A screenshot of a computer

Description automatically generated with low confidence

Memorandum

|  |  |
| --- | --- |
| to: | Travis Jensen |
| from: | Gregory S. Macfarlane, Ph.D., PE GRANT G. SCHULTZ, PH.D., PE, PTOE |
| subject: | IMT Deployment Optimization IMT PERFORMANCE: PHASE III November TAC MEETING AGENDA |
| date: | 20 March 2023 |
| CC: | Technical Advisory Committee MEMBERS |

# Agenda

1. IMT Performance Measures Phase III Project Update (Joel Hyer)
2. IMT Simulation Project Update (Daniel Jarvis / Brynn Woolley)
   1. IMT Vehicle Deployment
3. Additional discussion
4. Schedule next TAC Meeting: Proposed May 2023.

# IMT Performance Measures Phase III

## Project Update

Figure 1 illustrates the project schedule for the Analysis of Performance Measures of UDOT’s Traffic Incident Management Program: Phase III. Tasks 2 and 3 are complete, while Tasks 4 and 5 are currently underway. These tasks will need to be extended slightly based on the dates that the data were received.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | 2021 | | | 2022 | | | | | | | | | | | | 2023 | |  |
| Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Jan | Feb | Mar |
| 1 |  |  |  |  | TAC |  |  | TAC |  |  | TAC |  |  | TAC |  |  | TAC |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Review |
| KEY: |  | Task progress | | | | | |  | TAC meeting and updates | | | | | | | | | |

|  |  |
| --- | --- |
| **Task** | **Description** |
| Task 1: | Kickoff Meeting/TAC Meeting/Project Management |
| Task 2: | Literature Review |
| Task 3: | Collect Incident Management Performance Data |
| Task 4: | Analyze Performance Measures Data |
| Task 5: | Analyze User Cost Data |
| Task 6: | Compare Performance Measures |
| Task 7: | Report Results |

Figure 1. IMT Performance Measures Phase III project schedule and completion.

### Task 3: Collect Incident Management Performance Data

Data were collected for the months of March through August 2022. The final August 2022 data has now been received by the research team and is currently being processed.

### Task 4/5: Analyze Performance Measures Data/Analyze User Cost Data

The performance measures and user cost data are currently being evaluated. Although not all the data have been evaluated to analyze excess user costs, a summary of the performance measures is provided in the attached PowerPoint slides.

### Next Steps

The research team will continue to analyze crashes from July and August and will then begin more rigorous statistical analysis of performance measures and user cost data. Preliminary results will be shared at the next TAC meeting.

# IMT Deployment Optimization

## Project Update

Figure 2 illustrates the project schedule and completion percentages alongside the original task completion. Task 2 (Literature Review) was completed in Spring 2022, and the research team is fully engaged in building the simulation model. The team is developing scenarios for IMT response and should begin Task 4 by the next TAC meeting in May. Though modestly behind our target completion schedule at this point, the project is still scheduled to complete by Quarter 4 2023.

A picture containing chart

Description automatically generated

Figure 2. Project schedule and completion. Shaded bars represent task completion, and shadows represent original schedule. Blue line represents today (October 2022).

## Methodology

The method we are using to represent and analyze the IMT system is MATSim, an open-source framework used for mesoscopic modeling. MATSim is an activity-based model implemented in Java that models transport users as individual agents. MATSim is iterative, allowing users to adjust travel plans

The method we have chosen for representing and analyzing the IMT system is MATSim, an open-source framework used for mesoscopic modeling. MATSim is an activity-based, extendable, multi-agent simulation framework implemented in Java. MATSim models transport users as individual agents. MATSim is iterative and allows users to adjust travel plans during a single iteration, from iteration to iteration, or both. Details on the operation of MATSim and its theoretical background are given in Horni, et al. (2016). This research will use two primary features of MATSim:

1. Network change events, which allow the capacity or speed of a highway link to change at a particular point of the day.
2. On-demand vehicle services, which allow for agents to adapt to changing conditions, such as a capacity-reducing highway incident.

We are working with researchers at Technische Universität Berlin to represent IMT vehicles as an on-demand vehicle service and hope to demonstrate a working IMT response at the next TAC meeting. In this memo, we describe the network change events implementation and scenario generation.

### Network Change Events

MATSim represents transportation networks as digital links connecting nodes. Each link contains information on hourly capacity and free-flow speed; as volume on the link increases, the speed of travel is reduced until the link reaches its storage capacity. At this point, the link becomes a first-in, first-out (FIFO) queue receiving vehicles from upstream and passing the downstream. Details on MATSim’s traffic flow model are given by Flötteröd (2016).

The initial scenario network was obtained by converting the Wasatch Front Regional Council (WFRC) and Mountainland Association of Governments (MAG) travel demand model network into a file that could be used by MATSim. This network was selected because it already contained the best possible estimates of free-flow speed and capacity for roadways in Utah. One drawback of this network is that it does not contain all streets – local roads are omitted for computational simplicity. This drawback is not expected to affect the outcomes of the simulation.

In MATSim, network alterations are called “network change events” and can be used to change the capacity of network links. When running a network change event, there are three required elements, a start time, a capacity value, and a link identity. Each incident has two network change events, one at the beginning of the incident and one at the end. Figure 2 illustrates how network change events are written in Java code. This example is a code snippet from a MATSim incident scenario. An incident occurred at 9:13 am near 5800 S I-15 southbound. The incident reduced roadway capacity by 28% and was assigned to network link 24342. The second network change event shows that the link returned to total capacity at 9:23 am. The number of incidents per scenario ranges from 1 to 18, and each value has a distinct probability of occurrence.



Figure 2: Java code to execute a network change event in a MATSim scenario.

## Developing Scenarios

As part of the IMT Phase 3 research project, the project team analyzed Utah Highway Patrol incident data for several months in 2018. In total, the research analyzed 1071 incidents over 133 days. The number of incidents per day ranged from 1 to 18, and the distribution of incident frequency is shown in Figure 3. These data included 189 incidents with at least one responding IMT. These were then divided into five subcategories based on the total hours of excess travel time associated with that incident. In addition to the incident class, the team identified each incident’s coordinates, duration, and a time-weighted average of the capacity reduction associated with the incident.

Figure 3: Histogram of incidents per day analyzed in IMT Phase 3 project.

Table 1: Incidents by Severity Class

|  |  |  |
| --- | --- | --- |
| Severity Code | Hours of Excess Travel Time | Frequency |
| 1 | 0-200 | 66 |
| 2 | 200-400 | 35 |
| 3 | 400-800 | 37 |
| 4 | 800-2000 | 32 |
| 5 | 2000+ | 19 |

Incidents were modeled successfully in MATSim using this information. First, the incident data file was read into a CSV parser that recorded each incident's latitude and longitude coordinates. The Java class then used a method called "getNearestLinkExactly" to find the nearest MATSim link of each incident. This method reads the network file and the incident coordinates and outputs the link closest to the coordinates. The method was set to only consider links with at least three lanes to ensure that it only selected freeway links. Second, we wrote the Incident Number Selector script to determine the number of incidents in each simulation. For example, based on their observed frequencies, there is a 14.3% chance that the model will create 6 incidents and a 10.5% chance that it will create 10. Rather than simply producing the average number of daily UHP incidents, the script shows the variance in daily incident volume. Third, a script similar to the incident number selector is run for each incident. The Incident Class Selector script selects an incident class based on each class probability. An incident has a 34.9% of being categorized as Class 1 and a 10% of being categorized as Class 5. The script randomly selects an incident ID with the same class after the incident class has been assigned. Each incident ID is associated with an incident's location, duration, and capacity reduction. Finally, a network change event is created with the incident ID selected. An example of the code used to create a network change event is seen in Figure 1. Technically, two network changes occur for each incident, one for when the incident starts and another for when it ends.

The number of daily incidents and severity will vary each time the MATSim scenario is run. We ran three scenarios to illustrate the variance in incidents generated by the script. We recorded the number of incidents and their respective classes. These incidents were mapped using ArcGIS, and Figure 4shows the first scenario results. The presentation includes other figures showing the results of scenarios two and three. In Scenario 1, MATSim generated 13 incidents; in Scenario 2, there were seven incidents; and in Scenario 3, there were nine incidents.

Map

Description automatically generated

Figure 4: Random incidents from one run of the incident generator, .

The project is proceeding on schedule, as shown in Figure 1. The TAC members received a draft literature review prior to the TAC meeting. The literature review revealed many incident simulation studies, and many prior IMT optimization studies, but none that combined the two.

# References

Horni, A., Nagel, K. and Axhausen, K.W. (eds.) 2016 *The Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press. DOI: <http://dx.doi.org/10.5334/baw>. License: CC-BY 4.0

Flötteröd, G. 2016. Queueing Representation of Kinematic Waves. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 347–352. London: Ubiquity Press. DOI: <http://dx.doi.org/10.5334/baw.50>. License: CC-BY 4.0