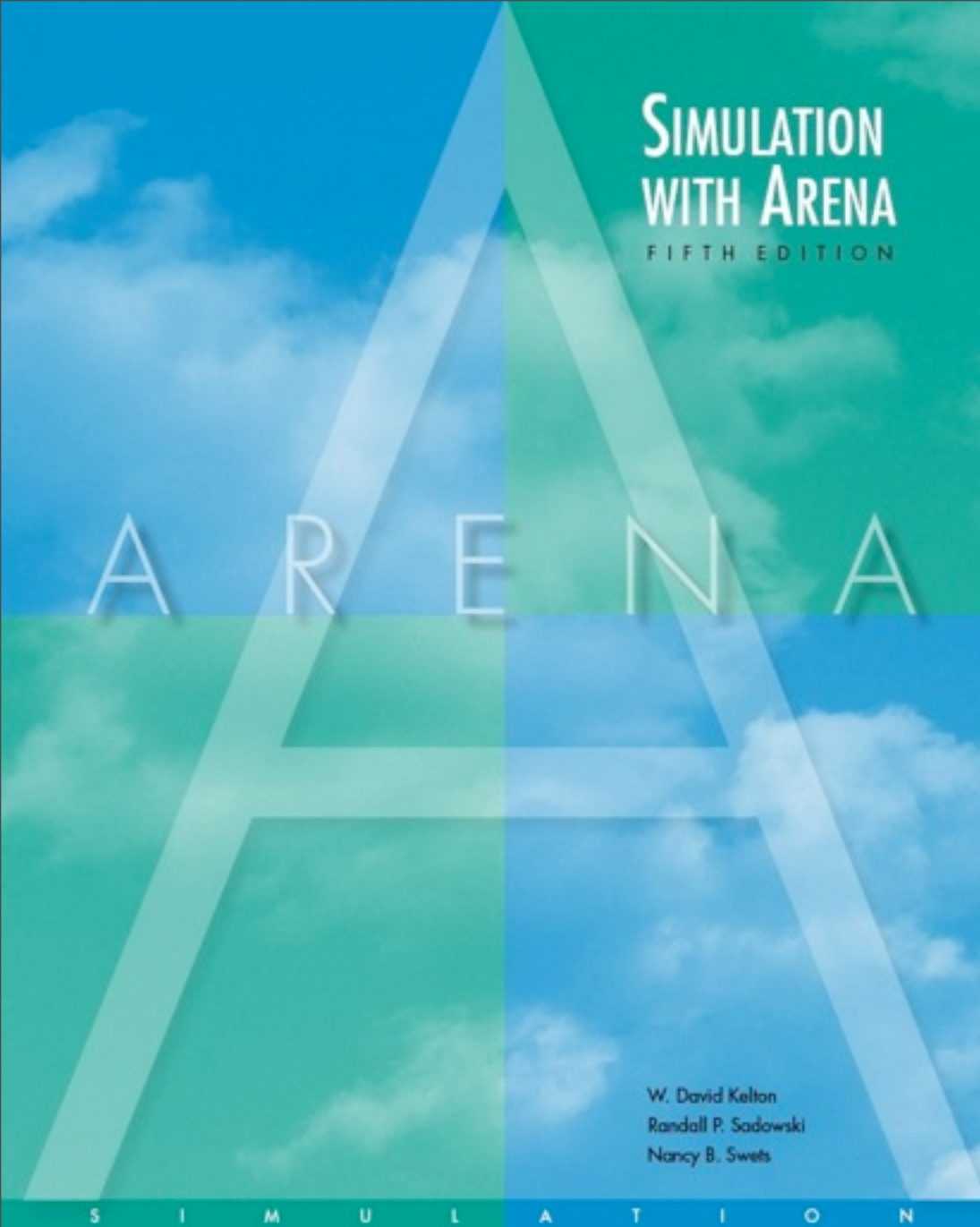




Modeling Detailed Operations

Chapter 5

Last revision August 20, 2009

What We'll Do ...

- **Model 5-1: Simple call center**
 - Lower-level modeling, Advanced Process panel
 - Three-way decisions, Variables, Expressions, Storages
 - Blocks panel
 - Terminating vs. steady-state operation
 - Logical (“fake”) entities
 - Terminating Condition in *Run > Setup*
- **Model 5-2: Enhanced call center**
 - Nonstationary Poisson arrival process
 - Sets – Resource, Counter
 - New Statistic data module Types
 - Counter, Time Persistent

What We'll Do ... (cont'd.)

- **Model 5-3: Enhanced call center with more output performance measures**
 - New Statistic data module Type
 - Output
 - Additional variable resources – look at staffing levels
- **Model 5-4: (s, S) inventory**
 - Not queueing
 - Choose to use low-level Blocks, Elements panels (SIMAN)
 - Can be done with higher-level panels

Model 5-1: Simple Call Center Setup

- **One phone number for customers to call in to**
 - 26 trunk lines, one needed for each call (incoming or outgoing, talking or on hold)
 - Arriving call finding no free trunk lines gets busy signal, goes away
 - Count number of such rejected calls
- **Calls arrive with interarrivals ~ EXPO (0.857) min.**
 - First call arrives at time 0
- **Three incoming call types**
 - Initial recording to decide ~ UNIF (0.1, 0.6) min.
 - Tech support (76%), sales (16%), order status (8%)

Model 5-1: Simple Call Center Setup (cont'd.)

- **Tech-support calls**
 - For product type 1 (25%), 2 (34%), or 3 (41%)
 - Recording/select time ~ UNIF (0.1, 0.5)
 - Needs qualified tech-support person
 - Two for type 1, three for type 2, three for type 3
No crossover to another type ... will allow this in Model 5-2
 - Separate FIFO queues for each type
 - Conversation time ~ TRIA (3, 6, 18) min. for all types
 - Then leaves system
- **Sales calls**
 - All the same
 - Four sales staff, all the same
 - One FIFO queue feeding all sales staff
 - Conversation time ~ TRIA (4, 15, 45)
 - Then leaves system

Model 5-1: Simple Call Center Setup (cont'd.)

- **Order-status calls**
 - All the same
 - Handled automatically by phone system
 - No limit on number in process at a time, except for trunk-line limit
 - “Conversation” time ~ TRIA (2, 3, 4)
 - After “conversation,” 15% of callers opt to talk to a person
 - Routed to sales staff, conversation lasts an additional TRIA (2, 3, 4)
 - Sales calls have higher priority (non-preemptive)
- **Center receives calls 8am – 6pm**
 - Must terminate arrival process at 6pm
 - Operate past 6pm if necessary to “flush out” all calls

Model 5-1: Simple Call Center Setup (cont'd.)

- **Output performance measures**
 - Number of calls attempted, rejected, completed
 - By call type – total time in system
 - By resource – time on hold, number of calls on hold
 - Resource utilization – of personnel, trunk lines
- **Terminating or steady-state**
 - Time frame of interest for each replication
 - Terminating – specific starting, stopping conditions (this model)
Stopping conditions could be of several forms – fixed time, count, condition (here)
 - Steady-state – output performance measures are a limit as simulated time $\rightarrow \infty$
 - Choice usually depends on intent of study, not on model logic

Model 5-1: Simple Call Center Modeling Panels

- **Basic Process**
 - Highest, fastest modeling level, usually the place to start
- **Advanced Process**
 - Smaller building elements, other functions, more detail
- **Advanced Transfer**
 - Entity movement, material handling
- **Blocks, Elements**
 - Lowest modeling level, SIMAN simulation *language*
 - Repeats some capabilities of higher-level panels
 - Some functions available only here
- **Other special-purpose panels**
 - License-dependent

Model 5-1: Simple Call Center Data Structure

- **Re-use data in several places**
 - Define once, global to whole model
 - Redefine once – modeling generality, user efficiency
- **Arena (global) *Variables***
 - Store numbers (not formulas)
 - Define, initialize in Variable data module (Basic Process)
 - Can change during run (Assign module, other ways)
 - Scalar, 1-d array (vector), 2-d array (matrix)
- **Arena (global) *Expressions***
 - Store formulas (as well as numbers, but can't change)
 - Use math ops, numbers, random variates, Attributes, Variables, ...
 - Define in Expression data module (Advanced Process)
 - Scalar, 1-d array (vector), 2-d array (matrix)

Model 5-1: Simple Call Center Arrivals, Direct to Service

- **Create attempted calls**
 - Entity type **Incoming Call**, change later
 - Max Arrivals = **MaxCalls**, Variable initialized to 999999
 - At 6pm (time 600 minutes) change this to 1 to stop arrivals ... later
 - Entities per Arrival = **CallsPerArrival**, Variable initialized to 1
 - At 6pm (time 600 minutes) change this to 0 to kill arrivals ... later
- **Entity data module**
 - **Incoming Call** Entity Type already there
 - For Initial Picture, select **Picture.Black Ball**
- **Record module for an attempted call**
 - Add 1 to Counter Name **Attempted Calls**
 - Results – Category Overview report, User Specified

More detailed description – mouse over modules, read Data Tips that pop up

Model 5-1: Simple Call Center Arrivals, Direct to Service (cont'd.)

• Decide module – Trunk Line Available?

- Type = 2-way by Condition
 - Select (logical) Expression for “If”
 - NR () is number of units of that resource that are busy now
 - MR () is number of units of that resource that exist now
- False – Record rejected call counter, Dispose
- True:
 - Seize a unit of **Trunk Line** Resource – Release later
 - Resources data module for Trunk Line and other Resource levels
 - Increment Variable **Total WIP** for number of active calls
 - Used in stopping rule at or after 6pm to sense if system is empty
 - Store module to animate entity during next Delay module
 - Add Storage animation separately, identify with this logical storage by name
 - Storage data module – entry made there by Store module
 - Delay module to listen to initial recording, make selection
 - Could have used Process module, but this is simpler, faster
 - Unstore module to make entity animation disappear

Alternate strategy – Queue module from Blocks panel ... details in text

Model 5-1: Simple Call Center Arrivals, Direct to Service (cont'd.)

- **Decide module – Determine Call Type**
 - Three-sided coin flip – Type = N-way by Chance
 - Add button for more sides of coin
 - Get new exit point for each Add, plus one for Else
 - Note that probabilities are entered as *percentages* (0-100, not 0-1)
 - Last entry is “else”
- **Direct call to one of tech support, sales, or order-status areas**

*Backed each area with colored box
Alternative way to organize – Submodels*

Model 5-1: Simple Call Center

Tech-Support Calls

- **Assign module**
 - Change Entity Type for separating out in results
 - Change Entity Picture for animation
- **Store – Delay – Unstore for recording, product type selection**
- **Decide module for product type**
 - Different three-sided coin flip
 - Direct to appropriate Process module for that product type
- **Process modules for tech-support service**
 - Seize-Delay-Release
 - Seize a unit from appropriate multi-unit Resource
 - Use **Tech Time** defined in Expression data module
- **Proceed to system exit logic ... later**

Model 5-1: Simple Call Center Sales Calls

- **Assign module – change Entity Type, Picture**
- **Process module**
 - Seize-Delay-Release
 - Seize a unit of **Sales** Resource
- **Sales calls priority over order-status calls that seek a person?**
 - Queue data module, **Process Sales Call.Queue**
 - Type = Lowest Attribute Value
Attribute Name = **Sales Call Priority**
Undefined for sales calls, so has value 0 ... will set to 1 for order-status calls that seek a person, putting sales calls ahead in the queue
 - Shared queue (with order-status calls seeking a person)
- **Proceed to system-exit logic**

Not the only way to do this

Model 5-1: Simple Call Center

Order-Status Calls

- **Assign module – change Entity Type, Picture**
- **Delay block (Blocks panel) for robo-chat**
 - Includes Store/Unstore logic – alternative to earlier method
 - No automatic entry in Storage data module, so must enter manually
- **Decide module**
 - No sales person required – go directly to system-exit logic
 - Sales person required:
 - Assign module – set **Sales Call Priority** Attribute to 1 so these will have lower priority than real sales calls
 - Seize module for a unit of **Sales** resource
 - Define Queue Name = **Process Sales Call.Queue** – shared with sales calls
 - Process module does not allow for specifying a shared queue, so can't use here
 - Delay for conversation with sales person
 - Release the unit of **Sales** resource
- **Proceed to system-exit logic**

Model 5-1: Simple Call Center System Exit

- All calls of all types come here when finished
- Release module – release the unit of Trunk Line resource seized upstream
- Assign module – decrement Total WIP variable
- Record module – increment Completed Calls counter
- Dispose of call

Model 5-1: Simple Call Center

Arrival-Cutoff Logic

- **Used to “choke off” arrival stream at 6pm**
- **Create a single “logical” entity at time 600 min. (6pm)**
 - Overkill on making sure just one is created
 - Time Between Arrivals = 999999 min., Max Arrivals = 1
- **Assign module to set Variable `MaxCalls` to 1**
 - Recall use of `MaxCalls` for Max Arrivals in Create module for attempted calls
- **Also in this Assign module, set `CallsPerArrival` to 0**
 - Since Create module will *always* schedule next arrival, and this makes the “size” of the next illegal arrival zero
- **Dispose of this single logical entity**

Creative use of such “logical” (a.k.a. “fake”) entities enhances modeling flexibility, power, detail

Model 5-1: Simple Call Center

Run > Setup

- Replication Parameters tab (other tabs as usual)
- Base Time Units = Minutes
- Replication Length = Infinite (the default)
- Terminating Condition field:

TNOW >= 600.0 && Total WIP == 0

Arena clock Variable *Greater than or equal to 600 minutes, (6pm)* *Logical "and"* *Variable we maintained in model* *Equality test for zero*

Base Time Units

It's 6pm or later *and* *there are no calls in the system.*

Could have used NR (Trunk Line) instead of Total WIP

Model 5-1: Simple Call Center Animation

- **Place three Storage animations**
 - Initial Recording Delay, Tech Call Recording Delay, Order Status Delay
 - Select proper Identifier in each from pull-down list
 - Graphic behaves like Queue animations
- **Four Queue animations**
 - Three tech-support call product types, sales
 - Came with four Process modules specifying Seize
- **Resource animations for three tech-support types, sales Resources**
 - Multi-unit Resource animations, as in Models 4-3, 4-4

Model 5-1: Simple Call Center Animation (cont'd.)

- **Variable animations for WIP at tech calls, sales**
 - For tech calls, Arena variable to animate is **Process Product Type 1 Tech Call.WIP, etc.** – pull-down list
 - For sales calls, must include order-status calls seeking a real person:
$$NR(Sales) + NQ(Process\ Sales\ Call.Queue)$$
- **Plot number of trunk lines busy,**
 $NR(Trunk\ Line)$
- **Labeling, background boxes as in model logic**

Model 5-1: Simple Call Center Results

(one replication ... sample of size only one!!)

- **Trunk-lines-busy plot**
 - Starts, ends at 0 – startup, termination logic
 - Capped at 26 during run
- **734 attempted calls (User Specified section)**
 - 643 completed, other 91 rejected
- **Sometimes see mixture of sales (green), order-status (blue) entities in sales queue**
- **Other “usual” outputs**
 - Times in system – separated out by call type
 - Queue lengths, times in queue – separated out by resource
 - Resource utilizations – normalized to [0, 1] by capacity

Model 5-2: Enhanced Call Center Changes

- Incoming calls' arrival rate varies over day
 - Probabilistic model – *Nonstationary Poisson process*
 - More in Section 12.3
 - Instead of a constant rate ($= 1 / \text{mean interarrival time}$), specify a rate *function*
 - Arena supports *piecewise-constant* rate function – “step” functions
 - Easy to specify, strong theoretical support
 - Rate-function specification:

In Arena, rates MUST be in arrivals per HOUR, regardless of base time units or time intervals

Table 5-2. Call Arrival Rates (Calls Per Hour)

Time	Rate	Time	Rate	Time	Rate	Time	Rate
8:00 - 8:30	20	10:30 - 11:00	75	1:00 - 1:30	110	3:30 - 4:00	90
8:30 - 9:00	35	11:00 - 11:30	75	1:30 - 2:00	95	4:00 - 4:30	70
9:00 - 9:30	45	11:30 - 12:00	90	2:00 - 2:30	105	4:30 - 5:00	65
9:30 - 10:00	50	12:00 - 12:30	95	2:30 - 3:00	90	5:00 - 5:30	45
10:00 - 10:30	70	12:30 - 1:00	105	3:00 - 3:30	85	5:30 - 6:00	30

Caution – it's easy to generate this incorrectly ... see text for details

Model 5-2: Enhanced Call Center Changes (cont'd.)

- **Sales-staff size varies over day**
 - Data in text, Schedule data module, Sales Schedule
- **Tech-support staff are partially cross-trained, work complicated schedule:**

Name	Product Lines	Time Period (30 minutes)																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Charity	1	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●					
Noah	1						●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●
Molly	1, 3			●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●			
Anna	1, 2, 3					●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	
Sammy	1, 2, 3				●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●		
Tierney	2	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●					
Aidan	2						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●
Emma	2				●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●		
Mya	3	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●					
Ian	3						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●
Christie	3				●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●		

Will use Arena Sets concept to implement this cross training

Model 5-2: Enhanced Call Center Changes (cont'd.)

- **4% of tech-support calls cannot be handled during the call, need offline back-office research**
 - Original call ends, same original talk-time distribution, gives up its trunk line, but not counted (yet) as completed
 - Case sent to back office (outside model boundaries), takes EXPO (60) minutes to resolve
 - Offline research may be carried over night, completed on a later day
 - Answer goes back to *same* tech-support person who took original call, with higher priority than incoming calls, but still might have to queue for this person
 - This tech-support person requests a trunk line for outgoing call, higher priority than incoming calls, but still might have to queue, talks for TRIA (2 ,4 ,9) min., call is *now* completed
 - Track number of each product type after research is done

Model 5-2: Enhanced Call Center Data Structure

- **Resources, Schedules**
 - Resource, Schedule data modules
 - Trunk Line – fixed capacity at 26
 - Sales – on Schedule **Sales Schedule**
 - 11 individual tech-support people on individual schedules
 - Caution – must fill out each schedule to all 22 half-hour periods, with leading/trailing 0's if necessary ... use Edit via Dialog or Spreadsheet, not graphical schedule editor
 - **Ignore** option to avoid shifting back schedule over multiple days
 - Include costing data for people in Resource data module
 - Define nonstationary arrival-rate function in Schedule module – **Arrival Schedule**
 - Enter trailing 0's in Edit via Dialog or Spreadsheet, not graphical schedule editor

Model 5-2: Enhanced Call Center Data Structure (cont'd.)

- **Sets – collect same-type items together**
 - Set, Advanced Set data modules (Basic, Advanced Process panels, resp.)
 - Refer to items in set by original name, or index (subscript) in set
 - Resource set for each tech-support product type
 - Members are those tech-support resources qualified
 - Individual resources already defined – Resource data module
 - Overlapping membership – some resources in multiple sets
 - Sets are ordered – here, put most versatile tech-support people at bottom, to “save” them for other calls ... Preferred Order in Seize
 - Will Seize from a set in model
 - Counter set – one for each hour
 - Count number of rejected calls in each hour
 - Individual counters already defined – Statistic data module
 - Use results later to decide when to increase staffing

Model 5-2: Enhanced Call Center

Modifying the Model

- **Call-arrivals, termination, Run > Setup**
 - Create module
 - Type = Schedule, Schedule Name = **Arrival Schedule**
 - Delete the entire arrival-cutoff section from Model 5-1
 - **Arrival Schedule** cuts off arrivals at 6pm, via 0 rate
 - Delete **Total WIP** variable used to terminate Model 5-1
 - Use built-in NR(**Trunk Line**) instead in Terminating Condition
 - Delete Assign modules used to manage **Total WIP**
 - Record module for rejected calls
 - Index into Counter Set **Rejected Calls** with index
$$\text{AINT} ((\text{TNOW}/60) + 1)$$
which is 1 for first hour, 2 for second hour, etc. (AINT truncates decimals toward zero)

Model 5-2: Enhanced Call Center

Modifying the Model (cont'd.)

- **Tech-support calls**
 - Same through **Determine Product Type** Decide
 - Add Assign modules for each product type thereafter
 - Entity Type to distinguish product type in reports
 - Entity Picture to distinguish product type in animation
 - Attribute **Tech Call Type** (1, 2, or 3 by product type) for routing
 - Process modules, Resources subdialogs
 - Type = Set
 - Set Name = Product **1**, **etc.**
 - Selection Rule = Preferred Order, to select earlier entries in set first
Recall – we put more versatile tech-support people lower in the set list
 - Save Attribute = **Tech Agent Index**
Entity attribute, carried along, in case of back-office research to send back to this same tech-support person for return call

Model 5-2: Enhanced Call Center

Modifying the Model (cont'd.)

- **Back office, returned tech-support calls – all new**
 - Entry via True branch (4%) in Decide module
Backoffice Research and Return Call?
 - Release this call's trunk line – going offline now
 - Delay (with storage) for EXPO (60) back-office research
 - Increment **Tech Return WIP(Tech Call Type)**
 - 1-dim. Variable array – defined in Variable data module
 - **Tech Call Type** is 1, 2, or 3, assigned in earlier Assign module
 - Decide module **Product Type?** based on Entity Type
 - Seize the same tech-support person – higher priority
 - *Then* seize a trunk line (higher priority), make return call
 - Then release this trunk line, tech-support person
 - Decrement **Tech Return WIP(Tech Call Type)**
 - Send entity to final Record, *after* trunk-line release there

Model 5-2: Enhanced Call Center

Modifying the Model (cont'd.)

- **Statistic data module**

- Ten Counter-type statistics, discussed earlier
- Four Time-Persistent statistics to track expressions
 - **Backoffice Research WIP** to track total number of cases in research, via **NSTO(Backoffice Research Storage)**
 - **Tech 1 Total Online WIP Stat, etc.**, to track number of that product type in back office via Expression **Tech 1 Total Online WIP, etc.**, defined in Expression data module as

Process Product Type 1 Tech Call.WIP + Tech Return WIP(1), etc.

- **No changes needed in sales-calls or order-status-calls section of Model 5-1**

Model 5-2: Enhanced Call Center

Modifying the Model (cont'd.)

- **Animation**

- Delete **Tech 1**, **Tech 2**, and **Tech 3** resource animations
- Change variables in three tech-support WIP displays to track total number of tech-support calls of that type present
- New back-office storage animation, variable animation for number present
- A new queue for each tech-support product type for return calls waiting for service
- Added a resource animation (from a .plb library) for each individual tech-support person
 - Grouped by product type, colors for capabilities

- **Results**

- Most rejected calls in hours 5-8 ... increase staff then ... ?

Model 5-3: Overall Call-Center Stats Setup

- **Develop overall operational-cost measure**
 - Two cost categories – staffing/resource, and poor service
- **Develop overall measure of service, % of calls rejected**
- **Add options for increased staffing, improvement**
- **Make 5 replications, focus on weekly costs**
 - IID replications, so will not carry over back-office research

Model 5-3: Overall Call-Center Stats

Staffing/Resource Costs

- **Resource data module – hourly costs for people**
 - \$20/hr. for each sales staffer
 - \$18/hr. – \$20/hr. for each tech-support, depending on skill
 - These salary costs paid when on duty, busy or idle
 - Summing, get \$12,820/week (details in text)
 - View all this existing staff as fixed

Model 5-3: Overall Call-Center Stats

Staffing/Resource Costs (cont'd.)

- **Increase sales, tech-support staff noon-4pm**
 - Variable **New Sales** = number of new sales staff
 - \$17/hr., 4 hrs./day, 5 days/week, so \$340/week for each add'l. staff
 - Schedule data module to add capacity – edit via dialog or spreadsheet, not graphical editor
 - Resource (**Sales**) already exists in Resource data module
 - Variables **New Tech 1, etc.**, and **New Tech All** for number of new tech-support people qualified as named
 - \$16/hr. for each one-product staff, \$18/hr. for each all-product staff
\$320/week for each single-product staff, \$360/week for each all-product staff
 - New entries in Resource data module
Larry, Moe, Curly, Hermann for 1, 2, 3, All, resp.
 - Schedule data module to add capacity – dialog or spreadsheet edit

Model 5-3: Overall Call-Center Stats

Staffing/Resource Costs (cont'd.)

- **Maybe increase number of trunk lines beyond 26**
 - \$98/week flat fee for each trunk line
- **Define Expression New Res Cost for all resource costs:**

New Sales*340

+ (New Tech 1 + New Tech 2 + New Tech 3)*320

+ New Tech All*360

+ 98*MR(Trunk Line)

- This does not depend on simulation results, only on setup

Model 5-3: Overall Call-Center Stats

Customer-Dissatisfaction Costs

- **Incur cost for caller wait on hold, past a threshold**
 - 3 min. for tech, 1 min. for sales, 2 min. for order-status
 - Beyond threshold, incur per-min. costs of \$0.368 for tech, \$0.818 for sales, \$0.346 for order-status
 - In practice, such costs are difficult to estimate
 - Three new Assign modules (orange backing) accumulate “excess” (beyond threshold) wait times on hold
 - Tech support (other two are similar): Variable **Excess Tech Wait Time** increased by **MAX(ENTITY.WAITTIME - 3, 0)**
ENTITY.WAITTIME is built-in Arena attribute holding all wait times (including in queues) so far ... luckily, there were none before the preceding Process module
 - At end, multiply excess wait times by per-min. costs, multiplied by 5 (to put on a weekly basis)
 - $5 \times \$0.368 = \1.84 for tech, $5 \times \$0.818 = \4.09 for sales,
 - $5 \times \$0.346 = \1.73 for order-status

Model 5-3: Overall Call-Center Stats

Overall Output Performance Measures

- **Statistic data module, Total Cost entry**

- Type = Output, computed only at end of replication

New Res Cost

+ Excess Sales Wait Time * 4.09
+ Excess Status Wait Time * 1.73
+ Excess Tech Wait Time * 1.84
+ 12820

- **Statistic data module, Percent Rejected entry**

- Counter **Total Rejected Calls** accumulated in new Record module in call-arrival area (orange backing)
 - Already accumulating hour by hour, but this is total over the day

- Type = Output

$100 * NC(\text{Total Rejected Calls}) / NC(\text{Attempted Calls})$

- NC is Arena function that returns the value of that counter

Model 5-3: Overall Call-Center Stats

Replication Conditions

- **Run > Setup > Replication Parameters, Initialize Between Replications**
 - Statistics? System? Details in text
 - Default is both – only way to get truly IID replications
 - Destroys overnight tech-support research jobs, but to do otherwise would complicate model – so accept
- **Run > Setup > Project Parameters**
 - Turned off all but Costing Statistics Collection, for speed
 - Costing required to get ENTITY.WAITTIME

Model 5-3: Overall Call-Center Stats Results

- **Results from five replications**

- Base Case – no additional staff, still 26 trunk lines

Total Cost = \$22,500.07

*Average over 5 replications
Conf. int. half-widths in output*

Percent Rejected = 12.9%

- Add 3 of each of five staff types, 3 more trunk lines

Total Cost = \$22,668.69

Is this better?

Percent Rejected = 1.6%

- **Use in Chapt. 6 for statistically valid experiments**

- Statistical precision
- Compare several alternatives, select best
- Search for configuration that minimizes cost, subject to upper limit on percent rejected

Model 5-4: (s, S) Inventory Simulation Setup

- Different kind of model – not queueing
- Use Blocks and Elements panels exclusively – SIMAN simulation language
 - Mostly just to demonstrate this capability
 - Could be done with higher-level panels we've been using
- Company carries a single discrete item (widgets) in inventory
- $I(t)$ = inventory level (an integer) at time t days past the beginning of the simulation; $I(0) = 60$
- Run simulation for 120 round-the-clock days

Model 5-4: (s, S) Inventory Simulation

Customer Demands Against Inventory

- **Customer interarrival times ~ EXPO (0.1) day (round the clock)**
 - First arrival not at time 0 but after an interarrival time past 0
- **Demand size is discrete random variable**
 - 1, 2, 3, 4 with respective probabilities 0.167, 0.333, 0.333, 0.167
- **If enough items are physically on hand in inventory to satisfy a demand, customer gets demand and leaves**
- **If demand > number of items on hand, customer gets whatever is there and the rest of the demand is backlogged ($I(t)$ becomes negative)**
 - If $I(t)$ was already negative, it just goes more negative

Model 5-4: (s, S) Inventory Simulation

Inventory Review, Replenishment

- **“Take inventory” just past midnight each day**
 - So at exactly times 0, 1, 2, ..., 119 (not 120 ... see below)
 - Two managerially-chosen constant integers $s = 20$ and $S = 40$ (must have $s < S$ if we change these values)
 - If $I(t) \geq s$, do nothing until next inventory evaluation exactly 24 hours later
 - If $I(t) < s$, order $S - I(t)$ items from supplier (order “up to” S)
 - Order does not arrive instantly from supplier, but after a *delivery lag* (a.k.a. *lead time*) $\sim \text{UNIF}(0.5, 1.0)$ day, so sometime during the last half of the day of ordering
 - In the meantime, inventory level could fall further from additional demands, so inventory level will not necessarily pop up to S when the order arrives, but to something less than S

Model 5-4: (s, S) Inventory Simulation

Cost Structure

- **Average ordering cost per day**
 - When an order is placed, incur a fixed cost of \$32, plus an incremental cost of \$3 per item ordered
 - If no order is placed at the beginning of a day, there's no ordering cost, not even the fixed cost
 - At end of simulation, divide total of ordering costs by 120
- **Average holding cost per day**
 - Whenever $I(t) > 0$, incur \$1 per day per item on hand
 - Average holding cost = $\int_0^{120} 1 \times \max(I(t), 0) dt / 120$
- **Average shortage cost per day**
 - Whenever $I(t) < 0$, incur \$5 per day per item in backlog
 - Average shortage cost = $\int_0^{120} 5 \times \max(-I(t), 0) dt / 120$

Model 5-4: (s, S) Inventory Simulation

Cost Structure (cont'd.)

- During periods when $I(t) = 0$ there's neither holding nor shortage cost incurred
- Overall performance measure
 - = Average *total* cost per day
 - = sum of average ordering, holding, and shortage costs per day
- Don't evaluate inventory at time 120
 - We might order and incur an ordering cost then, but order will never arrive
 - We'll fudge this, but an Exercise asks you to do it right

Model 5-4: (s, S) Inventory Simulation

Data Structure

- **Use Blocks, Elements panels exclusively**
 - Even for Variables, Expressions, Attributes, Entities, statistics collection, and run control
- **Variables Element (initialized, or default to 0 initially)**
 - `Inventory Level =  $I(t)$` , changes during run, initialized to 60
 - `Little s = s = 20`
 - `Big S = S = 40`
 - `Total Ordering Cost` accumulator
 - `Setup Cost = 32`
 - `Incremental Cost = 3`
 - `Unit Holding Cost = 1`
 - `Unit Shortage Cost = 5`
 - `Days to Run = 119.9999` (The Fudge)

Model 5-4: (s, S) Inventory Simulation

Data Structure (cont'd.)

- **Expressions element**
 - Define Interdemand Time, Demand Size, Evaluation Interval, Delivery Lag
 - *Cumulative* probabilities in DISC function for Demand Size
- **Attributes, Entities elements**
 - Just to define these objects
- **Project, Replicate elements**
 - Similar to *Run > Setup*
- **DStats element**
 - Request accumulation of integrals for total holding, shortage costs

Model 5-4: (s, S) Inventory Simulation

Data Structure (cont'd.)

- **Outputs element**

- Two entries, both of Data type “Output” so that they’re executed only at end of run, and reported
- Avg Ordering Cost computed
- Avg Total Cost added up
 - OVALUE returns most recent value
 - DAVG returns time-persistent average

Model 5-4: (s, S) Inventory Simulation

Logic for Customer Demands

- **Create block for arrival**
 - Entity Type is **Customer**
 - Uses **Interdemand Time** Expression
 - First Creation after an **Interdemand Time**
- **Assign block to decrement Inventory Level by a Demand Size**
 - **Demand Size** was defined as an Expression
 - Backlogging naturally happens
- **Dispose block for customer exit**
 - If backlogged, is accounted for automatically in the (simple) definition and tracking of **Inventory Level**

Model 5-4: (s, S) Inventory Simulation

Inventory Evaluation

- **Create block for Inventory Evaluator entities**
 - First Creation at time 0 – evaluate inventory at start of run
 - Interval is **Evaluation Interval**, defined as Expression
- **Branch block – somewhat like Decide module**
 - To determine whether to place an order now
 - Add “branches,” each evaluated as true or false
 - Clone of incoming entity sent out along each “true” branch, but at most Max Number of Branches will be sent out
 - So we set Max Number of Branches to 1 (default is ∞)
 - First branch of type “If” – if “true” we want to order
 - Second branch of type “Else” – if “true” it means that the first branch was “false” so we don’t order – just Dispose

Model 5-4: (s, S) Inventory Simulation

Placing an Order

- If we exit the Branch block via the top “If” branch, it must be that $I(t) < s$ so we want to order up to S
- **Assign block**
 - Define **Order Quantity** Attribute
 - Could have made this a Variable in this model with these parameters, but it’s more general for it to be an Attribute ... why?
 - Increment **Total Ordering Cost** Variable
- **Delay block for Delivery Lag**
- **Assign block to increment Inventory Level by the Order Quantity**
- **Dispose block**

Model 5-4: (s, S) Inventory Simulation Animation

- **Plot separate “in the black” and “in the red” curves**
 - If in backlog, red curve will be plotted in negative direction due to its Expression
- **Pair of Level (“thermometer”) animations**
 - Fill Direction for “in the red” is Down