ENTERPRISE TRANSPORT NETWORK EDITING IN TRANSPORT MODELING SOFTWARE

Minhua Wang, Ph.D., Citilabs, Inc.

INTRODUCTION

Transport Planning is a comprehensive process to promote the development of a multimodal transport system to ensure safe and efficient movement of people and goods while balancing environmental and community needs. Transport planning and demand forecast modelling processes rely on a wide variety of data and information. With the growth of the amount of data and the complexity of multimodal transport networks required to support planning and modelling studies, and with the growth of demands for enterprise transport network coding and integration with GIS, traditional desktop based transport network editing and file based data management for modern transport planning has become problematic in practice.
In an enterprise environment, transport network coding and network data editing should be based on a common network systems, or so called "master network", which combines multi-modal networks, multi-year networks, and multi-scenario networks in a single, integrated, and consistent spatial-temporal database. Such single and integrated database will eliminate network data redundancies, synchronize network data changes across multi-year networks and multi-scenario networks, better control versions of network editing to allow multi-user editing on the same network.

However, editing networks in a single and integrated database is not an easy job for most transport modelling software. There are two key issues need to be addressed: 1) handling of scenario based network editing; and 2) handling of network editing in an enterprise environment. Handling scenario based network editing is to allow storage of scenario networks with the master network to eliminate data storage redundancy and maintain data consistency. Handling network editing in an enterprise environment is to allow multiple users to concurrently access and edit the same network.

This paper will review the common transport network editing practice in the desktop based transport modelling software and to propose a solution to enable enterprise network editing with the transport modelling software. This solution includes an enterprise data model to store network data in an enterprise database, and enterprise network editing tools to handle both scenario network editing and multi-user network editing with a transport planning software.

TRANSPORT NETWORK EDITING WITH DESKTOP BASED TRANSPORT MODELING SOFTWARE

Most transport modelling software are single user based desktop software using file based data file (e.g., binary file) or simple database (e.g., Access) file. File based data file or simple database file usually does not have data management capability which causes problems for transport modellers when dealing with multi-modal network editing (e.g., highway network and transit network), and multi-year network editing (e.g., base year network and scenario year network) and multi-scenario network editing (e.g., alternative network or project based network), in particular, when there are multiple user editing the same network. The problems can be grouped into the following categories:

1. Data redundancy: data redundancy occurs when editing scenario networks or when multiple users edit the same network.

    a. When editing scenario network, a scenario network can be created either from the base year network or from a scenario network; for example, year 2020 network may be created from base year 2000 network or from scenario year 2010 network. With the file based network editing, scenario network changes are updated on a duplicate copy of the base year network or a scenario year network, causing redundant network data on the scenario networks

    b. When multiple users edit the same network, due to lack of data management capabilities, the network file has to be copied to multiple users, causing redundant data in multiple copies of the network file, also causing data inconsistency when changes are made in one user but not updated to others

2. Data inconsistency: data inconsistency occurs when a network is edited by multiple users or when base network elements are changed but the changes are not synchronized in the related scenario networks.

    a. Data inconsistency caused by multi-user editing: as mentioned above, when user B edit the same network as user A, a copy of the network file is made for user B, any changes user B made on the network will not be automatically visible and usable by user A, causing big headache for many transportation agencies trying to synchronizing network changes from multiple user edits on the same network system.

    b. Data inconsistency caused by parent network changes are not synchronized in child networks: as mentioned, due to lack of data management capabilities, changes made on the parent network are not automatically synchronized in the child networks, users has to manually identify those changes and make appropriate changes on the child networks individually. For example, due to initial error, a two-lane road was miscoded as one-lane on the base year network, there are five scenario networks are originated from the base year network, this simple change in the base year network has to be identified and changed in each of scenario year network files, otherwise, causing data inconsistency problem.

c. Data inconsistency caused by mutual related multi-modal transport networks: most multi-modal networks are spatially related, whether they are spatially connected at connection points (transfer stations) or they spatially overlaid such as bus route network and road network. When one network changes, it does not automatically synchronize changes in the related networks; for example, a highway network changes, the related public transit network is not synchronized.

3. Handling of multi-user concurrent editing: with the implementation of enterprise IT infrastructure, more and more transportation agencies move from single user network editing environment to multi-user network editing environment, or enterprise environment. In an enterprise environment, file based network file cannot handle multi-user concurrent editing due to file lock by the operating system and lack of concurrent transaction control. The only solution which allows multi-user concurrent editing is to implement enterprise database to store network data.

4. Data security: in a multi-user network editing environment, data security, that is who can edit what, becomes an important issue for a transport planning agency. Due to lack of user management capability, file based network file is not capable of implementing data security functions such as defining user roles and privileges on data editing.

WHAT IS ENTERPRISE TRANSPORT NETWORK EDITING

It has been widely demanded in the transport modelling user community that transport modelling software should support transport network editing within an enterprise environment. As the most network data originated from an enterprise GIS system and the implementation of enterprise IT infrastructure in many transport planning agencies, requirements for supporting enterprise network editing has become an increasing challenge for transport modelling software. These requirements may include:

- Maintain a single repository for network data and modeling data for access by multiple users based on “Master Network” data model
- Be able to edit and save scenario based network editing
- Allow multiple users to edit the same network data
- Allow multiple users to access the same modeling data
- Implement enterprise wide security policy on data maintenance, data access and data management
- Allow automatic synchronization of network changes among modeling data
- Manage scenarios in an enterprise environment to allow sharing of model scenarios
- Integrate with enterprise GIS system for better network data

The fundamental components for enterprise network editing include an enterprise transport modelling database as a single repository for all network data and modelling data, and a set of enterprise network editing tools to allow modelling software to access data stored in an enterprise database and to support multi-modal network editing, multi-scenario network editing, multi-user network editing, and to allow automatic synchronization of network changes among transport network data. The single network data repository requires an enterprise data model that is based on master network concept, supports enterprise data editing and data management, and seamlessly integrated with agency's enterprise GIS system, and can be implemented in different enterprise database management systems and different applications.

Master Network Concept and Data Model

Master network data model represents a collection of abstractive data entities derived from all data elements required for demand forecast modelling, transport system analysis and traffic studies, including highway networks, highway capacity data, land use data, public transport networks, public transit system data, time tabling data, junction data, origin-destination matrices, accessibility data, as well as GIS data support for demand forecast modelling. The key components of a master network data model include network data objects, scenario data objects and network relationships which are supporting the handling of scenario based network editing.

Network Data Objects in a Master Network Data Model

1. Highway network (road/street network) data objects: highway network data model for transport modeling is compatible with common GIS centerline network model with modeling characteristics. The difference between modeling network and GIS network is that in the modeling world, a network is consisted of links and nodes, a link is defined by two nodes: A node (starting node) and B node (Ending node) with directionality A to B or B to A; while in GIS network, a link is defined by a collection of vertexes including two end nodes without directionality. So, a highway network may include a geometry network with true shape road geometry, e.g., those derived from GIS centerlines, and a logic network with modeling attributes. Multiple networks can be defined and managed in a network metadata table.
2. Public transit (PT) network data model: PT network data model represents networks from different public transit types including bus network, rail network, light rail network, commercial rail network, metro/subway network, etc., within which bus network shares the same geometry network with highway network, while other transit networks use the designated geometry networks participating in the multi-modal transport network. The PT networks can be can be defined and managed in a PT network metadata table, and relationships between PT network and highway network can be explicitly defined in the metadata table. A PT network will contain three basic data objects: PT lines, PT Links and PT Line Stops; PT line stops are collection of physical stops for a specific PT line.

3. Junction data model: junction data model represents data objects associated with junction modeling (e.g., turning movement), including junction, approach, movement and stage. Junction data model is centered at junction, approaches, movement and stages are referenced by junction. Junction data has a linkage to highway network node through Node field as well as a reference to a specific highway network

Figure 1 illustrates a sample of network objects in a master network data model.

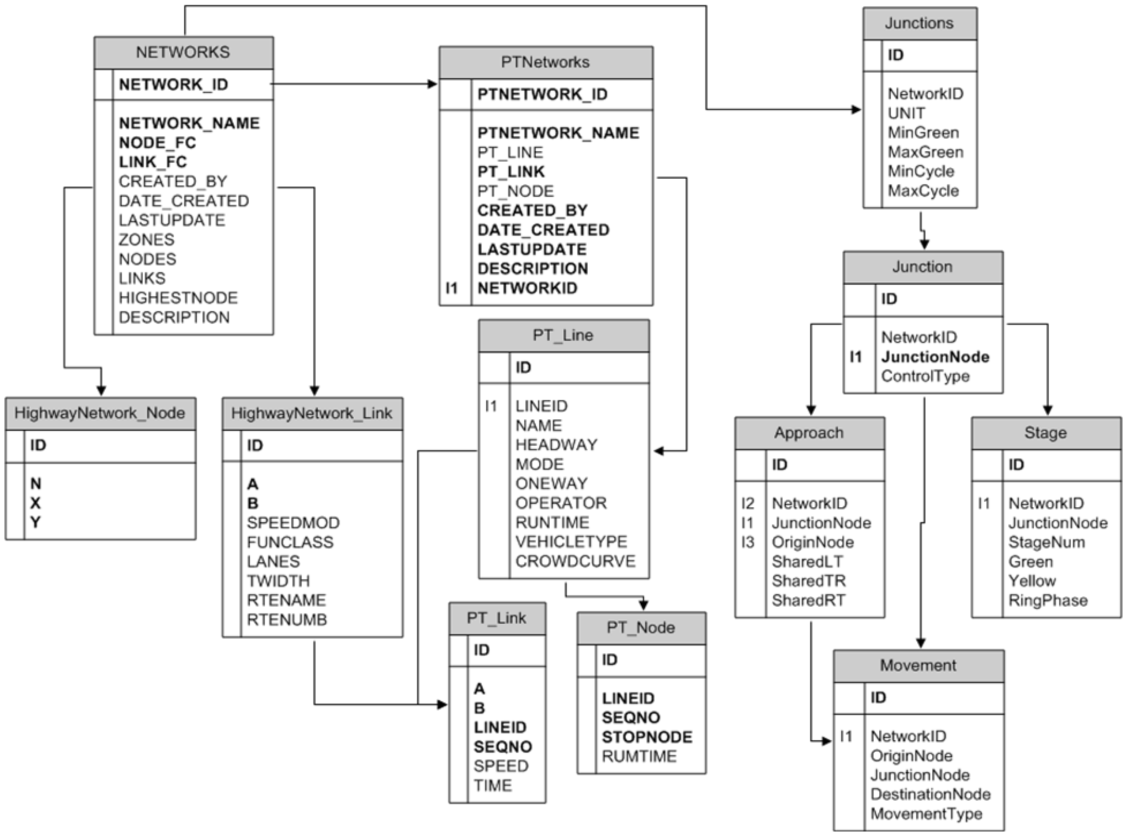


Figure 1 *– Master Network Data Model – Network Objects*

Scenario Data Objects and Scenario Data Management in a Master Network Data Model

Scenario, in demand forecast modelling, is defined as an alternative or time based changes to the base scenario. As a master network data model, scenario data objects represent the scenarios for source network data changes/updates, a typical example of data scenario is year based network updates, for example, a 2-lane link in year 2010 is changed to 4-lane link in 2020. All network data objects may have a scenario, including highway networks, PT networks and junctions. Each network data object scenario is defined in a scenario network metadata table and referenced by network data object identifier and scenario identifier. Scenario metadata is the key to track relationships between parent networks and scenario networks and search a scenario network. Figure 2 illustrates scenario data objects and scenario management in a master network data model.

Network Data Relationships in a Master Network Data Model

One of key aspects of a master network data model is the storage of network data relationships in database. These relationships not only include parent-child relationships between source network data objects and network data object scenarios, but also include relationships between highway networks and PT networks, as well as highway networks and junctions. These relationships are extremely important when cascading network changes to all related network data objects. Figure 2 illustrates an example of network data relationships in scenario metadata tables.

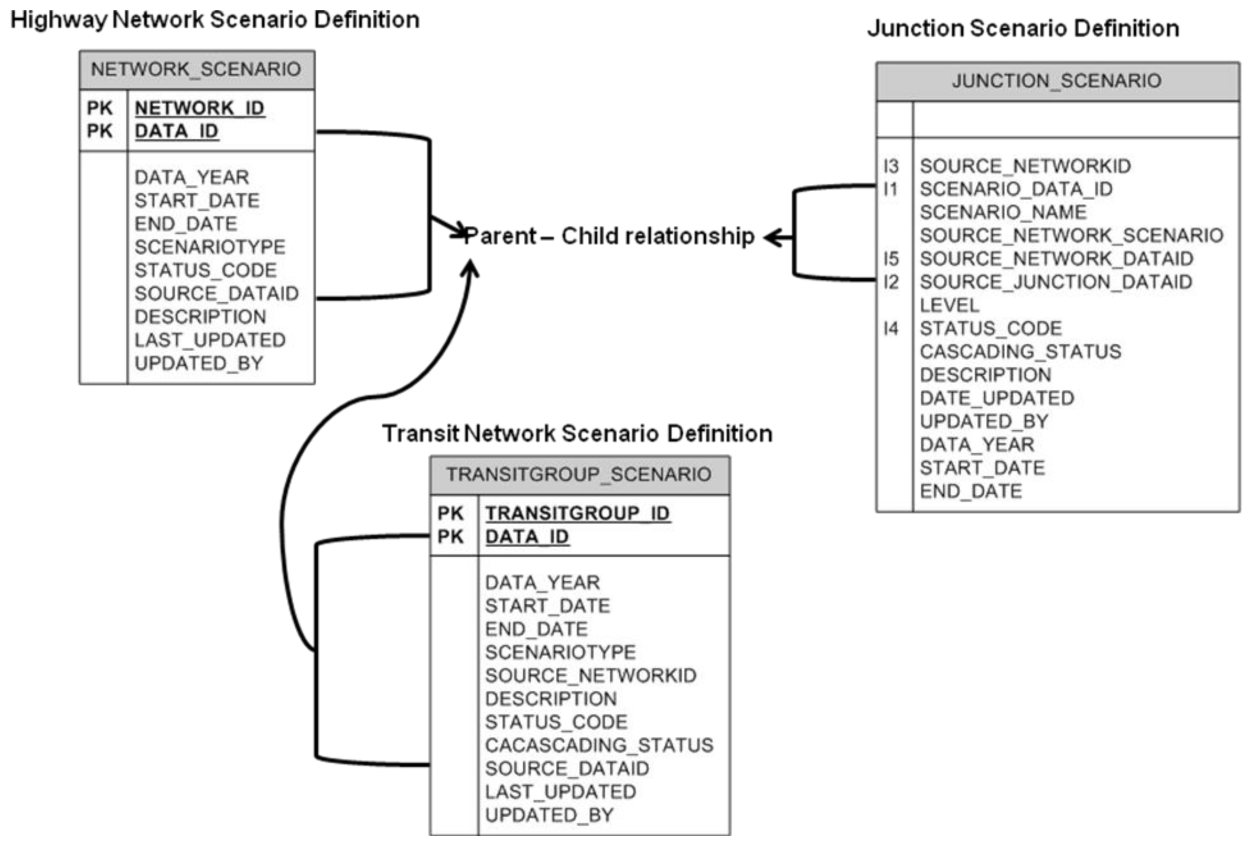


Figure 2 *– Master Network Data Model – Network Data Relationships in Scenario Metadata Tables*

Scenario based Transport Network Editing

Scenario based transport network editing is to edit scenario network within the master network. There are three key characteristics in scenario based network editing:

1. Store scenario network changes with base network: for scenario network, only changes to the base or parent network will be saved to eliminate redundant network data from the base or parent network in scenario network; by doing so, scenario changes will not override the base or parent network and the base or parent network elements will not be deleted

2. Track parent-child relationship in scenario network: a scenario network, either created from the base network or from a scenario network, will maintain single track parent-child relationship (single parent) to provide a mechanism for cascading base or parent network changes to the child scenario networks

3. Time stamp network editing and track network status in both base or parent networks and scenario networks to effectively query a scenario network from the master network; for example, a network link is split in a scenario network, that link in the base or parent network will be deleted but time stamped and set status to retired at the scenario, by doing so, the link will not be shown in the scenario network, but will be shown in the base or parent network.

With the scenario based network editing, the master network becomes a single source of transport network, eliminating headaches for synchronizing network editing from multiple users, multiple scenarios and multiple networks. It also provides a mechanism for cascading network changes within the master network. Figure 4 illustrates tracking scenario edits in network data objects:

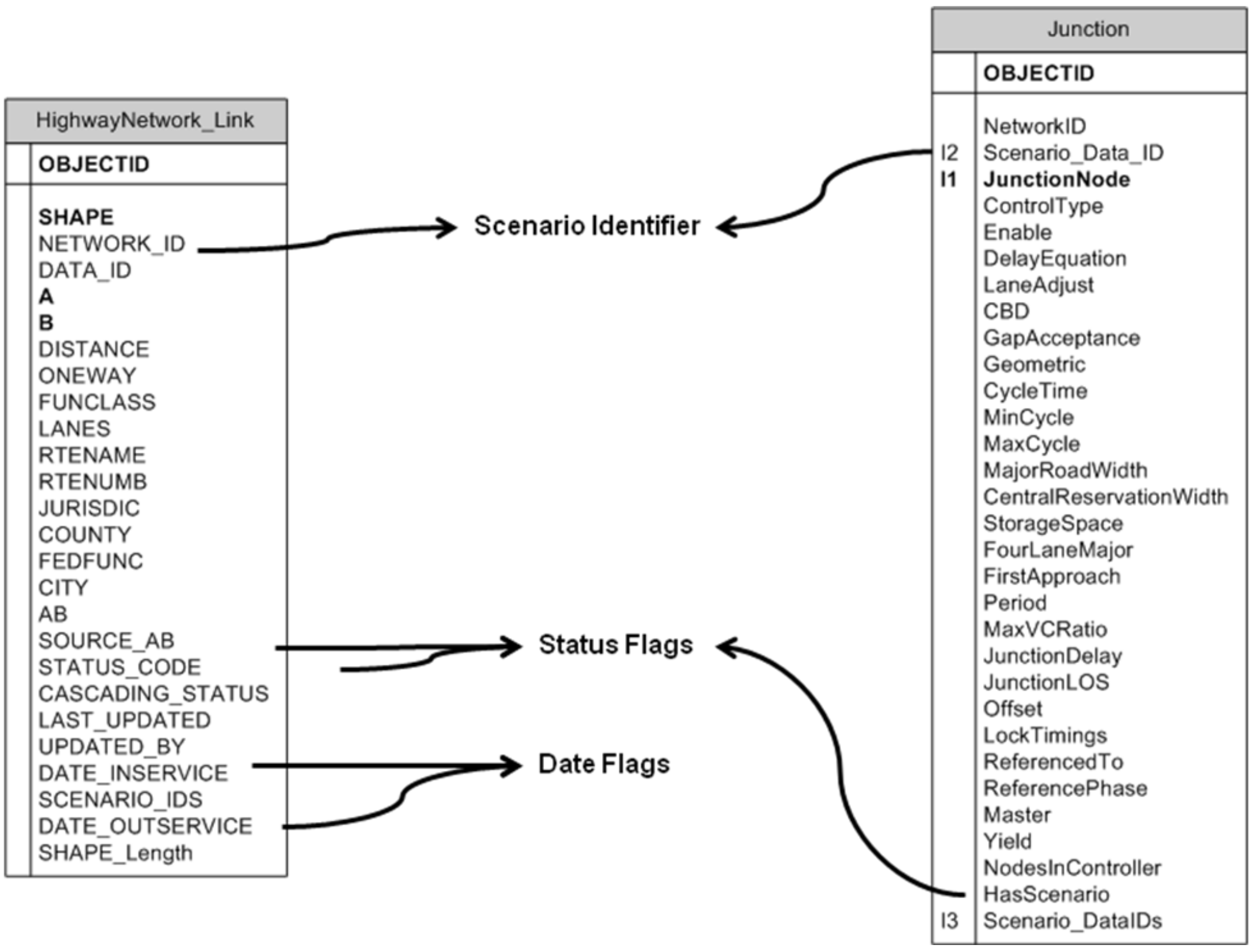


***Figure 3** – Example of Scenario Based Network Editing*

Cascading Network Changes

Cascading network changes is to synchronize network changes from the base or parent network to child scenario networks, or to synchronize network changes between multi-modal transport networks.

Cascading network changes requires two basic elements exist in the master network:

1. Relationships: including both parent-child relationships and multi-modal transport network relationships. The parent-child relationship ensures changes made in the parent network (due to errors or missing data) will be cascaded to all child scenario networks originated from the parent (including grant child, great grant child …); the multi-modal transport network relationship will assure changes from one network can be cascaded to other related networks. For example, a road network link is deleted in the base or parent network, this change can be cascaded in a related bus route network

2. Flag for cascading changes: not all network changes should trigger automatic cascading network changes. Flag for cascading changes provides a mechanism to allow user to determine if the changes should trigger an automatic cascading network changes or should manually handle the cascading changes. For example, a road network link is split in the base or parent network, this change may request cascading changes in a bus route network, but user has to manually determine how the change will be cascaded in the bus route network.

Cascading network changes is a complex process, in most cases it is up to the user to determine when and how the changes will be cascaded through the master network.

TRANSPORT NETWORK EDITING IN AN ENTERPRISE ENVIRONMENT

One of major differences between the enterprise transport network editing and single user based transport network editing is the handling of multiple users editing on the same network. Within an enterprise environment, network editing can occur with multiple users editing the same network (concurrent editing), or within a distributed environment (field editing, remote editing by contractor, etc.), so data protection, data synchronization and data integrity become critical issues for enterprise network editing. While protecting critical data from unauthorized users, the enterprise transportation modelling database should allow:

1) Concurrent network data editing: allow multiple users edit same data element (feature class or table) without data locking: the network is open for all authorized users to edit at the same time;
2) Versioned data editing: create protected versions of base data to allow designated user group to edit the data, versioned editing can be posted back to base data by administrator after reconciling conflicts between versions

3) Distributed data editing: create disconnected version of the base data (by check-in/check-out) to allow user edit the network in a distributed environment such as (at the field, or at a remote office); distributed editing is similar to versioned editing, changes made in the disconnected version can be posted back to the base data and conflicts between disconnected versions can be reconciled by administrator

Concurrent Network Editing in an Enterprise Environment

With the implementation of enterprise IT infrastructure, including enterprise databases, users from different locations within an organization are able to access the same data sources. When deploying transport modelling database in such environment, concurrent network editing by multiple users is inevitable. Managing concurrent network editing in an enterprise environment is not only to allow multiple users to edit the same network at the same time, but also to ensure edits from multiple users will not be overridden each other, and conflicts can be reconciled manually.

There are many different ways to manage concurrent editing issues in enterprise databases. One method that many database management systems use, for example, ESRI's geodatabase, is to hold concurrent edits (from multiple users) in several "delta" tables until the administrator posts the "delta" edits to the base and reconcile any conflicts manually.

Apparently, managing concurrent network editing requires a proper workflow to ensure multiple concurrent edits are synchronized and reconciled in time.

Version Management and Versioning Control Process

Version management is another way to manage multi-user network editing in an enterprise environment. A version represents a snapshot in time of the entire database. It contains all the datasets in the database. If managing concurrent network editing is for a single user group, then version management is for multiple user groups editing on the same dataset. Version management on an enterprise database allows multiple user groups to edit the same network based on different user groups' privilege on editing. For example, some transport agencies use network data shared with other user group, such as GIS, so GIS users and transport modelling users may edit the same network for different purposes. To avoid unnecessary conflicts in editing, two separate versions can be created from the same data source, each user group will only edit their own version, any edits from other user group are not visible until the administrator merge all versions back to the default version (root version) and reconcile any conflicts if exist.

One of key aspects in version management is to define user or user group roles and privileges, which determine what user can edit on what version. There are three types of versions can be created by either the administrator or by user: 1) private version; 2) public version; and 3) protected version:

- Private - only the owner may view the version and modify available datasets.
- Public - any user may view the version. Any user who has been granted read/write (UPDATE, INSERT, and DELETE or read/write) permissions on datasets can modify those datasets.
- Protected - any user may view the version, but only the owner may edit datasets to which he or she has read/write permission

Like managing concurrent editing, version management also requires a proper workflow in place to ensure edits in multiple versions will be synchronized to the target version, so users from multiple user groups can access the most updated network data. This workflow includes:

1. Create a version from the default version or target version and define user group to access and edit data on the version: the version can only be edited by the defined user group or user
2. Create a connection to the version for access by the users: instead of connecting to the source database, the connection only allows the users to access the version of the dataset
3. Load and edit data through the connection: using your application to edit the data through the connection
4. Reconcile conflicts: conflicts are detected during a reconcile when two or more users edit features that are in close proximity. There are two types of conflicts:

    - Those that arise when you save edits to a version and the same feature has been updated in that version in a different edit session (or is updated in one edit session and deleted in the other)
    - Those that occur when the same feature is updated in both the target version and the child version (or updated in one version and deleted in the other)

    When you reconcile, you may choose to either resolve conflicts in favour of the version you edited or resolve conflicts in favour of the target version.
5. Post changes to the default version: posting will merge your changes into the target version. Posting first saves your current edit session, then applies the target version to the current version. After posting, all the users will be able to access the most current data.

Distributed Network Editing

Distributed network editing is another way of managing versions in multi-user editing. Distributed network editing is to allow the user to check-out a version of the network to be edited in a disconnected environment, such as user's laptop or desktop disconnected from the agency's network. When the editing is completed at user's laptop or desktop, user can connect to the agency's network and check-in the edits to the source version. The check-out version can be in non-enterprise database format, for example, a check-out version can be saved as an Access database (.mdb). The process to synchronize edits in the check-out version to the source version is similar to the version management. The administrator will control the reconciliation and posting process.

Distributed network editing is very common in transport modelling practice. Many transport planning agencies hire consultants or contractors to edit the networks and develop the transport models. These consultants or contractors are normally working in a disconnected environment with the agencies' IT infrastructure. Merging edits from multiple disconnected network edits to the source version networks has become a challenging task for many transport planning agencies. The distributed network editing method will definitely help these agencies to synchronize network edits to the source version and to facilitate the transport modelling process.

ENTERPRISE TRANSPORT NETWORK EDITING WITH TRANSPORT MODELING SOFTWARE

Most transport modelling software does not support enterprise transport network editing due to the following reasons:

1. Network data are not stored in an enterprise database: file based network storage does not support enterprise network editing
2. Cannot handle concurrent network editing and versioning control: concurrent data editing and versioning control are the two basic functions of an enterprise database, without enterprise database, it is impossible to handle concurrent network editing and versioning control in transport modelling software
3. Cannot handle scenario based network editing and cascading changes: file based network editing disconnects a scenario network from its parent network, relationships between parent network and scenario network are not saved, which makes it difficult to cascade changes from a parent network to the child networks.

In order to support enterprise network editing, traditional transport modelling software needs a fundamental change in software architecture, changing from desktop based to client-server based or browser-server based architecture, changing from single user based licensing to concurrent user based licensing, changing from file based data storage to database based storage, and implementing an enterprise database with master network data model. This change will significantly improve the transport modelling software's functionality on data management and network editing to support enterprise network editing.

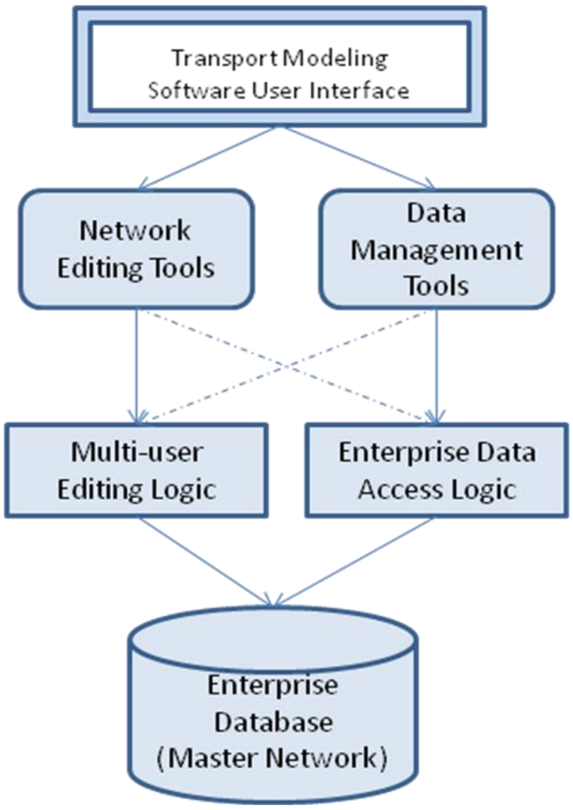

Transport Modeling Software User Interface
Network Editing Tools
Data Management Tools
Multi-user Editing Logic
Enterprise Data Access Logic
Enterprise Database (Master Network)

Figure 4 *– Transport Modelling Software Architecture for Enterprise Network Editing*

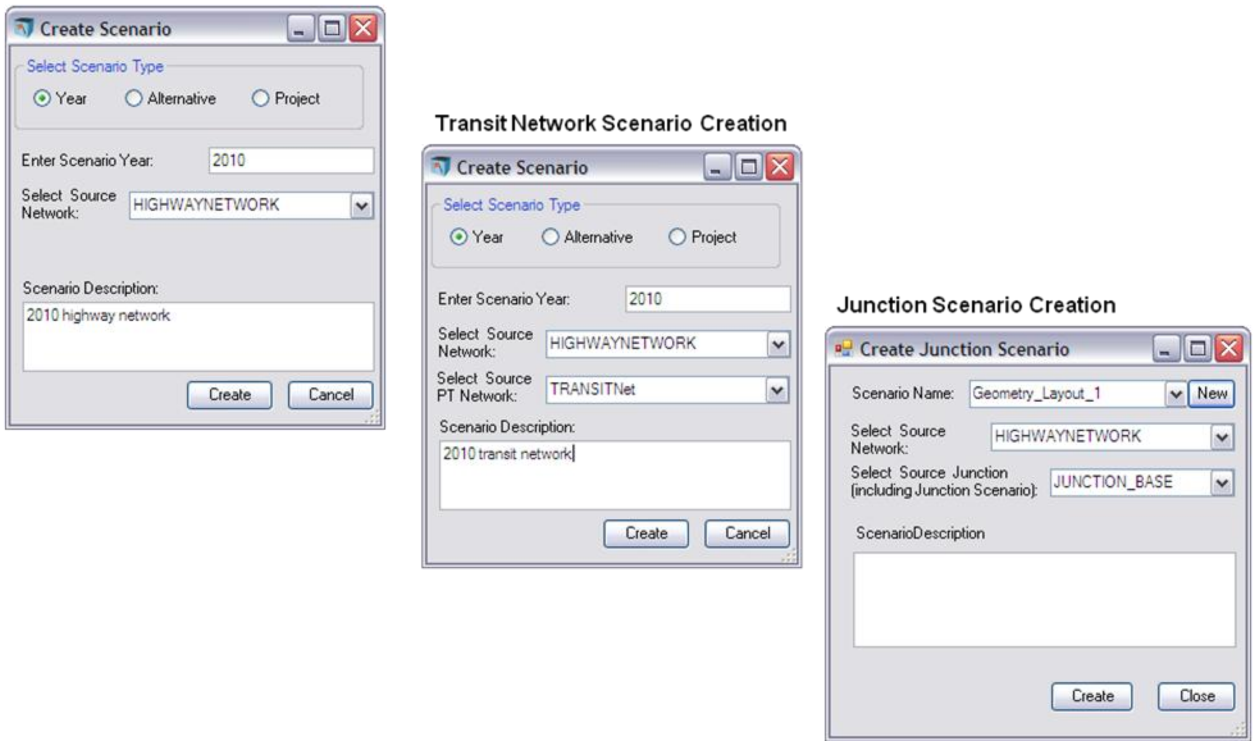
Data Management

Data management is a key function for handling enterprise transport network editing. When storing network data in a master network within an enterprise database, accessing a network is completely different than storing network data in a file or an Access database. Transport modelling software need to have a tool to manage network data stored in an enterprise database. This tool should have the following functions:

1. Create/Delete/Rename network: to mange network objects within the database, including scenario networks;
2. Copy/Paste network: to allow copying data within the same database or between different sources including scenario networks
3. Import/export network: to provide data transformation between different sources, including binary data file, database file and enterprise database, including scenario networks
4. Merge/join network with other attribute data: to provide data manipulation within and between data sources
5. Data versioning: to provide versioning control within enterprise database
6. Metadata management: update metadata within enterprise database

These functions allow a transport modeling software to access, manipulate and manage data within an enterprise database. Within these functions, managing scenario network data is very tricky. Some functions may not be applicable to scenario network, for example, the

Copy/Paste network function, if a single scenario network is copied, all parent-child relationships are lost, unless all child scenario networks are also copied; same as the Delete network function, should all child networks be deleted when a parent network is deleted.



Create Scenario
Select Scenario Type: [✓ Year] Alternative Project
Enter Scenario Year: 2010
Select Source Network: HIGHWAYNETWORK
Scenario Description: 2010 highway network

Transit Network Scenario Creation — Create Scenario
Select Scenario Type: [✓ Year] Alternative Project
Enter Scenario Year: 2010
Select Source Network: HIGHWAYNETWORK
Select Source PT Network: TRANSITNet
Scenario Description: 2010 transit network

Junction Scenario Creation — Create Junction Scenario
Scenario Name: Geometry_Layout_1
Select Source Network: HIGHWAYNETWORK
Select Source Junction (including Junction Scenario): JUNCTION_BASE
ScenarioDescription

***Figure 5** – Data Management Tools for Enterprise Network Editing – Create Scenario Network*

Support Scenario Based Network Editing

In the master network data model, there is significant difference how a scenario network is stored and is accessed than that in file based network storage. Since only the changes of a scenario network to its parent network are stored in the master network, access to a scenario network is a query to the master network based on scenario definitions.

This query consists of all the changed records in the scenario network and all unchanged records from the parent network. When editing such a query, the software must have built-in logic to deal with various editing scenarios:

1. Add new network elements: new links and nodes are added to the master network and identified as the scenario
2. Modify existing network elements (from parent network) including geometry and attributes: the existing network elements associated with the parent network will not be deleted or overridden, but flagged in status, and modified elements will be added to the master network and identified as the scenario
3. Split existing network link (from parent network): the existing link from the parent network will be flagged in status, and two new links after split will be added to the master network and identified as the scenario

4. Delete existing network link (from parent network): the existing network link associated with the parent network will not be deleted, but flagged in status

Due to the complexity of the scenario network editing, the software will also provide options for user to either save the edits to the source, or to an existing scenario or to a new scenario. The figure 6 illustrates the process when editing a scenario based network in transport modelling software:

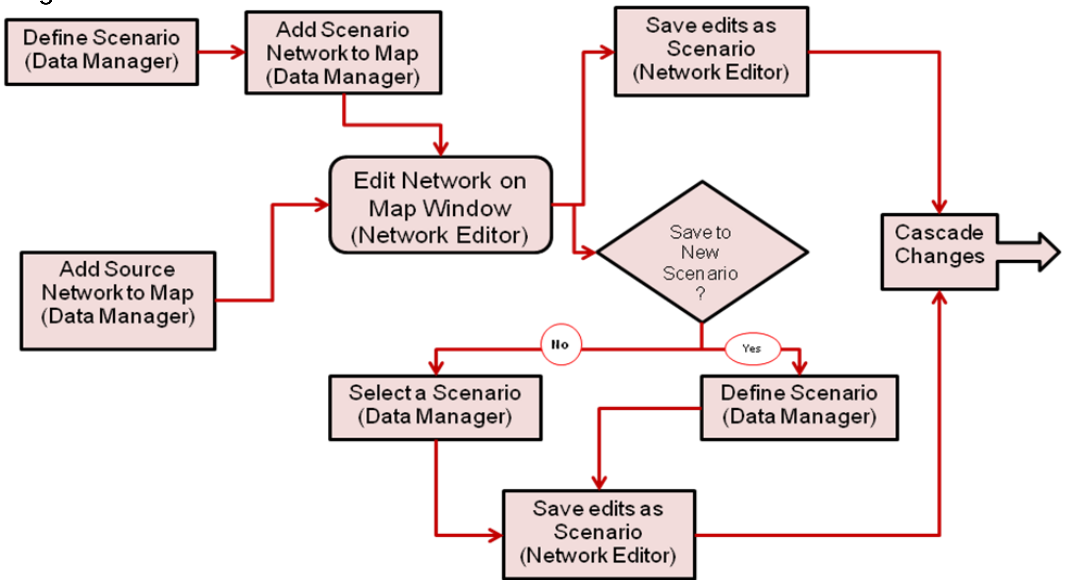

Define Scenario (Data Manager)
Add Scenario Network to Map (Data Manager)
Save edits as Scenario (Network Editor)
Edit Network on Map Window (Network Editor)
Save to New Scenario ?
Cascade Changes
Add Source Network to Map (Data Manager)
No
Yes
Select a Scenario (Data Manager)
Define Scenario (Data Manager)
Save edits as Scenario (Network Editor)

***Figure 6** – Scenario Based Network Editing Workflow*

Support Cascade Network Editing

The master network data model already provide a mechanism for cascading network changes, the software will implement this mechanism after a network is changed, including changes being made to the base network are cascaded to all child scenario networks, by doing so, cascading network changes becomes part of network editing process.

After a network has been modified, whether a network element's attributes have been modified, or a link is split or an element is deleted, the software will search all child scenario networks and related multi-modal transport networks to flag cascading changes. Then the software will prompt a dialog for user to select automatic cascading changes or manual cascading changes. When manual cascading changes is selected, the software will walk through each flagged record to allow user to determine if the change should be cascaded.

By adding cascading network changes to the network editing process, users no longer need to worry about inconsistency between different networks, eliminating the headache to manually search for cascading changes.

Support Versioning Control

For most transport modelling software users, version management or versioning control is a mystery. However version management is one of key aspects in enterprise network editing. Although most version management functions are provided by the backend database management system and handled by database administrator, transport modelling software also needs to provide functions to interface with the underlying database version management functions.

Some database management systems require that a dataset needs to be versioned before it can be edited. To comply with database management system's version management requirements, any data objects, including network objects created by the transport modelling software need to be registered as versioned.

As mentioned above, the database versioning control process also includes reconcile conflicts. Due to the complexity of the enterprise network editing, the conflicts not only include edits on the same entities by multiple users, but also include duplicated records due to editing on different versions, such as duplicated node number. When reconciling network editing conflicts, a validation procedures to check network topology will be needed.

Apparently, traditional transport modelling software will need to have some fundamental changes in order to support enterprise network editing process.

CONCLUSION

Transportation planning and demand forecast modelling processes rely on a wide variety of data and information. Editing transportation modelling data and maintaining data integrity are important steps in demand forecast modelling processes. This paper proposed a solution to implement a master network data model within an enterprise database to support enterprise network editing while maintaining data integrity. The solution addressed two fundamental issues in the enterprise network editing: the handling of scenario based network editing and the handling of network editing in an enterprise environment, and discussed the strategies for transport modelling software to support enterprise network editing.