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**

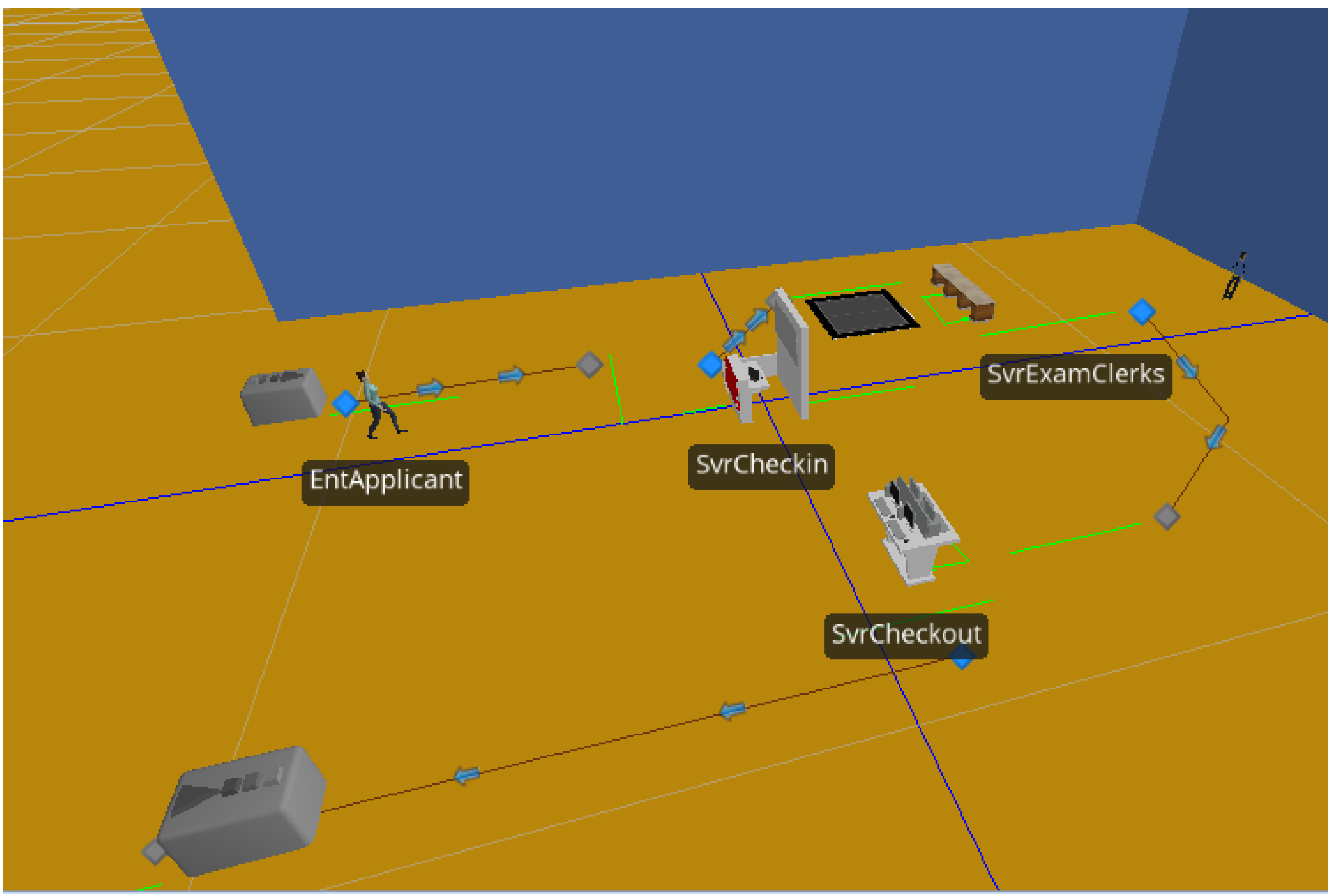
The source, named *SrcApplicantArrives* in my model, effectively represents the front door of the driver's license facility in this model. It generates the arrival events for the applicant entity.

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.

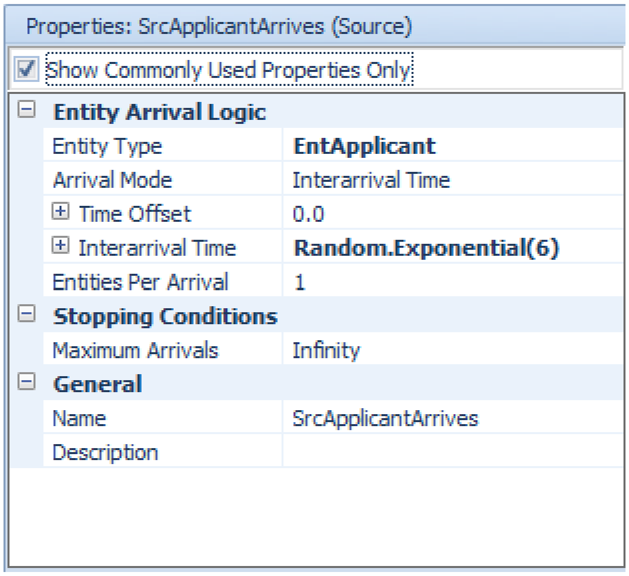
b) Simio Model Screenshots

The screenshots of my model in Simio follow:



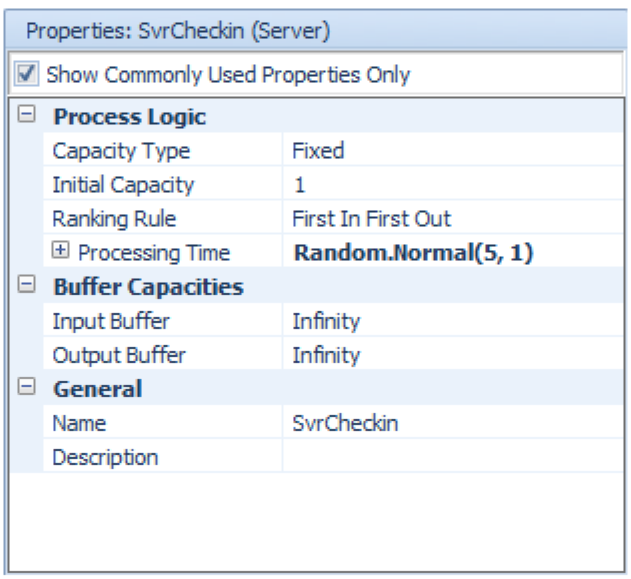
SrcApplicantArrives Properties

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



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”.



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...

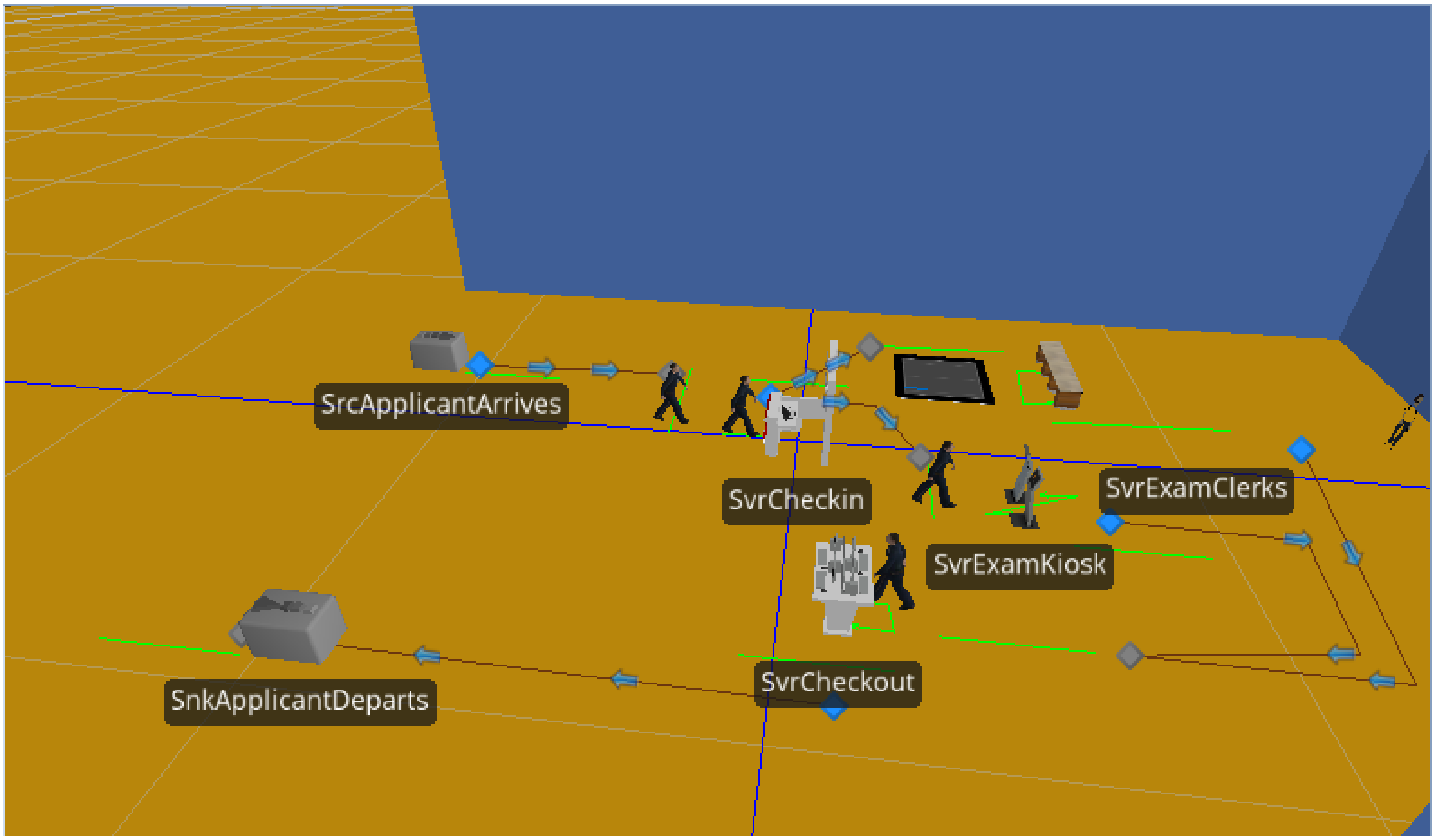
I set the simulation to run for 8 hours based on the concept of a business day. I created an Experiment with 10 Replications which was run to generate the following performance results:

Showing named view: "1cPerfMeasures"

Drop Filter Fields Here

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Scenario1 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 ρ value are generally close to the analytic solution. I didn't figure out how get a good number in queue value from my R program, but the analytic solution and Simio are quite close. For W, the expected system time, again the analytic solution and Simio are quite close… I'm not sure why my R prog is lower. Simio and my R program were run over 1000 customers, time was dependent on completion of all 1000 customers.

simType	p	Lq	W
Analytic	0.7000000	1.633333	23.33333
R Prog	0.6826126	NA	18.13304
Simio	0.6970520	1.621170	23.42610

```
RNGkind("Mersenne-Twister")
set.seed(020275)
maxCustomers <- 1000
iaRate <- 1/10
svcTimeRate <- 1/7

# Create a data frame of the new customers and their jobs
newJobs <- data.frame(customer=seq(1, maxCustomers),
                      iaMins=rexp(maxCustomers, rate=iaRate),
                      arrivalMins=rep(0, maxCustomers),
                      svcTimeMins=rexp(maxCustomers, rate=svcTimeRate),
                      timeSvcBegin=rep(0, maxCustomers),
                      queueWaitMins=rep(0, maxCustomers),
                      timeSvcEnd=rep(0, maxCustomers),
                      timeInSystem=rep(0, maxCustomers),
```

```r
                      svrIdleTime=rep(0,maxCustomers),
                      custInQueue=rep(0,maxCustomers))
# 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(1, nrow(simTable)))
{
  if(i == 1)
  {
    # Special handling for first row
    simTable[i,]$timeSvcBegin <- simTable[i,]$arrivalMins
    simTable[i,]$svrIdleTime <- simTable[i,]$arrivalMins
  }
  else
  {
    # Initialize for > first customer
    simTable[i,]$timeSvcBegin <- max(simTable[i,]$arrivalMins, simTable[i-1,]$timeSvcEnd)
    simTable[i,]$svrIdleTime <- ifelse(simTable[i,]$arrivalMins == simTable[i,]$timeSvcBegin,
                                   simTable[i,]$timeSvcBegin - simTable[i-1,]$timeSvcEnd, 0)
  }

  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 results table
summary(simTable)

totalMin <- max(simTable$arrivalMins)
totalMin
svrIdleMin <- sum(simTable$svrIdleTime)
svrIdleMin
sysUtilRate <- (totalMin - svrIdleMin) / totalMin
sysUtilRate
expectedTimeInSystem <- mean(simTable$timeInSystem)
expectedTimeInSystem
#
# Analytic Solution
#
# System Utilization Rate
rho <- iaRate / svcTimeRate
rho
# Variance
sigma2 <- 1 / svcTimeRate^2
sigma2
# Expected System Time
w <- (1/svcTimeRate) + (iaRate * (sigma2 + sigma2)) / (2 * (1 - rho))
w
# Expected number of customers in the queue
Lq <- (rho^2 *  (1 + sigma2 * svcTimeRate^2)) / (2 * (1 - rho))
Lq
```

DES 6.1

First some queue math:

```r
lambda <- 1 / 4
mu <- 1 / 3
rho <- lambda / mu
rho
```

```
## [1] 0.75
```

```r
Lq <- (rho^2) / (1 - rho)
Lq
```

```
## [1] 2.25
```

```r
Wq <- rho / (mu * (1 - rho))
Wq
```

```
## [1] 9
```

```r
costOfMechanicDelayPerHour <- 15 * Lq
costOfMechanicDelayPerHour
```

```
## [1] 33.75
```

Based on a average cost per hour of 33.75 for 2.25 delayed mechanics, it appears advisable to have a second tool-crib attendant (at $10/hour).

DES 6.2

First some math:

$$W_q = \frac{\lambda(1/\mu^2 + 1/\mu^2)}{2(1-\rho)}$$

$$3 = \frac{\lambda(1/(1.5)^2 + 1/(1.5)^2)}{2(1-(\lambda/1.5))}$$

$$3 = \frac{\lambda(1/2.25 + 1/2.25)}{2(1-(\lambda/1.5))}$$

$$3 = \frac{\lambda(0.444 + 0.444)}{2-(1.33\lambda))}$$

$$3(2-(1.33\lambda))) = \lambda(0.888)$$

$$6 - 4\lambda = \lambda(0.888)$$

$$\frac{6}{\lambda} - \frac{4\lambda}{\lambda} = 0.888$$

$$\frac{6}{\lambda} - 4 = 0.888$$

$$\frac{6}{\lambda} = 4.888$$

$$\lambda = \frac{6}{4.888} \approx 1.227$$

```
lambda <- 1.227
mu <- 1.5
rho <- lambda / mu

Wq <- (lambda * (1 / mu^2 + 1 / mu^2)) / (2 * (1 - rho))
Wq
```

```
## [1] 2.996337
```