Report on Standards Harmonisation

Including a GTFS to Transmodel / NeTEx mapping.

V1.0 revised with corrections for PT0302
Nicholas JS Knowles 2019.10.01

Table of Contents

1 Introduction

This paper summarises Standard harmonisation aspects of Transmodel v6.0.

The harmonisation work has involved comparison and analysis of other standards with Transmodel over a period of time, with the resulting modifications being integrated into a number of different areas of Transmodel, including some enhancements to the core Common Concepts model (for example, adding an abstract ROLE MODEL, and an abstract SECURITY LIST MODEL).

The requirements for additional harmonisation have been especially intensive in two particular areas:

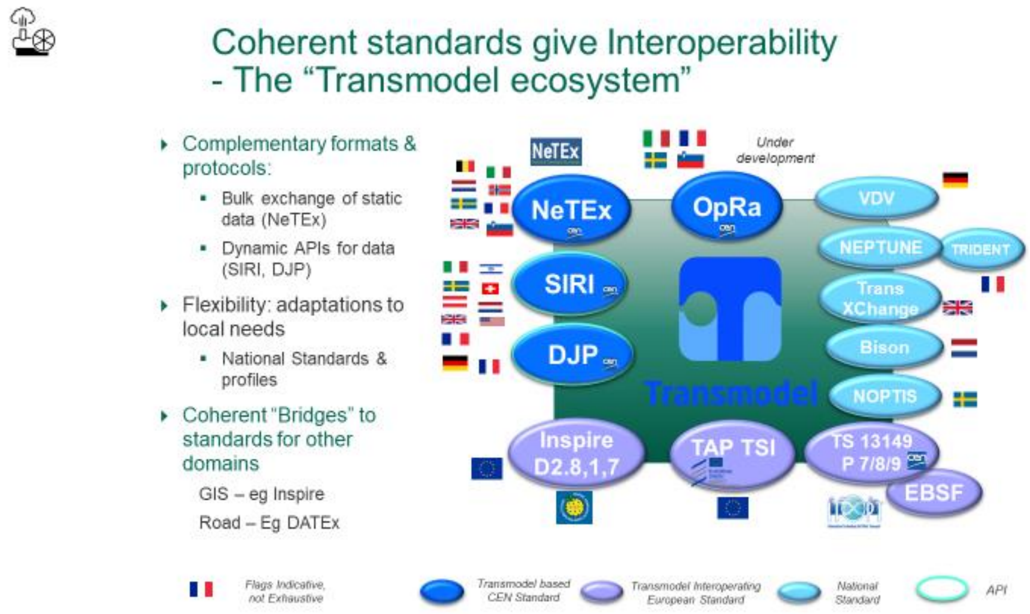
- **Harmonisation of the Passenger Information Model** (Part 6 of Transmodel v6.0). This is a natural consequence of this module being concerned with the mapping of Transmodel data elements to other APIs and formats.
- **Harmonisation of the Fare & Validation Models** (Part 5 of Transmodel v6.0). This is primarily because the Fares part of Transmodel has been extensively revised to meet new requirements from the CEN NeTEx format and other related conceptual standards (see below) – reflecting developments in modern ticketing technology.

The harmonisation work to compare different standards demonstrates the value and power of Transmodel as conceptual tool for the analysis and comparison of different representations of PT data, even if they use different terminology.

- A comparison of NeTEx and Google's General Transit Feed Specification (GTFS) provides an especially good example of this. Some notes are included in this report.
- A further presentation to show the strategic use of Transmodel for harmonising UK standards was developed and delivered to the British Standards Institute.

Figure 1 shows Transmodel as the basis for a family of interoperable data standards.

Figure 1 Transmodel related standards



2 Harmonisation of Passenger Information Models

The Transmodel v6.0 Passenger Information module (Part 6) includes a number of different submodels that show how Passenger Information may be mapped into views suitable for delivery as actual passenger information services, that is, in the form of APIs and other concrete formats. Such services are used by software applications to deliver PT information in both electronic and static formats.

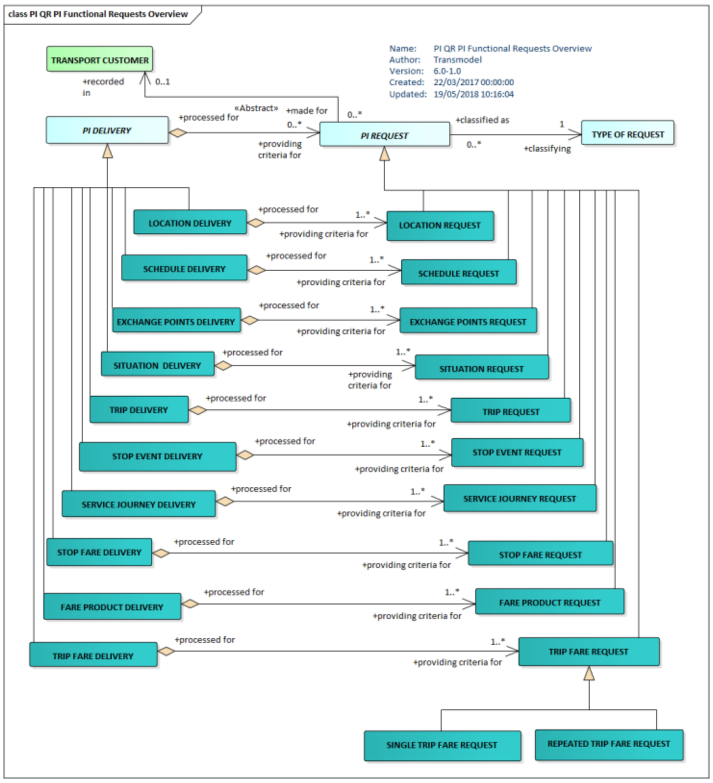
The Part 6 models have been very extensively revised and extended by the Transmodel v6.0 project, both to encompass enhancements to the underlying Transmodel models (for example a richer trip plan and fares model) and to harmonise them with some actual Transmodel based data formats so as to facilitate the mapping of model data to APIs.

In particular, the Transmodel model has been revised to be better aligned with two strategic CEN APIs (i) the CEN Open Access Distributed Journey Planning Mode (OA DJP), and (ii) the CEN SIRI functional services, the SIRI-SM and SIRI-SX, services being of particular significance.

To facilitate alignment of Transmodel terms for a given message pair in a concrete API, a specialised TM Query Model is defined that identifies the data elements needed to interpret and service the specific API. This model provides an “abstract query” as view of Transmodel concepts that can be readily mapped into different concrete formats; in the case of the DJP most of the messages and attributes even have identical names in the DJP to those in Transmodel v6.0.

Figure 2 summarises the PI Query MODELs used to harmonise standards APIs.

Figure 2 PI Queries in Transmodel v6.0



2.1 Harmonisation of PI Model with the Distributed Journey Planner

The CEN Distributed journey Planning (DJP) specification defines a standard API for trip planning related queries, including stop finding, journey plans, timetables and fares, all designed to be used in concert in customer information applications. Thus, for example a trip planner might use different queries in succession to first find available stops, then plan trips between stops, then get fares for those trips. The harmonisation of Transmodel with the DJP has a number of different aspects, summarised briefly below.

2.1.1 Revision of terminology to align with the Distributed Journey Planner

Transmodel terminology has be extensively revised at both at the message level (e.g. The TRIP OPTIMISATION TRANSACTION of Transmodel v5.1 is revised as a Transmodel v6.0 TRIP QUERY, made up of a TRIP REQUEST and a TRIP DELIVERY) and for individual elements and attributes (e.g. a Transmodel v5.1 TRIP becomes a Transmodel v6.0 TRIP PATTERN; the Transmodel v5.1 RIDE element becomes in Transmodel v6.0 a PT RIDE LEG, etc). See Appendix C of Part6 of the Transmodel Specification for a complete list of renamed terms.

2.1.2 Addition of new specific TM queries to represent DJP requests

The old Transmodel v5.1 model only had queries for journey planning (TRIP OPTIMISATION TRANSACTION) and rudimentary fare finding (FARE TRANSACTION). The DJP includes additional new specific queries that reflect common requirements of modern journey planners. Eight new queries have been added to Transmodel v6.0.

The following table shows the approximate equivalences between the DJP functional services and the TM Query model.

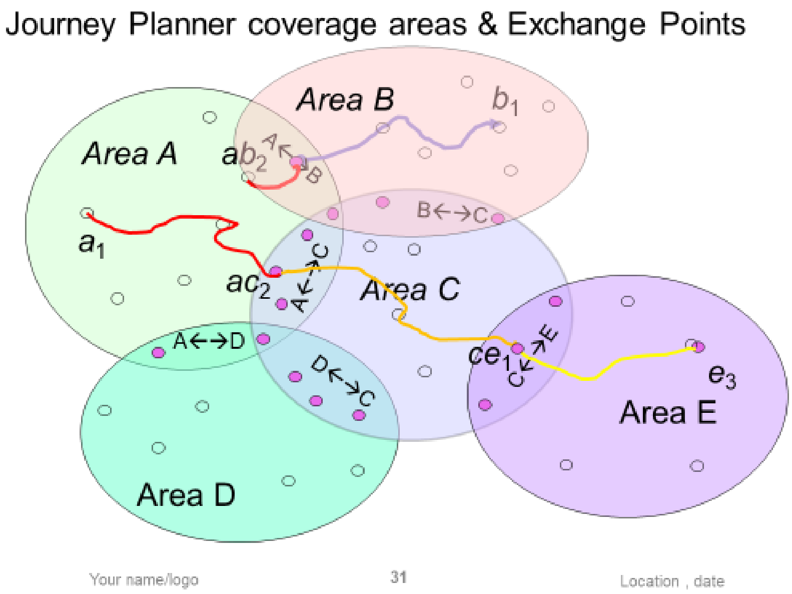
Table 1 OA DJP Service equivalences

OA DJP service	OA DJP Service	TM Query	Comment
EP	*Exchange Points*	*EXCHANGE POINTS QUERY*	New for Transmodel v6.0 from DJP.
LC	*Location*	*LOCATION QUERY*	New for Transmodel v6.0 from DJP
SC	*Schedule*	*SCHEDULE QUERY*	New for Transmodel v6.0 from DJP and other APIS, Including real time
SE	*Stop Event*	*STOP EVENT QUERY*	New for Transmodel v6.0 from DJP including real time
TP	*Trip*	*TRIP QUERY*	Renamed & Including real time
SJ	*Service Journey*	*SERVICE JOURNEY QUERY*	New for Transmodel v6.0 from DJP
FE	*Fare*	*SINGLE TRIP FARE QUERY*	Renamed and greatly enhanced

2.1.3 Modelling DJP Locations and Exchange Points

The simpler of the new DJP queries, such as for exchange points and for locations, can be regarded as implementation views that assemble existing Transmodel elements for use in queries. For example, the Exchange Point query returns the boundary points between areas covered by different journey planners (Figure 3); the underlying partitioning of stops into coverage areas for collaborating distributed trip planners can be modelled using existing Transmodel concepts such as an Administrative zone. The location query can be used to find stops and related places.

Figure 3 Exchange point concepts

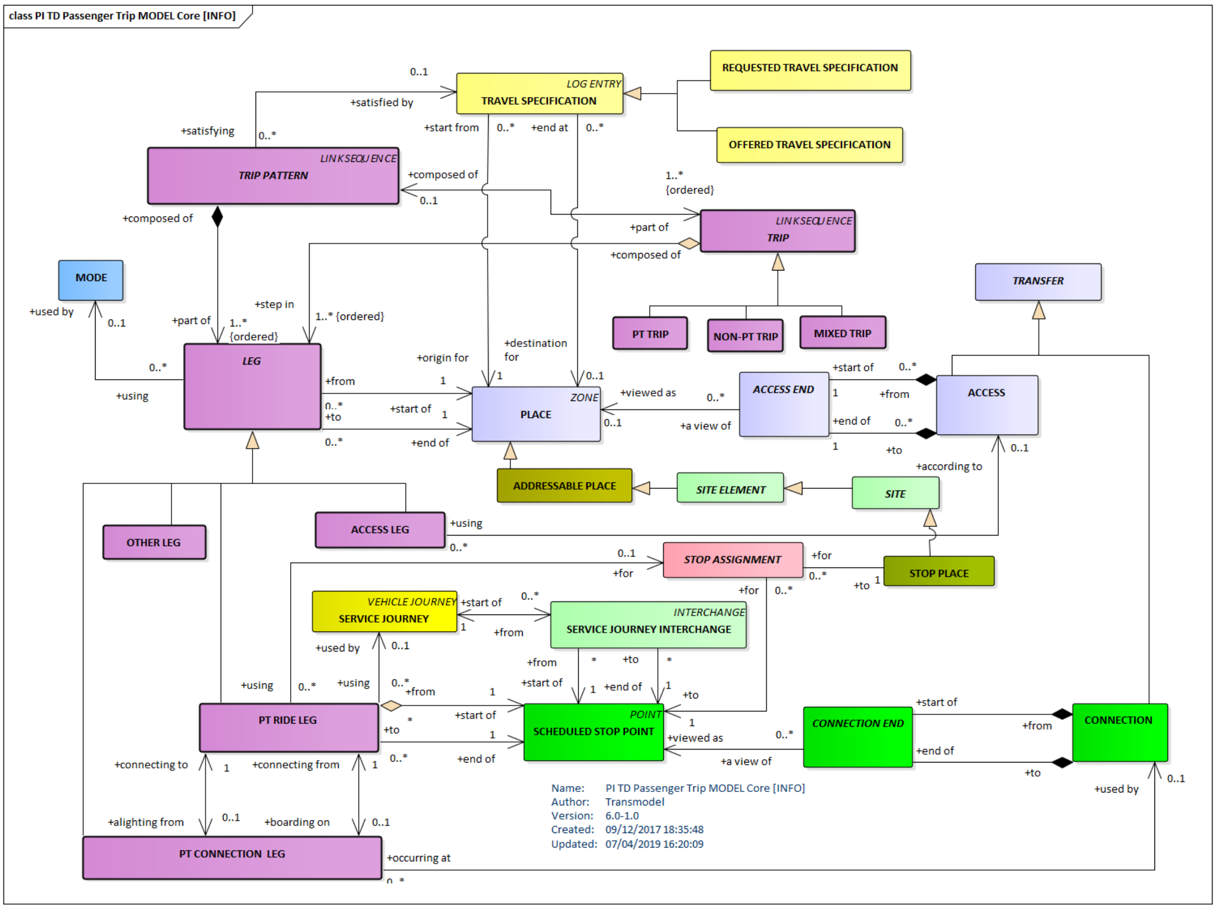


2.1.4 Addition of richer trip and travel models to support any mode DJP journeys

More complex queries, such as trip and fare queries, require a more complex intermediate "view model" to hold the results computed from the PT data. The old Transmodel v5.1 trip planning query (TRIP OPTIMISATION TRANSACTION) used a basic TRIP model to describe a user's itinerary (as say might be suggested by a trip planner) that assumed; (a) the use only of public transport (i.e. **not** other modes such as car or taxi); (b) had limited accessibility information; (c) did not cover any purpose-of-journey information – as is supported in Transmodel v6.0 to support the activities of personal trip planners and the production of passenger use statistics, etc.

The TRIP model in Transmodel v6.0 has been significantly enriched (Figure 4); (i) to support intermodal (i.e. PT and non PT) legs as found in the DJP and other standards; (ii) to support a Travel flow model so that passenger intent could be captured for statistical purposes (harmonising with an external requirements) and (iii) to include accessibility information for the legs and in particular the connection legs, as can be exchanged in NeTEx and elsewhere. The resulting itinerary of PT RIDE LEGs and PT CONNECTION LEGs can be mapped to the results of a DJP Trip Request.

Figure 4 Trip Model harmonised to use legs



2.1.5 Addition of richer fare models for use in fare query APIs

The old Transmodel v5.1 fare query (FARE TRANSACTION) had only a stub for both the inputs and the outputs to a Fare request and was not defined in any detail. The Transmodel v6.0 TRIP FARE QUERY has a much richer model (harmonised with the CEN NeTEx standard), defining an abstract API that can be mapped to concrete formats such as the DJP.

This FARE OFFER MODEL (Figure 5) is used to populate the results of a Fare Query such that (a) the Fare can be related to the journey elements being priced and (b) the Fare Offer can be related to the Fare Model elements representing the user's choices out of the selectable features of the offer; (c) the prices are separate from the elements being priced. This revised Transmodel 6.0 model now underpins the NeTEx model.

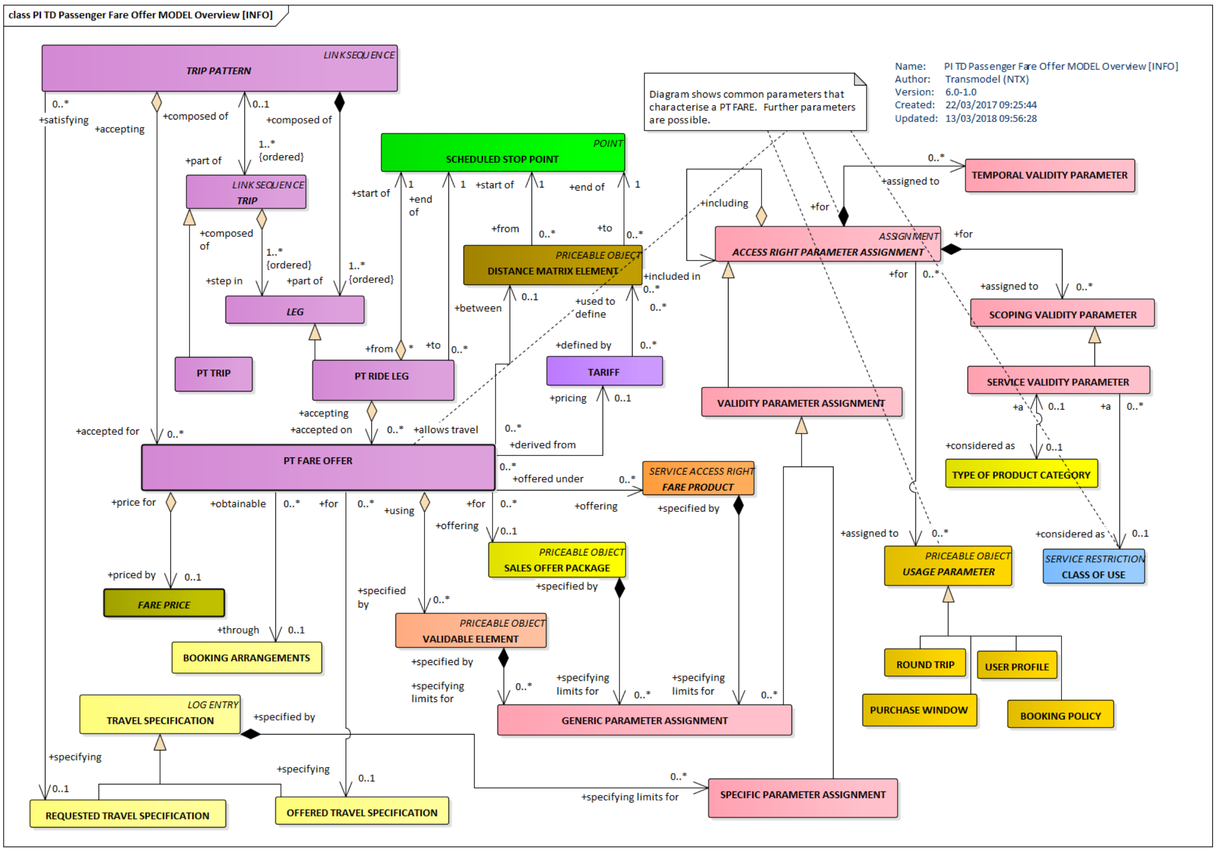


Figure 5 Overview of PT Fare Offer MODEL

2.2 Harmonisation with SIRI

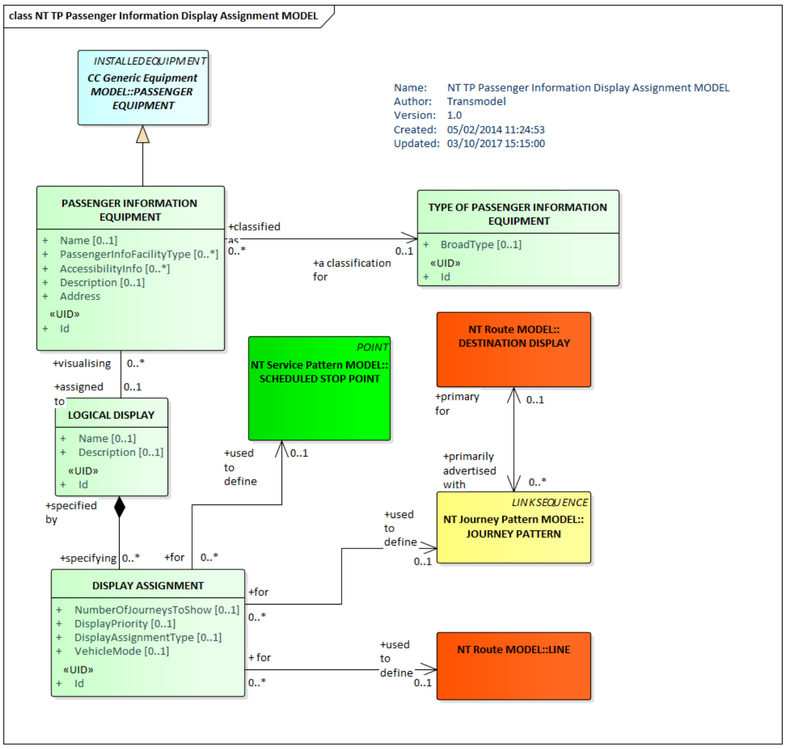
The harmonisation with the SIRI functional services has been improved both by improvements to the Transmodel v6.0 model, and by additional services in the Transmodel Part 6 Passenger Information Model.

2.2.1 Clarification of the Logical Stop Model

Historically, the mapping between the concept of a stop point in the Transmodel v5.1 model and a stop in the SIRI-ST and SIRI-SM services was not clear. In SIRI, the concept of a "stop" encompassed variously both a stop and a display (i.e. in Transmodel terms a PASSENGER INFORMATION EQUIPMENT) within a stop or other site. Analysis of this and other issues has led to enhancements to the Stop Model in Transmodel v6.0 (and also in NeTEx), allowing a more precise harmonisation of different standards. In particular the concepts of a LOGICAL DISPLAY and a DISPLAY ASSIGNMENT are introduced in order to articulate the distinct concepts (Figure 6).

Transmodel v6.0 also harmonises the separate physical stop (STOP PLACE, QUAY, BOARDING POINT, etc.) model from the IFOPT standard so that the relationship of physical equipment to a location within a station or other SITE can be represented – including the location of equipment such as displays.

Figure 6 Passenger Information Display MODEL



2.2.2 Addition of new specific Transmodel queries to represent SIRI requests

Some further specialised PI QUERies have been added to Transmodel v6.0 to clarify the mapping to certain SIRI services,

Table 2 shows the approximate equivalences between the SIRI functional services and the Transmodel v6.0 specific PI QUERY modes that provide abstract views of their content.

Table 2 SIRI Service Equivalences

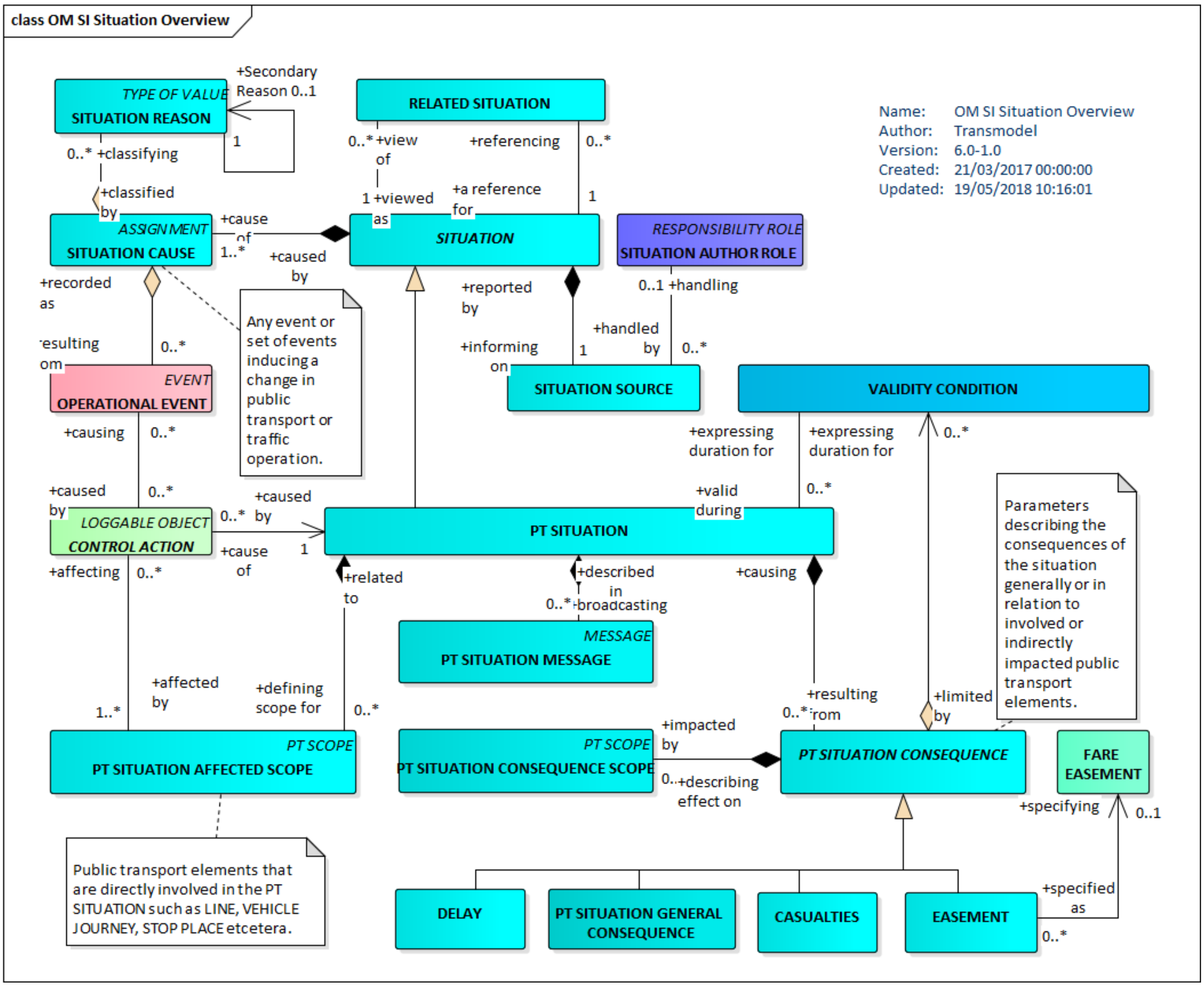
SIRI service	**SIRI Service**	**TM** *Query*	**Comment**
PT	*Planned Timetable*	*SCHEDULE QUERY*	Planned data only.
ET	*Estimated Timetable*	*SCHEDULE QUERY*	Including real time.
ST	*Stop Timetable*	*STOP EVENT QUERY*	Planned data only.
SM	*Stop Monitoring*	*STOP EVENT QUERY*	Including real time.
CT	*Connection Timetable*	Not described	Compose from INTERCHANGE, INTERCHANGE RULE, etc.
CM	*Connection Monitoring*	Not described	Compose from INTERCHANGE, INTERCHANGE RULE, etc.
GM	*General Messaging*	Not described	Compose from MESSAGE, etc
FM	*Facility Monitoring*	Not described	Compose from SITUATION, FACILITY, EQUIPMENT, etc.
SX	*Situation Exchange*	*SITUATION QUERY*	Added.

2.2.3 Development of the Message & Situation Models and SIRI-SX

Although Transmodel v5.1 had a Message Model, this was not elaborated either (a) in terms of the detailed structure of a message needed to describe the relevance of a message for a Passenger information system such that an application (for example a trip planner) could filter and process messages usefully, or (b) in terms of the life cycle of an incident, to describe messages for an evolving situation with successive updates and a need to integrate and reconcile successive bulletins. The SIRI-SX API introduced a significantly more elaborate model (with such features) and the Transmodel model has been accordingly extended to provide a generalised model to underpin the SIRI-SX as well as a mapping to other (for example, Datex2) situation models.

The enhanced Transmodel V6.0 general model for situations (Figure 7) separates out the different aspects of a Situation (identity, source, scope, nature, expected, impact, etc) from the audit and management aspects.

Figure 7 Overview of Situation Model



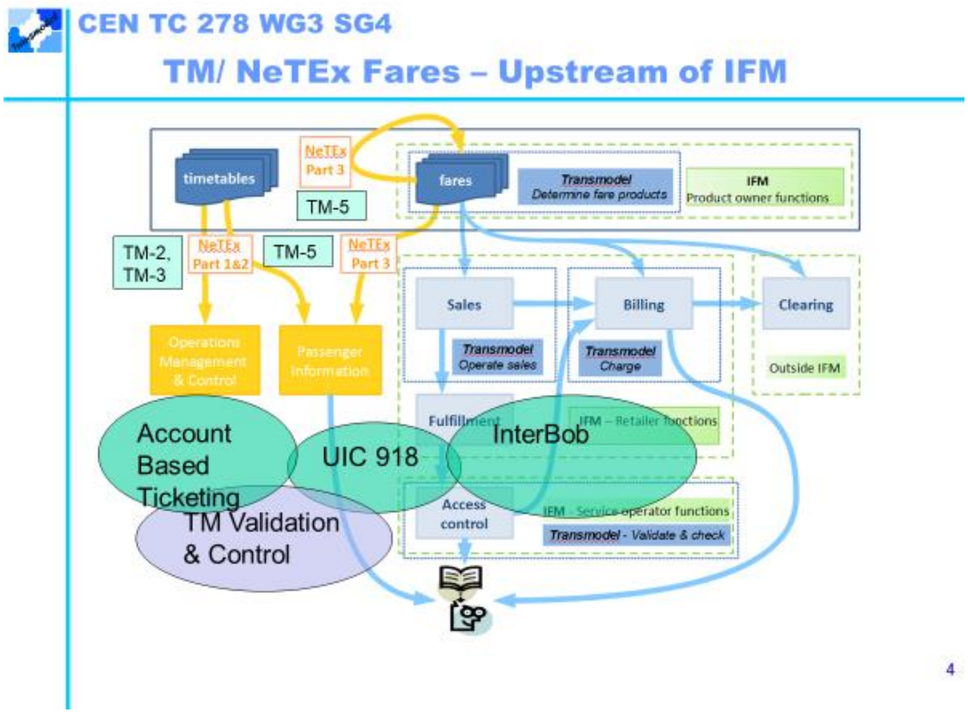
3 Harmonisation of Fare & Validation Models

Much of the standards harmonisation effort in the Transmodel 6.0 project has gone into considering the alignment of the Transmodel Fares Model and the Transmodel validation and Control Models with various other fare related standards.

These include both abstract high-a level standards covering processes and concepts (such as IFMS, and the ISO Account Based Ticketing report) with which it is helpful to align terminology and framework concepts as far as possible, and specific concrete formats (such as NeTEx v1.1) that include additional features not previously present in Transmodel.

Although the high-level standards predominately cover ticketing and back office functions that are downstream of the scope of Transmodel (Figure 8), attention to these standards allows Passenger Information systems to be aligned with them and the data used in trip planners and fare engines to be related to that generated by sales transactions and validation systems.

Figure 8 Some related Standards for TM Fares



3.1 Some relevant inputs

3.1.1 High Level Standards relevant for Fare Management & Validation architectures

- *IFMS (Public transport -- Interoperable fare management system* ISO 24014:2015),
  - A high-level architecture for Fare Management systems.
  - ➔Terminology for Roles, Events, etc.
- Smart card standards (*Identification card systems - Surface transport* EN 1545:2015*).*
  - Covers concepts and terminology relevant for electronic fare media.
- *Account-Based Ticketing State of the art report* (PD ISO/TR 20526:2017):

  - a report on account-based ticketing sponsored by the UIC that identifies architectures, roles and terminology needed for Account based ticketing (ABT)
  - ➔Terminology for Roles, Events, etc.

- *InterBoB* (Interoperabilité Back-Office Billettique) Smart Ticketing Alliance
  - An - architecture for Fare Back office settlement and clearing systems.
  - ➔Terminology for Roles, Control and Validation Events, etc.

3.1.2 Standards relevant for ticket media and validation

- *Data Exchange for Ticket Check and after sales operations with electronic information.* (UIC 918 – 4 (Validation): UIC standards for fare.
  - Concepts for validation and ticket content.

3.1.3 Standards relevant for product features & parameters

- *TAP TSI SPECIFICATIONs for Rail Fares:* European Rail Authority *B1 (NRT), B2 (IRT), B3 (Special).*
  - The European Rail Authority has a number of standards governing standard rail fares
  - ➔Concepts for routing restrictions (Series Constraints), distribution and fulfilment, and shared properties of fare offers.
- Full-Service Model, **(**Full Service Model Initiative**) -** A Reservation and Ticketing API for rail and other PT systems.
  - Equivalent concepts for tickets and fare APIS.

3.1.4 Concrete Fare data Standards and projects

The NeTEx 1.1 Standard contains many enhancements to cover additional business requirements. These are summarised in a section below. A number of these have arisen from active projects using NeTEx in the field. These include:

- BIP (Turin 5T)
  - A regional project providing integrated PI and fare data (e.g. identified gaps: support for Security features White lists, retail consortiums, etc)
- The Norwegian national fare project (Entur)
  - A project to represent multimodal fares in Norway in Transmodel/NeTEx (e.g. identified gap: Season Pass Suspension parameters).
- ERA Rail Domestic Fare project:
  - A project that examined existing European Domestic Rail fare products and product conditions in 28 European countries to look for gaps in Transmodel/NeTEx. (e.g. identified gaps: support for Subscription parameters.)

3.2 Alignment with NeTEx v1.1

The revised version 1.1 of the NeTEx schema and specification, in particular for the NeTEx Part3 (Fare) submodel, included many enhancements to the NeTEx fares model that have now been incorporated into Transmodel so that the two standards remain harmonised. As well as adding features to Transmodel v6.0 that had already been added to NeTEx v 1.0 and the first iteration of NeTEx v1.1, some further enhancements have come from recent projects in 2017-2018 in the field (as in BIP in Italy and the Entur national project in Norway) and projects such as the European Rail Authorities' study of domestic rail fares in 28 European countries.

The specific new features to improve harmonisation of the Transmodel fares model with that of NeTEx are summarised in the detailed change requests given in Appendix C of the Transmodel v6.0 Part5 (Fares) specification. They cover both network fare properties, access rights, fare product types and parameters, sale offer and distribution features, and validation and control features.

3.3 Fare Management Roles

Although back office ticketing and reservation systems and the settlement of fares are for the most part outside of the scope of Transmodel, certain concepts that underly back office architectures are relevant for relating passenger information to fares. Two particular examples are; (a) the roles involved in providing and managing fare products; and (b) the events involved in purchasing and using fare products.

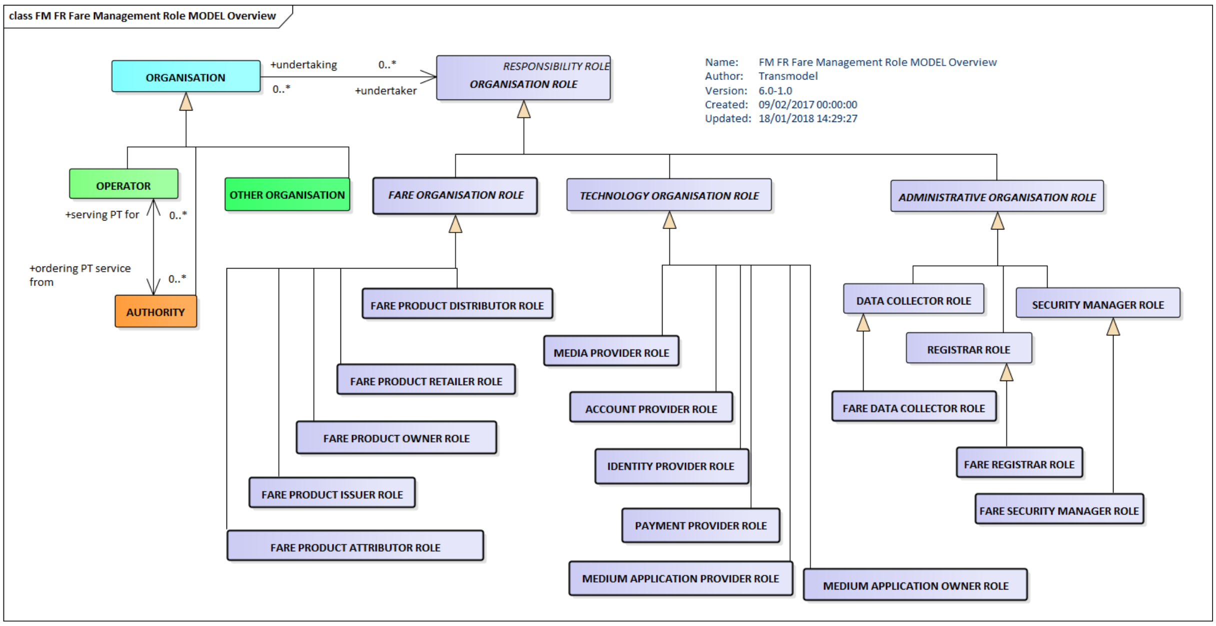
The harmonisation work for roles has involved reifying as Transmodel model elements the respective roles identified in fare management architectures, such as product owner, identity provider, etc., etc.

3.3.1 Product and Fare Management Roles

Figure 9 shows a number of named role relevant for defining Fare Products and for operating Validation and Control processes on public transport.

The roles may be undertaken by the same or different organisations. For example, classically an Operator both operated and owned the product definitions, sold them, distributed them, and validated and controlled their use. Now some or all of these tasks may be delegated to different stakeholders. Electronic ticketing and payment systems have also led to new roles such as providing security, collecting payment from the customer, providing an online account, etc.

Figure 9 Roles for fare Management



3.4 Control and validation

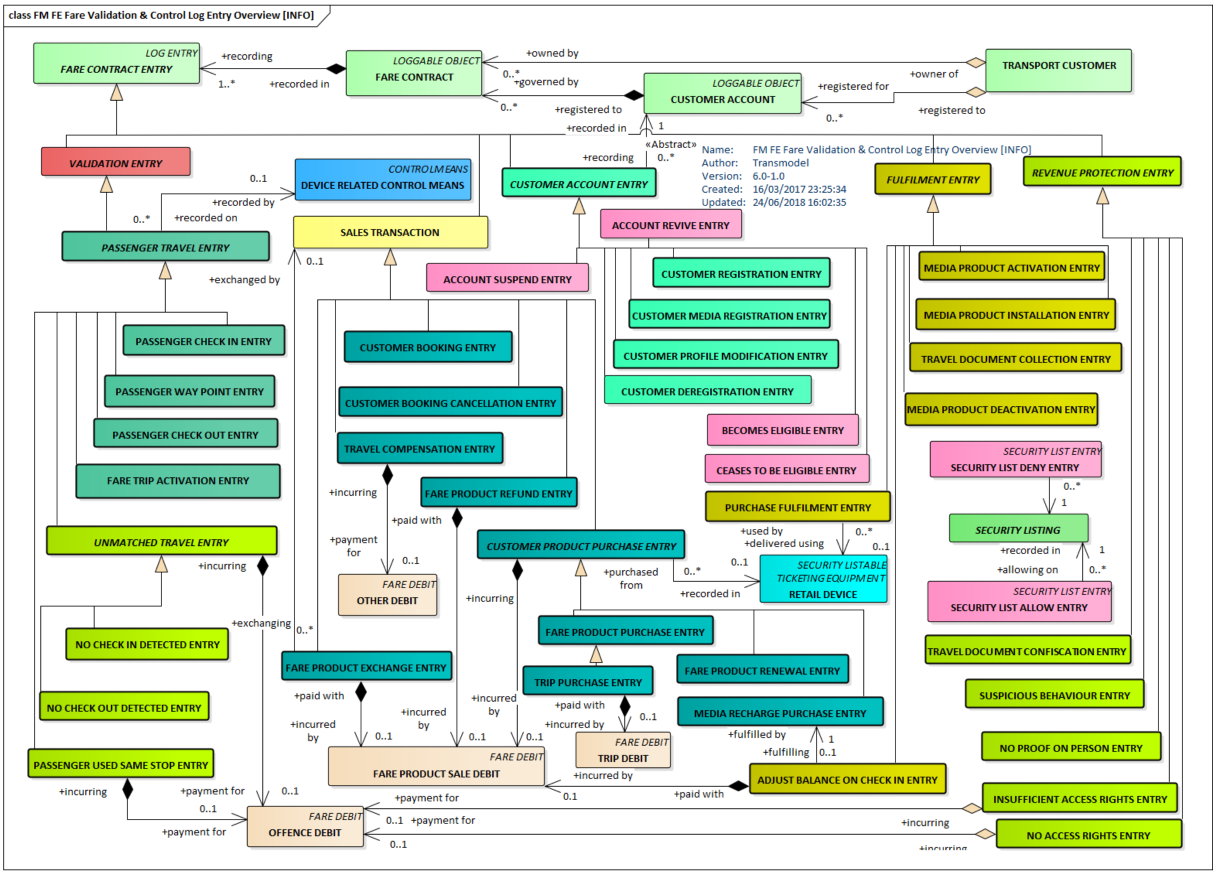
Similarly, in an era of electronic ticketing, the representation of validation and control events needs standardisation so that equivalent concepts can be identified in both passenger information, validation, and fare management systems. For example, when a user purchases a product, pays for a product, or consumes a product by validating it or checking in or out, etc. This becomes even more important when sophisticated Pay-As-You-Go products are available (such as Transport for London's capped Oyster product) as charging and billing are undertaken automatically, yet it still needs to be possible to give users a justification of their travel consumption in terms of their products, access rights and travel patterns.

The harmonisation work has included identifying and naming a common set of atomic events for travel purchase and consumption, together with the resulting recorded log entries (as Transmodel FARE CONTRACT ENTRies). Where possible, terminology was aligned with existing IFMS and other standards, in particular the ISO Account Based Ticketing report.

3.4.1 Validation Events

The various sets of log entries, coloured by functional area, are shown in Figure 10. See the Transmodel v6.0 Fares UML model for a separate more detailed model of each functional subgroup and the relationships of individual entries with other data elements.

Figure 10 Fare Contract Entries for Fare Management



4 Identifying Future requirements for Transmodel

The work on the Transmodel v6.0 project has been undertaken in parallel with ongoing work to develop a revised NeTEx 1.1 schema and also a new European Passenger Information Profile (EPIP). The activity of harmonisation, i.e. comparison of Transmodel with these other standards, has also identified further real-world requirements for Transmodel.

Many, but not all of the requirements for additional features have already been incorporated into the revised Transmodel v6.0 model. In particular, the later stages of harmonisation work (undertaken after the Transmodel v6.0 fare model had been stabilised) identified a few ancillary features that are candidates to add to a future release of Transmodel to meet business requirements. Some of these are noted here below.

4.1 Fare Related requirements

- **Subscriptions**: Usage parameter to represent the rules for subscribing to a product (payment intervals, notice periods, etc)
- **Season Pass Hiatus**: usage parameters to represent the rules for suspending a season pass for illness, holiday etc.
- **Eligibility Change Policy**: usage parameters with rules for product validity if a user's eligibility changes.
- **Sales Offer Entitlements**: some entitlements are defined between Sales Offer Packages rather than Fare Products.

4.2 General requirements

Some requirements are not specific to Fares:

- Explicit Frame dependencies to indicate prerequisite frames.
- Vehicle Journey Stop Assignment.
- Snow and Ice modes.
- Suspension and Partial Refund product usage parameters.
- Logical Seating plan Model.

4.2.1 Logical seating plan model

The choice of available accommodation and seating is in effect part of the passenger information set available to passengers and also forms part of the travel specification chosen or assigned to a user. The current Transmodel vehicle and Train models could be extended with a small submodel to describe the available seating on different services, for example sufficient to represent the example in Figure 11.

Figure 11 Example of a seating plan

4.2.2 Frame dependencies

Often a data set will be split among several Version Frames, with some frames depending on other frames as prerequisites. This dependency can in principle be derived from the individual low-level dependencies between elements in the frame and elements in perquisite frames but it is also useful to represent the frame level dependency.

5 Using Transmodel to compare NeTEx and GTFS

Transmodel can be used as an analysis tool to compare and establish equivalences between different standards, even those with a quite different origin and terminology – a form of conceptual harmonisation that facilitates interoperation between standards.

In the past this has been done notably to harmonise the Transmodel model of the transport network with geospatial standards such as GDF (Geographic Data Files) by establishing a common spatial feature model; in consequence, it is now relatively straightforward to integrate spatial data from INSPIRE data sets with transport data from Transmodel based data sets in applications that need to do so; thus Transmodel and Inspire can be considered as interoperable "sister" standards.

As a current example of Transmodel as an analysis tool, Transmodel has been used to compare the timetable and fare models of the General Transport Feed Specification (GTFS) with those of NeTEx. This exercise serves several purposes:

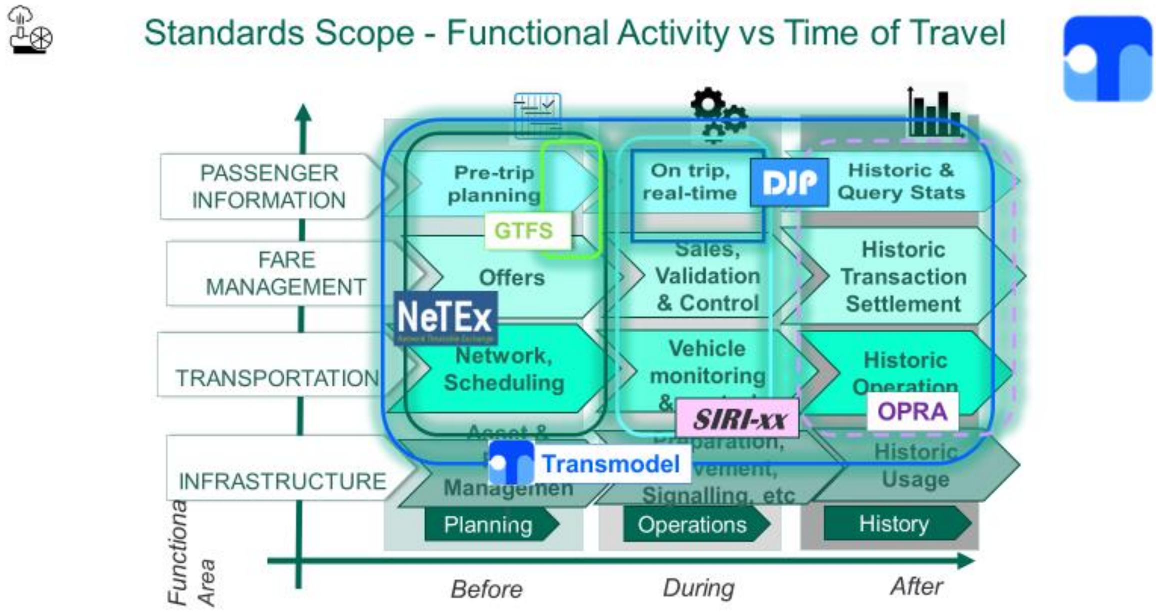
1. To characterise the precise scope of GTFS and untangle certain aspects of its representation, for example.
    a. Some concepts are referenced in GTFS but not specifically reified as GTFS CVS tables. (For example, GTFS ***Stop*** reference a Zone record that is not defined concretely).
    b. In GTFS in certain cases, multiple separate concepts are combined into a single entity. (For example, GTFS ***Fare_attributes*** includes both a price and a data about how a product may be purchased.)
    c. In GTFS in certain cases, concepts are not normalised but repeated on individual elements. (For example, GTFS ***Fare_attributes*** repeats data about how a product may be purchased on every single price.)
    d. Certain GTFS records overload a single data structure with multiple, separate concepts and concerns in different contexts, so that interpretation of content by a program is complex. (For example, a GTFS ***Stop*** may be used variously to represent a whole station or part of a station, or a GTFS ***Transfer*** may be used to present the possibility of a general connection between two stops or a specific interchange between trips).
2. To demonstrate how Transmodel/NeTEx includes all the necessary GTFS functionality.
3. To establish a mapping between GTFS and NeTEx that can be used to transform datasets from one format to another.
4. To characterise specific limitations of GTFS that restrict its use to certain applications (e.g. the lack of timing information), and in particular the limitations of its fare model.
5. To give a basis for translation of GTFS concepts into other languages (Since definitions of Transmodel concepts are available in many European languages.

5.1 High level comparison of scopes

A question frequently asked by stakeholders investigating PT standards for use in their organisation is "What is the difference between NeTEx and GTFS?" - or even "Why not just use GTFS?" - and it is of value to be able to articulate the nature of the differences, both at a high level, and in exact detail.

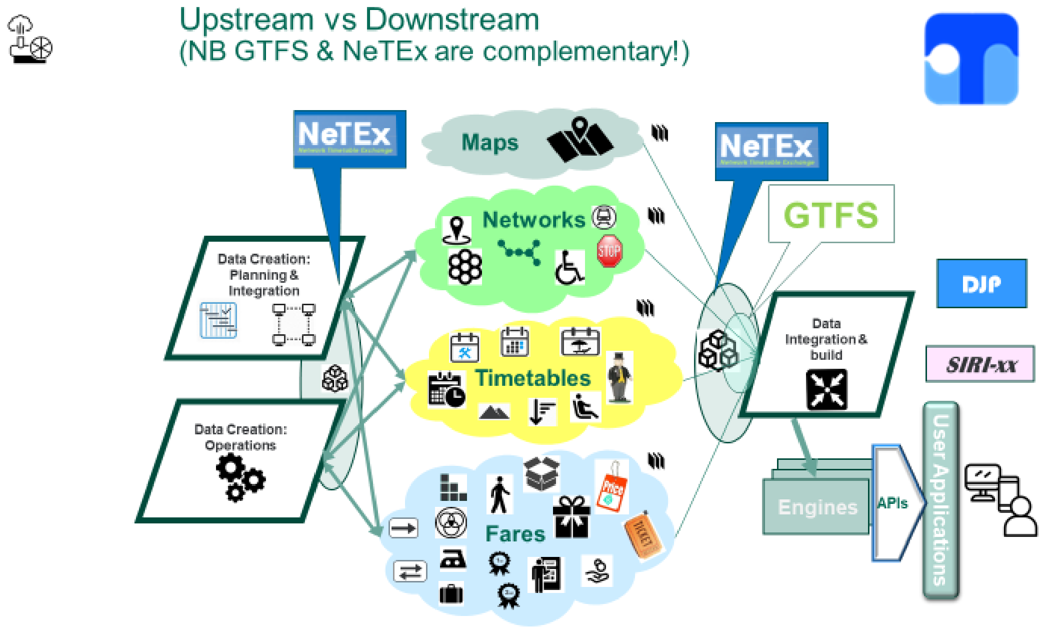
Within the domain of public transport information, Transmodel covers a wide scope, both in terms of different functional areas (network, vehicles, timetables, fares, control etc) and across different temporal phases for those domains (planning, operations, historical data etc), each of which aspect may require additional specific submodels. A starting point in any comparison is thus to give a relative indication of which areas are covered by the respective formats: GTFS focuses on final timetable data; NeTEx covers upstream functions as well (Figure 12).

Figure 12 Relative scope of different PT standards



One significant difference between Transmodel/NeTEx and GTFS concerns their use in upstream planning and data creation processes (Figure 13). Whereas GTFS is primarily intended for the delivery if finished timetables to third party trip planners, Transmodel/NeTEx is also capable of representing the many additional concepts such as service patterns, timing patterns, timing information, vehicle scheduling, day types, time demand types, etc, needed to create such data sets in the first place, as well as operational data sets never seen by the public (dead runs, driver and crew schedule, etc). That is to say, it is also intended to support planning and operational use cases rather than just the distribution of Passenger Information.

Figure 13 Use of standards in upstream and downstream processes.



In this regard, the standards may be seen as complimentary (providing there is a clear mapping between them); GTFS gives a relatively simple for distributing final timetable data to trip planners who adhere to the Google managed identifier system; Transmodel/NeTEx gives a richer format for assembling a and exchanging a fuller data set that can be used in any peer-to peer configuration between upstream and downstream users.

5.1.1 Enriching of GTFS with new Transmodel like features

GTFS was originally developed by Google Transit and its development has followed subsequent enhancements to Google Transit.

Like first generation European National formats, the original GTFS specification lacked a number of important features and so has been steadily evolving over time, with the addition in successive releases of concepts already available in Transmodel to enrich the expressiveness of GTFS. A degree of harmonisation is thus slowly taking place between GTFS and Transmodel. Examples of features that have been added to GTFS include;

1. Enhancing GTFS ***Stops*** to distinguish between timetabled stops and physical platforms (i.e. Transmodel STOP PLACE and QUAY).
2. Enhancing GTFS ***Stop_times*** with a ***stop_headsign*** to allow a change of heading at different stops (i.e. a simplified denormalised version of the Transmodel DESTINATION DISPLAY concept).
3. Enhancing GTFS ***Stops*** to distinguish between other physical locations within a transport interchange (i.e. IFOPT/Transmodel QUAY, ENTRANCE, BOARDING POSITION and ACCESS AREA concepts).
4. Adding GTFS ***Levels*** to indicate the physical level of components of a GTFS stop (IFOPT/Transmodel LEVEL).

5. Adding GTFS ***Pathways*** to specify detailed navigation paths within stations and transport interchanges (i.e. IFOPT/Transmodel LEVELs & PATH LINKs).
6. Adding GTFS Extension **route_types** (i.e. Transmodel SUB MODEs).
7. Enhancing GTFS ***Transfers*** to allow both trip specific transfer rules and general transfer rules between lines or stops (i.e. Just as Transmodel distinguishes between CONNECTIONS, SERVICE INTERCHANGEs and INTERCHANGE RULEs).
8. Enhancing GTFS ***Feed_info*** with feed contact attributes to distinguish between the publisher and the originator (i.e. DATA SOURCE versus ORGANISATION).
9. Adding Gtfs ***Translations*** to allow text in other languages (i.e. Transmodel ALTERNATIVE TEXT).

Note that Transmodel is an open standard and adopting its terminology and data structures is permitted by the terms of use for Transmodel and by the NeTEx GPL licence - and should be welcomed by both GTFS and Transmodel communities as (a) demonstrating the validity of Transmodel as a reference model; and (b) facilitating the interoperability of data between data sets in the two standards.

5.2 Analysing GTFS

In order to make a detailed and precise comparison with Transmodel/NeTEx, GTFS must also be modelled at least to some extent in a comparable notation.

GTFS does not have a formal model *per se*, but a UML model can be reverse engineered from the GTFS specification, albeit with some anomalies because some GTFS elements are overloaded to have different meanings in different contexts, or are not explicitly modelled but only implied by enumeration values or reference.

Note: This discussion is based on GTFS as of September 2019 see https://developers.google.com/transit/gtfs/

Note also that GTFS exists in two versions – (i) regular ***GTFS;*** and (ii) with ***GTFS Transit extensions*** – the latter being a set of provisional enhancements to GTFS to add additional feature that have already been adopted by the Google's own Google Transit feed. This comparison includes consideration of GTFS Transit extension features.

5.2.1 GTFS treated as The GTFS "Model"

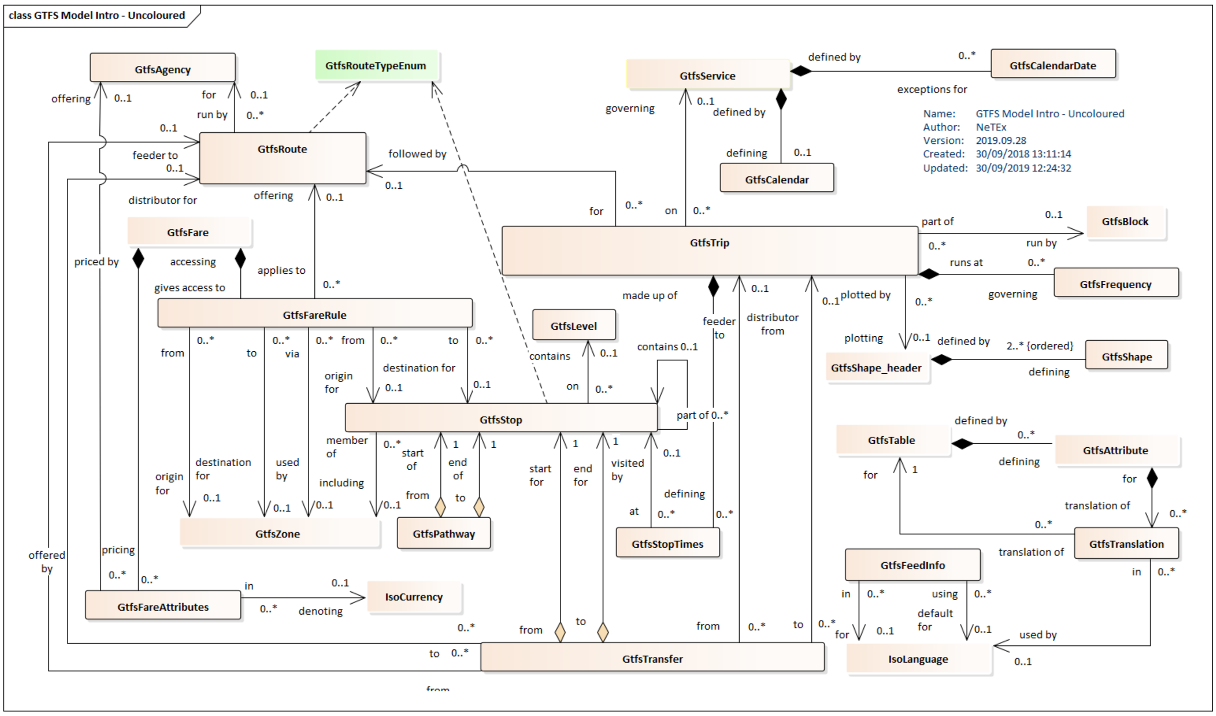
Our methodology is thus (a) to establish a GTS model in UML; then (b) to examine each GTFS element in turn to establish its Transmodel/NeTEx equivalent(s) as entities, attributes and relationships. In some cases the mapping is one-to-one, in others the mapping is more complex, usually because the Transmodel representation is more normalised or richer, breaking a concept down into two or more separate concerns and allowing for additional capabilities; but sometimes because the GTFS representation is overloaded, i.e. uses the same CSV table to represent different semantic concepts under different use cases, whereas Transmodel generally prefers to have a distinct entity for each concept.

Note that this comparison focuses on the basic semantic differences between the Transmodel and GTFS models. There are also some material differences between the technologies used in GTFS and NeTEx (such as the use of CSV versus XML, or the use of globally unique identifiers in NeTEx rather than Google issued ones in GTFS) that have some significant implementation consequences but are not compared in detail here. A third category of difference lies in the metadata of their semantic models: Transmodel/NeTEx has mechanisms such as Version-Frames that provide a uniform fine-grained model for grouping different versions of data for exchange (in practice, an important consideration when data is repeatedly exchanged between different stakeholders), whilst GTFS uses only a rudimentary set level versioning. Since metadata has semantics it is also considered.

5.2.1.1 The GTFS Model Introduction - Uncoloured

Figure 14 shows the basic GTFS model as a UML class model. Relationships are inferred from references held as "foreign key" attributes on specific elements.

Figure 14 GTFS Model Introduction – Uncoloured

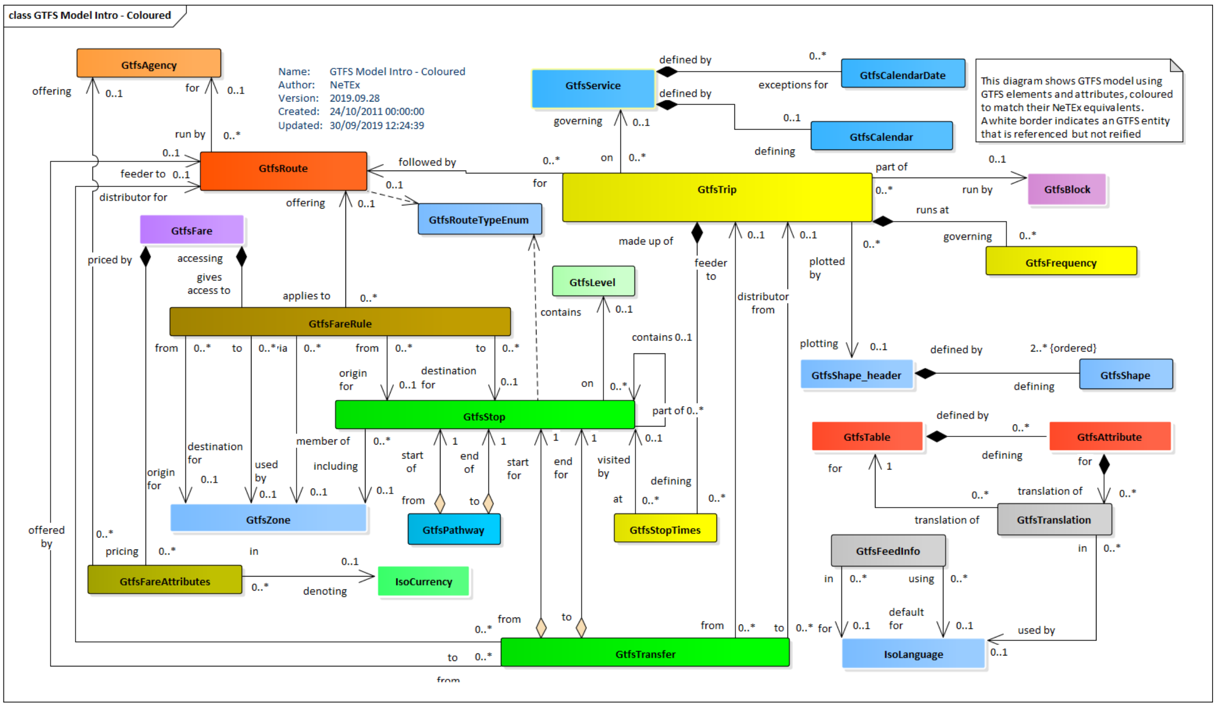


5.2.1.2 The GTFS model Introduction - Coloured

Figure 15Figure 15 shows the same GTFS model with the elements coloured, using the same set of colours used in Transmodel for equivalent functional areas. For example, yellow is used for GTFS Trip (Transmodel SERVICE JOURNEY). This presentation convention helps in making a high- level comparison and in achieving a gestalt grasp of a model when presented as a diagram.

Note that the diagrams also follow the Transmodel conventions for labelling relationships using role names at either end of an association. These should be read directionally from each end, for example; "***GtfsAgency*** *for* ***GtfsRoute***" / "***GtfsRoute*** *run by* ***GtfsAgency***".

Figure 15 GTFS Model Introduction – Coloured



5.2.1.2.1 Presentation conventions

Along with the use of colour, Transmodel presentation conventions for UML diagrams, as described in the annexes of the Transmodel v6.0 and NeTEx v1.1 specifications, are used in the diagrams in this paper. As additional conventions for this analysis; (i) The entity name used for the GTFS entity is generally the singular of the GTFS file name, prefixed by Gtfs, thus GTFS ***trips.txt*** becomes ***GtfsTrip;*** (ii) GTFS entities which are implied to exist by reference in the GTFS Model, but not actually reified as GTFS records are indicated by a white outline. For example, ***GtfsBlock*** and ***GtfsFare***.

Within the text of this mapping paper, GTFS entity names are given in bold italics, e.g. ***GtfsAgency***; (note however that in an UML diagram, italics denote an abstract class).

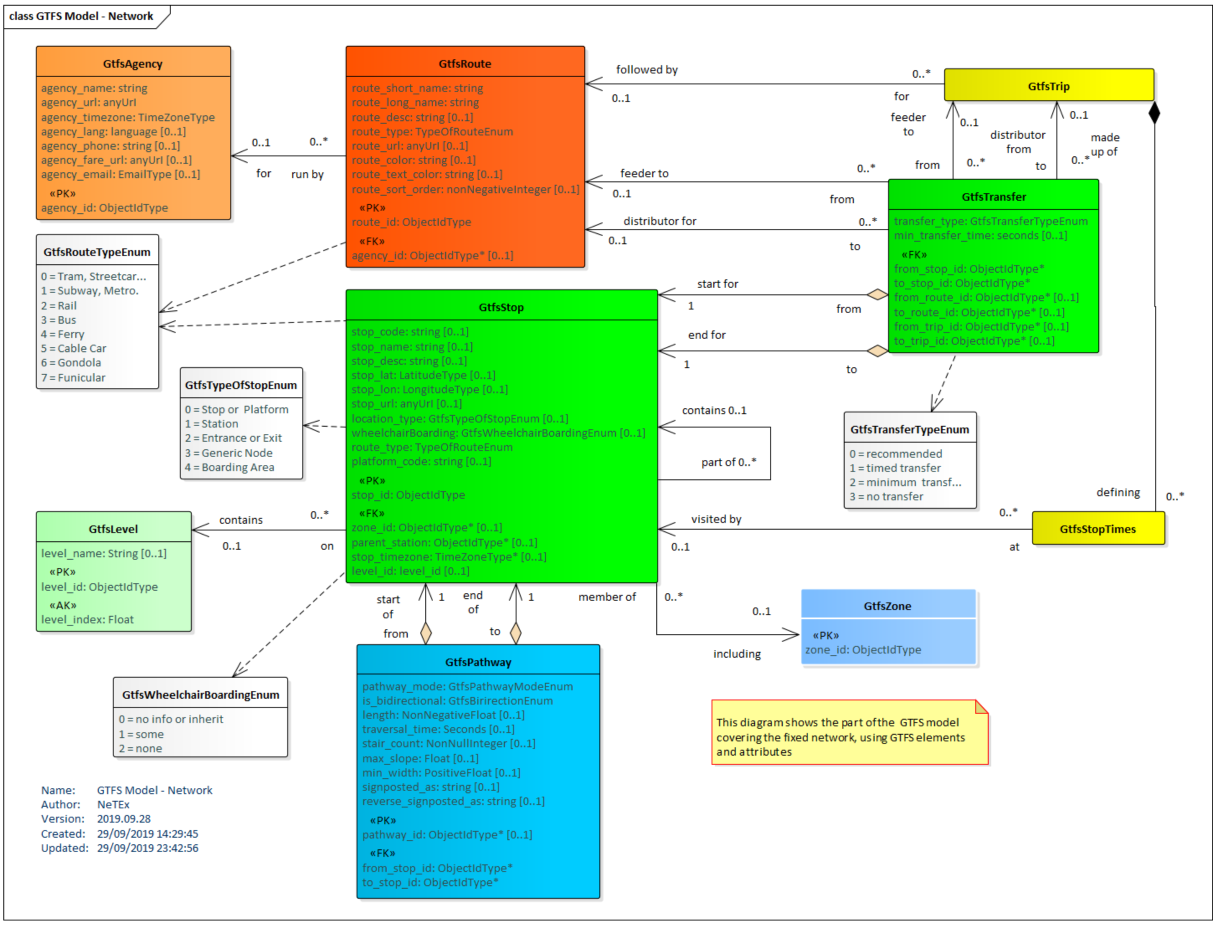
5.2.1.3 The GTFS Model – Detailed attributes

The GTFS model includes detailed attributes for specific elements; these are shown modularised by function into as four successive diagrams; starting with a model of the GTFS elements representing the fixed network in Figure 16.

5.2.1.3.1 The GTFS Network Model – Details

Figure 17 shows GTFS records describing the stops and routes of the fixed network; these may be shared between many different services and fares.

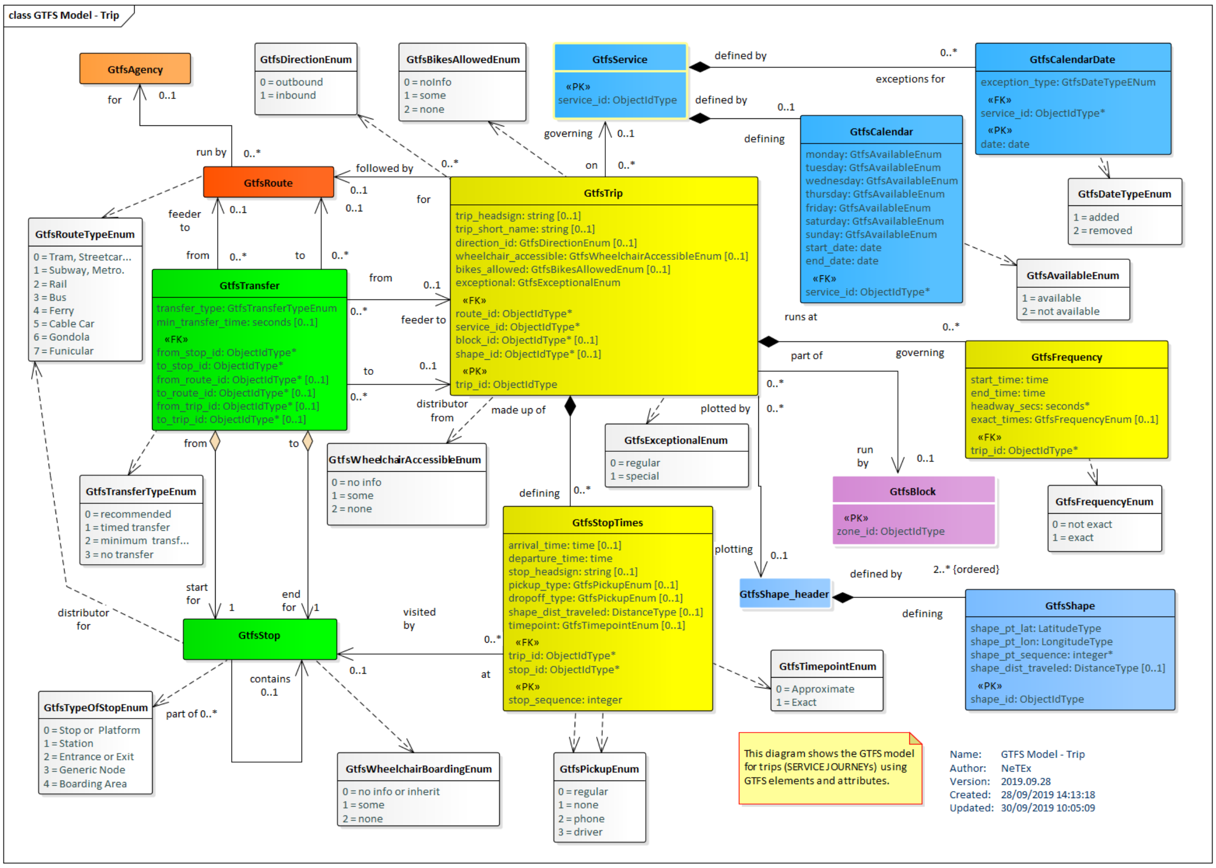
*Figure 16 GTFS **Network** Model – Detailed attributes.*



5.2.1.3.2 The GTFS Trip Model – Details

Figure 17 shows GTFS records describes the scheduled journeys over the network.

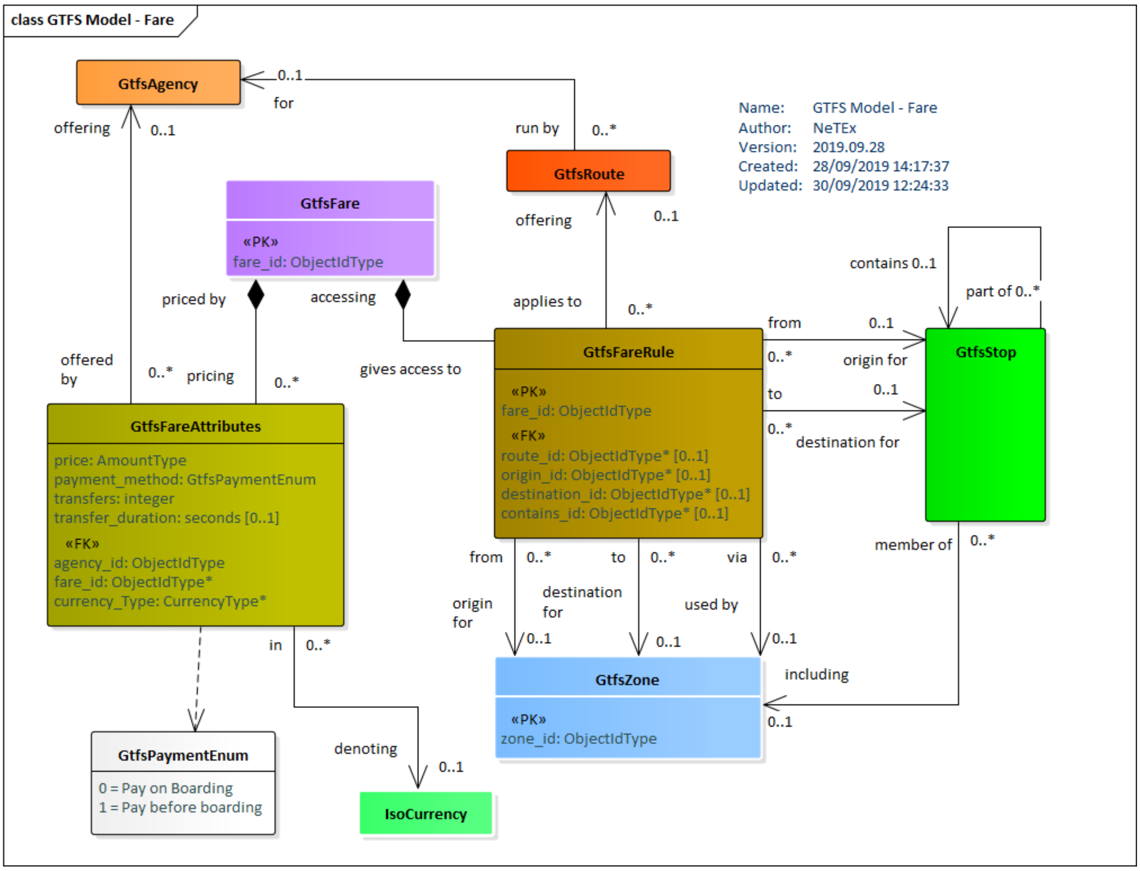
*Figure 17 GTFS **Trip** Model – Detailed attributes.*



5.2.1.3.3 The GTFS Fare Model – Details

Figure 18 shows GTFS records describes the prices of fares available for the network.

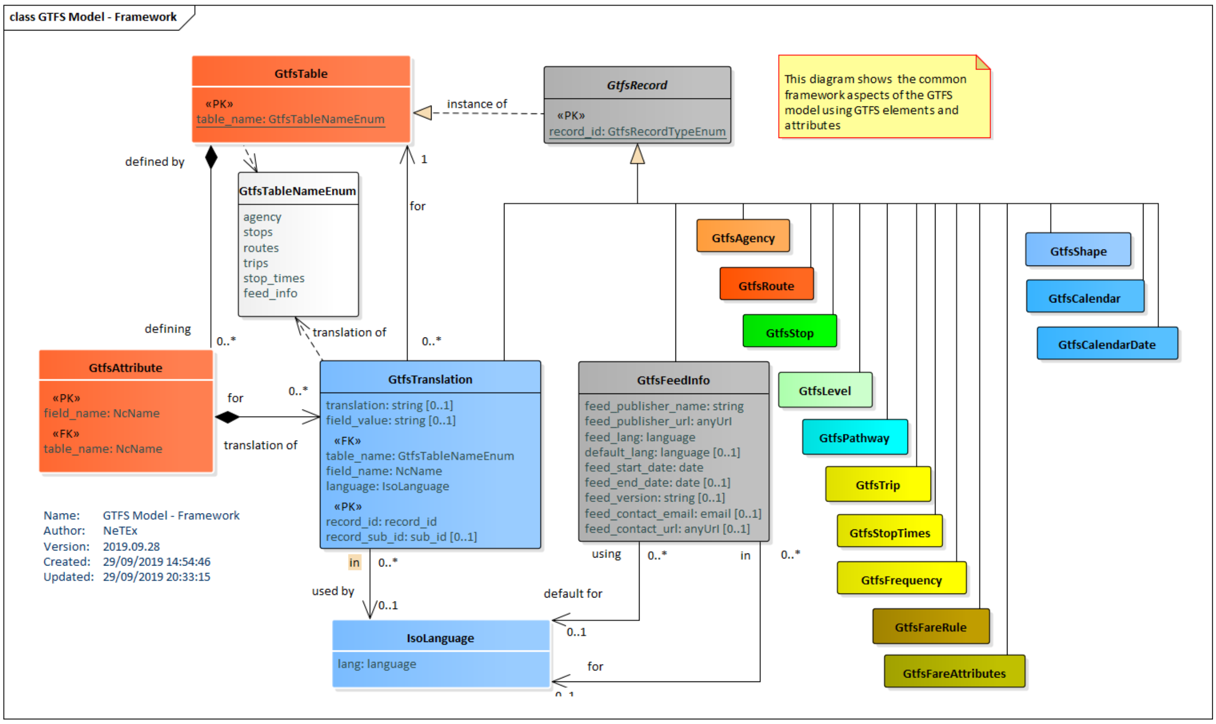
Figure 18 GTFS Fare Model – Detailed attributes.



5.2.1.3.4 The GTFS Framework Model – Details

GTFS has some common features used for all its different record types; (i) to describe the version and source of the data set (GTFS ***Feed_info***); and (ii) to describe natural language translations of specific fields in specific tables (GTFS ***Translation***). Figure 19 shows GTFS records describing the common framework elements.

*Figure 19 GTFS **Framework** Model – Detailed attributes.*



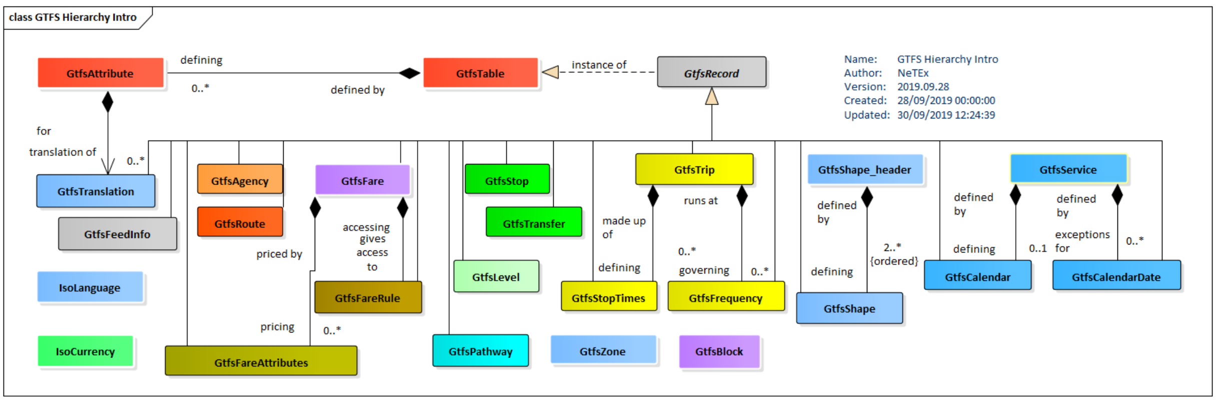
5.2.2 GTFS Hierarchies

The various GTFS CSV tables can be regarded as specialisations of an abstract GTFS ***Record*** that has a metamodel of record types (given as ***GtfsTable*** below) and attributes (given as ***GtfsAttribute*** below).

Furthermore, certain of the GTFS tables represent aggregations of other subrecords. For example, a GTFS ***Trip*** is composed of GTFS ***Stop_times***. Aggregations are indicated by black diamonds in a UML class diagram

Both specialisation and aggregation hierarchies for GTFS are shown in Figure 20.Figure 20

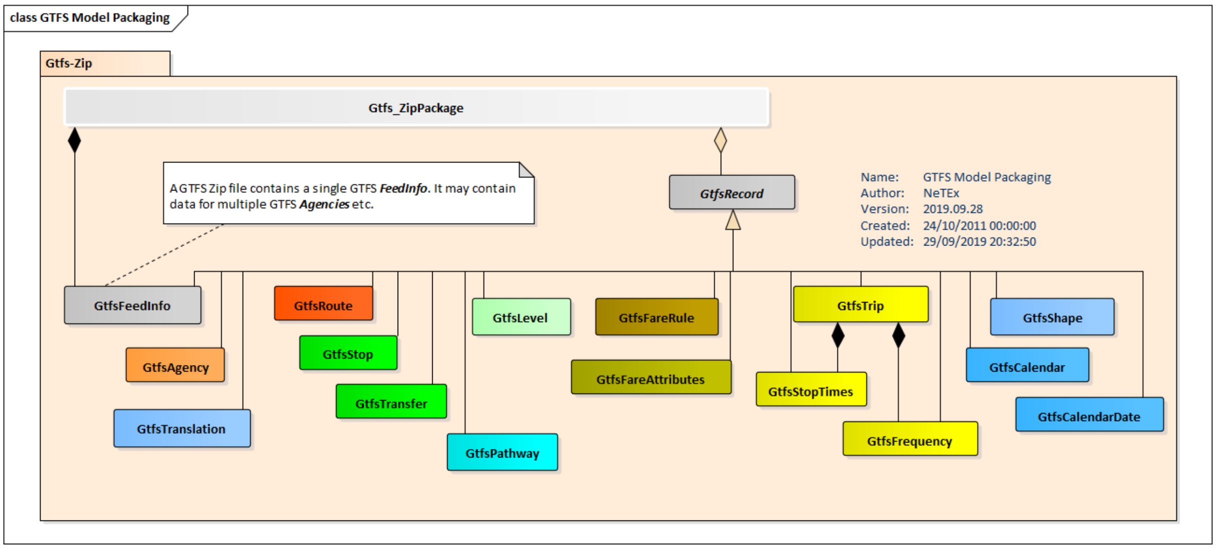
Figure 20 GTFS Model Hierarchies



5.2.2.1 Packaging a GTFS timetable

A GTFS data set comprises a set of individual CSV tables. To group them together for exchange as a coherent set with common versioning properties, they are placed in a zip file along with a GTFS ***Feed_info*** record with a single instance, describing the feed properties. This grouping is shown in Figure 21.

Figure 21 GTFS Model Packaging

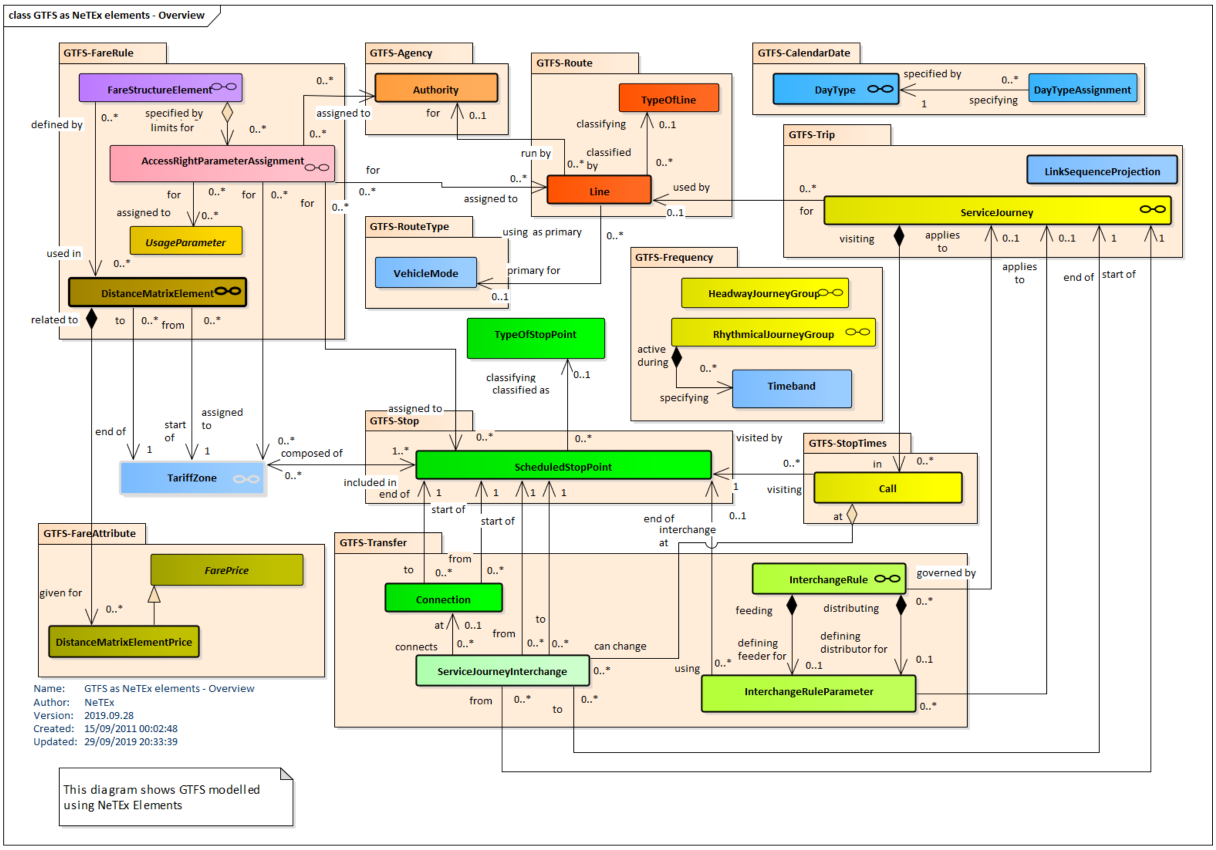


5.3 Overview of mapping of GTFS to Transmodel/ NeTEx

5.3.1 Transmodel/NeTEx elements equivalent to GTFS – Outline

Figure 22 summarises the core Transmodel/NeTEx elements needed to represent the GTFS model (it does not show every entity needed). Some correspondences are one-to-one, in other cases, the Transmodel representation is normalised into distinct elements, or spread over a supertype and a subtype (so that common properties can be reused in different subdomains), such that that several separate Transmodel/NeTEx elements are used together to represent a given GTFS element. This is indicated by a package showing the boundaries of the GTFS equivalent.

Figure 22 NeTEx elements equivalent to GTFS – Overview



5.3.1.1 Transmodel terminology

A GTFS/Transmodel comparison provides a good illustration of the importance of defining terminology precisely when comparing systems. In vernacular usage, many terms are used loosely in different contexts, so different words may be used for the same concept (e.g. “trip”, “journey”, etc), or even more confusingly, the same term may be used for what are in effect different concepts. For example, in colloquial English, “route” might be used to describe variously (i) the physical track of a specific vehicle, (ii) a particular sequence of stops followed in a specific direction by a vehicle, (iv) the sequence of stops in both directions traversed regularly by a public transport service; or (iv) the set of paths for a public transport journey, with variants, that are marketed under a common name, etc. There may be further variation arising from regional differences of dialect (e.g. American English “Agency” versus British English “Authority” or “Operator”). Transmodel aims to use consistently a separate, specific term for each concept, with official translations into multiple languages.

Sometimes these terms correspond readily to their colloquial usage (e.g. LEVEL), in other cases they are somewhat artificial so as to make the necessary distinctions from other highly specialised concepts (for example, RHYTHMICAL JOURNEY GROUP vs HEADWAY JOURNEY GROUP).

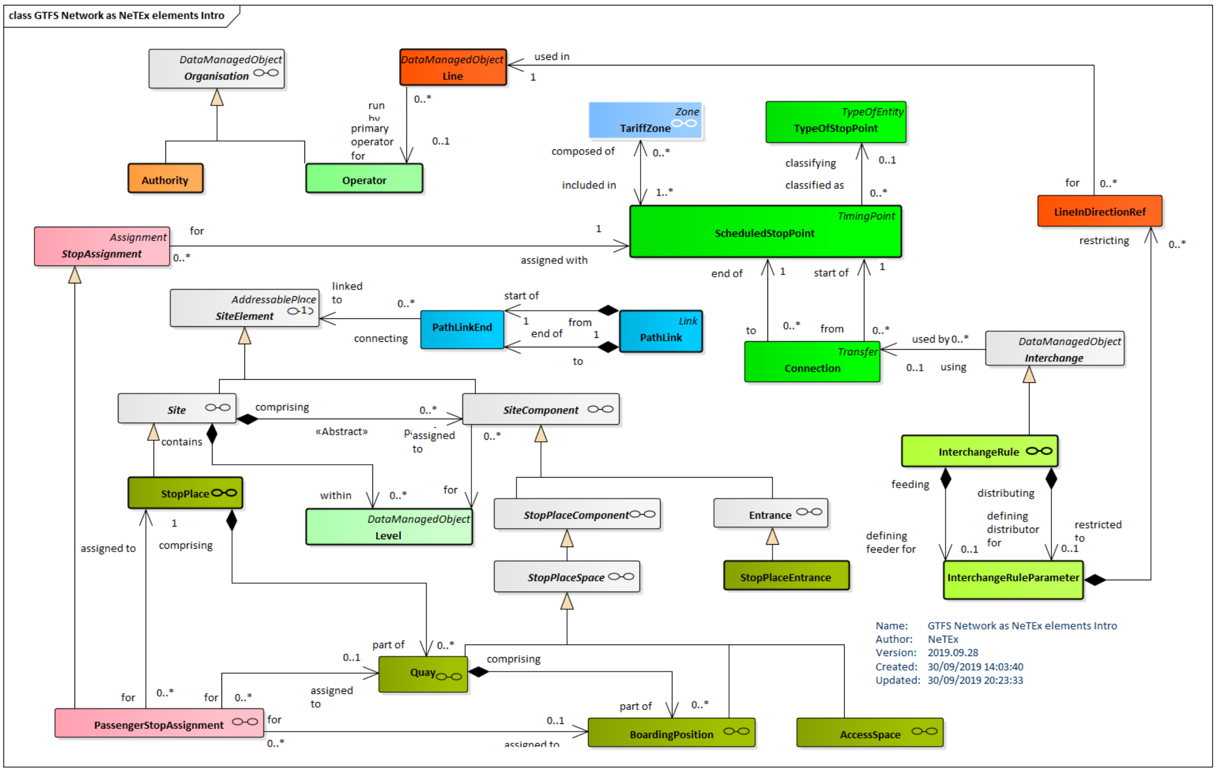
Several striking examples can be found in the GTFS to Transmodel mapping given later below of how Transmodel uses a restricted terminology to clarify the use of concepts, for example:

- A GTFS ***Route*** is a Transmodel LINE (Transmodel reserves the term ROUTE for the spatial path of a vehicle, and distinguishes a JOURNEY PATTERN – the sequence of stops in a given direction – from the concept of a named set of journey patterns in either direction – a LINE).
- A GTFS ***Trip*** is a Transmodel SERVICE JOURNEY (Transmodel reserves the term TRIP for the journey made by the passenger, while a JOURNEY or VEHICLE JOURNEY is a journey made by the vehicle. Transmodel further distinguishes between SERVICE JOURNEYs (which carry passengers) and DEAD RUNs (which don't), and between an individual SERVICE JOURNEY at a specific time and a TEMPLATE SERVICE JOURNEY, representing multiple service journeys at a specified frequency).
- A GTFS ***Service*** corresponds to two separate Transmodel SERVICE CALENDAR concepts: a DAY TYPE, characterising a day independently of any specific date, and a DAY TYPE ASSIGNMENT relating the DAY TYPE to a specific date in the calendar.
- A GTFS **Fare rule** corresponds to a Transmodel FARE STRUCTURE ELEMENT, in that it describes access rights to part of the network – however it also combines several other different concepts that are articulated separately in Transmodel so that they can be reused in different ways (e.g. DISTANCE MATRIX ELEMENT, covering specific access to network elements, and an ACCESS RIGHT ASSIGNMENT covering other access and usage factors).

5.3.1.2 GTFS Network as NeTEx elements

Figure 23 shows just the NeTEx elements needed to represent the GTFS elements that represent the fixed Network (i.e. GTFS ***stops***, GTFS ***transfers***, GTFS ***routes***, etc.), shown previously in Figure 16. Abstract elements are greyed out.

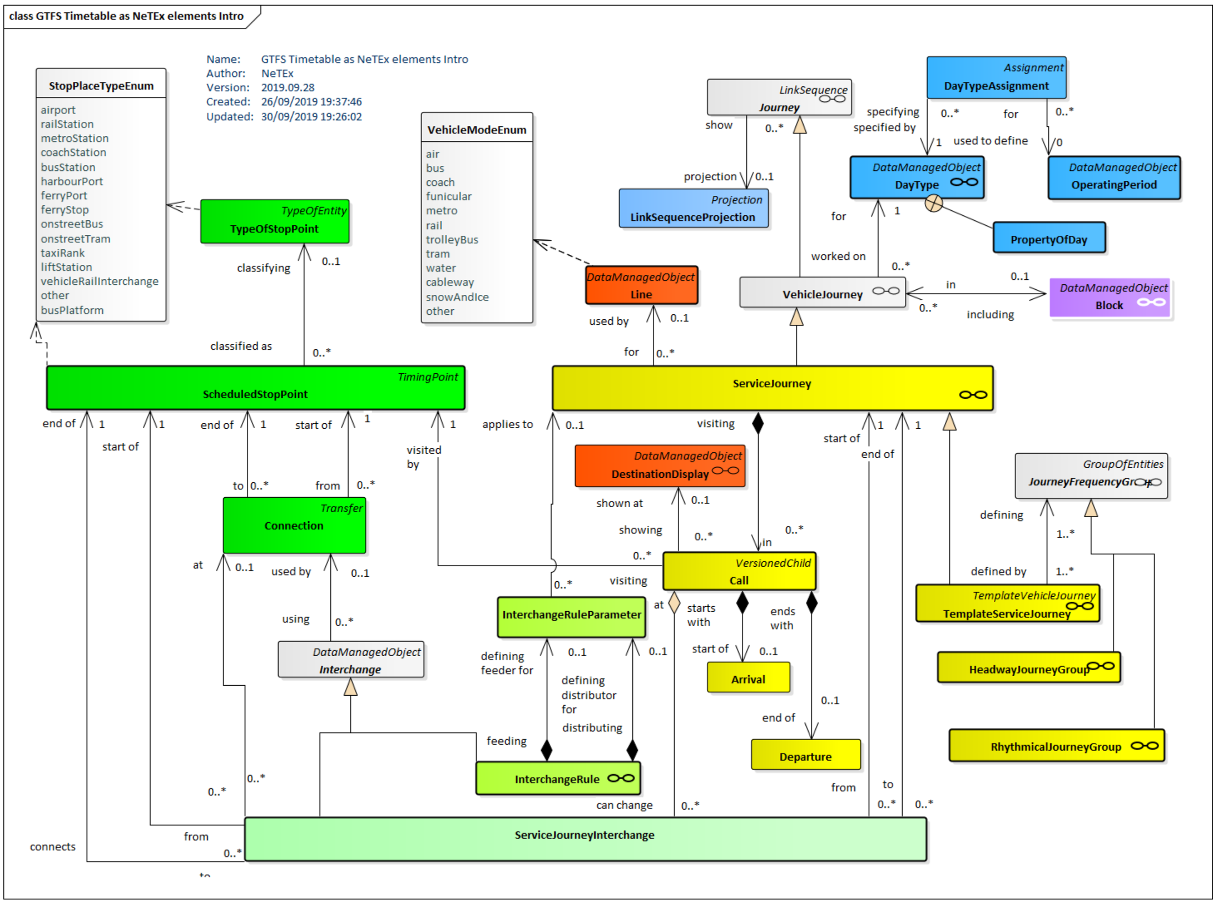
Figure 23 NeTEx elements equivalent to GTFS Network – detail



5.3.2.1 GTFS Timetable as NeTEx elements

Figure 24 shows the Transmodel/NeTEx elements covering equivalent function to the GTFS timetable model (i.e. GTFS ***trips***, GTFS ***stop_times***, GTFS ***calendar***, etc.) shown earlier in Figure 17.

Figure 24 NeTEx equivalents to GTFS Timetable elements

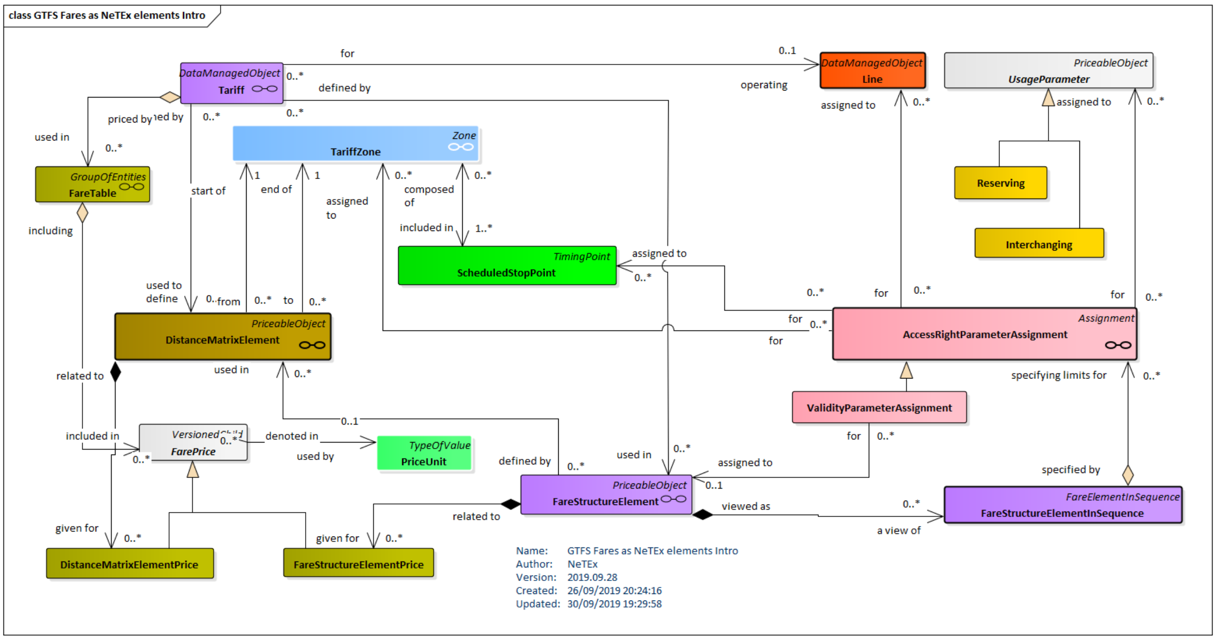


5.3.2.2 GTFS Fare as NeTEx elements

Figure 25 shows the NeTEx and elements covering equivalent function to the GTFS fare model (GTFS ***fare_rules***, GTFS ***fare_attributes,*** etc.) shown earlier in Figure 18.

Note that the GTFS fare model has a quite limited expressiveness, supporting only the simple use of certain tariff structures (one-to-zone, point-to-point, named zones, and flat), with very little information on products or their conditions of use or availability. Transmodel/NeTEx uses a richer model that adds additional abstractions in order to separate concerns, so that in particular, prices, tariff structures, access rights, fare products and sales offers (used to package up elements in different combinations for sale) are separate, reusable concepts. (Most of these additional features are not supported by GTFS and do not appear in the figure below). This makes it possible to describe many additional types of fares and to include additional information and conditions as to their use and availability. See further discussion at the end of this paper.

Figure 25 NeTEx equivalents to GTFS Fare elements – Introduction

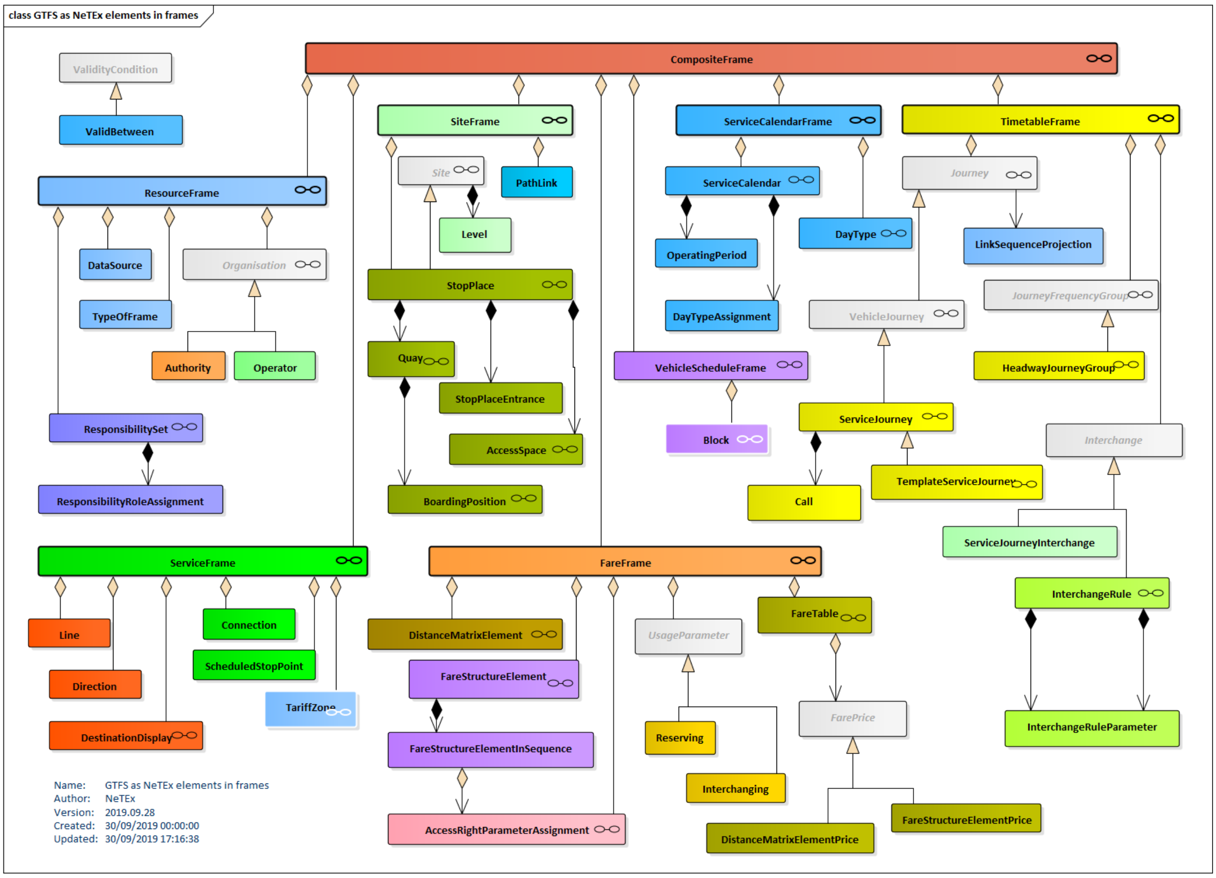


5.3.3 A GTFS zip as NeTEx elements in frame

As shown in Figure 21 above, GTFS packages a set of related data for exchange as a zip file containing separate CSV tables for each GTFS record. In contrast, NeTEx organises data as an XML document (or several documents, if desired), allowing data equivalent to many different types of GTFS record to be exchanged as a single file. Within the XML document, the different types of data content are grouped within specific types of Transmodel VERSION FRAME. For example, timetable (i.e. GTFS Trip) related elements are grouped in a TIMETABLE FRAME. Specific frames may themselves be grouped with a COMPOSITE FRAME with a single overall validity condition

Figure 26 shows the NeTEx elements equivalent to the GTFS model organised within version frames (other relationships between elements are not generally shown). Abstract elements have been greyed out.

Figure 26 NeTEx Frames of NeTEx elements to represent GTFS elements



5.4 Element-by-element detailed mapping of GTFS to Transmodel/NeTEx

The following diagrams take each GTFS element in turn and show its mapping to one or more equivalent NeTEx elements and attributes. For each GTFS element, a high-level view of the mapping without attributes followed by a detailed view with attributes is given. Additional NeTEx attributes that are not present in GTS are generally suppressed in the diagrams (in effect, this is the large majority of NeTEx attributes), as are additional NeTEx elements that are not supported in GTFS.

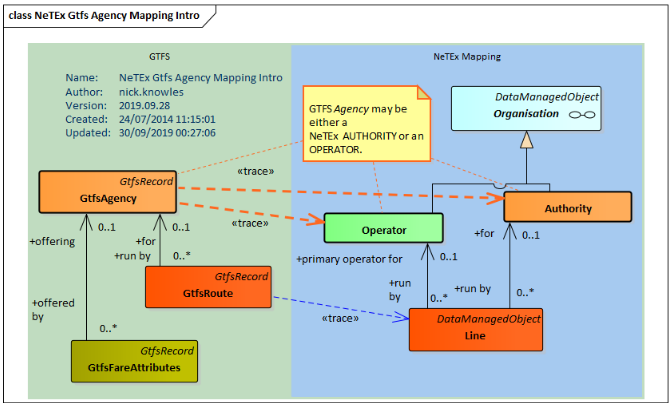
UML *trace* relationships (shown as dashed lines) are used to indicate the correspondence between GTFS and Transmodel/NeTEx elements.

5.4.1.1 GTFS Agency: NeTEx Mapping

5.4.1.1.1 GTFS Agency to NeTEx Mapping – Introduction

A GTFS **agency.txt** record (***GtfsAgency***) corresponds to a NeTEx transport ORGANISATION – which in Transmodel can be further specialised into an OPERATOR (Who runs a public transport service) and an AUTHORITY (who organises Public Transport for an area but doesn't necessarily actually operate services). Transmodel distinguishes between the two, but for most purposes they can be used interchangeably in the Transmodel model (Figure 27).

*Figure 27 GTFS **Agency** to NeTEx **Organisation** Mapping – Introduction*



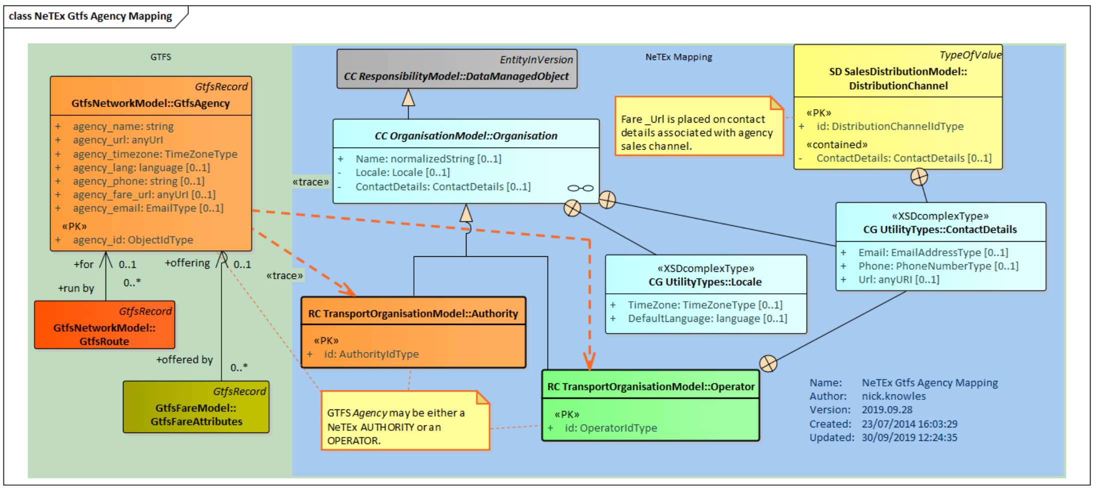
5.4.1.1.2 GTFS Agency to NeTEx Mapping – Details

A ***GtfsAgency*** maps to a Transmodel/NeTEx ORGANISATION; Transmodel further specialises ORGANISATION into different types, for example AUTHORITY and OPERATOR.

Equivalents to the detailed attributes of a ***GtfsAgency*** are found on NeTEx ORGANISATION; all the detailed attributes of ***GtfsAgency*** are common properties of a Transmodel/NeTEx ORGANISATION (Figure 28).

Note that the Transmodel representation also has a responsibility model (not shown) that allows the relevant responsibilities of different organisations to be characterised precisely (for example who originates and owns, data or operates different aspects of services). (in contrast. GTFS effectively assumes certain fixed responsibilities.)

*Figure 28 GTFS **Agency** to NeTEx **Organisation** Mapping – Details*



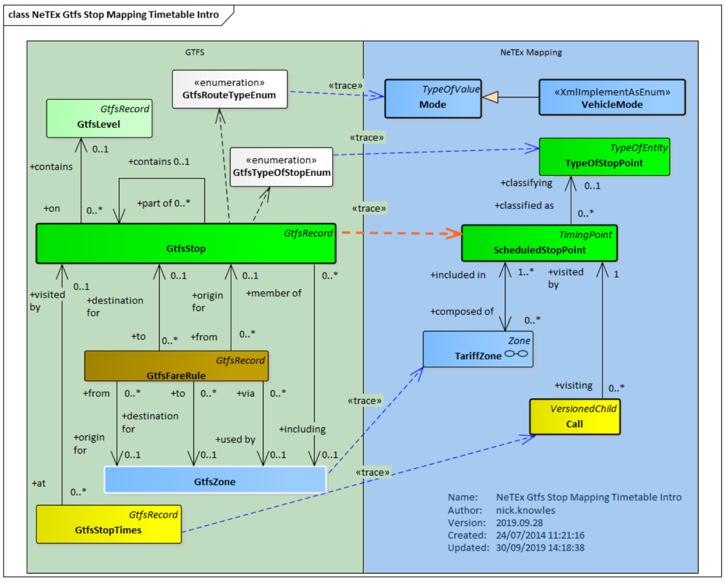
5.4.1.2 GTFS Stops: NeTEx Mapping

5.4.1.2.1 GTFS Timetabled Stop to NeTEx Mapping – Introduction

A GTFS **stops.txt** record ***(GtfsStop)*** includes both physical and timetabled stop concepts, corresponding variously to a Transmodel/NeTEx SCHEDULED STOP POINT (i.e. timetabled stop concept) and a STOP PLACE and or QUAY, (i.e. physical stop concept). The mapping of the timetable stop concept is one to one (Figure 29).

GtfsStop also describes the membership of the stop in a tariff zone for fares. Each ***GtfsStop*** may be assigned to a single ***GtfsZone*** for use in the model (though a ***GtfsZone*** is not actually reified as GTFS record, but merely referenced by ***GtfsStop*** and ***GtfsFareRule*** elements). If a stop belongs to multiple zones, then additional records are needed for each zone. Transmodel/NeTEx has a distinct TARIFF ZONE element and an individual SCHEDULED STOP POINT may be assigned to multiple TARIFF ZONEs.

*Figure 29 Timetabled GTFS **Stop** to NeTEx **ScheduledStopPoint** Mapping – Introduction*



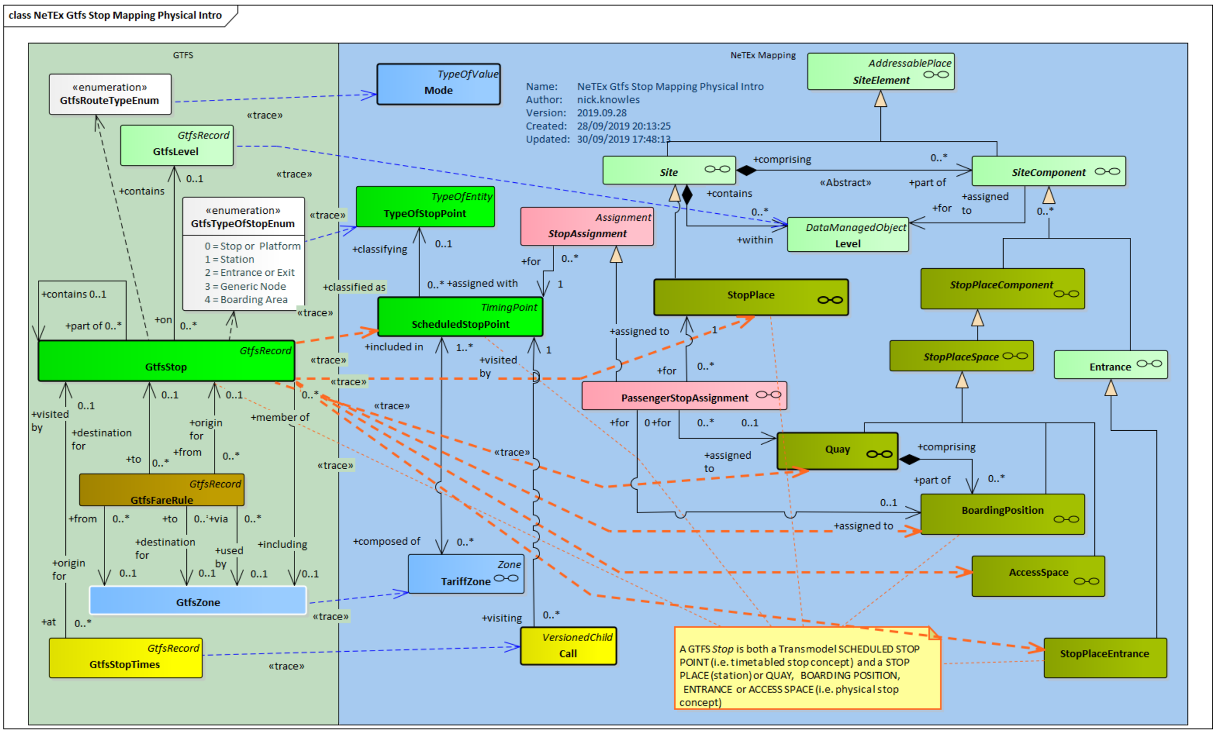
5.4.1.2.2 GTFS Physical Stop to NeTEx Mapping – Introduction

As well as being used for a timetabled stop, a ***GtfsStop*** can also represents physical stop concepts; a ***GtfsStop*** may represent either a whole station (GTFS ***location_type*** value. = "1") or a and an individual platform or quay by (GTFS ***location_type*** value ="0"). As a relatively recent enhancement to GTFS, a ***GtfsStop*** may also be used to represent the other IFOPT/Transmodel components of a physical stop; as indicated by the GTFS ***location_type*** enumeration; thus a GTFS "entrance" (Transmodel STOP PLACE ENTRANCE; "boarding area" (Transmodel BOARDING POINT); or "generic node") (Transmodel ACCESS SPACE) (Figure 30).

In Transmodel, the precise relationship between timetabled and physical stop elements can be described with a STOP ASSIGNMENT; so, for example, the platform allocated to a stop in a schedule may be changed. In NeTEx, if the identifier of the SCHEDULED STOP POINT and the STOP PLACE in QUAY is the same, the STOP ASSIGNMENT can be inferred implicitly and does not have to be stated. The physical stop (i.e. STOP PLACE elements, etc) are only needed for certain uses cases – e.g. in station navigation. For a simple timetable a SCHEDULED STOP POINT is sufficient.

In Transmodel/NeTEx it is also possible to build up hierarchies of stations representing a multimodal transport exchange, for example an airport with rail, metro, coach and bus stations.

*Figure 30 Physical GTFS **Stop** to NeTEx **StopPlace** Mapping – Introduction*

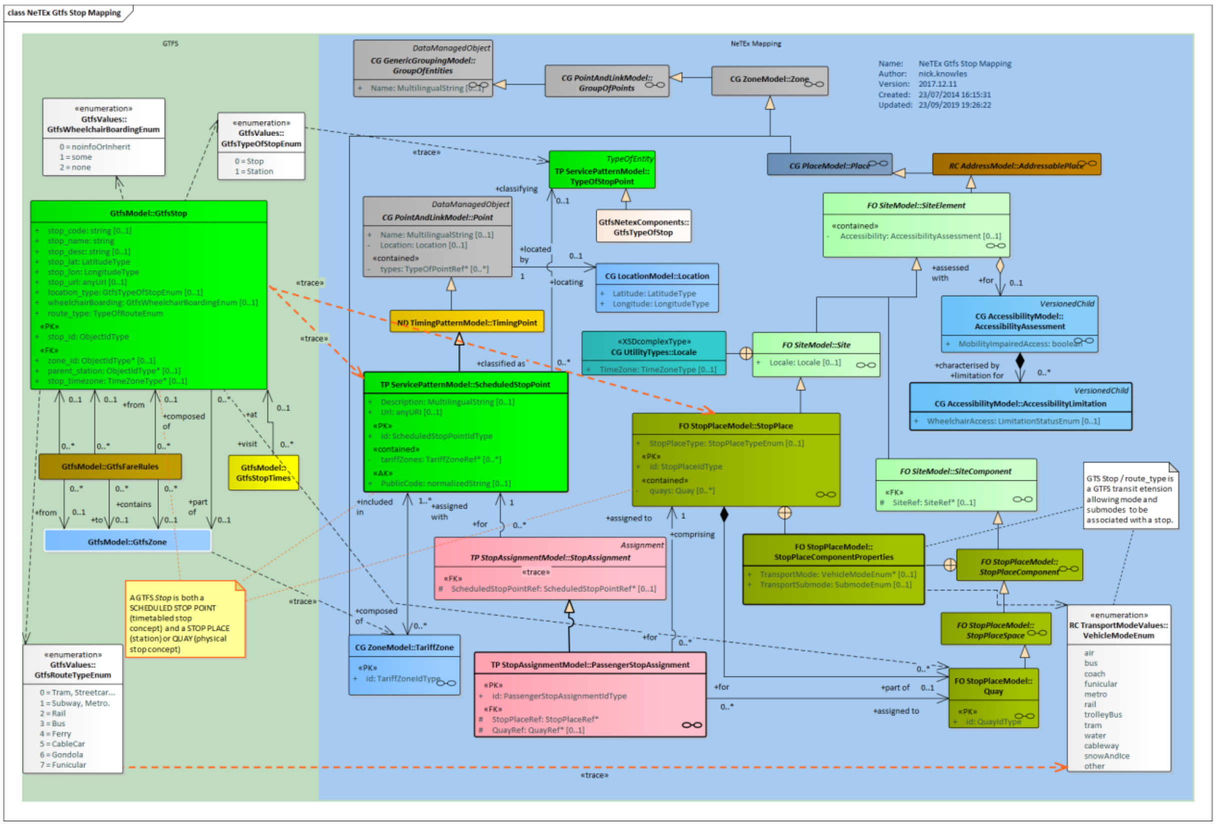


5.4.1.2.3 GTFS Stop to NeTEx Mapping – Details

GtfsStop attributes are mapped variously to SCHEDULED STOP POINT and STOP PLACE elements (Figure 31).

In Transmodel/NeTEx, a STOP PLACE is just one of a number of different types of SITE (others include POINTS of INTEREST and PARKING locations). SITEs and SITE COMPONENTs have common properties to locate them in space and to describe accessibility, facilities, equipment, etc. The ACCESSIBILITY ASSESSMENT and LOCALE properties (E.g. time zone and language) of a STOP PLACE and its parts (QUAY, ENTRANCE, etc) are thus inherited from SITE ELEMENT.

*Figure 31 GTFS **Stop** to NeTEx **StopPoint** and **StopPlace** Mapping – Details*



5.4.1.2.4 Overloading of GTFS Stop

Table 3 summarises the overloading of the ***GtfsStop*** concept with what in Transmodel/ NeTEx are separate concerns:

Table 3 Overloading of GTFS Stop Concepts

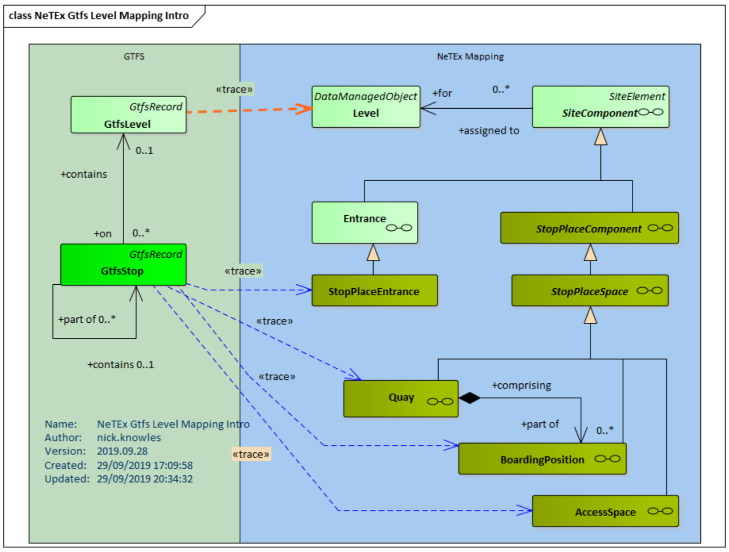
GTFS Usage	GTFS Stop location_type	GTFS Stop parent_station	Transmodel physical stop concept	Transmodel timetable stop concept
Whole Station	"1"	none	STOP PLACE	SCHEDULED STOP POINT
PT Access point / platform	"0"	required	QUAY	SCHEDULED STOP POINT
Entrance or Exit	"2"	required	ENTRANCE	SCHEDULED STOP POINT
Generic Node	"1"	required	ACCESS SPACE	SCHEDULED STOP POINT
Boarding Area		required	BOARDING POINT	SCHEDULED STOP POINT

5.4.1.3 *GTFS Levels: NeTEx Mapping*

5.4.1.3.1 GTFS Level to NeTEx Mapping – Introduction

A GTFS **levels.txt** record ***(GtfsLevel)*** is a physical stop concept corresponding one-to-one with a Transmodel/NeTEx LEVEL entity (Figure 32).

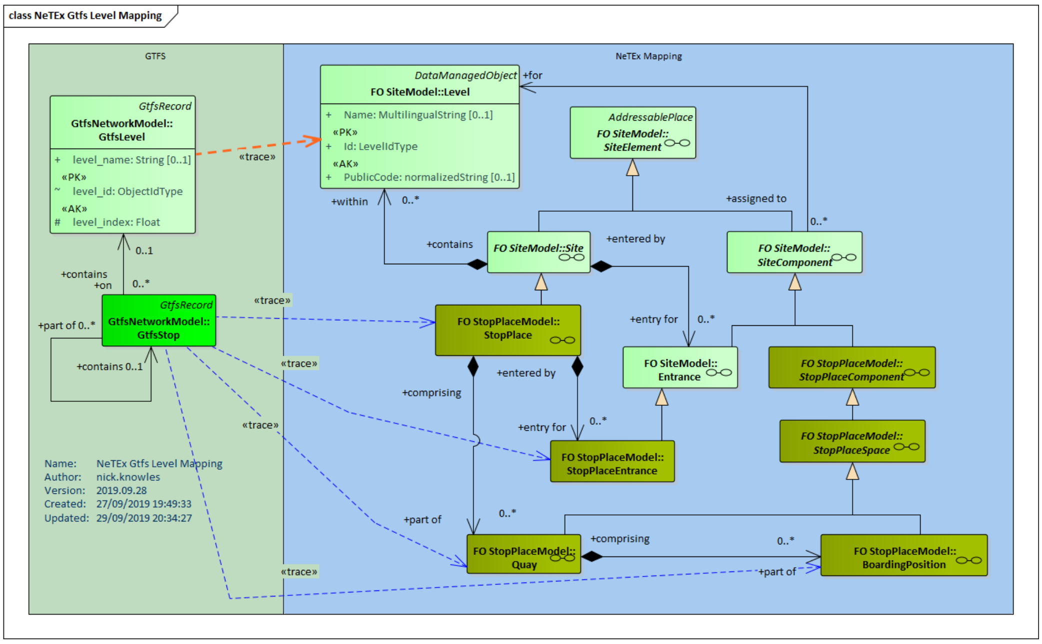
*Figure 32 GTFS **Level** to NeTEx **Level** Mapping – Introduction*



5.4.1.3.2 GTFS Level to NeTEx Mapping – Details

A ***GtfsLevel*** has a name and an index that is used both to label and to sequence the levels (Figure 33).

*Figure 33 GTFS **Level** to NeTEx **Level** Mapping – Details*

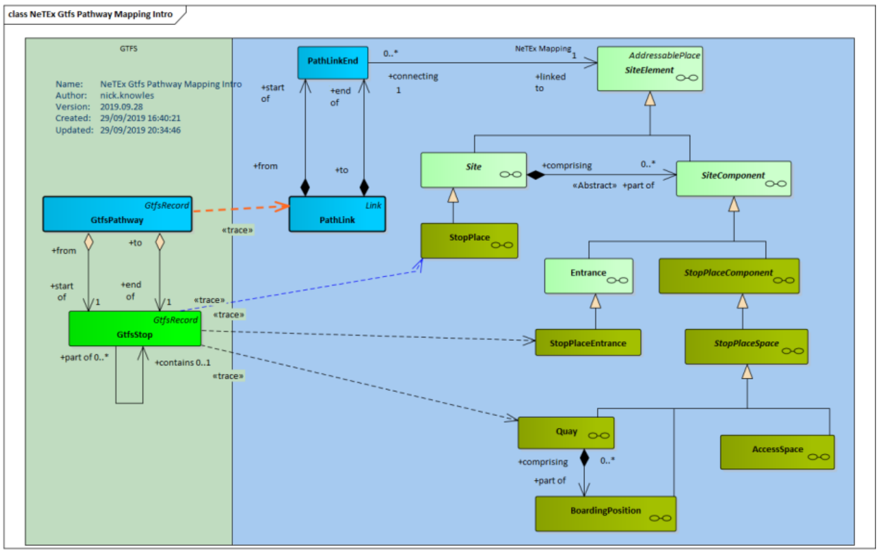


5.4.1.4 GTFS Pathways: NeTEx Mapping

5.4.1.4.1 GTFS Pathway to NeTEx Mapping – Introduction

A GTFS **pathways.txt** record (***GtfsPathway)*** is a physical stop concept connecting two subcomponents of a GTFS STOP (e.g. corresponding to a Transmodel/NeTEx PATH LINK entity ().

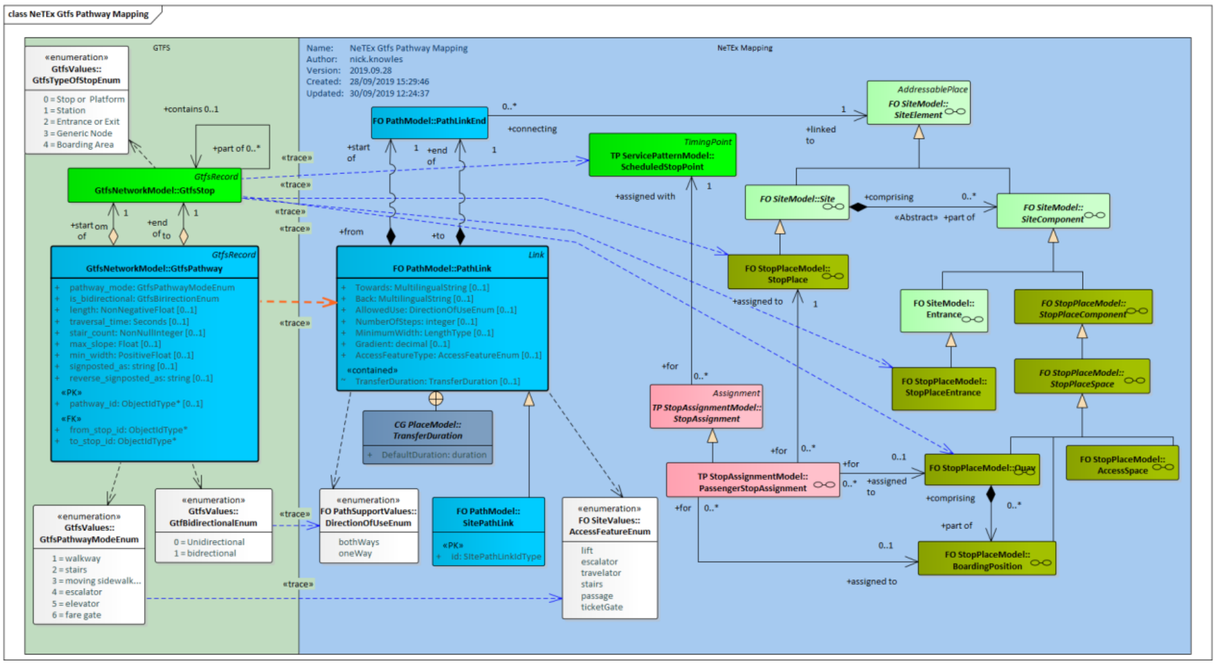
*Figure 34 GTFS **Pathway** to NeTEx **PathLink** Mapping – Introduction*



5.4.1.4.2 GTFS Pathway to NeTEx Mapping – Details

The detailed attributes of a GTFS ***Pathway*** (Figure 35) include a ***pathway_mode*** (NeTEx ***AccessFeatureType***) an ***is_bidirectional*** (NeTEx ***DirectionOfUse***) and ***signposted_as;*** (NeTEx ***Towards*** and ***Back*** attributes).

*Figure 35 GTFS **Pathway** to NeTEx **PathLink** Mapping – Details*



5.4.1.5 GTFS Transfers: NeTEx Mapping

5.4.1.5.1 GTFS Transfer to NeTEx Mapping – Introduction

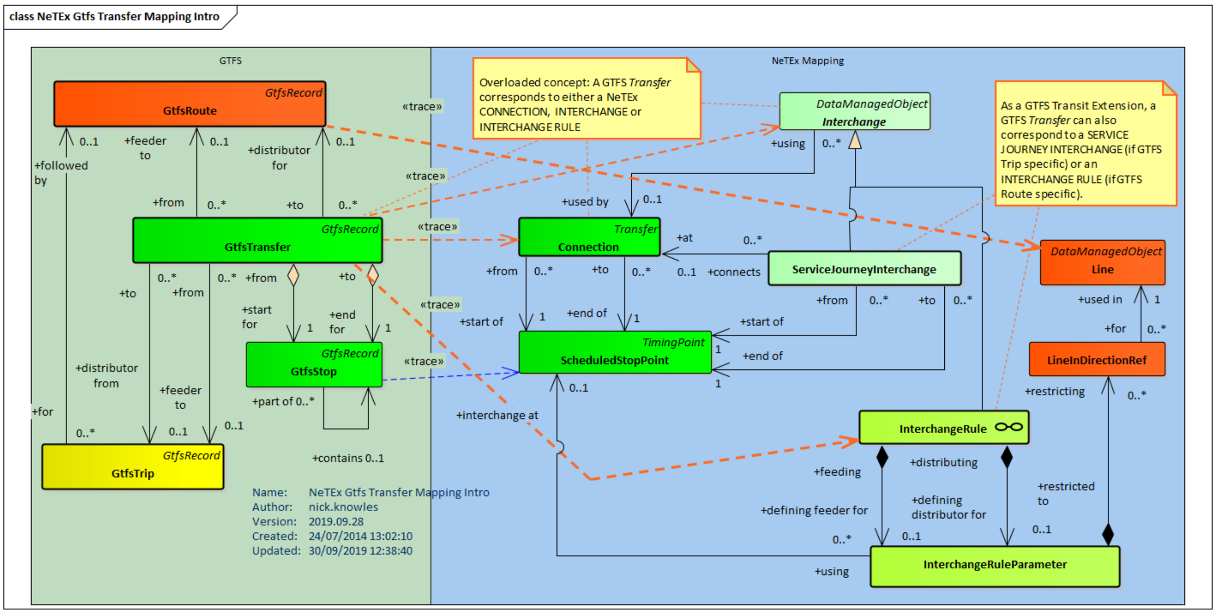
A GTFS ***transfers.txt*** record (***GtfsTransfer)*** includes both physical and timetabled connection concepts, and both timing and routing parameters, corresponding variously to the Transmodel/NeTEx CONNECTION, SERVICE JOURNEY INTERCHANGE and INTERCHANGE RULE entities (Figure 36).

The GTFS *Transit extensions* allow a further limiting of the applicability of the ***GtfsTransfer*** to specific ***GtfsRoute*** instances (i.e. Transmodel LINEs) or to specific pairs of feeder ***GtfsTrip*** and distributor ***GtfsTrip*** (i.e. Transmodel SERVICE JOURNEYs).

Transmodel, in contrast, uses different elements to represent the possibility of a connection between any two stops in the schedule (a Transmodel CONNECTION) and the time that should be allowed to make them, as opposed to an interchange just between two specific SERVICE JOURNEYs that service the two stops in close succession (a Transmodel SERVICE JOURNEY INTERCHANGE) – thus, different connection times, etc., may be specified for specific journeys. Rules that apply more generally to all the journeys of specific LINEs or routings may be described by an INTERCHANGE RULE.

Transmodel also allows generic defaults for transfers, filtered by, mode, operator, station, etc, to be set (not shown in the diagram as they not supported in GTFS) that can be used by trip planners when no specific timings are given (as is often the case).

*Figure 36 GTFS **Transfer** to NeTEx **Connection / Interchange** Mapping – Introduction*

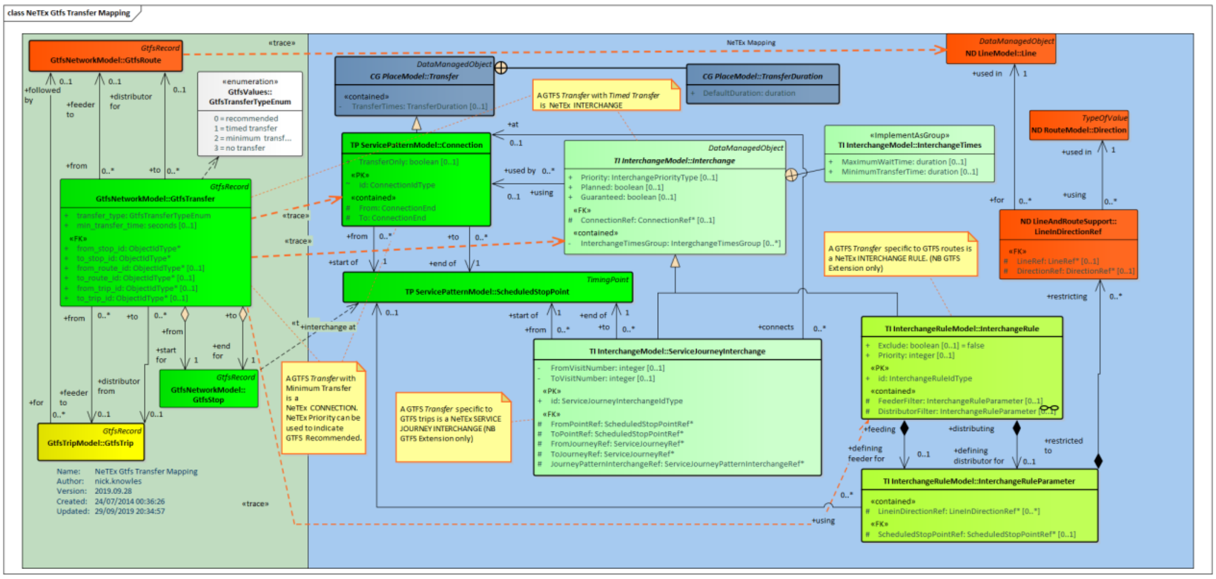


5.4.1.5.2 GTFS Transfer to NeTEx Mapping – Details

Figure 37 shows the mapping of detailed attributes for a ***GtfsTransfer*** (Figure 37).

Note: GTFS does not appear to handle the case of a SERVICE JOURNEY INTERCHANGE where the same VEHICLE visits the same stop twice with different attributes on each in separate interchange - since the ***from_trip_id*** and ***to_trip_id*** on a ***GtfsTransfer*** cannot distinguish between the two distinct ***GtfsTrip*** instances.

*Figure 37 GTFS **Transfer** to NeTEx **Connection / Interchange** Mapping – Details*



5.4.1.5.3 Overloading of GTFS Transfer

Table 4 summarises the overloading of the ***GtfsTransfer*** concept with what in Transmodel/NeTEx are separate concerns:

Table 4 Overloading of GTFS Transfer Concepts

GTFS Transfer Type	GTFS Possible Restriction	Transmodel / NeTEx mapping	Relevant NeTEx Attributes
Recommended=0	(Route)	CONNECTION + INTERCHANGE RULE	*InterchangeRule.priority +*
Timed Transfer=1	Route	CONNECTION + INTERCHANGE RULE + INTERCHANGE RULE PARAMETERs	*Interchange.MinimumTransferTime; Interchange.guaranteed; FeederFilter.InterchangeRuleParameter .LineInDirection.LineRef DistributorFilter.InterchangeRuleParameter .LineInDirection.LineRef*
Timed Transfer=1	Trip	CONNECTION + SERVICE JOURNEY INTERCHANGE	*Interchange.MinimumTransferTime ServiceJourneyInterchange .fromJourneyRef; ServiceJourneyInterchange .toJourneyRef;*
Minimum Transfer=2	Route	CONNECTION + INTERCHANGE RULE + INTERCHANGE RULE PARAMETERs	*Interchange.MinimumTransferTime; FeederFilter.InterchangeRuleParameter .LineInDirection.LineRef DistributorFilter.InterchangeRuleParameter .LineInDirection.LineRef*
Minimum Transfer=2	Trip	CONNECTION + SERVICE JOURNEY INTERCHANGE	*Interchange.MinimumTransferTime ServiceJourneyInterchange .fromJourneyRef; ServiceJourneyInterchange .toJourneyRef;*
No Transfer=3	(Route)	CONNECTION +	*Interchange.MinimumTransferTime; (FeederFilter.InterchangeRuleParameter .LineInDirection.LineRef*

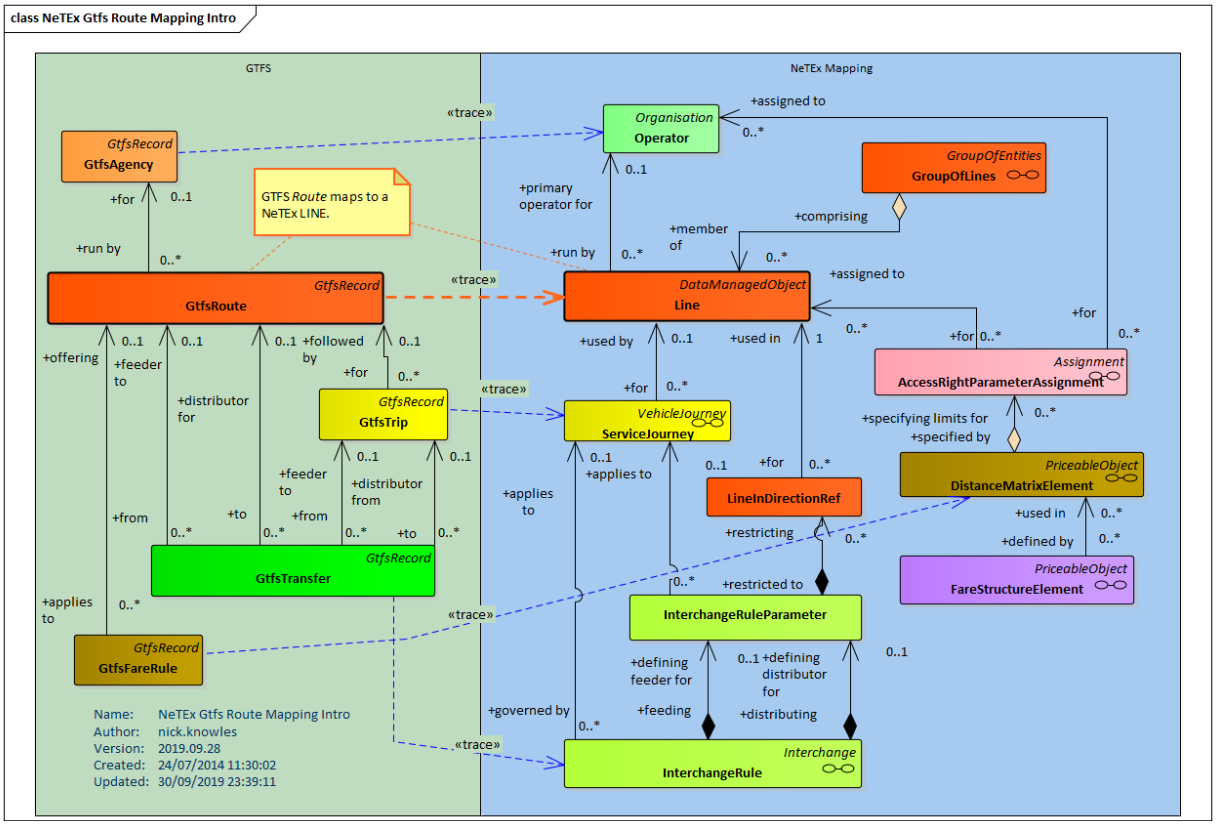
		INTERCHANGE RULE + INTERCHANGE RULE PARAMETERs	*DistributorFilter.InterchangeRuleParameter .LineInDirection.LineRef)*

5.4.1.6 GTFS Routes: NeTEx Mapping

5.4.1.6.1 GTFS Route to NeTEx Mapping – Introduction

A GTFS **routes.txt** record (***GtfsRoute)*** corresponds to a Transmodel/NeTEx LINE (Figure 38).

Figure 38 GTFS ***Route*** *to NeTEx* ***Line*** *Mapping – Introduction*

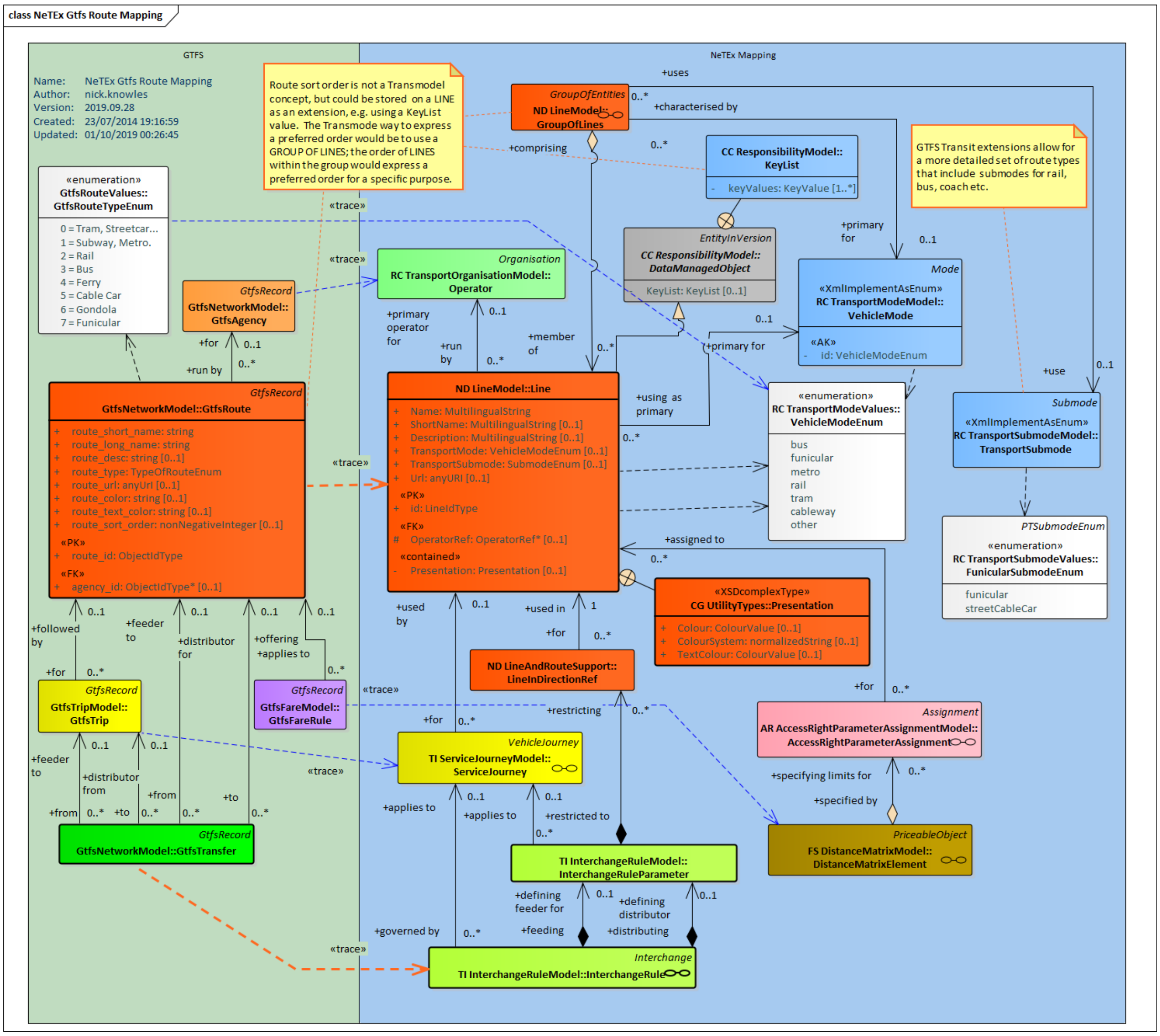


5.4.1.6.2 GTFS Route to NeTEx Mapping – Details

Detailed properties of a ***GtfsRoute*** include a GTFS ***route_type*** attribute, equivalent to a Transmodel MODE and or SUBMODE – see ***GtfsRouteType*** mapping later below (Figure 39).

A ***GtfsRoute*** can have a preferred ***route_sort_order*** to be specified to bias the order of presentation of LINEs in displays to the passenger. In Transmodel a GROUP OF LINES could be used to indicate the relative ordering of LINEs (as there might be different orders preferred for different use cases). In NeTEx it is also possible to add an extension value using the KeyList mechanism to hold arbitrary implementation attributes and this could also be used to add a sort order.

*Figure 39 GTFS **Route** to NeTEx **Line** Mapping – Details*



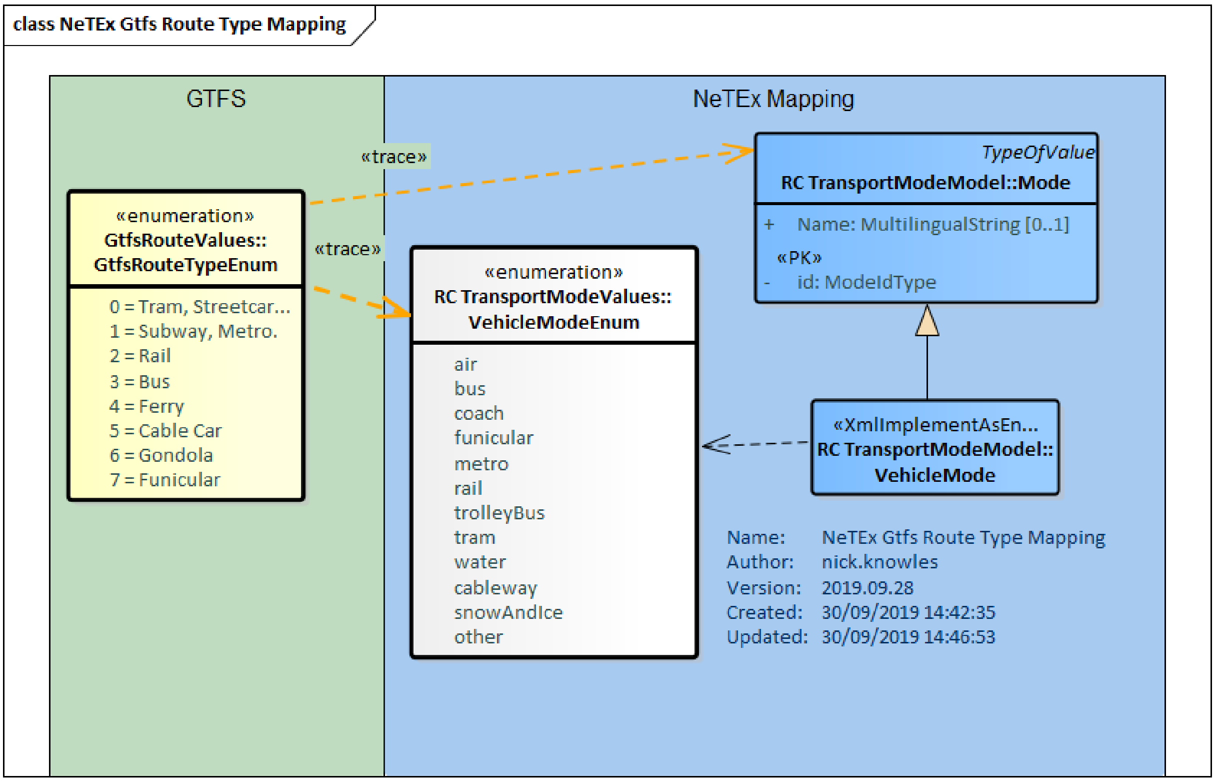
5.4.1.7 GTFS Route Types: NeTEx Mapping

5.4.1.7.1 GTFS Route Type to NeTEx Mapping – Introduction

The attributes of a ***GtfsRoute*** include a ***route_type*** (***GtfsRouteType),*** a set of enumerated values corresponding to a Transmodel/NeTEx MODE (and or SUBMODE). NeTEx also has an enumeration of fixed values that includes the GTFS values and provides a convenient implementation (Figure 40).

Note: Slightly curiously, GTFS does not distinguish between Coach and Bus MODEs – though the Google *Transit Extensions* support the distinction – see below.

*Figure 40 GTFS **Route_type** to NeTEx **Mode** – Introduction*



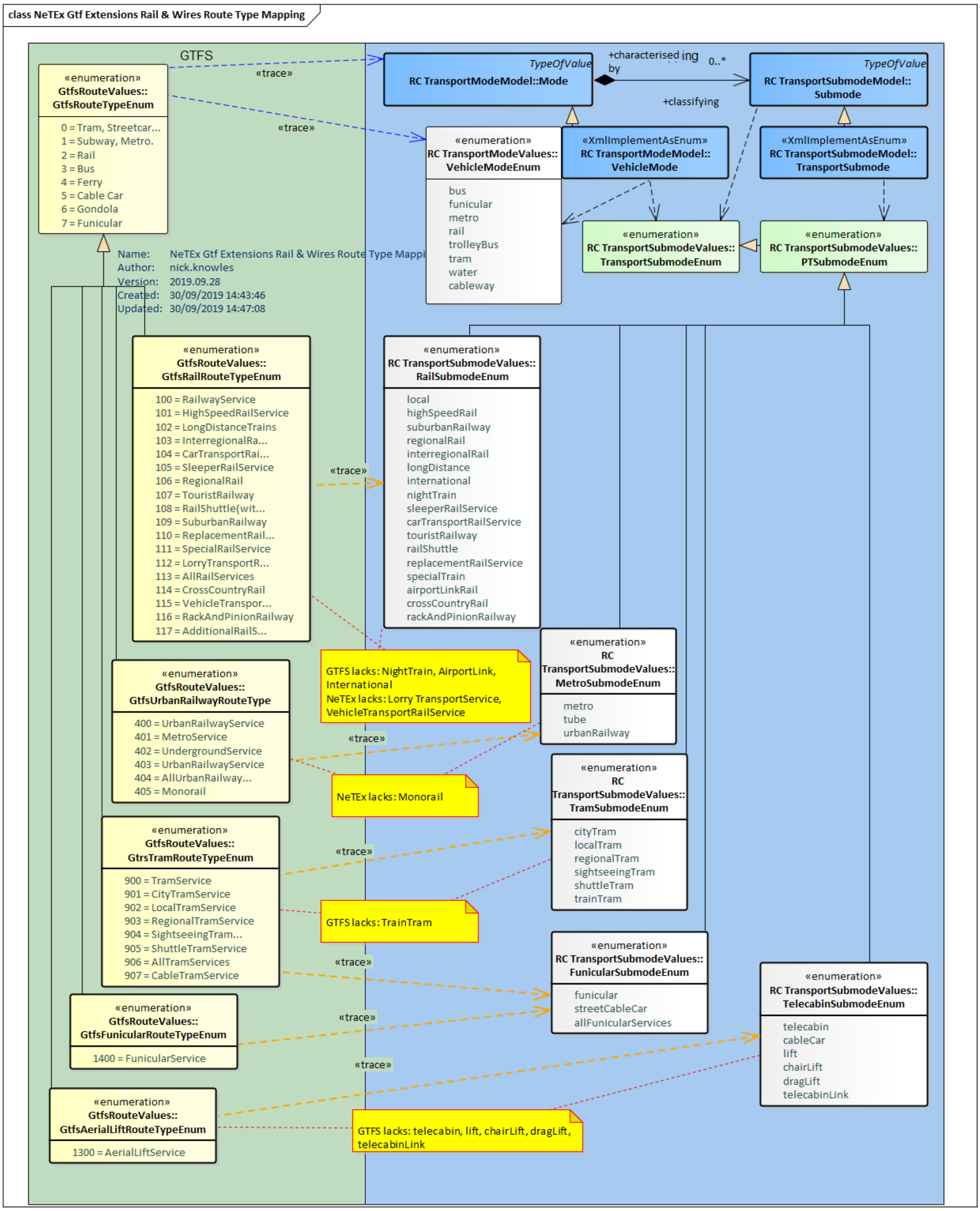
5.4.1.7.2 GTFS Extended Route Type to NeTEx Mapping – Rail & Wire Submodes

The GTFS *Transit Extensions* provide additional values for a ***GtfsRouteType,*** that further distinguish different types of route and correspond to a Transmodel/NeTEx SUBMODE (Figure 41).

NeTEx also has a set of enumerations of fixed values for SUB MODEs that include the GTFS values.

Figure 41 shows the mappings for Rail and other fixed path modes.

*Figure 41 GTFS Extended **Route_type** to NeTEx **Submode** – Rail & Wire submodes*

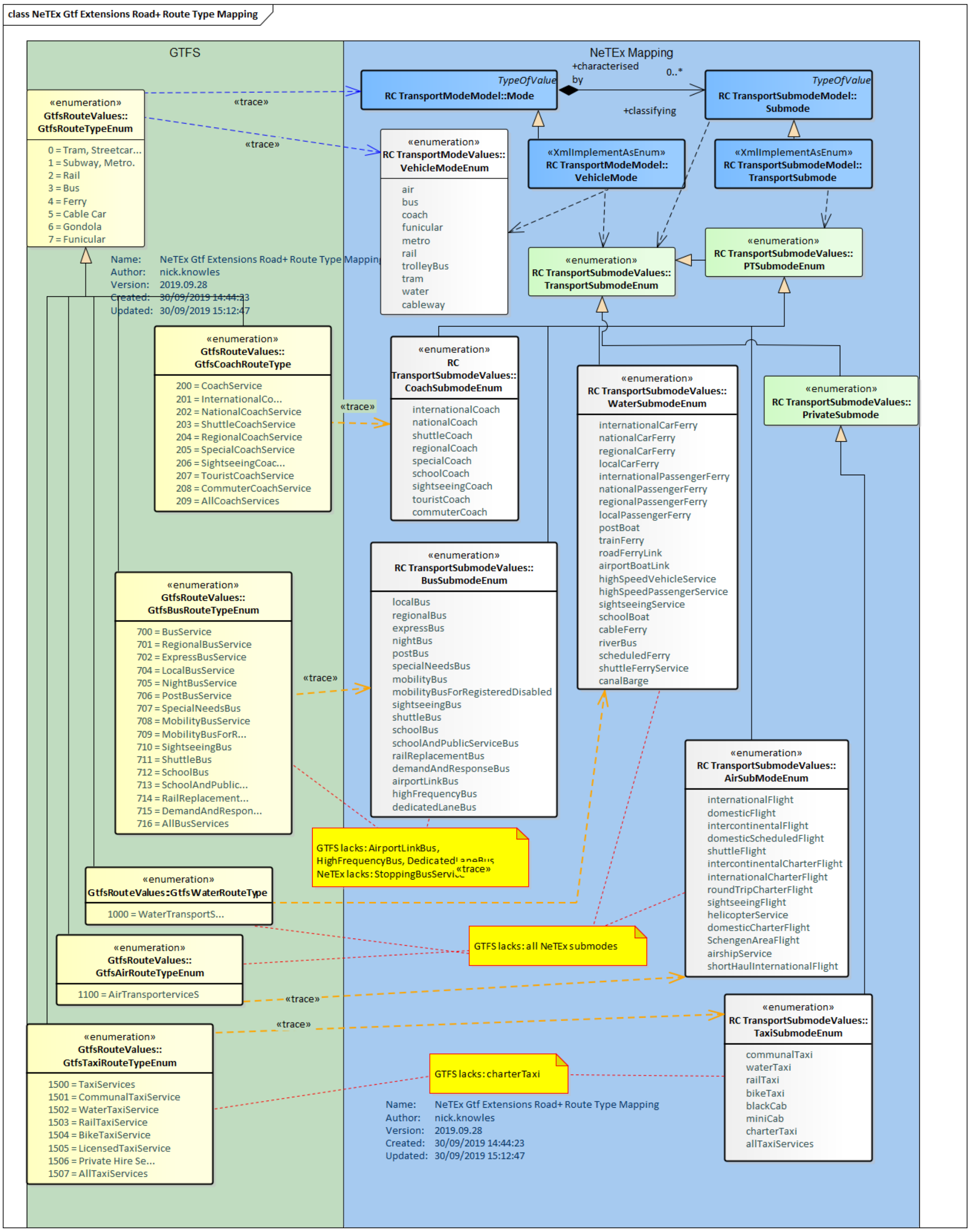


5.4.1.7.3 GTFS Extended Route Type to NeTEx Mapping – Road and other Submodes

The GTFS *Transit Extensions* provide additional values for a GTFS ***route_type*** that further distinguish different types of route and correspond to a Transmodel/NeTEx SUBMODE (Figure 42).

NeTEx also has a set of enumerations of fixed values for different groups of SUB MODEs that includes the GTFS values. Figure 41 shows the mappings for Road, ferry and other variable path modes.

*Figure 42 GTFS Extended **route_type** to NeTEx **Submode** - Road and other submodes*



a

5.4.1.8 GTFS Trips: NeTEx Mapping

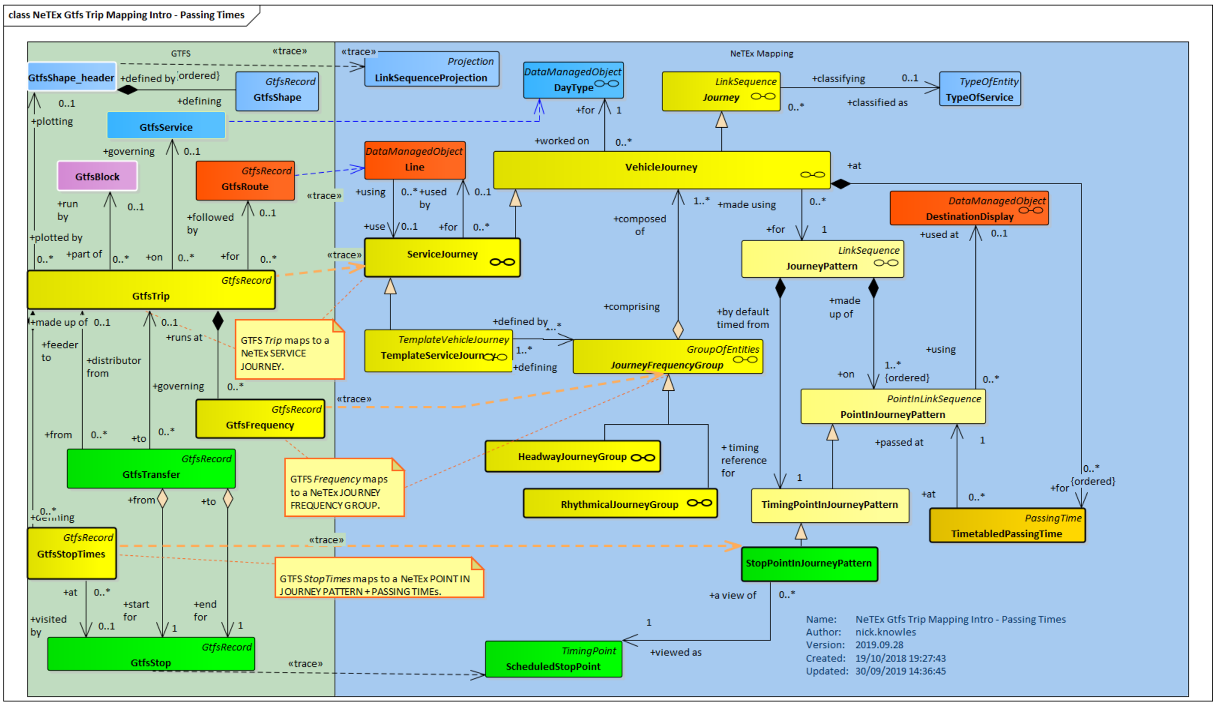
The primary component of a GTFS timetable is a ***Trip.*** The Transmodel/NeTEx equivalent is a VEHICLE JOURNEY

5.4.1.8.1 GTFS Trip to NeTEx Mapping Introduction – Passing Times

A GTFS **trips.txt** record (***GtfsTrip)*** corresponds to a Transmodel/NeTEx VEHICLE JOURNEY. More specifically: (i) an individual ***GtfsTrip*** running at a specific time corresponds to a SERVICE JOURNEY; (ii) a ***GtfsTrip*** running to a frequency as specified by a ***GtfsFrequency*** corresponds to a TEMPLATE SERVICE JOURNEY; the latter in effect defining a number of different SERVICE JOURNEYS running at a specified frequency.

The Transmodel/NeTEx representation also separates the SERVICE PATTERN of STOP POINTS in PATTERN from the PASSING TIMEs so that the same pattern can be used with different journeys. (in fact, the Transmodel/NeTEx representation also separates the TIMING PATTERN (not shown in diagram) from the SERVICE PATTERN so that different timings can be used with different SERVICE PATTERNs (Figure 43).

*Figure 43 GTFS **Trip** to NeTEx **ServiceJourney** Mapping using **Passing** Times– Introduction*

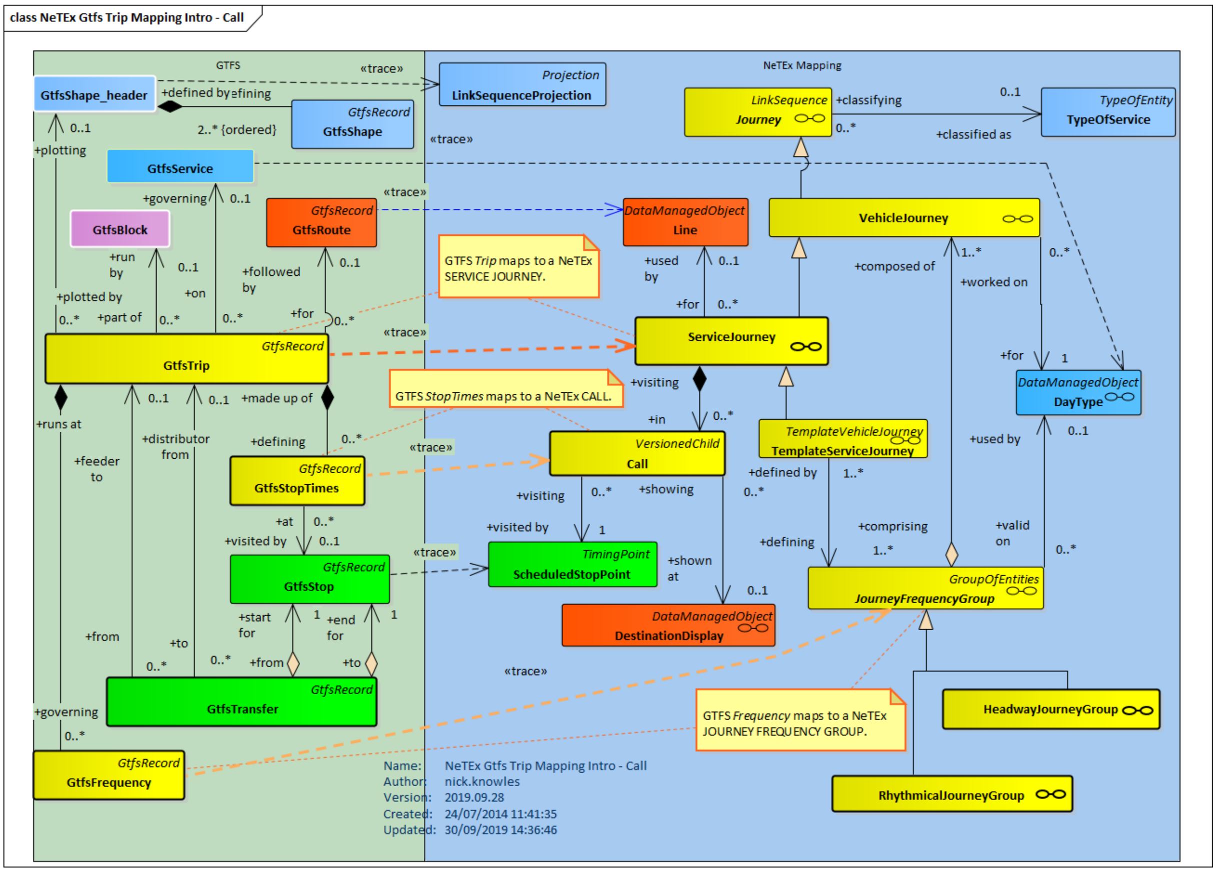


5.4.1.8.2 GTFS Trip to NeTEx Mapping Intro – using a Call

As a simplification for the implementation of timetable formats, NeTEx also supports the use of a CALL – a view element that assembles POINT IN JOURNEY PATTERN, PASSING TIMEs and other attributes into a single element that can be included in sequence in the description of a SERVICE JOURNEY instead of (or even as well as) separate POINT IN JOURNEY PATTERN, PASSING TIMEs, etc. A CALL gives a mapping very close to that of GTFS, since a CALL is largely equivalent to a ***GtfsStopTimes*** element (Figure 44)**.**

*Figure 44 GTFS **Trip** to NeTEx **ServiceJourney** Mapping using **Calls** - Introduction*

`



5.4.1.8.3 Overloading of GTFS Trip

Table 5 summarises the overloading of the ***GtfsTrip*** concept with what in Transmodel/NeTEx are separate concerns:

*Table 5 Overloading of GTFS **Trip** Concepts*

GTFS Usage	GTFS Trip timing	Transmodel Journey concept	Transmodel timing concept
Single Trip	***start_time***	SERVICE JOURNEY	DepartureTime
Trips running to a Headway frequency	***headway_secs***	TEMPLATE SERVICE JOURNEY	HEADWAY JOURNEY GROUP
Trips running at regular minutes past the hour	***exact_times***	TEMPLATE SERVICE JOURNEY	RHYTHMICAL JOURNEY GROUP

5.4.1.8.4 GTFS Trip to NeTEx Mapping – Details

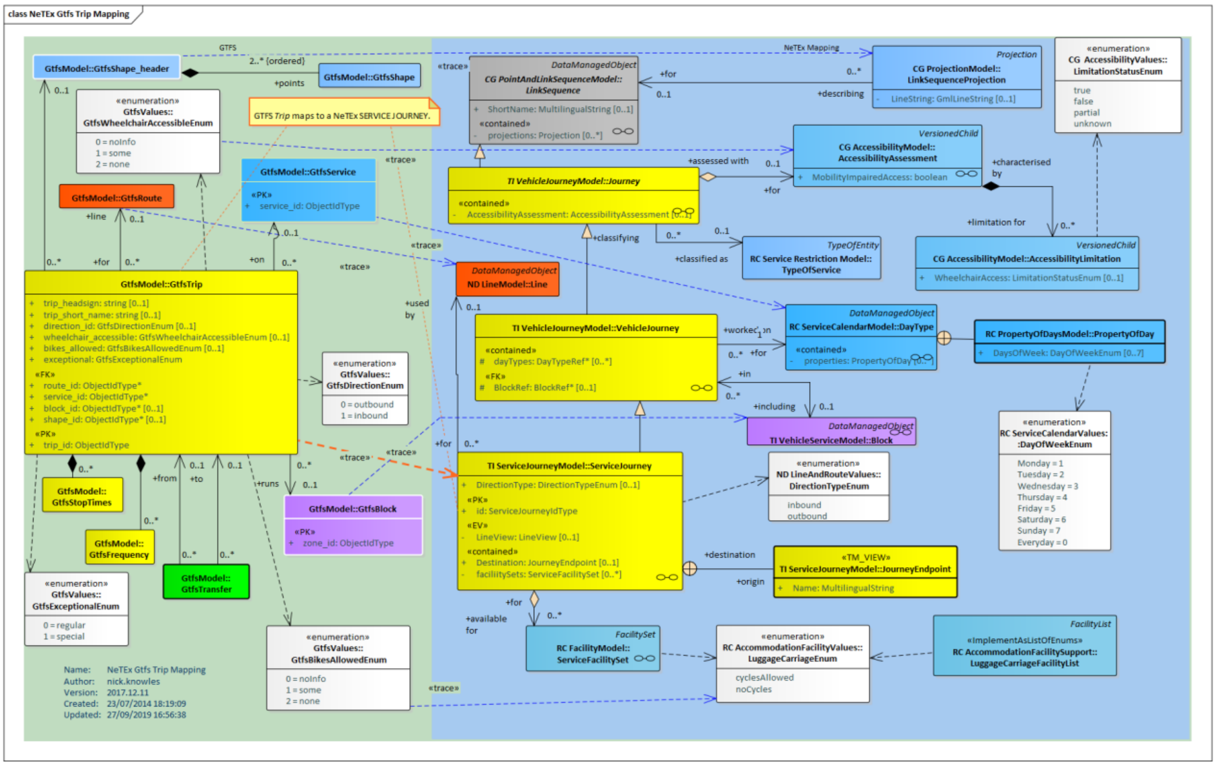
Unlike GTFS, Transmodel covers not just the passenger timetable, but also the planning and operational representations of the journey, so the mapping of a ***GtfsTrip*** to a Transmodel/NeTEx SERVICE JOURNEY, involves additional concepts (and enables additional capabilities):

(a) Transmodel describes planned journeys in the timetable as running on specific DAY TYPEs (for example "Monday to Friday", "Weekends", "Public Holidays", etc) rather than any specific dates. To arrive at an operational calendar, a specific calendar date is assigned to each day type and "dated" journeys additional defined with crews, vehicles etc. The ***GtfsService*** element (see later below) combines day type and calendar concepts.

(b) In a Transmodel, a SERVICE JOURNEY is just one type of VEHICLE JOURNEY – There are others, not shown in the diagram below, for example DEAD RUNs to position vehicles in place for service, that are part of the operational timetable as well and that can be described by Transmodel/NeTEx. GTFS covers only passenger information so a ***GtfsTrip*** corresponds to a SERVICE JOURNEY (Figure 45).

(c) A frequency-based journey that is described to the passenger in a timetable simply as a single journey at a given interval say "every five to then minute", is in fact operationally multiple journey carried out by different vehicles and crews. This is reconciled in the Transmodel/NeTEx model by describing such journeys in the timetable as TEMPLATE VEHICLE JOURNEYs, which act as placeholders for the actual DATED SERVICE JOURNEYs (not shown) which are used in the operational schedule. See ***GtfsFrequency*** later below.

*Figure 45 GTFS **Trip** to NeTEx **ServiceJourney** Mapping – Details*



5.4.1.9 GTFS Stop Times: NeTEx Mapping

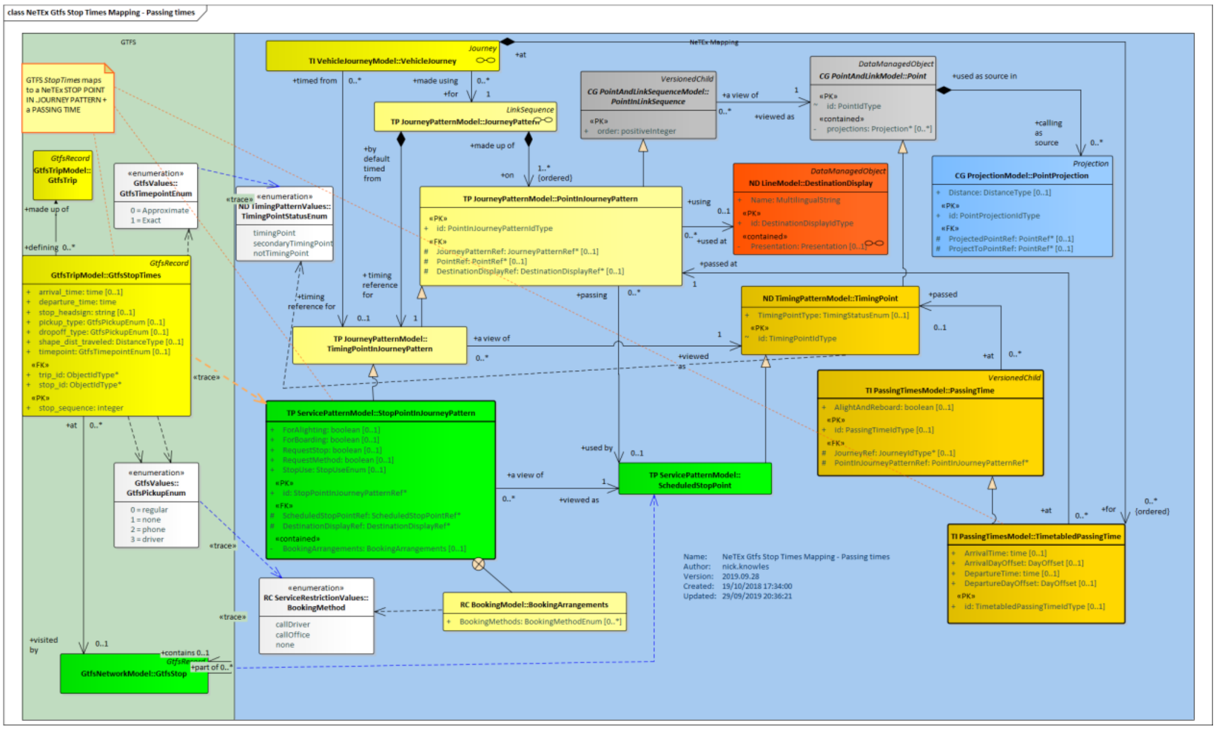
5.4.1.9.1 GTFS Stop Times to NeTEx Mapping: As Passing times – Details

The GTFS **stop_times.txt** record (***GtfsStopTimes)*** provides information about an individual visit to a stop in the course of a ***GtfsTrip***, i.e. NeTEx journey. A ***GtfsStopTimes*** can be mapped to separate Transmodel POINT IN JOURNEY PATTERN and PASSING TIME elements (Figure 46).

The Transmodel/NeTEx DESTINATION DISPLAY is a reusable version of the ***stop_headsign***. (allowing for constancy and efficient translation to other languages.

A Transmodel/NeTEx POINT PROJECTION (a general property of a shape) can be used to indicate the distance along the route plot (***GtfsShape***) that the stop lies. (Transmodel allows a separate detailed description of the route as points and links).

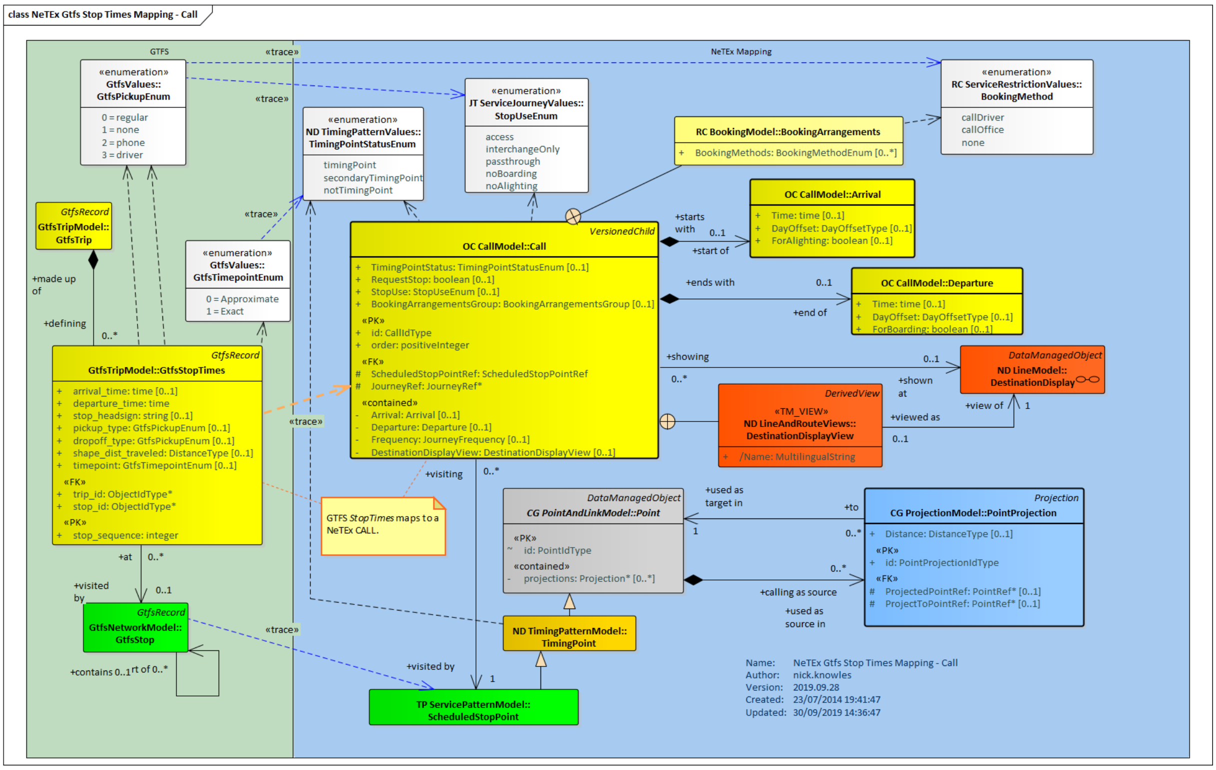
*Figure 46 GTFS **Stop Times** to NeTEx **PassingTimes** Mapping – Details*



5.4.1.9.2 GTFS Stop Times to NeTEx Mapping – As Call

An alternative simpler mapping of a ***GtfsStopTimes*** is to use a NeTEx CALL element. A CALL is a view element that assembles data attributes into a single element for convenience (Figure 47). The attributes of a CALL can all be derived through existing Transmodel relationships.

*Figure 47 GTFS **Stop Times** to NeTEx **Call** Mapping – Details*



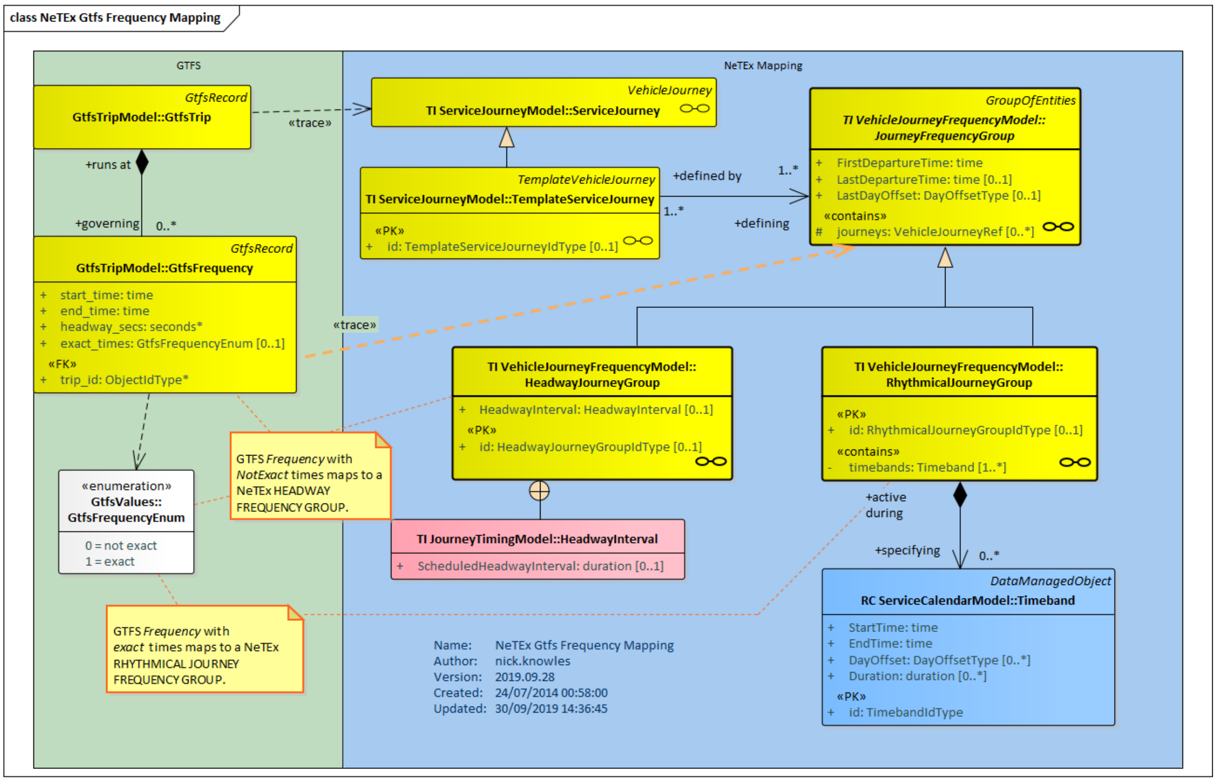
5.4.1.10 GTFS Frequencies: NeTEx Mapping

A GTFS **frequencies.txt** record ***(GtfsFrequency)*** provides interval times for frequency-based services.

5.4.1.10.1 GTFS Frequency to NeTEx Mapping – Details

The frequency data corresponds to that held by a Transmodel/NeTEx JOURNEY FREQUENCY GROUP, of which there are two specialisations; (i) if the ***GtfsFrequency*** is given as "***headway_secs***", then a Transmodel HEADWAY JOURNEY GROUP is use; (ii) if it is given as "***exact times***", then a Transmodel RHYTHMICAL JOURNEY GROUP is used (Figure 48).

*Figure 48 GTFS **Frequency** to NeTEx **JourneyFrequencyGroup** Mapping – Details*



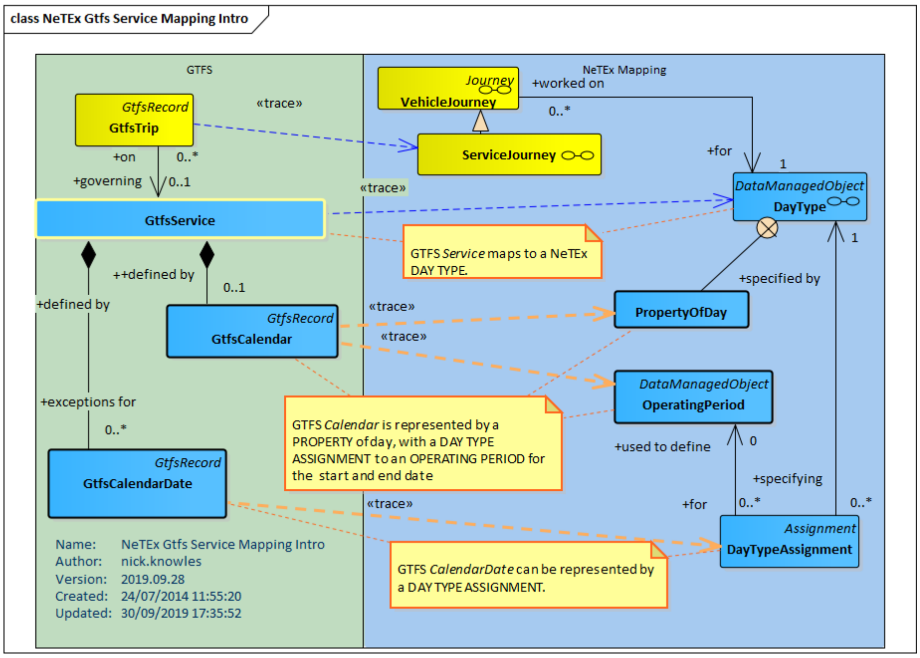
5.4.1.11 GTFS Service, Calendar and Calendar Dates: NeTEx Mapping

5.4.1.11.1 GTFS Service to NeTEx Mapping Intro

A GTFS **services.txt** record (Gtfs***Service)*** corresponds approximately to a Transmodel/NeTEx DAY TYPE and specifies temporal condition on a service. For example, “weekdays”. Transmodel in fact uses a more expressive model that separates planned and operational calendars; so that the validity of a temporal condition (i.e. when it starts and stops) is separate from the nature of the condition (day of week, holiday, season, etc). The Transmodel DAY TYPE is part of a reusable SERVICE CALENDAR concept that can be used to in different domains, for example to specify the availability of fare (which may have a different, more restricted, availability form that of the service itself, for example, for an off-peak fare).

A GTFS ***calendar_dates.txt*** record (***GtfsCalendarDate***) maps to a Transmodel/NeTEx DAY TYPE ASSIGNMENT that relates a DAY TYPE to a day in the calendar of a specific year (Figure 49).

*Figure 49 GTFS **Service** to NeTEx **DayType** Mapping – Introduction*



5.4.1.11.2 GTFS Service to NeTEx Mapping – Details

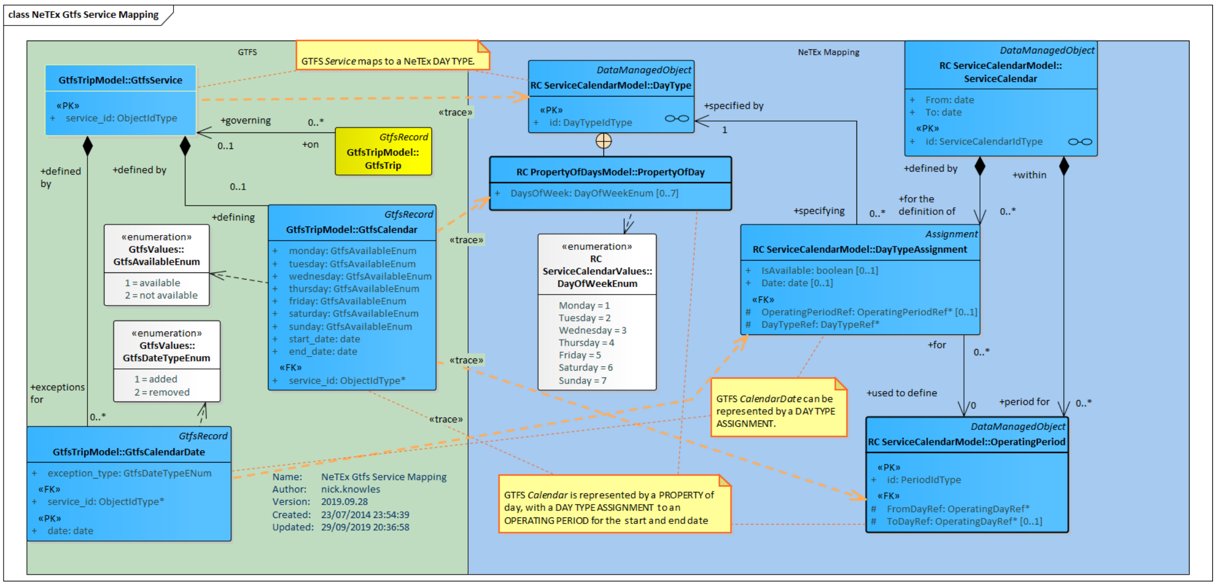
GtfsService aspects that are independent of calendar date (such as the day of week on a ***GtfsCalendar***) map to a Transmodel/NeTEx DAY TYPE and PROPERTY of DAY. ***GtfsService*** aspects that are limited to a specific calendar date map additionally to a DAY TYPE ASSIGNMENT and OPERATING DAY or OPERATING PERIOD (Figure 50).

A Transmodel/NeTEx SERVICE CALENDAR can be used to group multiple DAY TYPE ASSIGNMENTs and set a common start and end date for all elements (if there are multiple ranges on different GTFS Calendar elements, they can be specified as OPERATING PERIODs).

Note that NeTEx supports additional characterisations of PROPERTY OF DAY (“market day”, “match day”, etc.) that are not given in GTFS (not shown in diagram).

A Transmodel DAY TYPE does not have to be contiguous with a calendar day – for example it can run from 2am to 2am – it can be mapped to an OPERATIONAL DAY. (GTFS refers to this concept as a “Service day”) using a service calendar.

*Figure 50 GTFS **Service** to NeTEx **DayType** Mapping – Details*



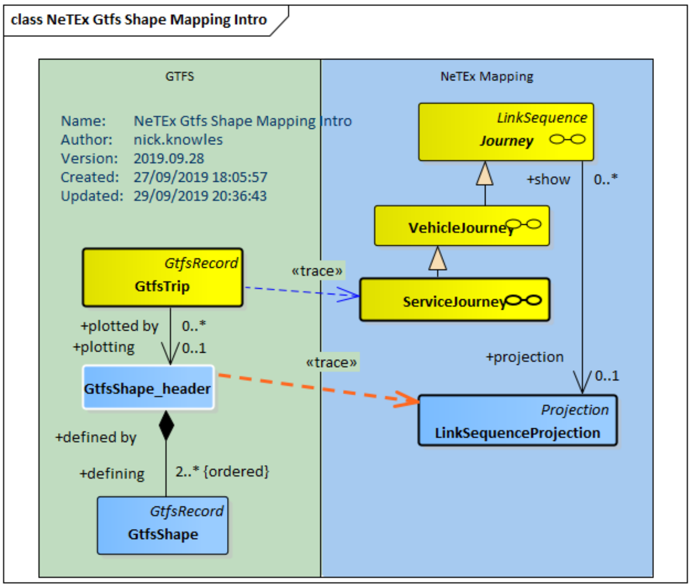
5.4.1.12 GTFS Shape to NeTEx Mapping

5.4.1.12.1 GTFS Shape to NeTEx Mapping Intro

A GTFS **shapes.txt** record (***GtfsShape)*** is a general ancillary element describing a spatial plot of any component as a sequence of points; it is used in GTFS to show the plot of a SERVICE JOURNEY. Each ***GtfsShape*** shape describes an individual point in a line; the line as a whole is not reified in the GTFS feed but is nonetheless modelled in this analysis by a ***GtfsShape_header*** element (Figure 51)***.***

The simplest mapping to NeTEx would be to embed an equivalent GML ***LineString*** on a LINK PROJECTION associated with the SERVICE JOURNEY.

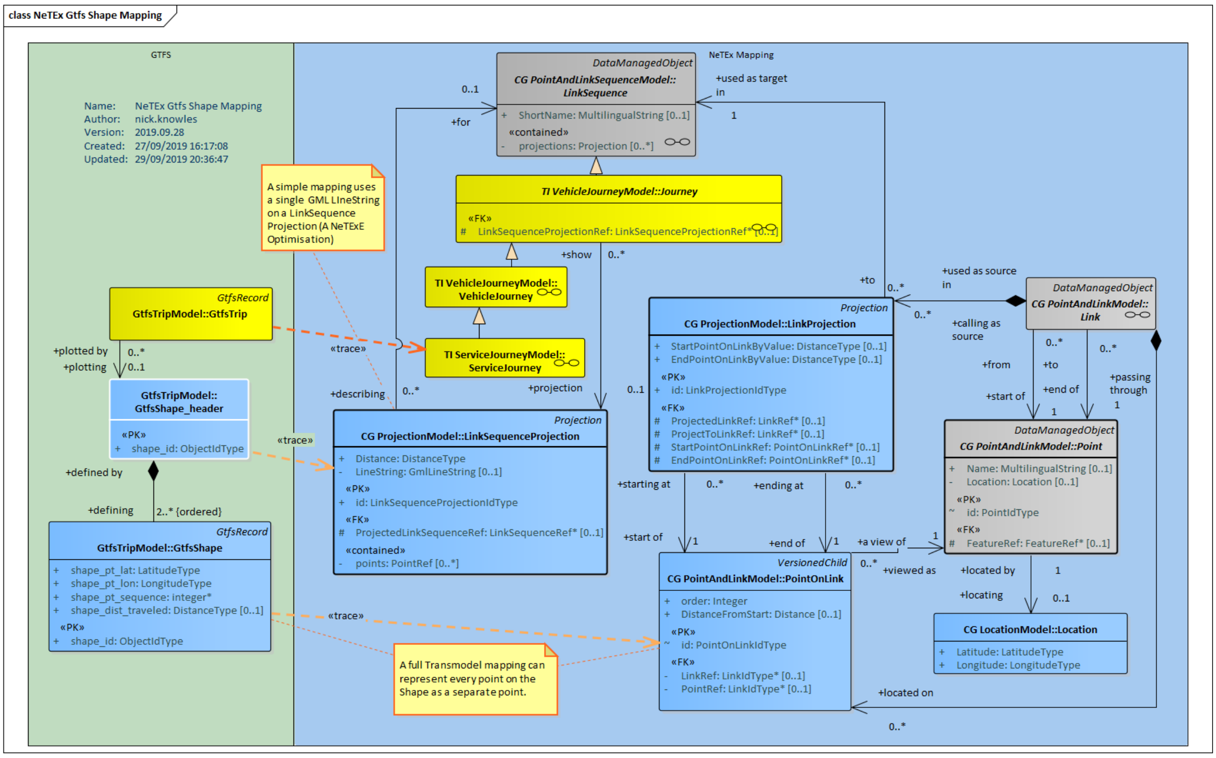
*Figure 51 GTFS **Shape** to NeTEx **Projection** Mapping – Introduction*



5.4.1.12.2 GTFS Shape to NeTEx Mapping details

A more elaborate mapping would map the ***GtfsShape*** approximately to a Transmodel/NeTEx DAY LINK PROJECTION with a POINT ON LINK for each GTFS SHAPE (Figure 52). This would allow the shape to be related to a persistent reusable spatial representation of the SERVICE PATTERN as points and links.

*Figure 52 GTFS **Shape** to NeTEx **Projection** Mapping – Details*



5.4.1.13 *GTFS Fare Rules: NeTEx Mapping*

Basic tariff structures are represented in GTFS by GTFS **fare_rules**, and the accompanying prices are represented with GTFS **fare_attributes**.

5.4.1.13.1 GTFS Fare Rule Mapping – Introduction

The GTFS **fare_rules.txt** record ***(GtfsFareRule)*** uses the same element in different ways to represent three different tariff structures - in effect “overloading” the semantics of the ***GtfsFareRule*** concept:

(a) Pairs of Origin/Destination ***GtfsStop*** instances or ***GtfsZone*** instances, each equivalent to a Transmodel/NeTEx DISTANCE MATRIX ELEMENT between Transmodel/NeTEx SCHEDULED STOP POINTs or TARIFF ZONEs.

(b) Allowed zones or sequence of ***GtfsZone*** instances that may be used. Equivalent to Transmodel/NeTEx FARE STRUCTURE ELEMENTs IN SEQUENCE, each restricted to specific zones with an ACCESS RIGHT PARAMETER ASSIGNMENT.

(c) The ***GtfsRoute*** that may be used, equivalent to a Transmodel/NeTEx FARE STRUCTURE ELEMENTs that has been restricted to a specific LINE by an ACCESS RIGHT PARAMETER ASSIGNMENT.

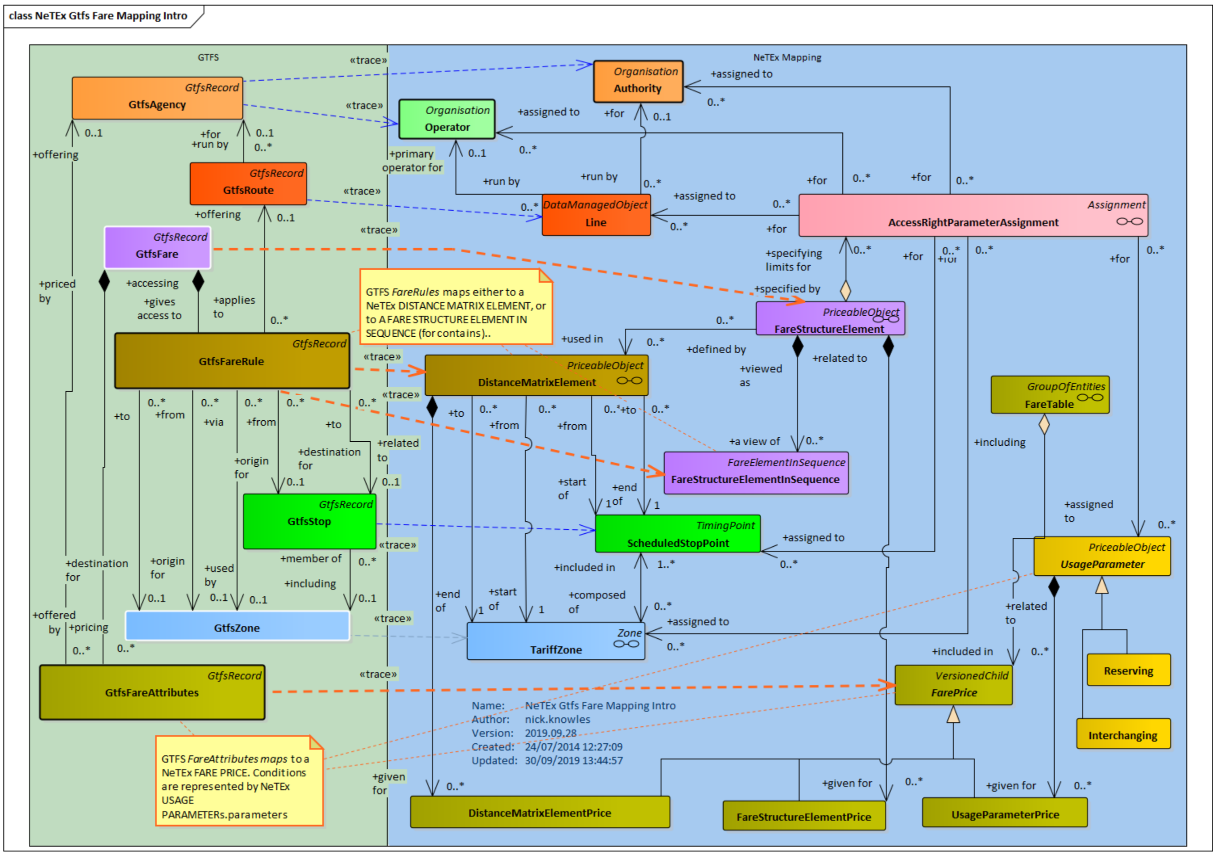
Thus, a ***GtfsFareRule*** corresponds to certain different combinations of Transmodel/NeTEx FARE STRUCTURE ELEMENT: for a point-to-point fare there will also be a one-to-one correspondence with a NeTEx/Transmodel DISTANCE MATRIX ELEMENT, used to describe the fare structure elements of a TARIFF. E.g. the point to point O/D pairs (Figure 53).

Note that in GTFS the ***GtfsZone*** instance (Transmodel/NeTEx TARIFF ZONEs) themselves are not reified as named elements, merely referenced.

Transmodel/NeTEx also supports a number of other tariff structures and fare products (e.g. stage fares, distance fares, season passes etc) as well as composite structures and products.

We Illustrate a mapping for each of the tariff structures separately below.

*Figure 53 GTFS **Fare_rule** to NeTEx **DistanceMatrixElement** Mapping – Introduction*

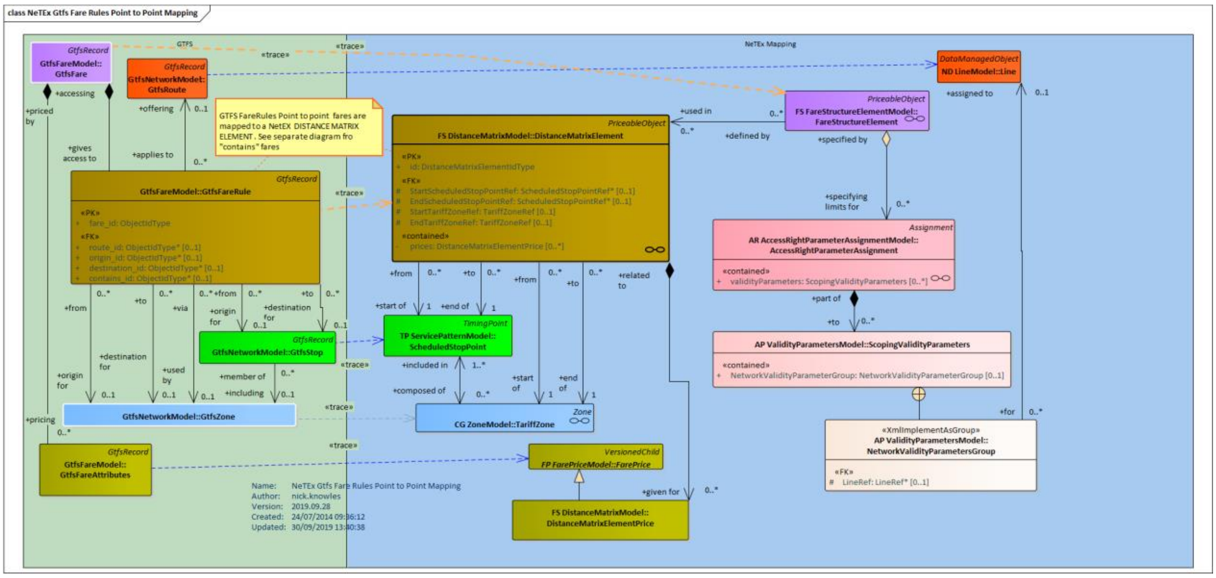


5.4.1.13.2 GTFS Fare Rule (Point to Point) to NeTEx Mapping – Details

In the common point-to-point-to-point tariff structure, each ***GtfsFareRule*** maps to a NeTEx DISTANCE MATRIX ELEMENT, for which a separate price i.e. DISTANCE MATRIX PRICE can be specified (Figure 54).

In GTFS, the ***route_id*** is repeated on every ***GtfsFareRule*** (although it would probably be the same for all the rules for a ***GtfsRoute***). In Transmodel/NeTEx, the LINE would usually be specified at the TARIFF or FARE PRODUCT level. For a simple mapping, if a LINE is specified, a FARE STRUCTURE ELEMENT can be added to group all the DISTANCE MATRIX ELEMENTs with the same line and an ACCESS PARAMETER ASSIGNMENT on the FARE STRUCTURE ELEMENT can be used to associate the access rights to a specific LINE.

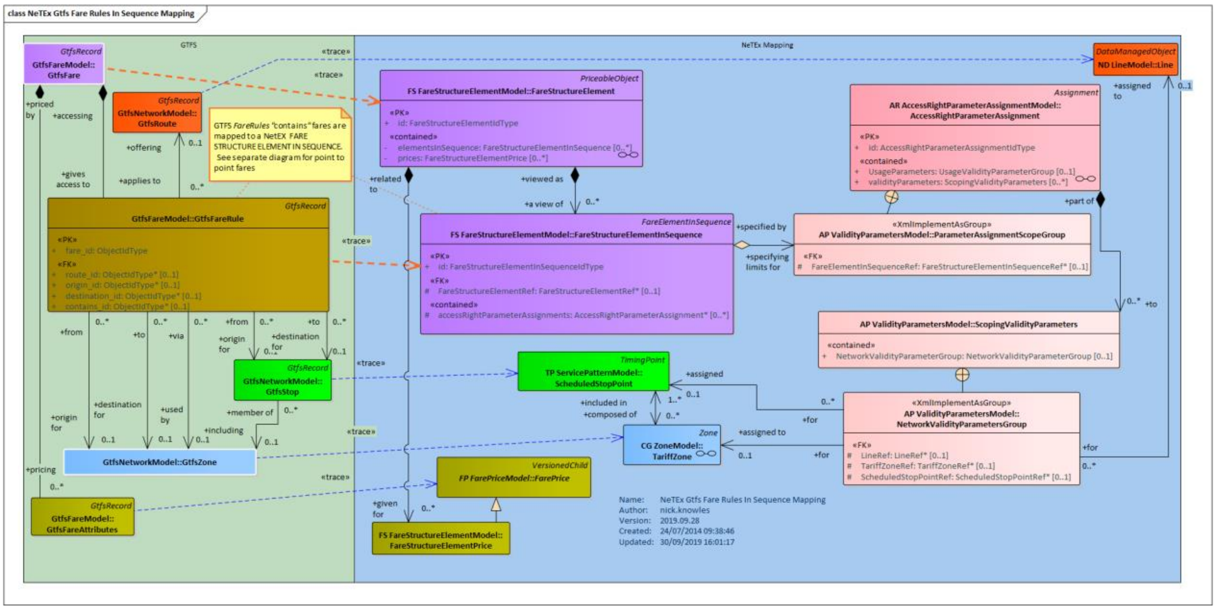
*Figure 54 GTFS **Fare_rule** to NeTEx **DistanceMatrixElement** Mapping – Details*



5.4.1.13.3 GTFS Fare Rule (in Sequence) to NeTEx Mapping – Details

The GTFS ***Fare_rule*** element may also be used to define a tariff structure that is for use of a specific list of zones in sequence, using the GTFS ***contains*** attribute (Figure 55).

*Figure 55 GTFS **Fare_rule** to NeTEx **FareElementInSequence** Mapping – Details*



5.4.1.13.4 Overloading of GTFS Fare_rules

Table 6 summarises the overloading of the GTFS ***GtfsFareRule*** concept with what in Transmodel/NeTEx are separate concerns:

Table 6 Overloading of GTFS ***Fare Rule*** *Concepts*

GTFS Usage tariff structure	GTFS tariff structure parameters	Possible Restrictions	Transmodel Tariff Structure concept
Zone-to-Zone (can also be used for zonal)	***origin_id*** ***destination_id***	(route_id)	DISTANCE MATRIX ELEMENT + FARE STRUCTURE ELEMENT
Point-To-Point	***origin_id*** ***destination_id***	(route_id)	DISTANCE MATRIX ELEMENT + FARE STRUCTURE ELEMENT
Zone in sequence	***origin_id;*** ***contains_id***	(route_id)	FARE STRUCTURE ELEMENT + FARE STRUCTURE ELEMENT IN SEQUENCE
Flat	**-**	Route_id	FARE STRUCTURE ELEMENT

5.4.1.14 GTFS Fare Attributes: NeTEx Mapping

5.4.1.14.1 GTFS Fare Attributes to NeTEx Mapping – Details

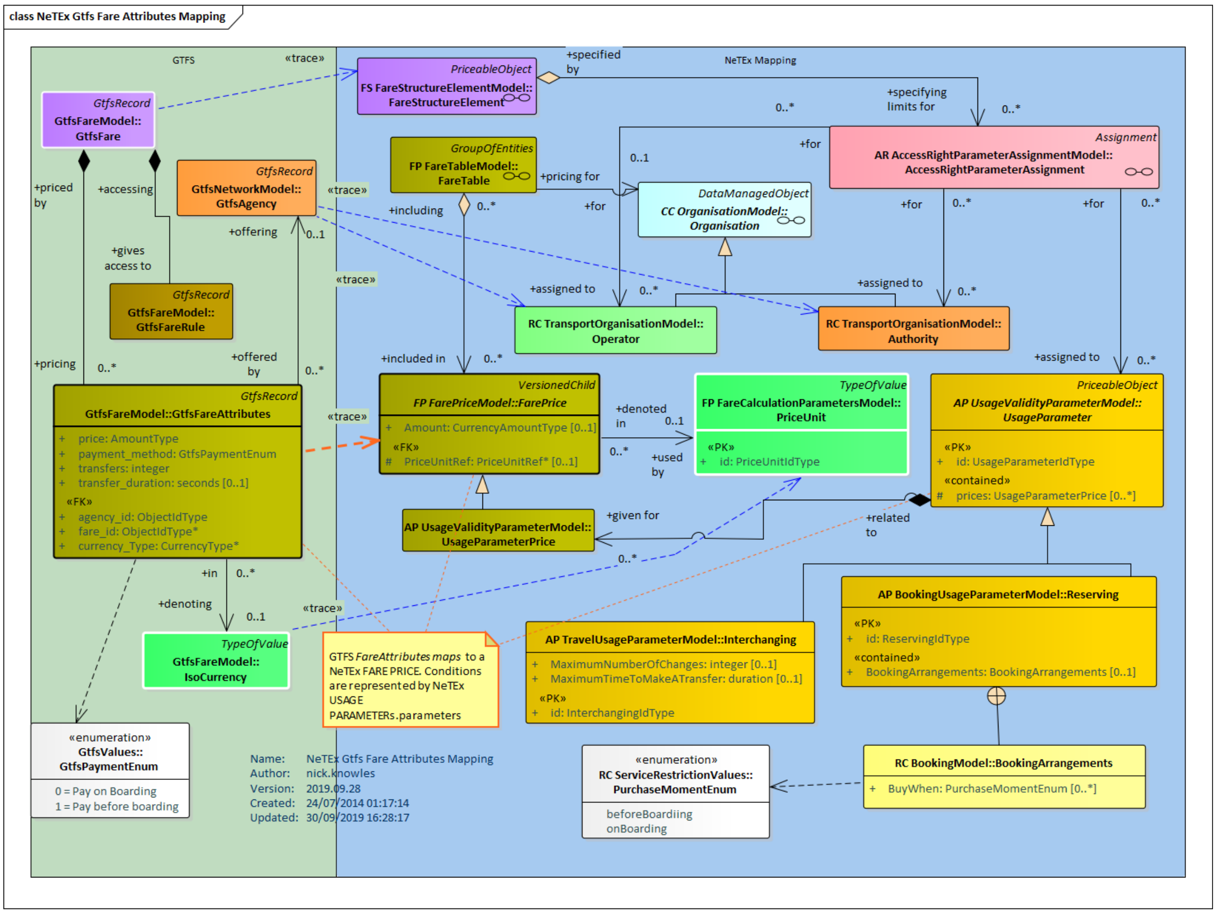
A GTFS **fare_attributes.txt** record (***GtfsFareAttributes)*** corresponds primarily to a Transmodel/NeTEx FARE PRICE element (in that it specifies the price for a tariff structure element), but also mixes in other usage conditions for the fare which in Transmodel/NeTEx are represented as separate USAGE PARAMETERs (which would normally be applied at the FARE PRODUCT or SALES OFFER PACKAGE LEVEL so as to be properly normalised) (Figure 56).

Specifically, (i) the ***transfers*** attribute of a ***GtfsTransfer*** (i.e. maximum number of transfer allowed), is specified on a Transmodel/NeTEx INTERCHANGING usage parameter and (ii) the ***payment_method*** attribute (i.e. whether payment is before or after boarding), is stated on a Transmodel/NeTEx RESERVING parameter. In a normal NeTEx representation it is likely that these would be common properties of a FARE PRODUCT that apply to all fares for the product. However, in order to achieve a simple GTFS mapping of just the tariff structure without introducing a FARE PRODUCT, these can equally well be associated with a FARE STRUCTURE ELEMENT and applied to all associated prices (at the expense of some denormalization, i.e. redundancy). A Transmodel/NeTEx ACCESS RIGHT ASSIGNMENT is used to associate these parameters with a FARE STRUCTURE.

Note that the GTFS model is potentially denormalised if it is repeating the same conditions on each ***GtfsFareAttributes***, i.e. price. If, however, a different price was associated with payment before boarding and payment on board, or with different numbers of permitted transfers, then it is not redundant; this would be represented in Transmodel by additional FARE STRUCTURE ELEMENTs with different parameter assignments.

Since for a large network here may be a large number of prices, NeTEx normally uses a FARE TABLE to group FARE PRICEs efficiently as nested tables so as not to repeat values.

*Figure 56 GTFS **Fare Attributes** to NeTEx **FarePrice** Mapping – Details*



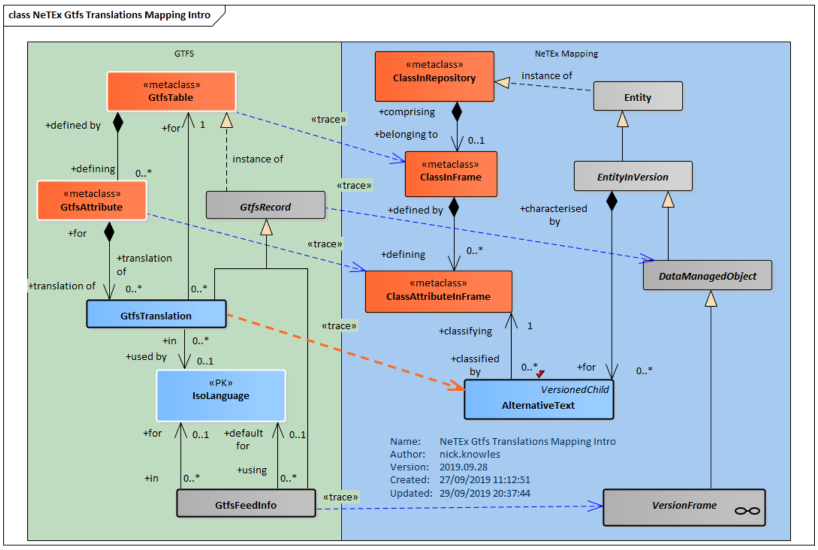
5.4.1.15 GTFS Translations: NeTEx Mapping

5.4.1.15.1 GTFS Translation to NeTEx Mapping – Introduction

A GTFS **translations.txt** record (***GtfsTranslation)*** holds alternative national language translations of text elements; Transmodel//NETEX has a similar ALTERNATIVE TEXT element that may be used to provide translations (Figure 57).

Note that NeTEx also has an ALTERNATIVE NAME mechanism to specify aliases for the main names of certain elements.

Figure 57 GTFS ***Translations*** *to NeTEx* ***AlternativeText*** *Mapping – Introduction*

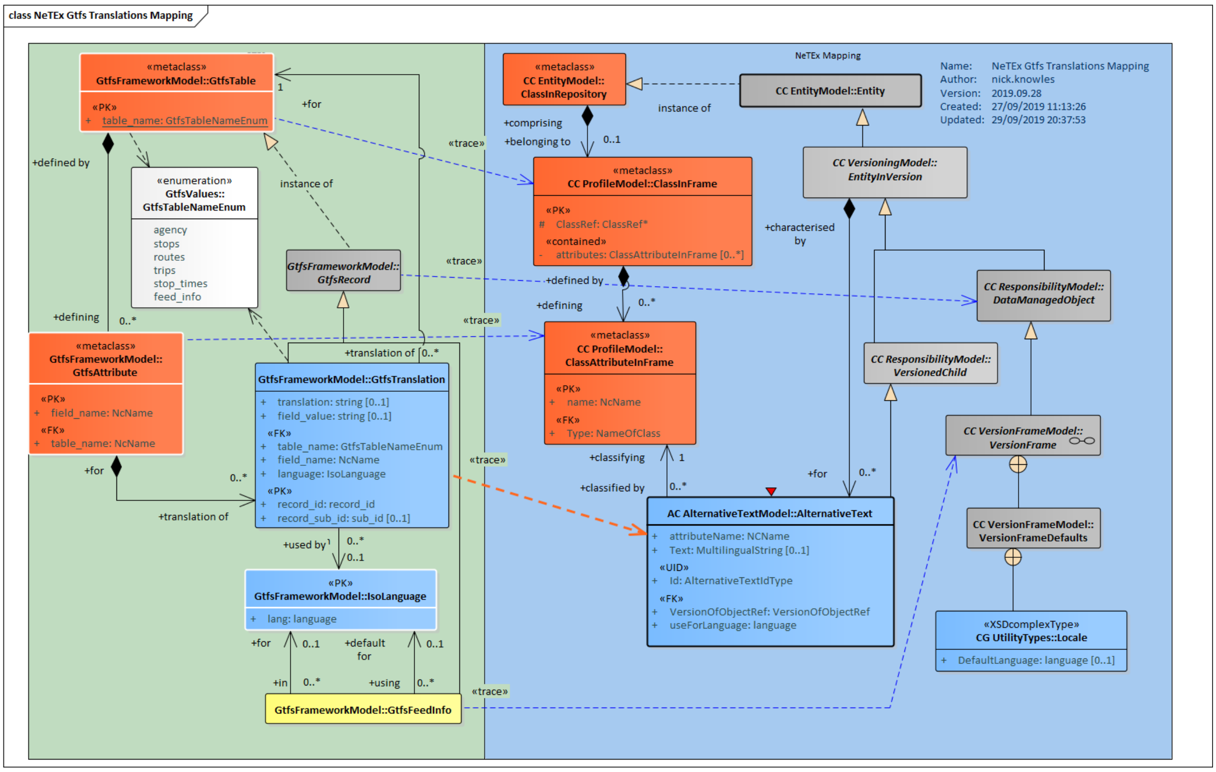


5.4.1.15.2 GTFS Translation to NeTEx Mapping – Details

A ***GtfsTranslation*** instance indicates the name of the GTFS attribute and the GTFS Record type for which it provides the translation (in effect GTFS metamodel properties). A NeTEx ALTERNATIVE TEXT similarly indicates the element (CLASS IN FRAME) and attribute name (CLASS ATTRIBUTE IN FRAME) for which it provides a translation (Figure 58).

In addition, the ***GtfsFeedInfo*** includes some language default attributes which may be used to indicate how Translation values should be used. Similar defaults may be placed on a NeTEx VERSION FRAME.

Figure 58 GTFS ***Translations*** *to NeTEx* ***AlternativeText*** *Mapping – Details*



5.4.1.16 GTFS Feed Info: NeTEx Mapping

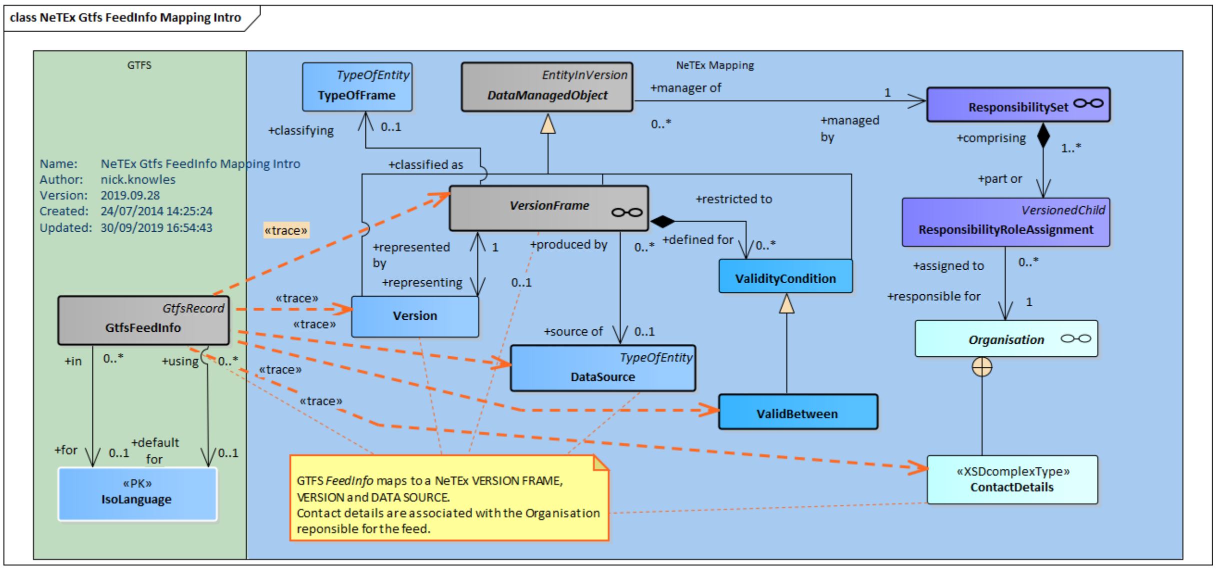
5.4.1.16.1 GTFS Feed Info to NeTEx Mapping – Introduction

A GTFS **feed_info.txt** record (***GtfsFeedInfo)*** holds overall version and validity data that in Transmodel/NETEX is associated with the VERSION FRAME holding the data elements. The VERSION FRAME can be associated with a DATA SOURCE (indicating the origin of the data and a VALIDITY CONDITION indicating the validity of the data (for GTFS this is a simple VALID BETWEEN condition defining a data range. A ***GtfsFeedInfo*** thus corresponds to several different Transmodel concepts (Figure 59).

If contact details are supplied, they may be associated with an ORGANISATION given as responsible for the feed.

In Transmodel/NeTEx it is possible to mark an instance of a version frame as being conformant to a particular set of rule or "profile" by means of a TYPE OF FRAME. Thus, a GTFS data set can be tagged with a "GTFS" TYPE OF FRAME to indicate that it contains GTFS content and uses GTFS compatible identifiers and other data values.

*Figure 59 GTFS **Feed_info** to NeTEx **VersionFrame** Mapping – Introduction*

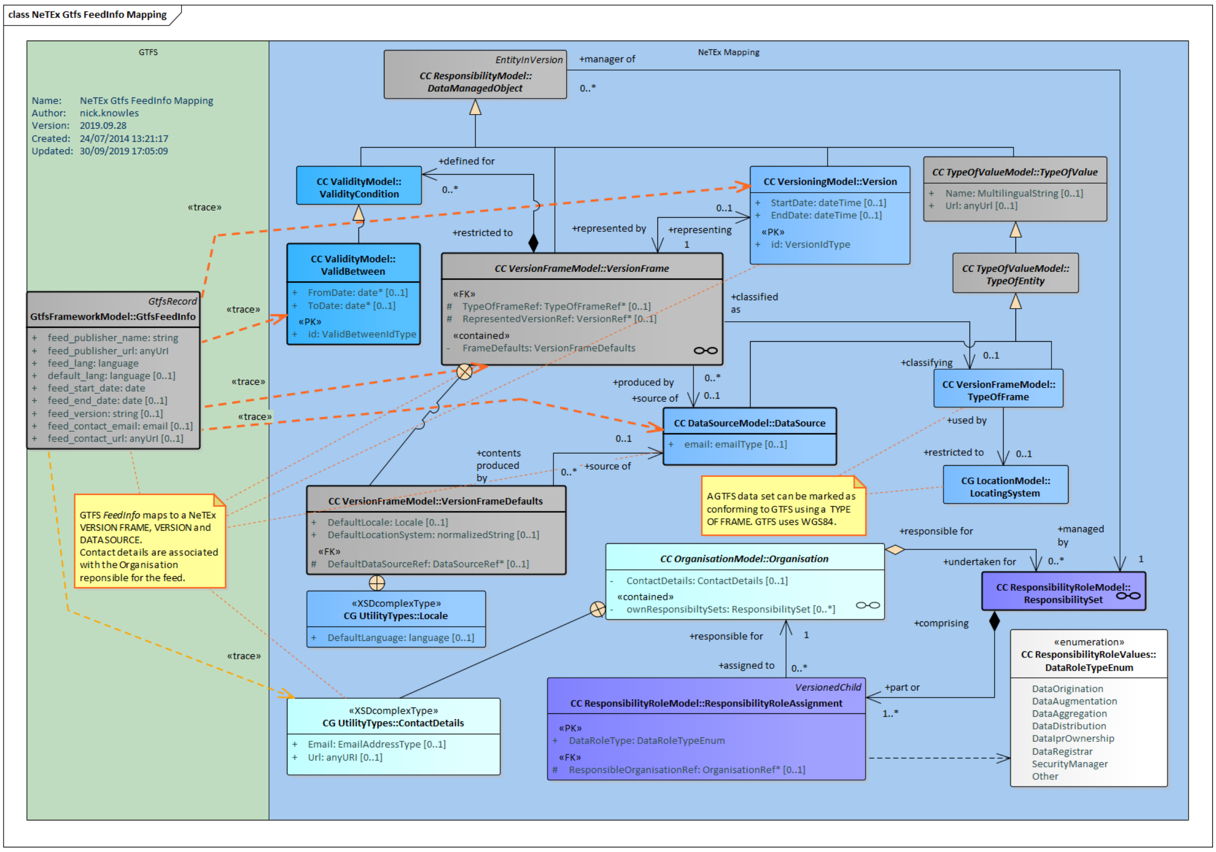


5.4.1.16.2 GTFS Feed Info to NeTEx Mapping – Details

The attributes from a ***GtfsFeedInfo*** are divided among a Transmodel/NeTEx VERSION FRAME, DATA SOURCE and VALIDITY CONDITION (Figure 60).

Note: A NeTEx implementation can indicate that a frame contains GTFS compliant data by means of a TYPE OF FRAME INSTANCE.

*Figure 60 GTFS **Feed_info** to NeTEx **VersionFrame** Mapping – Details*



5.4.2 Summary of GTFS and Transmodel/NeTEx equivalences

Table 7 summarises the GTFS to Transmodel/NeTEx mappings discussed in this paper.

Table 7 Mapping of GTFS Concepts to Transmodel/NeTEx

GTFS record	GTFS Model element	Transmodel/NeTEx concept	Comment
agency	***GtfsAgency***	ORGANISATION (OPERATOR, AUTHORITY)	
stops	***GtfsStop***	SCHEDULED STOP POINT + STOP PLACE	Overloaded. TARIFF ZONE not reified.
		SCHEDULED STOP POINT + STOP QUAY	
		ENTRANCE, BOARDING POINT, ACCESS SPACE	
routes	***GtfsRoute***	LINE	

trips	***GtfsTrip***	SERVICE JOURNEY	Overloaded
		TEMPLATE SERVICE JOURNEY + FREQUENCY GROUP	Overloaded
stop_times	***GtfsStopTimes***	STOP POINT IN JOURNEY PATTERN + PASSING TIMEs + DESTINATION DISPLAY	Denormalised
		CALL	Optimisation
calendar	***GtfsCalendar***	DAY TYPE + PROPERTY OF DAY + OPERATING PERIOD DAY TYPE ASSIGNMENT	Parent ***GtfsService*** not reified
calendar_dates	***GtfsCalendarDate***	DAY TYPE ASSIGNMENT	
fare_attributes	***GtfsFareAttributes***	FARE PRICE + ACCESS RIGHT ASSIGNMENT + USAGE PARAMETER (INTERCHANGING, RESERVING)	Denormalised
fare_rules	***GtfsFareRule***	FARE STRUCTURE ELEMENT + DISTANCE MATRIX ELEMENT	Overloaded. Parent ***GtfsFare*** not reified.
		FARE STRUCTURE ELEMENT + FARE STRUCTURE ELEMENT IN SEQUENCE	
shapes	***GtfsShape***	LINK SEQUENCE PROJECTION	Parent ***GtfsShape header*** not reified
frequencies	***GtfsFrequency***	HEADWAY FREQUENCY GROUP + HEADWAY INTERVAL	Denormalised
		RHYTHMICAL FREQUENCY GROUP	
transfers	***GtfsTransfer***	CONNECTION	Overloaded
		SERVICE INTERCHANGE	
		SERVICE INTERCHANGE RULE + INTERCHANGE RULE PARAMETER	
pathways	***GtfsPathway***	PATH LINK	
levels	***GtfsLevel***	LEVEL	
feed_info	***GtfsFeedInfo***	VERSION FRAME + DATA SOURCE + VALIDITY CONDITION + VERSION	Zip file used to group

5.5 Further High-Level comparisons of GTFS and NeTEx

In summary: the NeTEx model is more extensive and more complex than the GTFS representation for two fundamental reasons:

(a) It includes a lot more function, including planning, scheduling and operational aspects of timetables and a much richer fare model; it is intended not just for distribution of final timetables to third parties, but also for exchange of the planning and operational timetables.
(b) It separates different concerns into separate abstractions in the model, so as to achieve a high level of reuse of components and allow flexibility for future evolution of the model.

Although the Transmodel/NeTEx representation is more extensive than GTFS, it is not always more complex because GTFS overloads some elements with multiple alternative meanings (necessitating a complex interpretation by a program importing GTFS data), while Transmodel in general separates concerns and models each concept separately.

As a simple visualisation of the difference of scope of GTFS and Transmodel/ NeTEx timetable representations, the following two diagrams are offered:

1. The first (Figure 61) shows the Transmodel representation of a route and its timetable as a set of informational layers, each concerned with different aspects the model (physical route, timing information, service pattern, etc), the elements of each layer can be separately defined and repeatedly reused. So, for example, a given journey can be fully defined simply by a starting time and references to other existing elements. Timing information is held separately so that reusable sets of precise operational times can be exchanged (including wait times) independently of specific journeys, and also allowing different timings to be used at different times of day for the same service patterns.
2. The second (Figure 62) shows the GTFS representation, which holds only the full resolved timings for each journey. Data (apart from stop details) must be repeated on each journey.

Figure 61 Transmodel representation of a timetabled journey

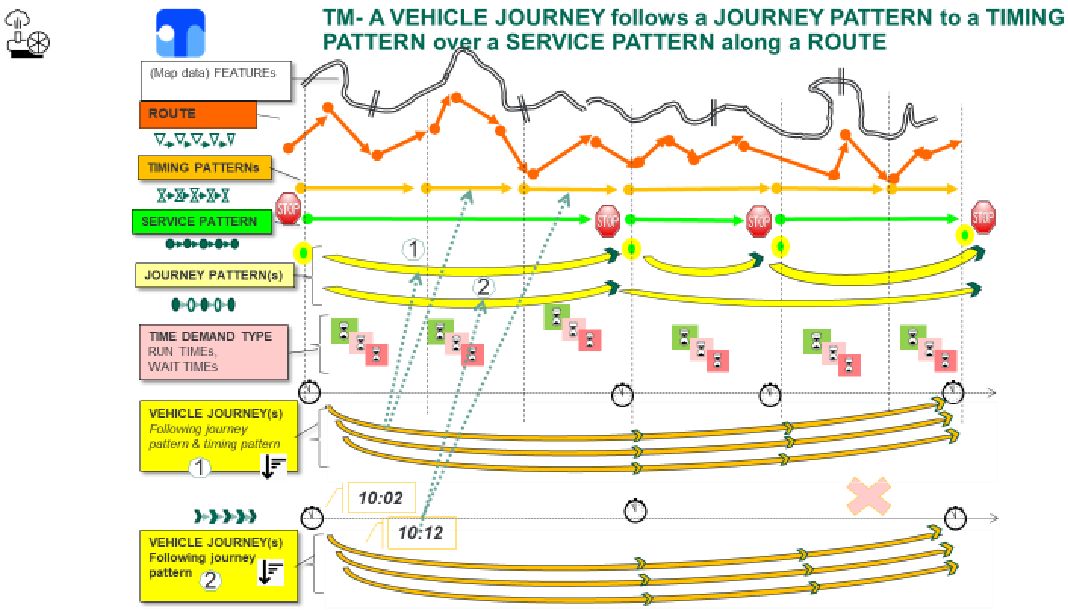
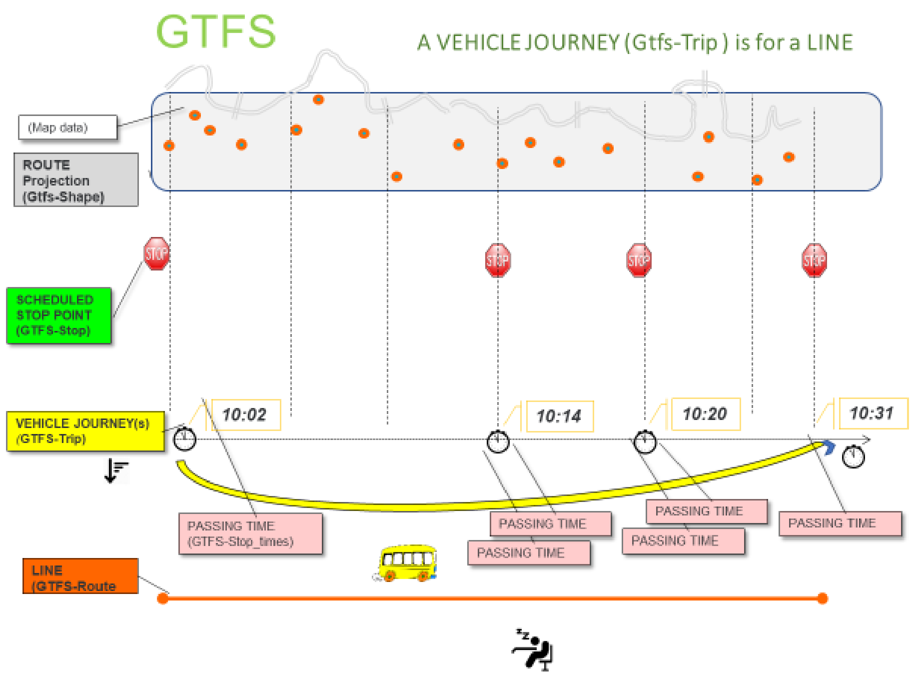


Figure 62 GTFS representation of a timetabled journey



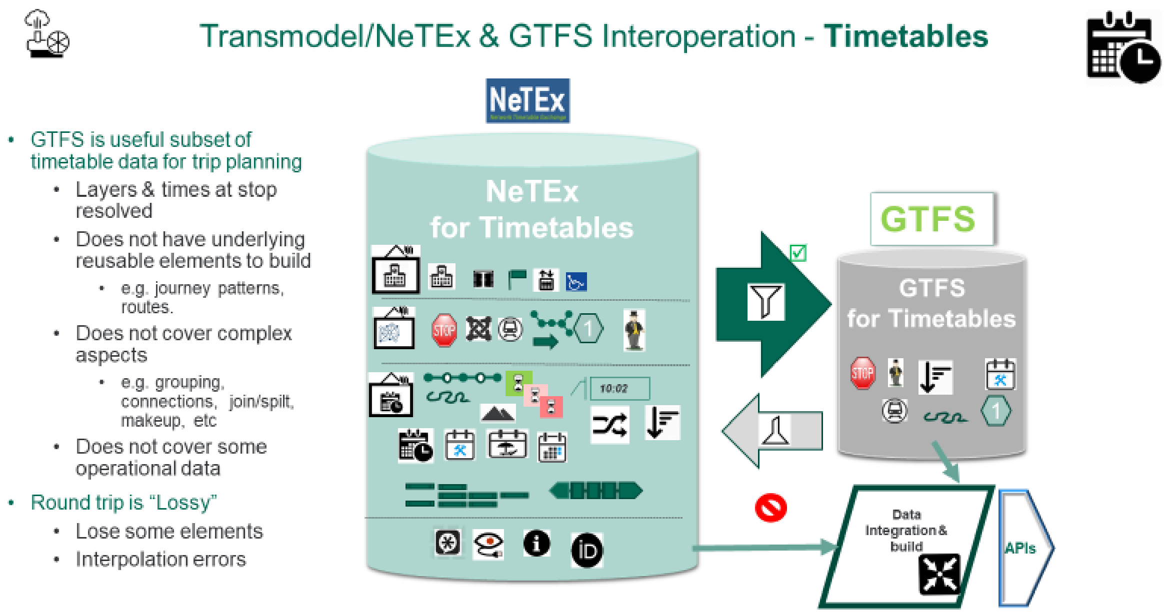
5.5.1 Comparison of relative scope for Timetables

Another visualisation of the relative scope of the two standards can be obtained by using an icon for each type of functional element that may be present in the exchange formats. In the following diagram (Figure 63), the potential Transmodel based NeTEx representation of a timetable is shown on the left, including separate reusable frames to contain the network and timetable elements, and including elements to describe the arrangement of journeys for presentation, detailed time conditions and footnotes, etc. The GTFS model has a much smaller scope - making it simpler to use for certain purposes, but not covering a number of planning and operational use cases.

GTFS does not include more complex journey concepts for passenger information such as the coupling of journeys (relevant for describing trains and services that change their service number.), train make up (needed to give accurate information about boarding positions and accessibility)

It is possible to transform a NeTEx representation into GTFS, using the Transmodel conceptual framework to reconcile terminology and concepts, and vice versa, but the round trip is "lossy" as GTFS lacks certain elements.

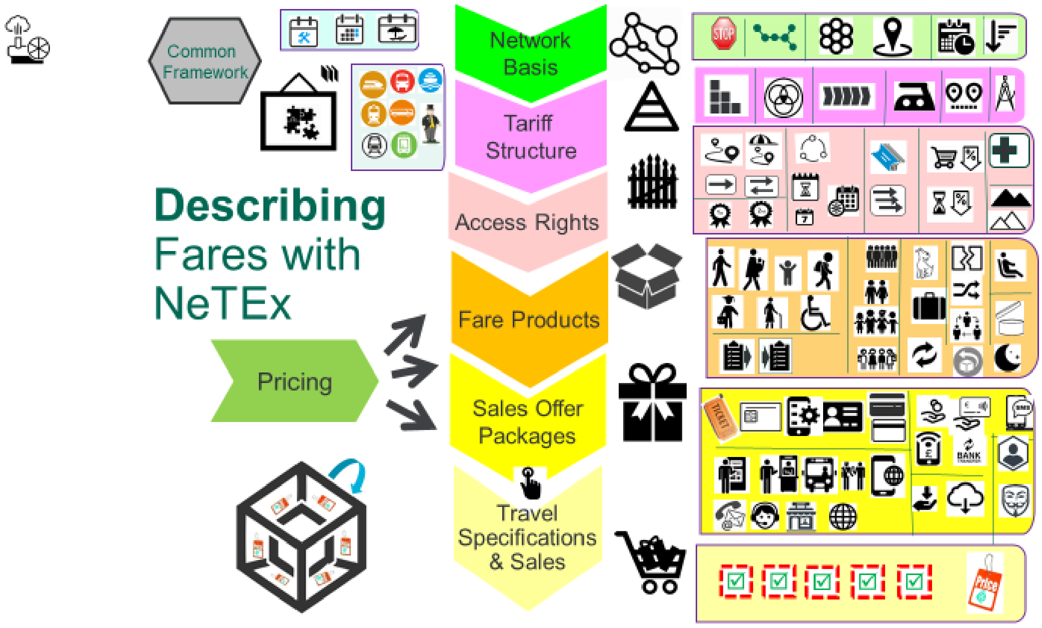
Figure 63 Comparison of Transmodel and GTFS Timetable model scope



5.5.2 Comparison of relative scope for Fares

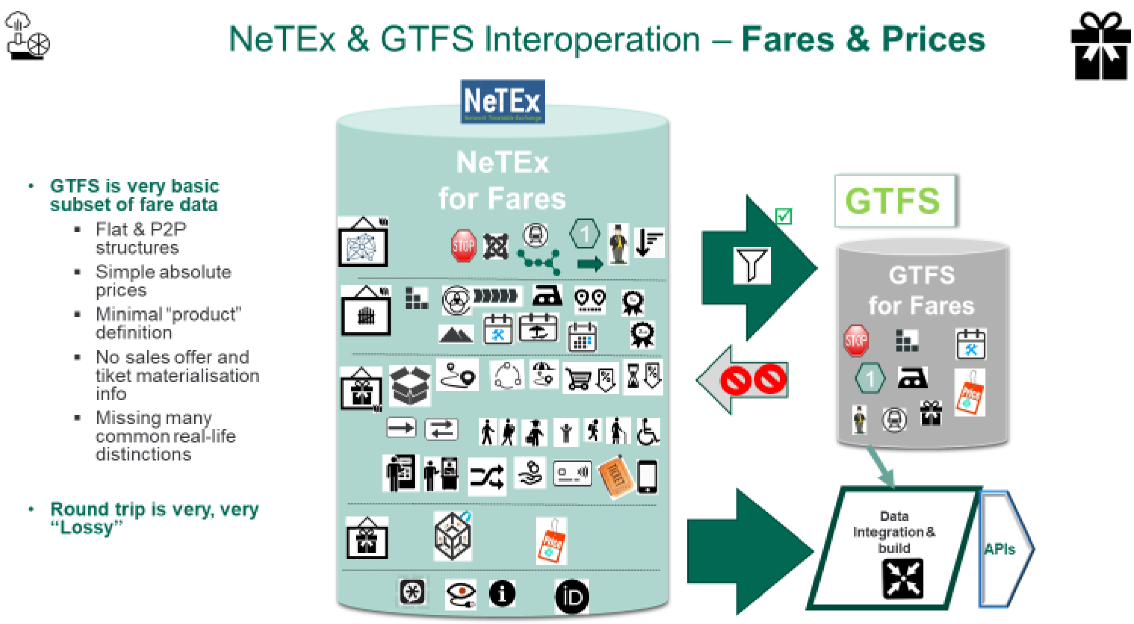
In order to make a similar comparison between Transmodel/NeTEx and GTFS for fares, we first introduce a set of icons to represent the different functional elements of the Transmodel Fares model (Figure 64). These correspond to the successive layers of components (Tariff structure, Access rights, Fare Products, Sales Offer Packages, etc) with which Transmodel product definitions are assembled.

Figure 64 Visualisation of Transmodel fare components



A similar visualisation to that used for timetables can be used to contrast the Transmodel and GTFS representations of fares (Figure 65). Again, the GTFS model has a much smaller scope - and is unable to represent a number of common product types and tariff structures, as well as ancillary aspects of fare information.

Figure 65 Comparison of Transmodel and GTFS fare model scope

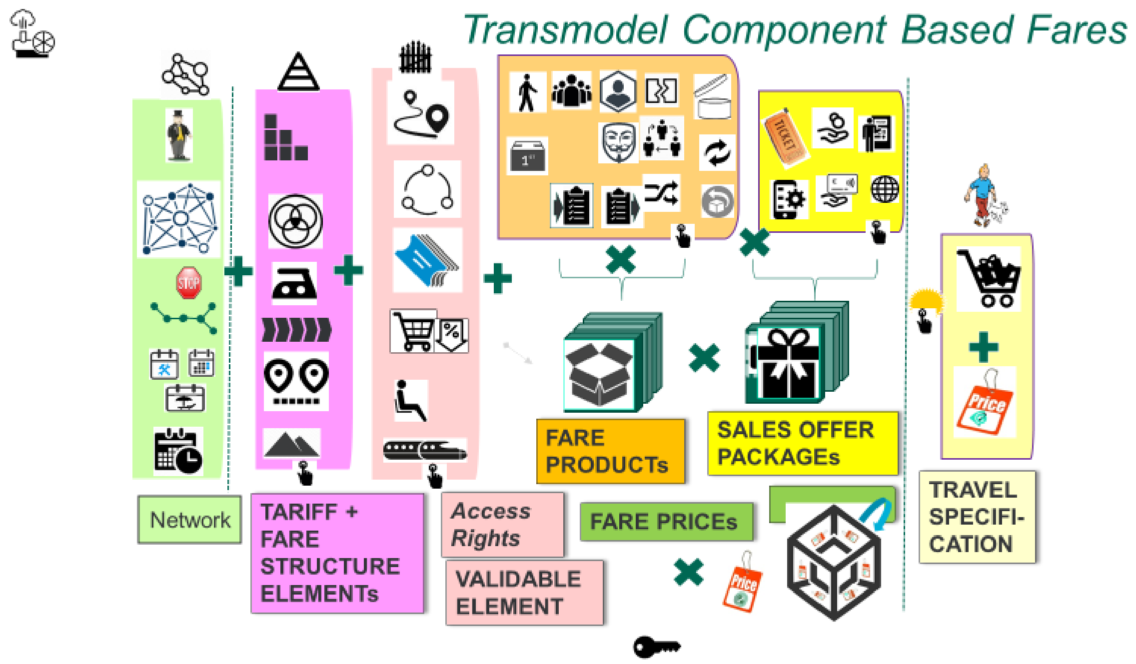


5.5.2.1 Component based fares

We can make the same comparison of fare scope in slightly more detail with the following visualisation that illustrates the Transmodel/NeTEx fare model as a component-based representation that builds upon successive layers of reusable elements (Figure 66). Thus, existing network and service elements such as stops and lines are referenced by tariff structure elements in different combinations to define access rights. These in turn are used to build fare products. Fare products themselves may be combined in different ways for different purposes as one or more sales offer package. When a user buys a product, they in effect specify a set of choices from the options available in a given sales offer and its products.

Such an approach allows a wide variety of products (trips, passes, carnets, discount cards etc) and materialisations (paper, electronic, account based, etc) for any possible transport mode (rail, bus, metro) to be covered with a single model.

Figure 66 Visualisation of Transmodel component-based fares

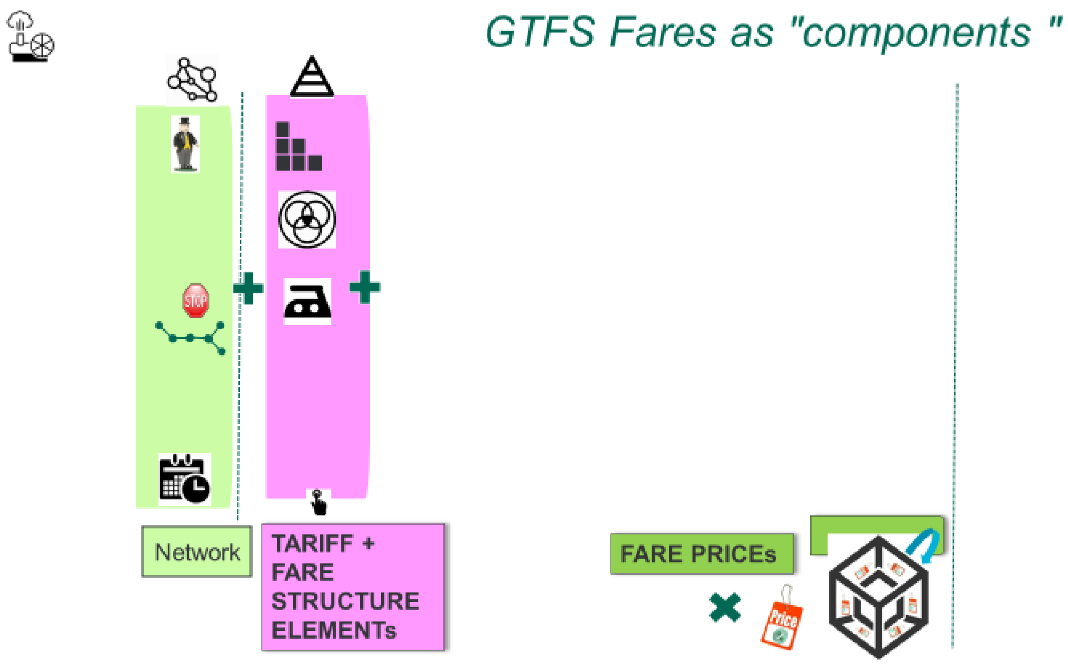


5.5.2.2 GTFS fare "components"

If we consider the GTFS fare model in similar term as a set of components (Figure 67), then two things are apparent – it is can be seen as being mainly concerned with simple tariff structure and price elements. It lacks the concepts of product, materialisation as tickets, as combinations of products and conditions to make sales offers, etc. Particularly egregious gaps are the lack of user types to allow a precise definition of who is eligible for a product and the very limited support for any type of season pass.

Other important capabilities of Transmodel concern the representation of price; Transmodel keeps the presentation of what is priced separate from the price itself (allowing separate sets of prices also allows prices to be specified for any combination of tariff structure, usage condition, product and packaging and marketing factor. Furthermore, Transmodel allow the separate exchange of pricing parameters (percentages, limits, rounding, etc), so that prices may be derived from other prices. (e.g. "child fare is half the adult fare").

Figure 67 Visualisation of GFS fares as components



5.5.3 Conclusion: The Cathedral versus the Bazaar

The GTFS and Transmodel formats can be seen to some degree as examples of two different approaches to systems design, as characterised in Eric Raymond's essay *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*.

- GTFS, as a de facto standard created to meet specific proprietary needs, is a classic product of the "bazaar"; it is minimalistic, ad-hoc, and covers just the fundamental requirements for distributing timetable data to third parties. It uses a simplified set of data structures with arbitrary representations and optimisations and is not concerned with any extraneous theoretical consideration of how public transport data is assembled or any wider requirements. This has significant advantages for expediency and simplicity, but also certain disadvantages in that some of the optimisations (such as overloading) make it progressively harder to add additional function over time to evolve the standard and others (e.g. denormalization) can lead to inefficiencies with large data sets.
- The Transmodel based NeTEx, the output of international teams of experts working under long term standards processes, is a much more extensive and considered specification, and so in Raymond's paradigm more of a "cathedral" (though whether in Gothic, Renaissance or Baroque style we leave to the reader). It has a uniform, overreaching architecture designed to provide a unified, joined up approach for developing strategic data systems for public transport and based on the comparison of many different systems, supporting many different use cases, in many different countries. Consequently, it includes additional abstractions and separations of concern that, while not always needed to meet simple requirements, support the more complex use cases and serve to "future proof" it for further evolution (just as a Cathedral built over several generations needs to have foundations capable of bearing the completed building). It is worth noting that Transmodel was

developed in response to dissatisfaction with first generation national PT standards for timetables, similar to GTFS (many in fact CVS based) that, over time, had become brittle and hard to evolve further - and which did not offer easy integration or reuse of data elements in operational and fare management systems.

The differences of the bazaar and the cathedral are reflected also in the choices of technology used in GTFS and NeTEx.

- GTFS uses CVS records, very much a "bazaar" technology, easy to use without tooling, but putting the onus on the purchaser to validate and interpret the contents (and allowing many different pagan gods may be followed when doing so). Each concept requires a separate flat file so there is a design propensity to optimise to reduce the number of repeating elements and to denormalise.
- NeTEx uses XML schema, like the masonry of a cathedral versus the tents of the bazaar, requiring more investment to learn how to use, as well the use of software tools to automate and facilitate binding, especially when parsing, but allowing a richer and subtler model. The ability to nest and cross-reference rich data structures within XML allows for a more straightforward correspondence to the representational model of a database (encouraging compliance with a consistent Transmodel "theology".) and greater flexibility for grouping and packaging data. The support for referential integrity and other validation checks in standard XML validators also helps to protect the consumer. These considerations can be burdensome for simple use cases but become more important when covering more complex data sets (e.g. complex journey coupling and splitting, or fares) where large numbers of interlocking elements are needed and when reusing data structures in many different workflows and problem domains.

The primary subject of Raymond's essay was not, however the nature of the software representations or the specific software technologies used, but the development processes; contrasting the transparent, open source mass-collaboration approach of "the bazaar" with a more closed, formal and phased process needed to design a "cathedral". In this respect however the differences between GTFS and NeTEx are actually less marked than they have been historically,

- The GTFS specification is published openly and is notionally under communal management, but in practice the adoption of most enhancements is led by the provision of official support for the features in the Google Transit feed (subject to Google's business objectives) and the updating of the official documentation.
- While the formal Transmodel and NeTEx specifications are subject to the open, but relatively slow CEN standardisation life cycle, they do already provide a specification map for modelling an extensive functional domain. Furthermore, the NeTEx schema itself is also available in Github, allowing for immediate collaborative fixes and provisional enhancements that can be reviewed and assimilated over time into the formal specification though the normal CEN processes.

s

6 Harmonising of standards as a strategic process

Data systems for public transport are complex and require a significant investment of resource and effort to implement and run. While standardisation is of great value for reducing the overall complexity of such systems, reducing the costs of platforms and tools and increasing interoperability and capability, it should be understood that the process of adoption is a gradual one that requires a strategic vision and patience and happens over years rather than months.

As has already been noted, Transmodel has significant value a conceptual tool that can be used to gradually align different standards so that they can converge on a common format. The following diagram gives a visualisation of how Transmodel has been used to align key National European standards over a 20-year period leading up to the EC ITS directive to encourage the use of pan-European standards for PT data. (Figure 68 - GTFS might additionally be included). The effective evolution of standards requires a two-way flow between the concrete formats used in the field that encounter real-life workflows and new business requirements, and the abstraction of those new features as new common concepts for the Transmodel reference model, with which convert formats can be harmonised. This evolution will need to continue.

Figure 68 Evolution of Fare standards