A Systematic Comparison for Consistent Scenario Development Using Microscopic Simulation Software

Date: December 17, 2024

Presenter: Abhilasha Saroj

Authors:

Abhilasha Saroj[1], Guanhao Xu[1], Yunli Shao[2], Ross Wang[1]

[1]Oak Ridge National Laboratory;
[2]University of Georgia

ORNL is managed by UT-Battelle LLC for the US Department of Energy

Outline

- Objective, Background, and Motivation
- Overview of Real Twin Tool
- Gaps in Literature
- Systematic Approach for Comparison across Microsimulation Software
- Case Study: Comparison of Scenario across Microsimulation Software – Downtown Chattanooga
  - Scenario Development
  - Scenario Performance Evaluation
  - Case Study: Comparison of Scenario across Microsimulation Software
- Conclusion and Future Work
- Traffic Simulation Use Case for XIL Cosimulation Demo
- Progress in Real-Twin Tool Development

Research Background

Background

- Scenarios for mobility research are studied on different platforms by different researchers; A cross-platform benchmark is needed
- With the rise in the use of machine learning, deep learning, reinforcement learning algorithm-based applications that are trained on simulation scenarios, it becomes crucial to have a consistent scenario that allows the use of the already developed algorithms on one platform to be used on another platform to further improve the algorithms or provide comparisons with newly developed algorithms

Evaluation and validation of mobility systems

Objective

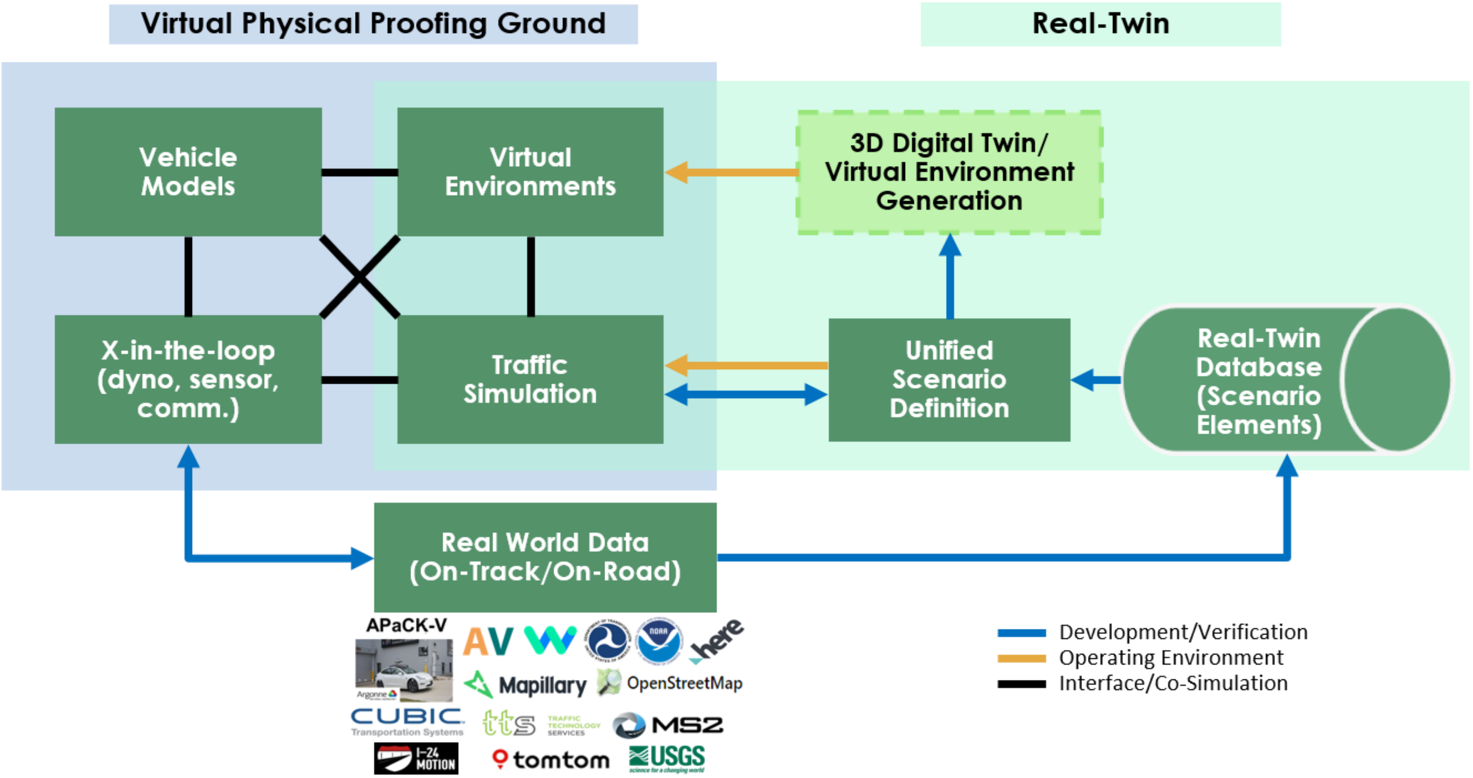
This study presents a systematic approach to compare the closeness of traffic scenarios developed to study vehicle and traffic control strategies for improved mobility and energy consumption on different microsimulation software platforms.

Research Motivation and Objective

Real-Twin Tool Development

- Provides a realistic scenario elements and attributes generation capability that ingests real data
- Provides a twin for analyzing decarbonization opportunities and evaluating mobility objectives

Real-Twin for Realistic Scenario Generation for Standalone Traffic Simulation Studies or for X-in-the-loop Studies



Research Objective

This study presents a **systematic approach to compare the closeness of traffic scenarios** developed **to study vehicle and traffic control strategies** for improved mobility and energy consumption on different microsimulation software platforms

Systematic Comparison Approach for Traffic Microsimulation Scenario Across Platforms

Key Gaps in Literature

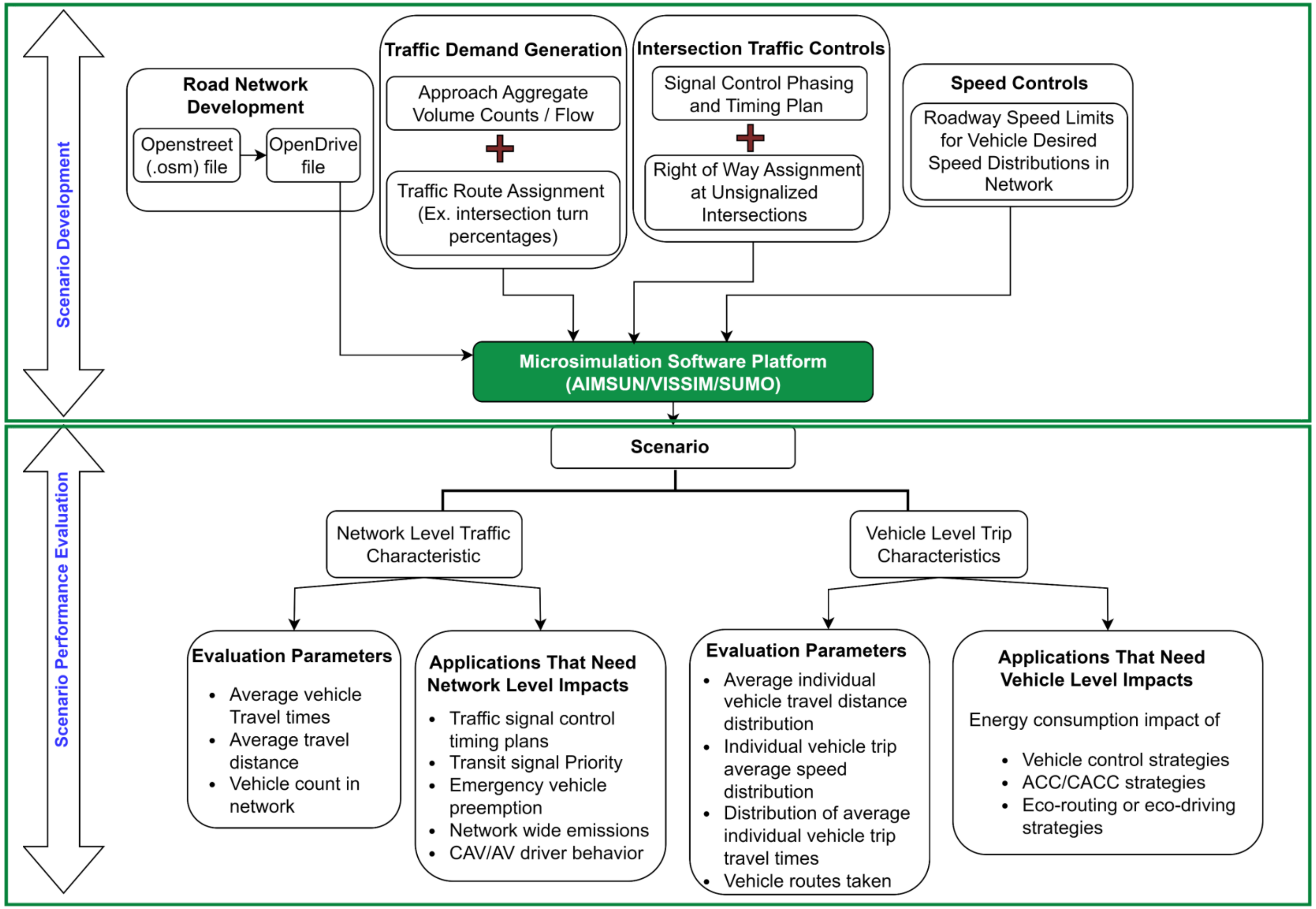
- Existing work **mostly focused on the network-level comparison of overall traffic characteristics, while vehicle-level evaluation is needed** for emerging vehicle and traffic control applications
- Very **few studies included a quantitative and systematic comparison** of results from different microscopic simulation software
- Comparative studies are needed as the **modeling platforms and tools are constantly being updated**

Systematic Comparison Approach for Traffic Microsimulation Scenario Across Platforms

- Systematic comparison approach to investigate the closeness of traffic scenarios across different microsimulation software platforms

- Scenario Development Comparison
  - Uniform inputs to the model
  - Software limitations

- Scenario Performance Evaluation
  - Network level traffic characteristics
  - Vehicle level characteristics
  - KPI for comparison based on application of study



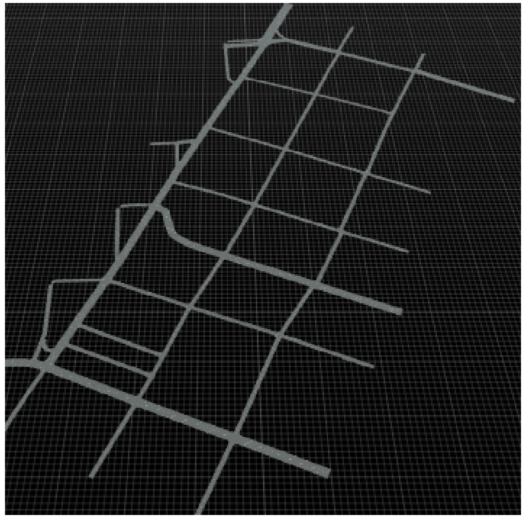
Case Study – Downtown Chattanooga, Tennessee

- Comparison of the network development process and performance measures from the three microsimulation platforms – VISSIM, SUMO, and AIMSUN
- Same scenario was created in the three platforms for a portion of the Downtown Chattanooga network

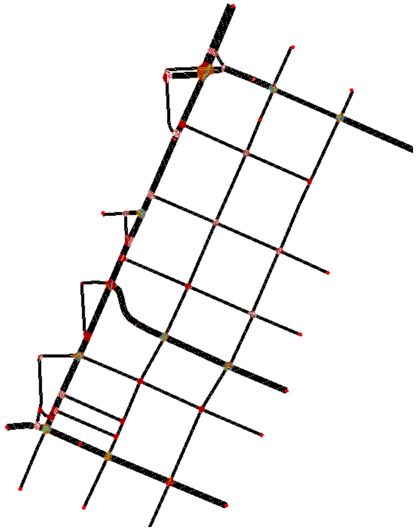
Scenario Development

- Network Development – OpenDRIVE
- Demand Generation
- Traffic Controls
- Speed Limits

ASAM OpenDRIVE®

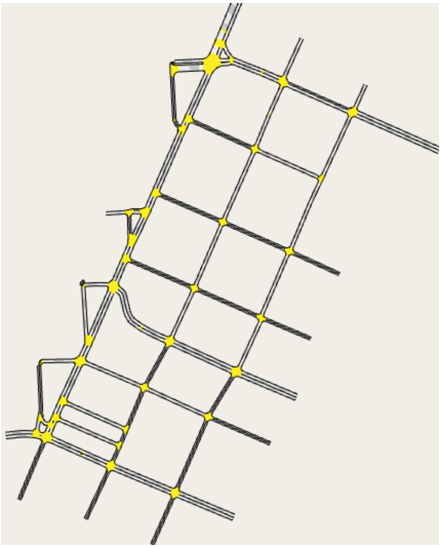


SUMO



VISSIM

aimsun.



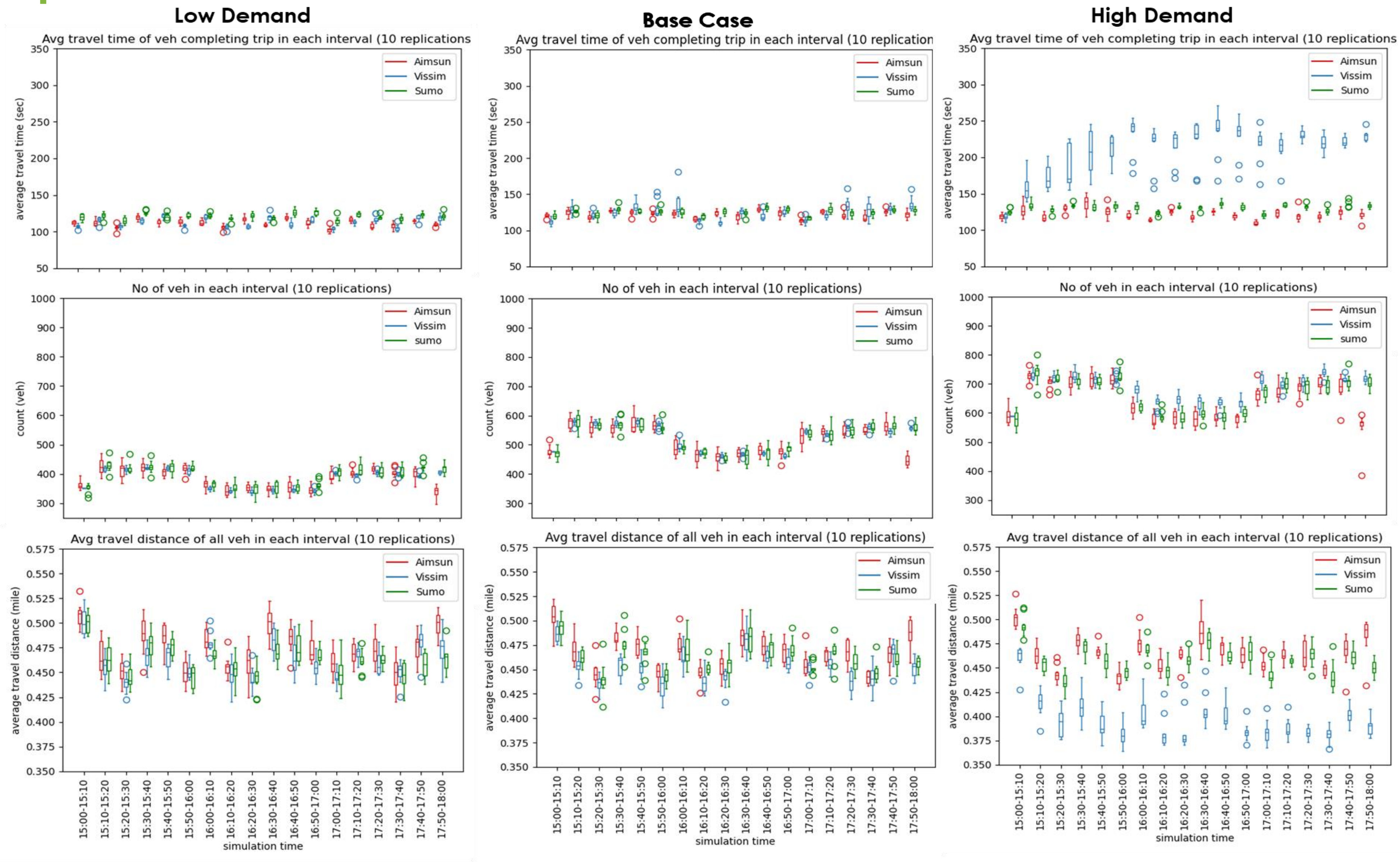
Case Study – Downtown Chattanooga, Tennessee

Network level comparison

- Number of vehicles, average vehicle travel time, and average distance traveled in the time interval

- Boxplots for 10 random seeds

- Network level **performance consistent for three platforms for low and base demand scenario but not for high demand**

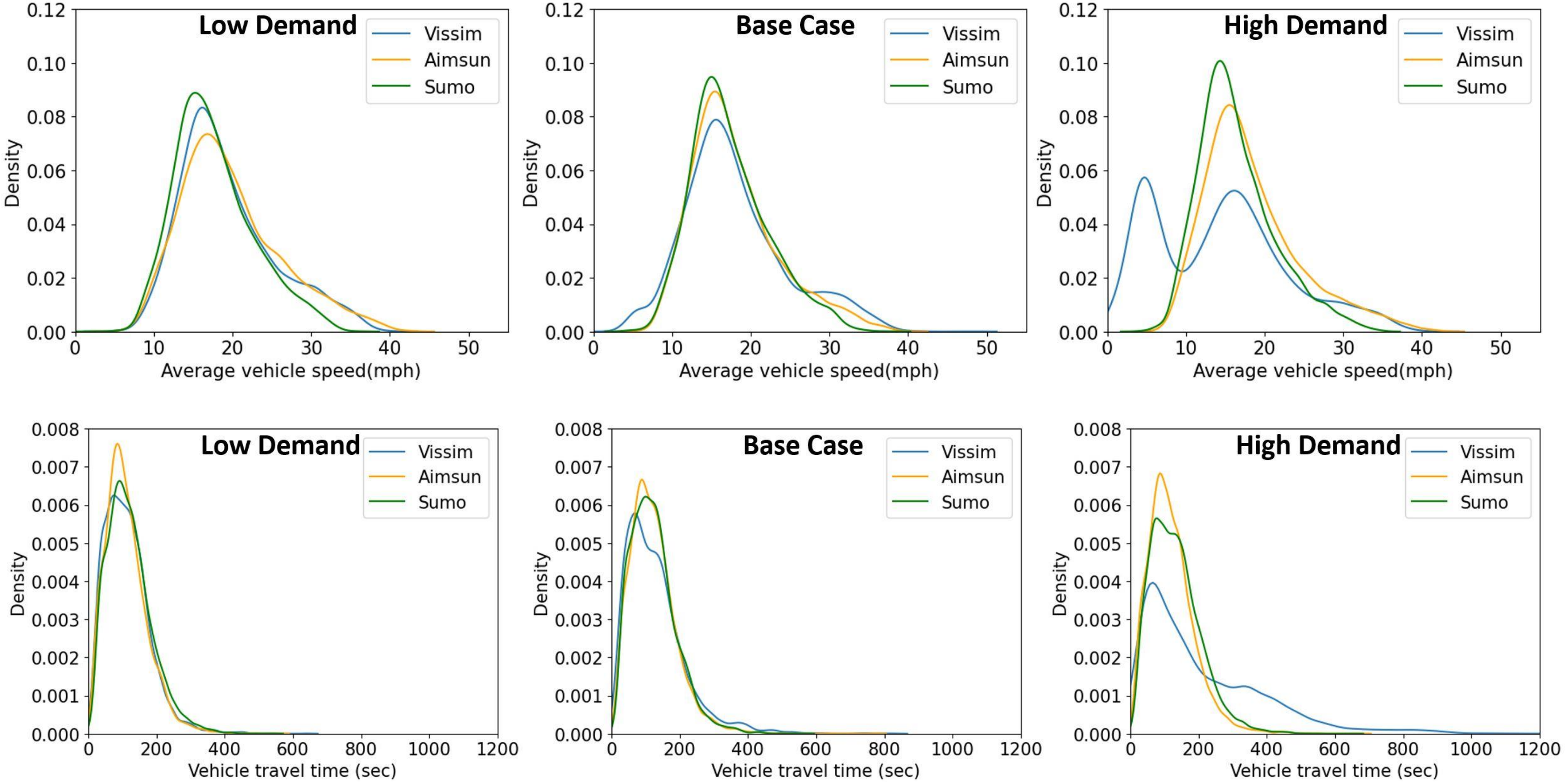


Case Study – Downtown Chattanooga, Tennessee

Vehicle Level Comparison

- Base case and low-demand case show consistent vehicle-level results from the three platforms

- High-demand case: VISSIM exhibit differences in average vehicle speed and average travel time distributions compared to AIMSUN and SUMO



- VISSIM had a higher number of vehicles experiencing lower average speeds under higher demand
- Differences in sensitivity of demand variation on vehicle-level performance in different software indicates a need for calibration to different demand scenarios

Case Study – Downtown Chattanooga, Tennessee

Vehicle Level Comparison

- In simulation implementation, the vehicles were assigned a turn at each junction approach based on the turn ratios

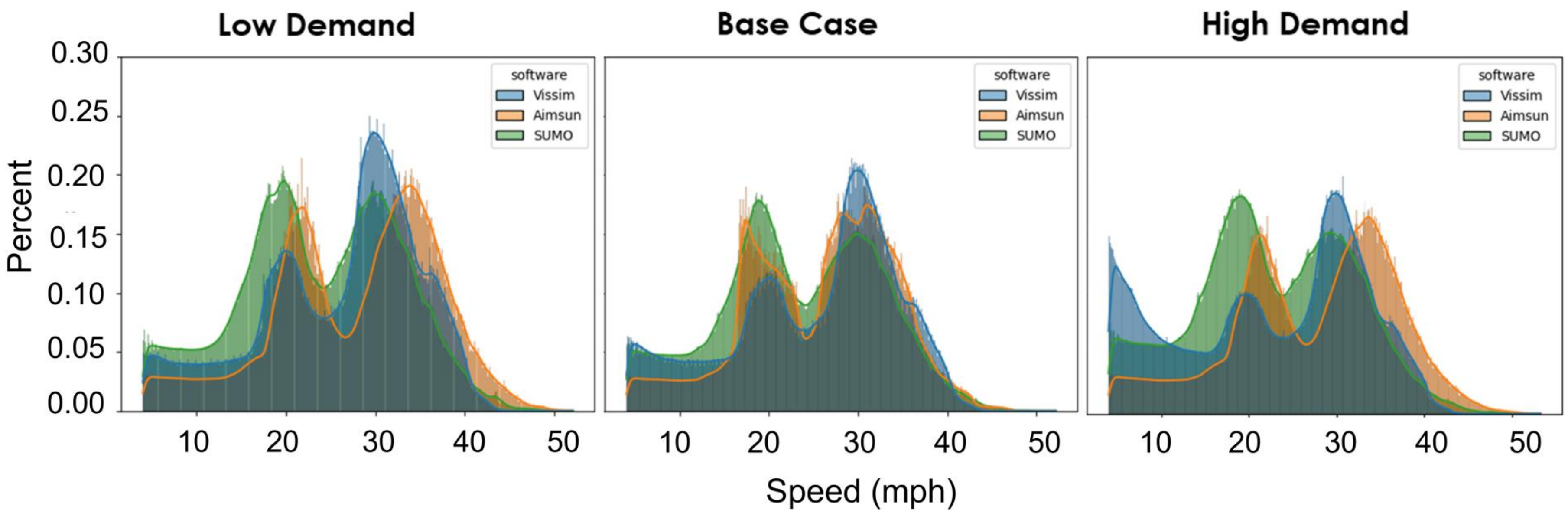
- The five most frequent routes and their orders were identical in the three platforms. However, the routes themselves may not be realistic (e.g., Route 4)

- This routing behavior may not be a concern when studying mobility or energy impacts at the network level where primarily only aggregated traffic performance matters

Case Study – Downtown Chattanooga, Tennessee

Vehicle Level Comparison

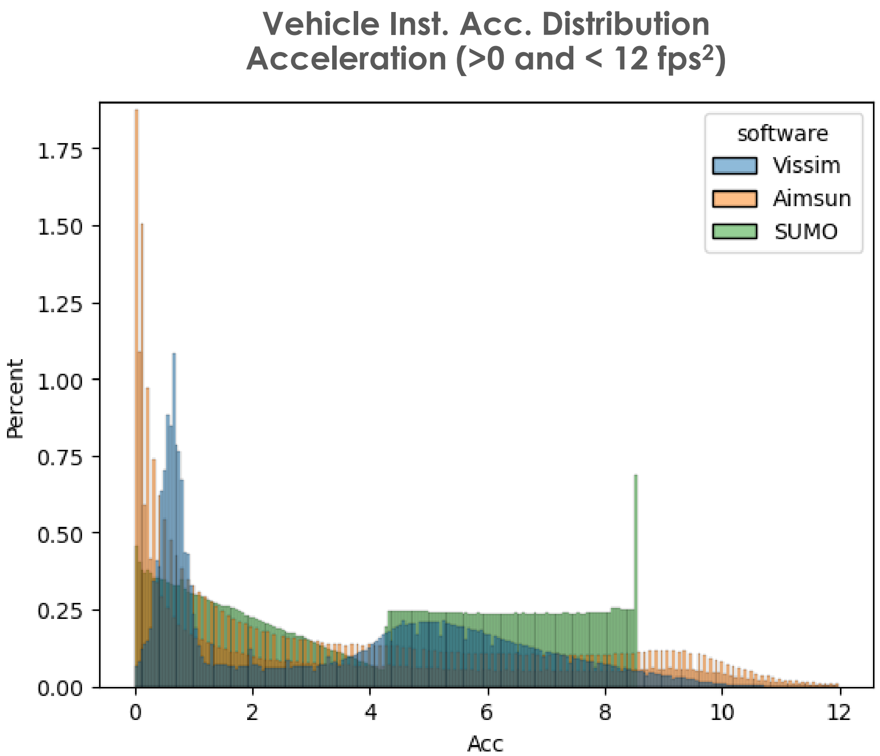
- Higher vehicle speeds observed in AIMSUN for low and high demand cases
- Increment in the frequency of vehicles with lower speeds from low demand to base to high demand for VISSIM

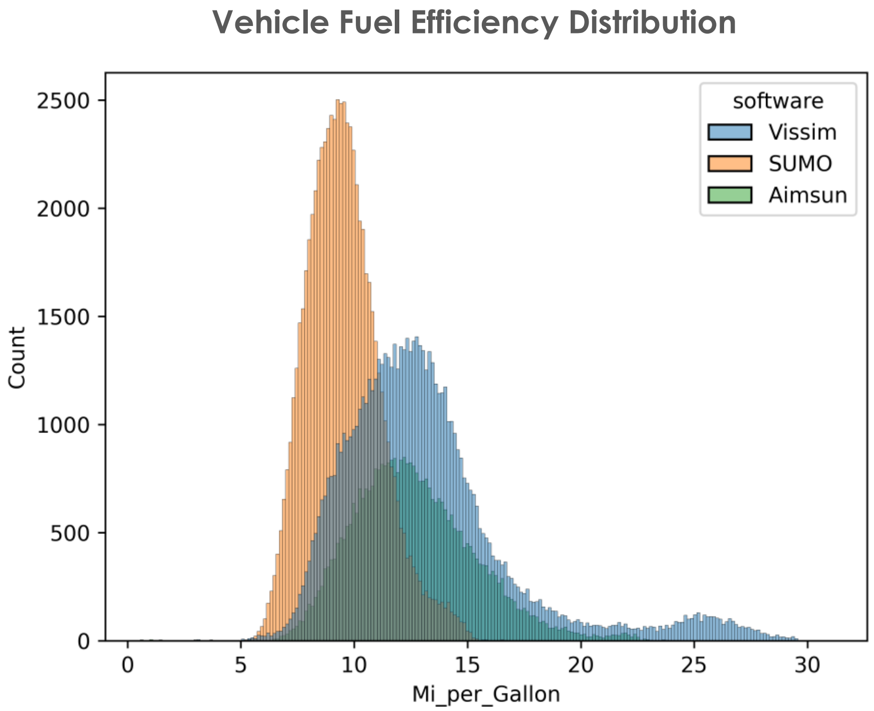


- Calibration of desired speed distribution parameters for different demand scenarios is important for comparable scenario development across different platforms

Case Study – Downtown Chattanooga, Tennessee

Vehicle Level Comparison – Preliminary Investigation for Energy Consumption Impacts





- Although network and vehicle level traffic performance measures similar energy consumption values differ
- Shows the importance of calibrating acceleration-related parameters such as desired acceleration/deceleration distribution, maximum acceleration/deceleration distribution, car following parameters, etc., especially for energy-focused applications

OAK RIDGE National Laboratory | NATIONAL TRANSPORTATION RESEARCH CENTER

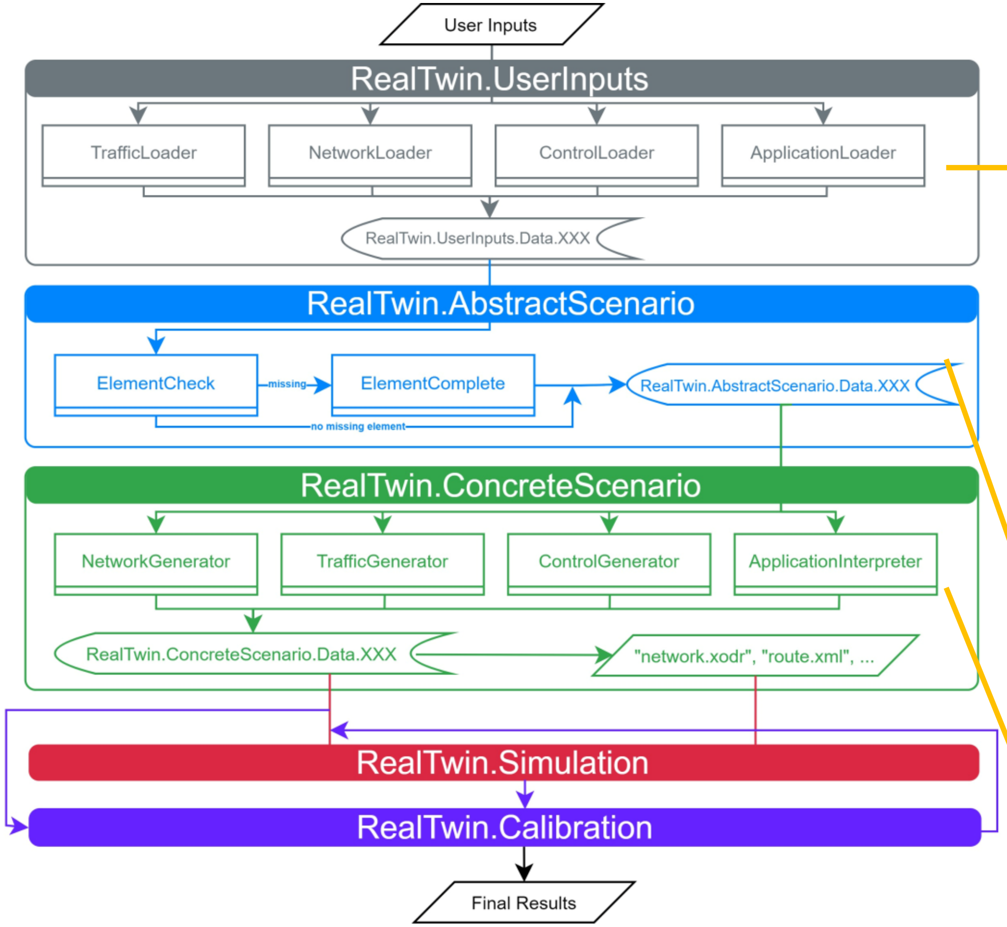
Key Conclusions

- Comparison of scenario development:
  - OpenDRIVE files can be imported into the three software compared in this study
  - Compared to AIMSUN and SUMO, VISSIM has the more comprehensive and complex underlying simulation models and requires attention to further calibrate the network layout

- Comparison of scenario performance evaluation:
  - For the base and the low demand cases, the network traffic characteristics and individual trip characteristics for the three software were consistent
  - Potential and effectiveness of using the scenario development approach followed in the paper to achieve consistent traffic simulation across different microscopic simulation platforms
  - Higher travel times with more variations across different random seeds were observed in VISSIM compared to SUMO and AIMSUN in the high demand case. This indicated VISSIM results in more stochasticity than SUMO and Aimsun
  - Vehicle fuel efficiency distributions in the base case scenario reveals that although conventional traffic characteristics are similar across the software, the fuel efficiency distributions and acceleration distributions differ
  - Need to calibrate driving behavior and acceleration distribution specific parameters for consistent scenario generation for energy consumption-focused applications

Progress on Real-Twin Tool

RealTwin Classes UserInputs, AbstractScenario, and ConcreteScenario to prepare all inputs – network, traffic, controls, and application parameters for the RealTwin.Simulation to generate simulation scenario in Aimsun, VISSIM and SUMO.



UserInput.yaml

Data from User Input File

Traffic:
- Volume
- Turning Ratio

Network:
- Network Name
- Network Bounding Box Vertices Coordinates (latitude, longitudes)
- Elevation Map

Control:
- Signal Timing Plan in UTDF format

Application:
- Application Type. e.g., AV
- Simulator
- Car Following Model
- Car Following Behavior Parameter Values

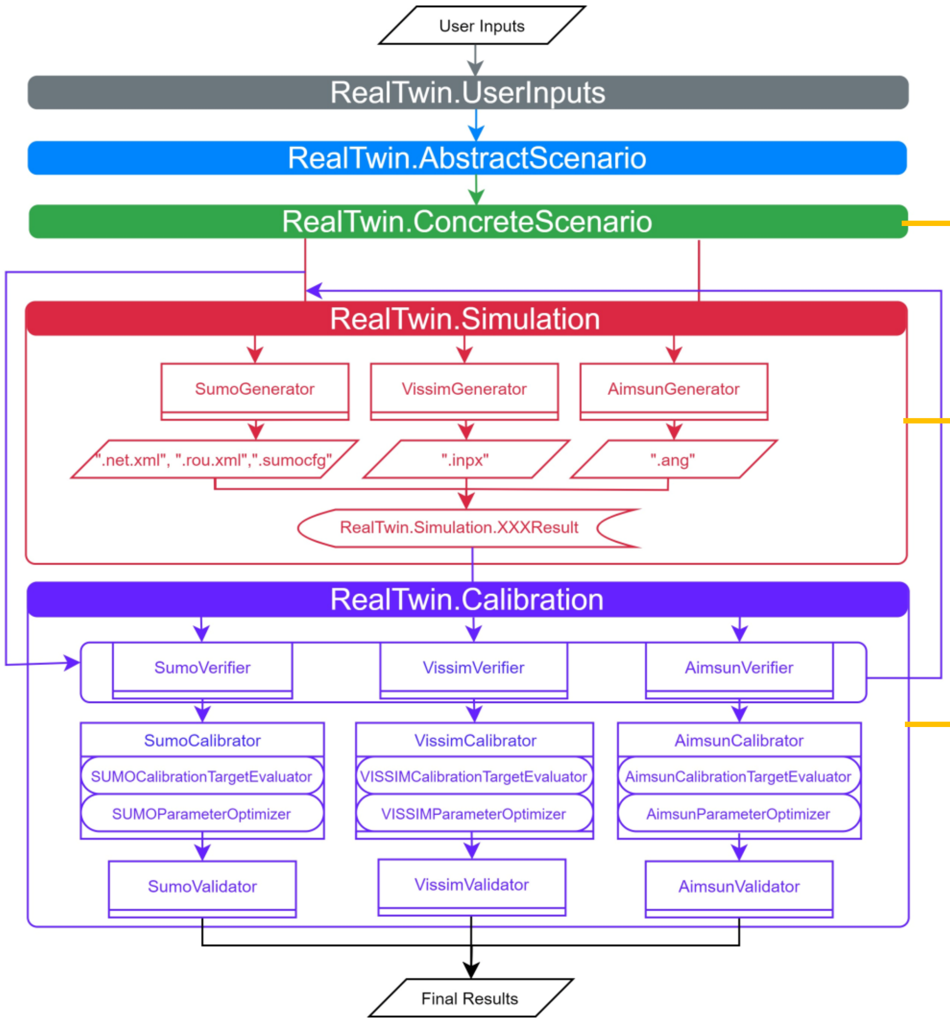
Intermediate Variables (storing user-provided input data)

Intermediate Variables from Traffic
Intersection Name, Latitude, Longitude, Approach, Interval Start Time, Interval End Time, Turn Movement Type, Count

Intermediate Variables (storing scenario elements with ID matched with OpenDRIVE network)

Intermediate Variables from Traffic include OpenDRIVE ID
Intersection Name, Latitude, Longitude, Interval Start Time, Interval End Time, Turn Movement Type, Count, **OpenDRIVE ID**

Progress on Real-Twin Tool



RealTwin Classes `Simulation` and `Calibration` used to generate simulation scenario in Aimsun, VISSIM and SUMO and to calibrate them.

Intermediate Variables
(intermediate variables updated to map with network and include all scenario elements)

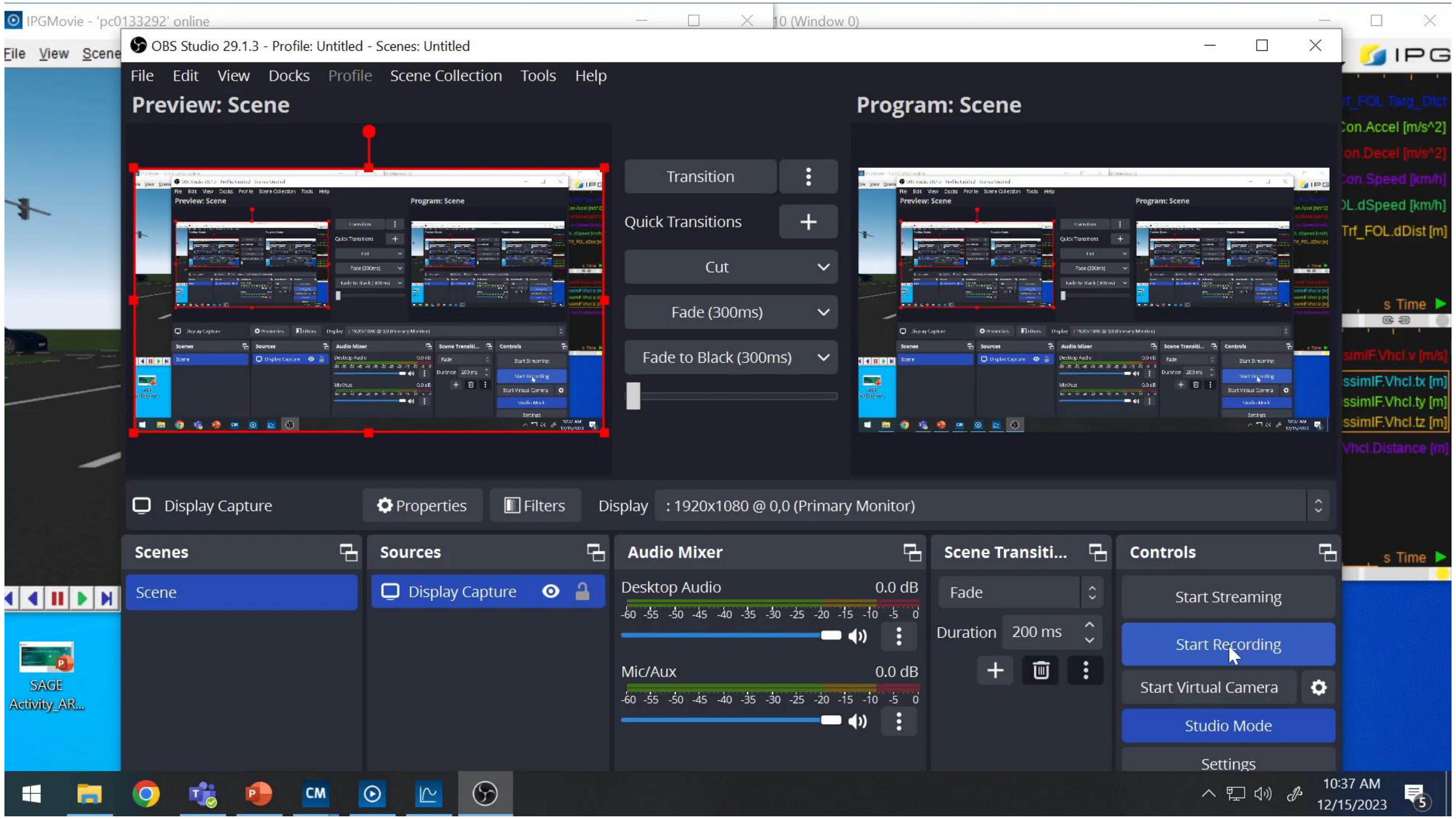
Network, Demand, Controls, and Application Parameters from `ConcreteScenario` used by the **Simulation** to generate scenario using the respective simulator.

Simulated scenario is calibrated in the Calibration module:

- **Verifiers**. Simulation outputs and `ConcreteScenario` intermediate variable values compared to verify simulation inputs.
- **Calibrators**. Calibrators include:
  - `CalibrationTargetEvaluators` to estimate calibration target measures.
  - `ParameterOptimizers` to optimize the parameters to reach the calibration target measure goal.
- **Validators**. Validate the verified and calibrated model with real-world data.

Use Case of Real-Twin for XIL Cosimulation

IPG CarMaker +Vissim Co-simulation



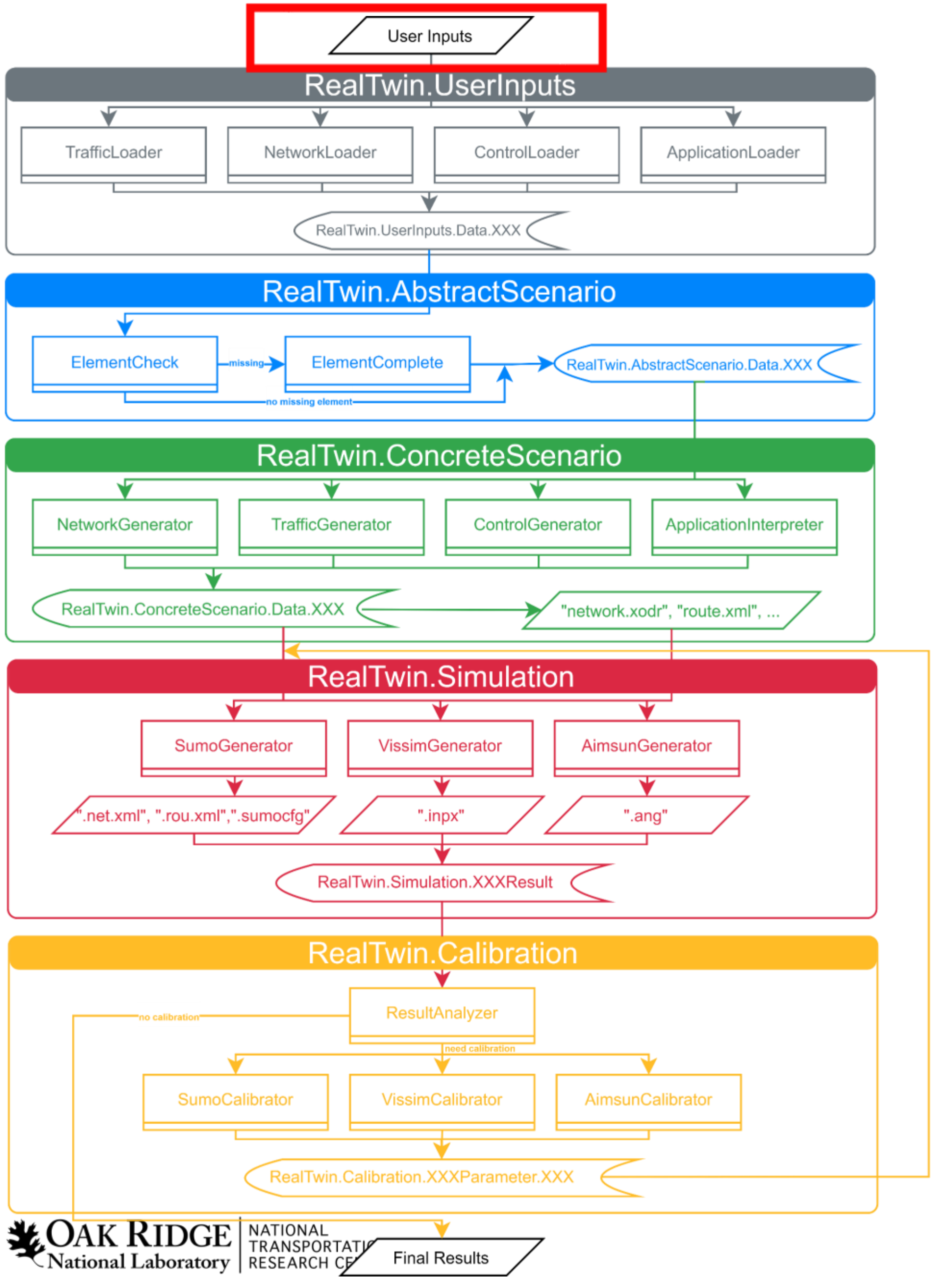
OAK RIDGE National Laboratory | NATIONAL TRANSPORTATION RESEARCH CENTER

Thank you

Questions?

Backup slides

UserInput Template



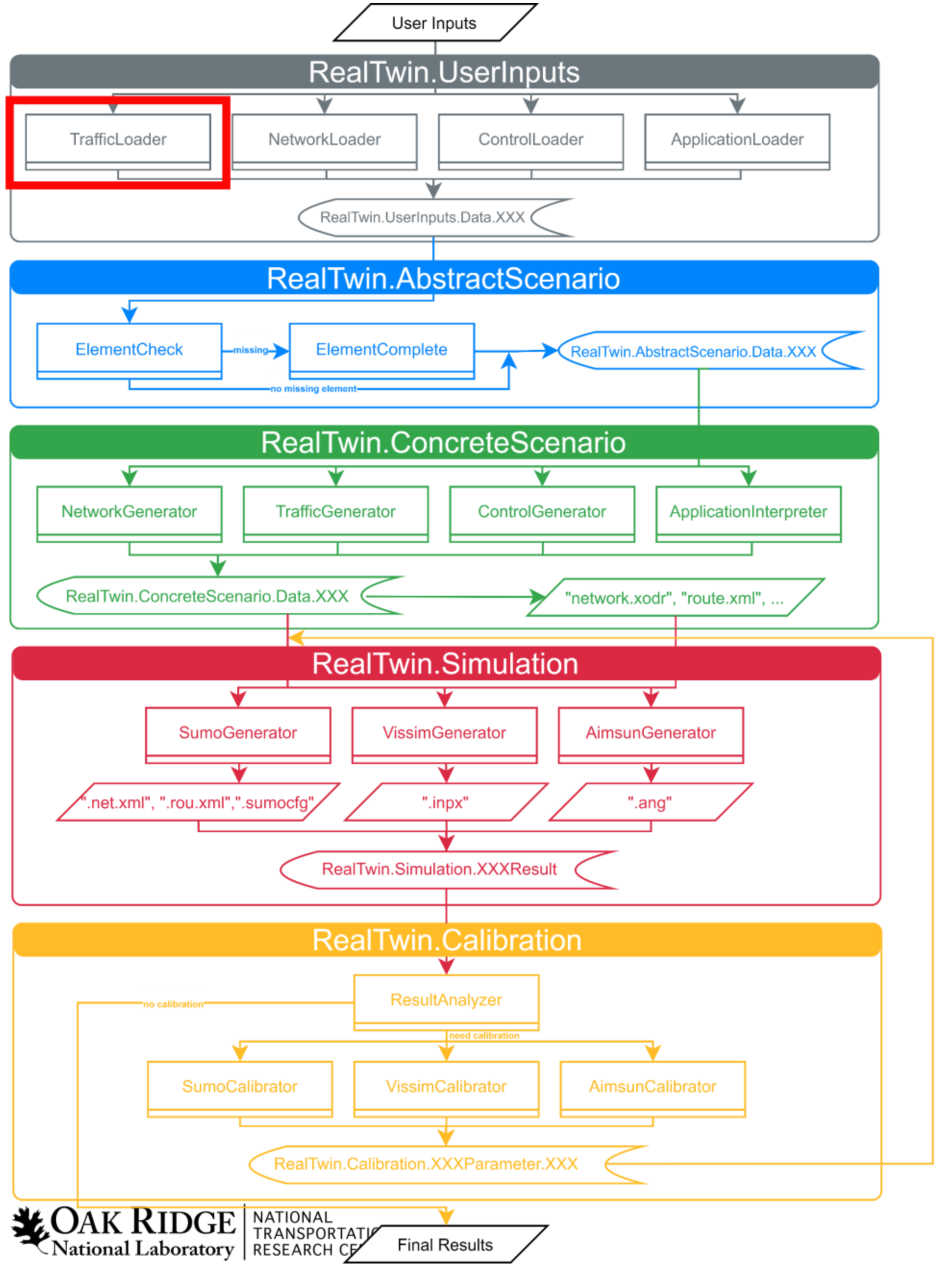
UserInputTemplate.yaml

*UserInputTemplate.yaml - Notepad

File Edit Format View Help

```
Traffic:
    Volume: GridSmart_demand.csv
    TurningRatio: GridSmart_demand.csv
Network:
    NetworkName: chatt
    NetworkVertices: (-85.14977588011192, 35.040346288414916), (-85.15823020212477, 35.04345144844759),
                     (-85.15829457513502, 35.043293338482925), (-85.14986171079225, 35.04018378032611)
    ElevationMap: chatt_elev.tif
Control:
    Signal: Synchro_signal.csv
```

Traffic



UserInputTemplate.yaml

```
Traffic:
    Volume: count.csv
    TurningRatio:turn.csv
```

Intermediate variable

```
pd.DataFrame(data, dtype =
{'IntersectionName':str, 'IntervalStart':int,
'IntervalEnd':int, 'Turn':str, 'Count':int})
```

Intermediate variable(using ID from OpenDrive)

```
pd.DataFrame(data, dtype =
{'IntersectionName':str, 'IntervalStart':int,
'OpenDriveFromID':int, 'Count':int})
```

chatt.rou.xml

```
<routes>
    <vType id="car"/>
    <vehicle id="4.0" type="car" depart="28803.52">
        <route edges="-280 -307 -327 -281 -314"/>
    </vehicle>
    <vehicle id="4.1" type="car" depart="28808.47">
        <route edges="-280 -307 -327 -281 -314"/>
    </vehicle>
```

Traffic

UserInputs

GRIDSMART — Turning Movement Counts - Sum

Intersection 0410) Shallowford Rd & Hickory Valley Rd
Date 7/10/2023

	Northbound				Eastbound				Southbound				Westbound				Unassigned			
	R	T	L	U	R	T	L	U	R	T	L	U	R	T	L	U	R	T	L	U
00:00		9	1		1	19	3		3	3	1			27	1					
00:15		4	1			6	3		6	2	2		1	12						
00:30		2	1			10	1		4	4	1			10	1					
00:45		2				7	4			6	1		1	14						
01:00		4				5	1							15	2					
01:15		2			1	10			1	2				10	1					
01:30					1	8	1		4					7	2					
01:45		3	1			11	1		1	3				7	2					
02:00		3				6	2		2	3				10	5					
02:15	2		1			7	1			2				3						
02:30		1				4				2				6	1					
02:45		2			1	3				3	1			7	1					
03:00		3	1			4								13						
03:15		1	1			6	2				2		1	12						
03:30		2				1			2	1				9	1					
03:45	2				1	8				3			1	16	1					

AbstractScenario

Volume

	index	Inters...	Interv...	Interv...	Turn	Count
0	0	Shallowf...	0	900	NBR	0
1	1	Shallowf...	0	900	NBT	9
2	2	Shallowf...	0	900	NBL	1
3	3	Shallowf...	0	900	NBU	0
4	4	Shallowf...	0	900	EBR	1
5	5	Shallowf...	0	900	EBT	19
6	6	Shallowf...	0	900	EBL	3
7	7	Shallowf...	0	900	EBU	0

Turning Ratio

	index	Inters...	Interv...	Interv...	Turn	Bound	Direct...	TurnR...
0	0	Shallowf...	0	900	NBR	N	R	0
1	1	Shallowf...	0	900	NBT	N	T	0.9
2	2	Shallowf...	0	900	NBL	N	L	0.1
3	3	Shallowf...	0	900	NBU	N	U	0
4	4	Shallowf...	0	900	EBR	E	R	0.043478...
5	5	Shallowf...	0	900	EBT	E	T	0.826086...
6	6	Shallowf...	0	900	EBL	E	L	0.130434...
7	7	Shallowf...	0	900	EBU	E	U	0

lookup table

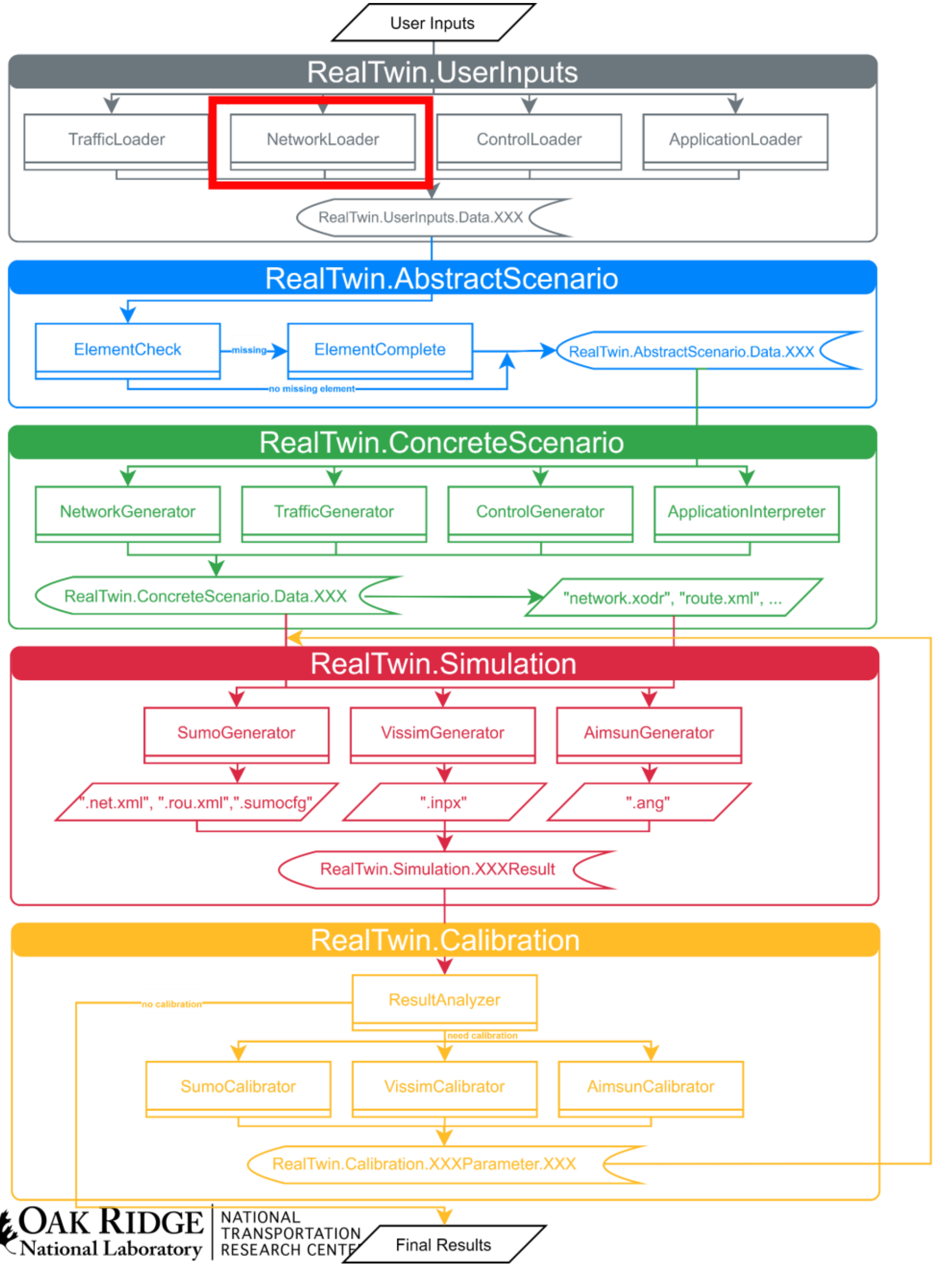
IntersectionName	Turn	OpenDrivi	OpenDriveToID
Shallowford Rd & Hickory Valley Rd	NBR	31	22
Shallowford Rd & Hickory Valley Rd	NBT	31	12
Shallowford Rd & Hickory Valley Rd	NBL	31	42
Shallowford Rd & Hickory Valley Rd	NBU	31	32

ConcreteScenario

	index	Interv...	Interv...	Inters...	OpenDriveFromID	Count
0	0	0	900	Shallowf...	11	7
2	2	0	900	Shallowf...	31	10
3	3	0	900	Shallowf...	41	23
1	1	0	900	Shallowf...	21	28
6	6	900	1800	Shallowf...	31	5
7	7	900	1800	Shallowf...	41	9

	index	Inters...	Interv...	Interv...	Turn	Bound	Direct...	TurnR...	OpenDriveFromID	OpenDriveToID
0	0	Shallowf...	0	900	NBR	N	R	0	31	22
1	1	Shallowf...	0	900	NBT	N	T	0.9	31	12
2	2	Shallowf...	0	900	NBL	N	L	0.1	31	42
3	3	Shallowf...	0	900	NBU	N	U	0	31	32
4	4	Shallowf...	0	900	EBR	E	R	0.043478...	41	32
5	5	Shallowf...	0	900	EBT	E	T	0.826086...	41	22
6	6	Shallowf...	0	900	EBL	E	L	0.130434...	41	12

Network



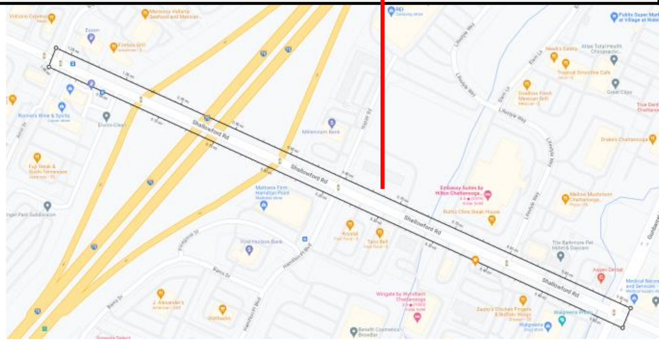
UserInputTemplate.yaml

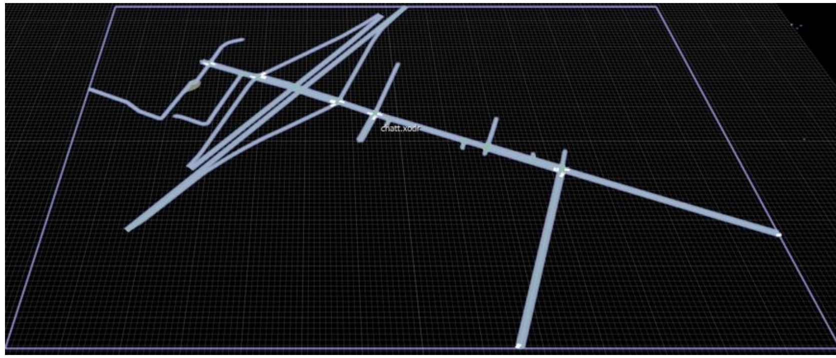
chatt.xodr (raw)

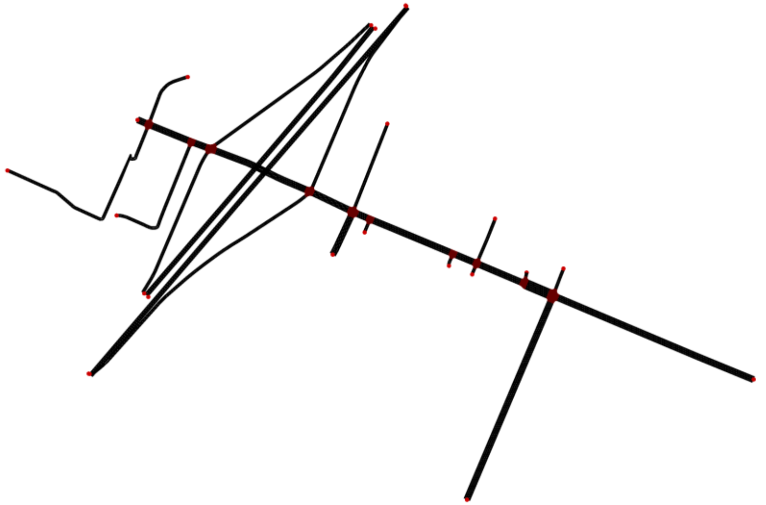
chatt.xodr (with updated elevation)

chatt.net.xml

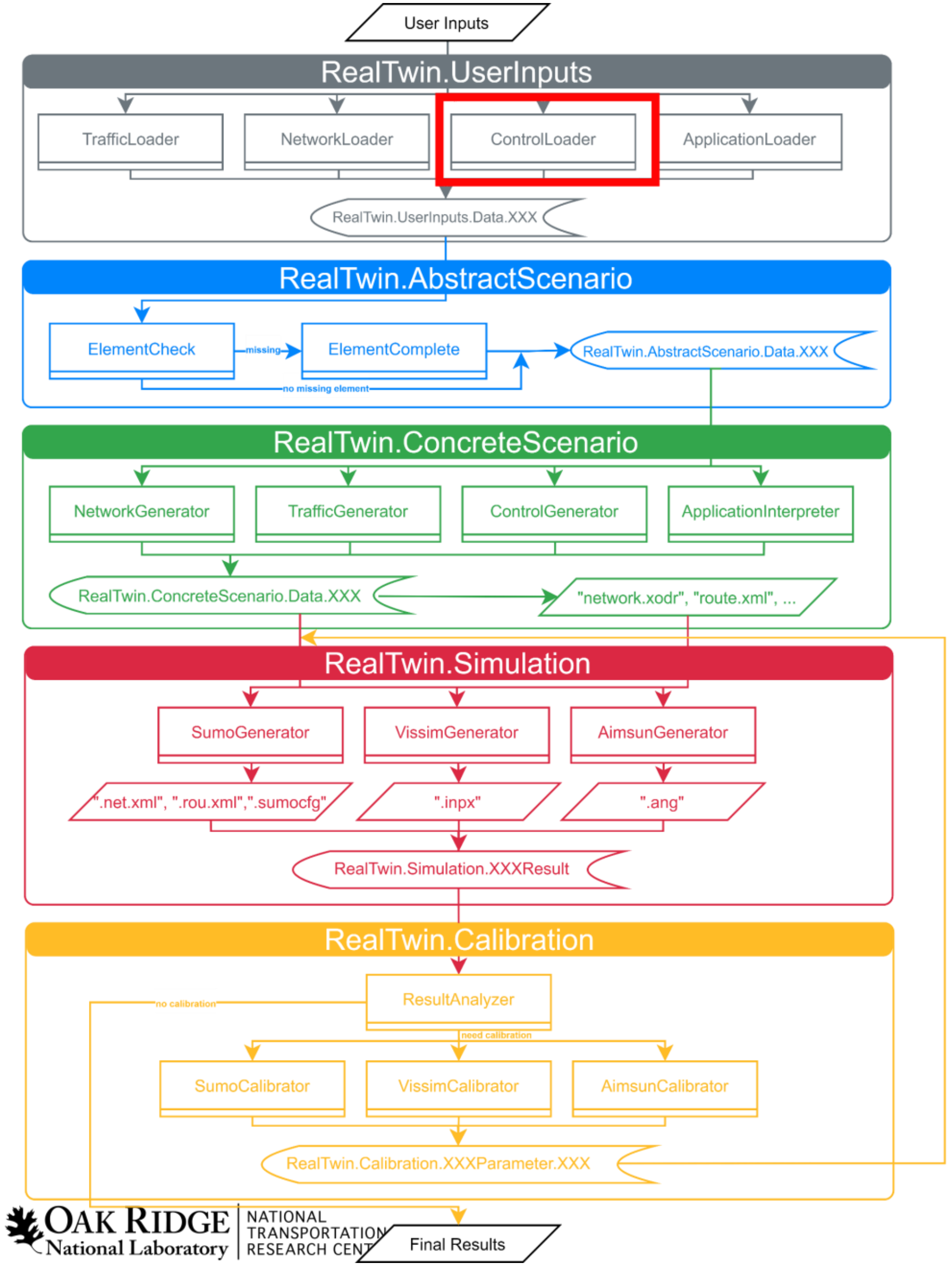
```
Network:
    NetworkName: chatt
    NetworkVertices: (long1, lat1), (long2, lat2),
                     (long3, lat3), (long4, lat4)
    ElevationMap: chatt_elev.tif
```







Control



UserInputTemplate.yaml

```
Control:
    SignalPlan: signal.csv (Synchro Universal Traffic
                            Data Format(UTDF) file)
```

Intermediate variable

```
pd.DataFrame(data, dtype = {'IntersectionID':
int,'PhaseID': int, 'Green': float, 'yellow': float,
'NBL', bool, 'NBT', bool, 'NBR', bool… }
```

Intermediate variable(using ID from OpenDrive)

tlLogic in chatt.net.xml

```
<tlLogic id="8" type="static" programID="0" offset="0">
    <phase duration="8.4" state="rrrrrrrrrrGG"/>
    <phase duration="3.0" state="rrrrrrrrrryy"/>
    ...
    <param key="linkSignalID:0" value="202556079_0"/>
    <param key="linkSignalID:1" value="202556079_0"/>
    ...
</tlLogic>
```