Introduction to Transportation Anaysis, Modeling and Simulation (AMS) Data Hub

Please add comments directly in this document about the structure of data tables, specifical data schema, data conversion and database implementation. Please feel free to write an email to one of the members below if you have more questions.

Revision History:

Version 1: This document was previously prepared and maintained by Xuesong Zhou: xzhou74@asu.edu and Brandon Nevers bnevers@kittelson.com from KAI, based on the project report for a FHWA research project titled: **The Effective Integration of Analysis, Modeling, and Simulation Tools.**

Version 2: We invite different modelers, agency users to provide comments.

Overall comments and questions can be left here. Special comments can be added as in-line comments.

A group of members and friends from Transportation Network Modeling Committee (ADB30) and IEEE ITSS Technical Committee on Travel Information and Traffic Management are currently reviewing this document.

Ömer Verbas: ismailverbas2012@u.northwestern.edu
Jennifer Duthie: jduthie@mail.utexas.edu;
Stephen Boyles: sboyles@mail.utexas.edu;
Michail Xyntarakis: MXyntarakis@camsys.com;

Date: January 08, 2014

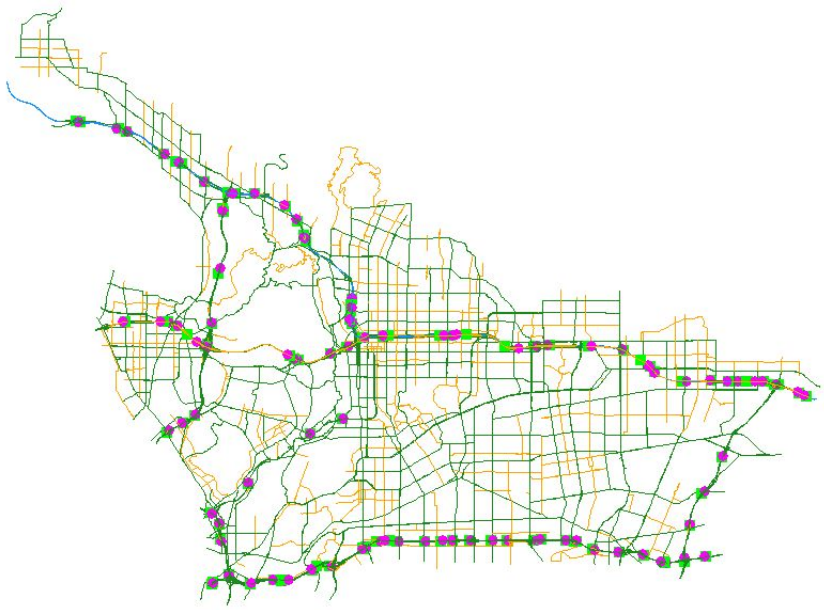
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Online resources:
1. sample Excel Data Set for West Jordan, Utah.
2. Pasedena, CA data handbook, network shape files and data, provided by Mygistics, through FHWA Research Data Exchange Program.
3. Coverted Pasedena data set in CSV format, visulizable by NeXTA.



Red circles and blue rectangles represent locations with speed and count data (from PeMS).

Simulation models used in transportation analysis are not well integrated among different domains (e.g., operations, safety, and environment) and for different levels of analysis (i.e., macro, meso, and micro). A recent FHWA research project developed a prototype data hub and data schema using the Network EXplorer for Traffic Analysis (NeXTA) open-source software tool to save users time to input data and to model and display results in a common format. The objective of the Analysis Modeling and Simulation (AMS) data hub is to define a prototype of operations and associated requirements that will allow for the effective integration of analysis modeling and simulation tools across various domains and scale. When they have been successfully integrated, these models will be able to easily exchange information and data at both the input and output levels. A further objective of this data hub schema and visualization prototype is to improve the effectiveness of the integration of operations through proof of concept and prototyping demonstrations.

In the recent FHWA AMS data hub project, the test applications validated the open-source data hub functionality by taking existing regional travel demand models from the respective regions, exporting the data to a dynamic traffic assignment model for mesoscopic analysis, exporting to a signal timing optimization tool, and then exporting to a microscopic simulation tool for detailed operations analysis. This document is aimed at model users, managers at modeling agencies, software developers, and researchers who are interested in advancing integrated modeling practices.

Many of the AMS tools have proprietary components. One solution to incorporate different existing industrial standards is to develop an open data format that allows software vendors to implement data conversion utilities from their own proprietary format, akin to the Open Document format which creates a standardized file format that allows users to open, edit, and save Microsoft Word documents from other applications such as Google Docs.

For maximum effectiveness, an open data format should be:

- Fully documented and available royalty-free;
- Developed through a publicly visible, vendor-independent process;
- Implementable by both proprietary and free/open source software;
- Interoperability among diverse traffic modeling and simulation tools; and
- Sufficient and unified in its representation at various spatial and temporal scales.

A multi-scale network editing and visualization tool is needed to integrate different simulation models across the full range of spatial and temporal resolutions. This data hub could serve future needs by enabling transportation researchers and software developers to build upon and expand its range of capabilities to other simulation packages.

The data hub visualization tool should have the ability to process and filter raw data from real-time sources including point, point-to-point and path measurements. Currently, many Radio Frequency Identification (RFID) RFID-based and Bluetooth-based Automatic Vehicle Identification (AVI) systems are widely deployed in real-time travel time estimation applications, while in-car navigation and smartphone applications using the Global Positioning System (GPS) technology has matured into a rapidly growing industry.

Visualization applied to multi-scale model integration is another emerging area of both research and practice. It is a critical component for both validating and reporting operational performance. The data hub visualization tool is expected to enable the visualization of input data and results as a central component.

Shown in Figure 1, the integrated multi-scale modeling platform offers the following unique features:

It enables tight interconnections between three critical entities: data, models and decision makers. By comparing simulation models using traffic data from multiple sources, the data hub model calibration utilities can extract useful traffic pattern information to construct multi-scale traffic state databases.

Through a cross-domain visualization interface, traffic management/planning organizations can better understand the complex model structure and multi-dimensional simulation results, and further increase the value of their contributions as they will achieve superior system representations, real-time response, and improved traffic control measures.

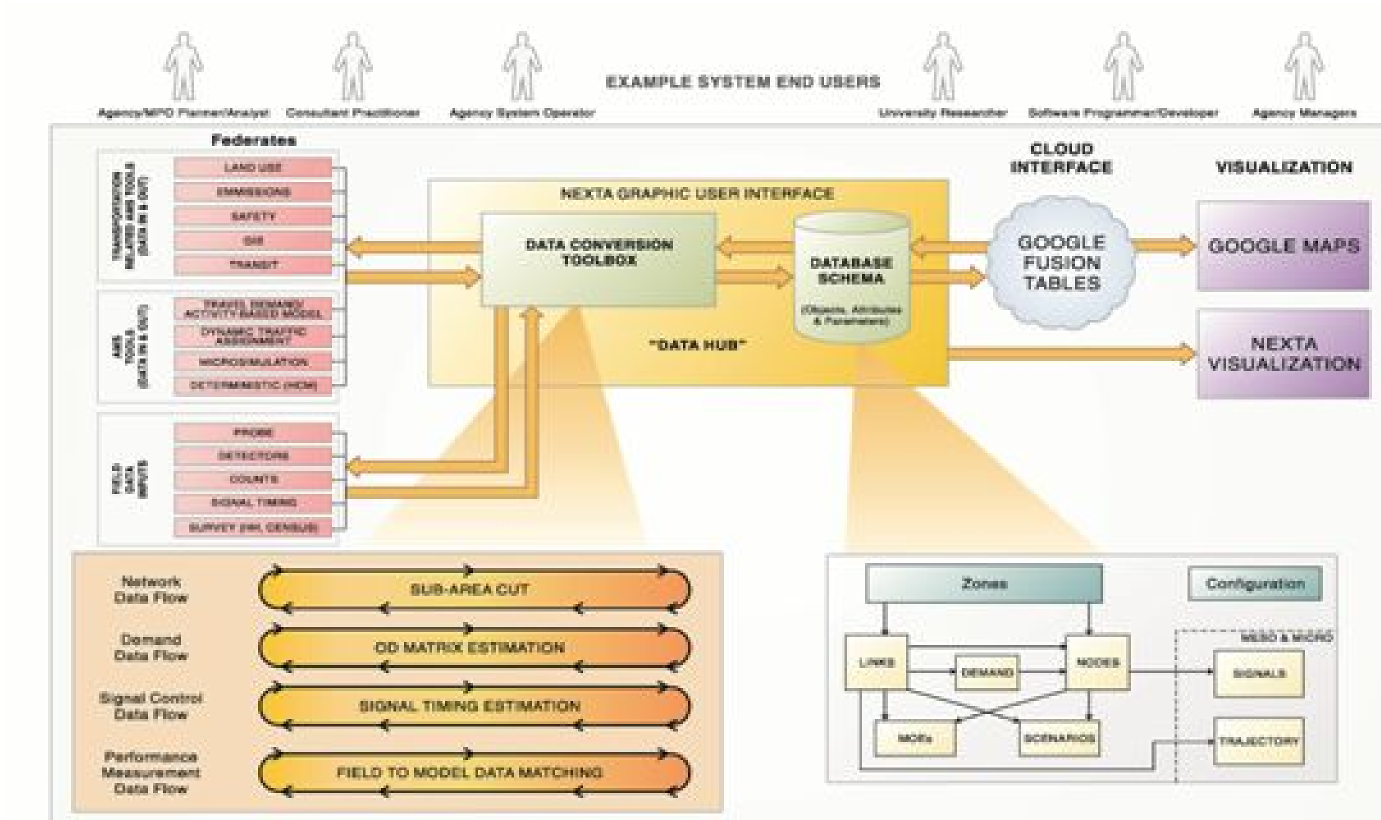


Figure 1. Data flow in AMS data hub and corresponding integrated multi-scale modeling platform

Network EXplorer for Traffic Analysis (NEXTA) aims to facilitate the preparation, post-processing and analysis of transportation assignment, simulation and scheduling datasets. The relationship of NEXTA and different scales of traffic analysis tools can be shown in Figure 2.

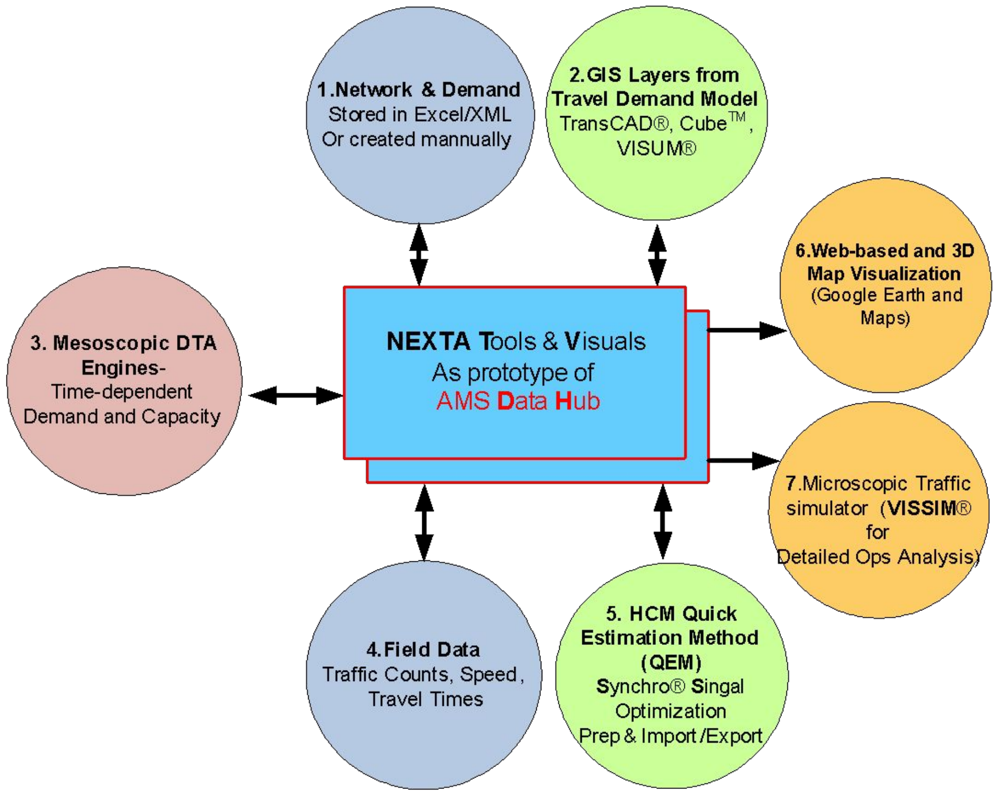


Figure 2. The relationship of NEXTA data hub tool and other modeling packages

As an open-source data hub processing and visualization tool, NeXTA supports the data linkage with the following tools/utilities:

(1) Network & Demand Data, created manually through NEXTA or prepared based on popular data file format (csv, Excel, XML). These data sets can be imported into NEXTA to serve as the base for mesoscopic DTA simulation and other modeling tools.

(2) GIS Layers from typical regional travel demand model: TransCAD®, Cube™, VISUM®. Typically, node, link, zone, centroid and connector layers can be directly imported through an open-source GIS shape file reader into the data hub.

(3) Network database, link-based Measure of Effectiveness (MOE) and vehicle trajectory files from mesoscopic DTA simulation.

(4) Field Data: Traffic counts, speeds, travel time from multiple locations and multiple days.

(5) HCM Quick Estimation Method (QEM) and Synchro® Singal Optimization Prep& Import/Export. The network and trajectory data from (2) and (4) should be converted by data hub tool (e.g. NeXTA) first to generate movement-specific turning volume.

(6) Web-based and 3D map visualization (through KML files for Google Earth and CSV file for Google Maps and Goolge Fusion Tables)

(7) Microscopic traffic simulation packages. E.g. a subarea of traffic network and path flow counts can be exported to VISSIM for detailed operational analysis.

DATA STRUCTURE

In this document, we will introduce the AMS data hub with a case where data is stored and represented in an Excel file and Comma Separated Values (CSV) file. Generally, for simple case, Excel file with multiple pages can be qualified with this task; for large-scale regional networks or with high-fidelity vehicle trajectory data, the primary data format is CSV. As the headers in the CSV files follow the suggested AMS data hub format, one can easily transfer the data from CSV files to a rational database, for example, MySQL or PostgreSQL, where the latter can directly use the geometric information coded in the CSV files. Those CSV files can be also readily available for adding/deleting/editing in Excel.

In NEXTA, a complete AMS data set is stored in a folder with multiple input and output files in CSV, with predefined standard AMS data hub field names for fast interchange or conversion among different data sources. The simple Excel representation is intended to provider readers a quick understanding on different data layers available in the current data hub schema. The CSV-based database format can incorporate large-scale networks and a huge volume of link MOE records and vehicle trajectory records.

Figure 3 shows a CSV database interconnected with four different data sources: (1) the default model setting csv files; (2) external csv files from other simulation models and tools, such as, modelLinkMOE, AgentTrajectory, AMS movement files; (3) the conversion of network and demand files from industrial standard to more accessible database for general mesoscopic DTA Simulators; (4) the csv files generated by DTA Simulators, like some output csv files.

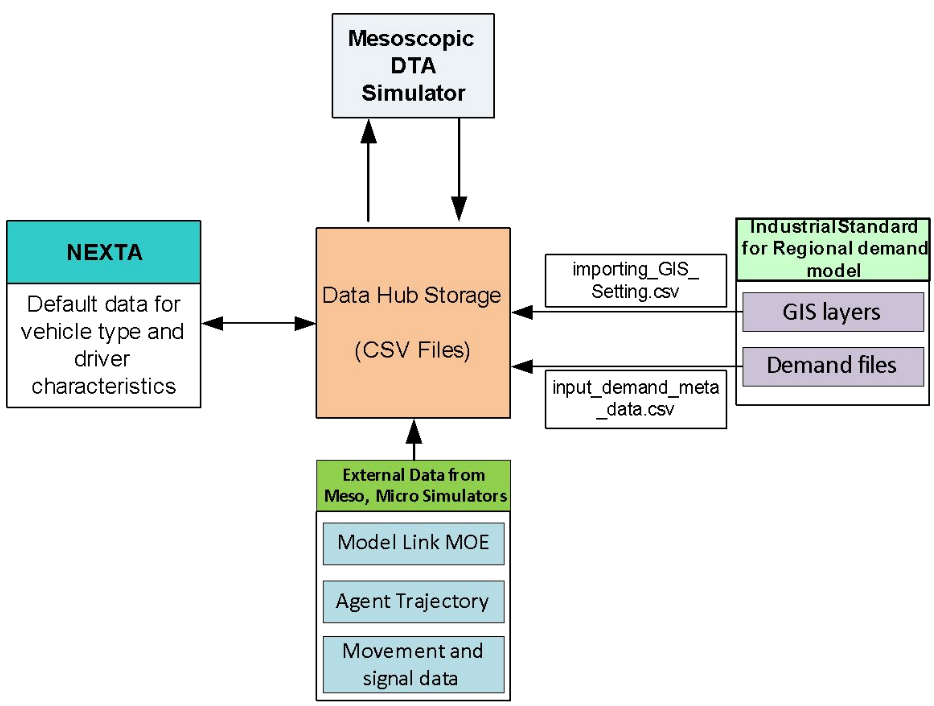


Figure 3. Data workflow from CSV database to different data sources and tools

Through the above data hub representation, specific benefits of the AMS data hub include the following:

- Capacity to quickly transfer most common travel demand models (TDMs) (TransCAD®, Cube, and Visum) into mesoscopic dynamic traffic assignment (DTA) resolution for enhanced operational evaluations.
- Tools to adjust network, demand, and signal timing to facilitate quality DTA evaluations (portable to multiple DTA engines such as and Dynamic Network Assignment-Simulation Model for Advanced Road Telematics (DynaSmart)). Dynamic Urban Systems for Transportation (DynusT), DTALite.
- Origin-destination matrix estimation (ODME) to calibrate model volumes to field counts.
- Tools and export/import utilities with signal timing optimization tool (Synchro™) to build confidence in signal timing plans for DTA and microsimulation use.
- Import/export tools from the data hub to a common microsimulation tool (Vissim™).

To offer a systematic perspective on the detailed data schemas, the data components in both Excel and CSV format are compared in Table 1. For most transportation data layers (e.g. node, link, zone), those two different formats use similar file/worksheet names with identical field names.

Table 1. Data Layers Represented in Different Environments/Applications

Excel Tables	**CSV Files for Large-Scale Applications**	**GIS Data Files and Regional Demand Data Files**	**DTA or Microscopic Simulation Data**	**External Traffic Sensor Data**	**HCM Capacity and Signal Timing Estimation Method**
Importable to NEXTA	*Readable and writable in NEXTA*	*Importable to NEXTA, through **importing_GIS_ setting** file*	*Importable to NEXTA*	*Visualizatio n in NEXTA*	*Exportable from NEXTA*
Configuration	input_link_type, input_node_control_type				
Node	input_node	GIS node shape file	Network file		Importable from Synchro UTDF format
Link	input_link	GIS link shape file		GIS link shape file	
Zone	input_zone	GIS zone shape file	Zone file		
Activity Location	input_activity_location	GIS zone or centriod or connector shape files	Generation and destination data for links/nodes		
Demand Matrix	input_demand_meta_dat a demand files	Various demand file format, including matrix, column and trip record files through **input_demand_meta_data**	Demand files		Turning volume, and path flow volume data
Model Link MOE	Model link MOE		Generated from simulator, for visualization in NEXTA		
Speed Sensor Data	Sensor_speed			Prepared by data vendor or state agencies, for visualizatio n, calibration in NEXTA	
Count Sensor Data	Sensor_count				
Agent Trajectory	Model_trace, Model_trajectory,		Generated from simulator, for visualization in NEXTA		
Signal Timing Plan	AMS_timing_plan		Usable by traffic simulator		Readable and writable in NEXTA
Phasing Data Movement Data	AMS_movement		Usable by traffic simulator		Readable and writable in NEXTA

The Excel file (titled DataHubExample.xlsx) contains the following worksheets: Configuration, Node, Link, Zone, ActivityLocation, DemandMatrix, ModelLinkMOE, SpeedSensorData, CountSensorData, SignalTimingPlan, MovementData, PhasingData, etc. The data structure diagram of different data layers is illustrated in Figure 4.

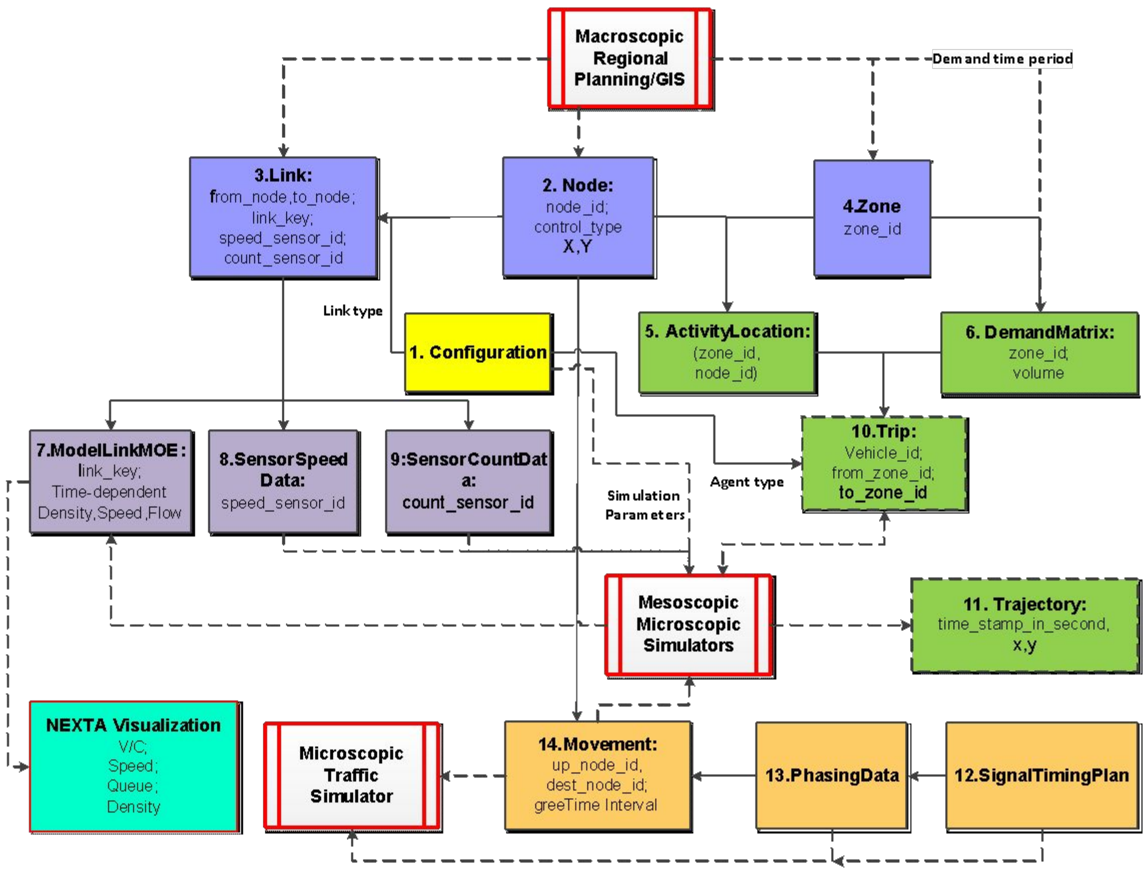


Figure 4. The data structure diagram in AMS data hub

The fundamental data components are **(2)** Node, **(3)** Link and **(4)** Zone, which build traffic network for further analysis. They can be manually input by users into an Excel/XML file, created inside NEXTA, or converted from Macroscopic Regional Planning/GIS packages. Meanwhile, when the demand from those packages is set with time period, the **(6)** DemandMatrix can be obtained.

In **(1)** Configuration, it provides basic setting values for some datasets, such as, the link type for **(3)** Link, the agent type for **(10)** Agent, and parameters for Mesoscopic DTA Simulator.

Through the data fields zone_id and node_id, **(5)** ActivityLocation is connected with **(2)**

Node and **(4)** Zone. Based on **(5)** ActivityLocation and **(6)** DemandMatrix, mesoscopioc and microscopic traffic simulation packages can receive and generate **(10)** Model Trip Data and **(11)** Model Trajectory data, which can be visualized and analyzed in NEXTA.

The link layer **(3)** can be connected with **(2)** Node, **(7)** ModeLinkMOE, **(8)** SpeedSensorData and **(9)** CountSensorData through a number of data keys, namely, from_node, to_node, link_key, speed_sensor_id, count_sensor_id, respectively. **(8)** SpeedSensorData and **(9)** CountSensorData can also be the data source for model calibration and OD demand matrix estimation. The connection of **(3), (7), (8)** and **(9)** is listed in Table 2.

Table 2. Connection of base link layer and related MOE and sensor data layers

Keys Defined in Link Layer (3)	Corresponding Data Table	Corresponding Key Referred in External Data Files	Example
Link_key	(7) ModeLinkMOE	Link_key	1285->5018
speed_sensor_id	(8) SpeedSensorData	speed_sensor_id	114+04361
count_sensor_id	(9) CountSensorData	count_sensor_id	VID14

In addition, **(12)** SignalTimingPlan can describe multiple time durations of signal timing plans within a day. **(13)** PhasingData and **(14)** MovementData details movement-specfic lane characteristics and green/yellow/red time intervals for a signal cycle.

For developers maintaining the existing packages, they can easily export simulation results from their current packages to the AMS open data format used in **(7)** ModeLinkMOE, **(10)** Model trip and **(11)** Agent trajectory. In particular, Link MOE layer **(7)** is used by NEXTA for visualization of V/C (volume-to-capacity), Speed, Queue, Density, etc. and vehicle trip data and trajectory **(10)** is the basis for advanced vehicle trajectory processing, which derives a rich set of reliability measures for the entire network, selected path and subareas.

The datasets **(12)** SignalTimingPlan, **(13)** PhasingData and **(14)** MovementData serve as the base for Microscopic Traffic Simulator and further detailed operational analysis.

DATABASE SCHEMA

This section describes a unified database schema that facilitates input/output data conversion in the short term and promotes data consistency and exchange in the long term.

Advantages of the database schema are as follows:

- It is "software neutral" meaning it can be applied using any software program.
- The database tables that are initially proposed can be expanded in the future.
- Software can be configured to read all or a subset of parameters in a table.
- A software vendor can add field parameters that are specific to its product.
- Links and nodes are geo-coded and thus are more easily integrate with GIS and other visualizers such as Google Maps®.
- Since the data structure is transparently and openly defined, it can be readily scrutinized, tested, and enhanced.
- The database schema can serve as a platform to encourage the development of analysis tools and visualization tools, especially from small software vendors and researchers.
- The database schema moves the practice of integrated modeling towards a standard for how modeling data should be stored and shared.

One major disadvantage of such an overarching, unified data schema is its size, which has an adverse effect on computation speed and efficiency. The data structure is understandably large in order to accommodate analysis models in various resolutions. As such, a single intersection analysis using the HCM method would utilize only a small portion of the data structure.

A description for each of the component of the database schema is provided in the following sub-sections. The descriptions are general in nature, but the database schema can be implemented in XML or a database format.

The following description illustrates the database schema, with a sample dataset from West Jordan, Utah.

(1) Configuration

This worksheet describes the general configuration parameters used for a typical mesoscopic traffic simulator. The listed parameters are for illustrative purposes and are not all-inclusive, especially for high-fidelity microscopic simulators or DTA simulators using different traffic flow models/theories. It should be noted that, there are also a few software tools within the same resolution can be based on different theories. In addition, software tools based on the same theory can have their own interpretation, assumptions, and default values.

As anticipated, configuration parameters vary significantly from one resolution to another and from model to model within one resolution. It is proposed to include the few common parameters among software packages in the configuration table and allow parameters specific to software to be added and stored on an as-needed basis.

Table 3. Sample Data in Configuration Table

Category	Key	Value	Value 2	Description
format_version	version	0.1		Number of version
parameter	length_unit	mile		If the unit should be mile. If not, please convert it.
parameter	travel_time_unit	min		If the unit should be min. If not, please convert it.
parameter	default_wave_speed	12		Default value
parameter	default_k_jam	220		Default value
parameter	demand_multiplier	1		Based on the demand setting
link_type	1	Freeway	f	There are different link types. The number (1,2..) represents link type, and the letter(f,h..) means link code.
link_type	2	Highway/Expressway	h	
agent_type	1	SOV		There are different agent types. The number represents agent type.
agent_type	2	HOV		
activity_mode	1	home		There are different activity modes. The number represents activity mode.
activity_mode	2	work		

(2) Node

A node is generally created at an intersection or where some roadway characteristics change (i.e., merge or diverge gore, change in free-flow or posted speed, change in horizontal or vertical alignment, etc.). Other reasons for the creation of nodes include the following:

- **Specific software requirements**: Some software tools require entry nodes, exit nodes, dummy nodes, or interface nodes to mimic a network. These nodes have minimal or no real-world properties.

- **Software limitations**: Some software tools include comprehensive link properties, while others do not. For the latter, nodes must be inserted to break up the link in order to model changes in the link properties.

- **TAZ centroids**: Previous practice has been to have a single node for each TAZ. This node is ideally located at the TAZ centroid, such that connectors from the centroid to the road network are representative of average travel times from the TAZ to the rest of the transportation network. The newer activity-based models often require much finer-grained representation of connections of different TAZ subareas (or even individual land parcels) to the transportation network. Hence, the trend is to have multiple nodes to represent different subareas within each TAZ.

It should be noted that some software tools create nodes automatically and store them internally when links are created. Some tools define nodes as locations where statistics need to be collected, and they have no physical attributes.

Challenges: Some challenges when integrating different datasets include the following:

- Some software tools have limitations on the number of nodes and numbering scheme.

- Some software tools do not use nodes explicitly. Vissim ™ uses connectors to store properties that may often be associated with nodes.

- Some software tools generate nodes automatically.

Proposed structure: The data structure proposed for nodes aims to replicate the physical real-world conditions as much as possible. The goal is to rely on software developers or conversion modules to aggregate the data into a format usable by specific software. Some suggestive guidelines for node creation and placement include the following:

- More than one node may be required to adequately represent connections between different subareas of the TAZ to the transportation network.

- Nodes should be used to represent signalized intersections, roundabouts, stop-control intersections, yield control intersections, major driveways, and merge/diverge areas.

- Intermediate nodes should be used to limit link length to 1,500 ft.

- For freeway merge and diverge locations, nodes should be placed where edge stripes meet.

- For intersections, nodes should be placed at the middle of the intersections. Node properties such as stop bar locations can be employed to delineate the intersections' dimensions.

- Node locations should be geo-coded so that they can be displayed in Google Maps®/Google Earth® or synchronized with various GIS layers.

A sample table 4 for nodes is shown as follows:

Table 4. Sample fields in node table

node_id	control_type	x	y	Description
1285	4way_stop_sign	-111.943	40.61645	It illustrates the location of one specific node with one control type.
1286	4way_stop_sign	-111.934	40.63078	
1358	pretimed_signal	-111.912	40.58352	
4899	unknown_control	-111.91	40.62578	

(3) Link

A *link* is generally defined as a roadway segment with uniform geometry and traffic operation conditions. However, each software tool may provide its own guidelines to suit its needs. Depending on software's methodologies and procedures, link properties requirements also vary. Additional differences are also introduced by the analyst as he/she molds the field data into a form required by the analysis software. Examples of link definitions and their properties are as follows:

- In CORSIM™, all links are directional and defined by the analyst with an upstream node and downstream node. Since a node does not have physical dimensions and link curvature is not accounted for, the analyst must often override CORSIM™ estimated link length with an actual field value.

- In an HCM-based tool, a weaving link is defined by a point at the merge gore where the edge stripes are 2 ft apart and by a point at the diverge gore where the two edges are 12 ft apart. In other analysis tools, these two points are loosely defined by the analyst.

- In a TDM like Visum, only major road segment are coded. As such, a link in Visum may be split up into three Vissim ™ links to account for significant driveways or minor intersections.

Challenges: Some challenges when integrating different datasets include the following:

- DTM tools typically define bidirectional links, while others only work with directional links.

- Some software tools require entry links, exit links, or dummy links.

- Some software tools require more link properties than others. For example, DTMs do not typically ask for lane-add/lane-drop, turn lanes, storage, median, lane width, etc.

Proposed Structure: The data structure proposed for links aims to allow analysts to rely on software tools or conversion modules to mold the physical field data into a format usable by the software.

Key features of the proposed link table include the following:

- Replicates the physical real-world data as much as possible.

- Complements or replaces formats that agencies use for roadway log/inventory.

- Geo-codes all links so that they can be displayed in Google Maps®/Google Earth® or synchronized with various GIS layers.

- Stores all link properties at the lowest common denominator and lets software tools aggregate the data to needed levels.

Sample fields in the link table are shown in Table 5, which defines the link type, begin and end nodes, the shape point path followed by the link, and the functional class. It should be noted that the fields can be defined to optimize data exchange among software platforms in the short term and progressively refined to replicate real-world conditions in the long term. Note that the table structure is normalized as with the node table structure.

- The directional link representation is highly suggested in the AMS link table, so directional link characteristics can be defined for capacity such as number of lanes, capacity, and free-flow speed. If the link in the existing database is bidirectional, there will be separate records for node A to node B and node B to node A in the AMS data hub format.

- Separate link MOE and sensor count tables are used to store traffic volumes for a particular scenario.

Table 5. Sample Fields in Link Table

Data field	Link 1	Link 2	Description
name	Cent	1300 West	
from_node_id	1285	4952	They are defined in “2.Node”
to_node_id	5018	5022	

link_key	1285->5018	4952->5022	It corresponds to field "link_key" in "7.ModelLinkMOE".
Count_sensor_id		VID14	It corresponds to field "speed_sensor_id" in "9.SensorCount".
Speed_sensor_id	114+04361	114+04425	It corresponds to field "count_sensor_id" in "8.TMCSpeed".
direction	1	1	
length	0.2384	0.5007	
number_of_lanes	7	1	
link_type_name	Freeway	Major arterial	Depended on the specific link type, the attributes of those links have similar values.
speed_limit	21	32	
lane_capacity_per_hour	1428.6	666.4	
link_type	1	4	
jam_density	220	120	
wave_speed	12	12	

(4) Zone

In the existing practice, zones are used in TDM and DTA. TDMs include TAZs that typically have number of automobiles per household, household income, and employment. DTA models also adopt the TAZs from TDMs. Synchro^{TM} also uses the zone concept to divide a network into zones to facilitate signal optimization or output for a portion of the network. Microsimulation software, most HCM-based tools, and field data do not use the zone concept.

Challenges: Generally, there are no obvious challenges to define zones for all resolutions. However, a few minor definition issues include the following:

- A link can be dissected by a zone boundary and as a result is physically located in more than one zone. Our solution is to map activity locations of such a zone to the node level, that is, to the upstream or downstream node of the link.

Proposed Structure: The data structure proposed for zones (see Table 6) meets current AMS needs and provides the flexibility to expand in the future. In this structure, a zone can have many activity locations defined in the activity location table 7.

Table 6. Sample Fields in Zone Table

zone_id	geometry	Description
1	<Polygon><outerBoundaryIs><LinearRing><coordinates> -111.940667,40.625617,0.0 -111.940667,40.621155,0.0 -111.947984,40.621155,0.0 -111.947984,40.625617,0.0 -111.940667,40.625617,0.0 </coordinates></LinearRing></outerBoundaryIs></Polygon>"	Zone identification number. A zone number used in OD demand table files must be defined here first. Zone numbers are not required to be consecutively sequential.
2	<Polygon><outerBoundaryIs><LinearRing><coordinates> -111.940863,40.607852,0.0 -111.949905,40.607852,0.0 -111.949905,40.611808,0.0 -111.940863,40.611808,0.0 -111.940863,40.607852,0.0 </coordinates></LinearRing></outerBoundaryIs></Polygon>"	

(5) ActivityLocation

Table 7. Sample Fields in ActivityLocation Table

zone_id	node_id	external_OD_flag	Description
1	5577	0	• zone_id: Zone identification number; • node_id: Node identification number associated with the zone identification number in the same row. Must defined in input_node.csv • external_OD_flag: Used to identify the type of activity location as non-external (default = 0) or external (-1 or 1). When 0, acts as both origin and destination.
2	11123	0	
3	11159	0	
4	11180	0	
5	11282	0	

(6) DemandMatrix

Different analysis resolutions require different traffic demand resolutions. Requirements for each of the four analysis resolutions are as follows:

- A traditional TDM typically outputs several types of trip tables. Production/attraction tables are intermediate person trip tables that are output from the trip generation and trip distribution process. O-D tables are person trip or vehicle trip tables that are used in the assignment process. Separate trip tables are produced by trip purpose and mode, and time-of-day trip tables are used for assigning passenger trips and vehicle trips to the transit and road networks. Different agencies have different time-of-day definitions. For example, a small agency might work only with daily trip tables, another agency might have a morning peak and an off-peak trip table, and a third agency may have separate trip tables for morning peak, afternoon peak, midday, and late night/early morning.

- Activity-based TDMs output individual trip records for individuals in a synthetic population that is intended to represent the population of the modeling region. Each trip record contains information that includes a weighting factor, travel mode, O-D, and some socioeconomic characteristics of the individual that could be used for more refined analyses (e.g., environmental justice analyses). Most activity models in use today rely on traditional network assignment procedures to get link volumes and level of service. As a result, these trip records are rolled up into traditional trip tables by mode and time of day.

- HCM-based tools typically analyze the peak 15-min volumes. An analyst may collect 1+ h of volume data during the peak morning and afternoon periods and then filter the data to isolate the heaviest 15-min interval for analysis.

- DTA models typically start with O-D matrices from a TDM and then adjust them to derive hourly or 15-min volumes.

- Microsimulation models often work with 15-min volumes that span the 1 to 2-h analysis period. Even though not frequently done, traffic variation every 1 or 5 min can be specified for each 15-min interval.

- Detailed traffic count data are kept for purposes of monitoring and validation.

In addition to the analysis practices mentioned, demand resolutions also vary in field data. For instance, many agencies collect and archive detector loop data. Raw data come in as time stamps which can then be aggregated to generate 1-min, 5-min, 15-min, hourly, 24-h, weekly, or monthly volumes. Due to processing speed and storage cost, not all of the data resolutions are stored nor are they retained for the same period of time.

Challenges: The brief summary in the previous section illustrates the difficulties encountered when attempting to integrate demand requirements for various modeling resolutions. Theoretically, traffic demands can be stored at the lowest common denominator (e.g., 20-s or 1-min volumes) to serve microsimulation analyses and then aggregated for other analysis resolutions. However, this approach is not currently practical for the following reasons:

- The original source of demand data for future years is from regional TDMs. Travel models output results by time period, which is typically at a very coarse resolution compared to that required by DTA and microsimulation models (e.g., a 3- to 4- h peak period). Current year traffic count data are typically used to break down travel model outputs into finer resolutions (i.e., 15 min or less).

- A large amount of 20-s or 1-min data can be unwieldy (i.e., difficult to store and work with).

- 20-s or 1-min data are not often collected and stored for all network links.
- Some link and node volumes are derived from land use data in TDMs. As a result, it is not possible to develop fine resolution volumes.

Proposed Structure: To satisfy various traffic demand levels, a demand meta database structure is introduced to link a set of files for each modeling resolution. It is envisioned that demands in one subset may be derived from another subset, from analysis results, or from field data.

The sample demand data is listed in Table 8. The more sophisticated definitional file for demand data can be found in the later demand meta data base (17) used by NEXTA.

Table 8. Dataset of DemandMatrix

zone_id	1	2	3	4
1	0	0	0	0
2	193.9329	39.3442	19.86503	136.8871
3	0	0	0	0
4	196.5152	16.803429	21.37807	96.21726

(7) ModeLinkMOE

MOEs are typically compiled for a node, a link, a corridor, a zone, or an entire network. Depending on the analysis level, MOEs can be generic or comprehensive. Some common MOEs include the following:

- **MOEs for nodes**: V/C ratio, average delay, entering volume, etc.
- **MOEs for links**: V/C ratio, density, queue, delay, speed, travel time, number of stops, number of lane changes, vehicle-miles traveled, emissions, fuel consumption, etc.
- **Path/corridor**: Travel time, queue, delay, progression, vehicle-miles traveled, number of lane changes, number of stops, etc.
- **Network**: Average speed, total vehicle-miles traveled, total vehicle-hours of delay, total vehicle-hours of move time, total fuel consumption, total emissions, etc.

Challenges: Some challenges with MOEs include the following:

- Some link MOEs can potentially be aggregated to generate path and network

MOEs. Space can be saved at the expense of computational speed.

- Emerging MOEs for reliability, safety, and sustainability have yet to be incorporated into mainstream AMS tools.

Proposed Structure: The MOEs table is proposed to comprise a number of subset tables, each of which contains MOEs for link, path and movement, respectively.

Based on this dataset, NEXTA can display the simulated results from other packages and compare it with the results from sensor data. Here, the field of "link_key" in the Model Link MOE table is referred to the field in link table **(3)**. Table 9 lists its detailed attributes.

Table 9. Dataset of ModelLinkMOE Table

link_key	1285->5018	1285->5018	1285->5018	1286->11125	1286->11125	1286->11125
start_time_in_min	932	933	934	931	932	933
end_time_in_min	933	934	935	932	933	934
link_hourly_volume	8.57	17.14	17.14	8.57	17.14	8.57
density	0.6	1.2	1.2	0.61	0.61	0.31
speed_per_hour	21	21	21	21	21	21
queue_length_percentage	0	0	0	0	0	0
cumulative_arrival_count	1	3	5	2	3	4
cumulative_departure_count	0	1	3	0	1	3

(8) SpeedSensorData

It is the speed data from TMC (Traffic Message Channel) for traffic analysis or simulation. Table 10 lists its attributes. The "speed_sensor_id" in speed sensor data should be defined in link table **(3)**, so the speed data can be analyzed and visualized at the link and path levels.

Table 10. Dataset of SpeedSensorData Table

speed_sensor_id	day_no	Min_0	Min_15	Min_30	Min_45	Min_60
114+04361	1	61.33333	60.66667	60	60	59.66667
114+04362	1	57.66667	58.33333	58.33333	59	59.33333
114+04363	1	61	61	61	61	61
114+04364	1	56.66667	57	57.66667	57.66667	58.33333

114+04365	1	58.33333	60	59.66667	59.33333	59
114+04366	1	57.33333	58	58.33333	58.66667	58.66667
114+04367	1	54.33333	57	56	56	58.33333

(9) CountSensorData

This dataset describes traffic counts at different observation time period. The key field "count_sensor_id" is defined in **(3)** Link, so it builds the connection between **(3)** and **(9)**. Table 11 lists its attributes. It should be remarked that, additional fields such as density and speed can be also provided for each record.

Table 11. Dataset of CountSensorData Table

Count_sensor_id	day_no	start_time_in_min	end_time_in_min	count
VID1	1	990	1050	2267.8
VID2	1	990	1050	2267.8
VID3	1	990	1050	1685.5
VID4	1	990	1050	1685.5
VID5	1	990	1050	1273
VID6	1	990	1050	1273

(10) Vehicle Identification Counts (Bluetooth based)

(11) Vehicle Trajectory

Vehicle trajectories can range from very detailed second-by-second lane-by-lane trajectory generated by microsimulation models to grossly estimated trajectories in macro or HCM-based tools in order to estimate emissions and fuel consumptions. In microsimulation, trajectories are generated for individual vehicles and used in animation and estimation of the most detailed MOEs if needed.

Challenges: For a large network analyzed over a long period of time, the amount of vehicle trajectory data generated can be unwieldy.

Table 12 describes the structure for vehicle trajectory data.

Table 12. Dataset of Vehicle Trajectory Table

trip_id	from_zone_id	to_zone_id	start_time_in_min	end_time_in_min	travel_time_in_min	demand_type	vehicle_type	number_of_nodes	path_node_sequence
2	18	41	930.02	936	5.98	1	2	4	1286;11125;5436;5578;
3	9	21	930.03	936	5.97	1	1	4	5356;5216;5868;6253;

(12) SignalTimingPlan

Signal operations and settings are traditionally simplified before being entered in an analysis tool. In most analysis tools, the ring and barrier setup is coded as sequential phases. Timing parameters such as minimum initial, splits, etc. are simplified as green, yellow, and all-red, and most controller settings such as red rest, recall mode, gap, etc. are ignored.

Challenges: Signal control challenges include the following:

- The traditional approach to simplify the signal settings for analysis cannot adequately depict advanced features (e.g., complex overlap phasing, priority treatment, dedicated pedestrian, or bicycle phase).
- Most if not all signal controllers are built modularly. A fully loaded signal cabinet can have many advanced functions and features that a bare-bone controller lacks. Capturing everything in the data structure can be overwhelming.
- Adaptive signal control is becoming increasingly popular. A variety of adaptive approaches already exist as well as controllers and their detection methods and requirements.
- Several standards for signal controller exist.

Proposed Structure: It is proposed that the data structure for signal control adopts the (National Electrical Manufactures Association) NEMA standards and includes all typical settings that directly control the signal head's display and functions. That is shown in Table 13, Table 14 and Table 15 in part.

Table 13. Dataset of SignalTimingPlan

'00:00	'00:15	'00:30	'00:45	'01:00	'01:15	'01:30	'01:45	'02:00
FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE

(13) PhasingData

Based on the timing plan, this dataset provides the data about phases in specific intersections (nodes). The nodes should be defined in advance in **(2)** Node. Table 14 lists its some attributes.

Table 14. Dataset of PhasingData Table

timing_plan_name	node_id	key	value	D1	D2	D3	D4	D5	D6	D7	D8
ALLDAY	1358	SplitInSeconds		10	10	10	10	10	10	10	10
ALLDAY	1358	BRP									
ALLDAY	1358	MinGreen									
ALLDAY	1358	MaxGreen									
ALLDAY	1358	VehExt									
ALLDAY	1358	TimeBeforeReduce									
ALLDAY	1358	TimeToReduce									
ALLDAY	1358	MinGap									

(14) MovementData

This dataset, shown in Table 15, can be generated from other packages or generated on the basis of **(12)** SignalTimingPlan and **(13)** PhasingData. It provides the data source for mesoscopic DTA simulation or microscopic traffic simulator. The nodes in "node_id", "up_node_id" and "dest_node_id" are defined in Table Node **(2)** in advance.

Table 15. Dataset of MovementData

timing_plan_name	node_id	up_node_id	dest_node_id	turn_type	turn_direction	sim_turn_hourly_volume	Lanes	speed	Lost time
ALLDAY	1289	4952	5018	Through	WBT	9	7	59	4
ALLDAY	1289	4952	11124	Left	WBL	0	0	59	4
ALLDAY	1289	4952	11125	Right	WBR	20	0	59	4
ALLDAY	1289	5018	4952	Through	EBT	2	7	48.3	4
ALLDAY	1289	5018	11124	Right	EBR	12	0	48	4
ALLDAY	1289	5018	11125	Left	EBL	0	0	48	4

(15) Configuration: input_link_type.csv

The input_link_type table allows users to define their own specific link types, as long as the flag variables are correctly used to identify how the different link types are connected/related (e.g., freeways connect to arterials using ramps). Only one flag may be used for each link type. Link types can also be used to determine how links are visualized in NeXTA. This file is required when using the network import tool to interpret the link type field in the link shape file. Table 16 lists the sample link type settings.

Table 16. Dataset of Link Type Table

link_type	link_type_name	type_code	default_lane_capacity	default_speed_limit	default_number_of_lanes
1	Freeway	f	1800	65	3
2	Principal arterial	a	1000	40	2
3	Principal arterial	a	1000	40	3
4	Major arterial	a	900	35	3
5	Minor arterial	a	850	30	2
6	Collector	a	650	25	1
7	Local	a	600	20	1
8	Frontage road	a	1000	45	2
9	Ramp	r	1300	30	2
10	Zonal connector	c	2000	100	2
100	Transit link	t	1000	40	1
200	Walking link	w	1000	5	1

(16) Configuration: input_node_control_type.csv

The input_node_control_type table defines the control type of nodes in the network in terms of control type name, unknown control, no control, yield sign, 2way stop sign, 4way stop sign, pretimed signal, actuated signal and roundabout. This file is required when using the network import tool and the control type field is read from the node shape file. Table 17 lists the sample node control type settings.

Table 17. Dataset of Node Control Type

control_type_name	control_type
unknown_control	0
no_control	1
yield_sign	2
2way_stop_sign	3
4way_stop_sign	4
pretimed_signal	5
actuated_signal	6
roundabout	7

(17) Demand Management: input_demand_meta_data.csv

It is ideal that all demand data obtained is just from CSV Database and therefore can be directly used by NEXTA/Mesoscopic DTA simulator. As a Data Hub, NEXTA has to tackle many different demand formats, and the input_demand_meta_data file is designed for this complex problem.

The input_demand_meta_data table is used to define the characteristics of demand data. Through a temporal demand profile table per record, users can define the proportion of demand in the network as a function of time, which is used to initiate trips in the simulation over the modeling horizon. This table can be used to supplement demand type information in an input demand table, where DTA simulator will use the temporal demand profile information in place of other time information.

- This meta-data file requires several entries, but the relevant entries are as explained below:
- Specify the demand file name (e.g., sov_14_15.mtx) and format (e.g., column)
- Specify the number of lines in the demand file to be skipped by NeXTA (8 for MTX file)
- Indicate whether subtotals are present in the last column (zero for none)
- Specify the loading start time and end time for the demand file (840 to 900, or 2PM to 3PM)
- Specify the demand types associated with the demand file (only demand_type1)

Table 18 illustrates part of sample demand meta settings.

Table 18. Demand Meta Settings Table

Field name	Sample vaule	Description
scenario_no	0	Scenario identification number
file_sequence_no	1	File identification number
file_name	input_demand.csv	Name of demand file
format_type	matrix	Input file format type
number_of_lines_to_be_skipped	1	The number of lines to be skipped at the beginning of demand file
loading_multiplier	0.25	Local multiplication factor applied to the number of trips in the demand file
start_time_in_min	300	Demand loading start time, which is the time

		gap in min from 0:00
end_time_in_min	660	Demand loading end time, which is the time gap in min from 0:00
apply_additional_time_dependent_profile	1	0: not use the time dependent profile in this table, that is, a flat demand pattern is used between time period [start_time_in_min, end_time_in_min] 1: use the time dependent profile in this table
subtotal_in_last_column	0	flag used for subtotal in last column of matrix demand file
number_of_demand_types	1	Number of demand types stored in demand file
demand_type_1	1	Such as, hov, sov
'07:15	0.07	Proportion of demand in specified time interval compared to 24-hour time period

You can also check the complete sample ***input_demand_meta_data.csv*** file online.

Here, **format_type** and **number_of_lines_to_be_skipped** should be emphasized due to its flexibility and complexity. In the learning document, it illustrates several demand formats, such as, column format, matrix format, full matrix format, DYNASMART, agent_csv, agent_bin, which also decide the number of lines to be skipped. All detailed explanations can be found at page 31-42 in the document, ***Data Structure and Workflow***.

(18) Conversion for large-scale applications: import_GIS_setting data

To import the GIS shape files and regional demand files in various format, we need import_GIS_setting file to describe the variables contained within the GIS shape file. It is a user-defined configuration file, and a user can use Excel to edit it. Depending on the type of data to be imported, the configuration file consists of several sections.

As defined in **Column A**, there are general sections of file, configuration, and sections for different network objects such as node, link, zone, and optional centroid and connector layers.

The "**key**" settings given in **Column B** are used to match and convert the shapefiles from the DBF files.

"**value**" in **Column C** is the matching name between NeXTA's "key" values, and those given in the shapefiles. Column "**allowed_values**" shows which definitions are related to "value" for binary values (such as km vs. mile, lane vs. link, etc).

Column "**required_or_optional**" shows which values must be defined for a successful import.

Filename of GIS layers

In this section, it defines the shape file names for GIS layers of nodes, links, zones, centroids, and connectors. In this example, the file names are listed in Table 19. The reference file names should correspond to the layers exported from planning packages. Node, link and zone are the required layers for a successful GIS network import. DTALite does not require connectors in its network representation, so users can leave empty values for keys "**centroid**" and "**connector**".

Table 19. Filename of GIS layers

section	key	value	required_or_optional
file_name	node	SLC_Network_Node.shp	desired
file_name	link	SLC_Network_Link.shp	required
file_name	zone	SLC_90thSouth_Zone.shp	desired
file_name	centroid		
file_name	connector		

The detailed explanation can be found at page 37-39 in the document, ***Data Structure and Workflow***.

Configuration

In this section, it describes general model attributes and import options that can accommodate different network coding conventions. These settings include the latitude/longitude coordinate system definition, length units, one way vs. two way links, and whether the capacity is given per lane, or per link. Its advanced settings also allow users to import link type-specific speed limit and capacity values, and to estimate the possible locations of signalized intersections, and link-based effective green time. Table 20 lists the sample configuration settings.

Table 20. Table Configuration settings

section	key	value
configuration	1) with_decimal_long_lat	yes

configuratio n	2) length_unit	mile
configuratio n	3) number_of_lanes_oneway_vs_twoway	oneway
configuratio n	4) lane_capacity_vs_link_capacity	lane
configuratio n	5) conversion_factor_for_obtaining_hourly_capacity	1
configuratio n	6) direction_0_as_oneway_vs_twoway	oneway
configuratio n	7) default_link_direction	oneway
configuratio n	8) node_number_threshold_as_centroid	0
configuratio n	9) use_default_speed_limit_from_link_type	no
configuratio n	10) use_default_lane_capacity_from_link_type	no
configuratio n	11) use_default_number_of_lanes_from_link_type	no
configuratio n	12) identify_from_node_id_and_to_node_id_based_on_geome try	no
configuratio n	13) create_connectors_for_isolated_nodes	no
configuratio n	14) identify_signal_intersection	yes
configuratio n	15) minimum_speed_limit_for_signals	28
configuratio n	16) maximum_speed_limit_for_signals	60
configuratio n	17) default_cycle_length_in_second	110
configuratio n	18) minimum_length_for_importing_links	0.00001

The detailed explanation can be found at page 39-42 in the document, ***Data Structure and Workflow.***

Node attributes

The import values are node ID, node name, the zone (TAZ) to which a certain node belongs to, and node control type. Table 21 lists the sample node attributes, from which node_id and TAZ are required. TAZ can be also set to the same as node_id.

Table 21. Node settings

section	key	value	required_or_optional
node	node_id	N	required when node file is provided
node	name		
node	TAZ	TAZID	optional, if not provided, then users can use node_number_threshold_as_centroid to specify the first n nodes directly as centroids of zones.
node	control_type		optional, if node control type has been specified.

Link attributes

In this section, it defines the link shapefile attributes. In order to code the corresponding attributes correctly, the user can access the link shapefile through GIS software, and read the link attributes. The available link attributes are from and to node, name, link ID, link type, transportation modes that link is open for, direction definition, number of lanes, hourly capacity, speed limit, and number of lanes, capacity, speed limit and link type for reversible lanes, if links are defined as two way links. Table 18 lists the sample link attributes.

Table 22. Link settings

section	key	value	required_or_optional
link	from_node_id	A	desired
link	to_node_id	B	desired
link	name	STREET	
link	link_id		
link	TMC		
link	link_type	FT	
link	mode_code		
link	direction	ONEWAY	
link	length	DISTANCE	desired
link	number_of_lanes	LANES	desired
link	hourly_capacity	CAP1HR1LN	desired
link	speed_limit	SFF	desired
link	r_number_of_lanes		
link	r_hourly_capacity		
link	r_speed_limit		
link	r_link_type		

The detailed explanation can be found at page 43-44 in the document, ***Data Structure and Workflow***.

Zone attributes

In this section, it defines the zone shapefile attributes. Only a zone id field is needed, if users have specified a value for key **centroid** in the file_name section. This zone id should be consistent with the zone id system used OD demand files. If the zone layer does not present, one can set a positive value (say 3000) for key **node_number_threshold_as_centroid** in section configuration to add zones (and the corresponding activity locations). Table 23 lists the sample zone attributes.

Table 23. Zone settings

section	key	value	required_or_optional	notes
zone	zone_id	Id	desired	required only if zone file is given in section file_name

Centroid attributes

In this section, it defines the centroid shapefile attributes. Fields node_id and TAZ can have the same or different names.

Connector attributes

In this section, it defines the connector shapefile attributes. Fields zone_end and node_end are referred to the upstream and downstream node ends of a connector. A connector can have its own speed limit, number of lanes, link type per record or use the default values given in the configuration keys: default_speed_limit, default_link_type, defaut_direction and default_number_of_lanes. Table 24 lists the sample connector attributes.

Table 24. Connector settings

section	key	value
connector	zone_end	
connector	node_end	
connector	length	
connector	number_of_lanes	
connector	hourly_capacity	
connector	default_speed_limit	
connector	default_link_type	

connector	defaut_direction	0
connector	defaut_direction	0
connector	default_speed_limit	60
connector	default_link_type	99
connector	defaut_direction	0
connector	default_number_of_lanes	7

Final output

In this section, it allows users to define if a two-way link should be applied with the offset or not, as a two-way link is converted to two directed links in NeXTA but still have the same geometry data. Table 25 lists the sample output settings.

Table 25. Final output settings

section	key	value
final_output	offset_link	yes

Finally, you can also check the complete sample import_GIS_setting.csv file online

(19) Additional transportation data layer: Transit data

Transit is coded as part of a regional transportation network. Transit is typically coded as a set of transit lines. Each transit line record contains a code for the mode (e.g., bus, rail, etc.), a code for the individual transit operator, a list of nodes along which the line runs, the service headway (with provision for specifying different peak and off-peak headways), and a code for the speed, which can be entered either as a factor related to the auto speed on a link or as an absolute speed (the latter is the format for transit lines with a separate right-of-way). Transit information also includes a fare matrix, which specifies TAZ-to-TAZ fares. Travel modeling software is becoming increasingly sophisticated with regard to transit, providing more detail on transit characteristics such as line specification, transit speeds, allowance for transit vehicle capacity limitations, and specification of allowable transfers between different transit routes.

Automated vehicle location (AVL) and automated passenger counting (APC) are being implemented on larger systems. AVL provides real-time transit vehicle arrival information at each transit stop or station. APC provides information on individual vehicle boardings and alightings at individual stops.

Google® has recently launched Google Transit®, which provides real-time transit information for over 120 transit operators in more than 475 cities in 36 U.S. States and 7 Canadian provinces. Development of this new application has entailed development of a new database schema called General Transit Feed Specification (GTFS) to allow transit operators to provide their information to Google® in a unified format. Because of its widespread and growing use, GTFS may provide a useful basis for defining the transit portion of the AMS database schema.

Challenges: Transit data are inherently complex. A full set of data on transit operations should include information on operator characteristics, route locations, stop locations, schedules, demand (e.g., demand by route and time of day and boardings and alightings by stop), and on-time performance versus schedule. Smaller operators do not collect all these data or may only collect them on a sample basis to fulfill Federal Transit Administration requirements for reporting to the National Transit Database.

Proposed Structure: The proposed structure consists of several tables that define transit routers and data on patronage by route. This structure is based in part on Google® GTFS.