Chapter 5
Modeling Basic Operations and Inputs
What We’ll Do …

• **Model 5.1: Electronic assembly/test system**
  – Modeling approaches
  – Stations, Transfers, Pictures

• **Model 5.2: Enhanced electronic assembly/test**
  – Resources, Resource States, Schedules
  – Saving statistical data
  – Output Analyzer (data display only)

• **Model 5.3: Enhancing the animation**
  – Queues, Entity Pictures, Resource Pictures
  – Adding Plots and Variables
What We’ll Do ... (cont’d.)

• Input analysis
  – Specifying input distributions, parameters
  – Deterministic vs. random input
  – Collecting and using data
  – Fitting input distributions via the Input Analyzer
  – No data?
  – Nonstationary arrival processes
  – Multivariate and correlated input data
Electronic Assembly/Test System
(Model 5.1)

- Produce two different sealed elect. 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 — Salvage or Scrap

**Simulation with Arena — Chapter 5 — Modeling Basic Operations and Inputs**
Part A

• Interarrivals: expo (5) minutes
• Transit times between all stations: 2 min.
  – No wait for person, cart to transfer — parts have their own feet (relax this assumption in Chapt. 7)
• Go to Part A Prep area
  – Process = (machine + deburr + clean) ~ tria (1,4,8)
• Go to Sealer
  – Process = (assemble + test) ~ tria (1,3,4) min.
  – 91% pass, go to Shipped; Else go to Rework
• Rework: (re-process + testing) ~ expo (45)
  – 80% pass, go to Salvage/Ship; Else go to Scrap
Part B

- Interarrivals: *batches* of 4, expo (30) min.
- Transit times between all stations: 2 min.
- Go to Part B Prep area
  - Process = (machine + deburr + clean) ~ tria (3,5,10)
- Go to Sealer
  - Process = (assemble + test) ~ norm (2.4, 0.5) 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 Salvage/Ship; Else go to Scrap
Run Conditions, Output, Animation

• Start empty & idle, run for 2,000 minutes

• Output:
  – Utilization of all resources
  – Number in each queue
  – Time in each queue
  – Cycle time (flowtime) separated out by shipped parts, salvaged/shipped parts, scrapped parts

• Animation:
  – Queues, busy/idle resources as before
  – Entity movement between stations (2 min. transfers)
Developing a Modeling Approach

- Define submodels, modules, data structures, control logic
- Appropriate level of detail — judgment call
- Often multiple ways to model, represent logic
- This model:
  - Separate Arrive modules for two part types
  - Separate Server modules for each Prep area
  - Inspect modules for Sealer and Rework
  - Depart modules for Shipping, Salvage, Scrap
  - Transfer times: Route
  - Attribute Sealer Time assigned at Arrival (parts have different times at the Sealer station)
Stations

• Up to now: no (zero) transfer times between stations — realistic??

• **Station**: Physical location for an activity (or a group of activities)

• Way to model entity flow, transfer generally

• Provide animation “launching, landing pads”

• Each Station has a unique name

• Station **marker**:  

• Can separate logical station, physical station
Station Transfers

• Send an entity from one station to another

• Modeling options:
  – *Connect* (zero time)
  – *Route* (possibly positive time, no constraints)
  – *Resource-constrained* (freeway, communications)
  – *Transporters*
  – *Conveyors*

• Animation facility for each transfer option
  – For Route: Route object from Animate toolbar
Roughing Out the Model

• New model window
• Attach Common Panel
• Place modules
  – Arrive (two)
  – Server (two, for Prep Areas)
  – Inspect (two, for Sealer and Rework)
  – Depart (three, for Shipping, Salvage, and Scrap)
  – Simulate
• Right mouse button — repeat last action
Part A Arrive Module

• Main dialog (default what’s not mentioned)
  – Enter Data
    • Station: Part A Arrive (type it in — first mention)
  – Arrival Data
    • Time Between: EXPO(5) (pull-down list)
    • Mark Time Attribute: Arrival Time (type it in)
  – Leave Data
    • Station: Part A Prep (type it in)
    • Route Time: 2

• Assign subdialog (button); Add... button
  – Attribute: Sealer Time (type it in)
  – Value: TRIA(1,3,4) (pull-down list)
Part B Arrive Module

• Same as for Part A Arrive, except:
  – Station: Part B Arrive
  – Batch Size: 4
  – Time Between: EXPO(30)
  – Leave Data Station: Part B Prep
  – Attrib. Sealer Time Value: NORM(2.4, 0.5)

• Each arrival creates four separate entities
  – Quadruplets separated at birth
  – Flow independently
  – Independent Sealer Time values assigned

- Exploit pull-downs where possible (Station names, Attribute names) for earlier definitions

- Main dialog
  - Enter Data
    - Station: Part A Prep [Part B Prep] (pull-down)
  - Server Data
    - Process Time: TRIA(1,4,8) [TRIA(3,5,10)]
  - Leave Data
    - Station: Sealer (type)
    - Route Time: 2

- Accept defaults for Resource name, Resource Statistics, subdialogs (Queue, Resource, etc.)
Sealer Inspect Module

• Main dialog
  – Enter Data
    • Station: Sealer (pull-down)
  – Server Data
    • Process Time: Sealer Time (attribute, must type in)
    • Failure Probability: 0.09 (type)
  – Pass Inspection Leave Data
    • Station: Shipping (type)
    • Route Time: 2 (type)
  – Fail Inspection Leave Data
    • Station: Rework (type)
    • Route Time: 2 (type)
Rework Inspect Module

• **Main dialog**
  
  – Enter Data
    • Station: **Rework** (pull-down)
  
  – Server Data
    • Process Time: **EXPO(45)** (type)
    • Failure Probability: 0.2 (type)
  
  – Pass Inspection Leave Data
    • Station: **Salvaged Parts** (type)
    • Route Time: 2 (type)
  
  – Fail Inspection Leave Data
    • Station: **Scrap** (type)
    • Route Time: 2 (type)
Depart Modules

• Three separate modules: Shipping, Salvaged Parts, and Scrap

• Main dialog for Shipping (others are similar)
  – Enter Data
    • Station: Shipping (pull-down)
  – Count
    • Individual Counter: select (accept default counter name)
  – Tally
    • Individual Tally: select
    • Attribute: Arrival Time (accept Interval default selection)
Simulate Module

• Specify termination rule (among other things)

• Main dialog
  – Project
    • Title: Electronic Assembly and Test
    • Analyst: Mr. Munchkin
  – Replicate
    • Length of Replication: 2000

• Accept defaults for
  – Date (computer clock)
  – Number of Replications (1)
  – Beginning Time (0.0)
  – Initialize everything between replications
Animated Routes

• **Paths to display entities during transfers**
  – Not necessary for numerical results
  – Just for animation to connect Stations

• **Animate panel, Route button**
  – Route dialog for appearance, orientation (just accept all defaults, hit OK)
  – Cursor changes to crosshairs
  – Click in beginning station, maybe click “corners” for polyline route, click in ending station
  – Repeat for all Routes to be animated (right click)
Run

- **Check** (if desired)
  - Find button to help find errors
- **Go** (will automatically pre-Check if needed)
- **Pause**
  - Step
  - Double-click on things to see status (debug)
- **Fast Forward**
  - Even faster: Run/Setup…/Mode/Batch Run (No Animation) before running
- II, remove toolbars during run (they return)
- Full-screen mode: Run/Setup…/Miscellaneous
Viewing the Results

• When done, asked if you want to see numerical results (text)
  – Uses Notepad or other viewer in separate window
  – Also saves as text file model_name.out

• Tally, Discrete Change (a.k.a. time-persistent), Counters areas (if present in model)
  – Columns for averages, min, max, number of observations or final value

• Half Width column:
  – For 95% confidence interval on steady-state (long-run) expected average
  – May not have enough data (see Chapter 7 ...)

Simulation with Arena — Chapter 5 — Modeling Basic Operations and Inputs
Enhanced Model
(Model 5.2)

• A Story
  – Original model shown to production manager
  – Pointed out that this is only the first shift of a two-shift day — on second shift there are two operators at Rework (the bottleneck station)
  – Pointed out that the Sealer fails sometimes
    • Uptimes ~ exponential, mean 2 hours
    • Repair times ~ exponential, mean 4 minutes

• Need: *Schedules, Resource States, Resource Failures*
Schedules

• Vary Capacity (number of units) of a resource over time

• Alternative to “Capacity” Capacity Type in Server, Inspect, Process modules

• Arena actually has four automatically defined Resource States, keeps statistics on all:
  – Idle (as before)
  – Busy (as before)
  – Inactive: capacity reduced to zero
  – Failed: model downtimes, unavailable
Schedules (cont’d.)

• **Server Data area:**
  – For Capacity Type, pick Schedule rather than Capacity (pull-down)
  – “Capacity” box changes to:
    • Schedule — name the schedule (defined below)
    • Choice between Preempt/Ignore/Wait — what if resource is busy when scheduled to go down? (See book.)
  – Get a new Schedule… button below — push it
    • Schedule subdialog
    • Add (capacity, duration) pairs
    • If all durations are specified, schedule repeats forever
    • If any duration is empty, it defaults to infinity
Ignore/Wait/Preempt

duration of scheduled capacity decrease

Ignore
Wait
Preempt

start of scheduled capacity decrease

time

Simulation with Arena — Chapter 5 — Modeling Basic Operations and Inputs
Resource Downtimes

• Bring one unit of a resource down — other units (if any) still up

• Resource… button
  – Downtime Name
  – Time Between Downtimes (anything — pull-down for distributions)
  – Downtime (anything, distribution pull-down)

• Can have multiple Downtimes (separate names) for a Resource
Resource Failures

• All units of a resource come down

• Resource… button
  – Failure Name
  – Based on entity Count or elapsed Time
  – Preempt/Ignore/Wait for come-down rule
  – If based on Count, the Count for uptime
    If based on Time, the Uptime
  – Downtime (anything, distribution pull-down)

• Can have multiple Failures (separate names) for a Resource
Saving Statistical Data

• **Observe, maybe save different kinds of data**
  – Non-default output performance measure
    • e.g., % of time queue length > 5
  – Postprocessing via Output Analyzer
    • Note that dynamic animated plots disappear when done
    • Statistical analysis of output data, statistical inference
  – Export to other applications (spreadsheets, etc.)

• **Save records of Time-Persistent data, Tallies, Counters, “Frequencies” (new)**

• **How?** *Statistics module* (Common panel)
The Statistics Module

• Five different areas, for different kinds of stats
• In an area, Add… button for what you want
  – Subdialog depends on type area (type of stat)
  – Option to save data to a (binary) file — “name.dat” (including the double quotes); name could include drive, path
• Time-Persistent area
  – Select data object, later dialogs react to selection
• Tallies area
  – Select Tally Name
• Other areas discussed later ...
Frequency Statistics

• A “finer” description of an output
• Record time-persistent occurrence frequency of a Variable, Expression, or State
• Example: Want to know % of time the Rework queue is of length 0, (0, 10], (10, 20], etc.
  – Statistics module, Frequencies area
  – Add… button
  – Expression: Variable, general expression
    • Arena function NQ(queue name): queue length
    • Others: NR(resource name): no. busy
      MR(resource name): no. available
  – Define categories (Constant or Range)
The Output Analyzer

- Separate application, also accessible via Tools menu in Arena
- Reads binary files saved by Arena
- Various kinds of output-data display, analysis
  - For now: just data-display functions
- Advisable (not required) — define, maybe save a data group (File/New or , then Add…)
  - List of output files of interest — one model or many
  - Eases tasks by “screening” for these files only
  - Save in file called whatever.dgr, Open next time
The Output Analyzer (cont’d.)

- **Plot** time-persistent data
  - Graph/Plot or
  - Can overlay several curves (Sensible? Units?)
  - Options for plot Title, axis Labels, crop axes

- **Moving-average plots**: “smooth” over time
  - Moving-average window Value
  - Exponential smoothing, Forecasting

- **Barcharts**: like Plot, cosmetically different

- **Histograms** of data
  - Beware: autocorrelation
Enhancing the Animation (Model 5.3)

• Get “Spartan” generic default animation for many things
  – Usually sufficient for verification, validation

• Often want to customize, enhance it a bit
  – More realism, impact

• Can pull animation away from model logic in model window
  – Useful for big models, complex animation
  – Set up Named Views for model logic, animation, or close-ups of parts of animation
Changing Animation Queues

- Lengthen (click, drag, maybe hold shift)
- Rotate to re-orient for realism
- Change the “form” of the queue from *Line* (the default) to *Point* — fixed places for entities
  - Double-click on the queue
  - Select Type to be Point
  - Click Points… button
  - Successively click Add for points, then OK
  - Drag them around on screen
  - *Check* Rotate box to show entities turning
Changing the Entity Pictures

• Distinguish between entity types, change them in process, realistically represent batches

• Default picture: above Simulate module

• Define different picture: Animate… button and subdialog in many modules, including:
  – Arrive: choose different Initial Entity Picture
  – Server: Change when entering or leaving
  – Inspect: Change when entering, “pass” leave, or “fail” leave

• Give desired Picture a name here
Changing the Entity Pictures (cont’d.)

• After defining names, must edit/create/read
  – Double-click on default picture above Simulate
  – Make sure Default picture is selected (depressed)
  – Copy, select the copied picture
  – Select name from Value pull-down to rename copy
  – Either
    • Double-click to edit (for artists only)
    • Open Picture library (.plb file), select desired picture from scrolling window, hit << button

• Reference point — where entity moves, sits

• Application: “hidden” batches (Model 5.3)
Changing Resource Pictures

• Realism, indicate state (Idle, Busy, etc.)
• Double-click, edit similarly to entity pictures
  – Artwork
  – Picture libraries (.plb files)
  – Example: Sealer resource in Model 5.3
• Seize point — place for realism (layers, etc.)
• Adjust size — Size Factor
• Multiple-capacity resources
  – Multiple seize points (Rework resource, Model 5.3)
Adding Plots and Animated Variables

• Animate module from Common panel
  – Alternative: Animate toolbar buttons, but Animate module is easier

• Select Data Object to observe

• Select Information to display (depends on Data Object selected)

• Check off mode(s) of display (default: all)
  – For Plots:
    • Have to guess at Max Y (maybe revise after run …)
    • History Points = no. of plot points to display at a time
Input Analysis: Specifying Model Parameters, Distributions

• **Structural** modeling: what we’ve done so far
  – Logical aspects — entities, resources, paths, etc.

• **Quantitative** modeling
  – Numerical, distributional specifications
  – Like structural modeling, need to observe system’s operation, take data if possible
Deterministic vs. Random Inputs

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

• **Random** (a.k.a. **stochastic**): model as a distribution, “draw” or “generate” values from to drive simulation
  – Transfer, Interarrival, Processing times
  – What distribution? What distributional parameters?
  – Causes simulation output to be random, too

• **Don’t just assume randomness away — validity**
Collecting Data

• Generally hard, expensive, frustrating, boring
  – System might not exist
  – Data available on the wrong things — might have to change model according to what’s available
  – Incomplete, “dirty” data
  – Too much data (!)

• Sensitivity of outputs to uncertainty in inputs

• Match model detail to quality of data

• Cost — should be budgeted in project

• Capture variability in data — model validity

• Garbage In, Garbage Out (GIGO)
Using Data: Alternatives and Issues

• Use data “directly” in simulation
  – Read actual observed values to drive the model inputs (interarrivals, service times, part types, …)
  – All values will be “legal” and realistic
  – But can never go outside your observed data
  – May not have enough data for long or many runs
  – Computationally slow (reading disk files)

• Or, fit probability distribution to data
  – “Draw” or “generate” synthetic observations from this distribution to drive the model inputs
  – We’ve done it this way so far
  – Can go beyond observed data (good and bad)
  – May not get a good “fit” to data — validity?
Fitting Distributions via the Arena Input Analyzer

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

• **Arena Input Analyzer**
  – Separate application, also accessible via Tools menu in Arena
  – Fits distributions, gives valid Arena expression for generation to paste directly into simulation model
Fitting Distributions via the Arena Input Analyzer (cont’d.)

- Fitting = deciding on distribution form (exponential, gamma, empirical, etc.) and estimating its parameters
  - Several different methods (Maximum likelihood, moment matching, least squares, …)
  - Assess goodness of fit via hypothesis tests
    - $H_0$: fitted distribution adequately represents the data
    - Get $p$ value for test (small = poor fit)

- Fitted “theoretical” vs. empirical distribution
- Continuous vs. discrete data, distribution
- “Best” fit from among several distributions
Data Files for the Input Analyzer

• Create the data file (editor, word processor, spreadsheet, ...)
  – Must be plain ASCII text (save as text or export)
  – Data values separated by white space (blanks, tabs, linefeeds)
  – Otherwise free format

• Open data file from within Input Analyzer
  – File/New menu or
  – File/Data File/Use Existing … menu or
  – Get histogram, basic summary of data
  – To see data file: Window/Input Data menu

• Can generate “fake” data file to play around
  – File/Data File/Generate New … menu
The Fit Menu

• Fits distributions, does goodness-of-fit tests
• Fit a specific distribution form
  – Plots density over histogram for visual “test”
  – Gives exact expression to Copy and Paste (Ctrl+C, Ctrl+V) over into simulation model
  – May include “offset” depending on distribution
  – Gives results of goodness-of-fit tests
    • Chi square, Kolmogorov-Smirnov tests
    • Most important part: *p-value*, always between 0 and 1:
      Probability of getting a data set that’s more inconsistent with the fitted distribution than the data set you actually have, if the the fitted distribution is truly “the truth”
      “Small” $p$ (< 0.05 or so): poor fit (try again or give up)
The Fit Menu (cont’d.)

• Fit all Arena’s (theoretical) distributions at once
  – *Fit/Fit All* menu or
  – Returns the *minimum square-error* distribution
    • Square error = sum of squared discrepancies between histogram frequencies and fitted-distribution frequencies
    • Can depend on histogram intervals chosen: different intervals can lead to different “best” distribution
  – Could still be a poor fit, though (check *p* value)
  – To see all distributions, ranked: *Window/Fit All Summary* or
The Fit Menu (cont’d.)

• “Fit” Empirical distribution (continuous or discrete): *Fit/Empirical*
  
  – Can interpret results as a Discrete or Continuous distribution
    
    • Discrete: get pairs (*Cumulative* Probability, Value)
    • Continuous: Arena will linearly interpolate *within* the data range according to these pairs (so you can never generate values outside the range, which might be good or bad)

  – Empirical distribution can be used when “theoretical” distributions fit poorly, or intentionally
Some Issues in Fitting Input Distributions

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

- Happens more often than you’d like
- No good solution; some (bad) options:
  - Interview “experts”
    - Min, Max: Uniform
    - Avg., % error or absolute error: Uniform
    - Min, Mode, Max: Triangular
      - Mode can be different from Mean — allows asymmetry
  - Interarrivals — independent, stationary
    - Exponential — still need some value for mean
  - Number of “random” events in an interval: Poisson
  - Sum of independent “pieces”: normal
  - Product of independent “pieces”: lognormal
Nonstationary Arrival Processes

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

• It can be critical to model this nonstationarity for model validity
  – Ignoring peaks, valleys can mask important behavior
  – Can miss rush hours, etc.

• Good model: Nonstationary Poisson process
Nonstationary Arrival Processes
(cont’d.)

• Two issues:
  – How to specify/estimate the rate function
  – How to generate from it properly during the simulation (will be discussed in Chapters 8, 11 …)

• Several ways to estimate rate function — we’ll just do the **piecewise-constant** method
  – Divide time frame of simulation into subintervals of time over which you think rate is fairly flat
  – Compute observed rate within each subinterval
  – Be **very** careful about time units!
    • Model time units = minutes
    • Subintervals = half hour (= 30 minutes)
    • 45 arrivals in the half hour; rate = 45/30 = 1.5 *per minute*
Multivariate and Correlated Input Data

• Usually we assume that all generated random observations across a simulation are independent (though from possibly different distributions)

• Sometimes this isn’t true:
  – A “difficult” part requires long processing in both the Prep and Sealer operations
  – This is positive correlation

• Ignoring such relations can invalidate model

• See textbook for ideas, references