7TH WCTR
SYDNEY AUSTRALIA
1995

TOPIC 12
GIS, LAND INFORMATION
SYSTEMS AND DATABASES

APPLYING THE TRANSPORT NETWORK RELATIONAL DATABASE TO A TURNING FLOWS STUDY AND A TRAFFIC NOISE MODEL

STUART J. THOMPSON-CLEMENT
Transport Systems Centre
University of South Australia
The Levels SA 5095, AUSTRALIA

JEREMY E. WOOLLEY
Transport Systems Centre
University of South Australia
The Levels SA 5095, AUSTRALIA

MICHAEL A.P. TAYLOR
Transport Systems Centre
University of South Australia
The Levels SA 5095, AUSTRALIA

Abstract

The Transport Network Relational Database is a set of relational tables for storing details of road transport networks. It is an effort to address the need for a common data format to aid those organisations involved with transportation modelling and analysis. The relational database approach is used due to its advantages of increased flexibility, the current emphasis on client/server computing and the ever-increasing power of the desktop PC.

INTRODUCTION

The Transport Network Relational Database (TNRDB) is a database structure for storing details of transport networks and is designed using the tenets of the relational database paradigm. It is intended to be used by transport system models for any variety of purposes in transportation engineering. Details currently included in the database design are physical dimensions such as node characteristics, the road sections in a network, lane dimensions, traffic management devices and safety barriers; logistic details such as turn restrictions, link use (eg parking arrangements), directional flow arrangements and speed limits; numeric data such as vehicle flow counts, turning flow counts and fleet composition statistics; and financial data such as tolls and parking costs. TNRDB is to be extended to cover indented bus stops; pedestrian crossings; the grouping or conceptual aggregation of the physical structures into centroids and corridors; travel time cost penalties; signal timing arrangements; bus routes; O-D flow matrices and fuel consumption statistics among others.

This paper discusses the impetus behind the development of a transportation database with a structure usable by many applications. The relational database approach is discussed by using the traditional file processing approach as a backdrop. This is followed by a description of the Transport Systems Centre (TSC) database approach with particular attention focussed on open database connectivity and client/server computing. An explanation follows about why the relational approach has been taken in two projects being carried out at the Transport Systems Centre and School of Civil Engineering of the University of South Australia: the Turning Flows Study and the development of the Traffic Noise Model, NetNoise. Each project is briefly described and the paper finishes with a description of projected work in which the TNRDB will be integrally involved.

TRANSPORTATION DATABASE AND SOFTWARE METHODOLOGIES

Transportation software inter-compatibility

There is a need in transport planning and analysis to use a data storage system that is compatible with many computer-based applications, from control programs to design and analysis tools. At present there are many such applications in use; each one using its own proprietary format for data input, manipulation and storage. There are considerable data on transport networks (in particular road transport networks) held and used at numerous locations within Australia but only in exceptional cases are data used at one location usable at another location without considerable data format manipulation.

Transportation software is renowned for being difficult to use. This is due largely to the complexity of the systems under study (eg urban road networks). Data preparation, focussing on 'network coding', has long been an unavoidable but tedious task in modelling and analysis. Previous attempts at common database structure formats have been rare partly because there has not been an urgent need for such a system (and hence no money has been made available for development and maintenance), partly because of the limitations of desktop computing power and partly because of the difficulties of inter-computer communications of the past. Horowitz and Pithavadian (1987) described one attempt to develop a transport network database structure. Recently the Australian Transport Software Integration System (Akçelik et al. 1994) has been proposed as an attempt to build an integrated data system for some of the Australian transportation software packages used in traffic signals design and urban traffic control. This system is a compromise between the traditional file processing approach and the database approach and consequently it is limited in that each package is required to read from and write to a common data file: all the relevant data managing is performed within the application software.

The Transport Network Relational Database was designed to enable commonality of format making the exchange of data between packages a relatively simple exercise. Using the relational database approach different transportation software developers would be able to design their database requirements using appropriate portions of the TNRDB and to disregard superfluous tables. Even the exchange of data between sites (eg the Transport Systems Centre and ARRB Transport Research) would be a matter of data transfer of the contents of relevant portions of the TNRDB using existing communications services. A particular need for commonality emerges in considerations of a connected hierarchy of transport network nodes (Taylor and Anderson 1983; Taylor 1991).

Relational databases v traditional file processing

Data used for computer-based transportation applications has traditionally been stored in some form of sequential file structure, and hence, in the absence of any recognised standards, each builder of a software system has defined and implemented the file structure necessary for that application. This means that for any application the definition of the data is embedded as a part of the program itself. Hence if the file structure is changed then so too must the programs that access the file be changed to suit. One disadvantage of this method is the lengthy development time for any query or report, as each must be specifically written. This translates to expensive processing of large flat files (Sayles et al. 1994) and since the cornerstone of the relational database approach is to have data reside in multiple independent tables (Date 1986; Elmasri and Navathe 1989), programmers have been able to retrieve information from a broader perspective than was previously possible and hence data retrieval has become cheaper. A further disadvantage of the traditional approach is that often two or more users maintain different files of the same data for two different applications. This clearly leads to wasted storage space due to the multiplicity of files, and hence redundant efforts would be made to keep common data current. This last problem is not confined to traditional file processing alone, but with the relational database approach using a common database structure it is a reasonably simple exercise to keep up-to-date data by combining two sets of data.

The database approach ideally has a single repository of data that is defined once and then maintained and accessed by various users within an organisation or even across organisational boundaries: this last is made possible by the rapidly advancing technology of the infobahn which enables many individuals to access data from many disparate sources. Wherever there is a single repository of data there is a need for some form of data management system and this applies equally to the database approach as to the traditional file processing approach. One advantage of the relational database approach is that the database management system (DBMS) software is not written for any particular database and hence the database approach using a DBMS can effect program-data independence (Ullman 1982). The DBMS holds a complete definition of the database in the database catalogue and the DBMS itself must refer to the catalogue to know the structure of the files for a particular database. Program-data independence is reflected in a reduction in application development time. This is a boast often heard whenever an application development environment is touted by its progenitors and one which rightly should be embraced with caution, but with a good DBMS there is indeed a reduction in prototype development time and *ad hoc* queries of the data of up to three or four tables are easy to build and execute. The combination of program-data independence and the use of a database catalogue provides users with a conceptual representation of the data, as users are not concerned with how the data is stored: hence the database approach encompasses data abstraction. This leads to the notion of support for multiple views of the data and since providing this is a simple exercise for most DBMSs the rules of primary database integrity are therefore built into the management system. Secondary integrity rules are included by the builder of an application and the DBMS must have some way of assimilating these seamlessly into the management processes for that application.

Briefly, a relational database consists of structure and data: as does any other database (Date 1983). The structure of a relational database can consist of one to many tables (as is the case with the TNRDB) but what makes the relational model distinctive is that the data in each row (record or tuple) of each table is unique and the data in one table (Table X) can be related to the data in another table (Table Y) by matching values in fields. This is known as a (relational) join. The most common form of join is to match the values of the primary key of one table with the values in an attribute field of a second table.

The values contained in the primary key of a table are unique for each record in that table and hence the primary key is usually a single data field (eg an ID number). In some tables the primary key may be a combination of fields. For example in a table of data showing the projects on which employees are working, the ID number of the employee combined with the project number could be used as the primary key. Similarly in a table of intersection nodes the longitude and latitude combined can uniquely identify each node record, though it is more usual and often easier to give a unique number to each node. Node numbering is also a requirement for network analysis procedures (eg minimum path calculations).

The Transport Systems Centre approach

The Transport Systems Centre has taken the relational database approach partly because of the increase in desktop computing power evident in recent years; partly because there are now several powerful database management systems on the market capable of handling the databases needed for applications in transportation modelling and analysis; and partly because of the gathering client/server revolution in the wake of the networking explosion. The proliferation of DBMSs goes hand-in-hand with the increase in computing power. The argument for the relational database approach is enhanced with the increasing ease with which links between commercially available software packages can be made. For example the open database connectivity (ODBC) paradigm enables an application package containing the necessary ODBC driver to query an SQL database and to use the data obtained for further calculations or reporting. It is most likely in practice that database tables would be exported to the user application rather than the tables being directly attached: this method offers the easiest and cheapest method of protecting the integrity of the original data though does not exclude some of the pervasive update problems of information management.

The client/server information processing model goes even further than this by splitting the tasks of an application between two or more machines. There is much confusion over the exact meaning and representation of the client/server model: this is not surprising since one of the main arguments for adopting such a system is that it can be very flexible. In all client/server models the server contains and manages the data while the application logic and presentation components can be distributed in varying amounts between the server and client (Philipson 1993). With a transportation database the fundamentals of a database management application would be provided and the onus for building the application logic and presentation layers would lie with either the keeper or the user of the data or some combination of both. With this approach it can be seen that accessibility by any user anywhere would be a matter of firstly electronic communication capabilities and secondly the level to which the user or client would be prepared to go to provide the application logic and presentation components of their application software system. This then does not depend on the DBMS employed by the server as the open database connectivity paradigm would allow any application package at the client end to descend to the direct query level of the database repository (ie the query portions of the database management component) and hence most logic and presentation components could be provided at the client's site.

So how does all this fit together? Suppose that the Transport Systems Centre wishes to build a software application program for a Travel Demand study. The data is to reside in a database of the TNRDB structure and the application is to be typically broken into three components: data management, application logic and presentation. The data management system could be provided by the Microsoft Access software package; the application logic could mostly be handled by Visual Basic with calls to more complex routines written in Pascal or C if necessary; and Visual Basic could be used for the presentation component or user interface. Alternatively Borland's Delphi environment is powerful and flexible enough to handle all three components. Of course not all of the tables of the full TNRDB description are necessary for the study. If the SA Department of Transport then wish to do further work using data from the original tables or on data resulting from work performed at the TSC and they use Paradox as their DBMS, then with the relevant ODBC driver the database can be readily accessed, as the driver itself does any and all necessary conversions transparently to the Paradox user at the DoT.

TNRDB MODELLING HIERARCHY

The hierarchy of models is a set of transport system level definitions and descriptions of some of the models used for analysis and simulation of various transportation scenarios. Its primary aim is to match particular modelling theories with relevant areas of application while providing a means to transfer data between these areas, as needed for purposes of analysis. The 7-level modelling hierarchy was proposed by Taylor and Anderson (1983). Later Brownlee et al. (1988) proposed the 5-level hierarchy. There are conceptual overlaps between the levels of both models. The comments below address the 7-level hierarchy though they could easily apply to the 5-level model with only minor changes required. The breakdown of the physical road network into the 7-level modelling hierarchy is shown in Table 1.

Level 1 contains the most network detail and level 7 contains the least. At the first level there can be simulation and analysis of individual vehicles, pedestrians and cyclists in a traffic stream or in a very localized area such as an intersection or carpark. This level can accommodate lane-based simulation and analysis. The highest level encompasses broad-brush sketch planning models concerned with the flow of people, vehicles and goods between regional, national and international precincts. One of the important aspects to consider when designing the TNRDB for use in the hierarchical modelling approach is to enable information transfer between levels and hence models. A detailed description of the way linkages between software packages can be used and the associated level of data checking can be found in Brownlee et al. (1988). Aggregation (supply generation) of information can be achieved as analysis moves from the lower level models to higher level models and disaggregation (demand generation) as the move is from higher level models to lower level models. The TNRDB is being designed so that aggregation and disaggregation operators can be built to perform those functions to extract the required data for software packages.

Table 1 Physical road network represented within the 7-level modelling hierarchy

Level	Objects	Data	Model
1	vertices arcs sectors nibs lanes	Lane-based, arc-based, vertex-based, individual vehicle descriptors, traffic signal settings, flows, turn counts, pedestrian activity, cyclists, parking details	microscopic; signal timing optimisation; covers single intersection; little, if any, forecasting; representation of individual traffic lanes on each approach
2	vertices arcs sectors	Aggregates of the above: average speed, gradient, flows etc	macroscopic; produces existing to short range forecasts
3	vertices arcs sectors nodes links	Aggregated link-based (usually) data: flows, travel time	dense network models (static); optimisation of fixed data; small-scale O-D matrices; flows on arterial/collector roads; turning movements at intersections; optimisation of route choice; covers a series of intersections
4	vertices arcs sectors nodes links	As for Level 3	dense network models (dynamic); as for Level 3 except dynamic rather than static results; short-range forecasts
5	nodes links centroids corridors	Household data, cars/residence, housing density, demography	strategic network: 4-step assignment process; future demand forecasts up to medium range; network performance statistics; all major arterial roadways and links usually represented - some aggregated corridor modelling
6	centroids corridors	Large-scale O-D matrices	regional land use impact models; short and long range forecasting; only major arterial roads represented or aggregated
7	centroids corridors	Economic data	sketch planning: econometric models

It is important to note that there need not be 7 separate viewpoint layers of a transport network to encompass the 7-level modelling hierarchy. For example Levels 3 and 4 have exactly the same conceptual view of a physical network: the difference between the two levels being that Level 3 is concerned with fixed information and data (eg O-D matrix values) and Level 4 includes dynamic analysis (eg what-if scenarios). Similarly for Levels 6 and 7: both view a network as major nodes with interconnecting corridors that can be either actual physical or conceptual roads.

Basis of the physical transport network

Following is a discussion of some of the terms used in the TNRDB and how they relate to the transport network itself. A complete discussion can be found in Thompson-Clement (1995). Most of the terms are commonly used in transport systems analysis and some are unique to the TNRDB. Where possible the usual meaning of a common word is retained but there are several terms used in different areas of network analysis (eg link) that are ambiguous and whose meaning is (usually) clear from the context of the area of study. Unfortunately most computer-based information schemes such as the TNRDB do not allow the cognitive licence of definition ambiguity and hence a detailed explanation of TNRDB terms is necessary.

The basis of the physical transport network comprises:

a. Centroids, nodes, vertices, and nibs;

b. Corridors, roads, links, arcs, lanes, and sectors.

The structure of an intersection node and examples of traffic flow are shown in Figure 1.

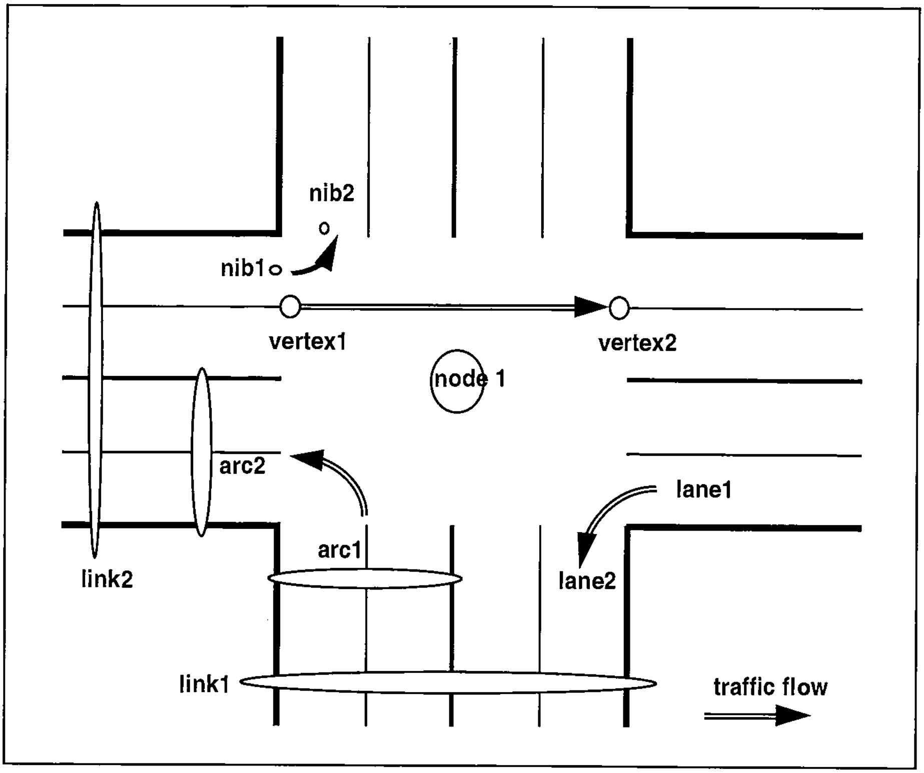


Figure 1 Intersection node components

Vertices are used in the first four levels of the modelling hierarchy ie up to and including the dense network model level. Vertices are connected by arcs. A vertex need not be part of an intersection: it may represent a corner of a road or the reference point for a curve of the roadway or the position of a road traffic management device.

An arc is a unidirectional part of a link and is defined by: road geometry in the horizontal plane; lane geometry; or speed. An arc is at least one lane wide. An arc is used at levels 1 to 4 of the hierarchy. Where there is a change in the speed limit of a section of road there is also an arc boundary. The present TNRDB does not handle the situation where there are different speeds in different lanes of the same arc.

Sectors are used to describe different physical zones within an arc eg for parking, no standing zones, taxi stands or bus stops. A sector is described in terms of the arc of which it is a part.

Nodes are specifically connected by links: the links can be of any class. A node must contain at least one vertex (an example of a single-vertex node is at the boundary point on a road that runs into two localities). Each node is related to a list of the vertices with which it is associated.

Links are usually bidirectional: but a one-way link consists of a single arc.

Nodes and links are usually not specifically referred to in analyses at Levels 1 and 2. A link is an aggregate of arcs and lanes at these levels and nodes an aggregate of vertices and nibs. For instance at the microscopic level where individual units in a traffic stream can be modelled, the broadest view is usually of movement from one vertex along an arc to another vertex. At the third level and higher, traffic movements can be expressed in terms of nodes and links, or vertices and arcs, or a combination of both (though rarely) and hence if data is available at a lower level they can be aggregated for use at higher levels. At the higher end of the modelling hierarchy nodes and links can at best describe Level 5: thereafter they are aggregated into centroids and corridors.

Centroids are used in the higher levels of the 7-level modelling hierarchy and are usually nodes though a centroid can be a place closely related to the transport network such as a sports park, shopping centre, exhibition ground, place of worship etc. Centroids can be an aggregation of nodes though this capability is not yet included in the TNRDB.

Connections among Levels 6 and 7 centroids of the land use impact assessment models and the sketch planning models can be likened to transport corridors where 'the spatial connections between system elements need only occur as notional representations' (Taylor 1991). A corridor does not necessarily occur along a physical stretch of roadway. Bearing in mind that corridors have not been included in the TNRDB, it is envisioned that aggregations of links into a transport corridor could be performed automatically in a DBMS based on the TNRDB, or accomplished interactively within the software application package being used for the analysis. An analytical study would more likely require a combination of both (Brownlee et al. 1988).

Roads consist of one or more links and may correspond to a corridor of Levels 6 and 7. Roads themselves are not a significant part of network analysis except as a named reference for relevant sections of roadway and hence the details of their component pieces (links, arcs, lanes etc) are stored in their own right rather than as attributes of the roads themselves.

The end of a lane is described by a nib. All nibs are part of an associated vertex and the latitude and longitude of the nib are ideally different to those of the vertex except where the vertex terminates a one-lane arc.

Traffic flow counts are described in terms of the relationship between different structures of the TNRDB and hence flows can be classified as nib-nib, vertex-vertex, node-node, lane-lane, arc-arc and link-link movements. A code designates to what type of flow the count refers: this can be from and to, or between structures. This method allows flexibility in handling intersection flows, mid-block flows, flows at the vertex of an arc, flows at the end of a lane, the unidirectional flow on a link and the bidirectional flow on a link. Centroid-centroid data is represented but corridor-corridor data are not included in the TNRDB at present.

TURNING FLOWS STUDY

The initial stage of the Turning Flows Study uses the program written by Professor Michael Taylor at the School of Civil Engineering, University of South Australia. The program utilises Kruithof's method to estimate turning movement flows at an intersection node when entry and exit link flows are known. The program selects *a priori* turning proportions based on leg types (arterial, collector and local). These proportions are based on Hauer's results for Toronto (Hauer et al. 1981), translated to left hand driving. Output is Q(i,j), a matrix of turning movement flows; hence a comparison between results obtained with the Toronto data and results obtained using Australian data from Adelaide, Melbourne and Geelong can be made.

The program can analyse 3- or 4-leg nodes and takes three inputs pertaining to leg *i*: the road type, the entry flow and the exit flow. Leg numbering is anticlockwise and for a 3-leg node (or T-junction) leg 1 is the left hand leg of the stem of the T, making leg 2 the stem of the T and leg 3 the right hand leg. For a 4-leg node leg 1 is the most northern leg, and legs 2, 3 and 4 are then numbered anticlockwise.

The Turning Flows Study extends the work of Hauer by using a larger database of nodes as well as details of the turning restrictions in operation in many of those nodes. For instance it includes considerations of signal phasing (eg provision of separate turning phases) and intersection geometry (eg separate turning lanes) in the selection of initial solutions for turning flow estimation.

Origin of the data

Data relating to signalised intersections of the Adelaide metropolitan area were obtained from two sources: the Adelaide City Council provided data on 62 intersections of the Central Business District and the South Australian Department of Transport provided data on 53 intersections from the remainder of the metropolitan area. Data for Melbourne and Geelong were provided by VicRoads (the Road Authority in Victoria, Australia). The data is being manipulated into TNRDB format to enable further analysis using software packages with, among other capabilities, the ability to perform SQL queries. SQL is the Structured Query Language that has become something of a standard for information retrieval from relational databases.

The data from VicRoads were contained in a turning movement count report file and a right turn control history report. The turning movement count report contains details of 540 surveys performed on 441 intersection nodes in the Melbourne and Geelong metropolitan areas from 1984 to 1989. The details contained in the report include the identifier for each survey; the extent of the survey (either a full 12-hour study or a peak hour turning movement study); the year of the survey; the map reference of the node as found in the Melway Melbourne Street Directory; a unique identifier for each intersection node; the start times of the am peak and pm peak hours; the day and date of the survey; intersection leg numbers; the directional bearing of the leg to the node; and the vehicle counts for the am peak, the pm peak and the 12-hour surveys.

The right turn control history report contains details of the changes made to the signal control operations at some of the intersection nodes of the turning movement count report. As the report name indicates it is the changes to the right turn control that are recorded and each record refers to one of the following types of change: no control to partial control; no control to full control; partial control to full control; and where the detector position has been changed. No control refers to a turning movement that is not under the influence of any right turn arrow and hence filter-turning is possible. Partial control is where a green arrow limits the right turn movement of vehicles, though filter-turning is possible during some part of the phase. Full control refers to a turn that is totally under the control of a turn arrow and hence motorists only see either the green or red arrow during a signal cycle.

The other relevant data contained in the report are: the unique identifier for each node; the direction of the right turn signal control change; the date the conversion was effected and the position of the vehicle detector in the intersection. Included with some records are comments that give additional information such as if a right turn lane is shared or if the turners have to cross a tram reserve.

Study outputs

For a study of this nature to be effective the database management system has to support a wide variety of queries on the roads and intersections of the study network. The queries relating to a particular intersection node would include finding all turning count studies for the node and returning a trace history of the changes made to the traffic light sequencing of the intersection. Road queries would include a list of all the nodes of which the road is a part and the road type(s). For analytical purposes some queries would need to produce aggregations and statistical results such as the mean turning proportion by turn type (eg right turn from primary arterial to collector with partial right turn control) and the distribution of the proportions.

The TNRDB is designed so that the data is held in a format that allows a DBMS to make these kinds of queries and calculations relatively easily. For instance, if the DBMS used is Microsoft Access then SQL aggregate functions such as the estimate of the standard deviation and the mean of a population can be used. If these are not satisfactory, or indeed not fast enough, then similar functions can be written in a different language such as Pascal or C and compiled into a form callable from within the DBMS.

NETNOISE MODEL

The NetNoise model was developed to provide traffic engineers and planners with a means of determining the distribution of noise across a road network area. NetNoise uses the Calculation of Road Traffic Noise (CoRTN) procedure (UK DoT 1988) which has found widespread application in Australia and New Zealand (TransitNZ 1994; Saunders et al. 1983). The CoRTN procedure allows the calculation of noise levels at a specific site using various traffic and site geometry parameters. NetNoise achieves spatial distribution of noise by superimposing a grid of receivers over the study area and calculating the noise level at each receiver. Contouring or Geographic Information System (GIS) software can then be used to derive a contour plot for the noise distribution. A more detailed description of NetNoise can be found in Woolley (1994).

NetNoise has the capability of modelling at levels four to six in the 7-level modelling hierarchy. Strategic planning and modelling can be carried out using links and nodes but where more site detail is required (such as dual carriageways, significant gradients and noise barriers) data input takes the form of arcs and vertices.

Input variables

Noise from road traffic is dependent on the volume of traffic, the speed at which the traffic is travelling and the amount of heavy vehicles in the traffic. Sound levels are also influenced by the road surface texture due to the interaction of tyres with the road surface. The nature of traffic flow is important as noise from free flowing traffic is quite different to that from interrupted flow (stop-go) traffic. When using NetNoise it is assumed that the majority of traffic in the mid-blocks of the links are freely flowing and that there is little congestion in the network. For both types of flow the presence of a positive gradient increases noise levels due to the extra energy required to overcome gravitational forces.

To determine a noise level at a specific site, the noise originating from the source (road traffic) must first be calculated and then adjusted according to the geometry between the receiver and the source. Two aspects of this adjustment are (1) the distance between the receiver and the source and (2) the propagation of the sound (ie how it gets from the source to the receiver). Propagation takes into account screening effects from structures such as sound barriers, earth mounds and buildings. Reflections from some surfaces and the absorption of sound by grass and vegetation may also be considered.

NetNoise has many data features common to other road traffic models and it has features that are unique for noise calculation purposes. The list of input variables required by NetNoise includes:

- node coordinates
- link geometry
- traffic volume
- traffic speed
- proportion of heavy vehicles in the traffic stream
- road surface type (chip seal, asphaltic concrete etc)
- road surface texture depth (optional)
- average link gradient
- presence of any obstructions or barriers
- distance between the source and receiver
- propagation characteristics of the terrain between the source and the receiver

At present, NetNoise reads input files which are in comma-delimited format structured according to the TNRDB protocol. The input requirement for proportion of heavy vehicles in the traffic stream highlights the ease with which a relational database may be used. Three tables have been created each containing data for single heavy vehicle counts, NAASRA classification counts and Australian Bureau of Statistics (ABS) data. NetNoise is able to use all three tables without the need to alter any input file structures. Another example is that of flow counts where flows may represent counts conducted over differing periods of time. At present, hourly, 11- and 12-hourly turning counts, 18-hour noise counts and 24-hour and Annual Average Daily Traffic (AADT) data is accommodated. Data manipulation can be performed by the database manager to generate the table necessary for use by NetNoise.

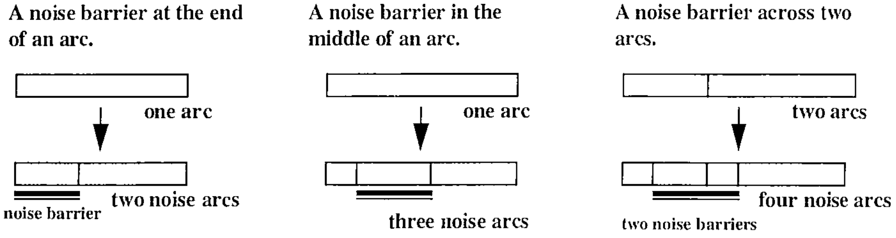


Figure 2 Examples of the use of noise arcs

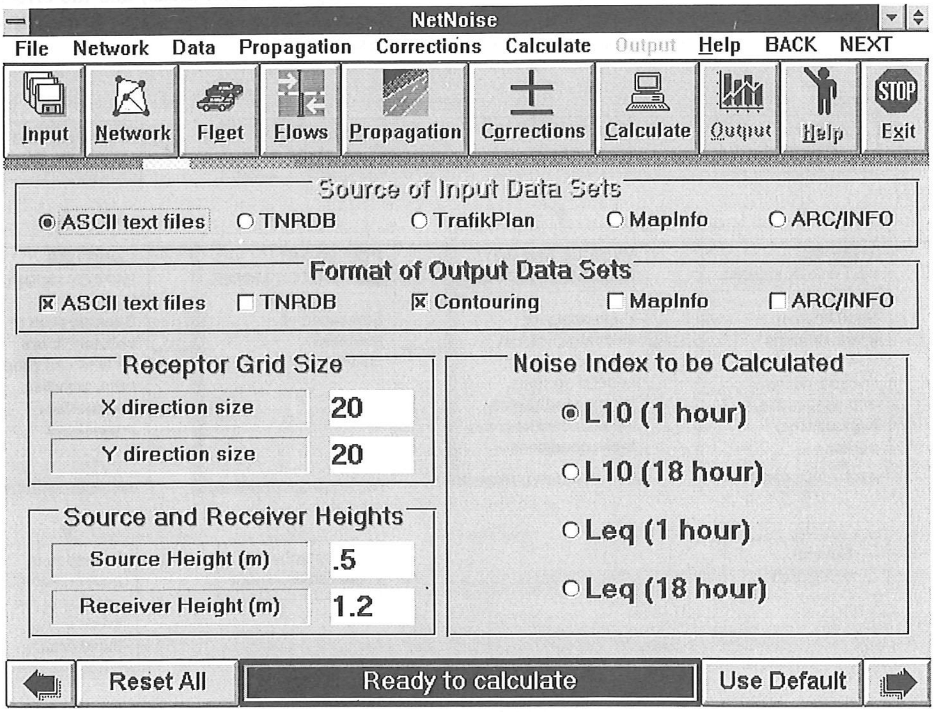
The TNRDB is most useful where there is a deviation from common data requirements. A 'noise arc' has been introduced to enable the storage of information relating to specific noise calculation requirements. CoRTN has a requirement that no arc should have characteristics which cause a difference of over 2 dB(A) along the entire length of the arc. Realistically this means that all significant changes in gradient, the presence of barriers, road surface and traffic conditions which cause a greater than 2 dB(A) change in noise level must be accommodated. At present, barriers in NetNoise must be the length of an arc and parallel to the arc. If a noise barrier is to be inserted into a network, at least one noise arc must be created. Noise arcs are a sub-class of arcs and their applications are best described in Figure 2. The concept can be applied similarly for road surface and gradient changes within an arc.

NetNoise output

Output from NetNoise takes many forms reflecting the importance of data availability to the program. Comma-delimited files are a useful form of output as they can be imported into spreadsheets for manipulation and some analysis. Files are also created for direct use by the MapInfo desktop mapper and ARC/INFO GIS. With the advent of greater computing capabilities, Object Linking and Embedding (OLE) will be used to place output directly into host applications such as Excel. This will avoid the need for converting and importing data but requires computer hardware comprising at least 8Mb of RAM to work and 16Mb to work efficiently. As the information from NetNoise principally represents the spatial distribution of noise, final noise levels at each receiver point are not output into the TNRDB format as there is no set relationship between the receiver points and the links (or arcs). Instead, noise levels emanating from the source (road traffic) are stored with the link or arc number as the primary key. This data is useful to determine noise levels along the links or arcs and is commonly applied for planning purposes but does not take into account propagation of sound from the source.

NetNoise development environment

The NetNoise application has an interactive Windows interface created using the Visual Basic Programming System (see Figure 3). Emphasis has been placed on guiding the user through network noise calculations and the setting up of input and output formats of which the TNRDB plays an integral part. Visual Basic v3.0 contains a copy of the Microsoft Access 1.1 engine known as 'Jet' and this combination allows access to many common database formats such as dBASE, FoxPro, Paradox and Btrieve.



Source of Input Data Sets
● ASCII text files ○ TNRDB ○ TrafikPlan ○ MapInfo ○ ARC/INFO

Format of Output Data Sets
☒ ASCII text files ☐ TNRDB ☒ Contouring ☐ MapInfo ☐ ARC/INFO

Receptor Grid Size
X direction size 20
Y direction size 20

Source and Receiver Heights
Source Height (m) .5
Receiver Height (m) 1.2

Noise Index to be Calculated
● L10 (1 hour)
○ L10 (18 hour)
○ Leq (1 hour)
○ Leq (18 hour)

Figure 3 The NetNoise interface developed using Visual Basic v3.0

Another useful feature incorporated into Visual Basic is the Crystal reports add-on which simplifies the task of report writing and formatting. At the TSC, a prototype TNRDB has been created using the Access database manager for use with the NetNoise model. The prototype contains tables which store common traffic data such as flows and speeds and specialised tables for use with NetNoise such as noise barriers and road surfaces. Future work on a study area in Adelaide will enable enough data to be collected to utilise some aspects of the TNRDB beginning with the linking of the traffic network model TrafikPlan and NetNoise. This work will culminate in the application of the IMPAECT supermodel described in the next section.

IMPAECT SUPERMODEL

Recent research at the TSC has been directed towards the development of a computer-based modelling system for the formal assessment of the environmental impacts of road traffic. The modelling system is known as IMPAECT, the Impact Model for the Prediction and Assessment of the Environmental Consequences of Traffic. The principal reason behind the development of IMPAECT is to include environmental planning decisions in the transport planning framework in such a way that potential environmental problems resulting from transport infrastructure development or modification can be identified in the planning stages, and solutions to those problems devised in conjunction with the development of the transport plans, rather than as remedial treatments or curtailment of the project at a later stage. An inherent part of the rationale behind IMPAECT is the idea that the manifestation of environmental impacts as they affect sensitive land uses is the desired model output for the planner and decision maker. Amounts of pollution emitted from the transport system or its individual components is not enough, the levels of immission of those pollutants by sensitive land uses (eg residential areas) and the effects of those pollutants on those land uses must be examined: see Wigan (1976), Brown and Patterson (1990) and Taylor (1994) for a further discussion of these issues.

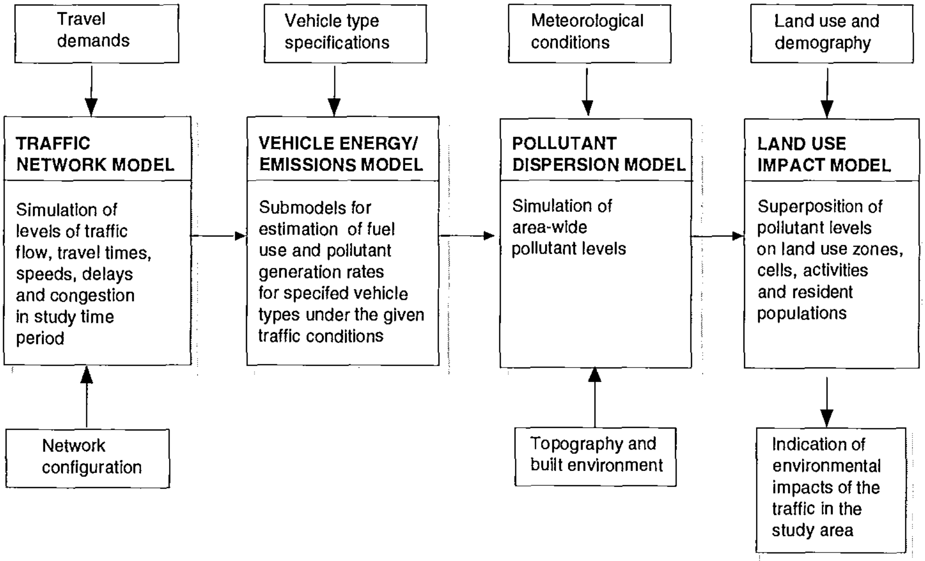


Figure 4 The IMPAECT supermodel system for assessing environmental impacts

IMPAECT is described as a supermodel because it consists of a connected set of individual models. These component models provide information on traffic conditions, travel movements, energy consumption, pollutant emissions and the distribution of pollutant loads across a study area. Each model has its own domain. It takes information from established databases or from other models, and its outputs are then used by other models or are taken into the wider analysis process.

The basic scheme of the system is given in Figure 4. A traffic network model is used to produce (by simulation or forecasting) the levels of traffic flow and travel conditions on a study area network, under the given transport plan or traffic management scheme. Models of vehicle fuel consumption and emissions under the modelled traffic conditions are then used to estimate the traffic system fuel usage and the levels and spatial distribution of pollution generation. This information, coupled with data on the meteorological conditions, may then be used as input to a pollution dispersion model, which estimates the spread of the pollution over the study area, so providing the modelled levels and spatial distribution of the pollution. The land use impact model superimposes the pollution levels on the land uses and populations in the study area to determine the likely sites and extent of environmental problems resulting from the traffic system.

The application of the modelling system of Figure 4 to predictions of environmental impact and energy use depends on the accuracy of the traffic model in reproducing travel conditions on the network, and the validity of the vehicle performance models. The application to environmental impact analysis is based on the premise that although the actual absolute levels of pollution may be affected by many other factors besides those included in the component models, the modelling system can reasonably detect relative differences in levels of pollution between alternative sets of traffic load distributions (eg under alternative transport systems management plans or alternative travel demand management schemes). Further, the modelling approach means that a number of pollutants can be included together under the same sets of conditions; eg noise, gaseous emissions, and fine particulates. Thus alternative schemes may be ranked on a number of environmental quality objectives, and comparisons made between them. A full description of the IMPAECT system is available in Woolley and Young (1994).

IMPAECT can be applied at levels 3 to 6 of the 7-level modelling hierarchy. Choice of a particular level depends on the extent of the geographic area to be considered and the kinds of transport problems to be investigated. Each level is distinguished by the particular network model to be applied and the matching set of fuel consumption and pollution emissions models as described in Taylor (1994) and including the NetNoise model. For example, studies at Levels 3 and 4 would apply dense network traffic models such as TrafikPlan or SATURN, whereas larger-scale studies (say Level 5) would employ strategic network models such as EMME-2 or QRS-II. Fuel and emissions models are selected from the family of models described by Taylor (1994) based on the initial fuel consumption modelling work by Bowyer et al. (1985).

TNRDB has a central role in IMPAECT, as it provides a database resource from which the various network models including NetNoise can extract the appropriate network descriptors and travel demand data that they require.

CONCLUSION

The Transport Network Relational Database is being designed for use by transport systems models for a variety of purposes in transportation engineering. Transport network analysis has significant demands for network data especially when a range of applications at different levels of detail are required. Different applications can use different sets of TNRDB tables and since the TNRDB is designed using the tenets of the relational database paradigm it has the capabilities to be managed by several database management systems utilising open database connectivity. Further, a database of the TNRDB design can be accessed by applications running in client/server configuration and hence its flexibility makes it a useful basis for transportation application data storage for use across a contemporary computer communications network. The TNRDB structure is being used in a study of turning movements and flows; in the network traffic noise prediction model, NetNoise; and is projected to be an integral part of the IMPAECT supermodel being developed at the Transport Systems Centre.

REFERENCES

Akçelik, R., Shepherd, R., Besley, M. and Lake, G. (1994) *Australian Transport Software Integration System (ATSIS) development report*, Report WD TE 94/009, Australian Road Research Board Ltd, Melbourne.

Bowyer, D.P., Akçelik, R. and Biggs, D.C. (1985) *Guide to fuel consumption analyses for urban traffic management*, Special Report 32, Australian Road Research Board, Melbourne.

Brown, A.L. and Patterson, N.S. (1990). Noise assessment when it matters: environmental evaluation integrated with road network planning, *Proceedings Australian Road Research Board Conference* 15 (7), 61-78.

Brownlee, A., Millican, R., Stephenson, B., Schnablegger, J. and Teply, S. (1988) Large scale integration of computer programs, *Proceedings of the Third International Conference on Computing in Civil Engineering,* 10-12 August 1988, Vancouver, 659-674.

Date, C. (1983) *An Introduction to Database Systems, Volume 2*, Second Edition, Addison-Wesley, Reading.

Date, C. (1986) *An Introduction to Database Systems, Volume 1*, Fourth Edition, Addison-Wesley, Reading.

Elmasri, R. and Navathe, S.B. (1989) *Fundamentals of Database Systems*, Benjamin/Cummings Publishing Co., California.

Hauer, E., E. Pagitsas and Shin, B.T. (1981) Estimation of turning flows from automatic counts, *Transportation Research Record 795*, Washington.

Horowitz, A.J. and Pithavadian, A.R. (1987) Toward a data structure for computer-aided design of transportation networks, *Transportation Research B* 21B (4), 317-321.

Philipson, G. (1993) In search of client/server, *Informatics*, August 1993, Australian Computer Society.

Saunders, R.E., Samuels, S.E., Leach, R. and Hall, A. (1983) *An evaluation of the UK DoE traffic noise prediction method.* Report ARR 122, Australian Road Research Board, Melbourne.

Sayles, J.S., Karlen, S., Molchan, P.and Bilodeau, G. (1994) *GUI-based Design and Development for Client/Server Applications*, John Wiley and Sons, Publishers.

Taylor, M.A.P. and Anderson, M. (1983) A hierarchical approach to modelling network flows, *Papers of the Australian Transport Research Forum* 8 (2), 136-149.

Taylor, M.A.P. (1991) Traffic planning by a 'desktop expert', *Computers, Environment and Urban Systems* 15 (3), 165-177.

Taylor, M.A.P. (1994) Incorporating environmental planning decisions in transport planning: a modelling framework, in Y. Hayashi and J.R. Roy (eds) *Transport, Land Use and the Environment*, Kluwer, New York.

Thompson-Clement, S.J. (1995) *The Transport Network Relational Database*, Working Paper 95/1, Transport Systems Centre, University of South Australia.

TransitNZ (1994) *Traffic Noise from Uninterrupted Traffic Flows*, Wellington, New Zealand.

UK DoT (Welsh Office) (1988) *Calculation of Road Traffic Noise (CoRTN)* HMSO, London.

Ullman, J. (1982).*Principles of Database Systems*, Second Edition. Computer Science Press.

Wigan, M.R. (1976) The estimation of environmental impacts for transport policy assessment, In: *Environment and Planning A* 8, 125-147.

Woolley, J.E. (1994) A road traffic network noise evaluation model, *Proceedings of AAS Canberra Australian Acoustical Society Annual Conference*, Canberra, 158-165.

Woolley, J.E. and Young, T.M. (1994) Environment: the third dimension of the land use-transport interaction, *Papers of the Australasian Transport Research Forum* 19 (1), 223-239.