Arena File Documentation

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 Dispatchers, Fleet Size, Shift Length, Specialist/Generalist, 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.

**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 an additional 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.

**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.

**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.

**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 the arrival rate parameters in Arena. 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. 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 v2v communications and those 30% produce 50% less tasks. I then modified the arrival parameters to account for that by setting 7 normal vehicles plus the equivalent of 1.5 vehicles for the other 3 autonomous vehicles for an 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. 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.

**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. In Low team coordination, 30 percent of operators are Dispatcher2. In high team coordination, 70 percent of operators are Dispatcher2. Note that the decrease in error probability is so low that it is not represented in the model.

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.

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 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.