0405324: Stochastic System Simulation

Lecture 4: Modeling Basic Operations

(To be covered by lab classes)

Last revision June 21, 2009

Content

- **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 animation**
  - Queues, Entity Pictures, Resource Pictures
  - Adding Plots and Variables
- **Model 4-4: Adding entity travel times**
  - Stations, Transfers, Routes, animation of entity movement
- **Finding and fixing model errors**

Electronic Assembly/Test System (Model 4-1)

The system to be modeled represents the final operations of the production of two different electronic units. The arriving parts are cast metal cases for the units that have already been machined to accept the electronic parts.

The first units, named Part A, are produced in an adjacent department, outside the bounds of this model, with inter-arrival times (to our model) being exponentially distributed with a mean of 5 minutes. Upon arrival, they are transferred to the Part A preparation area, with a transit time of 2 minutes. At the Part A preparation area, the mating faces of the cases are machined to ensure a good seal, and the part is then deburred and cleaned; the process time for the combined operation follows a triangular (1, 4, 8) distribution. The part is then transferred to the sealer, with a transit time of 2 minutes.

The second units, named Part B, are produced in a different building, outside this model's bounds, where they are held until a batch of four units is available; the batch is then sent to the final production area we are modeling. The time between the arrivals of successive batches of Part B to our model is exponential with a mean of 30 minutes. Upon arrival at the Part B preparation area, the batch is separated into the four units, which are processed individually.

Electronic Assembly/Test System (Model 4-1)

The processing for Part B preparation has the same three steps as at the Part A preparation area; the process time at this preparation area follow a triangular (3, 5, 10) distribution. The part is then sent to the sealer with a transit time of 2 minutes.

At the sealer operation, the electronic parts are inserted, the case is assembled and sealed, and the sealed unit is tested. The total process time for these operations depends on the part type: triangular (1, 3, 4) for Part A and normal (2.4, 0.5) for Part B. Ninety-one percent of the parts pass the inspection and are transferred directly to the shipping department. The remaining parts are transferred to the rework area where they are disassembled, repaired, cleaned, assembled again and re-tested. Eighty percent of the parts here are salvaged and transferred to the shipping department as reworked parts.

The remaining parts are transferred to the scrap area. The time to rework follows an exponential distribution with mean of 45 minutes, and is independent of the part type, or status. Assume all transfer times are 2 minutes.

Electronic Assembly/Test System (Model 4-1)

Run condition and 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 (cycle time)

- **First step- Draw the process flow diagram (conceptual model) of the system**
- **Second step- Model the system in ARENA following the step-by-step guide**

Electronic Assembly/Test System (Model 4-1)

Synopsis of the problem:

- Produce two different sealed elect. units (A, B)
- Arriving parts: cast metal cases machined to accept 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 (Model 4-1)

Part A

- **Interarrivals: expo (5) min.**
- **From arrival point, go immediately to Part A Prep**
  - 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

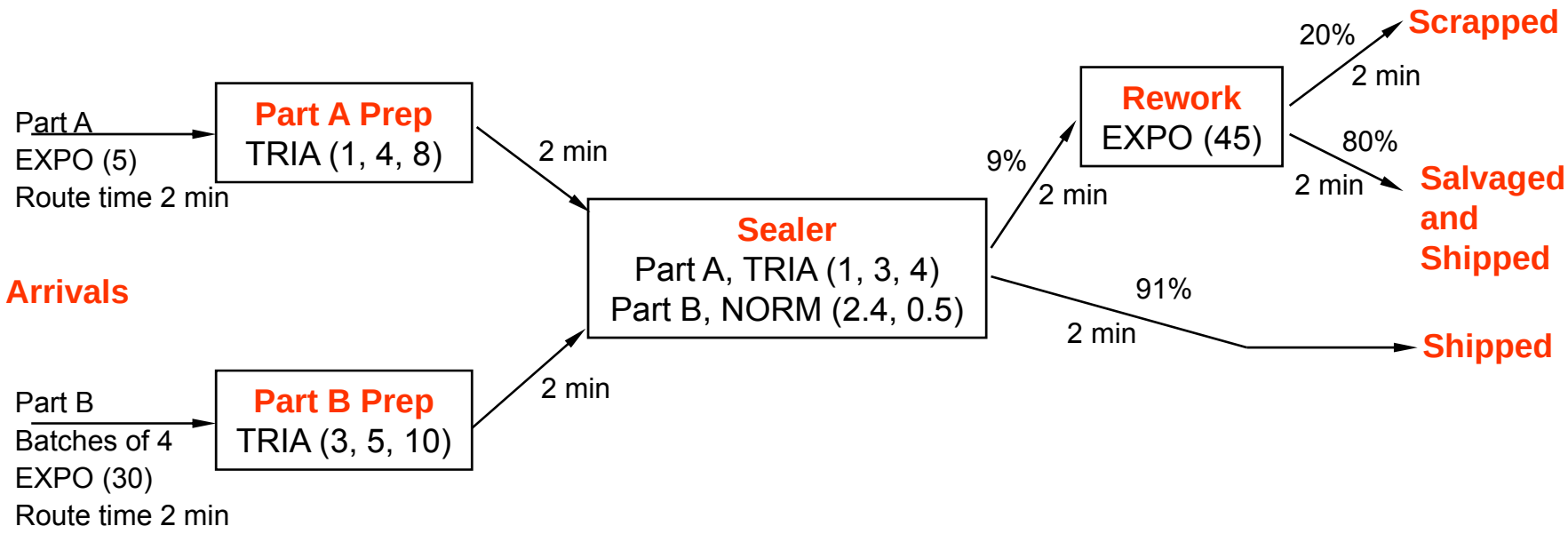
Electronic Assembly/Test System (Model 4-1)

Part B

- **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**
  - 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

Electronic Assembly/Test System (Model 4-1)

The system can be represented by the following process flow diagram:



Next, model the system in ARENA following the step-by-step guide

Electronic Assembly/Test System (Model 4-1): Modeling in ARENA

- **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 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)
  - 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 Model

- **New model window**
- **Attach Basic Process panel (if needed)**
- **Place modules**
  - Create (× 2)
  - Assign (× 2)
  - Process (× 4)
  - Decide (× 2)
  - Record (× 3)
  - Dispose (× 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

Part A Create Module

- **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 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`**

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

TNOW is internal Arena variable name for simulation clock; see Help > Arena Help > Contents > Variables, Functions, and Distributions > Variables > 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)

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`**

If several Resources were named (Add button), entity would have to Seize them all before 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`**

*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`**
- **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)`**

*Had to use general **`Expression`** choice for Delay Type since what we want (**`EXPO`**) is not directly on 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**
- **Want time in system (avg, max) of parts sorted 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 flowchart to collect and report various kinds of statistics from within model run as entities pass through it**
- **For Tally-type output performance measures**

Shipped Parts *Record* Module

- **Name: `Record Shipped Parts`**
- **Type: `Time Interval`**
  - Records time 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 clock value in Assign modules instantly after each entity was Created
- **Tally Name: `Record Shipped Parts`**
  - Determines label in reports

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

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**
  - Separate ones gets animation counts of three dispositions
  - Separate Dispose modules allows for differentially checking 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

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 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 list of pictures that's presented here … more later

Running 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 Results

- **Counters during animation for modules**
  - Create, Dispose, Decide – incremented when entity leaves
  - Process – number of entities currently in module
- **Asked at end if you want to see reports**
  - What you get depends on *Run> Setup> Project Parameters*
    - Looks like Rework area is 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 … Chapter 7
    - 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: Enhanced Electronic Assembly and Test System

- **Original model shown to production manager**
  - Pointed out that this is only first shift of a two-shift day — on second shift there are two operators at Rework (bottleneck station) … 16-hour days
  - Pointed out that 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)
- **Need:** ***Resource Schedules***, ***Resource States***, ***Resource Failures***

Run Conditions

- **Redefine a "day" to be 16 hours – *Run* > *Setup* > *Replication Parameters***
- **Change Replication Length to 10 (of these) days**

Schedules

- **Vary Capacity (no. 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 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 (cont'd.)

- **Define actual Schedule that 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 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

Schedules (cont'd.)

- Alternatively, right-click in 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 row, select Edit via Spreadsheet
  - Enter capacity Value, Duration pairs

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 (cont'd.)

- **Attach this Failure to 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)**

Frequencies

- **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], … for 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)

Frequencies (cont'd.)

- **Add another Frequency (in Statistic module) to give a finer description of Sealer states**
  - Produces 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 separate window)

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)**
- **Sealer is more congested (it now fails)**
- **Rework is less congested (50% higher staffing)**
- **Frequencies report suggests one rack suffices about 95% of the time, two racks all the time**
  - Standard vs. Restricted Percents – see text

Utilizations – Fine Points

- **Two utilizations reported for each Resource**
  - *Instantaneous Utilization* is time-average of ratio of number of units that are busy to number of units that are scheduled
    - By definition, counts periods when zero units are scheduled as zero-utilization periods
  - *Scheduled Utilization* is average number busy divided by average number available
    - No division-by-zero problem, assuming there were ever any units of Resource scheduled at all (if not, it shouldn't be in 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
  - More issues, even finer points – see text

Model 4-3: Enhancing 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
- **Pull animation away from logic in model window**
  - Useful for big models, complex animation
  - Named Views for model logic, animation, or close-ups
- **Default animation objects are connected to model logic and move with the module**
  - Identifiers, physical location (Shift-drag to decouple)

Changing Animation Queues

- **Lengthen (click, drag, maybe hold shift) to "hold" more entities**
  - Simulation logic, results OK if animated queue overflows
- **Rotate to re-orient for realism**
- **Change "form" of queue from *Line* (default) to *Point* — fixed places for entities**
  - Double-click on 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

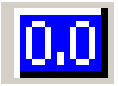
Changing Entity Pictures

- **Earlier – used Entity data module to assign different Initial Pictures to different Entity Types**
- **Can customize list, alter pictures in it**
  - *Edit > Entity Pictures*
  - Left column – names, pictures currently on 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 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 – Chapter 3**
- **Other animation objects from Animate toolbar**
  - Clock (TNOW), variety of formats
  - Level (thermometer) animation
  - Others discussed later

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

- **Generalize Model 4-3**
- **All part transfers now take 2 minutes (not instant)**
  - Model, animate this
  - Materially changes model logic, results
  - Two-minute transfer times for:
    - Arriving parts to prep areas
    - Departing parts to appropriate exit
    - All internal part transfers
  - Transfers take two minutes regardless of distance
    - Fix this (unrealistic) assumption in Chapter 8

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 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 transitway or transporters
  - *Route paths* represent Routes in flowchart/animation

Adding Route Logic – From Arrival

- **Stations and Station Transfers affect both model logic and 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 module (Advanced Transfer panel) – define entity's location
      Module Name vs. Station Name
    - Route module (Advanced Transfer panel) – send entity out
      Route Time, Destination Station
  - No direct connections exiting from Route modules – Route module's Destination Station Name defines that

Adding Remaining Route Logic

- **Add Station modules for entry to each Prep area**
  - Station names are **`Prep A Station`**, **`Prep B Station`**, and are 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 way to get two-minute transfer times:**
  - Insert a Process module with Action = Delay for 2 minutes on each relevant connection
  - Or, use Delay module from Advanced Process panel
- **This *would* work from modeling, numerical-output viewpoints**
- **But would not allow animation of part transfers, so we'll proceed with Stations and Routes**

Altering Animation – Stations

- **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 same logical station (e.g., to represent incoming, outgoing sides)
  - Can drag station markers around later

Altering Animation – Routes

- **Route button  from Animate Transfer toolbar**
  - Options for appearance of entities as they travel route
  - Get cross hairs; click in origin, destination Station Markers
    - Intermediate clicks for corners along route
  - Can drag around endpoints, corners later
  - Alternatively, use Route animation to create both Station markers and Route animation
    - Click for beginning Station marker
    - Intermediate clicks for route corners
    - Double-click for ending Station marker
    - Then go back and double-click on the two Station markers to define their logical Identifiers

Altering 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 route
- **Create a dishonest illusion to animate batch**
  - Assign module just after **`Part B Arrive`**
  - Add assignment of Entity Picture to **`Picture.Batch B`**
  - *Edit* > *Entity Pictures* to draw 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`**

Finding and Fixing Model Errors

- **If error prevents model from running, Arena will try to detect and lead you to it in Check or Run**
  - Undefined (or inconsistently spelled) Variables, Attributes, Resources
  - Unconnected modules
  - Duplicate names
  - Examples – see text
- **Highlight Active Module – selects active module during run animation**
- ***View* > *Layers* while running – change what shows during run animation**

Finding and Fixing Model Errors (cont'd.)

- **Module Break – stop when entity reaches module**
- **Debug Bar**
  - *View* > *Debug Bar*
  - Breakpoints, Calendar, Active Entity, Watch
  - Run Controller
  - Examples – see text

Continued in Lecture 5