*Sarah Sun, FHWA TMIP Outreach Manager*

Greetings, everyone. My name is Sarah Sun. I am moderating today's session. All of you are familiar with our webinar logistics. I will mention only a few things here quickly. 1) The webinar will be recorded and the recording will be posted at TMIP.org under Webinars' tab. 2) If you have trouble with Adobe connection during the webinar, simply close Adobe and open it again. 3) We do not have close caption today, but transcripts are in the Download Pod. Please feel free to download it to follow along. 4) If you have any audio problems, please use our toll free number to call in (1-888-675-2535, Participant Passcode: 8344566). That is all the administrative items I have.

I am going to skip the introduction slides on TMIP. If you are not familiar with TMIP, please download it from the Download Pod and read it at your leisure.

I hope you had a chance to listen to our last December webinar on TMIP-EMAT. If you did not, I encourage you to listen to it https://tmip.org/content/tmip-webinar-introducing-exploratory-modeling-and-analysis-tool-tmip-emat. The presentation slides are also in the Download Pod, along with today's presentation slides and transcripts.

**Disclaimer**

***The views and opinions expressed during this webinar are those of the presenters and do not represent the official policy or position of FHWA and do not constitute an endorsement, recommendation or specification by FHWA. The webinar is based solely on the professional opinions and experience of the presenters and is made available for information and experience sharing purposes only.***

Okay, now, let me introduce today's presenters

*Martin Milkovits* is a Principal in the Travel Demand Forecasting business line at Cambridge Systematics. He is the CS Project Manager for Phase 3 (Exploratory Modeling and Simulation Study). Mr. Milkovits has been involved in the design, development and testing of ACES future mobility conditions in regional travel models in several areas, including FDOT District 4, Colorado DOT, and through the recent development of a future mobility tool to evaluate GHG emissions for the City of Boston. Mr. Milkovits holds a BA in Philosophy of Mathematics from Colby College, an MS in Computer Science from Rivier College, and an SM in Transportation from Massachusetts Institute of Technology. He is a member of the Transportation Research Board Planning Applications Committee.

Jeffrey Newman is a Senior Associate in the Travel Demand Forecasting business line at Cambridge Systematics. Dr. Newman has experience in travel demand modeling and forecasting, for both regional and long-distance travel. He is the lead designer and developer of travel forecasting tools, including risk analysis methodologies, for the California High-Speed Rail Ridership and Revenue Forecasting model. Dr. Newman received a Doctorate in Civil Engineering from Northwestern University, as well as a Master of Public Administration and a Bachelor of Science degree in Policy Analysis from Cornell University. He serves on the Transportation Research Board Demand Forecasting and Aviation Economics committees, as well as on the Zephyr Foundation's Software Badging project management group.

**Recap of TMIP EMA Research History**

Before I turn it over to the project team, let me briefly recap the history of TMIP Exploratory Modeling and Analysis (EMA) research to put today's presentation into context. I am going to briefly talk about how the research got started, where it is now and where it is going for the next 15 months and beyond.

*How It Started* - I first learned about Exploratory Modeling and Robust Decision Making (RDM) from a 2015 TRB Annual Meeting workshop presentation that was given by Martin Wachs, who used to work at RAND. I started to read about deep uncertainty, EMA and RDM. After my initial research, I thought this could be a good addition to the TMIP Transportation Modeling and Analysis Toolbox (the Toolbox) https://tmip.org/library/toolbox, which was started in 2013.

The research has been going on for about 4 years now. In 2016, TMIP funded phase 1 of the EMA research. Phase 1 was not to carry out a full EMA investigation, but to test water by using the approach for a limited range of scenarios, investigating the possibility of using integrated activity based model (ABM) and dynamic traffic assignment (DTA) model for EMA. You can read the report at TMIP website https://www.fhwa.dot.gov/planning/tmip/publications/other_reports/feasibility_study/.

We have learned a lot in phase 1 and discovered that a lot of more work needs to be done in order to use EMA in transportation planning practices. Phase 2 followed, aiming to address some of the challenges encountered in Phase 1. However, due to funding and contract time limitation, some of the critical elements still cannot be fully addressed in phase 2, the project team (which are RSG/Caliper) made some recommendations for future research.

Based on the recommendations and lessons learned from phase 1 and phase 2, TMIP decided to develop an exploratory modeling and analysis tool (TMIP-EMAT) in the TMIP Exploratory Modeling and Simulation Study project (current phase 3).

Model run time was a big issue for EMA in phases 1 and 2. So the CS project team for phase 3 has enhanced the meta model from its California High Speed Rail project and incorporated it into TMIP-EMAT. Also, TMIP-EMAT allows users to use sketch planning or strategic models as core models to reduce model run time when performing EMA.

Another recommendation from phase 2 is investigating the best visualizations for explaining the EMA results. The RSG/Caliper team noted that this was especially important when presenting results of unknown input distributions. You could read the full phase 2 report at TMIP website https://www.fhwa.dot.gov/planning/tmip/publications/other_reports/model_impacts_cavs/. So, in phase 3, the visualizer from VisionEval is enhanced and incorporated into TMIP-EMAT.

To address the recommendation of improving the ease of using EMA process, TMIP-EMAT leverages on Jan Kwakkel's EMA Workbench, and the EMA process is being automated as much as possible, ranging from the experimental design, model inputs to visualizing the analysis results, as you will see in today's tutorial given by Jeff Newman, who is the architect of TMIP-EMAT.

*Where It Is Now* - As you know from last December webinar, we have 5 tasks in phase 3. In task 4, which was completed this October, San Diego Association of Governments (SANDAG), The Greater Buffalo-Niagara Regional Transportation Council (GBNRTC) and Oregon Department of Transportation (DOT) have beta tested TMIP-EMAT. They have been very helpful and provided us with a number of feedbacks, which will be addressed in task 5. In fact, one of our beta testers, Alex from ODOT has graciously agreed to present their experience in one of TMIP webinars next year. Task 5 started on Nov. 1, we are in the process of completing the work plan. Phase 3 is set to complete by February 2021.

*Where It Is Going* - In parallel with task 5 of phase 3 (Exploratory Modeling and Simulation Study), TMIP just started phase 4 of the EMA research. Phase 4 research is a joint effort with our Planning Oversight & Stewardship Team, focusing on integrating EMA into transportation planning process using TMIP-EMAT. Harlan Miller and I had our kickoff meeting with the RAND project team this past October.

That is a brief recap of TMIP EMA research. The RSG/Caliper project team presented phase 1 and phase 2 at “Beyond Scenario Planning – The Role of Models with Deep Uncertainty Workshop” on May 14, 2017 TRB Transportation Planning Application conference. CS team has been presenting parts of phase 3 such as scoping, metamodel and some visualizations at different venues such as various TRB conferences and agencies/universities. I hope this recap gives you a holistic view of TMIP EMA research.

*Acknowledgement* – Now I would like to take this opportunity to thank the beta testers, the project teams of all four phases, who share my enthusiasm about EMA and RDM, and thank Jan Kwakkel for making his EMA Workbench an open source. Last but certainly not least, I would like to thank Gloria Shepherd (my Associate Administrator), Ken Petty (my Office Director) and Brian Gardner (my team leader) for the financial support they have been giving to TMIP, especially my office director, Ken Petty who has always been championing TMIP. Without their funding support, there would be no TMIP or TMIP EMA research. Each and every one of those I mentioned today contribute to the success of TMIP EMA research so far.

*How You Can Help* – When you listen to today’s webinar, I hope you would keep an open mind. I think TMIP-EMAT can be a useful tool to help managing uncertainties and reducing risks, and facilitating efficient and effective transportation planning. The tool can also be used to enhance transportation model development practices as well. To ensure successful completion of TMIP EMA research, we need your help. If you are researchers, think about how you could leverage on TMIP-EMAT to further EMA research. If you are agency staff, start to use TMIP-EMAT in your modeling and analysis, and give us your feedbacks. We welcome any constructive comments and suggestions that you may have to help us prioritize activities in task 5 of phase 3 and beyond. Please send your feedbacks to Sarah.Sun@DOT.Gov with the subject line “TMIP-EMAT”.

With that, I will turn it over to CS project manager, Marty. Marty, please take it over.

*Jeffrey Newman, Senior Associate in the Travel Demand Forecasting business line at Cambridge Systematics*

**Getting Set Up**

In this tutorial, we will walk through setting up the Road Test example model to demonstrate how to get ready for exploratory modeling using TMIP-EMAT.

The Road Test model is a completely fictitious model representing traffic flow on a single highway link. The model is implemented as a python function with number of arguments defining the parameters of the BPR function, traffic volumes, infrastructure and financial investment choices, and more. It returns a number of output performance measures including travel times, the value of time savings, amortized investment costs. Documentation for the function is included on the TMIP-EMAT website, under Examples > Road Test > Core Model Definition.

One of the first things we need to do to use a core model with TMIP-EMAT is to define a model scope. We'll use a YAML file to set that up. The model scope includes information about all of the inputs and outputs of the core model. Inputs can be declared as constants, as policy levers that we can control, or as exogenous uncertainties that are outside our control. For each input we declare either a range or a set of discrete values that are possible, and for uncertainties we can also optionally declare a probability distribution. If we don't pick a distribution, a uniform distribution is assumed – this doesn't have any impact on the actual operation of the TMIP-EMAT tools, but it does impact how we'll interpret the results – if we don't put probabilities in, we can't get probabilities out.

The Output performance measures don't even need to have a range declared, as we'll discover that when we run the model.

Along with the model and the scope, we can set up a database to store model results. It isn't too important to have a database for a simple model like the road test, but for more complex models that take a while to run, the database of results is much more important. TMIP EMAT contains a simple database structure set up in SQLite, which is convenient because it requires no setup and no server… it just links to a file and its ready to use instantly with a one line command like this. Here, blah blah is the name of the file where the database will be stored.

Once you've got everything set up, you are ready to run the model. Check out the next video to see how to do that.

**Running the Model**

In this tutorial, we'll walk through running the Road Test example model to demonstrate how to get started using exploratory modeling with TMIP-EMAT.

You can run a model manually by explicitly defining every combination of inputs that will be evaluated, but the real power of TMIP-EMAT comes from running the model using automatically created experimental designs. There are several different ways you might want to do this.

This first is a univariate sensitivity test. In this design, TMIP-EMAT will create a series of experiments hitting the extreme values of each input parameter one at a time, pivoting off a common baseline set of inputs. This'll be a very familiar approach for many transportation modelers, as this kind of sensitivity testing is great for diagnostic analysis and to check for bugs – you've probably been doing this with your model manually before. Running TMIP-EMAT like this is great for the same reasons, to check if your core model and any code used to connect it to the TMIP-EMAT API is running without obvious bugs or other problems.

The second common experimental design used in exploratory modeling is the Latin Hypercube. This design selects a random set of experiments across multiple input dimensions, to ensure "good" coverage of the multi-dimensional modeling space. A Latin hypercube design is what you want if you are running experiments to be able to construct a meta model.

Lastly, you can create a Monte Carlo design, which is just a series of totally independent random experiments. You might choose this design for risk analysis, using a very large number of experiments, to get a good picture of the distributions of outcomes with a design that's simple to explain to stakeholders.

Regardless of which of these designs you want to create, TMIP-EMAT can create the design automatically. Just import the `design_experiments` function, pop in your scope, the sampler type to use, and – for LHS and MC samplers -- the number of experiments you want. There's a lot more this function can do too, check out the documentation online if you're interested.

For this example, we'll just create a Latin hypercube design with ten samples per factor.

The design that comes out is a pandas DataFrame. If you're familiar with Python, you'll know you can take this and do all sorts of interesting analysis directly with it. But the most important thing we're going to do right now is just run these experiments. We can do that with the `run_experiments` method from our model object. We can just give it the design we just created, and here we go…

[click run]

Depending on how many experiments you've defined and how long it takes your model to run, this might take a while. To make good use of the exploratory modeling tools, you're going to need to run some version of your model a lot – hundreds or thousands of times. If your model is big or slow, like a full scale travel demand model, TMIP-EMAT can help with that – go check out the meta-model tutorial video. But the road test model runs fast, so we're already done with 5000 model runs.

Once the experiments are complete, you'll be ready to move on to some visualizations.

**Scatter Plot Matrix and Feature Scoring**

In this tutorial, we're going take a look at using a scatterplot matrix to visualize experimental results. A scatterplot matrix is exactly what it sounds like: it's a grid of scatter plots, each displaying two dimensions of a multidimensional space.

To demonstrate this, we'll pull up a set of experimental results from the road test model using the `read_experiments` method, which will load up our prior model run results that we saved in the attached database.

A full scatterplot matrix might plot every dimension of data against every other dimension – this is the default for a lot of tools. But a more concise way to display one for exploratory modeling is to have a row of plots for each performance measure and a column for each input parameter. We can create something like this automatically using the `display_experiments` function in the `analysis` package. We give it the scope, so that it knows what the inputs and outputs are, plus the experimental results, and we'll get a neatly formatted scatterplot matrix to review. By default, the uncertainties are red and the policy levers are blue, so it's convenient to see which is which.

[ Looking at this]

The figures we get can give us a pretty good understanding of the relationships between the inputs and outputs. For example, if we look at this first row here, we can see that the primary driver of the no build travel time is the input flow. It's got a pretty strong but not linear relationship. There's also a little bit of a relationship here in the alpha and beta parameters, where the highest travel times are linked to the high parameter values, but it's clearly much less important than the flow. Everything else is pretty much random dots without any visible relationships. This make sense – the no build travel time shouldn't be responsive to changes in the build plan.

If we look at the build travel time, we can see input flow is still kind of important, but the relationship is less distinct, while now we're seeing that the expansion amount is also pretty important. It's also notably heteroskedastic – if the expansion is large, the variance on travel time is much lower than if the expansion is small. All of this makes sense.

These plots show a lot of details – for some purposes, maybe too much. We can use a tool called "feature scoring" to distill each of these plots down to a single number, which indicates how important each input is in determining the output. To do so, we just import the `feature_scores` function from the analysis package, and give it the scope and the data. We get an output like this, which is the same matrix of inputs and outputs we saw in the figures, but with just a summary importance number for each cell. It's color coded to highlight the most important input for each output in yellow, and other moderately important inputs in shades through green and blue. It's less detailed information than before, but easier to digest the entire matrix in one go – this really highlights that the flow and the amount of capacity expansion are really the important factors for almost everything, and many of the rest are really just not that big of a deal.

Everything so far looks fine, but let’s take a look at what we might see if something is seriously broken. The road test example includes an alternative scope, that uses a `lane_width` policy lever connected to a miscoded part of the model. We’re going to be able to choose lane widths between 8 and 12 feet, and you’d think that wider lanes would lead to more capacity. But as it turns out in the code, any deviation from exactly 10 feet results in massive extra delays. Let’s see what happens when we run this through our visualizer.

In this first row, everything looks great, but the coding error doesn’t impact the no build travel time. When we look at the next row, we should readily notice a problem way at the right, where we can see a distinct non-monotonic response function we were not expecting. Well, we were expecting it to cause this is an intentional error in the demo, but if this was a real model we wouldn’t have expected to see a “V” here. This is a big old red flag for the analyst, and definitely calls for someone with knowledge about the details of the core model to dive in and figure out what’s wrong.

Assuming you don’t have any fundamental problems like this, you’re probably ready now to move on to the interactive explorer.

**Scenario Discovery with the Interactive Explorer and PRIM**

In this tutorial, we're going take a look at using the interactive explorer to visualize experimental results. Unlike static visualizations, the interactive ones we're going to look at here require you to be running in a live Jupyter notebook, because we're going to have Python running behind the controls to update things on the fly.

For this explorer, we need to have pre-populated a set of experiments. We've got that done for the road test example model already, so I'm just going to load them from the database like this…

Then, to launch an explorer, we just import it from the "analysis" package, and create it like this, using the scope and pre-populated data.

The easiest way to get going, especially if you don't have too many inputs and outputs, is to just use the "complete" visualizer, which pops up an entire interface similar to that provided by the VisionEval tool. This'll create a set of histograms illustrating the data we put in. There is one histogram for each policy lever, uncertainty, and performance measure.

Each is accompanied by a range slider or toggle buttons underneath, depending on the data type – if it's categorical there are buttons, and if it's numeric there's a slider.

I can use these controls to select and highlight only a subset of the data, and the figures will all live update across the entire thing, showing me my selection in orange, against the blue background of the entire data. For example, if I want to look at just the experiments where we expand capacity by at least 50, I can slide this up here, and all the figures update. If I take a look at the performance measures now, I can see that my costs are concentrated in the higher end of the range, and there's a good chance my Net Benefits are going to be negative.

I can set the sliders on more than one thing at a time, like this.

I can even set them on my performance measures directly -- if I want to see what conditions lead to positive net benefits, I can drag this here, selecting only the positive values. Looking now back at the histograms above, I can see the orange areas are concentrated on the higher levels of input flow, and to some degree the higher values of time also.

The value of time one is a little more subtle, as the orange bars are all a lot smaller than the blue ones. I can visualize this a bit differently, by creating a kernel density viewer for the value of time plot, by making a new figure like this.

xp.viewers([ 'value_of_time', 'input_flow',], style='kde')

I'll make one for input flow too, so we can see how that changes too. The KDE figure renormalizes the orange areas so they act a little bit more like a probability distribution. Now we can see a lot more clearly how the distribution changes, as positive net benefits are going to be easier to get to if we have a higher end value of time. These new figures are still linked up with the rest of the figures and sliders, so if I go back up here and change my expansion value, let's say I'm willing to go for a number that's a bit smaller, we can see all the figures update together.

In addition to these histograms that are essentially all one dimensional, I can also create a two dimensional viewer like this.

This scatter plot is showing all the same data in blue and orange, but now I can visualize it across two dimensions.  I can pick over here which dimensions I want to see, let's say I want to look at input flow against value of time.  Now I can see the relationships here, and the orange dots in here are still all those experiments where I have positive net benefits and a bit above the minimum capacity expansion.  And I can see I can achieve these scenarios in these orange dots.  I can see here that I can get more wiggle room on the value of time, it can be a bit lower if I have a high flow.

If I switch this over to net benefits, I can actually see a green box appear, that's showing me the setting of the slider up here, right in the figure.

We could play with these for a while, and do all sorts of exploring – but what we have here are all hard limits.  I can set a specific minimum on my net benefits, and look at the universe of scenarios to meet that goal.  But what if I want to flip this the other way, and try to meet the performance measure goal by setting the constraints on the inputs?  TMIP-EMAT's got a tool for that too – it's called PRIM.

To use it, we can use the `prim` method of the Explorer tool.  We'll give this a target, let's say we want to get positive net benefits, so we can write it like this.

Now we've got a PRIM object here, which is going to systematically go through all the levers and uncertainties, and find ways to efficiently maximize the fraction of cases that will meet our goal by imposing a limited number of constraints.

We can see this by opening a tradeoff selector like this.  The tradeoff prim is making is trading coverage, on the X axis here, against density, on the Y axis.  Coverage is the fraction of all cases that meet the target that are inside our constraints, and density is the fraction of all cases inside our constraints that meet the target.

We start down here at the bottom right, with no constraints at all.  This is always going to have coverage of 1.0, because everything is inside the constraints.  It's got a density above zero, because some fraction of all the cases meet the target – in this case, about 28%.  Then we start applying constraints, cutting out some cases in the data.  When we can move the constraints in a little bit and only cut out cases that don't meet our target, we can move straight up.  But often we'll be chopping off a few outlier cases that do meet the target along with a host of cases that don't meet the target, so we shift a little left too.  We can work our way up up up, adding constraints, to make it more and more likely that the cases in the box meet the target.

I can click on any of these points, to instantly set the constraints in the connected explorer to these levels.  Then we can check out what the impacts are on all the other performance measures, click through to see different slices in the two-way viewer, whatever.

**The Parallel Coordinates Plot**

This is a parallel coordinates plot. It is composed of a number of vertical axes, one for each dimension in the set of pareto optimal solutions found by the optimization algorithm in TMIP-EMAT. In this example, we are looking at a set of results from a robust search in our road test example. We can see vertical lines for each of the four policy levers on the left, and a selection of robust performance measures on the right.

Each of the colored lines extending across the figure represents one possible complete policy to implement, which results in outcomes that are Pareto optimal across all of the relevant performance measure dimensions.

For the measures, each axis is oriented so that up is better. For maximization measures like those reporting benefits, this results in a normal axis with bigger values at the top. For minimization measures, like those reporting costs, the axis is flipped with the smaller numbers at the top, because smaller is better. This makes it easier to visualize tradeoffs, as the lines for two solutions will cross each other. In fact, that is one way to define what makes a set of pareto optimal solutions — every solution line running across this figure will cross over every other solution line at least once. For any possible solution that doesn't cross another, one of that pair of policies is dominated, and removed from the results during optimization.

Right now, the colors are determined by the "interest rate lock" lever, which is either true or false. But we can change them to reflect any dimension of the data in this figure, including performance measures. For example, we can change the colors to "Mean Net Benefits". This lets us see more easily how the mean net benefits are related to every other lever and robust measure shown, without needing to trace each line with our eyes across the figure.

Still, even after adjusting the colors, the figure can be a bit busy. If we want to focus on just a particular group of solutions, we can highlight them by selecting a range on a particular axis. For example, if we want to focus on solutions that yield the highest possible mean net benefits, we can click and drag right along that axis to highlight only these solutions.

Now it is much easier to see that maximizing the mean net benefits can be achieved by expanding capacity between about 20 and 40 units, and that this does involve some risk-taking, as the nearly worst case net benefits values fall away from the top of the range a bit. Similarly, the expected travel time savings is not as good as it could be.

If we want to particularly highlight the solutions with the best expected travel time savings, we can add an additional selection constraint by dragging along that axis to select only the best solutions there.

If it's a little hard to see the purple lines now amidst the unselected gray ones, we can change the colorization to "none" which will simply highlight all the selected solutions in red.

Simply clicking right on an axis that has a selection constraint will clear the constraint for that axis, or there is an option to clear all the selection constraints in the view menu as well.

If we find that the figure has too much going on, we can hide some of the vertical axes that are not important. In this example, I can quickly see that PayGo is the optimal debt type in every solution, and I'm not gaining a lot of additional insight into the tradeoffs by showing it in the figure, so I can take it out by unchecking this box here. Let's remove the amortization period too. There are options in the view menu to show or hide all the levers, uncertainties, or performance measures as a group, which can be handy if you've got  a lot of performance measures available but you want to focus on just a few.

If you want to save the current figure you are looking at as a PNG image that you can then embed in a report or a webpage to share with others, you can do so by clicking this camera button up here.