Homework 6

DATA604 Simulation and Modeling

Daniel Dittenhafer

April 21, 2016

1) Drivers License Facility Simulation

a) Home many 'source', 'server', 'sink' do we need to develop this model, what do those objects stand for in the real system?

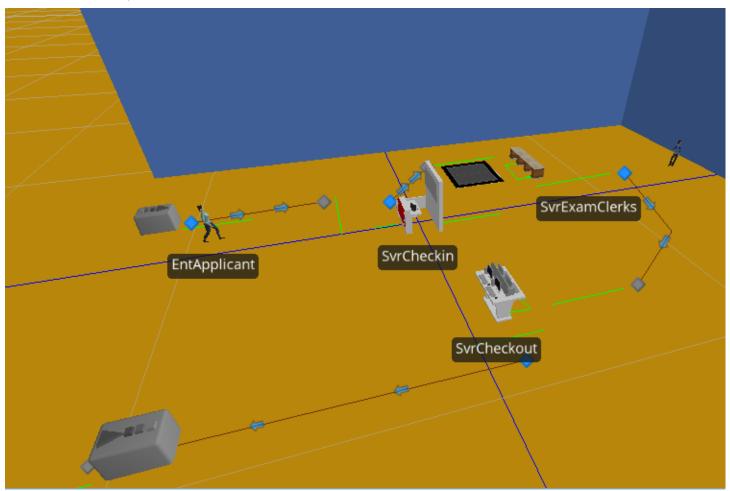- We need 1 **source**.
- 3 **servers**
- 1 **sink**

The source, named *SrcApplicantArrives* in my model, effectively represents the front door of the driver's license facility in this model. It generates the arrival events for the applicant entity.

The servers, named *SvrCheckin*, *SvrExamClerks* and *SvrCheckout* in my model, represent the check clerk, exam clerks and checkout computers, respectively.

The sink, named SnkApplicantDeparts, represents the exit door which enables the applicants to leave after they are done.
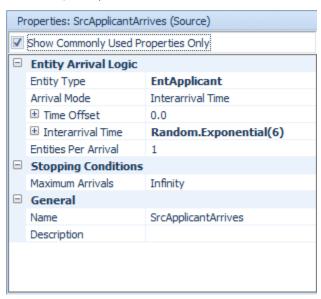
b) Simio Model Screenshots

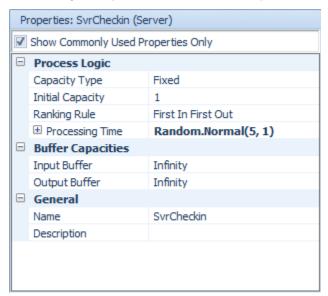The screenshots of my model in Simio follow:

SrcApplicantArrives Properties

I chose to use the exponential distribution for interarrival time since the question stated "approximately 10/hour" as opposed to "exactly"" 10/hour.

Properties: SrcApplicantArrives (Source)	
☑ Show Commonly Used Properties Only	
Entity Arrival Logic	
Entity Type	**EntApplicant**
Arrival Mode	Interarrival Time
⊞ Time Offset	0.0
⊞ Interarrival Time	**Random.Exponential(6)**
Entities Per Arrival	1
Stopping Conditions	
Maximum Arrivals	Infinity
General	
Name	SrcApplicantArrives
Description	

SvrCheckin Properties

For the checkin processing time, I chose to use the normal distribution with a mean of 5 minutes and standard deviation of 1 minute. Again, my rationale was that the question write stated "approximately 5 minutes".

Properties: SvrCheckin (Server)	
☑ Show Commonly Used Properties Only	
Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
⊞ Processing Time	**Random.Normal(5, 1)**
Buffer Capacities	
Input Buffer	Infinity
Output Buffer	Infinity
General	
Name	SvrCheckin
Description	

SvrExamClerks Properties

Properties: SvrExamClerks (Server)	
☑ Show Commonly Used Properties Only	
Process Logic	
Capacity Type	Fixed
Initial Capacity	**2**
Ranking Rule	First In First Out
⊞ Processing Time	**Random.Normal(8.8, 1)**
Buffer Capacities	
Input Buffer	Infinity
Output Buffer	Infinity
General	
Name	SvrExamClerks
Description	

SvrCheckout Properties

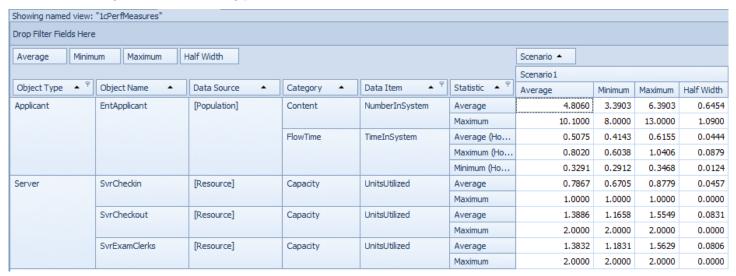| Properties: SvrCheckout (Server)	
☑ Show Commonly Used Properties Only	
Process Logic	
Capacity Type	Fixed
Initial Capacity	**2**
Ranking Rule	First In First Out
⊞ Processing Time	**Random.Normal(9, 1)**
Buffer Capacities	
Input Buffer	Infinity
Output Buffer	Infinity
General	
Name	SvrCheckout
Description	

c) Run the model and obtain the performance measures...

I set the simulation to run for 8 hours based on the concept of a business day. I created an Experiment with 10 Replications which was run to generate the following performance results:

Showing named view: "1cPerfMeasures"

Drop Filter Fields Here

						Scenario ▲			
						Scenario1			
Object Type ▲	Object Name ▲	Data Source ▲	Category ▲	Data Item ▲	Statistic ▲	Average	Minimum	Maximum	Half Width
Applicant	EntApplicant	[Population]	Content	NumberInSystem	Average	4.8060	3.3903	6.3903	0.6454
					Maximum	10.1000	8.0000	13.0000	1.0900
			FlowTime	TimeInSystem	Average (Ho...)	0.5075	0.4143	0.6155	0.0444
					Maximum (Ho...)	0.8020	0.6038	1.0406	0.0879
					Minimum (Ho...)	0.3291	0.2912	0.3468	0.0124
Server	SvrCheckin	[Resource]	Capacity	UnitsUtilized	Average	0.7867	0.6705	0.8779	0.0457
					Maximum	1.0000	1.0000	1.0000	0.0000
	SvrCheckout	[Resource]	Capacity	UnitsUtilized	Average	1.3886	1.1658	1.5549	0.0831
					Maximum	2.0000	2.0000	2.0000	0.0000
	SvrExamClerks	[Resource]	Capacity	UnitsUtilized	Average	1.3832	1.1831	1.5629	0.0806
					Maximum	2.0000	2.0000	2.0000	0.0000

d) Adding an optional "computerized exam kiosk"

2) M/M/1 Comparision

I wrote my developed queueing simulation program in R. The code follows:

```r
# Create a data frame of the new customers and their jobs
newJobs <- data.frame(customer=seq(1, 10),
                      iaMins=rexp(10, rate=1/10),
                      arrivalMins=rep(0, 10),
                      svcTimeMins=rexp(10, rate=1/7),
                      timeSvcBegin=rep(0, 10),
                      queueWaitMins=rep(0, 10),
                      timeSvcEnd=rep(0, 10),
                      timeInSystem=rep(0, 10))
# Determine overall arrival times
newJobs$arrivalMins <- cumsum(newJobs$iaMins)
# Join the existing and new jobs into one table
simTable <- newJobs
# Loop over the rows the compute the various activity and clock times
for(i in seq(2, nrow(simTable)))
{
  simTable[i,]$timeSvcBegin <- max(simTable[i,]$arrivalMins, simTable[i-1,]$timeSvcEnd)
  simTable[i,]$queueWaitMins <- simTable[i,]$timeSvcBegin - simTable[i,]$arrivalMins
  simTable[i,]$timeSvcEnd <- simTable[i,]$timeSvcBegin + simTable[i,]$svcTimeMins
  simTable[i,]$timeInSystem <- simTable[i,]$timeSvcEnd - simTable[i,]$arrivalMins
}

# Show the table
simTable
```