D13-GTF Specification

"Generalised Transportation-data Format" Specification document

Deliverable D13

Project: SPOTLIGHTS-TN (Thematic Network)

Contract No 1999-TN.10941

Project Co–ordinator

MCRIT SL (Barcelona)

WP Leader

MKmetric Gesellschaft für Systemplanung mbH (Karlsruhe, Germany)

WP Contributor

-

Other Partners

DTU, Technical University of Denmark (Copenhagen, Denmark)

Date: 28 November 2001

PROJECT FUNDED BY THE EUROPEAN COMMISSION UNDER THE TRANSPORT RTD PROGRAMME OF THE 5th FRAMEWORK PROGRAMME

Responsibilities & Contacts:

European Commission Responsible

Commission of the European Communities
Directorate–Transport and Energy (TREN)
Anna Panagopoulou
Rue de la Loi 200
B –1049 Brussels
Belgium

Tel: +32 (2) 296 78 94
Fax: +32 (2) 295 43 49
email:
anna.panagopoulou@cec.eu.int

WP Leader

MKmetric Gesellschaft für Systemplanung mbH
Durlacher Allee 49
D – 76131 Karlsruhe
Germany

Tel.: +49 721 961 60 0
Fax.: +49 721 961 60 99

Theory:
Dr. Benedikt Mandel
email: mandel@mkm.de

Project & Development:
Eduard Ruffert
email: ruffert@mkm.de

WP Contributor

Project:	SPOTLIGHTS
Technical Note #	5 © MKmetric
Description	**GTF Conceptual Model (**Specification**)**
Workpackage:	4
Title:	".gtf"
Deliverable:	D13 (draft)
Title:	GTF Specification
Author:	Dr. B. Mandel, E. Ruffert
Version:	**v0.6.r5 "Barcelona"**
Original Date:	1st January 2000
Last Updated:	30 November, 2001 By: E. Ruffert
Distributed to:	Project Officer (A. Panagopoulou), Project co-ordinator (MCRIT), Partners

List of Technical Notes

Comments	#	Doc. name	Title	Contents description	Updated
Distributed/handout to all partners	1	/p/spotlights/MKmetric/MKmetricTasks.doc	List of envisaged MKmetric Tasks	List of MKmetric's involvement in SPOTLIGHTS	26th January 2000
Distributed at the workshop DG/TREN 5th April 2000 in BXL	2	/p/spotlights/MKmetric/gtf/gtf.rtf	GTF Data Model Specification (Draft)	UML diagram and specification of GTF Data Model Objects and relationships.	4th April, 2000
Inception document, start-up discussion document between NEA, MKmetric (and partners)	3	/p/spotlights/MKmetric/gtfdocu/gtfoov02_r1.rtf	GTF Data Model Specification (Draft for Scientific Committee)	UML diagram and specification of GTF Data Model Objects and relationships.	6th April 2000
Initial preparatory discussion document	4	/p/spotlights/DataFormats/ComparisonFormats/GTF_Part1.doc	GTF Data Model	Evaluation and Comparison of data models and formats pertaining to modelling and exchanging of information. Suggestions for homogenisation & implementation.	8th December 2000
Update after BCN meeting 11th – 12th October 2001	5	/p/spotlights/DataFormats/ComparisonFormats/GTF_Specification.doc	GTF Conceptual Model	Update of specification document after Barcelona workshop	28th November 2001
Update after BCN meeting 11th – 12th October 2001	6	/p/spotlights/DataFormats/ComparisonFormats/GTF_overview.doc	GTF Overview, an executive summary of the GTF vision, task and the GTF CM	Update after Barcelona workshop	28th November 2001

Update after BCN meeting 11th – 12th October 2001	7	/p/spotlights/DataFormats/ComparisonFormats/GTF_adhocXML.doc	GTF ad hoc XML, an ad hoc specification for an XML mapping of the GTF CM structures	Update of specification document	28th November 2001
Update after BCN meeting 11th – 12th October 2001	8	/p/spotlights/DataFormats/ComparisonFormats/GTF_draftTIP.doc	draft GTF TIP, a draft for a specification for a run&command language for remote transportation models	Update of specification document	28th November 2001

GTF Conceptual Model (Specification)

Classification: spotlightsTN **Restricted**
Issued by: MKmetric GmbH
Durlacher Allee 49
76137 Karlsruhe
Germany
spotlights@mkm.de

Table of Contents

1 GTF CONCEPTUAL MODEL

Main modifications from previous version (v0.6r4 “Copenhagen”):

* the “TransportProduction” class was renamed to “Factor” because the previous name was objected to as it rather intuitively implies a too restricted view of the information that might be contained in objects from this class.

* the Factor class was promoted to a toplevelclass.

* this version of the documents are code named ‘Barcelona’version v0.6r5.

1.1 Introduction

This document specifies a GTF Conceptual Model on the abstract level (GTF-CM). It also specifies a first attempt at an implementation level specification of GTF. The specification was used to create sample C++ code to generate sample data in memory.

Basically, when one speaks of transportation modelling one is talking about two models that are used simultaneously to solve (or answer) a problem, one a model of the transportation infrastructure (i.e. a model describing the supply side of transportation) and another behavioural model (i.e. a model describing the demand side of transportation). These models reduce the complexities of the real world into manageable chunks. And, in principle, can be handled separately.

The infrastructure model defines networks (infrastructure supply side or abstract networks), vehicles (e.g. cars, trains, airplanes etc.) and services (facilities for loading and unloading at a port) etc. that are based on the real world (observable) 'things'.

The behavioural model defines

1. Abstractions of Zones, Zone features, choice alternatives etc., either in an aggregate or in a disaggregate fashion

2. The way the formulated actors of the problem domain react and decide, given sets of choice options. This model is often based on survey data. The more disaggregate the model (and therefore the required survey data) is, the more complex the model becomes mathematically. But disaggregate models have the advantage of being more accurate in forecasts and in analysing behaviour. (At this level there are many connections to social science, because both try to explain differences in behaviour of groups based on their social, economic etc. differences.)

Finally, the GTF-CM is another reduction of complexity, making the modelling information manageable for EDI. Grouping and classifying the modelling information reduces the complexity of the problem domain. For example, the concepts Zone and Junction are very different in the problem domain (and the usual models), but they share a common function of being ending points of Links[1]: Zones as 'end points' of Flow-Links and Junctions as 'end points' of (infrastructure network) Segment-Links. These kinds of abstractions are the gist of conceptual modelling and the contents of this chapter.

Some basic concepts from economic theory (i.e. supply side determinants, demand side determinants and the market where supply meets demand) were used to develop the concepts for the conceptual model. The next figure depicts the main conceptual

[1] Nouns in capital letters, e.g. "Zone", refer to classes defined in this document. These are part of the GTF conceptual model (GTF-CM).

classes used in modelling the information typically found in the problem domain.

Briefly described, the GTF-CM consists of:

The Factor class represents the determined generated or attracted movement, which induces the demand for movement and transport. Factor objects contain the data for, e.g. the GDP, age distribution, level of income etc. for one Zone, a group of Zones or an aggregation of Zones. The concept behind Factor and Zone is the following: an area itself (e.g. 10 km^2) cannot be the reason why traffic or movement is produced or attracted. The reason why there is movement to and from an area is that the area has a number of features that a traveller to an area takes an interested in. In this document these features, e.g. people, industry etc., are called Factor in analogy to the concept of production factors in economic theory. A Zone then, is the combination of an area (either in physical space or in the modelling space) and the Factor objects that are located in the Zone's area. The combination of Factor objects and a Zone is represented by the relationship 'activity', because a Factor in a Zone generates some sort of activity, either attracting movement or producing it.

The Zone class is a geographical coverage, which contains the factors (Factor objects) that generate or attract movement. It is connected to a network through the Terminator class and a Connector (-Link). The Connector is the virtual description of the impedance that is needed in average to enter / leave a Zone (and thus creating inter-zonal transportation / movement called a 'MatrixElement').

The Terminator class represents a virtual point for input & output (source & sink) of movement in networks. The Zone class is a virtual pool containing the Factor objects of an area of a Zone. In this context the virtual input & output points are Terminators (- in other texts these are sometimes referred to as 'connector nodes' or 'centroids').

A Node performs three functions. The first function is to relate (connect) a Zone to some point in the network as access and egress points. This function determines the Node as being a Terminator. The next function is that of being a Junction point in an infrastructure network. These Junctions describe topological aspects of networks, i.e. which Junctions are connected to which other Junctions. The Junctions and the connections (branch, arc, edge etc. later these connections will be summarised by the term Link) between them are the topological description of the infrastructure networks. And thirdly, the Node is the abstract (in the logical sense) super-class of Zones. A Node is called an abstract class, yet it is not in the strict sense used in object-oriented (OO) specifications, where abstract classes cannot be instantiated, but this specification allows instances of Node (the same holds for also e.g. Link and the other toplevel classes). The reason for this is, so that users of GTF can transmit information concerning networks without being restricted to the concrete semantics of 'transportation networks', where the talk is rather about Junctions and Segments. But what if users want to transmit information at a more abstract level? That is, they want to talk about Nodes and Links in the sense of graph theory without additional problem domain semantics. This requires being able to instantiate the Node class. Therefore the toplevel classes are not abstract in the OO sense, but abstract in the logical sense.

A Link is a topological relation between two Nodes. Depending on the combination of the types of starting or ending Node there are three types of Links: an infrastructure Segment-Link - a connection between two Junctions, a Connector-Link - between a Terminator and a Junction or vice versa, and a MatrixElement-Link - a connection between two Zones.

The MatrixElement class stores the result of Factors generating & attracting movement across the boundaries of Zones. It can therefore be described as a connection (relationship) between two Zones, i.e. in the sense of demand for transportation or observed/computed flows between Zones.

A Vessel is the abstraction of everything that increases the movement-count on any Link. Typical Vessels in transportation models are airplanes, cars, trains, airplays, trucks etc.

An ‘activity’ is a term for ‘determines demand of’. This term was chosen, as it describes better the concept of the association between class Factor and class Zone. ‘Activity’ can be seen as abstraction of the attractiveness of Zones or the potential for a visitor of a Zone to see sights etc. ‘Activity’ shall describe everything that induces movement/transportation to/from a Zone.

The ‘has Terminator/localises’relationship models the localisation of a Zone in a network.

The class Alternative comprises all the definitions for a Link that are derived from the modelling-side of the information. For example, the ‘main mode’might be ‘road’. This actually includes road segments as well as car ferry ship links etc. The Alternative class associates this information with a Link.

The Chain class represents all kinds of sequential information, e.g. Services and Paths. These are sequences of Nodes and Links in different forms.

The DynamicSegmentation class is used to attached information on Links which pertain either only to a part of a Link, e.g. a speed limitation for the first part of a Link or information pertaining to more than one Link, e.g. a portion of one Link and a partion of another Link marked by a milestone.

The Unit class describes all that can travel on Links, e.g. Persons and Goods.

The instances of the Grouping class can be used to collect any other objects represented by one Grouping object. This is useful for example to assemble all the objects referring to a “Scenario”. (Note: a “Scenario” is sometimes referred to as a “policy change” in other documents. In this document a “Scenario” means all enogeneous information that can be changed to describe the implementation of a specific measure or a change in the state of the world, e.g. the cost on some Link is increased or the GDP of some Zone changes by some percentage.)

Finally, Meta objects are used to specify lengths and other physical measurements needed by the values of the member attributes of the other classes in the conceptual

model, e.g. the length of a Link is stored in a Technical object attribute ‘length’ = 2. With an association of this object to a Dimension object of value = ‘km’, this specifies a Link of 2 km length.

These are the topmost classes in this model. They form the business logic layer extracted from the problem domain of strategic transport modelling. An applications programmer typically will have to add a user presentation layer, with classes like ‘map’, ‘presentation folder’, ‘projection’ etc. and a data layer which is the connection to a backend repository, e.g. to a database to store the information contained in a GTF transmission. (Note: For example, the pointer to OpenGIS objects in the GTFObject specification is the linkage between the business logic layer of GTF and a data layer, in this case specifically for GIS and the pointer KIF is the connection to another business logic layer, namely from the transportation problem domain to the ‘knowledge representation’ layer).

1.1 Overview Diagram

This figure depicts the highest level view on the GTF Conceptual Model representing the ‘vocabulary’ of the problem domain of transportation modelling.

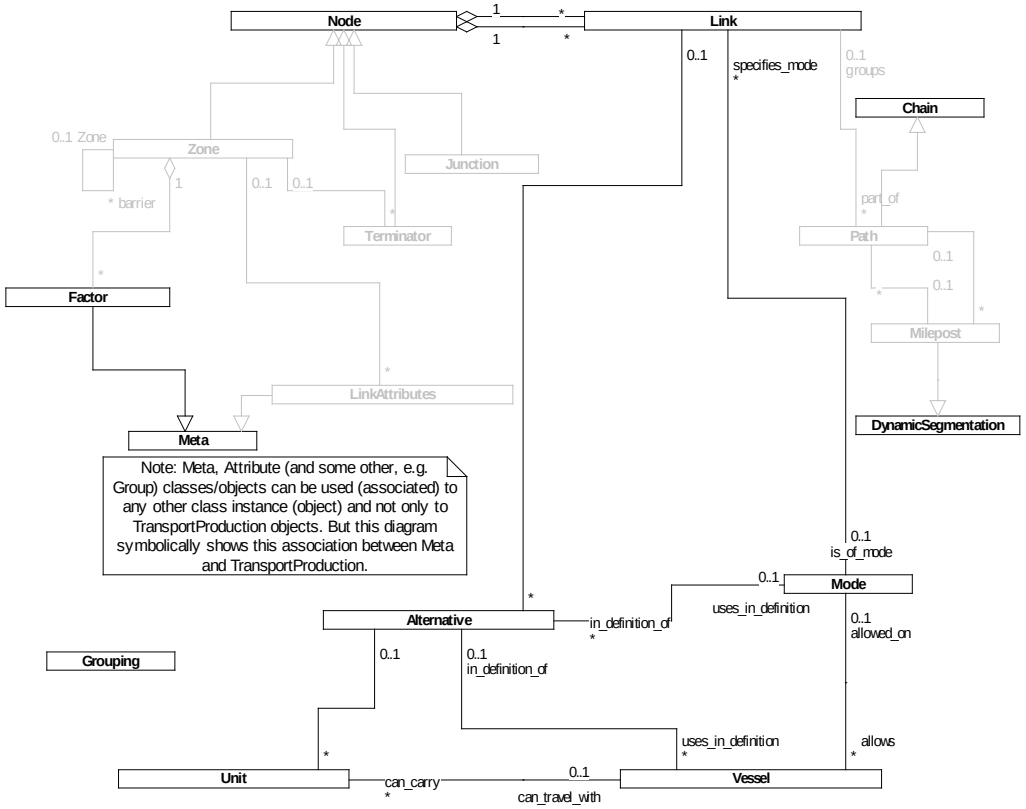


Figure 1: *Overview diagram*

1.2 Toplevel Classes

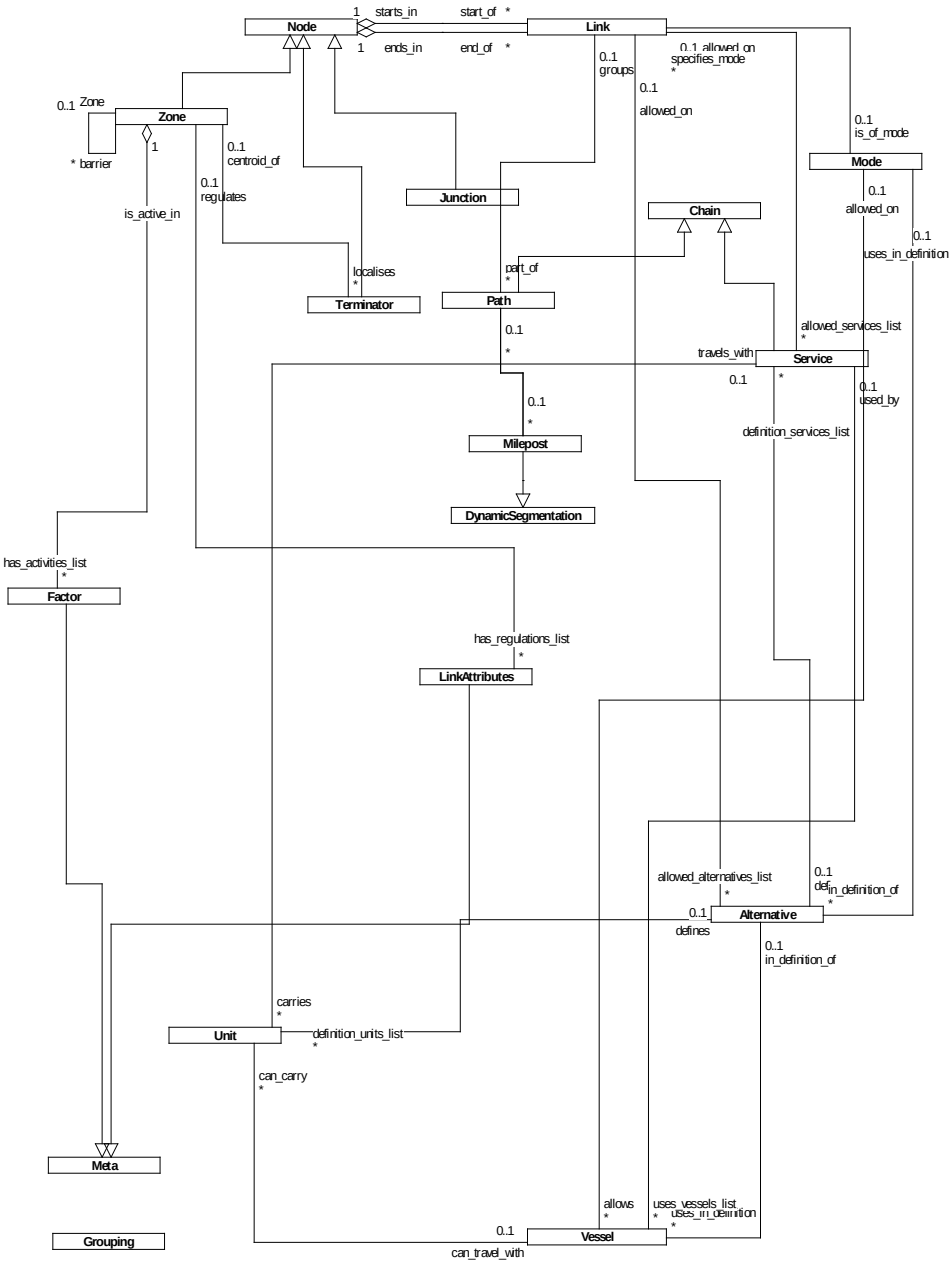


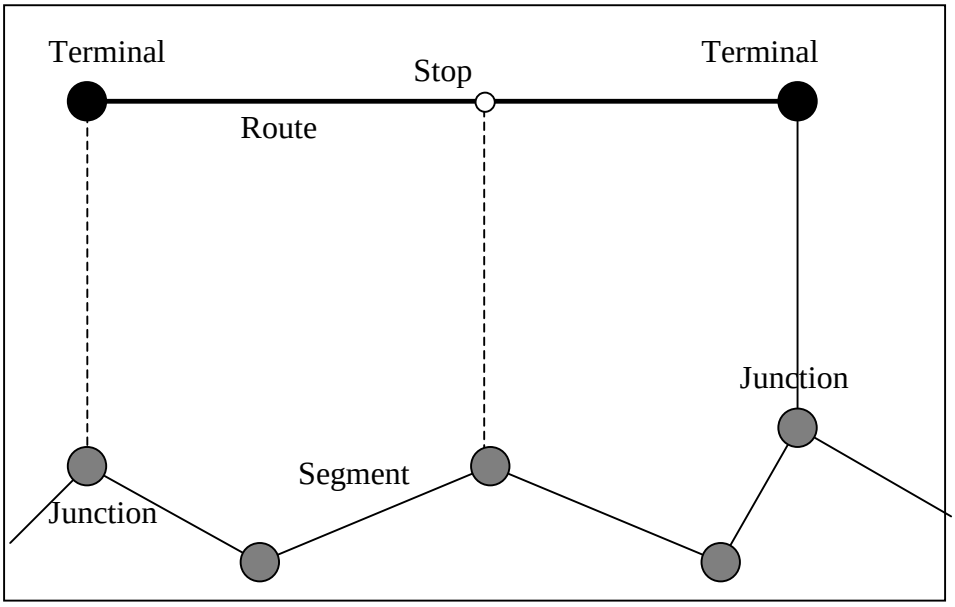
Figure 2: *toplevel classes*

2 USE OF GTF, EXAMPLES

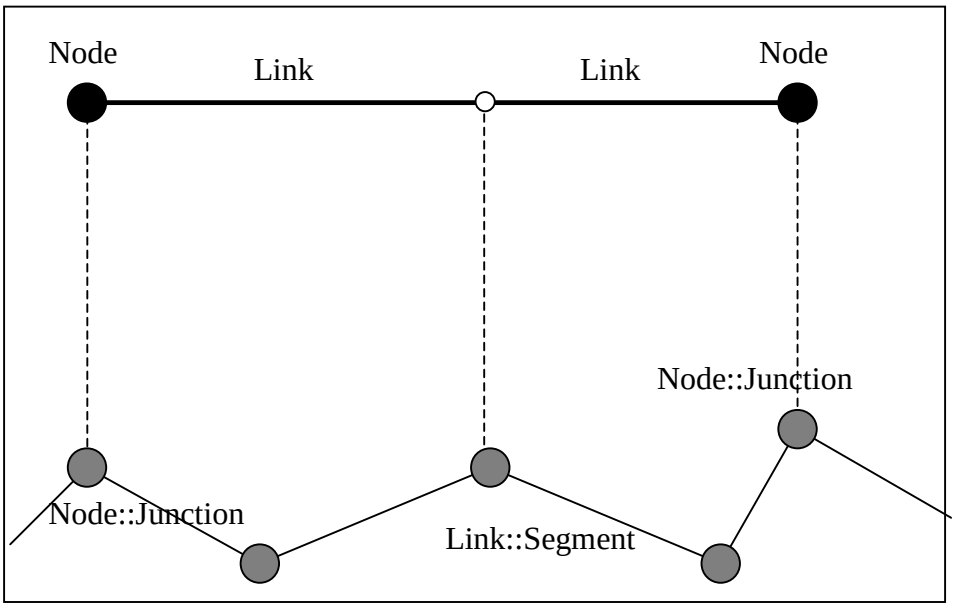
In the following, two examples are described. These shall explain the philosophy of how to use the GTF-CM to describe proprietary data structures.

2.1 Public transport: Route, Stops

Routes and stops can be seen as a logical (or semantic) layer on top of any transport infrastructure like roads etc.:



Using the GTF-CM, this would be represented as:

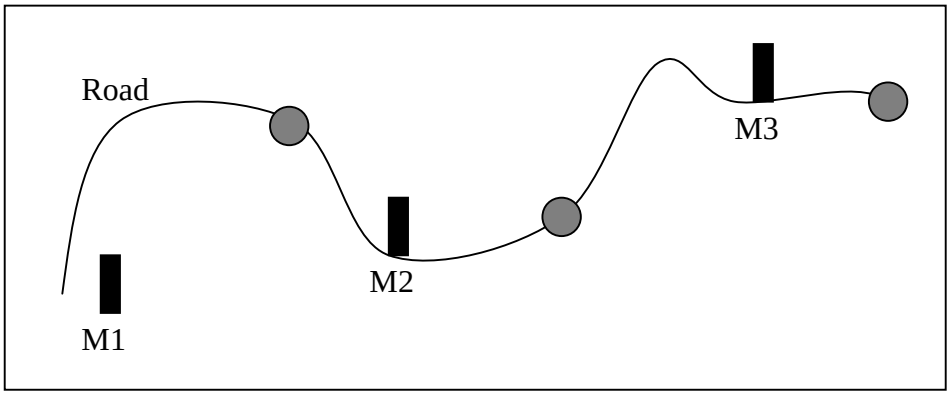


The point to note here, is the Route is split into two Links.

Using the super/sub association, the Node-Link-Node structure can be associated to another ‘virtual’ Node representing the complete Route or using the Path class, the structure is explicitly kept as a separate information in the representation in the GTF-CM.

2.2 Dynamic Segmentation

Many data sets rely on dynamically segmented information:



The initial idea in the GTF-CM was to add members to the Node class, called ‘distance_from_beginning’ and ‘distance_from_end’, to be able to specify the distance of a milepost/milestone relative from the start and ending Node of a Link.

As can easily be seen, this does not cover the above problem. Therefore, two new classes were added ‘DynamicSegmentation’ and ‘Milepost‘in order to represent the above problem completely and accurately. The Milepost class inherits the members ‘distance_from_beginning’ and ‘distance_from_end’ from its parent class, DynamicSegmentation. And DynamicSegmentation is defined in the GTF-CM, to be able to attach to any other GTF-CM class (object, in a concrete data set). Like this the above example can be described completely and accurately without losing information or making the description too complicated.

3 NODE

3.1 Classdiagram

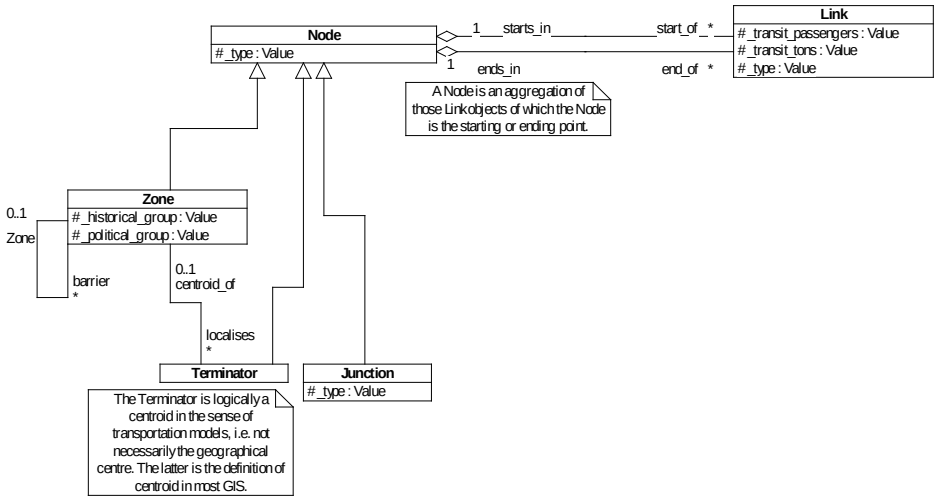


Figure 3: *Node diagram*

3.2 Parent class

3.2.1 Node

This class is the generalisation of the concept ‘start or ending point of Links’ and thus a generalisation of the Zone, Terminator and Junction concepts usually used in modelling and graph theory; and its function is to act as the starting / ending point of Link. Exactly two Node objects therefore determine the generic class Link.

To add a level of semantics one of the child classes should be used.

DEFINITION: abstraction of Terminator, Junction and Zone

FUNCTION: the starting / ending points of Links

3.2.1.1 Inherits

GTFClass

Abstract class from which all other classes are derived.

3.2.1.2 Relations

GTFDB <>--->> Node

List of all Node objects

Value --->> Node

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Node

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

starts_in(Node) <>--->> start_of(Link)

This object is the starting point of the Link

ends_in(Node) <>--->> end_of(Link)

This object is the ending point of the Link

3.2.1.3 Protected Members

Value _type

CODE LIST:

1. Stop

X Left for application programmers if runtime type identification of object types is not available

3.3 Child classes

3.3.1 Junction

The Junctions describe topological aspects of infrastructure networks, i.e. which Junctions are connected to which other Junctions.

If the Junction is used as an aggregation container of other classes, then the Junction is said to be a ‘network’ and one uses this kind of Junction to ‘zoom in’ and to ‘zoom out’ of a Junction in order to see its internal structure.

The English definition (in a computer science context) is:

zoom: <graphics>

To show a smaller area of an image at a higher magnification (‘zoom in’) or a larger area at a lower magnification (‘zoom out’), as though using a zoom lens on a camera. In the context of this specification the definition above is enriched by the concept of ‘zooming in’ to show further topological detail (not only graphical detail) associated with the Junction. But the basic idea of showing more detail or hiding it stays the same. The further detail a (sub-) network can associate with a Junction is, that the Junction is made up of other objects, e.g. a group of Junctions and Links that describes a railway station, an airport or generally terminals and their access and egress points as well as their ‘turns’ and ‘changes’ between Links (or Links of different modes). The super-/sub association of all toplevel classes defines the grouping of all the class instances that make up a specific disaggregation.

3.3.1.1 Inherits

Node

3.3.1.2 Relations

Value ---> > Junction

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation ---> > Junction

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

turn_to(Junction) <>--> turning_to(NodeRestrictions)

The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

turn_from(Junction) <>--> turning_from(NodeRestrictions)

The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

turn_at(Junction) <>--> turning_at(NodeRestrictions)

The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

3.3.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

3.3.2 Terminator

Pinpoints a Zone in space and connects the Zone to an infrastructure network via the Connector. A Zone is a virtual point for input & output (source & sink) of movement in infrastructure networks. For transportation models a Zone is a description of socio-economic and other information of a geographical area. The geographical connection between a Zone and the area it describes, is used to relate specific Factorobjects and their values to specific input and output points in networks.

3.3.2.1 Inherits

Node

3.3.2.2 Relations

centroid_of(Zone) --->> localises(Terminator)

A relationship connecting a Zone to an infrastructure network

GTFAssociation --->> Terminator

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

3.3.3 Zone

Is a virtual pool, which contains the Factorobjects of an area. For transportation models a Zone is a description of socio-economic and other information of a geographical area.

DEFINITION: an area in space (i.e. the modelling space)

DESCRIPTION: a Zone is the logical description of the people, industry etc. features that generate or attract transportation flows. That is, a Zone generates / attracts demand for transportation in an area or is the destination for such movement .

All information that is necessary to describe the Zone geographically is stored in the associated Shape objects or in the GIS objects pointed to by the OGISPointer member.

The ‘historical_group’ and ‘political_group’ member attributes contain information concerning the historically and politically associated groupings of this Zone, e.g. historically a Zone’s grouping might be ‘Germany’ and its politically associated grouping might be ‘Schengen’ or ‘EU’. The provided Meta object can also contain ‘time zone’ and ‘winter summer’ information pertaining to the Zone’s time-Zone or whether or not there is a difference concerning the winter / summer time. This kind of information is mostly relevant for the correct interpretation of time schedules. The syntax is ‘time_zone yes|no’, ‘winter_summer yes|no’.

Note: Zone structures on one (same) layer are not allowed to overlap, i.e. Zones on the same layer must be disjunct. A Zone can be a point at the highest level of detail.

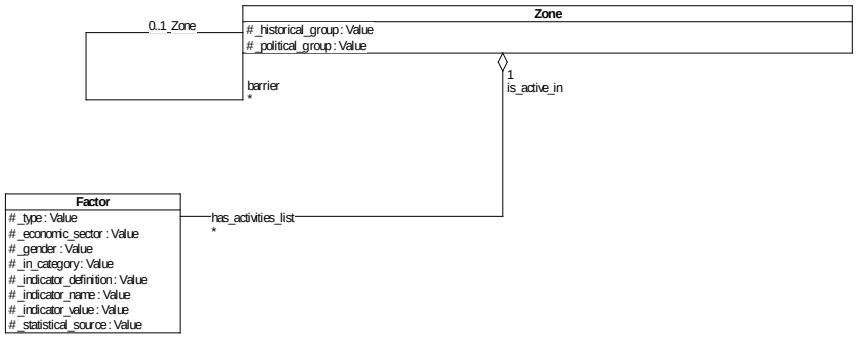


Figure 4: *Zone diagram*

3.3.3.1 Inherits

Node

3.3.3.2 Relations

GTFAssociation --->> Zone

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

is_active_in(Zone) <>-->> has_activities_list(Factor)

A relationship giving a Zone a list of Factorvalues (either in functional or constant form) from which the generated / attracted transportation demand for that Zone is computed

centroid_of(Zone) --->> localises(Terminator)

A relationship connecting a Zone to an infrastructure network

regulates(Zone) --->> has_regulations_list(LinkAttributes)

A relationship giving administrative regulations for a Link based on a Zone's, e.g. country

Zone --->> barrier(Zone)

This association defines separations (barrier) between Zones.

Models often depend on this crucial information, when e.g. two Zones are very similar, but one of them displays a different behaviour than the other (red-bus blue-bus problem). This happens when crucial information is missing in the model, like barriers between Zones, which make it more difficult to travel between two Zones, which are separated by a range of mountains (or by different languages, political

restrictions like needing a visa-pass to visit a country), than to travel between Zones, which are not separated by such a barrier.

The relationship to 'Barrier' objects is used for structuring the modelling information about the obstacles between Zones. For example, a Zone'A'might have two neighbours, Zone 'B' and Zone 'C'. The Factorobjects of both Zones might be the same, but the demand for movement, between Zone 'A' and the other Zones, might be different. So, to model the difference in the demand for movement a transportation model needs additional information, in this case the fact that there is a mountain between Zone 'A'and Zone 'B' and there is no mountain between Zone 'A' and Zone 'C'. The kinds of obstacles that are identified are languages (Is the same language spoken in Zones 'A', 'B' and 'C' or not?) and mountains / lakes / rivers. Note: country or in general Zone administrative boundaries or the boundaries for Zones derived from a Zone-ing system, e.g. NUTS, are represented graphically through the openGIS pointer.

3.3.3.3 Protected Members

Value _historical_group

List of flags defining the historical (country) structures the Zone belongs to, e.g. NUTS Codes (see NUTS Codes) or ISO country codes (see ISO country Codes).

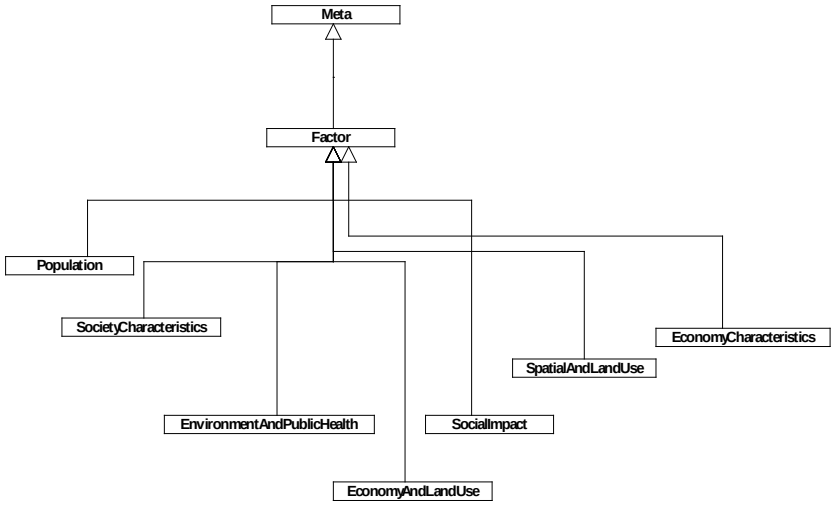
Value _political_group

List of flags defining the political flags the Zone belongs to, e.g.:

CODE LIST:
1. EU
2. Schengen
3. NAFTA
4. GUS
5. Asia
6. America
7.Australia
8. Africa
999 other
DEFAULT: 1

4 FACTOR

4.1 Classdiagram



4.2 Parent class

4.2.1 Factor

These determine the generated or attracted movement of a Zone, which together induce the demand for movement and transport. Factor objects contain for example the GDP, age distribution, level of income etc. for one Zone, a group of Zones or an aggregation of Zones. Additionally, since Factor is a subclass of Meta, any Factor object can be associated to any other GTF object in a data set, not only to Zone but e.g. to Node or Link etc. as many transportation models attach generation or production factors onto Nodes (Junctions etc.) or Links.

DEFINITION: zonal demand information / data

FUNCTION: to describe actors (or group of actors) that are used for Zones in the transportation model. Actors are the reason why movement and flows are generated or attracted.

DESCRIPTION: a Factor object is a piece of data (aggregated or disaggregated), e.g. socio-economic or other statistical data, that is used to compute / describe the potential for transportation demand that an actor / group of actors generate / attract. An actor might be a person, a group of persons, a firm, a group of firms or a branch of industry (globally or within a country). The information need not necessarily be associated to some ‘real world thing’ or even be pinpointed to a specific location. To

add a level of semantics one of the child classes should be used.

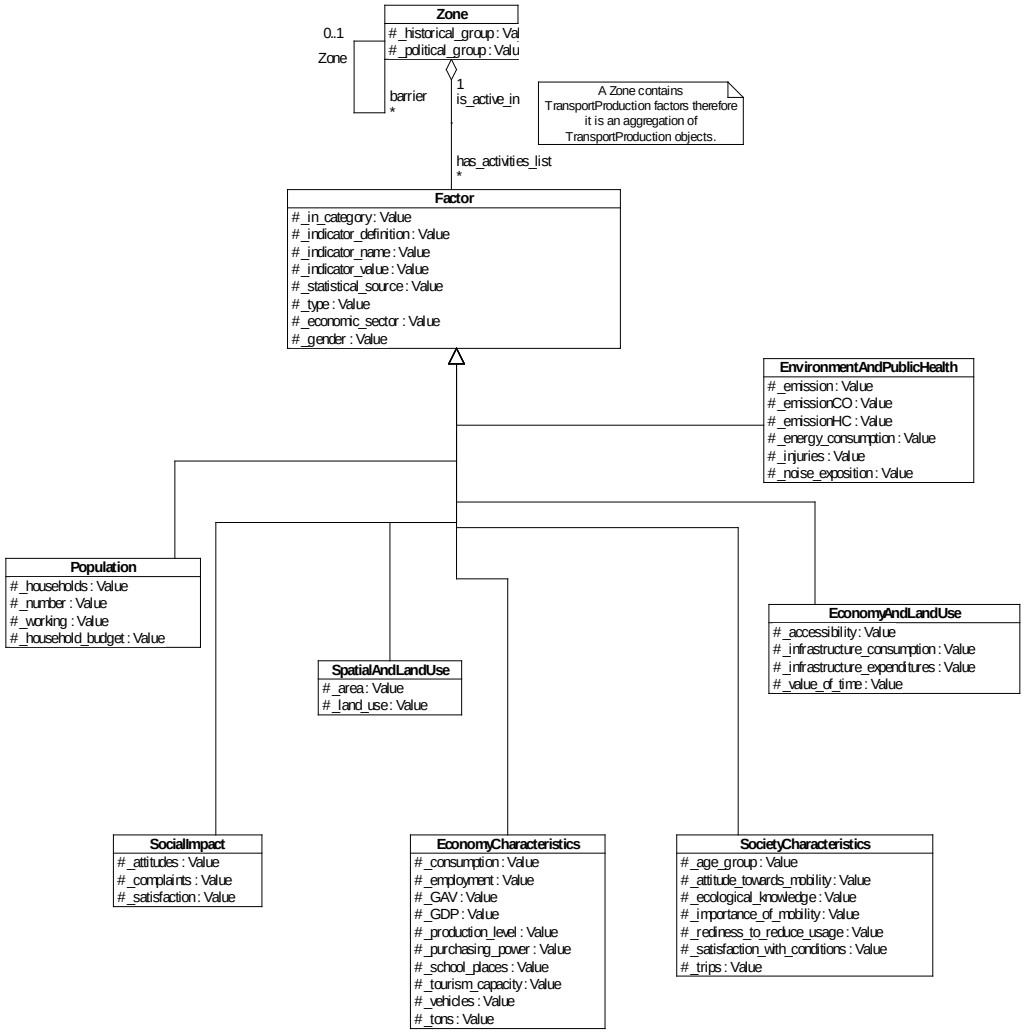


Figure 5: *Factordiagram*

4.2.1.1 Inherits

Meta

Abstract class from which all other classes are derived from

4.2.1.2 Relations

is_active_in(Zone) <>--->> has_activities_list(Factor)

A relationship giving a Zone a list of Factorvalues (either in functional or constant form) from which the generated / attracted transportation demand for that Zone is computed

GTFAssociation --->> Factor

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.2.1.3 Protected Members

Value _statistical_source

A code, typically from a standard list of statistical sources, e.g.

CODE LIST:
1. EUROSTAT
2. INFOSTAT
3. NACE-CLIO
999. Other
DEFAULT: 1

Value _indicator_name

The name of a specific statistical / economic value

CODE LIST:

1. POPULATION as defined by EUROSTAT

2. POPULATION as defined by INFOSTAT

999. Other

DEFAULT: 1

Value _indicator_definition

CODE LIST:

1. C = Constant

2. F = Function

DEFAULT: 1

RULE: If a function is specified (F) for a Factor's 'indicator definition' attribute, then the attribute 'indicator value'shall be a TEXT defining a function f (class instance Id...) using class instance Ids and attribute names, e.g. id the class instance ID101 is an instance of Population, then f(ID101.trips) = ID1.trips * 100, would define the value of this indicator as being 100 * the value of the 'trips' attribute value of the Population class instance with Id = 101. In this way, any kind of mathematical relationship between Factor objects and any value in the conceptual model can be defined.

Value _indicator_value

Actual value of this indicator

Value _in_category

CODE LIST:

1. Yes

2. No

DEFAULT: 1

RULE: ‘yes’ means: ‘check Meta object instances for references to this instance, because they contain further classification information’, e.g. ‘gender: male / female’ etc.

‘No’ means: no need to check.

Value _type

Left for application programmers if runtime type identification of object types is not available

Value _economic_sector

This is a container pertaining to the information on economic sectors in this Zone. See NACE.

Value _gender

This is a flag / qualifier that specifies the information of this Factor pertains to either the females or the males of the Zone.

4.3 Child classes

4.3.1 SpatialAndLandUse

Abstract parent class of spatial and land use Factor. TYPE = ‘Determinants of Transport’.

4.3.1.1 Inherits

Factor

4.3.1.2 Relations

GTFAssociation --->> SpatialAndLandUse

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.1.3 Protected Members

Value _area

This is a value pertaining to the ‘area’ness information of this Zone, e.g. km^2.

Value _land_use

This is a container of information pertaining to the land use in this Zone.

4.3.2 SocietyCharacteristics

Abstract parent class of society characteristics Factor. TYPE = ‘Determinants of Transport’

4.3.2.1 Inherits

Factor

4.3.2.2 Relations

GTFAssociation --->> SocietyCharacteristics

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.2.3 Protected Members

Value _ecological_knowledge

This is an indicator of the level of ecological knowledge (of the population) in this Zone.

Value _attitude_towards_mobility

This is an indicator of the attitude towards mobility (of the population) in this Zone.

Value _importance_of_mobility

This is an indicator of the importance of mobility for the personal living standard and

quality of life (of the population) in this Zone.

Value _satisfaction_with_conditions

This is an indicator of the satisfaction with the conditions of transport and traffic in this Zone.

Value _rediness_to_reduce_usage

This is an indicator of the readiness to reduce the usage of a certain vehicle type (of the population) in this Zone.

If used, must be combined with a Vehicle object defining the type of vehicle. Can be associated to a Vehicle object specifying a Vehicle instance and not a type.

Value _age_group

This is a container of information pertaining to the age groups in this Zone.

Value _trips

This is an indicator of the trips made by zonal population. Meta objects of further information can be attached, e.g. average distance of passengers trips made by zonal population, annual number of passenger- km generated by zonal population, annual number of domestic long-distance trips generated by zonal population, annual number of international long distance trips generated by zonal population, annual number of passenger-km in long distance transport generated by zonal population, average distance of passengers trips made by zonal population, annual distance travelled per inhabitant of Zone, annual mileage per car registered in Zone broken down by type of car owner, vehicle occupancy rate for trips of vehicles registered in Zone by vehicle type, annual domestic interzonal passenger transport flow, annual passenger-km occurring on territory of Zone (can be segmented using Meta objects by kind of infrastructure, by trip distance class, by day type, by time of day, by traffic conditions).

4.3.3 SocialImpact

Abstract parent class of social impact factors.

TYPE = ‘Impacts of Transport’

4.3.3.1 Inherits

Factor

4.3.3.2 Relations

GTFAssociation --->> SocialImpact

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.3.3 Protected Members

Value _complaints

This is an indicator of the population complaints caused by traffic in this

Value _satisfaction

This is an indicator of the stated degree of satisfaction with current traffic conditions in this

Value _attitudes

This is an indicator of transport-related attitudes (in general) in this Zone.

Segmentation must be done using Meta objects (e.g. negatively affected by social group).

4.3.4 Population

Class of population Factor. TYPE = ‘Determinants of Transport’

4.3.4.1 Inherits

Factor

4.3.4.2 Relations

Value --->> Population

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Population

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.4.3 Protected Members

Value _number

A flag / qualifier that specifies that the information of this Factor pertains to the population of the associated Zone.

If used (i.e. non Null) it must contain an entry ‘total’specifying a total population.

The segmentation by gender & age group must be coded using associated Meta objects. These define the classification (e.g. Meta-Gender ‘male / female’) and the value of the classification entry (e.g. male = 0, female = 1).

The segmentation by life cycle group must be coded using associated Meta objects. These define the classification.

The segmentation by car availability group must be coded using associated Meta objects. These define the classification.

Value _working

A flag / qualifier that specifies that the information of this Factorpertains to the population.

If used (i.e. non Null) must contain an entry ‘total’ specifying a total population.

The segmentation by gender & age group sector must be coded using associated Meta objects. These define the classification (e.g. male / female) and the value of the classification entry (e.g. male = 0, female = 1).

The segmentation by gender & economic sector must be coded using associated Meta objects.

The table can contain an entry ‘average working hours per week’.

Value _households

A flag / qualifier that specifies that the information of this Factorpertains to the population.

If used (i.e. non Null) must contain an entry ‘total’ specifying a total population.

The table can contain an entry ‘household size’. The table can contain an entry ‘income’. Average income etc. must be computes from these values or the entry ‘average income’ must be present.

The segmentation by number of cars owned must be coded using associated Meta objects. These define the classification (e.g. Meta-Cars owned) and the value of the classification entry (e.g. enumeration values 0,1,2, many).

Value _household_budget

A flag / qualifier that specifies that the information of this Factorpertains to the population.

If used (i.e. non Null) must contain an entry ‘total’ specifying a total population.

The table can contain an entry share devoted to transport in percent.

4.3.5 EnvironmentAndPublicHealth

Abstract parent class of environment and public health factors.

TYPE = ‘Impacts of Transport’

4.3.5.1 Inherits

Factor

4.3.5.2 Relations

GTFAssociation --->> EnvironmentAndPublicHealth

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.5.3 Protected Members

Value _energy_consumption

This is an indicator of the energy consumption in this Zone.

Segmentation by type must be done using Meta objects.

Segmentation by mode must be done using Vehicle objects.

Value _injuries

This is an indicator of the number of accident injuries in this Zone.

Segmentation by severity must be done using Meta objects.

Segmentation by mode must be done using Vehicle objects.

Value _emission

This is an indicator of the emission rates of gases, dust & particles (in general) in this Zone.

Segmentation by type must be done using Meta objects.

Value _emissionCO

This is an indicator of the emission rates of CO in this Zone.

Value _emissionHC

This is an indicator of the emission rates of HC in this Zone.

Value _noise_exposition

This is an indicator of the number of people exposed to noise above a certain level in this Zone.

4.3.6 EconomyCharacteristics

Abstract parent class of economy characteristics Factor.

TYPE = ‘Determinants of Transport’

4.3.6.1 Inherits

Factor

4.3.6.2 Relations

GTFAssociation --->> EconomyCharacteristics

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.6.3 Protected Members

Value _employment

This is an indicator of the employment in this Zone.

The segmentation by gender & age group must be coded using associated Meta objects. These define the classification and the value of the classification entry.

Value _GAV

This is an indicator of the gross added value (GAV) in this Zone.

Value _GDP

This is an indicator of the gross domestic product in this Zone.

Value _production_level

A flag / qualifier that specifies that the information of this Factorpertains to the

production level in this Zone.

The segmentation by economic sector must be coded using associated Meta objects. These define the classification and the value of the classification entry.

Value _purchasing_power

A flag / qualifier that specifies that the information of this Factorpertains to the purchasing power in this Zone.

The segmentation by age group & gender must be coded using associated Meta objects. These define the classification and the value of the classification entry.

Value _vehicles

This is an indicator of the number of Vehicles in this Zone. Should be segmented by type.

Segmentation can be done using Meta objects.

Value _consumption

This is an indicator of the consumption in this Zone.

Segmentation public & private can be done using Meta objects.

Value _tourism_capacity

This is an indicator of the hotel & accommodations capacity in this Zone.

Value _school_places

This is an indicator of the number of school places in this Zone.

Segmentation can be done using Meta objects (e.g. for segmentation be type).

Value _tons

This is an indicator of the tons handled in the Zone, e.g. annual number of tons originating from Zone, tons abroad, tons in domestic Zone, annual tons of goods in long distance transport originating from Zone, annual tons-km of goods in long distance transport originating from Zone average distances of goods transports generated by zonal economy (km) in domestic Zone, average distances (km) abroad, annual mileage per goods vehicle registered in Zone, loading factor (ratio ton-km/capacity-km) for goods vehicles registered in Zone, annual goods transport flow, annual tons-km occurring on territory Zone,

Segmentation by commodity type must be done using Meta objects.

Segmentation by domestic & international must be done using Meta objects.

Segmentation by vehicle type must be done using Meta objects.

Segmentation by kind of infrastructure must be done using Meta objects.

Segmentation by day type must be done using Meta objects.

Segmentation by time of day must be done using Meta objects.

Segmentation by traffic conditions must be done using Meta objects.

4.3.7 EconomyAndLandUse

Class of economy and land use Factor.

TYPE = ‘Impacts of Transport’

4.3.7.1 Inherits

Factor

4.3.7.2 Relations

GTFAssociation --->> EconomyAndLandUse

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

4.3.7.3 Protected Members

Value _accessibility

This is an indicator of the accessibility of this Zone.

Value _value_of_time

This is an indicator of the value of time for travellers in this Zone.

Value _infrastructure_consumption

This is an indicator of the consumption of infrastructure in this Zone e.g. in km^2.

Value _infrastructure_expenditures

This is an indicator of the infrastructure expenditures in this

5 LINK

5.1 Classdiagram

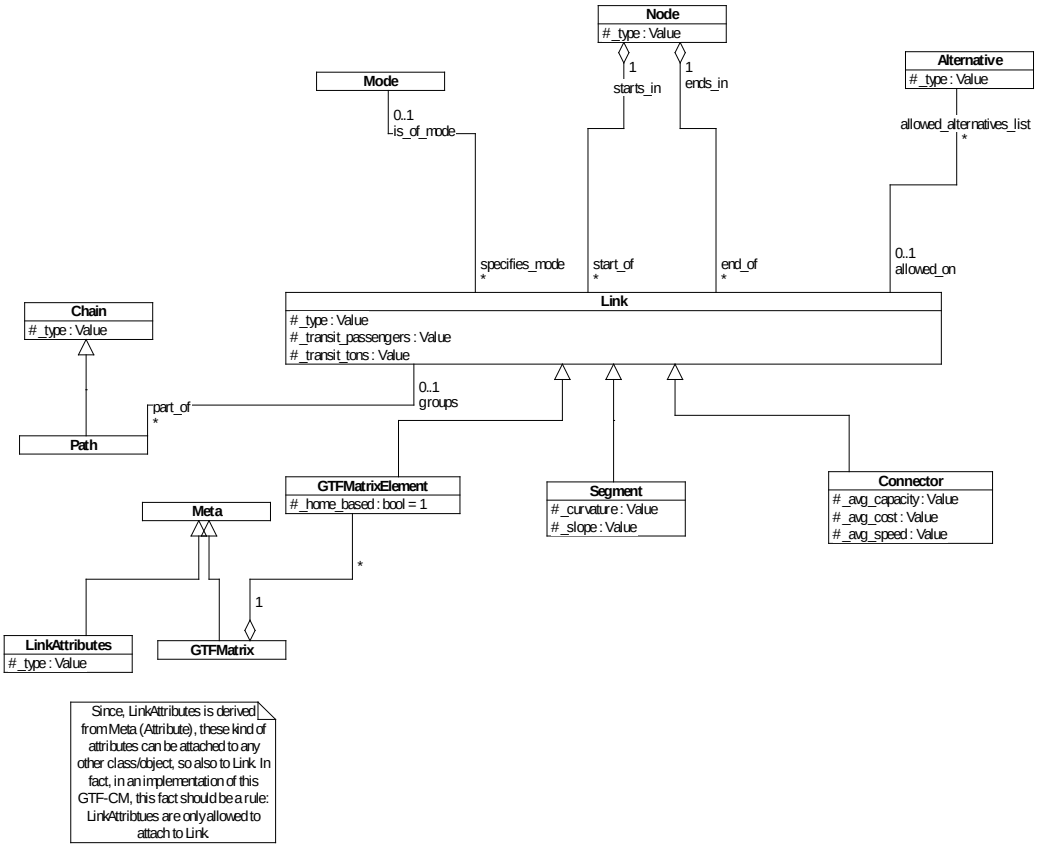


Figure 6: *Link diagram*

5.2 Parent class

5.2.1 Link

The Link class is not only an abstraction for all types of infrastructure network Links, but it also incorporates the connections between two Zones when modelling flows and the connection between a Zone and a Terminator. The three possible types of Link are (depending on the combination of Node types):

1. Segment: is a Link that is used to describe the supply-side of transport, i.e. infrastructure elements that supply the possibility of movement / transport, e.g. roads, rail tracks

2. Connector: is a Link that describes the avg. travel-times, costs, speeds describing the avg. disutility to reach (any) point in the Zone. ·

3. MatrixElement: is a Link that holds the flow information that results when two Zones are connected to describe the movement between two areas in space.

DEFINITION: a connection between exactly Nodes

FUNCTION: a logical carrier of information between two points

To add a level of semantics one of the child classes should be used.

5.2.1.1 Inherits

GTFClass

Abstract class from which all other classes are derived from

5.2.1.2 Relations

starts_in(Node) <>--<< start_of(Link)

This object is the starting point of the Link

ends_in(Node) <>--<< end_of(Link)

This object is the ending point of the Link

GTFDB <>--<< Link

List of all Link objects

is_of_mode(Mode) ---<< specifies_mode(Link)

The associated Mode objects define type of immobile infrastructure of the Link.

GTFAssociation ---<< Link

allowed_on(Link) ---<< allowed_alternatives_list(Alternative)

A relationship associating the (model's) choice-alternative structures with a Link

allowed_on(Link) ---<< allowed_services_list(Service)

A relationship allowing the usage of a Link by a Service

groups(Link) --->> part_of(Path)

5.2.1.3 Public Members

Value _direction

Enumeration:

0 = bi-directional

1 = uni-directional, the Link goes from the start Node to the end Node.

5.2.1.4 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

The type of the Link is defined by to the start and end Node types, see RULE.

Value _transit_tons

Number of transit tons

Value _transit_passengers

Number of transit passengers

5.3 Child classes

5.3.1 Connector

This is a Zone to Terminator Link or vice versa. Connector Links attach a Zone to some point in infrastructure networks as access and egress points.

5.3.1.1 Inherits

Link

5.3.1.2 Relations

GTFAssociation --->> Connector

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

5.3.1.3 Protected Members

Value _avg_speed

Average speed between the Zone and the Junction

Value _avg_capacity

Average capacity on this Connector Link between a Zone and a Junction

Value _avg_cost

Average cost of travelling between Zone and Junction

5.3.2 Segment

This is a Junction - Junction Link.

This class is a type of Link that is used to describe the supply-side of transport, i.e. infrastructure elements that supply the possibility of movement / transport, e.g. roads etc.

5.3.2.1 Inherits

Link

5.3.2.2 Relations

GTFAssociation --->> Segment

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

5.3.2.3 Protected Members

Value _curvature

Curvature information, e.g. %

Value _slope

Slope information, e.g. %

5.3.3 GTFMatrixElement

This is a Zone - Zone Link. It contains the result of Factorobjects generating & attracting movement across the borders of Zones. It can therefore be described as a connection (relationship) between two Zones. A MatrixElement is the container of information that exists when two specific Zones are connected, because there is

demand for movement between them.

DEFINITION: a connection between two Zones

FUNCTION: demand for transportation between Zones

DESCRIPTION: a MatrixElement is a specialisation of a Link. It describes the number of things (Vessels, Units) that move between Zones (potential, observed, computed) depending on each Zone's Factorobjects. The information can also be a passenger / freight matrix or a modal split matrix between Terminators which identify a Zone

5.3.3.1 Inherits

Link

5.3.3.2 Relations

GTFAssociation --->> GTFMatrixElement

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFMatrix <>-->> GTFMatrixElement

5.3.3.3 Protected Members

bool _home_based

Switch to qualify this Flow as

CODE LIST:

1. A not home based flow =false

2. A home-based flow = true

DEFAULT: 1

Value _trip

Private trip information or transport / transshipment information. A Trip object can contain the complete information of transport chains.

Trips are directed sequences of actions, e.g. first go to Hamburg for a business meeting then (while in Hamburg) visits City, then go to Paris for vacation. In this way descriptions of mobility can be defined. For example, the mobility of different social groups: a businessman has a different mobility profile than say, a single or a housewife or a worker. Workers' mobility could be described as the sequence of actions: 'waking up', 'driving to work', 'work' and 'driving home after work'. For a

single the sequence could be: ‘waking up’, ‘go get something for breakfast’, ‘taking the subway to work’, ‘work’, ‘take subway to shopping mall’, ‘buy food’, ‘go home’. The action of taking the subway would be coded using Trip objects. For each action a new Trip object must be used and then Chained together using the inherited sub-/super relation.

Trips are pre-defined ways through the network.

6 MODE

6.1 Classdiagram

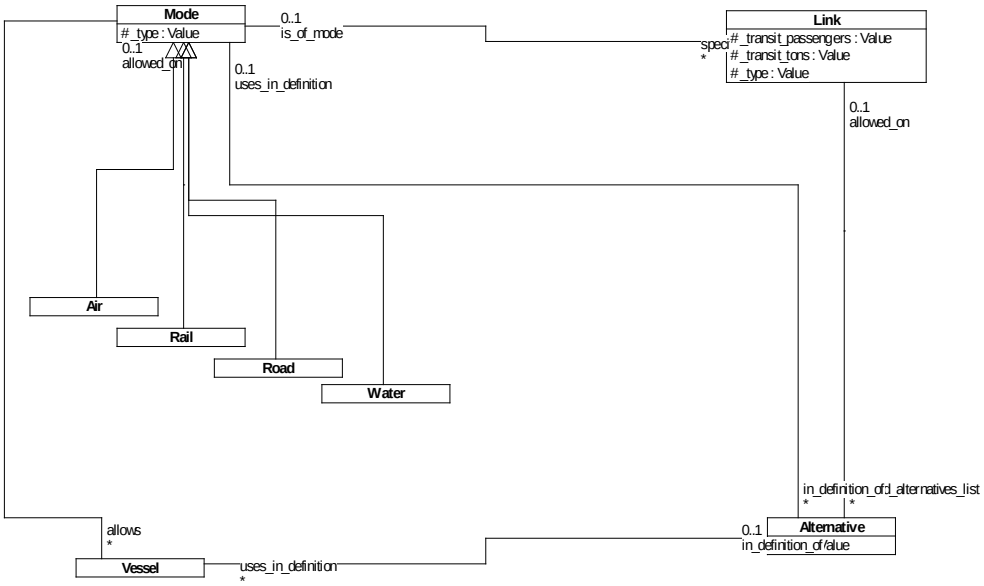


Figure 7: *Mode diagram*

6.2 Parent class

6.2.1 Mode

A Mode is the type of immobile infrastructure used by Vessels for the transportation of Units from Zone to Zone or between Junctions.

To add a level of semantics one of the child classes should be used.

6.2.1.1 Inherits

GTFAssociation

6.2.1.2 Relations

GTFDB <>-->> Mode

GTFAssociation --->> Mode

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

uses_in_definition(Mode) --->> in_definition_of(Alternative)

Alternatives can be defined also using Mode.

is_of_mode(Mode) --->> specifies_mode(Link)

The associated Mode objects define type of immobile infrastructure of the Link.

allowed_on(Mode) --->> allows(Vessel)

6.2.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification RTTI of object types is not available

6.3 Child classes

6.3.1 Air

The ‘immobile’ infrastructure for Airplanes is Air lane, i.e. ‘Air’.

6.3.1.1 Inherits

Mode

6.3.1.2 Relations

GTFAssociation --->> Air

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

6.3.2 Rail

The immobile infrastructure for Wagons is ‘Rail‘.

6.3.2.1 Inherits

Mode

6.3.2.2 Relations

GTFAssociation --->> Rail

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

6.3.3 Road

The immobile infrastructure for Cars is the ‘Road‘.

6.3.3.1 Inherits

Mode

6.3.3.2 Relations

GTFAssociation --->> Road

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

6.3.4 Water

The ‘immobile’ infrastructure for Ships is ‘Water‘.

6.3.4.1 Inherits

Mode

6.3.4.2 Relations

GTFAssociation --->> Water

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7 VESSEL

7.1 Classdiagram

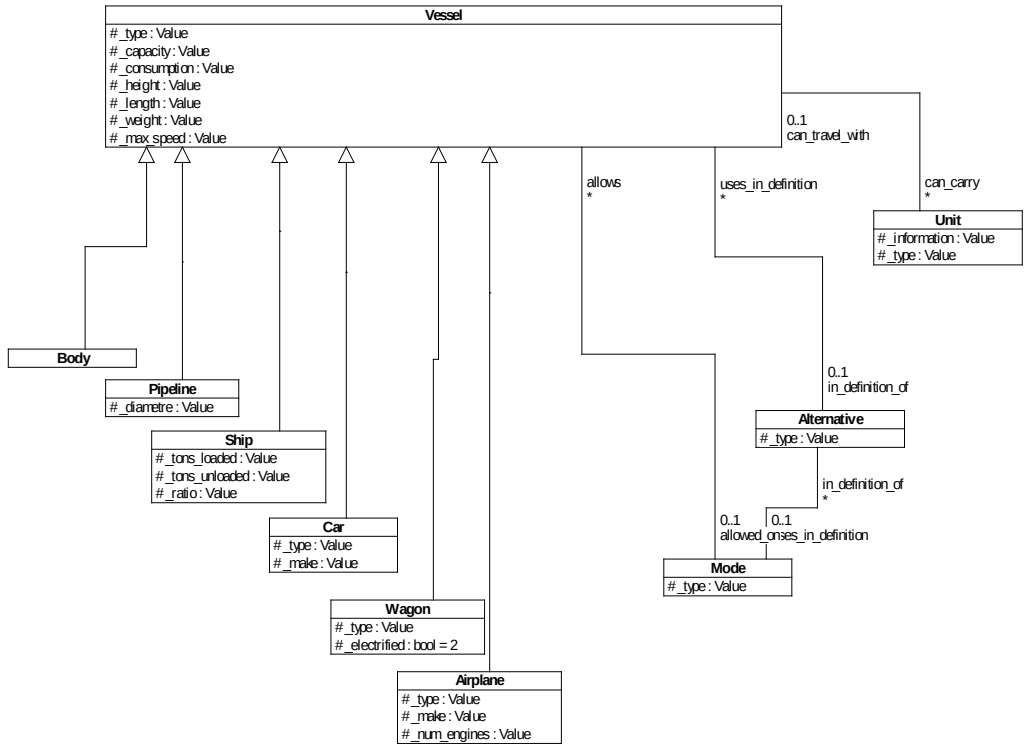


Figure 8: *Vessel diagram*

7.2 Parent class

7.2.1 Vessel

This is the abstraction of everything that increases the movement-count on any Link. In transportation models typical Vessels are cars, trains, aeroplanes, trucks etc. A term had to be found that is general enough to encompass e.g. ‘rolling stock’used for rail and ‘Coach’ for road transport.

DEFINITION: something using (travelling on) Links

FUNCTION: to model / ‘simulate’ a container of information that travels / moves on Links

DESCRIPTION: a Vessel is a logical view of entities that can use Links to travel / transport some Person / Good from one point (Node) to another. A Vessel

description contains all that characterises a Vessel object.

A Vessel is anything that moves and uses infrastructure classes, e.g. cars, planes, persons that use roads, rails, airways etc. Instances of Vessel are types of vessels / vehicles, e.g. an instance of Vessel - road - car is a Mercedes C and not a Mercedes C with the number plate XYZ of Mr Smith. There is also the possibility of a virtual Vessel like a human that uses the mode / vessel ‘walking’. For example for the access / egress points (where say the car is parked) of a railway station or airport to the points where one actually enters the railway station or airport, one has to walk. Thus one is using the mode / vessel ‘walking’.

7.2.1.1 Inherits

GTFAssociation

Abstract class from which all other classes are derived from

7.2.1.2 Relations

GTFAssociation --->> Vessel

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

in_definition_of(Alternative) --->> uses_in_definition(Vessel)

A relationship contributing to the definition of (modelling) Vessel (vehicles, modes) specifications in choice-alternative structures of a model.

used_by(Service) --->> uses_vessels_list(Vessel)

A relationship associating the Vessels used (as carriers) by a Service

allowed_on(Mode) --->> allows(Vessel)

GTFDB <>-->> Vessel

can_travel_with(Vessel) --->> can_carry(Unit)

measurement information used in describing Vessels

7.2.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

Value _weight

Weight of the Vessel. The dimension is defined by a Meta object.

Value _height

Height of the Vessel. A Meta object defines the dimension.

Value _length

Length of the Vessel. A Meta object defines the dimension.

Value _capacity

Capacity of the Vessel. The dimension is defined by a Meta object.

Value _consumption

Consumption of the Vessel. The dimension is defined by a Meta object.

Value _speed

Speed of the Vessel. A Meta object defines the dimension.

Value _max_speed

Maximum speed of this Vessel

7.3 Child classes

7.3.1 Airplane

Air Vessel qualifier specifying that the associated objects' information refers to Vessel information for the type air

7.3.1.1 Inherits

Vessel

7.3.1.2 Relations

GTFAssociation --->> Airplane

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7.3.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

Value _make

Make of airplane

Value _num_engines

Number of engines (the dimension of the value is defined by the associated Meta objects)

7.3.2 Body

Body is the Vessel of a human while e.g. walking.

7.3.2.1 Inherits

Vessel

7.3.2.2 Relations

GTFAssociation --->> Body

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7.3.3 Car

Vessel qualifier specifying that the associated objects' information refers to Vessel information for the type Road

7.3.3.1 Inherits

Vessel

7.3.3.2 Relations

GTFAssociation --->> Car

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7.3.3.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

Value _make

Name of car make

7.3.4 Pipeline

Pipelines are the container (= Vessel) of liquids, e.g. Oil, Water etc.

7.3.4.1 Inherits

Vessel

7.3.4.2 Relations

GTFAssociation --->> Pipeline

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7.3.4.3 Protected Members

Value _diametre

The diametre of the pipe.

7.3.5 Ship

Container / qualifier of water Vessel information

7.3.5.1 Inherits

Vessel

7.3.5.2 Relations

GTFAssociation --->> Ship

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7.3.5.3 Protected Members

Value _tons_loaded

Tons loaded at current time

Value _tons_unloaded

Tons unloaded at current time

Value _ratio

Loading factor (ratio ton-km/capacity-km) for goods

7.3.6 Wagon

Vessel qualifier specifying that the associated objects' information refers to Vessel information for the type Rail

7.3.6.1 Inherits

Vessel

7.3.6.2 Relations

GTFAssociation --->> Wagon

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

7.3.6.3 Protected Members

Value _type

CODE LIST:

1. Regional

2. High-speed

999. Other

DEFAULT: 1

bool _electrified

1. False = no

2. True = yes

DEFAULT: 1

8 CHAIN

8.1 Classdiagram

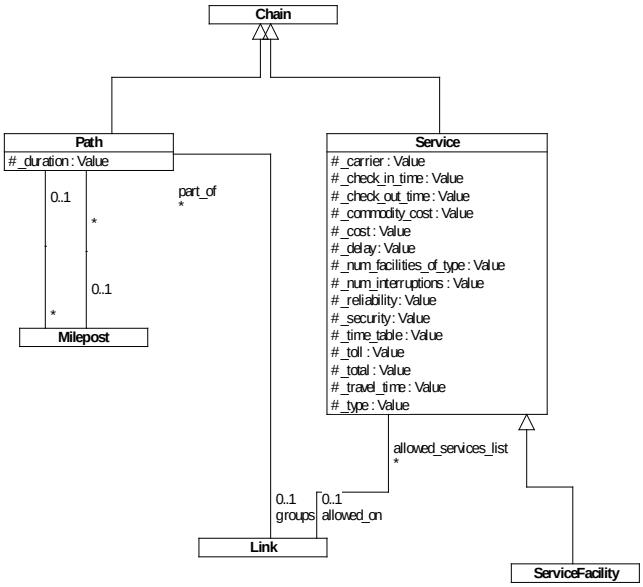


Figure 9: *Chain diagram*

8.2 Parent class

8.2.1 Chain

Super class of Service and Pathrepresenting sequences of objects. To add a level of semantics one of the child classes should be used. This class can be used to describe e.g. ‘Complex Demand’.

8.2.1.1 Inherits

GTFClass

Abstract class from which all other classes are derived from

8.2.1.2 Relations

GTFDB <>-->> Chain

List of Chain objects.

GTFAssociation --->> Chain

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

8.2.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

8.3 Child classes

8.3.1 Service

The class Service is the container for specifications concerning services, carriers etc. that use a Link.

FUNCTION: bundling of assistance to a user (=traveller) for travelling purposes

DESCRIPTION: A service provides a traveller with the means to travel with relevant choices already made in advance by the service operator. The Service class is a container for information pertaining to services, e.g. public transport. This class is a definition of a type of service, the used carrier Vessel(s), the level of security attributed to this type of service and the timetable for the service.

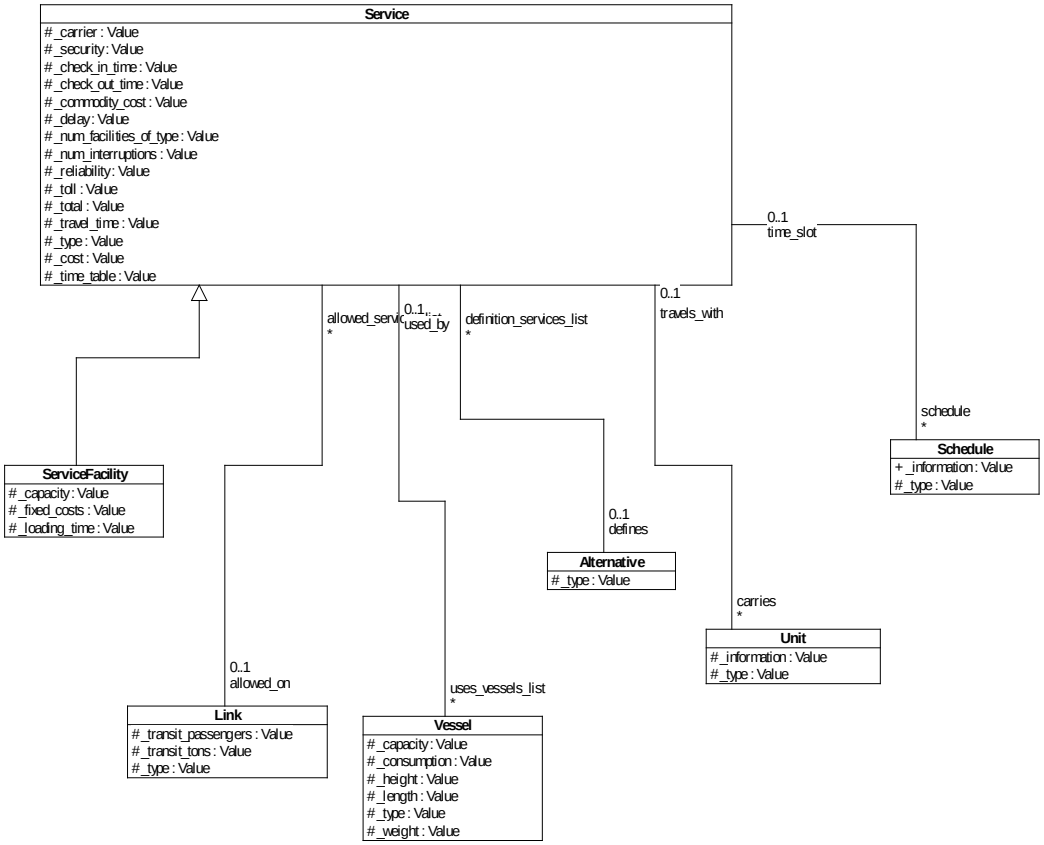


Figure 10: *Service diagram*

8.3.1.1 Inherits

Chain

Abstract class from which all other classes are derived from. It automatically associates all other classes / objects to an aggregation of Comment objects and an aggregation of GIS Shape objects.

8.3.1.2 Relations

allowed_on(Link) --->> allowed_services_list(Service)

A relationship allowing the usage of a Link by a Service

defines(Alternative) --->> definition_services_list(Service)

A relationship contributing to the definition of (modelling) Service specifications in choice-alternative structures of a model.

GTFAssociation --->> Service

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

time_slot(Service) --->> schedule(Schedule)

A relationship associating a time table / time slot (from SCHEDULE) to a Service

travels_with(Service) --->> carries(Unit)

A relationship defining purposes (e.g. business, private, vacation etc.) of the Service

used_by(Service) --->> uses_vessels_list(Vessel)

A relationship associating the Vessels used (as carriers) by a Service

8.3.1.3 Protected Members

Value _carrier

Name of carrier

Value _type

Type of carrier, typically from a standard code list of Service facilities along a Link.

Value _security

Security level

Value _total

A total number of something, e.g. direct flights

Value _reliability

A value defining the level of reliability of the Service

Value _toll

A value defining the tolls to be paid using the Service

Value _num_facilities_of_type

Number of facilities of this type of Service, e.g. 5 cranes available for loading / unloading at a specific port

Value _travel_time

Travel time of the Service on associated Link.

Value _commodity_cost

Cost per ton-km by associated commodity (good)

Value _delay

Delay time when using the Service

Value _num_interruptions

Number of interruptions while using the Service

Value _check_in_time

Check-in time for the Service

Value _check_out_time

Checkout time for the Service

Value _time_table

The Value map contains the Timetable for the service.

Value _cost

Transport price per ton-km of the Service on associated Link.

Segmentation by commodity type must be done using Meta objects.

8.3.2 ServiceFacility

Class containing information about facilities used by services, e.g. a loading / unloading facility at a port. This class holds information e.g. sightseeing points, regeneration points and gas stations.

8.3.2.1 Inherits

Service

8.3.2.2 Relations

GTFAssociation --->> ServiceFacility

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

8.3.2.3 Protected Members

Value _loading_time

Loading time of goods

Value _fixed_costs

Unloading time of goods

Value _capacity

Total capacity of goods

8.3.3 Path

8.3.3.1 Inherits

Chain

8.3.3.2 Relations

groups(Link) --->> part_of(Path)

Milepost --->> Path

A Milepost can be on many Paths objects, i.e. if different Paths use a smae Link with a Milepost on it..

GTFAssociation --->> Path

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

Path --->> Milepost

A single Path can contain many Mileposts.

8.3.3.3 Protected Members

Value _duration

Specifies how long a trip is undertaken . This is not the time it takes to go from A to B, because e.g. for a vacation trip moving from A to B might entail sight seeing etc.

9 DYNAMICSEGMENTATION

9.1 Dynamic Segmentation Classdiagram

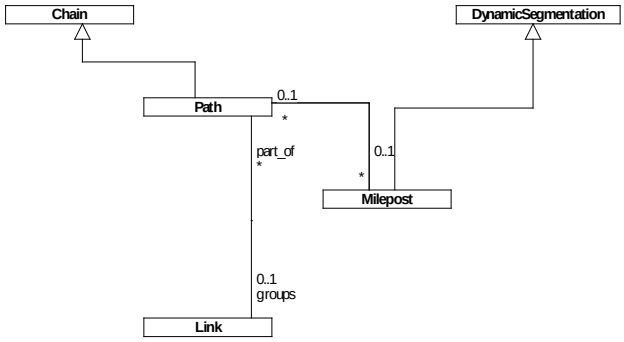


Figure 11: *Dynamic Segmentation diagram*

9.2 Parent class

9.2.1 DynamicSegmentation

Contains information of Milepostss, e.g. their position (distance from starting Node and distance form ending Node) and other data that is attached to a specific point of a Link. Use this class to associate of dynamical segmentation of e.g. attribute values along a Link. Note: Milestones & Dynamic Segmentation were used because an implicit approach of adding this kind of data implies splitting the Link into two at the relevant point, where Attribute values change. This can typically lead very easily to splitting the one Link into 100 or more. Thus many different ‘theme’ data must be incorporated into the data set.

Practically this is a concession to demand modelling and their algorithms, which usually are not designed to handle dynamically segmented information.

9.2.1.1 Inherits

GTFAssociation

9.2.1.2 Relations

GTFDB <>--->> DynamicSegmentation

GTFAssociation --->> DynamicSegmentation

This association is a component part of the implementation of the concept "an object

can be attached to any other object" used by some toplevel classes.

9.2.1.3 Protected Members

Value _distance_from_beginning

Distance in Metas from the beginning milestone / post of a defined section

Value _distance_from_ending

Distance in Metas to the end of the milestone / post of a defined section

9.3 Child classes

9.3.1 Milepost

Mileposts can be used to either describe the really existent mileposts (milestones) or to associate segments on Paths where Attribute object values change.

9.3.1.1 Inherits

DynamicSegmentation

9.3.1.2 Relations

Path --->> Milepost

A single Path can contain many Mileposts.

GTFAssociation --->> Milepost

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

Milepost --->> Path

A Milepost can be on many Paths objects, i.e. if different Paths use a smae Link with a Milepost on it..

10 ALTERNATIVE

10.1 Classdiagram

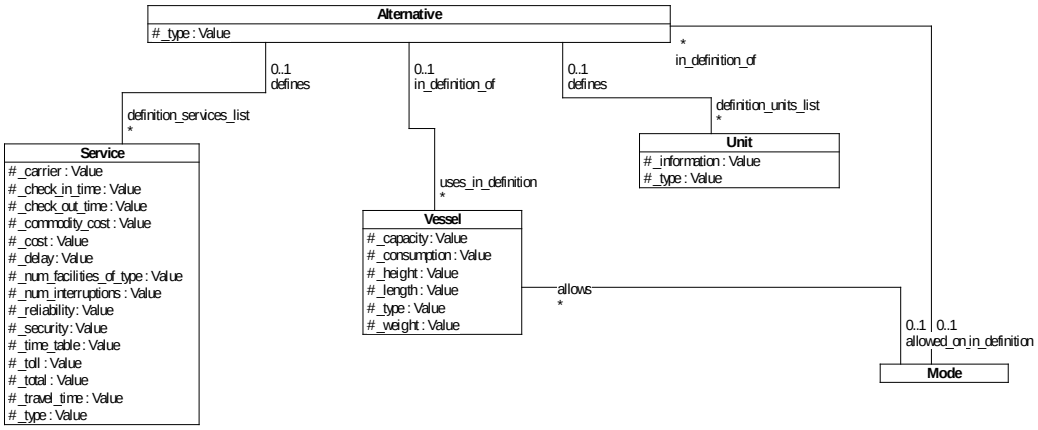


Figure 12: *Alternative diagram*

10.2 Parent classes

10.2.1 Alternative

This class comprises the relationships needed to define sets of options.

DEFINITION: is the container for information pertaining to the definitions of choice alternatives for a model

FUNCTION: a logical carrier of information for choice alternatives of models

DESCRIPTION: Transportation models use choice alternatives to describe the situation individuals (or the behavioural element being represented in the transportation model) face in certain situations. The transportation model then ‘decides’ which option the individual chooses by taking into account different aspects (socio-economic, economic, psychological etc.). Note, that this is one main philosophy used in the implementation of a transportation model.

From a transportation modelling point-of-view the infrastructure networks (i.e. the groupings of Junctions, Links etc. which form a logical whole) need to be distinguished according to different ‘main modes’ (or Alternatives), because models use these ‘main modes’ to differentiate elements of sets of choice alternatives. The instances of the Alternative class define choice options (or choice of combinations of available means of transport) for a model, through the combination of Units

(person/good ...), Services and Vessels. Certain Links can only be used by some of the Alternatives, because e.g. the Link cannot cope with a Vessel of some given tonnage used in the Alternative definition or some other restriction due to the definition in the model, i.e. an Alternative's main mode might not be allowed to use a Link, but otherwise (physically) it is allowed.

10.2.1.1 Inherits

GTFClass

Abstract class from which all other classes are derived from

10.2.1.2 Relations

allowed_on(Link) --->> allowed_alternatives_list(Alternative)

A relationship associating the (model's) choice-alternative structures with a Link

uses_in_definition(Mode) --->> in_definition_of(Alternative)

Alternatives can be defined also using Mode.

GTFDB <>-->> Alternative

GTFAssociation --->> Alternative

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

defines(Alternative) --->> definition_units_list(Unit)

A relationship contributing to the definition of (modelling) Unit (persons, goods) specifications in choice-alternative structures of a model.

defines(Alternative) --->> definition_services_list(Service)

A relationship contributing to the definition of (modelling) Service specifications in choice-alternative structures of a model.

in_definition_of(Alternative) --->> uses_in_definition(Vessel)

A relationship contributing to the definition of (modelling) Vessel (vehicles, modes) specifications in choice-alternative structures of a model.

10.2.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

10.3 Child classes

11 UNIT

11.1 Classdiagram

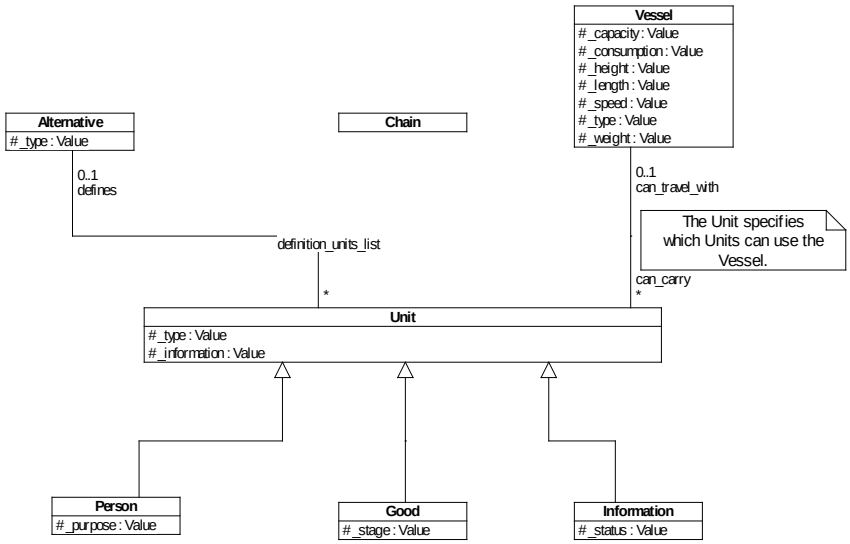


Figure 13: *Unit diagram*

11.2 Parent class

11.2.1 Unit

Units define the type of unit being moved or transported. This class contains all such information and associates this information with Alternatives and LinkAttributes etc.

Since it is derived from GTFAssociation, instances of this class can be attached to instances of any other object.

To add a level of semantics one of the child classes should be used.

11.2.1.1 Inherits

GTFAssociation

Abstract class from which all other classes are derived from

11.2.1.2 Relations

travels_with(Service) --->> carries(Unit)

A relationship defining purposes (e.g. business, private, vacation etc.) of the Service

defines(Alternative) --->> definition_units_list(Unit)

A relationship contributing to the definition of (modelling) Unit (persons, goods) specifications in choice-alternative structures of a model.

can_travel_with(Vessel) --->> can_carry(Unit)

measurement information used in describing Vessels

GTFDB <>-->> Unit

11.2.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification RTTI of object types is not available

Value _information

The actual data

11.3 Child classes

11.3.1 Good

Represents freight objects

11.3.1.1 Inherits

Unit

11.3.1.2 Relations

GTFAssociation --->> Good

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

11.3.1.3 Protected Members

Value _stage

The stage of a Good during the entire movement from A to B.

The stage member must be taken from the following

CODE LIST:

1. Waiting

2. Being loaded

3. Being unloaded

4. Moving

11.3.2 Information

The object of movement can be ‘information’.

11.3.2.1 Inherits

Unit

11.3.2.2 Relations

GTFAssociation --->> Information

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

11.3.2.3 Protected Members

Value _status

The status of the information being moved.

CODE LIST:

1. Original

2. Read

3. Changed

4. Marked for deletion

11.3.3 Person

Represents persons in movement.

11.3.3.1 Inherits

Unit

11.3.3.2 Relations

GTFAssociation --->> Person

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

11.3.3.3 Protected Members

Value _purpose

The purpose of a Person during the entire movement from A to B.

The purpose member must be taken from the following

CODE LIST:

1. Business

2. Private

3. Vacation

4. Shopping

12 META

12.1 Classdiagram

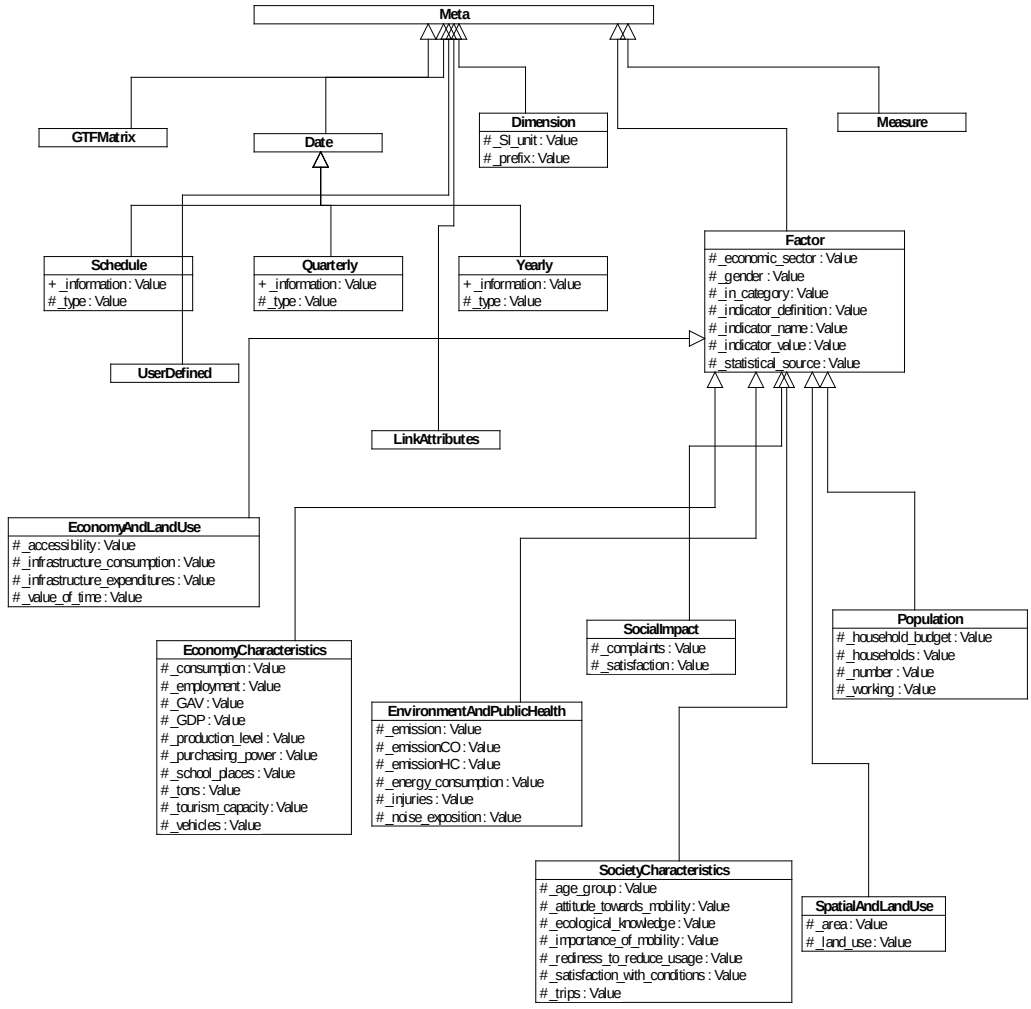


Figure 14: *Meta diagram*

12.2 Parent class

12.2.1 Meta

Summary of all meta-data information, e.g. dimensions, units of measurement etc. Meta-data objects are objects to define meta-information which do not pertain to modelling or network or other information, but are rather complementary information describing units of measurements etc. container of dimension meta-

information for other objects' values, e.g. km^2

To add a level of semantics one of the child classes should be used.

12.2.1.1 Inherits

Attribute

Abstract class from which all other classes are derived from

12.2.1.2 Relations

GTFDB <>--->> Meta

GTFAssociation --->> Meta

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.2.1.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

12.3 Child classes

12.3.1 GTFMatrix

This class uses MatrixElements to describe a complete matrix of values.

12.3.1.1 Inherits

Meta

12.3.1.2 Relations

GTFAssociation --->> GTFMatrix

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFMatrix <>--->> GTFMatrixElement

12.3.2 Date

A container of date objects qualifying other objects' information to be of a certain

time period

12.3.2.1 Inherits

Meta

12.3.2.2 Relations

GTFAssociation --->> Date

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.3 Quarterly

Quarterly time period information or just a qualifier of the associated objects specifying that the associated objects are relevant for a specific quarterly time period

12.3.3.1 Inherits

Date

12.3.3.2 Relations

GTFAssociation --->> Quarterly

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.3.3 Public Members

Value _information

List of quarterly time period information or just qualifier

12.3.3.4 Protected Members

Value _type

ENUM = 1 | 2 | 3 | 4 | 5

1. 1st quarter
2. 2nd quarter
3. 3rd quarter
4. 4th quarter
5. Quarterly

DEFAULT: 5

12.3.4 Schedule

Container of scheduled information for other objects

12.3.4.1 Inherits

Date

12.3.4.2 Relations

time_slot(Service) --->> schedule(Schedule)

A relationship associating a time table / time slot (from SCHEDULE) to a Service

GTFAssociation --->> Schedule

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.4.3 Public Members

Value _information

Schedule information or just qualifier

12.3.4.4 Protected Members

Value _type

ENUM = time_table | slot

1. Time_table
2. Slot

DEFAULT: 1

12.3.5 Yearly

Container of yearly information for other objects

12.3.5.1 Inherits

Date

12.3.5.2 Relations

GTFAssociation --->> Yearly

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.5.3 Public Members

Value _information

List of yearly time period information or just qualifier

12.3.5.4 Protected Members

Value _type

ENUM = Yearly | Specific_Year

The specific year is specified in the ‘information’attribute.

1. Yearly

2. Specific_Year

DEFAULT: 1

12.3.6 Dimension

Dimension specification of associated data. The dimension is specified using SI units and all mathematical operators.

12.3.6.1 Inherits

Meta

12.3.6.2 Relations

GTFAssociation --->> Dimension

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.6.3 Protected Members

Value _SI_unit

CODE LIST

1. Length metre [m]
2. Mass kilogram [kg]
3. Time second [s]
4. Electric current ampere [A]
5. Thermodynamic temperature kelvin [K]
6. Amount of substance mole [mol]
7. Luminous intensity candela [cd]

SI Derived Units:

Frequency hertz: Hz = 1/s
Force newton: N = m kg/s2
Pressure, stress pascal: Pa = N/m2 = kg/m s2
Energy, work, quantity of heat joule: J = N m = m2 kg/s2
Power, radiant flux watt: W = J/s = m2 kg/s3
Quantity of electricity, electric charge coulomb: C = s A
Electric potential volt: V = W/A = m2 kg/s3 A
Capacitance farad: F = C/V = s4 A2/m2 kg
Electric resistance ohm: Omega = V/A = m2 kg/s3 A2
Conductance siemens: S = A/V = s3 A2/m2 kg
Magnetic flux weber: Wb = V s = m2 kg/s2 A
Magnetic flux density, magnetic induction tesla: T = Wb/m2 = kg/s2 A
Inductance henry: H = Wb/A = m2 kg/s2 A2
Luminous flux lumen: lm = cd sr
Illuminance lux: lx = lm/m2 = cd sr/m2
Activity (ionizing radiations) Becquerel: Bq = 1/s
Absorbed dose gray: Gy = J/kg = m2/s2
Dynamic viscosity pascal second: Pa s = kg/m s
Moment of force meter newton: N m = m2 kg/s2
Surface tension newton per meter: N/m = kg/s2
Heat flux density, irradiance watt per square meter: W/m2 = kg/s3
Heat capacity, entropy joule per kelvin: J/K = m2 kg/s2 K
Specific heat capacity, specific entropy: J/kg K = m2/s2 K
Specific energy joule per kilogram: J/kg = m2/s2
Thermal conductivity watt per meter kelvin: W/m K = m kg/s3 K
Energy density joule per cubic meter: J/m3 = kg/m s2
Electric field strength volt per meter: V/m = m kg/s3 A
Electric charge density coulomb per cubic meter: C/m3 = s A/m3
Electric displacement, electric flux density coulomb per square meter: C/m2 = s A/m2
Permittivity farad per meter: F/m = s4 A2/m3 kg
Permeability henry per meter: H/m = m kg/s2 A2
Molar energy joule per mole: J/mol = m2 kg/s2 mol
Molar entropy, molar heat capacity joule per mole kelvin: J/mol K = m2 kg/s2 K mol
Exposure (ionizing radiations) coulomb per kilogram: C/kg = s A/kg
Absorbed dose rate gray per second: Gy/s = m2/s3

Value _prefix

SI unit prefix CODE LIST exponent (base 10) of decimal numbers: E n = 10^n

1. E 18 exa E
2. E 15 peta P
3. E 12 tera T
4. E 9 giga G
5. E 6 mega M
7. E 3 kilo k
8. E 2 hecto h
9. E 1 deca da
10. E -1 deci d
11. E -2 centi c
12. E -3 milli m
13. E -6 micro mu

14. E -9 nano n
15. E-12 pico p
16. E-15 femto f
17. E-18 atto a

12.3.7 Measure

These can be used to add ‘measure/dimension’ information that are not directly Physical dimensions (see the Dimension class). Rather these represent transport or traffic measures, e.g. average daily traffic.

12.3.7.1 Inherits

Meta

12.3.7.2 Relations

GTFAssociation --->> Measure

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.8 LinkAttributes

These are characteristics, which can be attributed to Links. These specifications are regulatory / administrative or defined by engineering science.

The reason why this kind of information is not modelled in the GTF-CM as member attributes of the Link class is that this class is the result of a ternary relationship between Link and other classes. As this association itself also contains attributes it cannot be modelled as an attribute to a class. In OO, the attributes of such ternary relationships are modelled as an own class called ‘objectified attributes’ or ‘association class’.

DEFINITION: a bundling of Link characteristics

FUNCTION: this class associates all the technical, statistical and movement specifications that come from Zone, Vessel and Unit and defines (physical) characteristics of a Link

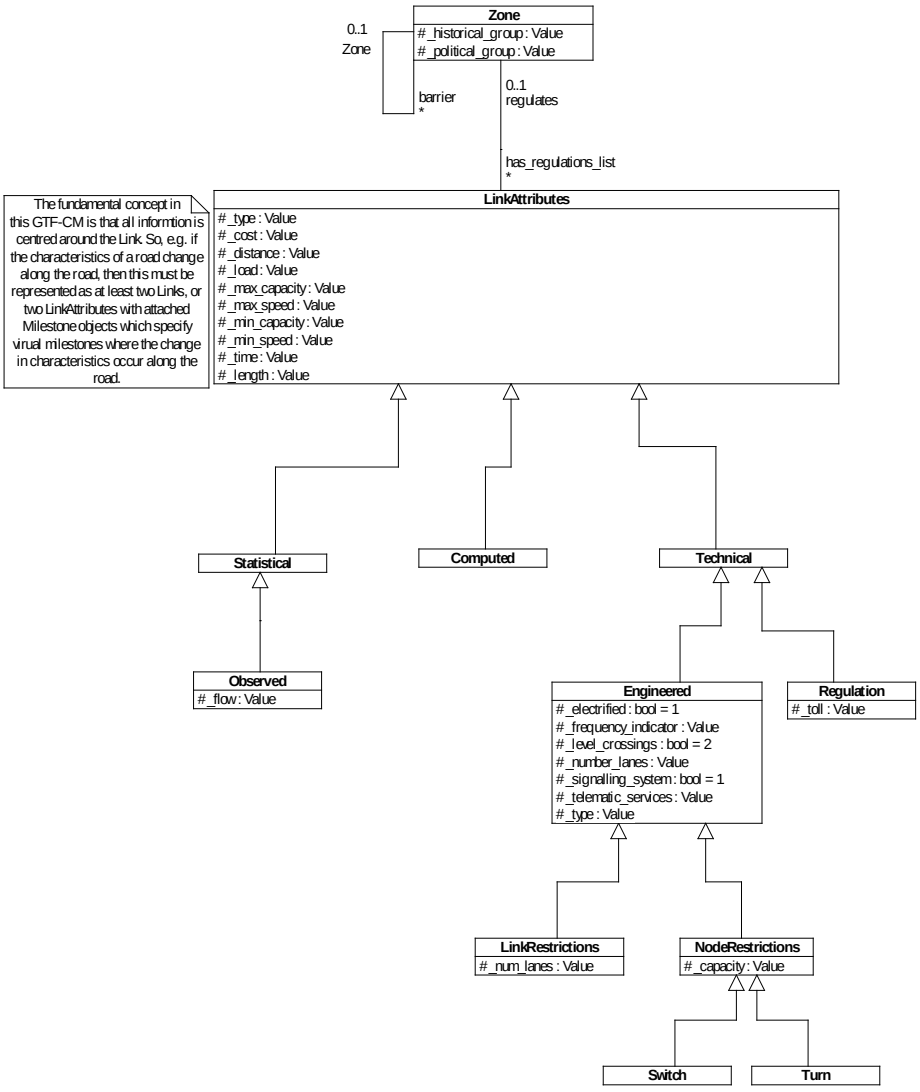


Figure 15: *LinkAttributes diagram*

12.3.8.1 Inherits

Meta

Abstract class from which all other classes are derived from

12.3.8.2 Relations

regulates(Zone) --->> has_regulations_list(LinkAttributes)

A relationship giving administrative regulations for a Link based on a Zone's, e.g. country

GTFAssociation --->> LinkAttributes

This association is a component part of the implementation of the concept "an object

can be attached to any other object" used by some toplevel classes.

12.3.8.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

Value _cost

Cost on associated Link. The dimension e.g. $, DM must be specified in an associated Meta object.

Segmentation by vehicle type must be done using Meta objects.

Value _distance

Distance of associated Link.

Value _load

Load on associated Link.

The value table of the _load member must contain an entry ‘total’.

The value table can contain an entry ‘proportion interregional traffic’.

The value table can contain an entry ‘proportion truck traffic’.

Segmentation by type of day must be done using Meta objects.

Value _max_capacity

Computed maximum capacity

Value _max_speed

Computed maximum speed

Value _min_capacity

Computed minimum capacity

Value _min_speed

Computed minimum speed

Value _time

Travel time on associated Link.

Can be replaced by an association to a Meta object (Meta-Distance).

Segmentation by vehicle type must be done using Meta objects.

Value _length

Length of associated Link. Must contain the transportation model relevant distance.

12.3.9 Computed

Contains all the relevant data regarding movements across Zone boundaries that are computed, not observed. Computed model results are located in instances of this class or in the instances of the derived objects

12.3.9.1 Inherits

LinkAttributes

12.3.9.2 Relations

GTFAssociation --->> Computed

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.9.3 Protected Members

Value _flow

Current computed movement in Units

12.3.10Statistical

Container of statistical information

12.3.10.1 Inherits

LinkAttributes

12.3.10.2 Relations

GTFAssociation --->> Statistical

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.11Observed

Contains e.g. surveys, flow counts etc. i.e. real observed flows.

12.3.11.1 Inherits

Statistical

12.3.11.2 Relations

GTFAssociation --->> Observed

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.11.3 Protected Members

Value _flow

Current computed flow in Units, Metas. The _flow Value-map can contain a matrix of values.

12.3.12Technical

Container of technical information, the _value table can contain a matrix.

12.3.12.1 Inherits

LinkAttributes

12.3.12.2 Relations

GTFAssociation --->> Technical

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.13Engineered

Contains all the engineering relevant data that describes a Link from an engineering point of view

12.3.13.1 Inherits

Technical

12.3.13.2 Relations

GTFAssociation --->> Engineered

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.13.3 Protected Members

Value _type

Left for application programmers if runtime type identification of object types is not available

Value _frequency_indicator

Allowed frequency of the associated Vessel type on this Link

Value _number_lanes

Number of lanes of associated Link

bool _electrified

CODE LIST:

1. Yes = y

2. No = n

DEFAULT: 1

bool _signalling_system

CODE LIST:

1. Yes = y

2. No = n

DEFAULT: 1

bool _level_crossings

CODE LIST:

1. Yes = y

2. No = n

DEFAULT: 1

Value _telematic_services

Telematic services available on the associated Link.

Segmentation by type must be done using Meta objects.

12.3.14Regulation

The Regulation class contains all the administrative / regulatory data that concerns a Link.

12.3.14.1 Inherits

Technical

12.3.14.2 Relations

GTFAssociation --->> Regulation

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.14.3 Protected Members

Value _toll

Toll according to administrative regulation on associated Link.

12.3.15LinkRestrictions

Container of technical / engineered restriction information pertaining to Links

For example, minimal height in tunnels, maximal allowed weight of trucks.

12.3.15.1 Inherits

Engineered

12.3.15.2 Relations

GTFAssociation --->> LinkRestrictions

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.15.3 Protected Members

Value _num_lanes

Number of lanes of the Link (or number of runways if Link is associated to Air mode)

12.3.16NodeRestrictions

This class represents restrictions e.g. an infrastructure allowed (signs) administrative ‘turn’. The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

12.3.16.1 Inherits

Engineered

12.3.16.2 Relations

turn_at(Junction) <>--> turning_at(NodeRestrictions)

The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

turn_from(Junction) <>--> turning_from(NodeRestrictions)

The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

turn_to(Junction) <>--> turning_to(NodeRestrictions)

The ‘turn’ restriction relationship defines which Node has this turning restriction, i.e. a turn from the ‘from’ Node to the ‘to’ Node at the ‘at’ Node.

GTFAssociation --->> NodeRestrictions

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.16.3 Protected Members

Value _capacity

Handling capacity of persons, cargo or goods. (E.g. pertains to Airport Terminal capacity if Junction is associated to Air mode.)

12.3.17Turn

A turning restriction.

12.3.17.1 Inherits

NodeRestrictions

12.3.17.2 Relations

GTFAssociation --->> Turn

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.17.3 Protected Members

Value _type

Enumeration:

1. left turn 2. right turn 3. number of lanes 4. signal

12.3.18Switch

12.3.18.1 Inherits

NodeRestrictions

12.3.18.2 Relations

GTFAssociation --->> Switch

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

12.3.18.3 Protected Members

Value _direction

Value _radius

Value _speed

Value _primary

Enumeration:

1. Yes 2. No, secondary

12.3.19UserDefined

Special class, so that the mapping to XML for user defined attributes can be done consistent to this conceptual model.

12.3.19.1 Inherits

Meta

12.3.19.2 Relations

GTFAssociation --->> UserDefined

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13 GROUP

13.1 Group Class Diagram

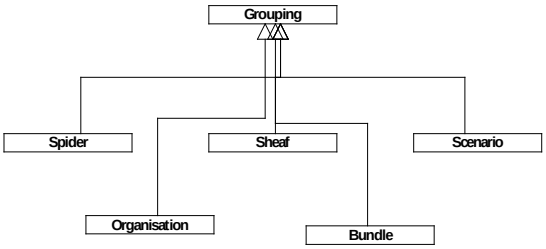


Figure 16: *Group diagram*

13.2 Parent class

13.2.1 Grouping

This class can be used to group business logic objects in order to define 'result sets'.

This class is not like the others in the toplevel. It is simply for grouping purposes. The other toplevels contain business logic, e.g. topologic information etc.

To add a level of semantics one of the child classes should be used.

13.2.1.1 Inherits

GTFAssociation

13.2.1.2 Relations

GTFDB <>--->> Grouping

GTFAssociation --->> Grouping

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.2.1.3 Protected Members

Value _type

Enumeration:

1. Sheaf 2. Corridor 3. Bundle 4. Spider 5. Scenario 6. Organisation

13.3 Child classes

13.3.1 Scenario

This grouping denotes the objects that form a specific case from a scenario. The scenario should be described in attached Comment objects.

13.3.1.1 Inherits

Grouping

13.3.1.2 Relations

GTFAssociation --->> Scenario

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.3.2 Organisation

This class denotes a grouping of objects because they are part of an organisation, e.g. a bus line of an operator.

13.3.2.1 Inherits

Grouping

13.3.2.2 Relations

GTFAssociation --->> Organisation

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.3.3 Bundle

This class denotes the grouped objects as a bundle. It should consist of a set of e.g. destination Zones for each Bundle.

13.3.3.1 Inherits

Grouping

13.3.3.2 Relations

GTFAssociation --->> Bundle

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.3.4 Sheaf

This class denotes the grouped objects being a Sheaf. It should consist of the necessary Nodes, Links and further Classes/Information determining e.g. the flows' starting Zone (or point), the ending Zone (or point) and the used Links with the constraint that these Links all have the same staring Junction and ending Junction.

13.3.4.1 Inherits

Grouping

13.3.4.2 Relations

GTFAssociation --->> Sheaf

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.3.5 Spider

This class denotes the grouped objects as a Spider. It should consist of a (central) Zone, and flows entering or leaving it.

13.3.5.1 Inherits

Grouping

13.3.5.2 Relations

GTFAssociation --->> Spider

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.3.6 Catchment

This class denotes the grouped objects as a Catchment Area. It should consist of all

Zones in one Catchment Area.

13.3.6.1 Inherits

Grouping

13.3.6.2 Relations

GTFAssociation --->> Catchment

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

13.3.7 Corridor

A Corridor is a grouping of all the elements along a specific area.

13.3.7.1 Inherits

Grouping

13.3.7.2 Relations

GTFAssociation --->> Corridor

14 FRAMEWORK

14.1 Framework Class Diagram

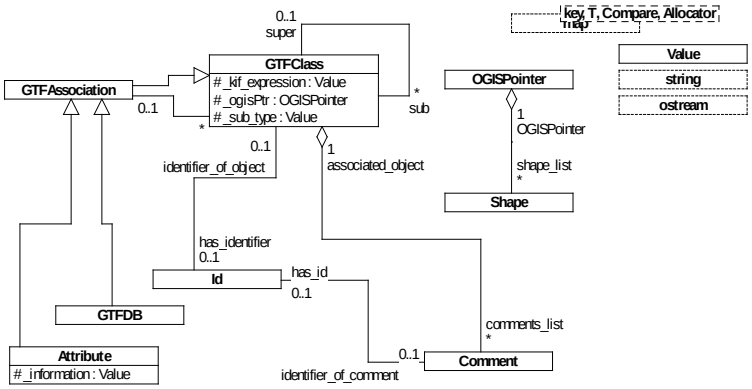


Figure 17: *Diagram: Framework classes*

14.2 Classes

These are utility classes that simplify things in the definition of the classes in the conceptual model in the business logic layer.

14.2.1 Attribute

A class to allow to attach ‘user defined’ attributes arbitrarily to all other business logic. The user can specify the attributes name. The value of the attribute is kept in the Value member attribute of this Class.

14.2.1.1 Inherits

GTFAssociation

Any Attribute instances can be attached to any business logic class instance.

14.2.1.2 Protected Members

Value _information

The actual data

14.2.2 Comment

A container of textual comments, which can be associated to any other object (class instance).

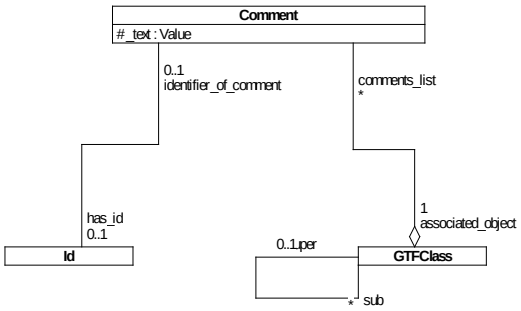


Figure 18: *Comment diagram*

14.2.2.1 Relations

associated_object(GTFClass) <>--->> comments_list(Comment)

List of comments for this object. The comments can be referenced by Id.

identifier_of_comment(Comment) ---> has_id(Id)

An Id object containing a unique identifier value for each object of this class

14.2.2.2 Protected Members

Value _text

Textual comment

14.2.3 GTFAssociation

Objects from this special Class automatically have 0..* associations to all other objects in the GTF-CM.

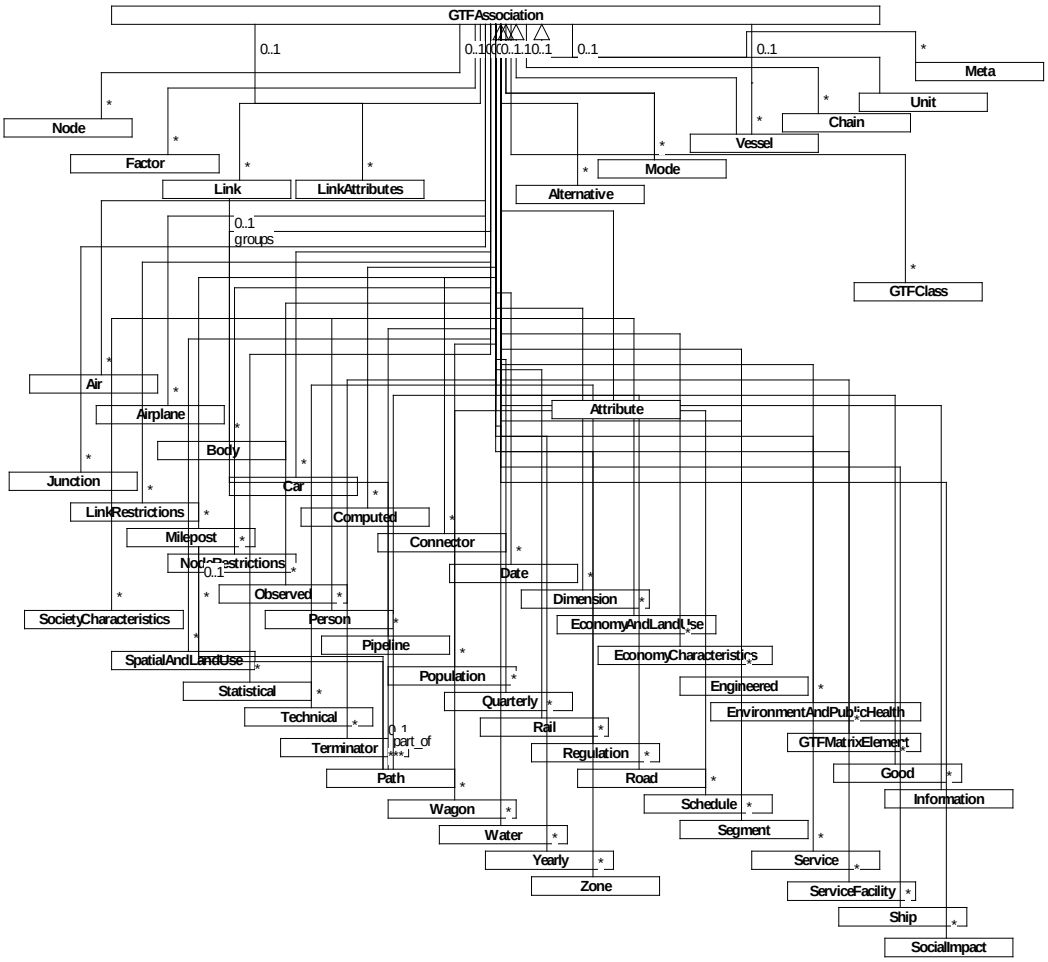


Figure 19: *GTFAssociation diagram*

14.2.3.1 Inherits

GTFClass

Abstract class from which all other classes are derived from

14.2.3.2 Relations

GTFAssociation --->> LinkAttributes

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Vessel

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Body

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Air

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Airplane

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Car

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Connector

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Computed

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Date

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Dimension

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> EconomyAndLandUse

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> EconomyCharacteristics

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Engineered

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> EnvironmentAndPublicHealth

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> GTFMatrixElement

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Good

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Information

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> LinkRestrictions

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> NodeRestrictions

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Observed

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Person

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Pipeline

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Quarterly

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Rail

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Regulation

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Road

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Schedule

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Segment

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Service

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> ServiceFacility

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Ship

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> SocialImpact

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> SocietyCharacteristics

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> SpatialAndLandUse

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Statistical

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Technical

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Terminator

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Wagon

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Water

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Yearly

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Zone

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> GTFClass

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Factor

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Link

GTFAssociation --->> Alternative

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Meta

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Bundle

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Catchment

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Chain

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> DynamicSegmentation

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Grouping

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> GTFMatrix

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Junction

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Measure

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Milepost

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Mode

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Node

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Organisation

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Path

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Population

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Scenario

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Sheaf

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Spider

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Switch

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Turn

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> UserDefined

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

GTFAssociation --->> Corridor

14.2.4 GTFDB

Topmost object which is just a facade in front of all the other classes. Could be implemented as the ‘Facade’ Pattern, see [GOF].

It is also used as the single top most object containing all others objects in a GTF data set of a transmission.

Note: To be absolutely correct, the associations of this class to the business logic Classes should all be 1 ... * instead of 0 ... *.

14.2.4.1 Inherits

GTFAssociation

GTFDB only needs to inherit all the aggregations Associations from the GTFAssociation Class so that it can be used as main information node (i.e. that can access all objects) in the GTF-CM.

14.2.4.2 Relations

GTFDB <>-->> Mode

GTFDB <>-->> Node

List of all Node objects

GTFDB <>-->> Link

List of all Link objects

GTFDB <>-->> DynamicSegmentation

GTFDB <>-->> Vessel

GTFDB <>-->> Chain

List of Chain objects.

GTFDB <>-->> Alternative

GTFDB <>-->> Unit

GTFDB <>-->> Meta

GTFDB <>-->> Grouping

14.2.5 GTFClass

Abstract class from which all other classes are derived.

Any GTFObject is a container, which can hold a number of any other class (i.e. class instances), so that one can define that, e.g. a Junction has some internal structure. This is done by defining a Junction and associated objects composing it through the super-/sub association.

DEFINITION: definition of some internal structure for a Junction or to group parts of an infrastructure mode into an aggregation

FUNCTION: container of groups of classes of the conceptual model

DESCRIPTION: The super-/sub association of this class is used in this model to, e.g. define different parts of an infrastructure network with different information. The objects associated through the super-/sub association represent e.g. its internal network structure, e.g. an airport Junction can be zoomed up to show the different access and egress points and terminals and their connections.

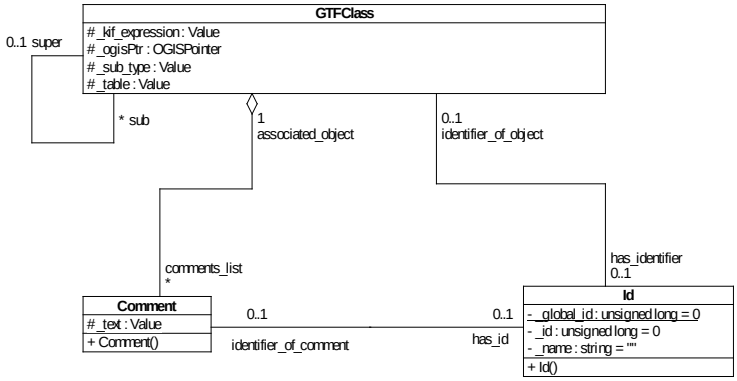


Figure 20: *GTFObject diagram*

14.2.5.1 Relations

GTFAssociation --->> GTFClass

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

associated_object(GTFClass) <>-->> comments_list(Comment)

List of comments for this object. The comments can be referenced by Id.

identifier_of_object(GTFClass) ---> has_identifier(Id)

An Id object containing a unique identifier value for each object it is part of. E.g. for a Junction object, since all Junction objects inherit from GTFObject they all automatically have unique identifiers since each Id object is unique by definition

super(GTFClass) --->> sub(GTFClass)

Structure of a GTFObject. A (every) GTFObject can be part of another GTFObject (which itself can contain more GTFObjects) or can be a container of other GTFObject itself. Thus, an explicit hierarchical structure of GTFObjects is possible. For example, railway stations, airports or even waterway locks can be described using the Junction/Link concepts as a hierarchy. This principle must be used to describe, ‘transport chains’.

14.2.5.2 Protected Members

OGISPointer _ogisPtr

The value of this member must be a valid ID of an external OpenGIS Feature catalogue.

Value _kif_expression

An ASCII Text representing a KIF expression. The length of the ASCII text for the expression is not limited, that is, this text can represent even a complete KIF knowledge base (i.e. many KIF expressions).

The object names in the KIF expressions must match the Class/Relationships/Members names from this conceptual model to be valid expressions.

Example of a KIF expression: (Junction London 3578). This would define a Junction named London with the identifier 3578.

Value _sub_type

Type of the grouping of objects associated through the sub association.

1. sub network, detailing of internal structure 2. dynamic segmentation 3. catchment area 4. complex demand 5. corridor

Value _table

A textual pointer to the table of a DB.

string _RTTIName

Name of the objects Class (Type) for use in RTTI (Run Time Type Identification). This gets set by each Class's Constructor.

int _RTTINumber

Number of the objects Class (Type) for use in RTTI (Run Time Type Identification). This gets set by each Class's Constructor.

14.2.6 Id

Unique numerical (id) for all instances of a class.

It is constructed in the following form (uniqueness constraint):

'<Date>#<Time>#<geographical location longitude>#<geographical location latitude>'

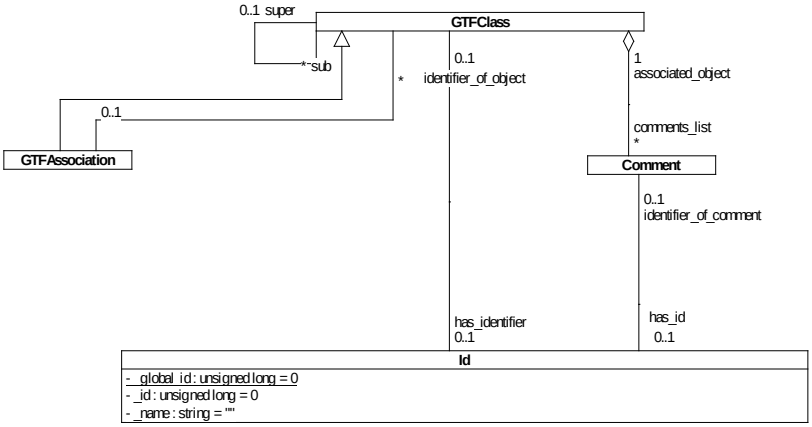


Figure 21: *Id diagram*

14.2.6.1 Relations

identifier_of_comment(Comment) ---> has_id(Id)

An Id object containing a unique identifier value for each object of this class

identifier_of_object(GTFClass) ---> has_identifier(Id)

An Id object containing a unique identifier value for each object it is part of. E.g. for a Junction object, since all Junction objects inherit from GTFObject they all automatically have unique identifiers since each Id object is unique by definition

14.2.6.2 Private Members

string _name

Name of GTF Class instance. The name is not a unique identifier, but can be used for external identification purposes, e.g. the Id.name attribute of a Junction object is 'Hamburg'. There might be several Junction objects called 'Hamburg', but they all have unique identifiers and not unique names.

static unsigned long _global_id

Internal id gets set (and incremented by 1) automatically for each new class instance (i.e. object) This is the class global id, i.e. it holds the total number of Id objects created

unsigned long _id

Identification number for the specific object this Id object is part of

14.2.7 OGISPointer

A pointer to an OpenGIS object (Feature, Relationship etc.). See OpenGIS specification.

A pointer that can be used to hold identifiers of OpenGIS classes external to data sets based on this conceptual model. For simplification purposes and not to force users of GTF to have to implement at least the most important parts of the OpenGIS abstract specification, OGISPointer objects are associated to generic Shape objects, which hold a list of unspecified length of x/y/z co-ordinate data that can be used alternatively to specify for example, polygons or other graphical visualisations.

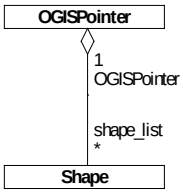


Figure 22: *OGISPointer diagram*

14.2.7.1 Relations

OGISPointer <>--->> shape_list(Shape)

List of shape objects defining the GIS view of the GTFObject

14.2.7.2 Protected Members

Value _value

The value of this member must be a valid identifier of a class in an external OpenGIS Feature catalogue.

The _value table must contain an entry ‘Catalogue’, naming an OpenGIS Feature catalogue.

The _value table must contain an entry ‘ID’ of a valid identifier from the external OpenGIS Feature catalogue referred to by the ‘Catalogue’ entry.

14.2.8 Shape

Container of shape information for GIS or any other display procedure / programme.

The ‘Shape’ class is the parent for the boundary / lakes / rivers / mountains classes. Instance objects from this class provide the container for lists of x/y/z-co-ordinates for the polygons that describe these graphically.

This class was added to the conceptual model in order to allow users of this conceptual model to specify graphical co-ordinates for say graphical display of the associated object without having to implement parts of the OpenGIS specification.

14.2.8.1 Relations

OGISPointer <>--->> shape_list(Shape)

List of shape objects defining the GIS view of the GTFObject

14.2.8.2 Protected Members

Value _coordinates

List of co-ordinate pairs in accordance to the specified projection

Value _projection

CODE LIST:

1. No projection / longitude - latitude
2. Mercator
3. Transverse mercator
4. Oblique mercator
5. Cylindrical equal area
6. Miller cylindrical
7. Equidistant cylindrical
8. Cassini
9. Albers equal area conic
10. Lambert conformal conic
11. Equidistant conic
12. Bipolar oblique conic conformal polyconic
13. Bonne
14. Orthographic
15. Stereographic
16. Gnomonic
17. General perspective
18. Lambert azimuthalequal area
19. Azimuthal equal area
20. Modified-stereographic
21. Space oblique
22. satellite tracking
23. van der grinten
24. sinusoidal
25. mollweide
26. eckert IV / VI

999. Other

DEFAULT: 1

List taken from [MAP87]

Value _zoom_level

Specifies the zoom-level of the coordinates.

Value _shape

The default SHAPE Value is rect, which defines a rectangular region using COORDS='left, top, right, bottom'. Other SHAPE Values are circle, which specifies a circular region using COORDS='centre-x, centre-y, radius'; poly, which specifies a polygonal region using COORDS='x1, y1, x2, y2, ..., xN, yN'. Coordinate Values are relative to the top left corner of the object and may be expressed as pixels or percentages. A percentage radius Value for circular regions is calculated relative to

the smaller of the object's width and height. If two or more regions overlap, the earliest specified region takes precedence.

14.2.9 Value

A helper class encapsulating any kind of types of values. This is a class, which serves as a container of 'attributes' of classes. The values can be of any type.

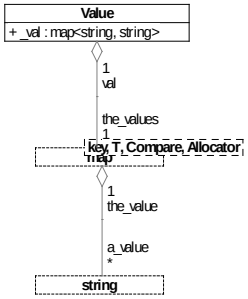


Figure 23: *Value diagram*

14.2.9.1 Relations

Value --->> Junction

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

Value --->> Node

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

Value --->> Population

This association is a component part of the implementation of the concept "an object can be attached to any other object" used by some toplevel classes.

14.2.9.2 Public Members

map<string, string> _val

map<> is a STL class Here in Value it is used as a hash table for key (string) -> value (string)

14.2.10Main

This is only used for implementation purposes (using a CASE Tool) and for compilation consistency it contains the main().

15 INDEX