Lab 4: Simulation with Arena

Your task in this lab is to simulate and animate a model of a two-station queueing system in the simulation modeling environment Arena. Read the description below before starting the lab.

The Situation
Leviathan Limited, a very large enterprise indeed, maintains a repair facility that reconditions certain expensive products when they fail. The repair facility consists of a repair station, an inspection station and a combined repair-and-inspection station. The repair station makes the first attempt to repair a product. The product then moves to the inspection station to verify that the repairs have been successful. Products that pass inspection are shipped back to their owner. The small number of products that fail inspection move to the combined repair-and-inspection station where they are repeatedly repaired and inspected until they work properly. The original vision for this configuration was to exploit the efficiency of specialized repair and inspection stations for most products, and have a general-purpose station for the particularly difficult cases.

Regrettably, profits have not been good for Leviathan Ltd., so management is searching for ways to make more efficient use of resources. An industrial engineer (IE) has noticed that the combined repair-and-inspection station is not heavily utilized; she speculates that this station can be eliminated by sending products that fail inspection back to the primary repair station. For this to be a viable option it must not lead to excessive delays, so the IE must predict the impact of the change to support her proposal.

Some rough estimates are available. Products arrive at a rate of about 5 per day, can be repaired at a rate of about 6 per day, and are inspected at a rate of about 8 per day. Roughly 10% of parts fail inspection, even if they have been repaired several times already. Simulate the proposed system for 30 days to estimate the average cycle time for repaired parts. Approximate all times as exponentially distributed.

Building the Model
The basic model is constructed as follows:

1. Using the Run: Setup... menu, set the Replication Length to 30 days, and set the Simulation Time Units to Days.
2. Use a Create module to model a Poisson arrival process. Remember that in Arena, `Random(Expo)` means exponential time between arrivals. Be sure to enter the mean time between arrivals, not the rate, and make sure the time unit is days.

3. Use a Process module to model the queue for repair. Make the name of this module Repair. Select Seize-Delay-Release as the action, and be sure to Add a resource to represent the repair person; give this resource the name Repair Person. For Delay Type choose Expression and use the function `EXPO(Mean)` to generate exponentially distributed service times. Be sure to specify the mean, not the rate, and make the time unit Days.

4. Use a Process module to model the queue for inspection. Make the name of this module Inspect. Select Seize-Delay-Release as the action, and be sure to Add a resource to represent the repair person; give this resource the name Inspector. For Delay Type choose Expression and use the function `EXPO(Mean)` to generate exponentially distributed service times. Be sure to specify the mean, not the rate, and make the time unit Days.
5. Use a **Decide** module to model passing or failing inspection. Decide 2-way By Chance with a 90% chance of passing (being True). Connect the False outcome to the **Repair** module.

   ![Decide module](image)

6. Use a **Dispose** module to model repaired items leaving the system.

7. Run the model and debug any errors that occur.

Once your model is running correctly, add additional animation as suggested below:

1. Click on the **Entity** spreadsheet icon. Click on **Initial Picture** and use the drop-down menu to select a picture to represent your broken part entities.

   ![Entity spreadsheet](image)

2. From the **Animate** buttons, select **Resource**. This will open a resource dialog box. Do the following:
   a. For the **Identifier** select **Repair Person**.
   b. Select and copy pictures for the **Idle** and **Busy** states using the `<<` key. You may have to open a picture library such as `Workers.plb`.
   c. Be sure to select **Seize Area**; this causes the entity (broken item) to appear to be being worked on by the server.
3. Now create a similar resource picture for the Inspector.

4. From the Animate buttons, select Clock. Place it and drag to size.

5. Add labels to the repair and inspect areas using the text tool. Add any other details or background colors you like by using the drawing tools.

6. Run your model and debug any errors.

7. Suppose we want to have the entity picture change after the item is repaired. To do this, place an Assign module between the two Process modules and connect them. Add the following assignment:

   **Type:** Entity Picture
   
   **Entity Picture:** pick something from the drop-down menu
Now when the entity passes through the Assign module its picture will change. Run your model to verify this.

**What You Should Do**

Now run the simulation for the full 30 days and observe the average cycle time statistic. Print out your model, write the average cycle time on it, and turn it in to Mr. Allen Lee by **12pm on March 14, 2005** at IS lab. **No late labs will be accepted!**
Appendix I: Using Category Overview Reports

Appendix II: Useful Arena Distribution Functions

<table>
<thead>
<tr>
<th>Arena Function</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETA(Alpha1, Alpha2)</td>
<td>Sample from beta distribution</td>
</tr>
<tr>
<td>ERLA(ExpMean, k)</td>
<td>Sample from Erlang distribution with k phases and mean k*ExpMean</td>
</tr>
<tr>
<td>EXPO(Mean)</td>
<td>Sample from exponential distribution</td>
</tr>
<tr>
<td>GAMM(a, b)</td>
<td>Sample from gamma distribution</td>
</tr>
<tr>
<td>LOGN(Mean, StdDev)</td>
<td>Sample from lognormal distribution</td>
</tr>
<tr>
<td>NORM(Mean, StdDev)</td>
<td>Sample from normal distribution</td>
</tr>
<tr>
<td>POIS(Mean)</td>
<td>Sample from Poisson distribution</td>
</tr>
<tr>
<td>TRIA(Min, Mode, Max)</td>
<td>Sample from triangular distribution</td>
</tr>
<tr>
<td>UNIF(Min, Max)</td>
<td>Sample from continuous uniform distribution</td>
</tr>
<tr>
<td>WEIB(a, b)</td>
<td>Sample from Weibull distribution</td>
</tr>
</tbody>
</table>