



Simulation on Logistics Operations

Allen Jiao Logistics Research Center, Nankai University



Electronic Assembly/Test System



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



Electronic Assembly/Test System



- **Part A Description:**
 - **Interarrivals: expo (5) min.**
 - **From arrival point, go immediately to Part A Prep Area**
 - **Process = (machine + deburr + clean) ~ tria (1,4,8) min.**
 - **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) min.**
 - **80% pass, go to Salvaged; Else go to Scrapped**



Electronic Assembly/Test System



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



Goals (Run Conditions, Output)



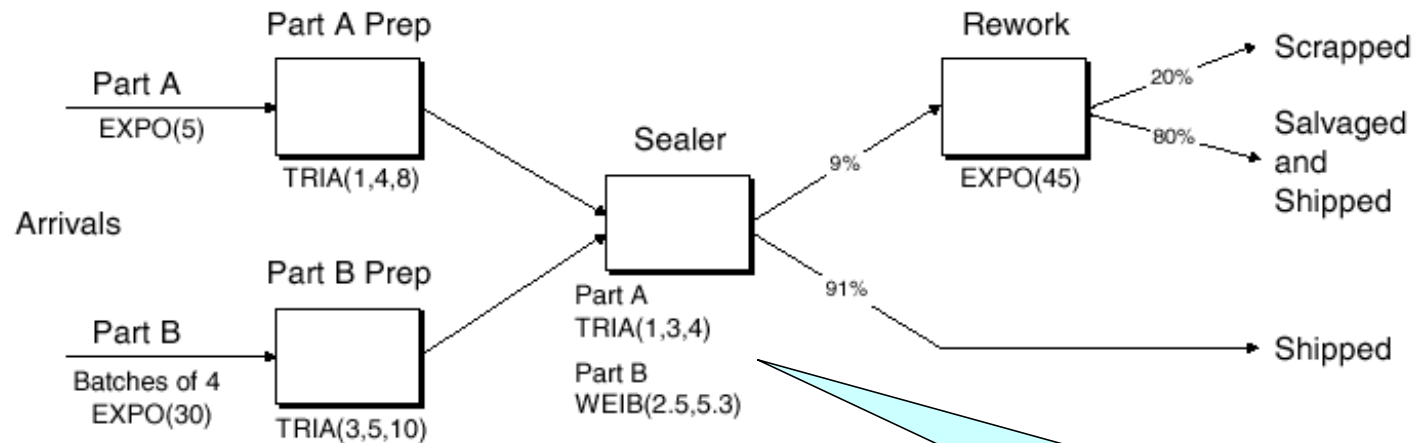
- Start empty & idle, run for **32** hours
- Collect statistics for each work area on
 - **Resource utilization**
 - **Number in queue**
 - **Time in queue**
- For each exit point (Shipped, Salvaged, Scrapped), collect total time in system (a.k.a. **cycle time**)



Model Analysis



- Formulate the model representation by draw a graph



One Server for two different parts with different times.
Kind a tough spot?

Model Analysis



- **Key points**
 - **Entities are the individual parts (two types)**
 - **Separately Create modules for two part types**
 - **Separately Process modules for each Prep area**
 - **Process modules for Sealer and Rework, each followed by a Decide module (2-way by Chance)**
 - **Depart modules for Shipped, Salvaged, Scrapped**
 - **Attribute *Sealer Time* assigned after Creates in Assign modules (since parts have *different* times at *the* Sealer)**
 - **Record modules just before Departs for time in system**



Clear the way for modeling



- **What is Expo()?**
 - **Shanghai Expo? No, Exponential not Exposition**
 - **Good for simulating inter-event times in random arrival**
 - **And good for breakdown process**
 - **Not good for process delay times**
- **What is Tria()?**
 - **When exact form distribution is not known, but estimates (or guesses) for the minimum, maximum, and most likely values are available**
 - **3 parameters for min, mode and max.**



Clear the way for modeling



- **What is weib()**
 - **Weibull (Beta, Alpha)**
 - **Weibull distribution is widely used in reliability models to represent the lifetime of a device.**
 - **If a system consists of a large number of parts that fail independently, and if the system fails when any part fails, then the time between successive failures can be approximated by the Weibull distribution.**
 - **Also used to represent non-negative task times**



Clear the way for modeling

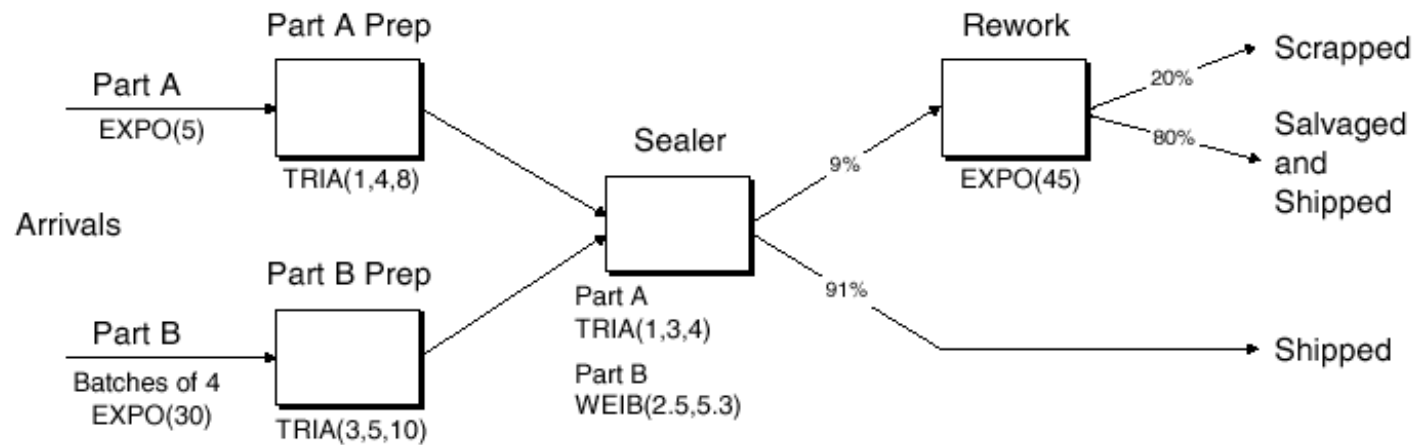


- **What is two way by chance?**
 - **It's what decide module build for.**
- **Arena collects and reports many output statistics by default, but sometimes not all you want**
- **Want time in system (average, maximum) of parts sorted by their exit point (Shipped, Salvaged, Scrapped)**
- **Record module can be placed in flowchart to collect and report various kinds of statistics from within model run as entities pass through it**
- **For Tally-type output performance measures**





- See file: *Model4 Electronic Assembly-Test System.doe*



Run > Setup for Run Control









- **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**



Running the Model



- **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**
 - Slider bar at top, or increase (> key), decrease (< key)
- **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



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
 - 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**



New Problems



- **Rework system actually operates two shifts a day, and on the second shift, there are two operators assigned to the rework operation.**
- **Periodically, the sealer machine breaks down, every $\text{Expo}(120)$ minutes; the time to repair also follows an $\text{expo}(4)$ minutes.**
- **How to build the function into the exited model?**



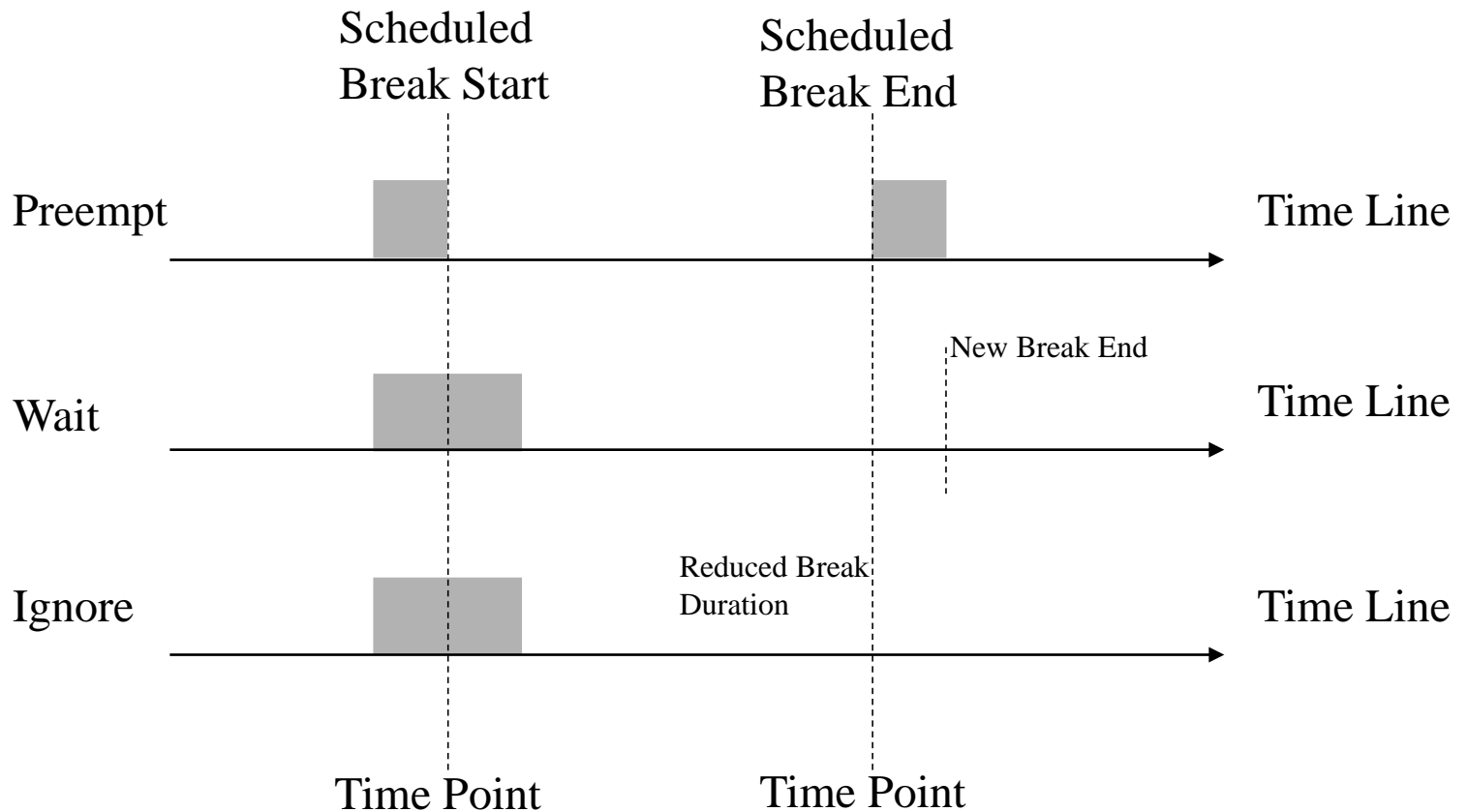
Schedules



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



Schedules Rules



Schedules



- **Define the actual Schedule the Resource will follow – Schedule data module**
 - Row already there since we defined **Rework Schedule**
 - **Format Type is Duration for entries based on elapsed time past simulation start time**
 - **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, with no Variables or Expressions involved**



Resource Failures



- Usually for 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 Time, Down Time, with Units for both



Resource Failures



- **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) acting on a resource**
- **Can re-use defined Failures for multiple Resources (operate independently if they involve random variables)**





- Thanks
- Q&A



Approaching the reality



- **Adding some new features**
 - **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 ~ expo (2) hours**
 - **Repair times ~ expo (4) min.**
 - **Wants to buy racks to hold rework queue**
 - **A rack holds 10 parts**
 - **How many racks should be bought?**
 - **Run for 10 days (16-hour days)**
- **See model changing**
 - **Using *Resource Schedules, Schedule, Failures, Run Parameters***

