Chapter 6
Intermediate Modeling and Terminating Statistical Analysis
What We’ll Do ...

• A small manufacturing system (Model 6.1)
  – Sequences
  – Variables and Expressions
  – Sets
  – Importing CAD drawings
  – Verification (a.k.a. debugging)

• Statistical analysis of output data: terminating simulations
  – Time frame of simulations
  – Replicating and confidence intervals (Model 6.2)
  – Comparing alternatives
A Small Manufacturing System

- Part arrivals, four cells, part departures
- Cells 1, 2, and 4: single machine each
- Cell 3: two machines — newer one 20% faster
  - Need: way to model non-identical resource units
- Circular layout of cells
- Parts enter at left, exit at right, travel only clockwise, all transfer times = 2 min.
A Small Manufacturing System
(cont’d.)

• Three separate part types
  – Inter arrivals (all types merged) ~ expo(13) minutes
  – 26% type 1, 48% type 2, 26% type 3

• Different part types follow different routes, have different (triangular) processing times:

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Cell/Time</th>
<th>Cell/Time</th>
<th>Cell/Time</th>
<th>Cell/Time</th>
<th>Cell/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 6, 8, 10</td>
<td>2 5, 8, 10</td>
<td>3 15, 20, 25</td>
<td>4 8, 12, 16</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 11, 13, 15</td>
<td>2 4, 6, 8</td>
<td>4 15, 18, 21</td>
<td>2 6, 9, 12</td>
<td>3 27, 33, 39</td>
</tr>
<tr>
<td>3</td>
<td>2 7, 9, 11</td>
<td>1 7, 10, 13</td>
<td>3 18, 23, 28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

– Need: way to change routing depending on part type — process plans
Sequences

• Sequences module on Common panel — a data module

• Define named sequences of stations to visit
  – Generally depends on entity type
  – Can assign attributes or variables at each station in sequence, which can depend on entity type
  – In Leave Data area of modules, select Seq (rather than Connect or Route) to get entity to follow a sequence
  – Must assign a sequence to entities doing this
Sequences (cont’d.)

• Arena internally keeps track of Sequence-following entity via automatic attributes
  – Sequence name, NS
  – Station (where entity is or is going to), M
  – Jobstep along the sequence, IS

• Normally, entity Arrives, is assigned a Sequence name, travels its route, exits
  – Can interrupt this sequence, jump forward or backward (tricky)

• Remember to define the “exit” station
Variables

• Allow re-use of the same number(s) in different places

• Can only be constant values, but any entity can reassign the value of a Variable

• Variables module from Common panel
  – Data module
  – Defines names, initial values of Variables
  – Can be a scalar, vector, or 2-dim. matrix

• Cannot involve arithmetic, entity attributes, other Variables, or distributions
Expressions

• Similar motivation to Variables — re-use the same “thing” in several places in the model
• A fixed “formula” or function that can involve arithmetic, entity attributes, other Variables, and distributions — very general
• However, the form of the expression cannot be changed during the simulation
• Expressions module from Common panel
  – Data module
  – Defines names, form of Expressions
  – Can be a scalar, vector, or 2-dim. matrix
Sets

- Group similar objects (resources, sequences, pictures, other) together under a single name
- Define: Sets module from Common panel
- Refer to objects in a Set by their original name (independent of the Set membership) or by an index into the Set
- An object can be a member of more than one Set (or not be in any Sets)
- Can form Sets of just about anything
  - A given Set must have the same kind of members
Sets (cont’d.)

• Perhaps the most common Sets: Resources
  – Allows dissimilar resources to be grouped — more general than multi-Capacity single Resources, where they all have to be identical
  – Entities can choose among members of a Resource Set according to preferences, rules
  – Can animate individual Resources in Set (state, picture) — unlike multi-Capacity single Resources

• Also group Sequences and entity pictures into sets for ease of access (via part-type number)
Modeling Approach

- There are usually several different ways to represent a model in software
- Often driven by data requirements, availability
- Decide on and write down ahead of time how things will be represented; here:
  - Sequences (in a Set) to control part flow
  - Assign entities in Sequences for process times except for Cell 1
  - Use Expression vector for process times at Cell 1
  - Use Variables for transfer times, speed factor at Cell 3
Building the Model

• Place the data modules (Sequences, Expressions, Variables, Sets, Simulate)
  – Defines names, initial values
  – Makes names available on pull-down lists for reference later while editing logic modules

• Place the logic modules, making use of names previously defined in data modules

• Animation — will import AutoCAD drawing

• Verification of expression of model in software

• Statistical design and analysis
Sequences Module

- Define station paths for different part types
- Name the sequences
  - Part 1 Process Plan, etc.
- Can define actions at each station as part of the definition of the sequences
  - Define Process Time attribute (except at Cell 1)
- Group the sequences into a Set (later)
- Assign a sequence, indexed by set, to each arriving part once its type is generated
- Don’t forget to define “exit” for each sequence
- Good (but not the only) way to model this
Expressions Module

• Many potential uses, but here to define process times at Cell 1 for different part types
• Name each expression: Cell 1 Times
• Scalar, vector, or 2-dimensional array
  – Use 3 x 1 vector here, one for each part type
  – Must index in correct order — here, by part type
• Can be very general expression
  – Here, process-time distributions at Cell 1
  – Entities will index into vector by their part type: Cell 1 Times(Part Index)
• Again, this is not the only way to model this
Variables Module

• Define global variables, their initial values
  – Like Expressions, these are global to the model
  – Unlike Expressions, they can only take on numerical values, not formulas or other variables, expressions, or attributes (but the numerical value can be changed during the run by any entity)
  – Here, use for two different variables
  – Speed factor at Cell 3: 2 x 1 vector Factor
    • 0.8 for index 1 (new machine), 1.0 for index 2 (old)
  – Value (2) of part transfer times: Transfer Time

• (Yet) again, this is not the only way to do this
Sets Module

• Group together (and name and index) “like” things — Resources, Queues, Storages, Stations, Pictures, Counters, Tallies, and “Other” (e.g., Sequences)

• Can make models more general, “cleaner”
  – In some cases, needed to model things properly, like workers (Resources) with different but overlapping skills

• Order of entry of set members is critical!

• Here, we’ll form sets of Resources, Sequences (“Other”), Pictures, and Tallies
Sets Module (cont’d.)

• **Resources set**
  – For the two different machines at Cell 3
  – Name the set **Cell 3 Machines**
  – Use pull-down lists where possible to define things
    • Depends on order in which you build, fill out model
  – Use: In Process module for Cell 3 (later …)

• **Sequences (“Other”) set**
  – For generality of number of part types
  – Name the set **Part Sequences**
  – No pull-down here in “Other” so have to remember names given in Sequences module
  – Use: In Arrive module (later …)
Sets Module (cont’d.)

• **Pictures set**
  - To show the part types differently in animation
  - Name the set *Parts*, the entries *Part 1*, etc.
  - Draw the pictures in Simulate module (later …)
  - Use: In Arrive module (later …)

• **Tallies set**
  - To separate out cycle times by part type
  - Name the set *Part Cycle Times*
  - Members are Tally names to appear in output
  - Use: In Depart module (later …)
Simulate Module

- Identify model and define Length of Replication, as in earlier models
- Also, draw the three entity Pictures and associate with names of members of Pictures set
  - Double-click on “generic” entity picture above module handle
  - Add new or edit existing pictures in list
    - Copy current last entry to add to bottom of list
    - Draw: shapes, size; colors for fill, line, text, etc.
    - Close drawing window to accept
  - Names (Values) must already be assigned in Sets
Arrive Module

**Main dialog**

- Station name: **Order Release**
- Time Between: **EXPO (13)**
- Mark Time Attribute: **Enter Time**
- Leave Data
  - Route: select (this is the default anyway)
  - Seq: select
  - Route Time: **Transfer Time**
• **Assign... subdialog**
  - Attribute: \textit{Part Index} = DISC(...) (\textit{cumulatives})
  - Other: Sequence = Sequence Set(Part Index)

• **Animate... subdialog**
  - Initial Entity Picture
    - Set Member: \textit{select}
    - Picture Set: \textit{Parts}
    - Set Index: \textit{Part Index}
Server Modules for Cells 1, 2, 4

• **Main dialog**
  - Station: **Cell 1** (or **Cell 2** or **Cell 4**)
  - Process Time:
    • If Cell 1: **Cell 1 Times (Part Index)**
      - Global Expression defined in Expressions module
    • If Cell 2 or 4: **Process Time**
      - Entity Attribute defined as part of Sequences
  - Leave Data:
    • Route: *select*
    • Seq: *select*
    • Route Time: **Transfer Time**
      - Global Variable defined in Variables module
Digression: Data Structures

• Why an Expression for processing times at Cell 1 rather than entity Attribute assigned in Sequences as for the other cells?
  – Frank answer: Just to show the use of Expression
  – Could easily have treated Cell 1 like the others

• Conversely, could have used Expression for processing times at Cells 3 and 4
  – Problem with Cell 2: Part 2 visits it twice with different processing-time distributions, so would have to indicate which visit somehow
  – Moreover, this is a very small model

• Moral: Think carefully about data structure!
Cell 3: Enter/Process/Leave

- Can’t use Server module — two different machines

- Two machines grouped into Resource Set called Cell 3 Machines
  - Server module cannot access into a Resource Set

- Modeling options
  - Advanced Server module
  - Enter → Process → Leave sequence
  - Identical in this case (but not always)
  - We’ll take the latter since it’s more general
    - Allows for tandem sequence of operations
Cell 3: Enter/Process/Leave (cont’d.)

- Enter module: generic “landing pad” for entities
  - Station: select
  - Station: Cell 3
  - Queue: Storage for unload time (can delete here)

- Process module: describe processing delay
  - More general than Server Data area of Server module — can reference into a Resource Set
  - Activity is to Seize a member of a Resource Set
  - Resource Set: Cell 3 Machines
  - Rule: Cyclical (cycle around available members)
  - Store Index in Att: Index (no. of chosen member)
  - Process Time: Process Time * Factor(Index)

0.8 for Index = 1 (new)
1.0 for Index = 2 (old)
Cell 3: Enter/Process/Leave (cont’d.)

• Leave module: generic “launching pad”
  – Must specify Station from which entity is leaving
  – From Station: Cell 3
  – Seq: check
  – Route Time: Transfer Time

• Must Connect Enter - Process - Leave
  – Could also use Labels but wouldn’t be visible

• Due to its position in the layout, want to re-orient these modules for right-to-left flow
  – Arrange the modules physically that way
  – Move entry (box) and exit (triangle) points
    • Flip triangle about vertical line to reorient
Resource Definitions

- Must define Cell 3 machine resources explicitly
- Resource modules
  - Separate ones for old, new machine
  - Need only define Name, Type, Capacity for each
    - Names: Cell 3 New, Cell 3 Old
    - Both are Capacity Type with Capacity = 1
  - Both have animation picture, but won’t be used (will use Enter module station) so can be deleted
  - Both have Individual Queue, but won’t be used (will use queue on Process module) so can be deleted
Parts’ Exit

• Use Depart module, as before

• Need to collect cycle times by part type
  – Station name: `Exit System`
  – Tally Set Member: `select`
  – Tally Set: `Part Cycle Time`
  – Set Index: `Part Index`
  – Type of Statistics: Interval (`select`)
    • Attribute: `Enter Time`
  – Will produce three different Tally reports, one for each part type
Animation

• Could run model at this point, would get correct numerical summary results
  – But part movement would not show in animation
• Pull animation away from logic, data modules
• Move, resize, reorient queues for realism
• Animate Routes (all movement possibilities)
  – Thick “bundles” of routes — Shift key, Snap to Grid
  – Heed clockwise direction
  – Draw lines to define route “lanes”
• Import AutoCAD dxf file for backdrop (see text)
• Fine-tune resource pictures
  – Layers for seize point
Verification

- System $\rightarrow$ Model $\rightarrow$ “Code”
- **Validation**: Is Model = System?
- **Verification**: Is “Code” = Model? (debugging)
- The Truth: Can probably never completely verify, especially for large models
Verification (cont’d.)

• Some techniques to attempt verification
  – Eliminate error messages (obviously)
  – Single entity release, Step through logic
    • Set Max Batches = 1 in Arrive
    • Replace part-type distribution with a constant
  – “Stress” model under extreme conditions
  – Performance estimation — like slide-rule decimal placement
  – Look at generated SIMAN .mod and .exp files
    • Run/SIMAN/View menu option
Statistical Analysis of Output Data: Terminating Simulations

- Random input leads to random output (RIRO)
- Run a simulation (once) — what does it mean?
  - Was this run “typical” or not?
  - Variability from run to run (of the same model)?
- Need statistical analysis of output data
- Time frame of simulations
  - Terminating: Specific starting, stopping conditions
  - Steady-state: Long-run (technically forever)
  - Here: Terminating
Modify Model 6.1 into Model 6.2

• Establish a realistic termination rule
  – There are many different ways to terminate
  – Really a modeling issue — what’s realistic?

• Process an incoming order of 100 parts
  – Set Max Batches = 100 in Arrive
  – Shuts off Arrival stream after 100 entities created
  – *In this model*, will cause termination when 100th part exits
    since there are no other events and the event calendar will
    become empty
  – Causes system to become less congested when the end of
    the simulation is approached
Modify Model 6.1 into Model 6.2
(cont’d.)

• Establish a single overall performance measure: Work in Process (WIP)
  – Variable WIP defined (Variables module)
  – Click WIP up 1 just after each arrival
    • Arrive module, Assign Variable WIP = WIP + 1
  – Click WIP down 1 just before each departure
    • No Assign subdialog in Depart module
    • Put Actions module just before Depart to click down
    • Re-wire Sequences, Routes to send entities here instead of to Exit System Depart module
    • Direct Connect to Exit System Depart module
  – Add Statistics module to request collection and reporting of Time-Persistent stats on WIP
Strategy for Data Collection and Analysis

• For terminating case, make IID replications
  – Simulate module: Number of Replications field
  – Check both boxes for Initialization Between Reps.
  – Get multiple independent Summary Reports

• How many replications?
  – Trial and error (now)
  – Approximate no. for acceptable precision (below)
  – Sequential sampling (Chapter 11)

• Save summary statistics across replications
  – Statistics Module, Outputs Area, save to files

• Maybe turn off animation (Run/Setup/Mode)
Confidence Intervals for Terminating Systems

• Output Analyzer on files saved from Outputs area (cross-replication) of Statistics module

• Define, read in, save Data Group(s)

• In Output Analyzer
  – *Analyze/Conf. Interval on Mean/Classical…* menu (or )
  – Add desired files; select Lumped for Replications
Confidence Interval Dialogs

Add files to Data Group
Select files for confidence intervals
Can change confidence level (95% is default)
Select “Lumped” Replications treatment to use all replications
Confidence Interval Results

- Interpretation of confidence intervals
  - What’s being estimated
  - Coverage, precision, robustness to non-normality
Automatic Text-Only 95% Confidence Intervals

At end of summary report, get 95% confidence intervals as above, in text-only format, if

- You ask for more than one replication, and
- You have a Statistics module with Outputs entries

Done only for output measures in Statistics module’s Outputs area

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Average</th>
<th>Half-width</th>
<th>Minimum</th>
<th>Maximum</th>
<th># Replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg WIP</td>
<td>11.327</td>
<td>1.7801</td>
<td>6.3666</td>
<td>18.704</td>
<td>20</td>
</tr>
<tr>
<td>Part 1 cycle avg</td>
<td>148.38</td>
<td>23.809</td>
<td>93.763</td>
<td>296.79</td>
<td>20</td>
</tr>
<tr>
<td>Part 2 cycle avg</td>
<td>189.47</td>
<td>25.602</td>
<td>124.38</td>
<td>296.29</td>
<td>20</td>
</tr>
<tr>
<td>Cell 4 avg Q length</td>
<td>1.5652</td>
<td>.64270</td>
<td>.48813</td>
<td>6.4610</td>
<td>20</td>
</tr>
<tr>
<td>Cell 2 avg Q length</td>
<td>1.7943</td>
<td>.66541</td>
<td>.45236</td>
<td>6.3355</td>
<td>20</td>
</tr>
<tr>
<td>Cell 1 avg Q length</td>
<td>1.7960</td>
<td>.52429</td>
<td>.45934</td>
<td>4.5093</td>
<td>20</td>
</tr>
<tr>
<td>Part 3 cycle avg</td>
<td>107.80</td>
<td>15.704</td>
<td>65.692</td>
<td>193.10</td>
<td>20</td>
</tr>
<tr>
<td>Cell 3 avg Q length</td>
<td>1.2976</td>
<td>.40001</td>
<td>.45141</td>
<td>3.3876</td>
<td>20</td>
</tr>
</tbody>
</table>
Half Width and Number of Replications

- Prefer smaller confidence intervals — *precision*
- **Notation:** \( n = \) no. replications
  \( \bar{x} = \) sample mean
  \( s = \) sample standard deviation
  \( t_{n-1,1-\alpha/2} = \) critical value from \( t \) tables

- **Confidence interval:**
  \[ \bar{x} \pm t_{n-1,1-\alpha/2} \frac{s}{\sqrt{n}} \]

- **Half-width** = \( t_{n-1,1-\alpha/2} \frac{s}{\sqrt{n}} \)
  - Want this to be “small,” say \(< h \) where \( h \) is prespecified

- Can’t control \( t \) or \( s \)
- Must increase \( n \) — how much?

*Simulation with Arena — Chapter 6 — Intermediate Modeling and Terminating Statistical Analysis*
Half Width and Number of Replications (cont’d.)

- Set half-width = \( h \), solve for
  \[ n = t_{n-1,1-\alpha/2}^2 \frac{s^2}{h^2} \]
- Not really solved for \( n \) (\( t, s \) depend on \( n \))
- Approximation:
  - Replace \( t \) by \( z \), corresponding normal critical value
  - Pretend that current \( s \) will hold for larger samples
  - Get
    \[ n \approx Z_{1-\alpha/2}^2 \frac{s^2}{h^2} \]
    \( s = \) sample standard deviation from “initial” number \( n_0 \) of replications
    \( n \) grows quadratically as \( h \) decreases.

- Easier but different approximation:
  \[ n \approx n_0 \frac{h_0^2}{h^2} \]
  \( h_0 = \) half width from “initial” number \( n_0 \) of replications
Comparing Alternatives

• Usually, want to compare alternative system configurations, layouts, scenarios, sensitivity analysis …

• Here: Transfer time (2 min) smells like a guess — does it matter if it’s, say, 1 vs. 3?
  – Call these alternatives A and B in Arena

• **Single measure of performance:** average WIP

• Make two sets of 20 replications, for A and B
  – Must rename output files to distinguish them
Comparing Alternatives (cont’d.)

- **Analyze/Compare Means** menu (no button)
  - Add comparable data files for A and B
  - Lumped Replications

![Image of the Analyze/Compare Means menu with options to add data files and select confidence levels and test types.]
Comparing Alternatives (cont’d.)

• Results:

```
<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>ESTD. MEAN</th>
<th>STANDARD</th>
<th>0.95 C.I.</th>
<th>MINIMUM</th>
<th>MAXIMUM</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg WIP</td>
<td>-2.5</td>
<td>3.64</td>
<td>1.6</td>
<td>5.36</td>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-4.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-3.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.857</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

• c.i. on difference misses 0, so conclude that there is a (statistically) significant difference