

# Modeling a Public Transport Network for Generation of Schematic Maps and Location Queries

Silvania Avelar and Raphael Huber  
Institute of Cartography, ETH Zurich  
CH-8093 Zurich, Switzerland  
Tel: +41 1 633 3031 Fax: +41 1 633 1153  
[avelar@karto.baug.ethz.ch](mailto:avelar@karto.baug.ethz.ch), [raphael.huber@gmx.ch](mailto:raphael.huber@gmx.ch)

## Abstract

Schematic maps, such as subway or transit maps, are produced by hand or purely graphics software at present. This is not only a timely process, but requires a skilled map designer. Automatic generation of schematic maps may improve the process but more importantly would extend the use of such maps to a larger audience. In order to produce such maps however, we need a database containing geographical information on network routes of the region under consideration. This database should also contain topological information in order to answer common location queries from the map user. One fundamental aspect in a GIS project is the data model, which describes how the geographic reality will be represented in the computer. This paper presents a data model to describe geographical and topological information of a public transport network. We have also built a user interface on top of the data model for supporting location queries.

**Keywords:** data model, transportation network, location queries

## 1. Introduction

Many cities provide a comprehensive public transport system, often integrating bus routes, suburban commuter train services and underground railways. Information about such transport systems are in general released in schematic maps (Monmonier, 1996). These maps indicate important topological information on transportation such as connectivity and stops. We need a database with transportation information of the city or region under consideration to generate schematic maps automatically (Avelar and Müller, 2000; Cabello et al., 2001).

Our database has two major purposes: to be used as a basis in the automatic generation of schematic maps, and as a general information system to users of public transportation networks. Regarding our first purpose, we need geographical information about transport routes. In order to answer user location queries, we also need to store topological characteristics of the transportation network, such as intersections of routes and rivers, adjacency of stations, and proximity of stations to special places.

In this paper we provide a conceptual data model designed to represent geographical and topological characteristics of a public transportation network. Data modeling aims at designing efficient and consistent databases. The result of this process is a conceptual scheme which reflects the requirements that the desired database must achieve (Halpin, 1995). Current data models used to represent streets and routes (e.g. ArcInfo georelational data model) integrate the cartography, network link, and attributes of the link into a single linear spatial object (Dueker and Butler, 2000). This work calls for an uncoupled approach of graphics, topology, and characteristics to facilitate comprehension and maintenance of the database.

Nielsen et al. (1997) described their experience in using ArcInfo to handle public transportation networks. They state that most GIS-packages, including ArcInfo, are currently not able to handle fundamental elements of public transport networks. In fact, our data model was implemented using ArcInfo and ArcView to build an experimental database with public transportation network information of a selected perimeter of Zurich (Purtschert, 2001). Some data types could not be directly defined in ArcInfo. For example, in ArcInfo there is a topological connection between nodes and links (arcs) given by the Arc-concept. The Arc Attribute table contains information on the 'from node

number' and the 'to node number'. However, the opposite information on which links are connected to a node is not contained in the Node Attribute table. The translation from transport model topology to GIS-topology has to be carefully treated, because transportation network models can demand a complex topology not very well covered by most traditional GIS-packages.

Different data models have also been proposed in geographic information systems for transportation (GIS-T). For example, Vonderohe et al. (1997) describe a data model prepared by transportation professionals and agencies in an effort to provide data sharing of linear systems for GIS-T. The central notion is that a common linear datum relates links and nodes of linear systems to the real world. Duecker and Butler (2000) discuss transportation data sharing and the need for a data model that holds transportation features as objects of interest, and not their graphical representations. They provide a data model in which cartography is directly connected to the transportation feature and not to the linear datum as in Vonderohe et al. (1997). Sharing transportation data and linear data models are an important issue, and a difficult one, because of the varied nature of applications that require data in specific forms. Our data model has an object approach as well. The transportation features were selected to better represent the requirements of our application.

We have also created a prototype tool which enables the user to interact with location information on transport routes. Our system provides visualization of the vector-based, cartographic information of a transportation database. It also answers user location queries related to the transport network by highlighting the map.

The following section presents user location queries for schematic maps and design considerations taken into account when planning our data model. Section 3 describes how the reality has been formalized in the data model. In section 4, we present our prototype tool to visualize and explore the transport network. Finally, section 5 gives our concluding remarks and outlines further work.

## **2. Requirements for the data model**

A schematic map aim is to provide topological information on transportation to answer user queries about directions (Monmonier, 1996). Here we consider some of such user queries for schematic maps to design our data model.

### **2.1 Cartographic queries**

We selected the following queries to be successfully addressed by our schematic maps:

Where is my destination on the map? In which direction do I need to go?

What is the name of the station at end of the line?

What bus/tram/etc. takes me from A to B? Do I need to change lines? If so, where and what line?

How many stops do I ride before I get off? How long will it take?

How do I get to the other side of a river or lake?

What is the nearest station to my present location? What is the nearest station to a given place?

The data model must reflect all necessary information, but not more, to generate automatic schematic maps from a database and also to answer the above location queries completely and efficiently. To answer location queries, the data model should support operations associated with topology (e.g., connectivity, crossing, proximity) and with network analysis (e.g., path finding and location). Next we give more details about what information is required to be stored in our database.

### **2.2 Design considerations of the data model**

Besides the geographical information about the spatial location of the transportation network elements for the generation of schematic maps, we also need to store lakes and rivers and their geographic shapes in the database. The algorithm to generate schematic maps can detect topological changes on

the intersection of roads (Avelar and Müller, 2000). This topological information is also not needed to answer previous queries. However topological information related to the intersection between transport routes and bodies of water is required to simplify the answer to queries like "How can I get to the other side of a river?".

We need to identify when changes between routes are possible for queries like "How do I get from A to B?". Transport routes can connect to each other at various stations along their routes. Most of the routes have only one way in both directions, but there exist stations which are served in one direction only. A public transport network often contains different networks, such as trains, trams and buses. A certain journey might use different transport modes. Additionally, transit service is time dependent, i.e., the best path between an origin and destination can change depending on the timing services available. At the same node, different routes may have different wait times. The model should provide means to describe such characteristics of a public transport network.

To answer proximity queries, 'nearTo'-relations among map features have to be explicitly identified and stored. For example, for the query "What is the nearest station to a given place?", we have to store special locations, such as zoos, parks, stadiums, theaters, etc., with their geographic location and the stations they are conveniently located next to.

### 3. The transport network data model

Our object model diagram is shown in Figure 1. The graphical notation used is an information structure based on Entity-Relationship (ER) Diagrams (Chen, 1977). This approach to conceptual design provides a simple way of declaring the objects, attributes, and relationships among them. The diagram symbols are summarized in the following paragraph for convenience.

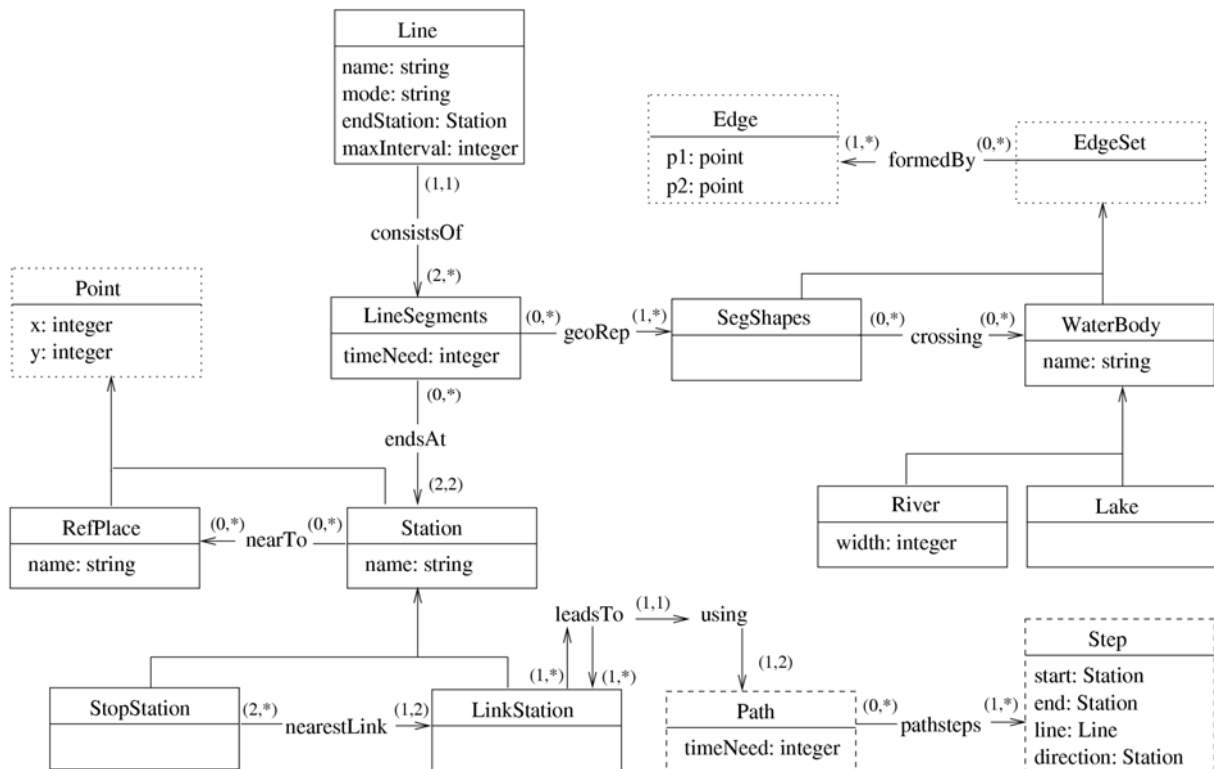


Figure 1: Transport network data model.

Entities may be seen as real-world objects which are of interest for the application under consideration. They are shown as boxes, which also include their main attributes. Relationships are used to model associations and are represented by lines. Entities are related to each other through verb-oriented

statements. A relationship without explicitly stated verbs generally can be read as an ownership relationship. The minimum and maximum number of occurrences of each entity that may participate in a relationship are characterized using (min,max)-notation. If an entity may or may not participate in the relation, then min is 0. The cardinality for 'many' is indicated by "\*". The arrows underline the semantics of the description.

### 3.1 Overview of transportation features and relationships

The objects used for modeling the public transport network can be categorized into three basic groups identified in Figure 1:

- *Transport features: elements of the transport network and other geographic objects required to answer user location queries.* A *Line* is a route composed of *LineSegments* (oriented, between two neighbor *Stations*), each of which may be formed from one or more *SegmentShapes* (for geometric shape of a line segment, non-oriented). The same *SegmentShape* can be shared by different *LineSegments*. *Stations* can serve one line only - *stop stations* - or be connection nodes, where travelers can change lines - *link stations*. We still introduced the concepts of *WaterBody*, e.g. rivers, lakes, and of *ReferencePlaces*, which are special places of the city located in the geographic context of the transportation network. *Relationships*: The topological relations we need to store contain information about the intersection of *SegmentShapes* and *WaterBody* ('crossing') and of proximity of *ReferencePlaces* to *Stations* ('nearTo').

- *Geographic locations: real world, earth-based location of transport features.* Occurrences of transport features can be linear between two points when we use the object *Edge*, or in a single geographic location, that is given by *Point*. To represent a sequence of linear segments or area elements, such as lakes, we use the collection *EdgeSet*, which are open or closed linearly connected sets of *Edges*.

- *Path objects: description of a path through the transportation network consisting of one or more segments.* A *Path* through the network is composed of *Steps*. A *Step* is a section you can go from one link station to another without changing the vehicle. *Relationships*: 'nearestLink', which assigns one or two link stations to every stop station; 'leadsTo', is a matrix with paths from every link station to every other link station as entries; 'using', which assigns a path to every matrix cell of 'leadsTo'. The detailed solution to model paths is described in the next subsection.

To populate the database, a number of constraints should not be violated. Such constraints should be considered when entering data and in the checking mechanisms of the data model. Some examples are:

C1. A *Line* has at least two *Stations*.

C2. A *Station* has at least one *Line*.

C3. All *LineSegments* of a *Line* must be of the same transport mode.

C4. A stop station has only one *Line*, and a link station has at least two *Lines*.

C5. There are at most two *LineSegments* in a stop station (coming from or going to) and at least two *LineSegments* at a link station (two in case of a terminal of two *Lines*).

C6. The first and last *Point* of the *SegmentShape.formedBy* of a *LineSegment* must be identical to the *Points* of *Stations* in *endsAt*.

C7. *Line.consistsOf* must form a cycle (to cover both directions of a *Line*).

C8. *Edges* of *EdgeSet* must be linearly connected.

### 3.2 Pre-calculated paths

We use pre-calculated paths to answer the "How do I get from A to B?" type of queries. Because the data concerning line routing in a public transport network is in general more or less static over several months or even years, it can be reasonable to perform the operation of calculating paths from one station to another one time in advance and store the results in a database. Queries can profit by accessing paths directly.

A path consists of steps, which are parts of the route between link stations without changing vehicle. A Path itself has an attribute called *timeNeed*. This is the sum of timeNeed for each Step and the times needed to change between the Steps at link stations (attribute *maxInterval* of object Line).

The straightforward approach would lead to a square  $m \times n$  adjacency matrix where  $m = n =$  the number of stations in whose cells the paths are stored. Because this matrix would become very large (e.g. for a city with 300 stations, 90000 cells), we searched for a simple way to reduce storage requirements. The idea is to store only paths from link station to link station and complete the actual paths during runtime of queries. Link stations are usually only about 20% or 40% of all stations, so the number of cells would result in about 15% of the original amount. To complete paths to a stop station, we use the stored data in the 'nearestLink' relationship. This relation is actually redundant, for performance reasons, and could also be calculated on the fly. Figure 2 illustrates the proposed solution. The algorithm is as follows.

Procedure GetPathfromAtoB (A, B, matrix);

Input: Stations A and B, matrix with Paths of linkstation x linkstation

Output: shortest pre-calculated Path from A to B

begin

if A is a stop station then

get nearest link stations to A in  $A_i$  else  $A_1 = A$ ;

if B is a stop station then

get nearest link stations to B in  $B_j$  else  $B_1 = B$ ;

get pre-calculated path<sub>k</sub> for combinations  $A_i \times B_j$ ; (\* maximum of 4 paths \*)

if A or B is a stop station then

extend path<sub>k</sub> by steps from A or B to respective nearest link stations;

compare all path<sub>k</sub> and return shortest;

end;

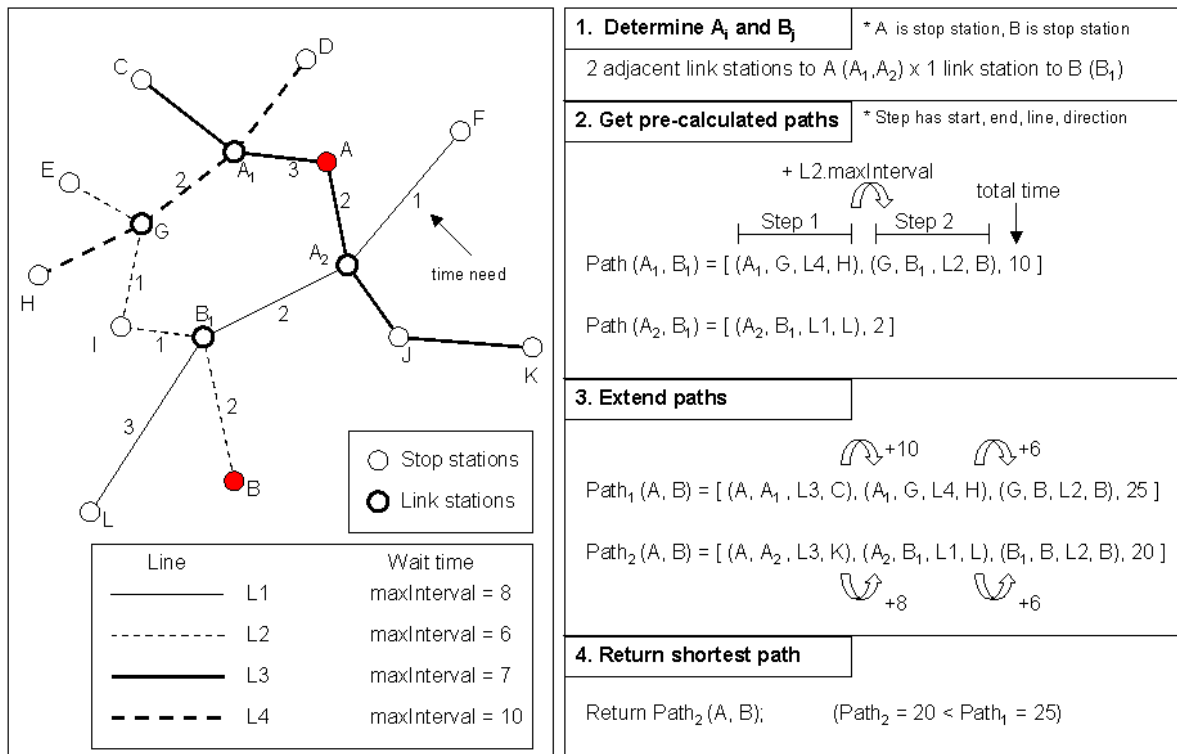


Figure 2: Illustration of use of pre-calculated paths.

#### 4. A visualization tool for cartographic queries: an example

The TNview (for Transportation Network viewing) tool enables a user to view the transport network and to explore and analyze geographical and topological information through cartographic queries. The experimental system was implemented using Java. We can start TNview either as an application or an applet embedded into a HTML page through Internet (Huber, 2001). The tool has the following structure:

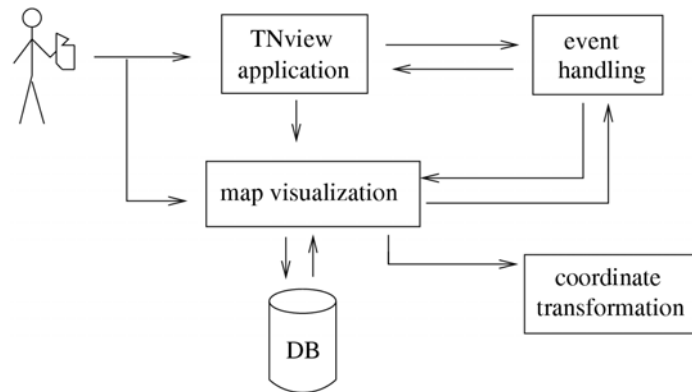


Figure 3: A framework for visualizing cartographic queries.

We created a data set to produce an experimental design and visualization. The database was built and accessed by using OMS Java (Kobler and Norrie, 2000). The transportation network of our scenario and the TNview tool is shown in Figure 4. The user interface provides location queries to obtain lines, stations, reference places and lakes, by name or geographical location. The text area displays attributes of the selected elements.

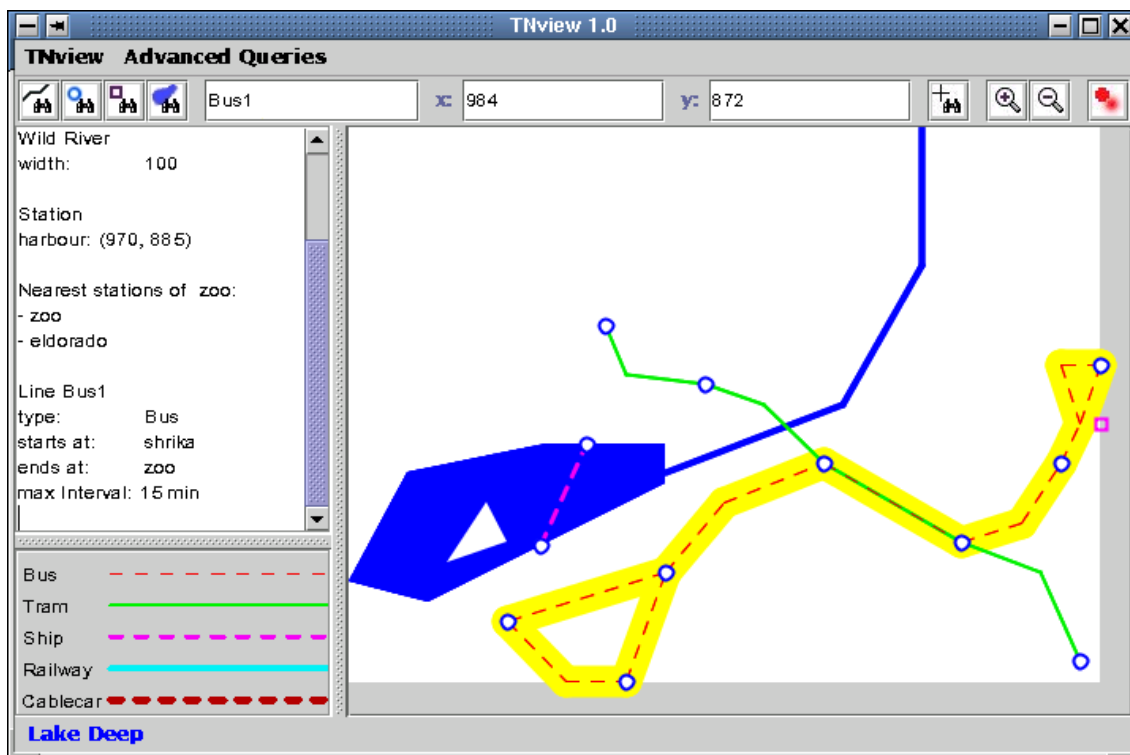


Figure 4: Visualization tool and scenario example.

Advanced queries, such as nearest stations to a reference place, are selected from the menu bar. Users can also select transport features directly from the map. By moving the cursor to a map element, its name is shown in the status bar. By double clicking on a map element, or according to queries, the selected element is highlighted on the map and its attributes are displayed in the text area (see line Bus1 in Figure 4). The text can be extracted by a copy/paste mechanism for further purposes. Each time an element needs to be displayed, its geographic coordinates are transformed into screen coordinates, and vice versa in case of element-at-position queries. A button gives access to toggle an anti-aliasing function for graphical treatment of lines. Zoom is also provided. This makes easy to browse around the map to find information about the surrounding area.

The map legend for mode of transportation is automatically generated. In case a route segment is shared by many modes of transportation (even all of them), TNview ensures that all modes have a different line specification, such that they are always visible on the map.

## **5. Conclusions**

Two major contributions are made in this paper: (1) we designed a data model for geographical and topological information of a public transport network and implemented it; and (2) we developed a prototype tool to support queries on the data model.

Data modelers can use this paper, and their previous experience, to build upon the described object data model in order to adapt it for other particular scenarios in which streams of objects at nodes in a network play a role. Roadways, railroads, transit systems, shipping lanes, and air routes have all linear features that can utilize the same basic network data model (Vonderohe et al., 1997).

The TNview tool provides data visualization and query capabilities. The aim in developing this tool was to provide fundamental options and building blocks to yield a map query based system which can be extended. The graphical tool can serve as a basis for a more sophisticated tool (e.g. supporting multiple views, editing of elements directly on the map display, and incremented with other queries). The idea is to have an electronic schematic map, which can answer user location queries.

Future work includes performance evaluation of the pre-calculated paths and the extension of the data model for including other features, e.g. street network and walkways. Although schematic maps offer little or no references to surrounding streets, this interaction can be useful when users want to find stations near to specific streets on the map.

## **Acknowledgments**

We are grateful to Lorenz Hurni and Moira Norrie for the support to the realization of this work. Thanks also to Roger Karrer and Ari Kahn for proof reading this paper.

## **References**

- Avelar, S. and Müller, M. (2000) Generating topologically correct schematic maps. In Proc. of 9th International Symposium on Spatial Data Handling, Beijing, pp. 4a.28-4a.35.
- Cabello, S., de Berg, M., van Dijk, S., van Kreveld, M. and Strijk, T. (2001) Schematization of road networks. In Proc. of 17<sup>th</sup> ACM Symposium on Computational Geometry, Massachusetts (to appear).
- Chen, P. (1977) The Entity-Relationship Approach to Logical Database Design, Q.E.D. Information Sciences, Wellesley, Massachusetts.

- Dueker, K. and Butler, J.A. (2000) A geographic information system framework for transportation data sharing. In Thill, Jean-Claude (ed.), "Geographic Information System in Transportation Research", Pergamon, pp. 13-36.
- Halpin, T. (1995) "Conceptual Schema & Relational Database Design", Prentice Hall Australia.
- Huber, R. (2001) Semester Project: A slim and open data model for public transport network using OMS. Institute for Information Systems, ETH Zurich, Switzerland.
- Kobler A. and Norrie M. (2000) OMS Java: An Open, Extensible Architecture for Advanced Application Systems such as GIS. In Proc. of International Workshop on Emerging Technologies for Geo-Based Applications, Ascona, Switzerland.
- Monmonier, M. (1996) "How to lie with maps". The University of Chicago Press.
- Nielsen, O.A., Israelsen, T. and Nielsen, E.R. (1997) GIS-based method for establishing the data foundation for traffic models, ESRI  
<http://www.esri.com/library/userconf/proc97/proc97/to600/pap573/p573.htm>
- Purtschert, M. (2001) Diploma work: Aufbau einer GIS-Datenbank zur Generierung von schematischen Karten des öffentlichen Verkehrs. Institute of Cartography, ETH Zurich, Switzerland.
- Vonderohe A., Chou C., Sun F., and Adams,T. (1997) A generic data model for linear referencing systems, Research Results Digest, number 218, Transportation Research Board, Washington.