Údarás
Náisiúnta Iompair
National Transport Authority

Irish National NeTEx Profile
for exchanging Irish Stop & Timetable data

Part 1 – Overview & Profile Elements

Version 1.0

Document Control Log

Date	Modified by	Comment	Version
2020.04.30	NJSK	Initial outline Draft	V0.01
2020 07.23	NJSK	Working Draft	V0.17
2020 08.05	NJSK	Draft for initial Review	V0.20
2020 08.12	NJSK	Split into two documents – separate out mappings	V0.21
2020 08.17	NJSK	Corrections. Also add RUN TIME and WAIT TIME sections	V0.22
2020.08.06	NJSK	Issued Draft for review	V0.23
2021 10.20	NJSK	Issued Version	V1.0

Contents

Page

1 Introduction

This document describes the core and network elements for the Irish NeTEx profile (EIRE_NP). The profile specifies how to encode Irish stop and timetable data using NeTEx XML, the common European format for the exchange of public transport data. The document is accompanied by examples of Irish public transport data for different modes of transport encoded as NeTEx XML.

The EC Intelligent Transport directive mandates the NeTEx format for the open exchange of travel information at National Access Points (NAP) across Europe. Work on the profile has been funded by the European Commission to further the EC's *Priority action A of ITS-Directive*.

The Irish NeTEx profile is based on a number of CEN standards as given in the normative references below, this document also references a number of legacy data formats from which the data is mapped – for a summary of these see the bibliography at the end.

The profile covers all modes of transport found in Ireland; bus, tram rail, and ferry modes.

The profile specification is in divided into two parts

- **Part1** – *Irish NeTEx Profile Specification* The specification of the profile elements. (This document)
- **Part2** – *Irish NeTEx Profile Mappings*. Detailed mappings of existing data sets (NaPTAN, GTFS, Hafas etc,)

1.1 Presentation Conventions

1.2 Relationship to European profile

The EIRE-NP provides modular sub-profiles for Stops and for Timetables. These are aligned with the common European Passenger Information Profile (EPIP) for exchanging stop and network data (TS 16614:PI Profile), and a similar structure is used in this document to that used for the European profile. Such an approach makes it easier for software developers and others who need to support both Irish and European profiles, giving an economy of scale to their development efforts.. The use of harmonised documents also facilitates the cross-border exchange of data, one of the main goals of the European profile.

National profiles based on the European profile may either chose to limit themselves to just the constructs in the EPIP, or also to allow additional elements to cover additional use cases. The EIRE-NP has a small number of augmentations, primarily to allow the inclusion of timing information.

1.2.1 Presentation Conventions in the text

NeTEx follows certain conventions for presentation of technical terms within text; in particular this helps to distinguish the use the same concept at different levels of abstraction (informal textual use, conceptual model, physical model, XML schema, etc). These conventions are given in detail in section 7, but two key points are mentioned here:

- Transmodel conceptual model elements are shown in UPPER CASE, for example “LINE”, “SCHEDULED STOP POINT”, “TYPE OF TARIFF”. If required to be in the plural, a lower case is used; for example, “SCHEDULED STOP POINTs”, “TYPEs OF TARIFF”.
- Concrete XML elements from the NeTEx schema are shown in bold italic, *for* example, ***Line, TypeOfTariff***. Compound names are camel cased without a space and are never pluralised, e.g., “***ScheduledStopPoint***”. The NeTEx element names correspond word for word with their corresponding Transmodel entities.

1.3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

CEN

- EN 12896, Road transport and traffic telematics - *Public transport - Reference data model* (*Transmodel*) Parts 1 to 9.
- CEN/TS 16614-*1 Network and Timetable Exchange (NeTEx) - Network description.*
- CEN/TS 16614-2 *Network and Timetable Exchange (NeTEx) - Timing information.*
- CEN/TS 16614-*3 Network and Timetable Exchange (NeTEx) - Fare description.*
- CEN/TS 16614:PI *Network and Timetable Exchange (NeTEx)— Passenger Information European Profile*
- EN 15531-1, *Public transport - Service interface for real-time information relating to public transport operations - Part 1: Context and framework.*
- EN 15531-2, *Public transport - Service interface for real-time information relating to public transport operations - Part 2: Communications infrastructure.*
- EN 15531-3, *Public transport - Service interface for real-time information relating to public transport operations - Part 3: Functional service interfaces.*
- CEN/TS 15531-4, *Public transport - Service interface for real-time information relating to public transport operations - Part 4: Functional service interfaces: Facility Monitoring.*
- CEN/TS 15531-5, *Public transport - Service interface for real-time information relating to public transport operations - Part 5: Functional service interfaces - Situation Exchange.*

UK Department of Transport (DfT)

- *NPTG & NaPTAN Schema Guide v2.5 2014*

The NeTEx XML schema can be downloaded from http://netex-cen.eu , along with available guidance on its use, example XML files, and case studies of national and local deployments.

See also bibliography at end.

2 The structure of this document

This document describing the Irish Profile (EIRE_NP-NT is arranged in three main parts as follows:

2.1 Introduction to NeTEx

Section 3 provides a brief introduction to NeTEx; it may be skipped by those already familiar with NeTEx.

Section 4 gives a brief summary of the scope of the profile and of the high-level mapping of Irish Data and is a useful place to begin if reading the profile for the first time.

5, 6 and -7 provide other contextual reference material; again, this may be skipped by those already familiar with NeTEx.

2.2 NeTEx elements relevant for the Irish Profile

Sections 8-10 contain the main body of the specification, a description of each the NeTEx elements needed to represent data, organised within three sections:

1. **Common core profile** describing elements that are shared by the two other profile parts given in this document and that may also be reused later on by other future profiles (Fare Profile, Accessibility Profile, etc.). The framework abstracts out common implementation 'plumbing for identifiers, versioning etc 'and can be ignored on a first reading.
2. **Stop profile**, describing the STOP PLACEs and their hierarchy.
3. **Network and Timetable profile**, describing the network topology and all related timing information (timetables). This constitutes the core content of the EIRE-NP.

NeTEx is modularised into a core framework and a number of functional modules that build on it; the modularisation is reflected in the documentation. The modularisation facilitates the separate exchange of stop data from that of timetable and from fare data (but it is technically still possible to exchange both of them together in a single document). The presentation of elements follows a similar order, starting with the most basic common elements and then describing their successive dependents.

2.3 Rules & Considerations for populating documents according to the Irish profile

Two additional sections (11 and 12) give general rules on creating and using documents conformant to the profile, and also more detailed mappings from each of the different legacy formats..

1. **Common rules**
2. **Data quality** and the use of validators
3. **Mapping** Irish NPTG, NaPTAN and GTFS, VDV452 and HRDF data

Some further annexes provide the NeTEx definitions of elements and other supporting tables.

3 Overview of NeTEx

This section provides a very brief overview of NeTEx for those that are new to it.

3.1 Purpose of NeTEx

The NeTEx Exchange format is a CEN Technical Standard for exchanging passenger information between computer systems. It covers many types of multimodal data, including stops and lines, timetables and also fare products and prices (Figure 1).

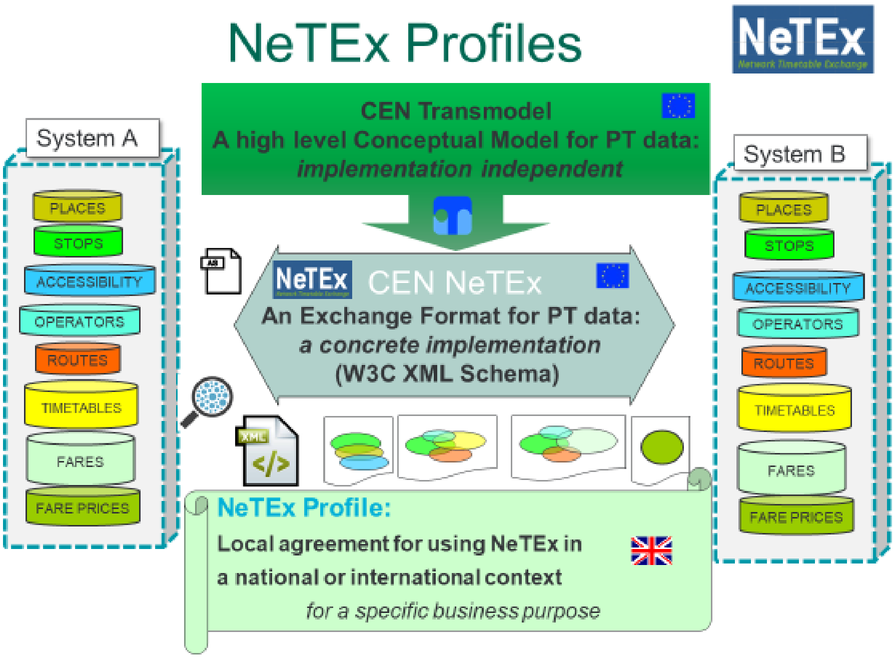

NeTEx Profiles
NeTEx
Network Timetable Exchange
CEN Transmodel
A high level Conceptual Model for PT data:
implementation independent
System A
System B
PLACES
STOPS
ACCESSIBILITY
OPERATORS
ROUTES
TIMETABLES
FARES
FARE PRICES
NeTEx CEN NeTEx
An Exchange Format for PT data:
a concrete implementation
(W3C XML Schema)
XML
</>
NeTEx Profile:
Local agreement for using NeTEx in
a national or international context
for a specific business purpose

Figure 1 – Use of NeTEx to transfer data between Systems

Data is exchanged as XML documents that conform to a precise XML schema that can be used to check that the contents have the right data types and values and are in the right order and self-consistent.

XML documents are electronic files where the start and stop of element is indicated with a tag. The allowed set of tags, as well as the rules for combining tags and validating the data are described by an XML schema a structured specification that can be read and interpreted by a computer (HTML, the tag language used to mark-up web pages is perhaps the best-known example of XML use). There are different schemas with different tags for different purposes; and just as there is one for HTML, there is also one for public transport data - NeTEx. XML can be contrasted with traditional ‘flat file’ formats such as CSV (Comma Separated variables) which comprise simple records comprising strings of bytes delimited with a comma. A schema gives a detailed definition of the meaning and structure of the content structure, making it possible to validate and process the data with general purpose tools.

NeTEx is designed to allow the integration of data coming from different systems using different identifier systems. It uses the w3C domain system to define globally unique namespaces for identifiers. This is an essential capability for supporting an open data model that is “future proof”.

Which is a document

3.2 Purpose of a NeTEx Profile

Transmodel has a wide scope and covers many different aspects of Public Transport, and many different phases from planning though to operations. NeTEx currently only covers a subset of its scope, providing in particular a concrete format to exchange static reference data elements relevant for passenger information, such as network elements, timetables and fares. A sister standard, SIRI is used to access dynamic data elements in real-time.

Even though it covers only a subset of Transmodel, NeTEx is nonetheless a rich and extensive standard and generalised to support a wide variety of use cases. A given implementation will typically only need to use a specific and limited set of the available NeTEx elements. Furthermore, implementors will normally want to impose additional constraints on their use of data (for example as to stakeholders, modes, data identifiers, granularity of data exchanges, etc) so as to simplify the task of implementation. To describe this set of restrictions a 'NeTEx Profile' is specified.

A profile in particular sets out:

- **Content**: Which data elements, data attributes and values for data attributes must be present (or should not be present).
- **Granularity** How the elements should be organised as documents; for example, ”there should be a separate document for each timetable of each line”, or “all the timetables of an operator should be in a single document”.
- **Organisation:** How the elements should be organised within documents. In a few cases NeTEx allows alternatives as to which choice of frames and the placement of specific elements. Limiting the choice can simplify an implementation.
- **Identifiers**: The code spaces and identifier systems to be used to ensure data elements are unique
- **Validation**: Any relevant specific local semantic rule s for validation and interpretation that at are additional to the regular NeTEx model.
- **Mapping**: Any relevant mapping to legacy data sets

NeTEx includes mechanisms to mark a given document as conforming to a particular profile (See Type of Frame add REF) and version of a profile so that a computer reading a NeTEx document can determine how it should be interpreted.

3.2.1 Extended profiles

Depending on its objectives, a profile may mandate different levels of conformance be followed.

1. **Strict conformance:** populating only those element types and attributes specified in the profile. An application consuming a document must be able to interpret and process all content elements.
2. **Augmented conformance:** populating only those element types specified in the profile but also allowing (i) extra attributes for the agreed elements (ii) the use of further enumeration values for the data attributes (iii) adding ***KeyValue*** instances. An application consuming a document may ignore such extra elements if they are irrelevant.

3. **Extended conformance:** Allowing the presence of additional types of NeTEx elements on an optional basis. An application consuming a document may ignore additional elements.

As discussed later, the EIRE_NP profile in effect specifies augmented conformance to the EPIP profile, that it allows additional elements from the schema to be present.

3.3 Use of NeTEx for Stop data

NeTEx includes elements with which to describe the stops and routes of a transport network for reference in timetables, fare tariffs, etc

These correspond to the concepts of Stop Point, Route etc found in NaPTAN and other standards such as GTFS.

The element may be associated with topographic places – equivalent to the Localities used in NaPTAN and NPTG.

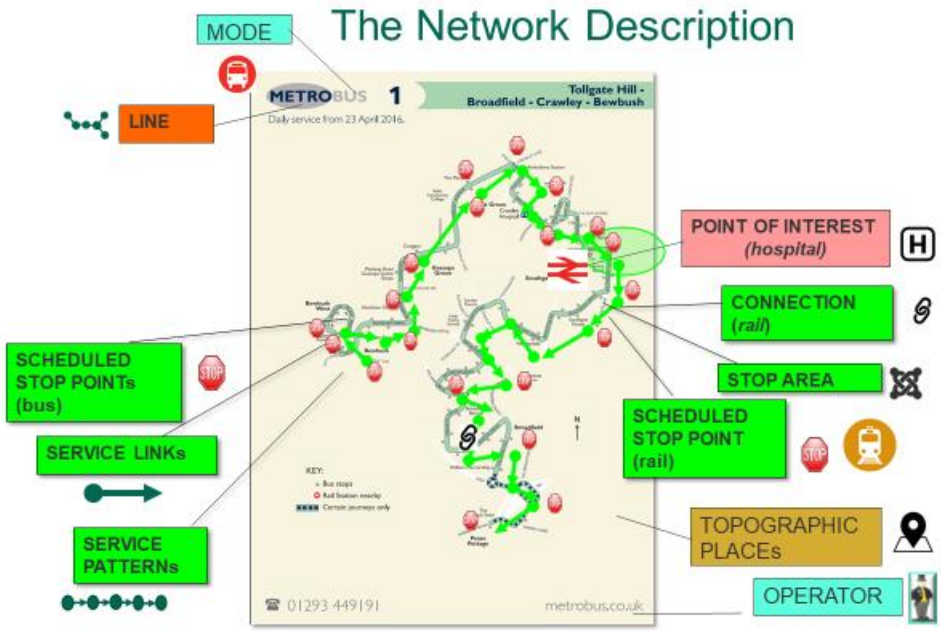


Figure 2 – Stops, routes and service patterns

3.4 Use of NeTEx for Timetable data

NeTEx can be used to defined timetables for any mode of transport, including both absolute and frequency-based timings, for fixed and flexible transport, and with many other features. Journeys may be based on shared journey patterns.

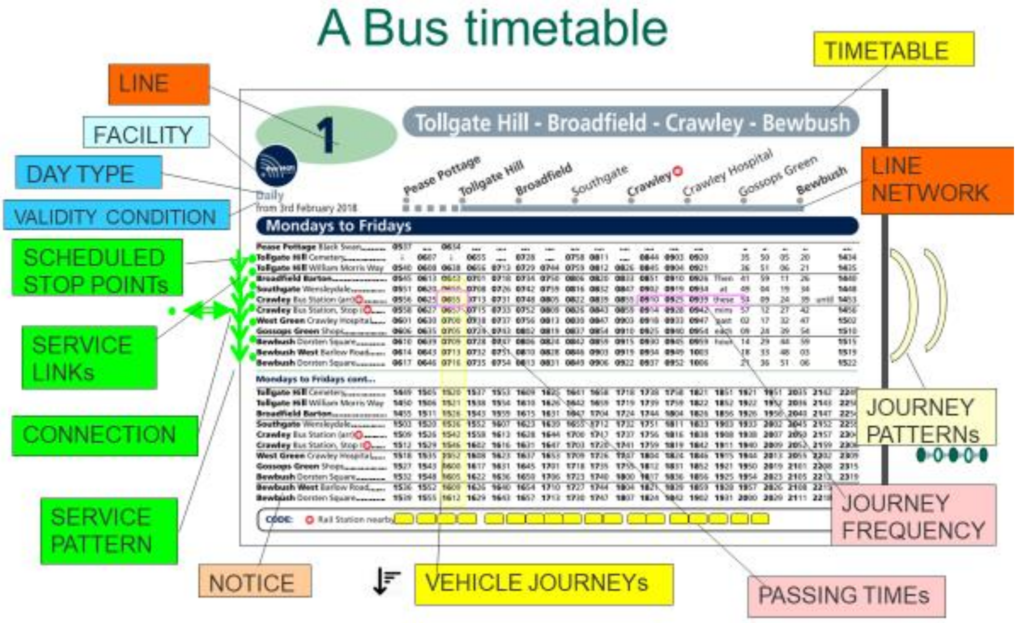


Figure 3 – Representing a bus timetable

The Transmodel representation of a timetable that is used in NeTEx separates out different concerns into separate layers that may be defined and reused independently for different purpose.

Thus, for example a service pattern might be used for both a timetable, for operation and for a tariff.

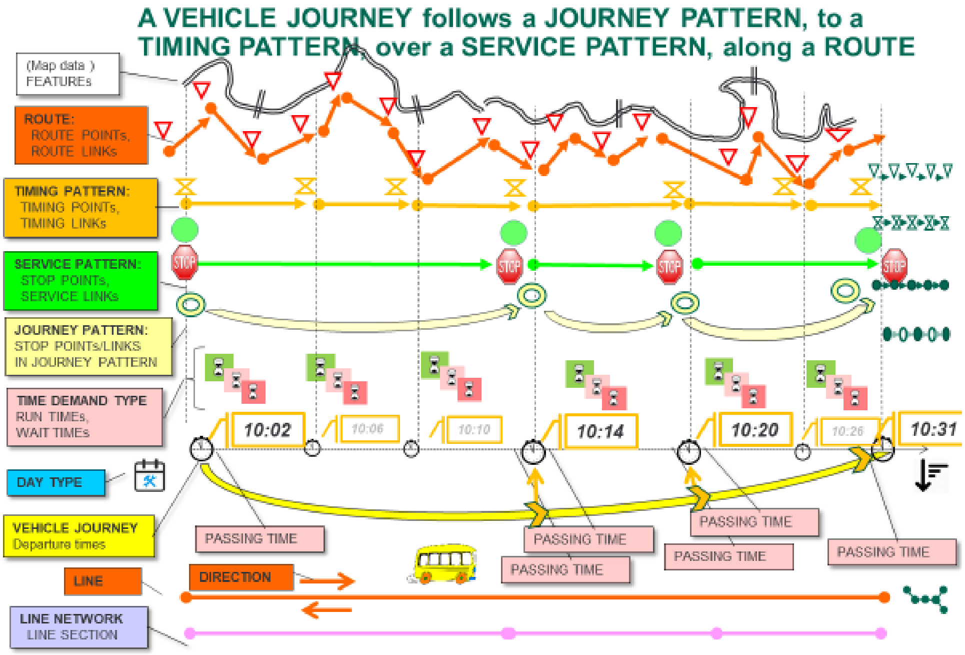

A VEHICLE JOURNEY follows a JOURNEY PATTERN, to a TIMING PATTERN, over a SERVICE PATTERN, along a ROUTE
(Map data) FEATUREs
ROUTE: ROUTE POINTs, ROUTE LINKs
TIMING PATTERN: TIMING POINTs, TIMING LINKs
SERVICE PATTERN: STOP POINTs, SERVICE LINKs
JOURNEY PATTERN: STOP POINTs/LINKS IN JOURNEY PATTERN
TIME DEMAND TYPE RUN TIMEs, WAIT TIMEs
DAY TYPE
VEHICLE JOURNEY Departure times
LINE
LINE NETWORK LINE SECTION
STOP
10:02
10:06
10:10
10:14
10:20
10:26
10:31
PASSING TIME
PASSING TIME
PASSING TIME
PASSING TIME
PASSING TIME
PASSING TIME
PASSING TIME
DIRECTION

Figure 4 – Components of a bus timetable

There may thus be multiple journeys sharing the same journey pattern and timing patterns. This allows for an efficient (that is precise but compact) description of journeys and other components.

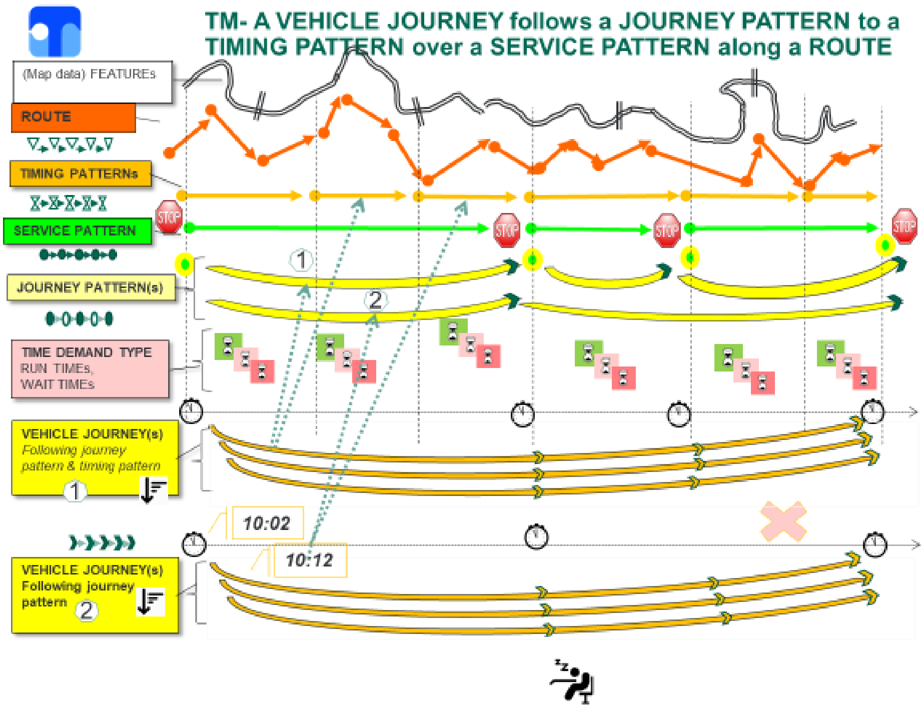

TM- A VEHICLE JOURNEY follows a JOURNEY PATTERN to a TIMING PATTERN over a SERVICE PATTERN along a ROUTE
(Map data) FEATURES
ROUTE
TIMING PATTERNs
SERVICE PATTERN
JOURNEY PATTERN(s)
TIME DEMAND TYPE
RUN TIMEs,
WAIT TIMEs
VEHICLE JOURNEY(s)
Following journey pattern & timing pattern
1
10:02
10:12
VEHICLE JOURNEY(s)
Following journey pattern
2
STOP
1
2

Figure 5 – Journeys making up a bus timetable

3.5 Other Introductory comments

3.5.1 Relationship to Transmodel

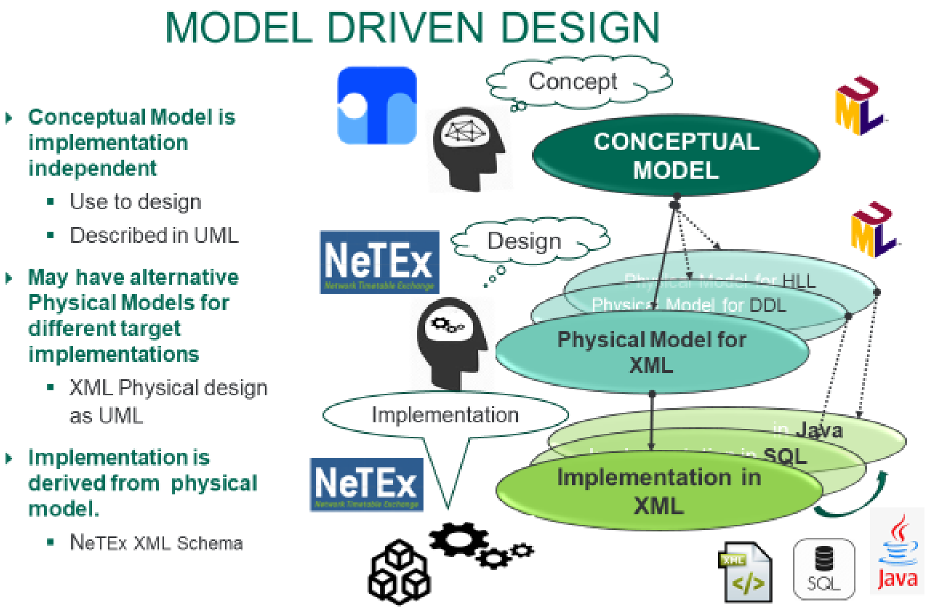
NeTEx itself is based on the European reference model for Public Transport data, *Transmodel*, which sets out an extensive conceptual model of entities and relationships describing many different functional areas of passenger information and transport operation as well as fares and control. Transmodel has recently (2016-2018) been revised to create an updated version 6.0, modularised into 8 functional parts. (Fare Management is part 5 of Transmodel)

- Transmodel as a conceptual model is concerned primarily with entities and relationships and prescribes only a basic set of attributes to describe the actual properties of elements.

- NeTEx, as a concrete format, provides a detailed set of elements and attributes and makes a number of implementation decisions as to how elements are grouped and identified for exchange, as well as assigning data types, identifiers, etc. to each attribute.

NeTEx implements only a subset of Transmodel 6.0, including most of Transmodel Part 1,2 and 3 covering timetables). Transmodel 6.0 includes some additional features not yet implemented in NeTEx 1.1

Transmodel defines a generalised, implementation independent model for PT data, using a consistent vocabulary and is specified in UML, a standard software modelling language. NeTEx transforms this general model, firstly into a design model for a physical implementation, and then into a W3C schema. Among the key benefits of this “model driven” approach are; (i) a systematic, modular, design with a high level of reuse of components and; (ii) the ability to evolve the model readily to meet future needs.



MODEL DRIVEN DESIGN

- Conceptual Model is implementation independent
  - Use to design
  - Described in UML
- May have alternative Physical Models for different target implementations
  - XML Physical design as UML
- Implementation is derived from physical model.
  - NeTEx XML Schema

Figure 6 – Model driven design

3.5.2 Caution on terminology

Transmodel & NeTEx use the same systematic vocabulary of terms for specific domain concepts (LINE, NETWORK, VEHICLE, TRANSPORT MODE, etc). Often the usage will be straightforward and the meaning intuitive (POINT OF INTEREST, TIMING POINT, FARE ZONE, etc.). The use of precise terms for each specific concept has major advantages for system specification and analysis, however it can impede the ready use of the standard;

- In some cases, these terms may appear non-intuitive or cumbersome (e.g., PASSENGER STOP ASSIGNMENT, TEMPLATE SERVICE JOURNEY, SERVICE LINK IN JOURNEY PATTERN, etc) and it requires some effort to become familiar with the usage.
- In other cases, where the use of the term corresponds to just one of several senses that a term has in common vernacular use (for example LINE vs ROUTE; DIRECTION vs DESTINATION DISPLAY, FARE PRICE vs TARIFF vs FARE PRODUCT vs SALES OFFER PACKAGE, etc), care should be taken to choose the correct term and to avoid confusion with the other vernacular senses.

3.5.3 XML schema technology

NeTEx uses W3C XML schema, a very well- established mainstream technology for describing data sets. W3C schema are used to describe documents for many of the most commonly exchanged data sets (HTML, Microsoft Office, etc) globally. Although relatively verbose compared to unvalidated formats (such as CSV or JSON), it has a number of important advantages for the long-term management of complex data sets.

- **Data quality:** The schema provides a detailed description of the expected content which can be used to perform many basic validity checks for data quality and referential integrity automatically. This is of great value for peer to peer exchange of data between many different stakeholders.
- **Flexibility:** The self-describing nature of schema documents makes it possible build applications that use data selectively, for many different purposes, and to evolve the schema without breaking backwards compatibility. Data elements can be used or ignored as needed.
- **Machine-readable**- the self-describing format can be used to automate most of the legwork of implementation, building parser bindings that present the data to a program interface in a ready to use form.
- **Widespread adoption:** there are a wide range of tools and many people with the necessary technology skill sets to draw on.

3.5.4 Evolving NeTEx

NeTEx is designed to be modular and extensible under a uniform framework.

As a CEN standard, NeTEx has a process and active working group to maintain and enhance the standard to meet additional business requirements. CEN has an establish governance process to develop and enhance standards by a consensus process, with formal review and voting procedures, NeTEx is managed by CEN TC 278 WG3 SG9.

This profile and other projects are likely to identify additional requirements that can be added to NeTEx and Transmodel in future. Enhancements can include both additional functions, optimised encodings of existing function and clarifications as to use.

Successive releases of NeTEx are identified by a version number; this study is based on NeTEx 1.1, a recently completed revised and enhanced version of NeTEx due for final release in 2019.

3.5.5 Handling large date sets

Public transport data sets can be technically challenging to standardise, not only because they cover a complex domain, but also because they can be large, requiring consideration of the scaling implications for processing. Furthermore, data and subsets of data are often exchanged repeatedly between many stakeholders, necessitating a careful coordination of versions in order to achieve coherency (with necessary metadata to achieve this). The NeTEx framework includes mechanisms to assist with the management of large datasets including:

❖ **Frames**: A container mechanism that allows data to be modularised into coherent subsets. For example, “all the Season ticket products for a specific operator”, or “all the Network elements describing a line and its stops”. There are different types of version frame for exchanging different types of data (Figure 7). Multiple frames may be placed in a single document so as to organise data into reusable subsets of different types of element.

❖ **Groups**: Groups of elements of a specific type may be grouped in named lists, allowing their repeated reuse as sets, and the assignment of common properties. For example, groups of sales offerings may be given shared properties, so that many similar products can be efficiently defined just by their additional properties.

❖ **Versioning and Validity conditions**. Fare management especially for multi-operator products will typically involve the exchange and aggregation of data from multiple stakeholders. There will be successive versions of data over time with updates having a specific validity period.

- **Systematic global identifiers**. In order to integrate diverse data sets a persistent, globally unique way of identifying elements of any types is needed. NeTEx employs a powerful uniform system of codespaces based on W3C namespaces.
- **Referential integrity checks:** Typically, a given data set will require a coherent set of elements all to be present in order to be usable. NeTEx makes extensive use of XML's built in validity-checking mechanisms to validate internal cross-references within a document to ensure completeness. Errors in data types, tag use and completeness will be automatically flagged by any off the shelf XML validator. For external references to entities in other documents a uniform approach (including versioned references) is used that facilitates programmatic checking.

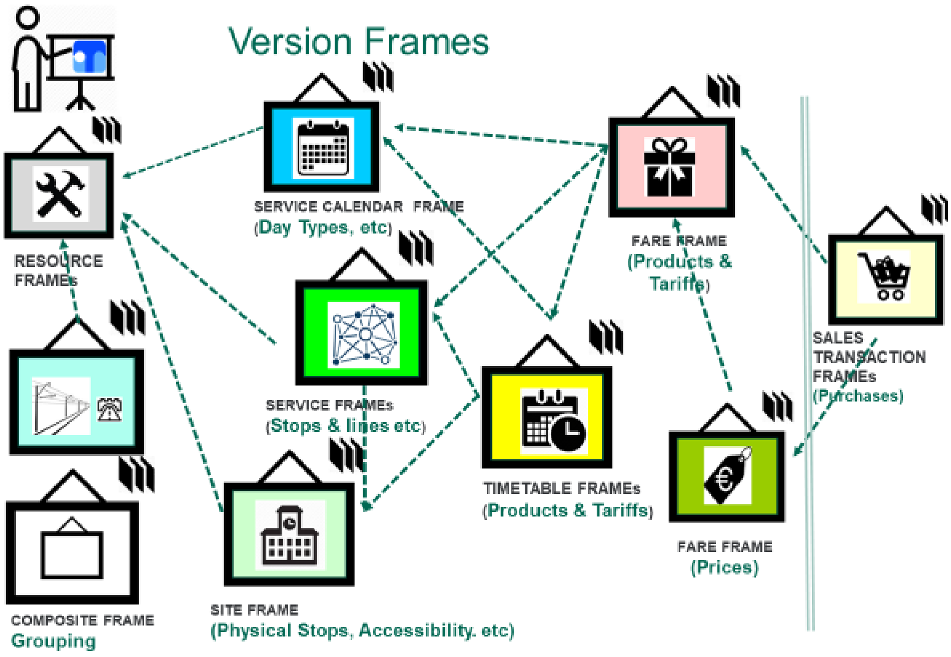


Figure 7 – Types of Version frame

4 The Eire NeTEx Profile

4.1 Profile Scope

The main purpose of the Irish Profile is to exchange timetable data for use in passenger information systems within Ireland and between Ireland and other countries. The profile requires;

1. Use of some common NeTEx Framework features (NeTEx Part1), as described in this document. These are based on Transmodel 6.0 Part1.
2. Use of use some common NeTEx Network components such as stops and lines (NeTEx Part1 & NeTEx Part2. These are based on Transmodel 6.0 Part2.

3. Use of a NeTEx Timetable components model (NeTEx Part2), These are based on Transmodel 6.0 Part3.

Note that this profile document omits many NeTEx capabilities and elements that are not needed for the Irish Profile.

A secondary purpose of the Irish Profile is to show how basic Irish data for stops and timetables from NaPTAN and various timetable formats can be mapped into the EPIP to provide an Irish data set that is conformant to the European profile.

4.2 The Passenger Information Profile

The Irish profile document is derived from the documentation for the common European Passenger Information Profile (EPIP) for exchanging stop and network (TS 16614:PI Profile) data, itself based on the full NeTEx specification v1.1 (EN 16614-1/2/3) and follows quite closely structure and content of the EPIP document. One of the advantages of this approach is to make it easier for those software developers who need to support both Eire and European profiles. The Irish Profile contains some further adaptions and extensions to meet Irish requirements. Where there is a significant difference form the EU profile this is noted. The document also draws on the UK National Profile to obtain mappings of NaPTAN and NPTG data

This Irish Profile specification can be regarded as condensed and focused version of the full NeTEx specification.

The EIRE_NP can be regarded as an 'extended' version of the EPIP. That is, it corresponds closely to the EPIP, but extends it by also including a few further NeTEx attributes and elements that are omitted from the EPIP but are available in the Irish data sets.

4.3 Primary Use case for the Passenger Information Profile

The EIRE_NP profile, like the EPIP, focuses on information relevant for feeding passenger information services; it therefore excludes most operational information that is not relevant for this purpose.

Typical use cases for the Passenger Information Profile are:

- Provision of data to a journey planner.
- Provision of data to a mapping tool to show the network on a map (possibly interactive).
- Provision of data to timetable printing/visualisation tools.
- Provision of data to a stop or line finder.

The profile has been designed to be concise and to focus on the data elements needed to fulfil the EC's Priority action A of ITS-Directive.

- In the interests of simplicity, a number of the more complex features specific to certain modes, in particular Rail, have also been omitted from the formal profile; for example, boarding position on train platforms

The profile supports certain common features that are important for particular subgroups of passengers:

- The ability to exchange translations of text elements in multiple alternative national languages.
- Basic accessibility information about sites and services.

For compatibility, the EIRE_NP profile retains all of the EPIP data elements, even though some of them are not relevant for current Irish business cases (i.e. no elements are dropped, though not all need to be used).

4.4 Use of the profile with Irish public transport data

The Irish NeTEx profile is intended to facilitate the creation of an Irish National Access Point (NAP), providing a central definitive source of transport data for national and international use.

A specific purpose of the Irish profile is thus to show how existing Irish stop and timetable data can be exchanged in a NeTEx XML document that is conformant with the common European Passenger Information profile.

The data comes from a variety of different legacy Irish data sets:

- **Operator data** comes from the Irish NOC data set. It is also possible to export operator data from the current NTA database (currently using MDV's DIVA data management software). Operator data of individual operators is also given in specific individual timetables.
- **Place data:** comes from the Irish NPTG data set provides a gazetteer of place (e.g., cities, towns and villages) s in Ireland served by public transport. This includes names in both English and Gaelic.
- **Stop data for** comes from NaPTAN. Note that bus and tram data is held as on street NaPTAN StopPoints. Rail, and Coach stations are indicated by a NaPTAN ***StopArea.***
- **Stop data for rail** also comes from Irish rail – using UIC codes for stops
- **Timetable data** comes in several different formats.
    - **Bus & Ferry:** Data: primarily VDV452 from the NTA database
    - **Tram**: available in GTFS format.
    - **Rail:** in HaFas format (The proprietary format for Hacon Gmbh – with similar content to VDV462) from Irish Rail.
- Accessibility attributes from the Inspire Transport Networks theme can also be integrated.

4.5 Use of Operator data

The EIRE_NP profile includes basic data about operators, taken from the existing Irish *National Operator Code* (NOC) database. Operator data is used with passenger information to indicate the providers and operators of services and to give contact details for answering queries and complaints. The NeTEx profile also be used to exchange the operator database as a single document.

In practice datasets for timetables and fares are usually provided by and specific to a given operator, so the use of unique operator identifiers is convenient for establishing unique codespaces for such data.

See Part 2 for a detailed mapping.

4.6 Use of NPTG and NaPTAN data

The EIRE_NP profile shows how the existing Irish National data sets for places (NPTG) and stops (NaPTAN) can be exchanged as XML documents conforming to the profile.

The mappings are summarised here. A more detailed mapping is given in Part2.

4.1 Summary of Irish NPTG Stop Mapping

- An NPTG ***NptgLocality*** is mapped to a NeTEx TOPOGRAPHIC PLACE.
- An NPTG ***AdministrativeArea*** is mapped to a TRANSPORT ADMINISTRATIVE ZONE, together with a RESPONSIBILITY SET that indicates the ORGANISATION responsible for managing data for the area and one or more CODESPACE ASSIGNMENTs to indicate the code prefix to use for stops in the zone.

4.2 Summary of Irish NaPTAN Stop Mapping

The following is a summary of mapping of Irish NaPTAN Stop data to NeTEx:

- When using stops in timetables
  - For Rail, bus and coach stations, a NaPTAN ***StopArea*** is mapped to a NeTEx STOP AREA.
  - For on street bus and tram stops, a NaPTAN ***StopPoint*** is mapped to a NeTEx SCHEDULED STOP POINT.
- For encoding physical stop data to provide information about stops.
  - For Rail, bus and coach stations, a NaPTAN ***StopArea*** is mapped to a NeTEx STOP PLACE, and STOP PLACE ENTRANCE.
  - For on-street stops, a NaPTAN ***StopArea*** is mapped to a NeTEx STOP PLACE.
  - For on-street stops, a NaPTAN ***StopPoint*** is mapped to a NeTEx QUAY, within the NeTEx STOP PLACE corresponding to NaPTAN ***StopArea.***
  - If there is no NaPTAN ***StopArea*** for a NaPTAN ***StopPoint***, (or pair of points) one is created with the same name as one of the stops. The opportunity should thus be taken to create STOP PLACEs to pair stops that are close together and have the same name..

- A STOP PLACE ASSIGNMENT can be used to associate a given SCHEDULED STOP POINT with a given STOP PLACE and QUAY

4.3 Use of Timetable data

The EIRE_NP profile shows how the Timetable elements in the EPIP can be populated with Irish data for the different Irish modes. The data is currently available in various formats for different transport modes (bus, tram, rail, as introduced in section 4.4 above.

4.3.1 Timetables in GTFS format

Timetable data for some other services, in particular for the LUAS trams, is available in the General Transit Feed Specification (GTFS) format. Developed by Google, GTFS is a simple CSV encoding of timetables using a representation that combines a POINT IN PATTERN with a resolved PASSING TIME FOR each stop . This can be considered an optimised view of the full timing data – equivalent to a CALL in NeTEx).

Unlike VDV452 or NeTEx, GTFS does not handle the more complex aspects of timetable data such as trains that join or split, default transfer times, separate physical stops, etc. A mapping from GTFS is indicated in Part2.

The relevant GTFS elements for mapping to the profile are summarised in Table 1.

Table 1 – GTFS / NeTEx main mappings

GTFS file	Scope		NeTEx element	NeTEx Frame
Agency.txt	Operators and Authorities	→	<Operator>	<ResourceFrame>
Stops.txt	Stops in timetable	→	<ScheduledStopPoint>	<ServiceFrame>
	Physical stops (Stations, stops, platforms, levels, entrances)	→	<StopPlace> <Quay> <Entrance>	<SiteFrame>
Transfer.txt	Interchange times	→	<Connection> <ServiceJourneyInterchange>, <InterchangeRule>	<ServiceFrame>
Routes.txt	Line name and facilities	→	<Line>	<ServiceFrame>
Trips.txt	Journeys	→	<ServiceJourney> + DestinationDisplay>	<ServiceFrame>
StopTimes.txt	Journey times at stops	→	<StopPointInPattern> + <PassingTimes> + <DestinationDisplay> &/ or <Call>	<TimetableFrame>
Frequency.txt	Frequency based journey timings	→	<HeadwayJourneyGroup>, or <RythmicalJourneyGroup> with <TemplateServiceJourney>	<ServiceFrame>
Calendar.txt	Days of operation	→	<DayType>, <OperatingPeriod> <DayTypeAssignment> (in <ServiceCalendar>)	<ServiceCalendarFrame>
Calendar_Dates.txt	Exception to days s of operation	→	<DayTypeAssignment> and <OperatingDay> (in <ServiceCalendar>)	<ServiceCalendarFrame>
Shapes.txt	Route plot	→	<LinkProjection> <LineString>	(parent)
Translations	Translations	→	<AlternativeText>, <AlternativeName>	(parent)
FeedInfo.txt	Validity and data source	→	<DataSource>ValidityCondition>, <ResponsibilitySet>	<CompositeFrame>
attributions.txt	Data credits			
fare_attributes	prices		<FarePrice>	<SiteFrame>
fare_rules	Fare products		<FareProduct>, <SalesOfferPackage>, <UsageParameter> etc	
levels.txt	Station levels		<Level>	
pathways.txt	Path h between stops		<PathLink>	

A more detailed mapping is given in Part2

4.3.1.1 High Level comparisons of GTFS and NeTEx

The GTFS format is optimised for brevity and gives an efficient format for the specific purpose of exchanging finalised timetable data. The NeTEx model is more extensive - and more verbose - than the GTFS representation for three fundamental reasons:

(a) It provides significant additional function, including the planning, scheduling and operational aspects of timetables and has a much richer fare model; it is intended not just for distribution of final timetables to third parties, but also for many other use cases, including the exchange of the planning and operational timetables, the exchange of data for operations, etc..

(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.

(c) It uses a self-describing XML schema, allowing the automatic validation of data types and referential integrity checks using off-the-shelf XML validators (GTFS requires a custom validator program to do the same).

In addition, NeTEx is intended to support the distributed peer-to-peer exchange and integration of data. It has mechanisms to allow different coding systems to be used alongside each other.

4.3.1.2 Comparison of GTFS and NeTEx representations of a timetable

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 shows the Transmodel representation of a route and its timetable as a set of informational layers, each concerned with different aspects of the problem domain (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 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 8 Transmodel representation of a timetabled journey

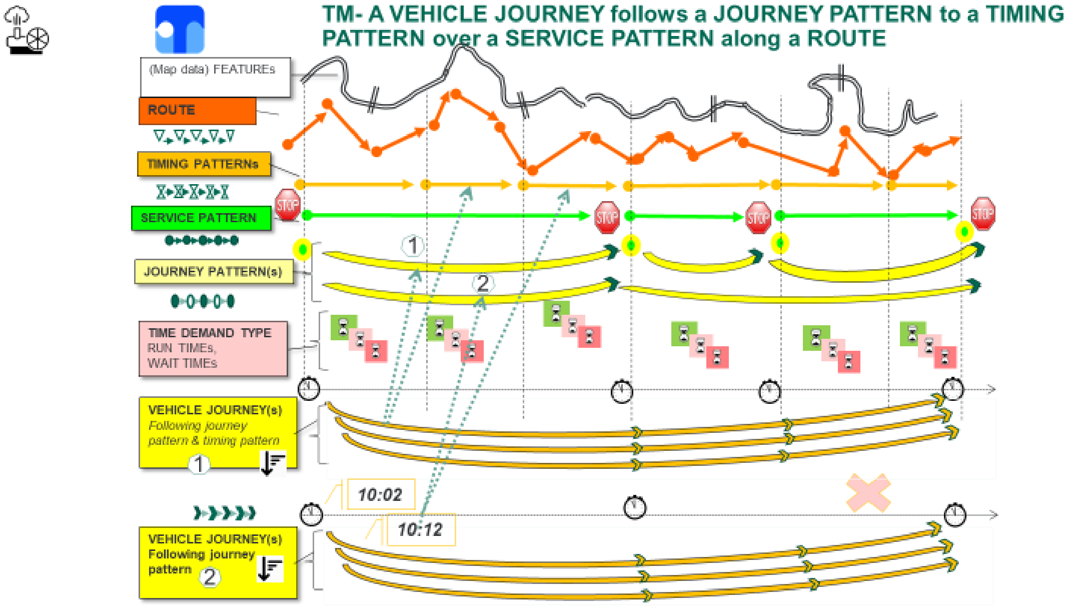
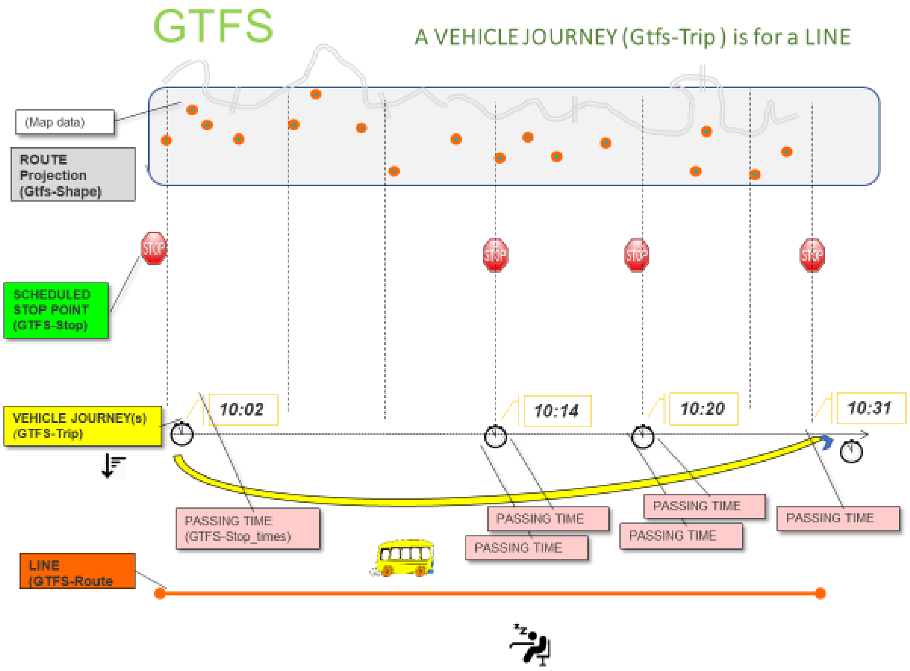

TM- A VEHICLE JOURNEY follows a JOURNEY PATTERN to a TIMING PATTERN over a SERVICE PATTERN along a ROUTE
(Map data) FEATUREs
ROUTE
TIMING PATTERNs
SERVICE PATTERN
JOURNEY PATTERN(s)
TIME DEMAND TYPE
RUN TIMEs,
WAIT TIMEs
VEHICLE JOURNEY(s)
Following journey pattern & timing pattern
1
VEHICLE JOURNEY(s)
Following journey pattern
2
10:02
10:12
STOP

Figure 9 GTFS representation of a timetabled journey



4.3.2 Timetables in VDV452 format

Timetable data for most bus lines is available in VDV452 format from the NTA database (currently an MDV DIVA database). VDV 452 is a German transport data standard for the exchange of timetables.

A canonical mapping from VDV452 to NeTEx is set out in the VDV publication *Recommendation 462- NeTEx VDV Application Handbook: Standardised exchange of line* *network and timetable data with the European standard CEN-TS 16441,* which provides detailed element by element equivalences. Note that The VDV 452 profile also includes a number of elements that are not present in the Irish bus data and may be omitted.

A high-level mapping from VDV 452 to NeTEx is summarised in the following table. A detailed mapping is given in the VDV462 publication. For the most part, there is a one-to-one correspondence between VDV and NeTEx concepts, and terminology is similar (since they are based on Transmodel).

Note however that VDV format, (like the UK TransXChange format), is intended to cover planning and scheduling use cases and not just the provision of Passenger information; it therefore separates timing data from service data. Thus, RUN TIMEs for a given SERVICE LINK, and WAIT TIMEs at a given stop, are represented separately from the SERVICE PATTERN itself. A JOURNEY itself states only an initial departure time, from which the passing times at stops can be computed using the SERVICE PATTERN and the timing information.

In consequence, these timings must be resolved into simple NeTEx PASSING TIMEs in order to generate an EPIP conformant timetable.

The underlying service and timing pattern information is of use for certain use cases, so the EIRE-NP allows this to be included as an optional profile augmentation if desired. This is an augmentation of the EPIP.

The Irish codespaces should be used on data elements, e.g., NaPTAN for stops.

Examples of the use of VDV452 timetable data are given to show the use of Irish codespaces, etc.

Table 2 VDV452 to NeTEx high level mapping

VDV #	**VDV 452 Class / Parent Element**		**NeTEx Containing Frame name**	**NeTEx Parent Element**
993	**BASE_VERSION_VALID**	→	***CompositeFrame***	***<CompositeFrame>***
485	**BASE_VERSION**	→	***CompositeFrame***	***<CompositeFrame>***
992	**TRANSPORT_COMPANY**	→	***ResourceFrame***	***<Operator>***
333	**OPERATING_DEPARTMENT**	→	***ResourceFrame***	***<OperationalContext>***
443	**VEHICLE**	→	***ResourceFrame***	***<Vehicle>***
293	**VEHICLE_TYPE**	→	***ResourceFrame***	***<VehicleType>***
290	**DAY_TYPE**	→	***ServiceCalendarFrame***	***<DayType>***
291	**OPERATING_DAY**	→	***ServiceCalendarFrame***	***<OperatingDay>***
348	**PERIOD**	→	***ServiceCalendarFrame***	***<ServiceCalendar>***
292	**DAY_TYPE_ASSIGNMENT**	→	***ServiceCalendarFrame***	***<DayTypeAssignment>***
295	**ACTIVATION POINT**	→	***InfrastructureFrame***	***<ActivationPoint>***
997	**STOP_TYPE**	→	***SiteFrame***	***<TypeOfPlace>***
998	**POINT_TYPE**	→	***ServiceFrame***	***<TypeOfPoint>***

246	ROUTE_SEQUENCE	→	*ServiceFrame*	*<ServicePattern>*
226	LINE	→	*ServiceFrame*	*<Route>*
226	LINE	→	*ServiceFrame*	*<JourneyPattern>*
994	DESTINATION	→	*ServiceFrame*	*<DestinationDisplay>*
229	STOP_POINT	→	*ServiceFrame*	*<ScheduledStopPoint>*
253	STOP	→	*ServiceFrame*	*<PassengerStopAssignment>*
299	LINK	→	*ServiceFrame*	*<ServiceLink>*
995	POINT_ON_LINK	→	*ServiceFrame*	*<PointOnLink>*
282	TRAVEL_TIME	→	*ServiceFrame*	*<JourneyPatternRunTime>*
999	WAIT_TIME	→	*ServiceFrame*	*<JourneyPatternWaitTime>*
996	ANNOUNCEMENT	→	*ServiceFrame*	*<Notice>*
332	JOURNEY_TYPE	→	*TimetableFrame*	*<TypeOfService>*
715	JOURNEY	→	*TimetableFrame*	*<VehicleJourney>*
222	TIMING_GROUP	→	*TimetableFrame*	*<TimeDemandType>*
308	JOURNEY_WAIT_TIME	→	*TimetableFrame*	*<VehicleJourneyWaitTime>*
432	SERVICE JOURNEY INTERCHANGE	→	*TimetableFrame*	*<InterchangeRule>*
232	SERVICE JOURNEY INTERCHANGE TIME	→	*TimetableFrame*	*<InterchangeRule>*
225	DEAD_RUN	→	*TimetableFrame*	*<DeadRun>*
247	DEAD_RUN_TIME	→	*TimetableFrame*	*<DeadRun>*
310	BLOCK	→	*VehicleScheduleFrame*	*<Block>*

4.3.3 Timetables in HaCon HaFas (HRDF)format

Timetable data for Irish rail is available in the proprietary HaFas HRDF (Rohdatenformat or Raw Data format) this is an old proprietary representation using fixed length records in a flat text file. See Part2 for a detailed mapping.

HRDF includes additional files – the ones listed here are just the ones used in the Irish data sample.

Table 3 Summary of Equivalences between HaFas HRDF and NeTEx

HRDF record	Scope	Description	NeTEx equivalent
eckdaten	Data valitiy	Validity period of data	***CompositeFrame /Validity Condition***
betrieb	Undertaking	Transport Operator	***Operator***
bahnhof	Stations	Station codes and names	***ScheduledStopPoint, StopPlace***

koord	Spatial coordinates	Spatial coordinates of stations	***Location***
maincoor	Spatial cordinates for	Spatial coordinates of other points	***Location***
bhfattr	Station Accessibility	Accessibility attributes for a station and other properties	***StopPlace/ AccessibilityAssessment, SiteFacilitySet***
umsteigb	Connections	Connections and transfer times between modes	***Connection/ServiceInterchange***
bhfrt	Permitted station routings	Routings	
kminfo	Ranking	Rrlative rankings for transfers	***Connctione***
fplan	Journeys and Passing timesf	**Service journey and passing times, with annotations**	***ServiceJourney or DatedServiceJourney , JourneyPart, PointinJourneyPattern, PassingTimes, NoticeAssignment, ServiceFacilitySet***
bitfield	Operational days	stores validity days for a specific ID in the form of a bitmap.	***DayTypeAssignment***
gleise	Platforms for	Platform assignments	***PassengerStopAssignment, ValidityCOnditions***
infotext	Notice	Text notices associated with journeys etc	***Notice***
attribut	Facilities	Site & Onboard facilities	***Facility***
metabhf	Station groups	Group of nearby stations	***StopArea***
sortkeys	Sort table for stations		***n/a***
attribue	Transport mode	Mode codes	***TransportMode***
zugart	Train type	“Means of transport”	***Submode, TypeOfService***
vardim	System parameters		***n/a***

4.1 Use of UD Rating Accessibility Data

The NeTEx schema can encode detailed accessibility data about stops, stations and vehicles.

The NTA's classification system for accessibility at bus stops – see PTIMS- *Universal Design Rating* [6] – can be mapped to NeTEx.

The UD rating system assigns a rating to a stop, either by manual survey or as computed from features such as shelter, pole, seating kerb, etc.

In NeTEx accessibility information is held on an ACCESSIBILITY ASSESSMENT element

Table 4 Summary of UD rating characteristics

UD Rating	Characteristics
A+	As for A with additional facilities, e.g., real-time
A	• Marked with a pole/shelter and plate code • Readily available and accurate information at stop: • Safe footpath to junctions nearby • Cassle kerbing/kerbing of sufficient height • Enough space for wheelchair users to wait and board • Well lit
B	• Marked with a pole/shelter • Accessible by a footpath • Having a kerbing which allows boarding onto a bus at the stop
C	• Not wheelchair accessible • No stop markings • Lack of infrastructure such as footpaths to junctions and kerbing
Unknown	Not yet classified

4.2 Data consistency and Quality

One of the goals of a profile is to simplify and enhance interoperability. However, mere syntactic compliance with the profile alone will not guarantee interoperability; the data being exchanged must also be of good quality, that is, complete and self-consistent as data, -- and correspond accurately to the external reality it is meant to be describing.

The profile therefore also includes a set of consistency and quality checking rules. Following these rules will not, of course, necessarily guarantee the absolute quality of any dataset (nor will it validate the data against the real-world), but will prevent many of the basic quality issues that are commonly encountered.

Many basic checks can be performed using any existing off the shelf XML validator (an XML validator checks a document against the specific rules encoded in a schema). A number of further domain specific rules might be checked automatically using a specially programmed validator. Such a validator could check for example that a SERVICE JOURNEY has PASSING

TIMEs for each stop, and that these are plausibly spaced. It could also check references to stops, localities operators, etc., against existing values in a database.

However, there are other quality measures that cannot necessarily be checked by a computer. For example, the lack of a STOP PLACE name can be detected, as can the use of a duplicate, but whether it is actually the correct one in use in the real-world requires human verification.

Additional quality checking rules are discussed later below in chapter 11.

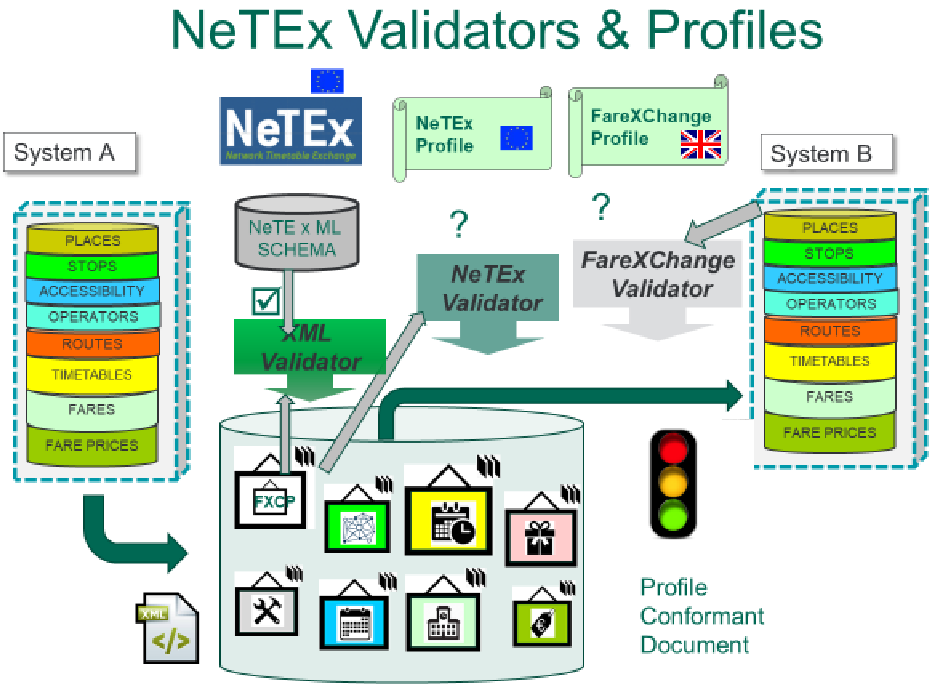


Figure 1 — Use of validators to check quality

4.3 NeTEx versions

4.3.1 NeTEx Specification Versions

Both the EPIP and the EIRE_NP specifications are based on the revised version 1.1 of NeTEx specification , issued in 2019.

If recourse to the NeTEx specification is needed, it is strongly recommended that the 1.1 version is used as it contains numerous clarifications and corrections to the original 1.0 document.

4.3.2 NeTEx Schema Versions

Likewise, the NeTEx 1.1 XML schema should be used.

NB The NeTEx working group keeps a stable updated ‘point release’ version of the schema which includes corrections and minor additions (e.g., a few additional enum values) to the original 1.1 schema; this is available on Github. This updated should be used in Irish implementations.

A ZIP of the recommended version is available on the Irish NeTEx profile site. httP://Netex.ie

5 Terms and definitions

For the purposes of this document, the terms and definitions given in EN 12896 (Transmodel V6) and CEN/TS 16614 (NeTEx) and the following apply.

5.1 attribute

property of an entity.

5.2 conceptual data model

description of a real-world domain in terms of entities, relationships and attributes, in an implementation independent manner. It should provide a structure on which the rest of the development of an application system can be based.

5.3 conformant

satisfying all the rules of a specification, both syntactic and otherwise; conformance may be strict, augmented, or extended.

5.4 consumer

a system that imports profile conformant data.

5.5 data domain

data structure (in this European Standard, a part of the Reference Data Model for Public Transport) made up of data related to each other, through the fact that there is a functional area or group of functions using this data set as a whole.

5.6 data model

description of a real-world domain in terms of data and relationships.

5.7 data provider

organisation responsible for providing data (this may be distinct from that of the data source or the producer).

5.8 dominant validity condition

validity condition that attaches to a version frame and applies to all its contents.

5.9 entity

object (data) that has its own existence (as opposed to an attribute).

5.10 external reference

a reference to another object in a document other than the document holding the reference.

5.11 fare management

all activities related to the collection of money from passengers.

5.12 functional area

arbitrarily defined set of activities, used, in this European Standard, to define the objectives and limits of the data model and exchange profile.

5.13 internal reference

a reference to another object in the same document as that holding the reference.

5.14 interoperability

ability of (sub)systems to interact with other (sub)systems according to a set of predefined rules (interface).

5.15 logical normalised model

relational data model that is not fully normalised, i.e., does not completely follow the normalisation rules and thus may be redundant.

5.16 object-oriented data model

data structure expressed according to principles that allow for a direct implementation as an object-oriented database, where information is represented in form of objects, i.e., respecting the principle of encapsulation meaning in particular that each data is accessed or modified through operations (methods) belonging to it.

5.17 operations monitoring and control

all activities related to the transportation process, i.e., real-time functions related to the driving and transportation of passengers according to given instructions, including the monitoring of the driving process and its control in case of deviations, as well as all activities that support the driving process (traffic light priority, track switching, bay selection, advance/delay advice, etc.). Such functions are often assisted by computer-aided tools, known as Automated Vehicle Monitoring (AVM).

5.18 passenger information

all activities related to informing the users either about the planned or about the actual transportation services.

5.19 producer

a system that outputs data intended to be conformant to a profile.

5.20 profile

subset of a standard selecting the needed concepts (entities and associated attributes) for a specific use case or set of use cases, and complemented by rules defined to restrict possibilities of divergent interpretations of open parts of the standard.

5.21 profile code value

metadata used to describe a profile; a standardised code value, unique with the namespace of the profile, that helps define the profile.

5.22 profile metadata

data describing the properties of a profile, such as the permitted frame types and restricted code values.

5.23 profile validator

software tool that checks the content of a profile conformant XML document to ensure that it is compliant with the profile specification.

5.24 real-time control

see Operations monitoring and control.

5.25 scheduling

see Tactical Planning.

5.26 specific frame

version frame that may only contain certain types entities concerned with a particular functional area, e.g., to describe a timetable.

5.27 strict conformance

conformance to a profile such that a document contains only the specified elements (i.e., that is not augmented or extended).

5.28 tactical planning

all activities related to the tactical planning of transportation, split into vehicle scheduling, driver scheduling, rostering.

5.29 XML document

A file containing data marked up with XML tags conforming exactly to an XML schema that specifies precise syntactic rules for the names, nesting and cardinalities of the tags and for the formats of the data values.

5.30 XML validator

A software tool that checks the content of an XML document to ensure that it is compliant with a schema.

6 Symbols and abbreviations

For the purposes of this document, the symbols and abbreviations given in CEN/TS 16614-1:2019 apply.

In addition:

AVMS	Automatic Vehicle Management System
DIVA	Dialoggesteuerte Informations und Verkehrsauskunft („Dialog controlled Traffic information”).
EPIP	European Passenger Information Profile.
EIRE_NP	Irish National NeTEx Passenger Information profile.
GIS	Geographical Information System.
GML	Geospatial Markup Language.
GTFS	General Transit Feed Specification
HaCon	Hannover Consulting (Creators of HaFas))
HaFas	HaCon Fahrplan-Auskunfts-System (Hacon Timetable Information System)
ITM	Irish Transverse Mercator (Irish OS Grid)
HRDF	HaFas Rohdatenformat (Raw Data) Format.

MDV	Mentz Datenverarbeitung GmbH (Creators of DIVA, a data management tool used by NTA)
NaPTAN	National Public Transport Access Node – specification & dataset
NPTG	National Public Transport Gazetteer – specification & dataset
NTA	National Transport Authority (Ireland)
NOC	National Operator Codes
NUTS	Nomenclature of Territorial Units for Statistics (Europe)
PTAN	Public Transport Access Node.
PTIMS	Public Transport Infrastructure Management System
RGB	Red Green Blue (Colour encoding system)
SBB	Schweitzer Bahn (Swiss Railways)
SMS	Short Message Service
NDA	National Disability Authority
UD Rating	Universal Design Rating

7 Presentation conventions

The presentation conventions used in this document are an extension of the ones used in the main NeTEx specification (and the same as those used in the EPIP).

7.1 Recap of NeTEx presentation conventions

NeTEx follows certain conventions for the presentation of technical terms:

- Transmodel conceptual model elements (‘ENTITIES’) are shown in UPPER CASE, for example, “LINE”, “SCHEDULED STOP POINT”. For readability, they may be pluralised with a lower case ‘s’, e.g., “some LINEs.”

- All XML elements in the NeTEx XML schema are shown in ***bold italic,*** *f*or example, ***Line***. Compound names are camel cased without a space and are never pluralised, e.g., “***ScheduledStopPoint”.***

- Properties that are implemented as the XML class attributes of a complex class are in ***bold lower-case italic***, for example, “***id”***, “***order”***.

- Properties of an entity that are implemented as an XML child element, are also capitalised and shown in bold for example, “***ScheduledStopPoint*** / ***Name”, “SiteElement / IsCovered***”. This is regardless of whether the child element is a simple XML type (e.g., *xsd:boolean*’) or a complex XML type (e.g., *PresentationStructure*).

- Restricted lists of enumeration values, (which are mostly in *lowerCamelCase)* are shown in *italics*; where values appear within free text, they are encased in single quotes to distinguish them from the plain text words, for example, ‘*busStop’, ‘canalBarge’, ‘rail’*.

- One-to-one relationships that are implemented as versioned references are shown as simple attribute names, for example, ***ScheduledStopPointRef, ZoneRef,*** though in fact the type of the ‘Ref’ attribute is a complex structure comprising both an identifier and a version number. When there is a reflexive relationship, or more than one relationship between the same two types the attribute names are qualified by a prefix to distinguish between them for example, ***ParentZoneRef, FromPointRef, ToPointRef***, etc.

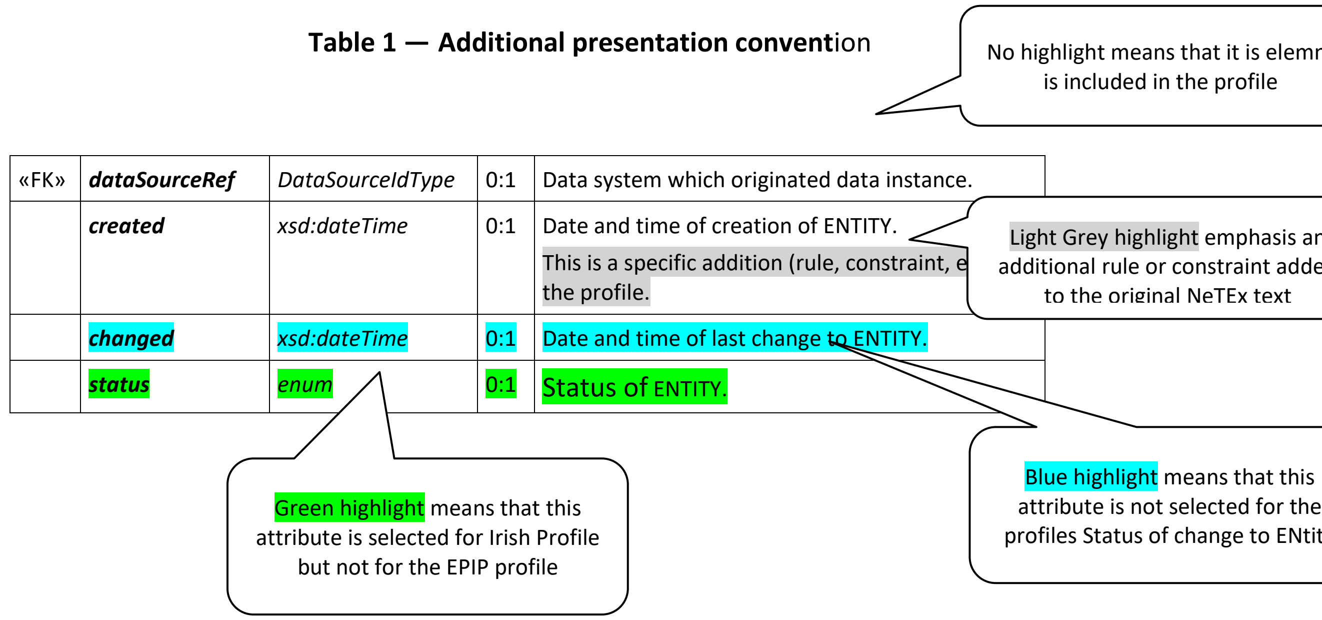
- One-to-many relationships that are implemented as collections whose names are lower camel-cased and are always plural, for example, ***stopPoints***, ***vehicleJourneys, prices;*** thus, *the relationship name provides the name of the wrapper tag which groups the members of the collection.* Such collections may variously comprise sets of simple versioned references (e.g., multiple instances of ***ScheduledStopPointRef)*** or of embedded elements whose full definition is included inline within the enclosing element (e.g., multiple instances of ***ScheduledStopPoint).***
- Data types are shown in *italic*; built-in xml types are shown in *italic* and *lowerCamelCase*, usually with the *xsd* namespace prefix, for example, *xsd:dateTime, xsd:normalizedString, xsd:integer*. NeTEx data types are shown in *italic* and *UpperCamelCase*, without a prefix. e.g., *LengthType, EmailType, DirectionTypeEnumeration*.
- The NeTEx data types that are complex (and hence have a further definition elsewhere) are shown with an underline, e.g., *DataManagedObject*, *Line*, etc. An exception is made for the common utility types such as *MultilingualString*, and all subtypes of *VersionOfObjectRef,* which are shown without an underline.
- All compound names of data types are camel cased without a space, for example, *ScheduledStopPointIdType*.
- Stereotype names are enclosed in guillements (chevrons), for example, «TM VIEW», «enum», «FK».

7.2 Additional profile presentation conventions

Several additional conventions beyond those used in NeTEx are used in the profile in order to cover specific aspects of the EPIP and the EIRE_NP:

(1) **Indication of which attributes are included in the profile.**
The profile uses highlighting to mark up the NeTEx model definitions as follows:

Table 1 — Additional presentation convention



No highlight means that it is elemr
is included in the profile

«FK»	***dataSourceRef***	*DataSourceIdType*	0:1	Data system which originated data instance.
	created	*xsd:dateTime*	0:1	Date and time of creation of ENTITY. This is a specific addition (rule, constraint, e the profile.
	changed	*xsd:dateTime*	0:1	Date and time of last change to ENTITY.
	status	*enum*	0:1	Status of ENTITY.

Light Grey highlight emphasis ar
additional rule or constraint adde
to the original NeTEx text

Green highlight means that this attribute is selected for Irish Profile but not for the EPIP profile

Blue highlight means that this
attribute is not selected for the
profiles Status of change to ENtit

Note: Unselected attributes (highlighted in blue) can still be used since they will still be valid for the NeTEx XSD. However, they are 'extensions' and are not required to be read and understood by the consumer. Moreover, they may be the source of issues for those deciding to use a reduced or optimised XSD. Therefore, using them should only be by agreement between consumer and producer.

EIRE ☞ Notes and comments specific to the Irish implementation are indicated by an Index pointer; ☞.

(2) Profile code values

Wherever possible, the EIRE_NP standardises the code values used to classify and characterise elements to a systematic set of documented values. In NeTEx, several different mechanisms are used to specify the permitted code values;

a. **Fixed enumerations** defined as part of the NeTEx schema (see presentation convention above e.g., '*riverBus'*). The EIRE_NP mandates a subset of the NeTEx codes - see Annex B for the permitted values.

b. **Specialisations** of TYPE OF VALUE, used to define open ended code sets which may be added to over time without changing the schema, and for arbitrary user classifications, for example, to record legacy entity classifications. EIRE_NP itself uses the TYPE OF VALUE mechanism in a few cases to specify additional standardised codes: these are assigned to an '*eix_metadata'* CODESPACE (https://netex.ie/eix) indicated by an *'eix'* prefix. (The PIP similarly uses an epip_data codespace) The identifiers of EIRE_NP values are shown in lower case italics (if necessary, with underscore marks between words), and appear within italics in text, for example, '*eix:administrative_district'*.

c. ***TypeOfFrame*** instances: The EIRE_NP uses several different TYPEs OF FRAME to specify the usage of VERSION FRAMEs in the profile. The identifiers of the ***TypeOfFrame*** instances are shown in upper case italics with underscore marks between words, for example, '*eix:EI_PI_LINE_*OFFER'**.**

(3) Indication of Abstract classes

The NeTEx schema makes extensive use of class *inheritance*; this significantly simplifies specification by avoiding repetition, since shared attributes are declared on a superclass and subclasses are then defined that have the common properties of the superclass without having to repeat them. Most superclasses are 'abstract' – that is, there is no actual instance of them; only the terminal subclasses are 'concrete'.

An inconvenience of using inheritance is that if one wishes to understand the properties of any single concrete class, one must look also at all of its superclasses (and it is not always obvious whether this should best be done before or after studying the concrete class). For this reason, the EIRE_NP profile includes the necessary abstract classes to understand the concrete classes, even though the concrete classes are never directly instantiated in a NeTEx document.

- The superclasses are flagged in headings by the suffix '(**Abstract)'** and are generally presented in top down order of inheritance (so they will appear before the concrete class of actual interest)**.**
- In UML diagrams (as for NeTEx and Transmodel), the names of abstract classes are shown in *italics*.

- Certain superclasses are technically not abstract in the NeTEx, but are not used as concrete classes in the EIRE_NP. These are flagged **'(Abstract in EIRE_NP)'** in this document.

A number of the NeTEx classes have subcomponents that make up their definition. These provide either essential further definitions (e.g., ***StopPointInSequence*** for a ***JourneyPattern***) or ancillary detail (for example, ***AlternativeText***, ***AlternativeName, TrainComponent,*** *et*c.); subcomponents are flagged in headings by the suffix '(**Subcomponent of XXX)',** where XXX is their usual parent. The ancillary details can be ignored on a first reading.

(4) Emphasis of key points to note

In this document, underscore is used to emphasise important points (since italic and bold emphasis is mostly used to distinguish XML elements).

7.3 Transmodel definitions and the EIRE_NP

The NeTEx specification uses the same formal definitions to describe its XML class elements as those used by Transmodel, the European standard reference model for PT data, to describe the corresponding ENTITies.

The EIRE_NP adheres to these definitions but, since it is intended to be as readable as possible, in certain places the EIRE_NP specification simplifies, annotates, reorders or paraphrases the formal definitions in the interests of clarity or brevity. In particular in its narrative, it adds comments and examples as to the purpose and intended use of specific features. It also makes similar revisions to the NeTEx definitions of attributes. Regardless of this, the original Transmodel definitions (For which see Annex A) should be considered definitive for the interpretation of concepts and the NeTEx specification definitive for the interpretation of additional attributes.

Note that for readability, the EIRE_NP specification also uses a different order of presentation of the elements compared to that used in the CEN Transmodel and NeTEx specifications.

8 NeTEx Framework Elements used in Profile

In order to simplify implementation, all NeTEx objects use uniform mechanisms for the common properties that are needed for effective data exchange, such as versioning, validity, ascribing responsibility and for grouping sets of data.

This section describes the common NeTEx Framework that is shared by all NeTEx functional parts in the profile.

8.1 DataManagedObject, Version and Codespace

DataManagedObject, together with ***Version*** and ***Codespace***, provides uniform, common properties for all NeTEx entities, such as an identifier, version etc.. ***DataManagedObject*** is the abstract "ur-object" of the NeTEx implementation and all first class NeTEx entities are specialisations of it.

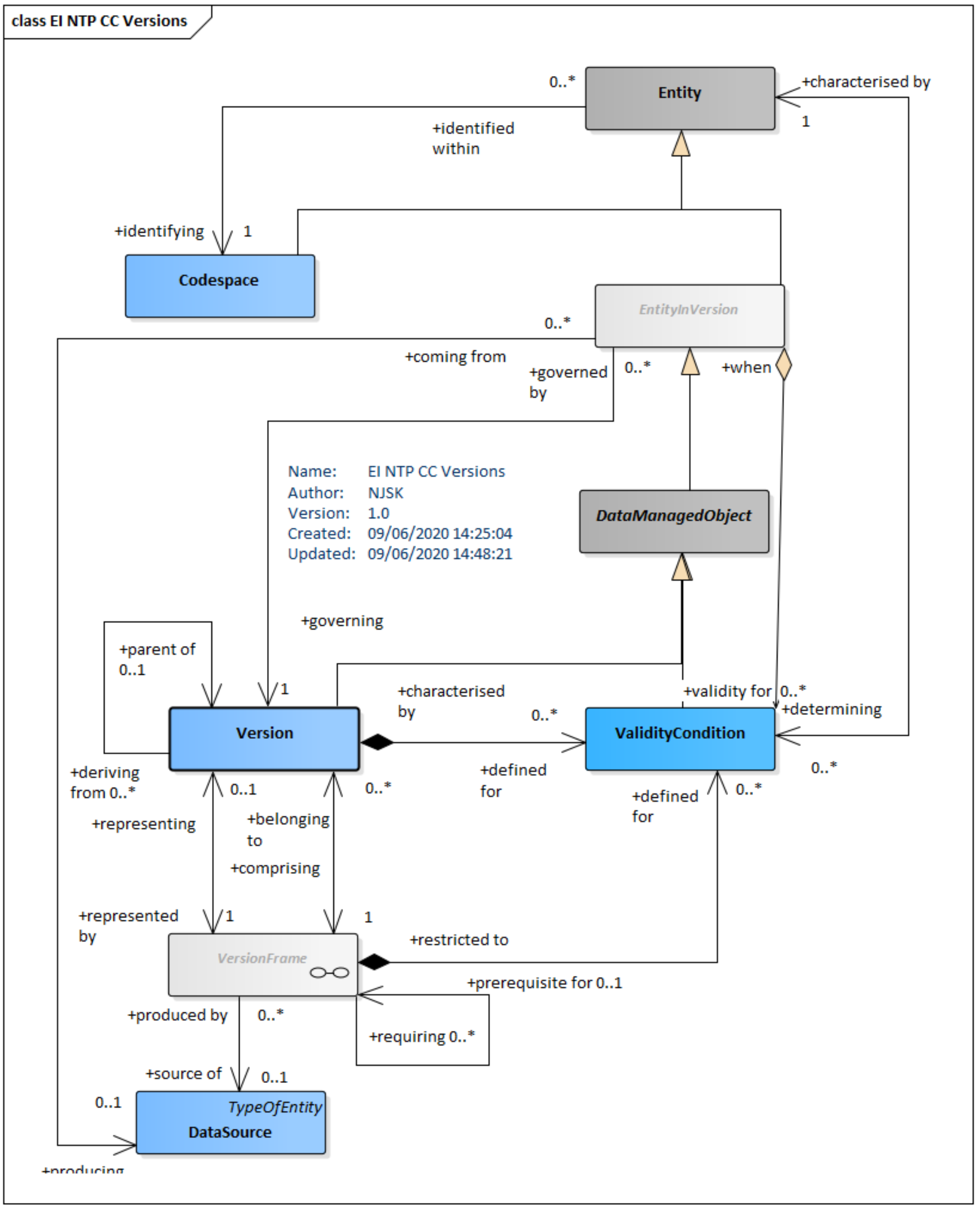


Figure 2 — Versions and Codespaces (physical model) (UML)

In Transmodel, an ENTITY represents an actual object in a repository having an instance of data present in an exchanged data set. Several versions, usually successive, of an ENTITY can be defined; normally a given data set will contain one particular version (but it is also possible to have multiple versions present in the same data set and/or repository if required).

Data exported to an XML document from a repository represents a snapshot of the state of a particular version of the data at a particular moment in time. The NeTEx XSD is therefore primarily concerned with instances of Transmodel's ENTITY IN VERSION**.** That is to say, data elements in a NeTEx XML document represent a particular version of each ENTITY. Even if an implementation of a repository holds only a representation of the current state of an ENTITY, rather than its entire history of versions, every time it makes an export it is in effect creating an ENTITY IN VERSION for the time of export; if the current version in the repository is changed, two successive exports would result in different states, i.e., different instances of the appropriate ENTITY IN VERSION.

In NeTEx, ENTITY IN VERSION is implemented as ***EntityInVersion***; this is further specialised as ***DataManagedObject*** which also reifies certain other separate Transmodel concepts into a single abstract XML class. It provides a convenient way to bring together common versioning, responsibility, and validity condition features from Transmodel, together with the XML extension mechanisms supported by NeTEx, in a way that is completely uniform for all NeTEx objects and so is simpler to implement.

NeTEx uses ***Codespaces*** to ensure that the identifiers of instances of elements in an XML document are unique, even if they come from many different sources – see chapter 11.

A ***ValidityCondition*** is used to attach conditions to elements to indicate when they are current and also for the circumstances in which they should be used (e.g., “In winter”). VALIDITY CONDITIONs can be attached to specific elements and also, through VERSION FRAMEs (see below), to whole sets of objects, so that it is possible to be precise about the conditions governing the coherence and relevance of data.

The ***DataManagedObject*** allows the assignment of a ***Version*** and responsibility attributes to an ***EntityInVersion*** and also one or more instances of a ***ValidityCondition*.**

8.1.1 Entity (Abstract in the EIRE_NP)

An ENTITY represents an actual object, instances of versions of which are present in an exchanged data set.

It specifies a unique identifier attribute that is inherited by all NeTEx elements.

Table 2 — *Entity* – XML Element

Classification	Name	Type	Cardinality	Description
«atr»	***nameOfClass***	*xsd:NMTOKEN*	0:1	Name of NeTEx XML class of which this is an instance. Used for reflection. Not used in simple applications.
«PK»	***id***	*ObjectIdType*	1:1	Identifier of ENTITY. Unique within CODESPACE. Abstract – always use the specific concrete identifier of the class that specialised ENTITY. See 11.4 for coding rules.

8.1.2 EntityInVersion (Abstract)

A specific version of an ENTITY, as found in a NeTEx document. The NeTEx ***EntityInVersion*** implements ENTITY IN VERSION by adding VERSION tracking attributes and VALIDITY CONDITIONs (see later below) for any NeTEx element.

NeTEx allows versioning to be either course-grained (a whole data set) and a fine-grained (every individual element) The EIRE_NP requires at least a minimal coarse-grained use, for which ***created*** and ***change***d attributes should be set on the VERSION FRAME.

Table 3 — *EntityInVersion* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*Entity*	::>	ENTITY ON VERSION inherits from ENTITY.
«FK»	***dataSourceRef***	*DataSourceIdType*	0:1	Data system which originated data instance.
«atr»	***created***	*xsd:dateTime*	0:1	Date and time of creation of ENTITY.
«atr»	***changed***	*xsd:dateTime*	0:1	Date and time of last change to ENTITY.
«enum»	***modification***	*ModificationEnum*	0:1	Nature of modification. Enumerated value: • *new;* • *revise* • *delete* • *unchanged* • *delta* If unspecified, assume value from parent of containing frame.
«PK»	***version***	*VersionIdType*	0:1	VERSION number of this instance of the ENTITY IN VERSION.
«enum»	***status***	*VersionStatusEnum*	0:1	Status of ENTITY in VERSION. Default is '*active*'. Enumerated value: • *active* | *inactive | other*.
«FK»	***derivedFromVersionRef***	*VersionIdType*	0:1	Reference to VERSION from which this VERSION of the ENTITY was derived. This is a *VersionIdType,* the same type as version, thus being a version number.
«FK»	***compatibleWithVersion-FrameVersionRef***	*VersionIdType*	0:1	Version of frame with which this version of ENTITY is compatible. Assumes a VERSION FRAME of the same id as current frame.
«FK»	***derivedFromObjectRef***	*ObjectIdType*	0:1	Identity of object from which this version of object was derived. Normally the same.
		CHOICE	**0**:1	Optimisation: A single VALID BETWEEN may be used without a wrapper tag.
«cntd»	***a*** ***validityConditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs conditioning entity.
	b ***ValidBetween***	*ValidBetween*	**0**:1	OPTIMISATION. Simple version of a VALIDITY CONDITION. Comprises a simple period. NB There is NO UNIQUENESS CONSTRAINT.

8.1.3 DataManagedObject (Abstract)

A ***DataManagedObject*** is a subclass of ***EntityInVersion*** that implements the common properties of all first-class NeTEx elements, such as:

— A ***ResponsibilitySet***, specifying data management and stakeholder roles for the object.

- A ***KeyList***: an optional set of an arbitrary list of key/value pairs to provide a simple extensibility mechanism. ***KeyList*** will not be used in the European profile (however it remains available for local extensions).
- An ***Extensions*** tag that allows the association of arbitrary user defined XML elements with an element. ***Extensions*** must **not** be used within the European profile.
- A reference to a ***Branding*** for the ENTITY to attribute a marketing content for presenting the element.
- ***AlternativeText*** a set of alternative variants of any textual attribute of the ENTITY (the main purpose being translations into other national language).

DataManagedObject is an implementation artefact and does not appear in Transmodel.

Table 4 — *DataManagedObject* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*EntityInVersion*	::>	DATA MANAGED OBJECT inherits from ENTITY IN VERSION.
«FK»	***responsibilitySetRef***	*ResponsibilitySetIdType*	1:1	Reference to a RESPONSIBILITY SET defining ownership and management responsibilities for objects.
«cntd»	***keyList***	*KeyValue*	0:*	A list of ***KeyValue*** pairs that may be associated with the object. This allows simple arbitrary user extensions.
«cntd»	***Extensions***	*ExtensionStructure*	0:1	User defined extension elements associated with elements.
«FK»	***BrandingRef***	*BrandingRef*	0:1	Reference to a BRANDING. (A ***TypeOfValue***) with no added attributes
«cntd»	***alternativeTexts***	*AlternativeText*	0:*	Additional Translations of text elements.

8.1.4 AlternativeText (Subcomponent of DataManagedObject)

It is sometimes necessary to provide several variants of a name or other descriptive text, especially if the information is required in several national languages. The ***AlternativeText*** is a generic way of providing such variants for any textual attribute of a ***DataManagedObject***. It can be seen as a complement to the **AlternativeName** mechanism (described later on), and can be used for any name, description or other text.

Note: The ***AlternativeName*** element ***(***rather than an ***AlternativeText)*** is expected to be used for aliases (e.g., *"King's Cross"; "London King's Cross*" "*King's Cross Railway Station*") of proper names, whereas ***AlternativeText*** is intended primarily for translations (e.g., "en.London", "fr.Londres", "it.Londra", "cn.倫敦", "ge.ლონდონი", etc). Can also be used for other attributes such as email and URLs which are for a specific language.

In the profile, ***AlternativeText*** should always be embedded inline within the parent element.

Figure 3 — AlternativeText – *XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	**AlternativeText** inherits from VERSIONED CHILD.

«PK»	***attributeName***	*xsd:NCName*	0:1	Name of text attribute for which this is the alternative text. Must be an existing attribute name
«PK»	***useForLanguage***	*xsd:language*	0:1	Name of language for which this is to be used.
«atr»	***order***	*xsd:integer*	0:1	Order of name.
«FK»	***DataManaged-ObjectRef***	*DataManagedObjectRef*	0:1	Reference to the DATA MANAGED OBJECT containing the attribute.
	Text	*MultilingualString* (Language + Text)	1:1	Variant of the text, in the specified language.

8.1.5 KeyValue (Subcomponent of DataManagedObject)

A **KeyValue** is a convenient way to exchange additional specific data attributes not supported in NeTEx. Lists of values can be associated with any **DataManagedObject**. Note that the interpretation and use of each ***KeyValue*** requires an additional specification of its meaning by the local systems that use them.

KeyList shall not be used to carry information that can be encode elsewhere in NeTEx.

Table 5 — *KeyValue – XML Element*

Classificat ion	Name	Type	Cardinality	Description
«PK»	***typeOfKey***	*xsd:normalizedString*	0:1	Type for KEY VALUE.
«PK»	***Key***	*xsd:normalizedString*	1:1	Key for KEY VALUE.
	Value	*xsd:normalizedString*	1:1	Value for KEY VALUE.

8.1.6 Version

NeTEx can support the exchange of data from systems with advanced versioning systems that track every change to every element - as recorded by a ***version*** attribute on ***DataManagedObject***. However, many legacy systems do not support such versioning capabilities and so the provision of *"fine-grained"* tracking data in a NeTEx document is optional in the EIRE_NP. However, the use of at least frame level versioning is mandatory in the EPIP and the EIRE_NP.

In Transmodel, VERSIONs are themselves first class objects and can be given descriptions that explain the purpose and validity of the VERSION. These can be exchanged in NeTEx as ***Version*** instances.

8.1.7 Codespace

A CODESPACE is similar to the XML concept of a *Namespace* that gives a context within which the names of elements and attributes in an XML schema must unique, (using the generic W3C domain mechanism), thus permitting the combining of elements from different schemas in the same aggregated model without name clashes.

In a similar way, NeTEx uses ***Codespaces*** to ensure that the identifiers of instances of elements in an XML document are unique, even if they come from many different systems and are being combined in the same document – see examples below. This allows data elements from different countries, different regions, and different organisations to be combined without clashes.

Having an explicit ***Codespace*** object also makes it possible for applications (such as a profile validator) to reflect on and compute over the elements within **Codespaces** in a generic manner (i.e., they can determine which coding system is being used for a given object and use this knowledge to inform the way they process data). The ***Codespace*** is thus used for three purposes:

(i) To give a context within which to identify objects uniquely.
(ii) To indicate to consumer systems which identifier system to use to interpret codes (if they care).
(iii) To manage the distributed allocation of new identifiers to new instances of objects.

The NeTEx schema includes XML uniqueness and referential integrity constraints so that (a) data can be validated to check each identifier is unique and (b) so that data can be validated to check any versioned reference to an entity is satisfied (i.e., that the referenced element is also present elsewhere in the document). The checks can be applied automatically using any off-the-shelf XML validator to any NeTEx document and are fundamental for establishing a basic quality level for data exchange. See section 11.16.

CODESPACEs thus prevent identifier clashes among data provisions coming from different countries (and regions, authorities and operators within countries). Different code structure systems may be used within different CODESPACES.

NB☞ In an Irish context, CODESPACEs should be used in particular to represent:

a) The scope of identifiers of locality data from the NPTG data set.
b) The scope of identifiers of stops from the NaPTAN data set.
c) The scope of identifiers of operators from the NOC data set used in timetable documents.
d) The scope of identifiers of timetable data from operators' data sets used in Timetable documents.

8.1.7.1.1 Use of codespaces to manage the distributed allocation of codes

For a large shared dataset, such as for example all the stops in a country, where the it is necessary to allocate unique codes but the responsibility for creating elements is delegated to multiple stakeholders, some sort of coordination mechanism is needed. This may be done either through a central registrar function, responsible for issuing all identifiers, or by a distributed system. Transmodel/NeTEx includes features to support a distributed approach.

In order to manage a uniform coding system for use by many stakeholders, each of whom may allocate codes, a CODESPACE may be partitioned into different TRANSPORT ADMINISTRATIVE ZONEs, based on spatial, modal or other criteria, each with a responsible ORGANISATION. Within the CODESPACE, the use of a specific prefix or prefix range value may then be reserved for the allocation of identifiers to new entities falling under the criteria of that zone. This allows the issuing of new codes to be delegated to the managers of each separate zone on a distributed basis. The central registrar function is needed only to setting up of the zones and issue the reserved ranges of prefixes in the first place.

NB☞ In an Irish Context, ADMINISTRATIVE ZONEs should be used in particular to represent:

a) The system of ADMINISTRATIVE AREAs provided by the NPTG and used to issue new NaPTAN stops.

A ***Codespace*** is specified as a path expression by using an internet domain, obtained through IANA (the Internet Assigned Numbers Authority), which provides a process for registering a domain as globally unique. For example, '*tfl.gov.uk*', '*bahn.de*', '*ratp.fr*', '*foo.com*', '*first.ei*', or '*sbb.ch'*. The path can be declared in the `<XmlnsUrl>` attribute of a ***Codespace*** element. Normally it is sufficient to use the existing domain of an organisation as the path.

Each ***Codespace*** has an associated *prefix* which may be used as a surrogate for the codespace throughout a document and is declared on a ***VersionFrame*** using a ***Codespace*** element. These prefixes provide an efficient way of encoding unique identifiers in an XML document (even if the identifiers come from multiple, heterogenous coding systems), because the short prefix can establish a unique context for each identifier on every element *id* in a document (rather than say, a long qualifying string or complex code structure). The prefix does not have to be the same as the ***id*** of the ***Codespace*** (For example, the examples in this document mostly use a prefix of '*epd'* for the '*epip_data'* ***Codespace*** and a prefix of '*epip'* for the '*epip_metadata'* ***Codespace***. Technically, the prefix is local to the document and different prefixes can be used for the same ***Codespace*** in different documents, but it is helpful to data consumers to use the same prefix values consistently across multiple documents.

Within a codespace, a data provider may use any system of coding identifiers they like; the only mandatory requirement is to provide persistent unique identifiers for each separate entity instance within the codespace.

NeTEx XML EXAMPLE CODESPACE declarations:

```
<CompositeFrame version="1.0" id="mybus:Timetable@0018">
        <codespaces>
            <Codespace id="naptStop_data">
                <Xmlns>naptStop</Xmlns>
                <XmlnsUrl>http://naptan.org.uk/stop</XmlnsUrl>
                <Description>NaPTAN for Eire/Description>
            </Codespace>
            <Codespace id="luas_cs">
                <Xmlns>luas</Xmlns>
                <XmlnsUrl>http://luas.ei/stops</XmlnsUrl>
                <Description>Dublin bus</Description>
            </Codespace>
        </codespaces>
```

Examples of elements that use of the above CODESPACEs:

```
<StopPlace id="naptStop:4701234567" version="any">
<StopPlaceRef ref="naptStop:4701234567" " version="any"/>
<Line id="luas:0018">, etc.
```

An application processing a document is expected to understand any necessary rules peculiar to validating or interpreting the identifiers in a specific ***Codespace***. For example, NaPTAN stop identifiers in the NAPTAN codespace have the structure '*999 0 XXXXXXXXX'* where '*999'* is an area prefix, '*0*' is fixed and '*XXXXXXXXX'* is a number unique within the area, whose length and nature depends on the mode of transport. An XML validator will not enforce the format rules, but an implementation that is aware of the CODESPACE and its rule system (as indicated by the domain url) is able to do so programmatically if it is relevant to do so.

Note that an application that is using rather than creating data (for example a system that is merely aggregating data to pass it onto others) can be agnostic about the internal structure of identifier. For the general purposes of emitting or parsing a NeTEx document, any such local rules are generally irrelevant; all that matters is that each entity instance can be uniquely identified.

Table 6 — *Codespace* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*Entity*	*::>*	CODESPACE inherits from ENTITY.
«PK»	***id***	*CodespaceIdType*	1:1	Identifier of CODESPACE. Unique within documents.
«AK»	***Xmlns***	xsd:*NMTOKEN*	1:1	CODESPACE prefix, unique within a given XML document e.g., *'naptStop'*
	XmlnsUrl	*xsd:anyURI*	0:1	CODESPACE path. Globally unique. For example, http://naptan.org.uk/naptan or http:/vdv.de/vdv/haltstelle/
	Description	*xsd:string*	0:1	Description of CODESPACE.

An application validator may be programmed to be aware of codespace format rules and apply further checks. For example, it could check that NaPTAN stop codes have the correct format for each mode and that the area prefixes are valid ones from the NPTG data set.

8.1.8 ValidityCondition

A ***ValidityCondition*** can be used to specify the conditional validity of a ***DataManagedObject.***

The full NeTEx specification allows complex, value dependent conditions to be constructed that may logically chain separate conditions to create a compound rule. For example, 'if the ***Season*** is "*winter*" and TIMEBAND is 24:00 to 06:00 and the TOPOGRAPHIC AREA is "*Lapland*" and the TRANSPORT MODE is "*SnowAndIce*" and the OPERATOR is '*Santa Claus'*, etc.

NB☞ The EIRE_NP does not include complex conditions, but allows the use of AVAILABILITY CONDITIONs, a sspecialisation of condition restricted to temporal parameters.

The producer should ensure that VALIDITY CONDITIONs are not contradictory. See section 11.2 later below.

Table 7 — *ValidityCondition* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	*::>*	Inherits from DATA MANAGED OBJECT. Note that use of most of the inherited values is suppressed.
«PK»	***id***	*ValidityConditionIdType*	1:1	Identifier of VALIDITY CONDITION.
	Name	*MultilingualString*	0:1	Name of VALIDITY CONDITION.
	Description	*MultilingualString*	0:1	Description of VALIDITY CONDITION. See comment on the use of *partial* for *AccessibilityLimitation*.
«FK»	***ConditionedObjectRef***	*ObjectRef*	0:1	Reference to object to which VALIDITY CONDITION applies. To be used only if (and mandatory in this case) the ValidityCondition is a standalone object (not embedded in another XML element).
«FK»	***WithConditionRef***	*ValidityConditionRef*	0:1	CONDITION with which this rule is logically ANDed.

8.1.9 ValidBetween

ValidBetween is a simplified view of a ***ValidityCondition***, having only a start date and end date. A start or end date may be omitted if unknown (thus specifying any time up to the ***ToDate***, or any time after the ***FromDate***), but at least one of them shall be provided.

A ***FromDate*** must be specified on a ***ValidBetween*** instance specifying the dominant condition on a VersionFrame.

Table 8 — *ValidBetween* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	Inherits from DATA MANAGED OBJECT. NB none of the inherited attributes are usable.
	FromDate	*xsd:dateTime*	0:1	Start date of VALIDITY CONDITION.
	ToDate	*xsd:dateTime*	0:1	End of VALIDITY CONDITION. Date is INCLUSIVE.

8.1.10 AvailabilityCondition

An AVAILABILITY CONDITION is a specific type of VALIDITY CONDITION used to specify a set of temporal conditions that can be associated with an ENTITY, for example that a STOP PLACE is open on a particular DAY TYPE.

Notes: AVAILABILITY CONDITIONs should not be used to condition VEHICLE JOURNEYs or other element for which a DAY TYPE may be specified directly. They can be used for example to specify temporal conditions on other arbitrary elements, for example, to state when a STOP PLACE or POINT OF INTEREST is open.

NB☞ Note that AVAILABILITY CONDITIONs are not included in the EPIP.

Table 9 — Table 5 — *AvailabilityCondition – Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*ValidityCondition*	::>	AVAILABILITY CONDITION inherits from VALIDITY CONDITION.
«PK»	***id***	*AvailabilityConditionIdType*	1:1	Identifier of AVAILABILITY CONDITION.
	FromDate	*xsd:dateTime*	0:1	Inclusive start date for validity of AVAILABILITY CONDITION.
	ToDate	*xsd:dateTime*	0:1	Inclusive end date for validity of AVAILABILITY CONDITION.
	IsAvailable	*xsd:boolean*	1:1	Whether the AVAILABILITY CONDITION makes the resource available or not available.
«FK»	***dayTypes***	*DayTypeRef*	0:*	DAY TYPEs during which AVAILABILITY CONDITION applies.
	validDayBits	*bitString*	0:1	Alternative way of stating the validity of days within the start and end date; 1 = *valid* and 0 = *not valid*. There must be one bit for each day.
«cntd»	***timeBands***	*TimeBand*	0:*	TIME BANDs during which AVAILABILITY CONDITION applies. N.B. Any required TIMEBANDs should be specified on the DAY TYPE.

8.1.11 DataSource

An ENTITY is linked to a DATA SOURCE, which is used to identify the system that originated the version of the ENTITY in the exchanged data set. Each ENTITY may have its own DATA SOURCE: for example, STOP PLACEs coming from the consolidated NAPTAN National registry of stops might all be declared as sourced from '*nptg:NaPTAN*', so that the XML tag for a STOP PLACE might read:

```
<StopPlace version="001" dataSourceRef="nptg:NaPTAN" id="naptStop:8220DB004722">.
```

Usually, all the data of a given type in a data set comes from the same source, so the ***DataSource*** is specified at the VERSION FRAME level using the ***DefaultDataSource*** attribute on the ***VersionFrame / FrameDefaults***.

DataSource is a specialisation of ***TypeOfValue*** but has no additional attributes of its own. See 8.5.1 below.

The ***DataSource*** is not necessarily the same as the provider of the data. For example, NaPTAN data is aggregated and issued centrally from a common ***DataSource***, but data from each area has a different ***ResponsibilitySet*** indicating it actual provider.

8.2 Relationships and entity references

In order to turn any computer-based representation of a data set (as an object model or database) into an NeTEx conformant data file that can be exchanged, it must be serialised by the producer system into an XML document – that is, a stream of characters grouped within XML tags. This requires in particular that the producer system encodes the relationships between entities (on the computer system represented by object pointers or keys in a database) in a way that can be correctly interpreted as the same relationships by any consumer system reading the XML document. This is typically done in two different ways:

(a) *Containment* – Related entities are grouped within the body of the referencing element. This is appropriate for implementing **composition** relationships where there is an inherent hierarchy, as say in the ***Codespace*** example above, or for the contents of a ***VersionFrame***.

(b) *Reference* – The relationship is implemented by a 'reference' attribute containing the identifier(s) of the related element. This is appropriate for **association** relationships which link two entities. Depending on the semantics and **navigability** of the relationship, a reference may be held on an entity at one or both ends of the relationship.

NeTEx XML EXAMPLES

For example, to serialise a ***StopPlace* → *quays*** composition relationship for two platforms of '*Castleconnell Train Station*', containment within a ***quays*** tag is used to embed the child ***Quay*** instances inline;

```
<StopPlace version="8.6" id=" naptStop:841GIR0071">
    <quays>
        <Quay version="8.4" id="naptStop:841GIR0071-1">
            <Name>Platform 1</Name>
        </Quay>
        <Quay version="8.6" id="naptStop:841GIR0071-2">
            <Name>Platform 2 </Name>
        </Quay>
    </quays>
</StopPlace>
```

For example, to serialise a many-to-one ***StopPlace*** **➔** ***TopographicPlace*** association for '*Castleconnell Train Station*' with the locality '*Castleconnell*, a versioned reference to the ***TopographicPlace*** is held on the ***StopPlace***:

```
<StopPlace version="001" id="naptStop:841GIR0071">
    <Name> Castleconnell Train Station</Name>
    <TopographicPlaceRef version="any" ref="nptgLocality:E0034695"> Castleconnell
</TopographicPlaceRef>
</StopPlace>
```

For one-to-many associations, where the referencing element holds multiple references, NeTEx always uses a wrapper tag based on the relationship name to group the references. For example, a ***StopPlace / type***s association that holds references to two different instances of ***TypeOfZone***.

```
<StopPlace version="001" id="napt:841GIR0071">
    <types>
            <TypeOfZoneRef version="any" ref="napt:Type_of_zone@GRLS"/>
            <TypeOfZoneRef version="any" ref="uicStationCat:476"/>
    </types>
</StopPlace>
```

Technical note: In fact, the schema encodes all such one-to-many-relationship as explicit specialisations of a ***OneRelationship*** complex structure, so that a software tool capable of reflection on the schema may interpret the model and apply various validation checks automatically.

Note: For flexibility, in the NeTEx schema every dependent child element includes an attribute to hold a reference to its parent. If the parent is evident from the context, as say when nesting child elements within a parent, the reference to the parent does not need to be populated.

In the EPIP and EIRE_NP the "**parent ref'** shall not be populated in such circumstances, both for brevity and to avoid the possibility of it being populated with the wrong value.

For example, the ***ParentZoneRef*** shown here is ***not*** needed on the ***Quay*** instances because it is already stated by virtue of them being embedded inline within the parent ***StopPlace***.

```
    <StopPlace version="8.6" id="naptStop:841GIR0071">
        <quays>
            <Quay version="8.4" id="naptStop:841GIR0071">
                <Name>Platform 1</Name>
                <ParentZoneRef version="8.6" ref="naptStop:841GIR0071">NOT
    NEEDED</ParentZoneRef>
            </Quay>
            <Quay version="8.6" id="napt: naptStop:841GIR0071">
                <ParentZoneRef version="8.6" ref="naptStop:841GIR0071">NOT
    NEEDED</ParentZoneRef>
                <Name>Platform 2 </Name>
            </Quay>
        </quays>
    </StopPlace>
```

8.2.1 VersionOfObjectRef

All NeTEx references are implemented not with a simple identifier value type but with a ***VersionOfObjectRef*** element which is specialised for each entity type (***LineRef, StopPlaceRef, ServiceJourneyRef***, etc.). This so as (a) to allow a version to be specified on the reference; and

(b) to enforce type correctness and facilitate the automatic validation of references using an XML validator.

Table 10 — VersionOfObjectRef – (Abstract)

Classifi-cation	Name	Type	Cardi nality	Description
	NameOfRefClass	*NameOfClass*	0:1	Reference to object type of referenced ENTITY that is in the specified VERSION FRAME
	created	*xsd:dateTime*	0:1	Date and time of creation of reference to ENTITY, i.e., when it was added to frame.
	changed	*xsd:dateTime*	0:1	Date when reference was last changed
«FK»	***ref***	*(ObjectIdType)+*	1:1	Reference to simple identifier of ENTITY that is in the specified VERSION FRAME. The *ObjectIdType* is specialised, e.g., *LineIdType*, *StopPlaceIdType*, etc
		ONLY ONE OF THE FOLLOWING		
«FK»	***version***	*VersionRef*	0:1	Reference to VERSION of ENTITY that is in a VERSION FRAME of the current dataset. Mutually exclusive with **versionRef**. Use "any" if the VERSION is unknown (note that this will trigger an XML automatic consistency check). The CODESPACE of the ***version*** and ***versionRef*** is assumed to be the same as that of the ***ref.***
«FK»	***versionRef***	*VersionIdType*	0:1	Identifier of the VERSION of the referenced entity. For use when reference is external (i.e., in another dataset) and a specific version is required. Mutually exclusive with **version.** Use "any" if the version is unknown. This will also make clear that the referenced object is external.

8.2.2 Integrity Checking of XML Documents

Missing or incorrect element references are a serious cause of error to a consuming system. NeTEx uses a powerful and very uniform approach to serialising relationships that allows the automated enforcement of referential integrity checks to ensure that references are complete and correct. References may be to a precise version of an object. The checking can be selectively relaxed on references to objects located in a separate document; thus, large data sets, (and also data sets that come from different producers), can be broken down into separate documents.

The distinction between internal (same document) and external object (separate document) references is made by choosing to use either the **version** attribute (*internal*) or **versionRef** attribute (*external*) on a given reference.

- References using the **version** attribute will be checked automatically by an XML validator working from the many constraints that are coded within the schema – i.e., are internal.
- This automatic consistency check is not done on references that use **versionRef**, which are treated as *external (*even if the referenced object is in fact in the same document).

For example, the following encoding of a ***StopPlace*** ➔ ***TopographicPlace*** reference will be checked to ensure that elsewhere in the document there is a definition of the specified ***TopographicPlace*** with the identifier *'nptg:E0034695'.*

```
    <TopographicPlaceRef version="2.2" ref="nptg:E0034695"/>
```

While, in this case the checking has been relaxed and the specified ***TopographicPlace*** does not necessarily have to be present in the document.

```
    <TopographicPlaceRef versionRef="2.2" ref="nptg:E0034695”/>
```

For readability and to assist debugging, by convention it is helpful to include an annotation on external references with the tag content as to the nature of the reference (since this may not be obvious from the ***ref*** identifier itself), for example, the text ‘*Wimbledon’* annotates the reference to *‘nptg:E0034695’.*

```
    <TopographicPlaceRef versionRef="2.2"
ref="nptg:E0034695">Wimbledon</TopographicPlaceRef>
```

For further details on the use of references in the profile see section 11.4.

8.2.3 Identity uniqueness scope

All the specialisations of ***DataManagedObject*** have an ***id***. Within a given document, the value of this must be unique within the class of element. For example, there can be only one ***SiteFrame*** with an ***id*** of “*foo*” in a document. In most cases, different classes of element may have the same *id*; thus, for example it is possible to have both a “*foo” **SiteFrame*** and a “*foo” **ServiceFrame*** in the same document.

The uniqueness and integrity constraints for NeTEx elements are implemented by XML language features as part of the schema, and so can be automatically enforced by any XML Validator. Note, however, that due to certain technical restrictions on the way the XML key constraints work and are currently implemented, certain classes of related NeTEx elements are currently subject to a wider restriction of uniqueness (that is, uniqueness is shared across a common superclass). For example, PLACEs of any sort must be unique, so it is not possible to have a ***PointOfInterest*** with ***id***=’foo’ and a ***StopPlace*** with ***id***=’foo’ in the same NeTEx conformant document.

Note that if an identifier coding system includes the name of a specific NeTEx class in the code structure (as does the EPIP common code structure used with the ‘*epip_data’ **Codespace***, – see 11.4) then all codes are inherently unique within their specific type. For example, the ***PointOfInterest*** with ***id***=’*epip:EI_NTA_PointOfInterest:foo*’ is distinct from a ***StopPlace*** with ***id***=’*epip:EI_NTA_stopPlace:foo*’.

Table 11 — NeTEx classes used in the profile with a shared uniqueness scope

Supertype	**NeTEx Concrete classes**	Comment
POINT	***ScheduledStopPoint, TimingPoint***	Useful to have a common scope because a ***ScheduledStopPoint*** is a specialisation of ***TimingPoint.***
LINK	***ServiceLink, RouteLink, PathLink,***	Useful to have a common scope so that Route, Service pattern and navigation paths are not confused.
ZONE & PLACE	***Topographic Place, Address*** ***Tariff Zone*** ***StopPlace, Quay, StopPlaceEntrance,*** ***PointOfInterest, PointOfInterestEntrance***	Unnecessarily over restrictive. For example, in some cases it would be useful to have a ***Quay*** or ***StopPlaceEntrance*** with the same id as their parent ***StopPlace***, or an ***Address*** with the same id as a ***StopPlace***.

LINK SEQUENCE	***ServicePattern, JourneyPattern***	Unnecessarily over restrictive; in some cases, it would be useful for a **ServicePattern** to have the same id as a corresponding ***JourneyPattern.***
LINE	***Line, FlexibleLine***	Useful to have common scope because a ***Line*** is a specialisation of ***FlexibleLine.***
JOURNEY	***ServiceJourney, TemplateServiceJourney***	Useful to have common scope as ***ServiceJourney*** and ***TemplateServiceJourney*** are always distinct.
ORGANISATION	***Operator, Authority***	Useful to have common scope as an ***Operator*** is distinct from an ***Authority***
FACILITY SET	***SiteFacilitySet, ServiceFacilitySet***	Useful to have common scope as they are distinct.

The identifiers of some child elements are further qualified by the use of an ***order*** attribute; they can thus share the same ***id*** value as their parent. Thus, for example ***ServicePattern*** *'phb:PHB@1@Outbound@JP1'* may have children that are uniquely identified by their order number, as follows:

```
<ServicePattern id=“phb:PHB@1@Outbound@JP1” version=”1.1”>
    <StopPointInSequence id=“ phb:PHB@1@Outbound@JP1” version=”1.1” order=”1”>
        <ScheduledStopPointRef ref=“ naptStop:3420678954” version=”any”/>
    </StopPointInSequence>
    <StopPointInSequence id=“ phb:PHB@1@Outbound@JP1” version=”1.1” order=”2”>
        <ScheduledStopPointRef ref=“ naptStop:3420678972” version=”any”/>
    </StopPointInSequence>
    <StopPointInSequence id=“ phb:PHB@1@Outbound@JP1” version=”1.1” order=”3”>
        <ScheduledStopPointRef ref=“ naptStop:3420678851” version=”any”/>
    </StopPointInSequence>
Etc., etc.

</ServicePattern>
```

Within an XML document, elements with at an ***order*** attribute must be serialised in sequence in accordance with the ascending value of their order attribute.

Note that the uniqueness and existence constraints are declared in the *NeTEx_publication.xsd* schema file and have the effect that all declarations of NeTEx elements must include an ***id*** and a ***version*** attribute in the start tag, even if it is just a dummy value (by convention '*any*')).

8.3 Responsibilities & Organisations

When data is being integrated from many sources it is important to be able to track the responsibilities for the management and licensing of data. This gives a means of contacting the relevant stakeholders for corrections and to obtain permissions.

NeTEx has a generalised responsibility model that can be used to describe the responsibilities for different RESPONSIBILITY ROLEs for any ENTITY. RESPONSIBILITY ROLEs are grouped as a RESPONSIBILITY SET using a RESPONSIBILITY ROLE ASSIGNMENT which specifies the responsible ORGANISATION. An ADMINISTRATIVE ZONE can be specified in addition to indicate the spatial extent of the ORGANISATION's jurisdiction.

NB☞ The responsibility model is a generalisation of the NPTG model which divides the country into a number of ***NptgAdministrativeArea*** instances in order to enable the distributed management of NaPTAN data. See mapping of NPTG data later.

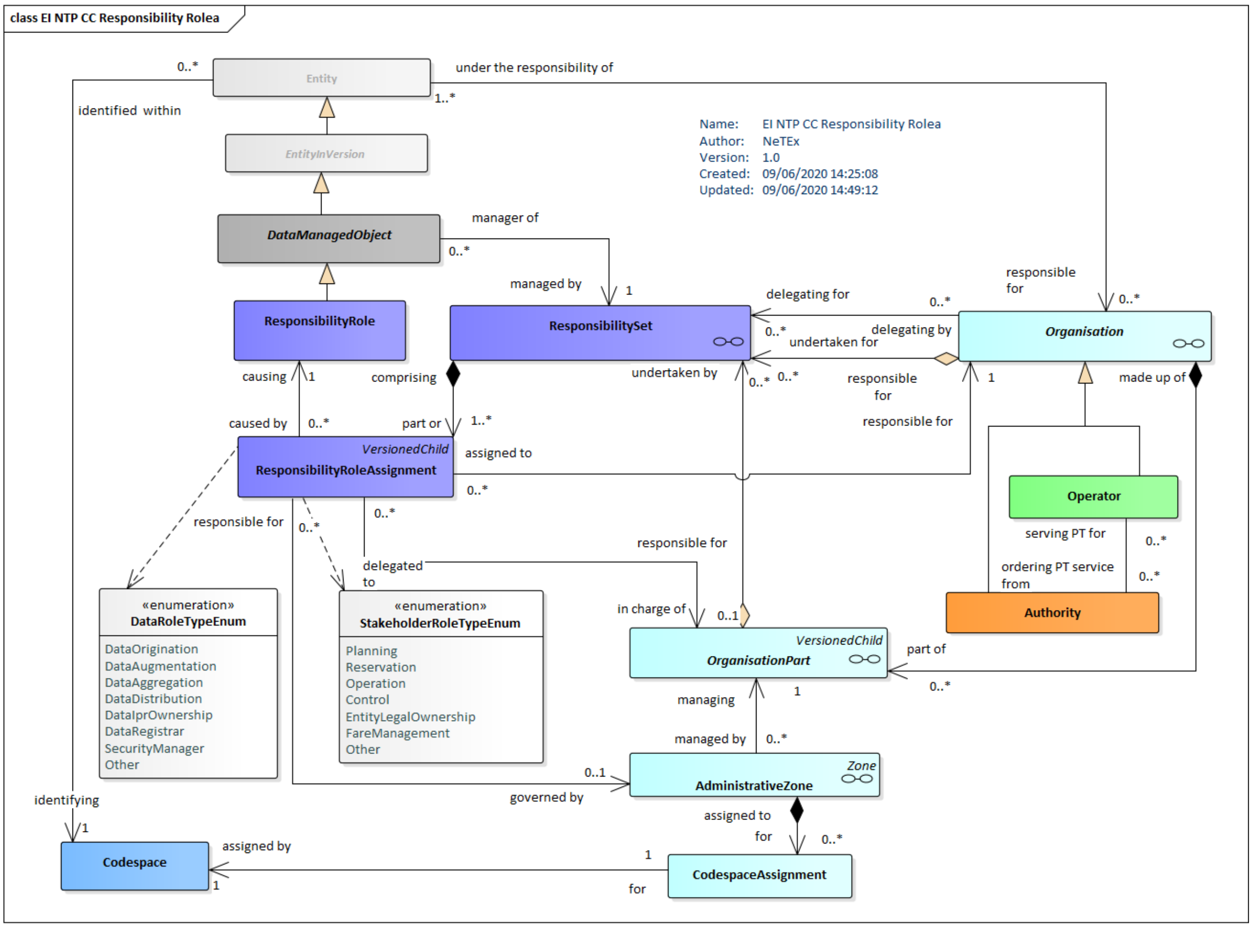


Figure 4 — ResponsibilitySet, Codespace, Administrative Zone – Physical Model (UML)

8.3.1 ResponsibilitySet

A ***ResponsibilitySet*** can be specified for a ***DataManagedObject*** to specify which organisations are responsible for different data management roles, such as creating, maintaining, distributing or licencing the data.

A ***ResponsibilitySet*** *is* normally used at a VERSION FRAME level to apply to all the elements in the frame, rather than on individual objects (though the latter is possible).

The subcomponents of a composite entity must belong to the same ***ResponsibilitySet*** as their parent. Thus, for example, all the instances of ***StopPointInJourneyPattern*** must belong to the same ***ResponsibilitySet*** as their parent.

NB☞ In the Irish NPTG and NaPTAN data, a ***ResponsibilitySet*** is used to indicate the NPTG administrative Area

NB☞ In the Timetable and Fare data provided by operators and other stakeholders, a ***ResponsibilitySet*** should be declared that states they are the provider of the data.

Table 12 — ***ResponsibilitySet*** – XML Element

Classific ation	Name	Type	Cardina lity	Description
::>	::>	*DataManagedObject*	::>	RESPONSIBILITY SET inherits from DATA MANAGED OBJECT.

«PK»	***id***	*ResponsibilitySetIdType*	1:1	Identifier of RESPONSIBILITY SET.
	Name	*MultilingualString*	0:1	Description of RESPONSIBILITY SET.
«AK»	***PrivateCode***	*PrivateCode*	0:1	A private code that uniquely identifies the element. May be used for inter-operating with other (legacy) systems.
«cntd»	***roles***	*ResponsibilityRoleAssignment*	1:*	RESPONSIBILITY ROLE ASSIGNMENTs making up the RESPONSIBILITY SET.

8.3.2 ResponsibilityRoleAssignment (Subcomponent of ResponsibilitySet)

A ***ResponsibilitySet*** is made up of one or more ***ResponsibilityRoleAssignment*** instances. Each RESPONSIBILITY ROLE ASSIGNMENT assigns of one or more roles to an ORGANISATION or an ORGANISATION PART as regards the responsibility it will have for specific data (e.g., ownership, planning, etc.) or the management of this data (e.g., distribution, updates, etc.).

Table 13 — *ResponsibilityRoleAssignment* – XML Element

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*VersionedChild*		RESPONSIBILITY ROLE ASSIGNMENT inherits from VERSIONED CHILD.
«PK»	***id***	*ResponsibilityRole-AssignmentIdType*	1:1	Identifier of RESPONSIBILITY ROLE ASSIGNMENT.
«FK»	***ResponsibilitySetRef***	*ResponsibilitySetRef*	0:1	Reference to a RESPONSIBILITY to which this RESPONSIBILITY ROLE belongs.
	Description	*MultilingualString*	0:1	Description of RESPONSIBILITY ROLE.
«enum»	***DataRoleType***	*DataRoleTypeEnum*	0:1	Data Role type of RESPONSIBILITY ROLE. • *creates* • *collects* • *validates* • *aggregates* • *distributes* • *redistributes* • *secures* • *owns* Note that only a subset of data related roles is selected for the EPIP, the selected values distinguish between an ORGANISATION originating the data '*creates*'; one collecting them from multiple sources but without changing them '*collects*' and one which integrates data sets (merging, correcting, enhancing, etc.) '*aggregates*'). NB☞ The EIRE_NP allows any role to be specified.
«enum»	***StakeholderRoleType***	*StakeholderRoleTypeEnum*	0:1	Stakeholder role type of RESPONSIBILITY ROLE. • *Planning* • *Operation* • *Control* • *Reservation* • *EntityLegalOwnership* • *FareManagement* • *DataRegistrar*

				- *SecurityManagement* - *other* Note that only a subset of data roles is selected for the EPIP, these distinguish between an ORGANISATION that does '*Planning'* (typically an Authority); one that operate '*Operation'* some aspect of the network (Operators, site managers, etc.); and one having the legal ownership (city, authority, etc.). This information is of relevance for passengers, especially if they have questions or complaints to address. EIRE☞ The FXCP also allows 'Registrar' roles (as for NPTG admin areas) and 'FareManagement'.
«FK»	***TypeOfResponsibility-RoleRef***	*TypeOfResponsibilityRoleRef*	0:1	Reference to a TYPE OF RESPONSIBILITY ROLE.
«FK»	***ResponsibilityRoleRef***	*ResponsibilityRoleRef*	0:1	Reference to a RESPONSIBILITY ROLE.
«FK»	***Responsible-OrganisationRef***	*OrganisationRef*	0:1	Reference to an ORGANISATION to which this RESPONSIBILITY ROLE is assigned.
«FK»	***ResponsiblePartRef***	*OrganisationPartRef*	0:1	Reference to an ORGANISATION PART to which this RESPONSIBILITY ROLE is assigned.
«FK»	***ResponsibleAreaRef***	*AdministrativeZoneRef*	0:1	Reference to an ADMINISTRATIVE ZONE to which this RESPONSIBILITY ROLE is assigned.

8.3.3 Organisation (Abstract)

The ORGANISATION entity represents an enterprise that is involved in planning, collecting or providing PT information. For example, a company providing a public transport information service, an authority, an operator, or a company providing an information collection service.

It is relevant to the profile as it is the supertype of OPERATOR and AUTHORITY.

Table 14 — *Organisation – XML Element (abstract)*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	ORGANISATION inherits from DATA MANAGED OBJECT.
«PK»	***id***	*OrganisationIdType*	1:1	Identifier of ORGANISATION. (abstract)
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Alternative public identifier of ORGANISATION.
«AK»	***PrivateCode***	*PrivateCode*	0:1	Alternative identifier of ORGANISATION.
«AK»	***CompanyNumber***	*xsd:normalizedString*	0:1	Registration number of ORGANISATION as a legally incorporated body.
«AK»	***External-OrganisationRef***	*ExternalObjectRef*	0:1	An alternative code that uniquely identifies the ORGANISATION. Specifically, for use in AVMS systems and for VDV compatibility.
	Name	*xsd:normalizedString*	0:1	Name of ORGANISATION.
	ShortName	*MultilingualString*	0:1	Short name of ORGANISATION.
	LegalName	*MultilingualString*	0:1	Legal name of ORGANISATION.
	TradingName	*MultilingualString*	0:1	Trading name of ORGANISATION. NB EPIP advocates use of Brand name (instead of TradingName). EIRE_NP Use Trading name. se a ***Branding*** element for the Brand Name.

«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for ORGANISATION. Use for aliases (whereas *AlternativeText* is for providing translations).
	Description	*MultilingualString*	0:1	Further description of ORGANISATION.
	Remarks	*MultilingualString*	0:1	Further remarks about ORGANISATION,
«cntd»	***Locale***	*Locale*	0:1	Locale of ORGANISATION specifying time zone, language, etc.
«cntd»	***ContactDetails***	*ContactDetails*	0:1	Contact details for ORGANISATION for public use.
«cntd»	***PrivateContactDetails***	*ContactDetails*	0:1	Contact details for ORGANISATION for privileged use, for example for immediate access to control centres.
«FK»	***OrganisationType***	*TypeOfOrganisationEnum*	0:1 1:1	Type of ORGANISATION. • *authority, operator* • *railOperator, railFreightOperator, statutoryBody, facilityOperator, travelAgent, servicedOrganisation, retailConsortium, other*
«cntd»	***typesOfOrganisation***	*TypeOfOrganisationRef*	0:*	References to a TYPE of ORGANISATION that classifies it.
	Status	*xsd:boolean*	0:1	Whether the organisation is active. Default is '*true*'.
«cntd»	***ValidityPeriod***	*ValidityCondition+*	0:1	VALIDITY Condition for organisation (deprecated: ***ValidityCondition*** is now inherited from EntityInVersion)
«cntd»	***parts***	*OrganisationPart*	0:*	Any component parts of an ORGANISATION, that is OPERATIONAL UNITs or DEPARTMENTs.
«cntd»	***ownResponsibilitySets***	*ResponsibilitySet | ResponsibilitySetRef*	0:*	Own RESPONSIBILITY SETs.
«cntd»	***delegatedResponsibilitySet***	*ResponsibilitySet | ResponsibilitySetRef*	0:1	Other ORGANISATION to which this one delegates one or more roles implementation.
«FK»	***delegatedFrom***	*(OrganisationRef)+*	0:1	Other Organisation that delegates to this Organisation. (TAP TSI B1.)

8.3.4 Operator

ORGANISATION is specialised to define the concerns that run Public Transport, specifically OPERATORs of Transport and AUTHORITies. Collectively these are called TRANSPORT ORGANISATIONs and are generally interchangeable in the schema, that is an AUTHORITY can be used in most places where an OPERATOR is indicated.

NOTE: In the original Transmodel definition, the generic term OPERATOR expressed a general responsibility for a CONCESSIONARY CONTRACT for public transport, including the operational responsibility for the execution of the contract.

Table 15 — *Operator – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Organisation*	::>	OPERATOR inherits from ORGANISATION.
«PK»	***id***	*OperatorIdType*	1:1	Identifier of OPERATOR.
«FK»	***CountryRef***	*CountryRef*	0:1	Reference to a country ISO 3166-
«cntd»	***Address***	*PostalAddress*	0:1	Postal ADDRESS of ORGANISATION.

«enum»	***PrimaryMode***	*VehicleModeEnum*	0:1	Primary TRANSPORT MODE of ORGANISATION. See STOP PLACE for allowed values.
«enum»	***OperatorActivities***	*OperatorActivitiesEnum*	0:*	Activities undertaken by the operator.
«cntd»	***CustomerService-ContactDetails***	*ContactDetails*	0:1	Contact details for ORGANISATION's Customer services.
«cntd»	***departments***	*Department*	0:*	DEPARTMENTs for Organisation.

8.3.5 Authority

An AUTHORITY is an ORGANISATION under which the responsibility of organising the transport service in a certain area is placed.

Note that if a LINE requires several AUTHORITies, this can be achieved through the use of ***additionalOperators;*** references to both OPERATORs and AUTHORITies are permitted.

Table 16 — *Authority – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Organisation*	::>	AUTHORITY inherits from ORGANISATION.
«PK»	***id***	*AuthorityIdType*	1:1	Identifier of AUTHORITY.
«cntd»	***Address***	*PostalAddress*	1:1	Postal ADDRESS of ORGANISATION.
«cntd»	***authorityTypes***	*TypeOfOrganisationRef*	0:*	Classification of AUTHORITY.

8.4 Frames

A **VersionFrame** provides a versionable container that allows a coherent set of related elements to be exchanged as a set, for example the timetable for a line, or the railway stations for a country.

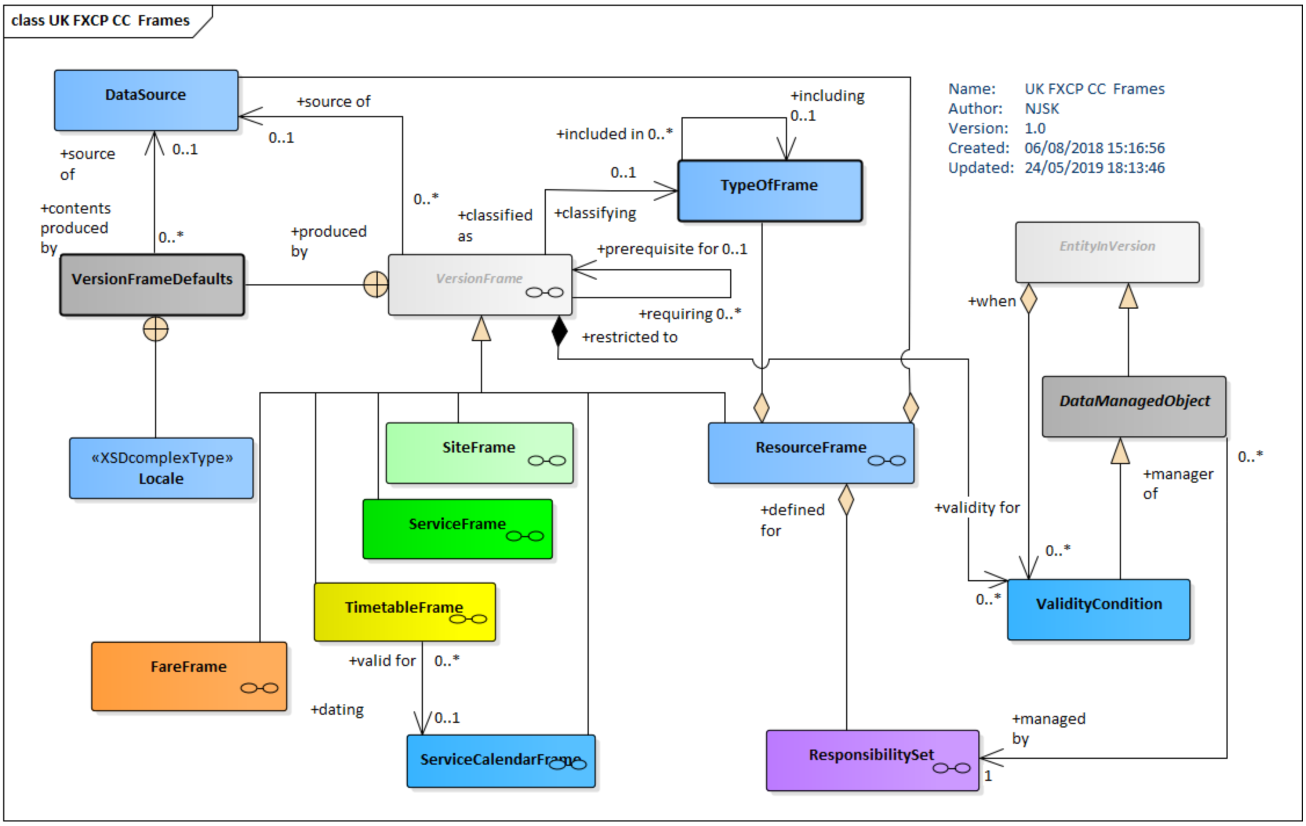


Figure 5 — Version Frames (UML)

Since they embody a coherent set of related elements (that is a complete set of elements, all of compatible version), frames are strongly involved in version management. Each VERSION FRAME instance also has one or more **ValidityConditions** attached to it (since it is a ***DataManagedObject*** in its own right and has a ***validityConditions*** relationship) that is used to express the validity of the frame and its contents as a whole See ***11.1-Version management*** for more details.

Note these frame validity conditions should not be confused with **contentValidityConditions** which are more specific validity conditions that apply to multiple elements within the frame.

The profile requires that VALIDITY conditions that apply to individual elements are embedded within the specific element to which they relate.

VersionFrame itself is abstract: NeTEx provides a set of specialised 'specific' concrete frames for specific purposes (***ServiceFrame***, ***SiteFrame***, etc.) each covering a functional subset of the NeTEx data elements; as well as a general purpose ***GeneralFrame*** that can contain any type of ENTITY IN VERSION.

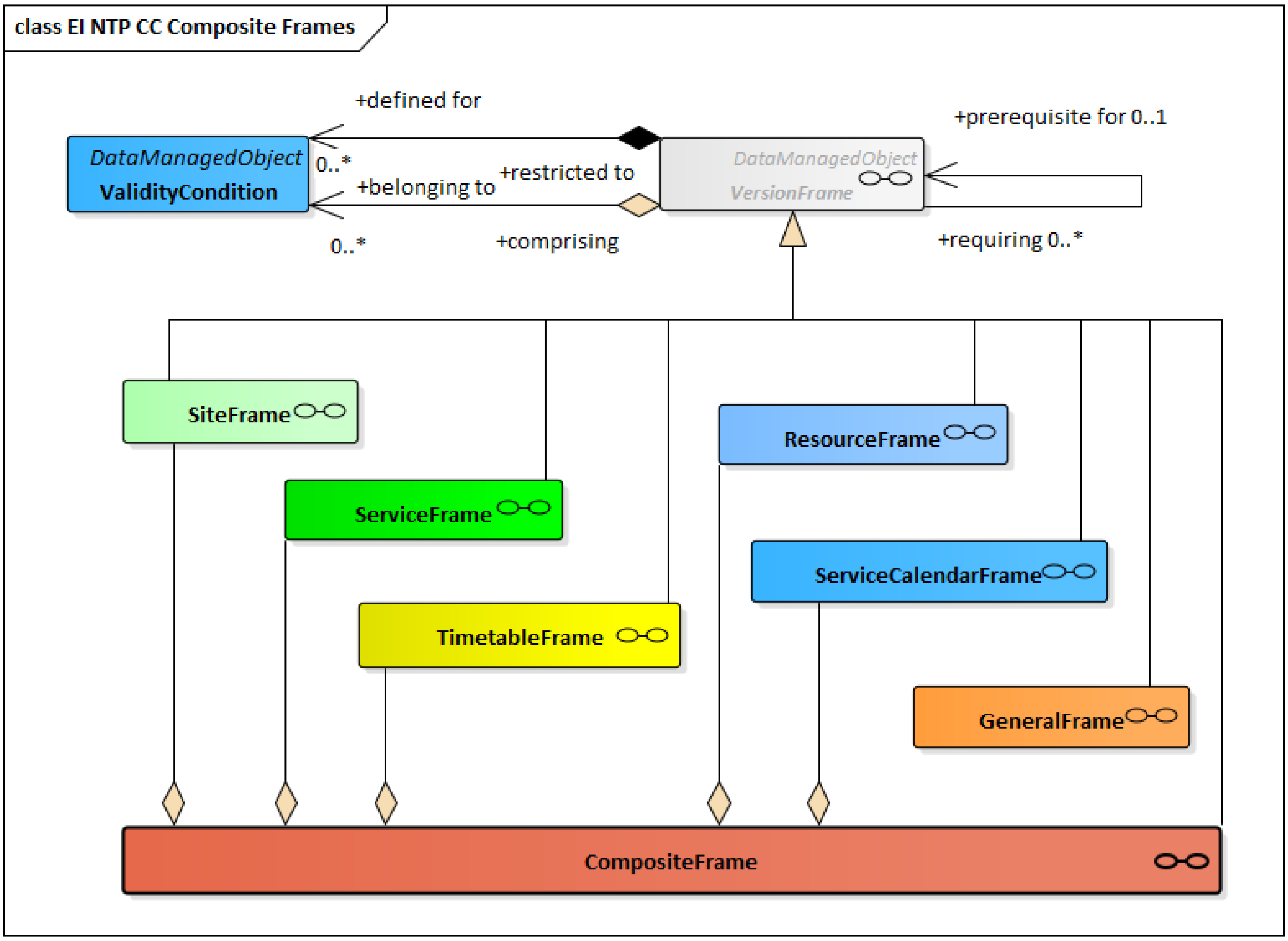


Figure 6 — Specific Version Frames (UML)

A ***VersionFrame*** can be assigned a ***TypeOfFrame***; an arbitrary user defined classification (whose use is reserved within the profile to a predefined set of values).

8.4.1 Version Frame (Abstract)

Regardless of their specific contents, all types of ***VersionFrame*** have a number of common properties, such as a name, codespaces, etc.

Table 17 — *VersionFrame* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	VERSION FRAME inherits from DATA MANAGED OBJECT. Note the “Dominant” VALIDITY CONDITION for the FRAME content is attached to the FRAME
«PK»	***id***	*VersionFrameIdType*	1:1	Identifier of VERSION FRAME.
	Name	*MultilingualString*	0:1	Name of VERSION FRAME.
	Description	*MultilingualString*	0:1	Description of VERSION FRAME.
«FK»	***TypeOfFrameRef***	*TypeOfFrameRef*	0:1	Reference to TYPE of VERSION FRAME.
«FK»	***BaselineVersion-FrameRef***	*VersionRef*	0:1	Previous baseline version frame that objects in this frame require.

«cntd»	***codespaces***	*Codespace*	0:*	CODESPACEs used in this frame. Normally there will be at least one. A default may be specified in the Frame defaults.
«cntd»	***FrameDefaults***	*FrameDefaults*	0:1	Set of default values to assume for values in the frame if not explicitly stated on individual elements.
«cntd»	***versions***	*Version*	0:*	Formal definitions of VERSIONS included in the FRAME.
«cntd»	***prerequisites***	*VersionFrameRef*	0:*	References to other version frames on which this frame depends. +v1.1
«cntd»	***traces***	*Trace*	0:*	TRACEs recording changes to ENTITIES in FRAME.
«cntd»	***contentValidity-Conditions***	*ValidityCondition*	0:*	Shared VALIDITY CONDITIONs used by elements in the FRAME. Note: this is not the dominant FRAME validity condition.
«cntd»	***layers***	*LayerRef*	0:*	Layers included in VERSION FRAMEs. +v1.1

8.4.2 TypeOfFrame

A TYPE OF FRAME classifies a VERSION FRAME. It provides a rudimentary form of metadata to describe the permitted contents of a profile (A profile validator may use such data to further check the contents of a document).

Both the EPIP and the EIRE_NP define a number of standardised ***TypeOfFrame*** instances; each one classifies a particular data set to be exchanged under the profile and must be assigned to the appropriate version frame in any profile conformant NeTEx document.

NB☞ The list of frame types in EIRE_NP is:

- ***'EI_PI_COMMON'***: common and generic object types possibly used across multiple other frames (for example: ***TypeOfValue***, ***DayType***, etc.)
- ***'EI_PI_STOP'***: all objects necessary for STOP PLACE and SITE descriptions (STOP PLACEs, QUAYs, ENTRANCEs, POINTs OF INTEREST, etc.).
- ***'EI_PI_NETWORK'***: all objects necessary for NETWORK descriptions (LINEs, ROUTEs, JOURNEY PATTERNs, etc.).
- ***'EI_PI_TIMETABLE'***: all objects related to timing information (VEHICLE JOURNEYs, PASSING TIMEs, etc.)
- ***'EI_PI_CALENDAR'***: all objects related to calendars and types of day.

Further Irish Profile frames for Fares are defined in Part3 EIRE_NP-FM.

The *'EI_PI_STOP', 'EI_PI_NETWORK'*, and *'EI_PI_TIMETABLE'* frame types carry the main "payload" data content for an EIRE_NP exchange for stops or timetables. The *'EI_PI_COMMON'*, and *'EI_PI_CALENDAR'* frames types carry ancillary data referenced by the payload frames.

A **CompositeFrame** is a specific type of frame that contains a set of VERSION FRAMEs to which the same VALIDITY CONDITIONs have been assigned. It is in effect a grouping mechanism or ‘container’. The EIRE_NP specification defines three composite frames with which to group the payload and ancillary frames:

- ‘***EI_PI_LINE_OFFER.’*** gathering all the information related to a single LINE; it shall contain ‘*EI_PI_COMMON’ ‘EI_PI_STOP’, ‘EI_PI_NETWORK’, ‘EI_PI_CALENDAR’* and ‘*EI_PI_TIMETABLE’*. specific frames.
- ***‘EI_PI_NETWORK_OFFER’***: gathering all the information related to a single NETWORK (so multiple lines); it shall contain *‘EI_PI_COMMON,’ ‘EI_PI_CALENDAR’, ‘EI_PI_STOP’*, ‘*EI_PI_NETWORK’* and ‘*EI_PI_TIMETABLE’* specific frames There may be multiple instances of a given frame type, for example a separate, , *‘EI_PI_NETWORK’, ‘EI_PI_TIMETABLE’* for each line.
- ‘***EI_PI_STOP_OFFER.’*** gathering all the information related to a STOPs for area; it shall contain ‘*EI_PI_COMMON’ ‘EI_PI_STOP,* specific frames.

A detailed description of these TYPEs OF FRAME is provided in 11.9.

One further ***TypeOfFrame*** *‘EI_PI_METADATA’* can be included in an EIRE_NP document to provide a set of static metadata references that can be used variously by both an XML validator and a parameterised profile validator. If present, internal references can be used for the ***TypeOfFrameRef*** instances, etc; otherwise, these must be external references that place the version number in a ***versionRef*** element.

Table 18 — *TypeOfFrame* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF FRAME inherits from TYPE OF VALUE.
«PK»	***id***	*TypeOfFrameIdType*	1:1	Identifier of TYPE OF FRAME. NB☞ EIRE_NP Value
«FK»	***TypeOfValidityRef***	*TypeOfValidityRef*	0:1	Reference to a TYPE OF VALIDITY.
«FK»	***FrameClassRef***	*NameOfClass*	0:1	Refence to a class of VERSION FRAME, that is the class of VersionFrame to be used to hold elements.
«cntd»	***classes***	*ClassInContextRef*	0:*	Classes to be present in frames conforming to the TYPE OF FRAME.
«cntd»	***includes***	*TypeOfFrameRef*	0:*	TYPEs OF FRAME contained in TYPE OF FRAME. Must not be cyclic.

NB☞ There is a one-for-one correspondence between EIRE_NP and EPIP frame types and an extended EPIP data set can be created simply by substituting ‘EU_PI’ for ‘EI_PI’: in the frame names.

8.4.3 Specific frames and General Frames

VersionFrame is specialised to make ‘Specific’ frames (***SiteFrame, ServiceFrame, etc.)*** for different functionally related data sets. There is also a ***GeneralFrame*** which can contain any type of object. They are very similar in concept, except that each specific frame only hosts a limited set of object types, and furthermore groups them by separate type (all the LINEs together, all the STOP PLACEs together, etc.) whilst a ***GeneralFrame*** places all objects in a single heterogenous list regardless of type.

NB☞ A specific ***VersionFrame*** is similar in concept to the outermost element in the NPTG, and NaPTAN schemas, each of which holds data elements for a specific purpose

NB☞ The EIRE_NP uses a number of different specific frame types: (***ResourceFrame, SiteFrame, ServiceFrame, TimetableFrame*** and ***ServiceCalendarFrame*** in this document, and ***FareFrame*** in Part3**.**

NB☞ Each frame must be assigned a ***TypeOfFrame*** (one of ‘*EI_PI_COMMON*’ ‘*EI_PI_STOP*’, ‘*EI_PI_NETWORK*’, ‘*EI_PI_CALENDAR*’, and ‘*EI_PI_TIMETABLE*’ - see 11.9.)

Note that the EPIP uses a very similar set of ***TypeOfFrame*** types to the EIRE_NP.

Table 19 — *GeneralFrame – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionFrame*	::>	GENERAL FRAME inherits from VERSION FRAME
«PK»	***id***	*GeneralFrameIdType*	1:1	Identifier of GENERAL FRAME.
«cntd»	***members***	*(EntityInFrame)*	0:*	Members of GENERAL FRAME: a DATA MANAGED OBJECT, an ENTITY_ENTITY or reference to a DATA MANAGED OBJECT (GeneralFrame¬Member)

8.4.4 Composite Frame

A **CompositeFrame** is a specific type of frame that contains a set of VERSION FRAMEs to which the same VALIDITY CONDITIONs have been assigned. It is in effect a grouping mechanism that allows different types of specific frame to be combined as a single data set.

NB☞ The EIRE_NP specification defines three TYPEs OF FRAME for **CompositeFrame**: *‘EI_PI_LINE_OFFER’ and ‘EI_PI_NETWORK_OFFER, and EI_PI_STOP_OFFER’*; see the summary descriptions of their permitted contents in 11.9.

Table 20 — *CompositeFrame – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionFrame*		COMPOSITE FRAME inherits from VERSION FRAME
«PK»	***id***	*CompositeFrameIdType*	1:1	Identifier of COMPOSITE FRAME.
«cntd»	***frames***	*VersionFrame*	1:*	Frames contained in COMPOSITE FRAME. Shall contain only specific frames of type *‘EI_PI_COMMON’, ‘EI_PI_STOP,’ ‘EI_PI_CALENDAR’, ‘EI_PI_NETWORK’ and ‘EI_PI_TIMETABLE ‘*.

8.4.5 FrameDefaults

The **FrameDefaults** specify default values for certain common properties of elements in a frame, such as the ***Codespace***, ***DataSource***, time-zone, etc. The defaults are applied to all elements in the frame for which an explicit value has not been specified. The use of defaults can both simplify export and reduce the size of documents.

Table 21 — *FrameDefaults* – XML Element

Classification	Name	Type	Cardinality	Description
«FK»	***DefaultCodespaceRef***	*CodespaceRef*	0:1	Default CODESPACE to assume for an identifier that does not have an explicit CODESPACE specified. The codespace must be defined elsewhere in the document.
«FK»	***DefaultDataSourceRef***	*DataSourceRef*	0:1	DATA SOURCE to use for elements in the frame which do not have a DATA SOURCE specified.

«FK»	***DefaultResponsibilitySetRef***	*ResponsibilitySetRef*	0:1	RESPONSIBILITY SET to use for elements in the frame which do not have a RESPONSIBILITY SET specified.
«cntd»	***DefaultLocale***	*Locale*	0:1	Default LOCALE to use to provide attribute values for elements in the frame which do not have a LOCALE element specified, for example language time zone, etc.
	DefaultLocationSystem	*xsd:normalizedString*	0:1	Default LOCATION SYSTEM to use for locations in the frame which do not have a LOCATION SYSTEM specified. For the EIRE_NP, WGS84 is recommended. NB☞ Irish Ordnance Service Grid References (OSGR) are also permitted.
«enum»	***DefaultSystemOfUnits***	*SystemOfUnitsEnum*	0:1	Default SYSTEM OF UNITs to use for dimensional attributes in the frame (*DistanceType, LengthType, WeightType,* etc) for which no units have been specified. • *SiMetres* • *SiMetresAndKilometres* • *Other*
	DefaultCurrency	*CurrencyType*	0:1	Default CURRENCY to use for amount attributes in the frame for which a currency has not been specified.

8.5 Generic elements

In order to achieve a simpler, uniform implementation, NeTEx model elements are based on a framework (taken from Transmodel) of abstract generic types that are specialised to create the concrete XML elements. The generic elements are abstracted to separate different concerns so that the model is robust against future changes and serve to provide common properties in a consistent manner to the elements that specialise them. There are also some concrete components such as TRANSPORT MODE that are used throughout the different parts of NeTEx. This section describes such elements.

Note that to make the EIRE_NP specification more readable, certain further ancillary NeTEx elements, such as ***BookingArrangements*** and ***Notice***, which are in fact part of the general NeTEx framework, are presented later under an appropriate element in a specific frame.

8.5.1 TypeOfValue (Abstract)

TYPE OF VALUE describes a simple coded value with a name, associated url and other descriptive properties; it can be specialised to make arbitrary sets of codes used to classify other NeTEx entities, for example TYPE OF POINT, TYPE OF LINK, TYPE OF SERVICE, etc. TYPEs OF VALUE can be used to exchange application metadata that is used to validate or collect data, such as the descriptions and allowed values for codes.

TypeOfValue is abstract: it is specialised to create concrete types for specific purposes, such as ***Branding***, ***DataSource***, etc.

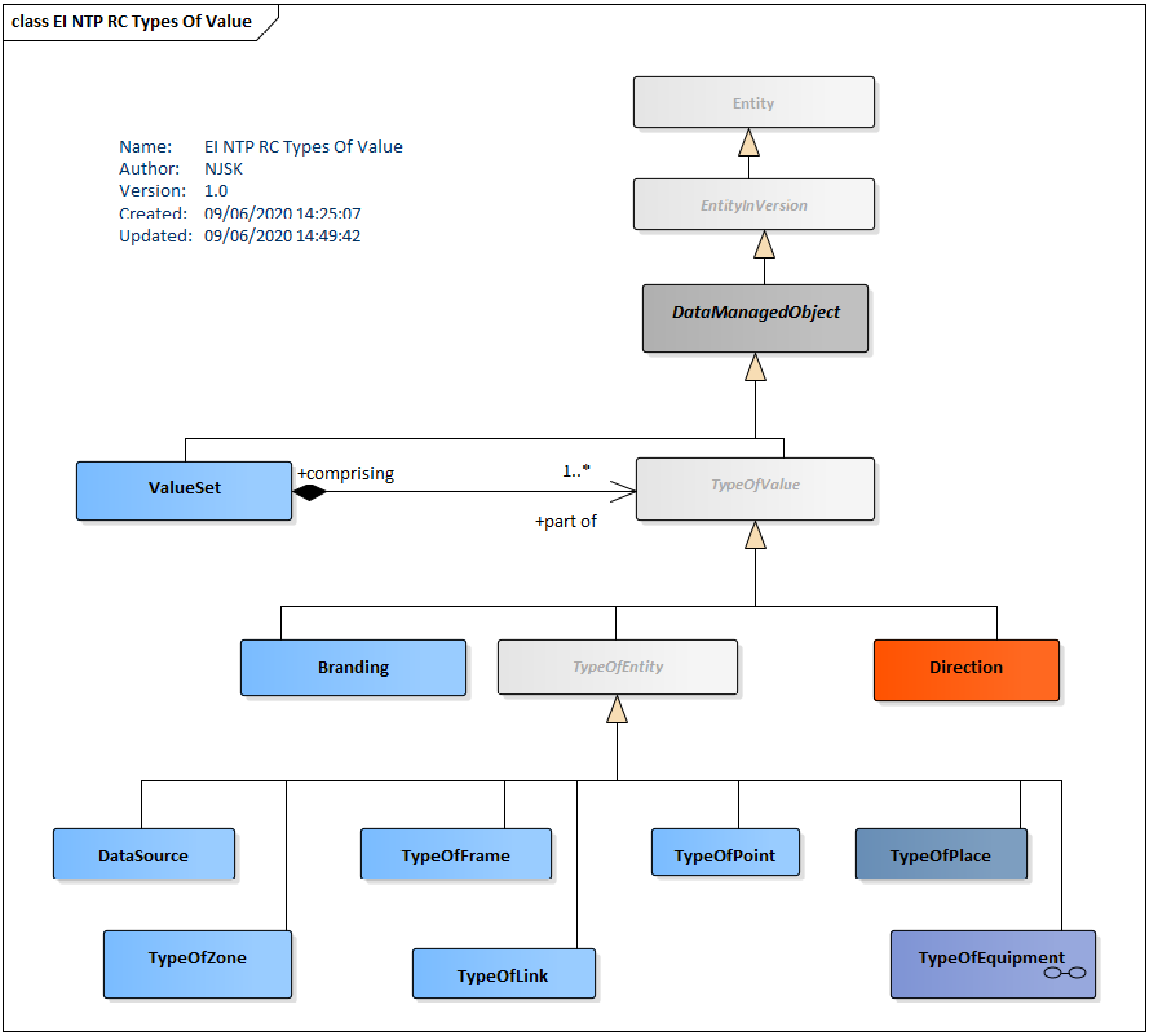


Figure 7 — Types of Value (UML)

In the EIRE_NP, in the interests of standardisation, fixed enumerated values are generally preferred, and the values encoded in schema suffice for most purpose.

Some limited us is made of ***TypeOfValue***, however, in particular ***TypeOfPlace*** instances are used to define values with which to classify a ***StopPlace*** as "*general*" or "*monomodal*". The EIRE_NP values can be considered as "profile metadata" inherent to the EIRE_NP and they can be referenced with external references without being explicitly encode in an EIRE_NP conformant document. However, an explicit encoding is allowed (and enables the checking of references by validators), provided that it conforms to the agreed definition; they should be placed in a *'EI_PI_METADATA'* frame.).

An augmented or extended profile might however want to add its own ***TypeOfValue*** instances. A particular use is to hold legacy classifiers of stops, etc, so that data sets can be exchanged losslessly in a round trip workflow

Table 22 — *TypeOfValue* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*DataManagedObject*</u>	::>	TYPE OF VALUE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*TypeOfValueIdType*	1:1	Identifier of TYPE OF VALUE.
	Name	*MultilingualString*	0:1	Name of TYPE OF VALUE.
	ShortName	*MultilingualString*	0:1	Short Name of TYPE OF VALUE.
	Description	*MultilingualString*	0:1	Description of TYPE OF VALUE.
	Image	*xsd:anyURI*	0:1	Default image for TYPE OF VALUE.
	Url	*xsd:anyURI*	0:1	URL associated with of TYPE OF VALUE.
«AK»	***PrivateCode***	*PrivateCode*	0:1	Private Code associated with TYPE OF VALUE.

NB☞ TYPEs of VALUE are used in the EIRE_NP to encode NaPTAN, NPTG classifications to facilitate roundtrip processing.

8.5.2 ValueSet

Many different categories of TYPE OF VALUE can be encoded in a NeTEx Document. A **ValueSet** can be used to group values that belong to a given category so that a consumer system can recognize them as a related set of alternative options. (Without the use of ***ValueSet*** instances, all the different instance of ***TypeOfValue*** would appear as an undifferentiated list and it would not necessarily be clear to which element they relate).

For example, the set of allowed legacy values for classifying NaPTAN stops can be grouped as a ***ValueSet***:

```
<ValueSet version="napt:v2.4" id="napt:NaPTAN_Stop_Classifications_Quays"
classOfValues="TypeOfPlace">
    <Name>NaPTAN Stop StopClassification values</Name>
        <values>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@BCT">
                <Name>NaPTAN Bus Coach Tram stop on street </Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@BCS">
                <Name> Bus / Coach bay / stand / stance within Bus / Coach
Stations.</Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@BCQ">
                <Name> Variable Bay.</Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@GAT">
                <Name>Airside area </Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@RPL">
                <Name>Rail Platform</Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@MET">
                <Name>Metro Station</Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@PLT">
                <Name>Metro Platform</Name>
            </TypeOfPlace>
            <TypeOfPlace version="napt:v2.4" id="napt:StopClassification@LPL">
            <Name>Lift Station Platform</Name>
        </TypeOfPlace>
        <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@TXR">
            <Name>Taxi Rank</Name>
        </TypeOfPlace>
        <TypeOfPlace version="napt:v2.1" id="napt:StopClassification@STR">
            <Name>Shared Taxi Rank</Name>
        </TypeOfPlace>
        <TypeOfPlace version="napt:v2.4" id="napt:StopClassification@SDA">
            <Name>Set Down Area</Name>
        </TypeOfPlace>
    </values>
</ValueSet>
```

Table 23 — *ValueSet* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	VALUE SET inherits from DATA MANAGED OBJECT.
«PK»	***id***	*ValueSetIdType*	1:1	Identifier of VALUE SET.
«atr»	***classOfValues***	*xsd:NCName*	0:1	Name of XML Class for VALUE SET contents.
	Name	*MultilingualString*	0:1	Name of VALUE SET.
«cntd»	***values***	*(TypeOfValue)*	0:*	Values in VALUE SET

8.5.3 GroupOfEntities (Abstract)

A set of ENTITies grouped together according to a PURPOSE OF GROUPING, e.g., grouping of stops known to the public by a common name.

Several different specialisations of GROUP OF ENTITies are used in the EIRE_NP: ***GroupOfLines***, ***GroupOfStopPlaces***, ***GroupOfServices***, etc (and ***Place*** is in effect a ***GroupOfPoints***).

Table 24 — *GroupOfEntities – XML Element (Abstract)*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	GROUP OF ENTITies inherits from DATA MANAGED OBJECT.
«PK»	***id***	*GroupOfEntitiesIdType*	1:1	Identifier of GROUP OF ENTITies.
	Name	*MultilingualString*	0:1	Name of GROUP OF ENTITies.
	ShortName	*MultilingualString*	1:1	Short Name of GROUP OF ENTITies.
	Description	*MultilingualString*	0:1	Description of GROUP OF ENTITies.
«FK»	***PurposeOfGroupingRef***	*PurposeOfGroupingRef*	0:1	PURPOSE OF GROUPING of GROUP OF ENTITies.
«AK»	***PrivateCode***	*PrivateCode*	0:1	Private code associated of GROUP OF ENTITies.
«cntd»	***(members)***	*(VersionOfObjectRef)+ | (GroupMember)+*	1:1	Reference to ENTITY included in the group. Each concrete specialisation must add its own set restricted to the correct type of member. References may be simple (e.g., a specialisation of ***VersionOfObjectRef***, or, if they have additional

				properties be a specialisation of GroupMember such as ***GroupOfServicesMember.***

8.5.4 Assignment (Abstract)

An ASSIGNMENT specifies a set of properties to be associated with another ENTITY on a temporary or permanent basis. An example is a STOP ASSIGNMENT, used to relate a stop in the time table with a physical stop. It may also allocate a platform (QUAY) for a journey for some or all journeys.

Table 25 — Table 6 – *Assignment* – XML Element (Abstract)

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	ASSIGNMENT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*AssignmentIdType*	1:1	Identifier of ASSIGNMENT.
«atr»	***order***	*xsd:positiveInteger*	0:1	Relative precedence of ASSIGNMENT.
	Name	*MultilingualString*	0:1	Name of ASSIGNMENT.
	Description	*MultilingualString*	0:1	Description of ASSIGNMENT.

8.5.5 Presentation (Subcomponent – Multipurpose)

A PRESENTATION defines a uniform set of values to control appearances of an element (on a map, a schematic map, a paper or screen description, etc.). It can be associated with a LINE, a TARIFF ZONE and other ENTITies.

Table 26 — *Presentation* – Type

Classificat ion	Name	Type	Cardinality	Description
	Colour	*ColourValueType*	0:1	Identifier of RGB colour value to use for ENTITY in user interface, e.g., LINE colour.
	ColourName	*xsd:string*	0:1	Text name of colour value to use for ENTITY in user interface, e.g., LINE colour.
	BackGroundColour	*ColourValueType*	0:1	Default RGB background colour value for graphics associated with ENTITY - e.g., of a LINE.
	BackgroundColour-Name	*xsd:string*	0:1	Name of the background colour value (in *ColourSystem*) for graphics associated with entity.
	TextColour	*ColourValueType*	0:1	Identifier of colour value to use for text for ENTITY in user interface, e.g., colour for name of LINE.
	TextColourName	*xsd:normalizedString*	0:1	Text name of colour value to use for text for ENTITY in user interface, e.g., name of colour for the name of LINE.
	TextFont	*xsd:normalizedString*	0:1	Identifier of font for text. +V1.1
	TextFontName	*xsd:normalizedString*	0:1	Name of font for text. +V1.1
	TextLanguage	*xsd:language*	0:1	Identifier of default language of text for ENTITY in user interface.
«cntd»	***infoLinks***	*InfoLink*	0:*	URL for image associated with entity e.g., icon.

	ColourSystem	*xsd:string*	0:1	Name of colour system that is used for ColourName and TextColourName: : may be RAL, https://en.wikipedia.org/wiki/RAL_colour_standard , DIN 6164 http://www.dtpstudio.de/atlas/farbsysteme/DIN%206164_bs00_3.htm, Pantone (be aware that Pantone is commercial), etc.

8.6 Generic Spatial elements

NeTEx uses a uniform model of POINTs and LINKs to represent the spatial extent of transport networks. The POINTs and LINKs can be assembled into LINK SEQUENCEs describing paths through the NETWORK.

There are a number of different specialisations of LINK SEQUENCE, each comprising its own respective set of POINTs and LINKs; for example, a SERVICE PATTERN is made up of SCHEDULED STOP POINTs and SERVICE LINKs a TIMING PATTERN of TIMING POINTs and TIMING LINKs.

A given LINK SEQUENCE can be subdivided into smaller subsections using a SECTION.

A LOCATION specifies the coordinates of spatial objects in a specified LOCATING SYSTEM. POINTs are 0-dimensional, LINKs are 1 dimensional. A third entity, ZONE, is used to specify 2-dimensional areas.

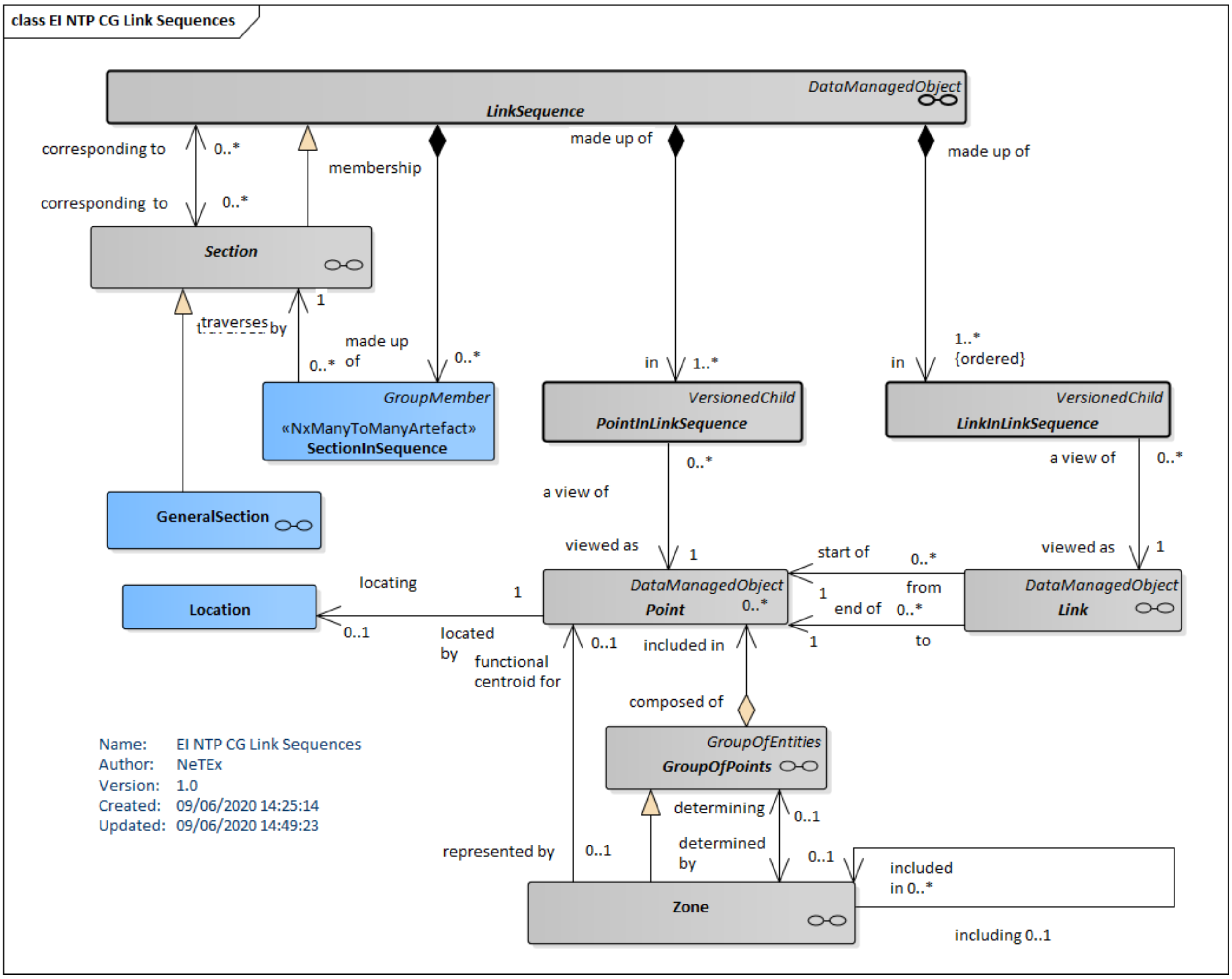


Figure 8 — NeTEx Generic spatial elements (UML)

8.6.1 Point (Abstract)

A 0-dimensional node of the network used for the spatial description of the network. POINTs may be located by a LOCATION in a given LOCATING SYSTEM.

Point is specialised to make the various Point types used in NeTEx, in particular ***ScheduledStopPoint***.

Table 27 — *Point – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	POINT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PointIdType*	1:1	Identifier of POINT.
	Name	*MultilingualString*	0:1	Name of POINT.
«cntd»	***Location***	*Location*	0:1	Location of POINT. Providing a ***Location*** for each Point with spatial coordinates is mandatory in the EIRE_NP.
«»	***PointNumber***	*xsd:normalizedString*	0:1	Arbitrary alternative identifier for the POINT.
«cntd»	***types***	*TypeOfPointRef*	0:*	Reference to one or more a TYPE OF POINT.
«cntd»	***projections***	*Projection*	0:*	Projections for POINT.
«cntd»	***groupMemberships***	*GroupOfPointsRef*	0:*	GROUPs of POINTs to which POINT belongs.

8.6.2 Location (Subcomponent of Point)

The position of a POINT with a reference to a given LOCATING SYSTEM (e.g., coordinates).

WGS84 coordinates can be given as separate ***Longitude*** and ***Latitude*** attributes; OSGR and other formats are stated in GML format.

NB☞ for NaPTAN, coordinates can thus be given in both WGS84 and OSGR at the same time.

Table 28 — *Location – XML Element*

Classification	Name	Type	Card-inality	Description
«FK»	***srsName***	*LocatingSystemNameType*	0:1	GML id of Type of LOCATING SYSTEM used. To be filled using the GML format (e.g. urn:ogc:def:crs:EPSG::4326 for WGS84, see http://www.epsg.org and http://www.opengeospatial.org/ogcUrnPolicy) Note: Only WGS84 and or OSGR may be used for the EIRE_NP.
	Longitude	*LongitudeType*	1:1	Longitude of LOCATION.
	Latitude	*LatitudeType*	1:1	Latitude of LOCATION.
	Altitude	*AltitudeType*	0:1	Altitude of LOCATION.
«cntd»	***Coordinates***	*gml:pos*	0:1	GML coordinates providing location in a specified LOCATION SYSTEM such as OSGR.
	Precision	*xsd:decimal*	0:1	Precision of coordinates.

8.6.3 Link (Abstract)

A LINK is an oriented spatial object of dimension 1 with view to the overall description of a network, describing a connection between two POINTs.

Different types of links are used to describe the different layers of a Transport Network, for example ***RouteLink***, ***ServiceLink***, etc.

Table 29 — *Link – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	LINK inherits from DATA MANAGED OBJECT.
«PK»	***id***	*LinkIdType*	1:1	Identifier of LINK.
	Name	*MultilingualString*	0:1	Name of LINK.
	Distance	*DistanceType*	0:1	Distance along LINK.
«cntd»	***types***	*TypeOfLinkRef*	0:*	Reference to one or more a TYPE OF LINK.
«cntd»	***LineString***	*gml:LineString*	0:1	GML linestring (detailed geometry) of the LINK.
«cntd»	***projections***	*Projection*	0:*	Projections for LINK.
«cntd»	***passingThrough***	*PointOnLink*	0:*	POINTs on LINK that LINK passes through. NOTE: Use ***LineString*** for LINK's shape
«FK»	(***FromPointRef***)	*PointRef*	1:1	Reference to point at which LINK starts. This an abstract reference – each specialisation of Link will reference a concrete reference, e.g., a ROUTE LINK will reference a ROUTE POINT, TIMING LINK a TIMING POINT etc
«FK»	(***ToPointRef***)	*PointRef*	1:1	Reference to point at which LINK ends. -Also Abstract.

8.6.4 LinkSequence (Abstract)

A LINK SEQUENCE describes a path through a network, such as a bus route. It can consist of an ordered set of POINTs IN SEQUENCE (referencing POINTs of a particular type) or LINKs in SEQUENCE (referencing LINKs of a particular type).

LinkSequence is abstract: it is specialised to create specific types of paths such as ***Route***, ***JourneyPattern***, ***ServicePattern***, etc.

Table 30 — *LinkSequence – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	LINK SEQUENCE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*LinkSequenceIdType*	1:1	Identifier of LINK SEQUENCE.
	Name	*MultilingualString*	0:1	Name of LINK SEQUENCE.
	ShortName	*MultilingualString*	*0:1*	*Short Name of LINK SEQUENCE.*
	Distance	*DistanceType*	1:1	Distance of LINK SEQUENCE. Units are as specified for Frame. Default is SI Metres.
	PrivateCode	*PrivateCodeType*	0:1	Private Code of LINK SEQUENCE.

«cntd»	***sectionsInSequence***	*sectionsInSequence*	0:1	SECTIONs that make up route. Can be used as an alternative to points in Sequence. POINTs and LINKs must be of the same type as sequence, e.g., ROUTE, ROUTE POINT, ROUTE LINK
«cntd»	***(pointsInSequence)***	*(XxxxPointInSequence)+*	1:*	An ordered sequence of POINT IN SEQUENCE elements that each reference a POINT of the specific type for the LINK SEQUENCE. For example, a ***ServicePattern*** holds a sequence of ***StopPointInJourneyPattern*** instances that each reference a ***ScheduledStopPoint***,
«cntd»	***(linksInSequence)***	*(XxxxLinkInSequence)+*	1:*	An ordered sequence of LINKs IN SEQUENCE elements that each reference a LINK of the specific type for the LINK SEQUENCE.

8.6.5 PointInLinkSequence (Abstract)

A POINT in a LINK SEQUENCE indicating its order in that particular LINK SEQUENCE.

Table 31 — *PointInLinkSequence – XML Element (Abstract)*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	POINT in LINK SEQUENCE inherits from VERSIONED CHILD.
«PK»	***id***	*PointInSequenceIdType*	1:1	Identifier of POINT in LINK SEQUENCE. (Abstract)
«atr»	***order***	*xsd:positiveInteger*	1:1	Order of POINT in LINK SEQUENCE within sequence.
«FK»	***LinkSequenceRef***	*LinkSequenceRef*	1:1	Reference to LINK SEQUENCE containing POINT in LINK SEQUENCE.
«containment»	***projections***	*Projection*	0:*	Projections of POINT in LINK SEQUENCE.
	Description	*MultilingualString*	0:1	Description of POINT in LINK SEQUENCE. +v1.1

8.6.6 Zone (Abstract)

A ZONE defines a two-dimensional PLACE within the service area of a public transport OPERATOR (ADMINISTRATIVE ZONE, TARIFF ZONE, ACCESS ZONE, etc.).

Table 32 — *Zone – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*GroupOfPoints*	::>	ZONE inherits from GROUP OF POINTs.
«PK»	***id***	*ZoneIdType*	1:1	Identifier of ZONE.
«cntd»	***members***	*PointRef*	0:*	List of points within ZONE.
«cntd»	***types***	*TypeOfZoneRef*	0:*	Type of ZONE.
«cntd»	***Centroid***	*Point*	0:1	Centre POINT of ZONE.
«cntd»	***gml:Polygon***	*gml:Polygon*		Polygon associated with zone.
«cntd»	***projections***	*Projection*	0:*	List of PROJECTIONs of ZONE.
«FK»	***ParentZoneRef***	*ZoneRef*	0:1	Reference to parent of ZONE.

8.7 Generic Purpose components

8.7.1 Locale (Subcomponent – Multipurpose)

One of the frame defaults is a ***Locale***; specifying parameters to be used when processing or presenting data in a given country or region.

Table 33 — *Locale* – XML Element

Classification	Name	Type	Cardinality	Description
	TimeZoneOffset	*TimeZoneOffset*	0:1	Time zone of Entity as offset in hours from GMT. plus, or minus. May be decimal for fractional differences.
	TimeZone	*xsd:normalizedString*	0:1	Name of Time zone associated with LOCALE.
	SummerTimeZoneOffset	*TimeZoneOffset*	0:1	Time zone as an offset in hours or part hours from GMT. Can be plus or minus.
	SummerTimeZone	*TimeZoneOffset*	0:1	Name of Summer Time zone associated with LOCALE.
«enum»	***DefaultLanguage***	*xsd:language*	0:1	Default language of LOCALE. (*en, fr, de,* etc)
«enum»	***languages***	*LanguageUsageEnum*	0:*	Other languages supported in LOCALE.

8.7.2 BookingArrangements (Subcomponent – Multipurpose)

Details of the booking arrangements for a LINE or SERVICE JOURNEY, e.g., if flexible.

Table 34 — *BookingArrangements* – Type

Classification	Name	Type	Cardinality	Description
«cntd»	***BookingContact***	*Contact*	0:1	Contact details or making a Booking.
«enum»	***BookingMethods***	*BookingMethodEnum*	0:*	Booking method for FLEXIBLE LINE. • *callDriver; callOffice; online*
«enum»	***BookingAccess***	*BookingAccessEnum*	0:1	Who can make a booking? • *Public; authorisedPublic; staff; other*
«enum»	***BookWhen***	*PurchaseWhenEnum*	0:1	When booking can be made. • *advanceOnly; untilPreviousDay; dayOfTravelOnly; advanceAndDayOfTravel; timeOfTravelOnly; subscriptionChargeMoment; other; advanceOnly*
«enum»	***BuyWhen***	*PurchaseMomentEnum*	0:*	When a purchase can be made. • *inAdvanceOnly, onReservation, beforeBoarding, beforeBoardingOnly, onBoarding, onBoardingOnly, afterBoarding, onCheckIn, onCheckOut, subscriptionOnly, other*
	LatestBookingTime	*MultilingualString*	0:1	Latest time in day that booking can be made.
	MinimumBooking-Period	*xsd:duration*	0:1	Minimum interval in advance of departure day or time that service may be ordered.
	BookingUrl	*xsd:anyURI*	0:1	URL for booking.
	BookingNote	*MultilingualString*	0:1	Note about booking the FLEXIBLE LINE.

8.7.3 ContactDetails (Subcomponent – Multipurpose)

Details on how to contact an ORGANISATION.

Table 35 — Table 7 — *ContactDetails – Type*

Classification	Name	Type	Cardinality	Description
	ContactPerson	*MultilingualString*	0:1	Name of contact person.
	Email	*EmailAddressType*	0:1	Email address in ISO format.
	Phone	*PhoneType*	0:1	Phone number.
	Fax	*PhoneType*	0:1	Fax number.
	Url	*xsd:anyURI*	0:1	Contact web site URL.
	FurtherDetails	*MultilingualString*	0:1	Further details of contact.

8.7.4 Transport modes and submodes

A MODE defines the means of transport used or available. NeTEx specialises MODE into TRANSPORT MODE, used to classify public transport systems for instance: bus, tramway, light rail, metro, long-distance rail, ferry.

A VEHICLE TYPE must belong to a single TRANSPORT MODE. For instance, the '*bus*' TRANSPORT MODE will include any sort of bus - standard, articulated, minibus, double-decker, etc.

A LINE is in general operated by only one TRANSPORT MODE: metro line, ferry line, etc.

EIRE_NP limits the use of MODEs to the fixed values provided as an enumeration (*VehicleMode*).

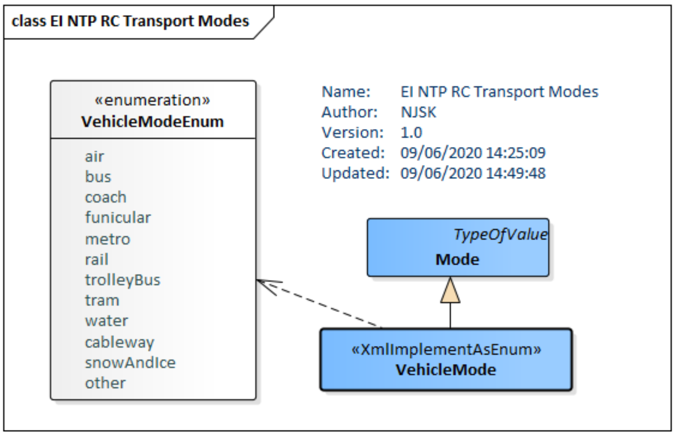


Figure 9 — Transport modes (UML)

The SUBMODE allows the TRANSPORT MODE to be further qualified with a subclassification. The subclassifications typically distinguish different contexts of operation for the mode, for example, for the '*bus*' mode, some submodes are '*schoolBus*', '*nightBus*', '*expressBus*', etc.).

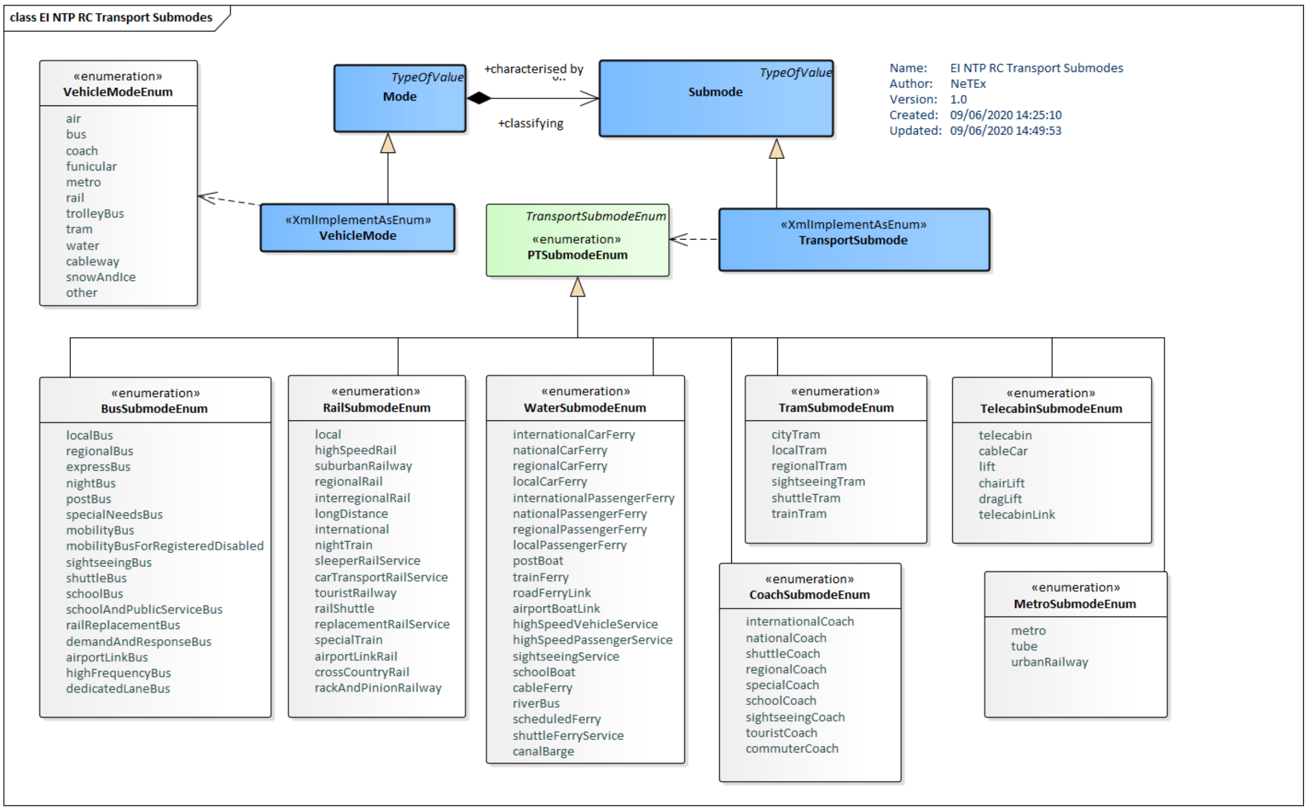


Figure 10 — Transport submodes (UML)

8.8 Accessibility

NeTEx uses a uniform set of components to describe the accessibility of sites and services, based on an ACCESSIBILITY ASSESSMENT. The use of a system of assessment makes it possible to compute the overall accessibility of a route for a given need.

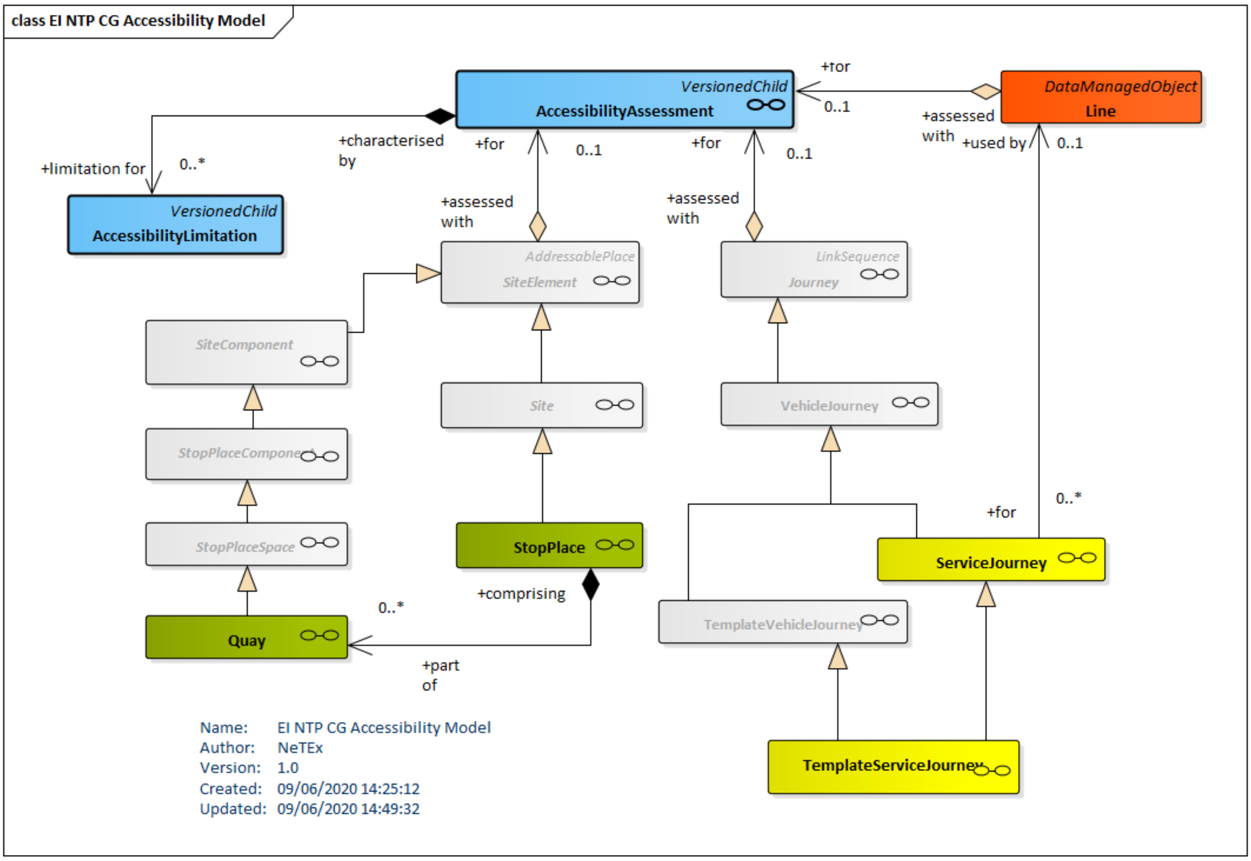


Figure 11 — Accessibility (UML)

An ACCESSIBILITY ASSESSMENT describes the accessibility of a SITE, a LINE or a SERVICE JOURNEY component (see later for the description of these ENTITies). An assessment is made up of ACCESSIBILITY LIMITATIONs, each describing the accessibility for a specific need, for example *Wheelchair, Step free, Escalator free, Lift free.*

NB☞ The EIRE_NP requires only basic accessibility data to be specified.

8.8.1 AccessibilityAssessment (Subcomponent – Multipurpose)

Assessment for accessibility purposes of limitations of environment and/or SUITABILITies for one or more USER NEEDs of a component

Table 36 — *AccessibilityAssessment – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	ACCESSIBILITY ASSESSMENT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*AssessmentIdType*	0:1	Identifier of ACCESSIBILITY ASSESSMENT.
	MobilityImpairedAccess	*LimitationStatusEnum*	1:1	Whether the overall assessment is that there is access for mobility impaired users. • *true* • *false* • *partial* • *unknown*
«cntd»	***limitations***	*AccessibilityLimitation*	0:1	The LIMITATIONs making up the ACCESSIBILITY ASSESSMENT.

«cntd»	***suitabilities***	*Suitability*	0:*	The SUITABILITies making up the ACCESSIBILITY ASSESSMENT.

8.8.2 AccessibilityLimitation (Subcomponent of AccessibilityAssessment)

A categorisation of the ACCESSIBILITY characteristics to indicate the usability by passengers with specific needs, for example, those needing wheelchair access, step-free access or wanting to avoid confined spaces such as lifts.

A small number of well-defined categories are used that are chosen to allow the consistent capture of data and the efficient computation of routes for different classes of user.

Table 37 — *AccessibilityLimitation – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*		ACCESSIBILITY LIMITATION inherits from VERSIONED CHILD.
«PK»	***id***	*LimitationIdType*	0:1	Identifier of ACCESSIBILITY LIMITATION.
«enum»	***WheelchairAccess***	*LimitationStatusEnum*	1:1	Whether there is access for wheelchair users. See values above. If absent assume *'unknown'*.
«enum»	***StepFreeAccess***	*LimitationStatusEnum*	0:1	Whether there is step-free access. See values above. If absent assume *'unknown'*.
«enum»	***EscalatorFreeAccess***	*LimitationStatusEnum*	0:1	Whether there is escalator-free access. See values above. If absent assume *'unknown'*.
«enum»	***LiftFreeAccess***	*LimitationStatusEnum*	0:1	Whether there is lift free access. See values above. If absent assume *'unknown'*.
«enum»	***AudibleSignsAvailable***	*LimitationStatusEnum*	0:1	Whether there are audible signs. See values above. If absent assume *'unknown'*.
«enum»	***VisualSignsAvailable***	*LimitationStatusEnum*	0:1	Whether there are visual signs. If absent assume *'unknown'*.

Note that the available values **LimitationStatus** are: `'true'`, `'false'`, `'unknown'` or `'partial'`; a value of *'partial'*" can have different meanings:

- *Temporal*, for example: the wheelchair access relies on a personal service only available at certain times, and/or on certain days.
- *Geographic*, for example: a station where there is, wheelchair access only to some QUAYs (not all).
- *Contextual*, for example: audio information is available only in case of disruption, or for a specific class of service.

For the EIRE_NP, whenever a "*partial*" ***LimitationStatus*** is used, then a "***ValidityCondition. Description***" shall be provided to explain why the accessibility is only partial (note that only the ***Description*** of the ***ValidityCondition*** can be filled). The contained textual information shall be suitable for presentation to the public without further editing.

NeTEx XML EXAMPLE

The following code fragment shows an ***AccessibilityAssessment*** that might be associated with a STOP PLACE>:

```
<AccessibilityAssessment version="any" id="mybus:SSP_02456A">
```

```
    <MobilityImpairedAccess>true</MobilityImpairedAccess>
<limitations>
    <AccessibilityLimitation>
        <WheelchairAccess>true</WheelchairAccess>
        <StepFreeAccess>true</StepFreeAccess>
    </AccessibilityLimitation>
</limitations>
```

8.9 Facility sets

Passenger Information can include advice as to the facilities available at a stop or onboard a service, for example restaurants, toilets for the disabled, WIFI, etc.

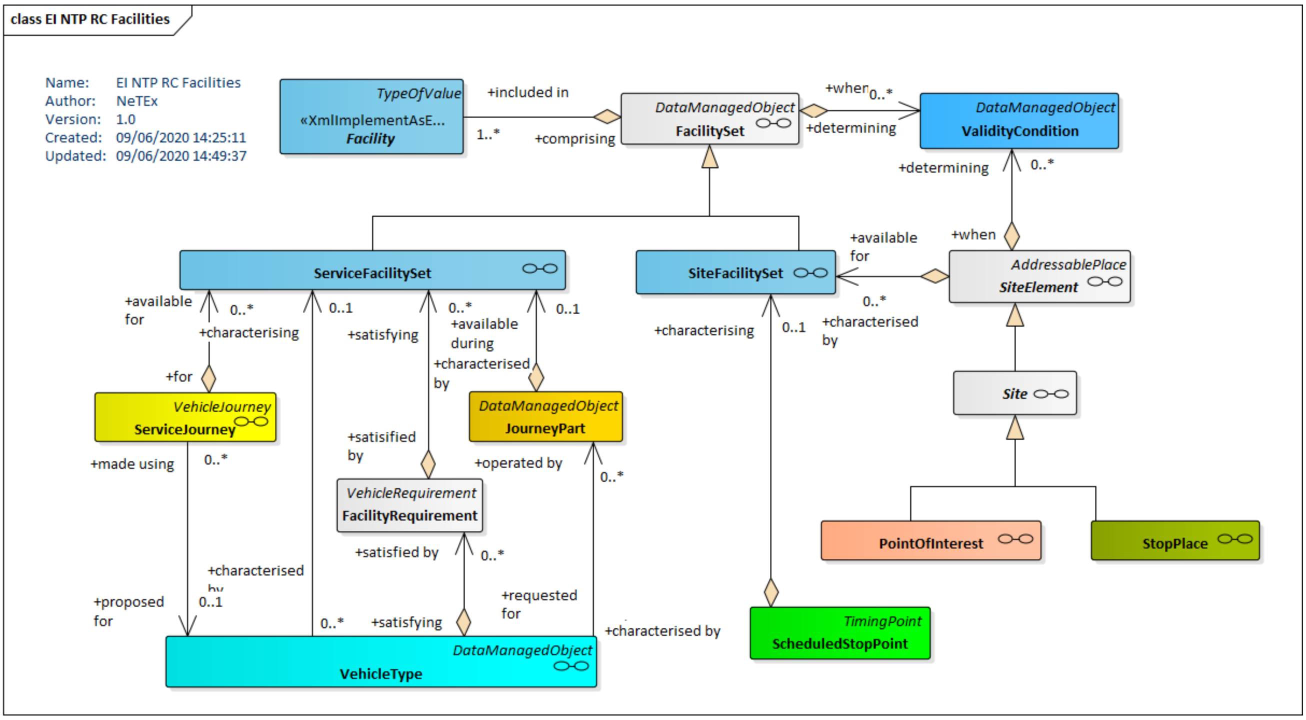


Figure 12 — Facility Sets (UML)

A FACILITY is a named amenity or other service that is available onboard or at a station, for example '*buffetCar*', '*leftLuggage*'. NeTEx provides a fixed set of standardised values for facility names; the list of all possible facility names is described in Annex B of this document.

A FACILITY SET describes a set of FACILITies that may be associated with an ENTITY.

A FACILITY SET may be subject to a specific VALIDITY CONDITION to indicate it is only valid at a specific time.

There are separate specialisations of FACILITY SET for SITEs and for SERVICEs.

SITE FACILITY SETs can only be associated with a STOP PLACE; SERVICE FACILITY SETs can only be used for a SERVICE JOURNEY and a JOURNEY PART.

Note that FACILITY data has not been provided historically for stops or services in Irish NaPTAN data.

8.9.1 Facility Set (Abstract)

FACILITY SET specifies the common properties of SITE FACILITY SET and SERVICE FACILITY SET.

Within a given instance of a FACILITY SET, the specified values are logically ANDed together. For example, '***FareClass*** =*firstClass*' and '***CateringFacility*** =*'restaurant'*.

Table 38 — *FacilitySet – XML Element (Abstract)*

Classification	Name	Type	Cardinality	Description
«PK»	***id***	*FacilitySetIdType*	1:1	Identifier of FACILITY SET.

«FK»	***ProvidedByRef***	*OrganisationRef*	1:0	ORGANISATION responsible for providing facilities in FACILITY SET.
	Description	*MultilingualString*	0:1	Description of FACILITY SET.
	TypeOfFacilityRef	*TypeOfFacilityRef*	0:1	Classification of FACILITY SET.
«cntd»	***other Facilities***	*TypeOfEquipment*	0*	Arbitrary user defined FACILITY types in FACILITY SET.
XGRP	***CommonFacility-Group***	*(xxxFacilityList)*	Nx(0:*)	A choice of ***FacilityLists***; each ***FacilityList*** is allows the selection of one or more FACILITies (each encoded as an enumerated value) on a particular topic. See Annex. The common lists may be used in any FACILITY SET. There are additional facilities that are specific to the SERVICE FACILITY SET and the SITE FACILITY SET.

8.9.2 ServiceFacilitySet

A SERVICE FACILITY SET describes the FACILITies available onboard a service. It additionally allows the type of ACCOMMODATION (seating, couchettes, etc) to be described.

Table 39 — *ServiceFacilitySet – XML Element*

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*FacilitySet*	::>	SERVICE FACILITY SET inherits from FACILITY SET.
«PK»	***id***	*ServiceFacilitySetIdType*	1:1	Identifier of SERVICE FACILITY SET.
XGRP	***ServiceFacilityGroup***	***xmlGroup of choices***	0:1	Facilities specific to SERVICE FACILITY SET. See Annex.
«cntd»	***accommodations***	*Accommodation*	0*	ACCOMMODATIONs in SERVICE FACILITY SET.
«cntd»	onboardStays	*OnboardStay*	0:*	ONBOARD STAYs in SERVICE FACILITY SET.

8.9.3 Accommodation (Subcomponent of ServiceFacilitySet)

Describes the accommodation characteristics available on a service, e.g. "*First Class Couchette with Shower* "

Table 40 — Table 8 — *Accommodation – XML Element*

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*VersionedChild*	::>	ACCOMMODATION inherits from VERSIONED CHILD.
«PK»	***id***	*AccommodationIdType*	1:1	Identifier of ACCOMMODATION.
	Name	*MultilingualString*	0:1	Name of ACCOMMODATION.
«FK»	***ServiceFacilitySetRef***	*ServiceFacilitySetRef*	0:1	Parent SERVICE FACILITY SET for ACCOMMODATION.. +v1.1
«enum»	***FareClass***	*FareClassEnum*	0:1	FARE CLASS of ACCOMMODATION. See Annex for allowed values.
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	CLASS OF USE of ACCOMMODATION.. +v1.1

«enum»	***Accommodation-Facility***	*Accommodation-FacilityEnum*	0:1	Type of accommodation Facility in ACCOMMODATION. See Annex for allowed values.
«enum»	***CouchetteFacility***	*CouchetteFacilityEnum*	0:1	Couchette Facility in ACCOMMODATION. See NeTEx.
	MaximumNumberOfBerths	*xsd:integer*	0:1	Maximum number of berths in ACCOMMODATION.
«enum»	***BerthFacility***	*BerthFacilityEnum*	0:1	Berth Facility in ACCOMMODATION. See NeTEx.
«enum»	***ShowerFacility***	*SanitaryFacilityEnum*	0:1	Shower Facility in ACCOMMODATION. See NeTEx.
«enum»	***ToiletFacility***	*SanitaryFacilityEnum*	0:1	Toilet Facility in ACCOMMODATION. See NeTEx.
«enum»	***GenderLimitation***	*GenderLimitationEnum*	0:1	Gender Limitation for ACCOMMODATION. E.g., Women only. See facility values.
«enum»	***NuisanceFacility***	*NuisanceFacilityEnum*	0:1	Nuisance Facility in ACCOMMODATION. See Annex B.
«enum»	***Passenger-CommsFacility***	*PassengersComms-FacilityEnum*	0:*	Communications facilities within the accommodation section See Annex B.

8.9.4 SiteFacilitySet

A SITE FACILITY SET describes the Facilities available at a fixed site.

Table 41 — *ServiceFacilitySet – XML Element*

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*FacilitySet*	::>	SITE FACILITY SET inherits from FACILITY SET.
«PK»	***id***	*SiteFacilitySetIdType*	1:1	Identifier of SITE FACILITY SET.
XGRP	***SiteFacilityGroup***	***xmlGroup***	0:1	Facilities specific to SITE FACILITY SET. See Annex B.

9 Stop Profile

The Stop profile part of the EIRE_NP, (EIRE_NP-Stop) describes stops for any mode of transport and their related properties.

It includes

- STOP PLACEs – Stations, bus stops, ferry ports, etc where passengers may alight or board from public transport.
- TOPOGRAPHIC PLACEs – Towns, cities, villages, regions etc associated with stop places and relevant for naming them and finding them in journey planners.
- TARIFF ZONEs – fare zones associated with stops.
- POINTs OF INTEREST – Places to which passengers might want to travel and to know which are the most suitable stops.

NB☞ The Stop Profile corresponds to the NaPTAN stop data content and its supporting NPTG data sets.

9.1 Defining places in NeTEx

For EIRE_NP stop data, the primary entity is the STOP PLACE. However, this entity inherits its properties through quite a deep hierarchy, and a basic knowledge of the hierarchy is required in order to have an accurate understanding of the model.

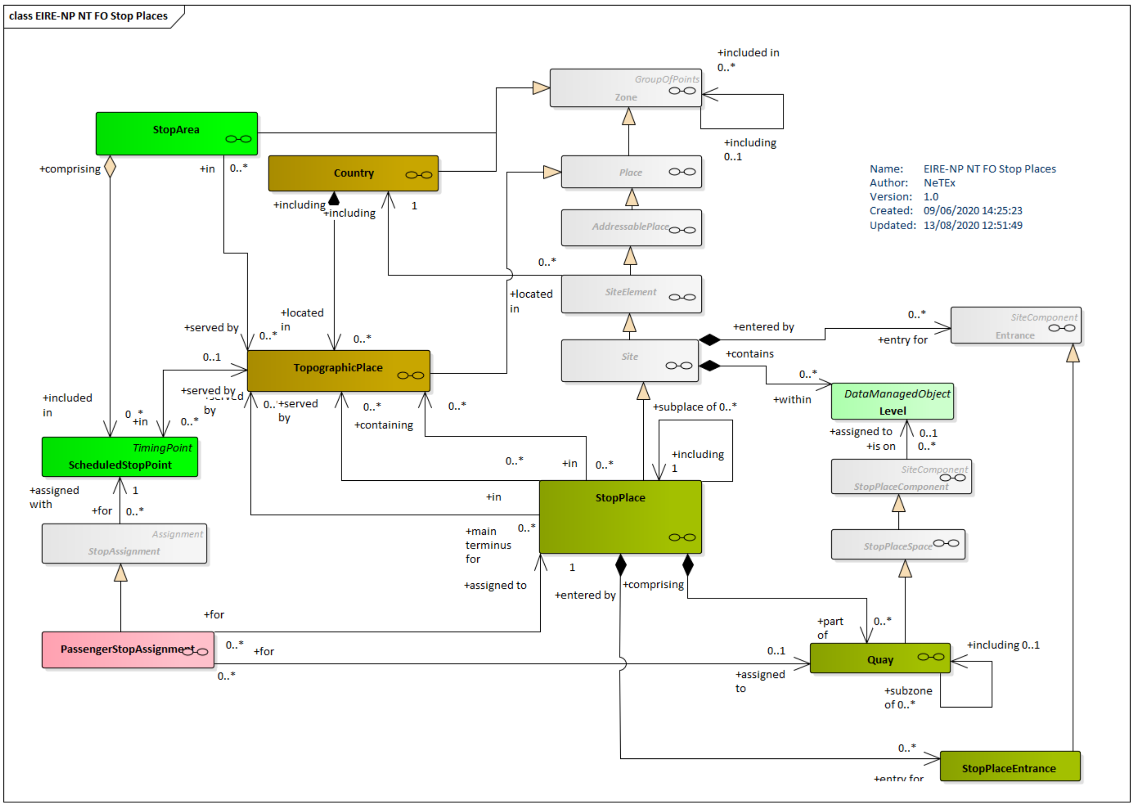


Figure 13 — Stop Places (UML)

For EIRE_NP stop data, the primary entity is the STOP PLACE. However, this entity inherits its properties through quite a deep hierarchy, and a basic knowledge of the hierarchy is required in order to have an accurate understanding of the model.

1. The top object in the inheritance tree for a STOP PLACE is a ZONE, describing a generic two-dimension object. A ZONE may be defined by a GROUP OF POINTs belonging to the ZONE, and may also be defined as a geometric area, bordered by a polygon (based on the GML Linear Ring feature).
    - A ZONE may nest other smaller ZONEs (this nesting may be multilevel). This is expressed by the reflexive relationship on ZONE (so a STOP PLACE may include other STOP PLACEs).
    - A ZONE may be represented by a single POINT (by the Centroid attribute of a Zone attribute). A centroid point defines a special kind of point that can be used as a point reference to the ZONE itself. This is useful for representing flexible transport systems (where a stop is often a ZONE).
2. The second level of the hierarchy is the PLACE, which represents any significant place that a transport model may wish to describe, and for which the possibility of travel may exist between. A PLACE can be specialised variously into a TOPOGRAPHIC PLACE (a named city, town or region) with a wide extent, or a specific ADDRESSABLE PLACE having an ADDRESS that is either a ROAD ADDRESS, or a POSTAL ADDRESS.
3. A SITE ELEMENT specialises ADDRESSABLE PLACE to add accessibility (see ACCESSIBILITY ASSESSMENT) and other properties common to any place that may be navigated through by a passenger.
4. A SITE specialises SITE ELEMENT to provide a general description of common properties of a physically situated location, such as a station or point of interest, including its entrances, levels, equipment, paths, accessibility properties, etc. The SITE is refined by specific subtypes such as STOP PLACE, POINT OF INTEREST, PARKING, etc. The basic elements making up a SITE are:
    - SITE COMPONENT (not shown): an element of a SITE describing part of its structure, such as a platform, concourse, ticket hall, entrance hall forecourt, room, retail area, etc. These may be specialised in submodels. (QUAY and STOP PLACE ENTRANCE are examples shown above)
    - LEVEL (not shown): an identified storey (ground, first, basement, mezzanine, etc.) within an interchange building or SITE on which SITE COMPONENTs reside.
    - ENTRANCE (not shown): a physical entrance or exit to/from a SITE. Each type of site will have its own ENTRANCE TYPEs (E.g., STOP PLACE ENTRANCE, POINT OF INTEREST ENTRANCE, etc).
5. The STOP PLACE describes different aspects of a physical point of access to transport, such as a stop or station. For a complex interchange, such as a station, this includes all the component areas of the station: the entrances, concourses, platforms, the levels they are on, etc. It also allows detailed accessibility attributes to be recorded at both the element and the station level.

A STOP PLACE represents a physical stop or station. Note that a STOP PLACE is a distinct concept from the representation of the stop in a timetable – the SCHEDULED STOP POINT. The two can be linked using a STOP ASSIGNMENT.

Physically, the SCHEDULED STOP POINT may correspond either to a whole STOP PLACE or to a specific QUAY (i.e., a SCHEDULED STOP POINT is a "logical stop" whose spatial position is conventionally assigned to the centroid of the STOP PLACE for journey planning purposes). Conversely, a physical STOP PLACE or a QUAY may be viewed in different functional contexts and be subject to several different assignments to different SCHEDULED STOP POINTs (see ***10.4.1-StopAssignment***).

The physical point of access to transport is always a QUAY. STOP PLACE ENTRANCEs describe the internal and external entrances to the STOP PLACE.

Because they inherit an "includes" relationship from ZONE, STOP PLACEs can be organised into a hierarchy to describe clusters of transport interchanges, such as a paired rail and tube station.

QUAYs can be nested; this allows one to represent composite platforms with two or more sides or named sections. One can thus journey plan to any required level of detail.

STOP PLACE ENTRANCEs describe points at which a passenger can access a stop place, normally on foot – an ACCESS MODE can be used to identify other permitted modes of entries such as cycle or car. (STOP PLACE VEHICLE entrance ENTRANCEs can be external, for example the main entrance, or internal, for example from an entrance concourse to a platform.

9.2 Stop Place hierarchies

To describe clusters of stops that are embedded within larger transport interchanges, STOP PLACEs can conveniently be nested (using the ***ParentSiteRef*** attribute of the inherited SITE description).

However, note that the EPIP specification imposes a constraint on the number of levels allowed for cross European data exchange. This is for two reasons (a) to simplify the semantics of applying different levels to a site and (b) to avoid interoperability issues (since many legacy systems only support a limited degree of nesting).

- A maximum of two levels of STOP PLACE will be accepted.
- Each nested level shall have a clear semantic:
  - An EPIP "General" STOP PLACE may have QUAYs or contain monomodal STOP PLACEs. It is expected that the contained monomodal STOP PLACE are of different modes, but this is not mandatory. An EPIP "General" STOP PLACE shall not be contained in another STOP PLACE. An EPIP "General" STOP PLACE may have STOP PLACE ENTRANCEs.
  - An EPIP "Monomodal" STOP PLACE shall have at least one QUAY, and shall not contain any other STOP PLACE; all the QUAYs of an "EPIP Monomodal" STOP PLACE shall be used for the same transport mode. A Monomodal STOP PLACE may have STOP PLACE ENTRANCEs.
- There is an implicit SITE CONNECTION between all the QUAYs contained in the same "General" STOP PLACE and "Monomodal" STOP PLACE hierarchy (meaning that, even if not formally described, there is the possibility of walking between any two QUAYs of a STOP PLACE, thus allowing passenger to make an INTERCHANGE over a CONNECTION).

This is summarised in the following two diagrams:

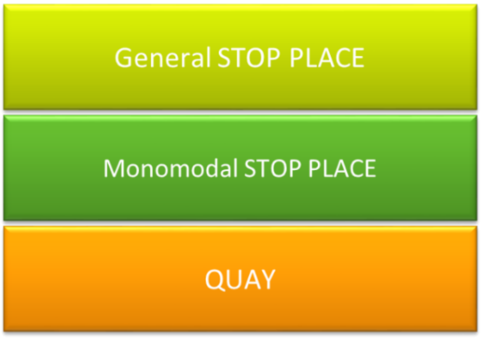


Figure 14 — EPIP Stop Place nesting hierarchy

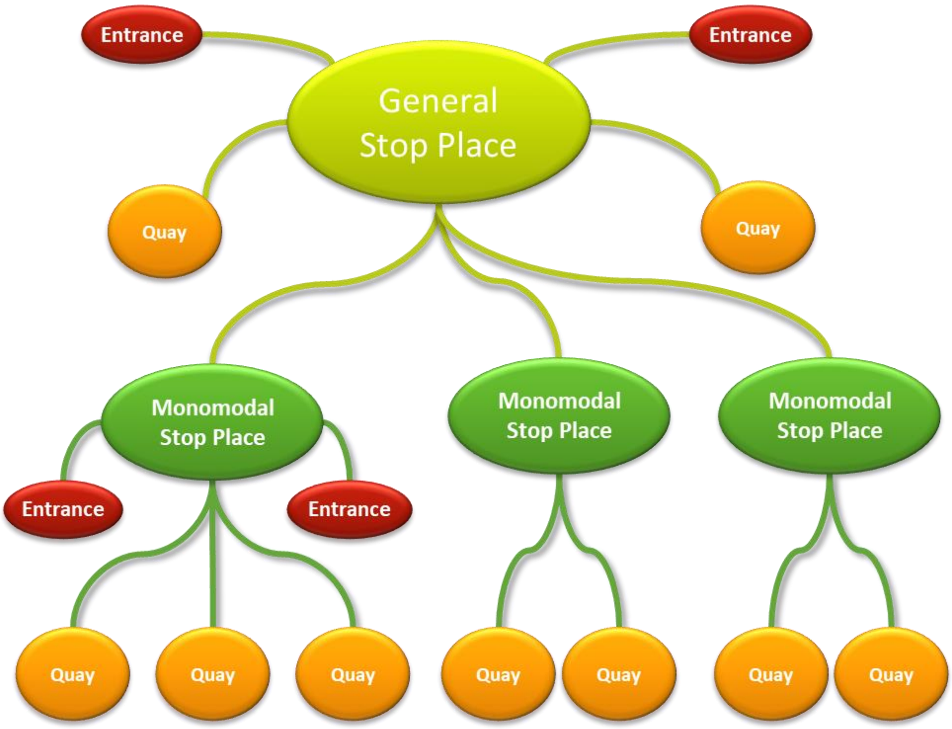


Figure 15 — EPIP Stop Place nesting example

The following table provides a compliance mapping across modes and quay types in order to build consistent “EPIP Monomodal” STOP PLACEs.

Table 42 — *Types of QUAYs and Mode compatibility*

Type of QUAY	Possible transport mode
railPlatform	*rail, intercityRail, urbanRail*
metroPlatform	*metro, funicular*
tramStop, tramPlatform	*tram*
busStop, busBay	*bus, coach, trolleyBus*
boatQuay, ferryLanding	*water*

telecabinePlatform	*cableway*
airlineGate	*air*

For very large clusters of stops, an additional level or organisation can be defined using another entity; a GROUP OF STOP PLACEs. For example, for an airport within which there are no walking connections between terminals, or to create a group with the main termini of a city that is to be used it as a general origin or destination for the city in a journey planner.

9.3 Places

As well as STOP PLACEs, several other PLACE entities are relevant for location finding in journey planners.

A TOPOGRAPHIC PLACE represents the named settlements (for example Cities, Towns, Villages, Regions, Districts, etc.) to which PT data, in particular STOP PLACEs (but also other entities), may be related. Some countries provide a standard gazetteer of the official names of such places to ensure that consistent (and official) names are used (for example, the National Public Transport Gazetteer).

A TOPOGRAPHIC PLACE can be located within a COUNTRY. TOPOGRAPHIC PLACEs may overlap. They may also be contained inside another TOPOGRAPHIC PLACE. They may have alternative official names in different national languages.

NB☞ NPTG Locality is a TOPOGRAPHIC PLACE.

POSTAL and ROAD ADDRESSes may be associated with SITEs and individual SITE ELEMENTs to facilitate stop finding and integration with navigation systems.

NB☞ NaPTAN POINTs include address data.

A POINT of INTEREST is another type of SITE used to represent a well-known attraction such as a museum, stadium, park, venue, etc. A POINT OF INTEREST may be categorised by one or more POINT OF INTEREST classifications. These can be organised into a POINT OF INTEREST CLASSIFICATION HIERARCHY which provides a way of exchanging standardised categorisations of the POINTs OF INTEREST.

POINTs OF INTEREST are available for use through the EPIP, as requested by the EC's *ITS Directive (priority action A),* although they do not correspond to an existing Irish PT data set and are not populated.

The NaPTAN v2.5 specification also allows for POINT OF INTEREST data (this was used by some stakeholders for the 2012 Olympics) but it not currently in general use.

NB☞ The EIRE_NP also allows the specification of a POINT OF INTEREST ENTRANCE to facilitate journey planning.

However, the actual POINTs OF INTEREST provided will be limited to those connected to STOP PLACEs.

9.3.1 Place (Abstract)

The PLACE entity defines topographically significant places that a transport model may wish to describe. It also allows the description of the possibility of connecting between them. A PLACE may be of dimension 0 (a POINT), 1 (a road section) or 2 (a ZONE).

Table 43 — *Place – XML Element* (Abstract)

Classification	**Name**	**Type**	**Cardinality**	**Description**

::>	::>	*Zone*	::>	PLACE inherits from ZONE.
«PK»	***id***	*PlaceIdType*	1:1	Identifier of PLACE. (Abstract)
«cntd»	***placeTypes***	*TypeOfPlaceRef*	0:*	Categorisation of PLACE. In the EPIP, STOP PLACEs must be assigned one of the EPIP categories. For example: `<TypeOfPlaceRef versionRef=”epip:1.0” ref="epip:monomodal"/>` **NB**☞ In the EIRE_NP , stops must be classified according to their NaPTAN classification. See STOP PLACE below.

NeTEx XML EXAMPLE

The following example shows a STOP PLACE that has been categorized with both the NaPTAN legacy categories, and the EPIP categories.

```
    <StopPlace version="any" id="naptStop: 8220DB000182"
responsibilitySetRef="nptgAdminArea:822">
        <Name>Glasnevin</Name>
        <StopPlaceType>onstreetBus</StopPlaceType>
        <quays>
            <Quay id="naptStop:260010966" version="1">
            ::::::
                    <placeTypes>
                        <TypeOfPlaceRef version="napt:v2.1"
ref="napt:StopClassification@BCT"/>
                        <TypeOfPlaceRef version="napt:v2.1" ref="napt:StopType@MKD"/>
                        <TypeOfPlaceRef version="epip:1.0" ref="epip:monomodal"/>

                    </placeTypes>
                    <RoadAddress version="any" id="naptStop: 8220DB000182@address">
                        <RoadName>Glasnevin Hill</RoadName>
                        <BearingCompass>SE</BearingCompass>
                    </RoadAddress>
                    <TransportMode>bus</TransportMode>
                    <QuayType>busStop</QuayType>
                </Quay>
            </quays>
        </StopPlace>
```

9.3.2 AddressablePlace (Abstract)

An ADDRESSABLE PLACE is a PLACE having an address, either a ROAD ADDRESS, or a POSTAL ADDRESS. It provides common address properties for specialisations such as STOP PLACE and POINT OF INTEREST.

ADDRESSes by themselves (i.e., for regular houses, buildings, etc that are not SITEs) are also exchangeable with NeTEx but are not part of the EPIP.

Table 44 — *AddressablePlace – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*Place*	::>	ADDRESSABLE PLACE inherits from PLACE.
«PK»	***id***	*AddressablePlaceIdType*	1:1	Identifier of an ADDRESSABLE PLACE.
	Url	*xsd:anyURI*	0:1	Default URL for ADDRESSABLE PLACE.
	Image	*xsd:anyURI*	0:1	Default image for ADDRESSABLE PLACE.

«cntd»	***PostalAddress***	*PostalAddress*	0:1	A POSTAL ADDRESS to which mail can be sent.
«cntd»	***RoadAddress***	*RoadAddress*	0:1	ADDRESS of a numbered building on a named road.

9.3.3 SiteElement (Abstract)

A SITE ELEMENT describes common properties of a spatially situated place.

Table 45 — *SiteElement – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*AddressablePlace*	*::>*	SITE ELEMENT inherits from *ADDRESSABLE* PLACE.
«PK»	***id***	*SiteElementIdType*	0:1	Identifier of SITE ELEMENT.
«cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	ACCESSIBILITY of SITE ELEMENT.
«cntd»	***AccessModes***	*AccessModeEnum*	0:*	Access modes by which SITE ELEMENT may be accessed. • *Foot, bicycle. boat, car. taxi, shuttle, ski, skate*
	NameSuffix	*MultilingualString*	1:1	Suffix to use to qualify name (mainly for bus modes). For example, two different bus stop QUAYs either side of the road might have the same name "*The White Horse"* but be distinguished by suffixes "*next to"* and "*opposite*".
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for the SITE ELEMENT. Used for stops that have multiple aliases.
	CrossRoad	*MultilingualString*	0:1	Name of nearest cross road or crossing street on which site is and which can be used to locate SITE. Mainly relevant for bus modes, where the cross road may provide necessary context to describe the stop to a user.
	Landmark	*MultilingualString*	0:1	Name of a nearby Landmark. Mainly relevant for bus modes.
«cntd»	***SiteElement-PropertiesGroup***	***xmlGroup***	0:1	Further properties of the element: see below.

Table 46 — *SiteElementPropertiesGroup* – Group

Classification	Name	Type	Cardinality	Description
«enum»	***PublicUse***	*PublicUseEnum*	0:1	Whether the SITE ELEMENT can be used by the general public. • *publicOnly, staffOnly, authorisedPublic-Only* • *disabledPublicOnly, all*
«enum»	***Covered***	*CoveredEnum*	0:1	Whether element is covered or outdoors. • *Indoors, outdoors, covered, mixed, unknown*
«enum»	***Gated***	*GatedEnum*	0:1	Whether element is within a gated area. • *gatedArea, openArea, unknown*
«enum»	***Lighting***	*LightingEnum*	0:1	How element is lit. • *wellLit, poorlyLit, unlit, other, unknown*

	AllAreasWheelchair	*xsd:boolean*	0:1	Whether all areas of component are accessible in a Wheelchair.
	PersonCapacity	*NumberOfPeople*	0:1	Number of people that can be in component at a time.
«cntd»	***facilities***	*SiteFacilitySet*	0:*	SITE FACILITY SET associated with SITE ELEMENT.

9.3.4 Site (Abstract)

A type (i.e., specialisation) of PLACE, such as a STOP PLACE, POINT OF INTEREST, to which passengers may wish to travel.

Table 47 — *Site – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*SiteElement*	::>	SITE inherits from SITE ELEMENT.
«PK»	***id***	*SiteIdType*	1:1	Identifier of SITE.
«FK»	***TopographicPlaceRef***	*TopographicPlaceRef*	0:1	Reference to a TOPOGRAPHIC PLACE with which the SITE is associated.
«cntd»	***additional-TopographicPlaces***	*TopographicPlaceRef*	0:*	Additional TOPOGRAPHIC PLACEs associated with SITE.
«enum»	***SiteType***	*SiteTypeEnum*	0:1	Type of SITE.
	AtCenter	*xsd:boolean*	0:1	Whether the site is central to the locality, referenced at the town centre.
«cntd»	***Locale***	*Locale*	0:1	Locale setting time zone, default language, etc., for the SITE.
«FK»	***OrganisationRef***	*OrganisationRef*	0:1	Reference to OPERATOR of SITE.
«FK»	***ParentSiteRef***	*SiteRef*	0:1	Parent SITE which contains this SITE. The ***ParentSiteRef*** will be used to describe nested SITEs and especially nested STOP PLACEs
«cntd»	***adjacentSites***	*SiteRef*	0:*	Adjacent Sites to SITE.
«FK»	***ContainedInPlaceRef***	*TopographicPlaceRef*	0:1	Parent SITE which contains this SITE.
«cntd»	***levels***	*Level*	0:*	LEVELs for SITE.
«cntd»	***entrances***	*Entrance*	0:*	ENTRANCEs to SITE.
«cntd»	***equipmentPlaces***	*EquipmentPlace*	0:*	EQUIPMENT PLACEs associated with SITE COMPONENT.
«cntd»	***placeEquipments***	*PlaceEquipment*	0:*	Items of fixed EQUIPMENT that may be located in places within the SITE ELEMENT.
«cntd»	***localServices***	(LocalService)	0:*	LOCAL SERVICEs that may be located in PLACEs within the SITE ELEMENT.

9.3.5 Address (Abstract)

Common attributes of an ADDRESS.

Table 48 — *Address – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Place*	::>	ADDRESS inherits from PLACE.
«PK»	***id***	*AddressIdType*	1:1	Identifier of an ADDRESS.

«FK»	***CountryRef***	*CountryEnum*	0:1	COUNTRY for ADDRESS.
	CountryName	*MultilingualString*	0:1	Name of country to use on address.

9.3.6 PostalAddress (Subcomponent of SITE in EIRE_NP)

A specialisation of ADDRESS representing a postal address – used in conjunction with GIS data, provides a means of locating a PLACE in a journey planner.

Used on STOP PLACE, QUAY, etc. to indicate the address. See NaPTAN Mapping later.

NB☞ Where two place names are needed to specify an address, 'suburb' can be used to be more specific. For example, for "*2 Brocas Way, Horton, Leighton Buzzard, Buckinghamshire*", "2" maps into ***HouseNumber***, "*Brocas Way*" maps into ***Street***, "*Leighton Buzzard*" into ***Town***, "*Horton*" into a ***Suburb,*** "*Buckinghamshire*" into ***Province***.

NB☞ Where there is both a building and a street number, a ***BuildingName*** can be used. For example, "*1501 High Flats, 16 High Road, Leyton*", "16" maps into ***HouseNumber***, "*1501 High Flats*" maps into ***BuildingName***, "*High Road*" maps into ***Street***, "*Leyton*" into ***Town.***

Table 49 — *PostalAddress – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Address*	::>	POSTAL ADDRESS inherits from ADDRESS.
«PK»	***id***	*PostalAddressIdType*	1:1	Identifier of POSTAL ADDRESS.
	HouseNumber	*xsd:normalizedString*	0:1	House or building number of POSTAL ADDRESS.
	BuildingName	*MultilingualString*	0:1	Building name of POSTAL ADDRESS.
	AddressLine1	*MultilingualString*	0:1	First line of POSTAL ADDRESS.
	AddressLine2	*MultilingualString*	0:1	Second line of POSTAL ADDRESS.
	Street	*MultilingualString*	0:1	Street name of POSTAL ADDRESS.
	Town	*MultilingualString*	0:1	Town of POSTAL ADDRESS.
	Suburb	*MultilingualString*	0:1	Suburb of POSTAL ADDRESS.
	PostCode	*PostCodeType*	0:1	Postcode.
	PostCodeExtension	*xsd:normalizedString*	0:1	Postcode extension.
	PostalRegion	*xsd:normalizedString*	0:1	Postal Region.
	Province	*MultilingualString*	0:1	Postal Province.
«FK»	***RoadAddressRef***	*RoadAddressRef*	0:1	ROAD ADDRESS associated with POSTAL ADDRESS.

9.3.7 RoadAddress (Subcomponent of SITE in EIRE_NP)

A specialisation of ADDRESS representing any position on a named road. Used in conjunction with GIS data, provides a means of locating a PLACE in a journey planner.

NB☞ Currently NaPTAN provides only a ***RoadName***. NeTEx allows both named sections and a road number to be given.

Table 50 — *RoadAddress – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Address*	::>	ROAD ADDRESS inherits from ADDRESS.

«PK»	***id***	*RoadAddressIdType*	1:1	Identifier of a ROAD ADDRESS.
«EK»	***GisFeatureRef***	*xsd:normalizedString*	0:1	Reference to a GIS feature identifier.
	RoadNumber	*xsd:normalizedString*	0:1	Number of ROAD.
	RoadName	*MultilingualString*	0:1	Name of ROAD or street.
«enum»	***BearingCompass***	*CompassEnum*	0:1	Compass bearing of ROAD at point of ADDRESS.
	BearingDegrees	*xsd:integer*	0:1	BEARING in degrees at point of ADDRESS.
	OddNumberRange	*RoadNumberRangeStructure*	0:1	Odd number range of ADDRESS.
	EvenNumberRange	*RoadNumberRangeStructure*	0:1	Even number range of ADDRESS on the road.

9.4 Topographic places

To help locate and name stops, all STOP PLACEs should be assigned to a TOPOGRAPHIC PLACE.

9.4.1 Topographic Place

A TOPOGRAPHIC PLACE is a geographical settlement which provides topographical context when searching for or presenting travel information, for example as the origin or destination of a trip. It may be of varying size (e.g., County, City, Town, Village) and of different specificity e.g., *'Greater London', 'London', 'West End', 'Westminster', 'St James's'*.

Table 51 — *TopographicPlace – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Place*	::>	TOPOGRAPHIC PLACE inherits from PLACE.
«PK»	***id***	*TopographicalPlaceIdType*	1:1	Identifier of a TOPOGRAPHIC PLACE.
	IsoCode	*IsoSubdvisionCodeType*	0:1	ISO code for a TOPOGRAPHIC PLACE.
«cntd»	***Descriptor***	*TopographicPlaceDescriptor*	1:1	Descriptive elements for a TOPOGRAPHIC PLACE. Only the **Name** will be filled for the EPIP. The EIRE_NP allows a Qualifier also to be specified for the elements.
«cntd»	***alternative-Descriptors***	*TopographicPlaceDescriptor*	0:*	Alternative descriptors for TOPOGRAPHIC PLACE. Can be used to provide translations of the fully qualified name
«enum»	***TopographicPlace-Type***	*TopographicTypeEnum*	0:1	Type of a TOPOGRAPHIC PLACE.
	PlaceCentre	*xsd:boolean*	0:1	Whether TOPOGRAPHIC PLACE lies at centre of the settlement.
	PostCode	*xsd:normalizedString*	0:1	Postcode or partial postcode associated with the area.
«FK»	***CountryRef***	*CountryRef*	0:1	Reference to COUNTRY of a TOPOGRAPHIC PLACE.
«cntd»	***otherCountries***	*CountryRef*	0:*	For TOPOGRAPHIC PLACEs that span borders, references to additional COUNTRY or COUNTRies that place lies in.
«FK»	***ParentTopographic-PlaceRef***	*TopographicPlaceRef*	0:1	Reference to a parent TOPOGRAPHIC PLACE that contains the TOPOGRAPHIC PLACE.
«cntd»	***adjacentPlaces***	*TopographicPlaceRef*	0:*	References to other TOPOGRAPHIC PLACEs that are immediately adjacent the TOPOGRAPHIC PLACE.

«cntd»	***containedIn***	*TopographicPlaceRef*	0:*	References to other TOPOGRAPHIC PLACEs within which the TOPOGRAPHIC PLACE is contained.
«cntd»	***accesses***	*AccessRef*	0:*	References to ACCESS links that relate to the TOPOGRAPHIC PLACE.

9.4.2 TopographicPlaceDescriptor (Subcomponent)

A TOPOGRAPHICAL PLACE must always have a name. It is not unusual for there to be several places with the same name in a country (for example, there are twelve settlements called "Hausen" in Germany, and eight "Newports"in the UK, etc.), or in different countries (there are also several "Hausen" in Switzerland and even "Paris, Texas").

In order to distinguish between the different instances in a consistent way, a qualifier name may be specified for a TOPOGRAPHIC PLACE using a ***TopographicPlaceDescriptor*** element (E.g., "Newtown, Westmeath", "Newtown, Louth", etc.). The qualifier is only needed in certain presentation contexts, so is held separately (and may be derived from the name of another place). So, for example a user in "Newtown, Westmeath" looking up stop departures would simply "Newtown", (since the context is already established) , while a user looking for "Newtown" in a Journey Planner for the whole of Ireland (or wider), will see the fully qualified name so as to distinguish it from the other homonyms.

If a full translation of the qualified name into other languages name is needed (as is the case in particular in areas where the language of place names is a political issue), alternative ***TopographicPlaceDescriptor*** instances can be specified (thus a ***TopographicPlaceDescriptor*** is in effect a special type of ALTERNATIVE NAME for TOPOGRAPHIC PLACEs).

Table 52 — *TopographicPlaceDescriptor – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	TOPOGRAPHIC PLACE DESCRIPTOR inherits from VERSIONED CHILD.
«PK»	***id***	*TopographicalPlace-IdDescriptorType*	1:1	Identifier of a TOPOGRAPHIC PLACE DESCRIPTOR. Only needed for alternative descriptors.
	Name	*MultilingualString*	1:1	Name of a TOPOGRAPHIC PLACE DESCRIPTOR.
	ShortName	*MultilingualString*	0:1	Short Name of a TOPOGRAPHIC PLACE DESCRIPTOR.
	QualifierName	*MultilingualString*	1:1	Name used to distinguish TOPOGRAPHIC PLACE from other similar named TOPOGRAPHIC PLACEs. This should not be included in the name but may be added by applications in context. The ***Qualifier*** should be in the same language as the ***Name***.
«FK»	***TopographicPlaceRef***	*TopographicPlaceRef*	0:1	Reference to another TOPOGRAPHIC PLACE that qualifies the name of the TOPOGRAPHIC PLACE.

9.4.3 Country

identifier. In the EIRE_NP, the countries do not need to be declared, but may be referenced as external references. For example,

```
              <CountryRef ref="ie" />
              <CountryRef ref="uk" refPrincipality="iso3166-2:GB-NIR">Northern
Ireland</CountryRef>
```

Table 53 — Table 9 — *Country – Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Place*	::>	COUNTRY inherits from PLACE.
«PK»	***id***	*CountryCodeIdType*	1:1	Identifier of COUNTRY.
«cntd»	***uicCode***	*PrivateCodeStructure*	01	UIC Code for Country.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative Names for Country.

Country Ids are implemented by an enumeration of allowed values.

NB☞ The ***Country*** element can be used to distinguish Ireland, England, Wales Scotland and Northern Ireland (Corresponding to the NPTG fixed codes). See later below.

9.5 Stop Places

Stop Places are made up of QUAYs and STOP PLACE ENTRANCEs, each of which may be classified.

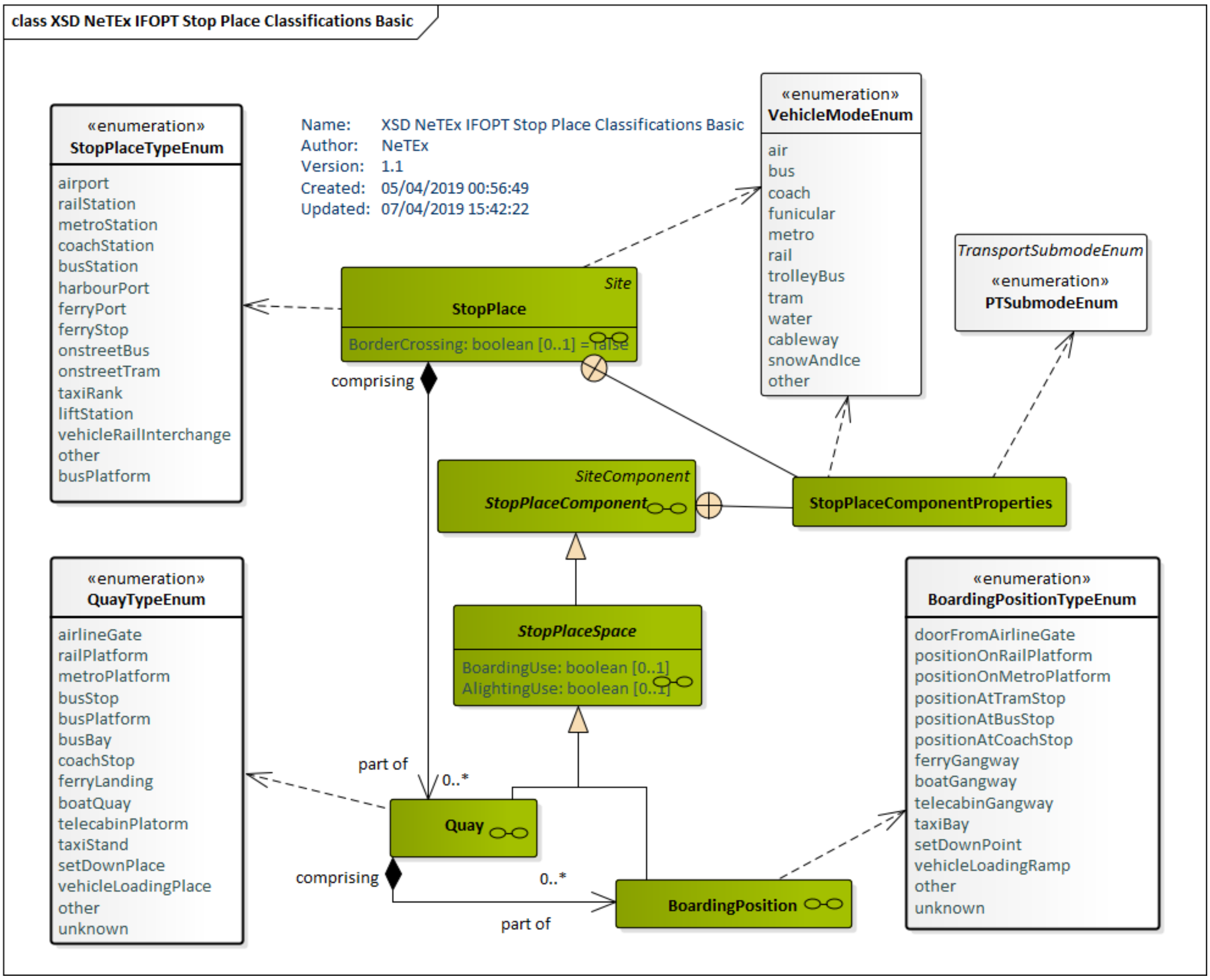


Figure 16 — Stop Place Classifications (UML)

9.5.1 StopPlace

A STOP PLACE, as discussed above, represents a station, stop, port, airport or any physical point of access to the transport network.

Table 54 — *StopPlace – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*Site*	*::>*	STOP PLACE inherits from SITE.
«PK»	***id***	*StopPlaceIdType*	1:1	Identifier of a STOP PLACE.
«AK»	***PublicCode***	*StopPlaceCodeType*	0:1	Code used to identify a STOP PLACE to the public as an alternative to a name.
	STOP PLACE COMPONENT PROPERTY GROUP (Common to all STOP PLACE COMPONENTs)			
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	The main TRANSPORT MODE of the STOP PLACE. • *air, bus, trolleyBus, coach, funicular, metro, tram, cableway, Rail, water, ferry, lift, other*
«enum»	***submode***	*TransportSubmodeEnum*	0:1	Submode enumeration associated with MODE.
«enum»	***OtherTransportModes***	*VehicleModeEnum*	0:*	The other TRANSPORT MODE available at the STOP PLACE.
«cntd»	***tariffZones***	*TariffZoneRef*	0:*	The TARIFF ZONEs associated with the STOP PLACE.
	STOP PLACE PROPERTY GROUP (Specific to STOP PLACE)			
«enum»	***StopPlaceType***	*StopPlaceTypeEnum*	1 :1	The type of the STOP PLACE. • *onstreetBus, onstreetTram, airport, railStation, metroStation, coachStation, ferryPort, harbourPort, ferryStop, liftStation, tramStation, vehicleRailInterchange, other*
	BorderCrossing	*xsd:boolean*	0:1	Whether STOP PLACE is a border crossing.
«cntd»	***unlocalisedEquipment***	*EquipmentRef*	0:*	Equipment available at STOP PLACEs but not specifically located.
	STOP PLACE TOPOGRAPHIC GROUP			
«cntd»	***servedPlaces***	*AccessRef*	0:*	TOPOGRAPHICAL PLACEs that the STOP PLACE STOP PLACEs.
«cntd»	***mainTerminusForPlaces***	*TopographicalPlaceRef*	0:*	TOPOGRAPHICAL PLACEs for which the STOP PLACE is a main terminus. Only certain stations will be deemed the main STOP PLACEs points. For example, London has many rail stations but only some are main line termini. Geographic containment is not necessarily implied. For example, London Gatwick and London Stansted airports are not in London, but are designated airports for London. Norwich station is not in Norwich, etc.
«enum»	***LimitedUse***	*LimitedUseEnum*		Categorisation of the stop as having topographic limitations which may affect its use in journey planners
«enum»	***Weighting***	*InterchangeUseEnum*	0:1	Default relative weighting to be used for stop place. Possible values are: • *noInterchange* • *interchangeAllowed* • *recommendedInterchange*

				- *preferredInterchange* (indicating that the StopPlace is specially designed to ease interchange, with path guidance paths, easy, secured and short walking path, etc.).
«enum»	***StopPlaceWeight***	*StopPlaceWeightEnum*	0:1	Type of expected INTERCHANGE at a STOP PLACE. For use in journey planner. +v1.1 - *international, national, regional, local*
	STOP PLACE PASSENGER GROUP			
«cntd»	***quays***	*Quay*	0:*	The QUAYs contained in the STOP PLACE, that is platforms, jetties, bays, taxi ranks, and other points of physical access to VEHICLEs.
«cntd»	***accessSpaces***	*AccessSpace*	0:*	ACCESS SPACEs within the STOP PLACE, i.e., STOP PLACE COMPONENTs that are not QUAYs, BOARDING POSITIONs or ENTRANCEs.
XGRP	***SiteAccessGroup***	**xmlGroup**	0:1	Common elements relating to SITE ACCESS; PATH LINKs, PATH JUNCTIONs, ACCESS LINKs, NAVIGATION PATHs
XGRP	***vehicleStoppingPlaces***	**xmlGroup**	0:*	VEHICLE STOPPING PLACEs within the STOP PLACE.

9.5.2 SiteComponent (Abstract)

A SITE COMPONENTs is a component of a SITE describing a part of its structure. SITE COMPONENTs share common properties for EQUIPMENT, management, accessibility and other features.

None of its properties, except ***LevelRef***, are used in EPIP.

NB☞ The EIRE_NP allows the use of EQUIPMENT to specify NaPTAN 2.5 equipment options.

Table 55 — *SiteComponent – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	*::>*	*SiteElement*	*::>*	SITE COMPONENT inherits from SITE ELEMENT.
«PK»	***id***	*SiteComponentIdType*	1:1	Identifier of SITE COMPONENT.
«FK»	***SiteRef***	*SiteRef*	0:1	Reference to parent of SITE COMPONENT.
«FK»	***LevelRef***	*LevelRef*	0:1	Reference to a LEVEL associated with the SITE COMPONENT.
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	Reference to CLASS OF USE associated with the SITE COMPONENT.
«cntd»	***checkConstraints***	*CheckConstraint*	0:*	CHECK CONSTRAINTs associated with SITE COMPONENT.
«cntd»	***equipmentPlaces***	*EquipmentPlace*	0:*	EQUIPMENT PLACEs associated with STOP PLACE COMPONENT.
«cntd»	***equipments***	*Equipment*	0:*	EQUIPMENT associated with SITE COMPONENT.
«cntd»	***localServices***	*LocalService*	0:*	LOCAL SERVICEs associated with SITE COMPONENT.

9.5.3 StopPlaceSpace (Abstract)

A STOP PLACE SPACE is an abstract component specifying properties common to all STOP PLACE COMPONENTs; it is specialised as a QUAY, or a STOP PLACE ENTRANCE.

Table 56 — *StopPlaceSpace – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	*::>*	*SiteComponent*	*::>*	STOP PLACE SPACE inherits from SITE COMPONENT.
	BoardingUse	*xsd:boolean*	0:1	Whether Passengers may use the component for Boarding vehicle transport.
	AlightingUse	*xsd:boolean*	0:1	Whether Passengers may use the component when alighting from vehicle transport.
	Label	*xsd:normalizedString*	0:1	Local Label given to Component, e.g., a Point Letter on a stop.
«cntd»	***entrances***	*StopPlaceEntrance*	0:*	ENTRANCEs to STOP PLACE SPACE.

9.5.4 Quay (Subcomponent of StopPlace)

A QUAY is a place such as a platform, bus stop, stance or quay side where passengers have access to PT vehicles, taxi, cars or other means of transportation. A QUAY may serve one or more VEHICLE STOPPING PLACEs and be associated with one or more STOP POINTs.

A QUAY may contain other sub QUAYs. A child QUAY must be physically contained within its parent QUAY. This allows one to represent composite platforms with two or more sides or named sections. One can thus journey plan to any level of detail.

Table 57 — *Quay – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*StopPlaceSpace*	*::>*	QUAY inherits from STOP PLACE SPACE.
«PK»	***id***	*QuayIdType*	1:1	Identifier of QUAY.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Code use to identify QUAY to the public.
«AK»	***PlateCode***	*xsd:normalizedString*	0:1	Asset Code use to identify QUAY.
«AK»	***ShortCode***	*xsd:normalizedString*	0:1	Short Code use to identify QUAY for near-band wireless.
«cntd»	***destinations***	*DestinationDisplayRef*	0:*	Destinations associated with one QUAY.
	CompassBearing	*CompassBearingType*	0:1	Bearing of street relative to QUAY in degrees (absolute bearing of front part of Vehicle when stopped at Quay).
«enum»	***CompassOctant***	*CompassOctantEnum*	0:1	Bearing of street relative to QUAY in compass quadrant.
«enum»	***QuayType***	*QuayTypeEnum*	0:1	Classification of QUAY.
«FK»	***ParentQuayRef***	*QuayRef*	0:1	Reference to parent of QUAY that wholly contains it.
«cntd»	***boardingPositions***	*BoardingPosition*	0:*	BOARDING POSITIONs within QUAY.

9.5.5 Entrance (Abstract)

An ENTRANCE describes any physical entrance or exit to or from a SITE. May be a door, barrier, gate or other recognisable point of access. It is specialised, for example, for STOP PLACE ENTRANCE, or POINT OF INTEREST ENTRANCE.

Table 58 — *Entrance – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description

::>	::>	SiteComponent	::>	ENTRANCE inherits from SITE COMPONENT.
«PK»	***id***	*EntranceIdType*	1:1	Identifier of ENTRANCE.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public code for ENTRANCE.
	Label	*xsd:normalizedString*	0:1	Label of ENTRANCE.
«enum»	***EntranceType***	*EntranceTypeEnum*	0:1	Type of ENTRANCE: • *opening* • *openDoor* • *door* • *swingDoor* • *revolvingDoor* • *automaticDoor* • *ticketBarrier* • *gate* • *other*
	IsExternal	*xsd:boolean*	0:1	Whether ENTRANCE opens externally.
	IsEntry	*xsd:boolean*	0:1	Whether ENTRANCE can be used for entry to SITE.
	IsExit	*xsd:boolean*	0:1	Whether ENTRANCE can be used for exit from SITE.
	Width	*LengthType*	0:1	Width of ENTRANCE.
	Height	*LengthType*	0:1	Height of ENTRANCE.
	DroppedKerbOutside	*xsd:boolean*	0:1	Whether there is a dropped curb outside ENTRANCE.
	DropOffPointClose	*xsd:boolean*	0:1	Whether ENTRANCE is near drop-off point.

9.5.6 StopPlaceEntrance (Subcomponent of StopPlace)

The STOP PLACE ENTRANCE is a physical entrance or exit to/from a STOP PLACE for a passenger.

They are relevant for journey planning to large transport interchanges.

The ENTRANCE will typically be a 'Logical entrance', that is an entrance to the building to which a Journey Planner can navigate. It may actually comprise several physical doorways – e.g., a revolving doorway with side doors. In NeTEx , the nature of the physical doorways and their accessibility can be specified using ENTRANCE EQUIPMENT elements – this is out of scope of the EIRE_NP.

NB☞ A Mapping of NaPTAN includes ***StopPoints*** which are STOP PLACE ENTRANCEs. (Only NaPTAN Main ENTRANCEs are given; QUAY and BOARDING POSITION ENTRANCE are not populated.)

Table 59 — *StopPlaceEntrance – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	SiteEntrance	::>	STOP PLACE ENTRANCE. inherits from SITE ENTRANCE.
«PK»	***id***	*StopPlaceEntranceIdType*	1:1	Identifier of STOP PLACE ENTRANCE.
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	Identifier of Primary TRANSPORT MODE of the group.
«enum»	***TransportSubmode***	*SubmodeEnum*	0:1	Identifier of TRANSPORT SUB MODE of the group.
«enum»	***OtherTransport-Modes***	*VehicleModeEnum*	0:*	Public transport MODEs which may be accessed through associated place.

«cntd»	***tariffZones***	*TariffZoneRef*	0:*	TARIFF ZONEs into which component falls.

9.6 Stops for Flexible Services

NeTEx, like NaPTAN, includes support for FTS (Flexible Transport Services) and DRT (Demand Responsive Transport). DRT and FTS often cover similar services; FTS being more generic since flexibility may not be directly linked to the demand, but may be related to particular operating needs or cost optimisations. The term ‘flexible’ is used in NeTEx and the following text to cover both concepts.

NB☞ Hail and ride or zone based flexible services which serve a section of road or an area can be represented in *NeTEx* using FLEXIBLE STOP PLACEs and FLEXIBLE QUAYs. This allows the sections of the road or other zones to appear as named “stops” in their own right in a journey planner. Note that flexible services can also use regular STOP PLACEs and QUAYs.

Note that The EPIP only covers flexible information at LINE and STOP PLACE level. This is mainly done through the ***BookingArrangements*** and the ***FlexibleLineType***. Services can be marked as FLEXIBLE SERVICES and the use of POINTs IN PATTERN can be marked as flexible. See ‘flexible services’ later

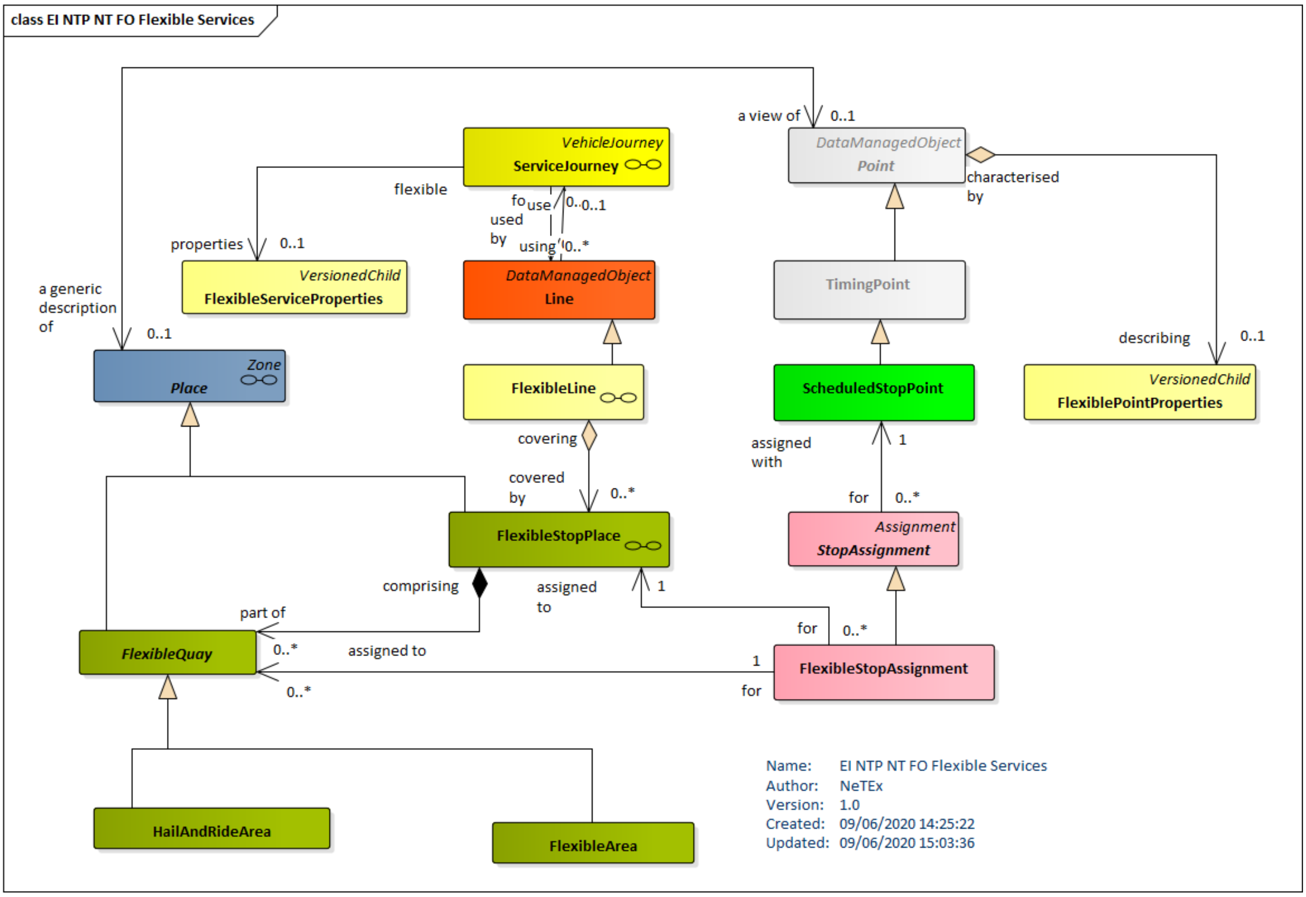


Figure 17 — Flexible StopPlace – Physical Model (UML)

9.6.1 FlexibleStopPlace – Model Element

A type of STOP PLACE for FLEXIBLE services, comprising one or more flexible zones (FLEXIBLE QUAYS) where vehicles may stop and where passengers may board or leave vehicles.

Table 60 — Table 10 – *FlexibleStopPlace* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*Place*	::>	FLEXIBLE STOP PLACE inherits from PLACE.
«PK»	***id***	*FlexibleStopPlaceIdType*	1:1	Identifier of a FLEXIBLE STOP PLACE.
	NameSuffix	*MultilingualString*	1:1	Suffix to use on Name.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for SITE ELEMENT.
«enum»	***TransportMode***	*VehicleModeEnum*	1:1	The main TRANSPORT MODE of the FLEXIBLE STOP PLACE. See TRANSPORT MODE for allowed values.
«cntd»	***Locale***	*Locale*	0:1	Locale setting time zone, default language etc., for the FLEXIBLE STOP PLACE. See Reusable components.
«AK»	***PublicCode***	*PublicCodeStructure*	0:1	Code used to identify a FLEXIBLE STOP PLACE to the public as an alternative to a name.
«cntd»	***areas***	*FlexibleQuay*	0:*	FLEXIBLE QUAYs of a FLEXIBLE STOP PLACE.

9.6.2 FlexibleQuay (Abstract))

A physical ZONE such as a section of a road where a flexible service is available on demand. The existence of the zone makes the services visible to journey planners looking for available services for an area.

Table 61 — Table 11 – *FlexibleQuay* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*Place*	::>	FLEXIBLE QUAY inherits from PLACE.
«PK»	***id***	*FlexibleQuayIdType*	1:1	Identifier of a FLEXIBLE QUAY.
	NameSuffix	*MultilingualString*	1:1	Suffix to use on Name.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for SITE ELEMENT.
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	Main TRANSPORT MODE of FLEXIBLE STOP PLACE. If not specified, same as parent STOP PLACE.
	BoardingUse	*xsd:boolean*	1:1	Whether Passengers may use the FLEXIBLE QUAY for Boarding vehicle transport.
	AlightingUse	*xsd:boolean*	1:1	Whether Passengers may use the FLEXIBLE QUAY for Alighting from vehicle transport.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public code of a FLEXIBLE STOP PLACE.
«FK»	***FlexibleStopPlaceRef***	*FlexibleStopPlaceRef*	0:1	Reference to a parent FLEXIBLE STOP PLACE.

9.6.3 FlexibleArea (Subcomponent of FlexibleStopPlace)

A FLEXIBLE AREA is the physical ZONE (i.e., 2-dimensional FLEXIBLE QUAY) within which a flexible service is available for pickup or drop off. It allows the projection of the zone onto a map.

Table 62 — Table 12 – *FlexibleArea* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*FlexibleQuay*	::>	FLEXIBLE AREA inherits from FLEXIBLE QUAY.
«PK»	***id***	*FlexibleAreaIdType*	1:1	Identifier of a Flexible AREA.

9.6.4 HailAndRideArea (Subcomponent of FlexibleStopPlace)

A physical section of road between a start and end point within which a Hail and Ride service is available (i.e., in effect a 1-dimensional FLEXIBLE QUAY).

Table 63 — Table 13 – *HailAndRideArea* – Element

Classifica tion	Name	Type	Cardinali ty	Description
::>	::>	*FlexibleQuay*	::>	HAIL AND RIDE AREA inherits from FLEXIBLE QUAY.
«PK»	***id***	*HailAndRideAreaIdType*	1:1	Identifier of a HAIL AND RIDE AREA.
	CompassBearing	*CompassBearingType*	0:1	Compass Bearing associated with a HAIL AND RIDE AREA in degrees
«enum»	***CompassOctant***	*CompassOctantEnum*	0:1	Compass Octant associated with a HAIL AND RIDE AREA (N, S, E W etc.). See utility types for allowed values.
«FK»	***StartPointRef***	*(PointRef)*	1:1	Start Point on road of the HAIL AND RIDE AREA.
«FK»	***EndPointRef***	(*PointRef*)	1:1	End Point on road of the HAIL AND RIDE AREA.

9.6.5 FlexiblePointProperties (Subcomponent – Multipurpose)

The set of characteristics describing the possible flexibility of a POINT.

It can mark a STOP POINT I JOURNEY PATTERN as having a flexible service.

In the EPIP, ***FlexiblePointProperties*** should be embedded within the ***StopPointInPattern*** which they describe.

Table 64 — Table 14 – *FlexiblePointProperties – XML Element*

Classifica tion	Name		Type	Cardin ality	Description
::>	::>		*VersionedChild*	::>	FLEXIBLE POINT PROPERTIES inherits from VERSIONED CHILD.
«PK»	***id***		*FlexiblePointPropertiesIdType*	1:1	Identifier of a FLEXIBLE POINT PROPERTIES.
			CHOICE	1:1	Element for which these are the flexible properties.
«FK»	***a***	***PointOnRouteRef***	*PointOnRouteRef*	0:1	POINT ON ROUTE for which these are the properties.
«FK»	***b***	***PointRef***	*(PointRef)*	0:1	POINT for which these are the properties.
	MayBeSkipped		*xsd:boolean*	0:1	Whether the POINT may be skipped.
	OnMainRoute		*xsd:boolean*	0:1	Whether the POINT is on the main ROUTE.
	PointStandingFor-AZone		*xsd:boolean*	0:1	Whether the POINT represents a FLEXIBLE ZONE.
	ZoneContainingStops		*xsd:boolean*	0:1	Whether the ZONE is defined by a GROUP of STOP POINTs (value =*'true'*) or is a geographical zone defined by its boundary.

9.7 GroupOfStopPlaces

A grouping of STOP PLACEs which will be commonly referenced for a specific purpose (stops inside an airport, main stops of a city, etc.).

Table 65 — *GroupOfStopPlaces – XML Element*

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*GroupOfEntities*	::>	***GroupOfStopPlaces*** inherits from ***GroupOfEntities***
«PK»	***id***	*GroupOfStopPlacesIdType*	1:1	Identifier of a GROUP of STOP PLACEs.
«cntd»	***members***	*StopPlaceRef*	0:*	Member STOP PLACEs of a GROUP of STOP PLACEs.
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	Identifier of Primary TRANSPORT MODE of the group. See STOP PLACE for allowed values.
«enum»	***TransportSubmode***	*SubmodeEnum*	0:1	Identifier of TRANSPORT SUB MODE of the group.

9.8 PointsOfInterest

9.8.1 Point of Interest

A POINT OF INTEREST (POI) is a type of PLACE to or through which passengers may wish to navigate as part of their journey and which is modelled in detail by journey planners.

POINTs OF INTEREST are required by the *EC Priority action A of the ITS Directive*. It is not expected that POINTS OF INTEREST will be provided in NeTEx datasets; they will in most cases be defined in external databases (INSPIRE, OSM, SPOI, or commercial datasets) and be referenced by SITE CONNECTIONs.

NB☞ The NaPTAN 2.5 Schema allows the definition of points of interest, but this has not been populated.

Table 66 — *PointOfInterest – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Site*	::>	POINT OF INTEREST inherits from SITE.
«PK»	***id***	*PointOfInterestIdType*	1:1	Identifier of: POINT OF INTEREST.
«cntd»	***classifications***	*PointOfInterest-ClassificationRef | PointOfInterestClassificationView*	0:*	Classification of the POINT OF INTEREST. In the EPIP, only the ***Name*** attribute of the ***PointOfInterestClassificationView*** will be used for this classification. No standard classification is predefined either by NeTEx, nor by the EPIP.
«cntd»	***spaces***	*PointOfInterestSpace*	0:*	POINT OF INTEREST SPACEs within the POINT OF INTEREST.
«cntd»	***nearTopographic-Places***	*TopographicPlaceRef*	0:*	TOPOGRAPHIC PLACEs near the POINT OF INTEREST.
XGRP	***SiteAccessGroup***	***xmlGroup***	0:1	ACCESS SPACEs within the STOP PLACE, i.e. STOP PLACE COMPONENTs that are not QUAYs, BOARDING POSITIONS, or ENTRANCEs.

9.8.2 PointOfInterestEntrance (Subcomponent of PointOfInterest)

A physical entrance or exit to/from a POINT OF INTEREST for passengers not in a vehicle. May be a door, barrier, gate or other recognizable point of access.

NB☞ POINT OF INTEREST ENTRANCEs are not in EPIP but are allowed in NaPTAN v2.5 so are included in the EIRE_NP. They are relevant for large POINTs OF INTEREST such as parks and stadiums, where journey planning to the centroid of the POINT OF INTEREST is insufficient or inaccurate.

Table 67 — Table 15 – *PointOfInterestEntrance* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*SiteEntrance*	::>	POINT OF INTEREST ENTRANCE inherits from SITE ENTRANCE.
«PK»	***id***	*PointOfInterestEntranceId*	1:1	Identifier of a POINT OF INTEREST ENTRANCE.
«FK»	***ParentRef***	*PointOfInterestRef*	0:1	Parent POINT OF INTEREST of ENTRANCE.

9.9 TariffZone

A ZONE used to define a zonal fare structure.

The relationship 'TARIFF ZONE – *members* - SCHEDULED STOP POINT' is navigable in both directions, so in NeTEx the references that implement may be held by the elements on either or both ends (i.e., on ***TariffZone*** or ***ScheduledStopPoint***).

In the EPIP and the EIRE_NP (as in NaPTAN) the reference should be on the ***ScheduledStopPoint*** (i.e., ***ScheduledStopPoint*** X should hold a reference to ***TariffZone*** Y).

Table 68 — Table 16 — *TariffZone – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Zone*	::>	TARIFF ZONE inherits from ZONE.
	id	*TariffZoneIdType*	1:1	Identifier of TARIFF ZONE.
«cntd»	***Presentation***	*Presentation*	0:1	Presentation features such as colour associated with TARIFF ZONE. +V1.1
«cntd»	***PrintedPresentation***	*PrintedPresentation*	0:1	Printed Presentation features such as colour associated with TARIFF ZONE. +V1.1

9.10 SiteFrame

A SITE FRAME is used to group the elements of the Stop Model.

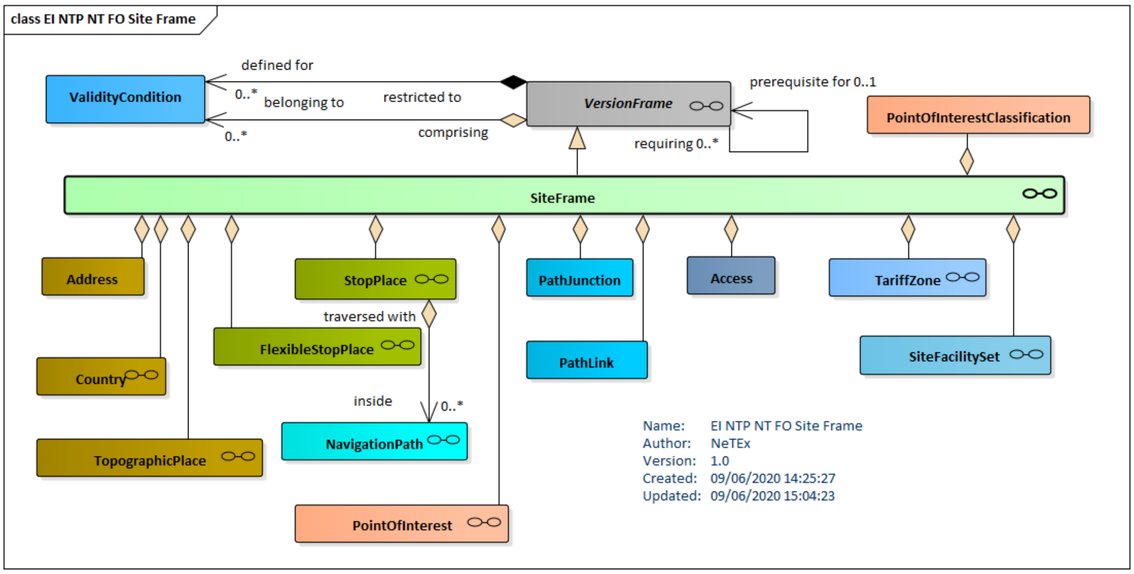


Figure 18 — Site Frame (UML)

10 Network and Timetable Profile

The Network and Timetable part of EIRE_NP (EIRE_NP-NT) specifies the data elements needed to describe a passenger timetable. The description is broken down into separate modules:

- A description of the network topology (lines, routes etc).
- A description of the possible patterns of use (stop points, journey patterns, service patterns etc).
- A description of the actual journeys and their times.
- A description of the day types and calendar controlling what types of journey are run on each day.

NB☞ The Network and Timetable Profile corresponds to different data sets for different modes. Bus and tram data is available in GTFS, while rail data is available in HaFas format..

10.1 Basic Network topology

The topology describes the permanent lines and routes of the transport Network.

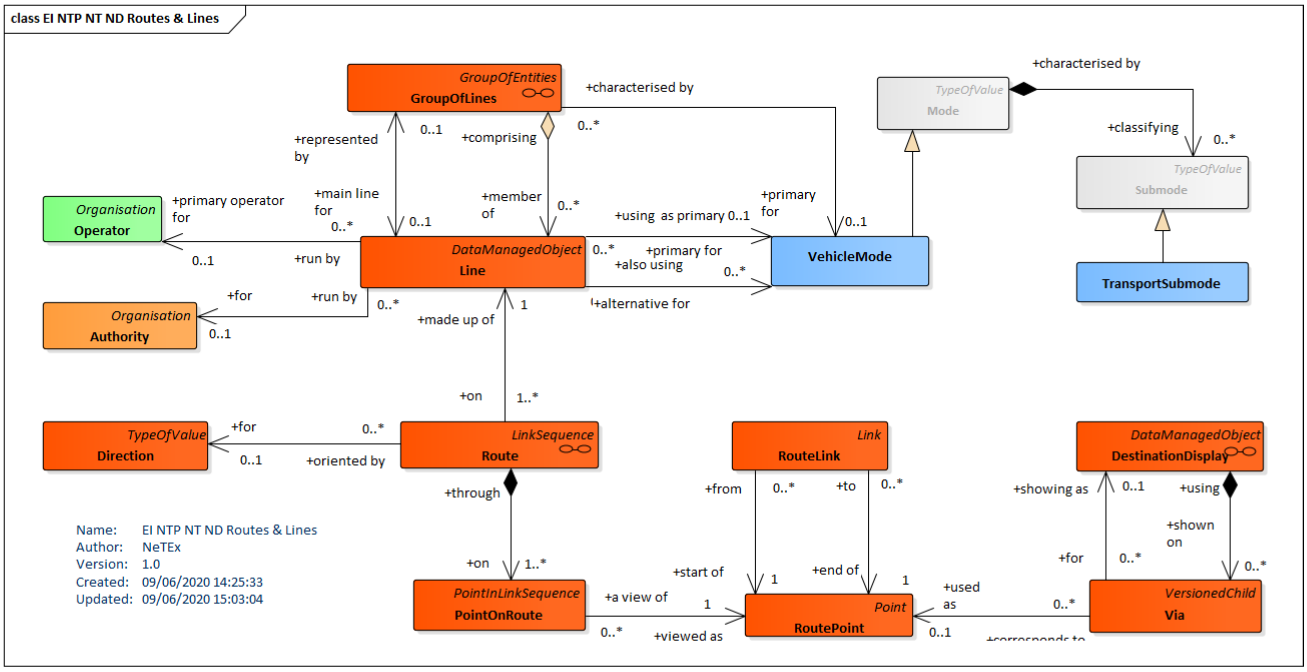


Figure 19 — Network Topology (UML)

Transmodel defines a LINE as a grouping of ROUTEs that is generally known to the public by a similar name or number. These ROUTEs are usually very similar to each other from a topological point of view, being variants of a core route with some deviations on certain parts only. Two ROUTEs using the same infrastructure path (or parallel tracks), but with opposite DIRECTIONs, will generally belong to the same LINE.

A LINE is associated with a main TRANSPORT MODE and SUBMODE but may also have secondary modes (for example a train line, which is operated by bus at certain times of the day or under certain circumstances).

A LINE is also associated with a primary OPERATOR or AUTHORITY; multiple secondary operators are also allowed.

LINEs may be grouped into GROUPs OF LINEs for particular purposes, such as fare harmonisation, day type assignment, or to group some kind of service categories (night buses, etc.).

10.1.1 Line

A grouping of ROUTEs that is generally known to the public by a similar name or number.

Table 69 — *Line – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	LINE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*LineIdType*	1:1	Identifier of LINE.
	Name	*MultilingualString*	1:1	Name of LINE.
	ShortName	*MultilingualString*	0:1	Short Name of LINE.
	Description	*MultilingualString*	0:1	Description of LINE.
«enum»	***TransportMode***	*AllVehicleModesOfTransportEnum*	0:1	Identifier of Primary TRANSPORT MODE of LINE. See STOP PLACE for allowed values.
«enum»	***TransportSubmode***	*TransportSubmodeEnum*	0:1	Submode of the TRANSPORT MODE of the LINE.
	Url	*any*	0:1	A URL associated with the LINE.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public identifier of LINE.
«AK»	***PrivateCode***	*xsd:normalizedString*	0:1	Alternative identifier of LINE.
«AK»	***ExternalLineRef***	*ExternalObjectRef*	0:1	An alternative code that uniquely identifies the LINE specifically for use in AVMS systems. For VDV compatibility.
		CHOICE		A TRANSPORT ORGANISATION of some sort.
«FK»	a **OperatorRef**	*OperatorRef*	0:1	Reference to OPERATOR of LINE.
«FK»	b ***AuthorityRef***	*AuthorityRef*	0:1	Reference to AUTHORITY of LINE.
«cntd»	***additionalOperators***	*OperatorRef*	0:*	References to additional TRANSPORT ORGANISATIONs for LINE. (May be OPERATORs or AUTHORITies).
«cntd»	***otherModes***	*VehicleModeEnum*	0:*	Additional transport MODEs for LINE.
«FK»	***OperationalContextRef***	*OperationalContextRef*	0:1	Reference to OPERATIONAL CONTEXT of LINE.
«enum»	***LineType***	*LineTypeEnum*	0:1	Classification of LINE.
«FK»	***TypeOfLineRef***	*TypeOfLineRef*	0:1	Reference to a TYPE of LINE.
«EK»	***ExternalProductCategoryRef***	*ExternalObjectRef*	0:1	An external product classification for all journeys of the LINE for use in AVMS systems. For VDV compatibility.
«FK»	***TypeOfProductCategoryRef***	*TypeOfProductCategoryRef*	0:1	Reference to a TYPE of PRODUCT CATEGORY. +v1.1
«FK»	***TypeOfServiceRef***	*TypeOfServiceRef*	0:1	Reference to a TYPE of SERVICE. +v1.1
	Monitored	*xsd:boolean*	0:1	Whether real-time data is normally available for LINE.
«cntd»	***routes***	*RouteRef*	0:*	Routes that follow the LINE.
«FK»	***RepresentByGroupRef***	*GroupOfLinesRef*	0:1	Identifier of GROUP OF LINES that should be used to represent this LINE.
«cntd»	***Presentation***	*Presentation*	0:1	Preferred presentation attributes to use when rendering LINE on maps, etc.

«cntd»	***Alternative-Presentation***	*Presentation*	0:1	Alternative presentation attributes to use when rendering LINE on maps, etc.
«cntd»	***PrintedPresentation***	*PrintedPresentation*	0:1	Preferred presentation attributes to use when rendering LINE on maps, etc. for printing. +v1.1
«enum»	***PaymentMethods***	*PaymentMethodEnum*	0:*	PAYMENT METHODs accepted for this LINE. See Service Restriction Model for allowed values. +v1.1
«cntd»	***typesOfPayment-Method***	*TypeOfPaymentMethod*	0:*	TYPEs OF PAYMENT METHOD accepted for this LINE. +v1.1.
«enum»	***PurchaseMoments***	*PurchaseMomentEnum*	0:*	LINE. See Service Restriction Model for allowed values. +v1.1
«cntd»	***ContactDetails***	*ContactDetails*	0:1	Contact details for OPERATOR of the LINE. (May be specific to the line)
«cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	The overall accessibility characteristics of the LINE.
«cntd»	***allowedDirections***	*AllowedDirection*	0:*	Directions allowed for this LINE.
«cntd»	***noticeAssignments***	*NoticeAssignment*	0:*	NOTICES assigned to LINE.
«cntd»	***documentLinks***	*InfoLinks*	0:*	Timetable documents associated with the LINE, e.g., pdf files +v1.1

10.1.2 AllowedLineDirection (Subcomponent of Line)

An allowed DIRECTION that can be used on a given ROUTE.

This can be used to associate particular DIRECTIONs with particular LINEs and to validate the selection of allowed values.

Table 70 — Table 17 – *AllowedLineDirection – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*DataManagedObject*	*::>*	ALLOWED LINE DIRECTION inherits from DATA MANAGED OBJECT.
«PK»	***id***	*AllowedDirectionIdType*	1:1	Identifier of ALLOWED LINE DIRECTION.
«FK»	***LineRef***	*LineRef*	0:1	Reference to a LINE of ALLOWED LINE DIRECTION.
«FK»	***DirectionRef***	*DirectionRef*	1:1	Reference to a DIRECTION of ALLOWED LINE DIRECTION.
«cntd»	***Presentation***	*Presentation*	0:1	Presentation to use for DIRECTION. +v1.1.

10.1.3 GroupOfLines

GROUPs of LINEs can be used to define several kinds of PT networks and sub-networks: what is usually called a 'public transport network' is in fact only a specific GROUP OF LINEs and a LINE may belong to several of them. For example, in Île-de-France, a LINE may belong not only to the Île-de-France *Mobilité* network (the whole Île-de-France network), but also to the *Noctilien* network (night buses) ant to the *PHEBUS* network (Versailles's town bus network).

Table 71 — *GroupOfLines – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*GroupOfEntities*	*::>*	GROUP OF LINEs inherits from GROUP OF ENTITies.
«PK»	***id***	*GroupOfLinesIdType*	1:1	Identifier of a GROUP OF LINEs.

«cntd»	***members***	*LineRef*	0:*	References to members of a GROUP of LINEs.
«FK»	***MainLineRef***	*LineRef*	0:1	Main LINE of a GROUP of LINEs.
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	Primary Transport MODE of GROUP OF LINES. See STOP PLACE for allowed values.

10.1.4 Network

A NETWORK is a named grouping of LINEs under which a Transport network is known.

Table 72 — ***Network – XML Element***

Classification	Name	Type	Cardinality	Description
::>	*::>*	*GroupOfLines*	*::>*	NETWORK inherits from GROUP OF LINEs.
«PK»	***id***	*NetworkIdType*	1:1	Identifier of NETWORK.
	Name	*MultilingualString*	1:1	Name of NETWORK. Note that by convention the name is mandatory (even if, as inherited from GroupOfEntities, it is not formally mandatory)
«FK»	***Transport-OrganisationRef***	*OperatorRef | AuthorityRef*	0:1	Transport organisation in charge of the NETWORK
«cntd»	***groupsOfOperators***	*GroupOfOperators*	0:*	GROUPs OF OPERATORs associated with the Network. May include both inclusion and exclusions. +v1.1
«cntd»	***groupsOfLines***	*GroupOfLines*	0:*	Groups of LINEs in NETWORK. Can be used to partition or classify the lines in a Network.
«cntd»	***tariffZones***	*TariffZone*	0:*	TARIFF ZONEs in NETWORK.

10.1.5 DestinationDisplay

For clarity to the passenger, it is important to use a consistent set of standardised headings for all services across all passenger information media.

A DESTINATION DISPLAY is an advertised destination of a specific LINE or JOURNEY PATTERN, usually displayed on a head sign or at other on-board locations. A reusable data set of such headings is kept to ensure consistency. Each DESTINATION DISPLAY may have one or more associated VIAs.

In EPIP, a DESTINATION DISPLAY can only be associated with a STOP POINT IN JOURNEY PATTERN.

The DESTINATION DISPLAY will often be the name of a stop, but might be an abbreviation, e.g., ‘Hospital’, a general phrase ‘All destinations or a direction e.g., ‘Northbound’.

NB☞ The FXC also allows a DESTINATION DISPLAY on a SERVICE JOURNEY.

Table 73 — ***DestinationDisplay – XML Element***

Classification	Name	Type	Cardinality	Description
::>	*::>*	*DataManagedObject*	*::>*	DESTINATION DISPLAY inherits from DATA MANAGED OBJECT
«PK»	***id***	*DestinationDisplayIdType*	1:1	Identifier of DESTINATION DISPLAY.
	Name	*MultilingualString*	0:1	Name of DESTINATION DISPLAY. Required.
	ShortName	*MultilingualString*	0:1	Short Name of DESTINATION DISPLAY.
	SideText	*MultilingualString*	0:1	Text to display on the side of the vehicle associated with DESTINATION DISPLAY.
	FrontText	*MultilingualString*	0:1	Front of vehicle text associated with DESTINATION DISPLAY.

	DriverDisplayText	*MultilingualString*	0:1	Text to display to DRIVER.
«AK»	***ShortCode***	*xsd:normalizedString*	0:1	Additional short code associated with DESTINATION DISPLAY.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public code (to be presented to the public) associated with DESTINATION DISPLAY.
«AK»	***PrivateCode***	*xsd:normalizedString*	0:1	Private short CODE associated with DESTINATION DISPLAY.
«cntd»	***vias***	*Via*	0:*	Text to show for VIA display.
«cntd»	***variants***	*DestinationDisplayVariant*	0:*	Variant Display texts for different media.

10.1.6 Via

A VIA is an optional part of a DESTINATION DISPLAY that distinguishes different possible paths between an origin and a destination.

Table 74 — *Via – XML Element*

Classification	Name		Type	Cardinality	Description
::>	*::>*		*VersionedChild*	*::>*	VIA inherits from VERSIONED CHILD.
	id		*ViaIdType*	1:1	Identifier of VIA.
«FK»	a	***Destination-DisplayRef***	*DestinationDisplayRef*	0:1	DESTINATION DISPLAY corresponding to VIA description.
	b	***Name***	*MultilingualString*	1:1	Name of VIA.
«FK»	***RoutePointRef***		*RoutePointRef*	0:1	Reference to ROUTE POINT corresponding to VIA name, if any.
«enum»	***ViaType***		*ViaTypeEnum*	0:1	Classification of meaning of VIA.

10.1.7 DestinationDisplayVariant

Sometimes different variants of a given DESTINATION DISPLAY are needed for different media channels (Electronic, printed, audio, mobile app, etc); this can be specified with a DESTINATION DISPLAY VARIANT.

Table 75 — *DestinationDisplayVariant – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*DataManagedObject*	*::>*	DESTINATION DISPLAY VARIANT inherits from DATA MANAGED OBJECT
«PK»	***id***	*DestinationDisplayIdType*	1:1	Identifier of DESTINATION DISPLAY VARIANT.
«FK»	***DestinationDisplayRef***	*DestinationDisplayRef*	0:1	Reference to a DESTINATION DISPLAY. Not needed as all DESTINATION DISPLAY VARIANTs must be embedded in line with a parent DESTINATION DISPLAY.
«enum»	***Destination-DisplayContext***	*Destination-DisplayContext*	0:1	Context in which to use the variant name. Default is ‘*any*’. +v1.1
«enum	***DestinationDisplayVariantMediaType***	*DeliveryVariantTypeEnum*	1:1	Type of DESTINATION DISPLAY VARIANT being: • *printed* • *textToSpeech* • *recordedAnnouncement* • *web* • *mobile* • *other*

	Name	*MultilingualString*	0:1	Name of DESTINATION DISPLAY.
	ShortName	*MultilingualString*	0:1	Short Name of DESTINATION DISPLAY.
	SideText	*MultilingualString*	0:1	Text to display on the side of the vehicle associated with DESTINATION DISPLAY VARIANT.
	FrontText	*MultilingualString*	0:1	Front of vehicle text associated with DESTINATION DISPLAY VARIANT.
	DriverDisplayText	*MultilingualString*	0:1	Text to display to DRIVER.
«cntd»	***vias***	*Via*	0:*	Text to show for VIA display.

10.2 Routes

The ROUTE entity represents a conventional way of describing a path through the physical infrastructure, to be used by regular PT services. A ROUTE is a linear feature composed of points and links specifically defined for that purpose. This sequence of points and links must be built in a way that identifies a path without any ambiguity.

The ROUTE entity represents an abstract concept that has in itself no real operational meaning. Its purpose is to describe a path independently of both the infrastructure pattern (e.g., ROAD ELEMENTs or RAILWAY ELEMENTs) and the operational pattern (e.g., sequence of SCHEDULED STOP POINTs). ROUTE is classically used as an interfacing object between operational planning and infrastructure description. The independence of the ROUTE definition serves to separate the concerns of the different layers allowing a modular exchange of data.

It may happen that the ROUTE description is not available from some OPERATORs (thus, although it physically must exist, data about it may not be available). When this occurs, the ROUTE information in the LINE description can be skipped, but the ROUTE can also be "aligned" with the SERVICE PATTERNs (using the SCHEDULED STOP POINTs as a base for creating ROUTE POINTs, the POINTs IN JOURNEY PATTERN for POINTs ON ROUTE, and the SERVICE LINKs for ROUTE LINKs).

10.2.1 RoutePoint

A ROUTE POINT is a POINT used to define the shape of a ROUTE through the network.

The same point may be used in different ROUTEs using a POINT ON ROUTE.

Table 76 — *RoutePoint – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Point*	::>	ROUTE POINT inherits from POINT.
«PK»	***id***	*RoutePointIdType*	1:1	Identifier of ROUTE POINT.
	ViaFlag	*xsd:boolean*	1:1	Whether ROUTE POINT is flagged as a via point.
	BorderCrossing	*xsd:boolean*	0:1	Whether point is a border crossing (defined for country borders).

10.2.2 RouteLink

A ROUTE LINK is an oriented link between two ROUTE POINTs allowing the definition of a unique path through the network. The same ROUTE LINK may be used in different ROUTEs.

There may be many ROUT LINKs describing the route of single SERVICE LINK.

Table 77 — *RouteLink – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*Link*	*::>*	ROUTE LINK inherits from LINK
«PK»	***id***	*RouteLinkIdType*	1:1	Identifier of ROUTE LINK.
	Distance	*DistanceType*	1:1	Distance of ROUTE LINK. Units are as specified for Frame (default is SI Metres). Inherited from LINK, but, by convention, mandatory in ROUTE LINK.
«FK»	***FromPointRef***	*RoutePointRef*	1:1	Reference to ROUTE POINT at which ROUTE LINK starts.
«FK»	***ToPointRef***	*RoutePointRef*	1:1	Reference to ROUTE POINT at which ROUTE LINK ends.
«FK»	***Operational-ContextRef***	*OperationalContextRef*	0:1	Reference to OPERATIONAL CONTEXT for ROUTE LINK

10.2.3 Route

An ordered list of located POINTs defining one single path through the road (or rail) network. A ROUTE may pass through the same POINT more than once.

Table 78 — *Route – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*LinkSequence*	*::>*	ROUTE inherits from LINK SEQUENCE.
«PK»	***id***	*RouteIdType*	1:1	Identifier of ROUTE.
«FK»	***LineRef***	*LineRef*	0:1	Reference to LINE to which ROUTE belongs.
«enum	***DirectionType***	*TypeOfDirectionEnum*	0:1	Type of Direction of ROUTE • *outbound; inbound; clockwise; anticlockwise*
«FK»	***DirectionRef***	*DirectionRef*	0:1	Reference to DIRECTION of ROUTE.
«cntd»	***pointsInSequence***	*PointOnRoute*	2:*	Points making up ROUTE.
«FK»	***InverseRouteRef***	*RouteRef*	0:1	Reference to the corresponding matching ROUTE in the opposite direction, if any.

10.2.4 PointOnRoute (Subcomponent of Route)

A POINT ON ROUTE indicates the sequence of a ROUTE POINT within a ROUTE. This intermediate object is required to allow a ROUTE to go several times through the same ROUTE POINT.

Table 79 — *PointOnRoute – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*PointInSequence*	*::>*	POINT ON ROUTE inherits from POINT IN SEQUENCE
«PK»	***id***	*PointOnRouteIdType*	1:1	Identifier of POINT ON ROUTE.
«PK»	***(order)***	*xsd:positiveInteger*	1:1	Order of POINT ON ROUTE within ROUTE (Inherited from POINT IN LINK SEQUENCE).
«FK»	***LinkSequenceRef***	*RouteRef*	0:1	Reference to a LINK SEQUENCE. i.e., ROUTE.
«cntd»	***projections***	*Projection*	0:1	Projection for the POINT ON ROUTE (on the infrastructure)

«FK»	***RoutePointRef***	*RoutePointRef*	1:1	Reference to ROUTE POINT that POINT ON ROUTE uses.
«FK»	***OnwardRouteLinkRef***	*RouteLinkRef*	0:1	Optional Reference to onward link to use - can be used to disambiguating where there are multiple links between the same point.

10.3 Service patterns

The SERVICE PATTERN model describes how services work the routes in the network and underpins the description of a timetable as a set of SERVICE JOURNEYs which follow the patterns.

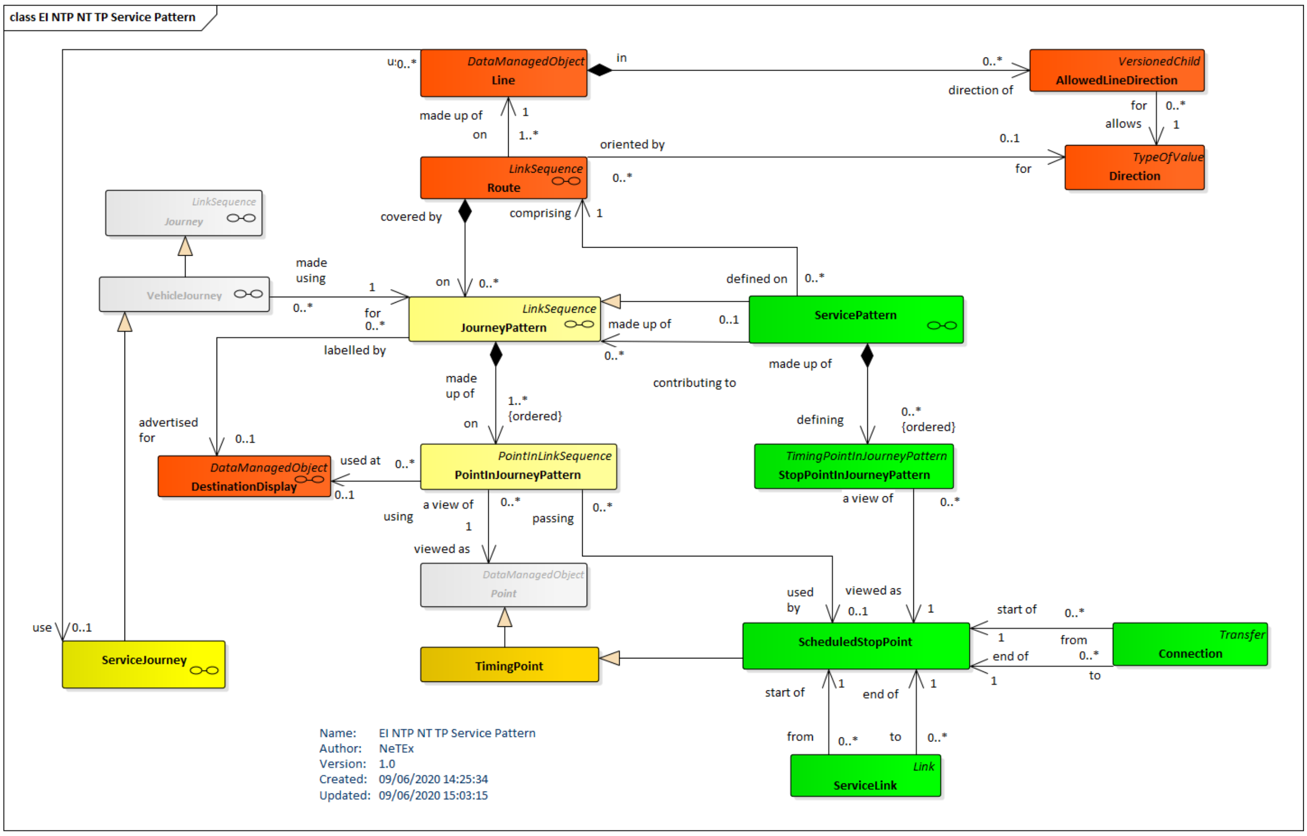


Figure 20 — Service description (UML)

In contrast to ROUTEs and LINEs, which describe the possible paths of vehicles through the road network, JOURNEY PATTERNs describe how the work of vehicles is actually performed. These work patterns describe the sequence of points where vehicles stop, attributed with timing information.

A SERVICE PATTERN is a view of a JOURNEY PATTERN, i.e., the vehicle service from the point of view of the passenger. Passengers are not interested in points relevant only for the scheduling process or for vehicle positioning, but in the SCHEDULED STOP POINTs, i.e., locations where they will be able to board or leave a vehicle.

A SERVICE PATTERN is made up of an ordered sequence of STOP POINTs IN JOURNEY PATTERN, with a possible SERVICE LINK between each pair of consecutive SCHEDULED STOP POINTs. The same SCHEDULED STOP POINT may be visited more than once in the same SERVICE PATTERN; there will be a separate STOP POINT IN JOURNEY PATTERN for each visit with a different ‘order’ attribute.

The possibility of making an interchange between two stops in a transport network is indicated by a CONNECTION.

10.3.1 JourneyPattern (Abstract in EIRE_NP)

The JOURNEY PATTERN model is concerned with the spatial description of services, i.e., spatial aspects of the work of the vehicles. The concerns of this model are different from those of describing the ROUTEs and LINEs. The latter are describing schematic paths of vehicles through the road network, whereas JOURNEY PATTERNs describe how the work of vehicles is performed. These work patterns describe the sequence of points where vehicles stop with specific points being attributed with timing information.

This means that a JOURNEY PATTERN consists of:

— An ordered sequence of SCHEDULED STOP POINTs to be served;

— An ordered sequence of TIMING POINTs at which timing information is scheduled. These TIMING POINTs may also be SCHEDULED STOP POINTs or other POINTs (such other points occur in data sets with timing information and with operational information such as DEAD RUNs). In the EPIP and in the FXC there are no TIMING POINTs that are not SCHEDULED STOP POINTs.

Note: NeTEx allows timing information to be held separately (for example the duration in minutes need to traverse a SERVICE LINK, or the time to wait at a SCHEDULED STOP POINT; this separate data can be used to calculate the actual passing times for many different journeys. In the EPIP only the calculated results are exchanged – as PASSING TIMEs.

NB☞ The EIRE_NP similarly uses only PASSING TIMEs – see discussion of NeTEx features not included by the EIRE_NP in introduction.

The working pattern described by a JOURNEY PATTERN is usually related to a ROUTE it covers. Several JOURNEY PATTERNs may use the same ROUTE (e.g., an express service skipping many stops and a 'slow' service stopping at all stops may both follow the same route).

Table 80 — *JourneyPattern – XML Element* (Abstract in EIRE_NP)

Classification	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	JOURNEY PATTERN inherits from LINK SEQUENCE.
«PK»	***id***	*JourneyPatternIdType*	0:1	Identifier of JOURNEY PATTERN.
«FK»	***RouteRef***	*RouteRef | RouteView*	0:1	Reference to ROUTE that JOURNEY PATTERN follows. Note that if the ROUTE is not available, then the ***RouteRef*** will have to be replaced by a ***RouteView*** (available in the XSD) in which the ***LineRef*** shall be filled in (if not the JOURNEY PATTERN is not connected to any LINE).
«enum»	***DirectionType***	*DirectionTypeEnum*	0:1	Direction type. See DIRECTION above
«FK»	***DirectionRef***	*DirectionRef*	0:1	DIRECTION of JOURNEY PATTERN (often used to distinguish groups of JOURNEY PATTERNs using the same branches (i.e., Routes) of a LINE). DIRECTION is available for the purposes of filtering (for example in a journey planner UI or timetable information system) but is not expected to be provided as descriptive information.

«FK»	***DestinationDisplayRef***	*DestinationDisplayRef*	0:1	DESTINATION DISPLAY associated with JOURNEY PATTERN.
«FK»	***TypeOfJourneyPatternRef***	*TypeOfJourneyPatternRef*	0:1	Type of JOURNEY PATTERN.
«FK»	***OperationalContextRef***	*OperationalContextRef*	0:1	OPERATIONAL CONTEXT associated with JOURNEY PATTERN
«FK»	***TimingPatternRef***	*TimingPatternRef*	0:1	Reference to a TIMING PATTERN.
«cntd»	***runTimes***	*JourneyPatternRunTime*	0:*	RUN TIMEs in JOURNEY PATTERN.
«cntd»	***waitTimes***	*JourneyPatternWaitTime*	0:*	WAIT TIMEs in JOURNEY PATTERN.
«cntd»	***headways***	*JourneyPatternHeadway*	0:*	HEADWAYs in JOURNEY PATTERN.
«cntd»	***layovers***	*JourneyPatternLayover*	0:*	LAYOVERs in JOURNEY PATTERN.
«cntd»	***pointsInSequence***	*PointInJourneyPattern*	0:*	POINTs building the JOURNEY PATTERN.
«cntd»	***linksInSequence***	*LinkInJourneyPattern*	*0:**	TIMING LINKs in JOURNEY PATTERN.

10.3.2 ServicePattern

A SERVICE PATTERN is made up of an ordered sequence of STOP POINTs IN JOURNEY PATTERN, with a SERVICE LINK between each pair of consecutive SCHEDULED STOP POINTs. As the same SCHEDULED STOP POINT may occur more than once in the same SERVICE PATTERN, a STOP POINT IN JOURNEY PATTERN is identified by that SERVICE PATTERN together with an ‘order’ attribute.

Each STOP POINT IN JOURNEY PATTERN references a specific SCHEDULED STOP POINT.

NOTE it should be noted that in the terminology of many existing systems and organisations, the sequence of such points is often called a “line”. However, the objective of Transmodel is the separation of concerns: LINEs are groupings of ROUTEs, i.e., schematic views of physical paths through the network, determined through ROUTE POINTs, whereas SERVICE PATTERNs are sequences of points of another type. Thus, a Transmodel LINE is conceptually different from a grouping of SERVICE PATTERNs even if a link exists (a LINE is a group of ROUTEs and each ROUTE is linked to one or more SERVICE PATTERNs). Of course, from the point of view of the precise physical path (along the road network) both types of points (ROUTE POINTs and SCHEDULED STOP POINTs) are on the INFRASTRUCTURE LINKs taken by the vehicle - but the ROUTE and the SERVICE PATTERN are not the same LINK SEQUENCEs.

Table 81 — *ServiceJourneyPattern – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*JourneyPattern*	*::>*	SERVICE JOURNEY PATTERN inherits from JOURNEY PATTERN.
«PK»	***id***	*ServiceJourneyPatternIdType*	1:1	Identifier of a SERVICE JOURNEY PATTERN.
«FK»	***ServiceJourney-PatternType***	*ServiceJourneyPattern-TypeEnum*	0:1	Type of SERVICE JOURNEY PATTERN. • *passenger* (assumed); • *garageRunOut; garageRunIn; turningManoeuvre; other*

10.3.3 PointinJourneyPattern (Abstract in EIRE_NP)

Each POINT IN JOURNEY PATTERN specifies the visit of a SERVICE to a specific POINT in a particular order within the pattern.

Table 82 — *PointInJourneyPattern – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*PointInLinkSequence*</u>	::>	POINT IN JOURNEY PATTERN inherits from POINT IN LINK SEQUENCE.
«PK»	***id***	*PointInJourneyPattern-IdType*	1:1	Identifier of POINT IN JOURNEY PATTERN. (abstract)
«PK»	***(order)***	*xsd:positiveInteger*	1:1	Order of POINT IN JOURNEY PATTERN within JOURNEY PATTERN (Inherited from POINT IN LINK SEQUENCE)
«FK»	***JourneyPatternRef***	*JourneyPatternRef*	0:1	Parent JOURNEY PATTERN. *(*a specialisation of **LinkSequenceRef***)*
«FK»	***PointRef***	*PointRef*	0:1	POINT associated with POINT IN JOURNEY PATTERN. (Abstract)
«FK»	***DestinationDisplayRef***	*DestinationDisplayRef*	0:1	DESTINATION DISPLAY associated with POINT IN JOURNEY PATTERN.
«cntd»	***vias***	*Via*	0:*	Via points associated with POINT IN PATTERN.
«cntd»	***FlexiblePointProperties***	<u>*FlexiblePointProperties*</u>	0:1	Flexible properties of a POINT.
	ChangeOfDestination-Display	*xsd:boolean*	0:1	Whether DESTINATION DISPLAY changes at this point.
	ChangeOfService-Requirements	*xsd:boolean*	0:1	Whether SERVICE REQUIREMENTs change at this point.
«cntd»	***noticeAssignments***	*NoticeAssignmentView*	0:*	NOTICE ASSIGNMENTs, footnotes, etc. associated with the POINT in PATTERN.

10.3.4 StopPointInJourneyPattern (Subcomponent of JourneyPattern)

Each POINT IN JOURNEY PATTERN specifies the visit of a SERVICE to a specific SCHEDULED STOP POINT in a particular order within the pattern.

Table 83 — *StopPointInJourneyPattern – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*PointInLinkSequence*</u>	::>	STOP POINT IN JOURNEY PATTERN inherits from POINT IN JOURNEY PATTERN. NB: In the Schema, ***StopPointInJourneyPattern*** inherits directly from ***PointInLinkSequence*** I.e. not ***PointInJourneyPattern)***, but repeats the attributes of a ***StopPointInJourneyPattern.***
«PK»	***id***	*StopPointInJourneyPattern-IdType*	1:1	Identifier of a STOP POINT IN JOURNEY PATTERN.
«PK»	***(order)***	*xsd:positiveInteger*	1:1	Order of POINT IN JOURNEY PATTERN within JOURNEY PATTERN (Inherited from POINT IN LINK SEQUENCE)
«FK»	***ScheduledStopPoint-Ref***	*ScheduledStopPointRef*	1:1	Reference to SCHEDULED STOP POINT which STOP POINT in JOURNEY PATTERN visits. NOTE: overwrite the TimingPointRef inherited from TIMING POINT IN JOURNEY PATTERN
«FK»	***OnwardService-LinkRef***	*ServiceLinkRef*	0:1	Next SERVICE LINK after STOP POINT in JOURNEY PATTERN visit.

				This attribute shall only be used if multiple SERVICE LINKs are available between two consecutive SCHEDULED STOP POINTs (otherwise the sequence of STOP POINTs IN JOURNEY PATTERN ➔ SCHEDULED STOP POINT is sufficient and remains the main way of describing the stop sequence in the EPIP).
	ForAlighting	*xsd:boolean*	0:1	Whether stop may be used for alighting.
	ForBoarding	*xsd:boolean*	0:1	Whether stop may be used for boarding.
«FK»	***DestinationDisplay-Ref***	*DestinationDisplayRef*	0:1	DESTINATION DISPLAY values to show at this point (also available as a ***DestinationDisplayView***).
«cntd»	***vias***	*Via*	0:*	Destinations that the service goes via
«cntd»	***FlexiblePoint-Properties***	*FlexiblePointProperties*	0:1	Flexible properties of a POINT.
	ChangeOfDestination Display	*xsd:boolean*	0:1	Whether DESTINATION DISPLAY changes at this point.
	ChangeOfService-Requirements	*xsd:boolean*	0:1	Whether SERVICE REQUIREMENTs change at this point.
	noticeAssignments	*NoticeAssignment*	0:*	NOTICEs for POINT IN JOURNEY PATTERN.
	RequestStop	*xsd:boolean*	0:1	Whether stop is a request stop for this journey. Default is false.
«enum»	***RequestMethod***	*RequestMethodTypeEnum*	0:1	Method to Request Stop in this particular service pattern; if none specified, as per stop.
«enum»	***StopUse***	*StopUseEnum*	0:1	Nature of use of stop. Default is '*access*'. • *access; interchangeOnly; passthrough; noBoardingOrAlighting*
	Booking-Arrangements	*BookingArrangements*	0:1	Booking Arrangements for stop if different from those for SERVICE JOURNEY.
	Print	*xsd:boolean*	0:1	Whether the stop is included in printed media. The default is '*true*'. +v1.1
«enum»	***Dynamic***	*DynamicAdvertisementEnum*	0:1	When the STOP POINT IN PATTERN is to be publicised in dynamic media. The default is '*always*'. +v1.1

10.3.5 Direction

A DIRECTION states the relative direction of use of a line, e.g. "North Bound", "South Bound", etc.

DIRECTION is available for informative and filtering purpose (for example in a journey planner UI or a timetable information system) but is not expected to be provided as a descriptive information. It will mainly be used for its name (inherited from DATA MANAGED OBJECT) and type.

Table 84 — *Direction – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*DataManagedObject*	*::>*	DIRECTION inherits from DATA MANAGED OBJECT.
«PK»	***id***	*DirectionIdType*	1:1	Identifier of DIRECTION.

«AK»	***ExternalDirectionRef***	*ExternalObjectRef*	0:1	An alternative code that uniquely identifies the DIRECTION specifically for use in AVMS systems. For VDV compatibility.
«enum»	***DirectionType***	*DirectionTypeEnum*	0:1	A fixed value e.g. *'outbound', 'inbound', 'clockwise', 'anticlockwise'* associated with this direction.
«FK»	***OppositeDirection-Ref***	*DirectionRef*	0:1	Reference to a DIRECTION that runs counter to this DIRECTION.

10.3.6 TimingPoint (Abstract in EIRE_NP)

TIMING POINT is present in the EPIP as it is further specialised by SCHEDULED STOP POINT, but it shall not be used as a standalone object in an EPIP or EIRE_NP conformant document.

Table 85 — *TimingPoint – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*Point*	*::>*	TIMING POINT inherits from POINT.
«PK»	***id***	*TimingPointIdType*	1:1	Identifier of a TIMING POINT.
«Enum»	***TimingPointStatus***	*TimingPointStatusEnum*	0:1	Timing status of POINT: default is 'timingPoint'. • *timingPoint* • *secondaryTimingPoint* • *notTimingPoint.*
	AllowedForWaitTime	*xsd:duration*	0:1	Default Wait time to use for TIMING PATTERNs that use TIMING POINT. TIME DEMAND TYPE specific values may be specified using a TIME DEMAND TYPE TIMING.

10.3.7 ScheduledStopPoint

A POINT where passengers can board or alight from vehicles.

Table 86 — *ScheduledStopPoint – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*TimingPoint*	*::>*	SCHEDULED STOP POINT inherits from TIMING POINT.
«PK»	***id***	*ScheduledStopPointIdType*	1:1	Identifier of a SCHEDULED STOP POINT.
«cntd»	***stopAreas***	*StopAreaRef*	0:*	STOP AREAs to which SCHEDULED STOP POINT belongs.
«cntd»	***tariffZones***	*TariffZoneRef*	0:*	TARIFF ZONEs to which SCHEDULED STOP POINT belongs. Note that a TariffZone is just a specialisation of a Zone without any additional attribute.
	ShortName	*MultilingualString*	0:1	Short Name of SCHEDULED STOP POINT.
	Description	*MultilingualString*	0:1	Description of SCHEDULED STOP POINT.
	Label	*MultilingualString*	0:1	Label of SCHEDULED STOP POINT.
«AK»	***ShortStopCode***	*xsd:normalizedString*	0:1	Short identifier of a SCHEDULED STOP POINT.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public Code of a SCHEDULED STOP POINT.
«AK»	***PrivateCode***	*xsd:normalizedString*	0:1	Alternative identifier of a SCHEDULED STOP POINT. Can be used to associate with legacy systems.

«AK»	***ExternalStopPointRef***	*ExternalObjectRef*	0:1	An alternative code that uniquely identifies the SCHEDULED STOP POINT specifically for use in AVMS systems. For VDV compatibility.
	Url	*xsd:anyURI*	0:1	URL associated with SCHEDULED STOP POINT.
«enum»	***StopType***	*StopPlaceTypeEnum*	1:1	Type of STOP PLACE. Restricted to an allowed value. • *onstreetBus; onstreetTram; airport; railStation' metroStation; busStation; coachStation; tramStation; harbourPort; ferryPort; ferryStop ;liftStation; vehicleRailInterchange* • *other*
	CompassBearing	*AbsoluteBearingType*	0:1	Heading of STOP relative to the street. Degrees from North. This should be considered as a derived value that can be used for presentation purposes when information about the physical stop is not available. The definitive value is the compass bearing found on the QUAY (i.e. physical stop) to which a SCHEDULED STOP POINT is assigned.
	Presentation	*Presentation*	0:1	Presentation values to use when rendering STOP POINT such as a colour and font.
«enum»	***VehicleModes***	*VehicleMode*	0:*	VEHICLE MODEs associated with SCHEDULED STOP POINT.
	ForAlighting	*xsd:boolean*	0:1	Default for whether stop may be used for alighting. May be overridden on specific services.
	ForBoarding	*xsd:boolean*	0:1	Default for whether stop may be used for boarding. May be overridden on specific services.
	RequestStop	*xsd:boolean*	0:1	Default for whether stop is a request stop. May be overridden on specific services.
«enum»	***RequestMethod***	*RequestMethodEnum*	0:1	Method of request stop. The default is '*noneRequired*'. See allowed values below.+ v1.1
«FK»	***CountryRef***	*CountryRef*	0:1	Reference to a country ISO 3166-1
«FK»	***TopographicPlaceRef***	*TopographicPlaceRef*	0:*	Principle TOPOGRAPHIC PLACE associated with SCHEDULED STOP POINT.
	AtCentre	*xsd:boolean*	0:1	Whether STOP POINT can be considered as being at the centre of a TOPOGRAPHIC PLACE. Default is '*false*'.

10.3.8 Service Link

A LINK between an ordered pair of SCHEDULED STOP POINTs.

Table 87 — *ServiceLink – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*Link*	*::>*	SERVICE LINK inherits from LINK. In the EIRE_NP, a schematic of the service is provided through the SERVICE LINK. Therefore, whenever the shape of the line is available, the detailed shape of the different links will have to be provided by the ***gml:LineString*** element of the SERVICE LINK.
«PK»	***id***	*ServiceLinkIdType*	1:1	Identifier of a SERVICE LINK.

«FK»	***FromPointRef***	*ScheduledStopPointRef*	1:1	Reference to SCHEDULED STOP POINT at which SERVICE LINK starts.
«FK»	***ToPointRef***	*ScheduledStopPointRef*	1:1	Reference to SCHEDULED STOP POINT at which SERVICE LINK ends.
«enum»	***VehicleMode***	*TransportModeEnum*	0:1	Mode of transport for which SERVICE LINK applies.
«FK»	***OperationalContext-Ref***	*OperationalContextRef*	0:1	Reference to OPERATIONAL CONTEXT used to distinguish SERVICE LINK from similar links between the same two points.

10.4 Assigning Stops

The Stop Assignment model relates the scheduled use of stops by services in the timetable with the actual physical STOP PLACEs and QUAYs that exist.

A SCHEDULED STOP POINT in the timetable may be allocated to entire STOP PLACE or to a QUAY within it.

Note: In many legacy systems, stop assignment is implicit – the same identifier is used both for the STOP PLACE and for the SCHEDULED STOP POINT and they are considered to be the same. A platform (QUAY) identifier is treated merely as an annotation and no attempt is made to model the physical stop. The Transmodel representation, though more elaborate, is more expressive in that it can describe platform changes, the temporary displacement of stop points, etc, the different boarding points for replacement services, the navigation between platforms, etc. It also makes it easier to integrate different code systems from different modes at multimodal interchanges.

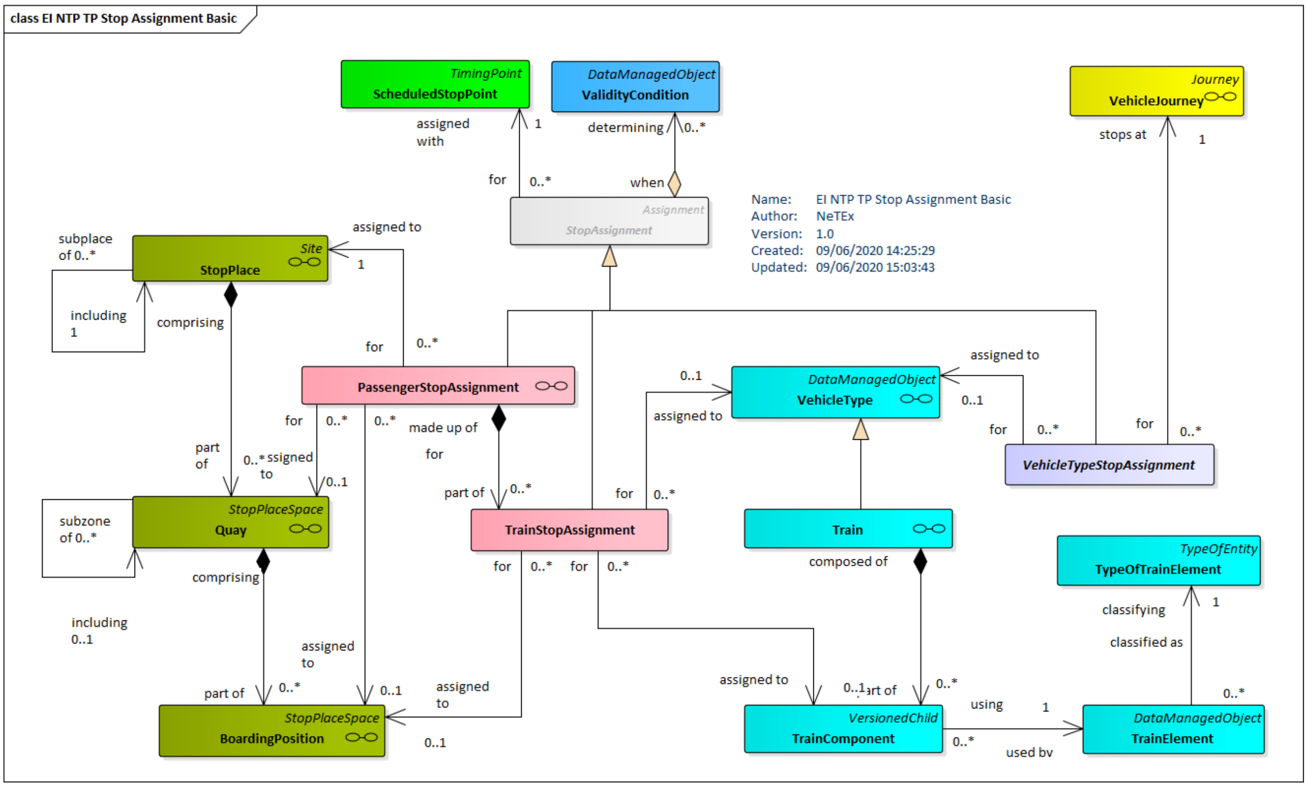


Figure 21 — Stop Assignments (UML)

10.4.1 StopAssignment (Abstract)

A STOP ASSIGNMENT is the allocation of a SCHEDULED STOP POINT to a physical stop, for either a Passenger JOURNEY or VEHICLE SERVICE. It is specialised to create a PASSENGER STOP ASSIGNMENT and a DYNAMIC STOP ASSIGNMENT.

Table 88 — *StopAssignment – XML Element (Abstract)*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	STOP ASSIGNMENT inherits from ASSIGNMENT.
«PK»	***id***	*StopAssignmentIdType*	1:1	Identifier of a STOP ASSIGNMENT.
«PK»	***(order)***	*xsd:positiveInteger*	1:1	Order of ASSIGNMENT (Inherited from ASSIGNMENT) Can be used to indicate precedence of rule.
	BoardingUse	*xsd:boolean*	1:1	Whether STOP ASSIGNMENT allows boarding at the stop.
	AlightingUse	*xsd:boolean*	0:1	Whether STOP ASSIGNMENT allows alighting at the stop.
	PrivateCode	*PrivateCodeStructure*	0:1	Private code associated with element.
«FK»	***ScheduledStopPoint-Ref***	*ScheduledStopPointRef*	1:1	Reference to a SCHEDULED STOP POINT to be assigned.

10.4.2 PassengerStopAssignment

The PASSENGER STOP ASSIGNMENT specialises STOP ASSIGNMENT to allocate a SCHEDULED STOP POINT to a specific STOP PLACE; it may also state a QUAY and BOARDING POSITION.

Table 89 — *PassengerStopAssignment – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*StopAssignment*	::>	PASSENGER STOP ASSIGNMENT inherits from STOP ASSIGNMENT.
«PK»	***id***	*PassengerStopAssignment-IdType*	1:1	Identifier of PASSENGER STOP ASSIGNMENT.
«FK»	***StopPlaceRef***	*StopPlaceRef*	1:1	Reference to STOP PLACE to which the SCHEDULED STOP POINT is assigned.
«FK»	***QuayRef***	*QuayRef*	0:1	Reference to a QUAY within the STOP PLACE to which the SCHEDULED STOP POINT is assigned.
«FK»	***BoardingPositionRef***	*BoardingPositionRef*	0:1	Reference to a BOARDING POSITION with QUAY at STOP PLACE to which the SCHEDULED STOP POINT is assigned.
«cntd»	***trainElements***	*TrainStopAssignment*	0:*	TRAIN STOP ASSIGNMENTs associated with PASSENGER STOP ASSIGNMENT. These can provide more detailed information about individual BOARDING POSITION and carriage alignments.

10.4.3 DynamicStopAssignment

The dynamic association of a SCHEDULED STOP POINT (i.e. a STOP POINT of a SERVICE PATTERN or JOURNEY PATTERN) with the next available STOP PLACE, QUAY or BOARDING POSITION within a STOP PLACE.

A DYNAMIC STOP ASSIGNMENT in effect specifies a change to another PASSENGER STOP ASSIGNMENT, for example to make a platform (QUAY) change; it may apply to all journeys at the SCHEDULED STOP POINT or to just the journeys following a particular JOURNEY PATTERN.

Table 90 — *DynamicStopAssignment – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DynamicStopAssignment*	::>	DYNAMIC STOP ASSIGNMENT inherits from PASSENGER STOP ASSIGNMENT.
«PK»	***Id***	*DynamicAssignmentIdType*	1:1	Identifier of DYNAMIC STOP ASSIGNMENT.
«FK»	***JourneyPatternRef***	*JourneyPatternRef*	0:1	JOURNEY PATTERN that DYNAMIC STOP ASSIGNMENT applies to + v1.1.
«FK»	***PassengerStopAssignmentRef***	*PassengerStopAssignmentRef*	0:1	PASSENGER STOP ASSIGNMENT that DYNAMIC STOP ASSIGNMENT overrides.

10.4.4 TrainStopAssignment (Subcomponent of Passenger Stop Assignment)

The TRAIN STOP ASSIGNMENT is the association of a TRAIN COMPONENT at a SCHEDULED STOP POINT with a specific STOP PLACE and also possibly a specific QUAY and BOARDING POSITION. It is used to indicate the correct boarding point to board a specific part of the train or a specific carriage in a TRAIN.

In EPIP it is used only to indicate a part of the train (using a text value in the ***EntranceToVehicle*** attribute): BOARDING POSITIONs are not in scope.

NB☞ Not used in the EIRE_NP

Table 91 — *TrainStopAssignment – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*StopAssignment*	::>	TRAIN STOP ASSIGNMENT inherits from STOP ASSIGNMENT.
	id	*TrainStopAssignmentIdType*	1:1	Identifier of TRAIN STOP ASSIGNMENT.
«FK»	***ScheduledStopPointRef***	*ScheduledStopPointRef*	0:1	Reference to a SCHEDULED STOP POINT to be assigned.
«FK»	***PassengerStopAssignmentRef***	*PasssengerStopAssignmentRef*	0:1	PASSENGER STOP ASSIGNMENT that this is part of
«FK»	***TrainRef***	*TrainRef*	0:1	TRAIN to which TRAIN STOP ASSIGNMENT applies.
	PositionOfTrainElement	xsd:*positiveInteger*	0:1	Relative position of TRAIN ELEMENT in TRAIN.
«FK»	***TrainComponentRef***	*TrainComponenRef*	0:1	TRAIN COMPONENT to which TRAIN STOP ASSIGNMENT applies. May include derived data as a TRAIN COMPONENT VIEW.
«FK»	***BoardingPositionRef***	*BoardingPositionRef*	0:1	Reference to a BOARDING POSITION with QUAY at STOP PLACE to which the SCHEDULED STOP POINT is assigned.
	EntranceToVehicle	*MultilingualString*	0:1	A specific ENTRANCE to the VEHICLE. E.g. Front, rear.

10.5 Connections, transfers and interchanging

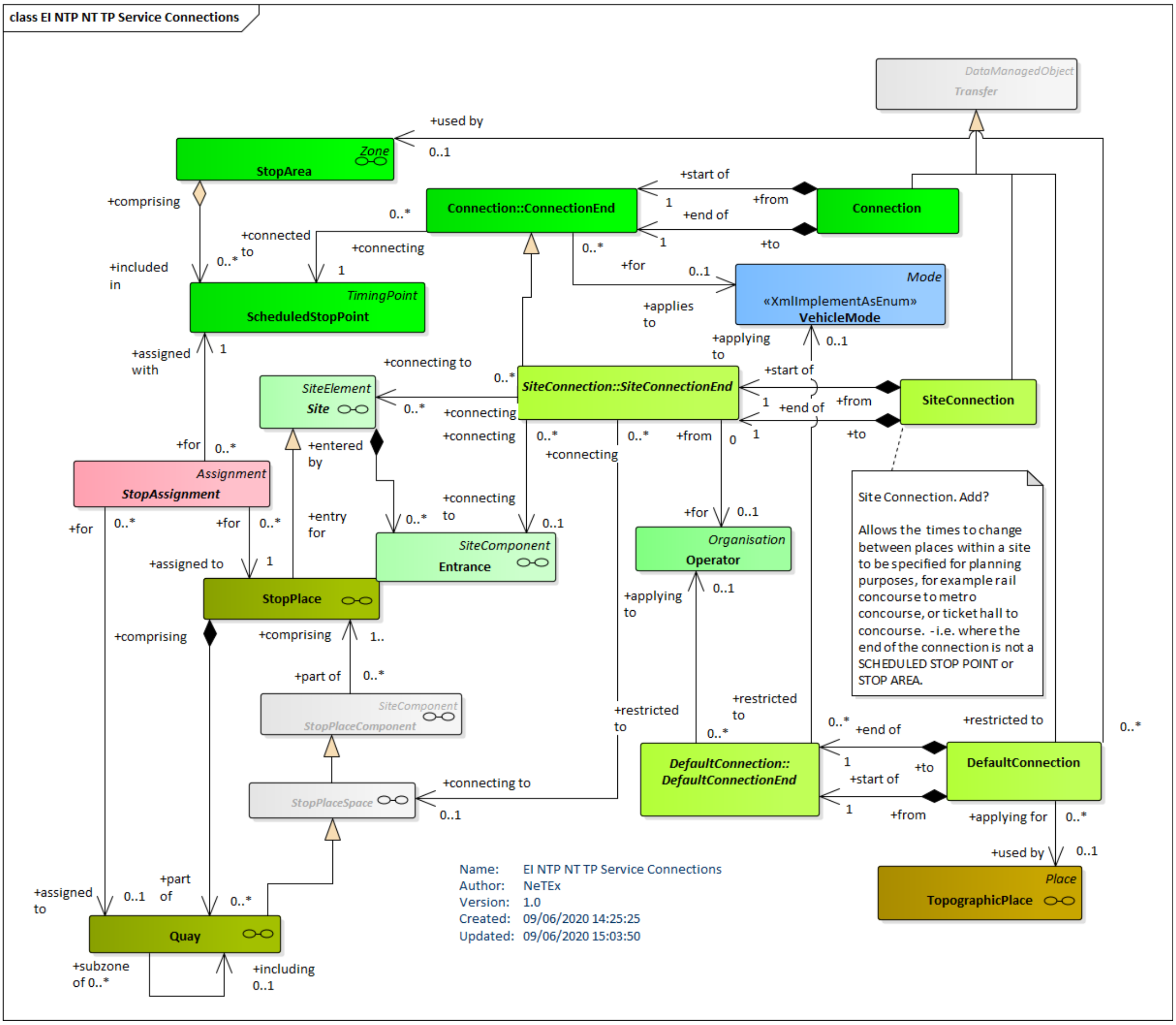
In a given transport network there will be interchange points where SCHEDULED STOP POINTs are close enough together to easily make a transfer between services; this is indicated by a CONNECTION.

Most journey planners allow transfer times for a transport interchange to be specified to some degree – usually as default exchange times to use either for all interchanges of a particular MODE, or at a specific station. Some allow individual transfer times between platforms to be stated. The NeTEx model allows a set of TRANSFER DURATIONs for journey planning to be exchanged with successive levels of precision as to scope (general; a specific station; a specific pair of stops), all inheriting from the TRANSFER concept. An OPERATOR or pair of OPERATORs may also be specified for the transfer.

Thus:

- A DEFAULT CONNECTION specifies default transfer times to be used for interchanging between specific modes at any stop in a TOPOGRAPHIC area (which might be the whole country).
- A SITE CONNECTION specifies default transfer times for any transfer between specific modes at a given site
- A CONNECTION specifies the possibility of making a transfer and transfer times between two SCHEDULED STOP POINTs.

Note that the NeTEx model has components (NAVIGATION PATH, PATH LINK, etc) that allow the very detailed specification of transfer times by a specific route under specific circumstances – but these are outside of the scope of the current EIRE_NP.



Site Connection. Add?

Allows the times to change between places within a site to be specified for planning purposes, for example rail concourse to metro concourse, or ticket hall to concourse. -i.e. where the end of the connection is not a SCHEDULED STOP POINT or STOP AREA.

Name: EI NTP NT TP Service Connections
Author: NeTEx
Version: 1.0
Created: 09/06/2020 14:25:25
Updated: 09/06/2020 15:03:50

Figure 22 — Connection (UML)

10.5.1 Transfer (Abstract)

A TRANSFER represents a pair of POINTs located sufficiently close together that they offer a passenger the possibility to reach one of these POINTs when starting at the other one in a timescale which is realistic when carrying out a trip.

A TRANSFER has a TRANSFER DURATION and may be specialised, for example as CONNECTION.

Table 92 — *Transfer – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	TRANSFER inherits from DATA MANAGED OBJECT.
«PK»	***id***	*TransferIdType*	1:1	Identifier of TRANSFER.
	Name	*MultilingualString*	0:1	Name of TRANSFER.
«FK»	***TypeOfTransferRef***	*TypeOfTransferRef*	1:1	Reference to identifier of a TYPE OF TRANSFER.
	Description	*MultilingualString*	0:1	Description of TRANSFER.
	Distance	*DistanceType*	0:1	Distance of TRANSFER link.
«cntd»	***TransferDuration***	*TransferDuration*	0:1	Scheduling TRANSFER TIMEs that apply to TRANSFER
«cntd»	***WalkTransfer-Duration***	*TransferDuration*	0:1	Walking TRANSFER TIMEs that apply to TRANSFER.
	BothWays	*xsd:boolean*	0:1	Whether TRANSFER can be traversed in both directions (with the same tings).
«cntd»	(***From***)	*(TransferEnd)*	1:1	Origin end of TRANSFER. Specialised in subclass
«cntd»	(***To***)	*(TransferEnd)*	1:1	Destination end of TRANSFER. Specialised in subclass

10.5.2 Transfer Duration (Subcomponent of Transfer)

Specifies transfer times for different sorts of user.

Table 93 — *TransferDuration – XML Element*

Classification	Name	Type	Cardinality	Description
	DefaultDuration	*xsd:duration*	1:1	Default time needed to make transfer.
	FrequentTravellerDuration	*xsd:duration*	0:1	Time for a traveller familiar with the journey to make the transfer.
	OccasionalTravellerDuration	*xsd:duration*	0:1	Time for a traveller unfamiliar with the journey to make transfer.

	MobilityRestrictedTraveller-Duration	*xsd:duration*	0:1	Time for mobility impaired travellers to make transfer.

10.5.3 Connection

A CONNECTION describes the physical (spatial) possibility for a passenger to change from one public transport vehicle to another to continue their trip, as determined by two SCHEDULED STOP POINTs. Different times may be necessary to cover the link between these points, depending on the kind of passenger (e.g., if they have impaired mobility). These times are inherited from TRANSFER.

Table 94 — *Connection – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Transfer*	::>	CONNECTION inherits from TRANSFER.
«PK»	***id***	*ConnectionIdType*	1:1	Identifier of a CONNECTION.
«AK»	***ExternalStop-PointRef***	*ExternalObjectRef*	0:1	An alternative code that uniquely identifies the CONNECTION specifically for use in AVMS systems. *NOTE For VDV compatibility.*
«cntd»	***From***	*ConnectionEnd*	1:1	Properties of the end at which CONNECTION starts.
«cntd»	***To***	*ConnectionEnd*	1:1	Properties of the end at which CONNECTION ends.
	TransferOnly	*xsd:boolean*	0:1	Whether when connecting at this stop passengers may only transfer and may not enter or exit at the station. If 'true' can only transfer.

10.5.4 ConnectionEnd (Subcomponent of Connection)

A *CONNECTION* end describes each end of a CONNECTION, that is, to which SCHEDULED STOP POINT it links.

Table 95 — *ConnectionEnd – XML Element*

Classification	Name	Type	Cardinality	Description
	Mode	*TransportModeEnum*	0:1	Reference to mode of SCHEDULED STOP POINT to which connection connects.
«FK»	***ScheduledStopPointRef***	*ScheduledStopPointRef*	0:1	Reference to SCHEDULED STOP POINT to which it connects.

10.5.5 SiteConnection

The SITE CONNECTION is similar to the CONNECTION but describes the physical (spatial) possibility for a passenger to go from a site (STOP PLACE or POINT OF INTEREST in the profile) to another.

Table 96 — *SiteConnection – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Transfer*	::>	SITE CONNECTION inherits from TRANSFER.
«PK»	***id***	*ConnectionIdType*	1:1	Identifier of SITE CONNECTION.

«cntd»	***From***	*SiteConnectionEnd*	1:1	Start end of SITE CONNECTION.
«cntd»	***To***	*SiteConnectionEnd*	1:1	Destination end of SITE CONNECTION.
	navigationPaths	*NavigationPath*	0:1	NAVIGATION PATHs between SITEs.

10.5.6 SiteConnectionEnd (Subcomponent of SiteConnection)

A ***SiteConnectionEnd*** is one end of a SITE CONNECTION link. It can reference either a STOP PLACE or a POINT OF INTEREST. If a STOP PLACE, it may also indicate a MODE and a QUAY. If no QUAY is given, then the SITE CONNECTION specifies a default for any QUAY of the stated MODE.

Note that the definitions of most POINTs OF INTEREST will come from datasets other than EPIP ones (e.g., from INSPIRE, OSM, commercial POI databases, etc.); therefore, a reference to a POINT OF INTEREST will typically be a reference to an external object see 8.2 for ID structure and external references).

Table 97 — *SiteConnectionEnd – XML Element*

Classification		**Name**		**Type**	**Cardinality**	**Description**
::>		::>		*ConnectionEnd*	::>	SITE CONNECTION END inherits from CONNECTION END.
«FK»		***StopPlaceRef***		*StopPlaceRef*	0:1	Reference to destination STOP PLACE of SITE CONNECTION.
a	«FK»	***StopPlaceSpaceRef***		*StopPlaceSpaceRef*	1:1	Reference to destination space of SITE CONNECTION. May be a QUAY or an ACCESS SPACE.
	«FK»	a	***AccessSpaceRef***	*AccessSpaceRef*	0:1	Reference to an ACCESS SPACE.
	FK»	b	***BoardingPositionRef***	*BoardingPositionRef*	0:1	Reference to a BOARDING POSITION.
	FK»	c	***QuayRef***	*QuayRef*	0:1	Reference to a QUAY.
	FK»	***StopPlaceEntranceRef***		*StopPlaceEntranceRef*	0:1	Reference to a STOP PLACE ENTRANCE.
b	«FK»	***PointOfInterestRef***		*PointOfInterestRef*	0:1	Reference to a POINT OF INTEREST.
	«FK»	***PointOfInterestSpaceRef***		*PointOfInterestSpaceRef*	0:1	Reference to a POINT OF INTEREST SPACE.
	«FK»	***PointOfInterestEntranceRef***		*PointOfInterestEntranceRef*	0:1	Reference to a POINT OF INTEREST ENTRANCE.
c	«FK»	***ParkingRef***		*ParkingRef*	0:1	Reference to a PARKING.
	«FK»	***ParkingAreaRef***		*ParkingAreaRef*	0:1	Reference to a PARKING AREA.
	«FK»	***ParkingEntranceRef***		*ParkingEntranceRef*	0:1	Reference to a PARKING ENTRANCE.
«FK»		***OperatorRef***		*OperatorRef*	0:1	Reference to the destination point of SITE CONNECTION.
		Label		*MultilingualString*	0:1	Label at end of site connection.

NeTEx XML EXAMPLE

The following example illustrates use of references to external POINTs OF INTEREST, the first coming from OSM in France and the second coming from INSPIRE in Slovakia. Note that the **versionRef** attribute is mandatory for external objects (the value can be "any" if the version is unknown or the real version number of the POINT OF INTEREST if it is known).

```
<PointOfInterestRef ref="FR_OSM_Poi:55711945" versionRef="any"/>


<PointOfInterestRef ref="SK_INSPIRE_Poi:SK.SOPSK.SKUEV0319" versionRef="any"/>
```

10.5.7 DefaultConnection

At the most general level, a DEFAULT CONNECTION sets a default time for a transfer between modes on any SITE, to be used if there is no more specific value for an individual site. It is of particular interest for journey planners when connection times are not provided. The ***from*** and ***to*** attribute can refer the same place, therefore meaning that the default connection information is valid for all connections internal to the referenced place.

Table 98 — *DefaultConnection – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Transfer*	::>	DEFAULT CONNECTION inherits from TRANSFER.
«PK»	***id***	*ConnectionIdType*	1:1	Identifier of DEFAULT CONNECTION.
«cntd»	***From***	*DefaultConnectionEnd*	1:1	Start end of DEFAULT CONNECTION.
«cntd»	***To***	*DefaultConnectionEnd*	1:1	Destination end of DEFAULT CONNECTION.
«FK»	***TopographicPlaceView***	*TopographicPlaceRef*	0:*	TOPOGRAPHIC PLACE to which DEFAULT CONNECTION applies.
«FK»	***StopAreaRef***	*StopAreaRef*	0:*	STOP AREA to which DEFAULT TRANSFER applies.
«FK»	***SiteElementRef***	*SiteElementRef*	0:*	SITE ELEMENT to which DEFAULT CONNECTION applies.

10.5.8 DefaultConnectionEnd (Subcomponent of DefaultConnection)

A ***DefaultConnectionEnd*** is one end of a DEFAULT CONNECTION. It can specify a MODE and an OPERATOR.

Classification	Name	Type	Cardinality	Description
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	Transport MODE of DEFAULT CONNECTION END. See STOP PLACE for allowed values.
«EV»	OperatorView	OperatorView	0:1	Reference to an OPERATOR, along with derived name, etc.

10.6 Vehicle journeys and passing times

A detailed description of each journey in a timetable is provided by a SERVICE JOURNEY.

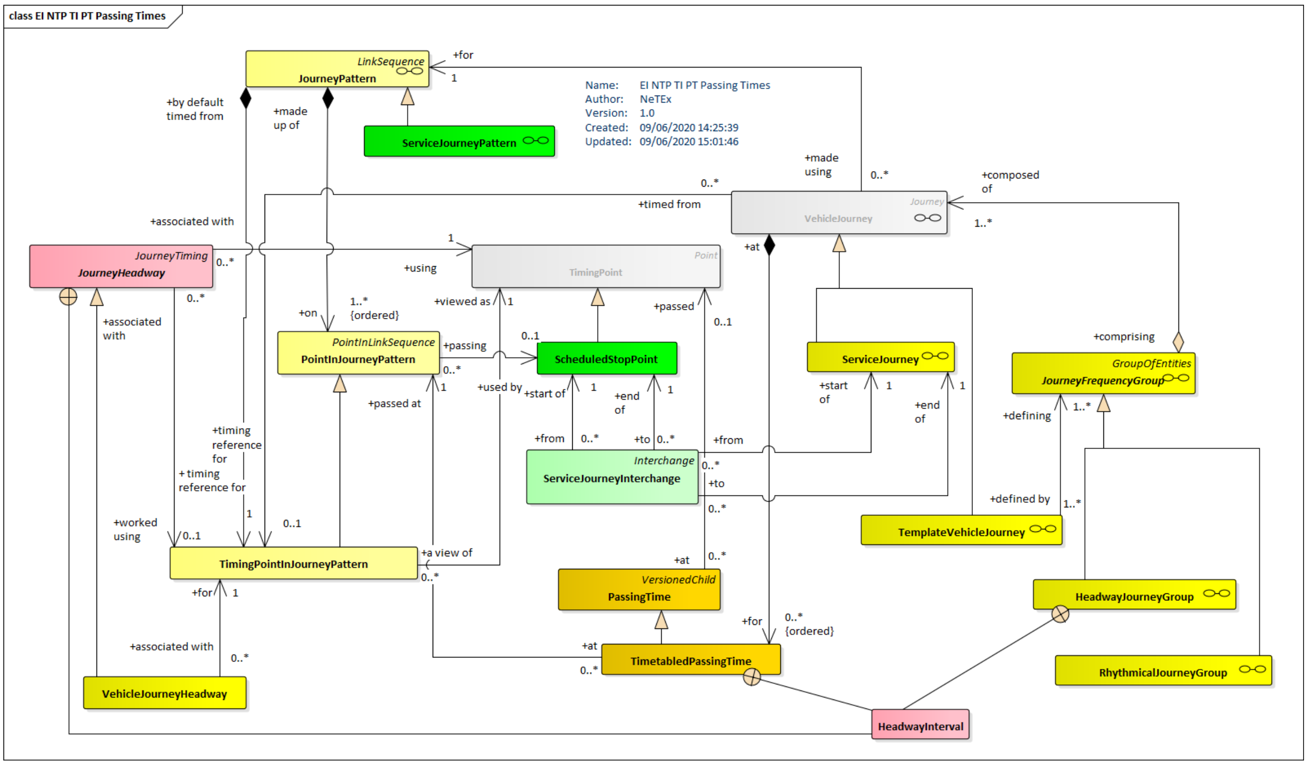


Figure 23 — Journeys and Passing Times (UML)

A VEHICLE JOURNEY is the defined movement of a vehicle using a specified JOURNEY PATTERN on a particular ROUTE. By virtue of being defined for a DAY TYPE, a VEHICLE JOURNEY is a class of journeys that would take place at the same time on each day of a specific DAY TYPE.

There are two main types of VEHICLE JOURNEYs: passenger-carrying SERVICE JOURNEYs and non-service DEAD RUNs used to position vehicles ready for service; the EPIP and EIRE_NP are only concerned with SERVICE JOURNEYs.

A SERVICE JOURNEY is a VEHICLE JOURNEY on which passengers will be allowed to board or alight from vehicles at stops. There are several different possible ways to define SERVICE JOURNEYs, in particular the two following:

— As the service between an origin and a destination, as advertised to the public.

— As the longest service during which a passenger is allowed to stay on the same vehicle. (Which might encompass several of the above).

A default VEHICLE TYPE may be proposed for a journey, chosen according to the time of day at which a SERVICE JOURNEY takes place, and the ROUTE and JOURNEY PATTERN it covers.

A SERVICE JOURNEY INTERCHANGE describes the scheduled possibility for the transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs and for a specific STOP POINT IN JOURNEY PATTERN of the SERVICE JOURNEY.

The facilities (grouped in a SERVICE FACILITY SET) available for the SERVICE JOURNEY are also important related information, especially when advertising to the public.

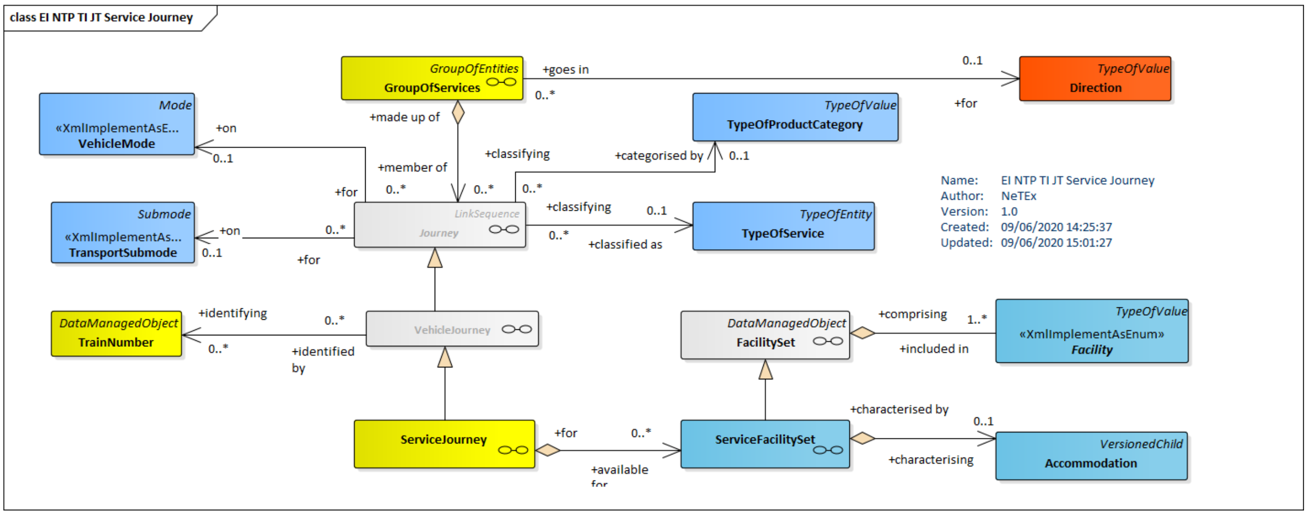


Figure 24 — Service Journey (UML)

10.6.1 Journey (Abstract)

The ***Journey*** element holds common properties for any JOURNEY (VEHICLE JOURNEY, SERVICE JOURNEY, DEAD RUN, etc).

Specialisations of Journey used in the EPIP and EIRE_NP include ***ServiceJourney*** and ***TemplateJourney.***

Table 99 — *Journey* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	JOURNEY inherits from LINK SEQUENCE.
«PK»	***id***	*JourneyIdType*	1:1	Identifier of JOURNEY.
«enum»	***TransportMode***	*VehicleModeEnum*	0:1	Transport MODE of JOURNEY. See STOP PLACE for allowed values.
«enum»	***TransportSubmode***	*TransportSubmode*	0:1	Transport Sub MODE of JOURNEY.
«AK»	***ExternalJourneyRef***	*ExternalObjectRef*	0:1	An alternative code that uniquely identifies the JOURNEY, specifically for use in AVMS systems. *NOTE For VDV compatibility.*
«FK»	***TypeOfProductCategoryRef***	*TypeOfProductCategoryRef*	0:1	PRODUCT CATEGORY of a JOURNEY.
«FK»	***TypeOfServiceRef***	*TypeOfServiceRef*	0:1	TYPE OF SERVICE of JOURNEY.
		CHOICE	:1	Can reference or embed inline
«FK»	**a** ***LinkSequenceProjectionRef***	*LinkSequenceProjectionRef*	0:1	Reference to LINK SEQUENCE PROJECTION to use to PROJECT JOURNEY onto map, etc.
«FK»	**b** ***LinkSequenceProjection***	*LinkSequenceProjection*	0:1	LINK SEQUENCE PROJECTION to use to PROJECT JOURNEY onto map, etc.
	Monitored	*xsd:boolean*	0:1	Whether the journey will be monitored in real time.
«cntd»	***journeyAccountings***	*JourneyAccounting*	0:*	JOURNEY ACCOUNTINGs that applies on JOURNEY.
«cntd»	***noticeAssignments***	*NoticeAssignment*	0:1	NOTICEs relevant for the whole JOURNEY.

10.6.2 ServiceJourney

A SERVICE JOURNEY is a VEHICLE JOURNEY on which passengers will be allowed to board or alight from vehicles at stops.

Table 100 — ServiceJourney – XML Element

Classification	Name	Type	Cardinality	Description
::>	*::>*	*Journey*	*::>*	SERVICE JOURNEY inherits from JOURNEY. It includes elements from VEHICLE JOURNEY.
«PK»	***id***	*ServiceJourneyIdType*	1:1	Identifier of SERVICE JOURNEY.
	ServiceAlteration	*ServiceAlterationEnum*	0:1	Status to journey - planned, cancelled, etc.
	DepartureTime	*xsd:time*	0:1	Time of departure.
	DepartureDayOffset	*DayOffsetType*	0:1	Day offset if the day of departure time of VEHICLE JOURNEY differs from the current OPERATING DAY.
cntd»	***Frequency***	*Frequency*	0:1	Frequency of Journey.
	JourneyDuration	*xsd:duration*	0:1	Total length of Journey. Can be computed from individual times. Add to Departure time to obtain JOURNEY arrival time.
«cntd»	***dayTypes***	*DayTypeRef*	0:1	DAY TYPEs for Journey.
«FK»	***RouteRef***	*RouteRef*	0:1	Reference to a ROUTE.
«FK»	***JourneyPatternRef***	*JourneyPatternRef*	0:1	Reference to a JOURNEY PATTERN.
«FK»	***TimeDemandTypeRef***	*TimeDemandType*	0:1	Reference to a TIME DEMAND TYPE used at start of JOURNEY.
«FK»	***TimingAlgorithmTypeRef***	*TimingAlgorithmTypeRef*	0:1	Reference to a TIMING ALGORITHM TYPE.
«FK»	***JourneyFrequencyGroupRef***	*JourneyFrequencyGroupRef*	0:1	Reference to a JOURNEY FREQUENCY GROUP.
«FK»	***VehicleTypeRef***	*VehicleTypeRef*	0:1	Reference to TYPE OF VEHICLE to use on SERVICE JOURNEY.
«FK»	***OperationalContextRef***	*OperationalContextRef*	0:1	Reference to an OPERATIONAL CONTEXT.
«FK»	***BlockRef***	*BlockRef*	0:1	Reference to a BLOCK.
«FK»	***CourseOfJourneysRef***	*CourseOfJourneysRef*	0:1	Reference to a COURSE OF JOURNEYs (Run).
	PublicCode	*xsd:normalizedString*	0:1	Public code for JOURNEY.
«FK»	***a*** ***OperatorRef***	*OperatorView*	0:1	Reference to an OPERATOR.
«EV»	***b*** ***OperatorView***	*OperatorView*	0:1	Simplified view of OPERATOR. All data except the identifier will be derived through the relationship.
«FK»	***a*** ***LineRef***	*LineRef*	0:1	Reference to a LINE.
«EV»	***b*** ***LineView***	*LineView*	0:1	Reference to LINE of SERVICE JOURNEY. May include derived values of LINE.
«EV»	***FlexibleLineView***	*FlexibleLine_DerivedView*	*0:1*	A group of FLEXIBLE ROUTEs of which is generally known to the public by a similar name or number and which have common booking arrangements.
«enum	***DirectionType***	*DirectionTypeEnum*	0:1	Type of DIRECTION. DirectionType is available for filtering purpose (for example in a journey planner UI or timetable information system) but is not expected to be provided as a descriptive information.

«EV»	***JourneyPatternView***		*JourneyPatternView*	0:1	Reference to JOURNEY PATTERN of SERVICE JOURNEY and derived values of PATTERN. NOTE: To be used as an alternative to ***JourneyPatternRef*** above in ***VehicleJourneyReferencesGroup***
«cntd»	***groupsOfServices***		GroupOfServiceRef	0:*	Grouping of services of a journey - for a multi-part journey only.
«cntd»	***timeDemandTypes***		TimeDemandTypeRef	0:*	Other TIME DEMAND TYPEs used in journey. NOTE: To be used as an alternative to ***TimeDemandTypeRef*** above in ***VehicleJourneyReferencesGroup***
«cntd»	***trainNumbers***		*TrainNumberRef*	0:*	TRAIN NUMBER s - derived through JOURNEY PARTs of a journey - for a multi-part journey only.
«cntd»	***Origin***		*JourneyEndpoint*	0:1	Origin of SERVICE JOURNEY.
«cntd»	***Destination***		*JourneyEndpoint*	0:1	Destination of SERVICE JOURNEY.
	Print		*xsd:boolean*	0:1	Whether this journey should be visible to the public in print channels.
«enum»	***Dynamic***		*DynamicAdvertisement-Enum*	0:1	When this journey should be visible to public in dynamic channels.
«cntd»	***waitTimes***		*VehicleJourneyWaitTime*	0:*	WAIT TIMEs for VEHICLE JOURNEY at different TIMING POINTs.
«cntd»	***runTimes***		*VehicleJourneyRunTime*	0:*	RUN TIMEs for VEHICLE JOURNEY over different TIMING LINKs.
«cntd»	***layovers***		*VehicleJourneyLayover*	0:*	LAYOVER times for VEHICLE JOURNEY.
«cntd»	***passingTimes***		*TimetabledPassingTime*	0:*	PASSING TIMEs for VEHICLE JOURNEY.
«cntd»	***parts***		*JourneyPart*	0:*	JOURNEY PARTs of a journey - for a multi-part journey only.
«cntd»	***calls***		*Call*	0:*	CALLs made by SERVICE JOURNEY.
«cntd»	***facilities***		*ServiceFacilitySet*	0:*	Facilities on SERVICE JOURNEY.
«cntd»	***groupsOfServices***		*GroupOfServices*	0:*	GROUP OF SERVICEs to which SERVICE JOURNEY belongs.
«cntd»	***checkConstraints***		*CheckConstraint*	0:*	CHECK CONSTRAINTs which apply to SERVICE JOURNEY, check in time, security time. These are advisory and not for journey planning.
	a	***PassengerCarrying-RequirementRef***	*PassengerCarrying-RequirementRef*	0:1	Reference to a PASSENGER CARRYING REQUIREMENT.
«EV»	***b***	***PassengerCarrying-RequirementsView***	*PassengerCarrying-Requirement*	0:1	Requirements for carrying passengers.
	TrainSize		*TrainSize*	0:1	Requirements for TRAIN SIZE.
«cntd»	***equipments***		*vehicleEquipment*	0:*	VEHICLE EQUIPMENT available on service.
			CHOICE		Properties may either be embedded or refer to a shared instance.
«FK»	***a***	***FlexibleServiceProperties Ref***	*FlexibleServiceProperties-Ref*	0:1	Reference to a FLEXIBLE SERVICE PROPERTIES.
«cntd»	***b***	***FlexibleServiceProperties***	*FlexibleServiceProperties*	0:1	Flexible properties of SERVICE JOURNEY.

10.6.3 PassingTime

Many different types of passenger information systems require data about the planned and actual arrival and departure times of vehicles; usually without requiring access to the detailed scheduling data used to compute them. In the conceptual model, such data is represented in a highly normalised form as PASSING TIMEs. Each PASSING TIME indicates the time of a vehicle at a point in a journey pattern. There can be different types of time (arrival, departure, estimated, observed, etc.)

The PASSING TIMEs that are the result of the scheduling process and to be published in a timetable are called TIMETABLED PASSING TIMEs.

The TIMETABLED PASSING TIMEs are normally generated a long time before the day of operation and are valid over a long period of time (e.g., summer, winter timetable).

An 'edge' case that has to be handled with care occurs on an OPERATING DAY when the time of a time zone is changed (i.e., from summer time to winter time and reverse). For the EPIP and EIRE_NP, the rule is that all the PASSING TIMEs of a SERVICE JOURNEY have to be provided using the same reference time: therefore, even if the time change is at 2:00 am, the correct sequence of passing times of a journey starting before 2:00 is, for example, 1:45 - 1:55 - 2:05 - 2:15, and not 1:45 - 1:55 - 3:05 - 3:15; nor 1:45 - 1:55 - 1:05 - 1:15.

Also note that some JOURNEYs may be skipped due to the time change: for example, a JOURNEY starting between 2:00 and 3:00 when 1h is added at 2:00 in the morning. As soon as there are no times between 2:00 and 3:00 (at 2:00 the time will immediately become 3:00), a JOURNEY operating in the corresponding OPERATING DAY and having a starting time between 2:00 and 3:00 will have to be ignored in order to realign the schedules. Conversely extra intercalary JOURNEYs may need to be injected when the clocks go back.

10.6.4 PassingTime (Abstract)

A PASSING TIME associates a time with a POINT IN JOURNEY PATTERN. It is specialised to create the different concrete times (TIMETABLED PASSING TIME, etc).

Table 101 — *PassingTime* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	*::>*	*VersionedChild*	*::>*	PASSING TIME inherits from VERSIONED CHILD.
«PK»	***id***	*PassingTimeIdType*	0:1	Identifier of PASSING TIME.
«FK»	***JourneyRef***	*JourneyRef*	0:1	Reference to JOURNEY for which this is a PASSING TIME.
	AlightAndReboard	*xsd:boolean*	1:1	Number of days that the departure day is after the start day of vehicle journey.
«FK»	***PointInJourney-PatternRef***	*PointInLinkSequenceRef*	0:1	POINT IN SEQUENCE for which this is a PASSING TIME. Refer to the SCHEDULE STOP POINT

10.6.5 TimetabledPassingTime (Subcomponent – Service Journey)

Table 102 — *TimetabledPassingTime* – XML Element

Classification	Name	Type	Cardinality	Description
::>	*::>*	*PassingTime*	*::>*	TIMETABLED PASSING TIME inherits from PASSING TIME.

«PK»	***id***	*TimetablePassingTimeIdType*	0:1	Identifier of TIMETABLE PASSING TIME.
	ArrivalTime	*xsd:time*	0:1	Arrival time at POINT IN PATTERN.
	ArrivalDayOffset	*DayOffsetType*	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.
	DepartureTime	*xsd:time*	0:1	Departure time at POINT IN PATTERN.
	DepartureDayOffset	*DayOffsetType*	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.
	WaitingTime	*xsd:duration*	0:1	Waiting time at POINT IN PATTERN.
«cntd»	***Headway***	*HeadwayInterval*	0:1	Headway interval at POINT IN PATTERN.
	LatestArrivalTime	*xsd:time*	0:1	Latest Arrival time at POINT IN PATTERN.
	LatestArrivalDayOffset	*DayOffsetType*	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.
	EarliestDepartureTime	*xsd:time*	0:1	Earliest Departure time at POINT IN PATTERN.
	EarliestDeparture-DayOffset	*DayOffsetType*	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.

10.6.6 GroupOfServices

A grouping of SERVICE JOURNEYs, GROUPs of SERVICEs are of use in particular for organising sets of similar journeys for presentation in tabular or in passenger information systems, for example “*all the weekday, outbound services for LINE 2*”.

The set of SERVICE JOURNEYs making up the complete timetable for a LINE (comprising services in both directions of travel and for different DAY TYPEs) will typically be partitioned into a number of different GROUPs of SERVICEs (for example “*Outbound, Weekdays”; “Inbound, Weekdays”; “Outbound Weekends and public Holidays”; “Inbound, Public holidays*”, etc).

Another use of GROUPs OF SERVICES is to segregate journeys for the same LINE that have very different SERVICE PATTERNs, for example “*express*” and “*stopping*” groups of journeys, or the journeys that go down a particular branch. (Note that any such groupings are usually irrelevant for journey planners, which are select the journeys directly, but are very important for arranging tabular representations of the timetable).

Table 103 — *GroupOfServices* – XML Element

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*GroupOfEntities*	::>	GROUP OF SERVICES inherits from GROUP OF ENTITIES.
«PK»	***id***	*GroupOfServicesIdType*	1:1	Identifier of GROUP OF SERVICEs.
«FK»	***dayTypes***	*DayTypeRef*	0:*	Reference DAY TYPEs applying to the GROUP OF SERVICEs.
	DirectionType	*DirectionTypeEnum*	1:1	Type of DIRECTION. See Part 1.
«EV»	***DirectionView***	*DirectionView*	0:1	Reference to common DIRECTION of SERVICE JOURNEYs in the GROUP OF SERVICE, with derived values.
	origin	*EndPoint_DerivedView*	0:1	View of derived data about origin of journey. E.g., ***ScheduledStopPointRef, Name, StopType, etc***

	destination	*EndPoint_DerivedView*	0:1	View of derived data about destination of journey.
«cntd»	***destinationDisplays***	*DestinationDisplay*	0:*	DESTINATION DISPLAYs associated with the GROUP OF SERVICEs. Can be used to describe VIA points.
«cntd»	***members***	*VehicleJourneyRef*	0:*	Members of a GROUP OF SERVICEs.
«cntd»	***noticeAssignments***	*NoticeAssignmentView*	0:*	NOTICE ASSIGNMENTs that apply to a GROUP OF SERVICE.

10.6.7 GroupOfServicesMember (Subcomponent of GroupOfServices)

An association that records membership of a GROUP OF SERVICEs by a JOURNEY. ***GroupOfServicesMember*** is a NeTEx implementation artefact that is not described in the Transmodel Conceptual model.

The reference to the JOURNEY may be direct (e.g., a ***ServiceJourneyRef***), or indirect (e.g., a ***TrainNumberRef*** or a ***ServiceDesignator***). The EPIP and EIRE_NP use only direct references to SERVICE JOURNEYs. Each ***GroupOfServicesMember*** may have its own ***noticeAssignments***.

Table 104 — Table 18 – *GroupOfServicesMember* – XML Element

Classification	Name		Type	Cardinality	Description
::>	*::>*		*GroupOfEntities*	*::>*	GROUP OF SERVICEs MEMBER inherits from GROUP OF ENTITies MEMBER.
«PK»	***id***		*GroupOfServicesMembers-IdType*	1:1	Identifier of GROUP OF SERVICEs MEMBER.
«PK»	(***order***)		*xsd:positiveInteger*	1:1	Inherited from GROUP OF ENTITIES MEMBER.
«FK»	***GroupOfServicesRef***		*GroupOfServicesRef*	0:1	Reference to a parent GROUP OF SERVICEs. Given by context.
			CHOICE	1:1	Reference to a JOURNEY.
«FK»	a	***JourneyRef***	*JourneyRef+*	0:1	Direct Reference to a JOURNEY.
«FK»	b	***TrainNumberRef***	*TrainNumberRef*	0:1	Reference to a JOURNEY by means of a TRAIN NUMBER REF.
«cntd»	c	***Journey-Designator***	*JourneyDesignator*	0:1	Reference to JOURNEY by value. See JOURNEY DESIGNATOR earlier.
«cntd»	d	***Service-Designator***	*ServiceDesignator*	0:1	Reference to SERVICEJOURNEY by value. See SERVICE DESIGNATOR earlier.
«cntd»	***noticeAssignments***		*NoticeAssignmentView*	0:*	NOTICE ASSIGNMENTs that apply to a GROUP OF SERVICE.

10.7 Very frequent journeys

When journeys run at short or regular intervals, rather than list all the journeys, passenger information systems often show multiple journeys as a single column in the timetable. For example, *"And then every 20 minutes until 6 pm"****.*** Note that this is a convention to simplify the presentation of schedules; the operational timetable implementing the schedule will still require separate DATED SERVICE JOURNEYS for each journey to be run.

The Journey Frequency model (see diagram for SERVICE JOURNEYs) allows multiple journeys to be aggregated using a TEMPLATE JOURNEY and an overall frequency to be specified for them, using for example a HEADWAY JOURNEY GROUP (e.g., '*every 10 minutes*').

Using an aggregated schedule can also significantly reduce the amount of data that needs to be exchanged.

10.7.1 TemplateVehicleJourney

A TEMPLATE VEHICLE JOURNEY is a repeating VEHICLE JOURNEY for which a frequency is specified as a JOURNEY FREQUENCY GROUP. It may thus represent multiple journeys.

The frequency of the journey is defined by a JOURNEY FREQUENCY GROUP. Two types of JOURNEY FREQUENCY GROUP are possible, *'headway'* (interval based*)*, and *'rhythmic'* (minutes past the hour). The EPIP is restricted to using only *'Headway'* JOURNEY FREQUENCY GROUPs.

NB☞The EIRE_NP allows either.

Table 105 — *TemplateVehicleJourney* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*Journey*	::>	TEMPLATE VEHICLE JOURNEY inherits from JOURNEY.
«PK»	***id***	*TemplateVehicleJourney-IdType*	0:1	Identifier of TEMPLATE VEHICLE JOURNEY.
«enum»	***TemplateVehicle-JourneyType***	*TemplateVehicleJourney-TypeEnum*	0:1	Type of TEMPLATE VEHICLE JOURNEY. • *headway* • *rhythmic* • *other*
«cntd»	***frequencyGroups***	*JourneyFrequencyGroup*	0:*	FREQUENCY GROUPs defining TEMPLATE VEHICLE JOURNEY.

10.7.2 JourneyFrequencyGroup (Abstract)

A JOURNEY FREQUENCY GROUP defines a set of JOURNEYs and a start and end time during which the frequency-based services will run. It is further specialised in order to describe specific frequency behaviour such as headway-based services or rhythmical services.

All the journeys in a JOURNEY FREQUENCY GROUP should follow the same JOURNEY PATTERN. When publishing a TEMPLATE JOURNEY, the JOURNEY PATTERN of the first journey in the group will be used to determine the stops visits for the group as a whole, i.e., at which to show the indicated frequency). EPIP requires at least one Journey to be specified in a JOURNEY FREQUENCY GROUP.

Table 106 — *JourneyFrequencyGroup* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*GroupOfEntities*	::>	JOURNEY FREQUENCY GROUP inherits from GROUP OF ENTITies.
«PK»	***id***	*FrequencyGroupIdType*	0:1	Identifier of FREQUENCY GROUP.
	FirstDepartureTime	*xsd:time*	0:1	Time of the first departure in FREQUENCY GROUP.
	LastDepartureTime	*xsd:time*	0:1	Time of last departure in FREQUENCY GROUP.
	DayOffset	*xsd:integer*	0:1	Offset of the end-time day from the start time.
«cntd»	***timeDemandTypes***	*TimeDemandRef*	0:1	TIME DEMAND TYPEs for which this FREQUENCY GROUP applies.

«cntd»	***journeys***	*VehicleJourneyRef*	1:*	Journeys belonging to FREQUENCY GROUP.

10.7.3 HeadwayJourneyGroup (Subcomponent of TemplateJourney)

A HEADWAY JOURNEY GROUP is a group of VEHICLE JOURNEYs following the same JOURNEY PATTERN and having the same headway interval between a specified start and end time (for example, '*every 10 minutes*').

Table 107 — *HeadwayJourneyGroup* – XML Element

Classification	Name	Type	Cardinality	Description
::>	*::>*	*JourneyFrequencyGroup*	*::>*	HEADWAY JOURNEY GROUP inherits from JOURNEY FREQUENCY GROUP.
«PK»	***id***	*HeadwayJourneyGroupIdType*	0:1	Identifier of HEADWAY JOURNEY GROUP.
	ScheduledHeadway-Interval	*xsd:duration*	0:1	Scheduled normal HEADWAY INTERVAL.
	MinimumHeadway-Interval	*xsd:duration*	0:1	Minimum HEADWAY INTERVAL.
	MaximumHeadwa-yInterval	*xsd:duration*	0:1	Maximum HEADWAY INTERVAL.
	HeadwayDisplay	*HeadwayUseEnum*	0:1	How headway is to be displayed to passengers.
	Description	*MultilingualString*	0:1	Description of Headway Journey Groups.

10.7.3.1 RhythmicalJourneyGroup – Model Element

A group of VEHICLE JOURNEYs following the same JOURNEY PATTERN having the same "rhythm" every hour (for example, '*runs at xxh10, xxh25 and xxh45 past the hour*') between a specified start and end time.

NB☞Only in EIRE_NP

Table 108 — Table 19 – *RhythmicalJourneyGroup* – Element

Classification	Name	Type	Cardinality	Description
::>	*::>*	*JourneyFrequencyGroup*	*::>*	RHYTHMICAL JOURNEY GROUP inherits from JOURNEY FREQUENCY GROUP.
«PK»	***id***	*RhythmicalJourneyGroup-IdType*	0:1	Identifier of RHYTHMICAL JOURNEY GROUP.
«cntd»	***timebands***	*Timeband*	0:*	Timebands describing RHYTHMICAL FREQUENCY GROUP. Each specifies some minutes past the hour.

10.8 Timing Information

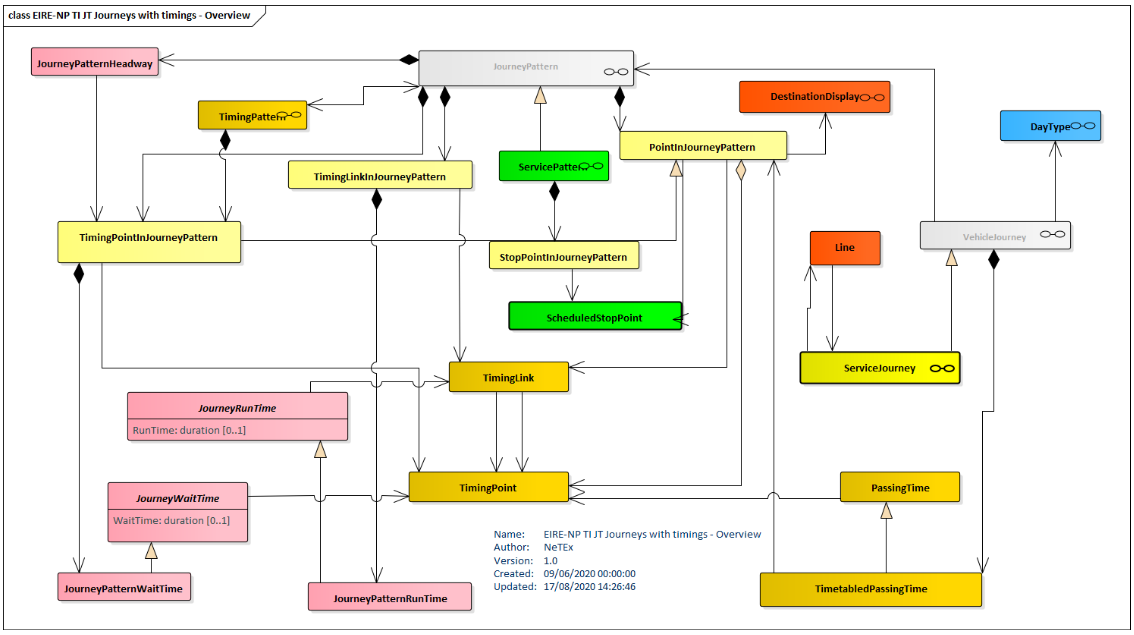
The EPIP profile excludes timing information from its scope. That is, the run times and wait times used to compute PASSING TIMEs for a given SERVICE JOURNEY are not included.

Such run times are useful for certain use cases, for example real-time stop prediction and for schedule creation, so they are permitted as augmentations in the Irish Profile.

In order to include timing information, RUN TIMEs (associated with a SERVICE LINK or TIMING LINK) and WAIT TIME elements (associated with a POINT IN PATTERN) are used.

10.8.1 Timings on a JOURNEY PATTERN

JOURNEY PATTERN RUN TIME and JOURNEY PATTERN WAIT time elements may be used to set defaults for a given SERVICE PATTERN. There may be different sets of RUN TIMEs and WAIT TIMEs for different TIME DEMAND TYPEs (e.g., peak, off-peak).



10.8.2 JourneyTime (Abstract)

A PASSING TIME associates a time with a POINT IN JOURNEY PATTERN. It is specialised to create the different concrete times (TIMETABLED PASSING TIME, etc).

Table 109 — *PassingTime* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	PASSING TIME inherits from VERSIONED CHILD.
«PK»	***id***	*PassingTimeIdType*	0:1	Identifier of PASSING TIME.
«FK»	***JourneyRef***	*JourneyRef*	0:1	Reference to JOURNEY for which this is a PASSING TIME.
	AlightAndReboard	*xsd:boolean*	1:1	Number of days that the departure day is after the start day of vehicle journey.
«FK»	***PointInJourney-PatternRef***	*PointInLinkSequenceRef*	0:1	POINT IN SEQUENCE for which this is a PASSING TIME. Refer to the SCHEDULE STOP POINT

10.8.3 JourneyRunTime (Abstract)t

The time taken to traverse a TIMING LINK in a particular JOURNEY PATTERN, for a specified TIME DEMAND TYPE. If it exists, it will override the DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.

Table 110 — Table 20 – *JourneyRunTime* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*JourneyTiming*	::>	JOURNEY RUN TIME inherits from JOURNEY TIMING.
«PK»	***id***	*RunTimeIdType*	1:1	Identifier of JOURNEY RUN TIME.
«FK»	***TimingLinkRef***	*TimingLinkRef*	0:1	TIMING LINK associated with timing.
	RunTime	*xsd:duration*	1:1	Run time of JOURNEY RUN TIME.

10.8.4 JourneyPatternRunTime (Subcomponent – Service Pattern)

The JOURNEY PATTERN RUN TIME is the time taken to traverse a TIMING LINK in a particular JOURNEY PATTERN, for a specified TIME DEMAND TYPE. If it exists, it will override the DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.

Table 111 — Table 21 – *JourneyPatternRunTime* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*JourneyRunTime*	::>	JOURNEY PATTERN RUN TIME inherits from JOURNEY RUN TIME.
«PK»	***id***	*JourneyPatternRunTimeIdType*	0:1	Identifier of JOURNEY PATTERN RUN TIME.
«FK»	***LinkRef***	*TimingLinkRef*	1:1	TIMING LINK with which TURNAROUND TIME is associated.
«FK»	***JourneyPatternRef***	*JourneyPatternRef*	1:1	JOURNEY PATTERN with which TURNAROUND TIME is associated.

10.8.5 JourneyWaitTime (Abstract)

The time a vehicle has to wait at a specific TIMING POINT IN JOURNEY PATTERN, for a specified TIME DEMAND TYPE. This wait time can be superseded by a VEHICLE JOURNEY WAIT TIME.

Table 112 — – *JourneyWaitTime* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*JourneyTiming*	::>	JOURNEY WAIT TIME inherits from JOURNEY TIMING.
«PK»	***id***	*WaitTimeIdType*	1:1	Identifier of JOURNEY WAIT TIME.
«FK»	***TimingPointRef***	*TimingPointRef*	1:1	POINT associated with of JOURNEY WAIT TIME.
	WaitTime	*xsd:duration*	1:1	Wait time of JOURNEY WAIT TIME.

10.8.6 JourneyPatternWaitTime (Subcomponent – Service Pattern)

The JOURNEY PATTERN WAIT TIME represents the time a vehicle has to wait at a specific TIMING POINT IN JOURNEY PATTERN, for a specified TIME DEMAND TYPE. This wait time can be superseded by a VEHICLE JOURNEY WAIT TIME.

Table 113 — Table 22 – *JourneyPatternWaitTime* – Element

Classification	Name	Type	Cardinality	Description

::>	::>	*JourneyWaitTime*	::>	JOURNEY PATTERN WAIT TIME inherits from JOURNEY WAIT TIME.
«PK»	***id***	*JourneyPatternWaitTimeIdType*	0:1	Identifier of JOURNEY PATTERN WAIT TIME.
«FK»	***JourneyRef***	*JourneyPatternRef*	1:1	JOURNEY PATTERN with which WAIT TIME is associated.

10.8.7 JourneyHeadway (Abstract)

Headway frequency information that is available for all the VEHICLE JOURNEY running on the JOURNEY PATTERN for a given TIME DEMAND TYPE, for the journey or at a run at a given TIMING POINT.

This is a default value that can be superseded by VEHICLE JOURNEY HEADWAY. This information shall be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services).

Table 114 — Table 23 – *JourneyHeadway* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*JourneyTiming*	::>	JOURNEY HEADWAY inherits from JOURNEY TIMING.
«PK»	***id***	*HeadwayIdType*	1:1	Identifier of JOURNEY HEADWAY.
	HeadwayInterval-Group	*HeadwayIntervalGroup*	1:1	Set of HEADWAY INTERVALs describing frequency of JOURNEY.

10.8.8 JourneyPatternHeadway (Subcomponent – Service Pattern)

The JOURNEY PATTERN HEADWAY is the headway frequency information that is available for all the VEHICLE JOURNEYs running on the JOURNEY PATTERN at the different TIMING POINTs IN JOURNEY PATTERN. This is a default value that can be superseded by the VEHICLE JOURNEY HEADWAY on a specific journey.

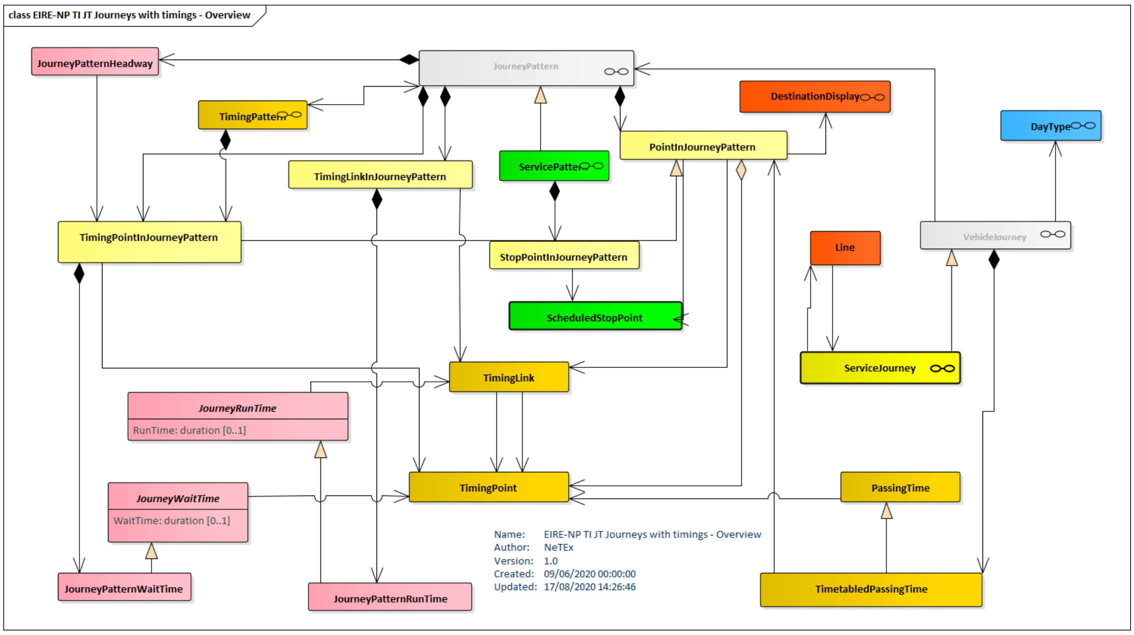
This information shall be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services).

Table 115 — Table 24 – *JourneyPatternHeadway* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*TimeDemandHeadway*	::>	JOURNEY PATTERN HEADWAY inherits from TIME DEMAND HEADWAY.
«PK»	***id***	*JourneyPatternHeadwayIdType*	0:1	Identifier of JOURNEY PATTERN HEADWAY.
«FK»	***JourneyRef***	*JourneyPatternRef*	1:1	JOURNEY PATTERN with which HEADWAY is associated.

10.8.9 Timings on a SERVICE JOURNEY

If timings are specific to a journey, VEHICLE JOURNEY RUN TIME and VEHICLE JOURNEY WAIT time elements may be used to override any SERVICE PATTERN values for a given SERVICE JOURNEY. The PASSING TIMES may be computed from these.



Name: EIRE-NP TI JT Journeys with timings - Overview
Author: NeTEx
Version: 1.0
Created: 09/06/2020 00:00:00
Updated: 17/08/2020 14:26:46

Figure 25 — Run times and wait times for a service pattern (UML)

10.8.9.1.1 VehicleJourneyRunTime – (Subcomponent – Service Journey)The VEHICLE JOURNEY RUN TIME is the time taken to traverse a specified TIMING LINK IN JOURNEY PATTERN on a specified VEHICLE JOURNEY. This gives the most detailed control over times and overrides the DEFAULT SERVICE JOURNEY RUN TIME and JOURNEY PATTERN RUN TIME and the DEFAULT DEAD RUN RUN TIME.

Table 116 — Table 25 – *VehicleJourneyRunTime* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*JourneyRunTime*	::>	VEHICLE JOURNEY RUN TIME inherits from JOURNEY RUN TIME.
«PK»	***id***	*VehicleJourneyRunTimeIdType*	0:1	Identifier of VEHICLE JOURNEY RUN TIME.
«FK»	***VehicleJourneyRef***	*VehicleJourneyRef*	0:1	VEHICLE JOURNEY for which this is a RUN TIME.

10.8.9.1.2 VehicleJourneyWaitTime – (Subcomponent – Service Journey)

The VEHICLE JOURNEY WAIT TIME is the time for a vehicle to wait at a particular TIMING POINT IN JOURNEY PATTERN on a specified VEHICLE JOURNEY. This wait time will override the JOURNEY PATTERN WAIT TIME.

Table 117 — Table 26 – *VehicleJourneyWaitTime* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*JourneyWaitTime*	::>	VEHICLE JOURNEY WAIT TIME inherits from JOURNEY WAIT TIME.
«PK»	***id***	*VehicleJourneyWaitTimeIdType*	0:1	Identifier of VEHICLE JOURNEY WAIT TIME.

«FK»	***VehicleJourneyRef***	*VehicleJourneyRef*	0:1	Reference to a VEHICLE JOURNEY with which this WAIT TIME is associated.

10.9 Transferring between services

If there is no direct SERVICE JOURNEY between the origin and the destination SCHEDULED STOP POINTs of a trip, a passenger will have to interchange between vehicles. The passenger will alight from the vehicle making a SERVICE JOURNEY at a particular SCHEDULED STOP POINT and transfer to board another vehicle (performing another SERVICE JOURNEY, usually on a different LINE) at the same or another SCHEDULED STOP POINT.

The transfer takes place at a CONNECTION. A CONNECTION indicates that two specified SCHEDULED STOP POINTs are suitably located for it to be possible for a passenger to walk between them when making an interchange from one public transport vehicle to another; it can also allocate specific time for a passenger to traverse the link (if note given, these may be defaulted from a DEFAULT CONNECTION or SITE CONNECTION). It is independent of any specific SERVICE JOURNEY.

If two specific SERVICE JOURNEYs are intended to coincide at a possible CONNECTION point, this can be expressed by a SERVICE JOURNEY INTERCHANGE.

Each SERVICE JOURNEY INTERCHANGE involves two different SERVICE JOURNEYs. The passenger makes the transfer from a *feeder* SERVICE JOURNEY to a *distributor* (or “fetcher”) SERVICE JOURNEY.

Sometimes a SERVICE JOURNEY INTERCHANGE expresses an “interchange” between two SERVICE JOURNEYs specifically planned to be operated by the same physical vehicle. This concept is for instance used for circular lines and coupled journeys. This means that passenger information should indicate that the passenger does not need to change vehicle as the transfer is implicit (see the ***StaySeated*** attribute below).

10.9.1 Interchange (Abstract)

An INTERCHANGE specifies common properties of interchanges. including whether it is advertised or not, if it is guaranteed or not and the maximum time a vehicle may wait for a connecting vehicle, beyond the planned departure time. It is specialised to define the SERVICE JOURNEY INTERCHANGE.

Table 118 — *Interchange* – XML Element

Classification	Name	Type		Cardinality	Description
::>	::>	*DataManagedObject*		::>	INTERCHANGE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*InterchangeIdType*		1:1	Identifier of INTERCHANGE.
	Name	*MultilingualString*		0:1	Name of INTERCHANGE.
	Description	*MultilingualString*		0:1	Description of INTERCHANGE.
«AK»	***PrivateCode***	*xsd:normalizedString*		0:1	Alternative key for INTERCHANGE.
«AK»	***ExternalInterchangeRef***	*ExternalObjectRef*		0:1	An alternative code that uniquely identifies the INTERCHANGE specifically for use in AVMS systems. *NOTE For VDV compatibility.*

«FK»	***ConnectionRef***	*ConnectionRef*		0:1	CONNECTION link over which the interchange place.
	Priority	*InterchangePriorityType*		0:1	Priority assigned to INTERCHANGE.
	StaySeated	*xsd:boolean*		0:1	Whether passengers can stay seated to make INTERCHANGE.
	CrossBorder	*xsd:boolean*		0:1	Whether INTERCHANGE involves crossing a na border.
	Planned	*xsd:boolean*		0:1	Whether INTERCHANGE is planned.
	Guaranteed	*ConnectionCertaintyEnum*		0:1	Whether INTERCHANGE is guaranteed, that is services may be held in order to ensure the co
	Advertised	*xsd:boolean*		0:1	Whether INTERCHANGE is advertised to the pu
	Controlled	*xsd:boolean*		0:1	Whether INTERCHANGE is controlled.
	StandardWaitTime	*xsd:duration*		0:1	Standard Time to wait at Interchange.
	MaximumWaitTime	*xsd:duration*		0:1	Maximum time that DISTRIBUTOR will wait aft planned departure time.
	MaximumAutomaticWaitTime	*xsd:duration*		0:1	Maximum automatic wait time for INTERCHAN
	StandardTransfer-Time	*xsd:duration*		0:1	Standard Time needed for transfer.
	MinimumTransfer-Time	*xsd:duration*		0:1	Minimum Time needed for transfer.
	MaximumTransfer-Time	*xsd:duration*		0:1	Maximum Time needed for transfer.
	ControlCentreNotify-Threshold	*xsd:duration*		0:1	Interval before CONTROL CENTRE should be no associated with SERVICE JOURNEY INTERCHAN
	TransferModes	*AccessModeEnum*		0:1	Out of vehicle TRANSPORT MODEs by which tr the interchange can be made. See Reusable co
«cntd»	***noticeAssignments***	*NoticeAssignmentView*		0:*	NOTICE ASSIGNMENTs that apply to the INTER

10.9.2 ServiceJourneyInterchange

A SERVICE JOURNEY INTERCHANGE specialises the INTERCHANGE entity; as well as having the common properties of an INTERCHANGE, it specifies at which POINTs in PATTERN of the respective feeder and fetcher SERVICE JOURNEYs the interchange takes place. Since the same SERVICE JOURNEY may visit the same stop more than once, an interchange relates to a POINT IN PATTERN rather than just a SCHEDULED STOP POINT.

Table 119 — *ServiceJourneyInterchange* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*Interchange*	::>	SERVICE JOURNEY INTERCHANGE inherits from INTERCHANGE.
«PK»	***id***	*ServiceJourneyInterchange-IdType*	1:1	Identifier of SERVICE JOURNEY INTERCHANGE.
«FK»	***FromPointRef***	*ScheduledStopPointRef*	1:1	SCHEDULED STOP POINT from which the interchange starts.

	FromVisitNumber	*xsd:integer*	0:1	Visit number of feeder journey (only needed if multiple visits).
«FK»	***ToPointRef***	*ScheduledStopPointRef*	1:1	SCHEDULED STOP POINT to which the interchange goes.
	ToVisitNumber	*xsd:integer*	0:1	Visit number of distributor journey (only needed if multiple visits).
«FK»	***FromJourneyRef***	*ServiceJourneyRef*	1:1	Reference to feeder SERVICE JOURNEY from which interchange is made.
«FK»	***ToJourneyRef***	*ServiceJourneyRef*	1:1	Reference to distributor SERVICE JOURNEY to which interchange is made.
«FK»	***ServiceJourney-PatternInterchange-Ref***	*ServiceJourneyPattern-InterchangeRef*	0:1	Reference to SERVICE JOURNEY PATTERN INTERCHANGE for which interchange is made.

10.10 Flexible Services

NeTEx can also be used to define Flexible services that do not necessarily follow a fixed route, but can operate over one or more areas area identified as FLEXIBLE STOP PLACEs. A FLEXIBLE SERVICE PROPERTIES element describes the properties of a specific service; a FLEXIBLE LINE describes common properties of the LINE.

Stop specific properties can be specified using a FLEXIBLE POINT PROPERTIES ELEMENT.

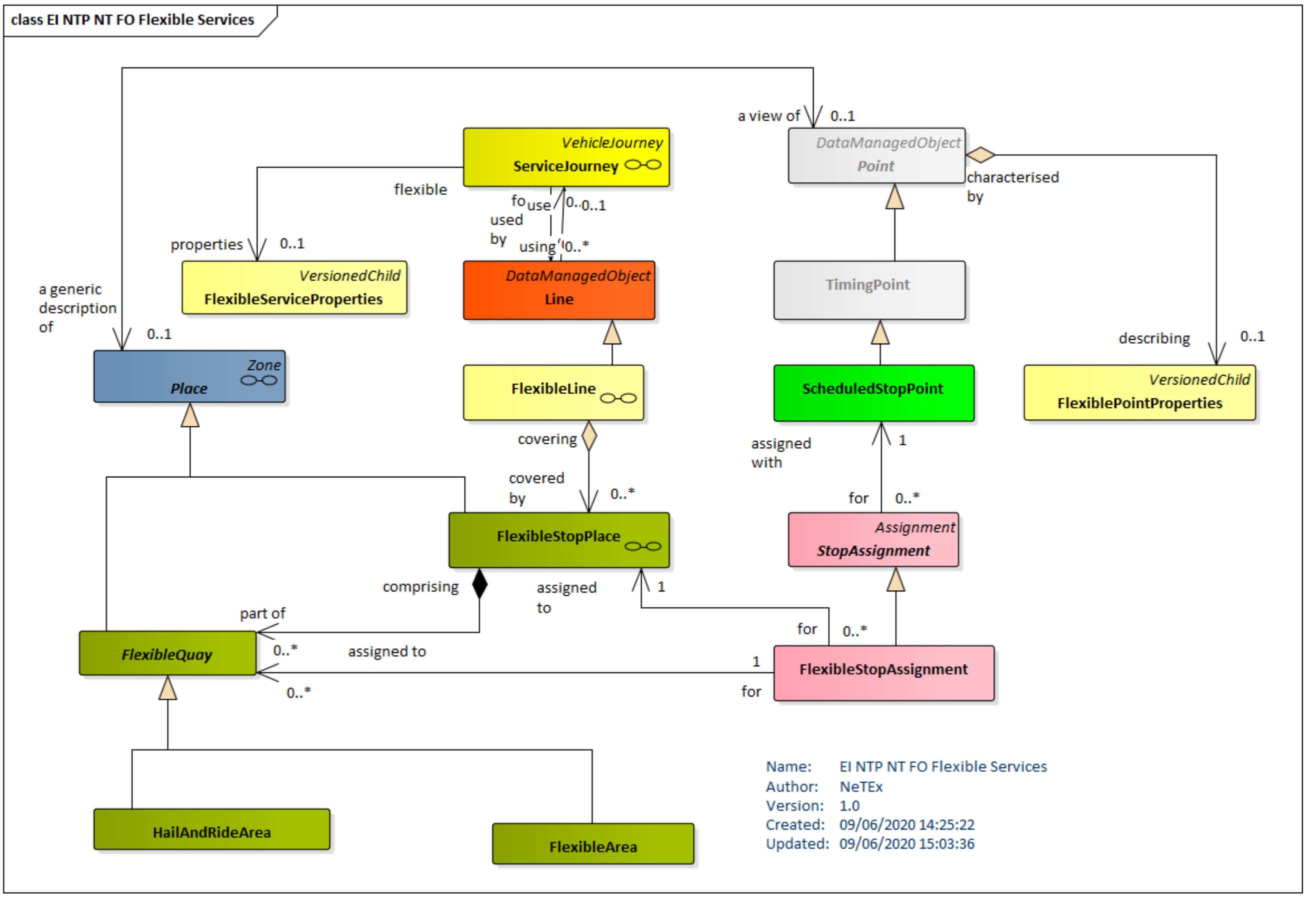


Figure 26 — Flexible Stops & Services (UML)

10.10.1FlexibleServiceProperties (Subcomponent of SERVICE JOURNEY)

FLEXIBLE SERVICE PROPERTIES describe additional properties of a FLEXIBLE or demand responsive (DRT) service. They can be attached to a SERVICE JOURNEY to make it flexible.

In NeTEx, A SERVICE JOURNEY may either embed its own FLEXIBLE SERVICE PROPERTIES, or reference a shared instance. EPIP requires each SERVICE JOURNEY to embed its own instance.

Table 120 — FlexibleServiceProperties – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	FLEXIBLE SERVICE PROPERTIES inherits from DATA MANAGED OBJECT.
«PK»	***id***	*FlexibleServiceProperties-IdType*	1:1	Identifier of FLEXIBLE SERVICE PROPERTIES.
«FK»	***TypeOfFlexible-ServiceRef***	*TypeOfFlexibleServiceRef*	1:1	TYPE OF FLEXIBLE SERVICE for which these are the properties.
«cntd»	***Booking-Arrangements***	*BookingArrangements*	0:1	Booking contact for FLEXIBLE SERVICE.
«enum»	***FlexibleServiceType***	*FlexibleServiceEnum*	0:1	Flexible service type. One of the following: • *fixedPassingTimes* • *dynamicPassingTimes* • *fixedHeadwayFrequency* • *notFlexible*
	CancellationPossible	*xsd:boolean*	0:1	Whether cancellation is always possible (meaning the Operator can decide to cancel a journey, usually because there are not enough passengers, or they are too busy to run the service.)
	ChangeOfTime-Possible	*xsd:boolean*	0:1	Flag saying that it may be moved (usually passing time update to optimise the service).

10.10.2FlexibleLine

A LINE that is operated flexibly (either to a flexible schedule or over a flexible area, or both).

FLEXIBLE LINE specialises LINE.

Table 121 — *FlexibleLine – XML Element*

Classification	Name	Type	Cardinality	Description
«PK»	***id***	*BookableLineIdType*	1:1	Identifier of FLEXIBLE LINE.
«enum»	***FlexibleLineType***	*FlexibleLineTypeEnum*	0:1	Type of FLEXIBLE LINE. • *corridorService; flexibleAreasOnly* • *hailAndRideSections; fixedStopAreaWide; freeAreaAreaWide; mixedFlexible; mixedFlexibleAndFixed; fixed; other*
«cntd»	***BookingArrangements***	*BookingArrangements*	0:1	Booking arrangements for FLEXIBLE LINE.

10.11 Trains and train coupling

A full representation of rail services must address additional complexities not found in other modes, such as trains that join or split, and TRAIN NUMBERs that are distinct from the identifiers of SERVICE JOURNEYs and may change over the course of the journey.

EPIP includes in its scope only a minimal representation concerned with the labelling of journeys and JOURNEY PARTs. It also allows the availability of FACILITies to be limited to a JOURNEY PART.

NB☞ These elements are not needed for the EIRE_NP bus Timetable data set.

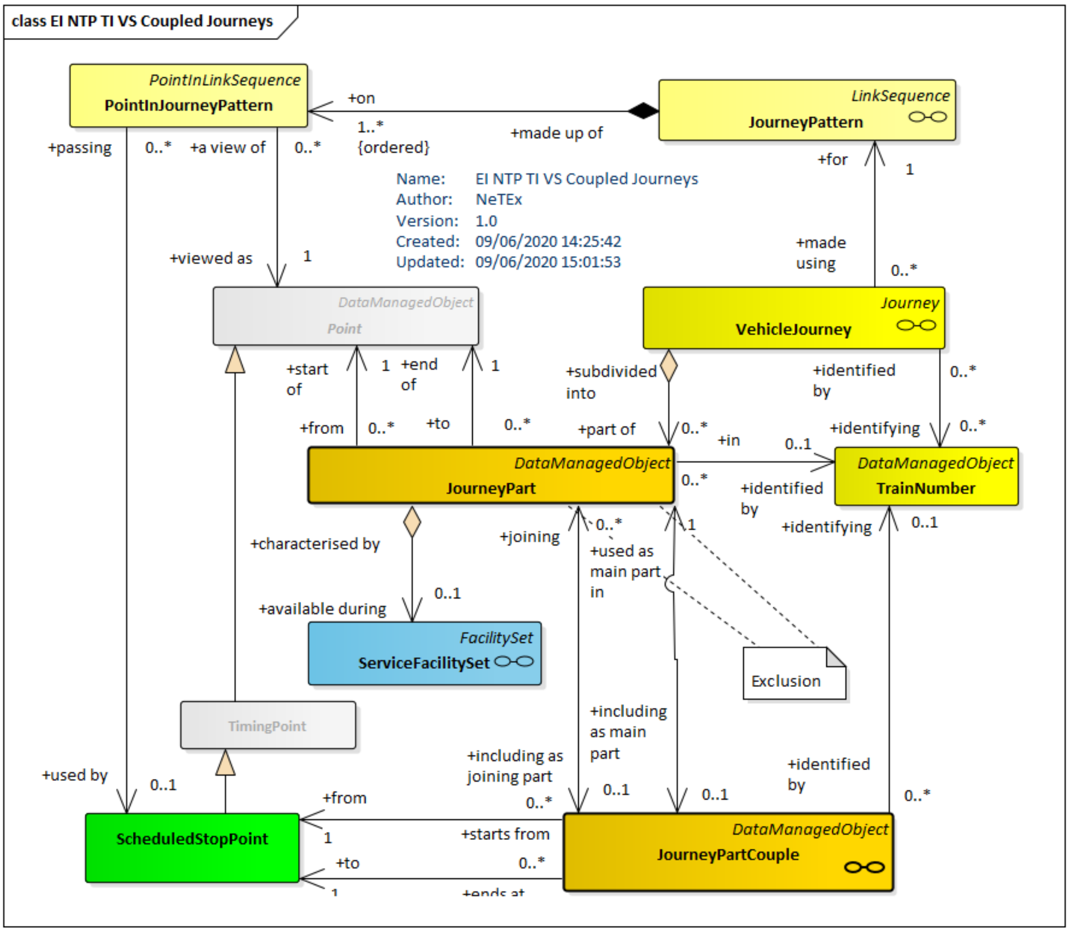


Figure 27 — Train Coupling (UML)

A SERVICE JOURNEY may be broken down into a number of JOURNEY PARTs, each with its own TRAIN NUMBER. JOURNEY PARTs from different SERVICE JOURNEYs may be coupled together with a JOURNEY PART COUPLE to indicate that they are joined in a single train for that section of their running. The same TRAIN NUMBER may be given to both SERVICE JOURNEYs by assigning it to the JOURNEY PART COUPLE.

10.11.1TrainNumber

The TRAIN NUMBER provides the specification of codes assigned to particular VEHICLE JOURNEYs when operated by TRAINs of COMPOUND TRAINs according to a functional purpose (passenger information, operation follow-up, etc.).

Table 122 — *TrainNumber* – XML Element

Classification	Name	Type	Cardinality	Description

::>	::>	*DataManagedObject*	::>	TRAIN NUMBER inherits from DATA MANAGED OBJECT
«PK»	***id***	*TrainNumberIdType*	1:1	Identifier of TRAIN NUMBER.
	Description	*MultilingualString*	0:1	Description of TRAIN NUMBER.
	ForAdvertisement	*xsd:normalizedString*	0:1	TRAIN NUMBER to use for advertisement to public if different from ID.
	ForProduction	*xsd:normalizedString*	0:1	TRAIN NUMBER to use for production purposes, for instance towards technical systems that require an odd or even value according to safety regulations if different from ID.

10.11.2JourneyPart (Subcomponent of ServiceJourney)

A JOURNEY PART is a part of a VEHICLE JOURNEY created to distinguish a specific functional purpose, for instance in situations when vehicle coupling or separating occurs. A JOURNEY PART can be demarcated by a start and end SCHEDULED STOP POINT. (NB A given stop may be visited only once within a JOURNEY PART).

In the EPIP, the JOURNEY PARTs should be embedded in-line with their parent SERVICE JOURNEY.

Table 123 — *JourneyPart* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	JOURNEY PART inherits from DATA MANAGED OBJECT.
«PK»	***id***	*JourneyPartIdType*	1:1	Identifier of JOURNEY PART.
	Description	*MultilingualString*	0:1	Description of a JOURNEY PART.
«FK»	***ParentJourneyRef***	*VehicleJourneyRef*	0:1	Parent VEHICLE JOURNEY to which JOURNEY PART belongs.
«FK»	***MainPartRef***	*JourneyPartCoupleRef*	1:1	Reference to JOURNEY PART COUPLE of the main JOURNEY of a JOURNEY PART.
«FK»	***JourneyPart-CoupleRef***	*JourneyPartCoupleRef*	0:1	Reference to JOURNEY PART COUPLE of the JOURNEY that JOURNEY PART joins.
«FK»	***TrainNumberRef***	*TrainNumberRef*	0:1	Reference to TRAIN NUMBER of JOURNEY PART
«FK»	***BlockPartRef***	*BlockPartRef*	0:1	REFERENCE to BLOCK PART of the JOURNEY that the JOURNEY PART joins.
«FK»	***FromStopPointRef***	*ScheduledStopPointRef*	0:1	Reference to from start SCHEDULED STOP POINT for JOURNEY PART.
«FK»	***ToStopPointRef***	*ScheduledStopPointRef*	0:1	Reference to end SCHEDULED STOP POINT for JOURNEY PART.
	StartTime	*xsd:time*	1:1	Start time of a JOURNEY PART.
	StartTimeDayOffset	*DayOffsetType*	0:1	Number of days after journey start time that start time is.
	EndTime	*xsd:time*	1:1	End time of a JOURNEY PART.
	EndTimeDayOffset	*DayOffsetType*	0:1	Number of days after journey start time that end time is.
	VehicleOrientation	*xsd:boolean*	0:1	Orientation of the vehicle carrying out the JOURNEY PART compared to the definition of the corresponding VEHICLE TYPE. True for forward.

«FK»	***PurposeOfJourney-PartitionRef***	*PurposeOfJourney-PartitionRef*	0:1	PURPOSE of JOURNEY PARTITION.
«cntd»	***facilities***	*ServiceFacilitySet*	0:*	Facilities available during a JOURNEY PART.
«cntd»	***journeyPartPositions***	*JourneyPartPosition*	0:*	Positions in Train of JOURNEY PART

10.11.3JourneyPartCouple

A JOURNEY PART COUPLE defines two or more JOURNEY PARTs of different VEHICLE JOURNEYs served simultaneously by a TRAIN set up by coupling their single vehicles.

As such it represents a “joined” train that has two SERVICE JOURNEYs coupled together for a section of their journey.

Table 124 — *JourneyPartCouple* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	JOURNEY PART COUPLE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*JourneyPartCoupleIdType*	1:1	Identifier of JOURNEY PART COUPLE.
	Description	*MultilingualString*	0:1	Description of a JOURNEY PART COUPLE.
	StartTime	*xsd:time*	1:1	Start time of a JOURNEY PART COUPLE.
	StartTimeDayOffset	*DayOffsetType*	0:1	Number of days after journey start time that start time is.
	EndTime	*xsd:time*	1:1	End time of a JOURNEY PART COUPLE.
	EndTimeDayOffset	*DayOffsetType*	0:1	Number of days after journey start time that end time is.
«FK»	***FromStopPointRef***	*ScheduledStopPointRef*	1:1	Reference to from start SCHEDULED STOP POINT for JOURNEY PART COUPLE.
«FK»	***ToStopPointRef***	*ScheduledStopPointRef*	1:1	Reference to end SCHEDULED STOP POINT for JOURNEY PART COUPLE.
«FK»	***MainPartRef***	*JourneyPartCoupleRef*	1:1	Reference to JOURNEY PART COUPLE of the main JOURNEY of a JOURNEY PART COUPLE.
«cntd»	***journeyParts***	*JourneyPartRef*	0:*	References to JOURNEY PARTs of JOURNEY that JOURNEY PART joins.
«FK»	***TrainNumberRef***	*TrainNumberRef*	0:1	Reference to the TRAIN NUMBER of JOURNEY PART.

10.11.4JourneyPartPosition (Subcomponent of JourneyPartCouple)

JOURNEY PART POSITION describes the relative position in the train of a JOURNEY PART from a given stop onwards. It may change in the course of the journey as train components are coupled and uncoupled.

Table 125 — Table 27 – *JourneyPartPosition*– XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	JOURNEY PART POSITION inherits from VERSIONED CHILD.
«PK»	***id***	*JourneyPartPositionIdType*	1:1	Identifier of JOURNEY PART POSITION.
«FK»	***ParentJourneyPart-Ref***	*ParentJourneyPartRef*	0:1	Parent of JOURNEY PART in which the position of the part is defined. Given by context.

«FK»	***ScheduledStopPoint-Ref***	*ScheduledStopPointRef*	0:1	SCHEDULED STOP POINT from which the position is valid.
	PositionInTrain	*xsd:integer*	0:*	Position of JOURNEY PART in TRAIN - from specified SCHEDULED STOP POINT until otherwise stated.

10.12 Vehicles and Trains

The NeTEx Vehicle and Train models (see Stop Assignment diagram earlier) allow properties of transport vehicles and the make-up of trains to be described. In the EPIP these are used to provide additional information about SERVICE JOURNEYs.

10.12.1VehicleType

VEHICLEs may be classified using a VEHICLE TYPE according to how they meet vehicle scheduling requirements as to capacity and on-board facilities (e.g., *standard bus, double decker,* etc.). A VEHICLE TYPE may be attached to a specific SERVICE JOURNEY to indicate that a vehicle of that type is needed to satisfy the requirements.

The VEHICLE TYPE is of particular interest to passengers with impaired mobility who may be limited to vehicles with particular accessibility features such as a wheelchair hoist.

Table 126 — *VehicleType – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	VEHICLE TYPE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*VehicleTypeIdType*	1:1	Identifier of VEHICLE TYPE.
	Name	*MultilingualString*	0:1	Name of VEHICLE TYPE.
	ShortName	*MultilingualString*	0:1	Short Name of DAY TYPE.
	Description	*MultilingualString*	0:1	Description of VEHICLE TYPE.
«AK»	***PrivateCode***	*PrivateCode*	0:1	Alternative Identifier for VEHICLE TYPE.
	ReversingDirection	*xsd:boolean*	0:1	Whether VEHICLE TYPE has a reversing direction.
	SelfPropelled	*xsd:boolean*	0:1	Whether VEHICLE TYPE is self-propelled.
«enum»	***TypeOfFuel***	*TypeOfFuelEnum*	0:1	Type of Fuel of VEHICLE TYPE.
	EuroClass	*xsd:normalizedString*	0:1	Euroclass of the vehicle type.
«cntd»	***PassengerCapacity***	*PassengerCapacity*	0:1	Total Passenger carrying capacity of VEHICLE TYPE. Use capacities
«cntd»	***capacities***	PassengerCapacity	0:*	Passenger carrying capacity of VEHICLE TYPE by FARE CLASS.
	LowFloor	*xsd:boolean*	0:1	Whether VEHICLE TYPE is low floor.
	HasLiftOrRamp	*xsd:boolean*	0:1	Whether VEHICLE TYPE has lift or ramp for wheelchair access.
	HasHoist	*xsd:boolean*	0:1	Whether vehicle has hoist for wheelchair access.
	Length	*LengthType*	0:1	Length of a VEHICLE of that TYPE.
	Width	*LengthType*	0:1	The width of a VEHICLE of the type.
	Height	*LengthType*	0:1	The height of a VEHICLE of the type.
	Weight	*WeightType*	0:1	The weight of a VEHICLE of the type.

«FK»	***IncludedIn***	*VehicleTypeRef*	0:1	Included in definition of VEHICLE TYPE.
«FK»	***ClassifiedAsRef***	*VehicleModelRef*	0:1	Classification of type as being of a particular VEHICLE MODEL.
«cntd»	***facilities***	*FacilitySet*	0:*	Actual facilities on VEHICLE
«cntd»	***canCarry***	*PassengerCarrying-RequirementRef*	0: *	Passenger carrying requirements satisfied by VEHICLE TYPE.
«cntd»	***canManoeuvre***	*ManoeuvringRequirementRef*	0:*	Manoeuvring requirements satisfied by VEHICLE TYPE.
«cntd»	***satisfiesFacilities***	*FacilityRequirement*	0:*	Facility requirements satisfied by VEHICLE TYPE.

10.12.2Train

A TRAIN is a vehicle composed of TRAIN ELEMENTs in a certain order, i.e., of wagons assembled together and propelled by a locomotive or one of the wagons.

Some rail passenger information systems give detailed information on train make-up and the relative location of carriages within the train so that passengers can board efficiently at their specified carriage, get on the right part of the train that will be splitting later in its journey, find the restaurant car or the toilets for the disabled, etc.

Table 127 — *Train – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VehicleType*	::>	TRAIN inherits from VEHICLE TYPE
«PK»	***id***	*TrainIdType*	1:1	Identifier of TRAIN.
	NumberOfCars	*xsd:nonNegativeInteger*	0:1	Number of cars in TRAIN.
«enum»	***TrainSizeType***	*TrainSizeEnum*	0:1	Categorisation of TRAIN SIZE. • *normal* • *short* • *long*
«cntd»	***components***	*TrainComponent*	0:*	Components of TRAIN.

10.12.3TrainElement (Subcomponent of Train)

A TRAIN ELEMENT is an elementary component of a TRAIN (e.g., wagon, locomotive).

Table 128 — *TrainElement – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	TRAIN ELEMENT inherits from DATA MANAGED OBJECT.
	id	*TrainElementIdType*	1:1	Identifier of TRAIN ELEMENT.
	Name	*MultilingualString*	0:1	Name of TRAIN ELEMENT.
	Description	*MultilingualString*	0:1	Description of TRAIN ELEMENT.
«enum»	***TrainElementType***	*TypeOfTrainElementEnum*	1:1	Classification of TRAIN ELEMENT. • *carriage; engine; carTransporter; sleeperCarriage; restaurantCarriage; luggageVan*
	FareClasses	*FareClassEnum*	0:*	FARE CLASSes for TRAIN ELEMENT. See Annex.

«cntd»	***PassengerCapacity***	*PassengerCapacity*	0:1	Capacity of TRAIN ELEMENT.
	Length	*LengthType*	0:1	Length of TRAIN ELEMENT.
	Width	*LengthType*	0:1	The width of a TRAIN ELEMENT.
«cntd»	***facilities***	ServiceFacilitySet	0:*	FACILITies of TRAIN ELEMENT.
«cntd»	***equipments***	*Equipment*	0:*	Equipment of TRAIN ELEMENT.

10.12.4TrainComponent (Subcomponent of Train)

A TRAIN COMPONENT is the specification of the order of TRAIN ELEMENTs in a TRAIN.

In EPIP, TRAIN ELEMENTs should be embedded in line within a TRAIN COMPONENT.

Table 129 — *TrainComponent – XML Element*

Classification	Name		Type	Cardinality	Description
::>	::>		*VersionedChild*	::>	TRAIN COMPONENT inherits from VERSIONED CHILD.
«PK»	***id***		*TrainComponentIdType*	1:1	Identifier of TRAIN COMPONENT.
	Label		*MultilingualString*	0:1	Label of TRAIN COMPONENT.
	Description		*MultilingualString*	0:1	Description of TRAIN COMPONENT.
«FK»	***TrainRef***		*TrainRef*	0:1	Reference to TRAIN of which this is a part.
«FK»	a	***TrainElementRef***	*TrainElementRef*	1:1	Reference to TRAIN ELEMENT associated with TRAIN COMPONENT.
«cntd»	b	***TrainElement***	*TrainElement*	1:1	TRAIN ELEMENT associated with TRAIN COMPONENT.

10.13 Notices

The NOTICE Model defines reusable text note elements that may be attached to timetables and other ENTITies as footnotes, used as announcements, etc. NOTICEs are associated with entities using a NOTICE ASSIGNMENT. NOTICEs may be classified with a TYPE OF NOTICE.

Each NOTICE may have several alternative formats as specified by a DELIVERY VARIANT, e.g., for displays, print, audio, use in a mobile app, etc.

NOTICEs are actually part of the NeTEx core framework and may be associated with any ENTITY, but are described here in the timetable part of the profile as their main use in the profile is to provide footnotes for SERVICE JOURNEYs, INTERCHANGEs, GROUPs of SERVICE and other timetable elements.

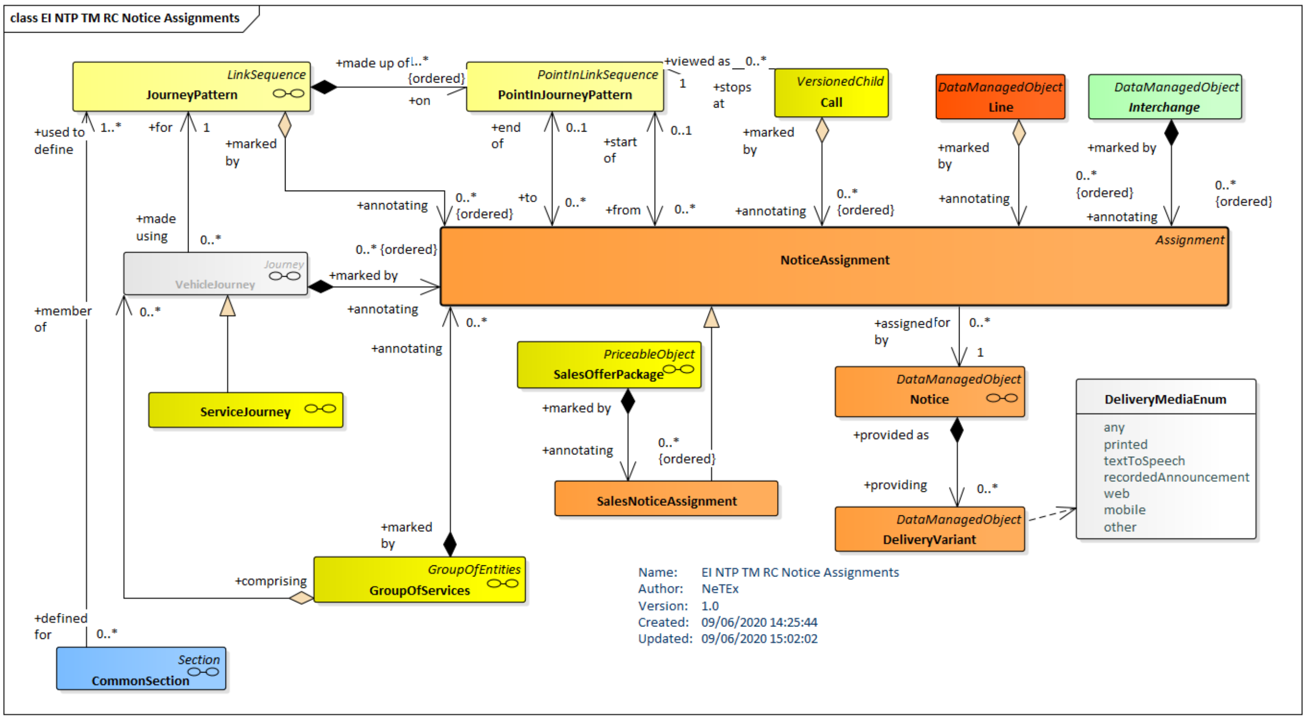


Figure 28 — Notices & Notice Assignments (UML)

10.13.1Notice (Subcomponent - Multipurpose)

A NOTICE carries additional information for passengers that may help them in planning or executing their trip. Footnotes are one type of NOTICE, a human-understandable text that may be made available in various delivery formats. Using a NOTICE ASSIGNMENT, NOTICEs can be assigned to stops of one service (POINT IN JOURNEY PATTERN), to stops in multiple services (COMMON SECTION), to JOURNEY PATTERNs, to one VEHICLE JOURNEY or more of them (GROUP OF SERVICES) and to INTERCHANGEs. Each NOTICE ASSIGNMENT can be restricted in its validity by specifying a VALIDITY CONDITION.

The same NOTICE can be used in many different NOTICE ASSIGNMENTs.

EPIP limits the use of NOTICE ASSIGNMENTs to SERVICE JOURNEYs, POINTs IN JOURNEY PATTERN (or between a couple of them) and SERVICE JOURNEY INTERCHANGEs. (In wider NeTEx use a NOTICE may be assigned to any ENTITY).

NB☞ EIRE_NP also allows them for GROUPs of SERVICEs.

Table 130 — *Notice – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	NOTICE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*NoticeIdType*	1:1	Identifier of NOTICE.
	Name	*MultilingualString*	0:1	Name of NOTICE.
	ShortName	*MultilingualString*	0:1	Short Name of Notice. Use Text for content. +v1.1.
	Text	*MultilingualString*	0:1	Content text for NOTICE. Note: use **AlternativeText** for translation

«AK»	***PublicCode***	*xsd:normalizedString*	1:1	Public code for NOTICE.
«AK»	***ShortCode***	*xsd:normalizedString*	1:1	Short code for NOTICE.
«AK»	***PrivateCode***	*PrivateCode*	1:1	Public Code for NOTICE.
«FK»	***TypeOfNoticeRef***	*TypeOfNoticeRef*	1:1	Reference to TYPE OF NOTICE.
	CanBeAdvertised	*xsd:boolean*	0:1	Whether NOTICE is advertised.
	DriverDisplayText	*MultilingualString*	0:1	Driver Display text associated with NOTICE.
«cntd»	***variants***	*DeliveryVariant*	0:*	DELIVERY VARIANTs for NOTICE.

10.13.2NoticeAssignment (Subcomponent - Multipurpose)

The assignment of a NOTICE to an ENTITY. For an ENTITY comprising a sequence of stops (i.e., A LINK SEQUENCE such as JOURNEY PATTERN, a SECTION, or a VEHICLE JOURNEY), it is possible to specifying at which POINT IN JOURNEY PATTERN the validity of the NOTICE starts and ends respectively.

In the EPIP and EIRE_NP, ***NoticeAssignments*** should be embedded in line within the element they annotate, and not placed separately. They may reference a NOTICE already defined in an earlier NOTICE ASSIGNMENT.

Table 131 — NoticeAssignment – *XML Element*

Classification	**Name**		**Type**	**Cardinality**	**Description**
::>	::>		*DataManagedObject*	::>	NOTICE ASSIGNMENT inherits from DATA MANAGED OBJECT.
«PK»	***id***		*TypeOfNotice-AssignmentIdType*	1:1	Identifier of NOTICE ASSIGNMENT.
Assignment group	***Name***		*MultilingualString*	0:1	Name of ASSIGNMENT.
	Description		*MultilingualString*	0:1	Description of ASSIGNMENT.
«FK»	***a***	***NoticeRef***	*NoticeRef*	0:1	Reference to a NOTICE i.e., footnote, note, announcement or other informational text element.
«FK»	***b***	***GroupOfNoticesRef***	*GeneralGroupOf-EntitiesRef*	0:1	A group of NOTICEs held in a GENERAL GROUP OF ENTITies.
	c	***Notice***	*Notice*	0:1	A note or footnote about any aspect of a service, e.g., an announcement, notice, etc. May have different DELIVERY VARIANTs for different media. Always prefer the ***Notice*** to the ***NoticeRef*** (only use NoticeRef for shared ***Notice***s).
«FK»	***NoticedObjectRef***		*VersionOfObjectRef*	0:1	Object with which NOTICE is associated. If given by context can be omitted.
«FK»	***LinkSequenceRef***		*LinkSequenceRef*	0:1	Reference to a LINK SEQUENCE.
«FK»	***SectionRef***		*SectionRef*	0:1	Reference to a SECTION.
«FK»	***StartPointInPatternRef***		*PointInSequenceRef*	0:1	POINT at which applicability of NOTICE starts.
«FK»	***EndPointInPatternRef***		*PointInSequenceRef*	0:1	POINT at which applicability of NOTICE ends. If absent same as Start Point.

	Mark	*xsd:string*	0:1	Mark associated with NOTICE. E.g., '+', '++', '*' etc
	MarkUrl	*xsd:anyURI*	0:1	URL for image associated with NOTICE.
«enum»	***PublicityChannel***	*PublicityChannelEnum*	0:1	How NOTICE is to be publicised. Default is all.
	Advertised	*xsd:boolean*	0:1	Whether NOTICE is advertised to the public.

10.14 Calendars

A SERVICE CALENDAR holds information about when services run. It provides elements to related a DAY TYPE to an actual calendar date.

Transmodel separates the concerns of the type of service appropriate for a particular DAY TYPE (Working days, weekdays, holidays, etc) from the actual calendar which states when the days occur (SERVICE CALENDAR, OPERATING PERIOD, OPERATING DAY,). A SERVICE journey is associated only with a DAY TYPE, not a calendar date. A DAY TYPE ASSIGNMENT relates a DAY TYPE to an actual calendar date.

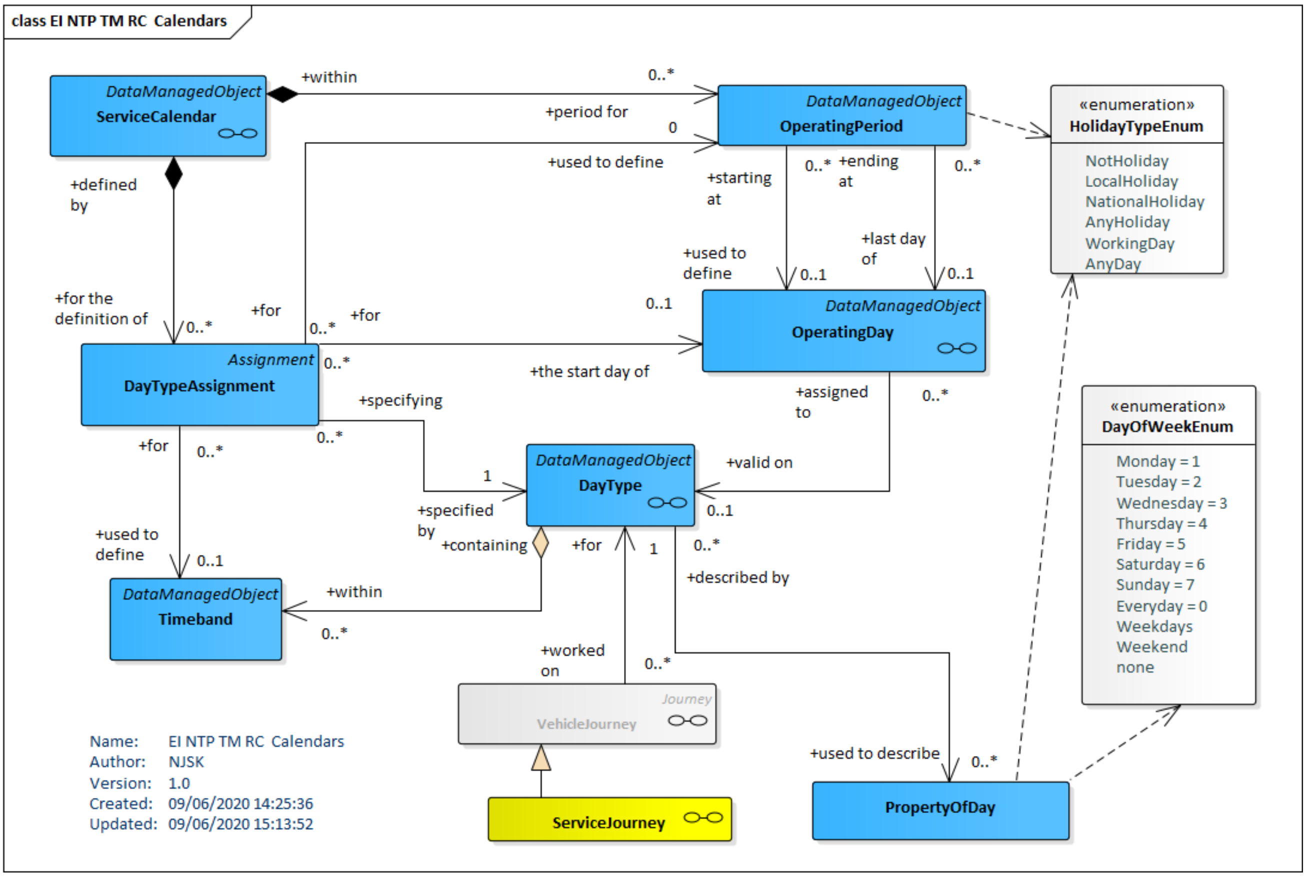


Figure 29 — Operating days and calendar (UML)

This approach allows the same SERVICE JOURNEY definitions to be used with different SERVICE CALENDARs in successive time periods. In effect the approach allows the conditions under which a journey should run to be specified independently of the actual calendar dates. A calendar with specific day assignments can then be used to determine the journeys to be run on an actual day.

10.14.1DayType

In Transmodel, a DAY TYPE is defined as a combination of various properties a day may have, and which will influence the transport demand and the running conditions (e.g., traffic flow for buses).

Any single condition that is relevant to the demand will be recorded as a particular PROPERTY OF DAY. For example, *"workday", "Sunday", "school holidays", "market day*" would each be a PROPERTY OF DAY. A workday during school holidays, which is a market day, would be a DAY TYPE, formed with the combination of those three PROPERTies OF DAY.

The most classical PROPERTY OF DAY is the DAY OF WEEK (e.g., "*Wednesday*"); more than one DAY OF WEEK can be specified for a give type, e.g., "Monday, Wednesday, Thursday" or '"*Saturday, Sunday". A*lso common are "public holiday", "Not a holiday" and "school term day". A DAY TYPE may associate different properties of the same type (e.g., "Tuesday or Thursday").

More than one ***PropertyOfDay*** instance can be specified for a single ***DayType*** in order to specify a compound condition.

- Properties values specified on the same PROPERTY OF DAY instance should be logically ANDed, for example ["*Monday" A*ND "Public Holiday" and "Winter"].
- Separate property instances should be logically ORed together, for example [["*Sunday*" AND "*Not Holiday*" AND "*Winter*"] OR ["*Monday*" AND "*Holiday*" AND "*Summer"*]].

Table 132 — *DayType* – XML Element

Classificat ion	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	DAY TYPE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*DayTypeIdType*	1:1	Identifier of DAY TYPE.
	Name	*MultilingualString*	0:1	Name of DAY TYPE.
	ShortName	*MultilingualString*	0:1	Short name of DAY TYPE.
	Description	*MultilingualString*	0:1	Description of DAY TYPE.
«AK»	***PrivateCode***	*PrivateCode*	0:1	Alternative Identifier for DAY TYPE.
	EarliestTime	*xsd:time*	0:1	Earliest start time of DAY TYPE.
	DayLength	*xsd:duration*	0:1	Length of DAY TYPE.
«cntd»	***properties***	*PropertyOfDay*	0:*	PROPERTies of DAY TYPE.
«cntd»	***timebands***	*Timeband*	0:*	TIMEBANDs of DAY TYPE.

10.14.2PropertyOfDay (Subcomponent of DayType)

A property which a day may possess, such as school holiday, weekdays, summer, winter, etc. This may be used to generate a description of a day type in many different natural languages.

Table 133 — *PropertyOfDay – XML Element*

Classificat ion	Name	Type	Cardinality	Description
	Name	*MultilingualString*	0:1	Name of PROPERTY OF DAY.
	Description	*MultilingualString*	0:1	Description of PROPERTY OF DAY.
«enum»	***DaysOfWeek***	*DayOfWeekEnum*	0:7	Days of week assigned to PROPERTY OF DAY.

«enum»	***WeeksOfMonth***	*WeekOfMonthEnum*	0:5	Weeks of month (1-5) assigned to PROPERTY OF DAY.
	a ***MonthOfYear***	*xsd:gMonth*	0:1	For those day types that occur on the same month every year; a month as a number.
	b ***DayOfMonth***	*xsd:gDay*	0:1	For those day types that occur on the same day of the month; a month as a number.
	c ***DayOfYear***	*xsd:gMonthDay*	0:1	For those day types that occur on the same day every year, month and day, assigned to PROPERTY OF DAY.
«enum»	***CountryRef***	*CountryEnum*	0:*	Country of Holiday type for PROPERTY OF DAY.
«enum»	***HolidayTypes***	*HolidayTypeEnum*	0:5	Holiday type assigned to PROPERTY OF DAY. • *AnyDay,* • *WorkingDay, NotWorkingDay* • *SchoolDay, NotSchoolDay* • *AnyHoliday, NotHoliday, LocalHoliday, RegionalHoliday, NationalHoliday* • *HolidayDisplacementDay, EveOfHoliday*
«enum»	***Seasons***	*SeasonEnum*	0:4	Season of year assigned to PROPERTY OF DAY. *Spring, Summer, Autumn, Winter, Perennially*
«enum»	***Tides***	*TideEnum*	0:4	State of tide assigned to PROPERTY OF DAY.
«enum»	***DayEvent***	*DayEventEnum*	0:1	Events happening on day. For example, market days. anyDay normalDay marketDay matchDay eventDay
«enum»	***Crowding***	*CrowdingEnum*	0:1	Relative busyness of day. • *veryQuiet, quiet, normal, busy* *, veryBusy*

10.14.3Timeband (Subcomponent of DayType)

A period in a day, significant for some aspect of public transport, e.g. similar traffic conditions or fare category.

In EPIP TIMEBANDs are limited to use within a DAY TYPE

The EIRE_NP also allows their use in AVAILABILITY CONDITIONs.

Table 134 — *Timeband – XML Element*

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*DataManagedObject*	::>	TIME BAND inherits from DATA MANAGED OBJECT. Note that most inherited values are suppressed.
«FK»	***id***	*TimebandIdType*	1:1	Identifier of TIME BAND.
	Name	*MultilingualString*	0:1	Name of TIME BAND.
	StartTime	*xsd:time*	1:1	Inclusive start time of TIME BAND.
	EndTime	*xsd:time*	1:1	Inclusive end time of TIME BAND.

	DayOffset	*xsd:integer*	0:1	Day offset of end time from start time. If same day, zero.
	Duration	*xsd:duration*	0:*	Length of day - Alternative to use of end time.

10.14.4ServiceCalendar

A SERVICE CALENDAR groups the calendar related ENTITies and provides in particular a set of assignments of OPERATING DAYs to DAY TYPEs.

Table 135 — *ServiceCalendar – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	SERVICE CALENDAR inherits from DATA MANAGED OBJECT.
«PK»	***id***	*ServiceCalenda-rIdType*	1:1	Identifier of SERVICE CALENDAR.
	Name	*MultilingualString*	0:1	Name of SERVICE CALENDAR.
	ShortName	*MultilingualString*	0:1	Short name of SERVICE CALENDAR.
	FromDate	*xsd:date*	0:1	Inclusive start date for validity of SERVICE CALENDAR.
	ToDate	*xsd:date*	0:1	Inclusive end date for validity of SERVICE CALENDAR.
	EarliestTime	*xsd:time*	0:1	Earliest time that days start SERVICE CALENDAR. Default to use if not specified on individual OPERATING DAY.
	DayLength	*xsd:duration*	0:1	Day length used with earliest time to work out end of day. Default to use if not specified on individual OPERATING DAY.
«cntd»	***dayTypes***	*DayType*	0:*	DAY TYPEs in SERVICE CALENDAR.
«cntd»	***timebands***	*Timeband*	0:*	TIMEBANDs in SERVICE CALENDAR. In the FXCP and EPIP, each DAY TYPE must declare its own timebands
«cntd»	***operatingDays***	*OperatingDay*	0:*	OPERATING DAYs in SERVICE CALENDAR.
«cntd»	***operatingPeriods***	*OperatingPeriod*	0:*	OPERATING PERIODs in SERVICE CALENDAR.
«cntd»	***dayTypeAssignments***	*DayTypeAssignment*	0:*	DAY TYPE ASSIGNMENTs in SERVICE CALENDAR.

10.14.5OperatingDay (Subcomponent of ServiceCalendar)

An OPERATING DAY is in most cases similar to a calendar day (i.e., an individual date in the calendar, such as ‘1st April 2019’), but may differ in its start and end times.

An assignment process, allocating DAY TYPEs to an OPERATING DAY, allows selection of the most appropriate schedule to meet the expected demand and to face the traffic conditions. This leads to an operational plan for every OPERATING DAY. The plan is completed by the assignment of physical resources to the theoretical work and amended as necessary to deal with unexpected circumstances (although such operational measures are outside of the scope of the profile).

Table 136 — *OperatingDay – XML Element*

Classification	Name	Type	Cardinality	Description

::>	::>	*DataManagedObject*	::>	OPERATING DAY inherits from DATA MANAGED OBJECT.
«PK»	***id***	*OperatingDayIdType*	1:1	Identifier of OPERATING DAY.
	CalendarDate	*xsd:date*	1:1	Calendar date of OPERATING DAY.
«FK»	***ServiceCalendarRef***	*CalendarRef*	0:1	SERVICE CALENDAR to which OPERATING DAY belongs.
	Name	*MultilingualString*	0:1	Name of OPERATING DAY.
	ShortName	*MultilingualString*	0:1	Short Name of DAY TYPE.
«AK»	***DayNumber***	*xsd:integer*	1:1	Day Number if different from ***@id***.
«AK»	***PrivateCode***	*PrivateCode*	0:1	SERVICE CALENDAR to which DAY TYPE belongs.
	EarliestTime	*xsd:time*	1:1	Start time of OPERATING DAY.
	DayLength	*xsd:duration*	1:1	Length of OPERATING DAY.

10.14.6DayTypeAssignment (Subcomponent of ServiceCalendar)

A DAY TYPE ASSIGNMENT allocates a DAY TYPE to a date, i.e., OPERATING DAY.

IN EPIP this is done using an OPERATING PERIOD.

Table 137 — *DayTypeAssignment – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	DAY TYPE ASSIGNMENT inherits from ASSIGNMENT
«PK»	***id***	*ShortTermAssignmentIdType*	1:1	Identifier of DAY TYPE ASSIGNMENT.
Assignment group	***Name***	*MultilingualString*	0:1	Name of ASSIGNMENT.
Assignment group	***Description***	*MultilingualString*	0:1	Description of ASSIGNMENT.
	ServiceCalendarRef	*CalendarRef*	0:1	SERVICE CALENDAR to which DAY TYPE ASSIGNMENT belongs.
«FK»	***OperatingPeriodRef***	*OperatingDayRef*	1:1	Reference to OPERATING PERIOD assigned by this DAY TYPE ASSIGNMENT. Mutually exclusive with OPERATING DAY (or date).
«FK»	***OperatingDayRef***	*OperatingDayRef*	1:1	Reference to OPERATING DAY assigned by this DAY TYPE ASSIGNMENT.
	Date	*xsd:date*	1:1	Date - may be used instead of OPERATING DAY.
«FK»	***DayTypeRef***	*DayTypeRef*	1:1	Reference to DAY TYPE assigned by this DAY TYPE ASSIGNMENT.
«FK»	***TimeBandRef***	*TimeBandRef*	0:1	Reference to TIME BAND assigned by this DAY TYPE ASSIGNMENT.
	IsAvailable	xsd:boolean	0:1	Whether available on assigned day.

10.14.7Operating Period

A continuous interval of time between two OPERATING DAYs which will be used to define validity.

Note that the ***UicOperatingPeriod*** will always be used in the context of the EPIP, in order to make the ***ValidDayBits*** mandatory (string of bits, one for each day in the period: whether valid or not valid on the day).

NB☞ The EIRE_NP also allows the use of regular OPERATING PERIODs.

Table 138 — *OperatingPeriod – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	OPERATING PERIOD inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PeriodIdType*	1:1	Identifier of OPERATING PERIOD.
«FK»	***ServiceCalendarRef***	*CalendarRef*	0:1	SERVICE CALENDAR to which OPERATING PERIOD belongs. Omit if nested within a calendar
	Name	*MultilingualString*	0:1	Name of OPERATING PERIOD.
	ShortName	*MultilingualString*	0:1	Short Name of OPERATING PERIOD.
«FK»	a ***FromOperatingDayRef***	*OperatingDayRef*	1:1	Start OPERATING DAY of OPERATING PERIOD.
	b ***FromDate***	*xsd:dateTime*	1:1	Start OPERATING DATE (inclusive).
«FK»	a ***ToOperatingDayRef***	*OperatingDayRef*	1:1	End OPERATING DAY of OPERATING PERIOD.
	b ***ToDate***	*xsd:dateTime*	1:1	End OPERATING DATE (inclusive).
«enum»	***HolidayType***	*HolidayTypeEnum*	0:*	Holiday type of OPERATING PERIOD.
«enum»	***Season***	*SeasonEnum*	0:*	Season of OPERATING PERIOD.

10.14.8UicOperatingPeriod (Subcomponent of ServiceCalendar)

The ***UicOperatingPeriod*** is an alternative optimised representation of an ***OperatingPeriod*** that uses a string of bits to indicate to which days within the period a specific ***DayType*** applies. It is associated with a DAY TYPE by a DAY TYPE ASSIGNMENT.

UicOperatingPeriod is preferred in EPIP because it is compact and simple to implement (and corresponds to some legacy encodings).

Table 139 — *UicOperatingPeriod – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*OperatingPeriod*	::>	UIC OPERATING PERIOD inherits from OPERATING PERIOD.
	ValidDayBits	*xsd:normalizedString*	1:1	String of bits (built of "0" and "1"), one for each day in the period: whether valid or not valid on the day. Normally there will be a bit for every day between start and end date. If bit is missing, assume available.
	DaysOfWeek	*DaysOfWeek*	0:1	Days of week to which correspond. (Up to first seven) bits

10.15 ServiceFrame

A SERVICE FRAME is used to group the elements of the Service Model.

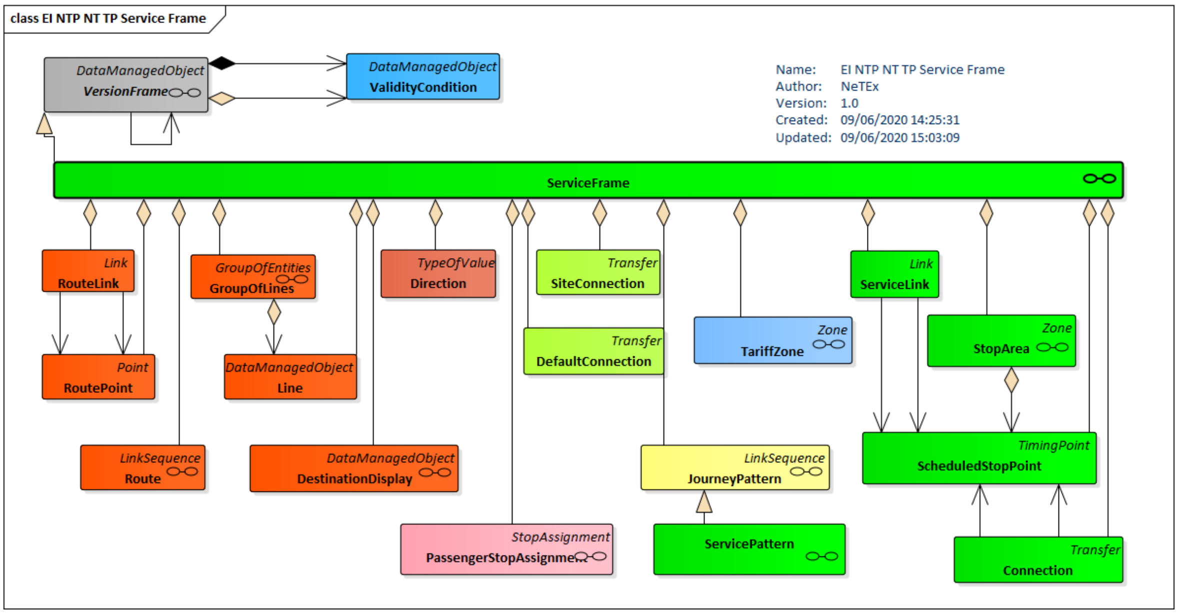


Figure 30 — Service Frame (UML)

10.16 Timetable

A Timetable FRAME is used to group the elements of the Timetable Model.

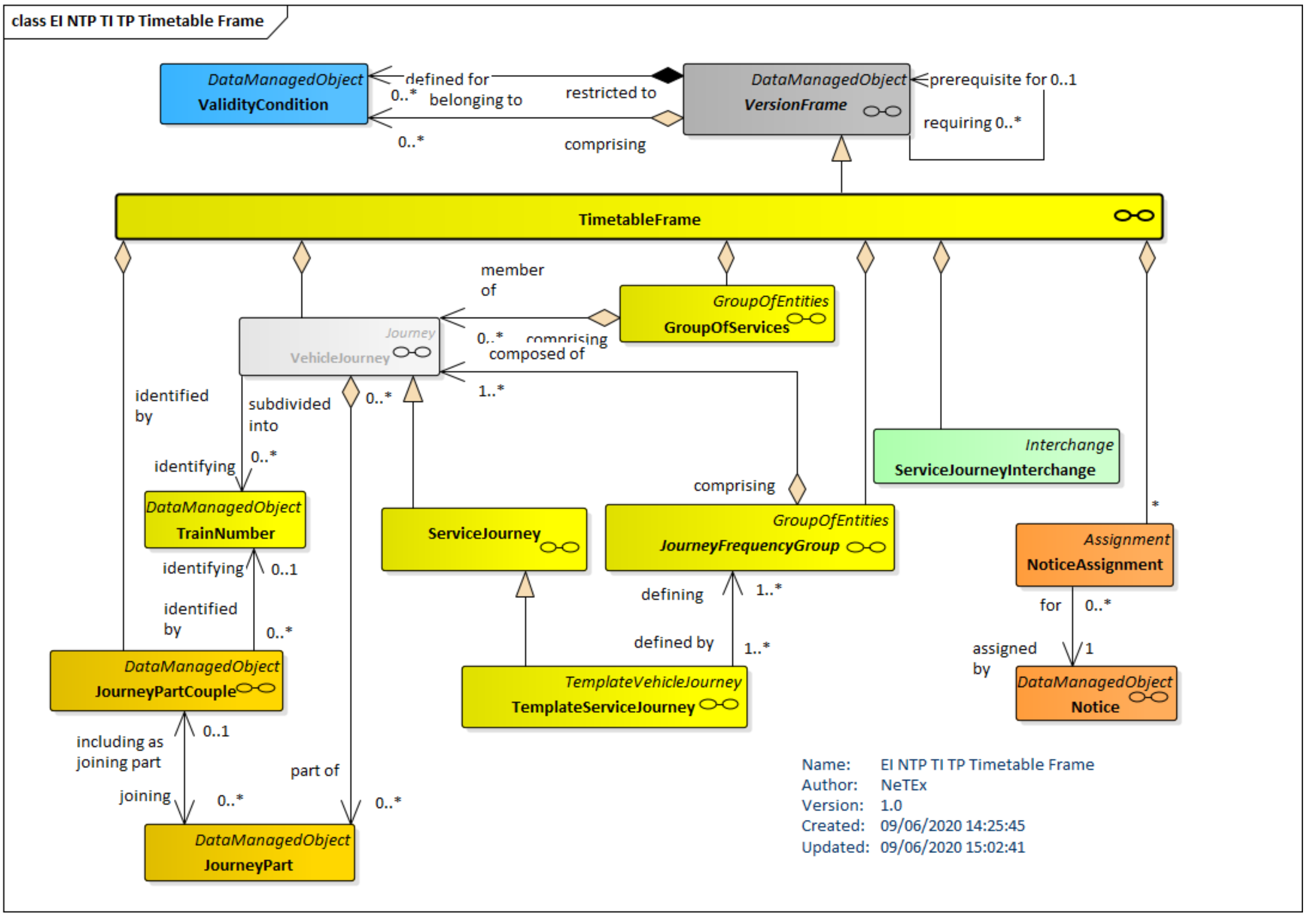


Figure 31 — Timetable Frame (UML)

11 Common rules

This section sets out common rules for different aspects of processing an EPIP compliant XML document.

11.1 Version management of data

The version and validity condition management under the EIRE_NP is the same as for the EPIP and is based straightforwardly on the use of VERSION FRAMEs.

The fundamental rules for object versioning are the following:

- All objects must have a unique ***id*** (for certain dependent child components this may be qualified by an *'order'*; in which case the *'id+order'* must be unique*).* The ***id*** shall be persistent throughout the life of a given object and successive exchanges of data for it; an update of an object shall not change the ***id***.
- All objects having an ***id*** shall also have a ***version*** (if the version is not managed or is unknown, the value of "*any*" should be used.
- As required by the concept of VERSION FRAME, all the versions of objects within a ***VersionFrame*** shall be coherent – that is consistent and usable as a whole. If coarse grained versioning is being used, it may be that all the elements have the

same version level, but this does not have to be the case; the actual ***version*** value can be stated on an individual object basis if desired, that is, does not have to be the same on all instances.

- Each VERSION FRAME must be assigned a specific ***version*** value (i.e., not "*any*"), furthermore:
  - The value must be a number (sequence of digits).
  - The version numbers used for a given VERSION FRAME must increase monotonically with each successive issue of the frame. That is to say, an update of a specific FRAME with the same *id*) shall have a version number higher than the previous one.

Modifying an object shall result in an increment in the version of the object (and that of any affected parent object that contains it). This is intended to make it efficient for a consuming system to detect changes within a data set (since any change to the detailed content is flagged at the highest level by a change in the version number of the container element as well). Thus, the essential rules for modifying object versions are as follows:

- If any of the values of the attributes of an object change, its version number must be incremented.
- If any of the references held by an object are added to, removed, or changed to reference a different object, its version number (i.e., the version number of the *referencing* object) must be incremented. Note that the version of a reference (held on the ***version*** attribute of a ***VersionObjectRef***) is that of the *referenced* object and for an association, is independent of the ***version*** attribute, (inherited from ***EntityInVersion***), of the *referencing* object itself**.**
- Likewise, if the version number specified on any of the references held by a referencing object is changed to indicate a different version of the same *referenced* object, the version number of the *referencing* object must be incremented.
- If an object is *composite*, that is, includes an aggregation of child elements that form part of its definition (as say, a SERVICE PATTERN is composed of STOPs IN JOURNEY PATTERN, or a SERVICE CALENDAR is composed of DAY TYPEs, etc), then if any subcomponent is modified (i.e., any child element is added, deleted or modified), the version number of both the modified child element and the containing parent must be incremented (but they do not have to be the same value). Similarly, if any element in a frame is materially modified and its version number incremented, then the version of the VERSION FRAME must also be incremented.

These four versioning rules can be summarised more succinctly as follows:

1. **A change to an attribute value requires a new version of an object.**
2. **A change to a reference held by the referencing object requires a new version of an object.**

3. **A change to a reference version held by the referencing object requires a new version of the referencing object.**
4. **A change to a subcomponent is also a change to its parent.**

Not all systems support the fine-grained provision of version numbers as described above; however, the same rules must be observed even when only populating version numbers at a coarser level; at its simplest this amounts to:

— If any object in a frame is modified, the version number of the frame must be incremented.

11.2 Use of validity conditions

NeTEx includes capabilities to specify complex validity conditions composed of multiple rules based on values in the data. In the EPIP, in the interests of simplicity, only very simple conditions are permitted, and the primary use is to state the applicability of data in a frame.

NB☞ The EIRE_NP also allows the use of slightly more complex AVAILABILITY CONDITIONS, but not the chaining or combining of multiple conditions as is also possible in NeTEx. Although AVAILABILITY CONDITIONs are quite simple in themselves, when they are specified at different levels (e.g., both on frame and payload elements) or on related elements (e.g., on both a STOP PLACE and its ENTRANCEs), complexity can arise and rules are needed to indicate how to combine them. Use of conditions should be limited and care should be taken that they are not contradictory.

The main rules for using VALIDITY CONDITIONs are the following:

1. **Frame dominance rule**: The **dominant** VALIDITY CONDITION is the one attached to the VERSION FRAME itself: if other VALIDITY CONDITIONs attached to individual objects within the frame contradict the dominant frame VALIDITY CONDITION (for example by having a wider temporal span), they will be ignored.

   — Note that as well the VALIDITY CONDITIONs that apply to the frame (specified with the ***validityConditions*** tag), frames may also contain declarations of shared VALIDITY CONDITIONs (specified with the ***contentValidityConditions*** tag) that are reference by some of the elements in the frame – these do not apply to the FRAME itself.

2. **Required frame condition rule:** A FRAME shall have at least one **dominant** VALIDITY CONDITION and it must give a start date.
3. **Composite frame precedence rule:** The **dominant** VALIDITY CONDITION attached to a COMPOSITE FRAME is applied to all the contained FRAMEs and is dominant over the other frames and their contents in the same way that each FRAME's own VALIDITY CONDITION is dominant over its contained objects.
4. **Successor Frame precedence rule:** If the temporal span of a **dominant** VALIDITY CONDITION for a successive version of a FRAME overlaps some or all of the span on of a previous version of the same FRAME (i.e., one with the same ***id***), the new version completely overrides the previous one for the overlapping period (thus

cancelling all the previous data for the period in question and replacing it with the data for the period from the successive FRAME).

5. **Self-consistency rule**: VALIDITY CONDITIONs for a given object should not be contradictory. For example, the same object should not have two conflicting conditions that apply at the same time "*Only in Summer*" and "*Only in Winter*".
6. **Model-consistency rule**: The consistency of VALIDITY CONDITIONs should not contradict the inherent semantics of the model. For example, if there is a condition stating that a STOP PLACE is closed at a specific time, then there should not be a condition stating that one of its STOP PLACE ENTRANCEs is open during that period – though the converse (just a specific ENTRANCE being closed, but the rest of the station open) would be meaningful and so possible.
7. **Full specific condition rule**: An AVAILABILITY CONDITION on a subcomponent should restate the full temporal condition, rather than presuming the inheritance from the parent component of any unspecified attributes. For example, if a STOP PLACE is available for "Monday to Sunday, except on public holidays", but one of its ENTRANCEs is only open on "Monday to Sunday", the separate condition specified on the ENTRANCE should also state "not on Public Holidays" (and not presume the inheritance of this restriction from the condition on STOP PLACE).

It should be noted that VERSION FRAME elements have two different associations with VALIDITY CONDITION:

(1) the ***validityConditions*** association, which specifies the conditions that apply to the validity of the frame. For the EIRE profile these should be limited to simple VALID BETWEEN conditions.

(2) The ***contentValidityConditions*** association which declares shared conditions that are specifically referenced by multiple elements in the frame - but do not condition the frame itself.

11.2.1 Temporal completeness

A frame providing data for a specified period must contain data for the primary payload (which will depend on the frame type) for the period. Thus, for example, a TIMETABLE FRAME should contain SERVICE JOURNEYs for the full extent of the dominant condition, and not contain gaps –notwithstanding some journeys not being available on certain day types within that period. Similarly, a FARE FRAME should contain FARE PRODUCTs or FARE PRICEs that are valid for the stated period of the frame.

11.3 Populating creation and change dates

All ***DataManagedObject*** instances have timestamp attributes that can be used to record their dates of original creation (***created***) and last modification (***changed***) – the latter being the time last changed as known to the producer system at the time of export from the producer system.

The population of such attributes in an exported EPIP conformant document is optional. However, if used they should not be confused with the date the XML document itself was created, which must be indicated by the ***PublicationTimestamp*** of the ***PublicationDelivery*** wrapper tag: see *11.14*.

- The ***created*** attribute is the date the entity was first created on the originating system. In the case of a reference, it is the date the reference was created.
- The ***changed*** attribute is the date the entity was last versioned (normally because of a material modification)
- If populated, the ***created*** and ***modified*** date attributes on a ***VersionOfObjectRef*** should reflect the date that the *reference* is changed (which may differ from the dates the *referencing* and the *referenced* object last changed).

The period of the frame's validity is indicated by 'dominant' a VALIDITY CONDITION attached to a VERSION FRAME. A simple ***ValidBetween*** condition is convenient for specifying this, for example:

NeTEx XML EXAMPLE

```
    <CompositeFrame version="3.0" id="epd:EI:STG:CompositeFrame-EI_PI_LINE_OFFER:STG-
BRM-1">
        <ValidBetween>
            <FromDate>2019-01-01T09:02:30.0Z</FromDate>
            <ToDate>2020-01:01T00:02:30.0Z</ToDate>
        </ValidBetween>
```

11.4 Codespaces and the structure of identifiers

The data exchanged with the EIRE_NP comes from multiple sources and stakeholders, is exchanged repeatedly, and may be integrated with other data sets in many different ways. It is therefore essential to make sure that identifiers are unique and stable throughout an object's lifecycle.

In NeTEx, the actual structure of an identifier can be arbitrary and is indicated by its CODESPACE, which also establishes a context within which a given code structure is unique (see 8.1.7). Multiple CODESPACEs can be used within a document. If the CODESPACE is the default CODESPACE for the frame, it may be omitted from elements in the frame.

A CODESPACE may also imply additional rules for the internal format of the code, such as minimum and maximum lengths, permitted characters, or a specific substructure, but any such rules are external to NeTEx and must be agreed by the implementers of producers and consumers of a dataset.

The purpose of a CODESPACE is to make NeTEx neutral as to the identifier systems used and so allow the concurrent existence of many different code systems within the same document. CODESPACEs avoid any unnecessary requirement for a common identifier structure and furthermore allow for extremely compact references with a document.

A single CODESPACE may cover multiple types of different element, provided the identifiers and class context are unique within it (see discussion of wider contexts under CODESPACE in section 8.1.7). For example, TOPOGRAPHIC PLACEs in the National Public Transport Gazetteer all begin with 'E,' NaPTAN SCHEDULED STOP POINT ids all begin with a numeric area prefix and have a 0, or 1 in the fourth place, while NaPTAN STOP AREA ids are similar to SCHEDULED STOP POINTs but have a 'G' in the fourth place (and moreover are different element types). Thus, they have distinct scopes and – so could all be placed in in a single 'NaPTAN' CODESPACE.

Similarly, a single '*eix_metadata'* CODESPACE can be used for all EIRE_NP metadata.

The following are some fictional examples of ***Codespace*** use for different identifiers:

Table 140 — Codespace examples

Examples	Codespace prefix and CODESPACE URI
`<TopographicPlace id="nptgLocality:E00034531">`	'nptgLocality = http://naptan.org.uk/nptgLocality
`<ScheduledStopPoint id="smirt:454e74">`	*'smirt'* = http://www.hdfmobilites.fr/smirt/stops
`<Line id="llp:22">`	*'llp'* = http://llp.sl/lines
`<Route id="enturRoute:985a">`	*'entRoute'* = http://www.entur.org/routes
`<ServiceJourney id="db:107678">`	*'db'* = http://bahn.de/reise
`<StopPlace id="ratpStop:43289">`	*'raptStop'* = http://ratp.fr
`<StopPlace id="naptStop:199G98765431">`	*'naptStop'* = http://naptan.org.uk/stops

11.4.1 Use of codespaces in the EIRE_NP

All identifiers in an EIRE_NP conformant document must be assigned a CODESPACE and be guaranteed to be unique within it.

NB☞ The EIRE_NP uses the codespaces of the Irish national data sets for data. See Table 28.

NB☞ Note that the EPIP defines additionally a Pan-European codespace for data elements - See Annex D. The EPIP standard codespace can be used as a general-purpose solution if an implementor does not have a preferred alternative, but is verbose and harder to use that the existing Irish codespaces. For the EIRE_NP, the use of simple codespaces is preferred (including the NAPTAN, NPTG and NOC for the various national data sets), but the EPIP system is used for EIRE_NP frame identifiers so as to give interoperability with the European profile.

11.4.2 Irish National codespaces

For Irish timetable and fare data, some elements (e.g., NaPTAN stops, NPTG Locations) come from existing national data sets that have existing well-defined public codespaces (see below). Most other data elements come from an operator and the choice of a code system and allocation of codes will be according to the operator's own internal system, i.e., fall within the operator's codespace. The operator themselves should be identified by an Irish National Operator code (NOC).

Table 28 shows the codespaces for the Irish national PT data sets that are relevant for the EIRE_NP.

Table 28 – Irish National Codespaces

Data Set	Content	NeTEx Element	Codespace	Preferred prefix
Administrative Areas	NPTG registrar data (***Country, Region, District, AdministrativeArea***)	ADMINISTRATIVE AREA, (+GENERAL ORGANISATION)	*nptg.org.uk/admin*	*nptgAdminArea:*
Locations	NPTG payload (**NptgLocality***)*	TOPOGRAPHICAL AREA	*nptg.org.uk/locality*	*nptgLocality:*

NPTG profile data	*NPTG metadata*	Fixed code values for classifying localities TYPE OF ZONE etc	*nptg.org.uk/nptg*	*nptg:*
NaPTAN profile data	*NaPTAN metadata*	Fixed code values for classifying stops TYPE OF ZONE etc	*naptan.org.uk/napt*	*napt:*
Stops, Stop areas	NaPTAN payload (StopPoint, StopArea)	SCHEDULED STOP POINT, STOP AREA	*naptan.org.uk/stops*	*naptStop:*
Operators	National Operator Code (NOC)	OPERATOR	*Nta.ei/noc*	*noc:*
EIRE_NP profile data	*Timetable metadata*	Fixed code values	*Nta.ei/eix*	*eix:*
EIRE_NP profile	*Timetable metadata*	DAY TYPE, SERVICE CALENDAR	*Nta.ei/eix*	*eix:*
Timetables	*Timetable* payload	Various NeTEx timetable elements	*Individual provider*	

11.4.3 Identifier structure for EIRE_NP version frames

The identifiers of profile frames also need to be unique and consistent, since the same frame may be exported repeatedly in successive versions and a consumer system needs to be able to reconcile the successive versions of the same data set.

The EIRE_NP uses the same system as the EPIP for the identifier structure of a VERSION FRAME. This is very similar to the EPIP identifier for other types of data object (see Annex D), and begins with the same four parameters to start. A frame identifier differs however in that the last part of the identifier is a ***[frame-topic],*** chosen to reflect the contents of the frame, rather than being just an arbitrary identifier.

The structure of a FRAME identifier is:

[country-code] : [provider-code] : [NeTEx-frame-type] _ [EIRE_NP-type] : [frame-topic]

Where:

- ***[country-code]*** is the ISO 3166-1 code (2 characters, capitalised) of the country of the ***[provider-code]***.
- ***provider-code]*** is a code uniquely identifying the locality or the provider within the country (may be a region code like the European NUTS code, an authority code, etc.).
  - For the EIRE_NP the [local code] should be the NOC Code if the data is provided by an operator, or a unique identifier e.g., NTA, TfL, TfN, if some other source.
  - Where an OPERATOR has multiple NOC Codes, the code used should be the most specific for the data. For example, the lowest relevant sub-division of the operator, e.g., FEDB for First Group’s bus operations in Dublin for routes

and timetables, or the Atco prefix of the local authority for NaPTAN stops for the administrative area.

- [***NeTEx frame-type]*** is one of the NeTEx ***VersionFrame*** types: ***CompositeFrame, ResourceFrame, SiteFrame, ServiceFrame***, ***TimetableFrame*** or ***FareFrame***)
- [***EIRE_NP-type]*** is the identifier of one of the EIRE_NP ***TypeOfFrame*** instances, viz:
  - Specific
    - ‘*EI_PI_COMMON’* for Resource data (***ResourceFrame***).
    - ‘*EI_PI_STOP’* for Stop data (***SiteFrame***).
    - ‘*EI_PI_NETWORK’* for a Network data (***ServiceFrame***).
    - *‘EI_PI_TIMETABLE* ‘for Timetable data (***TimetableFrame***).
    - *‘EI_PI_CALENDAR’* for Calendar data (***ServiceCalendarFrame***).
    - *‘EI_PI_METADATA’* for Common resource data (***ResourceFrame***).
  - Composite
    - *‘EI_PI_LINE_OFFER’* for a Line data set (***CompositeFrame***).
    - ‘*EI_PI_NETWORK_OFFER’* for a Network data set (***CompositeFrame***).
    - ‘*EI_PI_STOP_OFFER’* for a Stop data set (***CompositeFrame***).
- [***frame-topic***] is a designator of the frame content, constructed according to frame type from the main parameters that describe it. The designator should be a short name or code, fewer than fifteen characters, made only of letters (uppercase and lowercase), numbers and hyphen "-".
    - For ‘*EI_PI_LINE_OFFER*’: The LINE number (or name if no number is available.
    - For ‘*EI_PI_NETWORK_OFFER’:* The short name of the NETWORK.
    - For ‘*EI_PI_STOP_OFFER’*, name or identifier of OPERATOR, ADMINISTRATIVE ZONE or TOPOGRAPHIC PLACE for which the stops are included.

In an EIRE_NP compliant document, the elements making up the [***frame-topic]*** of the outer ***CompositeFrame s***hall also be cited as ***topics*** in the ***PublicationDelivery*** enclosing the frame***.*** This makes it possible for a consumer (human or machine) to interpret the [***frame-topic]*** and understand the frame’s intended scope.

NB☞ To convert an EIRE_NP profile document to an EPI one the ‘EI_PI’ is replaced with ‘EU_PI’.

11.4.4 EIRE_NP Offer Frame Examples

Note that in these examples, the “*epd*” codespace is for EPIP conformant data and so is used for the ids of VERSION FRAMEs. See Annex D.

NeTEx XML EXAMPLES:

(a) For an EI_PI_LINE_OFFER that specifies stops, networks and timetable for the *First Group Cork's* (FCRK) Line *'26'*:

```
<PublicationDelivery version="ntx:1.1"
::::
        <topics>
            <NetworkFrameTopic>
                <Current/>
                <NetworkFilterByValue>
                    <objectReferences>
                        <LineRef="frst:FCRKR@26"/>"
                    </objectReferences>
                    <NetworkRef="frst:FCRK"/>
                </NetworkFilterByValue>
            </NetworkFrameTopic>
        </topics>
::::
    <CompositeFrame version="3.0" id="epd:EI:FCRK:CompositeFrame-EI_PI_LINE_OFFER:
FCRK@26"
                    responsibilitySetRef="frst:data">
        <ValidBetween>
            <FromDate>2010-12-17T09:30:47.0Z</FromDate>
            <ToDate>2011-12-17T09:30:47.0Z</ToDate>
        </ValidBetween>
        <TypeOfFrameRef ref="eix:EI:NTA:EI_PI_LINE_OFFER:EIRE_NP" version="eix:v1.0"/>
        <codespaces>
            <Codespace id="frst_data
                <Xmlns>frst</Xmlns>
                <XmlnsUrl>https://www.firstgroup.ei/cork</XmlnsUrl>
                <Description>First data</Description>
            </Codespace>
            <Codespace id="eix_metadata">
                <Xmlns>eix</Xmlns>
                <XmlnsUrl>http://netex.ie/eix</XmlnsUrl>
                <Description>FXC metadata </Description>
            </Codespace>
            <CodespaceRef ref="epd_data">epd</ CodespaceRef>

        </codespaces>
        <FrameDefaults>
            <DefaultCodespaceRef ref="frst_data"/>
            <DefaultResponsibilitySetRef version="any" ref="frst:data"/>
        </FrameDefaults
        <frames>
            <SiteFrame version="5.6" id="epd:EI:FCRK:SiteFrame-EI_PI_STOP: FCRK@26">
                <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_STOP:EIRE_NP"
version="eix:v1.0"/>
                    <Etc., etc., Specific Frame contents>
            </SiteFrame>
            <ServiceFrame version="2.8" id="epd:EI: FCRK:ServiceFrame-EI_PI_NETWORK:
FCRK@26">
                <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_NETWORK:EIRE_NP"
versionRef="1.0"/>
                    <Etc., etc., Specific Frame contents>
            </ServiceFrame>
            <TimetableFrame version="2.78" id="epd:EI: FCRK:TimetableFrame-
EI_PI_TIMETABLE: FCRK@26">
            <TypeOfFrameRef ref=‟eix:NTA:TypeOfFrame_IE_PI_TIMETABLE:EIRE_NP"
version="eix:v1.0"/>
                    <Etc., etc., Specific Frame contents>
            </TimetableFrame>
    <ServiceCalendarFrame version="1.6" id="epd:EI: FCRK:ServiceCalendarFrame-
EI_PI_CALENDAR: FCRK">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_CALENDAR:EIRE_NP"
version="eix:v1.0"/>
                    <Etc., etc., Specific Frame contents>
            </ serviceCalendarFrame >
            <ResourceFrame version="1.6" id="epd:EI: FCRK:ResourceFrame-EI_PI_COMMON:
FCRK">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_COMMON:EIRE_NP"
version="eix:v1.0"/>>
                    <Etc., etc., Specific Frame contents>
            </ResourceFrame>
        <ResourceFrame version="1.0" id="epd:EI:NTA:ResourceFrame-
EI_PI_METADATA:EIRE_NP"
                    responsibilitySetRef=”eix:EI:NTA:ResponsibilitySet:EIRE_NP”>
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_METADATA:EIRE_NP"
version="eix:v1.0"/>
                    <Etc., etc., Specific Frame Fixed contents>
            </ResourceFrame>
        </frames>
    </CompositeFrame>
```

(b) For an *EI_PI_NETWORK_OFFER that* specifies stops, networks and timetables for a ‘*Phoenix bus’* network (showing here as separate embedded ‘*EI_PI_NETWORK’* & ‘EI_PI_*TIMETABLE* ‘ frames with modularised details for only two lines ‘PHB@1’ and ‘PHB@2’ ; there would be more). Some of the dependencies between the frames are also indicated as ***prerequisites***.

```
<PublicationDelivery version="ntx:1.1"
::::
        <topics>
            <NetworkFrameTopic>
                <Current/>
                <NetworkFilterByValue>
                    <places>
                        <TopographicPlaceRef ref="PHBS”/>
                    </places>
                    <NetworkRef="phb:Phoebus"/>
                </NetworkFilterByValue>
            </NetworkFrameTopic>
        </topics>
::::
    <CompositeFrame version="3.0" id="epd:EI:PHBS:CompositeFrame-
EI_PI_NETWORK_OFFER:Phoebus"
                    responsibilitySetRef=”phb:phoebus_data”>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_NETWORK_OFFER:EIRE_NP"
version="eix:v1.0"/>
        <codespaces>
            <Codespace id="phb_data">
                <Xmlns>mb</Xmlns>
                <XmlnsUrl>http://phoebus.co.ei/data</XmlnsUrl>
                <Description>Phoenix bus data</Description>
            </Codespace>
            <Codespace id="eix_metadata">
                <Xmlns>eix</Xmlns>
                <XmlnsUrl>http://netex.ie</XmlnsUrl>
                <Description>EIRE NP metadata </Description>
            </Codespace>
            <CodespaceRef ref="epd_data">epd</ CodespaceRef>
        </codespaces>
        <FrameDefaults>
            <DefaultCodespaceRef ref="phb_data"/>
            <DefaultResponsibilitySetRef version=”any” ref="phb:phoebus_data"/>
        </FrameDefaults
        <frames>
            <SiteFrame version="5.4" id="epd:EI:PHBS:StopFrame-EI_PI_STOP:Phoebus:phb”>
            <TypeOfFrameRef ref=" eix:EI:NTA:TypeOfFrame_IE_PI_STOP:EIRE_NP" "
version="eix:v1.0"/>
                    <Etc., etc., Specific Frame contents>
            </SiteFrame>
            <ServiceFrame version="5.7"
id="epd:EI:PHBS:ServiceFrame_IE_PI_NETWORK:PHB@1:phb”>
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_NETWORK:EIRE_NP" "
version="eix:v1.0"/>
                <prerequisites>
            <ResourceFrameRef version="5.0" ref="
epd:EI:PHBS:StopFrame_IE_PI_STOP:Phoebus:phb "/>
                </prerequisites>
                    <Etc., etc., Specific Frame contents>
            </ServiceFrame>
            <TimetableFrame version="2.4"
id="epd:EI:PHBS:TimetableFrame_IE_PI_TIMETABLE:PHB@1:phb”>
                <TypeOfFrameRef ref=" eix:EI:NTA:TypeOfFrame_TIMETABLE:EIRE_NP"
version="eix:v1.0"/>
                <prerequisites>
            <ServiceFrameRef version="any"
ref="epd:EI:PHBS:ServiceFrame_IE_PI_NETWORK:PHB@1:phb "/>
                </prerequisites>
                    <Etc., etc., Specific frame contents>
            </TimetableFrame>
            <ServiceFrame version="2.9"
id="epd:EI:PHBS:ServiceFrame_IE_PI_NETWORK:PHB@2">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_NETWORK:EIRE_NP"
version="eix:v1.0"/>
                <prerequisites>
                    <ResourceFrameRef version="any"
ref="SI:LLP:SiteFrame_IE_PI_STOP:Phoebus"/>
                </prerequisites>
                    <Etc., etc., Specific Frame contents>
            </ServiceFrame>
            <TimetableFrame version="2.9"
id="epd:EI:PHBS:TimetableFrame_IE_PI_TIMETABLE:PHB@2">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_TIMETABLE:EIRE_NP"
version="eix:v1.0"/>
                <prerequisites>
                <ServiceFrameRef version="any"
ref="epd:EI:PHBS:ServiceFrame_IE_PI_SERVICE:PHB@2"/>
                <ServiceCalendarFrameRef version="any"
ref="epd:EI:PHBS:ServiceCalendarFrame_IE_PI_CALENDAR:Phoebus "/>
                </prerequisites>

                    <Etc., etc., Specific Frame contents>
            </TimetableFrame>
            <Etc., etc., Service & Timetable frames for other lines>

    <ServiceCalendarFrame version="1.6"
id="epd:EI:PHBS:ServiceCalendarFrame_IE_PI_CALENDAR:Phoebus">
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_CALENDAR:EIRE_NP"
version="eix:v1.0"/>
                     <Etc., etc., Specific Frame contents>
    </ServiceCalendarFrame>
            <ResourceFrame version="1.6"
id="epd:EI:PHBS:ResourceFrame_IE_PI_COMMON:Phoebus">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_COMMON:EIRE_NP"
version="eix:v1.0"/>
                     <Etc., etc., Specific Frame contents>
            </ResourceFrame>
        <ResourceFrame version="1.0"
id="eix:NTA:EIRE_NP:ResourceFrame_IE_PI_METADATA:EIRE_NP"
                     responsibilitySetRef=”NTA:EIRE_NP”>
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_METADATA:EIRE)NP"
version="eix:v1.0"/>
                     <Etc., etc., Specific Frame Fixed value contents>
            </ResourceFrame>
        </frames>
    </CompositeFrame>
```

(c) For an ‘*EI_PI_STOP_OFFER’* with the NaPTAN stops for areas 822 and 847 , each placed in separate EI_PI_STOP frames:

```
<PublicationRequest>
    <RequestTimestamp>2021-01-01T00:00:00Z</RequestTimestamp>
    <Description>Request for NaPTAN data are </Description>
    <topics>
        <NetworkFrameTopic>
            <Current/>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_STOP_OFFER:EIRE_NP"
version="eix:v1.0"/>
                <NetworkFilterByValue>

                <objectReferences>
                    <TransportAdministrativeZoneRef ref="nptgAdminArea:822"/>
                    <TransportAdministrativeZoneRef ref="nptgAdminArea:847"/>
                </objectReferences>
                    <NetworkRef ref="NAPTAN" versionRef="napt:v2.4"/>
            </NetworkFilterByValue>
        </NetworkFrameTopic>
    </topics>
</PublicationRequest>
    <!--- =============== RESULTS =========== -->
<dataObjects>
    <CompositeFrame version="1.0"
                id="epd:EI:NTA:CompositeFrame_IE_PI_STOP_OFFER:NAPTAN-822+847:napt"
                    responsibilitySetRef="napt:stop_data" dataSourceRef="napt:NTA">
        <Name>Sample extract of NaPTAN data encoded a in NeTEx</Name>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_STOP_OFFER:EIRE_NP"
version="eix:v1.0"/>
        <codespaces>
            <CodespaceRef ref="eix_metadata"/>
            <CodespaceRef ref="epd_data"/>
            <CodespaceRef ref="napt_metadata"/>
            <CodespaceRef ref="nptg_metadata"/>
            <CodespaceRef ref="naptStop_data"/>
        </codespaces>
        <frames>
    <!--- ONE FRAME PER ADMIN AREAS -->
    <!-- ======= PAYLOAD= ========================================================= -->
```

```
            <SiteFrame id="epd:EI:NTA:SiteFrame_IE_PI_STOP:NAPTAN-822:napt"
version="9.6"
                    responsibilitySetRef="nptgAdminArea:822" dataSourceRef="napt:NTA">
                <Name>Sample Stop and Stop Area data encoded as NeTEx - Area 822/Name>
                <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_STOP:EIRE_NP"
version="eix:v1.0"/>
                <FrameDefaults>
                    <DefaultCodespaceRef ref="naptStop_data"/>
                    <DefaultDataSourceRef ref="napt:NTA" version="napt:v2.4"/>
                    <DefaultResponsibilitySetRef ref=" nptgAdminArea:822"
versionRef="nptg:v2.1"/>
                </FrameDefaults>
                    <!-- ======= NaPTAN Stop Data ====== -->
                <stopPlaces>
:::::::::: frame contents
                </stopPlaces>
            </SiteFrame>
            <SiteFrame id="epd:EI:NTA:SiteFrame_IE_PI_STOP:NAPTAN-847:napt"
version="5.4"
                     responsibilitySetRef="nptgAdminArea:847" dataSourceRef="napt:NTA">
                <Name>Sample Stop and Stop Area data encoded as NeTEx - Area 847</Name>
                <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame_IE_PI_STOP:EIRE_NP"
version="eix:v1.0"/>
                <FrameDefaults>
                    <DefaultCodespaceRef ref="naptStop_data"/>
                    <DefaultDataSourceRef ref="napt:NTA" version="napt:v2.4"/>
                    <DefaultResponsibilitySetRef ref=" nptgAdminArea:847"
versionRef="nptg:v2.1"/>
                </FrameDefaults>
                    <!-- ======= NaPTAN Stop Data ====== -->
                    <stopPlaces>
v
                    </stopPlaces>
                </SiteFrame>
```

11.4.5 Implementing codespaces

An implementor who is importing data from many different codespaces will need to use the codespace to distinguish different instances of data.

- Note that the actual codespace prefix used in a given document is arbitrary and not necessarily persistent: the same prefix could be assigned to different codespaces in different XML documents; the codespace url itself however must be constant for a given data element.
- Implementors will need to associate a unique surrogate key with each codespace to use in conjunction with the specific identifiers in order to make data globally unique

11.5 Classification of data elements

Data elements should be classified with the NeTEx built in enumerated values where appropriate. If NaPTAN, data is to be used in a “round-trip- exchange, i.e. to be imported back into a NaPTAN system, then it should also be annotated with a ***TypeOfValue*** instances indicating its NaPTAN classifications.

For example, NaPTAN ***StopPoint*** instances classified as ‘RLY’ and ‘RSE’ would be classified as NeTEx ***StopPlace*** components as follows.

```
    <StopPlace responsibilitySetRef="nptgAdminArea:110" version="1"
            <id="naptStop:9100ABDARE">
        <Name>Aberdare Rail Station</Name>
:::::
        <placeTypes>
            <TypeOfPlaceRef version="napt:v2.1" ref="napt:StopClassification@RLY"/>
            <!-- NaPTAN StopPoint.StopClassification.StopType -->
        </placeTypes>
        <TopographicPlaceRef ref="nptgLocality:E0054662" versionRef="1"/>
        <entrances>
            <StopPlaceEntrance id="naptStop:9100ABDARE0A" version="1">
                <Name>Aberdare Rail Station - Main Entrance</Name>
:::::::
                <placeTypes>
                    <TypeOfPlaceRef version="napt:v2.1"
ref="napt:StopClassification@RSE"/>
                </placeTypes>
:::::
            </StopPlaceEntrance>
        </entrances>
        <TransportMode>rail</TransportMode>
        <StopPlaceType>railStation</StopPlaceType>
        <quays>
```

See NaPTAN mappings in Part2 for further details.

11.6 Versions of the EIRE_NP

Future work on the EIRE_NP is likely to lead to some further refinement and extension of the profile to add new capabilities. While it is intended that any further changes will be fully backward compatible (such that existing documents can be used with the enhanced versions of the profile), if additions to the EIRE_NP are made in future, it will be useful for a consumer system to be able to determine on which version of the profile a dataset has been built.

Each successive version of the EIRE_NP is therefore given a formal version number, (this, the initial version, being v1.0). To state the version number being used in an EIRE_NP conformant document the ***version*** attribute on the ***TypeOfFrameRef*** must be used.

An EIRE_NP version number has the form **a.b** (a and b are integer and increase monotonically):

- **a**=main version.
- **b**=secondary version number.

The version number of the EIRE_NP profile being used on a frame should be stated on the ***TypeOfFrame*** reference that is mandatory for every frame in an EIRE_NP conformant document. The version number on the outer ***CompositeFrame*** is taken as the document's EIRE_NP version level. For example:

```
    <CompositeFrame version="1.4"
    id="epd:EI:FBRI:CompositeFrame:EI_PI_LINE_OFFER:FBRI@22">
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_LINE_OFFER"
    versionRef="eix:v1.0"/>
```

In line with normal NeTEx usage, the version of the NeTEx XML Schema being used should also be stated; this can be done in the ***version*** element on the ***PublicationDelivery*** element which contains the payload frames.

```
    <PublicationDelivery version="ntx:1.0"…
```

Note: The ***TypeOfFrame*** class is in effect additional EIRE_NP metadata that extends NeTEx and does not actually need not be included in the exchanged dataset – an external reference (with a ***versionRef*** rather than a ***version*** attribute) is sufficient (as above).

11.7 Dependencies between frames

Data of different types will necessarily be placed in different specific frames, and may well come from different producers and change at different latencies. For example, stop data may be exchanged within a ***SiteFrame*** as part of an '*EI_PI_STOP_OFFER',* while line data may be exchanged within a ***ServiceFrame*** together with schedule data in a ***TimetableFrame*** as part of an '*EI_PI_NETWORK_OFFER'*).

The dependencies between frames can be formally indicated by the ***prerequisites*** relationship on a ***VersionFrame***; this is useful in particular when the prerequisite frame is placed in a separate XML Document (so that internal integrity checking of individual object references to elements in the prerequisite frame must be suspended), as it gives an indication of the eternal data sets required to fully process the document. It is also helpful for manual inspection of a large data sets, for example a '*EI_PI_NETWORK_OFFER'* frame that contains many separate a '*EI_PI_NETWORK*' and '*EI_PI_TIMETABLE*' frames for different lines.

For the EIRE_NP and the EPIP, a version number should always be stated on a reference to a prerequisite version frame (i.e., in the ***version*** or ***versionRef*** attribute of the ***VersionOfFrameRef).***

Version Frames may be placed within an EIRE_NP document in any order.

11.8 Organisation and nesting of elements within an EIRE_NP document

The order and nesting of data elements in an XML document is largely dictated by the schema. Thus, for example, the schema definition for each specific version frame states which type of element it may contain and in which order, and each of these elements has subcomponents and attributes which must be nested within their parents only in a specified order.

However, in order to be flexible for different uses, the NeTEx schema (as discussed in section 8.1.10) allows some flexibility in the placing of referenced elements and in a number of cases they may either be specified inline where they are referenced, or be enclosed directly in the frame. An example is the use of a ***Notice***, which may either be declared in context, or be declared centrally in a frame and be referenced from elsewhere in the frame through a ***NoticeAssignment***. NeTEx also allows certain elements to be placed in more than one specific frame. (Note that the existence of alternatives does not present a technical difficulty to a consumer application, as mainstream XML parsing technology can be used to reassemble the document regardless of the organisation or elements).

To reduce the alternatives that implementors of the EIRE_NP need to support, in most cases a single alternative is specified in the profile. Thus:

— Each type of element is assigned to only one type of specific frame; that is, elements which are directly contained in a specific frame ("first class" elements)– their children will then be embedded in the same frame within their parent.) Note however, that generic subcomponents such as ***AlternativeName*** and ***AlternativeText*** may still appear in more than one type of frame in the context of their parent object).

- Ancillary objects are required to be embedded inline where they are first referenced (this also makes the document more readable for manual inspection). See tables in section. 11.9. For example:
  - ***TrainNumber*** instances are embedded within a ***ServiceJourney***.
  - ***ServiceFacilitySet*** instances are embedded with a ***ServiceJourney***.
  - ***SiteFacilitySet*** instances are embedded with a ***StopPlace***.
  - ***NoticeAssignment*** instances are embedded with a ***ServiceJourney (***or another parent***). Notices*** are embedded within the ***NoticeAssignment***.
- If an object is referenced many times over (as say for a commonly used ***Notice***), subsequent uses may reference the first use rather than embedding a repetitive copy.

This is specifically true for *NoticeAssignment* (always prefer the *Notice* to the *NoticeRef*).

11.9 Available types of frame

As explained in 8.3.3 the following TYPEs OF FRAME are available for use in the EIRE_NP:

Ancillary:

- *EI_PI_COMMON:* Used to exchange OPERATORs and other
- *EI_PI_CALENDAR:* Used to exchange DAY TYPEs of timetables

Payload:

- *EI_PI_STOP:* Used to exchange NaPTAN STOP Data and NPTG Localities
- *EI_PI_NETWORK: Used to exchange Stop data and LINEs for*
- *EI_PI_TIMETABLE*

Composite:

- *EI_PI_LINE_OFFER:* Used to exchange top and timetable for a single line.
- *EI_PI_NETWORK_OFFER:* Used to exchange top and timetable for a group of lines.
- *EI_PI_STOP_OFFER:* Used to exchange stop data for an area (like NaPTAN).

A further TYPE OF FRAME can be used to declare EIRE_NP specific metadata for validating the other frame types.

- *EI_PI_METADATA*

The following tables summarise the content of each TYPE OF FRAME.

11.9.1 Classes for SPECIFIC FRAMEs

The following notation is used:

Table 29 – Legend

	Unless nesting is indicated by an arrow, elements are contained directly in their frame type.

➔	Subelement is a component of preceding composite element: it must be embedded in-line within wrapper tags as per the NeTEx schema. (Additional arrows indicated additional subcomponent levels).
→	Subelement is an independent element referenced by another element: In EIRE_NP it should be nested within the referencing element that it annotates.
*	Parent is generic – the subelement may be nested within more than one type of element (If no parent is indicated, element is nested directly within a FRAME.
()	Classes in brackets are abstract
	The most important "payload" entities for EIRE_NP in each frame are highlighted in pink.
++	Elements in the Irish FXC profile but not in the EPIP are marked with a ++
'eix'	eix_metadata namespace =http:/netex.ieeix

EI_PI_ All Frames

Table 141 — TypeOfFrame: All

Name	**XML Type**	**Content XML Classes & Nesting**	**TM Entities & Composition**
classes	*ClassInContextRef*	*→***ValidityCondition*** *→***ValidBetween***	VALIDITY CONDITION [NeTEx only]
		*➔***AlternativeText***	➔ALTERNATIVE TEXT
		*➔***KeyValue***	[NeTEx only]
Data scope: ***ANCILLARY***	These data elements may be associated with any ***DataManagedObject*** in any EIRE_NP frame type.		

EI_PI_COMMON

Table 142 — TypeOfFrame: *EI_PI_COMMON*

Name	**XML Type**	**Content XML Classes & Nesting**	**TM Entities & Composition**
id	*TypeOfFrameIdType*	*'eix:EI_PI_COMMON'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***ResourceFrame***	RESOURCE FRAME
classes	*ClassInContextRef*	***ResponsibilitySet*** ➔***ResponsibilityRoleAssignment***	RESPONSIBILITY SET ➔RESPONSIBILITY ROLE ASSIGNMENT
		Operator ***Authority***	OPERATOR AUTHORITY
		ServicedOrganisation++	SERVICED ORGANISATION
		Branding	[NeTEx only]
		DataSource	DATA SOURCE
		ValueSet →(TypeOfValue)	[NeTEx only] TYPE OF VALUE
		VehicleType	VEHICLE TYPE
Data scope: ***ANCILLARY***	A ***ResourceFrame*** instance conforming to *'EI_PI_COMMON'* shall contain ancillary data elements referenced by the main payload frames *'EI_PI_STOP'* and/or *'EI_PI_TIMETABLE'*.		
Irish Data use	• For all data exchanges used to declare the RESPONSIBILITY SET (i.e., Admin areas being used) • When exchanging timetable data, used to define the OPERATORs, DAY TYPEs, SERVICED ORGANISATIONs, and VEHICLE TYPEs. • When exchanging fare data used to define the OPERATORs, and DAY TYPEs.		

EI_PI_CALENDAR

Table 143 — TypeOfFrame: *EI_PI_CALENDAR*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*'eix:EI_PI_CALENDAR'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***ServiceCalendarFrame***	SERVICE CALENDAR FRAME
classes	*ClassInContextRef*	***ServiceCalendar*** ➔***OperatingDay*** ➔***OperatingPeriod*** ➔***UicOperatingPeriod*** ➔***DayTypeAssignment***	SERVICE CALENDAR ➔OPERATING DAY ➔OPERATING PERIOD [NeTEx only] ➔DAY TYPE ASSIGNMENT
		→***DayType***	DAY TYPE
		➔*ServicedOrganisationDayType++*	SERVICED ORGANISATION DAY TYPE
Data scope: ***ANCILLARY***	A ***ServiceCalendarFrame*** instance conforming to '*EI_PI_CALENDAR'* shall contain ancillary data elements (DAY TYPES, etc) referenced by the payload ***TimetableFrame*** '*EI_PI_TIMETABLE*'.		
Irish Data use	• When exchanging timetable data, used to define the OPERATORs, DAY TYPEs, SERVICED ORGANISATIONs, and VEHICLE TYPEs. • When exchanging fare data used to define the OPERATORs, and DAY TYPEs.		

EI_PI_STOP

Table 144 — TypeOfFrame: *EI_PI_STOP*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*'eix:EI_PI_STOP'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***SiteFrame***	SITE FRAME
classes	*ClassInContextRef*	***StopPlace*** ➔***StopPlaceEntrance*** ***Quay*** ***FlexibleStopPlace++*** ➔***FlexibleArea++*** ➔***HailAndRideArea++*** ➔***FlexiblePointProperties++***	STOP PLACE ➔STOP PLACE ENTRANCE ➔QUAY FLEXIBLE STOP PLACE ➔ FLEXIBLE AREA ➔HAIL AND RIDE AREA ➔ FLEXIBLE POINT PROPERTIES
		TopographicPlace ➔***TopographicPlaceDescriptor*** ***Country***	TOPOGRAPHIC PLACE ➔TOPOGRAPHIC PLACE DESCRIPTOR COUNTRY
		PointOfInterest ➔***PointOfInterestEntrance++***	POINT OF INTEREST ➔ POINT OF INTEREST ENTRANCE
		*→***RoadAddress*** *→***PostalAddress***	ROAD ADDRESS POSTAL ADDRESS
		*→***SiteFacilitySet*** :::::➔***Facility***	SITE FACILITY SET ➔FACILITY
		*→***AccessibilityAssessment*** :::::➔***AccessibilityLimitation***	ACCESSIBILITY ASSESSMENT ➔ACCESSIBILITY LIMITATION
		GroupOfStopPlaces	(GROUP OF ENTITIES)
Data Scope: ***PAYLOAD***	A ***SiteFrame*** instance conforming to '*EI_PI_STOP' shall* contain stop data as required by the EIRE_NP ***CompositeFrame*** types *'EI_PI_LINE_OFFER'* or '*EI_PI_NETWORK_OFFER'***.** Thus, respectively (a) the STOP		

	PLACEs for a single LINE; or (b) the STOP PLACEs for a whole NETWORK. The network may be a specific regional network e.g., RATP, or a regional mode national, e.g., railway
Irish Data use	• When exchanging Stop data from NaPTAN, used to define the TOPOGRAPHIC PLACE taken from NPTG and the STOP PLACEs, taken from NaPTAN. • When exchanging Timetable data and fare data, used to define the SCHEDULED STOP POINTs from NaPTAN STOP POINTs and TARIFF ZONEs taken from NPTG or elsewhere.

EI_PI_NETWORK

Table 145 — TypeOfFrame: *EI_PI_NETWORK*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	'eix:*EI_PI_NETWORK*'	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***ServiceFrame***	SERVICE FRAME
classes	*ClassInContextRef*	***Line*** ➔***AllowedLineDirection*** ***FlexibleLine***	LINE ➔ALLOWED LINE DIRECTION FLEXIBLE LINE
		Direction	DIRECTION
		GroupOfLines ***Network***	GROUP OF LINES NETWORK
		TariffZone	TARIFF ZONE
		DestinationDisplay ➔***Via*** ➔***DestinationDisplayVariant***	DESTINATION DISPLAY ➔VIA ➔DESTINATION DISPLAY VARIANT
		RoutePoint	ROUTE POINT
		RouteLink	ROUTE LINK
		Route ➔***PointOnRoute*** ➔***SectionInSequence++*** →➔***GeneralSection (RouteLink)***	ROUTE ➔POINT ON ROUTE ➔SECTION IN SEQUENCE →➔GENERAL SECTION
		ScheduledStopPoint	SCHEDULED STOP POINT
		GeneralSection++ ➔***LinksOnSection++*** →➔***ServiceLink*** →➔ ***RouteLink***	SERVICE LINK ➔LINK ON SECTION SERVICE LINK ROUTE LINK
		➔*TransferDuration*** ***Connection*** ➔***ConnectionEnd*** ***SiteConnection*** ➔***SiteConnectionEnd*** ***DefaultConnection*** ➔***DefaultConnectionEnd***	(TRANSFER) CONNECTION [NeTEx only] SITE CONNECTION [NeTEx only] DEFAULT CONNECTION [NeTEx only]
		ServiceJourneyPattern ➔***StopPointInJourneyPattern*** ➔***SectionInSequence++*** →➔***GeneralSection (ServiceLink)***	SERVICE JOURNEY PATTERN ➔STOP POINT IN JOURNEY PATTERN ➔SECTION IN SEQUENCE →➔GENERAL SECTION
		ServicePattern	SERVICE PATTERN

		→StopPointInJourneyPattern *:::::*→**FlexiblePointProperties***	→STOP POINT IN JOURNEY PATTERN FLEXIBLE POINT PROPERTIES
		PassengerStopAssignment ***→TrainStopAssignment*** ***DynamicStopAssignment***	PASSENGER STOP ASSIGNMENT →TRAIN STOP ASSIGNMENT DYNAMIC STOP ASSIGNMENT
		****→Notice*** ****→Notice Assignment***	NOTICE NOTICE ASSIGNMENT
Data Scope: PAYLOAD	A ***ServiceFrame*** instance conforming to '*EI_PI_NETWORK'* shall contain schedule data for only a single LINE.		
Irish Data use	• When exchanging timetable data, used to define the SCHEDULED STOP POINTs, LINE, ROUTE and JOURNEY PATTERNs. • When exchanging fare data used to define the LINEs and VEHICLE JOURNEYs corresponding to the timetable entities.		

EI_PI_TIMETABLE

Table 146 — TypeOfFrame: *EI_PI_TIMETABLE*

Name	**XML Type**	**Content XML Classes & Nesting**	**TM Entities & Composition**
id	*TypeOfFrameIdType*	*'eix:EI_PI_TIMETABLE'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***TimetableFrame***	TIMETABLE FRAME
classes	>	***GroupOfServices*** ***→GroupOfServicesMember***	GROUP OF SERVICEs [NeTEx only]
		ServiceJourney ***→JourneyPart*** ***→TimetabledPassingTime*** ***→FlexibleServiceProperties*** ***TemplateServiceJourney*** ***→HeadwayJourneyGroup*** ***→ RhythmicalJourneyGroup++***	***SERVICE JOURNEY*** →JOURNEY PART ***→TIMETABLED PASSING TIME*** FLEXIBLE SERVICE PROPERTIES TEMPLATE SERVICE JOURNEY →HEADWAY JOURNEY GROUP →RHYTHMICAL JOURNEY GROUP
		****→ServiceFacilitySet***	SERVICE FACILITY SET
		JourneyPartCouple ***→JourneyPartPosition***	JOURNEY PART COUPLE [NeTEx only]
		ServiceJourneyInterchange	SERVICE JOURNEY INTERCHANGE
		VehicleType	VEHICLE TYPE
		****→TrainNumber***	TRAIN NUMBER
		Train ***→TrainComponent*** ****→TrainElement***	TRAIN →TRAIN COMPONENT TRAIN ELEMENT
		****→Notice*** ****→NoticeAssignment***	NOTICE NOTICE ASSIGNMENT (
Data Scope: PAYLOAD	A ***TimetableFrame*** instance conforming to '*EI_PI_TIMETABLE* 'shall contain schedule data for only a single LINE.		
Irish Data use	• When exchanging timetable data, used to define the SERVICE JOURNEYS. • When exchanging fare data used to define the LINEs and VEHICLE JOURNEYs corresponding to the timetable entities.		

11.9.2 Classes for COMPOSITE FRAMEs

EI_PI_LINE_OFFER

Table 147 — *TypeOfFrame for* *EI_PI_LINE_OFFER* - *(“EIRE_NP-LO”)*

Name	Type		Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*		*‘eix:EI_PI_LINE_OFFER’*	TYPE OF FRAME
FrameClassRef	*NameOfClass*		***CompositeFrame***	COMPOSITE FRAME
includes	*TypeOfFrameRef*	*1:1*	• *‘eix:EI_PI_COMMON’*	TYPE OF FRAME (RESOURCE FRAME)
		1:1	• *‘eix:EI_PI_STOP’*	TYPE OF FRAME (SITE FRAME)
		1:1	• *‘eix:EI_PI_NETWORK’*	TYPE OF FRAME (SERVICE FRAME
		1:1	• *‘eix:EI_PI_TIMETABLE’*	TYPE OF FRAME (TIMETABLE FRAME
		0:1	• *‘eix:EI_PI_CALENDAR’*	TYPE OF FRAME (SERVICE CALENDAR FRAME
Data Scope: CONTAINER	A ***CompositeFrame*** instance conforming to ‘*EI_PI_LINE_OFFER’* shall contain stop and timetable data for only a single LINE.			
Irish Data use	• Used to exchange a single timetable data			

EI_PI_NETWORK_OFFER

Table 148 — TypeOfFrame for** EI_PI_NETWORK_OFFER **– (“**EIRE_NP-NO**”)

Name	Type		Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*		*‘eix:EI_PI_NETWORK_OFFER’*	TYPE OF FRAME
FrameClassRef	*NameOfClass*		***CompositeFrame***	COMPOSITE FRAME
includes	*TypeOfFrameRef*	*1:1*	*‘eix:EI_PI_COMMON’*	TYPE OF FRAME (RESOURCE FRAME)
		*1:**	*‘eix:EI_PI_STOP’*	TYPE OF FRAME (SITE FRAME)
		*1:**	*‘eix:EI_PI_NETWORK’*	TYPE OF FRAME (SERVICE FRAME)
		*1:**	*‘eix:EI_PI_TIMETABLE’*	TYPE OF FRAME (TIMETABLE FRAME
		0:1	*‘eix:EI_PI_CALENDAR’*	TYPE OF FRAME (SERVICE CALENDAR FRAME
Data Scope: CONTAINER	A ***CompositeFrame*** instance conforming to ‘*EI_PI_NETWORK_OFFER*’ shall include stop and timetable information for two or more LINEs that make up a transport network.			
Irish Data use	Used to exchange multiple timetables as a single document.			

EI_PI_STOP_OFFER

Table 149 —

Table 150 — TypeOfFrame for** EI_PI_STOP_OFFER **– (“**EIRE_NP-SO**”)

Name	Type		Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*		*‘eix:EI_PI_STOP_OFFER’*	TYPE OF FRAME
FrameClassRef	*NameOfClass*		***CompositeFrame***	COMPOSITE FRAME

includes	*TypeOfFrameRef*	*1:1*	*'eix:EI_PI_COMMON'*	TYPE OF FRAME (RESOURCE FRAME)
		*1:**	*'eix:EI_PI_STOP'*	TYPE OF FRAME (SITE FRAME)
		*1:**	*'eix:EI_PI_NETWORK'*	TYPE OF FRAME (SERVICE FRAME)
Data Scope: CONTAINER	A ***CompositeFrame*** instance conforming to '*EI_PI_STOP_OFFER*' shall include one or more SiteFrame instances of type '*eix:EI_PI_STOP'* each giving information about stops for one or more MODEs, OPERATORs or ADMINISTRATIVE AREAs. Large data sets may be partitioned into multiple ***SiteFrame*** instances. an operator, region. The EIRE_NP also allows the inclusion of a '*EI_PI_NETWORK'* frame so that STOP ASSIGNMENTs may be used			
EIRE Data use	Used to exchange stop data from one or more NaPTAN areas as a single document.			

11.9.3 Classes for EIRE_NP Metadata FRAMEs

EI_PI_METADATA

Table 151 — TypeOfFrame: *EI_PI_METADATA*

Name	**XML Type**	**Content XML Classes & Nesting**	**TM Entities & Nesting**
id	*TypeOfFrameIdType*	*'eix:EI_PI_METADATA'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***ResourceFrame***	RESOURCE FRAME
classes	*ClassInContextRef*	***ResponsibilitySet*** ➔***ResponsibilityRoleAssignment***	RESPONSIBILITY SET ➔RESPONSIBILITY ROLE ASSIGNMENT
		TypeOfFrame	TYPE OF FRAME
		ValueSet →***TypeOfPlace***	[NeTEx only] TYPE OF PLACE
Data scope: ***ANCILLARY***	A ***ResourceFrame*** instance conforming to '*EI_PI_METADATA'* shall contain definitions of the fixed ***TypeOfFrame*** instances and ***TypeOfPlace*** instances defined by the EIRE_NP.		
EIRE Data use	Used to include profile metadata for validating Irish NPTG, NAPTAN, and timetable code values.		

11.10 Handling large data sets

Some data sets can be very large; for example, there will be several hundred thousand bus stops in an average sized European country, and even the timetable for a single line can be extensive; an urban bus route might have over 150 stops, and operate many different journey patterns on many different day types in two different directions; consequently, the network timetable for a large city can also be large (10+ MB).

Beyond a certain size, data files of any sort become hard to process and to transmit over the internet and so, for scalability, a modular approach to creating and handling data sets is needed.

The use cases for applications using data, together with the efficiency of the workflows that collect and distribute data, are also relevant for scaling decisions. For example, there is no point in aggregating data into a single large file (say all the bus stops in a country) if most users are only using a small subset of the data that concerns just their local area and which originates and can be distributed locally much more efficiently.

NeTEx offers great flexibility in modularising datasets and the three EIRE_NP or EPIP 'OFFER' frame types ('*EI_PI_STOP_OFFER', 'EI_PI_LINE_OFFER'* and '*EI_PI_NETWORK_OFFER*') have been chosen so that they can be used at different scales as needed. The following guidelines apply:

- Large data sets should be broken down into multiple files, for example by LINE (for timetables) or by area (for stop data exchanged without timetables). Thus, the likely size of the data set should be taken into consideration in choosing which type of EIRE_NP or EPIP 'OFFER' frame to use, and in deciding the selection parameters for the frame.
- Within an EIRE_NP or EPIP conformant document, data should be modularised into frames. For example, with an '*EI_PI_NETWORK_OFFER*' composite frame that holds data for multiple lines, the SERVICE PATTERNs for each LINE in the NETWORK should be placed in a separate ***ServiceFrame*** of EPIP type 'EI_PI_NETWORK' frame and the SERVICE JOURNEYs making up the schedule for each LINE should be placed in a separate ***TimetableFrame*** of EIRE_NP type '*EI_PI_TIMETABLE*'. The 'OFFER' frame will thus contain as many pairs of such 'NETWORK' and 'TIMETABLE' frames as there are lines in the network. Similarly, the stops in a 'EI_PI_STOP_OFFER' file should be broken down by administrative zone or operator into separate ***SiteFrame*** instances of EPIP Type '*EI_PI_STOP.'* The frame identifiers should indicate their respective contents (in the [***frame-topic***] parameter]. The ***prerequisites*** relationship can be used to indicate the specific dependencies between frames.
- Where data is split between files, the ***VersionFrame*** \ ***prerequisites*** relationship should be used to indicate dependencies on specific external data sets.

Note that the dimensions of different modes are very different (rail networks typically have far fewer stops and services) and so should be scaled differently. For example, in the UK, bus stop data (350,000+ instances) is packaged into separate XML documents for each administrative area (of which there are 140 or so) and is mainly used locally, while rail stop data (with only around 3,000 stops) is packaged into a single file for the whole country and is often used in a nation-wide context.

11.11 Responsibility for data

EIRE_NP and EPIP conformant documents should indicate the organisation responsible for providing data, using the ***responsibilitySetRef*** attribute from ***DataManagedObject***; a frame may contain data from multiple providers and different instances of ***ResponsibilitySet***. A reference to a ***ResponsibilitySet*** should be specified at least on the outermost container frame.

The OPERATOR or the AUTHORITY providing the data should be specified in the EI_PI_COMMON frame. This may also include one or more definitions of a ***ResponsibilitySet*** that indicate the roles.

NeTEx XML EXAMPLE

```
<CompositeFrame version="3.0" id="epd:EI:NTA:CompositeFrame_IE_PI_STOP_OFFER:822:NaPTAN"
                responsibilitySetRef=”nptgAdminArea:822”>
    <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_STOP_OFFER:EIRE_NP"
versionRef="eix:1.0" />
    <FrameDefaults>
        <DefaultCodespaceRef ref="naptan_data"/>
        <DefaultResponsibilitySetRef version=”any” ref="nptgAdminArea:822"/>
    </FrameDefaults
    <frames>
        <SiteFrame version="2.1" id="epd:EI:TFL:SiteFr:EIRE_NP»»
ame_IE_PI_STOP:822:NaPTAN">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_STOP:EIRE_NP"
versionRef="eix:1.0" />
                <Etc., etc., Specific Frame contents>
        </SiteFrame>
        <ResourceFrame version="2.1"
id="epd:EI:TFL:ResourceFrame_IE_PI_COMMON:822:NaPTAN">
            <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_COMMON:EIRE_NP"
versionRef="eix:1.0" />
                <Etc., etc., Specific Frame contents>
            <responsibilitySets>
                <ResponsibilitySet version="any" id= nptgAdminArea:822"
                    responsibilitySetRef="nptg:RS_nptg" >
        <Name>NTA provides Dublin data</Name>
                        <roles>
        <ResponsibilityRoleAssignment version="any" id="nptgAdminArea:822">
            <Description>Dublin data - Managed centrally</Description>
            <DataRoleType>collects</DataRoleType>
            <ResponsibleOrganisationRef version ref="noc:NTA"/>
            <ResponsibleAreaRef versionRef="any"  id="nptg:AdministrativeZone:822"/>
        </ResponsibilityRoleAssignment>
                        </roles>
                </ResponsibilitySet>
        </ResourceFrame>
                <Etc., etc., Specific Frame contents>
```

11.12 Completeness of data

NeTEx can in principle be used to exchange either a complete data set, or just the changes to it (a “delta”), as indicated by the ***modification*** attribute on ***DataManagedObject***. For a delta, just the elements and the attributes that have changed are included (though if a child component changes its parent must be included too so as to give a context).

> For simplicity, in the EIRE_NP only complete data sets are exchanged, replacing any previous versions in their entirety; deltas are not exchanged. Thus, the ***modification*** attribute value should one of be ‘*new*’, ‘*revise*’, or ‘*unchanged*’, not ‘*delta*’, or ‘*delete*’. The ***modification*** attribute only needs to be specified on the outermost container object – e.g., the ***CompositeFrame***, though it is acceptable to also include values for it on individual objects.

Within the frame, all values required by EIRE_NP for each element should be populated. A producer making an export must therefore handle four different cases:

1. Elements and attributes which are **mandatory in the NeTEx schema** (so must always be present).
2. Elements and attributes which are **optional in the NeTEx schema** but **mandatory in the EIRE_NP** (so shall always be populated).
3. Elements and attributes which are **optional in the EIRE_NP**; these should be populated if the producer holds a value, but should be omitted if the producer system holds no value (i.e., a null or zero value or an empty set). See discussion of ‘Empty values’ below.
4. Elements and attributes which are **outside the EIRE_NP**; these should follow the same principles as used for (3).

11.12.1Empty values

Many NeTEx elements and attributes are not mandatory. This leads to the question of what should appear in an XML document for them when there is no information held for them in a producer system?

In XML, if a field is optional and there is no value for it, three different actions are possible:

1) Omit the element altogether – i.e., have no XML tag. (This is the treatment required by EIRE_NP).

2) Include the element with no content between the start and end tags, for example: <**Name**></**Name**>. (Note that this is not allowed for attributes that have enumerated values).

3) Include the element as an empty tag, e.g., <***Name***/>. Syntactically this is identical to (2).

When importing a **new** data set, the consumer system will only populate its repository with the values that are present in the export. Note however that if an attribute has a default value in the schema, this will be assumed for it.

When importing an **update to an existing data set** that is a **complete** export (as is always the case for the EIRE_NP) - rather than, say, just a delta of changes) - a consumer system must not only update the content it holds with any values that are present in the export, but also remove any existing values it holds for attributes that were present in the first export but are now absent. In other words, a complete export replaces all the content of existing elements on the consumer system, including any dependent children it contains, and any references it holds to other entities.

Note that this rule could lead to loss of data in a consumer system if it aggregates data for the same element from two different producer systems, the second of which is only sparsely populated. In practice in such a case a consumer system might chose to create a new version of the element that merges the data attributes – however this requires a process to determine the correct action to take

In contrast, if a system is updating from a delta (unlike in the EIRE_NP) it only replaces the elements that are provided.

.

11.13 Data exchange

EIRE_NP data exchange will be implemented through the export and consumption of files that are XML documents. Each file will contain a single **PublicationDelivery,** itself containing a single version frame of one of the permitted EIRE_NP types).

Three different types of EIRE_NP conformant XML document will be accepted as files:

- **Line** XML documents containing a single '*EI_PI_LINE_OFFER*' ***CompositeFrame*** (describing the stops, topology and timetable of a single line).

- **Network** XML documents containing a single '*EI_PI_NETWORK_OFFER*' ***CompositeFrame*** (providing the stops, topology and timetable of a multiple lines) Within the ***CompositeFrame*** there may be multiple instances of the different EIRE_NP specific frame types, for example a separate ***TimetableFrame*** for each LINE.

- **Stop** XML documents containing a single '*EI_PI_STOP_OFFER*' ***CompositeFrame*** (describing all the stops of a country, an authority or an operator). It will contain one or more ‘*EI_PI_STOP*’ ***SiteFrame*** instances; large groups of stops may be broken down by area, operator or mode.

Note that the same data may be required in more than one data set, so there will be a degree of redundancy across multiple files, in particular for elements such as OPERATOR, STOP PLACE, CONNECTION, that are used in many lines and timetables. This is, of course, valid and normal; a consumer can use the version numbers and/or **changed** attributes to determine the most recent version.

11.14 PublicationDelivery

The NeTEx publication schema (*netex.publication.xsd*) describes a ***PublicationDelivery*** element that is suitable for exchange either as a stand-alone XML document, or embedded in a SIRI ***Delivery*** (that is the response to a SIRI request for NeTEx data). Only the stand-alone document is used for the EIRE_NP.

Thus, an EIRE_NP conformant document wraps a single ***PublicationDelivery*** element around one or more “payload” ***CompositeFrame*** instances of the required EIRE_NP ***TypeOfFrame***. The ***PublicationDelivery*** acts as a container and simple header and is annotated with a time stamp.

A ***PublicationRequest*** element may be used to indicate the ***Line, Network*** or topographic region to which the data applies.

Other NeTEx elements not included in the EIRE_NP profile may also be present (see discussion earlier) to augment or extend the content. It is also acceptable to include a ***ResourceFrame*** with EIRE_NP metadata instances (i.e., an EI_PI_METADATA) that can be used to validate the references to EIRE_NP ***TypeOfFrame*** and ***TypeOfValue*** instances.

Table 152 — *PublicationDelivery* –Element

Classification	Name	Type	Cardinality	Description
	version	*xsd:NMTOKEN*		Version identifier of NeTEx schema being used. A NeTEx ***Codespace*** prefix, e.g., “ntx:1.1” may be included to help readers understand that the version is the schema version “ ntx:1.1” and not the version of the EIRE_NP or of the data itself.
	PublicationTimestamp	*xsd:dateTime*	1:1	Time of output of data. Preferred format UTC. For example, *2017-07-26T13:52:00Z*
«FK»	***ParticipantRef***	*ParticipantCodeType*	1:1	Identifier of system providing data. Often this will be the same as the DATA SOURCE but it is possible there may be multiple participant systems belonging to a single data source.
«cntd»	***PublicationRequest***	*PublicationRequest*	0:1	Echo Request used to create bulk response. As a statement of intent as to what should be in the frame, it can be used to provide a high-level summary of the frame’s contents as parameters that can be used for manual and automated validation.
	PublicationRefresh-Interval	*xsd:duration*	0:1	How often data in publication is normally refreshed. This is only relevant for regular period batch exports, for example a weekly timetable update.
	Description	*xsd:normalizedString*	0:1	Description of contents.

«cntd»	***dataObjects***	*dataObjects_RelStructure*	0:1	NeTEx VERSION FRAMEs making up publication.

11.14.1Publication Request (Subcomponent of PublicationDelivery)

The ***PublicationRequest*** records which NeTEx objects were requested for inclusion in the document and the level of detail provided for them.

For EIRE_NP the ***PublicationRequest*** states the ***Line*** or ***Network*** whose data is contained in the frame**.**

Table 153 — *PublicationRequest* – Service Element

Classification	Name	Type	Cardinality	Description
	RequestTimestamp	*xsd:dateTime*	1:1	Time of request.
«FK»	***ParticipantRef***	*ParticipantCodeType*	0:1	Identifier of system requesting data.
	Description	*MultilingualString*	0:1	Text description of the filter.
«cntd»	***topics***	*NetworkFrameTopic*	0:*	One or more request filters that describe selection criteria for data to be included in output.
«cntd»	***RequestPolicy***	*NetworkFrameRequestPolicyStructure*	0:1	Policies to apply when fetching data specified by Topics.
«cntd»	***SubscriptionPolicy***	*FrameSubscription-PolicyStructure*	0:1	Policies to apply when subscribing to data specified by Topics.

11.14.2NetworkFrameTopic (Subcomponent of PublicationDelivery)

The ***NetworkFrameTopic*** specifies the selection criteria for including NeTEx objects in the publication.

For EIRE_NP only the ***NetworkFilterByValue*** is used in order to indicate the **Line** and/ or ***Network***. It should also indicate the region (e.g., the area for which stop data is provided, or in which a network lies) as a ***TopographicPlace***

Table 154 — *NetworkFrameTopic* – Service Element

Classification	Name		Type	Cardinality	Description
	Description		*MultilingualString*	0:1	Text description of the filter.
«cntd»	***sources***		*DataSourceRef*	0:*	DATA SOURCE for data.
«FK»	***CodespaceRef***		*CodespaceRef*	0:1	CODE SPACE for data to select.
«cntd»	***ResponsibilityRoleAssignment***		*ResponsibilityRole-Assignment*	0:1	RESPONSIBILITY ROLE ASSIGNMENT to use or selection criteria for data to be included in output.
	Temporal scope		CHOICE		
	a	***Current***	*EmptyType*	1:1	Return just the values from the producer current at the time the response is created.
»	b	***ChangedSince***	*xsd:dateTime*	1:1	Return the values that have changed since the specified point in time, including current and superseded values.
	c	***CurrentAt***	***ChangedSince***	*xsd:dateTime*	Return just the values that were current at the specified point in time.

«cntd»	d	***HistoricBetween***	*ClosedTimestamp-RangeStructure*	1:1	Return all values created or updated between the specified period, including current and historic values.
«cntd»	e	***selectionValidityConditions***	*ValidityCondition*	1:*	Validity conditions to apply when selecting data. Applies to frame.
«cntd»		***TypeOfFrameRef***	*TypeOfFrame-RefStructure*	0:1	Reference to a TYPE OF VERSION FRAME.
			CHOICE		
«FK»	a	***VersionFrameRef***	*VersionFrame-RefStructure*	1:*	Reference to a VERSION FRAME.
«cntd»	b	***NetworkFilterByValue***	*NetworkFilterByValueStructure*	1:0	Values for selecting by value

11.14.3 *NetworkFilterByValue* (Subcomponent of PublicationDelivery)

The ***NetworkFilterByValue*** specifies the selection criteria for including NeTEx objects in the publication.

Table 155 — *NetworkFilterByValue – XML Element*

Classification	Name	Type	Cardinality	Description
«FK»	***LayerRef***	*LayerRef*	0:1	Area containing objects to be returned.
«cntd»	***BoundingBox***	*BoundingBoxStructure*	0:1	Area containing objects to be returned.
«cntd»	***objectReferences***	*(VersionOfObjectRef)*	0:*	Reference to ENTITies to be returned. Use REF with specific value to identify an object. Use REF with blank value to specify all object of the object type. Use TypeOfValue to specify categories of object to be included. In EIRE_NP use to state the LINE if type of CompositeFrame is for EI_LINE_OFFER
cntd»	***NetworkRef***	*NetworkRefStructure*	0:1	Reference to a NETWORK.
cntd»	***places***	*(PlaceRef)*	0:*	Return all elements concerned with given place. EIRE_NP can state Topographical Place e.g., Country, region or area

11.14.4 PublicationDelivery example

The following fictional example shows a ***PublicationDelivery*** header for a ‘EI_PI_LINE_OFFER’ for *Phoebus Line 1*. The frames making up the delivery are also indicated but their contents are omitted.

NeTEx XML EXAMPLE:

```
<PublicationDelivery version="ntx:1.1"
    xsi:schemaLocation=http://www.netex.org.uk/netex NeTEx_publication.xsd
    xmlns="http://www.netex.org.uk/netex"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:ifopt="http://www.ifopt.org.uk/ifopt"
    xmlns:gml=http://www.opengis.net/gml/3.2
    xmlns:core=http://www.govtalk.gov.uk/core
    xmlns:siri=http://www.siri.org.uk/siri
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <PublicationTimestamp>2001-12-17T09:30:47.0Z</PublicationTimestamp>
        <ParticipantRef>mysis:SYS001</ParticipantRef>
        <PublicationRequest version="1.0">
            <RequestTimestamp>2001-12-17T09:30:47.0Z</RequestTimestamp>
            <topics>
                <NetworkFrameTopic>
                    <Current/>
                    <NetworkFilterByValue>
                        <objectReferences>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_LINE_OFFER:EIRE_NP"
versionRef="eix:v1.0"/>
                            <OperatorRef version="1.0" ref="noc:PHBS"/>
                            <LineRef ref="phb:Line@1"/>
                        </objectReferences>
                        <places>
                            <CountryRef ref=“ie”/>
                            <TopographicPlaceRef
                                 ref="nptgLocality: E0822022"/>
                        </places>
                    </NetworkFilterByValue>
                </NetworkFrameTopic>
        </topics>
        </PublicationRequest>
        <dataObjects>
            <CompositeFrame version="3.0"
id="epd:EI:PHBS:CompositeFrame_IE_PI_LINE_OFFER:Line-1:phb”
                        reponsibilityetRef="phb:timetable_data"/>
                <ValidBetween>
                    <FromDate>2019-01-01T09:02:30.0Z</FromDate>
                    <ToDate>2020-01:01T00:02:30.0Z</ToDate>
                </ValidBetween>
    <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_LINE_OFFER:EIRE_NP"
versionRef="eix:v1.0"/>
                <codespaces>
                    <Codespace id="phb_data">
                        <Xmlns>phb</Xmlns>
                        <XmlnsUrl>http://www.phoebus.ei/</XmlnsUrl>
                        <Description>Phoebus data</Description>
                    </Codespace>
                    <CodespaceRef id="eix_metadata">eix</CodespaceRef>
                    <CodespaceRef id="epip_data">epd</CodespaceRef>
                </codespaces>
                <FrameDefaults>
                    <DefaultResponsibilitySetRef ref="phb:timetable_data"/>
                    <DefaultCodespaceRef ref="eix_data"/>
                </FrameDefaults
                <frames>
                    <SiteFrame version="2.1"
id="epd:EI:PHBS:SiteFrame_IE_PI_STOP:Line@1:phb”>
                        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_STOP:EIRE_NP"
versionRef="eix:v1.0"/>
                                <prerequisites>
            <ResourceFrameRef version="any"
ref="epd:EI:PHBS:ResourceFrame_IE_PI_COMMON:Line@1"/>
                                </prerequisites>


        <Etc., etc., Specific Frame contents>
```

```
                    </SiteFrame>
                <ServiceFrame version="2.1"
id=“epd:EI:PHBS:ServiceFrame_IE_PI_NETWORK:Line@1:phb”>
    <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_NETWORK:EIRE_NP"
versionRef="eix:v1.0"/>
                                <prerequisites>
            <ResourceFrameRef version="any"
ref="epd:EI:PHBS:SiteFrame_IE_PI_COMMON:Line@1"/>
                                </prerequisites>

        <Etc., etc., Specific Frame contents>
                    </ServiceFrame>
            <TimetableFrame version="2.4"
id="epd:EI:PHBS:TimetableFrame_IE_PI_TIMETABLE:Line@1:phb”>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_TIMETABLE:EIRE_NP"
versionRef="eix:v1.0"/>
                                <prerequisites>
            <ServiceFrameRef version="any"
ref="epd:EI:PHBS:ServiceFrame_IE_PI_COMMON:Line@1"/>
            <ServiceCalendarFrameRef version="any"
    ref=”epd:EI:PHBS::ServiceCalendarFrame_IE_PI_CALENDAR:Line@1:phb"/>
                                </prerequisites>

        <Etc., etc., Specific Frame contents>
                    </TimetableFrame>
    <ServiceCalendarFrame version="1.6"
id="epd:EI:PHBS:ServiceCalendarFrame_IE_PI_CALENDAR:Line@1”>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_CALENDAR:EIRE_NP"
versionRef="eix:v1.0"/>
        <Etc., etc., Specific Frame contents>
                    </serviceCalendarFrame >
                <ResourceFrame version="1.6"
id="epd:EI:PHBS:ResourceFrame_IE_PI_COMMON:Line@1:phb”>
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_COMMON:EIRE_NP"
versionRef="eix:v1.0"/>
        <Etc., etc., Specific Frame contents>
                    </ResourceFrame>
                    <ResourceFrame version="1.0"
id="eix:EI:NTA:ResourceFrame_IE_PI_METADATA:FXC">
        <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_METADATA:EIRE_NP"
versionRef="eix:v1.0"/>      <Etc., etc., Specific Frame Fixed contents>
                    </ResourceFrame>
                </frames>
            </CompositeFrame>
    </dataObjects>
</PublicationDelivery
```

11.15 Names of files for EIRE_NP documents

It is helpful to developers and automated workflow applications if the names used for XML document give an indication of their contents and are uniform (and unique).

The following file name structure (which is the same as for the European profile) is recommended for any file disseminating EIRE_NP conformant documents:

[prefix] - [EIRE_NP-version] _ [country-code] _ [provider-code] _ [profile-type] _ [doc-topic] _[creation-date]

Where:

- **[prefix]** A fixed prefix agreed upon by the exchange partners, here ‘FX-PI ‘
- **[EIRE_NP-version]** Major version number of the EIRE_NP; currently ‘01’
- **[country-code]** Country ISO code.
- **[provider-code]** Producer code (use ‘NAP’ if coming from a national access point). Variable length. Normally this will be the same as the provider code on the composite frame within the NeTEx document.
- **[profile-type]** EIRE_NP Offer Profile type (LINE, NETWORK or STOP).
- **[doc-topic]** Topic according to profile subtype (as for the frame topic). All names are (short name, fewer than 15 characters, made only of letters (uppercase and lowercase), numbers and hyphen "**-**"). The EIRE_NP [doc-topic] should correspond to the [frame-topic] specified in 11.4.3 and the publication header (11.14). ‘+’ can be used to link multiple topics.
  - For ‘*EI_PI_NETWORK_OFFER’*, short name or code of network;
  - For ‘*EI_PI_LINE_OFFER’*: LINE number or name if no number is available
  - For ‘*EI_PI_*STOP_*OFFER’*, name or code of LINE, NPTG ADMINISTRATIVE ZONE or TOPOGRAPHIC PLACE for which the stops are included.
- **[creation-date]** Date of creation of the XML document (YYYMMDD). If multiple versions need to be distinguished within the same day a time-HHMM can be appended.
- All fields are separated as indicated either by a hyphen “-“ or by an underscore "_".

Example file names:

```
NX-PI-01_IE_NTA_STOP_NaPTAN-AdminArea-110_20160302.xml
NX-PI-01_IE_NTA_STOP_NaPTAN-AdminArea-822+847+110_20230909.xml
NX-PI-01_IE_NTA_STOP_NpTG-EI_20150705.xml

FX-PI-01_IE_PHBS_LINE_PHBS-1_20200704.xml
FX-PI-01_IE_ FCRK_LINE_ FCRK-48_20200704.xml
FX-PI-01_IE_FBRI_NETWORK_FBRI-WOE_20190202.xml
FX-PI-01_IE_ FCRK_LINE_ FCRK_20200704.xml

FX-PI-01_IE_TBD_STOP-OFFER_WIMBLEDON-accessibility_20148221.xml

NX-PI-01_GE_BVB_LINE-3_20161201.xml
NX-PI-02_FR_FR1_NETWORK_RATP_20171105.xml
NX-PI-01_IE_NTA_STOP_AdminArea-470+471_20171212.xml
```

11.16 Extending the profile

The choices made by the EIRE_NP-NT deliberately limit-what can be exchanged to the minimal subset of NeTEx needed for exchanging a timetable (and the EIRE_NP-FM to the elements needed to describe a fare tariff, products and prices). In practice, it may be that the EIRE_NP mostly fits a particular intended use (and that the EIRE_NP rules and principles are helpful to adopt), but certain extensions are required to meet slightly different local needs or a particular application.

Since the profile is merely an agreement on how to use the NeTEx and not a modification to the underlying schema, the full capabilities of NeTEx remain available for use as local extensions.

However, if users wish to deploy EIRE_NP based data sets with local extensions, but still keep the data set usable as an EIRE_NP conformant document that is readable by any EIRE_NP capable consumer system, the extensions should be used in a manner consistent with the EIRE_NP's principles and conventions. Therefore, the following rules shall apply to any extensions to the profile.

- For the addition of further NeTEx **attributes** not selected by the EIRE_NP, (i.e., Attributes available in the NeTEx schema for object types that are already selected by the EIRE_NP):
  - Excluded attributes are allowed, but the object semantic and consistency of the conformant data shall not be harmed when these attributes are ignored.
- For the addition of further NeTEx **object types** not selected in the profile (but available in the NeTEx schema):
  - It shall not be mandatory for additional objects to be present in order for the EIRE_NP part of the data set to be consistent (i.e., the EIRE_NP data in the profile's frames shall remain fully self-consistent even if there are some extensions).
  - Instances of any additional object types should preferably be placed in a separate VERSION FRAME; this may even be in a separate file. The frame should be given a ***TypeOfFrame*** with a persistent identifier and of a specified version level.
  - If placed in the same frame as the EIRE_NP conformant contents (as will necessarily be the case for small extensions, as when using an extra subcomponent or attribute that require the context of an existing object), the additional objects should follow the same behavioural rules as the objects in the profile as to identifiers, versioning, etc.
- For further custom **extensions** to NeTEx (for features not covered by the NeTEx schema):
  - The normal EIRE_NP principles should be applied (e.g., for stable unique identifiers, etc.)
  - Only the ***KeyValue*** extension mechanism is accepted (not the ***Extensions*** tag).
  - The ***KeyValue*** extension shall never be used if there is a corresponding NeTEx concept or attribute that expresses the same meaning – the native NeTEx mechanism should be used in preference.

We may thus characterise three ways of using an EIRE_NP profile (which might be assessed and noted by a profile conformance validator tool) as follows:

1. **Strict conformance:** populating only those element types and attributes specified in the profile.
2. **Augmented conformance:** populating only those element types specified in the profile but also possibly (i) extra attributes (including extra ***TypeOfValue*** instances with which to classify EIRE_NP elements), (ii) using additional enumeration values, (iii) adding ***KeyValue*** instances.
3. **Extended conformance:** Adding additional NeTEx object types as well as augmenting the existing EIRE_NP classes.

A producer system can indicate the nature of the intended conformance of a frame using a ***KeyValue***.

NeTEx XML EXAMPLE:

```
    <SiteFrame version="2.3" id="epd:EI:PHBS:SiteFrame_IE_PI_STOP:PHBS@1:ohb">
        <ValidBetween>
                <FromDate>2010-12-17T09:30:47.0Z</FromDate>
                <ToDate>2011-12-17T09:30:47.0Z</ToDate>
        </ValidBetween>
        <keyList>
            <KeyValue>
                <Key>epip:conformance</Key>
                <Value>strict</Value>
            </KeyValue>
        </keyList>
        <TypeOfFrameRef ref="eix:EI:NTA:EI_PI_STOP" versionRef="eix:1.0" />
    <TypeOfFrameRef ref="eix:EI:NTA:TypeOfFrame:EI_PI_STOP:EIRE_NP"
versionRef="eix:v1.0"/>
```

12 Profile compliance and data quality checking

The checking of profile conformance and of data quality is extremely important for achieving successful interoperability. Data producers need a means to verify that their output is correct and usable. Data consumers need to know that any EIRE_NP compliant document meets a reasonable set of quality criteria before they even attempt to use it. There also needs to be a process for resolving issues in exported data with the producer.

12.1 Methods of checking the quality of data

There are three main methods of checking data quality:

1. **XML validator:** A general purpose XML tool that automatically checks a document against the rules in an XML schema ("XSD") as to the use of tags, data types, referential integrity, etc. XML validators are not specific to NeTEx – they are included in most XML editors and in tools such as JAXB that are used to parse an XML document when importing it into a computer system. Thus, they can be used "off the shelf" and require no extra development effort. The result of an XML validation is binary – either a document passes or it fails. It is not advisable to import a document that has failed validation as it may not even parse correctly. An EIRE_NP document must validate against the *NeTEx_publication.xsd* schema using any mainstream validator.

2. **Profile validator:** A software tool specially written for a particular schema that reads a profile compliant document that has passed XML validation and checks for additional rules that are not enforced by the XML validator. (They are not enforced either because XML does not support such a check; or because the rule does not always apply and so an XML mechanism is not used since the applicability of XML constraints cannot be conditional).

    — There may be different degrees of severity associated with different profile rules, e.g., "fatal error", "minor error", "warning", etc.; and it may still be possible to import a document containing non-fatal errors into a consumer system and even to correct certain types of error automatically while doing so. For example, a validator could flag whether conformance of a given document was strict or augmented, or lacks particular attribute values required by the profile but not by the schema.

- A profile validator may apply many different types of quality rule, some of general applicability to the schema and some very specific to its business requirements. See further below.

3. **Hand-checking**. It can still be useful to inspect a document by hand, in particular when resolving conformance or quality issues between producer and consumer. A major advantage of XML over many other data formats (such as CSV or a binary representation) is that XML documents have a well organised structure and label every data element with meaningful tags, making it relatively easy for a human to read them. Most XML document editors include sophisticated formatting and search capabilities for examining documents. Organising and nesting the elements so that related elements are close together and in contexts can further improve readability – the EIRE_NP nesting rules are intended to facilitate this. The inclusion of names and labels on references can also help, and is recommended in particular for external references.

12.2 Quality issue resolution

Regardless of the method, a consumer who detects an issue in data will typically need to inform the relevant originator of the data about the issue in order to get it resolved (the originator is not necessarily the producer, who may just be acting as an aggregator of data coming from other sources). A ***DataSource*** with a contact email shall be specified at the frame level and more specifically if needed. A ***ResponsibilitySet*** shall be included to assign responsibility to a named organisation. The ***ContactDetails*** for an ***Operator*** or ***Authority*** must be populated.

12.3 Profile validator rules

Profile validators can vary in their degree of generality; for example, they may check just the rules inherent in a semantic model that apply to any document implementing a given schema, or also those additional rules that apply to a particular profile such as the EIRE_NP.

We can distinguish four levels of generality in such rules that a validator might implement.

1. **General purpose:** The rules and constraints that always apply since they are inherent in the semantics of the model on which the schema is based – in the case of EIRE_NP the Transmodel conceptual model as interpreted for NeTEx.

   - Many rules are inherent in the meaning of the conceptual model. For example, that a ***ServicePattern*** must contain at least two instances of ***StopPointInServicePattern*** (i.e., has at least an origin and a destination). A complicating factor however is that for the purposes of data exchange, the applicability of a rule may depend on the use case; that is the rule may be true, but the actual level of detail of data required by an application may mean that detailed data elements do not need to be exchanged (and may not even be populated in a model). For example, in principle a ***StopPlace*** will always have at least one ***Quay***, but if details about the ***Quay*** are not relevant to an application, there may be no need to exchange them. Thus, the relevance of many rules may be specific to a profile (see 12.3below).

- The rules applied can include heuristics as to the plausibility of the real-world geometry or physics of the entities they represent. Such rules can be used to detect basic errors in the data. For example, that successive ***ScheduledStopPoint*** instances used in a ***ServicePattern*** are an unlikely distance apart for the mode of transport, or that their sequence of ***PassingTime*** instances would require an excessive vehicle speed**.**
- A validator may draw on further data sets to check the correctness of data. For example, data from a GIS system (such as Open Street Map) could be used to check that the spatial coordinates of bus stops are adjacent to a road; addresses can readily be verified against an address data base; or a dictionary can be used to check that text elements are correctly spelt according to their indicated language.
- A profile validator may apply rules across multiple documents (XML validators usually work on only a single document); so that external references may be resolved and checked.

2. **Parameterised:** NeTEx includes a few elements that can be used to describe the use of data in specific profiles. A general-purpose validator can be parameterised to read these elements and apply the additional rules and constraints that they describe. See 'Metadata for Profiles' below.
   - Such rules can be used in particular to check which elements should be present in the profile (described by a ***TypeOfFrame***) and which code values can be used (described by a ***TypeOfValue***).
3. **Profile specific:** A profile specific profile will encode additional rules and restrictions that are specific to the business requirements of the profile's intended use.
   - Rules can include additional semantic assumptions: For example, in EPIP, the restriction of a ***StopPlace*** hierarchy to two levels.
   - Rules can constrain the permitted nesting and organising of data elements when multiple alternatives are possible.
   - External references to data in reference data sets can be checked against a database known to the validator. For example, references to a stop could be checked in against NaPTAN, or to an operator against the National Operator Code database.
   - A validator can check whether the document is a self-contained issue (i.e., entirely new or a complete replacement for a previous version) or a delta (i.e., may comprise only a partial dataset and also include deletions).
   - Rules can also relate to the **completeness**, **relevance** and **plausibility** of data content of a profile. See examples of each category in 12.5. below.

4. **Optional rules for custom use of a profile:** When using a profile, there may be additional strictures (for example, that accessibility data be present, or translation into an additional language) that apply only when using it for a particular application.

   - Such rules might be implemented by a separate, stricter profile validator – but might typically be treated as optional rules for a more general validator that can be selected at runtime, E.g., “check for translations into Welsh”, “check that accessibility data is present”, etc.

Note: As is the case with XML validators, it is likely that different profile validator tools from different developers will make a slightly different interpretation of the profile rules, as well as being subject to different "bugs"; consequently, some datasets may validate successfully with some tools but not with others. *In order to maximise interoperability between systems, it is recommended that shared open source tools be used (when available) that can be run by both producers and consumers alike.*

12.4 Metadata for profiles

NeTEx contains several element types and attributes that are of use for describing some of the additional validation rules of a specific profile – as such, they really describe metadata (data about data) rather than “payload” data content. They can be used to define frame types and reserved code values that apply to all profile conformant documents. The metadata mechanisms are described individually elsewhere but are summarised below. A general profile validator can be parameterised to use these values to apply additional checking of the content of a profile conformant document.

- ***TypeOfFrame***: specifies the elements and attributes allowed in a frame. (See 8.4.2).
- ***TypeOfValue***: specifies the allowed values for open code lists (***TypeOfValue*** is abstract; it is specialised as concrete types for different entities, e.g., ***TypeOfLine***, ***TypeOfPlace***, etc). A ***TypeOfZone*** is used in the EPIP to specify the monomodal and general stop classifications, and in the EIRE_NP to retain legacy NaPTAN classifications.). See 8.5.1.
- ***ValueSet***: groups a set of instances of ***TypeOfValue***. See 8.5.2.
- ***VersionFrame / prerequisites:*** indicates which other frames are needed to form a coherent set of data. See 8.4.3. This is relevant in particular for indicating the internal dependencies within a NETWORK_OFFER frame which may contain many different component frames, and for summarising external frame use.
- ***KeyList:*** can indicate optional characteristics for which a validator may check. Use in the x to indicate if conformance is meant to be *strict* or otherwise. See 8.1.5.

12.4.1 Profile metadata frames for EIRE_NP

A set of frames with metadata values are provided that can be used by implementers. These comprise a ***CompositeFrame*** with a separate metadata frame for the EIRE_NP, NPTG, NaPTAN, and timetable mappings. The relevant frames may be included in an EIRE_NP compliant document and used for internal referential integrity checking; an XML validator will check automatically that only the correct values have been used.

Data set	Contents	Frame of metadata
EIRE_NP	CompositeFrame	eix:EI:NTA:TypeOfFrame_IE_PI_METADATA_OFFER:EIRE_NP
EIRE_NP	Frame Types	eix:EI:NTA:ResourceFrame_IE_PI_METADATA_FRAMES:EIRE_NP
NPTG	NPTG code values	eix:EI:NTA:ResourceFrame_IE_PI_METADATA:NPTG:nptg
NPTG	NPTG Country definitions	eix:EI:NTA:ResourceFrame_IE_PI_METADATA:EIRE:nptg
NaPTAN	NaPTAN code values	eix:EI:NTA:ResourceFrame_IE_PI_METADATA:NaPTAN:napt

12.5 Summary of data quality rules for EIRE_NP and EPIP

The following table provides an overview of relevant quality checking rules that may be applied to an EPIP or EIRE_NP conformant document, grouped approximately according to the criteria discussed above and with the appropriate validation mechanism indicated. The list is not exhaustive.

Table 156 — Data quality checking rules

Type	Checking rule category	Description
A	**XML syntax conformance** (XML validator)	The syntax of the XML in the document shall fully conform to the NeTEx XSD, as proven by successful validation with an XML validator. This will check: — Tag names, tag order, tag nesting. — Whether elements are required by the schema and their permitted multiplicity (cardinality). — The formats of data attributes: — The use of built-in simple data types (*date, time, integer, boolean, decimal, lang, URI, etc*). — The use of NeTEx defined simple data types: (*ObjectIdType, Email, Longitude, Latitude, LengthType, DistanceType,* etc.) — The use of enumerated vales.
B	**Referential integrity** (XML validator – internal, General Profile validator - external)	Uniqueness and referential integrity constraints should be upheld; as proven by successful validation with an XML validator. — All identifiers should be assigned to the correct ***Codespace*** and must not accidentally correspond to those of another data provider. — All identifiers shall be unique within the document within their codespace and uniqueness scope. — Internal references to data elements shall be present in the same XML document. — A ***version*** value should be specified for all internal references so that they are checked by an XML validator. External references must also be correct (checkable with a Profile validator)

		— External references should be valid references. i.e., present in another available dataset. — A ***versionRef*** value should be specified for all external references so that th are flagged as such. — Referenced elements should correspond to the expected entity type and instance of the reference. (For example, the ***id*** of a reference to a ***StopPlac*** shall not correspond to the ***id*** of a ***Line,*** nor to a **StopPlace** other than the one expected, etc.). Reference to external resources should be correct. — Any specified Urls should exist. — Referenced images and other resources should exist and be of the expecte MIME type. — Etc.
C	**General profile conformance** (General Profile validator)	The populating of data values in a profile conformant document shall fully conform to the choices of the profile, as specified any metadata specified as NeTEx elements: — Only the objects required for *strict* conformance with the profile are necessarily expected to be used when a document is imported. — Any additional data elements, present say for an **augmented** or **extended** conformance, must uphold the profile principles for use (see ‘extending the schema’ above). In particular they must not overlap with the content of an of the elements present in **strict** conformance. It should be possible to use the **strict** content without reference to any of the additions. — A ***Codespace*** shall be given for all identifiers. — Data elements of a given class must be placed only in the version frame typ or types specified by the profile. — The version frames for the strict and augmented content shall be classified with the correct ***TypeOfFrame*** instances specified by the profile. — Etc.
D	**Common NeTEx semantic rules conformance** (General Profile validator)	Rules relating to the use of NeTEx elements arising from the real-world semanti of the objects that they represent; that cannot be or are not expressed in the XML but which apply to any NeTEx schema. — The **Submode** must be consistent with the ***TransportMode*** (‘*schoolBus*’ is n a submode of ‘*train*’). — A ***ServicePattern*** must have a minimum of two ***ScheduledStopPoints***; — The type and mode of a ***ScheduledStopPoint*** must be consistent with that o any ***StopPlace*** to which it is assigned with a ***PassengerStopAssignment***. (a ***ScheduledStopPoint*** with ***StopType=*** ‘*onstreetBus*’ should not be assigned t a ***StopPlace StopType=*** ‘*trains*’ and should be used by a ***Line*** of ‘*bus*’ mode)

		— Every ***StopPointInJourneyPattern*** shall, for all associated Journeys, have one associated ***PassingTime***; every stop except the last shall have a **DepartureTime;** the first stop shall not have an **ArrivalTime**, and the last stop shall *only* have an **ArrivalTime.** — ***PassingTimes*** should be successive along a ***ServicePattern*** in proportion to the known distances and a plausible speed for the ***Mode*** and ***VehicleType***. — ***DayOffSet*** values in a sequence of times should not decrease. — For a timetable, there should be a service (i.e., ***ServiceJourney*** instances) available on every calendar day. Days without services (per ***Line*** or ***Network***) should be checked and raise some warning. — The members of a group should be of the correct type for the group and should be unique within the group. — A ***ServicePattern*** should be contained in a similar spatial corridor to its corresponding "inverse" ***ServicePattern***, and should be the approximate reverse of it (i.e., share a minimum number of references to the same ***ScheduledStopPoint*** instances, used in a reverse order). — Instances of ***JourneyPart*** shall be given in sequential order within the ***ServicePattern*** of a ***ServiceJourney***. — Etc.
E	**Specific profile conformance** (EPIP & EIRE_NP Profile validators)	The populating of data values in a profile conformant document shall fully uphold the specific semantics of the profile: — The additional profile limitations on the normal Transmodel/ NeTEx semantics shall be followed. For example, the restriction of a ***StopPlace*** hierarchy to two levels. — A ***StopPlace*** must be categorised with a ***TypeOfPlace*** as ‘*epip*:*general* ‘ or ‘*epip*:*monomodal*’. NB☞ For the EIRE_NP, it can be assumed that all STOP PLACEs are *‘epip:monomodal’ unless specified otherwise* — The rules for encoding profile identifiers shall be followed. — The specified profile rules for organising and nesting the objects in a frame shall be followed. (In particular where the profile requires the inline embedding of ancillary elements that the NeTEx schema also allows to be placed separately and cross referenced). — The ‘*epip_data’* ***Codespace*** (with prefix ‘epd’) is reserved for EPIP metadata. NB☞ The ‘eix_*data* ***Codespace*** (with prefix ‘eix’ is reserved for EIRE_NP metadata. — Etc.
F	**Data Completeness**	All data required to make the data consistent and usable as a coherent set shall be available. The notion of ‘completeness’ depends on the intended use case; for

	(EPIP & EIRE_NP Profile validators)	example, whether all stops or just the stops that are used by a line should be present; or whether only current data is needed or future data as well. Examples for the EIRE_NP and EPIP include: — An EI_PI_LINE_OFFER ***CompositeFrame*** shall contain all the information (frames and content objects) required to properly describe a ***Line***. — Definitions must be present for all the ***StopPlaces*** used by a ***Line*** through it ***Route*** and ***ServicePattern*** instances***.*** — A ***StopPlace*** should reference at least one ***TopographicPlace***. — A ***ScheduledStopPoint*** shall be assigned to a ***StopPlace*** with at least one ***StopPlaceAssignment.*** — A ***Line*** should have one or more ***Route*** instances and its routes shall have o or more ***ServicePattern*** instances, — A ***ServicePattern*** shall have at least two ***ScheduledStopPoint*** instances. — A ***ServiceJourney*** shall have a ***ServicePattern.*** — Every ***ServiceJourney*** shall have at least one ***DayType.*** — A ***DayType*** will have sufficient ***PropertiesOfDay*** to characterise it. — A ***ServiceJourney*** shall have a ***ServicePattern.*** — ***ServiceJourney*** instances that have a similar ***Direction*** and ***DayType*** and ***JourneyPattern*** shall be grouped with a ***GroupOfServices***. — Data shall be valid and current for the specified validity period. — Etc.
G	**Data Plausibility** (General Profile validator)	Values of parameters should be plausible against common sense measures. (Fo example, the ***StopPlace*** instances for Dublin are not located on the equator; a bus doesn't drive at 200 km/h, in an urban context there is not 750 km distance nor a 17-hour travel time between two successive stops of a bus route, etc.). Such measures can be applied to objects with dimensional attributes such as length, distance and time. — Dimensional values should be in the units specified for their data type. — The spatial separation of stops should be appropriate to the mode of transport. — The spatial coordinates of ***StopPlace*** and ***TariffZone*** instances should lie within the bounds of their ***TopographicPlace***. — The distance between instances of ***StopPlaceComponent*** (***Quay***, ***StopPlaceEntrance***, etc) within a single StopPlace should allow a possible walk from one to the other within a reasonable time. — Nested instances of ***StopPlace*** should be sufficiently close to each other to allow a possible walk from one to the other within a few minutes.

		— Transfer times at an interchange (as specified variously on ***Connection***, ***SiteConnection***, and ***DefaultConnection***) should be appropriate to the ***Mode*** and size of the interchange and any specific ***Connection*** distances. — Dates should lie within a few years of the present; durations should be plausible for their usage. — The public identifiers used to label platforms lines, stops etc should be unique within their expected scope of use. (For example, many towns and operators will have a *‘Line 1’,* but there should only be one ‘*Line* 1’ (at least per ***Operator***) in a given town).
H	**Data Relevance** (Profile validator)	Documents should not contain irrelevant or unused instances of data (this requires a notion of which elements in a frame are the primary payload for the profile). For example, for the EPIP and the EIRE_NP: — Every ***StopPlace*** present in a document, or one of its components should be used by at least one ***ScheduledStopPoint***. — Every ***TopographicPlace*** present in a document, or one of its components should be used by at least one ***StopPlace*** *or* ***TariffZone’*** — Every ***ServicePattern*** should be used by at least one ***ServiceJourney.*** — Empty tags should not be used for unpopulated attributes (see discussion of Empty values *4*). — Ancillary elements such as ***DayType***, ***Operator***, ***Authority*** should all be referenced at least once. — Etc.
I	**Data correctness** (Manual inspection against real-world evidence)	Data values should accurately describe the real-world entities they are meant to represent. — For an EI_PI_NETWORK_OFFER, all the lines of a ***Network*** shall be described. — Names should be correct and spelled right. — Labels and identifiers of stops and services should be correct. — Spatial coordinates for objects are correct. — All the instances of ***ServiceJourney*** for a ***Line*** shall be specified. — All the instances of a ***Line*** for a given ***Network*** shall be specified. — Passing times should be correct. — Days of operation should be correctly stated as the properties of a ***DayType.*** — Etc.
J	**Application specific constraints**	The business requirements of some applications may require specific data attributes to be populated that are not mandatory in the profile. For example, a system providing information for deaf or blind persons may require that the ***AudibleSignsAvailable*** and ***VisualSignsAvailable*** always present in the data.

	(Custom profile validator)	Thus, it may be necessary to specify further stricter constraints on certain elements and usages the are beyond the scope of the strict EPIP.

12.6 Referential integrity *(automatic)*

As explained in section 8.1.10, **internal** references within a document shall specify the version of the referenced object in the ***version*** attribute and will be check by the XML validator; **external** references document should specify the version of the referenced object in the ***versionRef*** attribute (and so will be ignored by the XML validator).

Examples of uses of the version with an actual version and with the dummy "*any*" version:

NeTEx XML EXAMPLE

Declaration of a route 'PHB@1':

```
<Route version="5.6" id="phb:PHB@1>
</Route>
```

Versioned Internal and external reference to the route:

```
<RouteRef version="5.6" ref="phb:PHB@1">INTERNAL</RouteRef>

<RouteRef versionRef="5.6" ref="phb:PHB@1">EXTERNAL</RouteRef>
```

Unversioned internal and external reference to the route:

```
<Route version="any" id="phb:PHB@1">
</Route>

<RouteRef version="any" ref=" phb:PHB@1">INTERNAL</RouteRef>
<RouteRef versionRef="any" ref=" phb:PHB@1">EXTERNAL</RouteRef>
```

Annex A
(Normative)

Data Dictionary

This data dictionary repeats the definitions from Transmodel (v6.0) and NeTEx (v1.1), but selects only the concepts included in the EIRE_NP. Concepts written in uppercase are from the conceptual model (i.e., Transmodel); those in camel-case are specific to the NeTEx physical model. See Transmodel and NeTEx for further information.

A.1 Terms and definitions

For the purposes of this document, the terms and definitions given in CEN/TS 16614-1:2019 apply.

ACCESS MODE A characterisation of the passenger movement according to the means of transport different from public transport (e.g., walk, bicycle, etc).

ACCESSIBILITY ASSESSMENT The accessibility characteristics of an entity used by passengers such as a STOP PLACE, or a STOP PLACE COMPONENT. Described by ACCESSIBILITY LIMITATIONs, and/or a set of SUITABILITies.

ACCESSIBILITY LIMITATION A categorisation of the accessibility characteristics of a SITE, e.g., a STOP PLACE or a STOP PLACE COMPONENT to indicate its usability by passengers with specific needs, for example, those needing wheelchair access, step-free access or wanting to avoid confined spaces such as lifts. A small number of well-defined categories are used that are chosen to allow the consistent capture of data and the efficient computation of routes for different classes of user.

ACCOMMODATION A combination of accommodation characteristics available on a service, e.g. "First Class Couchette with shower and 2 bunks".

ADDRESS The descriptive data associated with a PLACE that can be used to describe the unique geographical context of a PLACE for the purposes of identifying it. May be refined as either a ROAD ADDRESS, a POSTAL ADDRESS or both.

ADDRESSABLE PLACE A type of PLACE that may have an address associated with it.

ALLOWED LINE DIRECTION An allowed DIRECTION that can be used on a given ROUTE. This can be used to validate the selection of allowed values.

ALTERNATIVE NAME Another name for a named ENTITY.

ALTERNATIVE TEXT Alternative text for any textual attribute of an ENTITY.

ASSIGNMENT The association of one or more other assigned ENTITies with a primary ENTITY in order to describe some characteristic of the 'assigned to' ENTITY. The assignment may itself also have attributes.

There may be multiple assignments applying to the same entity in which case an order and or VALIDITY CONDITIONs may apply in order to determine the current assignment to use.

AUTHORITY The organisation under which the responsibility of organising the transport service in a certain area is placed.

BOOKING ARRANGEMENTS Information on how to book a public transport service.

CODESPACE A CODESPACE defines the context within which an identifier of an object in a document is unique (i.e., it provides a namespace for data).

A CODESPACE may have a specific code set of rules associated with it for encoding the identifiers that are unique within its scope.

An application processing a document is expected to understand any rules peculiar to interpreting a specific CODESPACE.

CONTACT DETAILS Information on how to contact an ORGANISATION or its representatives.

CONNECTION The physical (spatial) possibility for a passenger to change from one public transport vehicle to another to continue the trip, determined by two SCHEDULED STOP POINTs. Different times may be necessary to cover the link between these points, depending on the kind of passenger.

CONNECTION END One end of a CONNECTION.

COUNTRY A jurisdictional geographic boundary. A COUNTRY normally has a two characters IANA identifier.

COMPOSITEFRAME A VERSION FRAME that may hold other VERSION FRAMES.

DataManagedObject An ENTITY IN VERSION that can be associated with a RESPONSIBILITY SET that describes who has responsibility for managing its data.

DATA SOURCE The DATA SOURCE identifies the system which has produced the data. References to a data source are useful in an interoperated computer system.

DAY OF WEEK A particular week day (from Monday to Sunday).

DAY TYPE A type of day characterised by one or more properties which affect public transport operation. For example: weekdays in school holidays.

DAY TYPE ASSIGNMENT The assignment of operational characteristics, expressed by DAY TYPEs, to particular OPERATING DAYs within a SERVICE CALENDAR.

DEFAULT CONNECTION The possibility to change from one mode of transport to another, specified at an area or national level as specified by a TOPOGRAPHIC PLACE, STOP AREA or SITE ELEMENT. It may be restricted to a specific MODE or OPERATOR or only apply in a particular direction of transfer, e.g., bus to rail may have a different time for rail to bus. It specifies default transfer times to use if no specific times are specified for individual CONNECTIONs.

DESTINATION DISPLAY An advertised destination of a specific JOURNEY PATTERN, usually displayed on a head sign or at other on-board locations.

DESTINATION DISPLAY VARIANT A variant text of a DESTINATION DISPLAY for use in a specific media or context.

DIRECTION A classification for the general orientation of ROUTEs.

DYNAMIC STOP ASSIGNMENT The dynamic association of a SCHEDULED STOP POINT (i.e., a SCHEDULED STOP POINT of a SERVICE PATTERN or JOURNEY PATTERN) with the next available STOP PLACE, QUAY or BOARDING POSITION within a STOP PLACE.

ENTITY Any data instance to be managed in an operational Version Management System. When several data sources coexist (multimodality and/or interoperability), an ENTITY has to be related to a given DATA SOURCE in which it is defined.

ENTITY IN VERSION The ENTITY associated to a given VERSION. ENTITY IN VERSION is restricted by ENTITY IN FRAME.

FACILITY A named amenity available to the public at a SITE or on a SERVICE. A facility has no further properties other than a name. An EQUIPMENT or LOCAL SERVICE is used to describe the further properties provided as part of particular facility.

FACILITY SET A set of FACILITies available for a SERVICE JOURNEY or a JOURNEY PART. The set may be available only for a specific VEHICLE TYPE within the SERVICE (e.g., carriage equipped with low floor).

FLEXIBLE LINE Specialisation of LINE for flexible service. As all the service on a LINE may not all be flexible, flexibility itself is described at JOURNEY PATTERN level (meaning that a separate JOURNEY PATTERN is needed for each type of flexibility available for the line).

FLEXIBLE POINT PROPERTIES Set of characteristics describing the possible flexibility of POINTs. A composition is used with POINT in order to avoid multiple inheritance.

FLEXIBLE SERVICE Additional characteristics of flexible service. A service may be partly fixed, partly flexible.

GeneralFrame A specialisation of VERSION FRAME that can contain data for ENTITies of any sort to which the same VALIDITY CONDITION has been assigned.

GROUP OF ENTITies A set of ENTITies grouped together according to a PURPOSE OF GROUPING, e.g., grouping of stops known to the public by a common name.

GROUP OF LINEs A grouping of LINEs which will be commonly referenced for a specific purpose.

GROUP OF SERVICEs A group of SERVICEs, often known to its users by a name or a number

GroupOfServices-Member An individual member of a GROUP OF SERVICEs.

GroupOfStopPlaces A grouping of STOP PLACEs which will be commonly referenced for a specific purpose.

HEADWAY JOURNEY GROUP A group of VEHICLE JOURNEYs following the same JOURNEY PATTERN having the same HEADWAY INTERVAL between a specified start and end time (for example, every 10 min). This is especially useful for passenger information.

INTERCHANGE The scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs.

JOURNEY Common properties of VEHICLE JOURNEYs and SPECIAL SERVICEs, e.g., their link to accounting characteristics.

JOURNEY FREQUENCY GROUP A group of JOURNEYs defined in order to describe special behaviour like frequency-based services or rhythmical services (for example “runs all xxh10, xxh25 and xxh45, etc.”; this is especially useful for passenger information).

JOURNEY PART A part of a VEHICLE JOURNEY created according to a specific functional purpose, for instance in situations when vehicle coupling or separating occurs.

JOURNEY PART COUPLE Two JOURNEY PARTs of different VEHICLE JOURNEYs served simultaneously by a train set up by coupling their single vehicles.

JourneyPartPosition Position in TRAIN of a JOURNEY PART, starting from a given stop.

JOURNEY PATTERN An ordered list of SCHEDULED STOP POINTs and TIMING POINTs on a single ROUTE, describing the pattern of working for public transport vehicles. A JOURNEY PATTERN may pass through the same POINT more than once. The first point of a JOURNEY PATTERN is the origin. The last point is the destination.

KeyValue An extension mechanism, comprising pairs of named keys and their associated values, allowing additional information to be included in a NeTEx document,

LINE A group of ROUTEs which is generally known to the public by a similar name or number.

LINK An oriented spatial object of dimension 1 with view to the overall description of a network, describing a connection between two POINTs.

LINK SEQUENCE An ordered sequence either of POINTs or of LINKs, defining a path through the network.

LOCATION The position of a POINT with reference to a given LOCATING SYSTEM (e. g. coordinates).

LOCATING SYSTEM The system used as reference for location and graphical representation of the network and other spatial objects. (E.g., WGS85, Lambert, OS, etc).

Locale Parameters relating to a specific locale to be used when processing and presenting data.

MODE Any means of transport. (Sometimes given as TRANSPORT MODE)

NETWORK A named grouping of LINEs under which a transport network is known.

NETWORK VALIDITY PARAMETER A type of VALIDITY PARAMETER related to the network structure.

NOTICE A text for informational purposes on exceptions in a LINE, a JOURNEY PATTERN, etc. The information may be usable for passenger or driver information.

NOTICE ASSIGNMENT The assignment of a NOTICE showing an exception in a JOURNEY PATTERN, a COMMON SECTION, or a VEHICLE JOURNEY, possibly specifying at which POINT IN JOURNEY PATTERN the validity of the NOTICE starts and ends respectively. For NeTEx Part 3, NOTICE ASSIGNMENT is also available for DISTANCE MATRIX ELEMENT and SERIES PATTERN.

OPERATING DAY A day of public transport operation of which the characteristics are defined within in a specific SERVICE CALENDAR. An OPERATING DAY may last more than 24 hours.

OPERATING PERIOD A continuous interval of time between two OPERATING DAYs which will be used to define validities.

OPERATOR A company providing public transport services.

ORGANISATION A legally incorporated body associated with any aspect of the transport system.

PASSENGER STOP ASSIGNMENT The allocation of a SCHEDULED STOP POINT (i.e., a STOP POINT of a SERVICE PATTERN or JOURNEY PATTERN) to a specific STOP PLACE for a PASSENGER SERVICE, and also possibly a QUAY and BOARDING POSITION.

PASSING TIME Time data concerning public transport vehicles passing a particular POINT; e.g., arrival time, departure time, waiting time.

PLACE A geographic place of any type which may be specified as the origin or destination of a trip. A PLACE may be of dimension 0 (a POINT), 1 (a road section) or 2 (a ZONE).

POINT A 0-dimensional node of the network used for the spatial description of the network. POINTs may be located by a LOCATION in a given LOCATING SYSTEM.

POINT IN JOURNEY PATTERN A STOP POINT or TIMING POINT in a JOURNEY PATTERN with its order in that JOURNEY PATTERN.

POINT IN LINK SEQUENCE A POINT in a LINK SEQUENCE indicating its order in that particular LINK SEQUENCE.

POINT OF INTEREST A type of PLACE to or through which passengers may wish to navigate as part of their journey and which is modelled in detail by journey planners.

POINT ON ROUTE A ROUTE POINT used to define a ROUTE with its order on that ROUTE.

POSTAL ADDRESS A specification of ADDRESS refining it by using the attributes used for conventional identification for mail. Comprises variously a building Identifier, Street name, Post code and other descriptors.

Presentation A set of parameters describing the visual appearance to be used for an ENTITY in displays and other media.

PublicationDelivery A NeTEx XML schema element that contains all the data being exchanged in a NeTEx conformant XML document.

PROPERTY OF DAY A property which a day may possess, such as school holiday, weekday, summer, winter etc.

QUAY A place such as platform, stance, or quayside where passengers have access to PT vehicles, Taxi, cars or other means of transportation. A QUAY may serve one or more VEHICLE STOPPING PLACEs and be associated with one or more STOP POINTS.
A QUAY may contain other sub QUAYs. A child QUAY must be physically contained within its parent QUAY.

RESPONSIBILITY SET A list of possible responsibilities over a one or more ENTITies IN VERSION., resulting from the process of the assignment of RESPONSIBILITY ROLEs (such as data origination, ownership, etc.) on specific data (instances) to ORGANISATIONs or ORGANISATION PARTs.

ROAD ADDRESS A specialisation of ADDRESS refining it by using the characteristics such as road number, and name used for conventional identification of a location along a road.

ROUTE An ordered list of located POINTs defining one single path through the road (or rail) network. A ROUTE may pass through the same POINT more than once.

ROUTE LINK An oriented link between two ROUTE POINTs allowing the definition of a unique path through the network.

ROUTE POINT A POINT used to define the shape of a ROUTE through the network.

SCHEDULED STOP POINT A POINT where passengers can board or alight from vehicles.

SERVICE CALENDAR A collection of DAY TYPE ASSIGNMENTs.

SERVICE FACILITY SET Set of FACILITies available for a specific VEHICLE TYPE (e.g., carriage equipped with low floor) possibly only for a service (or for a SERVICE JOURNEY or a JOURNEY).

SERVICE JOURNEY A passenger carrying VEHICLE JOURNEY for one specified DAY TYPE. The pattern of working is in principle defined by a SERVICE JOURNEY PATTERN.

SERVICE JOURNEY INTERCHANGE The scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs.

SERVICE LINK A LINK between an ordered pair of SCHEDULED STOP POINTs.

SERVICE PATTERN The subset of a JOURNEY PATTERN made up only of STOP POINTs IN JOURNEY PATTERN.

SITE A type of PLACE, such as a STOP PLACE, POINT OF INTEREST or ADDRESS, to which passengers may wish to travel. A SITE can have designated ENTRANCEs that represent the available points of access for different USER NEEDs.

SITE COMPONENT An element of a SITE describing a part of its structure. SITE COMPONENTs share common properties for data management, accessibility and other features.

SITE CONNECTION The physical (spatial) possibility to connect from one point to another in a SITE.

SITE CONNECTION END One end of a SITE CONNECTION.

SITE ELEMENT A type of PLACE specifying common properties of a SITE or a SITE COMPONENT to describe it., including accessibility.

SITE FACILITY SET Set of FACILITies available for a SITE ELEMENTS

STOP ASSIGNMENT The allocation of a SCHEDULED STOP POINT (i.e., a SCHEDULED STOP POINT of a SERVICE PATTERN or JOURNEY PATTERN) to a specific STOP PLACE, for either a Passenger JOURNEY or VEHICLE SERVICE.

STOP PLACE A place comprising one or more locations where vehicles may stop and where passengers may board or leave vehicles or prepare their trip. A STOP PLACE will usually have one or more well-known names.

STOP PLACE COMPONENT An element of a STOP PLACE describing part of its structure. STOP PLACE COMPONENTs share common properties for data management, accessibility and other features.

STOP PLACE ENTRANCE A physical entrance or exit to/from a STOP PLACE for a Passenger. May be a door, barrier, gate or other recognisable point of access.

STOP PLACE SPACE A physical area within a STOP PLACE, for example, a QUAY, BOARDING POSITION, ACCESS SPACE or EQUIPMENT PLACE.

STOP POINT IN JOURNEY PATTERN A POINT in a JOURNEY PATTERN which is a SCHEDULED STOP POINT.

SUBMODE A variant of a MODE, as for instance international or domestic rail (rail being the MODE).

TARIFF ZONE A ZONE used to define a zonal fare structure in a zone-counting or zone-matrix system.

TEMPLATE SERVICE JOURNEY A passenger carrying TEMPLATE VEHICLE JOURNEY. It may represent multiple journeys.

TEMPLATE VEHICLE JOURNEY A repeating VEHICLE JOURNEY for which a frequency has been specified, either as a HEADWAY JOURNEY GROUP (e.g., *every 20 minutes*) or a RHYTHMICAL JOURNEY GROUP (e.g., *at 15, 27 and 40 minutes past the hour*). It may thus represent multiple journeys.

TIME BAND A period in a day, significant for some aspect of public transport, e.g., similar traffic conditions or fare category.

TIMETABLED PASSING TIME Long-term planned time data concerning public transport vehicles passing a particular POINT IN JOURNEY PATTERN on a specified VEHICLE JOURNEY for a certain DAY TYPE.

TIMING POINT A POINT against which the timing information necessary to build schedules may be recorded.

TOPOGRAPHIC PLACE A geographical settlement which provides topographical context when searching for or presenting travel information, for example as the origin or destination of a trip. It may be of any size (e.g., County, City, Town, Village) and of different specificity e.g., Greater London, London, West End, Westminster, St James s.

A TOPOGRAPHICAL PLACE must always have a canonical gazetteer name. It may be necessary to use the hierarchical topographical relationships of the TOPOGRAPHICAL PLACE to establish a unique context with which to distinguish between two TOPOGRAPHICAL PLACEs with the same name.

TRAIN A VEHICLE TYPE composed of TRAIN ELEMENTs in a certain order, i.e., of wagons assembled together and propelled by a locomotive or one of the wagons.

TRAIN COMPONENT A specification of the order of TRAIN ELEMENTs in a TRAIN.

TRAIN ELEMENT An elementary component of a TRAIN (e.g., wagon, locomotive).

TRAIN NUMBER Specification of codes assigned to particular VEHICLE JOURNEYs when operated by TRAINs or COMPOUND TRAINs according to a functional purpose (passenger information, operation follow-up, etc.)

TRAIN STOP ASSIGNMENT The association of a TRAIN COMPONENT at a SCHEDULED STOP POINT with a specific STOP PLACE and also possibly a QUAY and BOARDING POSITION.

TRANSFER A couple of POINTs located sufficiently near that it may represent for a passenger a possibility to reach one of these POINTs when starting at the other one in a timescale which is realistic when carrying out a trip, e.g., ACCESS.

TransferDuration A set of times for TRANSFER between two Points.

TYPE OF ENTITY A classification of an ENTITies, for instance according to the domain in which they are defined or used.

TYPE OF FRAME A classification of a VERSION FRAME according to a common purpose. E.g., line descriptions, vehicle schedules, operating costs. A TYPE OF FRAME is ruled by a unique TYPE OF VALIDITY.

TypeOfValue A code value from an extensible set (i.e., which may be added to by user applications) that may be used to classify and validate the properties of ENTITies.

VALIDITY CONDITION Condition used in order to characterise a given VERSION of a VERSION FRAME. A VALIDITY CONDITION consists of a parameter (e.g., date, triggering event, etc.) and its type of application (e.g., for, from, until, etc.).

ValidBetween A simple VALIDITY CONDITION specifying a period between a start and an end date.

ValueSet A collection of ***TypeOfValue*** instances of the same type and relating to a single classification scheme.

VEHICLE JOURNEY The planned movement of a public transport vehicle on a DAY TYPE from the start point to the end point of a JOURNEY PATTERN on a specified ROUTE.

VEHICLE MODE A characterisation of the public transport operation according to the means of transport (e.g., bus, tram, metro, train, ferry).

VEHICLE TYPE A classification of public transport vehicles according to the vehicle scheduling requirements for mode and capacity (e.g., standard bus, double-deck, etc).

VERSION A group of operational data instances which share the same VALIDITY CONDITIONs. A version belongs to a unique VERSION FRAME and is characterised by a unique TYPE OF VERSION. E.g., NETWORK VERSION for Line 12 starting from 2000-01-01.

VERSION FRAME A set of VERSIONs referring to a same DATA SOURCE and belonging to the same TYPE OF FRAME. A FRAME may be restricted by VALIDITY CONDITIONs.

VersionFrameDefaults A set of values attached to a VERSION FRAME to be assumed as the default values to use for the objects in the frame (such as CURRENCY, CODESPACE, etc) that do not have their own specific value.

VIA A secondary heading relevant for a certain part of the JOURNEY PATTERN advertising an onward intermediate destination to supplement the advertised (final) destination of DESTINATION DISPLAY.

ZONE A two-dimensional PLACE within the service area of a public transport operator (administrative zone, TARIFF ZONE, ACCESS ZONE, etc.).

Annex B
(Normative)
Facility code list

COMMON FACILITIES

B.1 FacilityList

unknown

lift

wheelchairLift

escalator

travelator

ramp

automaticRamp

steps

stairs

slidingStep

shuttle

narrowEntrance

barrier

lowFloorAccess

validator

B.2 AccessibilityInfoFacilityList

audioInformation

audioForHearingImpaired

visualDisplays

displaysForVisuallyImpaired

largePrintTimetables

other

B.3 AccessibilityToolList

wheelchair

walkingstick

audioNavigator

visualNavigator

passengerCart

pushchair

umbrella

buggy

other

B.4 AccommodationAccess List

other

freeSeating

reservation

standing

B.5 AccommodationFacility List

unknown

seating

sleeper

singleSleeper

doubleSleeper

specialSleeper

couchette

singleCouchette

doubleCouchette

specialSeating

recliningSeats

babyCompartment

familyCarriage

recreationArea

panoramaCoach

pullmanCoach

standing

B.6 AssistanceFacilityList

personalAssistance

boardingAssistance

wheechairAssistance

unaccompaniedMinorAssistance

wheelchairUse

conductor

information

other

none

any

B.7 BoardingPermisssion

normal

earlyBoardingPossibleBeforeDeparture

lateAlightingPossibleAfterArrival

overnightStayOnboardAllowed

B.8 BookingProcessFacilityList

productNotAvailable

productNotBookable

bookableThroughInternationalSystem

bookableThroughNationalSystem

bookableManuallly

other

B.9 CarServiceFacilityList

unknown

carWash

valetPark

carValetClean

oilChange

engineWarming

petrol

B.10 CateringFacilityList

bar

bistro

buffet

noFoodAvailable

noBeveragesAvailable

restaurant

firstClassRestaurant

trolley

coffeeShop

hotFoodService

selfService

snacks

foodVendingMachine

beverageVendingMachine

miniBar

breakfastInCar

mealAtSeat

other

unknown

B.11 CouchetteFacilityList

unknown

T2

T3

C1

C2

C4

C5

C6

wheelchair

other

B.12 FamilyFacilityList

none

servicesForChildren

servicesForArmyFamilies

nurseryService

B.13 FareClasses

unknown

firstClass

secondClass

thirdClass

preferente

premiumClass

businessClass

standardClass

turista

economyClass

any

B.14 GenderLimitation

both

femaleOnly

maleOnly

sameSexOnly

B.15 GroupBookingFacility

groupsAllowed

groupsNotAllowed

groupsAllowedWithReservation

groupBookingsRestricted

unknown

B.16 LuggageCarriageFacilityList

unknown

noBaggageStorage

baggageStorage

luggageRacks

extraLargeLuggageRacks

baggageVan

noCycles

cyclesAllowed

cyclesAllowedInVan

cyclesAllowedInCarriage

cyclesAllowedWithReservation

vehicleTransport

B.17 MealFacilityList

breakfast

lunch

dinner

snack

drinks

B.18 MedicalFacilityList

unknown

defibrillator

alcoholTest

B.19 MobilityFacilityList

unknown

lowFloor

stepFreeAccess

suitableForWheelchairs

suitableForHeaviliyDisabled

boardingAssistance

onboardAssistance

unaccompaniedMinorAssistance

tactilePatformEdges

tactileGuidingStrips

B.20 NuisanceFacilityList

unknown

smoking

noSmoking

familyArea

childfreeArea

noAnimals

breastfeedingFriendly

mobilePhoneUseZone

mobilePhoneFreeZone

B.21 PassengerCommsFacilityList

unknown

freeWifi

publicWifi

powerSupplySockets

telephone

audioEntertainment

videoEntertainment

businessServices

internet

postOffice

postBox

B.22 PassengerInformationEquipmentList

timetablePoster

fareInformation

lineNetworkPlan

lineTimetable

stopTimetable

journeyPlanning

interactiveKiosk

informationDesk

realTimeDepartures

other

B.23 PassengerInformationFacilityList

nextStopIndicator

stopAnnouncements

passengerInformationDisplay

realTimeConnections

other

B.24 RetailFacilityList

unknown

food

newspaperTobacco

recreationTravel

hygieneHealthBeauty

fashionAccessories

bankFinanceInsurance

cashMachine

currencyExchange

tourismService

photoBooth

B.25 SafetyFacilityList

ccTv

mobileCoverage

sosPoints

staffed

B.26 SanitaryFacilityList

none

toilet

wheelChairAccessToilet

shower

washingAndChangeFacilities

babyChange

wheelchairBabyChange

shoeShiner

other

B.27 ServiceReservationFacilityList

reservationsCompulsory

reservationsCompulsoryForGroups

reservationsCompulsoryForFirstClass

reservationsCompulsoryFromOriginStation

reservationsRecommended

reservationsPossible

reservationsPossibleOnlyInFirstClass

reservationsPossibleOnlyInSecondClass

reservationsPossibleForCertainClasses

groupBookingRestricted

noGroupsAllowed

noReservationsPossible

wheelchairOnlyReservations

bicycleReservationsCompulsory

reservationsSupplementCharged

unknown

B.28 TicketingFacilityList

unknown

ticketMachines

ticketOffice

ticketOnDemandMachines

mobileTicketing

B.29 TicketingServiceFacilityList

purchase

collection

cardTopUp

reservations

exchange

refund

renewal

excessFares

other

all

B.30 UicProductCharacteristicList

tariffCommunVoyageurs

allInclusivePrice

eastWestTariff

trainWithTcvAndMarketPrice

noPublishedTariff

B.31 UicTrainRate

normal

discountInTrainOtherThanTGV

specialFare

supplement

noPublishedTariff

B.32 VehicleFacilityList

Unknown

wheelchairLift

manualRamp

automaticRamp

steps

slidingStep

narrowEntrance

validator

<u>SERVICE FACILITIES</u>

Annex C Populating NaPTAN data

The NaPTAN specification includes a set of recommendations for how NPTG and NaPTAN data should be populated; covering such aspects as naming stops and locations and allocating codes. These guidelines remain relevant when populating data elements in a NeTEx format.

The following is a simplified extract of the NaPTAN guidelines taken from the DfT NaPTAN v2.5 guide, annotated with NeTEx equivalence].

C.1 Populating the NaPTAN Data

The names of stops and places in *NaPTAN* model, as for the National Gazetteer, should be carefully chosen to correspond to the way real-world places are perceived by users, and so that the relationships between them are useful for the intended computational purposes.

Consideration should be given to how locality name and stop name complement each other, as they may often be used in combination. In particular, when applications such as journey planners present lists of stop names for users to choose from, the locality name may be combined with the stop name to give an appropriate context within which to recognise the stop, for example, to distinguish '*Cosham, High Street'* from '*Farnham, High Street',* but omit it of the context is clear, e.g., in the context of Cosham, just *'High Street'*.

C.2 Choosing NaPTAN Points

On-street Points

On-street points of access to public transport are represented as points [i.e., NeTEx QUAYs]:

- For individual on-street **Bus Stops (**also **Coach Stops),** there should be a *NaPTAN* ***Bus*** stop point *for every physical stop;* even if a stop is the unmarked pair to another stop, it should always have its own separate *NaPTAN* identifier and definition (of type '*BCT'* located at its physical position. [i.e., NeTEx QUAY]
    - ***StopArea*** elements are used to group individual poles into larger groupings such as pairings (of type '*GBPS')* and on-street clusters (of type '*GCLS'*) (see below). [i.e., a NeTEx STOP PLACE should be created, or added if there is no existing NaPTAN ***StopArea***]
- For **Coach Stops***,* a ***StopArea*** of type '*GCCH'* can be used to associate the stop with Coach Service coverage. '*GCCH'* stop areas have a stop area code (*900G*) and are allocated centrally. [In NeTEx this is an additional STOP PLACE ASSIGNMENT].
- For **Taxi Ranks**, there should be a NaPTAN stop point for the head of the taxi rank, of type ***Taxi*** ('*TXR'*), or ***SharedTaxi*** ('*STR')* if an official taxi sharing scheme operates from the rank. [a NeTEx STOP PLACE should also be created].

Off-street Points

For stations, termini and other interchange facilities, there should be an individual *NaPTAN* stop point for each "entrance" from the public thoroughfare to the facility [i.e., NeTEx STOP PLACE ENTRANCE], and another ***AccessArea*** stop point instance for the "access side" [I.e., NeTEx STOP PLACE]: All stops should have the same ***CommonName***, with a different ***Indicator*** value [NeTEx ***NameSuffix*** attribute] to distinguish them if necessary.

- For **Airports**: For each terminal, there should be a *NaPTAN* ***Entrance*** point for each main area of check-in desks (of type '*AIR*'), [i.e., NeTEx STOP PLACE ENTRANCE] and another single ***AccessArea*** point to represent the “air-side” (of type ‘*GAT’*) [i.e., NeTEx STOP PLACE SPACE]. Entrance records are provided by the relevant Local Administrative Area.

  - A ***StopArea*** element (of type *‘GAIR’*) should be used to group the air entrances, access area, and any other connecting stop points such as taxi ranks and individual bus stops around the terminal. [A parent STOP PLACE should be created. A NeTEx STOP AREA may also be created]
  - The Access Area (‘*GAT*) points will be provided centrally (they will have identifiers beginning with *920*) and do not need to be provided by other administrative areas.

- For **Ferry Terminals and Ports:** There should be a *NaPTAN* ***Entrance*** point for the main entrance gate to the docks or ferry terminal (all of type *‘FTD’*) [i.e., a NeTEx STOP PLACE ENTRANCE], and each secondary entrance (also of type *‘FTD’*), and another single ***AccessArea*** (of type ‘*FER*’) point [i.e., STOP PLACE SPACE] to represent the general area berths from which the ferries depart. Entrance records are provided by the relevant Local Administrative Area.
  - A ***StopArea*** element (of type *‘GFTD*) [i.e., NeTEx STOP PLACE] should be used to group the ferry entrances, access area, and also any other connecting stop points such as taxi ranks and individual bus stops.
- For **Rail Stations**: There should be a *NaPTAN* ***Entrance*** stop point for the main entrance to the station (of type *‘RSE’*) [i.e., NeTEx STOP PLACE ENTRANCE], an additional stop point for each secondary entrance (also of type *‘RSE’*), and another to represent the “track side”, that is the main area of public access to the platforms (of type *‘RLY’*) [i.e., NeTEx QUAY]. Entrance records are provided by the Local Administrative Area.
  - The main entrance should be the primary *NaPTAN* stop point, i.e., be encoded with a 0 as the last digit (Secondary entrances have non-zero digits). All entrances should indicate their nature in the indicator text e.g., ‘main entrance’, ‘side entrance’.
  - The ***AccessArea*** (‘*RLY’*) and ***RailPlatform*** (‘*RPL’*) points will be provided centrally (they will have identifiers beginning with *‘910’*) and do not need to be provided by other administrative areas.
  - A ***StopArea*** element (of type *‘GRLS’*), provided centrally, should be used to group the rail entrances, access area, and any other connecting stop points such as taxi ranks and individual bus stops.
  - Note that in the Irish NaPTAN data set, only a StopArea is provided for a station.
- For **Bus and Coach Stations**: There should be a *NaPTAN* ***Entrance*** point for the main entrance (of type ‘*BCE’*), and each secondary entrance gate (also of type ‘*BCE’*). There may be a single ***AccessArea*** point (of type *‘BCQ’*) to represent the general bays from which the buses depart. There may additionally or instead also be one or more specific ***Bay stop*** points of (of type BCS) if individual poles are identified. All records for Bus and Coach Stations are provided by the Local Administrative Area.
  - A ***StopArea*** element (of type *‘GBCS’*) should be used to group the station entrances, access area and any other connecting stop points such as taxi ranks and individual bus stops.
- For **Metro & Underground Stations**: There should be a *NaPTAN* ***Entrance*** point for the main entrance to the station (of type *‘TMU’*) [i.e., NeTEx STOP PLACE ENTRANCE], and each secondary entrance (also of type *‘TMU’*), and another single ***AccessArea*** point to represent the “rail side”, that is the main area of public access to the platforms (of type ‘*MET*’) [i.e., NeTEx QUAY]. Entrance records are provided by the Local Administrative Area.
  - A ***StopArea*** element (of type *‘GTMU’*) [STOP PLACE]should be used to group the station entrances, access area, and any other connecting stop points such as taxi ranks and individual bus stops.

    - The ***AccessArea*** (‘*MET’*) and Metro ***Platform*** (‘*PLT’*) points will be gathered locally, but compiled and entered centrally.
- For **Tram Stops** Tram stops are treated as stations. There should be a *NaPTAN* ***PLT*** stop point *for every physical platform,* located at its physical position. And a ***PLT*** stop to represent the pair.
- For ***Telecabine*** (Lift & Cable Car Stations**)**: There should be a *NaPTAN* ***Entrance*** point for the main entrance to the station (of type ‘LCE*’*), and each secondary entrance (also of type ‘*LCE’*), and another single ***AccessArea*** point to represent the “lift side”, that is the main area of public access to the platforms (of type ‘*LCB’*). Entrance records are provided by the Local Administrative Area.
    - A ***StopArea*** element (of type ‘*GLCB’*) should be used to group the lift station entrances, access area, and any other connecting stop points such as taxi ranks and individual bus stops.
    - The ***AccessArea*** (‘*LCB’*) and Metro ***Platform*** (‘*LPL’*) points will be gathered locally, but compiled and entered centrally.

The NaPTAN ‘Transport side’ stops (‘*GAT’, ‘FER’, ‘RLY’*, ‘*MET’, ‘LCB’* areas*,* and *‘FBT’, ’RPL’* and *‘PLT’, ‘LPL’* access points) represent the boarding points to transport vehicles within the station or interchange building. At present FTD can also be used in the absence of FBT elements at Ferry Terminals.

C.3 Allocating an Identifier (“AtcoCode”) for a NaPTAN Stop Point

The NaPTAN stop identifier (AtcoCode***)*** is intended to be unique for a given stop point within Ireland. The prefix part is reserved to specific ranges so as to manage the distributed concurrent allocation of unique codes by different stakeholders. The ***AtcoCode*** has a general form of:

[**Database-prefix** {3}]] + [**Flag** [{1}] + [**Local- identifier** {up to 8 alphameric characters}]

Where:

1. The [**Database prefix**] is the three-character ***AtcoAreaCode*** of the ***AdministrativeArea*** responsible for managing the stop (which includes special values for rail stations, coach locations, ferry ports and airports).
2. The [**Flag**] normally has a value of ‘0’. Historically ‘1’ was used to encode stops belonging to another administrative area – this is not now needed so its use within NaPTAN constitutes an error.
3. **[Local-identifier**] is an identifier of the stop, unique within the scope of the ***AtcoAreaCode.***
    - **Rail Station Entrances.** The designated form is ‘AAA*0XXXXXXXn’* where AAA comprises the ***AtcoAreaCode***, ‘0’ is a fixed flag, *XXXXXXX* is the Network Rail TIPLOC code (generally alphabetic, capitalised, up to seven characters) for the station, and *n* is a zero character for the main entrance, and a sequence number for the other entrances. For example, *‘4000FARNHAM0’, ‘4000FARNHAM1”.*
    - **Coach Station Entrances.** The preferred form of number for Coach station entrances is *AAA0YYYYYn* where is the ***AtcoAreaCode*** of the ***AdministrativeArea*** responsible for managing the stop, ‘0’ is a fixed flag, *YYYYY* is the National Coach code (5 digit numeric) for the coach station, and *n* is a zero character for the main entrance and a sequence number for the other entrances.
    - **Transport side Access Areas.** The stop point codes of the ‘Transport side’ stops (GAT, FER, MET, RLY, and FBT, RPL, PLT) are assigned centrally from special national prefixes ranges beginning with ‘9’. The numbers of all other points begin with a local area prefix. For example, ‘*4000FARNHAM0’.*
    - **On-Street Stops.** The preferred form of numbers for on-street stops is *AAA0YYYYYYYY* where AAA is the ***AtcoAreaCode*** of the ***AdministrativeArea*** responsible for managing the stop.’0’ is a fixed flag. YYYYYYYY is a unique locally-allocated code of up to 8 alpha-numeric characters

C.4 Allocating NaPTAN (SMS) Codes for NaPTAN Stop Points

NaPTAN allows a short code to be specified for each stop, the ***NaptanCode***. This is intended as a unique reference for use in public facing systems such as SMS and web query apps.

The *NaPTAN* short code is distinct from the ATCO code (the latter is in effect a system identifier). A ***NaptanCode*** can only be used for a single stop

C.5 Mandatory NaPTAN Code features

In order to achieve nationwide uniqueness, a ***NaptanCode*** has a systematic structure.

[area-prefix] : [area -identifier]

Where:

- **[area-prefix]** Each prefix is unique within Eire (and also the UK) and assigned to a specific NPTG Admin area.
- The prefixes are normally three characters, but London is treated as a special case and uses a single digit - '1'.
- **[area-identifier]** a code that is unique for each stop within the area.; usually four characters

C.6 Choosing NaPTAN Stop Areas

The choice and naming of *NaPTAN* stop areas is closely related to the choice of stop points, and the names of related *NaPTAN* stop points and stop areas generally should be the same.

StopArea instances *should only be used to group stops that constitute a localised interchange in easy walking distance*, such as a bus bay, or a pair of opposite bus stops, or the various access points around a rail station. Stop areas must **not** be used to group stops in a wide area arbitrarily. For example, a stop area must **not** be used to create a general stop grouping for all the stops of a town centre; instead, a *NPTG* locality for the town centre should be used, and one or more of the stop groups and or stop points be associated with the *NPTG* locality.

As a general rule, a ***StopArea*** should not group stop points that are more than 250m apart.

Stop areas may be nested in hierarchies to build up a simple interchange description. Stop area names should correspond to the main stop points. For example, the '*Farnham Rail Station'* stop area might include subsidiary bus and stop areas, each containing various stop pairs near the station.

In principle there should be a stop area:

- For every pair of on-street bus poles (*GPBS*).
- For every cluster of on-street bus poles (*GCLS*).
- For every airport (*GAIR*).
- For every ferry terminal or port (GFTD).
- For every rail station (*GRLS*).
- For every bus or coach station (*GBCS*).
- For every metro station (*GTMU*).
- For every coach service association (*GCCH*).
- For every lift or cable car station service (*GLCB*).

The ***StopArea*** for the main travel mode can be used as a parent for the stop areas of subsidiary modes, for example the airport mode can contain a stop area for a rail station that serves the airport.

For a complex interchange, stop areas should be organised into a hierarchy. For example, an airport might contain child stop areas for its Rail and Metro stations, and several for its bus stops. When assembling ***StopArea*** instances into a hierarchy, the parent area should be chosen using the relative rankings shown in *Table 1*.

Table 1 – Precedence of StopArea Types

	Code	Type	Ranking
Off Street	GAIR	Airport	1
	GFTD	Ferry / Port	2
	GRLS	Rail Station	3
	GTMU	Metro Station	4
	GBCS	Bus or Coach Station	5
	GLCB	Lift or Cable Car Station	6
	GCCH	Coach Stop	7
On Street	GCLS	On-street Bus / Coach stop cluster (more than two stops in the same general location).	8
	GPBS	On-street Bus/ Coach stop pair	9

➔Note that in many cases, additional ***StopArea*** instances may be inferred by automated processes that augment the manually created *NaPTAN* stop data, for example grouping stop points by (i) by spatial proximity of location, and/or or (ii) semantic similarity of stop point, street name or other descriptor, together with (iii) transport mode. In practice these derived groupings may either be instantiated as actual ***StopArea*** instances in a database used by the journey planner, or be dynamically recomputed every time a search is made.

For some interchanges, notably rail stations, there may be multiple stop areas describing different parts of the same station (or two different encodings of the same station for historic reasons). If this is the case, they should be organised hierarchically with one of them chosen as the “main” root station and others as subsidiary (i.e., ***not*** using circular references with each one being part of the other).

C.7 The Naming of Stop Points and Stop Areas

The allocation of effective names to public transport access points is an important aspect of *NaPTAN’s* purpose.

Whilst rail stations and airports generally have well-known names, some types of PTAN, in particular bus stops, do not always have obvious or intuitive names. The *NaPTAN* ***StopPoint*** element provides a number of ‘descriptor’ subelements for specifying text descriptions of stops, and *NaPTAN* sets guidelines for populating the elements in a consistent way that will result in useful name phrases in applications, i.e., that enable the use of text-based searches to find the stop. See also the examples given later in Part III.

StopPoint descriptors may include:

- A ***CommonName***. The simple name for the stop. ‘Simple’ means that qualifiers such as the locality or street name should not be included as a *component* part of the ***CommonName*** - See ‘Descriptor Atomicity’ below and further comments below. A street name by itself may however be used as the complete simple ***CommonName*** of the stop, if that is the most appropriate concept (see "Street Style" later below). Thus, for example, a ***CommonName*** of "*Opp St Mary's Upper Street Islington*" is non-conformant because it repeats data that is already contained by the other atomic descriptor elements.

  - Assuming a ***Landmark*** style of naming - i.e. that *"St Mary's"* is the best simple name by which users can recognise the stop, a more conformant representation would be: ***CommonName***: *"St Mary's"; **Landmark**: "St Mary's"; **Indicator:** "Opp"; **Street** : "Upper Street"; **NptgLocality**: "Islington"* – which contains all the information necessary to create a label of *"Opp St Mary's, Upper Street, Islington"* if needed, but also allows other presentation forms.
  - A ***ShortCommonName***: Provides a version of the name to use restricted format media. By default, the regular ***CommonName*** will be used. <u>Note that for the Irish NaPTAN data set, it is the ***ShortCommonName*** that is always populated, and the ***CommonName*** is defaulted from it.</u>

- Assuming a ***Street*** style of naming - i.e., that "'Upper Street" is the best simple name, a more conformant representation would be: ***CommonName***: *"Upper Street"; **Landmark**: "St Mary's"; **Indicator:** "Opp 27"; **Street**: "Upper Street"; **NptgLocality**: "Islington".* The nearest ***Landmark*** should be shown in the data; for example, *'Red Lion Public House'.* The nearest cross-street (***Crossing***) may also be used as the ***CommonName***, for example: *'Folly Lane'.*
- An ***Indicator*** phrase, giving the relationship of the stop to the entity used as the common name, for example 'o/s' i.e., outside, 'behind', etc. The ***Landmark***, ***Street*** or ***CommonName*** should <u>**not**</u> be repeated in the ***Indicator,*** as this breaks the principle of descriptor 'atomicity' (see below). Thus, if the ***CommonName*** is *'Red Lion'*, the ***Indicator*** should say just *''o/s'*, and <u>not</u> *'Red Lion (o/s)' or 'o/s 'Red Lion'*. Stop numbers, Bay Numbers, etc are also relevant values for the ***Indicator***.
- The name of the ***Street*** on which the stop point lies. The street should always be specified as it provides an alternative search value for finding the stop, and also can provide additional context with which to recognize the stop in stop finders.
- Where both a Point of Interest ***Landmark*** and a ***Crossing*** are useful for identifying the stop, the nearest intersection may be given separately using the ***Crossing*** element.

Additional elements useful for describing the stop include:

- The compass ***Bearing*** towards which the vehicle is pointing when proceeding down the street past the stop. For example: 'SW'.
- Other descriptive ***Notes*** about the stop point. These are not public facing – they provide information only to users of the database.

Stop Name Uniqueness

*NaPTAN **StopPoint*** name phrases should be unique within their ***NptgLocality*** (including any parent or grandparent locality); that is, the combination of ***CommonName*** and ***Indicator*** elements should be unique.

The descriptor elements that make up stop names should be chosen so that when combined as a 'name phrase', they make a meaningful name that uniquely identifies the stop. The following is one possible order of combination:

[locality] ([locality-qualifier]), [common-name] ([indicator])

Table 2 shows some examples of preferred forms

CommonName	Indicator	Locality			Qualifier	Preferred full name
Red Lion	*o/s*	*Blacko*			-	*Blacko, Red Lion (o/s)*
Health Centre	*opp*	*Cosham*			-	*Cosham, Health Centre (opp)*
Tilford Road	-	*Farnham*			-	*Farnham, Tilford Road*
Woolworths	*opp*	*Gillingham*			*Kent*	*Gillingham (Kent), Woolworths (opp)*

Table 2 – Examples of Preferred Stop Names

Descriptor 'Atomicity'

The different descriptor elements may be combined by applications into name phrases in different ways in different circumstances (see discussion in section 1.1.1.1). Thus, the ***Landmark***, ***Street*** and ***Indicator*** elements should avoid repeating the same proper nouns as content, as this results in verbose and unintelligible compound name phrases: such *as 'o/s Red Lion Red Lion (o/s)'*.

Similarly, common names should not include the ***NptgLocality*** / ***Name*** or ***NptgLocality*** / ***Qualifier*** name unnecessarily, as again this leads to unhelpful descriptive name phrases when the elements are combined. For example, unnecessary repetition might result in '*Gillingham (Kent), Woolworths Gillingham Kent (opp)*.' In the case of rail stations and other termini, it is often the case that the locality name is included in the formally adopted common name (Table 3).

CommonName	Indicator	Locality	Qualifier	Preferred full name
Farnham Rail Station		Farnham	-	Farnham, Farnham Rail Station

Table 3 – Example Preferred Form for Rail Station Names

As an illustration, *Table 4* shows some example name elements for a stop;

Table 5 shows some of the different ways that an application might choose to create name phrases from the elements.

	Element	Value
NPTG Locality	***AdministrativeArea / ShortName:***	*Lancs*
	NptgLocality / Name:	*Blacko*
Stop Descriptors	***CommonName:***	*Red Lion*
	Landmark:	*Red Lion*
	Indicator:	*Opp*

Table 4 – Example Name Elements

Possible Derived Names
Red Lion
Red Lion (opp)
Blacko, Red Lion
Blacko (Lancs), Red Lion
Blacko, Red Lion (opp)
Blacko (Lancs), Red Lion (opp)
Gisburn Road, Red Lion
Gisburn Road, Red Lion (opp)
Blacko, Gisburn Road, Red Lion
Blacko (Lancs), Gisburn Road, Red Lion
Blacko, Gisburn Road, Red Lion (opp)
Blacko (Lancs), Gisburn Road, Red Lion (opp)
Red Lion, Blacko
Red Lion, Blacko (Lancs)
Red Lion (opp), Blacko
Red Lion (opp), Blacko (Lancs)
Red Lion, Gisburn Road, Blacko
Red Lion, Gisburn Road, Blacko (Lancs)
Red Lion (opp), Gisburn Road, Blacko
Red Lion (opp), Gisburn Road, Blacko (Lancs)

Table 5 – Ways of Deriving Names from Descriptors

Bus Stop Naming Styles

Where there is not an established name for a stop point, a new ***CommonName*** should be issued. When devising bus stop names, consideration should be given to the finding of the stop by name or partial name in computer-based stop finders; the choice of the best actual common name depends on how the stop name needs to be distinguished from other nearby stops, so that in practice any of the following naming styles may be appropriate:

1 **Locality Style:** Name the stop after the locality it serves, for example '*Little Gidding Centre'.* In some cases the actual stop name will be a generic name like *Town Centre*'. Use of the locality name as a ***CommonName*** should generally be avoided, as it is not very specific or informative and does not necessarily help users locate the stop with the locality. It is better to use a landmark (e.g. "*The Poets Arms")* or crossing name (e.g. "*High Street")* within the locality. The ***NptgLocality*** name can, of course, always be associated with the stop and used in names if appropriate to the context (e.g. "*The Poets Arms, Little Gidding*").

2 **Landmark Style**: Name the stop after the landmark or point of interest it serves, for example ' *'British Museum', 'Town Centre', 'St Trinian's School', 'Boots',* if necessary giving the relation to the landmark as the ***Indicator***. For example, '*British Museum' + 'O/s'*. The landmark may also be the street or crossing name, but a ***Street*** must also be given. This is a preferred style as it helps users relate stops to their surroundings.

3 If there is no obvious landmark, the name of a road on which the stop lies may be appropriate as long as there is only one set of stops on that road.

3.1 **Street Style:** If the road is short, and has only a single stop or pair of stops, in the street then the name of the road the stop is on may be appropriate as a ***CommonName*** if there is no other obvious style. This should be with an ***Indicator*** such as a house number, for example ‘*o/s 34’.*

3.2 **Crossing Style:** For a longer road on which there are two or more pairs or clusters of stops, then common names based on the nearest cross-street or a landmark are to be preferred, without the name of the road on which they are located (as this is available if needed from the ***Street***). The ***Indicator*** should be set to ‘nr’ or ‘adj’ for a stop on the same side of the road, ‘opp’ for a stop on the other side of the road. This is a preferred style as it helps users relate stops to their surroundings. The use of the ***Crossing*** rather than the ***Street*** name as the ***CommonName*** is preferable as it allows the future addition of more stops in the same street without ambiguity.

4 **Particular Style:** Give the stop a name that does not follow any of the above styles because of some other local usage: for example: ‘*Rail Replacement Services’.* This approach should only be used in exceptional circumstances.

General Rules for the Names of Stop Points

The following general rules should be applied to stop Common Names and other textual stop descriptor elements:

- **Capitalization**: The preferred style of stop names, place names and street names in *NaPTAN* is ‘title case’, that is lower case with the first letter of each significant word in upper case, for example, ‘*Milton Keynes’.* Prepositions and articles within a name should be in lower case *‘Isle of Man’, ‘Hole-in-the-Wall Lying-in Hospital’. ‘Kirkby-in-Furness High Street’, ‘Cley-next-the-Sea’,* **not** *‘Cley Next The Sea’.* Prepositions and articles derived from Latin or other languages should not be capitalised either; ‘*St George's-super-Ely’, ‘Poulton-le-Fylde’.*
- **Character Set**: Only uppercase and lower case letters should be used. Specifically, the use of digits, non-alphabetic characters, and any punctuation characters other than apostrophes, hyphens and ampersands should be avoided in names. Numbers should be spelt out e.g. *'Seven Sisters’*, ***not*** *'7 Sisters'.* Certain punctuation characters must ***not*** be used as they are disallowed by the schema. Note that non-letter characters such as ampersand (‘&’will need to be encoded as XML entities (e.g. &amp;) within XML content.
- **Hyphenation:** Names should be hyphenated according to the preferred form of usage by residents, for example, *‘Dudington-Fineshade’, ‘Lawton-Gate’.* Prepositions in some British place names are hyphenated, for example. *‘Lilford-cum-Wigestead’, ‘Hinton-In-the-Hedges, ‘Laughton-en-le-Morthen’, ‘Rhyd-y-Pandy’, ‘Ty’n-twr’.* Where there is a choice of usage the hyphenated form is preferred.
- **Use of Periods:** Full stops must not be used within names. For example, use just ‘*St’* rather than ‘*St.’*; do not put a final period on names.
- **Use of Commas:** Commas must not be used within names as they are conventionally used to indicate concatenation of elements when formatting names. See section 1.1.1.1 below.
- **Apostrophes:** Apostrophes should be used in accordance with the preferred local usage, and be consistent with the locality name. For example, *“Robinson’s end”, ““Cross o' th' Hands*”**, “***St Mary’s”, “Top o' th' Knowl High Street”.*

Indicator phrases: **Standard terms of relation should be used in the content of** *Indicator.* **See**

- Table 6 for details of preferred values *for **Indicator**.*

Group	Indicator term	
	source	normalised
Relation (should appear in front of common-name)	opposite	opp
	opp	
	outside	o/s
	o/s	
	adjacent	adj
	adj	
	near	nr
	nr	
	behind	
	inside	
	by	
	in	
	at	
	on	
	just before	
	just after	
	corner of	
Description (should appear after common-name)	corner	cnr
	cnr	
	drt	
	Stop	
	stance	
	stand	
	bay	
	platform	
	entrance	
	main entrance	
	side entrance	
	front entrance	
	back entrance	
	rear entrance	
	north entrance	N entrance
	east entrance	E entrance
	south entrance	S entrance
	west entrance	W entrance
	north east entrance	NE entrance
	NE entrance	
	north west entrance	NW entrance
	NW entrance	
	south east entrance	SE entrance
	SE entrance	
	south west entrance	SW entrance
	SW entrance	
	N entrance	
	E entrance	
	S entrance	
	W entrance	
	arrivals	
	departures	
	Northbound	N-bound
	N-bound	
	Southbound	S-bound
	S-bound	
	Eastbound	E-bound
	E-bound	
	Westbound	W-bound
	W-bound	
	NE-bound	
	NW-bound	
	SW-bound	
	SE-bound	

Table 6 – Preferred Phrases to Use in Indicator

The words "Stop", "stand", "stance", "bay", "platform", "entrance" can be followed by an alphanumeric string to allow for Stop codes e.g. A, 1, A1, 1A, 23, FG, AB27, etc. with the numeric part limited to one or two digits and the alpha part to one or two characters either before or after the numeric – all in an unbroken string (of up to 4 characters).

Words which indicate a relationship (nr, opp, o/s, adj, at etc) can be followed by an alphanumeric string to allow for house numbers (e.g. opp 23, o/s 76a). In this case the numeric component should permit values to 9999, with or without a single following alpha character.

In output systems, stops which have an indicator in NaPTAN which does not match one of the preferred values (including those which do not have an indicator where one is required) should be given a normalised indicator based on the value of the bearing for the Stop – so in this situation a Stop with a bearing of "N" will have a normalised indicator of "N-bound".

- **Use of Ampersand:** The ampersand character '&' is preferred to the word 'and' as a conjunction, for example, '*Bat & Ball'.*
- **Use of Abbreviations:** Abbreviations should be avoided, for example '*Great Missenden*' and **not** '*Gt Missenden', 'North Wootton'* **not** '*N.Wootton'*. The exception to this is the prefix for '*Saint*', where '*St'* should always be used, without a full stop, for example '*St Ives', 'Ilketshall St John'.* Although names and other text descriptors in the *NaPTAN* database can be up to 48 characters long, it is preferable if they can be kept to less than 24 characters.
  - Where needed, standardised abbreviations should be used. See Appendix C.12.
- **Spacing**: Words should be single spaced.
- **Use of forward Slash:** The uses of forward or backwards slashes or vertical bars in stop common names to denote alternatives must be avoided. Alternative names should be specified explicitly as separate descriptor entries.
- **Stop types :** A stop type should not be referred to in either the ***CommonName*** or the Indicator for a stop. If a stop is a Hail-and-Ride (HAR), an unmarked (CUS) or a flexible zone (FLX) stop type, then this information is available from the stop type field and it is for output systems to interpret this data and to add to its display (Hail-and-Ride), (unmarked) or (Demand Responsive Zone) as relevant or whatever else might be appropriate to the specific output system.

Naming Of Particular Types of Stop

Naming of Rail Stations

Rail station names should include the suffix phrase 'Rail Station' in their names, for example, '*Cork Rail Station'*.

Rail station names should use the definitive names used on the Irish Rail Website

http:/ https://www.irishrail.ie/travel-information/your-travel .

Naming of Airports

Airport stops should have the word 'Airport' or 'Terminal' in their name, for example, *'Southampton Airport'; 'Heathrow Terminal 1' + 'London Heathrow Terminal 1'.*

Airport names should be the definitive IATA name. Other names may be specified as alternative names.

Naming of Stop Areas

Stop area names should be the same as the common names of the main stops in the stop area.

C.8 The Classifying of Bus Stops and Other PTANs

The *NaPTAN* model provides a number of ways of classifying the stop:

- Whether the stop is *active* or *inactive*. See discussion in the NaPTAN guide.The modes of transport it supports (bus, rail etc).
- For bus stop point there are additional attributes:
  - Whether the stop is marked or not (For example many rural bus stops are not).
  - Whether it is normally a timing point in a schedule.

C.9 Associating Stop Points and Stop Areas with NPTG Localities

Every ***StopPoint*** has a primary ***NptgLocality*** within which it is situated. The *NPTG* locality specified for a stop point or stop area should be the most specific (i.e. the most precise as to area) available. For example, use a suburb of a city in preference to the whole city.

In addition ***StopPoint*** instances may also be associated with a number of alternative ***NptgLocality*** instances.

Certain major ***StopPoint*** instances may further be associated with particular ***NptgLocality*** instances as the main stop points for the locality; for example, the rail stations. Main stop points are normally central, but do not have to be physically contained within the immediate locality – for example the rail station for Norwich is a main stop for the town, but is not located at the centre.

The ***Town*** and/or ***Suburb*** on the ***Place*** element of a ***StopPoint*** should only be specified if they differ from the names of the *NPTG* locality specified for the ***StopPoint***. If they are the same, they will be derived automatically through the reference.

The association of stop areas with an ***NptgLocality*** is indirect – through the associations of the ***StopPoint*** instances within the ***StopArea***. All stop points in a stop area should be associated with the same primary *NPTG* locality, and also have similar associations as the other stops for any alternative localities.

C.10 Geocoding of Stop Points - Location

All *NaPTAN* ***StopPoint*** instances have a geocode, i.e. a spatial ***Location*** associated with them that specifies their map coordinates.

- The Irish *NaPTAN* database uses OS Grid coordinates and data should be *submitted* geocoded with Grid coordinates. For Eire ITM grid may be used (Irish Transverse Mercator).
- The *NaPTAN* schema supports the exchange of stops with both WGS 84 and grid co-ordinates, and both are provided in the *distributed* data.

The usage of location depends on the stop point classification (see *Table 7*); for on-street points and off-street entrance points, the location should be an exact single point of the doorway or pole. For logical stops representing a zone or access area, the location should be a central point chosen to give a sensible visualisation of the area on a map; and depending on type, may also be accompanied by a more detailed description of the coordinates, as for example for a hail and ride section.

Group		Type	Sub Type	Description	Shape	Location	
On Street	**Bus**	*BCT,*	MKD	On-street Bus / Coach Stop.	Point		2.0
			CUS	Custom / Unmarked stop	Point		2.0
			HAR	Hail & Ride zone.	Line	Centre of line	2.0

			FLX	Flexible Zone.	Polygon	Centre of area	2.0
		BCS		Bus / Coach bay / stand / stance within Bus / Coach Stations.	Point		2.0
	Taxi	*TXR*		Taxi Rank (head of).	Point		2.0
		STR		Shared Taxi Rank (head of	Point		2.0
	Car	*SDA*		Setdown and Pick up Area	Point	Centre of area	2.4
Off Street	**Entrance**	*AIR*		Airport Entrance.	Point		2.0
		FTD		Ferry Terminal / Dock Entrance.	Point		2.0
		TMU		Tram / Metro / Underground Entrance.	Point		2.0
		RSE		Rail Station Entrance.).	Point		2.0
		BCE		Bus / Coach Station Entrance.	Point		2.0
		LCE		liftOrCableCarStationEntrance	Point		2.4
	Access Area	*GAT*		Airport Interchange Area..	Point	Centre of area	2.0
		BST		Bus / Coach Station Interchange Area.	Point	Centre of area	2.0
		FER		Ferry or Port Interchange Area.	Point	Centre of area	2.0
		MET		Underground or Metro Interchange Area..	Point	Centre of area	2.0
		RLY		Railway Interchange Area.	Point	Centre of area	2.0
		LCB		liftOrCableCarAccessArea	Point		2.4
	Bay / Platform	*FBT*		Ferry Terminal / Dock Berth.	Point		2.0
		RPL		Rail Platform.	Point		2.0
		PLT		Underground or Metro Interchange Platform..	Point		2.0
		BCS		Bus / Coach bay / stand / stance within Bus / Coach Stations.	Point		2.0
		BCQ		Bus / Coach Station Variable Bay.	Point		2.1
		LPL		liftOrCableCarPlatform	Point		2.4

Table 7 – Stop Point Location Types

1.1.1.1Considerations for Effective Naming of Stops in Journey Planners:

Some useful insight into the effective naming of stops can be obtained by considering how stop names are used in the software interfaces that interact with end users, as for example in a journey planner stop or place finder.

1.1.1.2 Presentation of Stop Names in Disambiguation Lists

When displayed in lists in place finders, stop names will typically be prefixed by a locality name in order to provide users a context within which to recognise the common name, and to distinguish the stop name from other similar names. For example, if you enter ‘*High Street’* without a town name, there might be many possible candidates, so the locality may be added as a prefix, *‘Oxford, High Street’.*

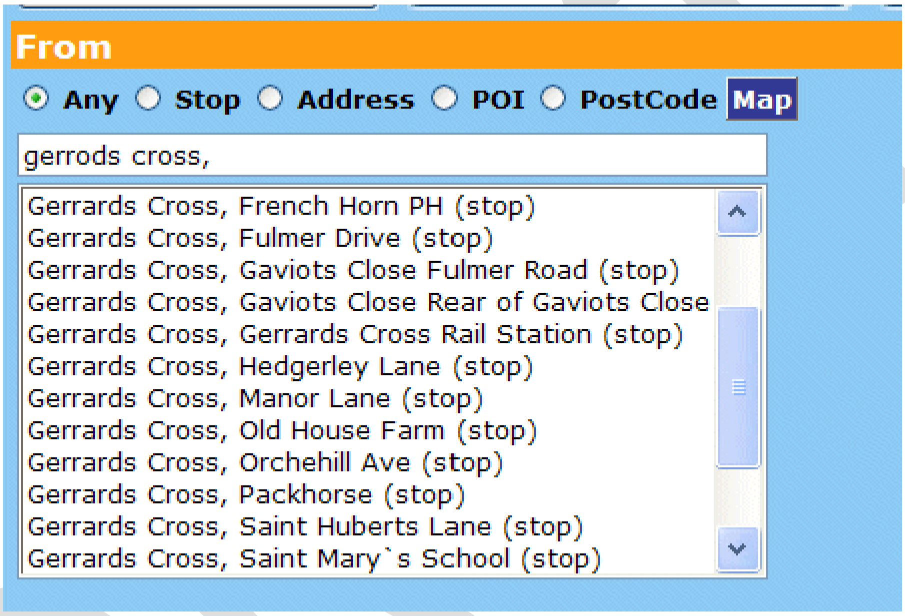
When displayed in a list in a user interface, disambiguated names will normally have a general format that is made up of several elements:

{NPTG Locality Name (+Optional Locality Qualifier)}+ {Stop Common Name} {Stop Indicator}

Note however, that different application user interfaces may vary the order in which they use to combine the elements into a 'name phrase' for presentation; for example the order 'Stop Name + Locality Name + Stop type' may also be used, or in other circumstance the locality name and/or qualifier may be omitted; for example on a map, where the context is already given.

Figure 12-1 shows an example from the South East region journey planner using Locality Name + Stop Name for bus stop points (with 'stop' appended on the end. Thus for instance, the '*Packhorse*' ***StopPoint*** in the ***NptgLocality*** '*Gerrards Cross*' would appear as: '*Gerrards Cross, Packhorse (stop)*'*.*

Note the example demonstrates the use of fuzzy phonetic matching to tolerate common types of spelling errors in the enquiry input ('*gerrods cross*').



gerrods cross,

Gerrards Cross, French Horn PH (stop)
Gerrards Cross, Fulmer Drive (stop)
Gerrards Cross, Gaviots Close Fulmer Road (stop)
Gerrards Cross, Gaviots Close Rear of Gaviots Close
Gerrards Cross, Gerrards Cross Rail Station (stop)
Gerrards Cross, Hedgerley Lane (stop)
Gerrards Cross, Manor Lane (stop)
Gerrards Cross, Old House Farm (stop)
Gerrards Cross, Orchehill Ave (stop)
Gerrards Cross, Packhorse (stop)
Gerrards Cross, Saint Huberts Lane (stop)
Gerrards Cross, Saint Mary`s School (stop)

From the SELTA region journey planner stop finder, Courtesy MDV

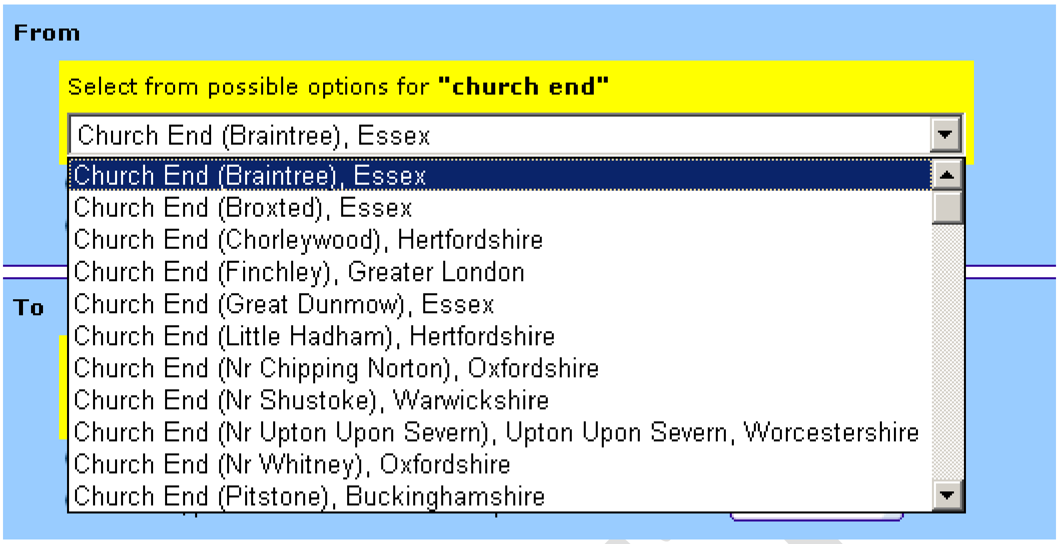
Figure 12-1 – Example of Stop Names in a List

The locality qualifier can be used in applications if the locality needs to be distinguished from other similarly named localities. For example, the '*Packhorse*' ***StopPoint*** in the ***NptgLocality*** '*Ashford* would appear as: '*Ashford (Kent), Packhorse (stop)''.*

The use of hyphens can facilitate the intelligibility of names, for example '*Sutton-on-the-Forest, Huby*' is slightly easier to read and recognise than *'Sutton on the Forest, Huby'*.

The avoidance of embedded commas in names is especially important; '*On the Forest, Sutton, Huby' is* considerably harder to interpret. Similarly trailing articles as in *'Dunks, The, High Street, The'* give rise to difficulties.

As a further *example*, *Figure 12-2* shows the results of using a place name of '*Church End*' in the Transport Direct Portal Journey planner – the various instances are distinguished by both a qualifier and an administrative area.



From

Select from possible options for **"church end"**

Church End (Braintree), Essex

- Church End (Braintree), Essex
- Church End (Broxted), Essex
- Church End (Chorleywood), Hertfordshire
- Church End (Finchley), Greater London
- Church End (Great Dunmow), Essex
- Church End (Little Hadham), Hertfordshire
- Church End (Nr Chipping Norton), Oxfordshire
- Church End (Nr Shustoke), Warwickshire
- Church End (Nr Upton Upon Severn), Upton Upon Severn, Worcestershire
- Church End (Nr Whitney), Oxfordshire
- Church End (Pitstone), Buckinghamshire

From the Transport Direct Portal Journey Planner – Atos Origin.

Figure 12-2 – Example of Ambiguous Place Names

1.1.1.3 Matching of Stop Names by Stop & Location Finders

When processing input search strings, stop finders will generally:

- Use specific special characters as delimiters (for example comma to mark the end of a locality), or commands (for example '*' for wildcard).
- Ignore extra spaces in names.
- Ignore hyphens and apostrophes.
- Ignore the difference between upper and lower case.
- Understand some common abbreviations.
- Support fuzzy and partial searches, and tolerate some common types of typing and spelling errors.

1.1.1.4 Implications for Stop Naming

We note some particular implications of the use of stop names in software user interfaces for the naming of stops:

- It is preferable if the stop common name does not repeat the locality name unnecessarily -- so as to avoid for example the informationally redundant '*Gerrards Cross, Gerrards Cross Packhorse'.* Applications may always themselves add in the locality if appropriate. However, where the locality name is an integral part of the name, for example *'Tonbridge School',* or *'Farnham Rail Station',* it should be used, even though this might result in some repetition (e.g. *Tonbridge, Tonbridge School* or *'Farnham, Farnham Rail Station').*
- The inclusion of separators such as commas in stop names generally makes them harder to interpret in lists.
- Lists may include stops of different types, so including a type phrase ('Rail Station', 'Airport', 'Coach Station') for stop type other than bus stops helps users.
- Simple names ('*Boots', 'St Mary's Church', 'Hospital', 'High Street'*), are preferred to composite names ('*Boots High Street*', '*St Mary's Church Fenham Green' 'Hospital -Furlong Road*'', '*High Street Bus Station*'). Again applications may always themselves add in the locality or other context if appropriate. Where there are two or more stops on the same road, then common names based on the nearest cross-street or a landmark are to be

preferred, without the name of the road on which they are located - since this can be obtained from the ***Street*** element of the database.

- As an exception to this rule it is however useful to include the town name in the names of Rail stations.
- The assigning of correct NPTG localities is very important.
- The consistent use of capitalisation and hyphenation improves intelligibility. Names held in the NPTG database should be in a definitive form and consistent style.
- The preferred way of populating NaPTAN name elements is so as to lead to easily recognizable names when the descriptor elements are combined by applications into a name phrase in a particular order. The preferred order is *'NptgLocality (Qualifier), Common Name (Indicator)*'. In choosing names it is helpful (i) to test them by concatenating the elements in the suggested order and considering the resulting name phrase for sense, and (ii) to compare the name phrase to those of adjacent stops to see if they are helpful in distinguishing the stop from the others.

1.1.1.5Implications for NPTG Locality Naming

Similarly considerations apply to the naming of NPTG Localities:

- Names should generally be the simple name of the locality.
- It is useful to create distinct elements to represent the central areas of towns and cities. For the names of Town and City Centres, it is useful to include the Town name as part of the name, e.g. *'Shirley Town Centre', 'Winchester City Centre'*.
- Consideration should be given as to whether a Locality name is unique within Ireland, and if it is not, a qualifier should be added.

C.11 Populating Accessibility data

NaPTAN model is intended to allow an incremental approach to capturing accessibility data, that is, data may captured to different degrees of detail according to the available resources. An overall assessment should always be provided, with further detail as available.

It should be noted that accessibility depends not just on the stop, but also on the capabilities of the vehicles (e.g. low floor, wheel chair spaces etc.) and services (e.g. assistance) that visit the stop. NaPTAN provides a means of specifying stop related data and also of indicating whether service at the stop is generally accessible or not. However to provide completely accurate information additional data is needed from other sources such as GTFS.

1.1.1.6On Street Stops

Typically the capture of top accessibility data for on-street stops such as bus coach and tram stops is more straightforward than for complex off street sites such as stations and airports, since the stop itself is directly accessible Table 8 indicates the relative priority of different elements.

On street stops are normally accessible directly at street level one and may simply be tagged as accessible or not. If all the services that visit the stop are accessible it is useful to tag the stop further, for example with low floor/hoist, wheelchair/mobility scooter.

Any boarding assistance service will usually be provided by the driver or conductor so if offered at will be available at all times.

Element	***Attribute***	Usual value		**Note**
Stop-Accessibility	***MobilityImpairedAcess***	true	1	**To be specified**
	WheelchairAccess	true	1	**To be specified**
	StepFreeAccess	(true)	2	Usually true
	LiftFreeAccess	(true)	4	Not usually relevant
	EscalatorFreeAccess	(true)	4	Not usually relevant
	AssistanceService	(unknown)	2	**Useful**
	InfoUrl		3	Useful
	ServicesAtStopUsuallyAccessible	(unknown)	**2**	**Useful**
	Note		3	
DayType	***DaysOfWeek***		4	Not usually relevant
	Timeband		4	Not usually relevant
	PublicHolidays		4	Not usually relevant
AccessVehicle-Equipment	***LowFloor***	true	2	Useful
	HighFloor	true	2	Useful
	Hoist	true	2	Useful
	HoistOperatingRadius		2	Useful
	Ramp	false	2	Useful
	Boarding Height		4	Not usually relevant
	Gap to Platform		5	Not usually relevant
	Width of Access area		4	Not usually relevant
	Height of Access area		4	Not usually relevant
	AutomaticDoors		3	Usually true
	SuitableFor	(unknown)	2	**Useful to further characterize wheelchair, mobility Scooter, etc.**
	AssistedBoardingLocation	(BoardAtAnyPoint)	5	Not usually relevant
	GuideDogsAllowed	(true)	3	Usually true

Table 8 – Populating on-street stops

1.1.1.7Off street Stops

For off street stops such as stations, access to platforms may involve paths that use steps, lifts or escalators and it is helpful to indicate these. In addition it may be relevant to indicate if the assistance is only available at particular times or needs booking. Table 9 indicates the relative priority of different elements.

Element	***Attribute***	Usual value		**Note**
Stop-Accessibility	***MobilityImpairedAcess***	(unknown)	1	**To be specified**
	WheelchairAccess	(false)	1	**To be specified**
	StepFreeAccess	(unknown)	1	**To be specified**
	LiftFreeAccess	(true)	**2**	**Useful**
	EscalatorFreeAccess	(true)	**2**	**Useful**
	AssistanceService	(false)	2	**Useful – may need booking**
	InfoUrl		3	Useful
	ServicesAtStopUsuallyAccessible	(unknown)	**2**	**Useful**
	Note		3	
DayType	***DaysOfWeek***		3	Useful
	Timeband		3	Useful
	PublicHolidays		3	Useful
AccessVehicle-Equipment	***LowFloor***	(false)	4	Not usually relevant
	HighFloor	(false)	4	Not usually relevant
	Hoist	(false)	4	Not usually relevant
	HoistOperatingRadius		4	Not usually relevant
	Ramp	(true)	**2**	**Useful**
	Boarding Height		4	Useful
	Gap to Platform		4	Useful
	Width of Access area		4	Useful
	Height of Access area		4	Useful
	AutomaticDoors		3	Usually true
	SuitableFor	(unknown)	2	**Useful to further characterize wheelchair, mobilityScooter, etc**
	AssistedBoardingLocation	(BoardAtAnyPoint)	**2**	**Useful**
	GuideDogsAllowed	(true)	3	Usually true

Table 9 – Populating off-street stops

C.12 Standard Abbreviations for Topographical Features

The following standard abbreviations for topographical features and other terms are preferred when naming stops. They should be used only where it is essential that the full name be abbreviated (to meet constraints of field-lengths in a database, for instance).

C.13 Terms for Relationship

Relationship	Abbreviation
Adjacent	Adj
Near	Nr
Opposite	Opp
Outside	O/s
Great	Gt
Greater	Gtr
Little	Lt
Upper	Upr
Middle	Mdl
Lower	Lwr
East	E
West	W
North	N
South	S
Saint	St (1)

C.14 Topographical Features

Topographical Feature	Abbreviation
Alley	Al
Approach	App
Arcade	Arc
Avenue	Ave, Av
Back	Bk
Boulevard	Bvd
Bridge	Bri
Broadway	Bway
Buildings	Bldgs
Bungalows	Bglws
Business	Bsns
Causeway	Cswy
Centre	Ctr
Church	Chu, Ch
Churchyard	Chyd
Circle	Circ
Circus	Ccus
Close	Clo, Cl
College	Col
Common	Comn
Corner	Cnr
Cottages	Cotts
Court	Ct
Courtyard	Ctyd
Crescent	Cres
Cross-roads	Xrds
Drive	Dri, Dr
Drove	Dro
Embankment	Embkmt
Esplanade	Espl
Estate	Est
Gardens	Gdns
Gate	Ga
Green	Grn, Gn
Grove	Gro
Heights	Hts
Hospital	Hosp
Industrial	Ind
Infirmary	Inf
Interchange	Intc
Junction	Jct
Lane	Ln, La
Manor	Mnr
Mansions	Mans
Market	Mkt
Mews	Mws
Mosque	Msq
Mount	Mt
Orchard	Orch
Palace	Pal
Parade	Pde
Park	Pk
Passage	Pass
Place	Pl
Police Station	Pol Stn
Precinct	Prec
Promenade	Prom
Quadrant	Quad
Road	Rd
Roundabout	Rdbt
Square	Sq
Stairs	Strs
Station	Stn
Steps	Stps
Street	St (1)
Subway	Sub
Synagogue	Syng
Terrace	Ter, Terr
Temple	Tmpl
Trading	Trdg
Turn	Tn
View	Vw
Villas	Vs
Walk	Wlk
Way	Wy
Yard	Yd

(1) St as prefix means ‘Saint’. St as suffix means ‘Street’.

C.15 Common Place name Acronyms

Term	Abbreviation
Football Club	FC
Her Majesty’s Prison	HMP
Post Office	PO
Public House	PH

Royal Air Force	RAF

C.16 Common Place name Adjectives

Adjective	Abbreviation
National	Ntl
British	Brt
Royal	Ryl
Scottish	Sct

Annex D
(informative)

European Passenger Information Profile - Identifier rules

For comparison, the EPIPI rules for identfiers are given here

D.1 Use of codespaces in EPIP

All identifiers in an EPIP conformant document must be assigned a CODESPACE and be guaranteed to be unique within it.

A common ‘epip_data’ CODESPACE is available for use in EPIP profiles; it specifies an EPIP identifier structure that can be used to create unique identifiers within a single pan-European context.

The ‘epip_data’ identifier structure is quite verbose and encodes meaning in the code structure (normally considered a bad practice) but has certain virtues: it allows for uniform and unique identifiers within a single wordwide CODESPACE and it provides a convenient ready-made solution for those who do not have already have an agreed identifier structure.

Use of the “epip_data” CODESPACE and identifier structure on frame identifiers is mandatory for the EPIP.

Use of the “epip_data” CODESPACE and identifier structure on frame contents is recommended but optional for the EPIP.

D.2 Identifier structure for EPIP data objects.

The EPIP codespace for data elements is ‘*epip_data*‘ (http:/netex-cen.eu/epip_data/), by convention indicated by the prefix ‘*epd*’ in EPIP conformant XML documents. The prefix can be omitted if declared as the default on the containing frame).

The EPIP identifier structure for objects is the following:

(epd:)[country code] : ([local code]) : [object type] (_ [epip-type]) : [technical-identifier] : [ID provider for shared IDs]

Where:

- ***[country-code]*** is the ISO 3166-1 code (2 characters, capitalised) of the country of the ***[local-code]***.
- ***[local-code]*** is a code uniquely identifying the locality or the provider within the country (may be a region code like the European NUTS code, an authority code, etc.). The European NUTS code is recommended here. However, this code is not mandatory if the other elements make the code unique (but a placeholder colon ":" must still be present).!
- ***[object type]*** is the NeTEx element type (***ServiceJourney***, **PassengerStopAssignment**, **Line**, etc. using the exact XML tag name in UpperCamelCase without a space) and is included to avoid any collision if the same [***technical-identifier]*** values are used for different types of object.
- ***[epip type]*** is an optional qualifier for the ***[object type]*** ,used to allow a common [***technical-identifier]*** to be used for closely related objects of the same element type. In the EPIP version 1.0 it is used just to distinguish monomodal and general instances of **StopPlace**, the resulting qualified ***[object-type]*** will be **StopPlace-general** or **StopPlace-monomodal** instead of **StopPlace** (see 9.2-Stop Place). (Note that in future prolifes and profile versions, additional values are expected to be added.)
- ***[technical-identifier]*** is a technical identifier for the object, it can be whatever code the producer system prefers, but must satisfy the xml *NCName* data type (i.e. consist of upper-case or lower case non-accented characters, numbers, "-" or "_") but shall be <u>unique</u> for the object within the other identifier parameters and codespace and be <u>stable</u> (that is, *persistent*;a given object must not change its identifier).

- ***[ID-provider-for-shared-IDs]*** is used to provide a reference to the ID provider (reference data system) when there is one (in effect the real CODESPACE!). When this non-mandatory field is present, it ensures the uniqueness of the ***[technical-identifier]***.
- All the "**:**" separating characters are mandatory (even if a field is empty) and are contiguous without spaces.

Examples *(purely fictional, just illustrating the coding rule)*

‘epd’= netext-cen.eu/epip_data

```
<ScheduledStopPoint id=”epd:FR:SMIRT:ScheduledStopPoint:454e74”>
<Line id=”epd:SI:LLP:Line:22”>
<Route id=”epd:NO::Route:985a:ENTUR”>
<ServiceJourney id=”epd:DE:DB:ServiceJourney:985a”>
<StopPlace id=”epd:FR:RATP:StopPlace_monomodal:43289:FR1”>
<StopPlace id=”epd:EI::StopPlace_general:199G98765431:NaPTAN”>
```

D.3 Identifier structure for EPIP version frames

The identifiers of EPIP frames also need to be unique as the same frame may be exported repeatedly in successive versions and a consumer system needs to be able to reconcile the successive data sets.

The identifier structure for a VERSION FRAME is very similar to the pattern defined above for other types of data object, and has the same four parameters to start, though for a frame, the [***epip-type]*** must be one of the set EPIP ***TypeOfFrame*** identifiers permitted by the EPIP (See *11.9*); and must of course correspond to the ***TypeOfFrameRef*** used within the frame.

A frame identifier differs however in that a ***[frame-topic]*** (that in effect is a “***[technical-identifier]”*** for a frame is chosen to reflect the contents of the frame.

The structure of a FRAME identifier is:

[country-code] : [local-code] : [frame-type] _ [epip-type] : [frame-topic]

Where

- [***frame-type]*** is one of the NeTEx ***VersionFrame*** types: ***CompositeFrame, GeneralFrame, ResourceFrame, SiteFrame, ServiceFrame***, ***TimetableFrame*** or ***GeneralFrame***)
- [***epip-type]*** is the identifier of one of the EPIP ***TypeOfFrame*** instances, viz:
  - Specific
    - *‘EU_PI_COMMON’* for Resource data (***ResourceFrame***).
    - *‘EU_PI_STOP’* for Stop data (***SiteFrame***).
    - *‘EU_PI_NETWORK’* for a Network data (***ServiceFrame***).
    - *‘EU_PI_TIMETABLE* ‘for Timetable data (***TimetableFrame***).
    - *‘EU_PI_CALENDAR’* for Calendar data (***ServiceCalendarFrame***).
    - *‘EU_PI_METADATA’* for fixed EPIP resource data (***ResourceFrame***).
  - Composite
    - *‘EU_PI_LINE_OFFER’* for a Line data set (***CompositeFrame***).

  - *‘EU_PI_NETWORK_OFFER’* for a Network data set (***CompositeFrame***).
  - *‘EU_PI_STOP_OFFER’* for a Stop data set (***CompositeFrame***).

- [***frame-topic***] is a designator of the frame content, constructed according to frame type from the main parameters that describe it. The designator should be a short name or code, fewer than fifteen characters, made only of letters (uppercase and lowercase), numbers and hyphen "**-**".
  - For *‘EU_PI_LINE_OFFER’*: The LINE number (or name if no number is available.
  - For *‘EU_PI_NETWORK_OFFER’:* The short name of the NETWORK.
  - For *‘EU_PI_STOP_OFFER’*, name or identifier of OPERATOR, ADMINISTRATIVE ZONE or TOPOGRAPHIC PLACE for which the stops are included.

In an EPIP compliant document, the elements making up the [***frame-topic]*** of the outer ***CompositeFrame s***hall also be cited as ***topics*** in the ***PublicationDelivery*** enclosing the frame**.** This make it possible for a consumer (human or machine) to interpret the [***frame-topic]*** and understand the frame’s intended scope.

NeTEx XML EXAMPLES:

For a PI_LINE_OFFER that specifies stops, networks and timetable for RATP *Line ‘1’*:

```
<PublicationDelivery version="ntx:1.1"
::::
        <topics>
            <NetworkFrameTopic>
                <Current/>
                <NetworkFilterByValue>
                    <objectReferences>
                        <LineRef="FR:RATP:Line:1"/>"
                    </objectReferences>
                    <NetworkRef="FR:RATP:Network:Noctilien"/>
                </NetworkFilterByValue>
            </NetworkFrameTopic>
        </topics>
::::
    <CompositeFrame version="3.0" id="FR:RATP:CompositeFrame_EU_PI_LINE_OFFER:RATP-1"
                    responsibilitySetRef=”FR:RATP:ResponsibilitySet:RATP”>
        <ValidBetween>
            <FromDate>2010-12-17T09:30:47.0Z</FromDate>
            <ToDate>2011-12-17T09:30:47.0Z</ToDate>
        </ValidBetween>
        <TypeOfFrameRef ref="epip:EU_PI_LINE_OFFER" versionRef="1.0" />
        <codespaces>
            <Codespace id="epip_data">
                <Xmlns>epd</Xmlns>
                <XmlnsUrl>http://netex-cen/epip_data</XmlnsUrl>
                <Description>EPIP data</Description>
            </Codespace>
            <Codespace id="epip_metadata">
                <Xmlns>epip</Xmlns>
                <XmlnsUrl>ttp://netex-cen/epip</XmlnsUrl>
                <Description>EPIP metadata </Description>
            </Codespace>
        </codespaces>
        <FrameDefaults>
            <DefaultCodespaceRef ref="epip_data"/>
            <DefaultResponsibilitySetRef version=”any” ref="FR:RATP:ResponsibilitySet:RATP"/>
        </FrameDefaults
        <frames>
            <SiteFrame version="2.1" id="FR:RATP:SiteFrame_EU_PI_STOP:RATP-1">
                <TypeOfFrameRef ref="epip:EU_PI_STOP" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </SiteFrame>
            <ServiceFrame version="2.1" id="FR:RATP:ServiceFrame_EU_PI_NETWORK:RATP-1">
                <TypeOfFrameRef ref="epip:EU_PI_NETWORK" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </ServiceFrame>
            <TimetableFrame version="2.4" id="FR:RATP:TimetableFrame_EU_PI_TIMETABLE:RATP-1">
                <TypeOfFrameRef ref="epip:EU_PI_TIMETABLE" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </TimetableFrame>
    <ServiceCalendarFrame version="1.6" id="FR:RATP:ServiceCalendarFrame_EU_PI_CALENDAR:RATP-1">
                <TypeOfFrameRef ref="epip:EU_PI_CALENDAR" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </ serviceCalendarFrame >
            <ResourceFrame version="1.6" id="FR:RATP:ResourceFrame_EU_PI_COMMON:RATP-1">
                <TypeOfFrameRef ref="epip:EU_PI_COMMON" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </ResourceFrame>
        <ResourceFrame version="1.0" id="epip:EU:CEN:ResourceFrame_EU_PI_METADATA:EU"
                    responsibilitySetRef=”epip:EU:CEN:ResponsibilitySet:TC278_WG3_SG9”>
                <TypeOfFrameRef ref="epip:EU_PI_METADATA" versionRef="1.0" />
                    <Etc., etc., Specific Frame Fixed contents>
            </ResourceFrame>
        </frames>
    </CompositeFrame>
```

(d) For an *EU_PI_NETWORK_OFFER that* specifies stops, networks and timetables for Llubjana (showing here separate contained ‘*EU_PI_NETWORK’* & ‘*EU_PI_TIMETABLE ‘* frames with modularised details for only two lines ‘LPP-1’ and ‘LPP-2’ ; there would be more). Some of the dependencies between the frames are also indicated as ***prerequisites***.

```
<PublicationDelivery version="ntx:1.1"
::::
        <topics>
            <NetworkFrameTopic>
                <Current/>
                <NetworkFilterByValue>
                    <places>
                        <TopographicPlaceRef ref="LLJUBJANA”/>
                    </places>
                    <NetworkRef="SI::Network:LLP"/>
                </NetworkFilterByValue>
            </NetworkFrameTopic>
        </topics>
::::
    <CompositeFrame version="3.0" id="epd:epd:EI:PHBS:CompositeFrame_EU_PI_NETWORK_OFFER:LLJUBJANA"
```

```
                    responsibilitySetRef=”epd:EI:PHBS:ResponsibilitySet:LPP”>
        <TypeOfFrameRef ref="epip:EU_PI_NETWORK_OFFER" versionRef="1.0" />
        <codespaces>
            <Codespace id="epip_data">
                <Xmlns>epd</Xmlns>
                <XmlnsUrl>http://netex-cen/epip_data</XmlnsUrl>
                <Description>EPIP data</Description>
            </Codespace>
            <Codespace id="epip">
                <Xmlns>epip</Xmlns>
                <XmlnsUrl>ttp://netex-cen/epip</XmlnsUrl>
                <Description>EPIP metadata </Description>
            </Codespace>
        </codespaces>
        <FrameDefaults>
            <DefaultCodespaceRef ref="epip_data"/>
            <DefaultResponsibilitySetRef version=”any” ref="epd:EI:PHBS:ResponsibilitySet:LPP"/>
        </FrameDefaults
        <frames>
            <SiteFrame version="2.1" id="epd:EI:PHBS:SiteFrame_EU_PI_STOP:LLJUBJANA">
                <TypeOfFrameRef ref="epip:EU_PI_STOP" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </SiteFrame>
            <ServiceFrame version="2.1" id="epd:EI:PHBS:ServiceFrame_EU_PI_NETWOK:PHB@1">
                <TypeOfFrameRef ref="epip:EU_PI_NETWORK" versionRef="1.0"/>
                <prerequisites>
                    <ResourceFrameRef version="any" ref="SI:LLP:SiteFrame_EU_PI_STOP:LLJUBJANA"/>
                </prerequisites>
                    <Etc., etc., Specific Frame contents>
            </ServiceFrame>
            <TimetableFrame version="2.4" id="epd:EI:PHBS:TimetableFrame_EU_PI_TIMETABLE:PHB@1">
                <TypeOfFrameRef ref="epip:EU_PI_TIMETABLE" versionRef="1.0"/>
                <prerequisites>
                    <ServiceFrameRef version="any" ref="SI:LLP:ServiceFrame_EU_PI_SERVICE:PHB@1"/>
                </prerequisites>
                    <Etc., etc., Specific frame contents>
            </TimetableFrame>
            <ServiceFrame version="2.1" id="epd:EI:PHBS:ServiceFrame_EU_PI_NETWORK:PHB@2">
                <TypeOfFrameRef ref="epip:EU_PI_NETWORK" versionRef="1.0" />
                <prerequisites>
                    <ResourceFrameRef version="any" ref="SI:LLP:SiteFrame_EU_PI_STOP:LLJUBJANA"/>
                </prerequisites>
                    <Etc., etc., Specific Frame contents>
            </ServiceFrame>
            <TimetableFrame version="2.4" id="epd:EI:PHBS:TimetableFrame_EU_PI_TIMETABLE:PHB@2">
                <TypeOfFrameRef ref="epip:EU_PI_TIMETABLE" versionRef="1.0" />
                <prerequisites>
                    <ServiceFrameRef version="any" ref="SI:LLP:ServiceFrame_EU_PI_SERVICE:PHB@2"/>
                    <ServiceCalendarFrameRef version="any"
ref="SI:LLP:ServiceCalendarFrame_EU_PI_CALENDAR:LLJUBJANA "/>
                </prerequisites>

                    <Etc., etc., Specific Frame contents>
            </TimetableFrame>


            <Etc., etc., Servcie & Timetable frames for other liness>
```

```
    <ServiceCalendarFrame version="1.6"
id="epd:EI:PHBS:ServiceCalendarFrame_EU_PI_CALENDAR:LLJUBJANA">
                <TypeOfFrameRef ref="epip:EU_PI_CALENDAR" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </ serviceCalendarFrame >
            <ResourceFrame version="1.6" id="epd:EI:PHBS:ResourceFrame_EU_PI_COMMON:LLJUBJANA">
                <TypeOfFrameRef ref="epip:EU_PI_COMMON" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </ResourceFrame>
        <ResourceFrame version="1.0" id="epip:EU:CEN:ResourceFrame_EU_PI_METADATA:EU"
                    responsibilitySetRef="epip:EU:CEN:ResponsibilitySet:TC278_WG3_SG9">
                <TypeOfFrameRef ref="epip:EU_PI_METADATA" versionRef="1.0" />
                    <Etc., etc., Specific Frame Fixed value contents>
            </ResourceFrame>
        </frames>
    </CompositeFrame>
```

(e) For an '*EU_PI_STOP_OFFER'* with the stops for VBB in Berlin divided between two EU_PI_STOP frames:

```
<PublicationDelivery version="ntx:1.1"
::::
        <topics>
            <NetworkFrameTopic>
                <Current/>
                <NetworkFilterByValue>
                    <objectReferences>
                        <OperatorRef="DE:VBB:Operator:VBB"/>"
                        <TariffZoneRef="DE:VBB:TariffZone:Berlin"/>"
                    </objectReferences>
                    <NetworkRef="DE:VBB:Network:VBB"/>
                </NetworkFilterByValue>
            </NetworkFrameTopic>
        </topics>
::::

    <CompositeFrame version="3.0" id="epd:DE:VBB:CompositeFrame_EU_PI_STOP_OFFER:VBB">
                    responsibilitySetRef="DE:VBB:ResponsibilitySet:VBB">
        <TypeOfFrameRef ref="epip:EU_PI_STOP_OFFER" versionRef="1.0" />
        <codespaces>
            <Codespace id="epd">
                <Xmlns>epd</Xmlns>
                <XmlnsUrl>http://netex-cen/epip_data</XmlnsUrl>
                <Description>EPIP data</Description>
            </Codespace>
            <Codespace id="epip">
                <Xmlns>epip</Xmlns>
                <XmlnsUrl>ttp://netex-cen/epip</XmlnsUrl>
                <Description>EPIP metadata </Description>
            </Codespace>
        </codespaces>
        <FrameDefaults>
            <DefaultCodespaceRef ref="epip_data"/>
            <DefaultResponsibilitySetRef version="any" ref=" DE:VBB:ResponsibilitySet:VBB"/>
        </FrameDefaults
        <frames>
            <SiteFrame version="2.0" id=" DE:VBB:SiteFrame_EU_PI_STOP:Berlin@Inner">
            <TypeOfFrameRef ref="epip:EU_PI_STOP" versionRef="1.0"/>
                <Etc etc.
            </SiteFrame>
            <SiteFrame version="2.0" id=" DE:VBB:SiteFrame_EU_PI_STOP:Berlin@Outer">
            <TypeOfFrameRef ref="epip:EU_PI_STOP" versionRef="1.0"/>
                <Etc etc.
            </SiteFrame>
            <ResourceFrame version="1.6" id=” DE:VBB:ResourceFrame_EU_PI_COMMON:Berlin">
                <TypeOfFrameRef ref="epip:EU_PI_COMMON" versionRef="1.0" />
                    <Etc., etc., Specific Frame contents>
            </ResourceFrame>

        </frames>
    </CompositeFrame>
```

Bibliography

General

[1] ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times. (http://www.w3.org/TR/xmlschema-2/ – isoformats)*

[2] ISO 639/IETF 1766, *Tags for the Identification of Languages*

[3] WGS 1984 Location Referencing World Geodetic Standard 1984

[4] ISO 3166-2, Codes for the representation of names of countries and their subdivisions.

[5] ISO/IEC 19501-1:2002, *Unified Modelling Language (UML) – Part 1: Specification*

[6] XML, Extensible Mark-up Language (XML) 1.0 W3C Recommendation 04 February 2004, available at http://www.w3.org/TR/2004/REC-xml-20040204.

[7] PTIMS-UD Public Transport Infrastructure Management System Universal Design Rating V1 NTA April 2020

Public transport data formats

[8] NaPTAN & NPTG Schema User Guide 2.5 UK Department for Transport 2013

[9] VDV 452, 7/13 Standard VDV Route Network / Timetable Interface including enhancements: • Transfer definitions including their validity VDV https://www.vdv.de/

[10] *VDV Recommendation 462- NeTEx VDV Application Handbook: Standardised exchange of line network and timetable data with the European standard CEN-TS 16441 vdv-462-netex-recommendation-v00-22-english.pdfx*

[11] GTFS Google Transit Feed Specification. https://developers.google.com/transit/gtfs/reference

[12] HRDF – HaFas Rohdatenformat (HaFas raw data format). Hacon gmbh, https://opentransportdata.swiss/de/cookbook/hafas-rohdaten-format-hrdf/). https://de.wikipedia.org/wiki/HAFAS

[13] INFO+ HRDF-Exportschnittstelle 5.20.39 Spezifikation der nach HRDF 5.20.39 exportierten Datei- und Zeilentypen in INFO+ SBB CF FFS june 2016. https://www.fahrplanfelder.ch/fileadmin/fap_daten_test/hrdf.p