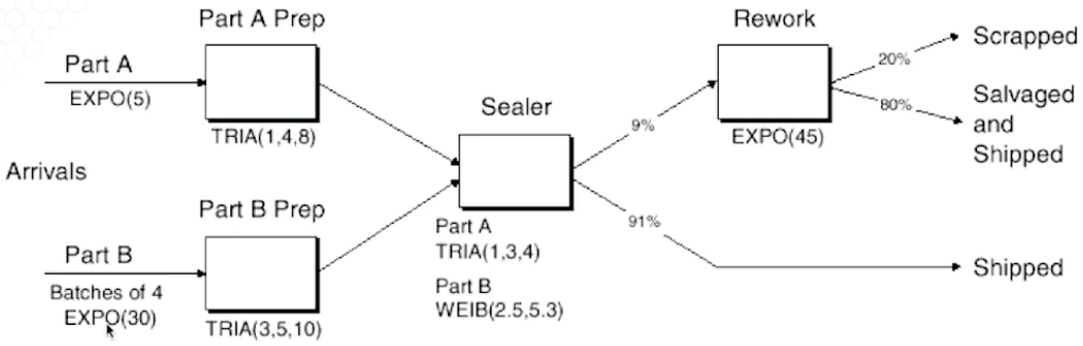
Lesson 15 Two channel Manufacturing Example

L15.1 Simple two channel manufacturing example

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.

I stole this flowchart from KSZ…



- 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).

- 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?)

```
((Entity.Type == Part A) * TRIA(1,3,4)) + ((Entity.Type == Part B) * WEIB(2.5,5.3))
```

L16.1 Fake Customer

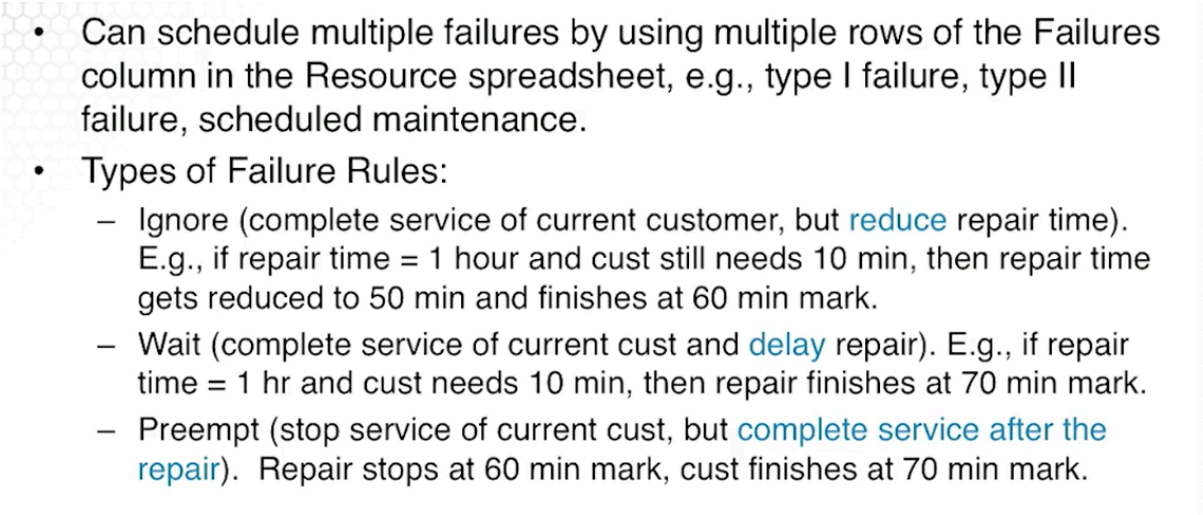
- You can use "fake" customers to accomplish various tasks during a simulation.
- Not actual customers that you care about in terms of waiting times or use of resources.
- Demo Time will explain all!
  - Calculate normal probabilities
  - Keep track of which time period you're in (part of Call Center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

- Fake customers can be used to schedule machine breakdowns, keep track of which time period the simulation is currently in, and carry out other duties not associated with actual customers

L17 Advanced Process Template

- Seize Delay Release Expression Failure
  Complex version of seize delay release, like seize- assign-delay-release
  Complicated Seize's and Release's that might depend on sets of servers
- The Process module does not allow us to use Seize, Delay, Release with sufficient generality.

L18 Resources Failures + Maintenance



L19 The Blocks Template

- Contains a lot of blocks.
- Also the primitive version of seize delay release.
- Alter block

L20 Sets

- Resource sets
- Cross functionality
- Be careful with Seize-Delay-Release for a resource set
- Various Seize Selection Rules

L21 Description of Call Center

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?

L22.1 Call Center Demo
Submodel, Fake customers used as. Timer, Use Queue with capacity 0 to kick out callers
NHPP arrivals, 3 resource sets for Tech support, Ticky Seize-Delay-Release