Arena File Documentation – Written by Samuel Liu

To validate the Java model developed by Richard, I created models in the commercial discrete event simulation modeling software, Arena. The parameters I evaluated these models on were Number of Dispatchers, Fleet Size, Shift Length, Task specialization, Fleet Heterogeneity, Autonomy, Operator Strategy, Team Coordination, AI, and Exogenous events.

**Number of Dispatchers:**

A model with fixed task creation parameters was created, and run with 2 dispatchers, 3 dispatchers, and 4 dispatchers. We expected the workload for each dispatcher and the waiting time for each task to decrease as the number of dispatcher increased, which was confirmed by the model. Arrival parameters were multiplied by 6, and all service parameters were updated with the distributions at the end of this document

**Fleet size:**

The same model for number of dispatcher was used in this scenario, except with a fixed number of dispatchers. The arrival parameters were doubled and tripled, which replicated 2 and 3 vehicles (or fleets). We expected dispatcher utilization and wait times to increase as the number of tasks created increased, which was verified to be true. The model was run with 1 dispatcher and all service parameters were updated with the normal distributions from the model parameters given at the end of this document.

**Shift Length:**

The same model as above was used again. This time, the resource (dispatchers) and arrival parameters were held constant, while the shift length was varied. All models include a 30 minute warmup period where the resource takes 30 minutes to reach steady-state processing speed. We expected utilization and wait times to remain similar as the shift length increased, while the number of tasks completed would increase linearly. This hypothesis was confirmed by our testing. However, this model does not account for the fact that dispatchers are given breaks and that fatigue may slow down processing times in an actual dispatch center. The shift lengths ranged from 120 minutes to 480 minutes, in 120 minute increments.

**Specialist/Generalist:**

A specialist is a dispatcher who only processes certain tasks, while a generalist is a dispatcher who processes all tasks. In our model, Specialist 1 completes tasks 1,2,3, and 4 while Specialist 2 completes task 5,6, and 7. Due to the dissimilar arrival rates and processing times, one dispatcher’s utilization is still significantly higher than the others. One thing to keep in mind is that specialists can normally process the tasks they specialize in faster than generalists can process their tasks, which was not accounted for in the model (until using sets later in team coordination). Both generalists and specialists here process tasks at the same rate. Two dispatchers were used in each scenario.

**Fleet Heterogeneity:**

Fleet heterogeneity validates the representation of handling tasks coming from two separate fleets with different levels of autonomy. For instance, one fleet may have a vehicle-to-vehicle communication link that eliminates communication tasks for the dispatcher, which will reduce the number of tasks to process. This was modeled by altering all arrival rate parameters in Arena by a set percentage. As an example, for a fleet of 10 normal vehicles and 10 autonomous vehicles with 50% fewer tasks generated, I would multiply the task arrival parameters by 15 instead of 20 for two fleets of 10 normal vehicles. We also tested for constant arrival parameters, but increased and decreased processing times. This isn’t entirely representative of real heterogenous fleets because not every task is reduced the same amount through autonomy. Some tasks may be eliminated completely, some with shorter processing times, and some remain the same.

**Autonomy:**

Autonomy was validated in a similar manner to fleet heterogeneity. Say 30% of a fleet has vehicle-to-vehicle communications and those 30% produce 50% less tasks. To make calculation simple, I used a fleet of 10 vehicles. I then modified the arrival parameters to account for 30% autonomy by setting 7 normal vehicles plus the equivalent of 1.5 vehicles for the other 3 autonomous vehicles for an overall arrival parameter of 8.5.

**Operator Strategy:**

This evaluates the efficiency of dispatchers using the First-in-First-Out method, Shortest Time First, and on order of task importance. FIFO is the default queue strategy in Arena. Shortest Time First was determined by ranking the tasks from shortest to longest mean processing time. Tasks with the same mean processing time were assigned identical priorities. The priorities are listed next to the process time at the end of this document. After creation, each task is assigned an attribute number according to its entity type. The queue is then arranged so the shortest tasks are pushed to the front of the queue and the longer tasks are bumped to the end of the queue. Task importance, or priority, is handled in a similar manner to shortest-time-first. Instead of shortest time, however, each entity type is assigned a priority based on the task’s time sensitivity. For these methods to be evaluated thoroughly, a queue must exist, but arena generates an error when the queue exceeds 150 so it is difficult to make a model that works for all 3 using the same arrival, processing, and dispatcher constants. To work around this issue, I changed the 30 minute warmup time to 0, which means the operators work at 100% of the steady-state rate from the very beginning, and I changed the replication length to 120 minutes so the queue of 150 tasks would not be exceeded.

**Team Coordination:**

Team coordination is an internal communications task that only certain dispatchers called Dispatcher2 can process. It then decreases the amount of service time for other tasks for that dispatcher. This task is independent of fleet size, meaning it does not scale up as the number of vehicles or increases. As a result, it can significantly increase dispatcher utilization for small fleets, but make almost no difference in large fleets. In Low team coordination, 30 percent of operators are Dispatcher2. In high team coordination, 70 percent of operators are Dispatcher2. To easily model the changing autonomy level, there are 10 human dispatchers. When an AI is introduced in “Equal Teammates,” there is an 11th nonhuman dispatcher that processes tasks.

We tested 3 different models on this team coordination platform, the first being “Equal Teammates”. In this case, AI\_1 is a processor that acts similarly to other operators, but it can also perform “Recordkeeping,” “Referencing,” and “Actuation” tasks with 0 processing time and 0 error. To model this, I gave those tasks to AI\_1 only, and placed AI\_1’s priority on all other tasks to last, which leaves the AI\_1 resource free for as long as possible. This can be accomplished through the **sets** feature, which allows skills-based selection for seizing a resource. Set 1 is ordered dispatcher 1, dispatcher 2, where all dispatcher 1’s are utilized before calling upon dispatcher 2’s even though they take the same amount of time to perform a task. This is so dispatcher 2’s can be free to respond to tasks they are more efficient at if a task were to arrive. Set 2 is ordered dispatcher 2, dispatcher 1, to handle such events.

The second model on this platform is called “Operator Assistant.” In this case, the AI does not process tasks, but it does speed up the tasks for operators. In our particular case, it reduces service time of each operator by 50% on “Actuation,” “Directive Mandatory,” “Recordkeeping,” and “Referencing.” When combined with team coordination, all tasks except for “Direct Courtesy 1,2” have their service times cut in half, and “Referencing” is further cut down to 25% of the original service time.

The third model is called “Team Assistant.” The AI reduced service time on “Team Coordination” by 50% and reduced error probability for all operators to 50% of their original value.

**Exogenous Events:**

There are two types of exogenous events. Type 1 generates an extra task with a low arrival rate (appx 1 per shift) and a high service time (~30 mins) and error probability (50%) to the task list. This can simulate a medical emergency. Type 2 increases the arrival rate for all tasks by 10%, which is representative of a bad weather scenario.

**Additional Notes:**

1. While all models count the number of undetected failures, the failure rate is not pulled from an exponential curve. All failures are based on the mode failure for that task, and there is a 50% chance of failure detection. Failure rates due to autonomy were not changed because failure rates were too low to be statistically useful.
2. Whenever a dispatcher is more efficient at completing a task, those tasks are given higher priority to the more efficient dispatcher (e.g. dispatcher 2 has 50% shorter process times for actuation. Any new actuation task will look for free dispatcher 2’s before being assigned to a dispatcher 1.)
3. There is a 30 minute warmup time for all simulations where operator productivity gradually rises to the steady-state value, except for the operator strategy (FIFO, priority, STF) models.
4. The Random(EXPO) expression for create boxes takes an exponential distribution around the mean task arrival time. Expression EXPO(number) can be used for task creation or processing times, and pulls numbers around the mean task arrival/processing time. The lambdas given in my parameters were tiny and represented the inverse of the mean time we needed, so I used EXPO(1/lambda) to correct the issue. I validated this by using running the simulation multiple times with changing parameters for the Comms\_Other task, which I was given a lambda value of 7.5188 minutes for processing time. As a very frequently occurring event, Comms\_Other should be taking about 10 seconds to process, which is more representative of EXPO(1/7.5188) mins.
5. Comms\_Other has a different arrival parameter than given in the parameter documentation sheet. With a lambda of .95, a task would be arriving almost every minute. I altered this to 0.095, so one event is created approximately every 10 minutes.
6. Entities may not match perfectly in the graphs excel doc because some models in “number of dispatchers” and “fleet size” had incorrectly labelled entities. The issue only appears in those two folders. The correct numbers are there, but arrival rates and service parameters may be labelled as a different task. The models have been fixed, but the graphs have not been updated. To compare accurately, run the model and read the numbers off the report.

**Parameters**

**Actuation Uniform (0.5,2) #4 (Rank by processing time)**

**DM Uniform (0 0.5) #2**

**DC Uniform (0.167 2.5) #5**

**DC\_2 Uniform (.167 2.5) #5**

**Comms\_Other (EXPO 1/7.5188) #1**

**Recordkeeping Uniform (0.05, 1.5) #3**

**Referencing U (0.05,1.5) #3**