
Managing models in the age of open data

Paper

Prepared For

AITPM National Conference 2016

By



26 July 2016

Peter Davidson Consulting Pty Ltd trading as TransPosition
Level 5, 199 George St Brisbane, QLD 4000

Table of Contents

| | | |
|-----|--|----|
| 1 | Introduction | 2 |
| 1.1 | The use of spatial databases | 2 |
| 1.2 | The 4S model | 3 |
| 1.3 | Underlying principles | 4 |
| 2 | Infrastructure Network – Road and Active Transport | 7 |
| 2.1 | The infrastructure network | 7 |
| 2.2 | Data sources | 8 |
| 2.3 | Licensing Issues | 8 |
| 2.4 | Creating a network from GIS layers | 9 |
| 2.5 | Node numbers | 10 |
| 2.6 | Network Connection Points | 10 |
| 2.7 | Directional Points – Link Transitions | 11 |
| 3 | Public Transport Networks | 16 |
| 3.1 | Multiple layers | 16 |
| 3.2 | PT Fares | 16 |
| 3.3 | Data Sources | 17 |
| 4 | Network options | 18 |
| 5 | Conclusions | 21 |

Abstract

Traditionally transport models have been developed through a largely manual process, manipulating and editing input data to create a stand-alone transport network for modelling. This was appropriate when most data had to be manually collected, and the model could be the source of truth. That is not the case now, since there are many sources for good quality data that can be used for modelling. This should change the way in which we manage our models; they should be defined by a repeatable assembly process so that they can be easily updated.

This paper addresses some of the key principles that should be adopted when managing data, and briefly mentions some of the legal and operational challenges. It also discusses some of the approaches that have been used and developed by the authors to manage data required for modelling, focused on building road and public transport networks and managing scenarios.

1 Introduction

When the first transport models were created, there were few sources of data – almost everything had to be manually collected, coded and mapped. At this time it was reasonable that a transport network would be created by hand, and when it came time to test options, the natural choice was to go in and edit the network. Similarly for zoning systems, demographics, intersection data and counts. Today the context for modelling is very different – there are many sources of data and much of it is maintained and improved by other people for their own purposes. However many of our approaches to modelling data have carried through with little change. Too often the approach is to extract, copy and edit data – losing the links to the source and obscuring the provenance of each input and assumption. Scenarios are often prepared by copying and editing, and even when options are individually coded, they are usually done in a way that would not survive an update to base data.

This paper discusses some of the approaches that have been used and developed by the authors to manage data required for modelling. Three aspects of transport model data are addressed – road and active transport networks; public transport networks; and the management of network options. This paper shows how road network and option data can be specified through loose overlays to externally maintained data (such as Queensland's State Digital Road Network, or the Open Street Map network), so that every element has a clear "source of truth". This allows the model to be regularly updated with the most current underlying data.

We have also developed similar tools for managing the complex inter-relations between sources for transport and land use data, and connecting this information to the transport network. These approaches will be discussed in a future paper.

1.1 The use of spatial databases

Before discussing new ways of processing data, it is useful to consider how that data is stored. Transport modelling has existed for much of the history of computing; in fact the first urban transportation study, the Detroit Metropolitan Area Traffic Study (DMATS, 1955) was prepared using IBM 407 electromechanical accounting machine. For this reason, many of the standard approaches to storing model data were developed without reference to the more recent developments in data management. It is still common for model data to be stored in proprietary formats, with all inputs and outputs mediated by the software package. Some systems use file based data, and some have their own consolidated data bank. Even where data is stored outside of the model, often using a Geographical Information System (GIS), the situation is sometimes not much better, with deeply nested folders of files on a network share.

A much better approach is to use the technology that has been developed for storing, managing and accessing data – the relational database management system (RDBMS). These exist as a number of commercial and open source packages, including Oracle,

Microsoft SQL server, IBM DB2, PostgreSQL and MySQL/MariaDB. These systems store data as tables (with rows and columns) with clear and enforced relationships. They are designed for shared access to large amounts of data, and allow for standardised access and analysis of the data using the Structured Query Language (SQL). Almost all systems now support the storage and analysis of geographical data, which can be accessed by most GIS packages.

There are many advantages to using a spatial database for storing transport modelling data. These include

- maintain data consistency through referential integrity
- maintain clear separation between data and process (see following section)
- a consistent location for storing all data, including model inputs and outputs, survey data, traffic counts, demographics etc
- powerful analysis and querying, using SQL
- integrates with all other tools, including statistical analysis, scripting, custom software
- independent of GIS and modelling package – allows a variety of systems to be used
- easily allow multiple users to access the data, with fine-grained control over security

Some of the more recent versions of transport modelling packages allow the database to be used for inputs and outputs, but even if they do not, then simple scripting can often be used to convert the data.

The disadvantages are

- the commercial packages can be expensive, although open source packages are available, as are free versions of the commercial software for smaller databases (<10GB)
- needs a computer to be configured as server
- some training is required to set up and use the system

1.2 The 4S model

TransPosition has developed a new modelling approach that seeks to provide a complete alternative to traditional four step modelling. It is based on a very flexible random utility model with Monte Carlo sampling of an integrated route/mode/destination choice structure. It is also very efficient; a multimodal model of the whole of Australia with every single road, every public transport service and every off-road walk/cycle path can be run in 6 hours. It eliminates many of the artificial constructs of traditional models (such as zones, centroids, matrices and skims) and is usually run with complete networks. In doing so it vastly simplifies the process of creating new models, since many of the manual processes are eliminated – this includes coding centroid connectors; choosing which roads to be included; and preparing consistent zonal demographics.

Many of the processes described in this paper have been developed alongside the 4S model, and are particularly well suited to its flexible and streamlined structure. Nonetheless, all of the methods are still applicable to traditional models.

1.3 Underlying principles

There are a number of principles that are useful guides to the way in which data should be treated.

1.3.1 The robustness principle

Be conservative in what you do, be liberal in what you accept from others

— Jon Postel (1980)

This principle was first stated in an early specification of the Transmission Control Protocol (RFC 761), one of the core foundations of the internet. In the context of this paper, it seeks to prefer flexible, fuzzy use of data rather than brittle use. For example, one way of ensuring consistency between two data sources (for example, intersection details and road data) is to have a common identifier (a node number, for example). But this is a brittle connection – if we seek a new source for one of the data sets then it is unlikely to have the same identifier. We could also use an exact coordinate – if the coordinates match exactly then the data should join. But again this is brittle – if one of the data sets is reprojected, or has come from a different base map, then the data will not match. A liberal approach is to allow coordinates to match within a tolerance. This is a fuzzy approach, that may cause problems with invalid linkages. However it is often possible to have a fuzzy connection in one dimension (a coordinate, for example), that is constrained by another fuzzy connection in another dimension (a data attribute). This allows maximum flexibility in dealing with changes to source data – an important consideration when other people are independently changing the source data.

1.3.2 Maintain good metadata

Without metadata, any data is just a collection of bytes. Metadata is “data about data” and is essential to interpret and contextualise data; recognise its source; and understand any limitations. There are international standards for metadata (particularly ISO 19115:2003 Geographic Information – Metadata) and metadata exchange formats (SDMX and DDI), and these are useful for formalising the process. In many cases a more informal approach may be suitable.

1.3.3 Separation of data and processing

Often what looks like data is actually a combination of data and processing. One example is a spreadsheet, where some cells are entered and some are calculated. Sometimes this is systematic and obvious, but often the data cells are intermingled with the calculated cells, or the equations change throughout the data. This makes it much harder to understand what is the actual data. If this spreadsheet is then exported to another format, or imported into a database or modelling software, then the distinction is completely lost. A less obvious example can come from GIS layers that have been subject to some manually selected algorithms (like buffering or a spatial join) – once the algorithm has been run the layer is modified but the processing history is lost. This again makes it impossible to differentiate between the original data and the processed results. A better system would store each component separately – the actual unmodified data; easily repeated queries and processes; and any stored outputs.

One useful tool to maintain separation is the database view. A view can be used in many ways like a table, but it is defined by a SQL query. The query can be quite complex – it can join tables; set default values; apply overrides; and transform data. This makes it much easier to audit and understand later in the project than a simple list of data values.

1.3.4 Everything goes into the database

This is not a philosophical principle, but a practical one – if everything is in the database then it is always possible to join and process data using common tools; and there is never a doubt as to the definitive version of a data set. This is not true of any file based system.

There are very few data sets that cannot be stored in a spatial database – even skims and matrices can be usefully stored and processed. The possible exception to this rule is very large data sets that are not easily analysed, such as aerial photography; CAD designs; or LIDAR data. There are methods for storing this sort of data in a database, but it may not be worth the effort.

The application of this rule may mean some processing work when data is provided; a common problem is data in a spreadsheet or document that is not well structured. The following principle on data normalisation can make this more complicated. However if the data is useful, it is generally better to take the effort of bringing it into a structured format suitable for entry into a database. The original files should be kept for auditing and verification purposes.

Another useful approach that we adopt is to ensure that any spatial data that goes into the database is reprojected into a common spatial reference system (SRS) – we use Long/Lat WGS84 (EPSG:4326) coordinates because we deal with data from all around the world, and it is consistent with coordinates collected from GPS. For more localised databases a more localised SRS may be more suitable – for example the ABS uses Long/Lat GDA94 (EPSG:4283).

1.3.5 Data normalisation

[Every] non-key [attribute] must provide a fact about the key, the whole key, and nothing but the key.

— Bill Kent (1983)

The main idea of data normalisation is that data should not be repeated – any duplicate data can lead to data integrity errors since the repeated data may be inconsistent. This can be managed by ensuring that every table has an index, and only attributes related to that index are stored in the table. Any secondary data should be stored in a separate table, with foreign key relationships between the tables. This leads to the 'Relational' in Relational Database Management System. There are times where strict normalisation can be difficult, particularly for information designed to be read by humans, but this can usually be managed by database views and reports.

2 Infrastructure Network – Road and Active Transport

Generally the most important component of any transport model is the network. In a multi-modal model this will consist of three parts – the road network; the public transport network; and the active transport network (including attributes of the road network on cycling and footpaths). At a more abstract level a multi-modal network can be seen as an infrastructure network (with varying attributes) and a service network that operates on top of the infrastructure network (for public transport services). The management of the service network is considered later in this paper.

2.1 The infrastructure network

The infrastructure network contains all road links, as well as any off-road walk and cycle paths. It may also contain rail lines, and ferry links, so that these services have links that they can operate on.

A key aspect of this network is geometry and connectivity. In the traditional approach this is done with a series of links and nodes; each link is defined by an *Anode* and a *Bnode* with a fixed number of attributes on each link. The links will generally have the following attributes

- Length
- Allowable modes
- Posted speed
- Road type / hierarchy
- Capacity or information that can be used to derive capacity, such as number of lanes

This approach is very easy to understand, and it has a long history. It is the default approach used by most modelling packages. There are, however, a number of weaknesses to this approach.

1. In order to make changes to the network, new nodes must be added, and often links will need to be split. This must be done using some centralised allocation of nodes to ensure that there are no numbering conflicts. And even then, the merging of changes to separate networks is somewhat complex.
2. Many of the link attributes will have been set using simple rules or default values. Some will have been manually adjusted. However it is impossible to know whether any particular attribute is a default or an override.
3. It is difficult to trace any changes to attributes. This can be done in documentation or revision control, but it is difficult to see in the data.
4. The network is generally created from street centrelines and possibly attributes from other sources. However once it has been created, the links to those original

sources are broken. If there is an update to the external source, it is generally a labour intensive process to incorporate these changes.

The last of these problems is key; the traditional approach to constructing networks is a semi-automatic/semi-manual process that creates a new stand-alone artefact. This artefact must be maintained and updated independently of its sources. The primary goal of the work described in this paper is to eliminate any manual processing from network creation and make it a repeatable, automatic process. Two secondary goals are to make it fast enough that it can be repeated every time the model is run; and to make it “fuzzy” enough that it can still work even if there are changes to the underlying spatial data.

2.2 Data sources

There are a range of possible sources for network data, depending on the level of detail required. At the most basic level is street centreline data, often obtained from processing of a Digital Cadastral Database (DCDB). In many jurisdictions (particularly those with Open Data policies) this data is freely available, but it is quite limited. It generally contains only the horizontal alignment of the road, and its name. Nonetheless it can be used as a good starting point for basic network and connectivity data.

There are also a number of commercial products that contain road networks, some with full routing information such as speeds, turn restrictions and congestion estimates. These can provide good quality data, but do lead to some licensing risks, as discussed below.

Another alternative is to use open source data, the most prominent of which is OpenStreetMap (www.openstreetmap.org). This is a crowd-sourced database of world-wide mapping data, including – road networks; points of interest; commercial centres; schools; airports; parking; and many other elements. The quality of the data is generally quite good, but because it is dependent on people’s contributions, there are areas with missing or inconsistent data. The big advantage of this data source is that if there are any errors or omissions they can be easily fixed. Any use of the data should involve some scrutiny and allow for time to be spent correcting problems, but these will then be incorporated into the main database. This makes it an ideal platform for government agencies to get behind; any effort spent on improving the data will improve the quality for the whole community.

2.3 Licensing Issues

Disclaimer: This section discusses copyright and licensing. The authors have no legal qualifications and any comments here should not be taken as legal advice.

The licensing of data in Australia is somewhat complex, and has been refined in 2009 and 2010 by two significant court cases *IceTV Pty Ltd v Nine Network Australia Pty Ltd*

and Telstra Corporation Limited v Phone Directories Company Pty Ltd. Both of these cases dealt with simple directory-type data (TV schedule and phone directories), and were judged to have insufficient independent intellectual effort to qualify for copyright protection. A transport network will generally involve much more labour and creative exercise of skill and judgement, and so may still be subject to copyright and thus limited by the licensing conditions of the data provider.

A transport modelling network derived from a commercial road network product is likely to be classified under copyright as a derivative work, but unless the original license allows for it, there may be problems in providing the network to others.

The constraints on distributing data are not present for OpenStreetMap – the data is licensed under the Open Data Commons Open Database License (ODbL) which allows free copying and distributing of the data as long as you credit OpenStreetMap and its contributors. However there is a requirement that any derivative databases be issued with a compatible open license, which may make it difficult to prevent downstream users of the network from exercising their full freedoms under the ODbL. This can cause its own problems if the modelling networks are intended to have only limited availability.

The data independent techniques described in the remainder of this paper may make it possible to avoid creating a derivative database, by separating out the base network data from the separately licensed modelling additions. Instead of distributing a fully prepared network, the distribution would include just the additional information and tools necessary for downstream users to generate their own networks. This extra information can be independently licensed, and downstream model users can obtain their own licence to the base data (either commercially or under an open license).

2.4 Creating a network from GIS layers

There are two ways of viewing an infrastructure network; geographically as a series of polylines; or topologically as a series of links and nodes. Geographical Information Systems (GIS) usually focus on the first view, and transport models focus on the second. Creating a transport network is partly just a conversion between these two ways of viewing the network. Note that in some cases, such as OpenStreetMap, the spatial layer is constructed from an explicit topological model.

The key task is to identify network connectivity from the spatial connection between links. This can be complex if the street layer has not been constructed with connectivity in mind. The most obvious way of identifying links that should connect is looking for coincident points, but this is sensitive to very minor changes to point locations. However if the connections are too fuzzy, with too much flexibility in determining which points coincide then the final topology may be incorrect. One example is a freeway overpass, where the two roads do not connect but may have points which are very close to each other. Fortunately many roads layers have been constructed to be both spatially and topologically correct, so this is often not a problem. However problems

can emerge if multiple sources are being combined – at state borders, for example. This can also occur if modelling options (such as a new road) are being combined with base infrastructure data. In these cases the coordinates will almost certainly not align exactly.

2.5 Node numbers

The most common way of identifying links and nodes in transport models is to use node numbers. This approach works well when the network is static but is problematic if the network is to be automatically built or is part of a hierarchy of models. We have found that the best approach is to specify everything geographically using the techniques described in this paper. If node numbers are required for the running of the model then they should be assigned as necessary but not used as persistent identifiers. Thus no information should be coded with node numbers. This means that traffic counts; turn bans; toll locations; and centroid connectors should all be coded geographically.

2.6 Network Connection Points

The approach that we have adopted to this problem is to explicitly identify Network Connection Points. These are indications to the network building process that some flexibility should be assumed in spatial coordinates at that location. In effect they become localised attractors that draw other points in range to their location. The construction of Network Connection Points must be done manually based on an investigation of the networks to be joined. The coder can then ensure that there is no ambiguity in the connections – if necessary the connection points can have a modified tolerance in areas where there are multiple potential connections. The Network Connection Points can also be used for error checking – if the final network does not have two links connected at each connection point then something has gone wrong.

When we apply these network connection points, all polylines in range are searched to find the potential connections. For each of these lines the point of closest approach is identified – this may be at the end of the line or at some intermediate point. The algorithm then applies the following logic.

For each connection point

 Find all lines within range (using manual tolerance)

 For each line in range

 Find the point of closest approach between the line and the connection point

 Identify if the point is an endpoint or an intermediate point

 If the intermediate point is close to the end of the line

 Use the end point instead

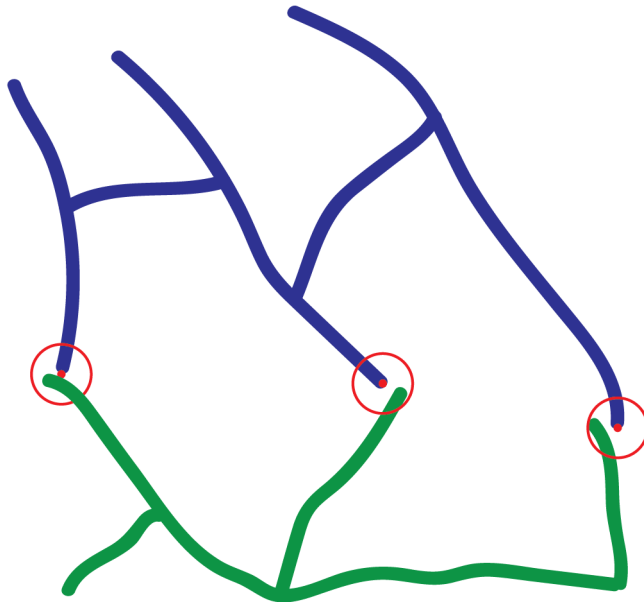
 Add the point to a set of adjustment points

 If all adjustment points are end points

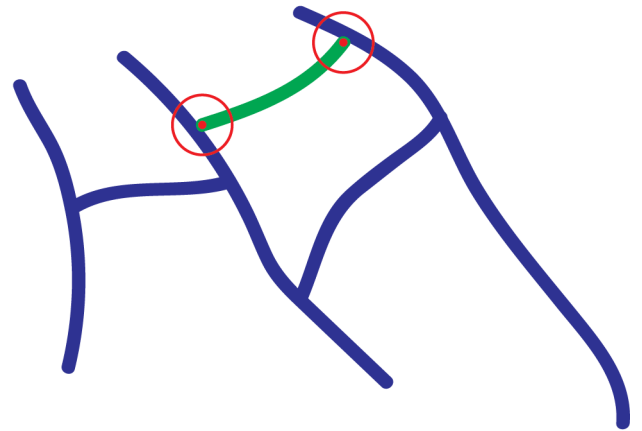
 Extend each of the adjustment lines to the connection point

Figure 2.1 Connection Points

NETWORK CONNECTION POINTS - JOINING NETWORKS



OPTION CONNECTION POINTS



Else if only one adjustment point is an intermediate point

Extend all of the end point adjustment lines to the intermediate point

Else

Extend all end points to the connection point

Add a new mid point to all intermediate points to the connection point

For the network options used for testing scenarios we find it useful to separately identify Option Connection Points – these are the points on the network to which the option should connect. Note that these do not have to align with existing nodes in the base network.

2.7 Directional Points – Link Transitions

As discussed earlier, the base information available from external road databases generally does not have all of the information needed to build a transport model. Even the more detailed commercial sources that contain full routing information do not generally have sufficient information to develop capacity estimates and speed/flow relationships. It is tempting to simply edit the data – add some additional fields and code the new information into the database. An alternative approach is to construct the model network and then do the editing there. In fact this is the most common approach, where network attributes are edited in the transport modelling software. However this breaks the connection to the source data; if a new version of the roads database becomes available then this information is lost. It also makes tracking data provenance more difficult.

Other approaches to data have been used. The approach that was commonly used in the pre-GIS days, and particularly for declared state roads, was to use chainage (or kilometrage) lengths along a gazetted roadway. So a change in speed or number of lanes could be identified by a starting chainage and an ending chainage. The benefit of this is that it is not dependent on the specific network details – adding nodes or changing which roads are included will not change the data. It also allows directional data to be specified. However using this approach more widely is somewhat problematic for a number of reasons.

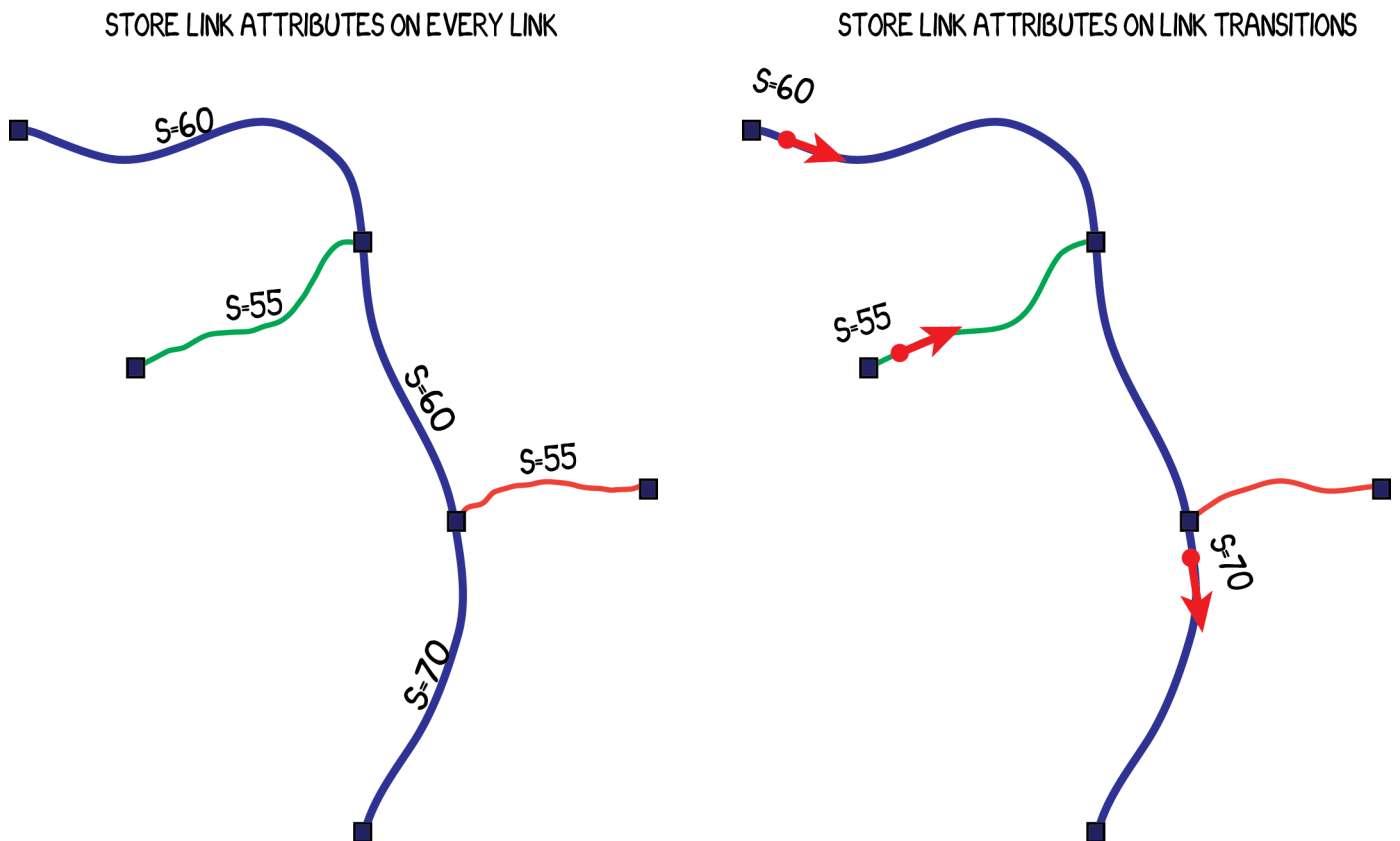
1. It depends on clearly identified roads, with a readily identified starting point
2. It requires each road to have a nominated direction
3. It depends on consistent measurements of length, which may not be possible with data from multiple sources
4. Any change to the road alignment will break all chainage measurements downstream of the change

An alternative approach is to code data at a point, and allow that data to be spatially joined with the road network. This may be suitable for data that applies at a point (such as a traffic count, bus stop or pedestrian crossing) but is tedious to use for any data that applies along a length of road (such as number of lanes or posted speed).

The best approach that we have found is a fusion of the point and the chainage approaches which we call Link Transitions. A link transition is simply a point with a bearing (mathematically it is a unit vector) – this corresponds to the chainage point on a road, but it is specified using standard Cartesian coordinates rather than linear coordinates along the road. The link transition can specify the start or end of an attribute change, and will automatically apply that attribute to all sections of the road between the transitions points. The bearing allows the direction of the attribute change to be identified, and is similar to the with-gazettal or against-gazettal attributes on a chainage point.

Using a fully specified bearing may appear to be overkill – a very coarse bearing (or even N/S/E/W) could be used to identify which direction of a road is being referred to. However a full angular bearing allows for better road identification when the locational data is fuzzy. This is particularly important in areas of high network density, such as ramps on a freeway interchange. One of the goals of this approach is to allow for some differences in the underlying base data, but a simple point could easily be within range of multiple roads. By qualifying the match on the bearing, the algorithm can ensure the correct identification of the road even if the coordinates are incompatible. Of course an exact match of bearing is not required, but the combination of location and bearing can eliminate most ambiguities. Any remaining problems can be identified and solved through more careful coding (placing the link transition at a slightly different location that is less ambiguous).

Figure 2.2 *Link Transitions*



However one problem remains – since we do not have a clearly gazetted road, how can we identify the start and end of each road. The only real candidate is to use the road name to determine road identity. This is somewhat problematic, since some roads are not named; there is not always consistency on when names change; and the same name could be used on multiple roads. In order to make the process work, we first construct a unique road name identifier. This is prepared by identifying contiguous sections of identically named roads. By requiring them to be contiguous we avoid problems of similarly named roads, and isolated sections of unnamed roads. Some flexibility in the contiguous test is desirable, though, since we have found that there are sometimes small sections of differently named roads at ramps and roundabouts that can sometimes break what should be a single road.

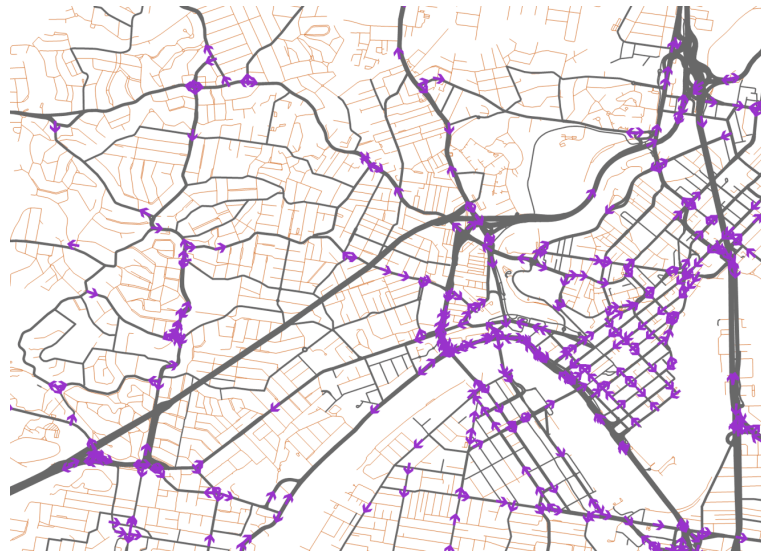
To simplify coding, and eliminate extraneous end points, we have adopted a range of different Link Transition types.

1. Merge Start – notes that the attribute changes at the closest intersection back from this point, and applies to all sections of this road until the end (unless an End Transition is found)
2. Split Start – splits the road at the transition point, and applies the attribute change
3. Merge End – notes the end of an attribute change at the next intersection forward from this point

4. Split End – splits the road at the transition point, and ends the previous attribute change
5. Single Link – applies an attribute change only to the specified link (however that has been defined)
6. Single Block – applies the attribute change from the previous intersection through to the next intersection

The link transitions are easily created in any GIS system. Although it is a different way of thinking about data we have found that people can become quickly proficient at coding attributes using this approach. To make it easier to understand how the attributes will apply to a road, we create a GIS layer with the road network coloured by the unique road name identifier.

Figure 2.3 *Example application of Link Transitions*



The figure above shows a real-world application of link transitions. In this case it is being used to add posted speeds and number of lanes to a simple street-centreline based network. The LinkTransitions are combined with a smart system for setting default values based on hierarchy and surrounding land use. This means that only exceptional data must be coded - many links can stay at their default values. The link transitions shown here have been coded manually by undertaking virtual site inspections using Google StreetView. Each major road is rapidly “traversed” looking for speed signs and lane changes.

In other cases the link transitions can be created from actual site surveys using a custom smart-phone app and GPS determined points and bearings. They can also be created from existing location data, such as speed sign or traffic count databases. In this case the bearings must be added to the data; this can be done semi-manually by snapping the points to lines; attaching the bearings; and then reviewing the automatically assigned bearings. Depending on the source the directionality may also need to be checked. Any road chainage data can be converted automatically. We have

also done automatic conversion of data that has been coded using street/cross street identifiers.

3 Public Transport Networks

3.1 Multiple layers

Public transport networks are more complex than road networks – it is best seen as three layers built on top of each other. At the base is the infrastructure layer; then the service layer; and then the individual timetable trips.

The infrastructure layer can be managed using the techniques described in the previous section as long as rail lines, ferry routes, and dedicated bus-ways are included in the infrastructure network (with suitably qualified mode limitations).

The service layer shows the routes followed by each service and identifies the locations of stops. When focusing on the PT network all that really matters is the stop-to-stop linkages. However when examining traffic congestion or the interaction between buses and cars, the sequence of links making up the route is important. Traditional PT coding explicitly identifies this sequence of links, but in most cases it can be automatically determined by building shortest paths between stops. Where there is ambiguity intermediate dummy stops can be introduced to force a particular path. This approach is much better than explicitly identifying links because it is robust to changes in the underlying infrastructure network.

The individual timetable trips record the expected arrival and departure times at each stop. Some transport models use only service frequencies rather than a full timetable. In this case the trips layer is not required and the frequencies can be stored in the service layer. Implicit in the timetable is the travel time between stops; a frequency based PT network needs to store average stop-to-stop times separately. The 4S Model uses complete timetable PT and thus makes full use of the trips layer. There are complexities in updating a timetable to incorporate congestion because any change in a stop-to-stop time will alter the whole timetable and is likely to have follow on effects on other services that use the same vehicle. We have explored methods to address this problem but they are beyond the scope of this paper.

3.2 PT Fares

Another difference between the PT network and the infrastructure network is due to the presence of fares on PT travel. Fares can be complex, particularly in the treatment of multi-trip discounts; time of day variations; student and pensioner discounts; multi-modal fares; and varying fare zones. Generally a fare zone system can capture most of the important details, but this is an area where there is real complexity that cannot be fully managed with a general solution.

3.3 Data Sources

In the past most public transport networks were constructed manually from published timetables. With many PT operators moving to integrated fleet management and timetabling systems, a newer approach was to use extracts directly from the operators' databases. But the best approach now is to use the General Transit Feed Specification (GTFS). This is a multi-layered data specification that has most of the timetable details described above. There are two big data sets missing in the GTFS that are necessary for most PT modelling - fare and capacity information.

As mentioned earlier, fares are complex and probably need to be managed manually. They can, however, be specified in a spatial format, so that they can be consistently applied to modified PT networks.

Information on PT capacity depends on vehicle characteristics that are not specified in the GTFS feed. This is an operational detail that again needs to be manually managed.

4 Network options

Preparing the base network is only the first step in managing a transport modelling network. The primary function of transport models is to test scenarios, so it is important that the coding and management of scenarios is done in a way that is compatible with the use of open data.

Network option coding is most often done in the same way that networks are developed – the network file is directly edited. This is sometimes done through “copying and pasting”, where the base network is copied as the starting point for a new network year. All of the network alternatives that apply in that year (including new and changed road links, intersection changes and transit network changes) are directly coded in the net network file. This approach has many obvious problems:

- it is difficult to prepare new alternatives that have different combination of options
- if the base network is changed then it must be manually copied through all of the other copies
- it can be difficult to compare results across the different networks
- managing the multitude of files that result (with only minor differences) can be challenging

An improvement to this approach is to code up network alternatives as a list of changes - effectively recording the sequence of operations that must be done to implement the option. The operations can be to add, delete or change a link. A scenario can then be assembled from the list of alternatives; the network process simply run the operations in order. This approach can work well, but is difficult to implement without unique network identifiers (*Anodes* and *Bnodes*).

By building on the earlier concepts of Network Connection Points and Link Transitions, it is possible to code networks in a way that is easy to change and can be compatible with a change in the underlying network. The process uses the following data layers

- OptionLinks - this contains all the new links that are to be added in any scenario (includes link attributes)
- OptionLinkConnectionPoints - shows where the new links should connect to the existing network
- OptionNodes - identifies nodes that are added or changed in any scenario (includes node attributes)
- OptionLinkTransitionPoints - shows the start and stop of any changes in link attributes (includes link attributes)

Each record in these tables (with the exception of the OptionLinkConnectionPoints) is tagged with a special OptionCode - a string that defines which option the change should apply to. The option codes allow all of the various changes to the network to be combined into meaningful groups. These could be complete scenario specifications (for example, 'Planned2031' for all changes that apply in the planned network for 2031)

but are much more useful if they define easily identified localised (atomic) changes. An example list of option codes could be

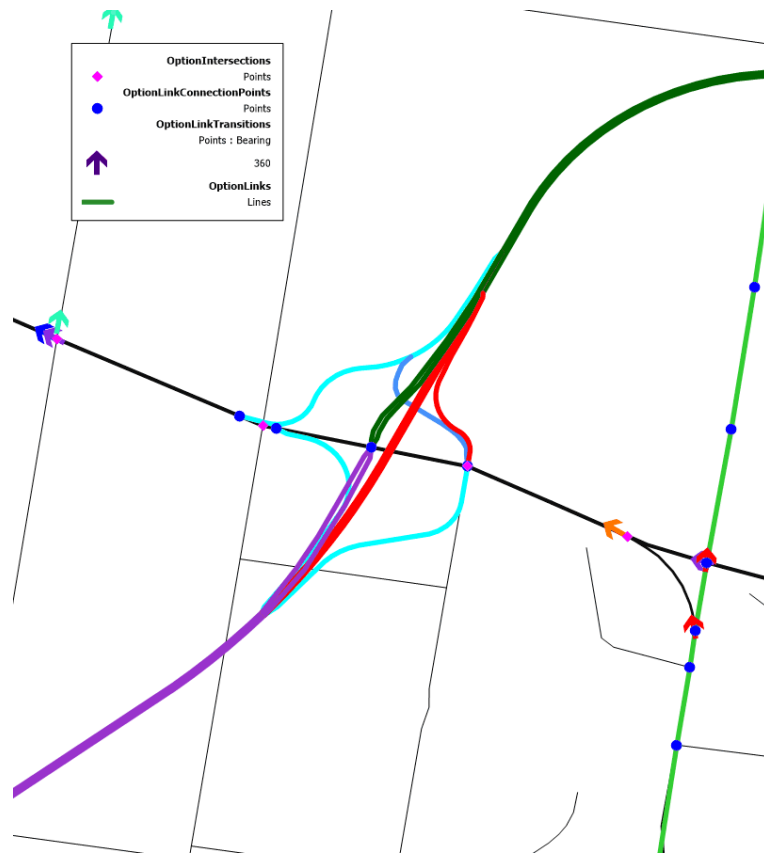
- Base: OptionCode for all base year features
- MainSt: Main St upgrade
- BypassNorth: the northern section of a new town bypass
- BypassSouth: the southern section of a new town bypass
- BoundaryStSignals: a set of new signalised intersections

Each list option may include multiple new links or changed attributes. This could also include marking that a link should be removed in an option. Since all of these options are specified spatially, with fuzzy connectivity, they are robust enough to survive a change in the underlying network. They can also be easily applied to network from other models.

Network scenarios can be constructed from sets of network options. The system that we use allows for multiple levels of grouping to simplify scenario management. Using the prior option codes, we might have the following scenario sets

- Base_2016: Base
- Full_Bypass: BypassNorth, BypassSouth
- Proposed_2026: Base, Full_Bypass, BoundaryStSignals
- Optional_2026: Proposed_2026, MainSt

Figure 4.1 Option Coding



5 Conclusions

This paper calls for a change in the way that we think about modelling data; rather than focusing on creating new data sets we should be focused on developing processes to bring data together. This ensures that the most recent data can always be used; allows a hierarchy of models to be more easily maintained; and can eliminate some of the legal issues associated with creating a derivative copyrighted work.

The change is best managed by rethinking the whole process of acquiring, creating and maintaining data. Rather than seeing it as a series of ad-hoc processes that are done when needed, the transformation and augmentation of source data for modelling should be treated systematically.

This may require more involvement from specialists in data management, including database administrators, GIS operators and software engineers. It probably also requires some reduced reliance on specific transport modelling software for managing the data, and increased use of standardised tools such as relational database systems.

A number of specific technical approaches have been identified by the authors to enhance the ability of spatial databases to store information about infrastructure networks - Link Connection Points; Link Transition Points; and Network Option management. Alongside suitable tools for incorporating them into the model assembly process, these approaches can allow the various external data sources to be enhanced with all additional information needed for modelling.

By adopting the principles and techniques described in this paper, models can be constructed more quickly and cheaply, and always kept up to date. Data and scenarios can be shared between different models within a model hierarchy; different groups of modellers; and between different modelling platforms.

References

Kent, William. 1983. "A Simple Guide to Five Normal Forms in Relational Database Theory." *Communications of the ACM* 26 (2). ACM: 120–25.

Postel, J. 1980. "RFC 761 - Transmission Control Protocol." Request for Comments. RFC 761; RFC Editor. doi:10.17487/rfc761.