A Manager-Friendly Platform for Simulation Modeling and Analysis of Call Center Queueing Systems

Robert Saltzman
Vijay Mehrotra

Robert Saltzman © 2004
Introduction

• For a manager, even an animated simulation model can be a barrier to doing analysis
• How to provide an Excel interface to make an Arena model easier to experiment with?
• Here, we apply the ideas to call center systems, but it could fit many other settings
• Many call centers can be represented by an $X/X/N/K+X$ queueing system…
X/X/N/K+X Queueing System

- No assumptions about:
  - Interarrival time distribution (1st X)
  - Service time distribution (2nd X)
  - Abandonment time distribution (Last +X)
- N identical servers working in parallel
- K = System capacity = N + MaxQ
- MaxQ = Maximum number allowed in queue
The Simulation Model is Straight-Forward

• Resource capacity is set at beginning of run
• Callers are created and move through the model:
  – They *balk* if there’s no room in the Q
  – They *abandon* if they wait too long in the Q
  – Otherwise, they get served by one of N agents
• Perform. measures are ordered in Outputs element
• Model can run by itself or from Excel
How to get Excel data into Arena model?

• Variables & attributes can be read in directly via ReadWrite & File modules in Arena 7:
  => *Could* do this with N, MaxQ and SLTarget
• However, expressions can’t be read in this way:
  => *Problem* with the 3 time distributions
• Some run-time parameters can’t be altered once the run starts:
  => *Problem* reading in Warm Up Time
• So, I used VB/Excel rather than ReadWrite/Arena

Robert Saltzman © 2004
Input Expressions

IATDist, STDist, and AbTDist each require a 2-step process to get transferred to the model

1. In Arena: modify tag on the corresponding row of the Expression data module so that this row can be identified by VB, e.g., tag = “InterArrivalDistrib”

2. In VB:
   a) Locate this tag’s index with the find method
   b) Set a variable to point to the module containing this tag
   c) Set this module’s “Value” operand to the desired input
Input Variables

NServ, MaxQ and SLTarget each require a similar 2-step process to get transferred to the model

1. In Arena: modify tag on the corresponding row of the Variable data module so that this row can be identified by VB, e.g., tag = “MaxQCap”

2. In VB:
   a) Locate this tag’s index with the find method
   b) Set a variable to point to the module containing this tag
   c) Set module’s “Initial Value” operand to the desired input
Input Replication Parameters

RepLength, NReps, and WarmUp each have a corresponding Arena model object variable:

- oModel.ReplicationLength = Str(RepLength)
- oModel.NumberOfReplications = Str(NReps)
- oModel.WarmUpPeriod = Str(WarmUp)
Sending Arena Output to Excel

- Fortunately, this is easier than the input side 😊
- Order the performance measures using the *Outputs* element in Arena model
- For each performance measure, there are SIMAN objects that contain the average, 95%CI half-width, min & max across replications
- **EX**: Average waiting time in line is stored in oSIMAN.OutputAverageAcrossReplications(2)
Conclusion

• Model is \textit{easy} for a non-simulationist to use
• This Excel interface can be \textit{combined} with other VB macros to quickly analyze a large number of scenarios
• For more details:
  – See Chapter 10 of Arena textbook
  – See our working paper: email me if interested