Lesson 8. Custom Performance Measures in JaamSim

1 An example

- Suppose you want to compute a "non-standard" performance measure that is not automatically computed by JaamSim
- We can take advantage of JaamSim's flexibility and compute our own custom performance measures
- Let's consider the following example

Example. Customers visit the neighborhood hair stylist Fantastic Dan for haircuts. The customer interarrival time is exponentially distributed with mean 20 minutes. Each haircut takes Fantastic Dan anywhere from 15 to 25 minutes, uniformly distributed. This time also includes the initial greetings and the transaction of money at the end of the haircut. Fantastic Dan works 8 hours a day without breaks.

Simulate 1 day of Dan's operations. What is the average sojourn time?

- The **sojourn time** is the time a customer spends in a queueing system from arrival to departure
  - In Lesson 1, we called this **waiting time**
- JaamSim doesn't automatically compute the average sojourn time, like it does for average delay at each queue
- However, we can still compute the average sojourn time with attributes and the Assign and Statistics objects
- The JaamSim file for this lesson contains a model of Fantastic Dan's shop as described above
- Let's modify it to compute the average sojourn time

2 Our plan

- Define two attributes for the Customer entity:
  - one for a customer's arrival time,
  - the other for a customer's sojourn time
- When a customer arrives, record their arrival time in these attributes
- When a customer departs, compute their sojourn time, and record it in these attributes
- Compute the average sojourn time

3 Defining attributes

- First, let's configure each Customer entity to have two attributes: `ArrivalTime` and `SojournTime`
- For Customer, make the following changes:

Keyword	Value
AttributeDefinitionList	{`ArrivalTime 0 s`} {`SojournTime 0 s`}

4 Recording a customer's arrival time

- Recall that the Assign object assigns one or more attribute values to an incoming entity.
- Let's start by inserting an Assign object between Entrance and DanQ, and naming it RecordArrivalTime
- Modify Entrance as follows:

Keyword	Value
NextComponent	RecordArrivalTime

- Configure RecordArrivalTime as follows:

Keyword	Value
NextComponent	DanQ
AttributeAssignmentList	{ 'this.obj.ArrivalTime = this.SimTime' }

  - `this.SimTime` is the current simulation time
  - `this.obj.ArrivalTime` is the value of `ArrivalTime` for the incoming Customer entity

5 Recording a customer's sojourn time

- Next, let's compute and record each customer's sojourn time
- Insert an Assign object between FantasticDan and Exit, and name it RecordSojournTime
- Modify FantasticDan as follows:

Keyword	Value
NextComponent	RecordSojournTime

- Configure RecordSojournTime as follows:

Keyword	Value
NextComponent	(leave blank for now)
AttributeAssignmentList	(what should this be?)

6 Computing the average sojourn time

- Now that each customer has an attribute containing their sojourn time, we can get the average sojourn time over all customers with the Statistics object
- The Statistics object collects statistical information on the entities it receives
- Insert a Statistics object between RecordSojournTime and Exit, and name it SojournTimeStats
- Modify RecordSojournTime so that it routes entities to SojournTimeStats
- Configure SojournTimeStats like this:

Keyword	Value
NextComponent	Exit
UnitType	TimeUnit
SampleValue	`this.obj.SojournTime`

7 Running the simulation and finding the average sojourn time

- Run the simulation once
- Take a look at the OutputViewer for SojournTimeStats
- What is the average sojourn time?
- What is the minimum and maximum sojourn time?