

Transport object platform (TOP)

A. H. Kaas & R. D. Frederiksen

Atkins Transportation Scandinavia, Denmark.

Abstract

The operation of public transport systems is inherently a rather complex matter. Not only does a public transport system rely on a given infrastructure; it is also dependent on the available rolling stock and the possible timetable.

Despite the interdependence between different data, public transport companies often structure their data in a non-holistic way; e.g. by making separate departments responsible for infrastructure, timetable and rolling stock data, respectively. This tendency is strengthened by the deregulation of the public transport sector in many countries making different companies responsible for data of the same transport system.

For this reason, data are often placed on different software platforms; Timetable and rolling stock data in different relational databases, infrastructure data are often divided in tabular data stored in a relational database, and geographical data stored in either CAD or GIS. Some data are even stored in closed proprietary formats inside transport modelling software packages.

With the introduction of object-oriented GIS based on standard relational databases, an elegant solution to these problems is now possible. The answer is to create an intelligent, rule-based, open and extensible object-oriented data model.

Making a data model intelligent and rule-based involves building functions (methods) into the data model itself, rather than into the client of the data model, e.g. into a transport model package. Based on defined rules, these functions can ensure data integrity at all times. More advanced functions can be programmed to modify the underlying data so that otherwise illegal edits can be made without compromising data integrity.

1 Introduction

This paper describes a new rule-based, object-oriented model for describing public transport systems. The implementation of this approach – the Transport Object Platform (TOP) – utilises the new object oriented approach now available in GIS to handle topologic complexities beyond the possibilities of earlier, non-object oriented GIS.

2 The Conceptual Model of TOP

In the following, the conceptual model behind TOP is described. The object class names are written in *Italics* concatenated with capital letters starting the individual words, e.g. *TransportEdge*.

Overall, the TOP data model, can be described as consisting of 4 main parts:

- The Physical Network consisting of edges, junctions and turns. Turns are mainly used in conjunction with the road network. But they can also describe restrictions in e.g. rail switches.
- The Route Network describes scheduled routes on top of the underlying physical edges. A *Route* connects a series of *Stops*. A *StopPattern* shows which of the *Stops* along the route that are actually stopped at. While the *TimePattern* describes the travel time from *Stop* to *Stop*. The *run* describes one specific departure. *Routes* can be grouped in order to describe a single public service with variations in the *Route*, *StopPattern* and *Timetables*. In the implementation, *TimePattern* and *StopPattern* was combined into a single table, *TimePattern*.
- Transit Terminals describe junctions in the public route network, and the possibilities of movement (*Transfers*) between stops within the terminal. *StopGroups* are aggregations (unions) of *Stops*, and *Terminals* unions of *Stops* and *StopGroups*.
- The Demand group of objects describe data elements commonly used in transport modelling. *CatchmentAreas* (e.g. zones) are used to divide a model area into a collection of aggregated elements. A *Terminator* is the network representation of the *CatchmentArea* in the form of a node. This is connected to the relevant *TransportNodes* and *Stops* using *Connectors*. *Matrices* are used to store relevant information described on a *Catchment-to-Catchment* level, for instance number of travellers, travel time etc.

As part of the process developing the conceptual model of TOP, a review was made of the most widely used model applications on the market. This led to the addition of some specialised objects to describe Terminals and Demand, as explained more thoroughly in the following sections.

2.1 Physical Networks

As mentioned above, the physical network consists of *TransportEdge*, *TransportJunction* and *Turn* objects (figure 1).

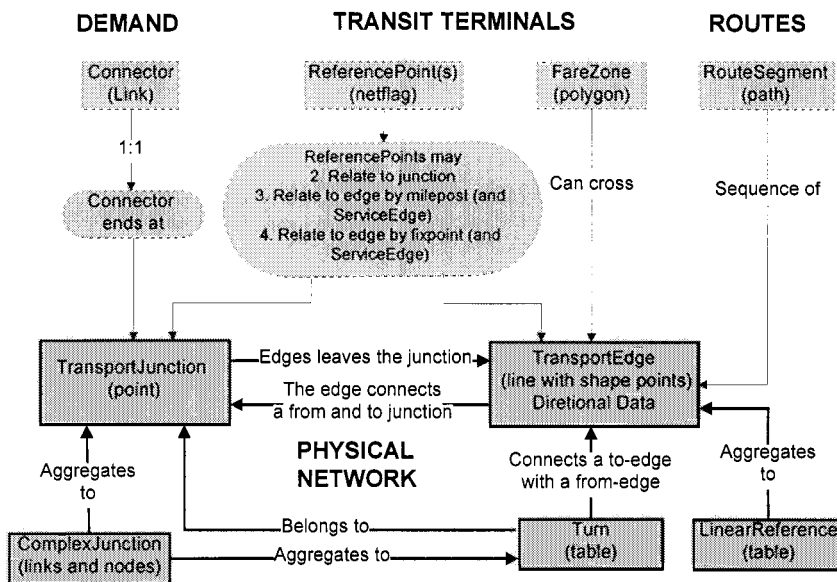


Figure 1: The conceptual model for physical networks

The *TransportEdge* and *TransportJunction* describe physical infrastructure objects, while the *Turn* objects are used to describe movements allowed at the *TransportJunctions*.

Data concerning road signs, pavement management etc., can be described by linear referencing (similar to dynamic segmentation) on top of the *TransportEdges*.

ComplexJunctions describe junctions that are too complex to describe by a simple turntable. This data type is sometimes referred to as *sub networks*, e.g. in the GTF format (Nielsen et al, 2001 [1]).

2.2 Terminals

Terminals are the most complex group of objects in TOP (figure 2). This is partly because of the versatility needed to describe terminals for a wide range of public transport modes, and partly caused by the wide range of conceptual models used by various public transport operators, companies, departments of transport and planning software.

The *Stop* is the basic building block used to describe a *Terminal*. A *Stop* has always a geographic location described by co-ordinates given by the

ServicePoint. In some cases this location might be coincident with a *TransportJunction* (as the case in some older transport modelling packages). This is the most simplified geographic definition of the stop. In other cases, the stop will have co-ordinates different from junctions in the physical network; i.e. rail platforms versus rail tracks.

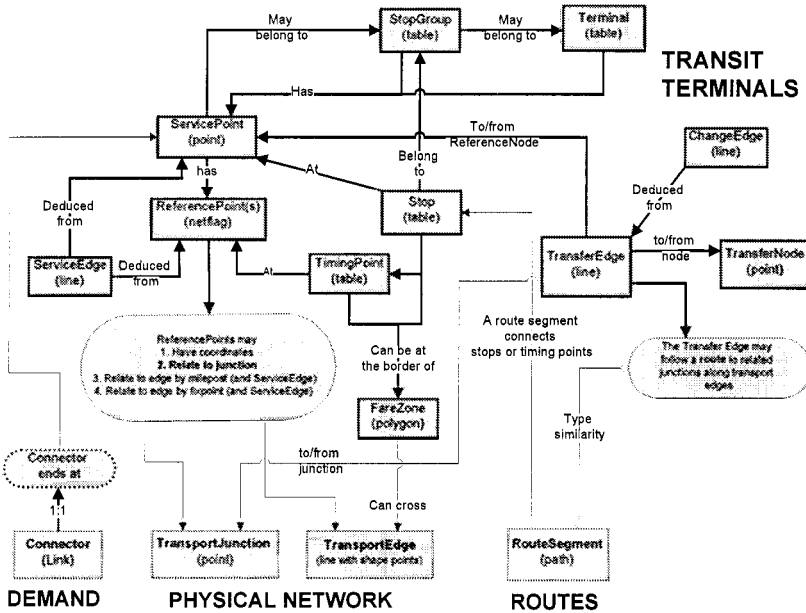


Figure 2: The conceptual model for terminals

The connection of a *Stop* to the relevant nearby *TransportEdges* is shown by placing a *ReferencePoint* on the edge (figure 3). This represents the ability of passing public routes to use the stop. A *ServiceEdge* connects the *ReferencePoint* to the *Stop* (*ServicePoint*).

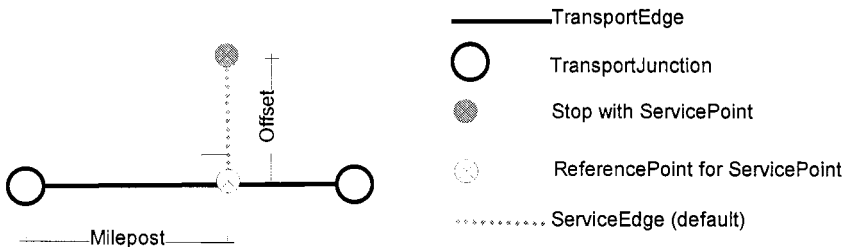


Figure 3: Different approaches to define a stop

The *ServicePoint* of the *Stop* can be defined simply by its co-ordinates. However, it can also be defined by a milepost along the *TransportEdge* (linear referencing) and an offset (figure 3).

A certain *Stop* can be connected to several physical networks by multiple *ReferencePoints*, e.g. a bus stop connected to the road where the bus runs, as well as the pedestrian network leading to the stop.

A *Stop* can also be connected to other *Stops* nearby, through *TransferEdges*. One example is when *Stops* are grouped together in a *StopGroup*, where the *TransferEdges* describe the connection between the *StopGroups'* *ServicePoints* and *Stops*. Note that a *ServicePoint* for a *StopGroup* is described similar in the data model as a *ServicePoint* for a *Stop*, but that it has a more aggregated interpretation: It can relate to a more aggregated physical network, making it possible to 'zoom out' from the very detailed representation of the network using *Stops*.

The *StopGroup* can be used to ease the aggregation of a public transport network, e.g. for the use in the timetable presented to the public. For instance, main timetables for rail seldom describe from which platform (*Stop*), the train departs from, this information is only displayed at the station or used at the control centre.

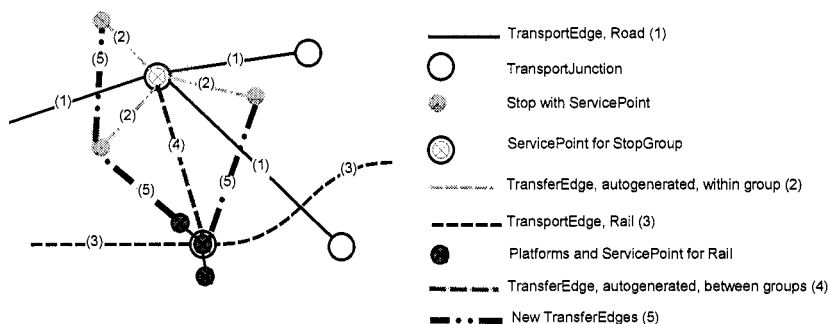


Figure 4: Relationships between objects in a terminal

TransferEdges between the *ServicePoint* and the *Stops* can be generated automatically, while one may need to define more *TransferEdges*, e.g. in the case where the path between two *Stops* in a *StopGroup* would be unnecessarily long, using auto-generated *TransferEdges* (as in figure 4).

StopGroups can be grouped together into *Terminals*. The *ServicePoints* within a *Terminal* can then be connected with auto-generated *TransferEdges*, and manually defined *TransferEdges* can connect *Stops*, *StopGroups'* *ServicePoints* and *Terminals'* *ServicePoints* as needed.

A *ChangeEdge* is an auto-generated edge that is not stored permanently in the model, but used to display all possible links between *Stops*, or to show all possible transfers within a transit terminal. *ChangeEdges* are accordingly closely related to tools and procedures built on TOP.

TransferEdges describe the possibility of changing (walking) from one stop to another. As such, it is an aggregated representation of the underlying walk paths. This relation to a more detailed network can be described by using

RouteSegments – similar as for public transport routes (see section 2.3) - or by using linear referencing.

Another possibility is to build a sub network within the *Terminal* of *TransferEdges* and *TransferNodes*. However, the same can be obtained by building a physical network of *TransportEdges* and *TransportNodes* restricted to walk as mode.

Finally, a *TimingPoint* can be used to denote a *ServicePoint* of special interest. As an example important checkpoints for a schedule, a bypass section at a rail line, or the passage between two *FareZones*.

Also the *Terminals* can contain connectivity rules, so that certain *Stops* can only be connected within certain *TransportEdges* (e.g. a rail line cannot stop in a ferry berth). Also a number of rules can define which modes, vehicles, units and companies and departments that are allowed to use the specific *Stops*, e.g. that a platform is too short for certain trains or that a *TransferLink* is not allowed to be used for containers containing dangerous goods.

2.3 Route Networks

The public routes are described on top of the physical infrastructure network (figure 5). This is made possible by the *RouteSegment* object.

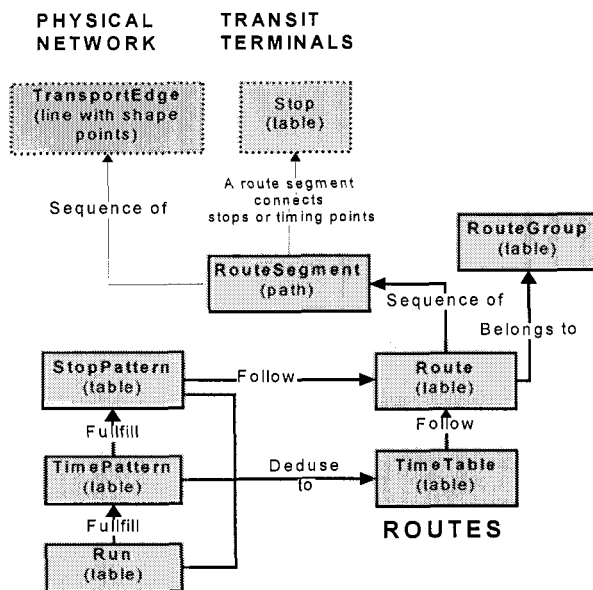


Figure 5: Relationships between objects in a terminal

The *RouteSegment* is the definition of a connection between two *Stops*. It describes the path between the two stops through the network, i.e. through a

sequence of *TransportEdges* (and parts of *TransportEdges*) along the path (figure 6). Thereby, *RouteSegments* are described unambiguously; which would not be the case if the path was described by a sequence of nodes, as in some exchange formats (e.g. that other paths could connect the two *ServicePoints* in figure 6).

A *Route* describes the route of a public transportation service as a sequence of *RouteSegments*. This means that a *Route* is basically a series of stops connected by a path.

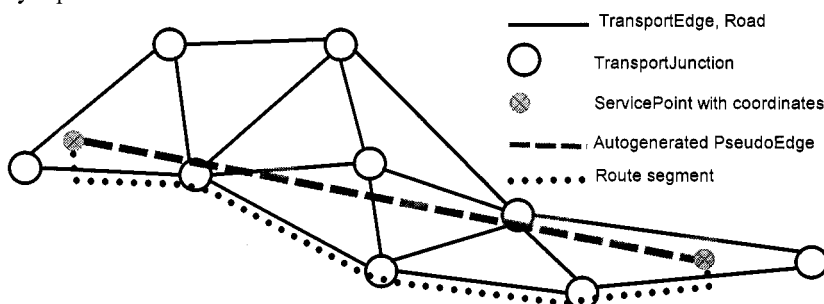


Figure 6: Relationships between objects in a terminal

Detailed information is added to the *Route*, by using a *TimePattern* and a *StopPattern*. The *TimePattern* is a table that corresponds to the sequence of *RouteSegments* in the *Route*. It describes the relative time needed to travel from stop to stop. The *StopPattern* describes which of the *Stops* in the sequence that the *Route* actually stops at. An example is an express train only stopping at certain stations (*Stops*) along the route, while the slow train has more stops.

The users are not restricted to define one *TimePattern* or one *StopPattern* in conjunction with a *Route*. A series of *Runs* are then specified for the route, and for each *Run* a *TimePattern*, a *StopPattern* and a time of departure from the first stop is specified.

For a highly non-regular timetable (typically long distance services), each run has its own *TimePatterns* and *StopPatterns*. While regular services may only have one *TimePattern* and *StopPattern*, e.g. a metro line running each 3 minutes along the same path and stopping at the same stops throughout the day.

It is noted that several *Routes* can use the same *RouteSegment*, e.g. in the case of a bottleneck in a rail network, or bus lines between two islands using the same ferry.

Finally, the *RouteGroup* object is used to group together a collection of *Routes* and their *Runs*. This makes it possible to describe public transport lines whose timetable or path varies with time, as a whole.

As illustrated in figure 7 the representation of a timetable based public transport service in TOP, is performed using 3 layers of data:

- A physical network, describing roads, railroads, bike and balk paths etc.

These elements describe physical objects, and are easily modelled as simple network features.

- On top of the physical network part of the public transportation network is described. The data described in this layer is basically *Stops*, *Terminals*, *Transfers* and *RouteSegments*. *RouteSegments* are the necessary as building blocks for a detailed description of public routes. None of the elements contain any time dependencies.

The *RouteSegments* are defined using a table of linear references to the underlying infrastructure edges. That basically means a table, which describes what *TransportEdges* and parts of *TransportEdges*, a given *RouteSegment* consists of.

- Finally, as the topmost network layer, a combination of features and tabular data provides a detailed, time dependent description of the public routes

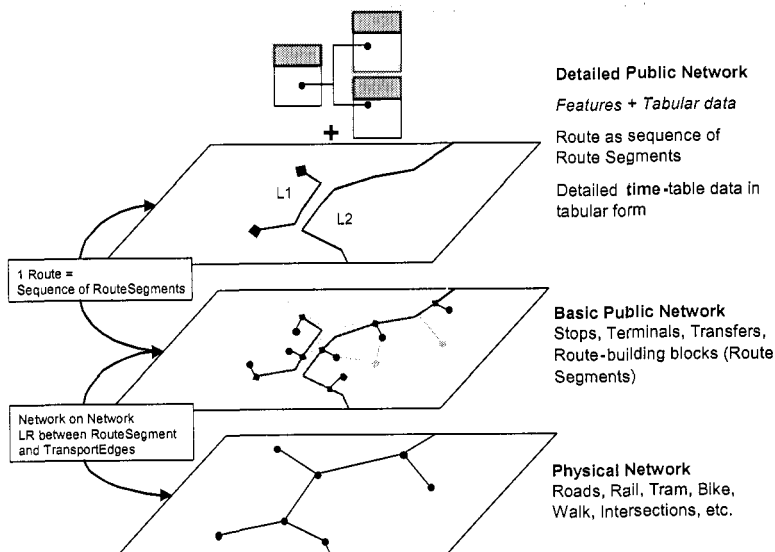


Figure 7: The representation of a timetable based public transport service: Interaction between objects in the physical network, route network and terminals.

TOP contains functionality to ensure the consistency of the relationship between *Routes* and the physical network, e.g. so that describe that rail routes can only run on rail tracks. In addition, a number of rules can define modes, vehicles, units and company/departments that are allowed to use the specific *Routes*.

2.4 Demand

The *Demand* group of objects is used to describe the need for transportation (figure8).

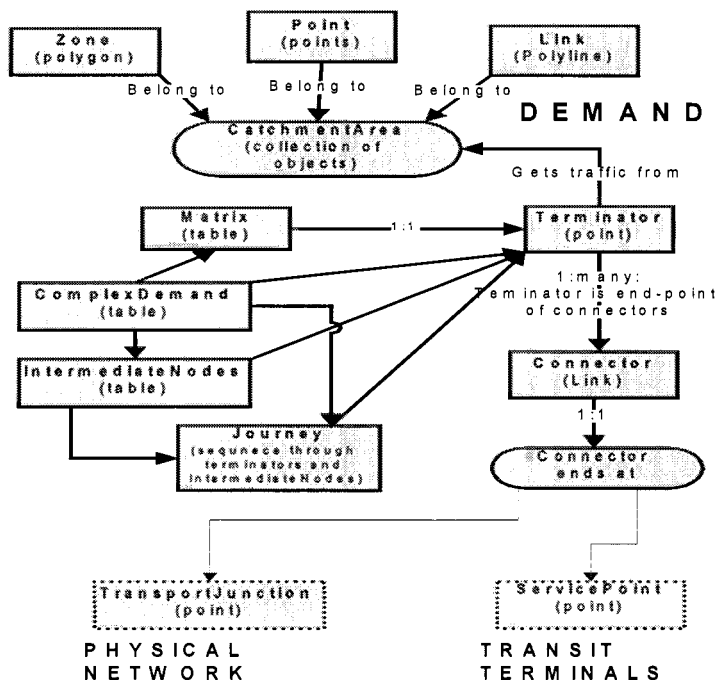


Figure 8: Conceptual model for Demand

The *Terminator* - a generalisation of the zone centroid commonly used in transport models - represents an entry/exit point as a Junction. The *Terminator* can be connected to all kind of objects that are aggregated representations of real-life objects that generates or attracts traffic, e.g. zones, junctions, shopping centres, or roads in a residential area. These are grouped together as the *CatchmentArea*.

The *Matrix* describes data on a *Terminator*-to-*Terminator* basis. Data being described is typically traffic (persons, goods, cars) or travelling times, travel distances or other calculated results.

The *Connector* object connects a *Terminator* to the transport network at a *TransportJunction* or at a *ServicePoint*. The latter can represent a *Stop*, a *StopGroup* or a *Terminal*. A *Terminator* can be connected to a network by several *Connectors*. Note that route choice application may need rules to prevent paths using a sequence of *Connectors*, i.e. *Connectors* may only be used from the 'from-*Terminator*' and for the 'to-*Terminator*' along the path.

Compared to traditional traffic models, the *CatchmentArea* is a generalisation of zones, the *Terminator* of zone centroids, and the *Connector* of fictive links. In a future version of TOP, a *ComplexDemand* object will probably be introduced. Its purpose is to generalise the *Terminator*-to-*Terminator* description of traffic flow, extending it to a travel chain, with a number of mandatory destinations or transportation modes between the *Terminators*, at intermediate nodes. Accordingly,

a *Journey* can describe a sequence of traditional 'trips' in transport models. This may be used in activity based traffic models, which describe the daily activities and trip chains derived from this. Or the *Journey* can describe the distribution of goods by the use of e.g. travelling salesman algorithms.

3 Conclusions and perspectives

The paper described the intentions behind the Transportation Object Platform (TOP) developed as an extension to the GIS *ArcInfo 8*.

By utilising the object-oriented approach in ArcInfo, it was possible to build far more complex topological models for transportation network than what has been possible previous. This makes it possible to eliminate the many proprietary solutions for data and models in today's practice of transport modelling and management. A practical example is the transfer of the Copenhagen-Ringsted Model (Nielsen et al 2000 [2] & Kaas [3]) to TOP. The first step integrated the many data models and sources already embedded in KRM. The next allows for integrating stops within stop groups into the model. TOP can accordingly increase the consistency of data models, and provide a platform for building more advanced tools for the editing, visualising and analyses of transport data.

In practice, TOP can ease and support the work in public transport companies, operators of public transport, freight companies, infrastructure owners, planning authorities, and other organisations, that works with infrastructure and transport modelling. In some cases as is, in other by extending TOP's domain by new objects.

Furthermore, TOP is a software tool for implementing new and more efficient multi-modal transport models. This is both due to the built-in routines for data management, and due to the facilities to generate the needed sub graph of the network, as well as to import calculation results from these. This is both of interest for research institutions and software developers. Several Danish Research projects are utilising this for both passenger and freight transport modelling.

References

- [1] Nielsen, O. A., Mandel, B. & Ruffert, E. (2001). *Generalised Transportation-data Format (GTF): Data, Model and Machine Interaction*. European Transport Conference (PTRC), Proceedings, Seminar on Applied Transport Methods, September, Cambridge.
- [2] Nielsen, O.A., Hansen, C.O., Daly, A. (2000). *A large-scale model system for the Copenhagen - Ringsted railway project*. International Association for Travel Behaviour Research Conference. Gold Coast, Australia, 2000.
- [3] Kaas, A. H. (2000) *Punctuality model for railways*. COMPRAIL 2000 in Bologna.