SA421 – Simulation Modeling Fall 2017
Assoc. Profs. D. Phillips and N. Uhan

Lesson 3. Introduction to JaamSim

1 Overview

- **JaamSim** is a free and open source software package designed specifically for discrete-event simulation
- To learn the basics of JaamSim, let's use it to tackle the following problem:

Problem 1. The Nimitz Coffee Bar (NCB) has a line where customers wait to place their order with the cashier. Once they have ordered, two baristas complete their order. Order completion is conducted on a first-in, first-out (FIFO) basis.

Model the customer interarrival times as an exponential random variable with a mean of 41 s; the cashier times as a gamma random variable with mean 2 s and shape parameter 4; and the server times as a gamma random variable with mean 49s and shape parameter 3.

a. Simulate 1 day of NCB's operations. Assume NCB is open continuously for 8 hours a day.
b. What is the average delay customers experience at the cashier? At the barista?
c. What is the time average number of customers waiting at the cashier? At the barista?

2 Before we begin...

- You can find the JaamSim User Manual and examples of how to use various JaamSim model objects here:

```
http://jaamsim.com/downloads.html
```

- Launch JaamSim and adjust the view options:
  - Uncheck Options » Show Axes and Options » Show Grid
  - Activate the 2D button to turn on the 2D view

3 Creating model objects

- We can add objects to our model by dragging them from the **Model Builder** into the **View Window**
- Basic controls in the View Window:

Mouse/Keyboard Action	Effect
Left Drag	Pans the window view
Left Click	Selects point of interest
Control + Left Drag	Moves object – must select object first
Shift + Left Drag or Scroll Wheel	Zoom in/out

- First, let's add a **SimEntity** object (under **Process Flow**) to our model to represent a customer in our system
- It's nice to see the name of the objects, so right-click the new SimEntity object and select Show Label
- Double-click on the label and rename the SimEntity to "Customer"

- In a similar fashion, define the following objects:

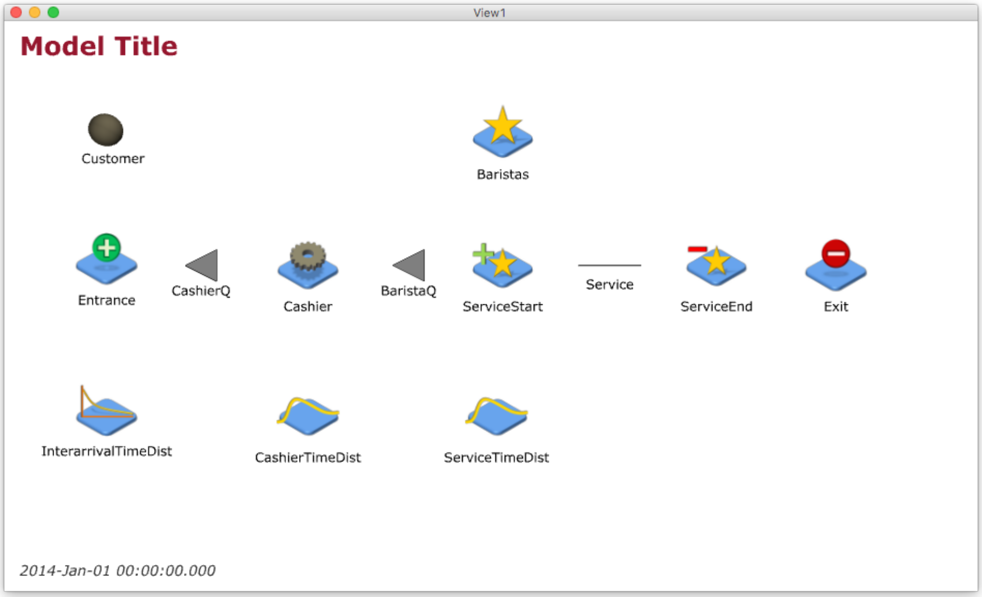
Object Type	Name
EntityGenerator	Entrance
Queue	CashierQ
Server	Cashier
Queue	BaristaQ
Resource	Baristas

Object Type	Name
Seize	ServiceStart
EntityDelay	Service
Release	ServiceEnd
Entity Sink	Exit

- The above objects represent the different parts of the system that a Customer encounters
- Next, again in a similar fashion, define the following **Probability Distribution** objects:

Object Type	Name
ExponentialDistribution	InterarrivalTimeDist
GammaDistribution	CashierTimeDist
GammaDistribution	ServiceTimeDist

- At this point, your View Window should look like this:



4 Defining the objects and putting them together

- Next, we need to modify the **inputs** of the objects we created
- By modifying their inputs, we can connect the objects together and define characteristics of the system

4.1 InterarrivalTimeDist

- InterarrivalTimeDist is an **ExponentialDisribution** object, which generates random variates from an exponential distribution
- Select the InterarrivalTimeDist object

- In the **Input Editor**, make the following changes:

Keyword	Value	What is this?
UnitType	TimeUnit	The unit type for the random variates returned by this object
Mean	41 s	Mean of the exponential distribution

- Note that you need to specify units for the mean!

4.2 CashierTimeDist and ServiceTimeDist

- CashierTimeDist and ServiceTimeDist are **GammaDisribution** objects
- These objects generate random variates from a gamma distribution
- For CashierTimeDist, make the following changes:

Keyword	Value	What is this?
UnitType	TimeUnit	The unit type for the random variates returned by this object
Mean	2 s	Mean of the gamma distribution
Shape	4	Shape parameter for the gamma distribution

- For ServiceTimeDist, make the following changes:

Keyword	Value	What is this?
UnitType	TimeUnit	The unit type for the random variates returned by this object
Mean	49 s	Mean of the gamma distribution
Shape	3	Shape parameter for the gamma distribution

4.3 Entrance

- Entrance is an **EntityGenerator** object, which creates a series of entities and passes them to another object
- Make the following changes to Entrance:

Keyword	Value	What is this?
NextComponent	CashierQ	This is where the generated entities go next
InterArrivalTime	InterarrivalTimeDist	The elapsed time between one generated entity and the next
PrototypeEntity	Customer	The entity to be copied by the EntityGenerator

4.4 CashierQ and BaristaQ

- CashierQ and BaristaQ are **Queue** objects
- This type of object is a location for entities to wait for processing by a Server or Seize object (others as well)
- The default input values are OK here

4.5 Cashier

- Cashier is a **Server** object, which processes an incoming entity and then passes it to another object
- Note that a Server object alone cannot represent multiple identical parallel servers, like the baristas in our problem
- Make the following changes to Cashier:

Keyword	Value	What is this?
NextComponent	BaristaQ	This is where the incoming entity goes next
WaitQueue	CashierQ	Queue of entities waiting for processing by this Server
ServiceTime	CashierTimeDist	The time required to process the incoming entity

4.6 Baristas

- Baristas is a **Resource** object, which represents a pool of identical processing units
- Make the following changes to Cashier:

Keyword	Value	What is this?
Capacity	2	The number of identical resource units available

4.7 ServiceStart

- ServiceStart is a **Seize** object, which allocates one or more units of a Resource to an incoming entity
- Make the following changes to ServiceStart:

Keyword	Value	What is this?
NextComponent	Service	This is where the incoming entity goes next
WaitQueue	BaristaQ	Queue of entities waiting to seize the specified Resources
ResourceList	Baristas	Resources seized by the incoming entity

4.8 Service

- Service is an **EntityDelay** object, which delays an incoming entity before passing it to another object
- This kind of object is often used to model service times or travel times
- Make the following changes to Service:

Keyword	Value	What is this?
NextComponent	ServiceEnd	This is where the incoming entity goes next
Duration	ServiceTimeDist	The time required to process the incoming entity

4.9 ServiceEnd

- ServiceEnd is a **Release** object, which de-allocates one or more units of a Resource from an incoming entity
- Make the following changes to ServiceEnd:

Keyword	Value	What is this?
NextComponent	Exit	This is where the incoming entity goes next
ResourceList	Baristas	Resources to be released

5 Final steps

- Activate the → button to visualize how the objects are connected
- Define the simulation time:
  - Select **Simulation** in the **Object Selector**
  - Make the following changes in the Input Editor

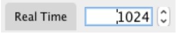
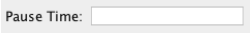
Keyword	Value
RunDuration	8 h

- Give the model a meaningful title:
  - Select **Title** under **Graphics Objects** > **Overlay Text** in the Object Selector
  - Make the following changes in the Input Editor

Keyword	Value
Format	'Nimitz Coffee Bar'

6 Running the simulation and examining output

- The toolbar contains controls for running the simulation:

Toolbar item	What does it do?
▶	Starts, pauses, and resumes the simulation run
⏮	Stops the model, clears generated entities, sets simulation time to zero
 Real Time 1024	If activated, simulation is run at a constant multiple of wall-clock time
 Pause Time:	The time at which the simulation is automatically paused

- Go ahead and run the simulation!
- Speed it up if you like, or deactivate the **Real Time** button to have the simulation run as fast as possible, without animations
- Let's find the average delay at the cashier
- Select the CashierQ object
- In the **Output Viewer**, you can see the outputs for the cashier queue at the end of the simulation run
- **QueueLengthAverage** is the time-average number of customers in the queue
- **AverageQueueTime** is average delay
- You can find these performance measures for the BaristaQ in a similar fashion

7 Mess around!

- Try making some small changes to the simulation, e.g.
  - Number of baristas
  - Mean of interarrival time, cashier time, and service time distributions
- Run the simulation with your changes and see what happens
- Make sure to reset the model with ⏮ before running it again