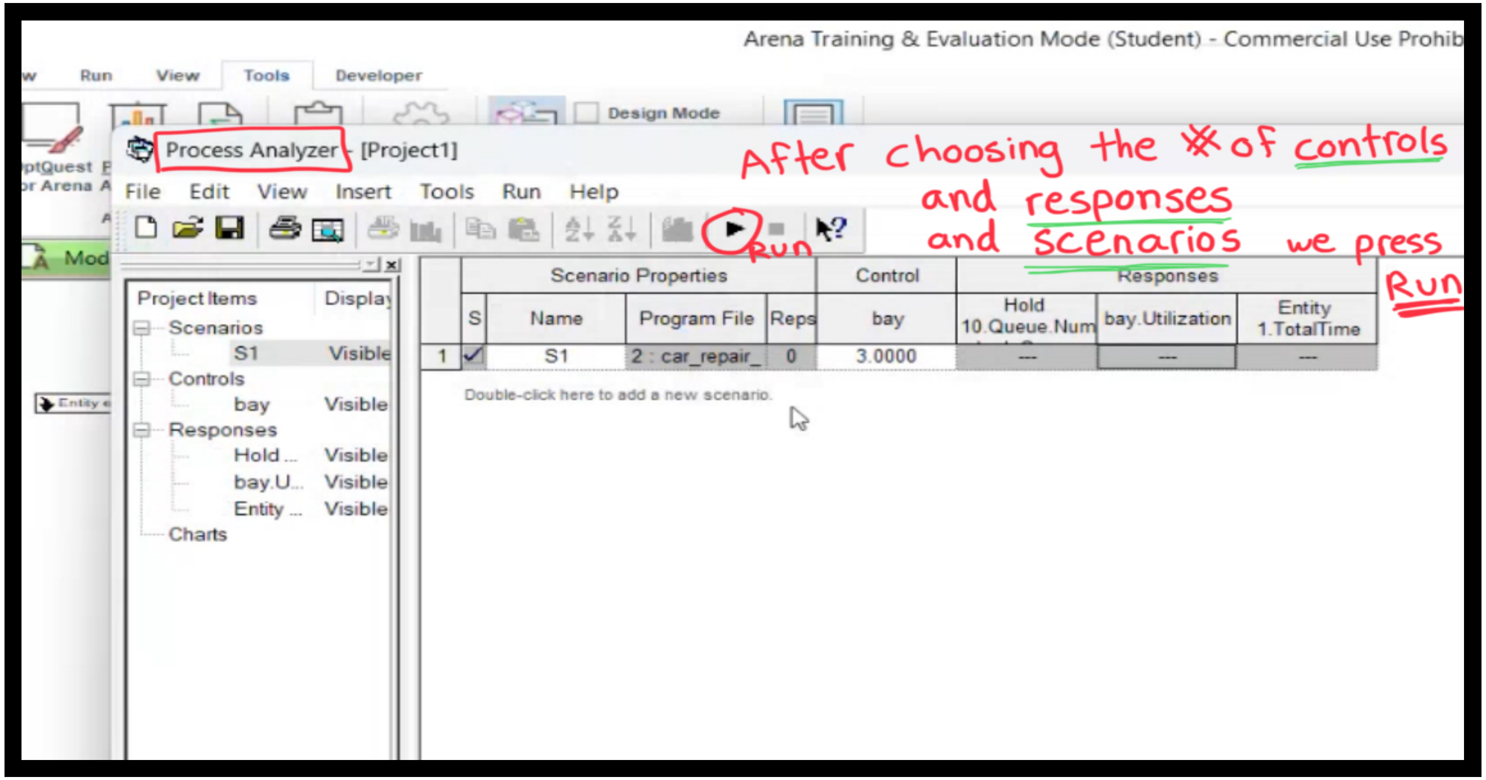


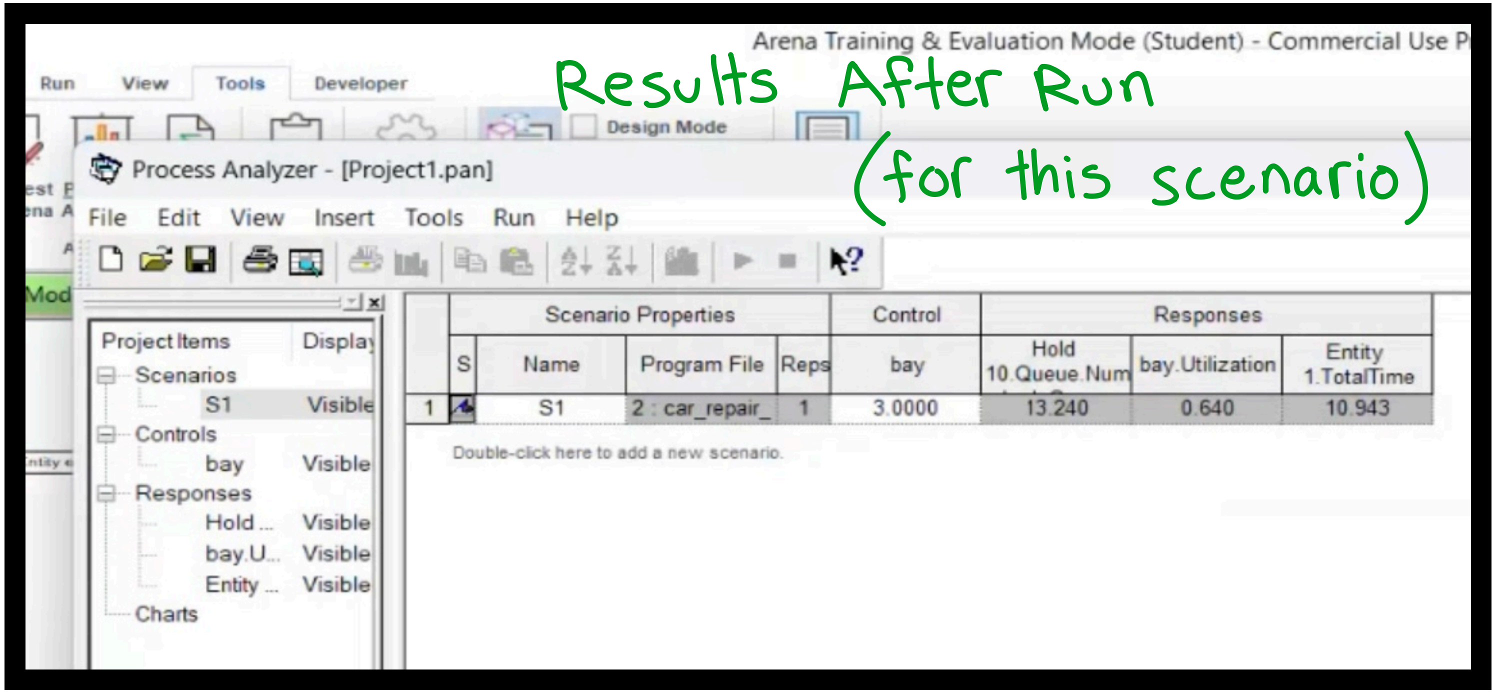
	Scenario Properties				Controls			Responses	
	S	Name	Program File	Reps	Max Load	Max Wait	Wait Allowance	Daily Profit	Daily Late Wait Jobs
1		Base Case	1 : Model 06-04.p	100	24.0000	5.0000	1.0000	492.628	0.699
2		- - -	1 : Model 06-04.p	100	20.0000	1.0000	0.5000	326.596	0.257
3		+ - -	1 : Model 06-04.p	100	40.0000	1.0000	0.5000	489.135	0.259
4		- + -	1 : Model 06-04.p	100	20.0000	7.0000	0.5000	328.400	0.892
5		+ + -	1 : Model 06-04.p	100	40.0000	7.0000	0.5000	480.321	1.771
6		- - +	1 : Model 06-04.p	100	20.0000	1.0000	2.0000	326.596	0.083
7		+ - +	1 : Model 06-04.p	100	40.0000	1.0000	2.0000	489.135	0.076
8		- + +	1 : Model 06-04.p	100	20.0000	7.0000	2.0000	328.400	0.284
9		+ + +	1 : Model 06-04.p	100	40.0000	7.0000	2.0000	480.321	0.590

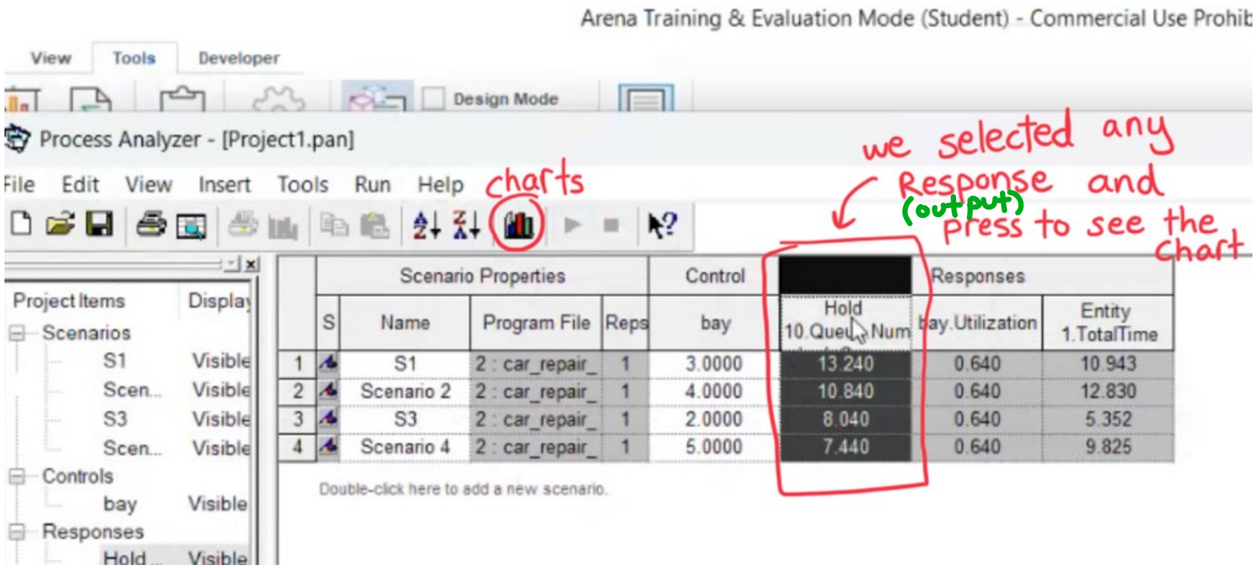
What to make of all this?
Statistical meaningfulness?

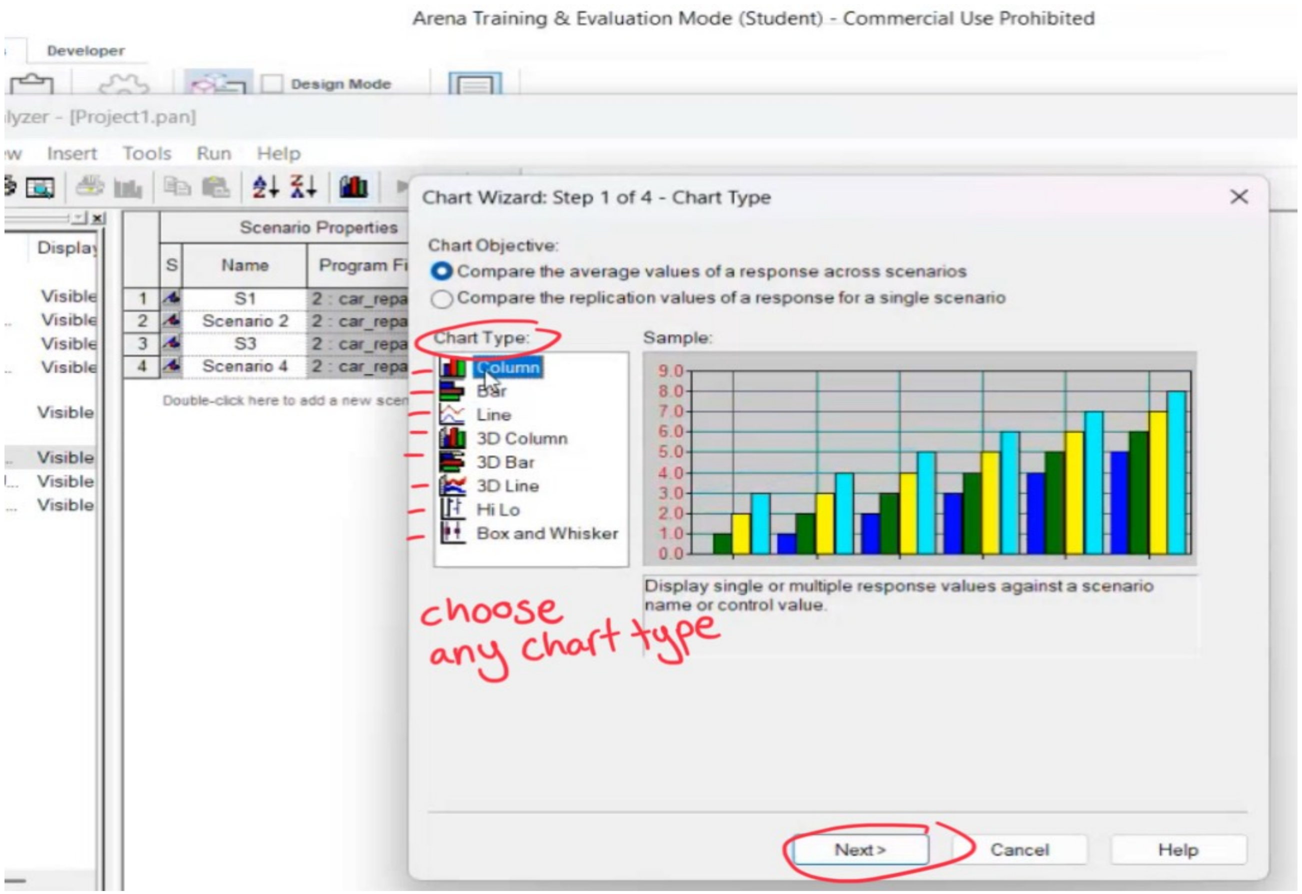

Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Insert New Object...
Links...
Arena Visual Designer
Visualization
OptQuest for Arena
Design Mode
Process Analyzer - [Project1]
PAN
File
Edit
View
Insert
Tools
Run
Help
Project Bar
Data Definition
Discrete Processing
Decisions
Grouping
Input Output
Animation
Material Handling
Distance
Network
Network Link
Transporter
Activate
Allocate
Free
Project Items
Scenarios
Controls
Responses
Charts
Project
Status
Scenario Properties
S
Name
Program File
Reps
Double-click here to add a new scenario.
Scenario Properties
Name:
Scenario 1
Tool Tip Text:
Scenario Name
Program File
C:\Users\mabar\OneDrive\Documents\car_repair_shop.p
Browse...
file will be in (.p) format
Status:
Version: 2
Program File OK
OK
Cancel
Help

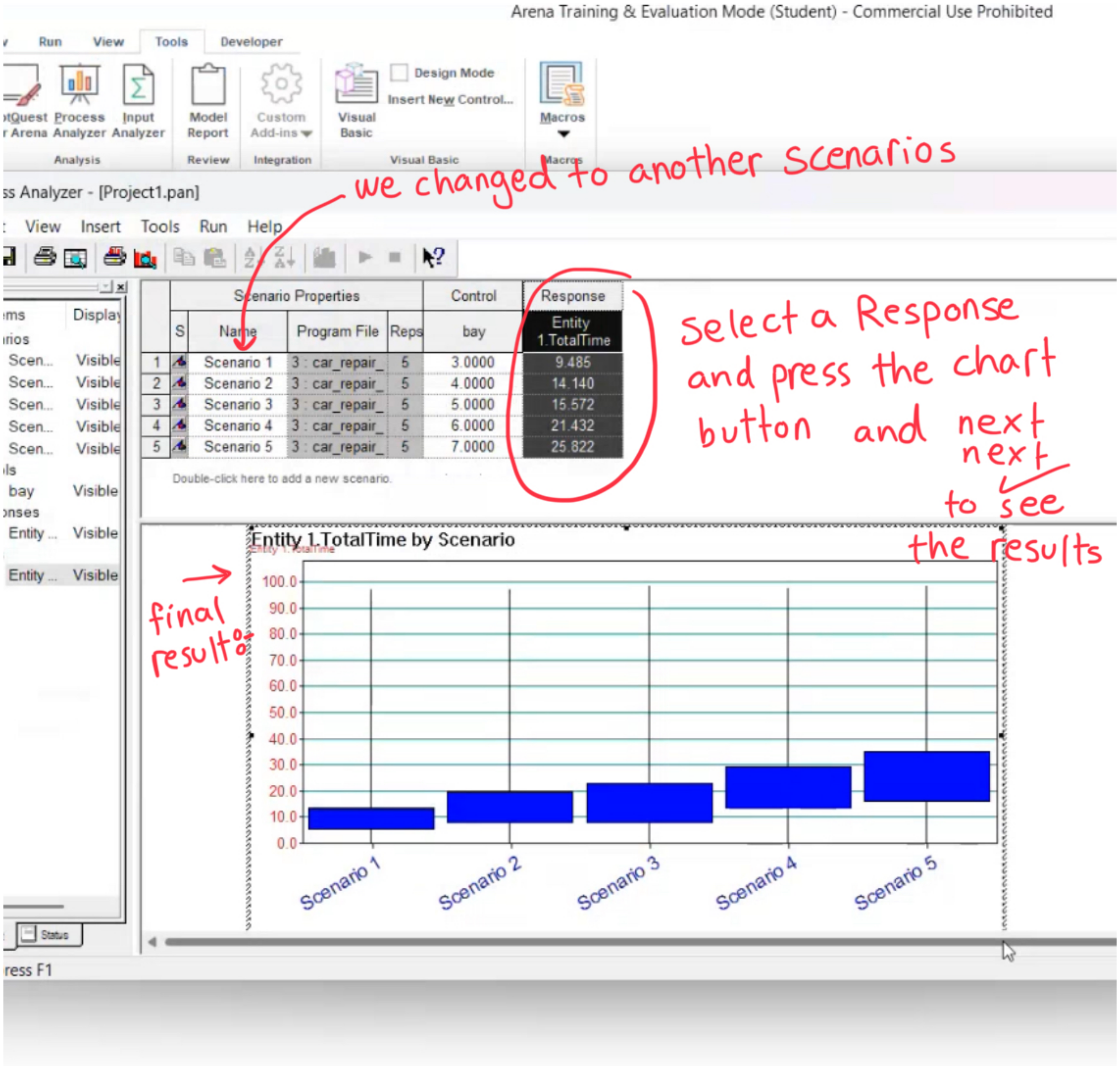

Arena Training & Evaluatio
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Insert New Object...
Links...
Arena Visual Designer
Visualization
OptQuest for Arena
Design Mode
Process Analyzer - [Project1]
File
Edit
View
Insert
Tools
Run
Help
Scenario...
Control...
Response...
Chart...
Project Bar
Data Definition
Discrete Processing
Decisions
Grouping
Input Output
Animation
Material Handling
Distance
Network
Network Link
Transporter
Project Items
Scenarios
S1
Controls
bay
Visible
Responses
Charts
rio Properties
Control
Program File
Reps
bay
2 : car_repair_
0
3.0000
Double-click here to add a new scenario.
choose any of them
and then the resource
or Queue or..... we are
intrested in


Arena Training & Evaluation Mode (Student) - Commercial Use Prohib
Run View Tools Developer
Design Mode
Process Analyzer - [Project1]
File Edit View Insert Tools Run Help
After choosing the ※ of controls and responses and scenarios we press Run
Run
Project Items Display
Scenarios
S1 Visible
Controls
bay Visible
Responses
Hold ... Visible
bay.U... Visible
Entity ... Visible
Charts
Scenario Properties
Control
Responses
S Name Program File Reps bay Hold 10.Queue.Num bay.Utilization Entity 1.TotalTime
1 S1 2 : car_repair_ 0 3.0000 --- --- ---
Double-click here to add a new scenario.


Arena Training & Evaluation Mode (Student) - Commercial Use P
Run View Tools Developer
Design Mode
Results After Run (for this scenario)
Process Analyzer - [Project1.pan]
File Edit View Insert Tools Run Help
Project Items Display
Scenarios
S1 Visible
Controls
bay Visible
Responses
Hold ... Visible
bay.U... Visible
Entity ... Visible
Charts
Scenario Properties
Control
Responses
S Name Program File Reps bay Hold 10.Queue.Num bay.Utilization Entity 1.TotalTime
1 S1 2 : car_repair_ 1 3.0000 13.240 0.640 10.943
Double-click here to add a new scenario.


Arena Training & Evaluation Mode (Student) - Commercial Use Prohib
View Tools Developer
Design Mode
Process Analyzer - [Project1.pan]
File Edit View Insert Tools Run Help
charts
we selected any Response (output) and press to see the chart
Scenario Properties
Control
Responses
Project Items
Display
Scenarios
S1 Visible
Scen... Visible
S3 Visible
Scen... Visible
Controls
bay Visible
Responses
Hold... Visible
S
Name
Program File
Reps
bay
Hold 10.Queue.Num
bay.Utilization
Entity 1.TotalTime
1 S1 2 : car_repair_ 1 3.0000 13.240 0.640 10.943
2 Scenario 2 2 : car_repair_ 1 4.0000 10.840 0.640 12.830
3 S3 2 : car_repair_ 1 2.0000 8.040 0.640 5.352
4 Scenario 4 2 : car_repair_ 1 5.0000 7.440 0.640 9.825
Double-click here to add a new scenario.


Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
Developer
Design Mode
lyzer - [Project1.pan]
Insert Tools Run Help
Scenario Properties
Chart Wizard: Step 1 of 4 - Chart Type
Chart Objective:
Compare the average values of a response across scenarios
Compare the replication values of a response for a single scenario
Chart Type:
Column
Bar
Line
3D Column
3D Bar
3D Line
Hi Lo
Box and Whisker
Sample:
Display single or multiple response values against a scenario name or control value.
choose any chart type
Next >
Cancel
Help



we changed to another scenarios

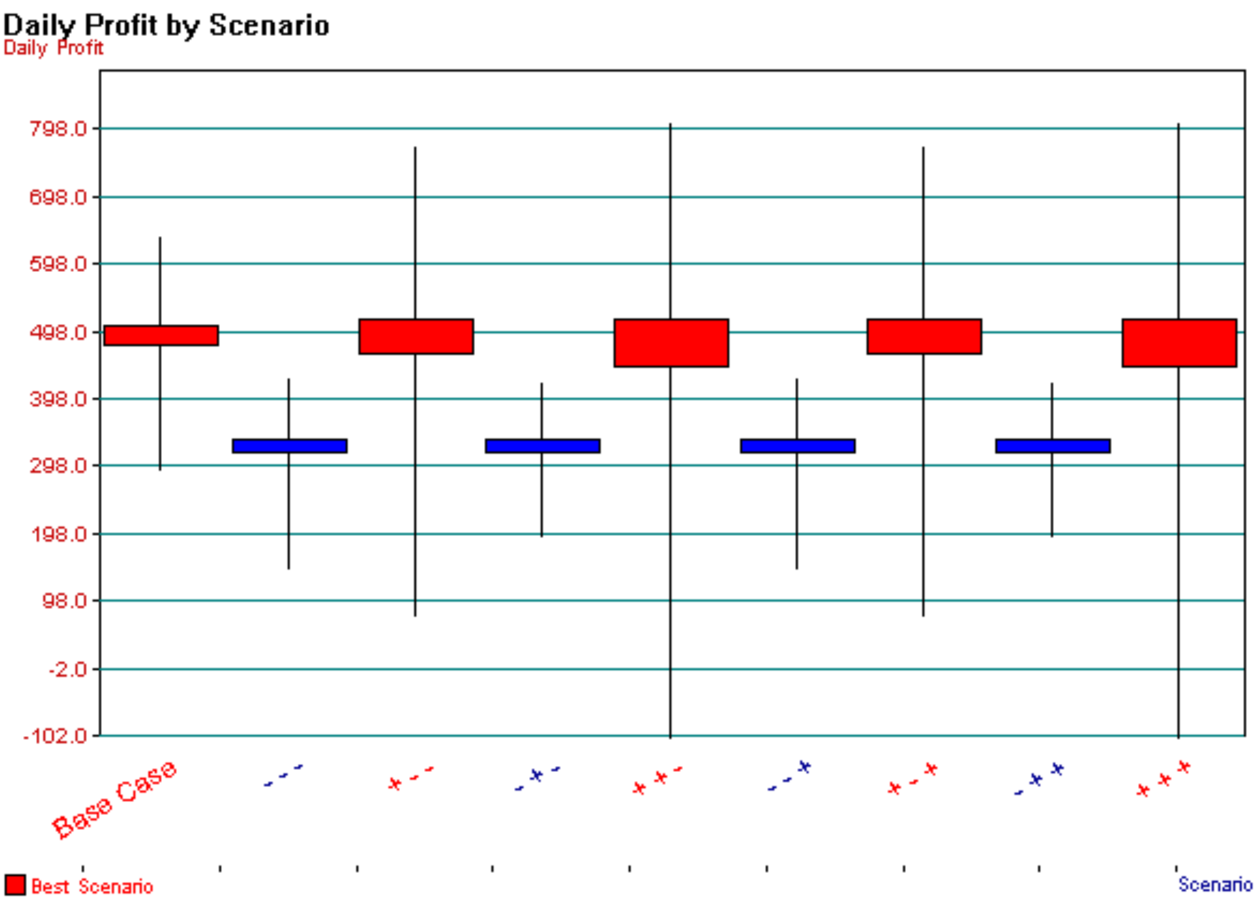
select a Response and press the chart button and next next to see the results

final result

Statistical Comparisons with PAN

- **Model 6-4 alternatives were made with 100 replications each**
  - Better than one replication, but what about statistical validity of comparisons, selection of "the best"?
- **Select Total Cost column, *Insert > Chart* (or  or right-click on column, then Insert Chart)**
  - Chart Type: Box and Whisker
  - Next, Total Cost; Next defaults
  - Next, Identify Best Scenarios
    - Bigger is Better, Error Tolerance = 0 (not the default)
    - Show Best Scenarios; Finish

Statistical Comparisons with PAN (cont'd.)



رسومات توضيحية ولكنها ليست احصائية

- **Vertical boxes: 95% confidence intervals**
- **Red scenarios statistically significantly better than blues**
  - More precisely, red scenarios are 95% sure to contain the best one
  - Narrow down red set – more replications, or Error Tolerance > 0
  - More details in book

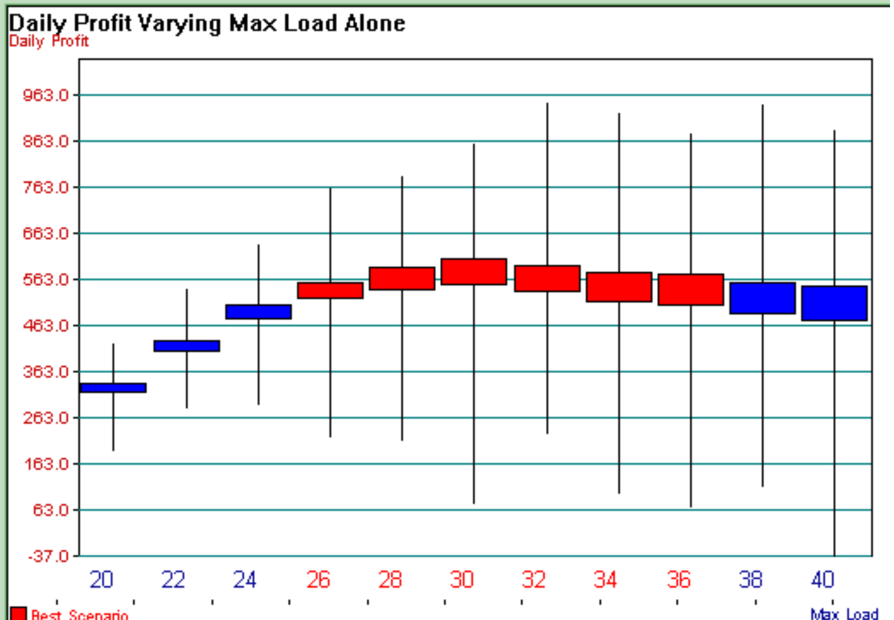
A Follow-Up PAN Experiment

- **From 2^3 factorial experiment, it's clear that `Max Load` matters the most, and bigger appears better**
  - It's factor 1, varying between "−" and "+" in each scenario as ordered there, creating clear down/up/down/up pattern
  - Could also see this by computing *main effects* estimates
    - Consult an experimental-design text
- **Eliminate other two factors (fix them at their base-case levels) and study `Max Load` alone**
  - Let it be 20, 22, 24, ..., 40
  - Set up a second PAN experiment to do this, created chart as before

A Follow-Up PAN Experiment (cont'd.)

	Scenario Properties				Control	Responses	
	S	Name	Program File	Reps	Max Load	Daily Profit	Daily Late Wait Jobs
1		20	1 : Model 06-04.p	100	20.0000	328.825	0.600
2		22	1 : Model 06-04.p	100	22.0000	419.706	0.659
3		24	1 : Model 06-04.p	100	24.0000	492.628	0.699
4		26	1 : Model 06-04.p	100	26.0000	539.121	0.799
5		28	1 : Model 06-04.p	100	28.0000	564.532	0.808
6		30	1 : Model 06-04.p	100	30.0000	581.258	0.908
7		32	1 : Model 06-04.p	100	32.0000	564.251	0.938
8		34	1 : Model 06-04.p	100	34.0000	545.189	0.979
9		36	1 : Model 06-04.p	100	36.0000	540.660	1.039
10		38	1 : Model 06-04.p	100	38.0000	523.838	1.018
11		40	1 : Model 06-04.p	100	40.0000	511.547	0.977

Double-click here to add a new scenario.



- **Here, profit-maximizing `Max Load` is about 30**
- **But `Daily Late Wait Jobs` keeps increasing (worsening) as `Max Load` increases**
  - At profit-maximizing **`Max Load`** = 30, it's 0.908 job/day, which seems bad since we only take 5 wait jobs/day
  - Would like to require that it be at most 0.75 job/day ... still want to maximize **`Daily Profit`**
  - Allow other two factors back into the picture ...

Searching for an Optimal Alternative with OptQuest

- **The scenarios we've considered with PAN are just a few of many possibilities**
- **Seek input controls maximizing `Daily Profit` while keeping `Daily Late Wait Jobs` ≤ 0.75**
- **Formulate as an optimization problem:**

Maximize `Daily Profit` — Objective function is the simulation model

Subject to $20 \le$ `Max Load` ≤ 40
$1 \le$ `Max Wait` ≤ 7
$0.5 \le$ `Wait Allowance` ≤ 2.0
— Constraints on the input control (decision) variables

`Daily Late Wait Jobs` < 0.75 — An output *requirement*, not an input *constraint*

Could also have constraints on linear combinations of input control variables (but we don't in this problem)

  - Reasonable starting place – best acceptable scenario so far: (the base case, actually)
  - Where to go from here? Explore all of feasible three-dimensional space exhaustively? ***No***.

OptQuest

- **OptQuest searches intelligently for an optimum**
  - Like PAN, OptQuest
    - Runs as a separate application … can be launched from Arena
    - "Takes over" the running of your model
    - Asks that you identify the input controls and the output (just one) response objective
  - Unlike PAN, OptQuest
    - Allows you to specify constraints on the input controls
    - Allows you to specify requirements on outputs
    - Decides itself what input-control-value combinations to try
    - Uses internal heuristic algorithms to decide how to change the input controls to move toward an optimum configuration
- **You specify stopping criterion for the search**

Using OptQuest

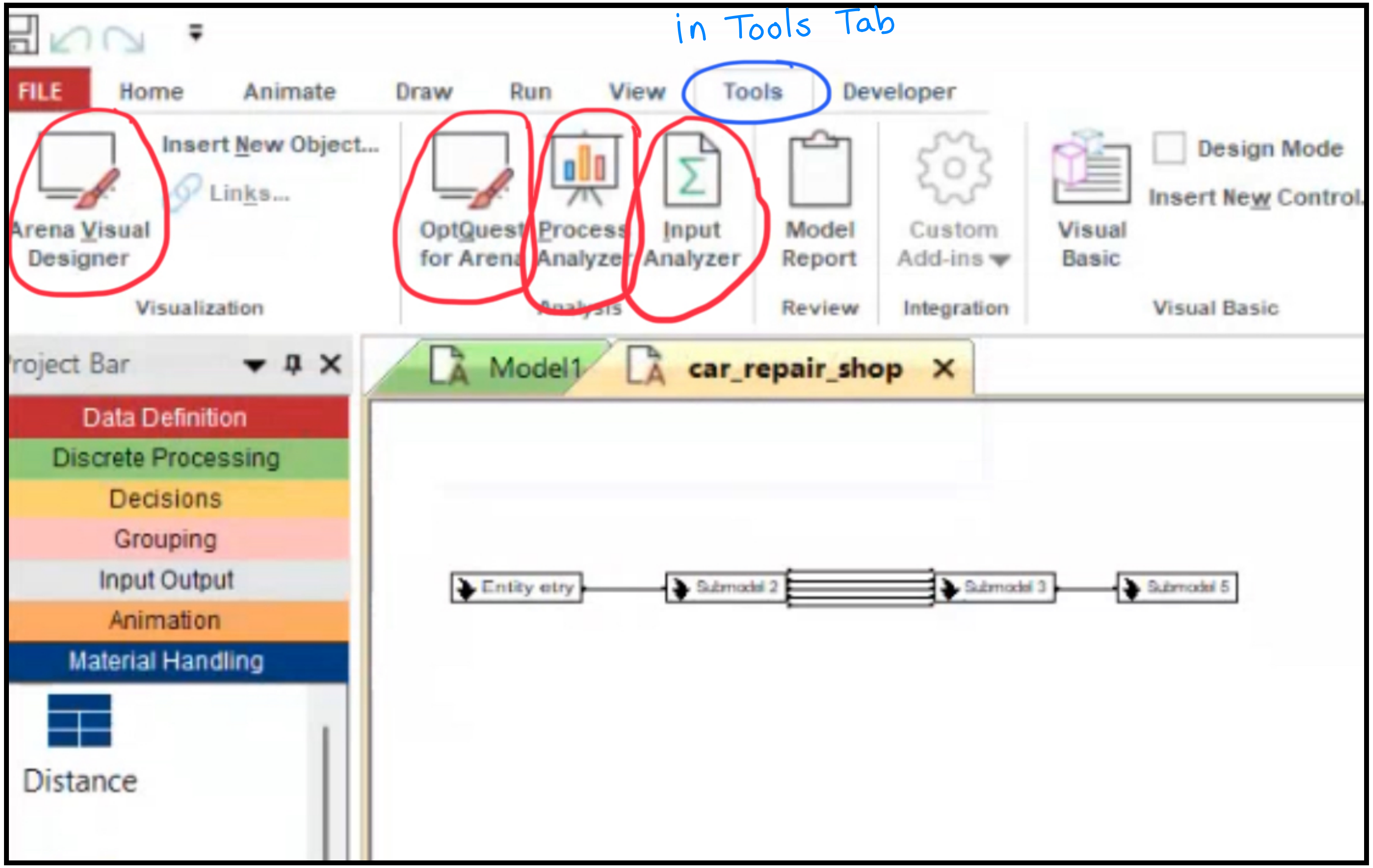
for Optomization

مرتبط باد (PAN)
لكن يجب أن يكون عندي
Upper Bound و Lower Bound

- *Tools > OptQuest for Arena*
- **New session (*File > New* or Ctrl+N or 🗋)**
  - Make sure the desired model window is active
- **Select controls – Variables, Resource levels**
  - **`Max Load`**, Lower Bound = 20, Upper Bound = 40, Conts.
  - **`Max Wait`**, Lower Bound = 1, Upper Bound = 7, Discrete (Input Step Size 1)
  - **`Wait Allowance`**, Lower Bound = 0.5, Upper Bound = 2, Conts.
- **Constraints – none here other than earlier Bounds**
- **Objective and Requirement**
  - **`Daily Profit`** Response – Select Maximize Objective
  - **`Daily Late Wait Jobs`** Response – Select Requirement, enter 0.75 for Upper Bound

Using OptQuest (cont'd.)

- **Options window – computational limits, procedures**
  - Time tab – run for 20 minutes
  - Precision tab – vary number of replications from 10 to 100
  - Preferences tab – various settings (accept defaults)
- **Can revisit Controls, Constraints, Objective and Requirements, or Options windows via**
- **Run via wizard (first time through a new project), or *Run > Start* or**
- ***View > Status and Solutions* and *View > Performance Graph* to watch progress**
- **Can't absolutely guarantee a true optimum**
  - Usually finds far better configuration than possible by hand



الفرق بين الـ (Process Analyzer) والـ (OptQuest)

احنا بنحدد الـ

scenarios
controls
responses

Chapter(6)

*Separate Results for each Replication

Run → Setup → Replication Parameters → Initialize between Replications

☐ Statistics → if we do ☑ initialize for statistics it will take every replicate alone
if not ☐ it will take commulative summary reports

☐ System → if we do ☑ initialize for system it will Start from Scratch
if not ☐ it will take Commulative for the Replications number

اذا سوينا (✓) بصير في نهاية كل Replicate
اذا مثل شوى من القطع بصفرهم و بنفذ كمان مرة

*Confidence Intervals for terminating Systems

→ the information is in category overview report

other confidence levels → in output Analyzer

Confidence Interval → $\bar{X} \pm t_{n-1,1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}}$

Half Width → $t_{n-1,1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}}$

$n \cong n_0 \frac{h_0^2}{h^2}$

*Comparing two Alternatives

-Add File (...) to statistic Module ⟨ column / Row , two versions ⟨ Base case / MoreBooking case

ومرات بنشوف اذا ال two versions بيسوو overlap or Not

*Compare Means Via the Output Analyzer

Output Analyzer → for statistical Analysis
→ is a seperate Application that operates on (.dat files)

-open Output Analyzer (seperate Application from Arena)
-new → Add (button) to select .dat File then ok → it will give me the output Graph
-we can go to Analyze → choose
→ Batch/Truncate
→ Conf. Interval on Mean
→ Compare means
→ compare Variance
→ one way-Anova

*Evaluating Many Alternatives with the Process Analyzer (PAN)

Process Analyzer → used to compare many Alternatives
→ operates on (.p files)
→ Collection of scenarios)

→ Tools tab → Process Analyzer
→ New → [redacted] → choose .p File Format
→ double click to add scenarios
→ from insert
→ Scenario → Row → choose Any of them and then
→ Control → column → the resource of Queue...
→ Response → column → we are intrested in
→ Select (the scenarios) then Run → the empty Responses will be filled

$2^3 = 8$ + Base case

	Scenario Name	Controls			Responses	
		Max Load	Max. wait	Allowance	Profit	wait Job
		Base case				
1		take All possibilities				
2						
3						
4						
⋮						

we can press (Chart) and select any Response then ok

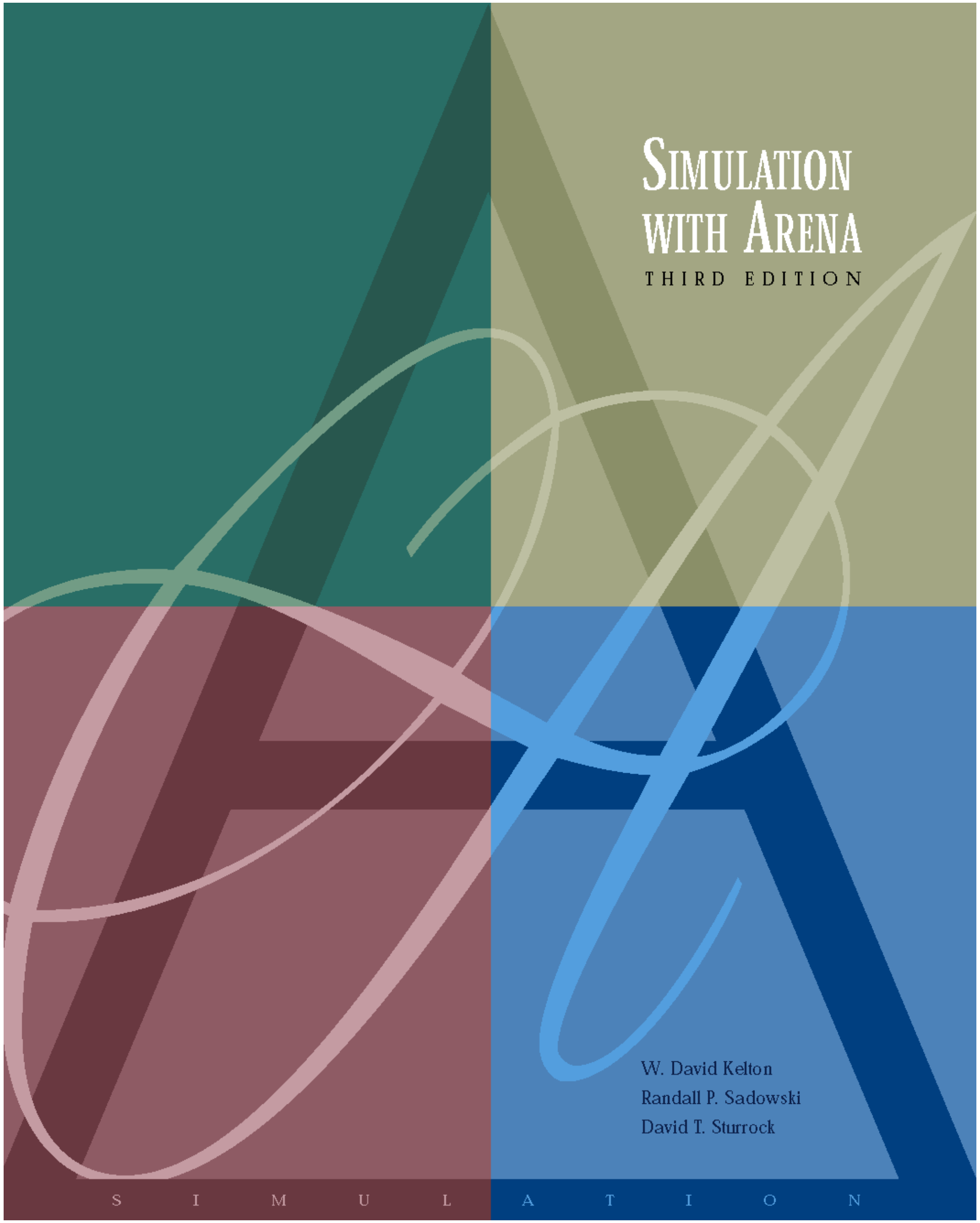
Appear after Run

*Searching for an Optimal Alternative with OptQuest

OptQuest → for optimization
→ it asks to identify the iput controls and the Response (one Response)
+ it Allows to specify constains on the input

PAN مرتبط بال
لكن يجب يكون عندي Upper Bound and Lower Bound

→ Tools Tab → OptQuest
→ New → Select controls and (LB, UB)
→ Objective and Requirement



Arena Integration and Customization

Chapter 10

Last revision July 14, 2003

What We'll Do ...

- Reading and Writing Data Files (ReadWrite)
- ActiveX™ and Visual Basic® for Applications (VBA)
- Creating Modules with Arena Professional Edition

There is courses
about (Decision Analysis)

Reading and Writing Data Files

- Reading entity arrivals from a text file
- Reading and writing Microsoft Access and Excel files

more useful

- Advanced reading and writing

Reading Entity Arrivals From a Text File

- **Why data-driven simulations?**
  - Model validation
  - Evaluating how a particular scenario is handled
  - Modeling a specific arrival pattern
  - Assumes historical data exist and can be transformed for use in simulation

Array → index i+1

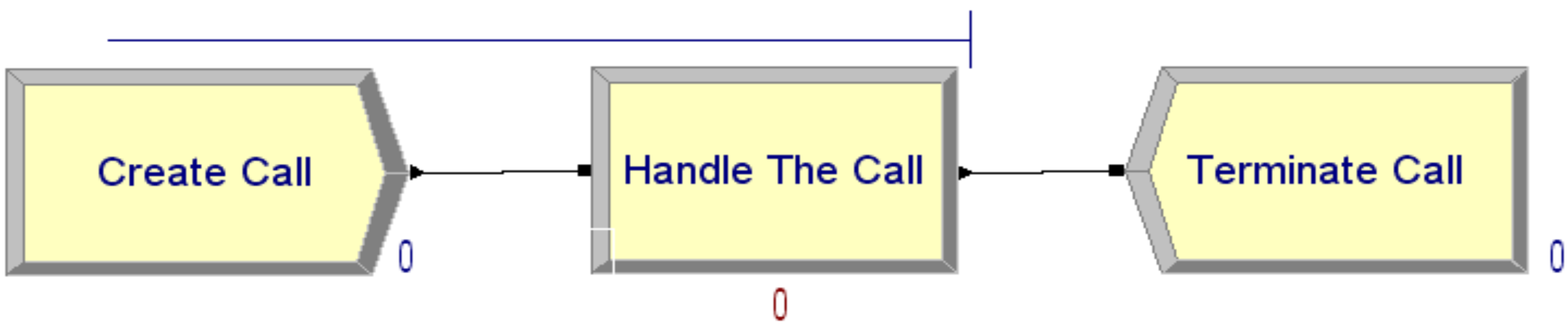
Reading Entity Arrivals from a text file

مثلاً" اذا بدنا نكرر ال Exact Day

Simple Call Center Model

- Single call stream
- Single agent resource
- Random call processing time

اذا بدي اخليه يتبع scenario
معين



External Call Center Data

certain scenarios انفذ الهدف انو

- **Historical call arrival times**
- **Model 10-02 Input.txt**
  - ASCII file (e.g., Notepad, saved as text from Excel)
  - Absolute simulation arrival times

    1.038457

    2.374120 → 1.038457 + 2.374120

    4.749443

    9.899661

    10.525897 → time between

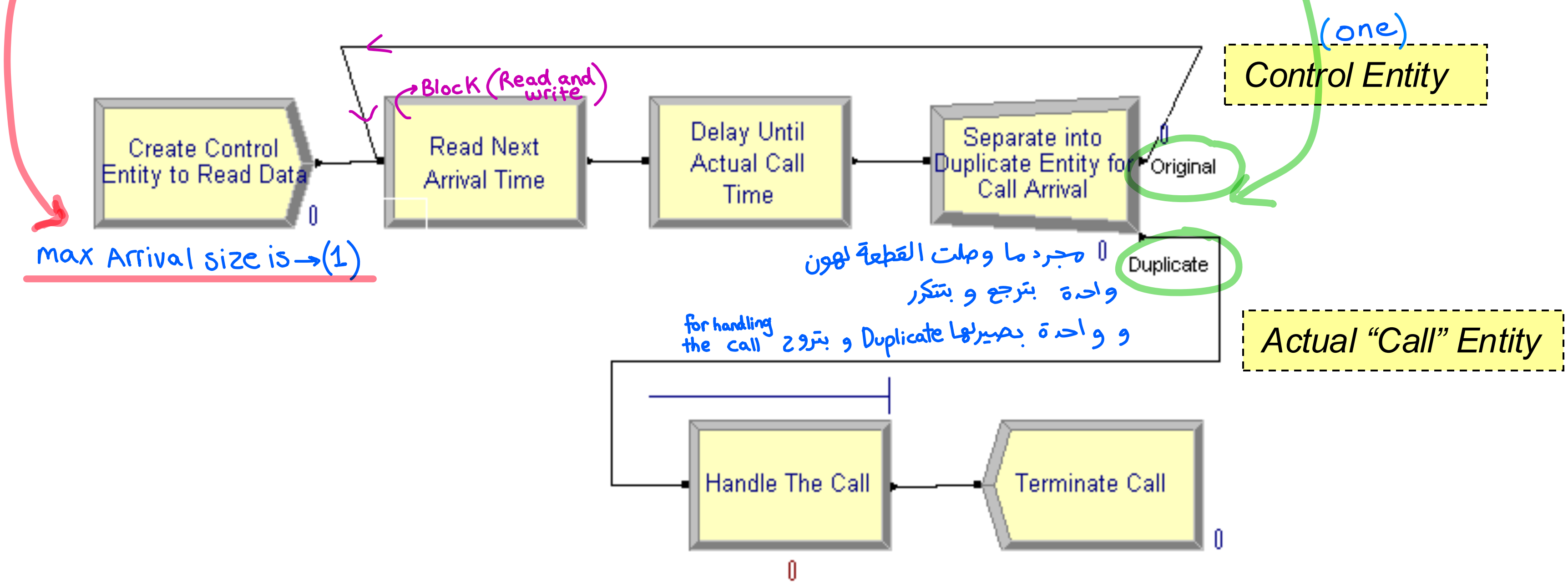
    17.098860

or → Array
or → Read from sequential file
→ Read from excell or Access file

هنا البرمجة تتم على أكثر من محور

Model Logic to Read Data

- **Can't use simple time between arrivals**
- **Control entity**
  - Create only one
  - Duplicate to send actual "call" entity into model



Model Logic to Read Data (cont'd.)

- **ReadWrite module (Advanced Process)**
  - *Arena File Name*: description (actual disk filename is specified in File module)
  - *Assignments*: model variables/attributes to be assigned based on data read from file (*Call Start Time* attribute)
- **Delay/Duplicate Logic**
  - File contains "absolute" times; Delay module holds entity for a time *interval*
  - Delay control entity for interval until actual arrival time of call $\boxed{(\textit{Call Start Time - TNOW})}$ Commulative لأنو
  - Create a duplicate (Separate module) to dispatch actual call into model. Original entity loops back to read next time.

we can do it from any text File
Such as Notepad

Model Logic to Read Data (cont'd.)

- **File data module (Advanced Process)**
  - *Name*: Name referenced in other Arena modules.
  - *Access Type:* Sequential indicates to read in order.
  - *Operating System File Name*: The name used by file system. May be relative or fully qualified.
  - *End of File Action*: What to do when all records are read.

File

	Name	Access Type	Operating System File Name	Structure	End of File Action	Initialize Option	Comment Character
1	Arrivals File	Sequential File	Model 10-02 Input.txt	Free Format	Dispose	Hold	No

to read in order

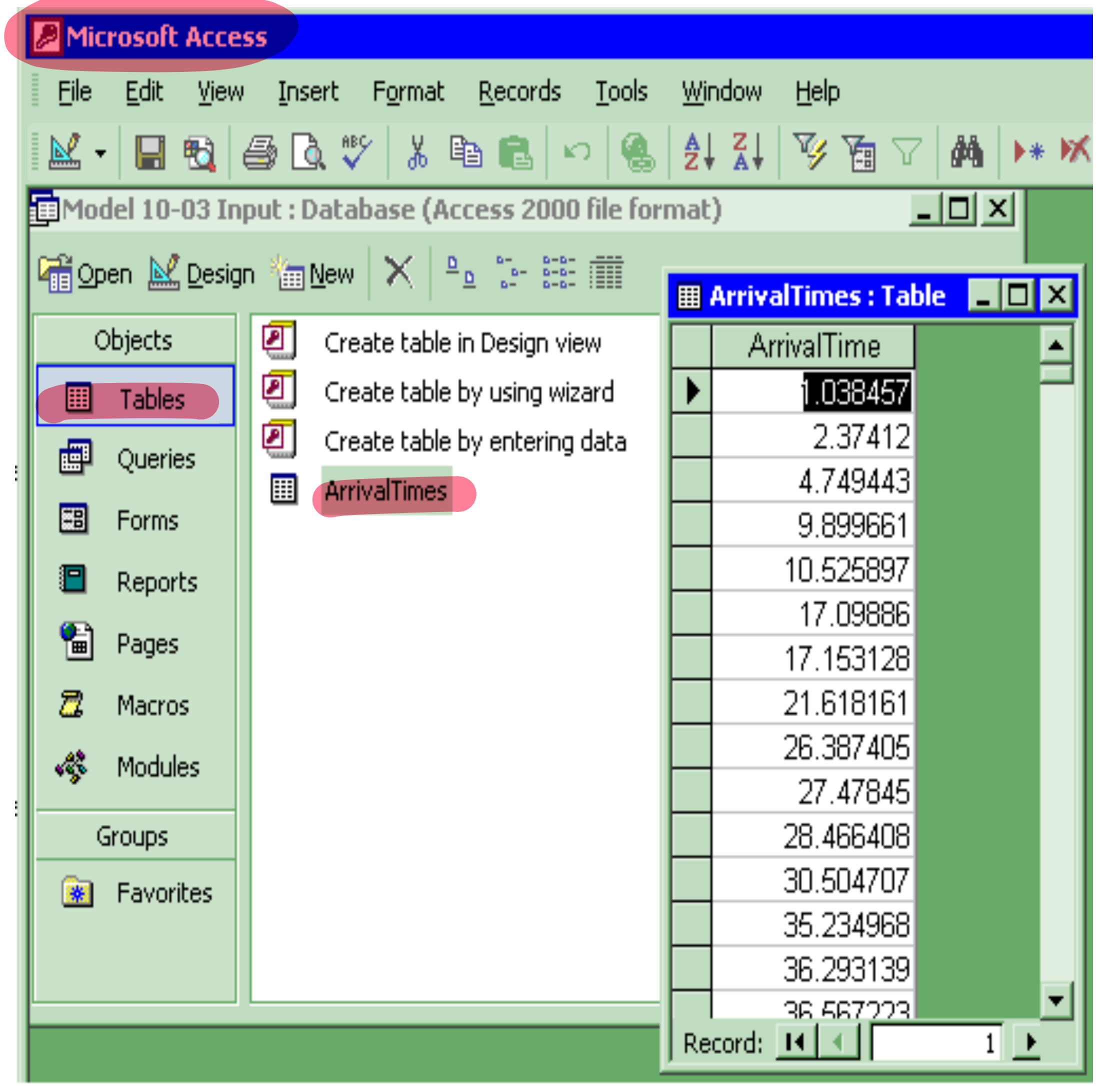
Run Termination

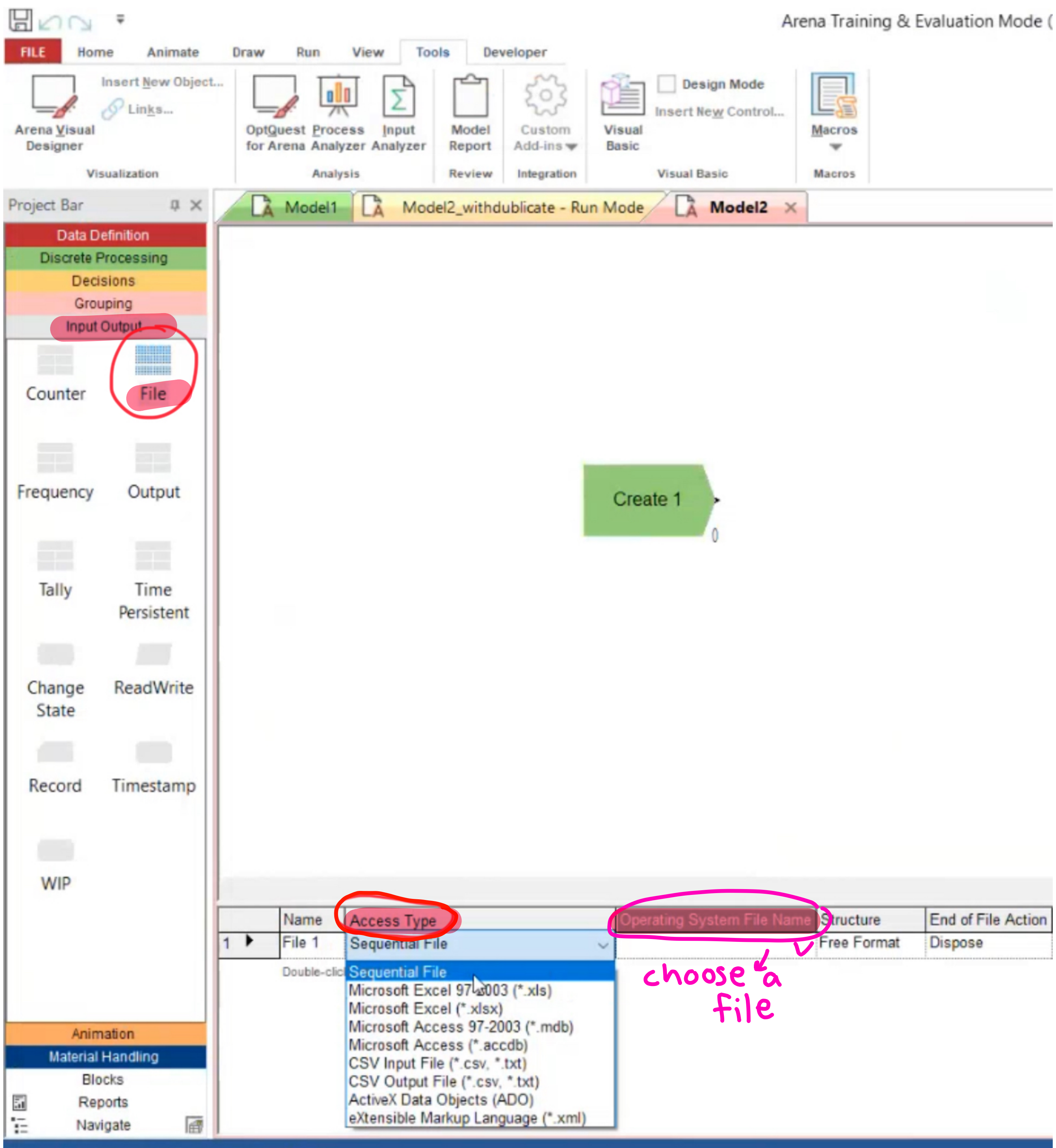
- **Run Setup options**
  - Maximum replications / simulation end time always terminates the simulation run.
- **System empties**
  - If no entities on calendar and no other time-based controls, run may terminate earlier than setup options dictate.
  - The control entity is disposed after it reads the last data value.

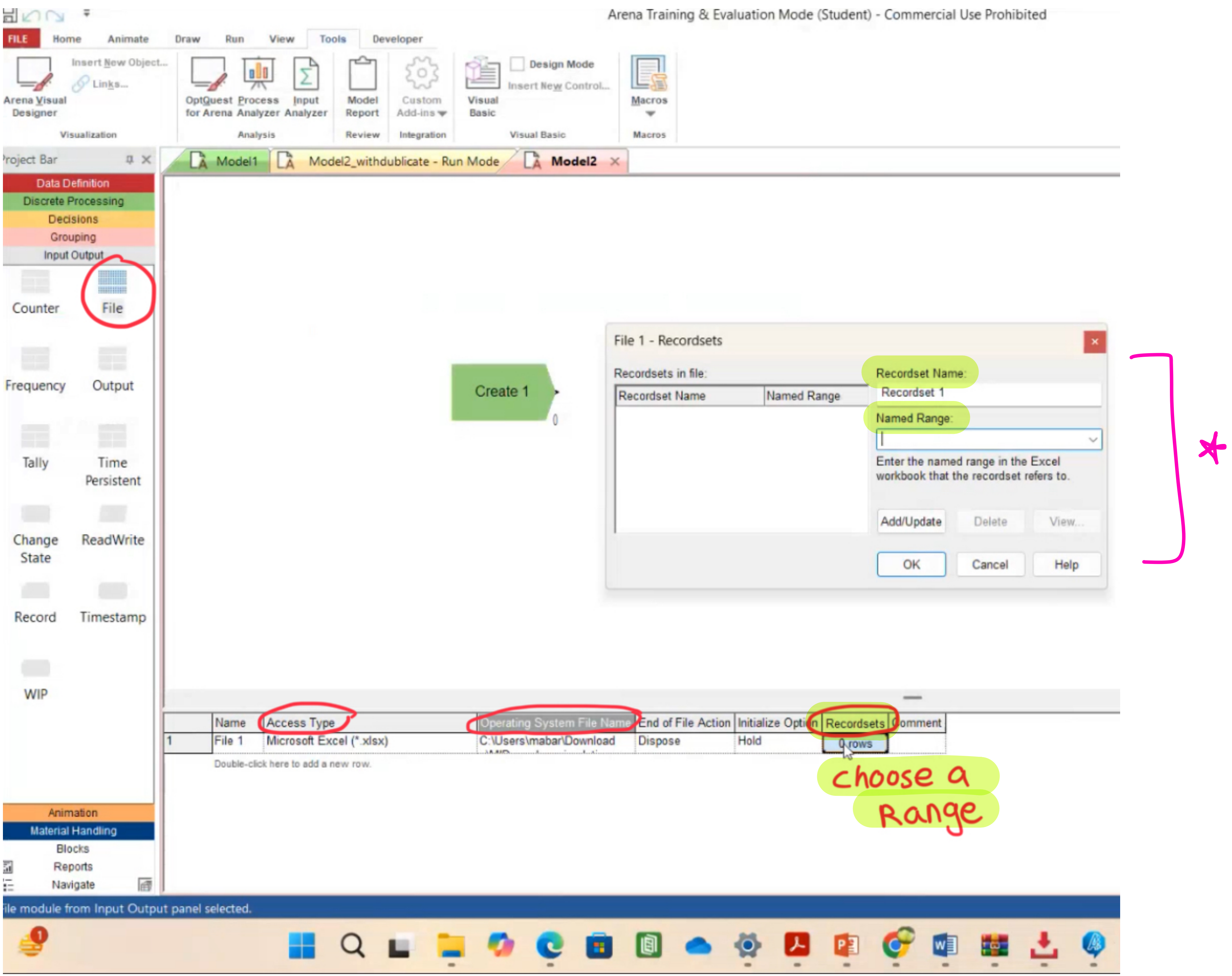
Reading Access Files

يعني بدل ما نقرأ من ال Notepad استعملنا ال Microsoft Access

- **Sample Access data**
  - Model 10-03 Input.mdb
  - Table: ArrivalTimes




Arena Training & Evaluation Mode (
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Arena Visual Designer
Insert New Object...
Links...
Visualization
OptQuest for Arena
Process Analyzer
Input Analyzer
Analysis
Model Report
Review
Custom Add-ins
Integration
Visual Basic
Design Mode
Insert New Control...
Visual Basic
Macros
Macros
Project Bar
Model1
Model2_withdublicate - Run Mode
Model2
Data Definition
Discrete Processing
Decisions
Grouping
Input Output
Counter
File
Frequency
Output
Tally
Time Persistent
Change State
ReadWrite
Record
Timestamp
WIP
Animation
Material Handling
Blocks
Reports
Navigate
Create 1
0
Name
Access Type
Operating System File Name
Structure
End of File Action
1
File 1
Sequential File
Free Format
Dispose
Double-clic
Sequential File
Microsoft Excel 97-2003 (*.xls)
Microsoft Excel (*.xlsx)
Microsoft Access 97-2003 (*.mdb)
Microsoft Access (*.accdb)
CSV Input File (*.csv, *.txt)
CSV Output File (*.csv, *.txt)
ActiveX Data Objects (ADO)
eXtensible Markup Language (*.xml)
choose a file



Arena Training & Evaluation Mode (Student) - Commercial Use Prohibited
File 1 - Recordsets
Recordsets in file:
Recordset Name
Named Range
Recordset Name:
Recordset 1
Named Range:
Enter the named range in the Excel workbook that the recordset refers to.
Add/Update
Delete
View...
OK
Cancel
Help
Create 1
Name
Access Type
Operating System File Name
End of File Action
Initialize Option
Recordsets
Comment
File 1
Microsoft Excel (*.xlsx)
Dispose
Hold
0 rows
Double-click here to add a new row.
choose a Range
*

Reading Access Files

- **File data module (Advanced Process)**
  - *Access Type*: Microsoft Access (*.mdb)
  - *Operating System File Name*: Model 10-03 Input.mdb
  - *Recordsets*: Click to load the Recordsets Editor

وين بالزبط موجود ال Access جوا ال file
و ال Range يلي بدنا نقرأه

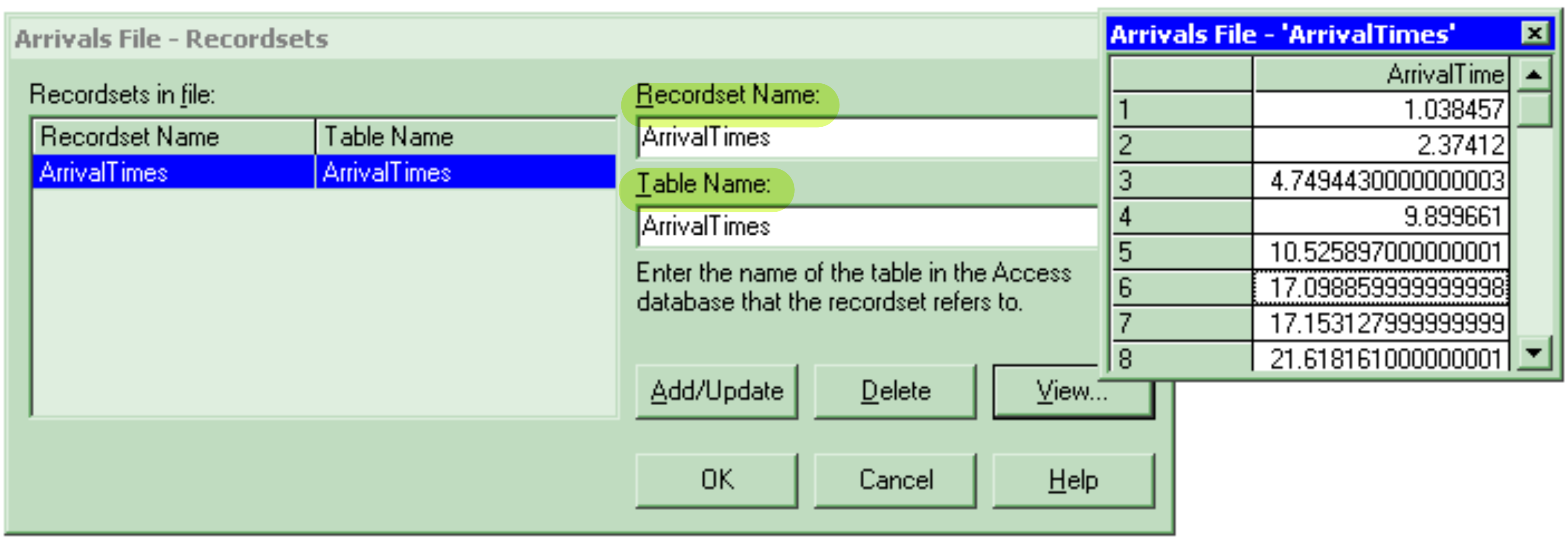
- **Important:**
  - Never name an access file the same as the model name or it will conflict with the automatic output database file.

	Name	Access Type	Operating System File Name	End of File Action	Initialize Option	Recordsets
1	Arrivals File	Microsoft Access (*.mdb)	Model 10-03 Input.mdb	Dispose	Hold	0 rows

For Help, press F1 (-161, 3266)

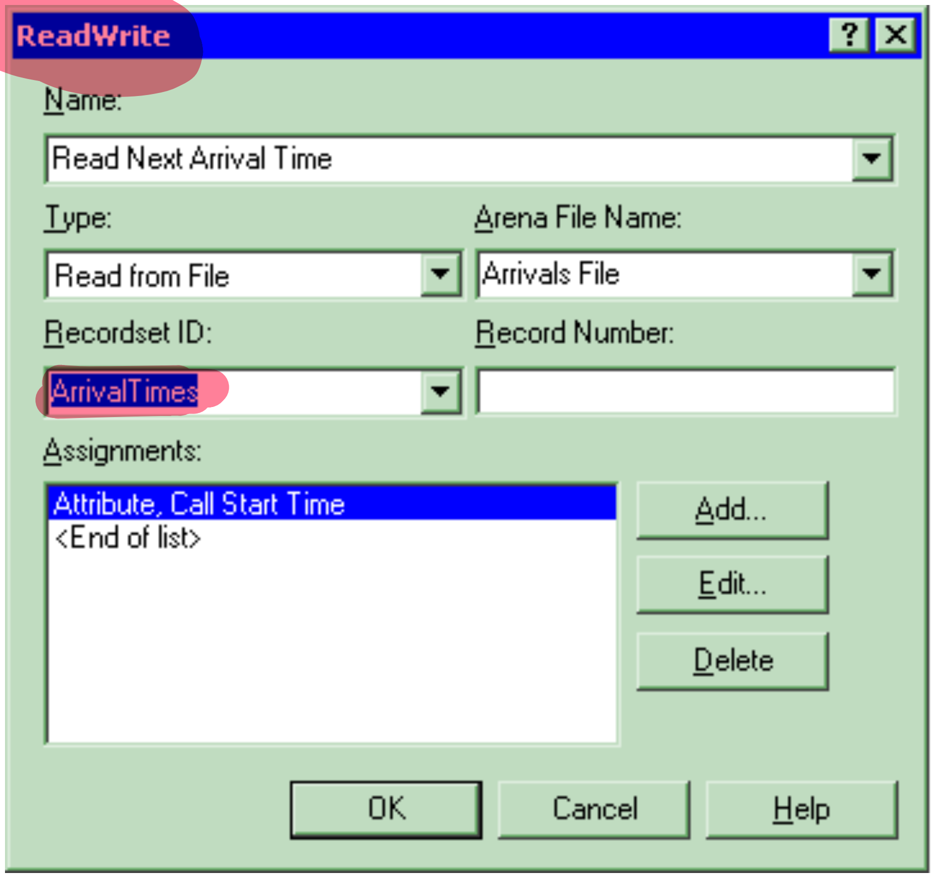
Reading Access Files

- **Recordsets Editor**
  - Associates a recordset name with a table
  - Table must already exist
  - *View* allows you to see a sample of the real data



Reading Access Files

- **ReadWrite module (Advanced Process)**
  - *Recordset ID*: Same as defined in Recordsets Editor

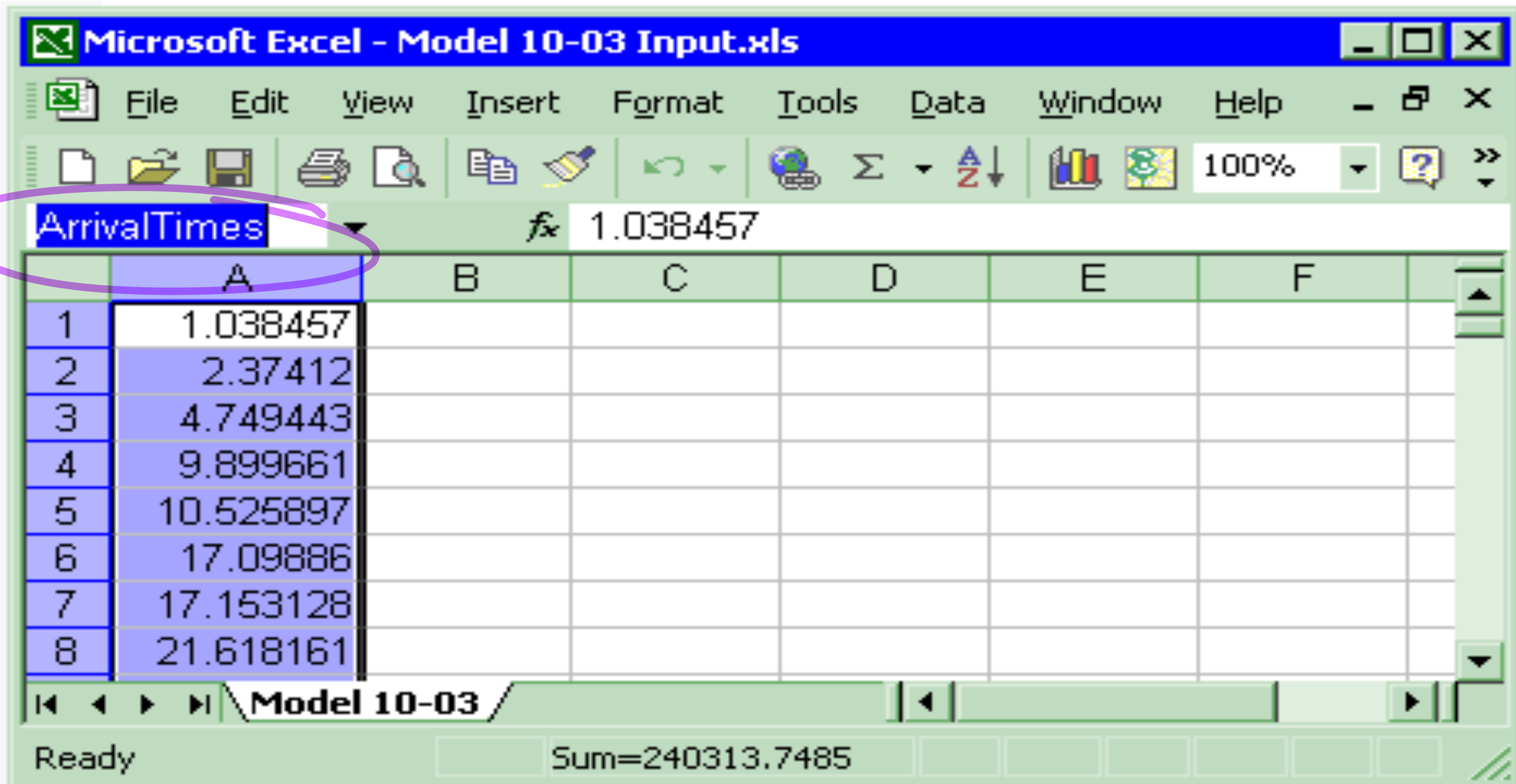


Reading Excel Files

- **Excel is not a relational database but has many similarities:**
  - *An Excel workbook is similar to an Access database file.*
  - *The rows and columns in a rectangular named range in an Excel worksheet are similar to the rows and columns of an Access table.*

Reading Excel Files

- **Sample Excel data**
  - Model 10-03 Input.xls
  - Named Range: ArrivalTimes



Reading Excel Files

- **File data module (Advanced Process)**
  - *Access Type*: Microsoft Excel (*.xls)
  - *Operating System File Name*: Model 10-03 Input.xls
  - *Recordsets*: Click to load the Recordsets Editor

Block

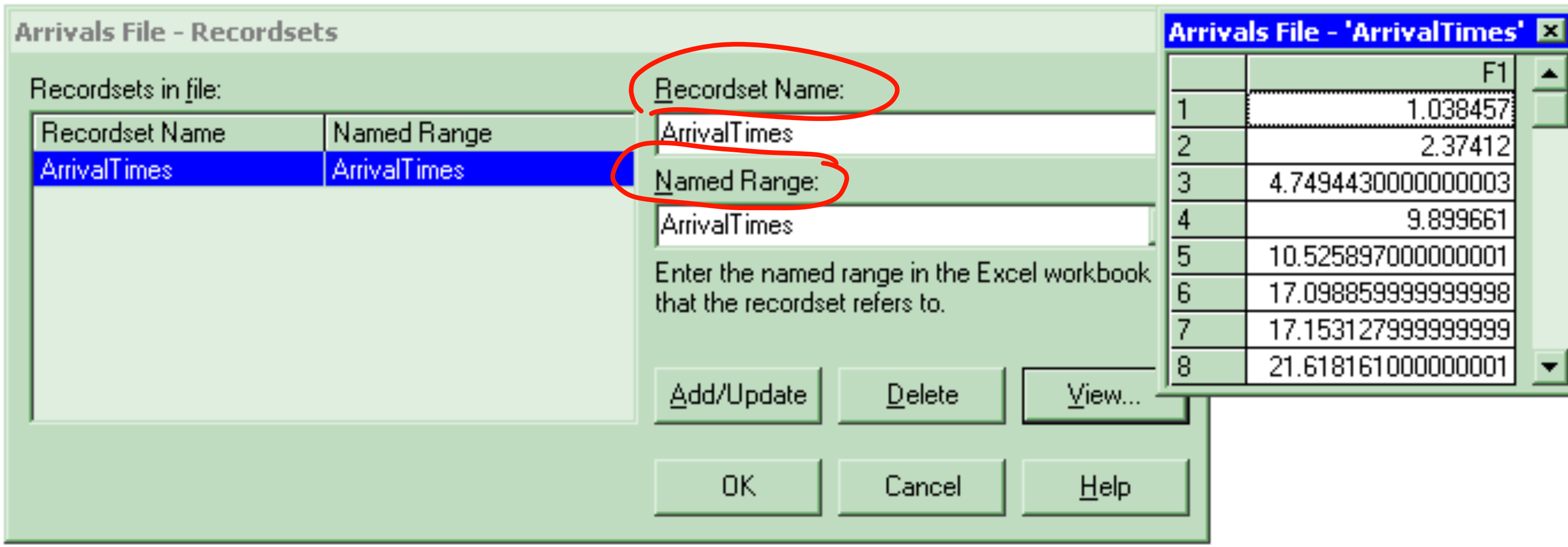
$\boxed{\text{File}}$ from (input-output)

	Name	Access Type	Operating System File Name	End of File Action	Initialize Option	Recordsets
1	Arrivals File	Microsoft Excel (*.xls)	Model 10-03 Input.xls	Dispose	Hold	1 rows

For Help, press F1 (-593, 3618)

Reading Excel Files

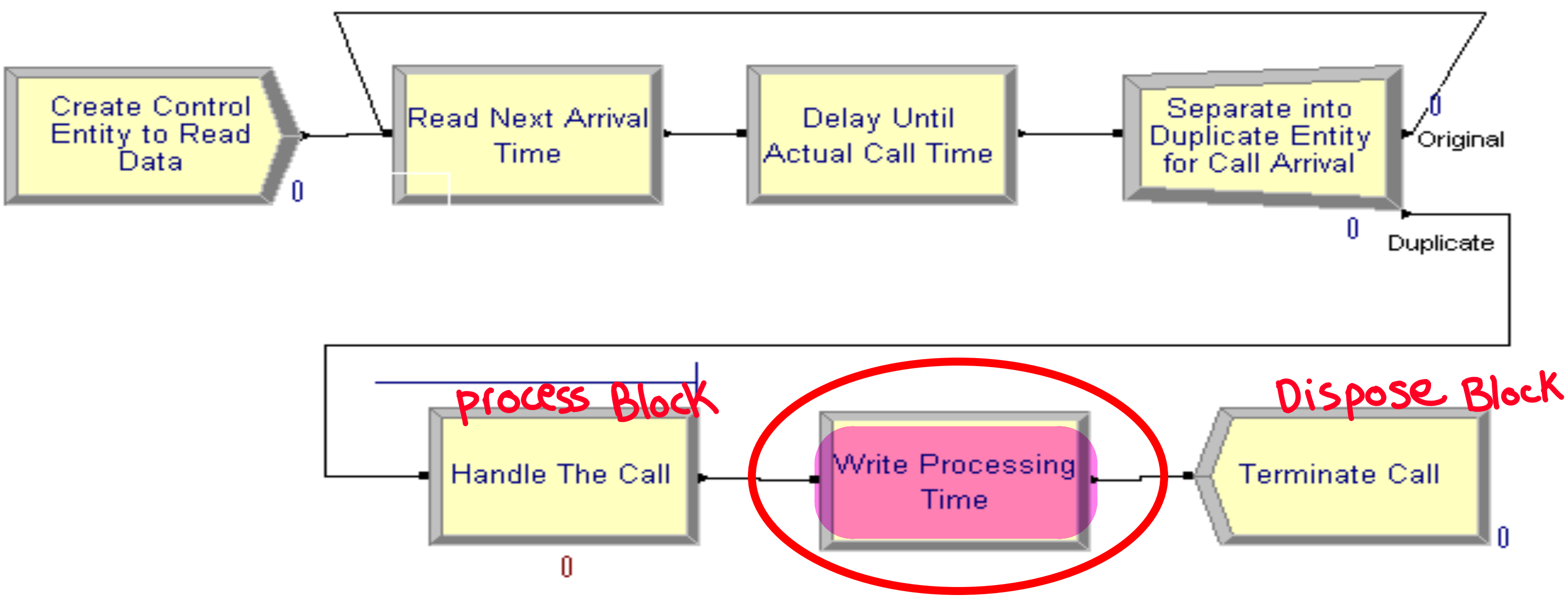
- **Recordsets Editor**
  - Associates a recordset name with a named range
  - Named range must already exist
  - *View* allows you to see a sample of the real data



Writing Access and Excel Files

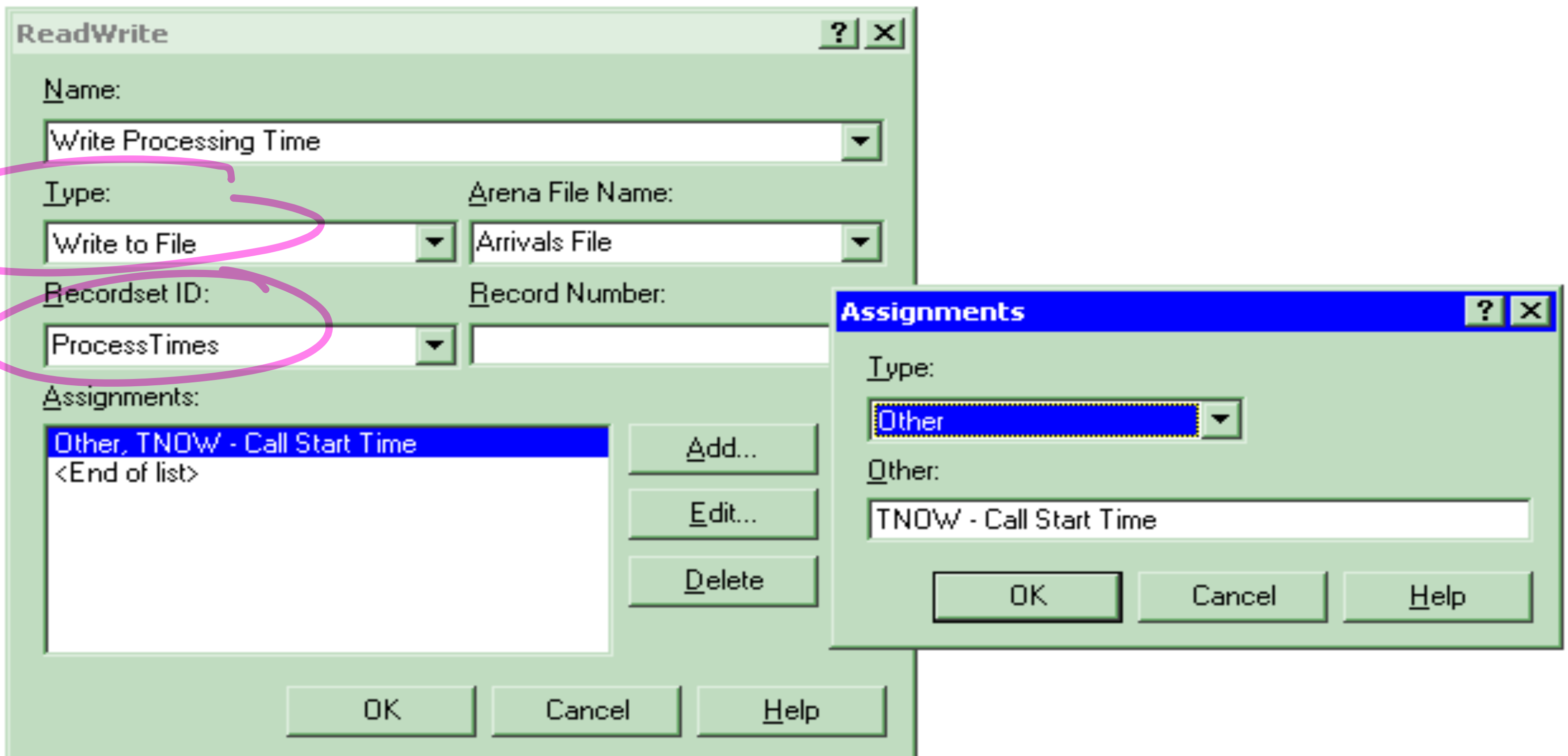
we can read and write from a named Range in Excel

- ***The file:***
  - *The table or named range must already exist.*
  - *An Excel named range should be formatted as numeric.*
- **ReadWrite module:**
  - Add new module to specify which data to write.



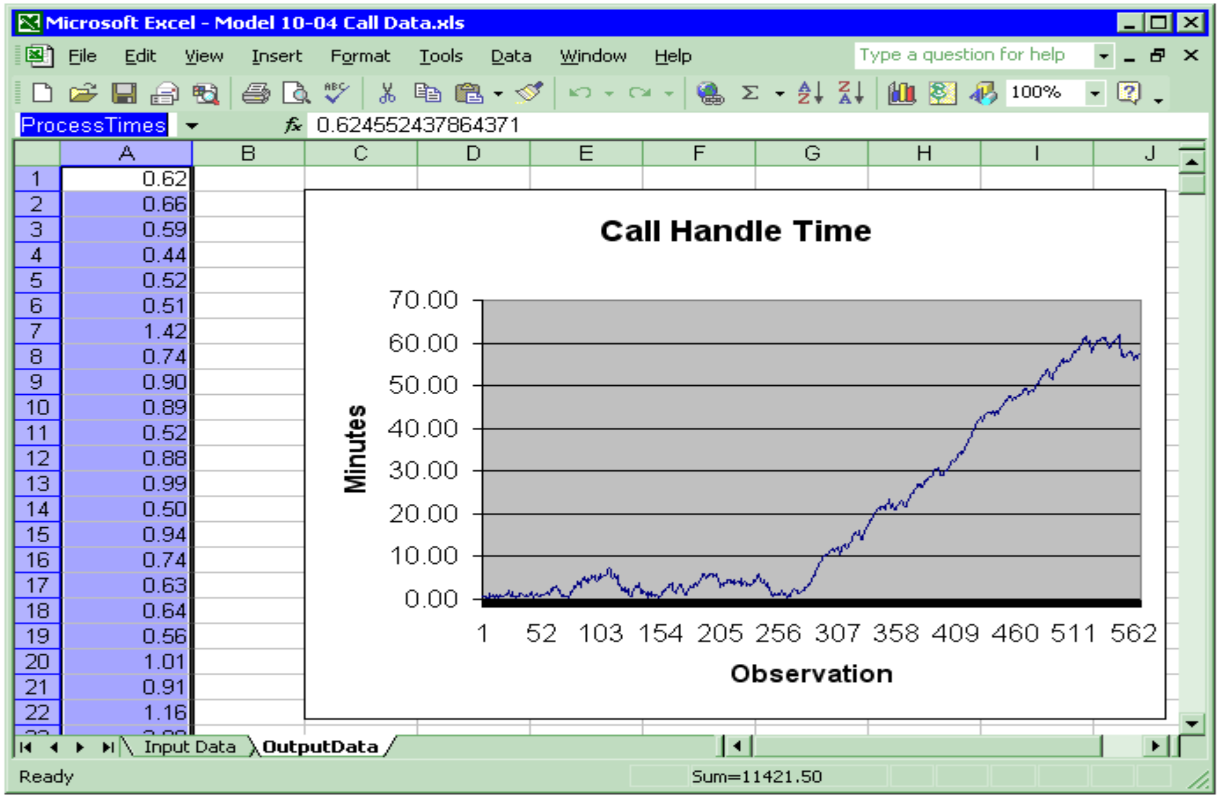
Writing Access and Excel Files

- **ReadWrite module:**
  - Use *Type* of *Write To File*
  - *Use Recordset ID as before.*



Writing Access and Excel Files

- **Spreadsheet options:**
  - You may predefine a plot on the named range and the plot will be built dynamically as data is added to the file.



Advanced Reading and Writing

- **Formatting can be used to handle text files with fields not delimited by spaces:**

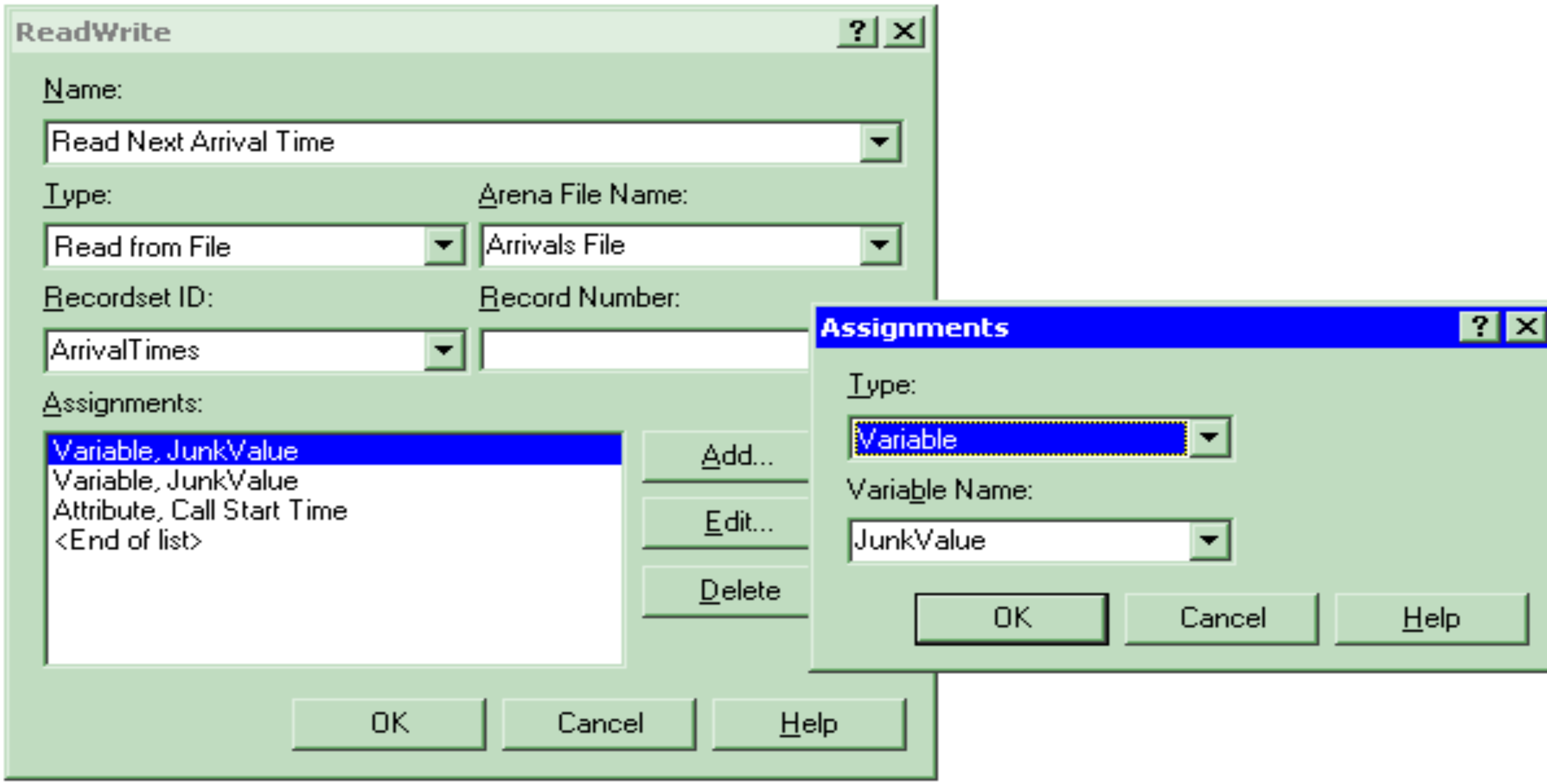
```
Part 1  1.038
Part 27 2.374
Part 5194.749
Part 67 9.899
Part 72 10.52
Part 16217.09
```

	Name	Access Type	Operating System File Name	Structure	End of File Action	Initialize Option	Comment Character
1	Arrivals	Sequential File	Model 10-02 Input.txt	"(8x,f8.3)"	Dispose	Hold	No

For Help, press F1 (145, 4932)

Advanced Reading and Writing

- **Skip columns in Access & Excel by using dummy variables:**



Advanced Reading and Writing

- **Advanced data access is available using *Access Type* of *Active Data Objects (ADO)* and a *Connection String:***
  - ***Excel With Headings Using ADO***

    Provider=Microsoft.JET.OLEDB.4.0;
    Data Source=C:\Documents\Book1.xls;
    Extended Properties=""""Excel 8.0; HDR=Yes;""""

  - ***SQL Commands Using ADO***

    Driver={SQL Server};
    Server=RSI-Joe; Database=BizBikes;
    Uid=BizWareUser; pwd=MyPassword

Use two double quotes for each embedded double quote

What We'll Do ...

- Reading and Writing Data Files (ReadWrite)
- ActiveX™ and Visual Basic® for Applications (VBA)
- Creating Modules with Arena Professional Edition

is extremely helpfull in engineering

ActiveX Automation

معناه ممكن ال Excell يفتح ال Arena
و ال matlab " " "
و هكذا

- **Program applications to "automate" tasks**
  - Act on themselves (e.g., macros in Excel)
  - Act on other applications (e.g., Arena creating Excel file)
- **External programming languages**
  - C++, Visual Basic®, Java, etc.
- **Visual Basic for Applications (VBA) programming embedded in application**
  - Microsoft Office®, Visio®, AutoCAD®, Arena®, …
- **Both types work together (e.g., Arena VBA controlling Excel)**

مثلاً "يوجد Developer جوا ال Excel Arena لتطوير البرنامج و التعامل الخارجي مع Activex و هكذا

Application Object Model

- **Objects**: application ***components*** that can be controlled
- **Properties**: ***characteristics*** of objects
- **Methods**: ***actions*** performed on or by objects

Arena Objects	Properties	Methods
Application	Visible	Show
Model	Name, State	Close, Go
View	Background Color	Zoom In
…		

Name three arena objects?

Visual Basic for Applications (VBA)

- Included with Arena
- Full Visual Basic programming environment
- Code stored with Arena model (*.doe*) file
- UserForms (dialogs) for custom interfaces
- Code-debugging tools
- Comprehensive online help
- Visual Basic Editor window: “child” of Arena (Tools/Show Visual Basic Editor) 

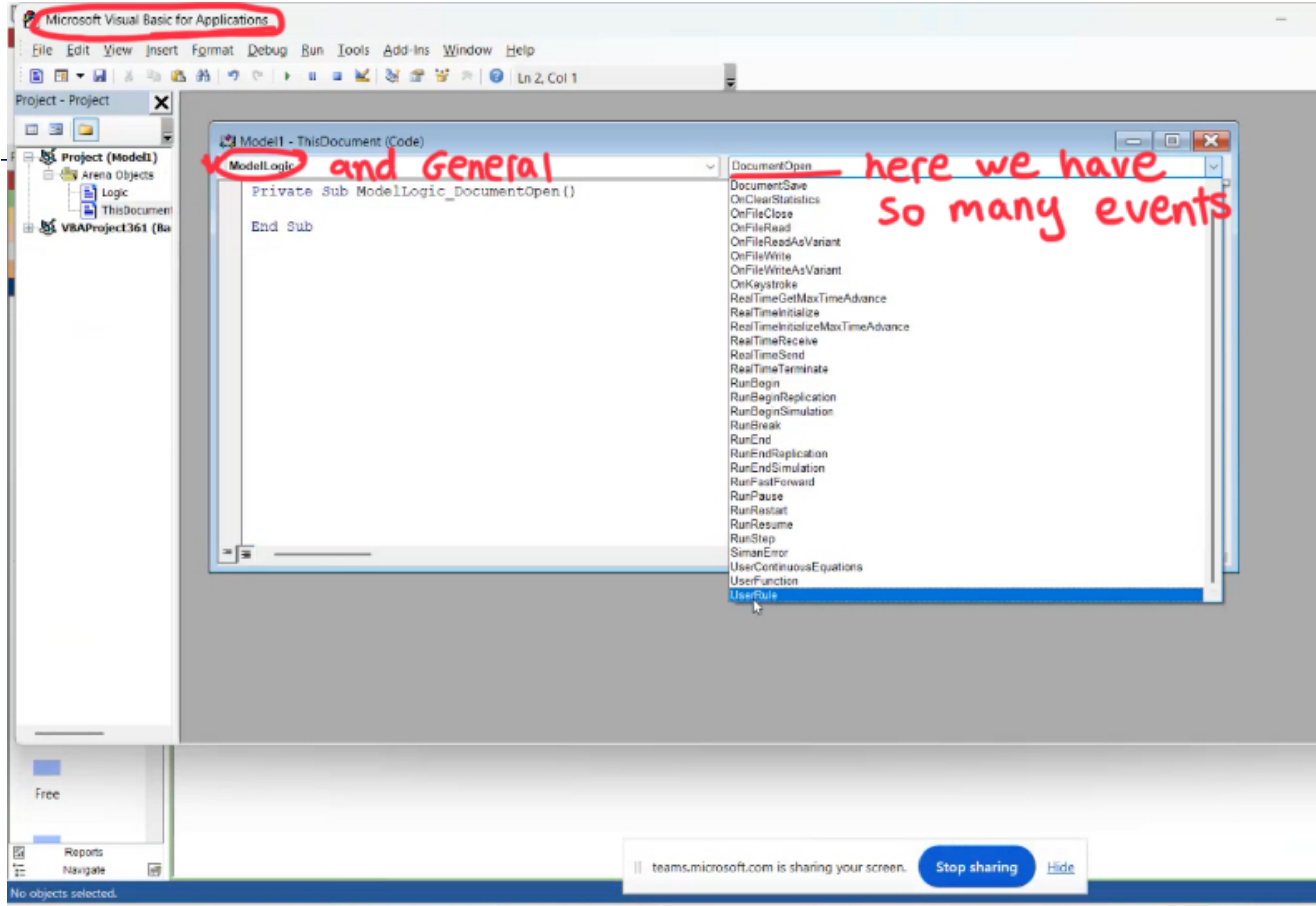
اما Event موجود من الأساس
او Event احنا بنضيفه

Built-in Arena VBA Events

- **ThisDocument: accesses objects, events in Arena's object model**

Visual Basic

- **Built-in VBA events: locations where VBA code can be activated**
  - Pre-run events (e.g., DocumentOpen)
  - Arena-initiated run events (e.g., RunBegin, RunEndReplication)
  - Model/user-initiated run events (e.g., UserFunction, VBA_Block_Fire)
- **Type code in Visual Basic Editor to populate an event**


Microsoft Visual Basic for Applications
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 2, Col 1
Project - Project
Project (Model1)
Arena Objects
Logic
ThisDocument
VBAProject361 (Ba
Model1 - ThisDocument (Code)
ModelLogic
and General
DocumentOpen
here we have so many events
Private Sub ModelLogic_DocumentOpen()
End Sub
DocumentSave
OnClearStatistics
OnFileClose
OnFileRead
OnFileReadAsVariant
OnFileWrite
OnFileWriteAsVariant
OnKeystroke
RealTimeGetMaxTimeAdvance
RealTimeInitialize
RealTimeInitializeMaxTimeAdvance
RealTimeReceive
RealTimeSend
RealTimeTerminate
RunBegin
RunBeginReplication
RunBeginSimulation
RunBreak
RunEnd
RunEndReplication
RunEndSimulation
RunFastForward
RunPause
RunRestart
RunResume
RunStep
SimanError
UserContinuousEquations
UserFunction
UserRule
Free
Reports
Navigate
teams.microsoft.com is sharing your screen.
Stop sharing
Hide
No objects selected.

Simulation Run VBA Events

- **Arena/VBA sequence of events when model runs:**

Name four arena evens?

1. **RunBegin**
2. *Arena checks and initializes the model*

Module data available

3. **RunBeginSimulation**
4. **RunBeginReplication**
5. *Arena runs replication*

OnKeystroke, UserFunction, etc.

6. **RunEndReplication**
7. **RunEndSimulation**

Simulation run data available

8. *Arena terminates the simulation run*
9. **RunEnd**

Module data available

Arena's Object Model

- **Model-window objects: items placed in model window, such as:**
  - Modules
  - Connections
  - Lines
- **SIMAN object: simulation run data, such as:**
  - Variable values
  - Queue lengths
  - Simulation time
- **Structural objects: access general functions**
  - Application
  - Panels

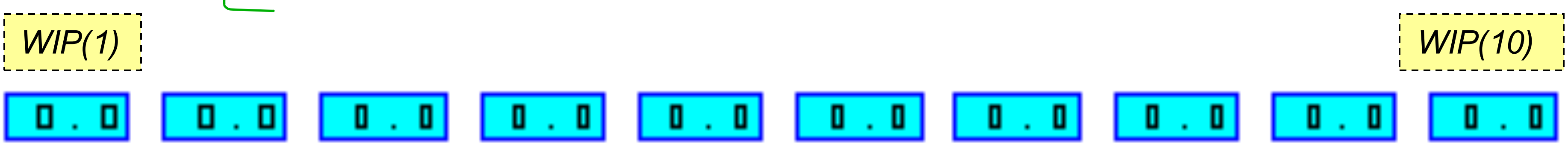
Sample: Create Ten Status Variables

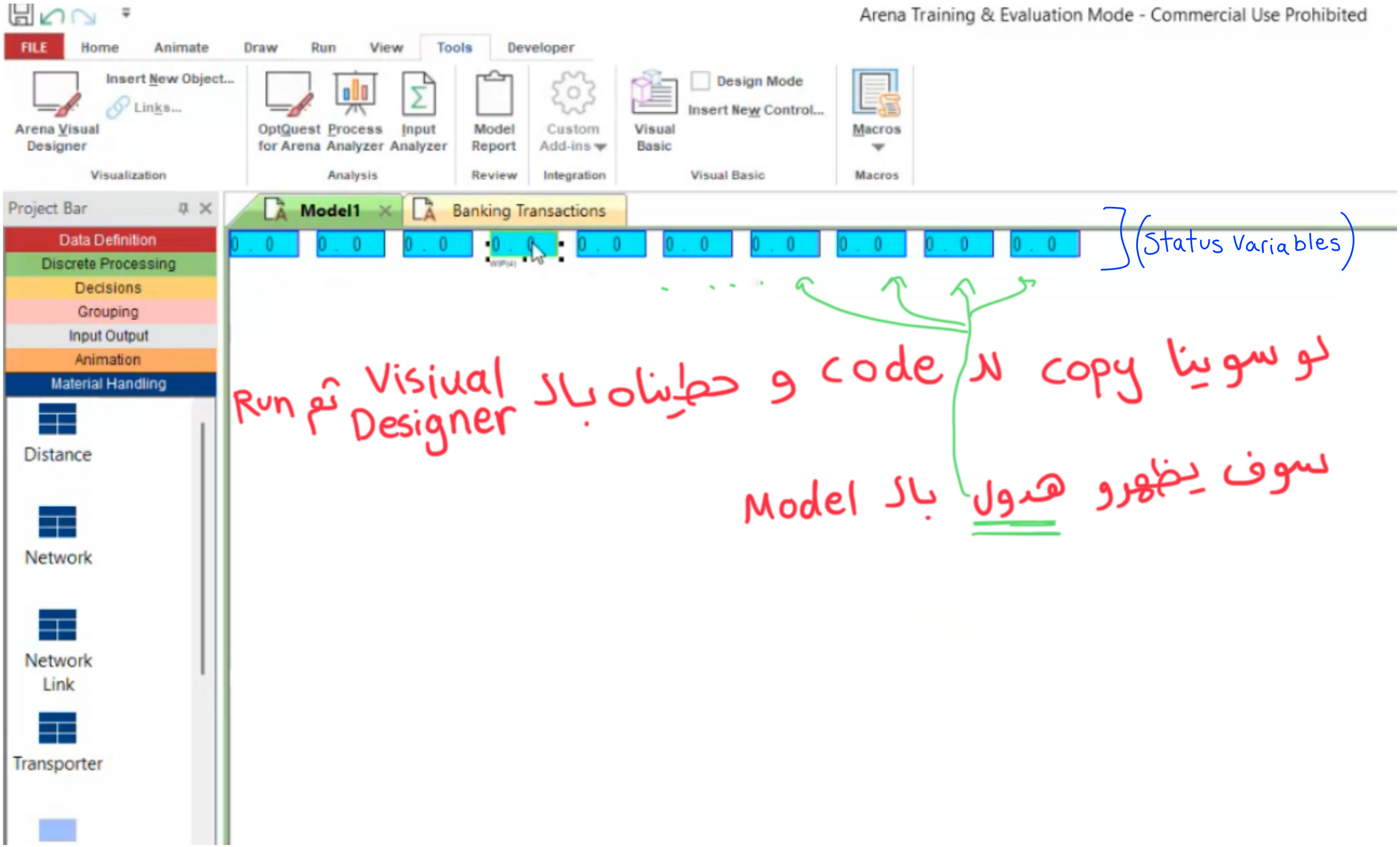
نسوي
copy
و نسوي
paste
داخل ال
Visiual
Designer

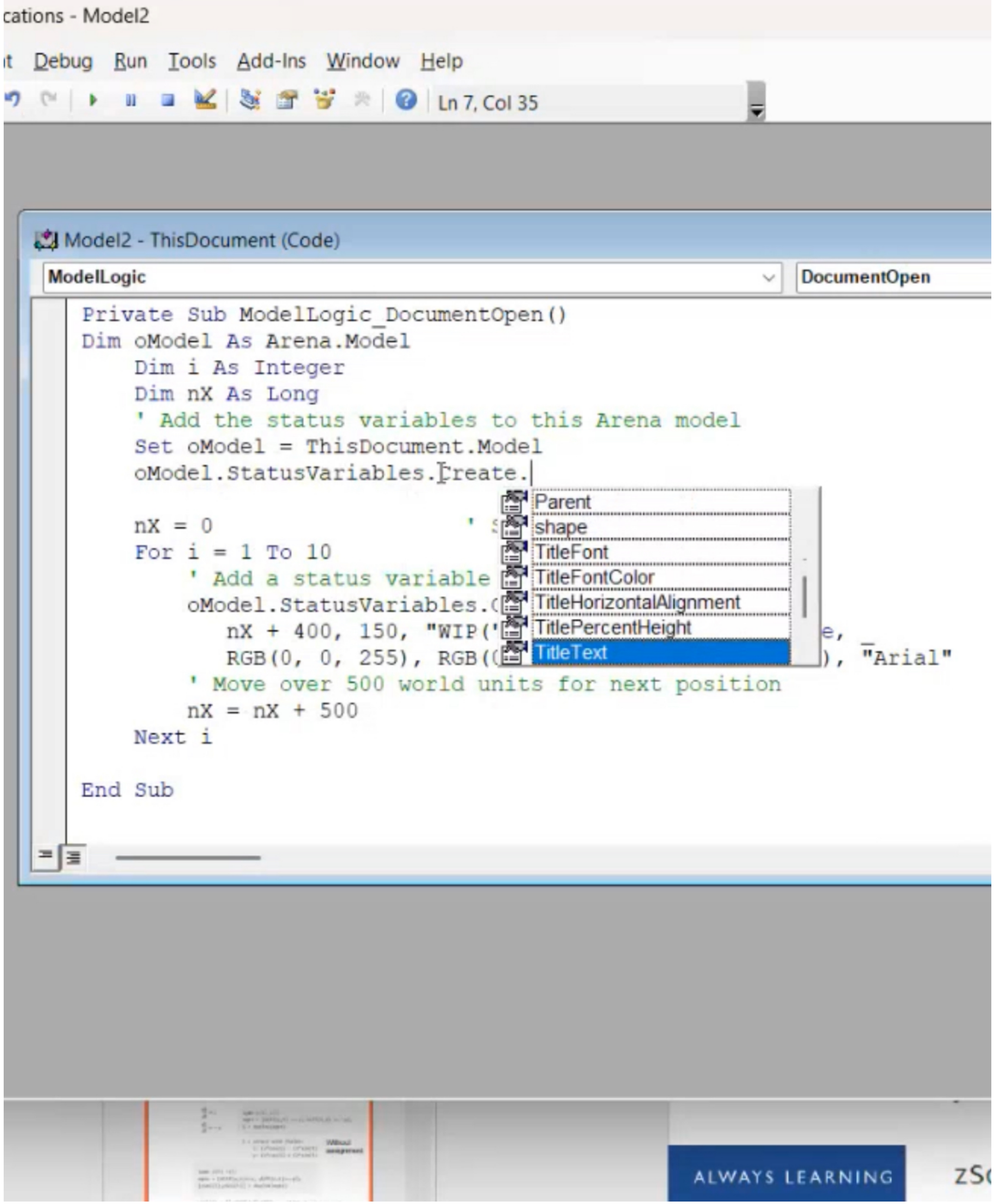
```
Dim oModel As Arena.Model
Dim i As Integer
Dim nX As Long

' Add the status variables to this Arena model
Set oModel = ThisDocument.Model

nX = 0                      ' Start at x position 0
For i = 1 To 10
    ' Add a status variable to the model window
    oModel.StatusVariables.Create nX, 0, _
        nX + 400, 150, "WIP(" & i & ")", "**.*", False, _
        RGB(0, 0, 255), RGB(0, 255, 255), RGB(0, 0, 0), "Arial"
    ' Move over 500 world units for next position
    nX = nX + 500
Next i
```

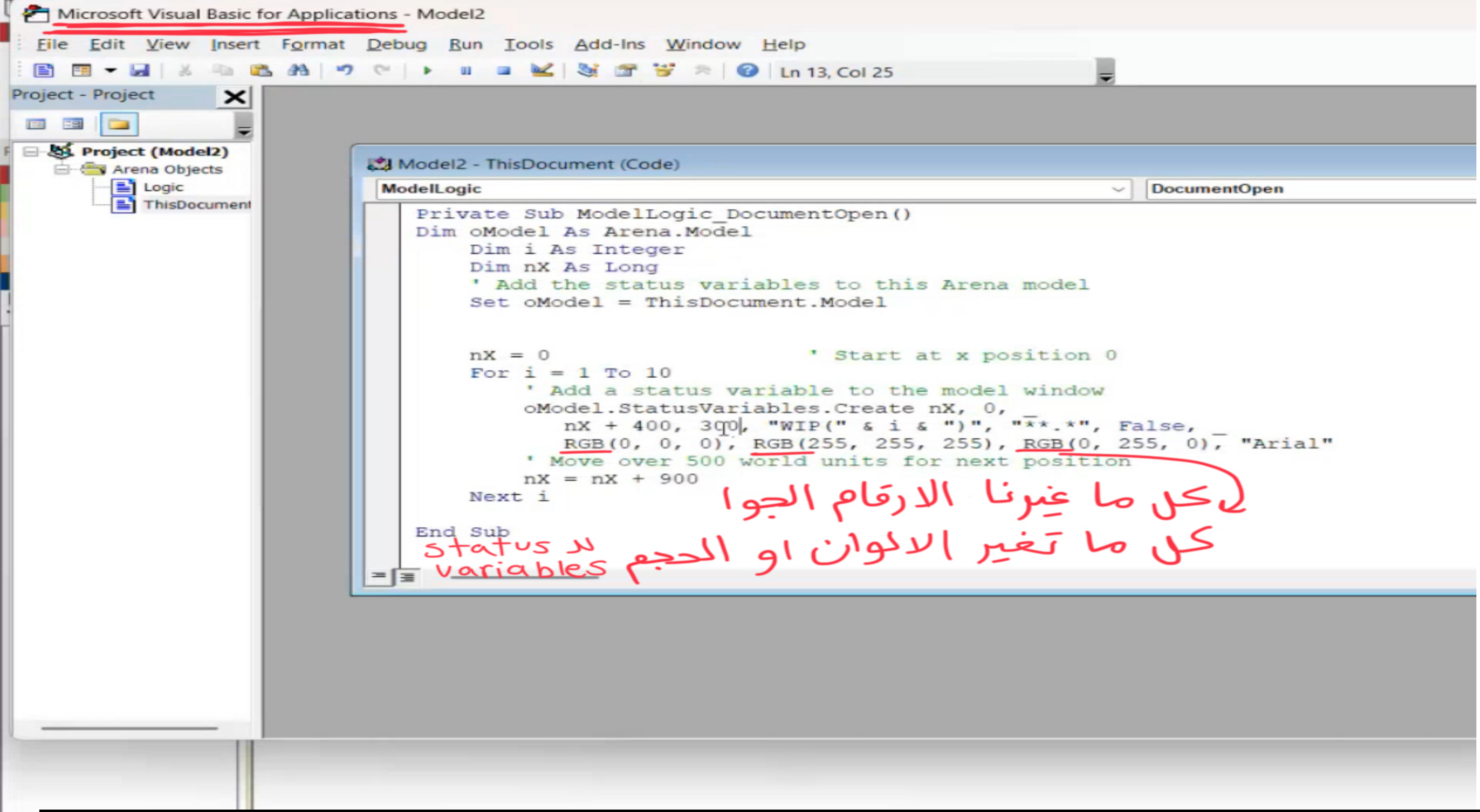



Arena Training & Evaluation Mode - Commercial Use Prohibited
FILE
Home
Animate
Draw
Run
View
Tools
Developer
Insert New Object...
Links...
Arena Visual Designer
Visualization
OptQuest for Arena
Process Analyzer
Input Analyzer
Analysis
Model Report
Review
Custom Add-ins
Integration
Visual Basic
Design Mode
Insert New Control...
Visual Basic
Macros
Macros
Project Bar
Data Definition
Discrete Processing
Decisions
Grouping
Input Output
Animation
Material Handling
Distance
Network
Network Link
Transporter
Model1
Banking Transactions
(Status Variables)
لو سوينا copy لل code و حطيناه بال Visual Designer ثم Run
سوف يظهرو جدول بال Model



يمكن التعديل

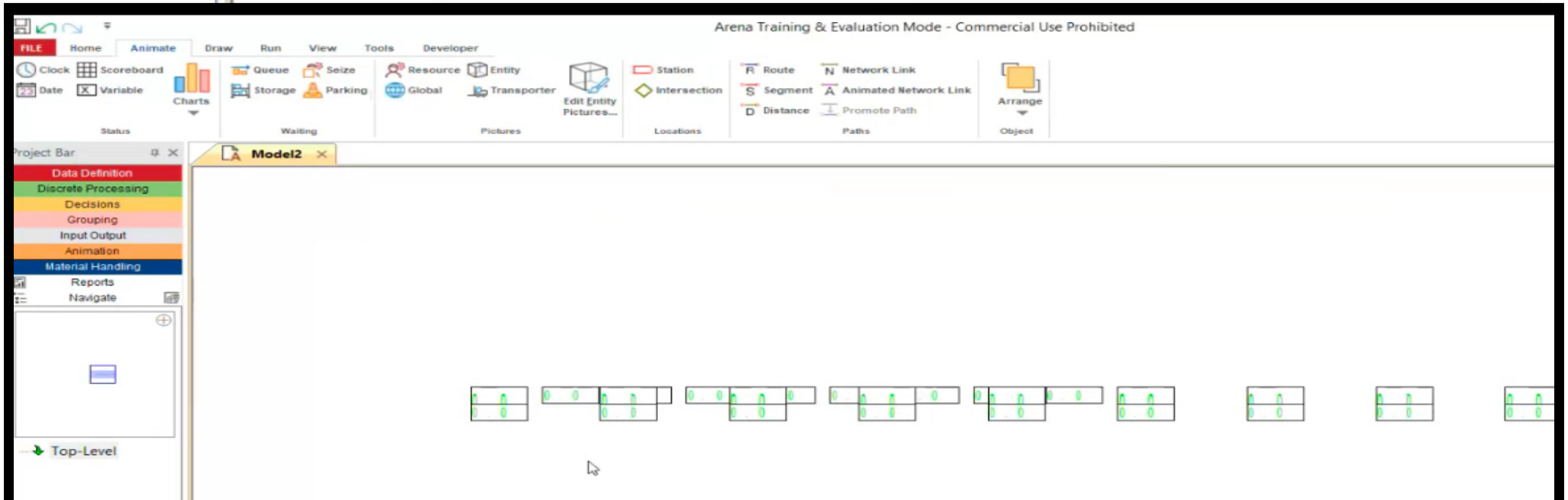
على ال Model



كل ما غيرنا الارقام الجوا

كل ما تغير الالوان او الحجم

لـ status variables



Sample: Assign Variable Value During Run

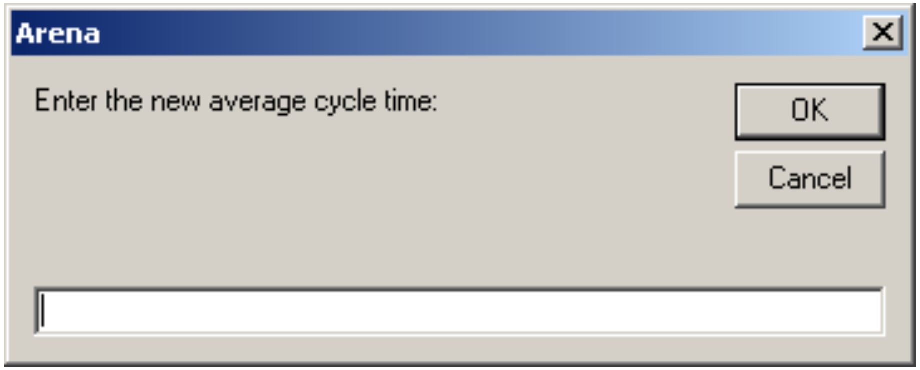
```
Dim oSIMAN As Arena.SIMAN
Dim nVarIndex As Long
Dim sNewValue As String

' Prompt for a new value
sNewValue = InputBox("Enter the new average cycle time:")

' Assign their answer to the Mean Cycle Time variable
```

اسم الـArena زمان

```
Set oSIMAN = ThisDocument.Model.SIMAN
nVarIndex = oSIMAN.SymbolNumber("Mean Cycle Time")
oSIMAN.VariableArrayValue(nVarIndex) = sNewValue
```



in Excell
→ Developer Tab
→ Macro

in Tools Menu

Arena Macro Recorder

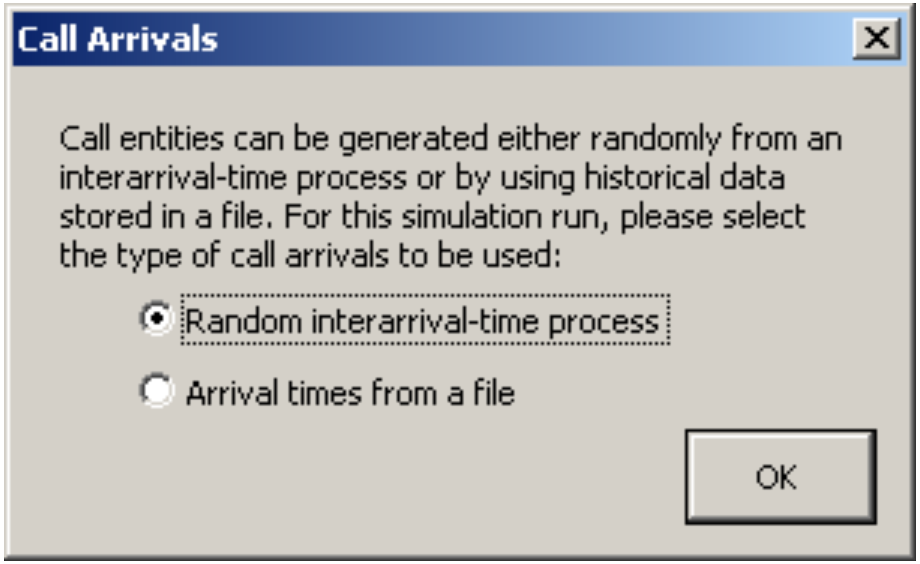
in Arena
→ Tools tab
→ Macro

So we can Record a Macro and take the code for what we did

- A **Macro** is a VBA function to perform a task.
- Macro recording automatically creates the VBA code to reproduce the actions you take while performing the steps in the task.
- Use the **Record Macro** toolbar to start/stop and pause/resume recording.
- Useful for automating repetitive tasks.
- Ideal for learning VBA commands and prototyping functions.

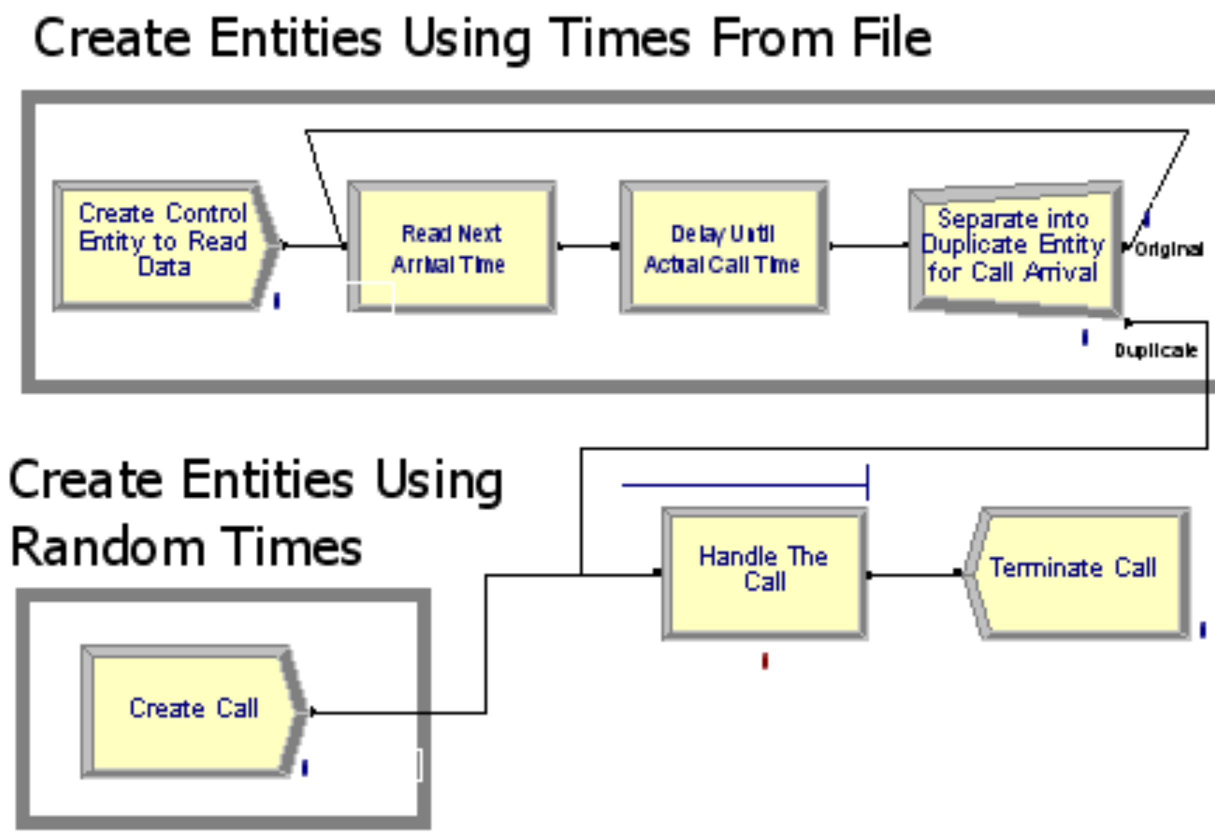
Model 10-05: Presenting Arrival Choices to the User

- **Prompt at beginning of run**
  - Generate entities via random process … or …
  - Generate based on arrival times stored in a file



Our Approach

- **Both sets of logic placed in model window and connected to start of call logic (Process module)**

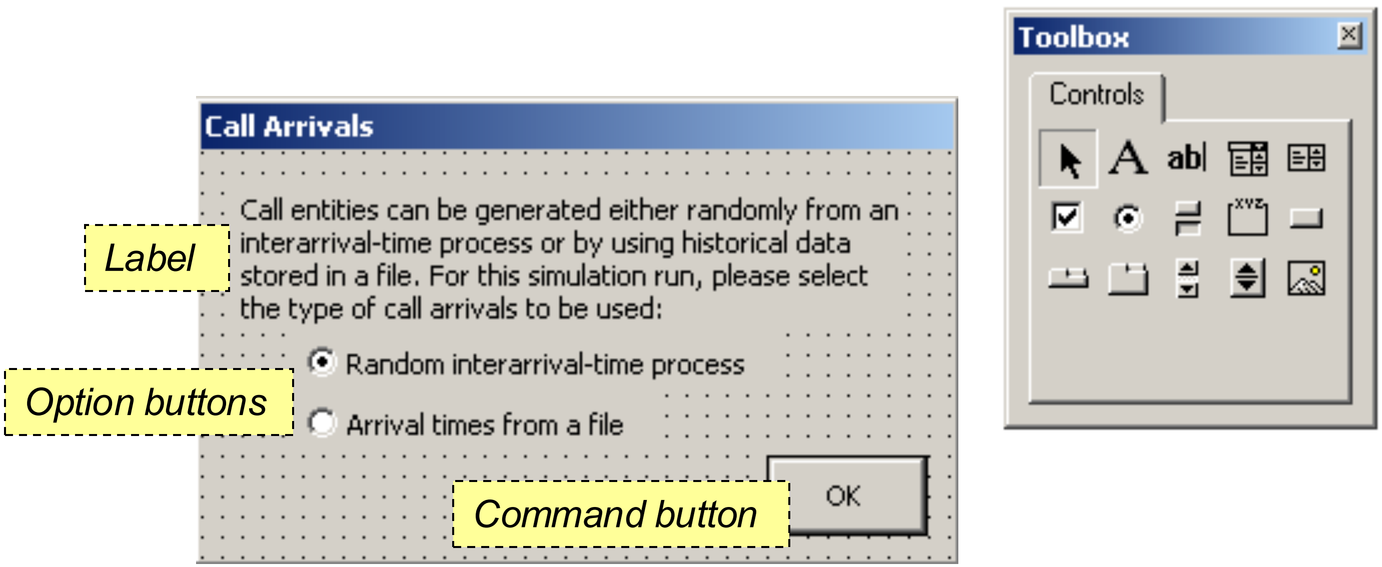


Our Approach (cont'd.)

- **Change Max Arrivals field in Create module to turn "on" or "off" its generation of entities**
- **Random interarrival-time process**
  - ***Create Call*** module: ***Infinite***
  - ***Create Control Entity to Read Data*** module: ***0***
- **Arrival times from a file**
  - ***Create Call*** module: ***0***
  - ***Create Control Entity to Read Data*** module: ***1***
- **Give unique "tag" to each Create module (so VBA code can find them)**

VBA UserForm

- **Insert/UserForm menu in Visual Basic Editor**
- **Drop controls from Control Toolbox (labels, option buttons, command button)**



Show the UserForm

- At beginning of run (ModelLogic_RunBegin), show the form:

```
Option Explicit
Private Sub ModelLogic_RunBegin()
    ' Display the UserForm to ask for the type of arrivals
    frmArrivalTypeSelection.Show

    Exit Sub
End Sub
```

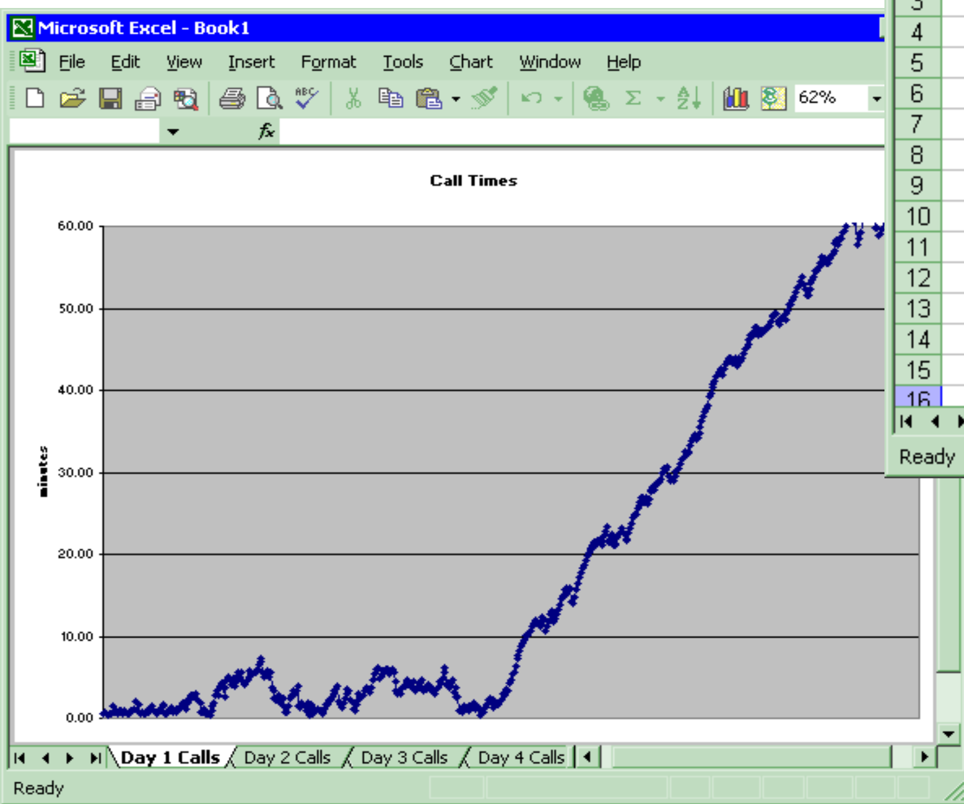
- Program control passes to the form until it's closed
- Arena run "suspended" while form is in control

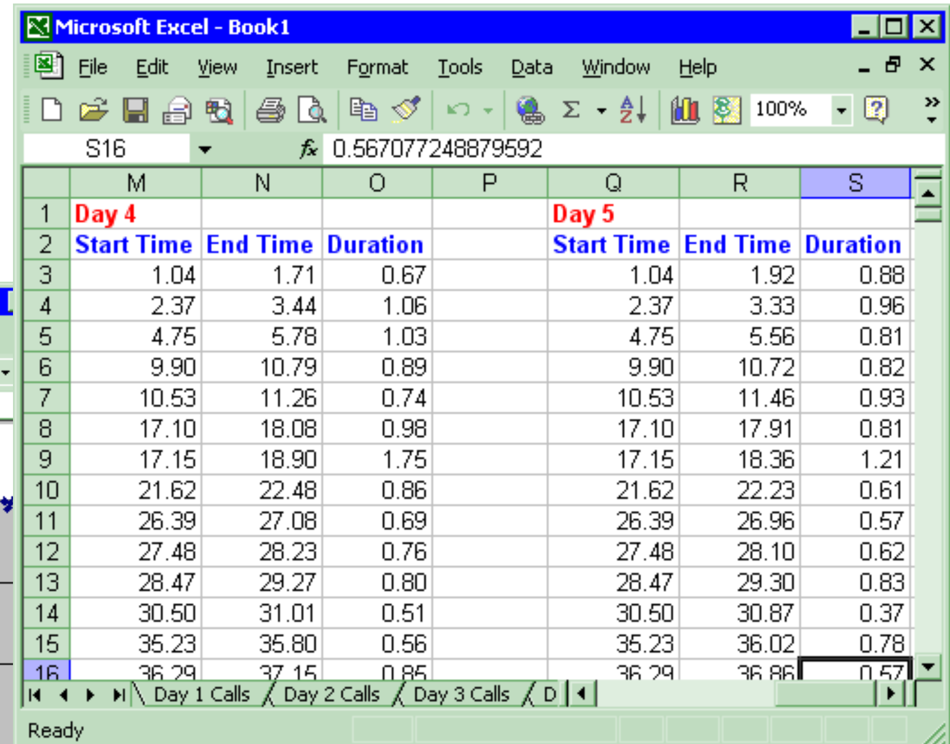
Change Module Data On OK

- **When user clicks OK button on form, modify the Create module data**
  - Find the Create modules
  - Set the Max Arrivals fields
  - Play a sound
  - Close the UserForm
- **When form is closed, simulation run commences with the new data values in the Create modules**

Model 10-06: Record Call Data in Microsoft Excel

- **Our goal:**
  - Raw call data tables
  - Daily call duration charts





	M	N	O	P	Q	R	S
1	Day 4				Day 5		
2	Start Time	End Time	Duration		Start Time	End Time	Duration
3	1.04	1.71	0.67		1.04	1.92	0.88
4	2.37	3.44	1.06		2.37	3.33	0.96
5	4.75	5.78	1.03		4.75	5.56	0.81
6	9.90	10.79	0.89		9.90	10.72	0.82
7	10.53	11.26	0.74		10.53	11.46	0.93
8	17.10	18.08	0.98		17.10	17.91	0.81
9	17.15	18.90	1.75		17.15	18.36	1.21
10	21.62	22.48	0.86		21.62	22.23	0.61
11	26.39	27.08	0.69		26.39	26.96	0.57
12	27.48	28.23	0.76		27.48	28.10	0.62
13	28.47	29.27	0.80		28.47	29.30	0.83
14	30.50	31.01	0.51		30.50	30.87	0.37
15	35.23	35.80	0.56		35.23	36.02	0.78
16	36.29	37.15	0.85		36.29	36.86	0.57

Using ActiveX Automation in VBA

- **Reference the Excel Object Library**
  - Tools/References menu in Visual Basic Editor
  - Check the Microsoft Excel Object Library
  - Establishes link between Arena VBA and Excel
- **Object variables from application's object model**
  - *Excel.Application*, *Excel.Workbook*
  - *Arena.SIMAN*
- **Starting Excel**
  - *CreateObject*: starts application, returning "handle" to the program (stored in *oExcelApp* variable)
  - *oExcelApp.Workbooks.Add*: similar to "File/New" in Excel

Retrieving Simulation Data

- **ThisDocument**
  - Built-in variable accessing the Arena model
  - Use only within Arena's VBA

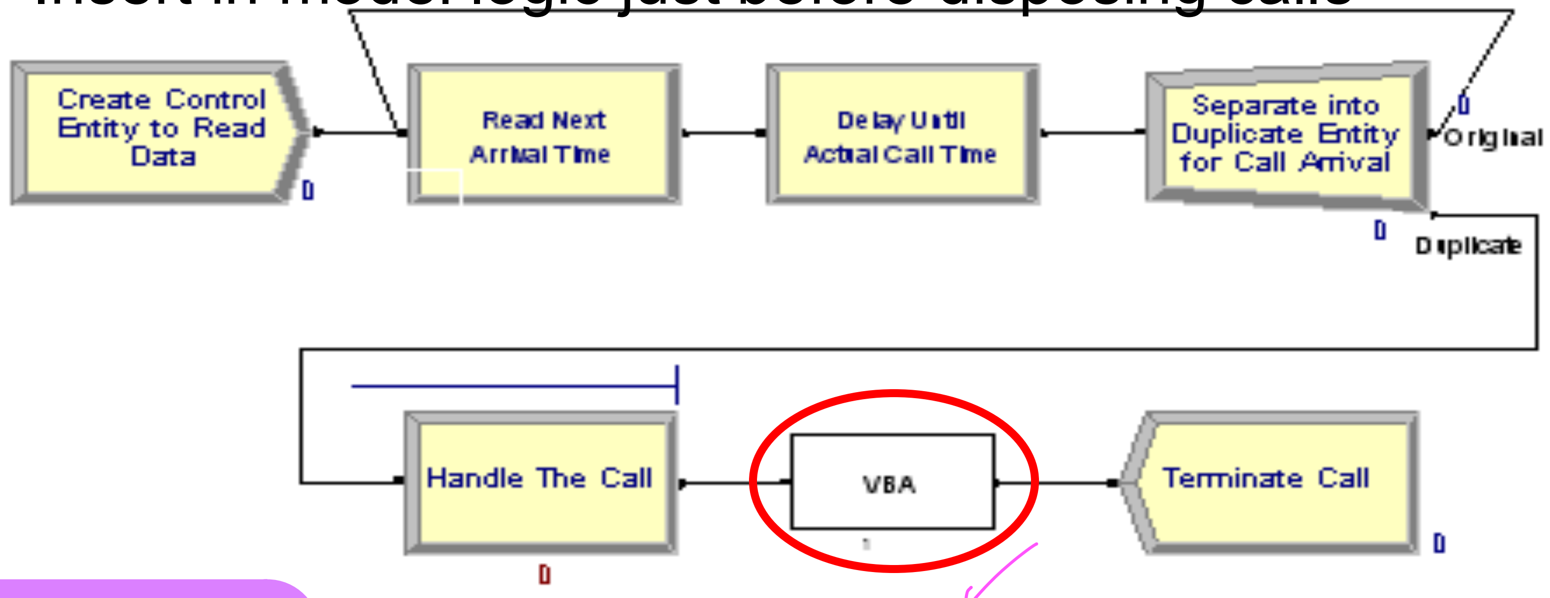
- **ThisDocument.Model.SIMAN**
  - Used to access simulation run data
  - Browse (**F2**) in VBA window for full list of variables
  - Active only when simulation run data is available -- i.e., built-in events:
    - after (and including) ModelLogic_RunBeginSimulation
    - before (and including) ModelLogic_RunEndSimulation

Our Approach

- **VBA ModelLogic_RunBeginSimulation**
  - Called ***once*** at the beginning of the simulation run
    - Start Excel with a new spreadsheet ("Workbook")
    - Format header rows for data worksheet
- **VBA ModelLogic_RunBeginReplication**
  - Called at the beginning of ***each replication***
    - Write headers for the three columns and the Day
    - Format the data columns

Our Approach (cont'd.)

- **VBA Module (Blocks panel)**
  - Insert in model logic just before disposing calls



- **VBA Code**
  - VBA modules numbered as they're placed, with corresponding VBA_Block_<n>_Fire events in VBA

Our Approach (cont'd.)

- **VBA_Block_1_Fire**
  - Called each time an entity enters the VBA Block in the model
  - Retrieve data from running simulation via SIMAN object (stored in *oSIMAN* variable)
  - Row and columns into which to write data stored in global VBA variables (*nNextRow*, *nColumnA*, *nColumnB*, *nColumnC*)

Our Approach (cont'd.)

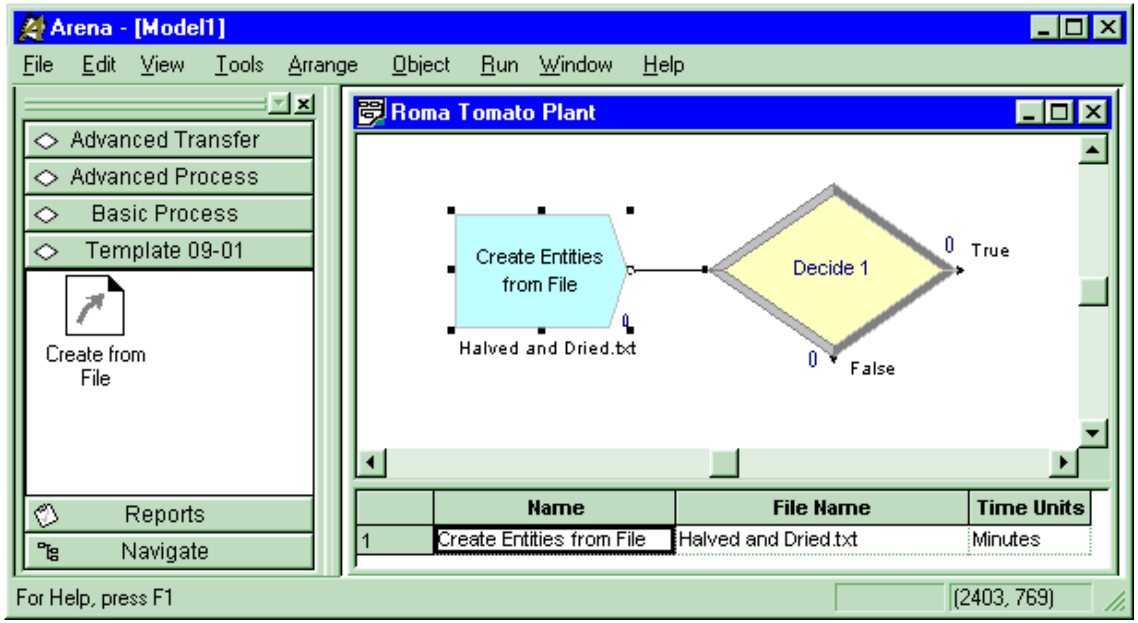
- **ModelLogic_RunEndReplication**
  - Called at end of each replication
  - Creates the chart and updates the global variables
  - Hint: Use Excel macro recording for "skeleton" code (e.g., for formatting commands, creation of chart); copy into Arena VBA and adjust variable names (e.g., add *oExcelApp* to access Excel)
- **ModelLogic_RunEndSimulation**
  - Turn *DisplayAlerts* off (overwrites *.xls* file if it exists)
  - *SaveAs* method to give filename
  - Excel still running. Could use *oExcelApp.Quit*

What We'll Do ...

- Reading and Writing Data Files (ReadWrite)
- ActiveX™ and Visual Basic® for Applications (VBA)
- Creating Modules with Arena Professional Edition

Creating Modules with Arena Professional Edition

- **Template 10-01:**
  ***Create from File* module**
  - Template (*.tpo*) can be attached and used in Academic Arena
  - Place and edit like any other module
  - Need Research/ Professional Arena to create/edit template source (*.tpl*)



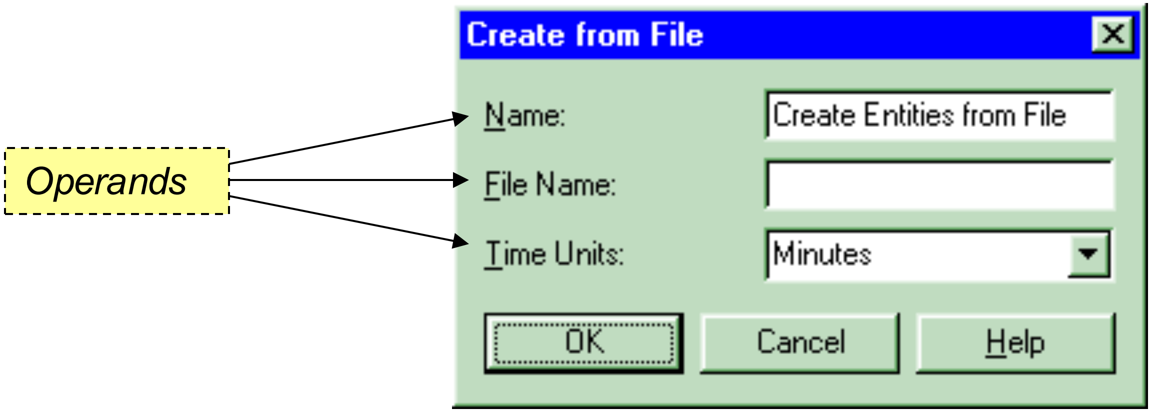
Panel Icon and User View

- ***Panel Icon***: "Button" displayed in template panel to represent the module

- ***User View***: Graphics objects placed in the model window with an "instance" of the module
  - Module handle
  - Entry, exit points
  - Animation
  - Operand values

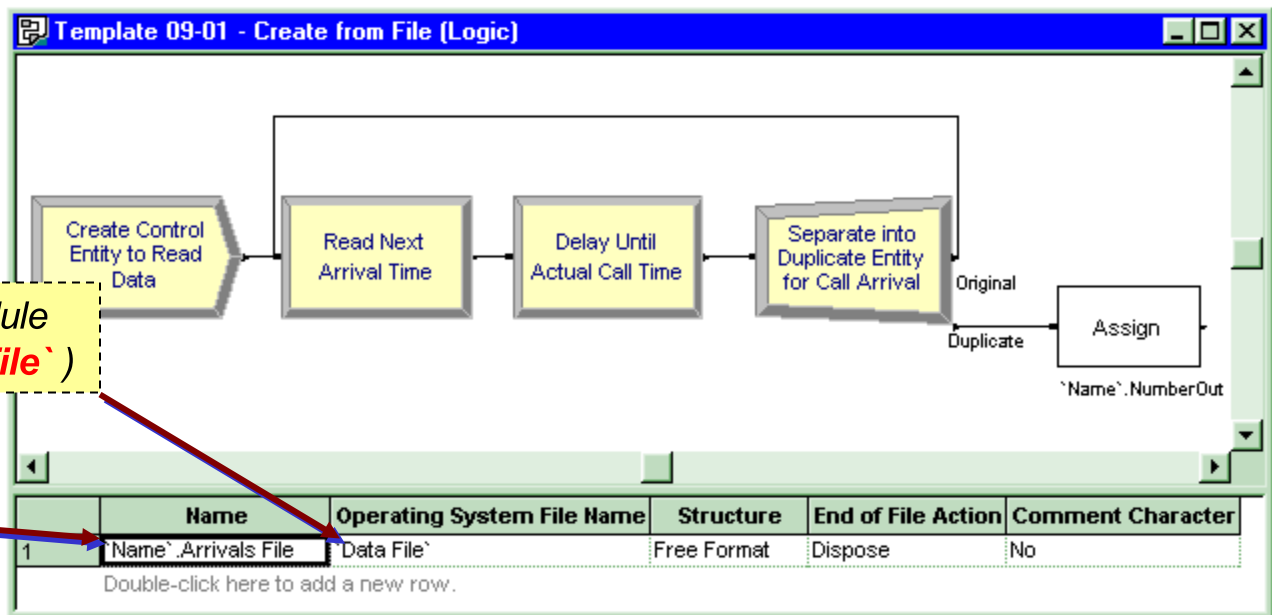
Operands

- ***Operands*: Fields that can be filled out by the modeler**
  - Allow different data for each instance of the same module type in a model
  - Passed down to logic via back quotes (`)
  - Entry and exit point operands permit entities to move through underlying module logic



Module Logic

- ***Module Logic*: A "submodel" containing module instances**
  - Can paste from a model into a module definition's logic window
  - Same interface as for building models



*Reference to module operand (**`Data File`**)*

*Reference to module operand (**`Name`**)*

Uses of Templates

- **Commercial templates**
  - Arena templates (Basic Process, Advanced Process, etc.)
  - Contact Center, Packaging templates
  - …
- **Application-focused templates**
  - Terminology, modeling capabilities designed for a particular application environment (e.g., mining, material handling, order processing)
  - “Personal” / utility templates
  - Reuse what you learn
  - Share your modeling techniques

Summary

- **We have just barely scratched the surface of Arena customization and interactions with other software including:**
  - Reading and writing various types of external data files.
  - Visual Basic for Applications
  - Interacting with Microsoft Office
  - Building custom applications with templates.
- **There are many other ways of customizing Arena and allowing Arena to interact and exchange data with other software.**

Chapter 7 Random-Number Generation

Banks, Carson, Nelson & Nicol

Discrete-Event System Simulation

مبني على فكرة توليد رقم عشوائي .بين ال zero وال one

و بشكل أساسي سوف نستخدم (باقي القسمة) واسم العملية Linear Congruential Method

Purpose & Overview

- Discuss the generation of random numbers.

- Introduce the subsequent testing for randomness:
  - Frequency test
  - Autocorrelation test.

Properties of Random Numbers

- Two important statistical properties:
  - Uniformity
  - Independence.
- Random Number, R_i, must be independently drawn from a uniform distribution with pdf:

$$f(x) = \begin{cases} 1, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

$$E(R) = \int_0^1 x\,dx = \left.\frac{x^2}{2}\right|_0^1 = \frac{1}{2}$$

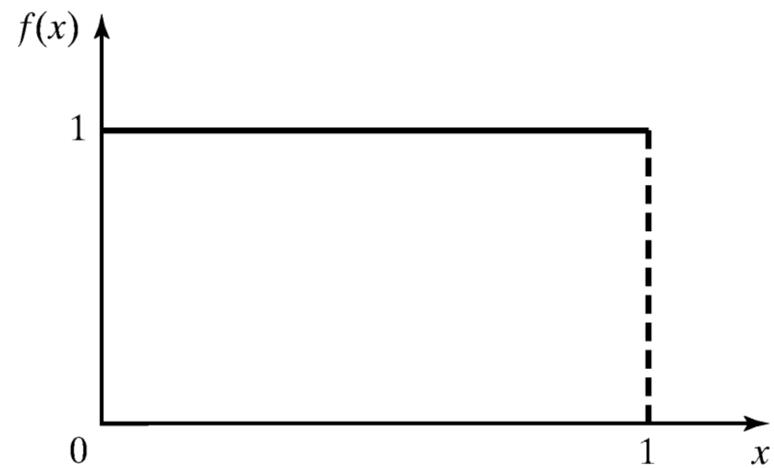


Figure: pdf for random numbers

Generation of Pseudo-Random Numbers

- "Pseudo", because generating numbers using a known method removes the potential for true randomness.
- Goal: To produce a sequence of numbers in $[0,1]$ that simulates, or imitates, the ideal properties of random numbers (RN).
- Important considerations in RN routines:
  - Fast
  - Portable to different computers
  - Have sufficiently long cycle
  - Replicable
  - Closely approximate the ideal statistical properties of uniformity and independence.

Techniques for Generating Random Numbers

- Linear Congruential Method (LCM).
- Combined Linear Congruential Generators (CLCG).
- Random-Number Streams.

Linear Congruential Method

مطلوب
ال Simple Method

[Techniques]

- To produce a sequence of integers, $X_1, X_2, \ldots$ between 0 and $m-1$ by following a recursive relationship:

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0,1,2,\ldots$$

الجواب لايقل عن (الصفر)
ولايزيد عن ال(m-1)

The multiplier

The increment

The modulus

- The selection of the values for a, c, m, and X_0 drastically affects the statistical properties and the cycle length.
- The random integers are being generated $[0, m-1]$, and to convert the integers to random numbers:

the Random number is always between zero and one

$$R_i = \frac{X_i}{m}, \quad i = 1,2,\ldots$$

قسمنا على m
فأصبح الجواب
بين الصفر و الواحد

Example

[LCM]

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:

$a * x + c$

$X_1 = (17*27+43) \bmod 100 = 502 \bmod 100 = 2, \qquad R_1 = 0.02;$

$X_2 = (17*2+43) \bmod 100 = 77, \qquad R_2 = 0.77;$

$X_3 = (17*77+43) \bmod 100 = 52, \qquad R_3 = 0.52;$

…

فقط إلى هنا المطلوب من هذا الفايل

سؤال اكيد جاي بالفاينل

بدل ما يعطينا الارقام لل Hand Simulation احنا بنسوي Generate

Random-Variate Generation

عليه ٦ أو ٧ علامات بالفاينل

Exponential

Uniform

Discrete

Normal

لتحويل الارقام يلي طلعت معنا بين ال zero و ال one إلى Distributions

Need for Random-Variates

- We, usually, model uncertainty and unpredictability with statistical distributions
- Thereby, in order to run the simulation models involving uncertainty, we need to get samples from these statistical distributions
- Here, we assume that the distributions (type and parameters) are already specified
- ARENA comes with ready functions to sample from specified distributions.
- It is still useful to know how it is done
- We will generate the variates always using random numbers, whose generation is discussed previously

Inverse Transform Technique -- Exponential Distribution

Simplest Function that can be inverted

- To generate samples from exponential distribution we use the inverse transform technique (we want x as a Function of y)

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Step 1. Compute the cdf of the desired random variable X, F(x).

Step 2. Find the inverse of F(x) function

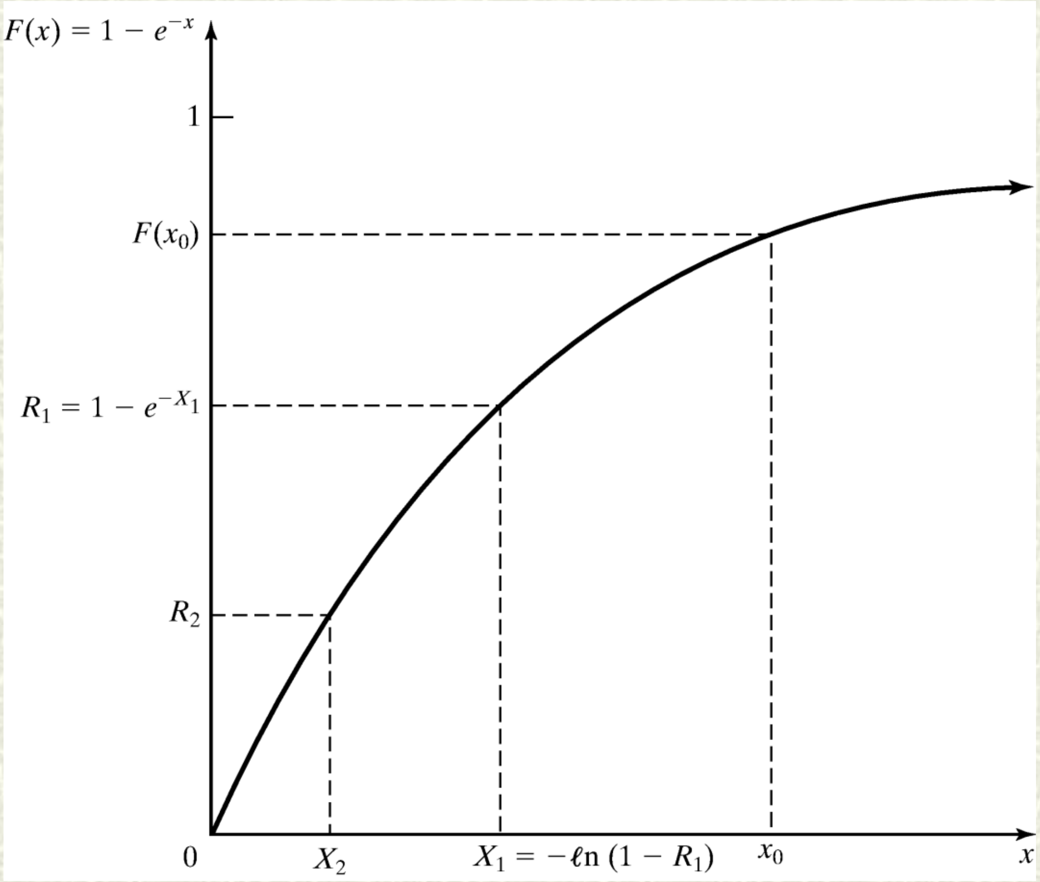
$$y = F(x) \Rightarrow y = 1 - e^{\lambda x} \Rightarrow x = F^{-1}(y) = -\frac{1}{\lambda}\ln(1 - y)$$

is a Random Stream between 0 and 1

Step 4. Generate uniform random variables R_1, R_2, R_3, … and compute the desired random variates by

$$X_i = F^{-1}(R_i) = -\frac{1}{\lambda}\ln(1 - R_i)$$

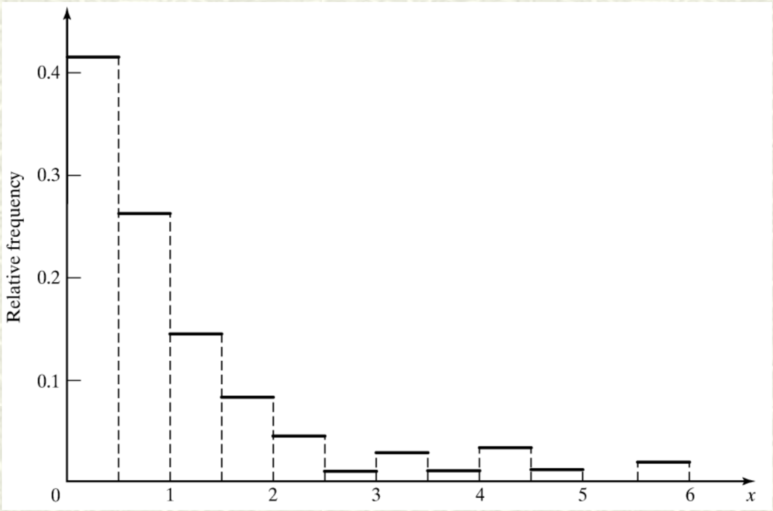
Inverse Transform Technique -- Exponential Distribution



Inverse Transform Technique -- Exponential Distribution

- Example: Generate *200* variates X_i with distribution $exp(\lambda = 1)$

Histogram



  - Check: Does the random variable X_1 have the desired distribution?

Proof?

- Can you prove that the numbers you have generated are indeed samples from an exponential distribution?

Other Distributions

- Uniform Distribution [UN(a,b)] (X = a + (b-a)R)
  - Does it really work?
- Weibull Distribution

$$f(x) = \begin{cases} \dfrac{\beta}{\alpha^{\beta}} x^{\beta-1} e^{-\left(x/\alpha\right)^{\beta}}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

  - Derive the transformation
- Triangular ….
- The moral is if we can find a closed-form inverse of the cdf for a distribution we can use this method to get samples from that distribution

Continuous Functions without a Closed-Form Inverse

- Some distributions do not have a closed form expression for their cdf or its inverse (normal, gamma, beta, …)
- What can be done then?

R ⟶ رقم بين الصفر و الواحد

- Approximate the inverse cdf
- For the standard normal distribution:

$$X = F^{-1}(R) \approx \frac{R^{0.135} - (1-R)^{0.135}}{0.1975}$$

- This approximation gives at least one-decimal place accuracy in the range [0.0012499, 0.9986501]

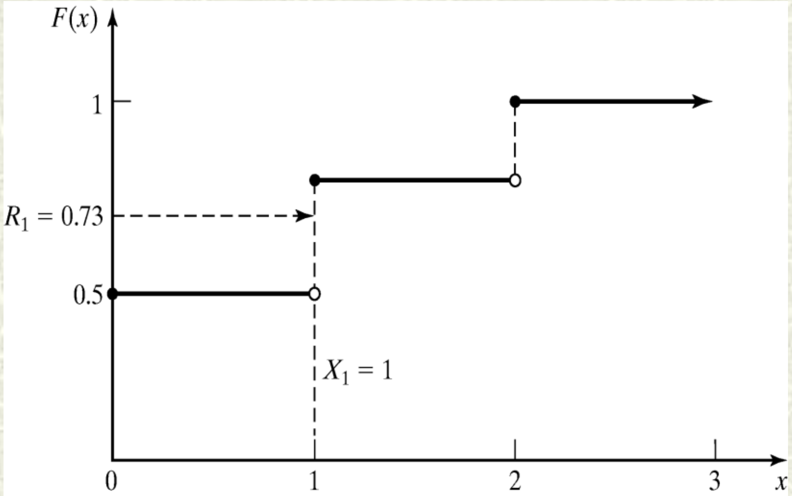
Discrete Distributions

This CDF Kind of have Jumps

- An Empirical Discrete Distribution
  - p(0) = P(X=0) = 0.50
  - p(1) = P(X=1) = 0.30
  - p(2) = P(X=2) = 0.20 عشرين % من الحالات

$$F(x) = \begin{cases} 0, & x < 0 \\ 0.5, & 0 \le x < 1 \\ 0.8, & 1 \le x < 2 \\ 1.0, & 2 \le x \end{cases}$$

commulative



- Can we apply the inverse transform technique?

Discrete Distributions

- Let $x_0 = -\infty$, and $x_1, x_2, \ldots, x_n$, be the ordered probability mass points for the random variable X
- Let R be a random number

If Statement

$$if \quad F(x_{i-1}) < R \leq F(x_i) \;\Rightarrow\; X = x_i$$

Discrete Distributions

- A Discrete Uniform Distribution

$$p(x) = \frac{1}{k}, \quad x = 1, 2, \ldots, k$$

$$F(x) = \begin{cases} 0, & x < 1 \\ \frac{i}{k}, & i \le x < i+1, \ for \ \ i \in [1, k-1] \\ 1, & k \le x \end{cases}$$

$$if \quad \frac{i-1}{k} < R \le \frac{i}{k} \quad \Rightarrow \quad X = i$$

$$or \quad X = roundup(kR)$$

Discrete Distributions

- The Geometric Distribution

$$p(x) = p(1-p)^x \,, \quad x = 0, 1, 2, \ldots$$

$$F(x) = 1-(1-p)^{x+1}, \quad x = 0, 1, 2, \ldots$$

- Some algebraic manipulation and …

$$X = \ roundup\left(\frac{\ln(1-R)}{\ln(1-p)} - 1\right)$$

?

يعني الرقم يلي مش ضمن الـRange ما بقبله

Acceptance-Rejection Technique

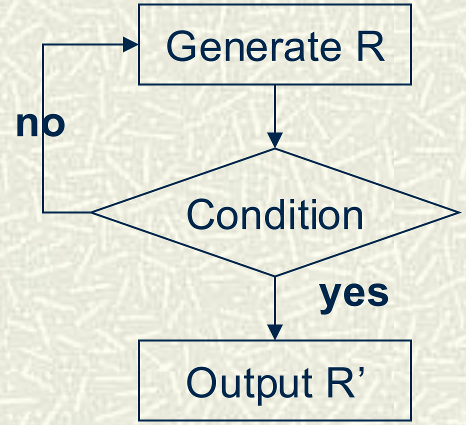
- Useful particularly when inverse cdf does not exist in closed form, a.k.a. thinning
- Illustration: To generate random variates, $X \sim U(1/4,\ 1)$

Procedures:

Step 1. Generate R ~ U[0,1]

Step 2a. If R >= ¼, accept X=R.

Step 2b. If R < ¼, reject R, return to Step 1



- R does not have the desired distribution, but R conditioned (R') on the event $\{R \geq ¼\}$ does.
- Efficiency: Depends heavily on the ability to minimize the number of rejections.

Acceptance-Rejection Technique

- Poisson Distribution

$$p(n) = P(N = n) = \frac{e^{-\alpha}\alpha^n}{n!}, \quad n = 0, 1, 2, \ldots$$

  - N can be interpreted as number of arrivals from a Poisson arrival process during one unit of time
  - Then time between the arrivals in the process are exponentially distributed with rate α

$$N = n \iff \sum_{i=1}^{n} A_i \le 1 < \sum_{i=1}^{n+1} A_i$$

Acceptance-Rejection Technique

$$\sum_{i=1}^{n} A_i \le 1 < \sum_{i=1}^{n+1} A_i \Leftrightarrow \sum_{i=1}^{n} -\frac{1}{\alpha}\ln R_i \le 1 < \sum_{i=1}^{n+1} -\frac{1}{\alpha}\ln R_i$$

$$\Leftrightarrow \prod_{i=1}^{n} R_i \ge e^{-\alpha} > \prod_{i=1}^{n+1} R_i$$

- Step 1. Set n = 0, and P = 1
- Step 2. Generate a random number R_{n+1} and let P = P. R_{n+1}
- Step 3. If $P < e^{-\alpha}$, then accept N = n. Otherwise, reject current n, increase n by one, and return to step 2
- How many random numbers will be used on the average to generate one Poisson variate?

Direct Transformations

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dx$$

- Consider two normal variables Z_1 and Z_2

$$B^2 = Z_1^2 + Z_2^2 \sim \boxed{Chi - square} \text{ with two degrees of freedom}$$

(Exponential with parameter 2)

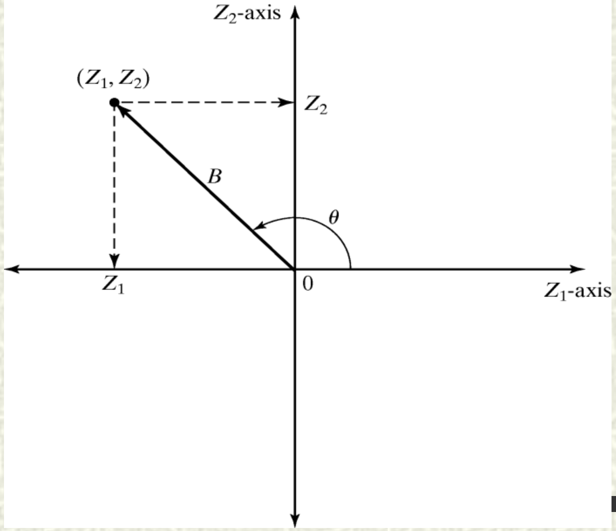
$$\theta = \tan^{-1}\left(\frac{Z_1}{Z_2}\right) \sim Uniform[0, 2\pi]$$

الطول $\rightarrow$ $B = \sqrt{-2\ln R}$

they are two Random Normally distributed

$$Z_1 = \sqrt{-2\ln R_1}\cos(2\pi R_2)$$

$$Z_2 = \sqrt{-2\ln R_1}\sin(2\pi R_2)$$



Direct Transformation

- Approach for normal(μ, σ^2):
  - Generate $Z_i \sim N(0,1)$

$$\boxed{X_i = \mu + \sigma Z_i}$$

- Approach for lognormal(μ, σ^2):
  - Generate $X \sim N(\mu, \sigma^2)$

$$\boxed{Y_i = e^{X_i}}$$

اطلع
X

Convolution Method

- Erlang Distribution
  - An Erlang-K random variable X with parameters (K, θ) (1/θ is the mean, K is the stage number) can be obtained by summing K independent exponential random variables each having mean 1/(K θ)

$$X = \sum_{i=1}^{K} X_i$$

$$X = \sum_{i=1}^{K} -\frac{1}{K\theta} \ln R_i = -\frac{1}{K\theta} \ln\left(\prod_{i=1}^{K} R_i \right)$$

Hand Simulation

تم بنعمل