DTALite Interfacing Simulation Platform

Multiresolution Simulation Platform

Booz Allen Hamilton

Arizona State University

TABLE OF CONTENTS

Introduction

This repository contains the DTALite Interfacing source-code, executable file, user's guide, and NEXTA graphical user interface (GUI). This version of DTALite allows external users to embed their own simulation scenarios into DTALite when simulating on the fly, such as adaptive ramp metering, signal control, and work zone management. This customized version of DTALite allows users to simulate and evaluate their traffic management projects without providing all the typically required input by DTALite. The interfacing frequency between DTALite and users' algorithms is adjustable to meet various project needs. The simulation tool is coded in C++ language and is developed by Arizona State University.

DTALite Interfacing

The DTALite Interfacing is an executable file recognized as a .exe extension. DTALite uses a computationally simple but theoretically rigorous traffic queuing model in its lightweight mesoscopic simulation engine. To reduce data preparation efforts, it only requires a minimal set of static traffic assignment data and some time-dependent OD demand pattern estimates. Its built-in parallel computing capability dramatically speeds-up the analysis process by using widely available multi-core CPU hardware.

DTALite's four major modeling components include:

1) Time-dependent shortest path finding, based on a node-link network structure;
2) Vehicle/agent attribute generation, which combines an origin–destination (OD) demand matrix with additional time-of-day departure time profile to generate trips;
3) Dynamic path assignment module, which considers major factors affecting agents' route choice or departure time choice behavior, such as (i) different types of traveler information supply strategies (e.g. historical, pre-trip, and/or en route information, and variable message signs) and (ii) road pricing strategies where economic values are converted to generalized travel time;
4) A class of queue-based traffic flow models that can accept essential road capacity reduction or enhancement measures, such as work zones, incidents, and ramp meters.

DTALite Input Files and Execution

The DTALite simulation platform requires 18 basic input files recognized as .csv and .txt native format and are listed as follows:

1) input_node.csv
2) input_node_control_type.csv
3) input_link.csv
4) input_link_type.csv
5) input_zone.csv
6) input_activity_location.csv
7) input_pricing_type.csv
8) input_demand_type.csv
9) input_vehicle_type.csv
10) input_VOT.csv
11) input_vehicle_emission_rate.csv
12) input_demand_meta_data.csv
13) input_MOE_settings.csv
14) input_scenario_setting.csv
15) DTA_settings.txt
16) ODME_settings.txt
17) input_sensor.csv
18) input_subarea.csv

Users are encouraged to reference the following documentation for more information on the input data structure https://sites.google.com/site/dtalite/.

The DTALite interfacing executable file is labeled 'DTALite_64.exe' and must be located within the same directory as the 18 mandatory input files and other optional supplementary input files. Once the DTALite Interfacing file is executed and the agent output results have been produced, the following 11 output files are produced and recognized as .csv and .bin extension:

1) Output_summary.csv
2) output_multi_scenario_results
3) agent.bin
4) Output_agent.csv
5) Output_ODMOE.csv
6) Output_pathMOE.csv
7) output_NetworkTDMOE.csv
8) Output_linkMOE.csv
9) Output_linkTDMOE.csv
10) output_MovementMOE.csv
11) output_vehicle_emission_MOE_summary.csv

Graphical User Interface (GUI)

The following Figure 1 shows an example of the NEXTA GUI application. The program is loaded from the network reference file labeled tempe_new.tnp which contains references to all 18 mandatory input files.

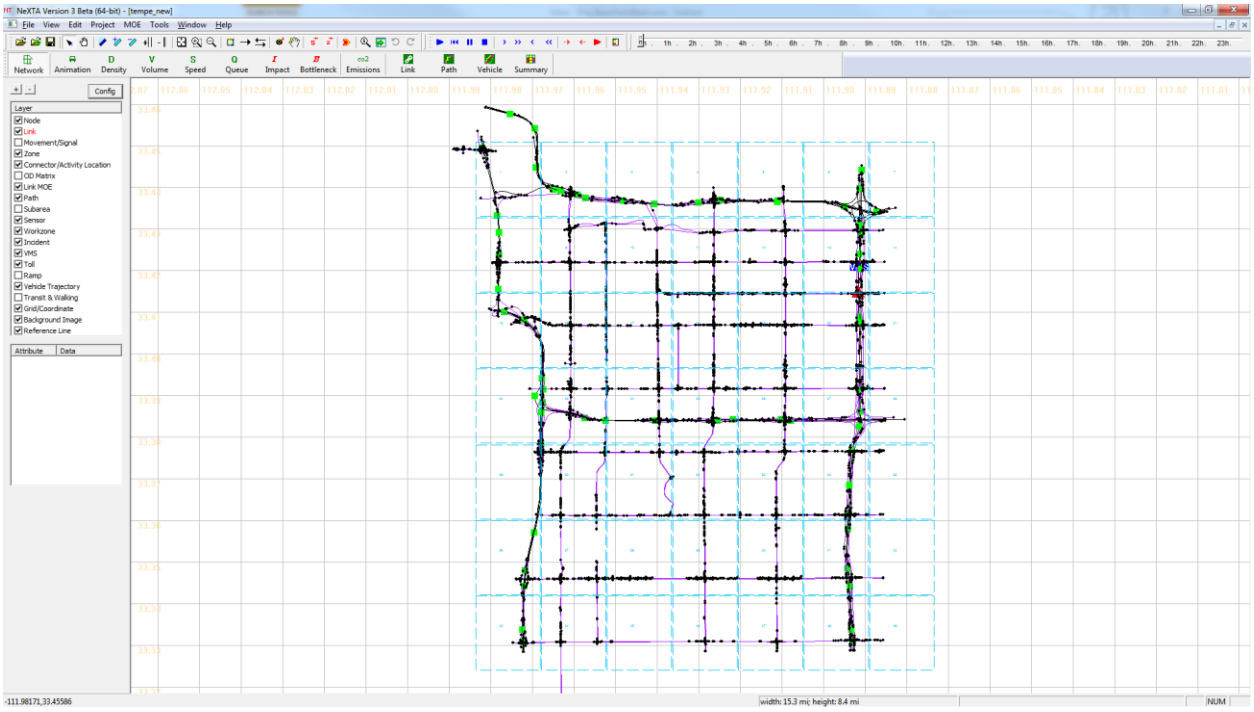

Figure 1: Sample NEXTA Graphical User Interface

There are two methods of executing the DTALite Interfacing traffic simulation. The first is to double click on the 'DTALite_64.exe' executable file with all the necessary input files located within the same folder. The second option to execute the simulation is through the NEXTA GUI by selecting the 'Run Simulation'

button located at the top ribbon. The following option pane for 'Review Simulation/Assignment Settings' shown in Figure 2 below.

There are textboxes to display input files to review the network summary and options to select the simulation performance options for the following: Link Traffic Flow Model; Signal Control Representation; and Traffic Assignment Method. Once appropriate inputs have been selected, users will click on the 'Run Simulation' button within the option box to initiate the DTALite Interfacing program.

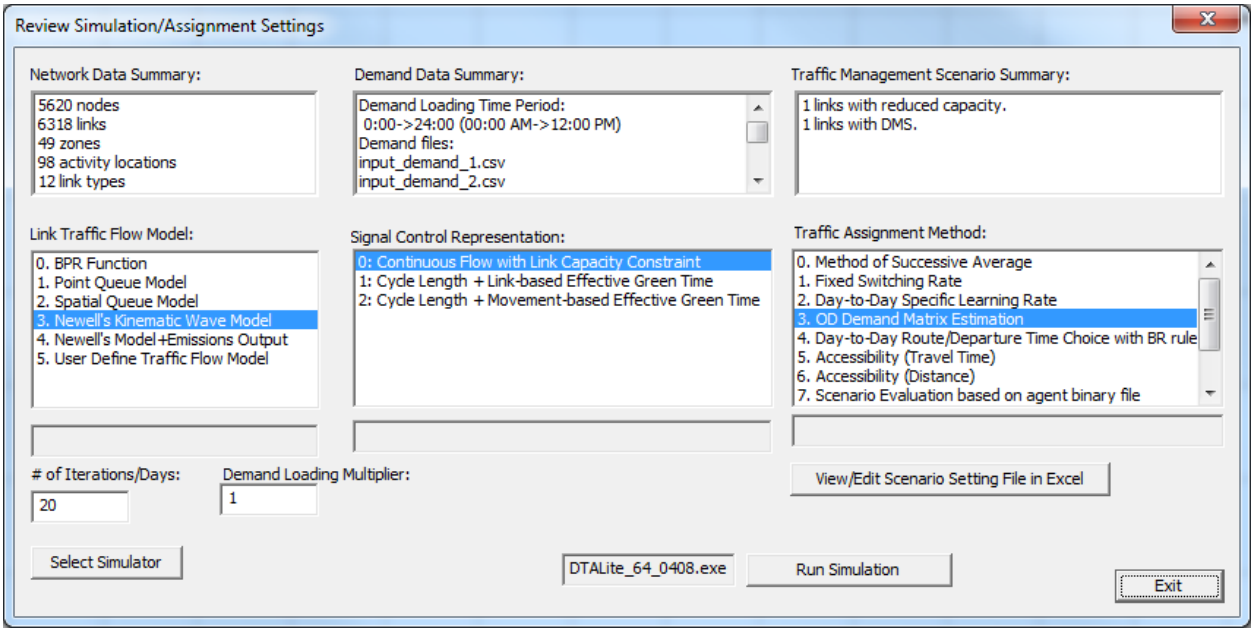

Figure 2: NEXTA Review Simulation/Assignment Settings

To display the results generated from the simulation run, there are various options located on the top ribbon and configuration side bar as shown in Figure 3. Users are able to display results in various animations showing the network vehicle trajectory, link density, link volume, link speed, link queue, impact, bottleneck, and emissions.

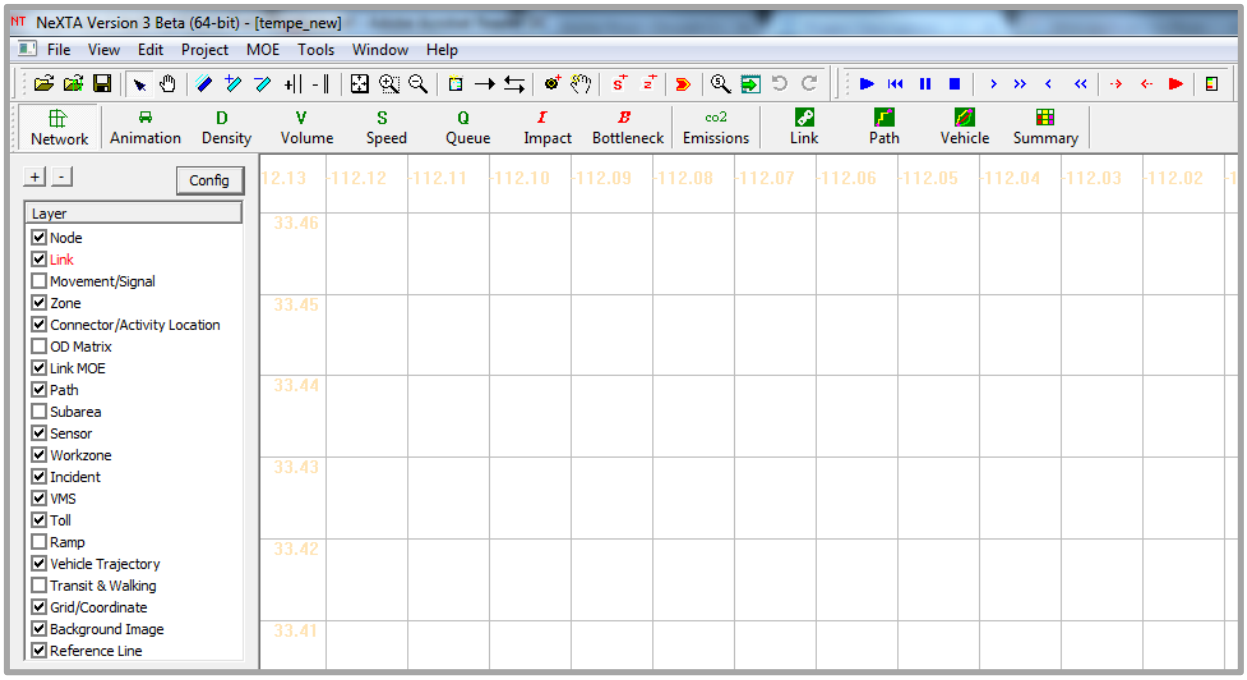

Figure 3: Result Display Ribbon and Configuration Tab

The configuration button on the left allows for alterations of data display for nodes and links.

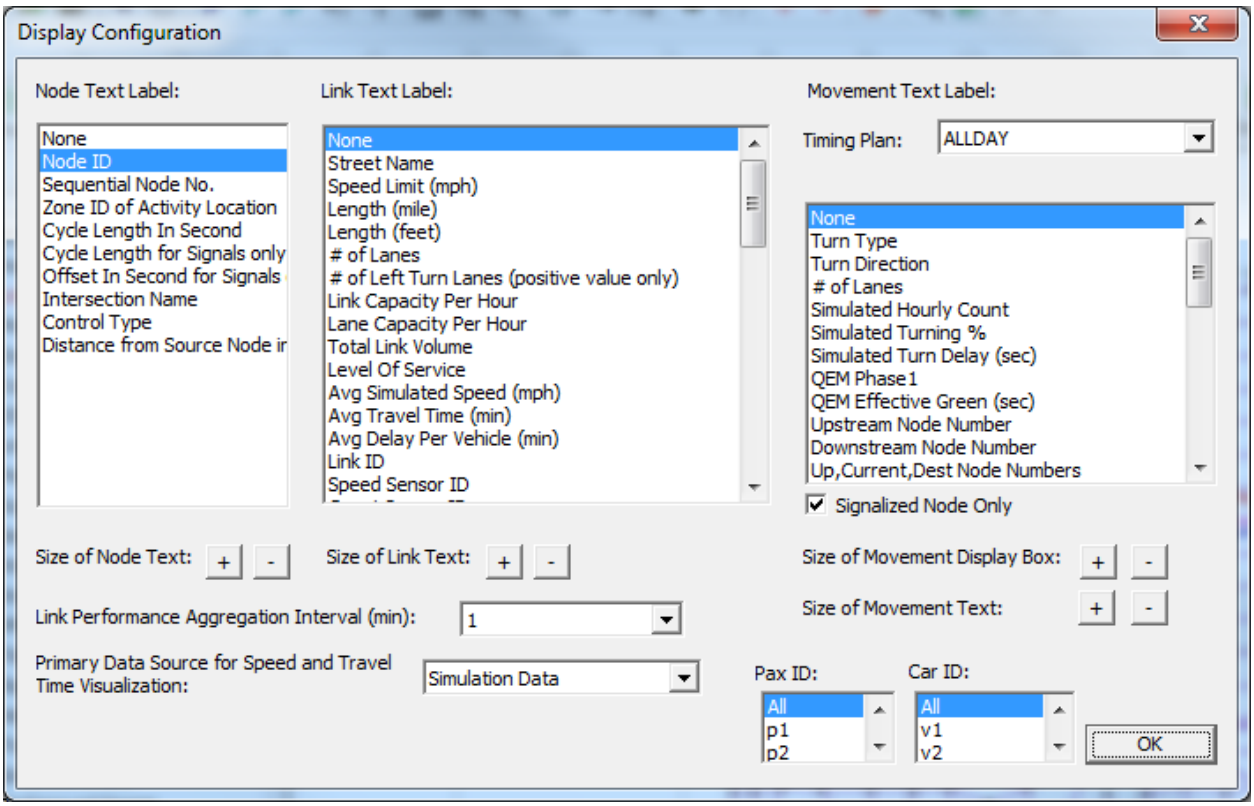

Figure 4: Display Configuration Pane

DTALite/NEXTA Data Structure

Since NeXTA uses DTALite for transportation network analysis, not all variables required as inputs to DTALite are required as inputs for visualization in NeXTA, and not all variables required as inputs to NeXTA are required as inputs to DTALite. The following Figure 5 describes the general work flow for the NeXTA data hub. The large data structure diagram on the next page shows the relationships between different input tables and their variables.

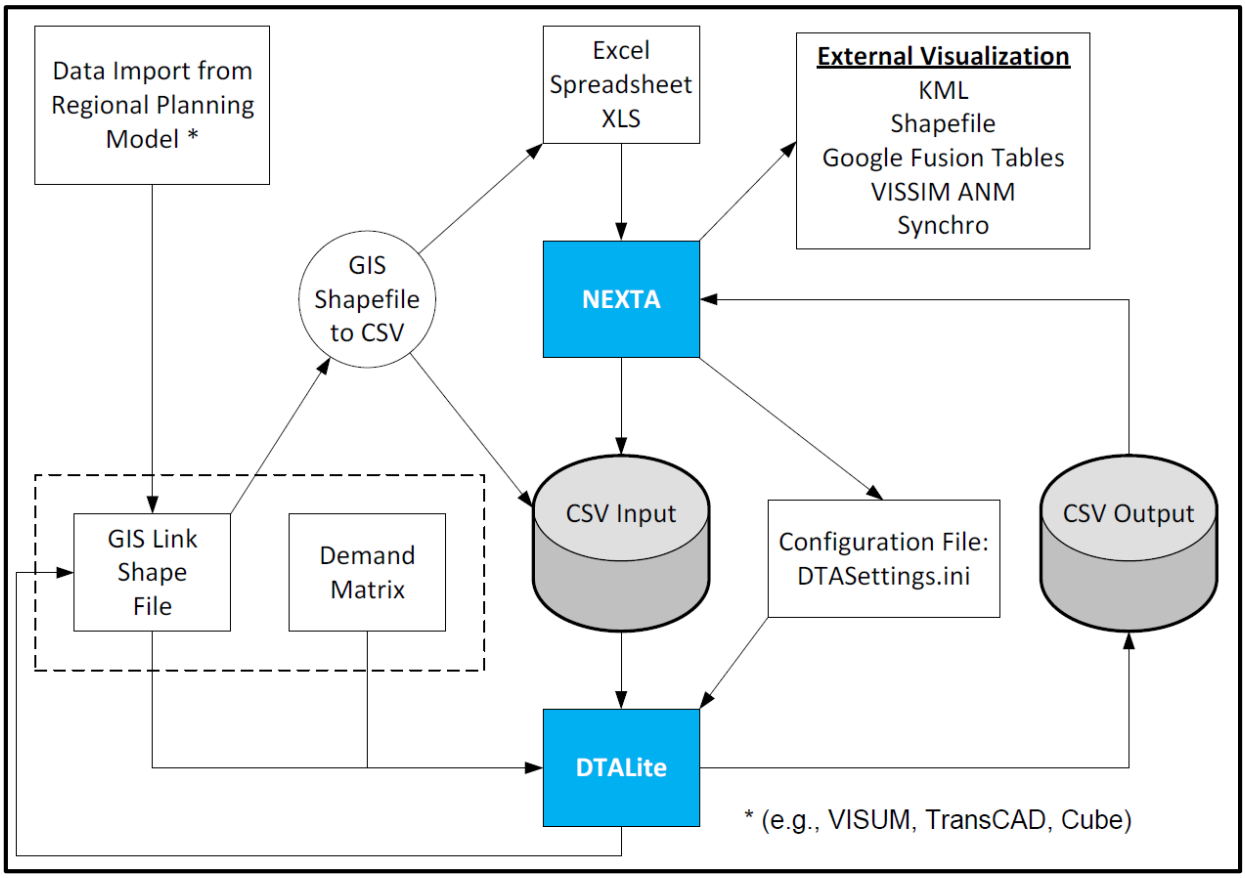

Figure 5: NeXTA/DTALite Data Input and Output Workflow Diagram

System Architecture

Component 1: Open data exchange format

This modified version of DTALite can use the original static assignment network data "as is", without requiring additional signal control data. That is, the spatial queuing model in the simulation engine will use the calibrated link capacity value (used for the BRP travel time function) to consider demand and supply interactions. To model possible demand variations, the demand input will include (1) regular OD demand table and time-of-day demand profile and (2) dynamic OD demand table under special events. To consider road capacity reductions/variations, the link capacity input on each simulated day can be adjusted by different random events.

Component 2: Traffic flow model

To capture the queue dynamics at typical bottlenecks (e.g. lane drop, merge and weaving segments), DTALite embeds a simplified spatial queue model with additional shockwave propagation tracking capabilities. This model uses Newell's cumulative-count based traffic flow theory to consider queue propagation and spillback. In comparison, state variables in CTM are density, incoming and outgoing flows and speed, while the state variables in Newell's model are incoming and outgoing cumulative flow counts, from which link-specific density, flow volume, travel time, and space-mean speed measurements can be easily derived.

For simplicity, DTALite will not initially simulate detailed signal control logics at each simulation interval, and the capacity impact of signalized intersections, stop signs and yields signs are implicitly considered through the link capacity attributes (used by the BPR function) from an external data input.

Component 3: Traffic assignment

Conventional static traffic assignment methods generally assume users' perfect information and deterministic capacity. To represent realistic travel behavior under day-dependent and stochastic travel times, this research will implement a new class of day-to-day learning algorithm under different traffic information provision strategies, in which travelers select routes according to their experienced and predicted average travel times. Specifically, in this model, commuters will use their experienced travel times to update the perceived travel time on different paths; and a certain percentage of commuters (e.g. 5%-15%) will switch to the least average travel time path every day.

Component 4: Time-dependent shortest path algorithm

Traffic assignment models require computationally efficient shortest path algorithms as a core solution routine. DTALite adopts an OpenMP technique to utilize multiple processors for shortest path calculation. OpenMP is a new interface standard that can facilitate programmers to decompose computational tasks to different processors in C++ and Fortran. Using this technical, DTALite will assign different origins to different processors to calculate the shortest path tree for a single origin and a single departure time. To reduce coding complexity, we will use C++ based standard template library (STL) to concisely represent multi-dimensional vectors and lists.

Component 5: Free Visualization and Analysis Interface

DTALite uses the NEXTA visualization package to import the static traffic assignment data in commonly used Excel and text format. The exiting NEXTA package already has the following capabilities for network-wide mobility analysis: (1) Link MOE, network MOE, vehicle trajectory visualization; (2) Bottleneck identification and (3) Subarea extraction and analysis. In this research, we will further develop a travel time reliability analysis module in NEXTA to systematically quantify the impact of different delay sources: demand vs. supply, recurring vs. non-recurring, using multiple days of simulation results.