



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Introduction

Module Overview

Last Module: We looked at general principles behind simulation languages. How do they work?

This Module: We'll learn about Arena, a popular simulation language.

Idea: Go through a series of tutorials that will allow you to fearlessly simulate real-world systems.

More Overview

This module is built on numerous mini-tutorials.

Many will be self-contained and will focus on a specific topic...

...Yet the tutorials are organized in groups that build to a common goal (e.g., a non-trivial demo).

Each lesson contains written notes, but you must **WATCH** the videos to see the “live” stuff.

I Gotta Have More Overview!

.

Rough Groupings of Topics:

- A. Introductory material ← now
- B. Modeling multi-channel systems
- C. Building to a call center model
- D. Some demos on interesting miscellaneous models

Pre-Show Notes...

- Arena is easy, but if it's not your cup of tea (whatever reason),...
- ...There are many other simulation languages around.
- Arena is a good choice to enable you to learn those languages.
- In any case, we'll do lots of examples from which you may find useful applications.

The Next Few Lessons

Our first job is to cover some basics:

- Process-Interaction [← next](#)
- Let's Meet Arena!
- The Arena Basic Template
- Create-Process-Dispose Modules
- The Process Module
- Resources, Schedules, Queues
- The Decide Module

Summary

This Time: Discussed what's coming up in this module on the Arena computer simulation language.

Next Time: A brief review of the Process-Interaction modeling approach. We'll use this to get our subsequent terminology straight and to re-introduce P-I.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Process-Interaction Review

Lesson Overview

Last Lesson: We simply discussed what's coming up in this Arena module.

This Lesson: A quick review of the Process-Interaction approach that Arena uses.

Idea: It's just a flowchart!

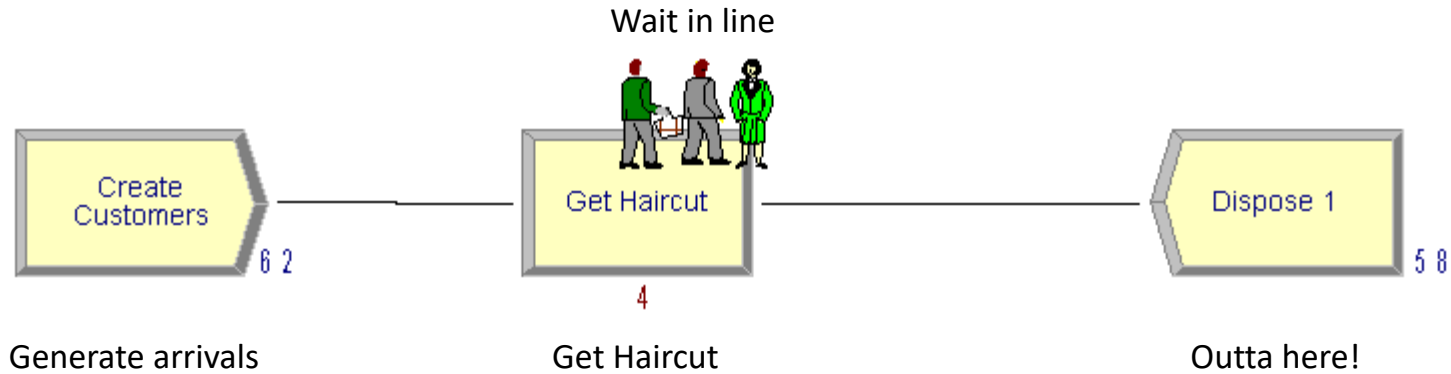
Process-Interaction

- Consider a generic customer (*entity*) and the sequence of events and activities (*processes*) it undergoes as it moves through the system.
- At any time, the system may have many entities interacting with each other as they compete for resources.
- Arena takes this *process-interaction* “world view.”

P-I (cont'd)

- The entities flow through a **network of modules (blocks)** that describe their behavior.
- The network can be represented as a process **flowchart**.
- Example: People show up at the barber, get served (maybe after waiting in line), and then leave.
- In Arena:
 - Create (generate cust arrivals)
 - Process (use the barber)
 - Dispose (outta here)

If It Quacks Like a Flowchart...



Summary

This Time: Talked about the Process-Interaction modeling approach and how it relates to Arena.

Next Time: A first look at Arena. How to get it, and a sneak peak.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Let's Meet Arena!

Lesson Overview

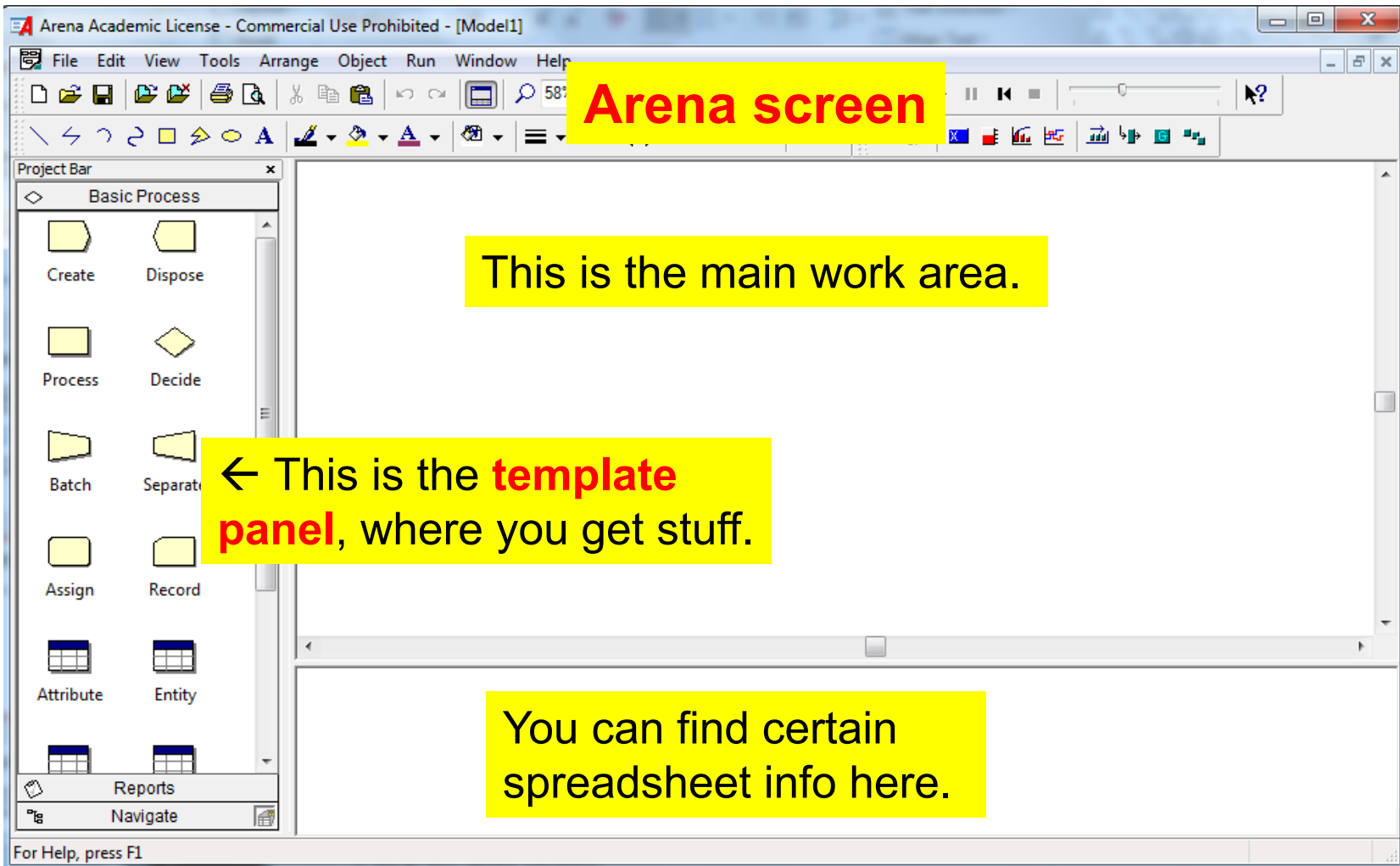
Last Lesson: Reviewed the Process-Interaction approach. This is what Arena uses.

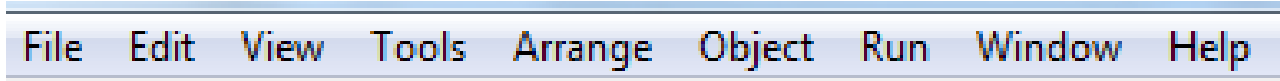
This Lesson: Let's finally meet Arena!

We'll show how to get the software, and then take a first look.

Getting Arena

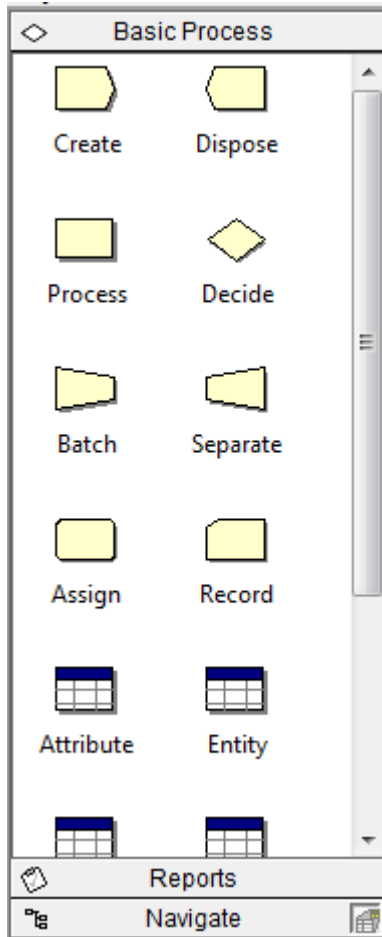
- Easy to download and install free student version of Arena.
- <https://www.arenasimulation.com/academic/students>
- This is a Windows product.
- If you're indeed using Windows, then there is certain Arena stuff you'll occasionally use that's stored deep down in a "Rockwell Software" directory (don't worry about this for now).
- If you don't have Windows, it ain't a problem...
- You can also use Arena via the Georgia Tech virtual labs.
- Or you can really be nerdy and partition your disk to use a Windows OS.





- File: Lets you do the usual New, Open, Close, Save,..., but also allows you to import different template panels and background pix.
- Edit: Allows you to edit entity pix, insert objects, other nice stuff.
- View: See various toolbars, customize your “named views”, etc.
- Tools: Lots of cool toys, including Input Analyzer, OptQuest, AVI capture, macros,...
- Arrange, Object, Window: Various visualization aids.
- Run: Set a run up and make it go as fast as you want or step-by-step.





- This is the **Basic Process** template panel.
- It does very basic stuff that a Univ. of Georgia student might be able to handle on a good day.
- The panel consists of **modules** (I call 'em **blocks** sometimes) such as Create, Dispose, Process, etc., that will be connected together as a flowchart for our simulation model.
- The other items such as Attribute, Entity, etc., are related to **spreadsheets**.
- **Demo Time!** Look at the screen, go buttons, speed, zooming, etc.

Summary

This Time: Finally got to meet Arena! We downloaded Arena and then reviewed what the main screen looks like.

Next Time: We'll look at the Basic Process template in more detail. The components of this template will allow us to put together our first simulations.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

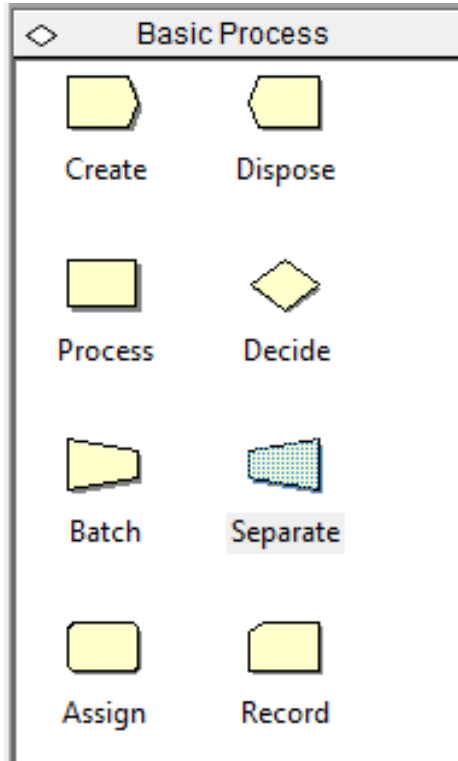
Basic Process Template

Lesson Overview

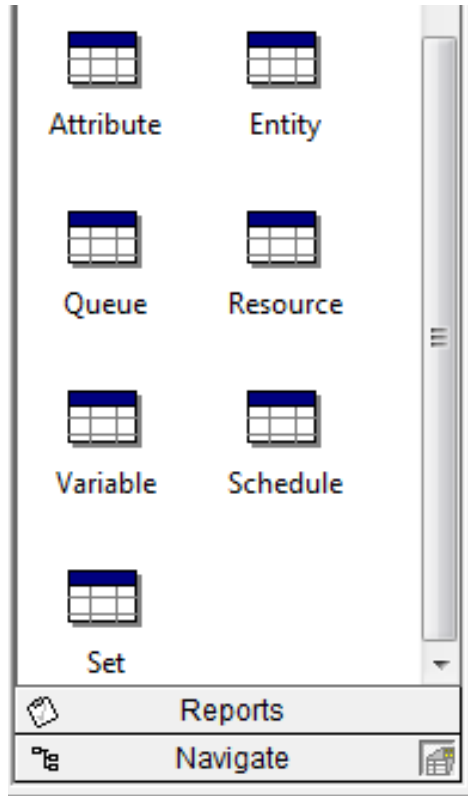
Last Lesson: We were introduced to Arena and went on our first date with it.

This Lesson: We'll look at the Basic Process template in some detail. The items from this template will allow us to put together our initial simulations.

Idea: Click and drag items to build the simulation.



- This is the top half of the **Basic Process** panel.
- It does very basic stuff that a Univ. of Georgia student might be able to handle on a good day.
- More-advanced templates available for non-UGA students: File > Template Panel > Attach
- The panel consists of **modules** such as Create, Dispose, Process, etc., that will be connected together as a flowchart representation for our simulation model.
- Example: **Create** generates customer arrivals.
- You'll drag these modules over to the work area to build the flowchart, fill in some numbers, and then hit the go button to run the simulation.



- This is bottom half of the **Basic Process** panel.
- These items such as Attribute, Entity, etc., are **spreadsheets** that are both informational and which will allow us to change certain system parameters.
- Example: The **Variable** spreadsheet defines global quantities such as WIP that are updated as the simulation progresses. **Resource** keeps track of the names and capacities of the different resources (servers).
- **Demo Time!** Look at a couple of modules and spreadsheets. Load a new template. Build a model.

Summary

This Time: Reviewed the modules and spreadsheets available on the Basic Process template.

Next Time: We'll learn about the Create, Process, and Dispose modules, and then build our first official simulation.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Create-Process-Dispose
Modules

Lesson Overview

Last Lesson: Learned about the Basic Process template.

This Lesson: We'll discuss the Create, Process, and Dispose modules, and then build our first official simulation.

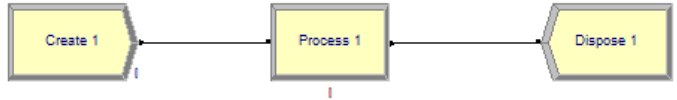
Idea: The modules are easy, but it turns out that there's a lot going on inside of them.

Create-Process-Dispose

- Create: Periodically generate customer arrivals.
- Process: Have work performed on the customers; maybe have to wait in line.
- Dispose: Customers leave the system after service.
- Fun Facts:
 - “Dispose” is called “Terminate” in the language GPSS
 - “Send to Die” in Automod.

How to Use

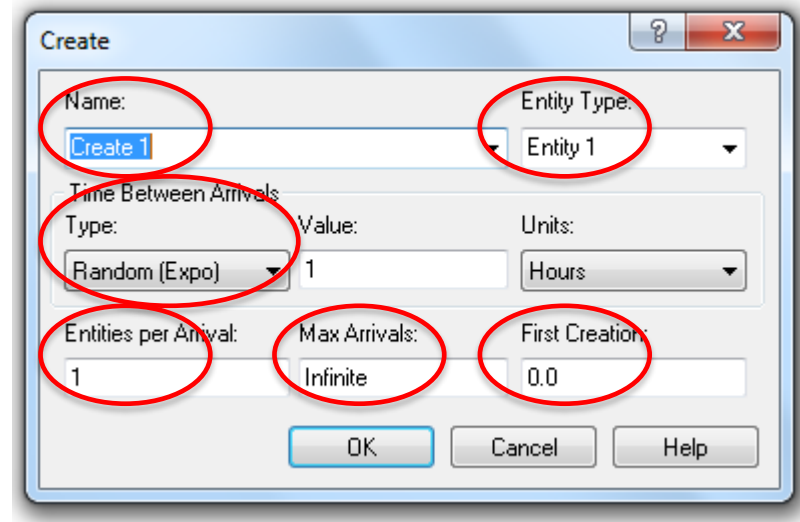
- Drag and drop from template.
- Modules (usually) connect automatically.



- Connects are “instantaneous” for customers.
- Hit “go” and see guys move...
- ...But customers don’t look good and no lines form. Be patient and see demos.

Deeper Dive into Create

- Click into modules and see required inputs.
- Example: Create
- Fields for
 - The name of the module
 - The type of entity (customer)
 - Interarrival distribution
 - # of customers per arrival
 - Maximum # of arrivals allowed
 - Time that the first guy shows up



Process and Dispose Modules

- Process module has fields for
 - The name of the module
 - Type of action. This is where you can try to reserve a server or free a server who's currently in use. **More on this in next lesson.**
 - How long will you be delayed?
I.e., what is the service time?
- Dispose module
 - Nothing deep here – you just get rid of entities.
 - You can name the module.

The screenshot shows the 'Process' dialog box with the following settings:

- Name:** Process 1
- Type:** Standard
- Logic:**
 - Action:** Delay (circled in red)
- Delay Type:** Triangular (circled in red)
- Units:** Hours
- Allocation:** Value Added
- Minimum:** .5
- Value (Most Likely):** 1
- Maximum:** 1.5
- ☒ Report Statistics

Buttons: OK, Cancel, Help

Demo Time!

- Create-Process-Dispose
- Connecting the modules
- Go over input fields
- Expressions and Build Expression
- CONT and DISC functions
- Some other Arena-defined functions

Summary

This Time: Discussed aspects of the Basic Process template's Create-Process-Dispose modules.

Next Time: Deep dive on the Process module. It does a lot more than you may think!



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Details on the Process

Module

Lesson Overview

Last Lesson: Learned about the Create-Process-Dispose modules.

This Lesson: We'll learn more goodies about the Process module.

Idea: It allows you to grab servers, use them, and then let them go for the next guy to use. And along the way, it automatically sets up a queue!

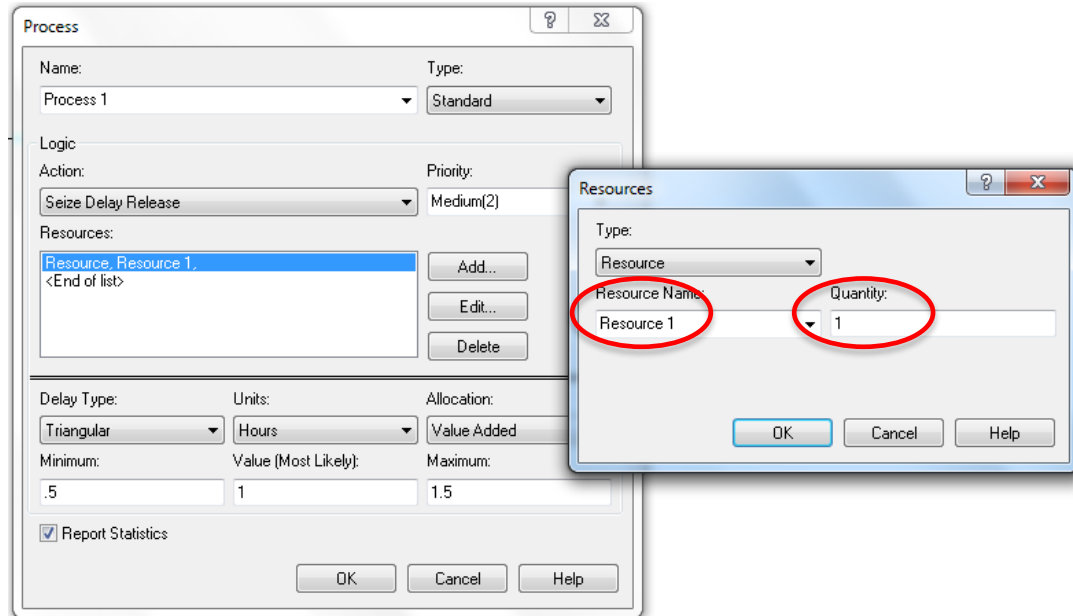
Seize-Delay-Release

You'll take one of the following “actions” in the Process module:

- **Delay:** Spend time in the Process (self-service).
- **Seize-Delay-Release:** Grab at least one resource (server), spend time getting served, and then free the server for the next customer. If you Seize and the server isn't available, you may have to wait in a **queue**. www.youtube.com/watch?v=eAyVgPtC5Zc
- **Seize-Delay:** Grab at least one resource and spend time getting served. Remember to Release the server sometime later, else he gets deadlocked and a giant line will form!
- **Delay-Release:** Use a previously Seize'd server for a while, and then free him for the next guy to use.

Resource Dialog Box

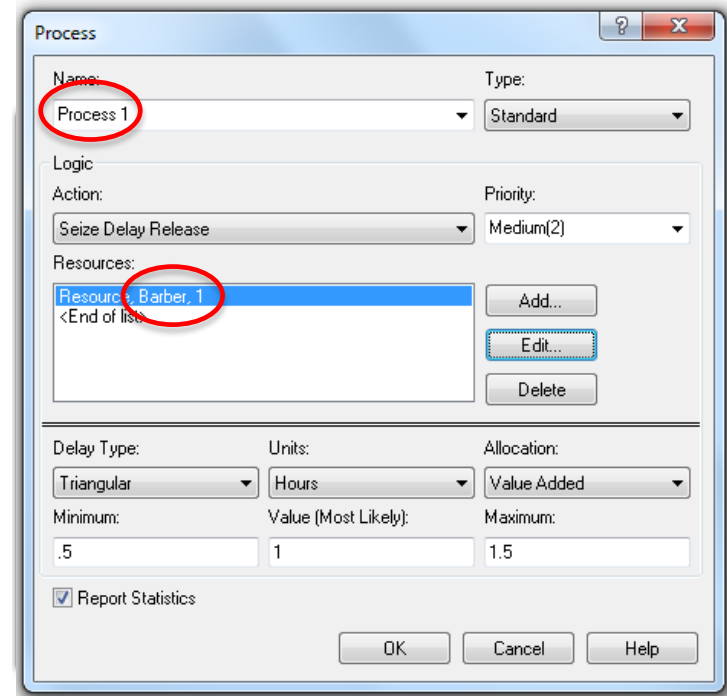
If you do a Seize or Release, a dialog box pops up asking which and how many resource(s) you want to **Add** (Seize) or **Delete** (Release).



Resource Dialog Box (cont'd)

Example:

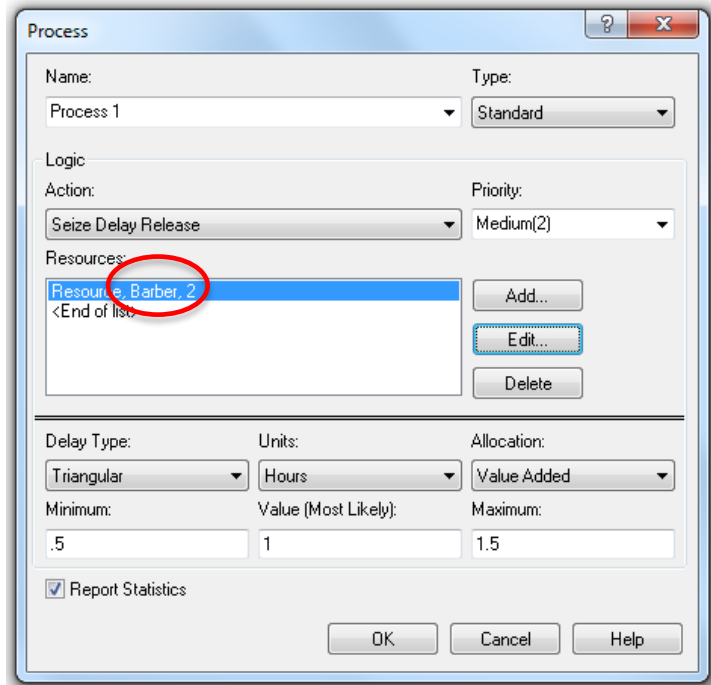
- A customer walks into the Process module, and does a Seize-Delay-Release to grab and use **one** unit of the resource Barber.
- Hit the **Add** button, name the resource Barber, and set Quantity = 1.
- The Process is given a default name of “Process 1” and includes the resource as well as the default queue “Process 1.Queue” (talk about later).
- **Process = Resource + Queue**



Resource Dialog Box (cont'd)

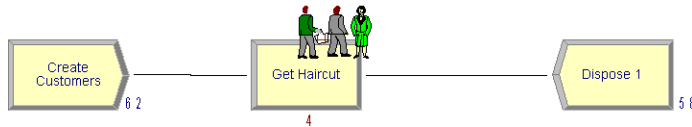
Slightly Different Example: Now every customer uses **TWO** barbers. (They all have lots of flowing locks like your gentle teacher). How to do?

- Set Quantity = 2 in Add or Edit box.
- Distinction: There may be 5 barbers in the store, but each customer needs exactly 2 of them. We'll show in the next lesson how to use the Resource Spreadsheet to set the Barber's "capacity" to 5.



Demo Time!

- Different permutations of Seize-Delay-Release
- Notice that a queue magically pops up!



- Multiple resources
- Warning about deadlocks

Summary

This Time: Learned lots more about the Process module, including the Seize-Delay-Release sequence and the associated queue that magically gets set up.

Next Time: We'll finally take a look at some Arena spreadsheets: Resource, Schedule, and Queue.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Resource, Schedule, and
Queue Spreadsheets

Lesson Overview

Last Lesson: Talked about the Seize-Delay-Release functionality of the Process module – how do you reserve and use servers?

This Lesson: Now it's time to look at some of those spreadsheets you may have noticed.

You can change resource capacities, the types of queues, etc.

Resource Spreadsheet



Click on the spreadsheet in the Basic Process template, and you get a list of the resources that the model is using (at the bottom of the screen).

- Example: Two resources, Barber and Pedicurist. The resource Barber has fixed capacity 4 (four barbers on duty); there's only 1 Pedicurist.

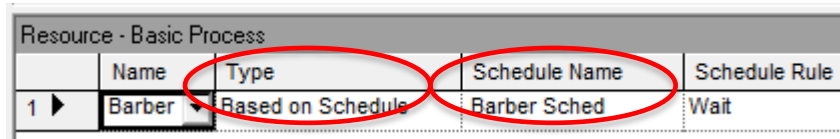
Resource - Basic Process								
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures
1 ▶	Barber	Fixed Capacity	4	0.0	0.0	0.0		0 rows
2	Pedicurist	Fixed Capacity	1	0.0	0.0	0.0		0 rows

- Reminder: Capacity is the number of the resource's servers on duty, not necessarily the number of the servers that a customer requests.
- Note: A resource's servers are regarded as **identical and interchangeable**.
- The resources are automatically sent to the spreadsheet when we define them in the Process module. Or, we can just double-click a new row and define a resource right there in the spreadsheet.
- Can change **fixed capacity**, or make a **schedule** that varies the capacity over time.

Schedule Spreadsheet

Can set schedules for [servers](#).

- Example: Go to the Resource Spreadsheet and change the Type to “Based on Schedule”. The dialog box will change slightly to accommodate a schedule name (which we’ve called “Barber Sched”).



Resource - Basic Process				
	Name	Type	Schedule Name	Schedule Rule
1 ▶	Barber	Based on Schedule	Barber Sched	Wait

- Now go over to the Schedule Spreadsheet and you’ll see that Barber Sched is already there, waiting for your input.



Schedule - Basic Process					
	Name	Type	Time Units	Scale Factor	Durations
1 ▶	Barber Sched	Capacity	Hours	1.0	0 rows

- The key is to enter stuff into the “Durations” box.

Schedule Spreadsheet

Can set schedules for **arrivals**.

- Example: Go to a Create module and change the Type to “Schedule”. The dialog box will change slightly to accommodate a schedule name (which we’ve called “Arrival Sched”).
- Now go over to the Schedule Spreadsheet and you’ll see that Arrival Sched is already there, waiting for your input.
- We’ll see in the demo how to enter stuff into the “Durations” box.

Create

Name: Create 1 Entity Type: Entity 1

Time Between Arrivals




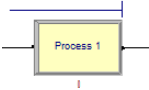
Type: Schedule Schedule Name: Arrival Sched

Entities per Arrival: 1 Max Arrivals: Infinite

OK Cancel Help

Schedule - Basic Process					
	Name	Type	Time Units	Scale Factor	Durations
1	Barber Sched	Capacity	Hours	1.0	0 rows
2	Arrival Sched	Arrival	Hours	1.0	0 rows

Demo Time!

- Resource Spreadsheet
- Resource and Arrival Schedules
- Queue Spreadsheet 

- Resource Animation 
- Queue Animations 

Summary

This Time: Discussed the Resource, Schedule, and Queue Spreadsheets, along with some of their ramifications.

Next Time: We'll discuss the Decide module, which allows us to make probabilistic and conditional decisions that affect the entities' paths through the model.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

The Decide Module

Lesson Overview

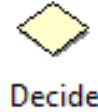
Last Lesson: The Resource,
Schedule, Queue Spreadsheets

This Lesson: The Decide module,
which allows customers to make
probabilistic and conditional
choices about their paths.

Idea: A flexible module enabling
you to go this way and that way!

www.youtube.com/watch?v=z4uWfs3sH7s

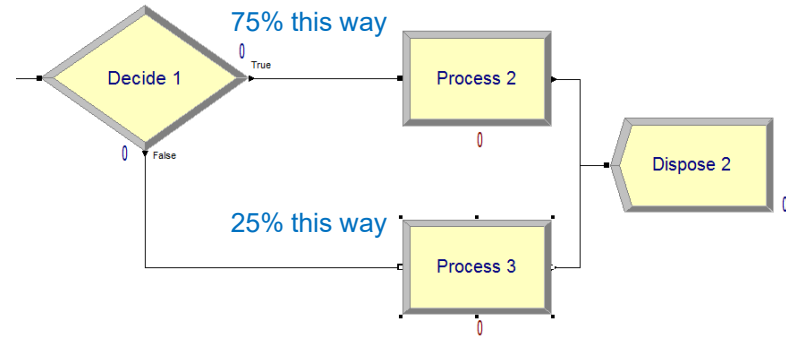
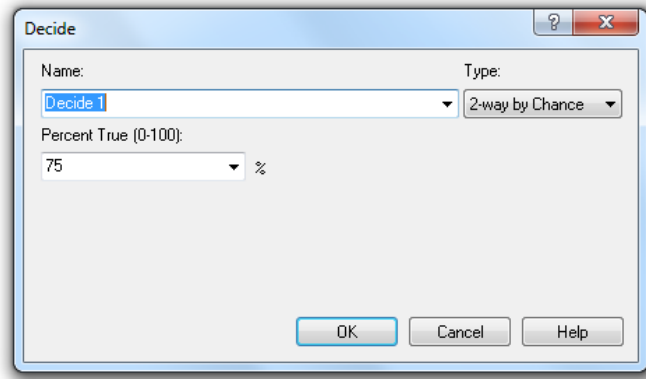
Decide Module



- When an entity gets to a Decide module, he can...
 - Randomly go to either of two locations (“2-way by Chance”) (probabilities are given in percentages)
 - Randomly go to any of various locations (“N-way by Chance”)
 - Go to either of two locations if a condition is satisfied (“2-way by Condition”)
 - Go to any of various locations if a condition is satisfied (“N-way by Condition”)

- # Decide Module

- Example: Go to Process 2 with prob 0.75, and Process 3 w.p. 0.25.



- Demo Time!

Summary

This Time: The Decide module, which gives customers lots of choices about moving through the model.

Next Time: The Assign module, which allows us to give values to attributes and variables, and even assign graphics to entities.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

The Assign Module

Lesson Overview

Last Lesson: The Decide module, which makes probabilistic and conditional decisions about where customers go.

This Time: The Assign module, which allows us to give values to attributes and variables, and even assign graphics to entities.

Idea: Very power, general tool.

The Next Few Lessons

Cover material that'll eventually allow us to simulate multi-channel customer flows:

- The Assign Module ← now
- Attribute, Variable, Entity Spreadsheets
- Arena Internal Variables
- Displaying Stuff
- Batch, Separate, Record
- Run Setup and Control
- Two-Channel Mfg Example

Attributes

Each customer passing through the system has various properties (**attributes**).

- Tom is 6' tall, weighs 160 lbs, loves baseball, and has LDL cholesterol = 108.
- Justin B. is 4'11" tall, weighs 280 lbs, loves eating lard, and has LDL = 543.
- Both guys have 4 attributes, though other guys may have different numbers of attributes.
- Attributes need to be **numerical**; for example, BB = 11 and lard = 28.

Variables

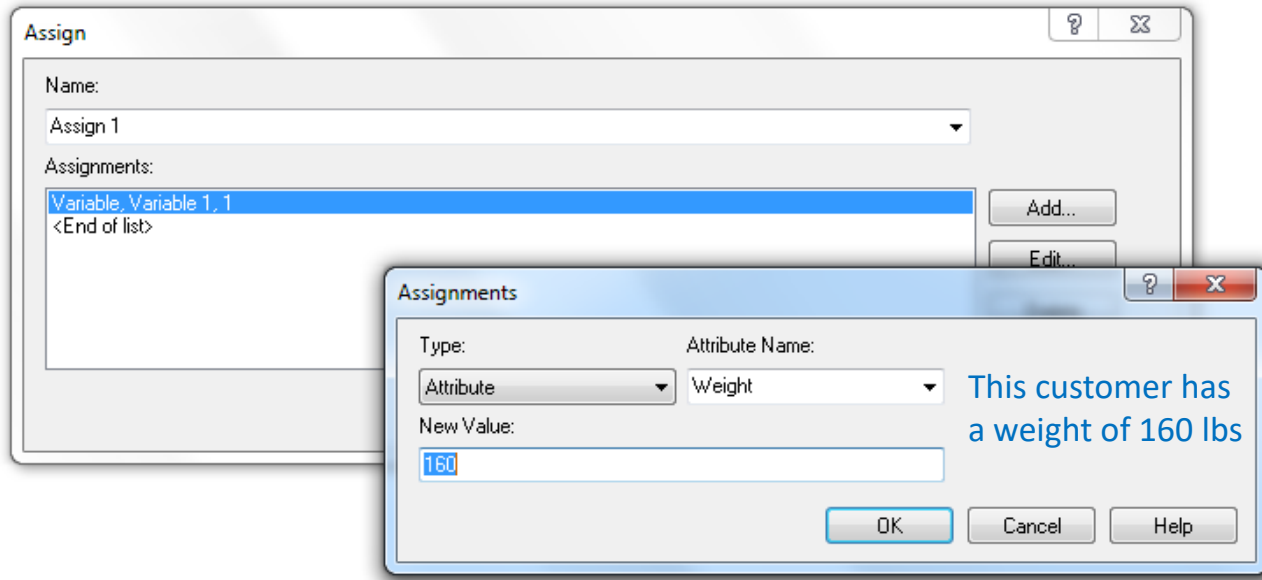
- Unlike attributes, whose values are specific to each customer, **variables are global**.
- If you change a variable anyplace in the Arena program, it gets changed everywhere.
- Example: WIP might be incremented when an entity is Create'd, and decremented if it's Dispose'd – these events could occur anywhere.

- Assign Module



Assign

- So how can you change attributes, variables, and other stuff?
- Use the very powerful and flexible **Assign** module.



Demo Time!

- Use of Assign for attributes and variables
- Use of Assign for entity pictures
- Using Decide and Assign together

Summary

This Time: Talked about Attributes, Variables, and then the Assign module.

Next Time: The Attribute, Variable, and Entity Spreadsheets.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Attribute, Variable, and Entity
Spreadsheets

Lesson Overview

Last Lesson: Discussed the Assign module, and how it can be used to (i) give values to attributes and variables, and (ii) change entity graphics.

This Lesson: The Attribute, Variable, and Entity spreadsheets.

• Spreadsheets



Attribute



Variable



Entity

- The **Attribute Spreadsheet** keeps track of existing **attributes** that you might define in an Assign.
- Can make your own (double click).

	Name	Rows	Columns	Data Type	Initial Values
1 ▶	height	3		Real	3 rows
Double-click here to add a new row.					

- Can be vectorized.

Initial Values	
1	1.3
2	1.2
3	1.8

- **Variable Spreadsheet** pretty much the same.
- **Entity Spreadsheet** allows you to set initial picture.
- **Demo Time!**

Entity Type	Initial Picture
Entity 1 ▼	Picture.Report

Summary

This Time: Easy discussion about the Attribute, Variable, and Entity Spreadsheets.

Next Time: We'll discuss a few Arena "internal" variables that are automatically calculated as the simulation proceeds. These have a great variety of uses.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Arena Internal Variables

Lesson Overview

Last Lesson: The Attribute, Variable, and Entity Spreadsheets.

This Lesson: Talk about Arena's so-called "internal" variables that are automatically calculated as the simulation proceeds.

Idea: Incredibly useful. Help you make decisions based on the simulation's current state.

• Internal Variables

- Dirty Little Secret: Arena keeps track of and continuously updates lots of stuff as the simulation runs.
- Good for making decisions, drawing graphs, etc.
- Examples:
 - `TNOW` = current simulated time
 - `NR(Barber)` = # of resource Barber's servers now working
 - `NQ(Process 1.Queue)` = # customers in that queue
 - `Create 1.NumberOut` = # of customers who have so far left the module named Create 1
- Huge list is available if you sniff around Build Expression.
- **Demo Time!** Go to the shortest line.

Summary

This Time: Introduced internal variables such as TNOW, NR(), etc. They're constantly updated and provide lots of relevant info.

Next Time: How to display variables, graphs, and results.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Displaying Variables, Graphs,
and Results

Lesson Overview

Last Lesson: Talked about some useful internal variables that Arena constantly updates.

This Lesson: How can we display the values of certain variables in real time, construct graphs, and produce output result files?

• Getting Information

- How to display information as the simulation is running?


- Arena provides lots of capability.

- For example, consider the toolbar...



- You can get an analog or digital clock. 

- Calendar 

- Variable displays, which keep track of the values of variables in real time. 

- Histograms 

- Graphs 

- When the simulation is over, it will automatically generate an **output report** giving info on server usage, queue length, customer waits and cycle times, and other user-defined quantities.

- **Demo time**, including drill press example!

Summary

This Time: Discussed ways to display the values of certain variables in real time, construct graphs, and produce output files.

Next Time: A quick tutorial on the Batch, Separate, and Record modules.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Displaying Variables, Graphs,
and Results

Lesson Overview

Last Lesson: Talked about some useful internal variables that Arena constantly updates.

This Lesson: How can we display the values of certain variables in real time, construct graphs, and produce output result files?

• Getting Information

- How to display information as the simulation is running?


- Arena provides lots of capability.

- For example, consider the toolbar...




- You can get an analog or digital clock. 

- Calendar 

- Variable displays, which keep track of the values of variables in real time. 

- Histograms 

- Graphs 

- When the simulation is over, it will automatically generate an **output report** giving info on server usage, queue length, customer waits and cycle times, and other user-defined quantities.

- **Demo time**, including drill press example!

Summary

This Time: Discussed ways to display the values of certain variables in real time, construct graphs, and produce output files.

Next Time: A quick tutorial on the Batch, Separate, and Record modules.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Batch, Separate, and Record
Modules

Lesson Overview

Last Lesson: Looked at a bunch of ways to obtain information during and after the run – e.g., histograms, graphs, output reports.

This Lesson: Discuss the Batch, Separate, and Record modules.

These are the last modules for the Basic Process template... We're making good progress!

• Batch Module



Batch

- Combine (“batch”) multiple customers into one “super”-customer.
- Batch Size = 3 accumulates 3 guys before sending off the super-customer.
- If info about the individual customers in a batch won’t be needed later on, then choose Type = Permanent.
- But if you want eventually to reconstitute the original members, set Type = Temporary.
- Temporary batches will need to be **split** before being Dispose’d (keep watching).

The screenshot shows a 'Batch' dialog box with the following fields and options:

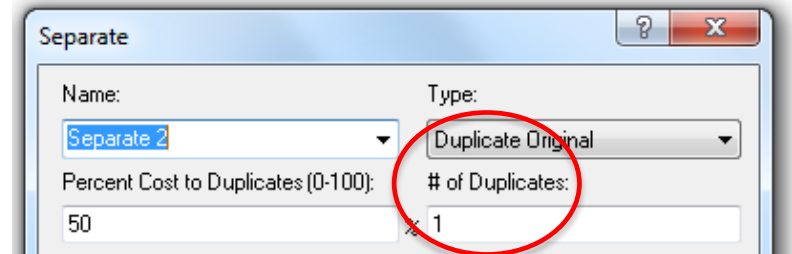
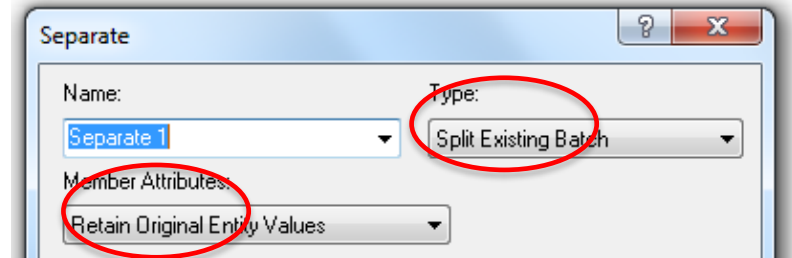
- Name:** Batch 1
- Type:** Permanent (circled in red)
- Batch Size:** 3 (circled in red)
- Save Criterion:** Last
- Rule:** Any Entity
- Representative Entity Type:** (empty dropdown)
- Buttons:** OK, Cancel, Help

Separate Module



Separate

- Duplicate a single entity, or split multiple entities that had been combined in a Batch module.
- If dealing with a **Permanent** batch, usually use Duplicate Original to get several customers all with the same attributes.
- If dealing with a **Temporary** batch, use Split Existing Batch to reproduce customers with original attributes.

A screenshot of the 'Separate' module dialog box. The 'Name' field is set to 'Separate 2'. The 'Type' dropdown is set to 'Duplicate Original'. The 'Percent Cost to Duplicates (0-100)' is set to 50. The '# of Duplicates' is set to 1. Red circles highlight the 'Type' dropdown and the '# of Duplicates' field.A screenshot of the 'Separate' module dialog box. The 'Name' field is set to 'Separate 1'. The 'Type' dropdown is set to 'Split Existing Batch'. The 'Member Attributes' dropdown is set to 'Retain Original Entity Values'. Red circles highlight the 'Type' dropdown and the 'Member Attributes' dropdown.

Record Module



Record

- Collect statistics when an entity passes through the module.
- We'll talk more about this module as we encounter it in future examples.
- **Demo Time!** Now we'll look at a couple of permutations of the Batch and Separate modules.

Summary

This Time: Finished off the Basic Resources template with the Batch, Separate, and Record modules.

Next Time: We'll look at a bunch of Run Setup and other control functionalities.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Run Setup and Control

Lesson Overview

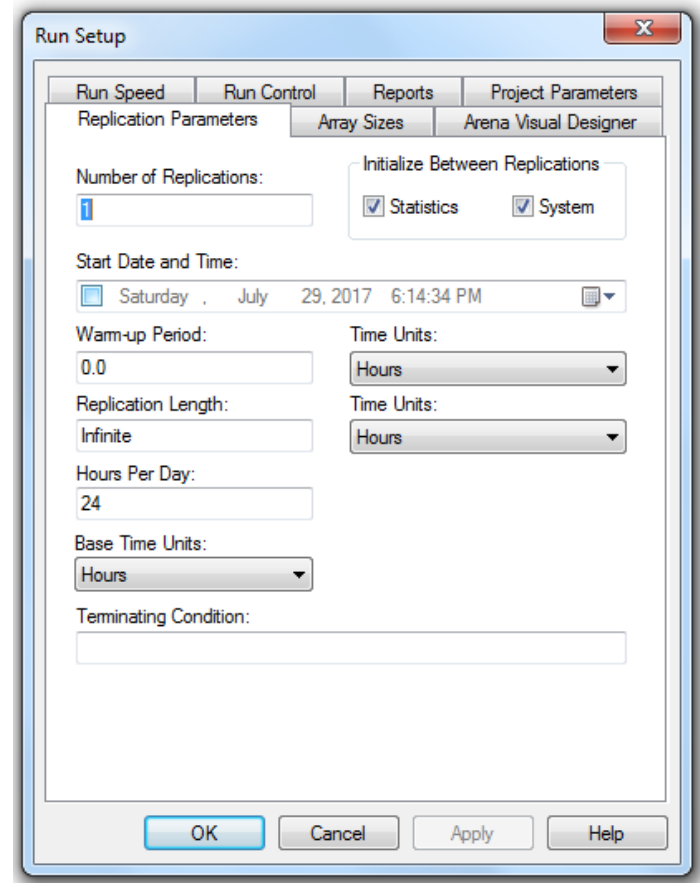
Last Lesson: Finished off the Basic Resources template with the Batch, Separate, and Record modules.

This Lesson: We'll be looking at a bunch of easy little trivia points to help you get your simulation runs going efficiently.

• Run Setup

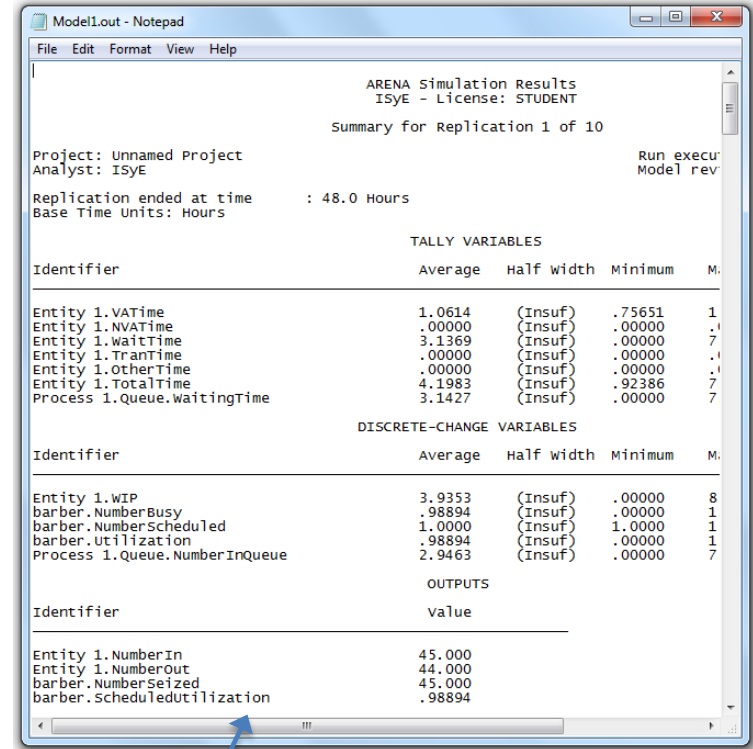
Run > Setup gives you **lots** of stuff.

- Replication Parameters tab
 - Number of Reps: # of indep runs.
 - Initialize Between Reps: Do you start the stats collection and/or system state from scratch for each replication?
 - Warm-up Period: How long to run before you start keeping data.
 - Rep Length: How long each run is.
 - Terminating Condition: Any special ways to stop the simulation (besides rep length condition from this tab or max arrivals from Create module)?



• Run Setup (cont'd)

- Run Speed tab: Often need to speed up or slow down the simulation by more than what you can do on the main screen.
- Reports: Arena has a variety of reports that it gives upon completion of a run.
 - Contains info on customer waits, lengths of queues, server utilizations, user-defined variables, etc.
 - Category Overview summarizes all reps.
 - Category by Replication gives tedious info about each rep.
 - SIMAN Report gives concise text file.



ModelOut - Notepad

File Edit Format View Help

ARENA Simulation Results
ISyE - License: STUDENT
Summary for Replication 1 of 10

Project: Unnamed Project
Analyst: ISyE
Run execution Model revision

Replication ended at time : 48.0 Hours
Base Time Units: Hours

TALLY VARIABLES

Identifier	Average	Half width	Minimum	Maximum
Entity 1.VATime	1.0614	(Insuf)	.75651	1.0614
Entity 1.NVATime	.00000	(Insuf)	.00000	.00000
Entity 1.WaitTime	3.1369	(Insuf)	.00000	7.00000
Entity 1.TransTime	.00000	(Insuf)	.00000	.00000
Entity 1.OtherTime	.00000	(Insuf)	.00000	.00000
Entity 1.TotalTime	4.1983	(Insuf)	.92386	7.00000
Process 1.Queue.WaitingTime	3.1427	(Insuf)	.00000	7.00000

DISCRETE-CHANGE VARIABLES

Identifier	Average	Half width	Minimum	Maximum
Entity 1.WIP	3.9353	(Insuf)	.00000	8.00000
barber.NumberBusy	.98894	(Insuf)	.00000	1.00000
barber.NumberScheduled	1.0000	(Insuf)	1.0000	1.0000
barber.Utilization	.98894	(Insuf)	.00000	1.00000
Process 1.Queue.NumberInQueue	2.9463	(Insuf)	.00000	7.00000

OUTPUTS

Identifier	Value
Entity 1.NumberIn	45.000
Entity 1.NumberOut	44.000
barber.NumberSeized	45.000
barber.ScheduledUtilization	.98894

SIMAN report

Run Control

Run > Setup > Run Control gives a variety of ways that you can run the simulation.

Example: Batch mode, which turns off all graphics and result in extremely fast runs.

Demo Time! Let's...

- Look at some reports.
- Stop the simulation the first time the queue size hits 4.

Summary

This Time: Learned how to do various setup and run control tasks. For instance, we heard about independent reps, run termination from the setup menu, and output reports.

Next Time: A small case study for a two-channel manufacturing system, where we put a lot of the previous material together.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Simple Two-Channel
Manufacturing Example

Lesson Overview

Last Lesson: Discussed various various run setup and control tasks. E.g., how to implement indep reps, run speed, reports.

This Lesson: Demo a two-channel manufacturing system, where we finally put a lot of the previous material together.

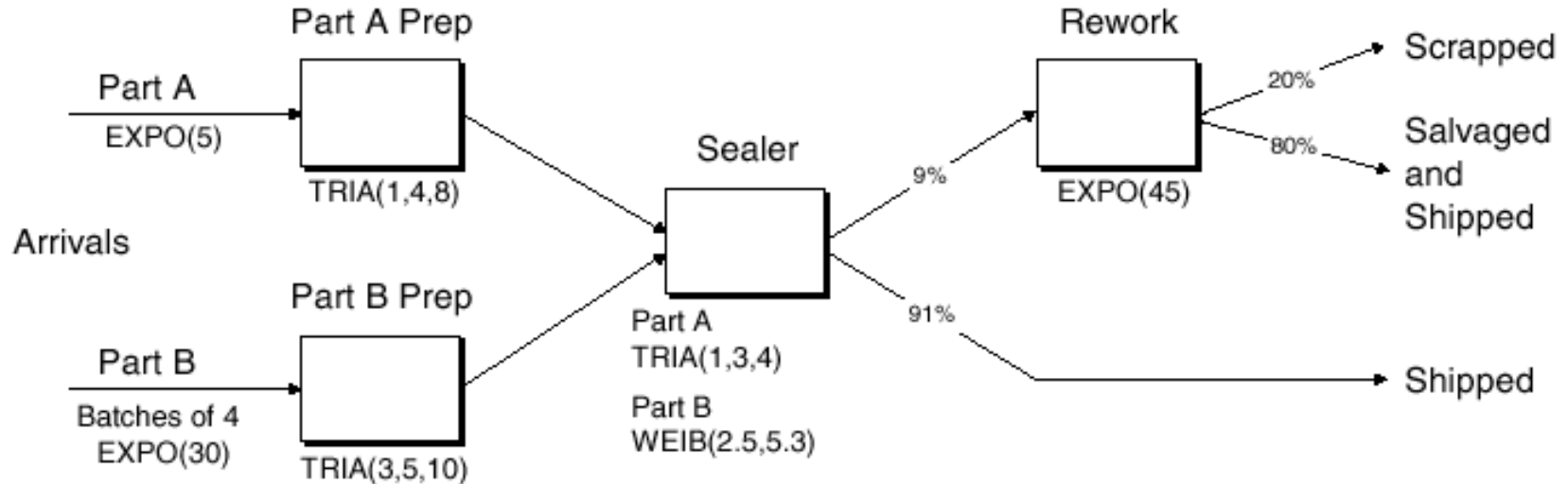
Watch how we use attributes!

• The Story

- This is Model 4-1 “An Electronic Assembly and Test System” from the KSZ (2015) text, *Simulation with Arena*.
- Two different arrival streams
 - Type A parts show up 1-at-a-time; Type B’s 4-at-a-time.
 - Type A’s show up a little more often than Type B’s.
- A’s feed into a Prep A server; B’s go to Prep B. Different service times.
- Then the parts get processed by the **same** Sealer server, but again with **different service time distributions**.
- All parts undergo an inspection. If they pass, they exit.
- If they don’t pass, they go to a Rework server, and then another inspection. Whether or not they pass, they exit the system.

• The Story (cont'd)

I stole this flowchart from KSZ...



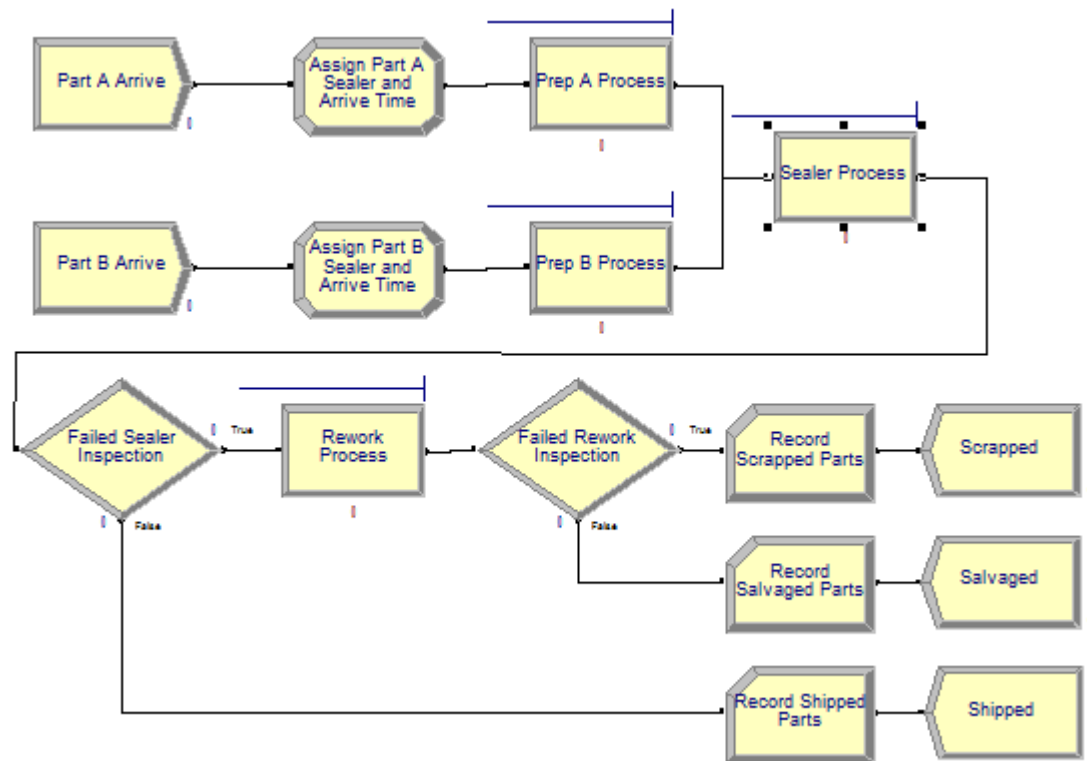
- The Story (cont'd)

- How to handle different A and B service times at Sealer?
 - Trick 1: **Pre-assign** the service times as an **attribute** (Sealer Time) in an Assign module immediately after each customer arrives. Use that attribute regardless of being a Type A or B part.
 - Trick 2: While we're at it, use the Assign to store each customer's arrival time as an **attribute**. Use the Arena variable TNOW to do so.
- Record departure times just before parts get Dispose'd, This will allow us to get average cycle times (depart – arrival times) for any of the 3 types of parts (pass on first try, pass on second, fail both).

• Let's Get Logical

- Dave's Alternative Trick: Is there another way to model the process time at the Sealer without having to assign a Sealer Time attribute for A's and B's?
- Yes! It involves a little work, but it's nice to know.
 - Note that the entity types (Part A and Part B) are assigned in the respective Create modules.
 - Instead of assigning the Sealer Time as an attribute in the Assign module, we can just wait until the Sealer Process module and use the following **logical** expression, where $(x=y) = 1$ if $x=y$, and 0 otherwise. (See why?)
 $((\text{Entity.Type} == \text{Part A}) * \text{TRIA}(1,3,4)) + ((\text{Entity.Type} == \text{Part B}) * \text{WEIB}(2.5,5.3))$

- Demo Time!



Summary

This Time: Our first “real” system – a small manufacturing system.

Learned a couple of nice tricks involving attributes and logic.

Next Time: Fake Customers!



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Fake Customers

Lesson Overview

Last Lesson: Demo'd a small electronic assembly and test system.

This Lesson: We'll talk about “fake” customers!

Idea: These aren't real customers at all... their purpose is just to do some task that Arena needs.

The Next Few Lessons

We'll now cover material that will lead us to the simulation of a call center:

- Fake Customers ← now
- The Advanced Process Template
- Resource Failures + Maintenance
- The Blocks Template
- The Joy of Sets
- Description of Call Center
- Call Center Demo

Fake Customers

- You can use “fake” customers to accomplish various tasks during a simulation.
- Not actual customers that you care about in terms of waiting times or use of resources.
- **Demo Time** will explain all!
 - Calculate normal probabilities
 - Keep track of which time period you’re in (part of Call Center example coming up)
 - Breakdown demon

Summary

This Time: Talked about fake customers. We'll be using these several times in upcoming applications.

Next Time: Now that we're experts in the Basic Process template, we'll move on to the Advanced Process template for more good stuff!



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

The Advanced Process
Template

Lesson Overview

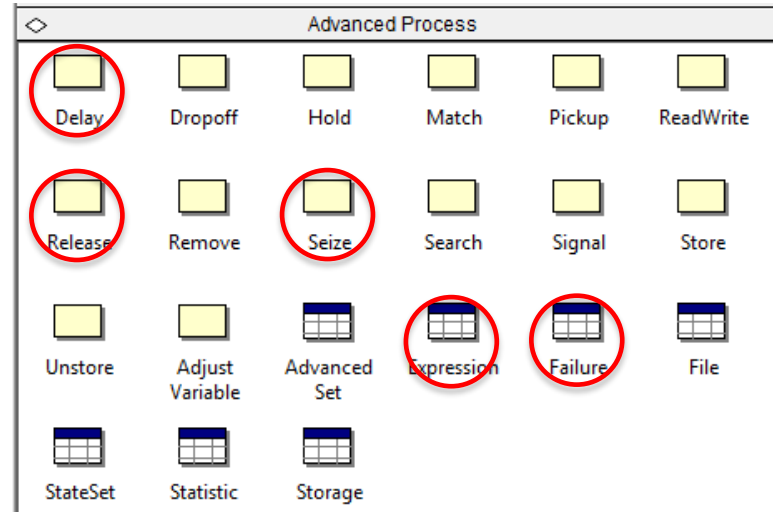
Last Lesson: Introduced “fake” customers, which can be used for a variety of purposes, including arrivals of breakdown demons.

This Lesson: We'll add to our arsenal of modules by introducing the Advanced Process template!

Idea: Lots of great new stuff, though I don't expect you to memorize everything.

• Advanced Process Template

- File > Template Panel > Attach (and hopefully you find it in the Rockwell Software \ Arena \ Template directory).
- Lots of new modules and spreadsheets.
- I don't expect you to memorize them all.
- Concentrate for now on a few of these (comment on others later):
 - Seize, Delay, Release modules
 - Expression, Failure spreadsheets.



• Seize, Delay, Release

- Why do we have separate Seize, Delay, and Release modules? Aren't they right there in the Process module?
- Yup, but sometimes you need to do things that are too complicated for the Process module version of Seize–Delay–Release.
- Now you can handle things like...
 - Seize–Assign–Delay–Release
 - Non-symmetric multiple Seize's and Release's
 - Complicated Seize's and Release's that might depend on sets of servers (as in the upcoming Call Center example)
- Demo Time!
 - Will talk about some of these Seize, Delay, Release issues.
 - Will also sneak in some wisdom about the Expression spreadsheet.

Summary

This Time: Learned about the Advanced Process template, which gives us a new bag of tricks to play with!

Next Time: How to deal with resource failures and maintenance.



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Resource Failures and
Maintenance

Lesson Overview

Last Lesson: Learned about the Advanced Process treasure chest.

This Lesson: We'll study model breakdowns and maintenance in a way that's more direct than via fake customers.

Idea: It's just a certain type of schedule!

• Failures

- You can cause resource failures by scheduling “breakdown demons” (fake customers with high priority). But this isn’t elegant.
- Better way is to use the Resource and Failure spreadsheets in conjunction with each other.

Resource - Basic Process						
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use
1 ▶	Drill Press ▼	Fixed Capacity	1	0.0	0.0	0.0

Failures

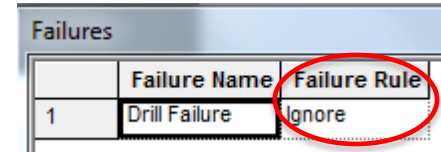
	Failure Name	Failure Rule
1	Drill Failure	Ignore

StateSet Name	Failures
	1 rows

Failure - Advanced Process					
	Name	Type	Count	Down Time	Down Time Units
1 ▶	Drill Failure	Count	10	Expo(30)	Minutes

• Failures (cont'd)

- Go to Resource spreadsheet in the Basic Process template.
- Click on Failures column.
- Add a Failure Name.
- Choose Failure Rule (will discuss in a minute).
- Go to the Failure spreadsheet in the Advanced Process template, where you'll see your new failure name.
- Choose type of failure:
 - Count (failure after a certain # of arrivals)
 - Time (after a certain amount of time)
- Choose downtime (for repair): Can be any expression.



	Failure Name	Failure Rule
1	Drill Failure	Ignore

• Remarks

- Can schedule multiple failures by using multiple rows of the Failures column in the Resource spreadsheet, e.g., type I failure, type II failure, scheduled maintenance.
- Types of Failure Rules:
 - Ignore (complete service of current customer, but **reduce** repair time). E.g., if repair time = 1 hour and cust still needs 10 min, then repair time gets reduced to 50 min and finishes at 60 min mark.
 - Wait (complete service of current cust and **delay** repair). E.g., if repair time = 1 hr and cust needs 10 min, then repair finishes at 70 min mark.
 - Preempt (stop service of current cust, but **complete service after the repair**). Repair stops at 60 min mark, cust finishes at 70 min mark.
- **Demo Time!**

Summary

This Time: How ironic! We
succeeded in learning how to
implement failures in Arena!

Next Time: A quick intro to the Blocks
template, which gives us a
krazy number of new things to
look at... but we'll be very gentle
about this.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

The Blocks Template

Lesson Overview

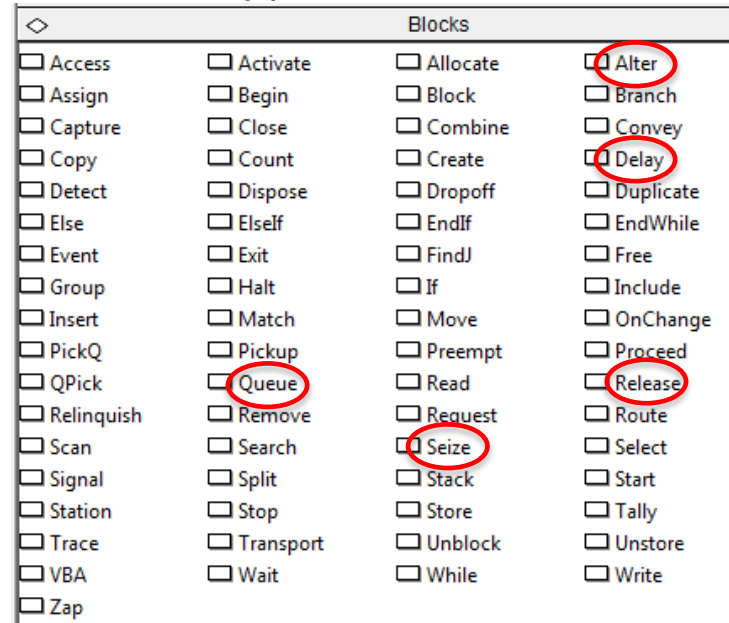
Last Lesson: Discussed the implementation of resource failures.

This Lesson: We'll add to our bag of tricks by introducing the Blocks template.

These are lots or blocks in this template, but don't be intimidated.

• Blocks Template

- File > Template Panel > Attach (and hopefully you find it in the Rockwell Software \ Arena \ Template directory).
- Huge number of blocks!
- Do not be scared!
- They're sort of a self-contained language (related to SIMAN).
- Very specialized and low-level.
- Very useful, but we'll only need a few for now:
 - Seize, Delay, Release, Queue, Alter



- Eh??

- Yikes! Why are we seeing stuff like Seize, Delay, Release **yet again!?**
At least two reasons...
 - Arena has been built in many layers over the years, and these are what's left from SIMAN – the “original” incarnation of Arena.
 - But in any case, and for whatever reason, certain primitive blocks such as the Queue block can't even connect to a Seize module from the Advanced Process template or a Process module from the Basic Process template. So you're stuck!
- **Demo Time!**
 - A Queue-Seize-Delay-Release example using primitive blocks (we'll see this again in the Call Center example coming up).
 - Using the Alter block to change the number of resources.

Summary

This Time: The Blocks template!
Later on, we'll find a number of interesting uses for some of these guys.

Next Time: The joy of sets! How to use certain vectorized sets to make modeling easier.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

The Joy of Sets

Lesson Overview

Last Lesson: Talked about the Blocks template, which we'll occasionally use to help with specialized little issues.

This Lesson: What are sets, and how do they enhance our modeling ability?

Idea: We'll especially concentrate on **resource sets**.

Sets

- As in baby math class, a set is just a group of elements. Elements are allowed to belong to more than 1 set.
- “Elements” can mean a lot of things in Arena, and so there are various types of sets:
 - Resource (e.g., a set of servers)
 - Counter
 - Tally
 - Entity Type
 - Entity Picture
 - ...and tougher stuff later

• Resource Sets

- Use the Set spreadsheet in the Basic Process template to define sets.
- We'll just study resource sets for now.
- A “vanilla” **resource** has identical, interchangeable servers. But a **resource set** can have **distinct** servers, with different schedules, service speeds, service specialties, etc.
- Call Center Example: Three products with the following resources
 - Product 1: Charity, Noah, Molly, Anna, Sammy
 - Product 2: Tierney, Sean, Emma, Anna, Sammy
 - Product 3: Shelley, Jenny, Christie, Molly, Anna, Sammy
 - Molly, Anna, and Sammy all have degree of cross-functionality.
 - All 11 servers turn out to have different schedules (more on that later)

• Resource Sets

- To define the resource set Product 1, choose Type = Resource, click in Members, then enter Product 1's servers under Resource Name.

Set - Basic Process

	Name	Type	Members
1 ▶	Product 1	Resource	5 rows
2	Product 2	Resource	5 rows
3	Product 3	Resource	6 rows
4	Tech Calls	Tally	3 rows
5	Return Time	Tally	3 rows
6	Busy Lines	Counter	22 rows

Double-click here to add a new row.

Members

	Resource Name
1	Charity
2	Noah
3	Molly
4	Anna
5	Sammy

Double-click here to add a new row.

- Product 1's "Preferred Order" is to try Charity first. If she's not available, try Noah. Save Sammy for last because he's cross-functional. Other orders are possible.

• Seize-Delay-Release

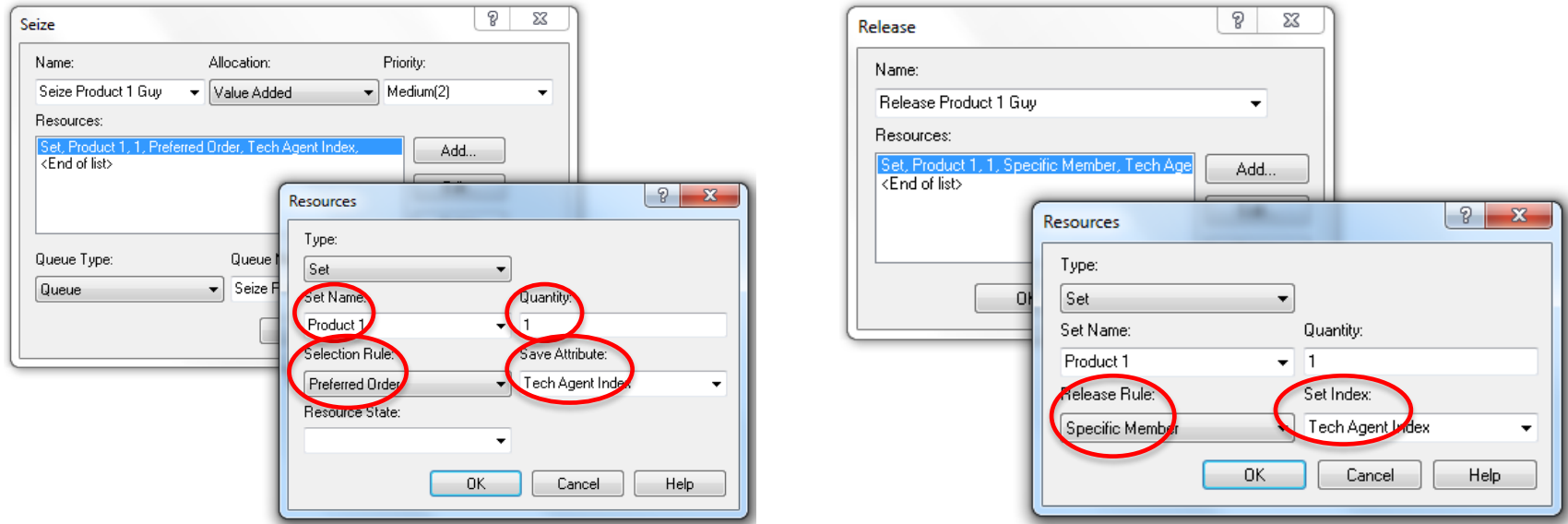


- Have to be a little careful with Seize-Delay-Release for a resource set.
- The problem is that you have to make sure that you Release the **same guy** that you originally Seize'd. (If you release a random server, some other customer may lose his server!)



- Let's Seize a server from the set Product 1 in Preferred Order. Make sure we remember who it is. Then when we're done, Release that same guy.

• Seize-Delay-Release (cont'd)



- Seize Quantity = 1 guy from the set Product 1 in the Preferred Order. Save the “name” of that server in the customer’s Save Attribute called Tech Agent Index.
- Later, Release that Specific Member stored in Tech Agent Index.

• Remarks

- Various Seize Selection Rules are possible:
 - Cyclical
 - Random
 - Preferred Order
 - Specific Member
 - Largest Remaining Capacity
 - Smallest Number Busy
- This stuff will be very useful when we do our Call Center example.
- Demo Time! A simple model with 4 servers.

Summary

This Time: How joyful! We introduced sets, especially resource sets. Then we showed how to use Seize-Delay-Release with such sets.

Next Time: We'll describe in Plain English the Call Center example that seems to be all the rage these days.



Computer Simulation

Module 5: Arena


Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Description of Call Center
Example

Lesson Overview

Last Lesson: Mainly discussed resource sets, and how to use Seize-Delay-Release with them. Have to be a little careful. 

This Lesson: What we've all been waiting for! Let's start describing the Call Center example.

When we're done with this, you can be consultants!

Call Center Description

Program is arranged in [submodels](#).



- Create and Direct Arrivals
 - How often do calls show up and where do they go?
- Tech Support Calls
 - What kind of tech support do you need?
- Returned Tech Calls
 - Sometimes the guy has to get back to you.
- Sales Calls
- Order Status Calls
- Time Period Counter – What ½-hour period of the day is it?

Arrivals

Calls show up according to a **nonhomogeneous Poisson process**.

- The center accepts calls from 8:00A till 6:00P.
 - The arrival rates change every half hour (see table below from KSZ).
 - Also see **Arrival Schedule** in the Basic Process Schedule spreadsheet.

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



← These rates are **per hour**
(even though time intervals
are 30 min)

- A few staff actually stay at work until 7:00P, in order to let the calls at the end of the day clear out. (You have to explicitly model 0 arrivals for the last two half-hour segments of the Arrival Schedule.)

Phone Lines

There are 26 phone lines. If you get a busy signal, you balk and get killed off.

- Use a Queue block with capacity 0 to try to Seize a line. If the Seize fails, there's no place to go in the Queue, and out you go.
- The Queue and Seize both come from the [Blocks](#) template panel. This is the only place where you can get a Queue block, and it only connects to this kind of Seize. (Recall that you can also use a Seize within a Process module in the Basic Process template, and there's another Seize in the Advanced Process template!)

Types of Calls

There are 3 general types of calls, which we'll handle with an "N-way by Chance" Decide module. Each customer hears a recording for a $UNIF(0.1, 0.6)$ amount of time, while he makes his choice of call type.

- 76% go to Tech Support.
- 16% go to Sales.
 - There are 7 identical, faceless Sales guys that we'll need to schedule (but not on an individual basis).
 - Sales calls take $TRIA(4, 15, 45)$.
- 8% go to Order Status.
 - Most callers can do self-service in a $TRIA(2, 3, 4)$ amount of time. But...
 - 15% of customers need a Sales guy for $TRIA(3, 5, 10)$.

Tech Support

There are 3 specific types of Tech Support calls (which we'll again handle via a Decide module):

- 25% are for Product 1.
- 34% are for Product 2.
- 41% are for Product 3.
- Each customer gets another recording for a $\text{UNIF}(0.1, 0.5)$ amount of time, while he makes his choice.
- All Tech Support calls require staff for a $\text{TRIA}(3, 6, 18)$ duration of time.

Call Backs

4% of Tech Support calls require additional investigation.

- This additional investigation is carried out by another group of staff that we won't worry about. The customer's original server is freed up when he makes the determination that more research is needed.
- The investigation takes an EXPO(60) amount of time.
- At that point, the original Tech Support guy will call the customer back using one of the 26 phone lines (but with high priority).
- That call takes TRIA(2,4,9) time.

Tech Support Info

Tech Support staff have some interesting issues.

- They all have 8 hr days + 30 min for lunch.
- They all have different schedules. We'll need Resource Schedules to model this on an individual basis.
- They all have different product expertise. We'll need Sets to model this. Here are the Preferred Orders for the sets:
 - Product 1: Charity, Noah, Molly, Anna, Sammy
 - Product 2: Tierney, Sean, Emma, Anna, Sammy
 - Product 3: Shelley, Jenny, Christie, Molly, Anna, Sammy

Tech Support Staff Schedules (from KSZ)

[illegible]

Summary

This Time: Gave a long, tedious verbal description of the Call Center model.

Next Time: Gigantic Demo Time!
Now we get to see the Arena program in its full glory!



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Call Center

Lesson Overview

Last Lesson: Plain English description of the Call Center model, along with some Arena details (but no actual demo).

This Lesson: It's Demo Time!

Idea: We're putting together ideas from many of the recent lessons. Everything from fake customers to resource sets.

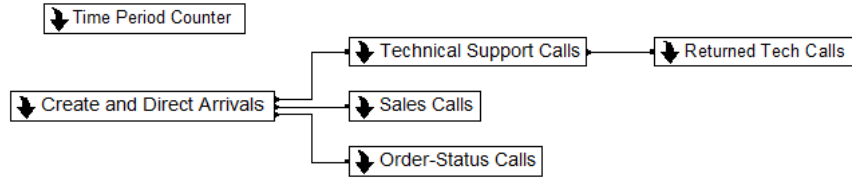
Be On The Lookout

As we go thru the model, keep an eye out for some interesting things...

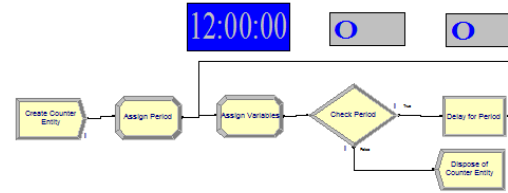
- Submodels
- Fake customers used as timers
- Use Queue with capacity 0 to kick out callers getting busy signal
- NHPP arrivals
- 3 resource sets for Tech Support
- Tricky Seize-Delay-Release
- Quirky callback procedure

Demo Time!

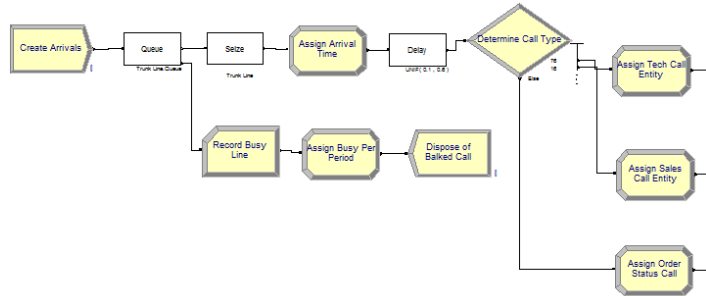
Top-Level View



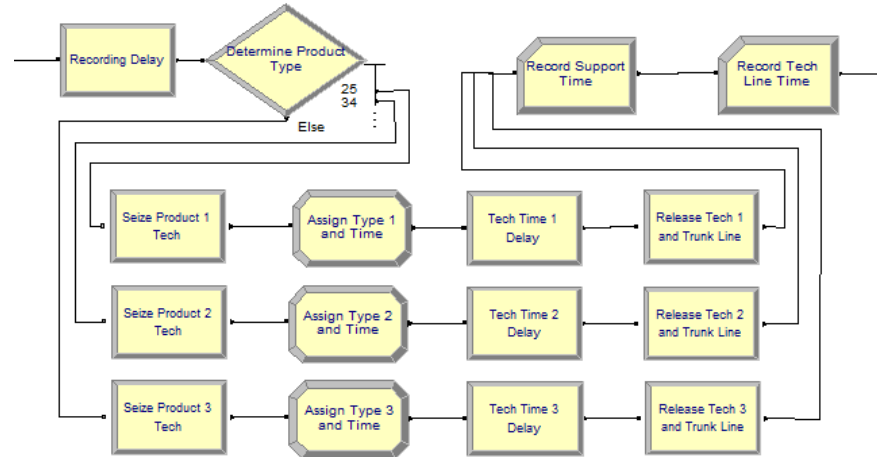
Time Period Counter



Arrivals Submodel



Tech Support Submodel



Summary

This Time: We put together a lot of stuff and finally demo'd the Call Center example in all of its radiant glory.

Next Time: We'll describe an Inventory system in Plain English before doing an analogous demo.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

An Inventory System

Lesson Overview

Last Lesson: Call Center simulation demo.

This Lesson: Plain English description of simple Inventory system and then a demo.

The Next Few Lessons

We'll now look at several demos.

- An Inventory Model ← now
- One Line vs. Two Lines?
- A Re-entrant Queue
- SMARTS files and Rockwell demos
- A Manufacturing System (with transporters, conveyors, ...)

• Description of (s,S) Policy

- Simulate the widget inventory stock over time.
- Use modules from the Basic and Advanced panels.
- Customer interarrivals are $\text{EXPO}(0.1)$.
- Demand size is a DISC function.
- Demand is always “met” (backlogged, if necessary).
- Inventory is “taken” at the start of each day.
- If inventory is below s , we order up to S .
- Delivery lead time is $\text{UNIF}(0.5,1)$ before order arrives (pretty quick, but by the time it arrives, other customers will have arrived).
- Order costs (set-up + incremental), holding costs, penalty costs.

- Description (cont'd)

- Calculate average total cost per day over 120 days.
- Inventory, **unit costs**, and all of the other parameters are **variables**.
- Inventory is decremented by demands, incremented by orders.
- Interarrival times, demands, lead time, are **expressions** from Advanced Process template.
- **Accumulated costs** are calculated in the **Statistic spreadsheet** in the Advanced panel.
- Daily inventory review is conducted by a “fake” customer.

Demo Time!

Summary

This Time: Discussed and demo'd a nice little (s,S) inventory policy.

Next Time: One Line, Two Lines,
Red Line, Blue Line?

<https://www.youtube.com/watch?v=yauPbZKzChs>



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

One Line vs. Two Lines?

Lesson Overview

Last Lesson: We worked on an (s,S) inventory policy example.

This Lesson: Should we use one line feeding into two parallel servers or separate lines feeding into two individual servers?

Idea: We'll use a very cool trick called **common random numbers** to do an apples-to-apples comparison.

• Game Plan

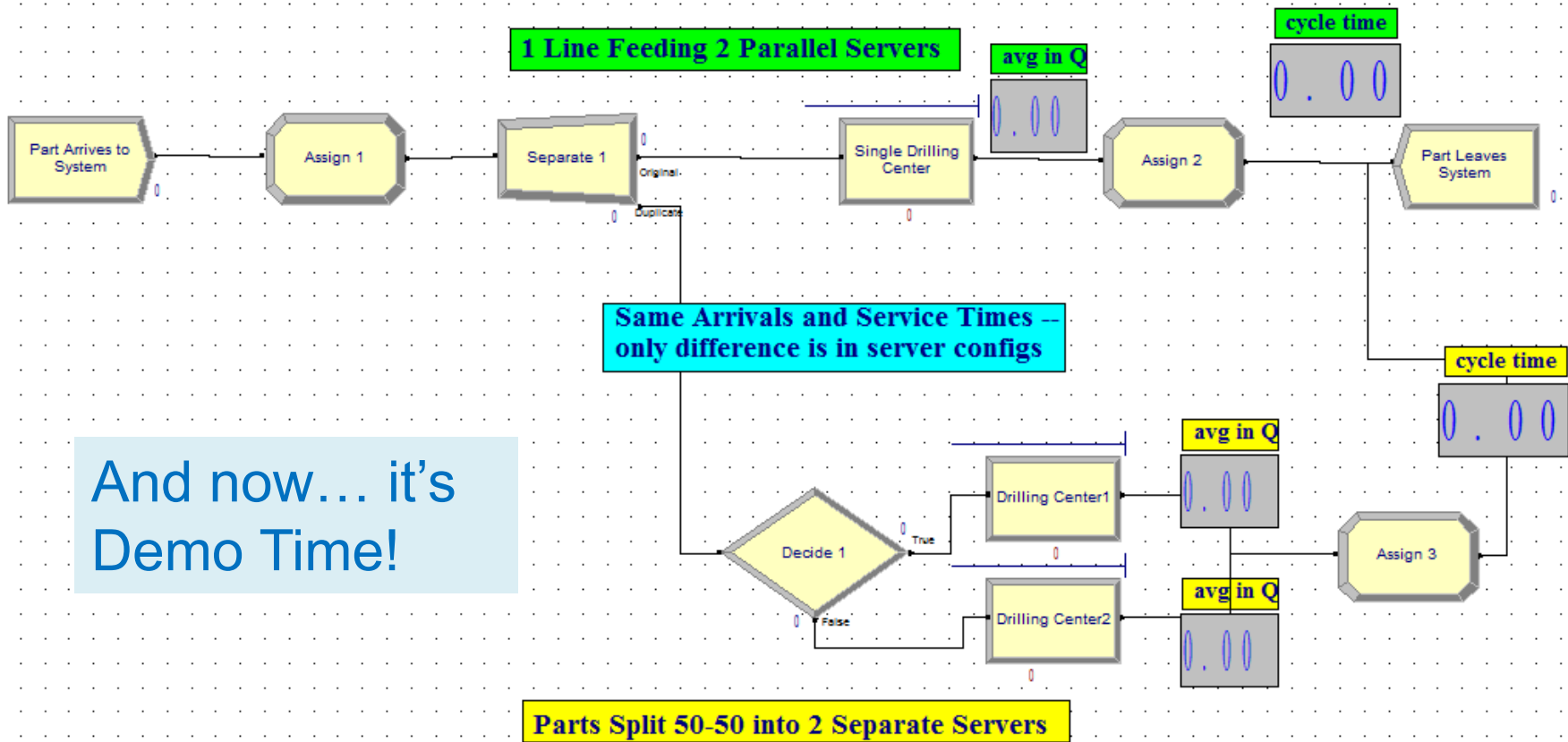
- Option A: Customers show up and join **one line in front of two identical servers**. They go to whichever server is available first.
- Option B: Customers randomly choose which of **two lines (in front of single servers)** to join.
- Will compare which of A or B is better by using:
 - The exact same customer arrivals (thanks to a Separate module), and
 - The same service times for a particular customer whether or not he's doing A or B (thanks to an early Assign module).
- Almost certainly, Option A is better (why?)
- Matching up the arrival and service times makes things easier for us statistically. **“Common Random Numbers”**

1 Line Feeding 2 Parallel Servers

Same Arrivals and Service Times --
only difference is in server configs


And now... it's
Demo Time!

Parts Split 50-50 into 2 Separate Servers



Summary

This Time: Used Arena to compare two different server configurations for a simple queueing system.

Next Time: We'll look at a  re-entrant queueing system. Very non-intuitive, very kool. Simulation was the first way people used to analyze these things.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Crazy Re-entrant Queue

Lesson Overview

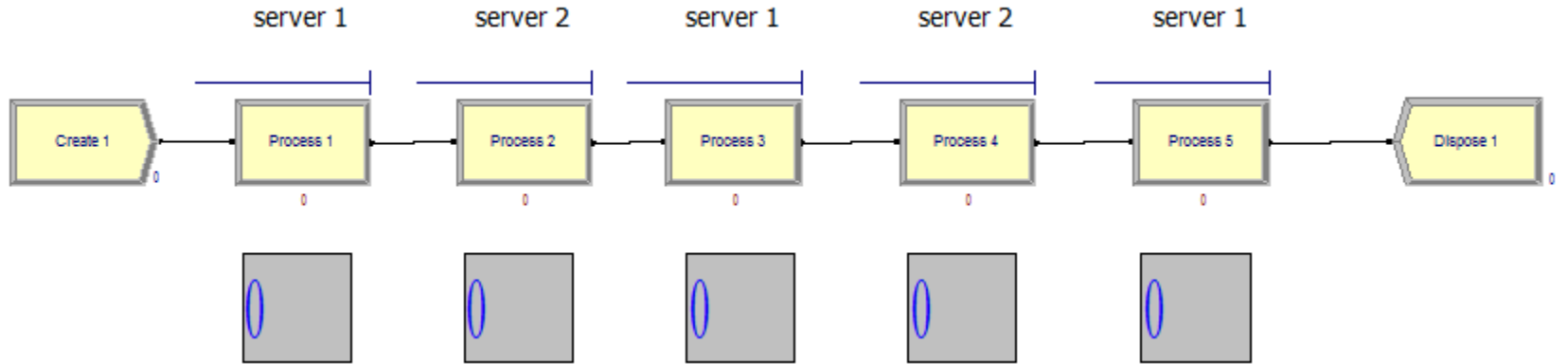
Last Lesson: We compared different queueing strategies – one line vs. two lines.

This Lesson: A more-interesting system – a re-entrant queue.

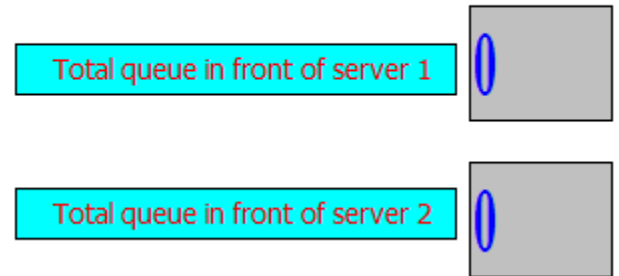
Idea: If you re-use servers and mess around with priorities, you can get some pretty non-intuitive outcomes!

Re-entrant Queues

- Customers go to Server 1, then to Server 2, then back to Server 1, back to 2, and finally back to 1. That is, $1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1$.
 - These returns are why we call the queues “re-entrant”.
 - They’re depicted in Arena as 5 separate Process modules with Seize-Delay-Release trios. (It’s OK to Seize the same server in different Process modules.)
- This seems like a perfectly boring model. Let’s make it interesting.
 - All service times are exponential. Here are the means:
 $1 \text{ (0.1)} \rightarrow 2 \text{ (0.5)} \rightarrow 1 \text{ (0.1)} \rightarrow 2 \text{ (0.1)} \rightarrow 1 \text{ (0.5)}$
 - ...and the customer priorities:
 $1 \text{ (low)} \rightarrow 2 \text{ (high)} \rightarrow 1 \text{ (medium)} \rightarrow 2 \text{ (low)} \rightarrow 1 \text{ (high)}$
 - Thus, on the customer’s 3rd visit to Server 1, he has a high-priority EXPO(0.5) service time.



Get ready for... Demo Time!
You're gonna see mind-blowing behavior!



Summary

This Time: So cool, so very cool.
Words escape me.

Next Time: We're going to get
smart... literally!

SMARTS files! 😊



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

SMARTS and Rockwell
Files

Lesson Overview

Last Lesson: Re-entrant Q's and phenomenal behavior!

This Lesson: Rockwell has hidden away some beautiful little gems: SMARTS files and demos.

Idea: These make great bedtime reading.

• SMARTS + Rockwell Demos

- SMARTS: Very nice tutorial files organized by subject area.
 - E.g., there's a SMARTS file to create customers with a time-dependent arrival rate that varies according to an equation (an **expression**).
 - 100s of SMARTS files
- Rockwell has also prepared many professional demos.
- Libraries > Documents > Rockwell Software > Arena
- Some Typical Examples that we'll demo:
 - SMARTS - Arrivals Varying Rate via Expression
 - SMARTS - Queues Evaluate Conditions Before Proceeding
 - Rockwell Emergency Room
 - Rockwell Mission to Mars

Demo Time!

Summary

This Time: Looked at several Rockwell SMARTS and demo files.

Next Time: We'll demo a easy-looking yet fairly sophisticated manufacturing cell, along with a couple of variations involving transporters and conveyors.



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Manufacturing System

Lesson Overview

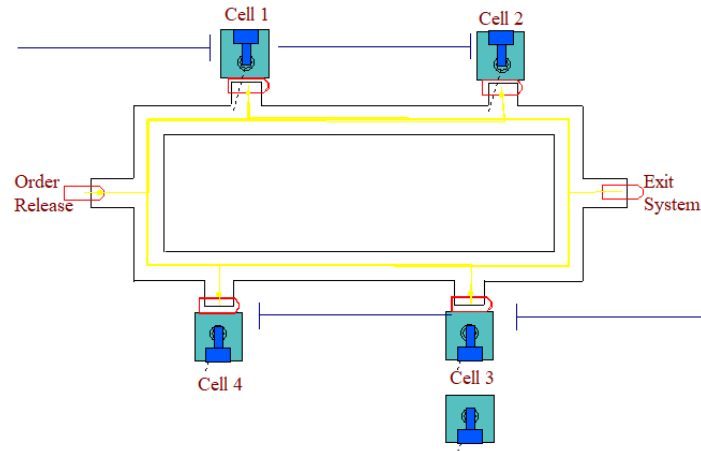
Last Lesson: Cool SMARTS and Rockwell demos.

This Lesson: We'll demo a easy-looking yet fairly sophisticated manufacturing cell, along with a couple of variations involving transporters and conveyors.

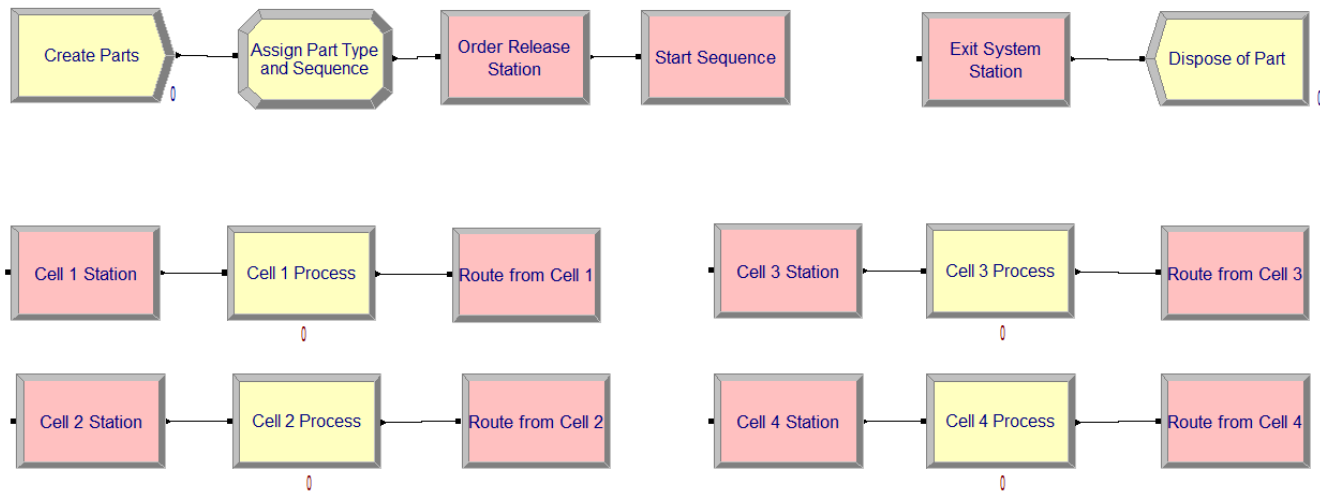
Idea: Introduce movement and multiple part paths. Extremely useful stuff that generalizes!

• Description

- Manufacturing cell with 3 types of parts.
- Each part type follows a different path (“sequence”) thru the system.
 - Different service times at each station (depending on part type and place in visitation sequence).
 - E.g., Part type 2’s visit stations 1 > 2 > 4 > 2 > 3, each with different service times.
- Movement requires the Advanced Transfer template
 - Route, Station, Enter, Leave modules
 - Sequences spreadsheet
 - “Advanced Sets” (which is where the sequences come in).
- Parts can move in a variety of ways:
 - By themselves, transporters, conveyors.
 - Requires construction of transporter and conveyor paths



Demo Time!



Summary

This Time: We did demos on a small, yet sophisticated manufacturing cell, along with some variants involving movement.

Next Time: Details and generalizations of this model



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Manufacturing System in Detail

Part 1. Introduction + Advanced Transfer Panel

Coming Up...

A bunch of additional lessons related to our **Manufacturing Example!**

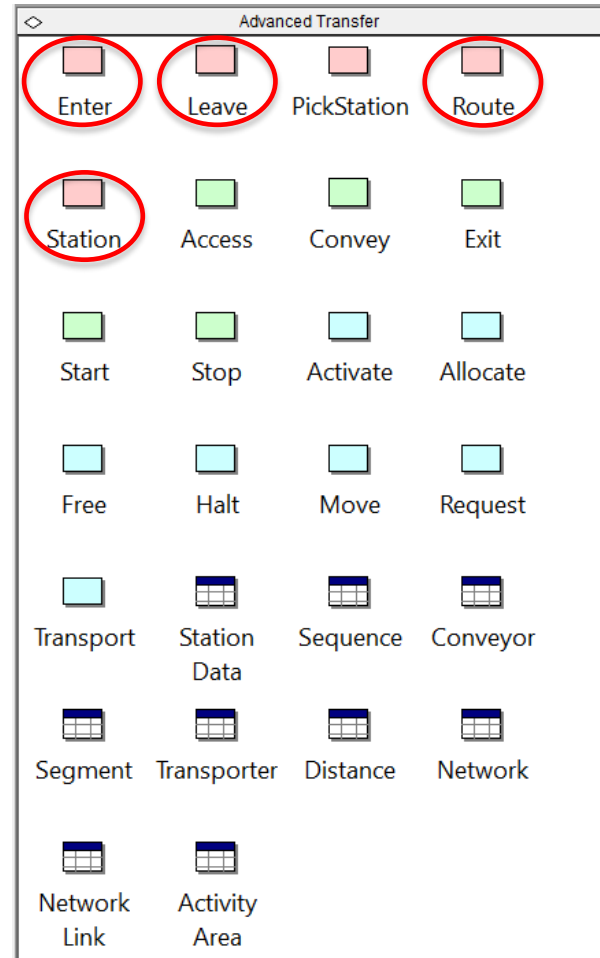
1. **Advanced Transfer Panel Modules** related to movement
 - Station, Route, Enter, Leave
2. **Sequences** of visitation locations
3. Sets and **Advanced Sets** (e.g., sets of Sequences)
4. Detailed walk-through of the model
5. Parts can move in a variety of ways:
 - By themselves
 - Via **transporters**
 - Via **conveyors**
 - Requires construction of transporter and conveyor paths

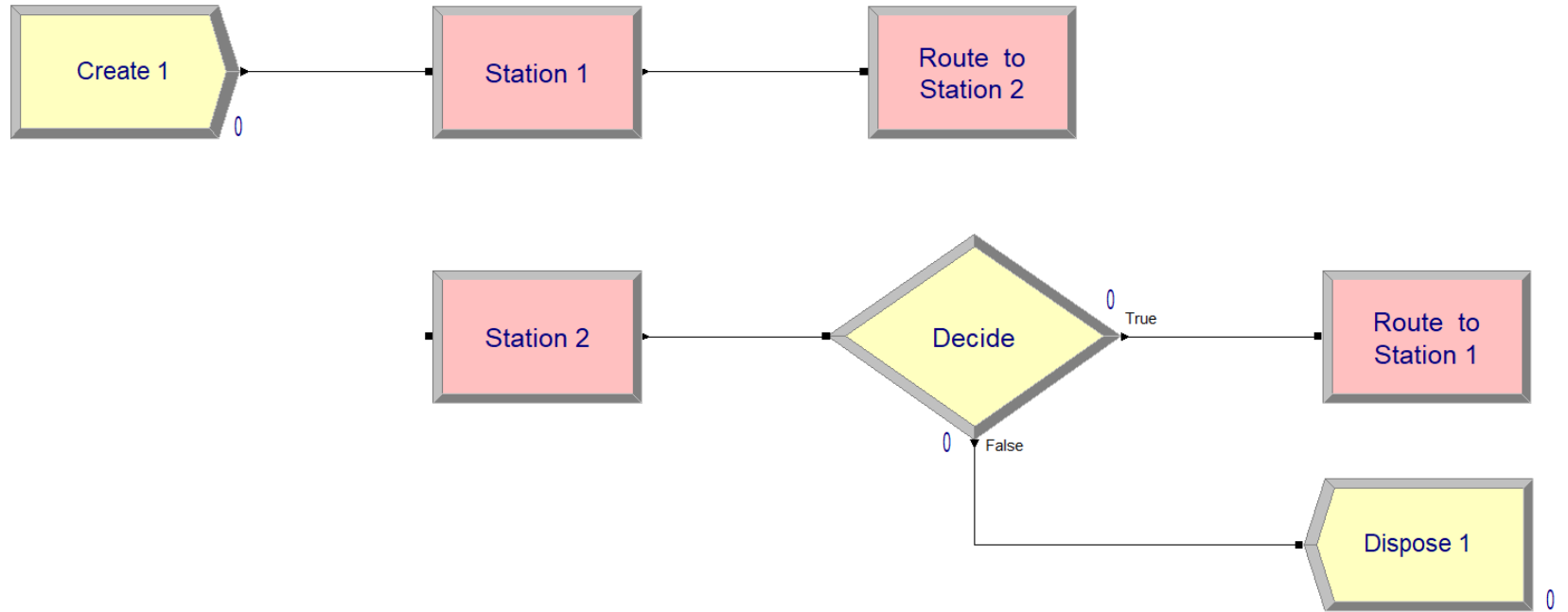
Advanced Transfer

An entire panel related to movement

1. This Lesson: **Where** you move
 - **Station:** Somewhere you can go to
 - **Route:** Tells you where you're going.
Similar to Connect, except it takes time
 - **Enter:** General way to get into a Station
 - **Leave:** General way to depart a Station
2. Subsequent Lessons: **How** you move

Mini-Demo with Station and Route!







Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Manufacturing System in Detail

Part 2. Sequences

Coming Up...

A bunch of additional lessons related to our **Manufacturing Example!**

1. **Advanced Transfer Panel Modules** related to movement
 - Station, Route, Enter, Leave
2. **Sequences of visitation locations**
3. Sets and **Advanced Sets** (e.g., sets of Sequences)
4. Detailed walk-through of the model
5. Parts can move in a variety of ways:
 - By themselves
 - Via **transporters**
 - Via **conveyors**
 - Requires construction of transporter and conveyor paths

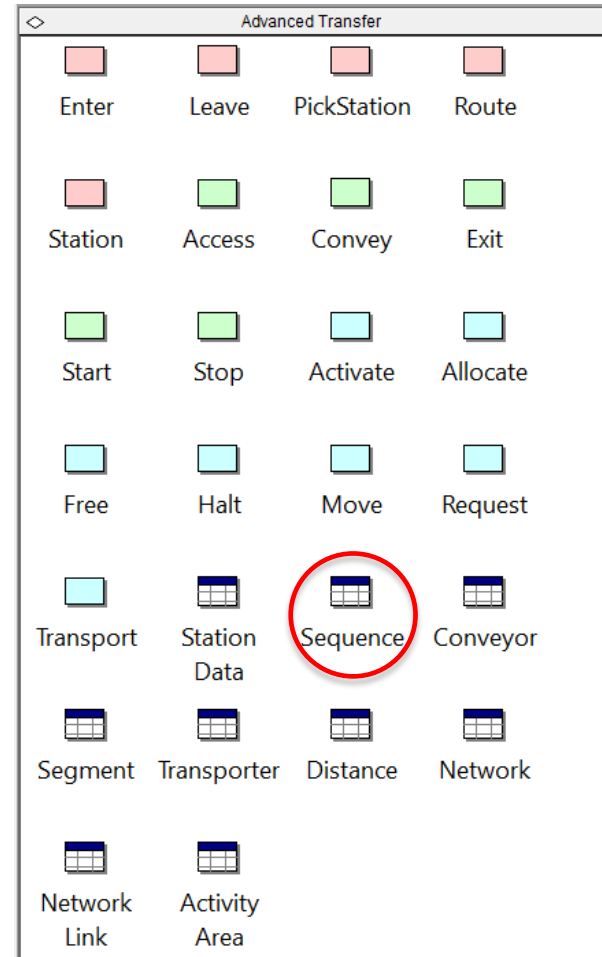
Sequences

Each type of customer follows a different path (**sequence**) thru the system.

- A sequence goes from station to station
- Maybe have different service time distribution at each station (depending on customer type and place in visitation sequence).

Example:

- Part Type 1's visit stations $1 > 2 > 3 > 4$
- Part Type 2's visit $1 > 2 > 4 > 2 > 3$
- Part Type 3's visit $2 > 1 > 3$
- Each with different service times distributions



Three different Sequences

Defining Sequences and Assigning Attributes

Sequence - Advanced Transfer		
	Name	Steps
1	Part 1 Process Plan	5 rows
2 ▶	Part 2 Process	
3	Part 3 Process	
Double-click here to add a new row.		

Steps Part Type 2 Sequence				
	Station Name	Step Name	Next St	Assignments
1	Cell 1	Part 2 Step 1		0 rows
2	Cell 2	Part 2 Step 2		1 rows
3	Cell 4	Part 2 Step 3		1 rows
4	Cell 2			
5	Cell 3			
6	Exit Syst			
Double-click here to add a new row.				

Assignments Type 2 Process Time at Cell 2			
	Assignment Type	Attribute Name	Value
1	Attribute	Process Time	TRIA(4 , 6 , 8)
Double-click here to add a new row.			

Demo Time!



Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Manufacturing System in Detail

Part 3. Advanced Sets

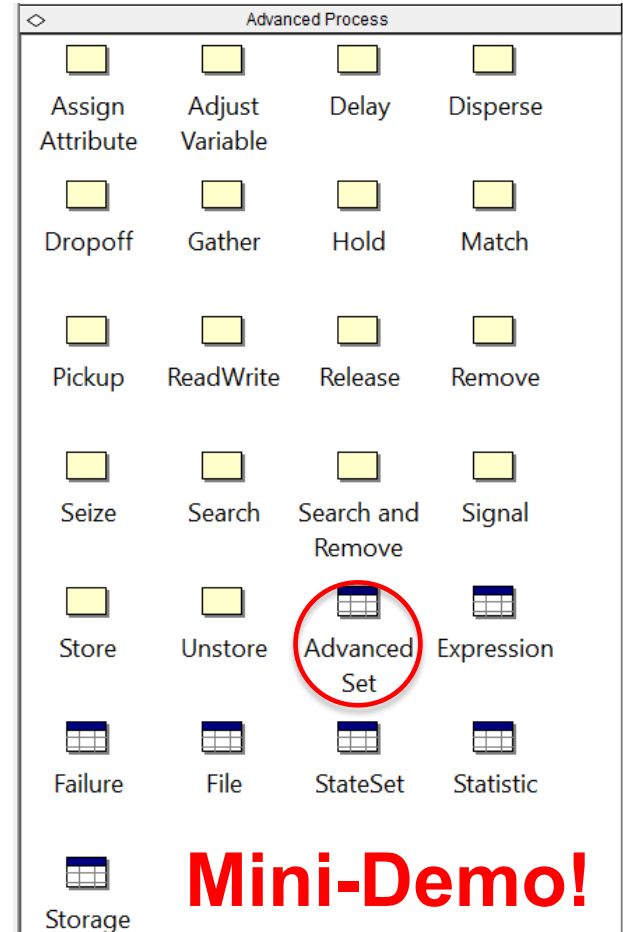
Coming Up...

A bunch of additional lessons related to our **Manufacturing Example!**

1. **Advanced Transfer Panel Modules** related to movement
 - Station, Route, Enter, Leave
2. **Sequences** of visitation locations
3. **Sets and Advanced Sets** (e.g., sets of Sequences)
4. Detailed walk-through of the model
5. Parts can move in a variety of ways:
 - By themselves
 - Via **transporters**
 - Via **conveyors**
 - Requires construction of transporter and conveyor paths

Advanced Sets

- Found in Advanced Process panel
- Used for **kraAaZy** types of sets, e.g., **sets of sequences!**
- Our Advanced set will be called **Part Sequences**, and will consist of the “elements” (sequences):
{Part 1 Process Plan, Part 2 Process Plan, Part 3 Process Plan}
- Think of Part Sequences as a vector,
Part Sequences(2) = Part 2 Process Plan



Mini-Demo!



Computer Simulation

Module 5: Arena

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Manufacturing System in Detail

Part 4. Model Walk-Through

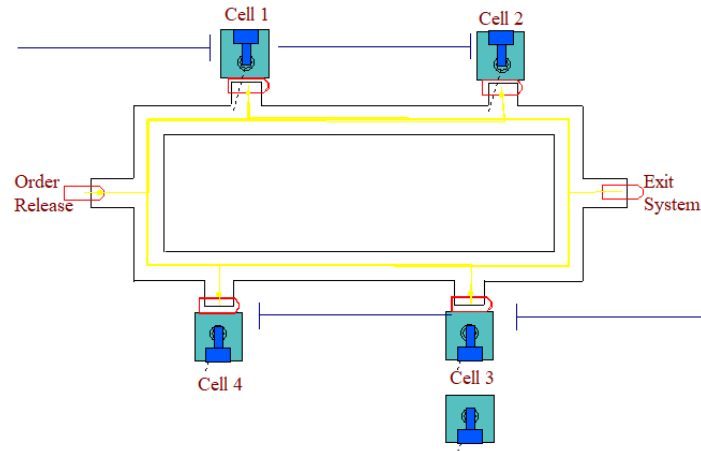
Coming Up...

A bunch of additional lessons related to our **Manufacturing Example!**

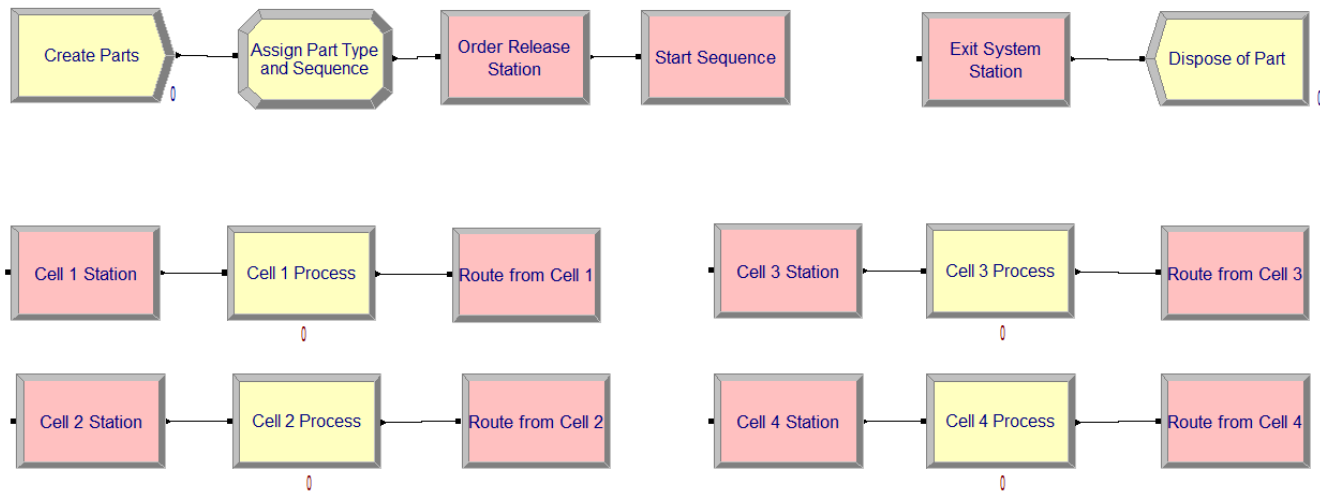
1. **Advanced Transfer Panel Modules** related to movement
 - Station, Route, Enter, Leave
2. **Sequences** of visitation locations
3. Sets and **Advanced Sets** (e.g., sets of Sequences)
4. **Detailed walk-through of the model**
5. Parts can move in a variety of ways:
 - By themselves
 - Via **transporters**
 - Via **conveyors**
 - Requires construction of transporter and conveyor paths

• Description

- Manufacturing cell with three types of parts.
- Each part type follows a different path (“sequence”) thru the system.
 - Different service times at each station (depending on part type and place in visitation sequence).
 - E.g., Part Type 2’s visit stations 1 > 2 > 4 > 2 > 3, each with different service times.
- Movement requires the Advanced Transfer template
 - Route, Station modules (will look at Enter and Leave modules later)
 - Sequences spreadsheet
 - Advanced Sets (which is where the sequences come in).
- Service times usually handled in Sequence definitions (except Cell 1)
- Cell 3 has two servers: An old slow guy, and a young 20% faster guy



Demo Time!





Computer Simulation

Module 5: Arena

Dave Goldman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

A Manufacturing System in Detail

Part 5. Transporters and Conveyors

Coming Up...

A bunch of additional lessons related to our **Manufacturing Example!**

1. **Advanced Transfer Panel Modules** related to movement
 - Station, Route, Enter, Leave
2. **Sequences** of visitation locations
3. Sets and **Advanced Sets** (e.g., sets of Sequences)
4. Detailed walk-through of the model
5. **Parts can move in a variety of ways:**
 - **By themselves**
 - **Via transporters**
 - **Via conveyors**
 - **Requires construction of transporter and conveyor paths**

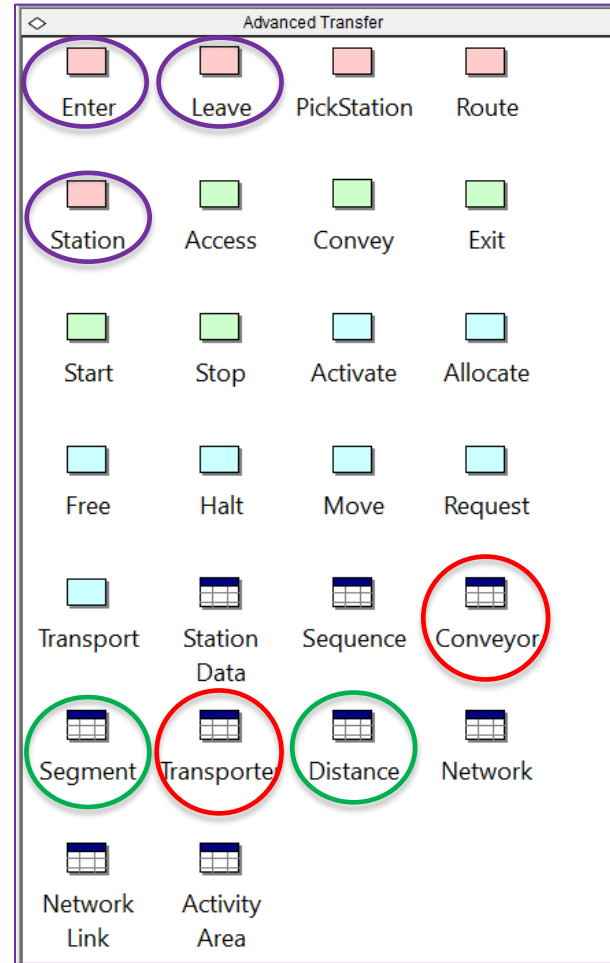
Other Ways to Move

We'll look at **ENTER – LEAVE** modules for Station-to-Station movement.

Different way you can travel, via...

- **Resource** (e.g., an assistant)
 - Seize-Release
- **Transporter** (e.g., a car)
 - Request-Free
 - Requires **Distance** Set
- **Conveyor**
 - Access-Exit
 - Requires **Segment** Set

Demo Time!



Summary

Last Few Lessons: We discussed in detail a small, yet sophisticated manufacturing cell, along with some variants involving movement.

This ends our module on Arena.

- We'll still see Arena later on.
- We only scratched the surface.

Next Time: An entire module on generating $\text{Unif}(0,1)$'s!!!