CEN TC 278

Date: 2019 -04

prTS 16614:PI Profile

TC WI 00278457

Secretariat: NEN

Public transport — Network and Timetable Exchange (NeTEx)— Passenger Information European Profile

Einführendes Element — Haupt-Element — Ergänzendes Element

Élément introductif — Élément central — Élément complémentaire

ICS:

CCMC will prepare and attach the official title page

Document type:

Document subtype:

Document stage:

Document language: E

Contents

European foreword

This document (WI 00278307) has been prepared by Technical Committee CEN TC 278 “Intelligent transport systems”, the secretariat of which is held by NEN.

This document is currently submitted to the CEN Enquiry.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association, and supports essential requirements of EU Directive(s).

For relationship with EU Directive(s), see Annex C, which is an integral part of this document.

A list of all parts in the TS 16614 series can be found on the CEN website.

This document presents a Passenger Information European Profile of the European Technical Specification known as “NeTEx”. NeTEx provides a framework for specifying communications and data exchange protocols for organizations wishing to exchange scheduled Information relating to public transport operations.

NeTEx is made up of three parts defining a single European Standard, which provides a complete exchange format for public transport networks, timetable description and fare information.

- Part 1 is the description of the public transport network topology exchange format. It also contains use cases shared with part 2, and modelling rules and the description of a framework shared by all parts.
- Part 2 is the description of the scheduled timetables exchange format.
- Part 3 is the description of the fare information exchange format.

Part 1 is fully standalone, parts 2 and 3 rely on part 1.

The 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.

This document is highly technical, and special care has been taken on keeping the text readable. This has been done through a set of editorial rules enhancing usual CEN writing rules:

- To avoid confusion with usual wording, Transmodel terms are in capital letters (JOURNEY PATTERN for example).
- To avoid confusion with usual wording, attribute names are in bold/italic style and use camelcase style with no spaces (***JourneyPattern*** for example).
- To avoid confusion with usual wording, attribute types are in italic style and use camelcase style with no spaces (*TypeOfEntity* for example).

Introduction

Public transport services rely increasingly on information systems to ensure reliable, efficient operation and widely accessible, accurate passenger information. These systems are used for a range of specific purposes: setting schedules and timetables; managing vehicle fleets; issuing tickets and receipts; providing real-time information on service running, and so on.

This European Technical Specification specifies a Profile of Network and Timetable Exchange (NeTEx) for public transport. NeTEx is intended to be used to exchange information between Public Transport organisation systems containing scheduled public transport data. It can also be seen as a complement to the SIRI (Service Interface for Real-time Information) standard (EN 15531 series), as SIRI needs reference data exchanged in the scope of NeTEx before any possible real-time exchange.

Well-defined, open interfaces have a crucial role in improving the economic and technical viability of public transport Information Systems of all kinds. Using standardized interfaces, systems can be implemented as discrete pluggable modules that can be chosen from a wide variety of suppliers in a competitive market, rather than as monolithic proprietary systems from a single supplier. Interfaces also allow the systematic automated testing of each functional module, vital for managing the complexity of increasing large and dynamic systems. Furthermore, individual functional modules can be replaced or evolved, without unexpected breakages of obscurely dependent functions.

NeTEx improves a number of features of public transport information and service management:

- Interoperability – the standard will facilitate interoperability between information processing systems of the transport operators by:
  (i) introducing common architectures for message exchange;
  (ii) introducing a modular set of compatible information services for real-time vehicle information;
  (iii) using common data models and schemas for the messages exchanged for each service;
  (iv) introducing a consistent approach to data management.
- Technical advantages include the following: reusing a common communication layer shared with SIRI for all the various technical services enables cost-effective implementations, and makes the standard readily extensible in future.

A profile is an ancillary document to the standard which specifies additional rules for implementation in a given context. The profile contains information such as:

- Details of the objects used in an exchange.
- Details on the options proposed by the standard.
- Details on optional elements.
- Precision on the identifier codes to be used.
- Advice on grouping elements.

The reason for having a detailed profile specification is that it facilitates implementation. A developer intending to implement a certain service or type of service in a given environment need only implement the smaller set of options and parameters specified in the profile, and is is given a number of additional rules that restrict or simplify the required processing of data.

This European Passenger Information Profile (EPIP) for NeTEx is for exchanging passenger information; it describes a minimal information set to feed passenger information services in a European wide and multimodal context. It covers basic network and timetable data as explained further below.

1 Scope

This technical specification is a profile of CEN/TS 16614 series. It focuses on information relevant to feed passenger information services and excludes operational and fares information.

NeTEx is dedicated to the exchange of scheduled data (network, timetable and fare information) based on Transmodel V6 (EN 12986) and SIRI (CEN/TS 15531-4/5 and EN 15531-1/2/3) and supports information exchange of relevance to public transport services for passenger information and AVMS systems.

As for most data exchange standards, defining subsets of data and dedicated rules for some specific use case is of great help for implementers and for the overall interoperability. This subset is usually called profile and this profile targets passenger information as only use case.

1.1 Profiles – a Rationale

Standards are by their nature and definition, consensus documents. In the case of the CEN and ISO standards, and more particularly in the field of traffic applications, these are established at an international level, with a significant number of different stakeholders involved in their development and discussion. This means that CEN and ISO standards take into account requirements that are far beyond many local implementations of such standards. Implementing a standard for a complex data model (such as that for transport) represents a significant investment, so care is taken to achieve a comprehensive and well abstracted solution and standards are written to have with the longest possible life and stability.

These factors tend to result in standards documents being large and detailed, consequently requiring significant effort to read and understand – and this is certainly the case for Transmodel and NeTEx. It may also be difficult to see how the standard can best applied in practice, since there are many subtle choices to make out of all the possibilities and there is insufficient space to include extensive examples in the formal documentation.

Another cause of complexity is that standards such as NeTEx and SIRI (since they were developed to harmonise different national standards), include features and options whose purpose is to ensure compatibility with the different systems developed in specific countries. For example, SIRI includes services dedicated to the advanced management of connection guarantees, as implemented in the German VDV standard, but used only in a few countries such as Switzerland and Germany; NeTEx also includes features to help compatibility with practices peculiar to particular countries, such as are found in the French NEPTUNE, British TransXChange, Swedish NOPTIS, etc. standards.

In addition, local or national specificities may require the specific use of a specific data sets and specific identifier systems for particular information. For example, the UK has a national system for the identification of stops (NaPTAN), which is naturally required to be used within NeTEx and SIRI exchanges, but which is not relevant for other European countries except in a few cross-border services.

Finally, some elements proposed by the standards are optional and relevant only for a particular business context: so that it needs to be decided if these items are relevant for a particular implementation or not.

A profile is intended to address some of the above issues by offering a predefined set of choices for use in specific context, and setting out additional explicit rules that help to simplify implementation.

Additionally, a profile can be accompanied by the definition of specific testing procedures to assess the conformance of the implemented solution with the profile.

A profile remains fully compliant with the standard; it merely implements a subset, using a well-defined code system.

From a practical point of view, profiles can be seen as a set of implementation guidelines; instead of having to face the challenge of analysing the whole standard, discovering the relevant parts for a given application, and then adjusting optional values and parameters for the intended application, a profile can be specified to address the needs of a particular application and then used for any subsequent similar initiatives.

1.2 The European Passenger Information Profile (EPIP)

The EPIP focuses on information relevant for feeding passenger information services, it therefore excludes operational information that is not relevant for this purpose. Fare information, apart from basic Tariff zones, though relevant for passengers, is also considered as out of scope, but will be addressed by a future separate profile.

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 as concise as possible and to focus on the data elements needed to fulfil the EC's *Priority action A of ITS-Directive*.

- This means that features specific to a particular country will not in general not be included (the profile is designed to be sharable across Europe, and data sets compliant with it are expected to be usable by any NeTEx conformant passenger information system in Europe).
- Each country may also define a further national NeTEx Profile (for internal use nationally), extending the European profile and providing additional information.
- In the interests of simplicity, a number of the more complex features specific to certain modes have also been omitted from the formal profile, for example boarding position on train platforms.

The EPIP is divided into three parts:

1. **Common core 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.)
2. **Stop Profile**, describing the STOP PLACEs and their hierarchy.
3. **Network and Timetables Profile**, describing the network topology and all related timing information (timetables).

Dividing the profile into three parts allows for a modular implementation, with a minimal overlap between separate documents and their corresponding components. The modularisation also facilitates the separate exchange of stop data from that of timetable and other information (but it is, of course, still possible to exchange them together). One of the reasons for this partition is that there are more and more countries developing separate national or regional stop database registries who require the ability to exchange only the stop description.

The EPIP supports certain common features that are important in a European context:

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

1.3 Data consistency and Quality

One of the goals of a profile is to simplify and enhance interoperability. However, mere syntactic compliance with the EPIP 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 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. A number of rules may be controlled automatically, thus for example, it is possible to check that a SERVICE JOURNEY has PASSING TIMEs for each stop, and that these are plausibly spaced; but other quality measures cannot necessarily be checked. For example, the lack of a STOP PLACE name can be detected, as can the use of a duplicate, but not whether it is actually the correct one in use in the real-world.

1.4 NeTEx versions

The EPIP is specified for the revised version 1.1 of NeTEx, issued in 2019. It may also be used with the original version 1.0 of NeTEx, issued in 2014, though certain elements are of course lacking.

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.

2 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.

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.

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.*

3 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.

3.1
attribute
property of an entity.

3.2
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.

3.3
conformant
satisfying all the rules of a specification, both syntactic and otherwise; conformance may be strict, augmented, or extended.

3.4
consumer
a system that imports profile conformant data.

3.5
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.

3.6
data model
description of a real-world domain in terms of data and relationships.

3.7
data provider
organisation responsible for providing data (this may be distinct from that of the data source or the producer).

3.8
dominant validity condition
validity condition that attaches to a version frame and applies to all its contents.

3.9
entity
object (data) that has its own existence (as opposed to an attribute).

3.10
external reference
a reference to another object in a document other than the document holding the reference.

3.11
fare management
all activities related to the collection of money from passengers.

3.12
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.

3.13
internal reference
a reference to another object in the same document as that holding the reference.

3.14
interoperability
ability of (sub)systems to interact with other (sub)systems according to a set of predefined rules (interface).

3.15
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.

3.16
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.

3.17
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).

3.18
passenger information
all activities related to informing the users either about the planned or about the actual transportation services.

3.19
producer
a system that outputs data intended to be conformant to a profile.

3.20
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.

3.21
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.

3.22
profile metadata
data describing the properties of a profile, such as the permitted frame types and restricted code values.

3.23
profile validator
software tool that checks the content of a profile conformant XML document to ensure that it is compliant with the profile specification.

3.24
real-time control
see Operations monitoring and control.

3.25
scheduling
see Tactical Planning.

3.26
specific frame
version frame that may only contain certain types entities concerned with a particular functional area, e.g. to describe a timetable.

3.27
strict conformance
conformance to a profile such that a document contains only the specified elements (i.e. that is not augmented or extended).

3.28
tactical planning
all activities related to the tactical planning of transportation, split into vehicle scheduling, driver scheduling, rostering.

3.29
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.

3.30
XML validator
A software tool that checks the content of an XML document to ensure that it is compliant with a schema.

4 Presentation conventions

The conventions used in this document are an extension of the ones used in the main NeTEx document.

4.1 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,*** *fo*r 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.*** When there is a reflexive relationship, or more than one relationship between the same two types these are qualified by a prefix to distinguish between them for example, ***ParentZoneRef, FromPointRef, ToPointRef***, etc.
- One-to-many relationships that are implemented as collections are lower cased and are always plural, for example, ***stopPoints***, ***vehicleJourney;*** *thus, the relationship name is used as a wrapper tag*. Such collections may variously comprise sets of simple versioned references (e.g. multiple instances of ***ScheduledStopPointRef)*** or of embedded elements whose full defining 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».

4.2 Additional EPIP presentation conventions

A few additional conventions have been added beyond those used in NeTEx in order to cover specific aspects of EPIP:

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

Table 0 - Additional presentation convention

«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, etc.) for the profile.

No highlight means that it
identical or similar (same mea
to the original NeTEx tex

Light Grey highlight emphas
additional rule or constraint a
to the original NeTEx tex

(2) Profile code values

Wherever possible, the EPIP 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 EPIP 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. EPIP itself uses the TYPE OF VALUE mechanism in a few cases to specify additional standardised codes: these are assigned to an "*epip_metadata"* CODESPACE (https://netex-cen.eu/epip) indicated by an *'epip'* prefix. The identifiers of EPIP values are shown in lower case italics (if necessary, with underscore marks between words), and appear within italics in text, for example, '*epip:monomodal'*.
c. ***TypeOfFrame*** instances: The EPIP uses several 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, '*epip:EU_PI_LINE_*OFFER'**.**

(3) Indication of Abstract classes

NeTEx 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 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 EPIP 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 EPIP. These are flagged **(Abstract in EPIP)**
  In this document

(4) Subcomponent Classes

A number of EPIP classes have subcomponents that make up their definition. These provide ancillary detail (for example ***AlternativeText***, ***AlternativeName, TrainComponent***) and are flagged in headings by the suffix '(**Subcomponent of XXX)',** where XXX is their usual parent. They can be ignored on a first reading.

(5) 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).

4.3 Transmodel definitions and EPIP

The NeTEx specification uses the same formal definitions to describe its XML class elements as those used by Transmodel to describe the corresponding ENTITies.

The EPIP adheres to these definitions but, since it is intended to be as readable as possible, in certain places the EPIP 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 EPIP specification also uses a different order of presentation of the elements compared to that used in Transmodel and NeTEx.

5 Framework

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

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

5.1 DataManagedObject, Version and Codespace

DataManagedObject, together with ***Version*** and ***Codespace***, provides uniform, common properties for all NeTEx entities. ***DataManagedObject*** is the abstract "upper object" of the NeTEx implementation and all first class NeTEx entities are specialisations of it.

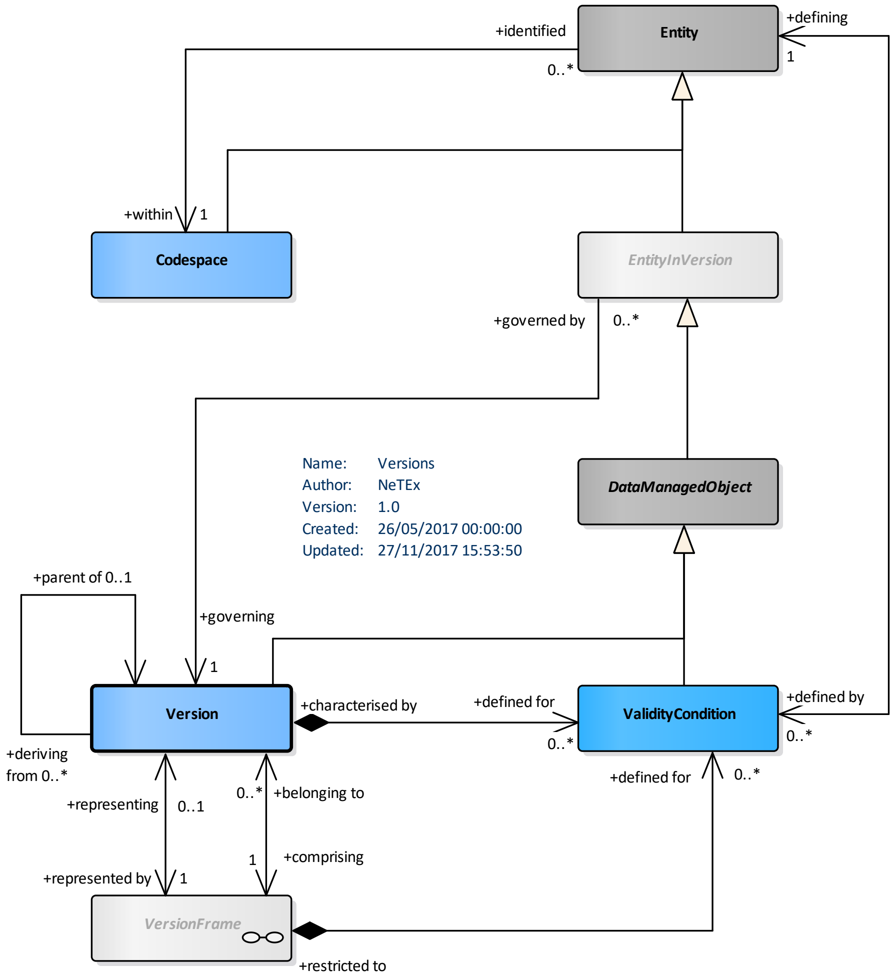

Figure 1 — Versions and Codespaces (physical model) (UML)

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 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 a basic 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*** it to ensure that the identifiers of instances of elements in an XML document are unique, even if they come from many different sources – see ***5.1.7-Codespace*** and and also ***8.4-Codespaces and the structure of identifiers***.

A ***ValidityCondition*** is used to attach conditions to elements to indicate when they are current or 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*****.**

5.1.1 Entity (Abstract)

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

Table 1 — *Entity* – XML Element

Classification	Name	Type	Cardinality	Description
«PK»	***id***	*ObjectIdType*	1:1	Identifier of ENTITY. Unique within CODESPACE. Abstract – always use the specific concrete identifier. See ***8.4-Codespaces and the structure of identifiers*** for coding rules.

5.1.2 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.

Table 2 — *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* • *unchanged* 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*'. Only '*active*' is supportd in EPIP
				0:1	
«cntd»					
	b	***ValidBetween***	*ValidBetween*	**0**:1	OPTIMISATION. Simple version of a VALIDITY CONDITION. Comprises a simple period. NO UNIQUENESS CONSTRAINT.

5.1.3 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*** shall **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 3 — *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»				
«FK»	***BrandingRef***	*BrandingRef*	0:1	Reference to a BRANDING. (A ***TypeOfValue***) with no added attributes
«cntd»	***alternativeTexts***	*AlternativeText*	0:*	Additional Translations of text elements.

5.1.4 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).

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

Table 4 — 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. Shall 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.
	Text	*MultilingualString* (Language + Text)	1:1	Variant of the text, in the specified language.

5.1.5 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.

In EPIP, extensions to elements will be based only on the ***Key***/***Value*** mechanism (i.e. the ***Extensions*** tag will not be used). Furthermore, **KeyList** shall not be used to carry information that can be encode elsewhere in NeTEx.

Table 5 — *KeyValue – XML Element*

Classification	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.

5.1.6 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 EPIP. However frame level versioning is mandatory in EPIP.

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.

In EPIP the ***Version*** element itself not populated; a simple version number in the ***version*** attribute of an EPIP element is used to indicate the version.

5.1.7 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 shall unique, (using the generic W3C domain mechanism), thus permitting the combining of elements from different schemas in the same aggregated model. 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 placed 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 that (a) check each identifier is unique and (b) that 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 XML validator to any NeTEx document and are fundamental for establishing a basic quality level for data exchange. See section 8.15.

In the context of EPIP, CODESPACEs prevent identifier clashes among data provisions coming from different countries (and regions, authorities and operators within countries. Different code structures may be used within different CODESPACES – see ***8.4-Codespaces and the structure of identifiers*** for further discussion.

In order to manage a uniform coding system for use by many stakeholders, it may be desirable to partition a CODESPACE into different 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 the zone on a distributed basis. The central registrar function is needed only to setting up of the zones and issue the prefixes in the first place. ADMINISTRATIVE ZONES are not exchanged directly in the strict EPIP, but can be referenced with an external reference from a ***ResponsibilitySet*** (see 5.1.10)

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*', or '*sbb.ch*'. The path can be declared in the `<XmlnsUrl>` attribute of a ***Codespace*** element.

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 in that 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), even if the identifiers come from heterogenous coding systems. 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.

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

XML EXAMPLE CODESPACE declarations:

```
    <CompositeFrame version="1.0" id="mybus:Line1">
            <codespaces>
                <Codespace id="napt">
                    <Xmlns>napt</Xmlns>
                    <XmlnsUrl>http://naptan.org.uk/naptan</XmlnsUrl>
                    <Description>UK NaPTAN</Description>
                </Codespace>
                <Codespace id="era_stops">
                    <Xmlns>era</Xmlns>
                    <XmlnsUrl>http://era.org.eu/stops</XmlnsUrl>
                    <Description>European Rail Authority</Description>
                </Codespace>
            </codespaces>
```

Examples of elements that use of the above CODESPACEs:

```
        <StopPlace id="napt:4701234567">
        <StopPlaceRef ref="napt:4701234567">
        <StopPlace id="era: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 UK 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 (for example a system that is merely aggregating data to pass it onto others might be agnostic about the structure of identifiers). 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. *'napt'*
	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.

5.1.8 ValidityCondition

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

EPIP restricts use of VALIDITY CONDITIONs to just a simple name (the full NeTEx specification allows complex, value dependent conditions to be constructed).

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.
	Description	*MultilingualString*	0:1	Description of VALIDITY CONDITION. See comment on the use of *partial* for *AccessibilityLimitation*

5.1.9 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*** shall be specified on a ***ValidBetween*** instance specifying the dominant condition on an EPIP 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 AVAILABILITY CONDITION. Date is INCLUSIVE.
	ToDate	*xsd:dateTime*	0:1	End of AVAILABILITY CONDITION. Date is INCLUSIVE.

5.1.10 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.

In EPIP, 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).

Table 9 — *ResponsibilitySet* – XML Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	RESPONSIBILITY SET inherits from DATA MANAGED OBJECT.
«PK»	***id***	*ResponsibilitySetIdType*	1:1	Identifier 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.

5.1.11 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 10 — *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* • *aggregates* Only a subset of data related roles are selected for the EPIP, these 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)*
«enum»	***StakeholderRoleType***	*StakeholderRoleTypeEnum*	0:1	Stakeholder role type of RESPONSIBILITY ROLE. • *Planning* • *Operation* Only a subset of data roles are 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.
«FK»	***Responsible-OrganisationRef***	*OrganisationRef*	0:1	Reference to an ORGANISATION to which this RESPONSIBILITY ROLE is assigned.
«FK»	***ResponsibleAreaRef***	*AdministrativeZoneRef*	0:1	Reference to an ADMINISTRATIVE ZONE to which this RESPONSIBILITY ROLE is assigned.

5.1.12 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 UK 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="napt:910GWIMBLDN">.
```

Usually all the data 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 5.4.1 below.

5.2 Relationships and entity references

In order to turn any computer-based representation of a data set (as an object model or database) into an EPIP conformant NeTEx data file that can be exchanged, it shall 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 on 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.

XML EXAMPLES

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

```
<StopPlace version="8.6" id="napt:910GWIMBLDN">
    <quays>
        <Quay version="8.4" id="napt:910GWIMBLDN1">
            <Name>Platform 1</Name>
        </Quay>
        <Quay version="8.6" id="napt:910GWIMBLDN2">
            <Name>Platform 2 </Name>
        </Quay>
    </quays>
</StopPlace>
```

For example, to serialise a many-to-one ***StopPlace* ➔ *TopographicPlace*** association for Wimbledon station, a versioned reference to the ***TopographicPlace*** is held on the ***StopPlace***:

```
<StopPlace version="001" id="napt:910GWIMBLDN">
    <Name>Wimbledon Rail Station</Name>
    <TopographicPlaceRef version="any" ref="nptg:E0034695">Wimbledon</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:910GWIMBLDN">
    <types>
            <TypeOfZoneRef version="any" ref="napt:Type_of_zone@GRLS"/>
            <TypeOfZoneRef version="any" ref="uicStationCat:476"/>
    </types>
</StopPlace>
```

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 (and in the EPIP shall not be). For example, the ***ParentZoneRef*** shown here is ***not*** needed on the ***Quay*** instances because they are embedded inline within the parent ***StopPlace*..**

```
<StopPlace version="8.6" id="napt:910GWIMBLDN">
    <quays>
        <Quay version="8.4" id="napt:910GWIMBLDN1">
            <Name>Platform 1</Name>
            <ParentZoneRef version="8.6" ref="napt:910GWIMBLDN">NOT NEEDED</ParentZoneRef>
        </Quay>
        <Quay version="8.6" id="napt:910GWIMBLDN2">
            <ParentZoneRef version="8.6" ref="napt:910GWIMBLDN">NOT NEEDED</ParentZoneRef>
            <Name>Platform 2 </Name>
        </Quay>
    </quays>
</StopPlace>
```

5.2.1 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 11 — 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
«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 (choice)				
«FK»	***version***	*VersionIdType*	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 NeTEx's XML automatic consistency check). The CODESPACE of the ***version*** and ***versionRef*** is assumed to be the same as that of the ***ref.***
«FK»	***versionRef***	*VersionRef*	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.

5.2.2 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.

```
<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;

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

For further details on the use of references in the EPIP see section 8.4

5.2.3 Identity uniqueness scope

All the specialisations of ***DataManagedObject*** have an ***id***. Within a given document, the value of this shall 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 technically possible (but not advisable) 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 shall 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 a specific type in the code structure (as does the EPIP common code structure use with the ‘*epip_data’* ***Codespace***, – see 8.4.2) then all codes are inherently unique within their specific type.

Table 12 — NeTEx classes used in EPIP with a shared uniqueness scope

Supertype	**EPIP Concrete classes**	Comment
POINT	***ScheduledStopPoint, TimingPoint***	Useful to have common scope

LINK	***ServiceLink, RouteLink, PathLink,***	Unnecessarily over restrictive
ZONE & PLACE	***Topographic Place, Address*** ***Tariff Zone*** ***StopPlace, Quay, StopPlaceEntrance,*** ***PointOfInterest,***	Unnecessarily over restrictive
LINK SEQUENCE	***ServicePattern, JourneyPattern***	Unnecessarily over restrictive
LINE	***Line, FlexibleLine***	Useful to have common scope
JOURNEY	***ServiceJourney, TemplateServiceJourney***	Useful to have common scope
ORGANISATION	***Operator, Authority***	Useful to have common scope
FACILITY SET	***SiteFacilitySet, ServiceFacilitySet***	Useful to have common scope

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 ‘foo’ may have children that are children uniquely identified as follows:

```
<ServicePattern id=“foo” version=”1.1”>
    <StopPointInSequence id=“foo” version=”1.1” order=”1”>
    </StopPointInSequence>
    <StopPointInSequence id=“foo” version=”1.1” order=”2”>
    </StopPointInSequence>
    <StopPointInSequence id=“foo” version=”1.1” order=”3”>
    </StopPointInSequence>
Etc.

</ServicePattern>
```

In EPIP within an XML document, elements with at an ***order*** attribute shall 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 shall include an ***id*** and a ***version*** attribute in the start tag, even if it is just a dummy value (by convention ‘*any*’)).

5.3 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.

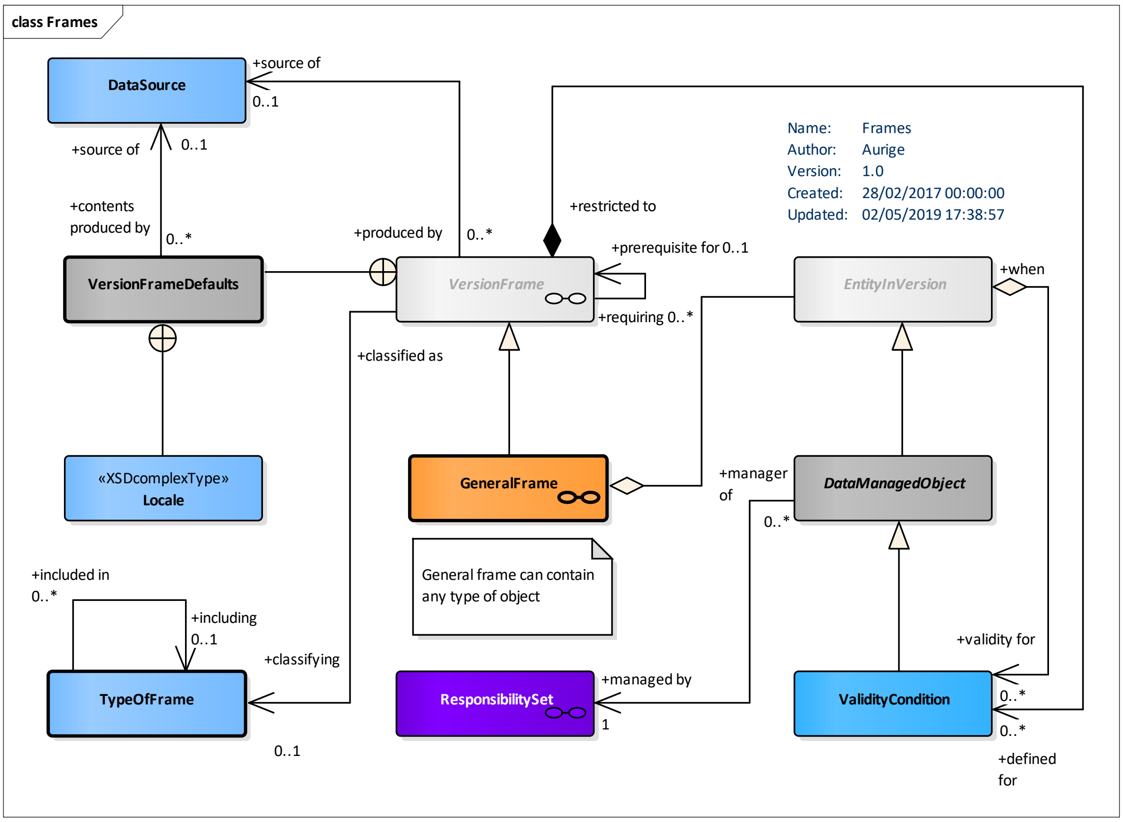

Figure 2 — Version Frames (UML)

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 ***8.1-Version management*** for more details.

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

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

5.3.1 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 13 — *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»	***prerequisites***	*VersionFrameRef*	0:*	References to other version frames on which this frame depends. +v1.1
«cntd»	***contentValidity-Conditions***	*ValidityCondition*	0:*	Shared VALIDITY CONDITIONs used by elements in the FRAME. Note: this is not the dominate FRAME validity condition.

5.3.2 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).

The EPIP defines a number of standardised ***TypeOfFrame*** instances; each one classifies a particular data set to be exchanged under the EPIP and shall be assigned to the appropriate version frame in any EPIP conformant NeTEx document. All frame identifiers are defined within the EPIP ***Codespace*** with a prefix of '*epip:*'.

The list of frame types in EPIP is:

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

The '*EU_PI_STOP', 'EU_PI_NETWORK'*, and '*EU_PI_TIMETABLE'* frame types carry the main "payload" data content for an EPIP exchange. The '*EU_PI_COMMON'*, and '*EU_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. The EPIP specification defines two composite frames with which to group the payload and ancillary frames:

- '***EU_PI_LINE_OFFER.'*** gathering all the information related to a single LINE; it shall contain '*EU_PI_COMMON' 'EU_PI_STOP', 'EU_PI_NETWORK', 'EU_PI_CALENDAR'* and '*EU_PI_TIMETABLE'*. specific frames.
- ***'EU_PI_NETWORK_OFFER'***: gathering all the information related to a single NETWORK (so multiple lines); it shall contain *'EU_PI_COMMON,' 'EU_PI_CALENDAR', 'EU_PI_STOP'*, '*EU_PI_NETWORK'* and '*EU_PI_TIMETABLE'* specific frames There may be multiple instances of a given frame type, for example a separate, , *'EU_PI_NETWORK', 'EU_PI_TIMETABLE'* for each line.
- '***EU_PI_STOP_OFFER.'*** gathering all the information related to a STOPs for area; it shall contain '*EU_PI_COMMON' 'EU_PI_STOP,* specific frames.

A detailed description of these types of VERSION FRAME is provided in 8.10.

One further ***TypeOfFrame*** *'EU_PI_METADATA'* can be included in an EPIP 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 shall be external references that place the version number in a ***versionRef*** element.

Table 14 — *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. Shall be one of the following: *EU_PI_COMMON, EU_PI_STOP, EU_PI_NETWORK, EU_PI_TIMETABLE, EU_PI_LINE_OFFER, EU_PI_NETWORK_OFFER, EU_PI_CALENDAR*
«FK»				
«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. Shall not be cyclic.

5.3.3 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 specific frames only host a limited set of object types, and furthermore group 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.

EPIP will only use the ***ResourceFrame, SiteFrame, ServiceFrame, TimetableFrame*** and ***ServiceCalendarFrame*** specific frames, but for brevity the actual frame descriptions are omitted – see the summary descriptions of the permitted contents of *‘EU_PI_COMMON’ ‘EU_PI_STOP’, ‘EU_PI_NETWORK’, ‘EU_PI_CALENDAR*’, and *‘EU_PI_TIMETABLE’* in 8.10.

In the EPIP, a given type of NeTEx object is only permitted in within one single type of frame.

Table 15 — *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)

5.3.4 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.

The EPIP specification defines two TYPEs OF FRAME for **CompositeFrame**: *'EU_PI_LINE_OFFER' and 'EU_PI_NETWORK_OFFER'*; see the summary descriptions of their permitted contents in 8.10.

Table 16 — *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 *'EU_PI_COMMON', 'EU_PI_STOP,' 'EU_PI_CALENDAR', 'EU_PI_NETWORK' and 'EU_PI_TIMETABLE '*.

5.3.5 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 17 — *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.
«FK»	***DefaultDataSourceRef***	*DataSourceRef*	0:1	DATA SOURCE to use for elements in the frame which do not have a DATA SOURCE specified.
«FK»	***DefaultResponsibility-SetRef***	*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 EPIP WGS84 is recommended.
«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. • *SiMetresAndKilometres* (only value accepted in EPIP)
	DefaultCurrency	*CurrencyType*	0:1	Default CURRENCY to use for amount attributes in the frame for which a currency has not been specified.

5.4 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 of the EPIP describes such elements.

Note that to make the EPIP 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.

5.4.1 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.

In the EPIP, 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 EPIP values can be considered as "profile metadata" inherent to the EPIP and they can be referenced with external references without being explicitly encode in an EPIP 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 *'EU_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 18 — *TypeOfValue* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	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.
	Description	*MultilingualString*	0:1	Description of TYPE OF VALUE.

5.4.2 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, EPIP specifies two types of STOP PLACE '*monomodal*' and '*general*: these could be formally encoded and exchanged along with the data that references them (so that any use of them is validated) as follows:

```
<ValueSet version="1.0" id="epip:Types_of_StopPlace" classOfValues="TypeOfPlace">
    <Name>Types of EPIP Stop Place</Name>
        <values>
            <TypeOfPlace version="1.0" id="epip:general">
                <Name>General stop place</Name>
            </TypeOfPlace>
            <TypeOfPlace version="1.0" id="epip:monomodal">
                <Name>Single mode stop place</Name>
            </TypeOfPlace>
        </values>
</ValueSet>
```

Table 19 — *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

5.4.3 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 EPIP: ***GroupOfStopPlaces***, ***GroupOfServices***, etc (and ***Place*** is in effect a ***GroupOfPoints***).

Table 20 — *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*	0:1	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 shall 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.***

5.4.4 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.

Table 21 — *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 **1:1**	Location of POINT. Mandatory in the EPIP.
«»	***PointNumber***	*xsd:normalizedString*	0:1	Arbitrary alternative identifier for the POINT.
«cntd»	***projections***	*Projection*	0:*	Projections for POINT.

5.4.5 Location (Subcomponent of Point)

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

Table 22 — *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: WGS84 Is highly recommended for the EPIP.
	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.
	Precision	*xsd:decimal*	0:1	Precision of coordinates.

5.4.6 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.

Table 23 — *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»	***LineString***	*gml:LineString*	0:1	GML linestring (detailed geometry) of the LINK.
«FK»	(***FromPointRef***)	*PointRef*	1:1	Reference to point at which LINK starts.
«FK»	(***ToPointRef***)	*PointRef*	1:1	Reference to point at which LINK ends.

5.4.7 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 path such as ***Route***, ***JourneyPattern***, ***ServicePattern***, etc.

Table 24 — *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.
	Distance	*DistanceType*	0:1	Distance of LINK SEQUENCE. Units are as specified for Frame. Default is SI Metres.
«cntd»				

«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***,

5.4.8 PointInLinkSequence (Abstract)

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

Table 25 — *PointInLinkSequence – XML Element (Abstract)*

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	POINT in LINK SEQUENCE inherits from VERSIONED CHILD.
«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.
	Description	*MultilingualString*	0:1	Description of POINT in LINK SEQUENCE. +v1.1

5.4.9 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 26 — *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»	***Centroid***	*Point*	0:1	Centre POINT of ZONE.
«cntd»	***gml:Polygon***	*gml:Polygon*		Polygon associated with zone.

5.4.10 Transport Mode and Submode

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 shall 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.

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

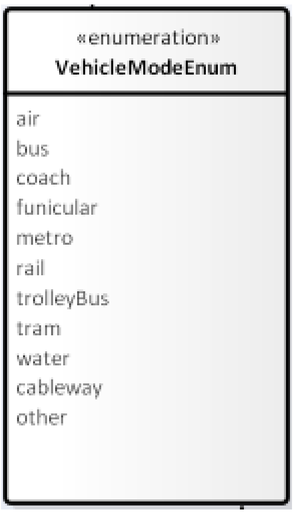

Figure 3 — Transport modes

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.).

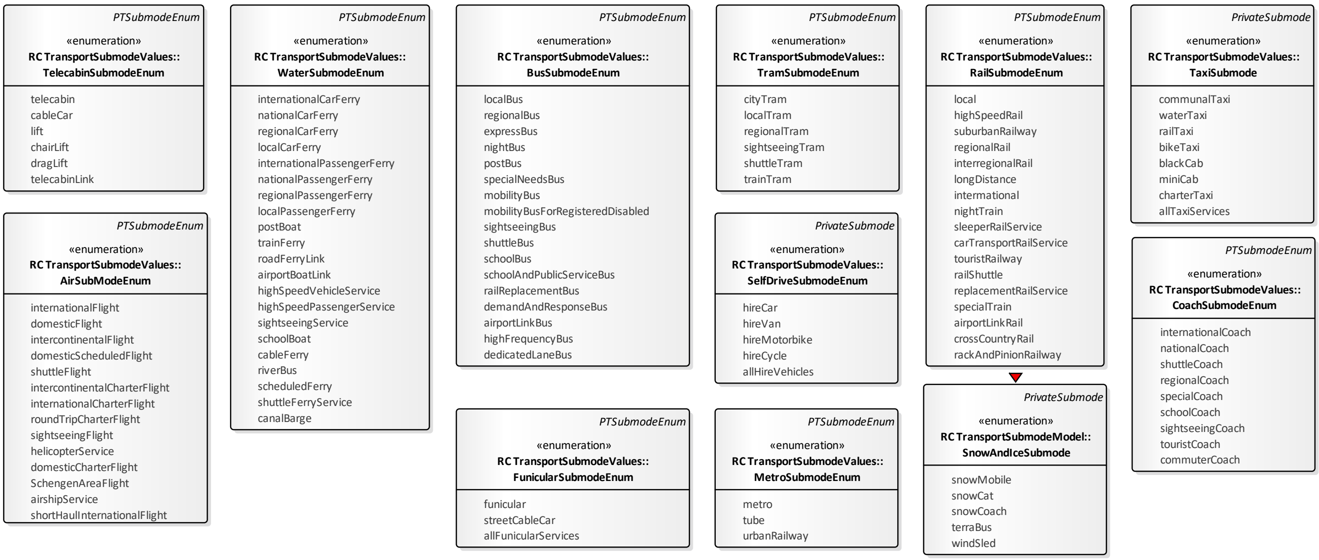

Figure 4 — Transport submodes

5.4.11 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 27 — Table 1 – *Assignment* – XML Element (Abstract)

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	ASSIGNMENT inherits from DATA MANAGED OBJECT.
	Name	*MultilingualString*	0:1	Name of ASSIGNMENT.
	Description	*MultilingualString*	0:1	Description of ASSIGNMENT.

5.4.12 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 28 — *Presentation* – Type

Classification	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 backgropund 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
«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.

5.4.13 Locale (Subcomponent – Multipurpose)

One of the frame defaults is a ***Locale***; specifying parametersand properties to be used when processing data in a country or region (time zone, language, etc.).

Table 29 — *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.
«enum»	***DefaultLanguage***	*xsd:language*	0:1	Default language of LOCALE. (*en, fr, de,* etc)
«enum»	***languages***	*LanguageUsageEnum*	0:*	Other languages supported in LOCALE.

5.4.14 Booking Arrangements (Subcomponent – Multipurpose)

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

Table 30 — *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*
«enum»	***BuyWhen***	*PurchaseMomentEnum*	0:*	When purchase can be made. • *inAdvanceOnlyy* • *onReservation* • *beforeBoarding* • *beforeBoardiingOnly* • *onBoarding*

				- *onBoardingOnly* - *afterBoarding* - *onCheckIn* - *onCheckOut* - *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.

5.4.15 ContactDetails (Subcomponent – Multipurpose)

Details on how to contact an ORGANISATION.

Table 31 — Table 2 — *ContactDetails – Type*

Classification	Name	Type	Cardinality	Description
	Email	*EmailAddressType*	0:1	Email address in ISO format.
	Phone	*PhoneType*	0:1	Phone number.
	Url	*xsd:anyURI*	0:1	Contact web site URL.

5.5 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.

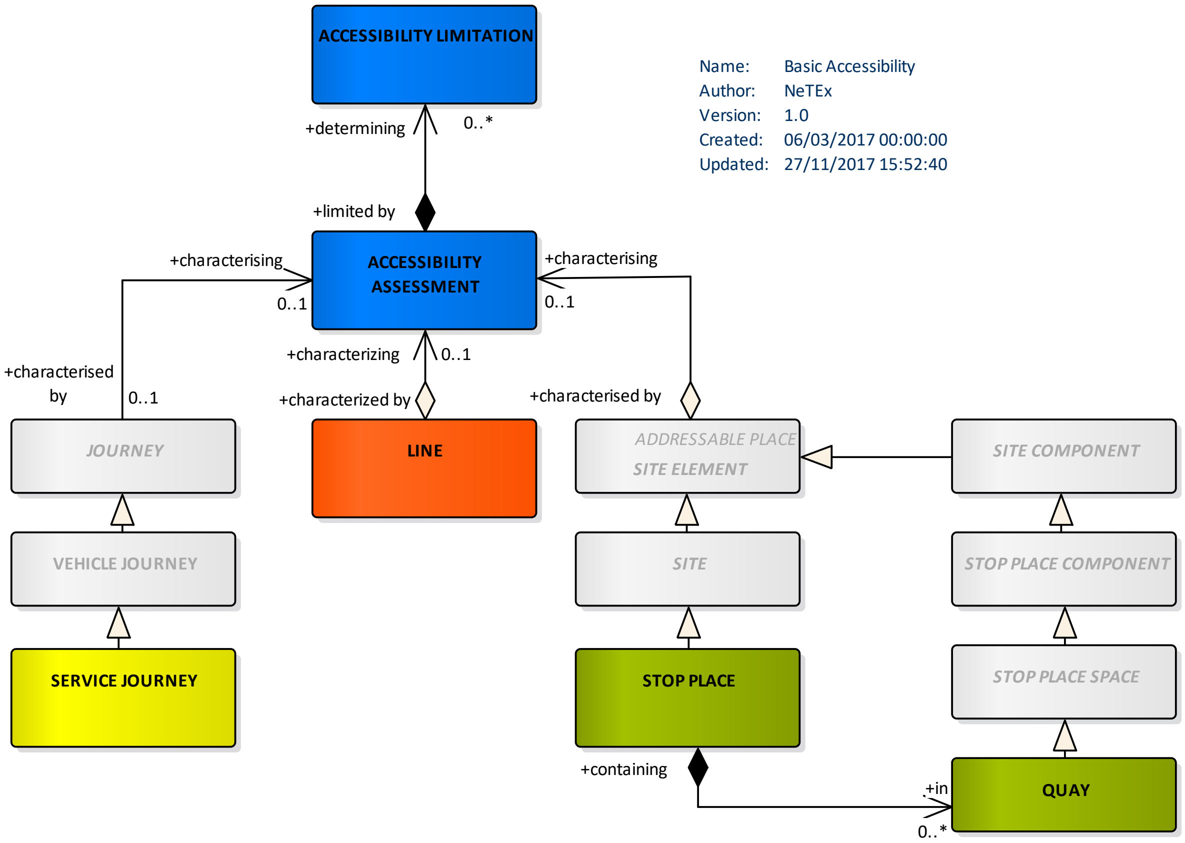

Figure 5 — Accessibility (UML)

An ACCESSIBILITY ASSESSMENT describes the accessibility of a SITE, a LINE or a SERVICE JOURNEY components (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.*

Note that accessibility assessment methods and rules are country dependent and out of the scope of the EPIP: the exchanged information will be based on the local assessment methods and rules of the country of the provider.

5.5.1 AccessibilityAssessment (Subcomponent – Multipurpose)

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

Table 32 — *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.

5.5.2 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 33 — *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 only in case of disruption, or for a specific class of service.

In the context of EPIP, whenever a "*partial*" ***LimitationStatus*** is used, then a "***ValidityCondition->Description***" shall be provided accordingly 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.

5.6 Facility sets

Passenger Information can include advice as to the facilities available at a stop or onboard a service.

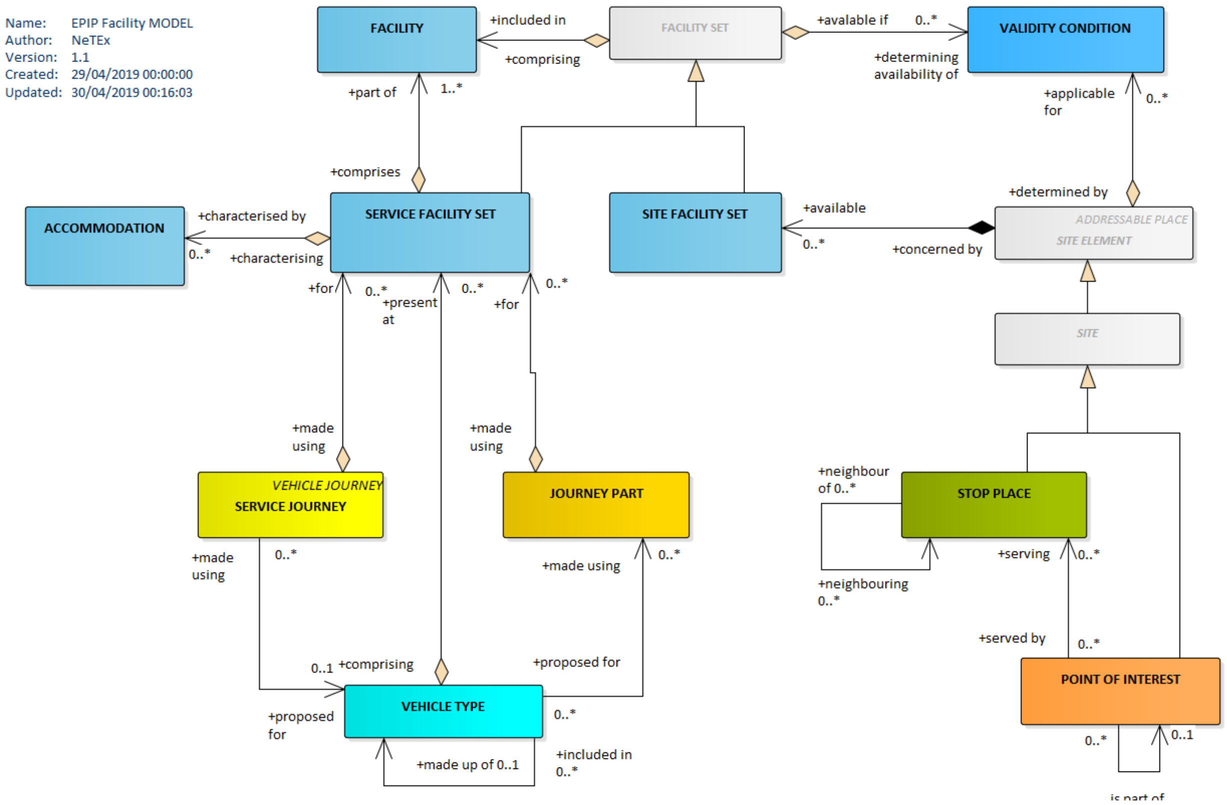

Figure 6 — Facility Sets (UML)

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 codes is described in an annex 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.

For EPIP, SITE FACILITY SETs can only be associated with a ***StopPlace;*** SERVICE FACILITY SETs can only be used for a ***ServiceJourney*** and a ***JourneyPart***.

5.6.1 FacilitySet (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 34 — *FacilitySet – XML Element (Abstract)*

Classification	Name	Type	Cardinality	Description
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 theSITE FACILITY SET.

5.6.2 ServiceFacilitySet

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

Table 35 — *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 hoices***	0:1	Facilities specific to SERVICE FACILITY SET. See Annex.
«cntd»	***accommodations***	*Accommodation*	0*	ACCOMMODATIONs in SERVICE FACILITY SET.

5.6.3 Accommodation – (Subcomponent of ServiceFacilitySet)

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

Table 36 — *Accommodation – XML Element*

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

«PK»	***id***	*AccomodationIdType*	1:1	Identifier of ACCOMMODATION.
«enum»	***FareClass***	*FareClassEnum*	0:1	FARE CLASS of ACCOMMODATION. See Annex for allowed values.
«enum»	***Accommodation-Facility***	*Accommodation-FacilityEnum*	0:1	Type of accommodation Facility in ACCOMMODATION. See Annex for allowed values.
«enum»	***NuisanceFacility***	*NuisanceFacilityEnum*	0:1	Nuisance Facility in ACCOMMODATION. See Annex B.
«enum»	***Passenger-CommsFacility***	*PassengersComms-FacilityEnum*	0:*	Communications facilities within the accomodation section See Annex B.

5.6.4 SiteFacilitySet

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

Table 37 — *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.

6 Stop Profile

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

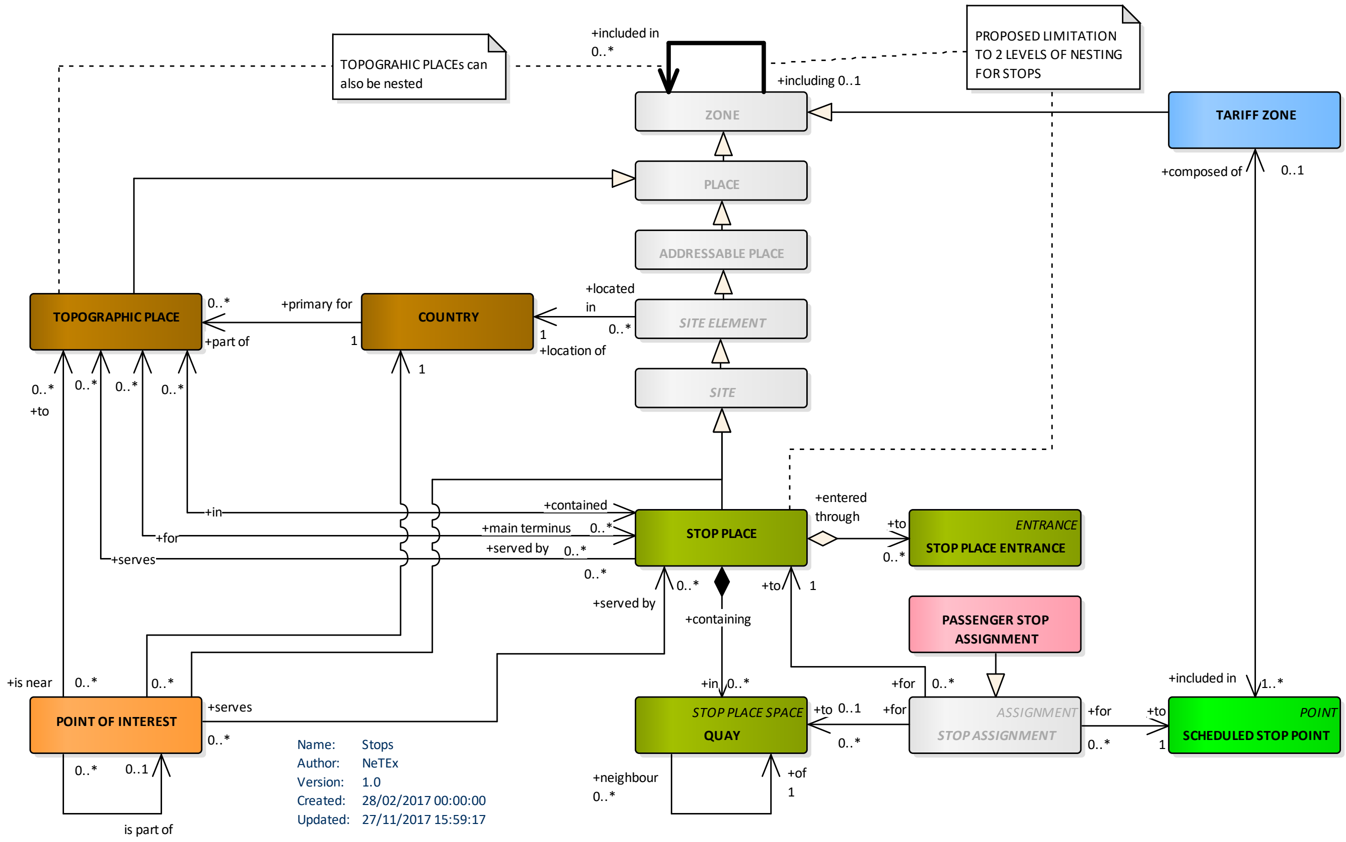

Figure 7 — Stop Places

For EPIP 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.

The top object in the inheritance tree 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).

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.

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.

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).

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 ***7.4.1-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 so 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.

6.1 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, having an unlimited number of hierarchy levels, especially without a clear semantic for each level, may lead to interoperability issues (since many legacy systems only support a limited degree of nesting). Therefore EPIP constrains the use of the NeTEx stop model to a mode aware structure requiring the following properties:

- 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 a 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:

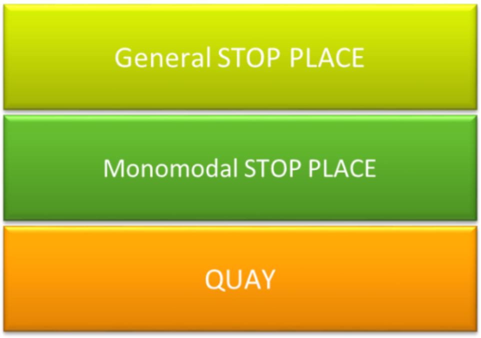

Figure 8 — Stop Place nesting hierarchy

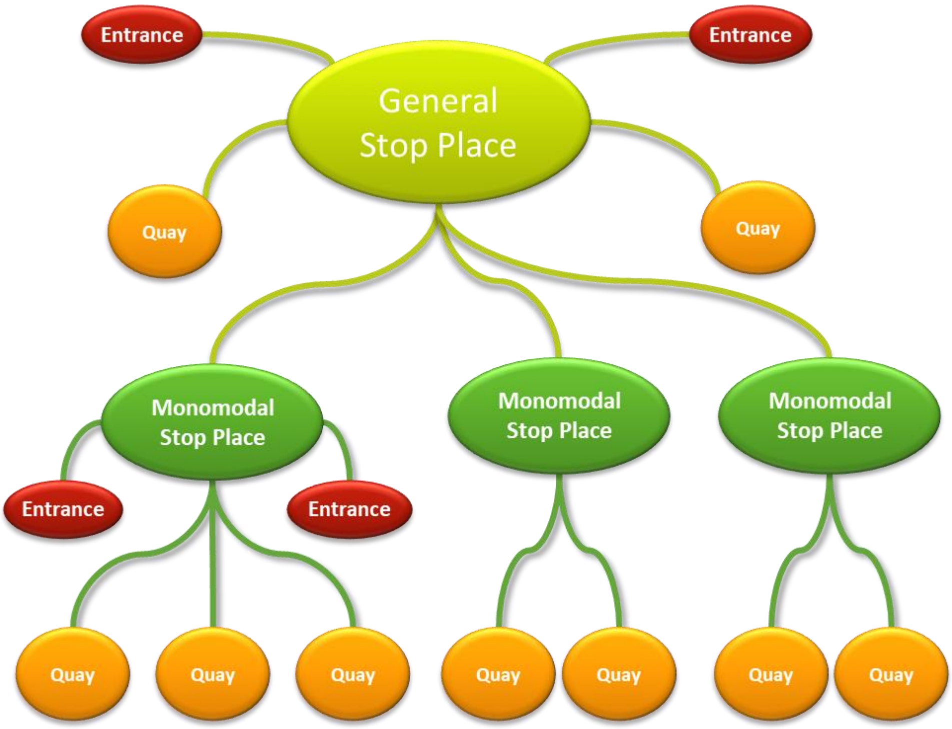

Figure 9 — Stop Place nesting example

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

Table 38 — *Types of QUAYs and Mode compatibility*

Type of QUAY	Other compatible QUAY types	Possible transport mode
railPlatform	none	*rail, intercityRail, urbanRail*
metroPlatform	none	*metro, funicular*
tramPlatform	*tramStop*	*tram*
tramStop	*tramPlatform*	*tram*
busStop	*busBay*	*bus, coach, trolleyBus*
busBay	*busStop*	*bus, coach, trolleyBus*

boatQuay	*ferryLanding*	*water*
ferryLanding	*boatQuay*	*water*
telecabinePlatform	none	*cableway*
airlineGate	none	*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.

6.2 Other types of Place

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 UK 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.

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

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 through the EPIP, as requested by the EC's *ITS Directive (priority action A)*. However, the actual POINTs OF INTEREST provided will be limited to those connected to STOP PLACEs.

6.2.1 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 39 — *Place – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*Zone*	::>	PLACE inherits from ZONE.
«cntd»	***placeTypes***	*TypeOfPlaceRef*	0:*	Categorisation of PLACE. In EPIP (see *Stop* Place), STOP PLACEs shall be assigned one of the following two values: • 'epip:*monomodal*' • 'epip:*general*' For example: `<TypeOfPlaceRef versionRef="epip:1.0" ref="epip:monomodal_stop_place"/>`

6.2.2 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 40 — *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.

6.2.3 SiteElement (Abstract)

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

Table 41 — *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*
«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 42 — *SiteElementPropertiesGroup* – Group

Classification	Name	Type	Cardinality	Description
«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*

6.2.4 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 43 — *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»				
«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»	***levels***	*Level*	0:*	LEVELs for SITE.
«cntd»	***entrances***	*Entrance*	0:*	ENTRANCEs to SITE.

6.2.5 Address (Abstract)

Common attributes of an ADDRESS.

Table 44 — *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.

6.2.6 PostalAddress (Subcomponent of SITE in EPIP)

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.

Table 45 — *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	*xsd:normalizedString*	0:1	Building name of POSTAL ADDRESS.
	AddressLine1	*xsd:normalizedString*	0:1	First line of POSTAL ADDRESS.
	AddressLine2	*xsd:normalizedString*	0:1	Second line of POSTAL ADDRESS.
	Street	*xsd:normalizedString*	0:1	Street name of POSTAL ADDRESS.
	Town	*xsd:normalizedString*	0:1	Town of POSTAL ADDRESS.
	Suburb	*xsd:normalizedString*	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	*xsd:normalizedString*	0:1	Postal Province.
«FK»	***RoadAddressRef***	*RoadAddressRef*	0:1	ROAD ADDRESS associated with POSTAL ADDRESS.

6.2.7 RoadAddress (Subcomponent of SITE in EPIP)

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.

Table 46 — *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	*xsd:normalizedString*	0:1	Name of ROAD.
«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.

6.2.8 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 47 — *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.

	Descriptor		1:1	Descriptive elements for a TOPOGRAPHIC PLACE. Only the **Name** will be filled for the EPIP.
«enum»	***TopographicPlace-Type***	*TopographicTypeEnum*	0:1	Type of a TOPOGRAPHIC PLACE.
	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.
	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.
	containedIn	*TopographicPlaceRef*	0:*	References to other TOPOGRAPHIC PLACEs within which the TOPOGRAPHIC PLACE is contained.

6.2.9 TopographicPlaceDescriptor (Subcomponent)

A TOPOGRAPHICAL PLACE shall 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. “Newport, Gwent”, “Newport, Salop”, 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 “Newport, Gwent” looking up stop departures would simply “Newport”, (since the context is already established) , while a user looking for “Newport” in a Journey Planner for the whole of Britain (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 48 — *TopographicPlaceDescriptor – XML Element*

Classification	Name	Type	Cardinality	Description

::>	::>	<u>*VersionedChild*</u>	::>	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.
	QualifierName	*MultilingualString*	0:1	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***.

6.2.10 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 POIs 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.

Table 49 — *PointOfInterest – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*Site*</u>	::>	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»	***nearTopographic-Places***	*TopographicPlaceRef*	0:*	TOPOGRAPHIC PLACEs near the POINT OF INTEREST.

6.3 Stop Places

6.3.1 StopPlace

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

Table 50 — *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* • *taxi* • *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* • *busStation*

				- *airport* - *railStation* - *metroStation* - *coachStation* - *ferryPort* - *harbourPort* - *ferryStop* - *liftStation* - *tramStation* - *vehicleRailInterchange* - *taxiStand* - *other*
	BorderCrossing	*xsd:boolean*	0:1	Whether STOP PLACE is a border crossing.
STOP PLACE TOPOGRAPHIC GROUP				
«enum»	***Weighting***	*InterchangeUseEnum*	0:1	Default relative weighting to be used for stop place. Possible values are: - *noInterchange* - *interchangeAllowed* - *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				
	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.

6.3.2 SiteComponent (Abstract)

A SITE COMPONENTs is an 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.

Table 51 — *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»	***LevelRef***	*LevelRef*	0:1	Reference to a LEVEL associated with the SITE COMPONENT.

6.3.3 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. None of its properties except ***Label*** are used in EPIP.

Table 52 — *StopPlaceSpace – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*SiteComponent*	::>	STOP PLACE SPACE inherits from SITE COMPONENT.
	Label	*xsd:normalizedString*	0:1	Local Label given to Component, e.g. a Point Letter on a stop.

6.3.4 Quay (Subcomponent of Quay)

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 shall 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 53 — *Quay – XML Element*

Classification	Name	Type	Cardinality	Description

::>	::>	<u>*StopPlaceSpace*</u>	::>	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.
	CompassBearing	*CompassBearingType*	0:1	Bearing of street relative to QUAY in degrees (absolute bearing of front part of Vehicle when stopped at Quay).
«enum»	***QuayType***	*QuayTypeEnum*	0:1	Classification of QUAY.
«FK»	***ParentQuayRef***	*QuayRef*	0:1	Reference to parent of QUAY that wholly contains it.

6.3.5 Entrance (Abstract)

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

Table 54 — *Entrance – XML Element* (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*SiteComponent*</u>	::>	ENTRANCE inherits from SITE COMPONENT.
«PK»	***id***	*EntranceIdType*	1:1	Identifier of ENTRANCE.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public code for SITE 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.

6.3.6 StopPlaceEntrance

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.

Table 55 — *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.

6.4 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 56 — *GroupOfStopPlaces – XML Element*

Classification	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.

6.5 TariffZone

A ZONE used to define a zonal fare structure.

The TARIFF ZONE to members SCHEDULED STOP POINT relation 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 EPIP the reference should only be on the ***ScheduledStopPoint***.

Table 57 — Table 3 — *TariffZone – XML Element*

Classificat ion	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

7 Network and Timetables

The Network and Timetable part of EPIP (EPIP-NT) specifies the data element 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.

7.1 Basic network topology

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

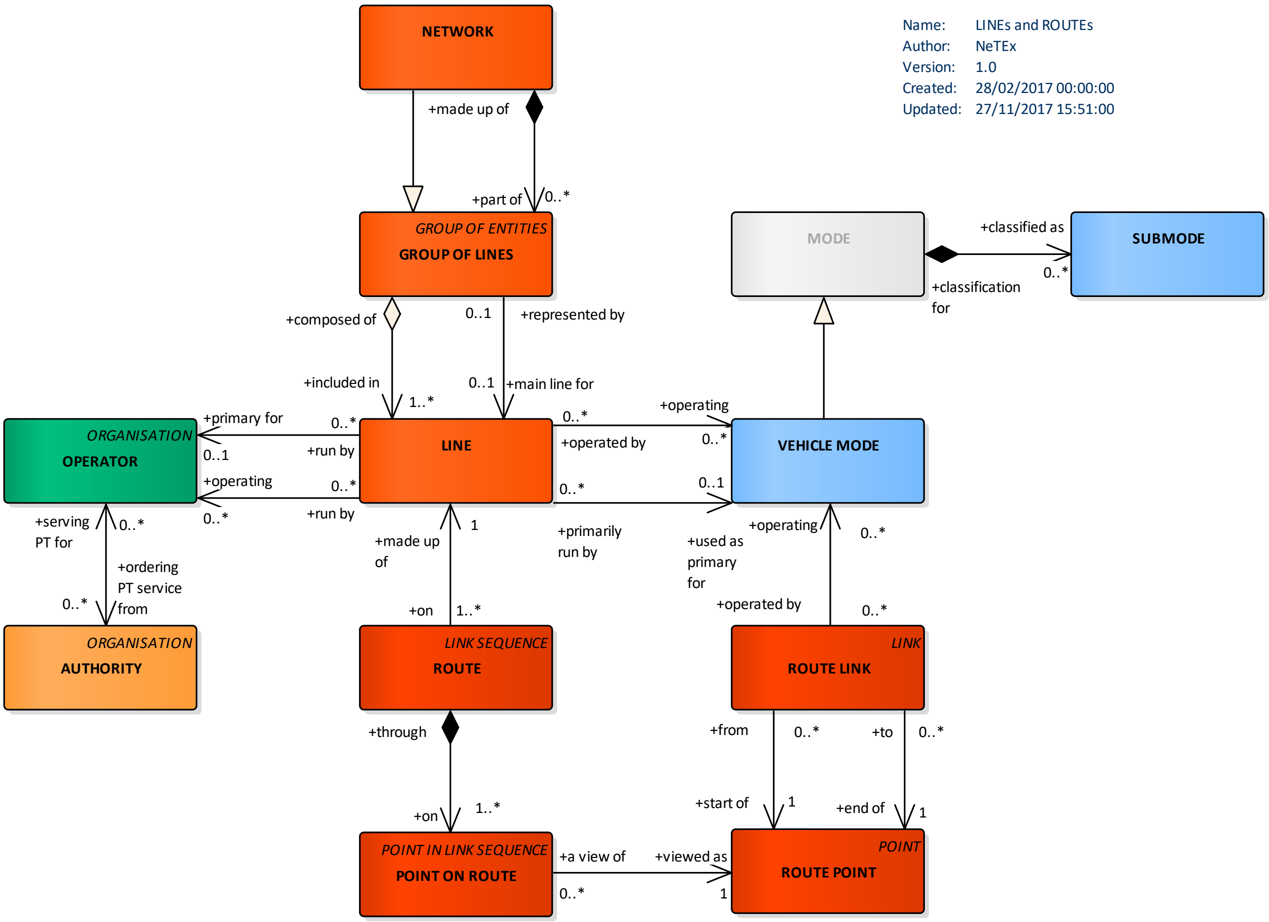

Figure 10 — Network Topology (UML)

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 (see ***5.4.10-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.).

7.1.1 Line

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

Table 58 — *Line – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*DataManagedObject*</u>	::>	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.
	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.

	TypeOfLineRef	*TypeOfLineRef*	0:1	Reference to a TYPE of LINE.
	Monitored	*xsd:boolean*	0:1	Whether real-time data is normally available for LINE.
«cntd»	***routes***	*RouteRef*	0:*	Routes that follow the LINE.
«cntd»	***Presentation***	*Presentation*	0:1	Preferred presentation attributes to use when rendering LINE on maps, etc.
«cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	The overall accessibility characteristics of the LINE.
	allowedDirections			
«cntd»	***noticeAssignments***	*NoticeAssignment*	0:*	NOTICES assigned to LINE.
«cntd»	***documentLinks***	*InfoLinks*	0:*	Timetable documents associated with the LINE, e.g pdf files +v1.1

7.1.2 AllowedLineDirection (Subcomponent of Line)

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

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

Table 59 — Table 4 – *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»	***DirectionRef***	*DirectionRef*	1:1	Reference to a DIRECTION of ALLOWED LINE DIRECTION.

7.1.3 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 all Île-de-France network), but also to the *Noctilien* network (night buses) ant to the *PHEBUS* network (Versailles's town bus network).

Table 60 — *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.

7.1.4 Network

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

Table 61 — *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»	***groupsOfLines***	*GroupOfLines*	0:*	Groups of LINEs in NETWORK. Can be used to partition or classify the lines in a Network.

7.1.5 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. It may have one or more associated VIAs.

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

Table 62 — *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.
	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.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public code (to be presented to the public) associated with DESTINATION DISPLAY.
«cntd»	***vias***	*Via*	0:*	Text to show for VIA display.
«cntd»	***variants***	*DestinationDisplayVariant*	0:*	Variant Display texts for different media.

7.1.6 Via

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

Table 63 — *Via – XML Element*

Classification	Name		Type	Cardinality	Description
::>	*::>*		*VersionedChild*	*::>*	VIA inherits from VERSIONED CHILD.
	id		*ViaIdType*	1:1	Identifier of VIA.
	b	***Name***	*MultiLingualString*	1:1	Name of VIA.

7.1.7 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 64 — *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.
				DESTINATION DISPLAY VARIANTs shall be embedded in line with a parent DESTINATION DISPLAY
«enum	***DestinationDisplayVariantMediaType***	*DeliveryVariantTypeEnum*	1:1	Type of DESTINATION DISPLAY VARIANT being: • *printed* • *textToSpeech* • *recordedAnnouncement*

				- *web* - *mobile* - *other*
	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.
«cntd»	***vias***	*Via*	0:*	Text to show for VIA display.

7.1.8 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 EPIP as the supertype of OPERATOR and AUTHORITY.

Table 65 — *Organisation – XML Element (abstract)*

Classification	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*DataManagedObject*	::>	ORGANISATION inherits from DATA MANAGED OBJECT.
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Alternative public identifier of ORGANISATION.
«AK»	***CompanyNumber***	*xsd:normalizedString*	0:1	Registration number of ORGANISATION as a legally incorporated body.
	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			NOT USED IN EPIP Use Brand name (instead of TradingName) in the EPIP
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for ORGANISATION. Dedicated to aliases (where AlternativeText is more focused on translations).
	Description	*MultilingualString*	0:1	Further description of ORGANISATION.

«cntd»	***ContactDetails***	*ContactDetails*	1:1	Contact details for ORGANISATION for public use.
«FK»	***OrganisationType***	*TypeOfOrganisationEnum*	1:1	Type of ORGANISATION. - *authority* - *operator* - *railOperator* - *railFreightOperator* - *statutoryBody* - *facilityOperator* - *travelAgent* - *servicedOrganisation* - *other*

7.1.9 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.

Table 66 — *Operator – XML Element*

Classificat ion	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.

7.1.10 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 67 — *Authority – XML Element*

Classificat ion	Name	Type	Cardinalit y	Description

::>	::>	*Organisation*	::>	AUTHORITY inherits from ORGANISATION.
«PK»	***id***	*AuthorityIdType*	1:1	Identifier of AUTHORITY.
«cntd»	***Address***	*PostalAddress*	1:1	Postal ADDRESS of ORGANISATION.

7.2 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 shall 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 shall 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).

7.2.1 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 68 — *RoutePoint – XML Element*

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

7.2.2 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.

Table 69 — *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.

7.2.3 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 70 — *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.

7.2.4 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 71 — *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***	*LinkSequenceRef*	0:1	Reference to a LINK SEQUENCE
«FK»	***RoutePointRef***	*RoutePointRef*	1:1	Reference to ROUTE POINT that POINT ON ROUTE uses.

7.3 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.

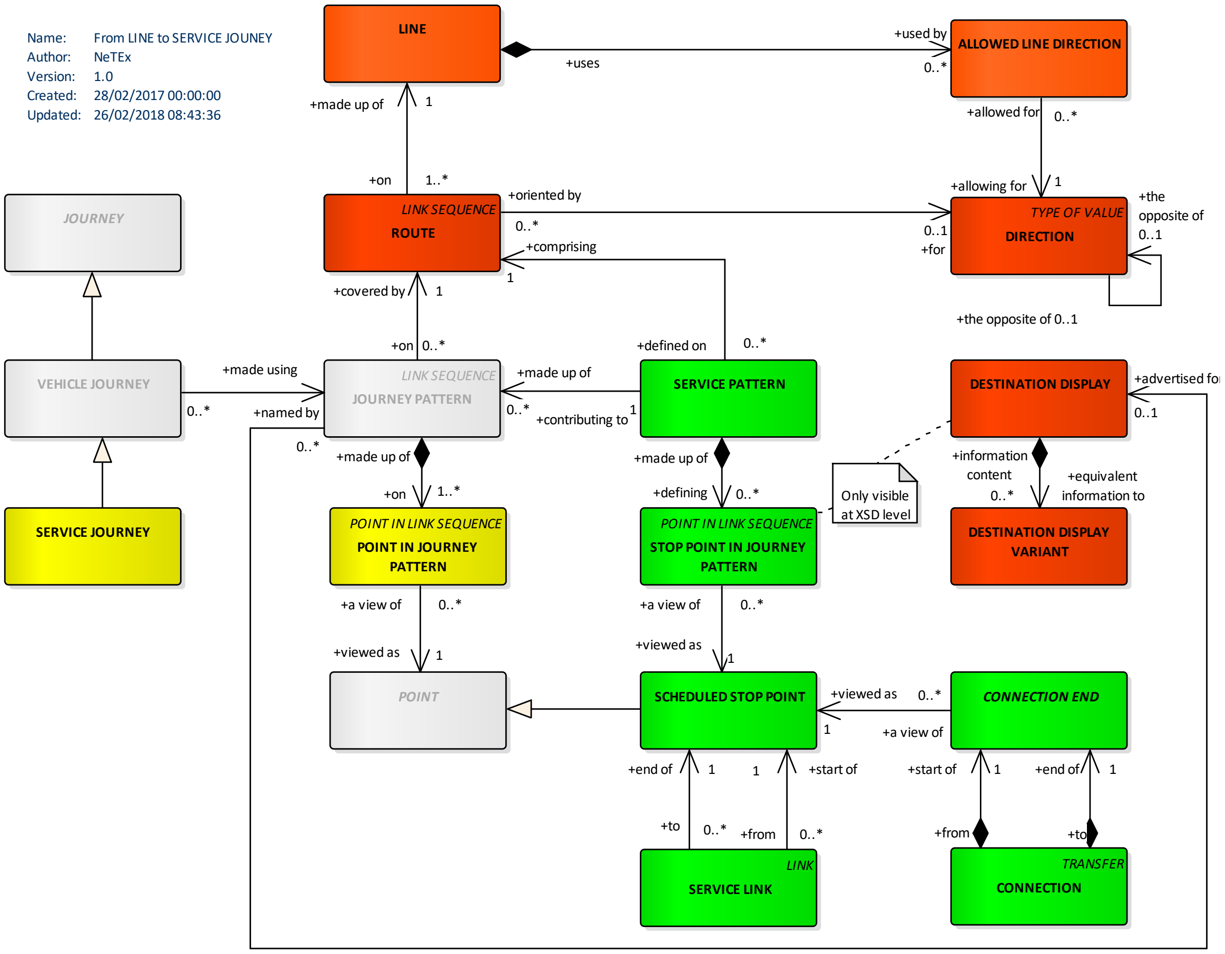

Figure 11 — Service description (UML)

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.

7.3.1 JourneyPattern (Abstract in EPIP)

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. In the EPIP 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.

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 72 — *JourneyPattern – XML Element* (Abstract in EPIP)

Classification	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	JOURNEY PATTERN inherits from LINK SEQUENCE.
«FK»	***RouteRef***	*RouteRef* | *RouteView*	0:1	Reference to ROUTE that JOURNEY PATTERN follows.

				Note that if the ROUTE is not available, then the this ***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).
«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.
«cntd»	***pointsInSequence***	*PointInJourneyPattern*	0:*	POINTs building the JOURNEY PATTERN.

7.3.2 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 73 — *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.

7.3.3 PointinJourneyPattern (Abstract in EPIP)

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

Table 74 — *PointInJourneyPattern – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*PointInLinkSequence*	*::>*	POINT IN JOURNEY PATTERN inherits from POINT IN LINK SEQUENCE.
«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»	***DestinationDisplayRef***	*DestinationDisplayRef*	0:1	DESTINATION DISPLAY associated with POINT IN JOURNEY PATTERN.
«cntd»	***FlexiblePointProperties***	*FlexiblePointProperties*	0:1	Flexible properties of a POINT.
	ChangeOfDestination-Display	*xsd:boolean*	0:1	Whether DESTINATION DISPLAY changes at this point.

7.3.4 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 75 — *StopPointInJourneyPattern – XML Element*

Classification	Name	Type	Cardinality	Description
::>	*::>*	*PointInLinkSequence*	*::>*	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***).
«cnd»	***FlexiblePoint-Properties***	*FlexiblePointProperties*	0:1	Flexible properties of a POINT.
	ChangeOfDestination Display	*xsd:boolean*	0:1	Whether DESTINATION DISPLAY changes 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.

7.3.5 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 76 — *Direction – XML Element*

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

7.3.6 TimingPoint (Abstract in EPIP)

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 conformant document.

Table 77 — *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: deafult is 'timingPoint'. • *timingPoint* • *secondaryTimingPoint* • *notTimingPoint.*

7.3.7 ScheduledStopPoint

A POINT where passengers can board or alight from vehicles.

Table 78 — *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»	***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.
«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.
«enum»	***StopType***	*StopPlaceTypeEnum*	0:1	Type of STOP PLACE. Restricted to an allowed value. • *onstreetBus* • *onstreetTram* • *airport* • *railStation* • *metroStation* • *busStation* • *coachStation* • *tramStation* • *harbourPort* • *ferryPort* • *ferryStop* • *liftStation* • *vehicleRailInterchange* • *other*
	Presentation	*Presentation*	0:1	Presentation values to use when rendering STOP POINT such as a colour and font.

7.3.8 Service Link

A LINK between an ordered pair of SCHEDULED STOP POINTs.

Table 79 — *ServiceLink – XML Element*

Classification	Name	Type	Cardinality	Description
::>	::>	*Link*	::>	SERVICE LINK inherits from LINK. In the EPIP, 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.

7.4 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.

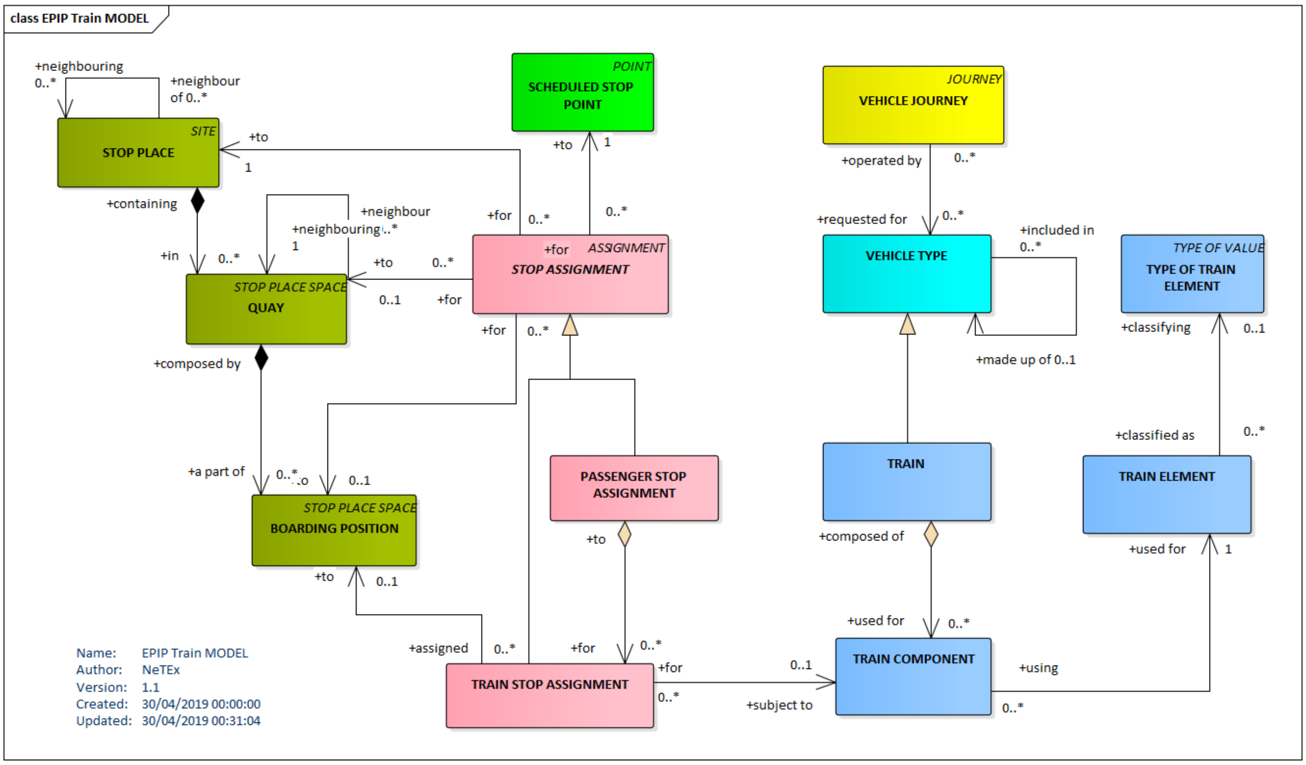

Figure 12 — Stop Assignments (UML)

7.4.1 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 80 — *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.
«FK»	***ScheduledStopPoint-Ref***	*ScheduledStopPointRef*	1:1	Reference to a SCHEDULED STOP POINT to be assigned.

7.4.2 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 (NeTEx also accepts BOARDING POSITION, but they are not used in EPIP).

Table 81 — *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.
«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.

7.4.3 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 (BOARDING POSITIONs are not used in EPIP) within a STOP PLACE.

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

Table 82 — *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»	***PassengerStop-AssignmentRef***	*PassengerStopAssignment-Ref*	0:1	PASSENGER STOP ASSIGNMENT that DYNAMIC STOP ASSIGNMENT overrides.

7.4.4 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 (BOARDING POSITIONs are not used in EPIP). 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.

Table 83 — *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»	***ScheduledStopPoint-Ref***	*ScheduledStopPointRef*	0:1	Reference to a SCHEDULED STOP POINT to be assigned.
«FK»	***PassengerStop-AssignmentRef***	*PasssengerStop-AssignmentRef*	0:1	PASSENGER STOP ASSIGNMENT that this is part of

«FK»	***TrainRef***	*TrainRef*	0:1	TRAIN to which TRAIN STOP ASSIGNMENT applies.
	PositionOfTrain-Element	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.
	EntranceToVehicle	*MultilingualString*	0:1	A specific ENTRANCE to the VEHICLE. E.g. Front, rear.

7.5 Connections, transfers and interchanging

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.

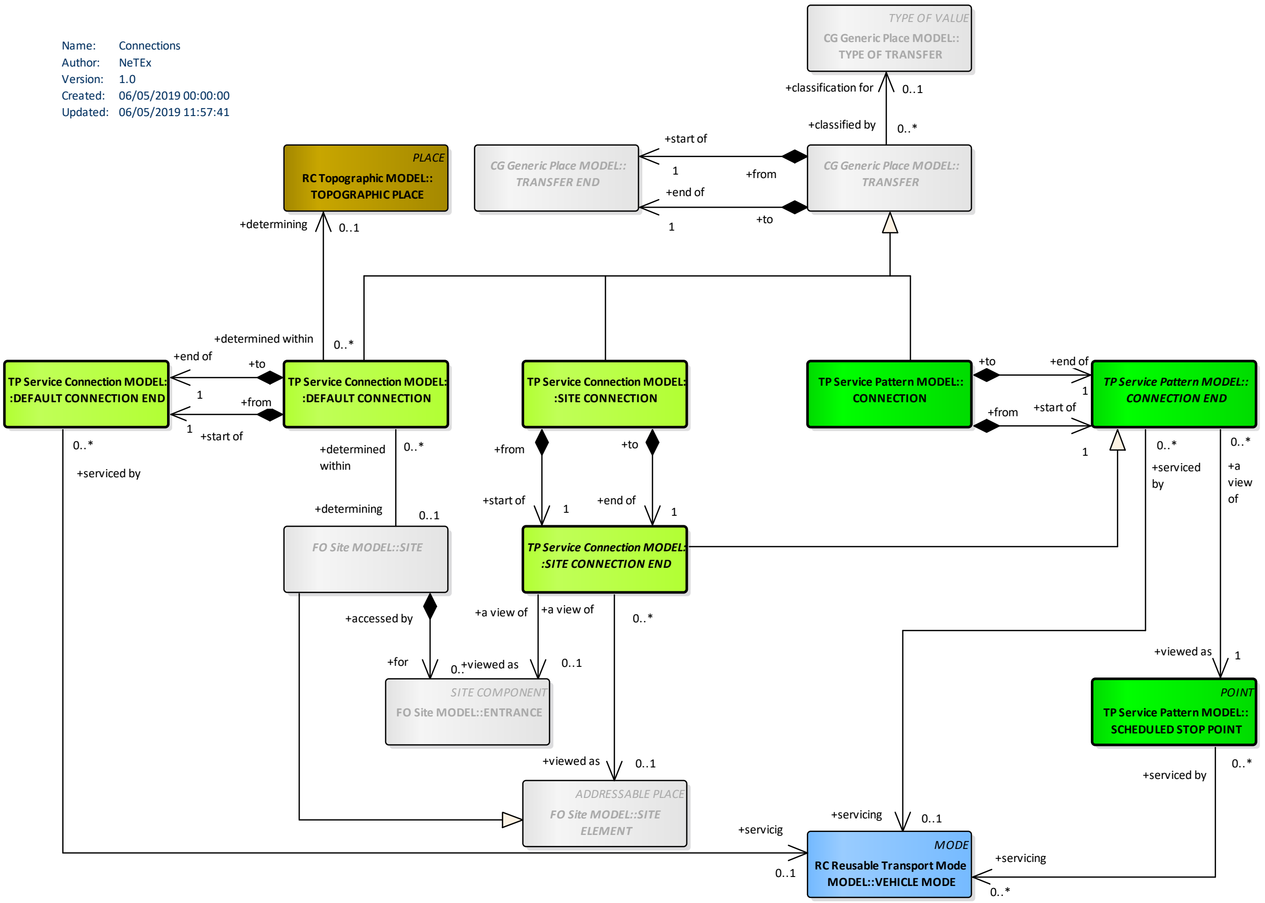
Name: Connections
Author: NeTEx
Version: 1.0
Created: 06/05/2019 00:00:00
Updated: 06/05/2019 11:57:41
TYPE OF VALUE
CG Generic Place MODEL:: TYPE OF TRANSFER
+classification for
0..1
+classified by
0..*
PLACE
RC Topographic MODEL:: TOPOGRAPHIC PLACE
CG Generic Place MODEL:: TRANSFER END
+start of
1
+from
+end of
1
+to
CG Generic Place MODEL:: TRANSFER
+determining
0..1
+determined within
0..*
+end of
+to
1
+from
1
+start of
TP Service Connection MODEL: :DEFAULT CONNECTION END
TP Service Connection MODEL: :DEFAULT CONNECTION
TP Service Connection MODEL: :SITE CONNECTION
TP Service Pattern MODEL:: CONNECTION
+to
+end of
1
+from
+start of
1
TP Service Pattern MODEL:: CONNECTION END
0..*
+serviced by
+determined within
0..*
+determining
0..1
+from
+to
+start of
1
+end of
1
0..*
+serviced by
0..*
+a view of
FO Site MODEL::SITE
TP Service Connection MODEL: :SITE CONNECTION END
+accessed by
+a view of
+a view of
0..*
+for
0..
+viewed as
0..1
SITE COMPONENT
FO Site MODEL::ENTRANCE
+viewed as
1
POINT
TP Service Pattern MODEL:: SCHEDULED STOP POINT
+viewed as
0..1
+serviced by
0..*
ADDRESSABLE PLACE
FO Site MODEL::SITE ELEMENT
+servicing
0..1
+servicig
0..1
MODE
RC Reusable Transport Mode MODEL::VEHICLE MODE
+servicing
0..*

Figure 13 — Overview of NeTEx connections for EPIP (UML)

7.5.1 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 84 — *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»	***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

7.5.2 Transfer Duration (Subcomponent of Transfer)

Specifies transfer times for different sorts of user.

Table 85 — *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.

7.5.3 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 86 — *Connection – XML Element*

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

7.5.4 ConnectionEnd (Subcomponent of Connection)

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

Table 87 — *ConnectionEnd – XML Element*

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

7.5.5 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 EPIP) to another.

Table 88 — *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.

7.5.6 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 ***8.4-Codespaces and the structure of identifiers*** and ***8.6-Dependencies between frames*** for ID structure and external references).

Table 89 — *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»					
	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.

XML EXAMPLE

The following example illustrates use of references to external POIs, 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 POI if it is known).

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


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

7.5.7 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 90 — *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»	***SiteElementRef***	*SiteElementRef*	0:*	SITE ELEMENT to which DEFAULT CONNECTION applies.

7.5.8 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.

7.6 Vehicle journeys and passing times

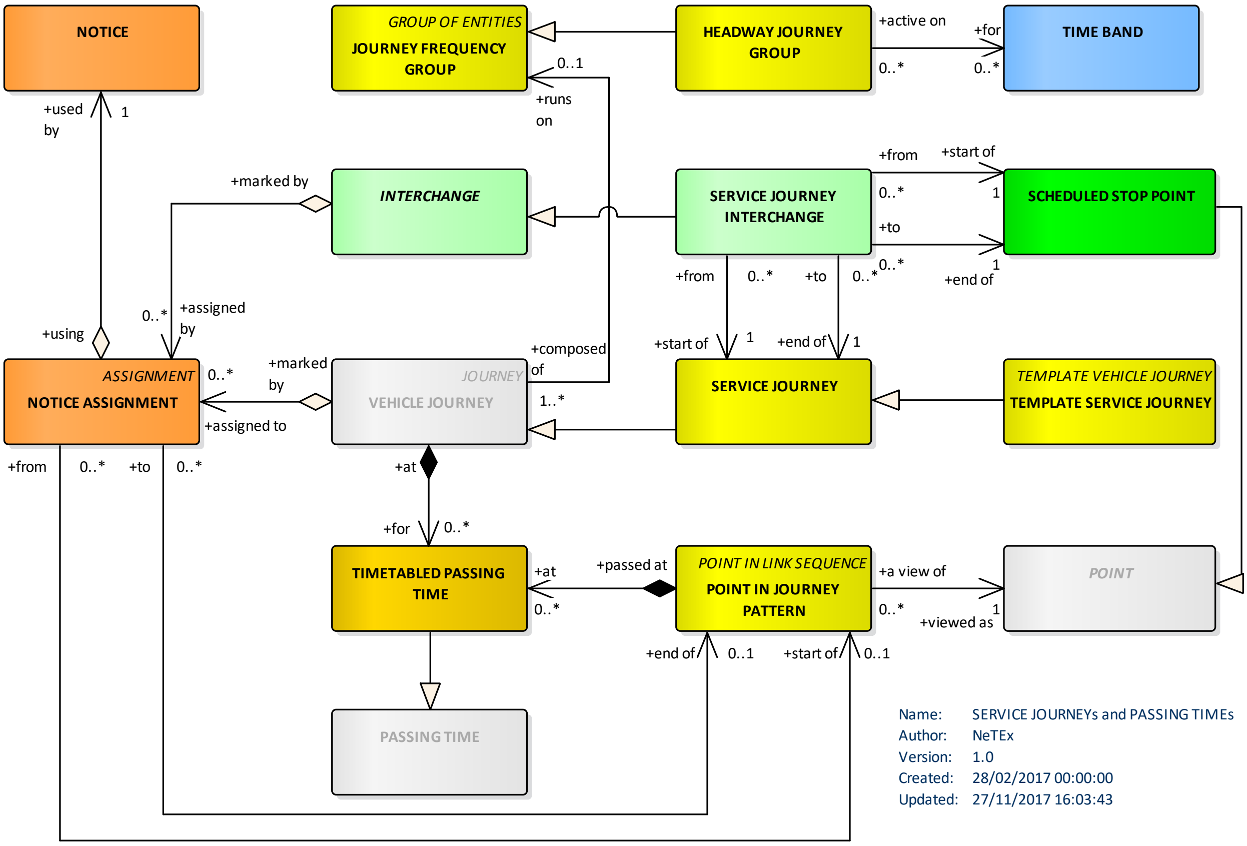

Figure 14 — Journeys and Passing Times (UML)

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; the EPIP is 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 pecific 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.

7.6.1 Journey (Abstract)

The ***Journey*** element holds common properties for any Jurney.

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

Table 91 — *Journey* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	JOURNEY inherits from LINK SEQUENCE.
«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.
«cntd»	***noticeAssignments***	*NoticeAssignment*	0:1	NOTICEs relevant for the whole JOURNEY.

7.6.2 ServiceJourney

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

Table 92 — 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.
	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»					
	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»	***JourneyPatternRef***		*JourneyPatternRef*	1:1	Reference to a JOURNEY PATTERN.
«FK»	***VehicleTypeRef***		*VehicleTypeRef*	0:1	Reference to TYPE OF VEHICLE to use on SERVICE JOURNEY.
«FK»	***a***	***OperatorRef***	*OperatorView*	0:1	Reference to an OPERATOR.
«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.
«cntd»	***trainNumbers***		*TrainNumberRef*	0:*	TRAIN NUMBERs - derived through JOURNEY PARTs of a journey - for a multi-part journey only.
«cntd»	***passingTimes***		*TimetabledPassingTime*	1:*	PASSING TIMEs for VEHICLE JOURNEY.
«cntd»	***parts***		*JourneyPart*	0:*	JOURNEY PARTs of a journey - for a multi-part journey only.
«cntd»	***facilities***		*ServiceFacilitySet*	0:*	Facilities on SERVICE JOURNEY.
	TrainSize		*TrainSize*	0:1	Requirements for TRAIN SIZE.
«cntd»	***b***	***FlexibleServiceProperties***	*FlexibleServiceProperties*	0:1	Flexible properties of SERVICE JOURNEY.

7.6.3 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, 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.

7.6.4 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 93 — *PassingTime* – XML Element (Abstract)

Classification	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	PASSING TIME inherits from VERSIONED CHILD.
«FK»	***PointInJourney-PatternRef***	*PointInLinkSequenceRef*	1:1	POINT IN SEQUENCE for which this is a PASSING TIME. Refer to the SCHEDULE STOP POINT

7.6.5 TimetabledPassingTime (Subcomponent – Service Journey)

Table 94 — *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.

«cntd»	***Headway***	*HeadwayInterval*	0:1	Headway interval at POINT IN PATTERN.

7.6.6 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 95 — *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.
	origin	*EndPoint_DerivedView*		View of derived data about origin of journey. E.g. ***ScheduledStopPointRef, Name, StopType, etc***
	destination	*EndPoint_DerivedView*		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.

7.6.7 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 uses only direct references to SERVICE JOURNEYs. Each ***GroupOfServicesMember*** may have a noticeAssignment

Table 96 — *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
			CHOICE	1:1	Reference to a JOURNEY.
«FK»	a	***JourneyRef***	*JourneyRef+*	0:1	Direct Reference to a JOURNEY
«cntd»	***noticeAssignments***		*NoticeAssignmentView*	0:*	NOTICE ASSIGNMENTs that apply to a GROUP OF SERVICE.

7.7 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.

7.7.1 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.

Table 97 — *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*
«cntd»	***frequencyGroups***	*JourneyFrequencyGroup*	0:*	FREQUENCY GROUPs defining TEMPLATE VEHICLE JOURNEY.

7.7.2 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 98 — *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»	***journeys***	*VehicleJourneyRef*	1:*	Journeys belonging to FREQUENCY GROUP.

7.7.3 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 99 — ***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.
	Description	*MultilingualString*	0:1	Description of Headway Journey Groups.

7.8 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).

7.8.1 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 100 — ***Interchange*** **– XML Element**

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	INTERCHANGE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*InterchangeIdType*	1:1	Identifier of INTERCHANGE.
«FK»	***ConnectionRef***	*ConnectionRef*	0:1	CONNECTION link over which the interchange takes place.
	StaySeated	*xsd:boolean*	0:1	Whether passengers can stay seated to make INTERCHANGE.
	CrossBorder	*xsd:boolean*	0:1	Whether INTERCHANGE involves crossing a national border.
	StandardTransfer-Time	*xsd:duration*	0:1	Standard Time needed for transfer.

7.8.2 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 101 — *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.

7.9 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.

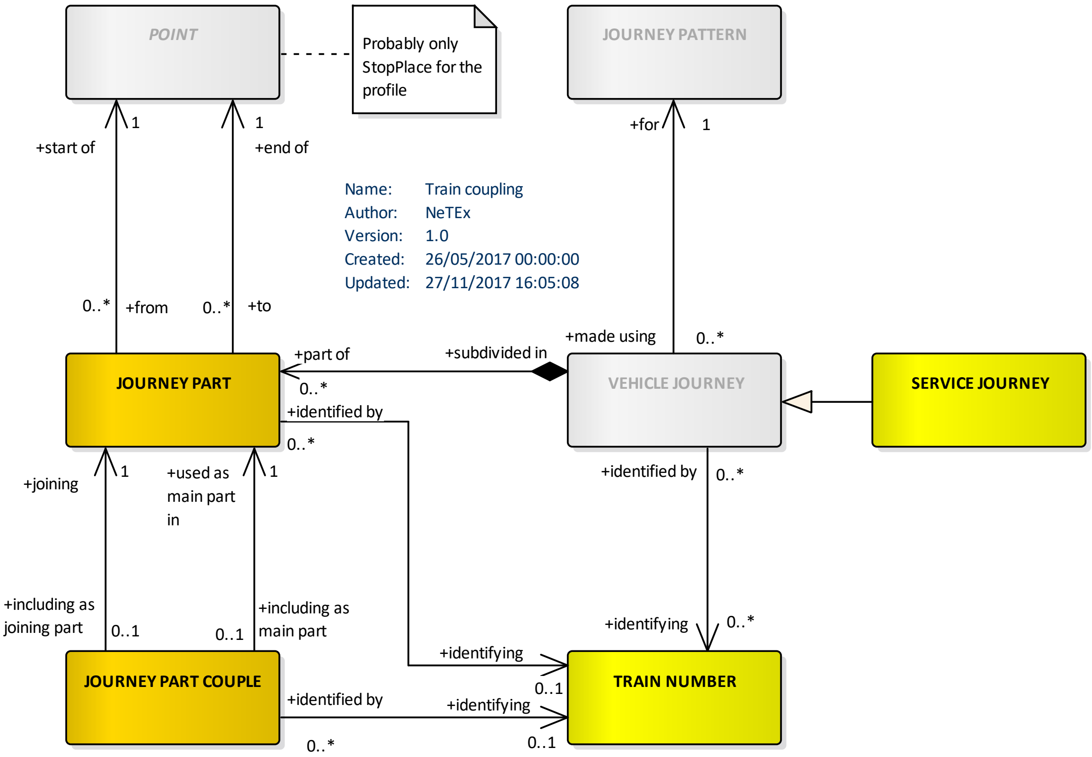

Figure 15 — Train Coupling (UML)

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.

7.9.1 TrainNumber

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 102 — *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.

7.9.2 JourneyPart (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 103 — *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»	***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»	***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.
«cntd»	***facilities***	*ServiceFacilitySet*	0:*	Facilities available during a JOURNEY PART.
«cntd»	***journeyPartPositions***	*JourneyPartPosition*	0:*	Positions in Train of JOURNEY PART

7.9.3 JourneyPartCouple

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 104 — *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.

7.9.4 JourneyPartPosition (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 105 — Table 5 – *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»	***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.

7.10 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.

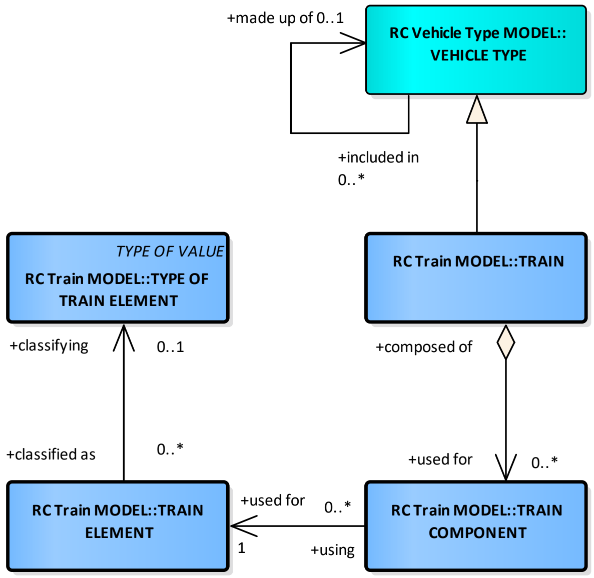

Figure 16 — Vehicles and Trains (UML)

7.10.1 VehicleType

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 106 — *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.
	Description	*MultilingualString*	0:1	Description of VEHICLE TYPE.
	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»	***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.

7.10.2 Train

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 passsenger information systems gived 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 107 — *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.

7.10.3 TrainElement (Subcomponent of Train)

A TRAIN ELEMENT is an elementary component of a TRAIN (e.g. wagon, locomotive).

Table 108 — *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* • *sleeperCarriage* • *restaurantCarriage* • *luggageVan*
	FareClasses	*FareClassEnum*	0:*	FARE CLASSes for TRAIN ELEMENT. See Annex.

7.10.4 TrainComponent (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 109 — *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.
«cntd»	b	***TrainElement***	*TrainElement*	1:1	TRAIN ELEMENT associated with TRAIN COMPONENT.

7.11 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, or example 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 EPIP as their main use in EPIP is to provide footnotes for SERVICE JOURNEYs and GROUPs of services.

7.11.1 Notice (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).

Table 110 — *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.
	Text	*MultilingualString*	0:1	Content text for NOTICE. Note: use AlternativeText for translation
«AK»	***PublicCode***	*xsd:normalizedString*	0:1	Public code for NOTICE.
«FK»	***TypeOfNoticeRef***	*TypeOfNoticeRef*	0:1	Reference to TYPE OF NOTICE.

7.11.2 NoticeAssigment (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, ***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 111 — 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.
«FK»	***a***	***NoticeRef***	*NoticeRef*	0:1	Reference to a NOTICE i.e. footnote, note, announcement or other informational text element.
	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»	***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.

7.12 Calendars

A SERVICE CALENDAR holds information about when services run.

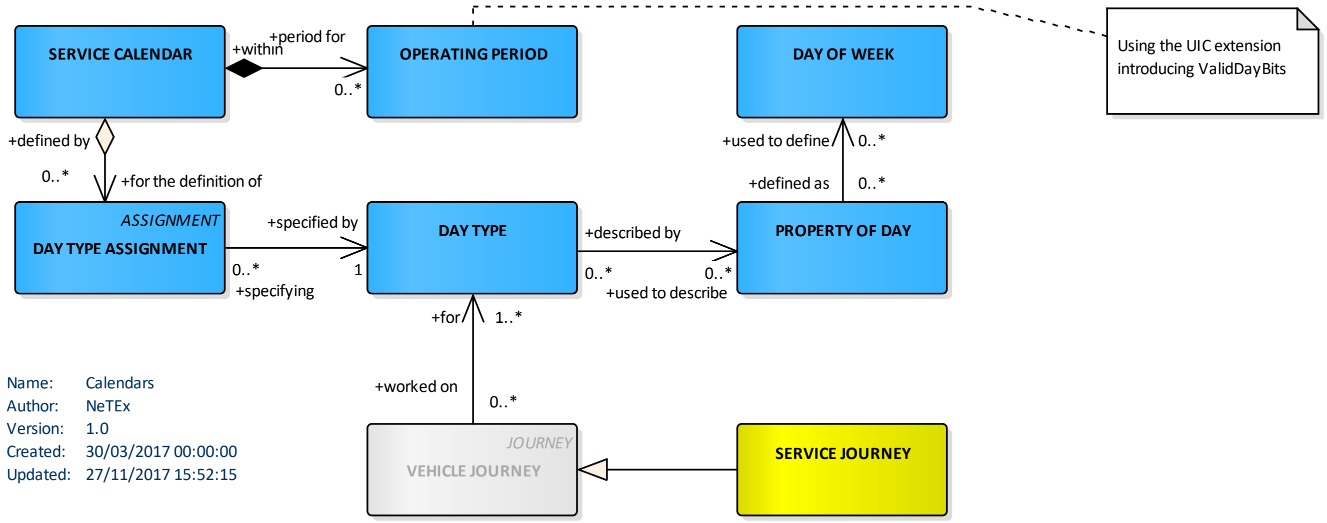

Figure 17 — Operating days and calendar (UML)

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.

This approach allows the same SERVICE JOURNEY definitions to be used with different SERVICE CALENDARs in successive time periods.

7.12.1 DayType

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”); also 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 instance should be logically ANDed, for example [“*Monday”* AND “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 112 — *DayType* – XML Element

Classification	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.
	Description	*MultilingualString*	0:1	Description of 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.

7.12.2 PropertyOfDay (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 113 — *PropertyOfDay – XML Element*

Classification	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*
	Seasons	*SeasonEnum*	0:4	Season of year assigned to PROPERTY OF DAY. • *Spring,* • *Summer,* • *Autumn, Winter,* • *Perennially*
	Crowding	*CrowdingEnum*	0:1	Relative busyness of day. • *veryQuiet* • *quiet* • *normal* • *busy* • *veryBusy*

7.12.3 Timeband (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

Table 114 — *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.
	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.

7.12.4 ServiceCalendar

A SERVICE CALENDAR groups the calendar related ENTITies and provides in particular a set of assignments of OPERATING DAYs to DAY TYPEs.

Table 115 — *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.
	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.
«cntd»	***dayTypes***	*DayType*	0:*	DAY TYPEs in SERVICE CALENDAR.
«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.

7.12.5 OperatingDay (Subcomponent of ServiceCalendar)

An OPERATING DAY is in most cases similar to a calendar day (i.e. an indidual 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 (such operational measures are outside of the scope of EPIP).

Table 116 — *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.
	EarliestTime	*xsd:time*	1:1	Start time of OPERATING DAY.
	DayLength	*xsd:duration*	1:1	Length of OPERATING DAY.

7.12.6 DayTypeAssignment (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 117 — *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.
	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.
	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.
	IsAvailable	xsd:boolean	0:1	Whether available on assigned day.

7.12.7 Operating Period (Abstract in EPIP)

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).

Table 118 — *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.
	Name		*MultilingualString*	0:1	Name of OPERATING PERIOD.
	b	***FromDate***	*xsd:dateTime*	1:1	Start OPERATING DATE (inclusive).
	b	***ToDate***	*xsd:dateTime*	1:1	End OPERATING DATE (inclusive).

7.12.8 UicOperatingPeriod (Subcomponent of ServiceCalendar)

The ***UicOperatingPeriod*** is an alternative optimised representation of an ***OperatingPeriod*** that uses a string of bits to indicate to which days with 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 119 — *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

7.13 Flexible services

NeTEx 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.

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.

7.13.1 FlexibleLine

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

FLEXIBLE LINE specialises LINE.

Table 120 — *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.

7.13.2 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, ***FlexiblePointProperies*** should be embedded within the ***StopPointInPattern*** which they describe.

Table 121 — *FlexiblePointProperties – XML Element*

Classification		Name	Type	Cardinality	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.

7.13.3 FlexibleServiceProperties (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 122 — 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).

8 Common rules

This section sets out common rules for different aspects of processing an EPIP compliant XML document.

8.1 Version management of data

The version and validity condition management under the EPIP is straightforward and is based on the use of VERSION FRAMEs.

The fundamental rules for object versioning are the following:

— All objects shall have a unique ***id*** (for certain dependent child components this may be qualified by an *'order';* in which case the *'id+order'* shall 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 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 (but their actual ***version*** value can be stated on an individual object basis, that is, does not have to be the same on all instances).
- Each VERSION FRAME shall be assigned a specific ***version*** value (i.e. not "*any*"), further more:
  - The value shall be a number (sequence of digits).
  - The version numbers used for a given VERSION FRAME shall 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 shall 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) shall 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 shall 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 shall 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 shall 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 requires a new version of an object
3. A change to a reference version requires a new version of an 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 shall 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 shall be incremented.

8.2 Use of validity conditions in EPIP

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 a frame.
The main rules for using VALIDITY CONDITIONs are the following:

- The **dominant** VALIDITY CONDITION is the one attached to the FRAME: 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.
- A FRAME shall have at least one **dominant** VALIDITY CONDITION and it shall give a start date.
- 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).
- The **dominant** VALIDITY CONDITION attached to a COMPOSITE FRAME is applies 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.
- 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*".

8.3 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 shall be indicated by the ***PublicationTimestamp*** of the ***PublicationDelivery*** wrapper tag: see *8.13*.

- 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:

XML EXAMPLE

```
<CompositeFrame version="3.0" id="epd:FR:RATP:CompositeFrame-EU_PI_LINE_OFFER:RATP-1">
    <ValidBetween>
        <FromDate>2019-01-01T09:02:30.0Z</FromDate>
        <ToDate>2020-01:01T00:02:30.0Z</ToDate>
    </ValidBetween>
```

8.4 Codespaces and the structure of identifiers

The data exchanged with EPIP 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 5.1.7). 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 shall 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 differ types of elements, provided the identifiers and class context are unique within it. For example, TOPOGRAPHIC PLACEs in the UK National gazetteer all begin with 'E,' UK NaPTAN SCHEDULED STOP POINT ids all begin with a numeric area prefix and have a 0, or 1 in the fourth place, while UK NaPTAN STOP AREA ids are similar to SCHEDULED STOP POINTs but have a 'G' in the fourth place (and moreover are different element types) – so they could all be placed in in a single 'NaPTAN' CODESPACE. Similarly, a single '*epip_metadata'* CODESPACE can be used for all EPIP metadata.

The following are some fictional examples of ***Codespace*** use for different identifiers:

Table 123 — *Codespace examples*

Examples	Codespace prefix and CODESPACE
`<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="napt:199G98765431">`	'napt' = http://naptan.org.uk/stops

8.4.1 Use of codespaces in EPIP

All identifiers in an EPIP conformant document shall 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.

8.4.2 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 ":" shall 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 6.1-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 shall 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 shall 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:UK::StopPlace_general:199G98765431:NaPTAN”>
```

8.4.3 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]*** shall be one of the set EPIP ***TypeOfFrame*** identifiers permitted by the EPIP (See *8.10*); and shall 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:

(epd:)[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.

XML EXAMPLES:

(a) 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>
```

(b) 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:SI:LPP:CompositeFrame-EU_PI_NETWORK_OFFER:LLJUBJANA"
                    responsibilitySetRef="SI:LPP: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="SI:LPP:ResponsibilitySet:LPP"/>
        </FrameDefaults
        <frames>
            <SiteFrame version="2.1" id="SI:LPP: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="SI:LPP:ServiceFrame-EU_PI_NETWOK:LPP-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="SI:LPP:TimetableFrame-EU_PI_TIMETABLE:LPP-1">
                <TypeOfFrameRef ref="epip:EU_PI_TIMETABLE" versionRef="1.0"/>
                <prerequisites>
                    <ServiceFrameRef version="any" ref="SI:LLP:ServiceFrame-EU_PI_SERVICE:LPP-1"/>
                </prerequisites>
                    <Etc., etc., Specific frame contents>
            </TimetableFrame>
            <ServiceFrame version="2.1" id="SI:LPP:ServiceFrame-EU_PI_NETWORK:LPP-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="SI:LPP:TimetableFrame-EU_PI_TIMETABLE:LPP-2">
                <TypeOfFrameRef ref="epip:EU_PI_TIMETABLE" versionRef="1.0" />
                <prerequisites>
                    <ServiceFrameRef version="any" ref="SI:LLP:ServiceFrame-EU_PI_SERVICE:LPP-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="SI:LPP: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="SI:LPP: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>
```

(c) 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>
```

8.5 Versions of the EPIP

Future work on the EPIP is likely to lead to some further refinement and extension of the profile. 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 EPIP 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 EPIP is therefore given a formal version number, (this, the initial version, being v1.0). To state the version number being used in an EPIP conformant document the ***version*** attribute on the ***TypeOfFrameRef*** shall be used.

An EPIP 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 EPIP profile being used on a frame should be stated on the ***TypeOfFrame*** reference that is mandatory for every frame in an EPIP conformant document. The version number on the outer ***CompositeFrame*** is taken as the document's EPIP version level. For example:

```
<CompositeFrame version="1.4" id=”epd:FR:RATP:CompositeFrame:EU_PI_LINE_OFFER:L22">
    <TypeOfFrameRef ref="epip:EU_PI_LINE_OFFER" versionRef="1.0"/>
```

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 EPIP 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).

8.6 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 '*EU_IP_STOP_OFFER'* while line data may be exchanged within a ***ServiceFrame*** together with schedule data in a ***TimetableFrame*** as part of an '*EU_IP_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 integrity checking of individual object references to elements in the prerequisite frame shall 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 '*EU_PI_NETWORK_OFFER'* frame that contains many separate frames for different lines.

For 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 EPI document in any order.

8.7 Organisation and nesting of elements within an EPIP document

The order and nesting of data elements in an XML document is largely dictated by the schema. Thus, for example, each type of frame states which type of element it may contain, and each of these elements has subcomponents and attributes which shall be nested within their parents in a specified order.

However in order to be flexible for different uses, the NeTEx schema (as discussed in section 5.2) 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 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 for implementors of the EPIP, 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 in the case of 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. 8.10. 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 other parent***).*** ***Notices*** are embedded within the ***NoticeAssigment***.
- 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*).

8.8 Responsibility for data

An EPIP conformant document 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 EU_PI_COMMON frame. This may also include one or more definitions of a ***ResponsibilitySet*** that indicate the roles.

XML EXAMPLE

```
<CompositeFrame version="3.0" id="UK::CompositeFrame-EU_PI_STOP_OFFER:490:NaPTAN"
                responsibilitySetRef=”UK:TfL:ResponsibilitySet:082:NPTG”>
    <TypeOfFrameRef ref="epip:EU_PI_STOP_OFFER" versionRef="1.0" />

    <FrameDefaults>
        <DefaultCodespaceRef ref="epip_data"/>
        <DefaultResponsibilitySetRef version=”any” ref="UK:TfL:ResponsibilitySet:082:NaPTAN"/>
    </FrameDefaults
    <frames>
        <SiteFrame version="2.1" id="UK:TFL:SiteFrame-EU_PI_STOP:490:NaPTAN">
            <TypeOfFrameRef ref="epip:EU_PI_STOP" versionRef="1.0" />
                <Etc., etc., Specific Frame contents>
        </SiteFrame>
        <ResourceFrame version="2.1" id="UK:TFL:ResourceFrame-EU_PI_COMMON:490:NaPTAN">
            <TypeOfFrameRef ref="epip:EU_PI_COMMON" versionRef="1.0" />
                <Etc., etc., Specific Frame contents>
            <responsibilitySets>
                <ResponsibilitySet version="any" id="UK:TfL:ResponsibilitySet:082:NaPTAN"
                    responsibilitySetRef="nptg:RS_nptg" >
                             <Name>TfL provides london data</Name>
                         <roles>
                             <ResponsibilityRoleAssignment version="any"
                                      id="UK:TfL:ResponsibilitySet:082:NaPTAN ">
                                 <Description>490 London data - Managed centrally</Description>
                                 <DataRoleType>collects</DataRoleType>
                                 <ResponsibleOrganisationRef version ref="noc:Tfl"/>
                                 <ResponsibleAreaRef versionRef="any"
                                     ="nptg:AdministrativeZone:82"/>
                             </ResponsibilityRoleAssignment>
                         </roles>
                </ResponsibilitySet>
        </ResourceFrame>
                <Etc., etc., Specific Frame contents>
```

8.9 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 EPIP 'OFFER' frame types ('*EU_PI_STOP_OFFER', 'EU_PI_LINE_OFFER'* and *'EU_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 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 EPIP 'OFFER' frame to use, and in deciding the selection parameters for the frame.
- Within an EPIP conformant document, data should be modularised into frames. For example, with an '*EU_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 'EU_PI_NETWORK' frame and the SERVICE JOURNEYs making up the schedule for each LINE should be placed in a separate ***TimetableFrame*** of EPIP type '*EU_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 'EU_PI_STOP_OFFER' file should be broken down by administrative zone or operator into separate ***SiteFrame*** instances of EPIP Type '*EU_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.

8.10 Available types of frame

As explained in 5.1.12 the following TYPEs OF FRAME are available for use in the EPIP:

Ancillary:

- *EU_PI_COMMON*
- *EU_PI_CALENDAR*

Payload:

- *EU_PI_STOP*
- *EU_PI_NETWORK*
- *EU_PI_TIMETABLE*

Composite:

- *EU_PI_LINE_OFFER*
- *EU_PI_NETWORK_OFFER*
- *EU_PI_STOP_OFFER*

A further TYPE OF FRAME can be used to declare EPIP specific metadata for validating the other frame types.

- *EU_PI_METADATA*
-

The following tables summarise the content of each TYPE OF FRAME.

8.10.1 Classes for SPECIFIC FRAMEs

‘*epip’* namespace =http:/nete-cen.eu/epip

The following notation is used:

	Unless nesting is indicated by an arrow, elements are contained directly in their frame type.
➔	Subelement is a component of preceding composite element: it shall 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 EPIP 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 EPIP in each frame are highlighted in green.

Table 124 — 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 EPIP frame type.		

Table 125 — TypeOfFrame: *EU_PI_COMMON*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*'epip:EU_PI_COMMON'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***ResourceFrame***	RESOURCE FRAME
classes	*ClassInContextRef*	***ResponsibilitySet*** →***ResponsibilityRoleAssignment***	RESPONSIBILITY SET →RESPONSIBILITY ROLE ASSIGNMENT
		Operator ***Authority***	OPERATOR AUTHORITY
		TypeOfFrame ***Branding***	TYPE OF FRAME [NeTEx only]
		DataSource	DATA SOURCE
		ValueSet →(TypeOfValue)	[NeTEx only] TYPE OF VALUE
		VehicleType	VEHICLE TYPE
Data scope: ***ANCILLARY***	A ***ResourceFrame*** instance conforming to *'EU_PI_COMMON'* shall contain ancillary data elements referenced by the main payload frames '*EU_PI_STOP*' and/or '*EU_PI_TIMETABLE*'.		

Table 126 — TypeOfFrame: *EU_PI_CALENDAR*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*'epip:EU_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
Data scope: ***ANCILLARY***	A ***ServiceCalendarFrame*** instance conforming to *'EU_PI_CALENDAR'* shall contain ancillary data elements (DAY TYPES, etc) referenced by the payload ***TimetableFrame*** *'EU_PI_TIMETABLE'*.		

Table 127 — TypeOfFrame: *EU_PI_STOP*

Name	**XML Type**	**Content XML Classes & Nesting**	**TM Entities & Composition**
id	*TypeOfFrameIdType*	*'epip:EU_PI_STOP'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***SiteFrame***	SITE FRAME
classes	*ClassInContextRef*	***StopPlace*** ***➔StopPlaceEntrance*** ***➔Quay***	STOP PLACE ➔STOP PLACE ENTRANCE ➔QUAY
		TopographicPlace ***➔TopographicPlaceDescriptor***	TOPOGRAPHIC PLACE ➔TOPOGRAPHIC PLACE DESCRIPTOR
		PointOfInterest	POINT OF INTEREST
		****→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 *'EU_PI_STOP' shall* contain stop data as required by the EPIP ***CompositeFrame*** types *'EU_PI_LINE_OFFER'* or *'EU_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		

Table 128 — TypeOfFrame: *EU_PI_NETWORK*

Name	**XML Type**	**Content XML Classes & Nesting**	**TM Entities & Composition**
id	*TypeOfFrameIdType*	'epip:*EU_PI_NETWORK'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***ServiceFrame***	SERVICE FRAME
classes	*ClassInContextRef*	***Line*** ***➔AllowedLineDirection***	LINE ➔ALLOWED LINE DIRECTION

		FlexibleLine	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***	ROUTE ➔POINT ON ROUTE
		ScheduledStopPoint	SCHEDULED STOP POINT
		ServiceLink	SERVICE LINK
		****➔TransferDuration*** ***Connection*** ➔***ConnectionEnd*** ***SiteConnection*** ➔***SiteConnectionEnd*** ***DefaultConnection*** ➔***DefaultConnectionEnd***	(TRANSFER) CONNECTION [NeTEx only] SITE CONNECTION [NeTEx only] DEFAULT CONNECTION [NeTEx only]
		ServiceJourneyPattern ➔***StopPointInJourneyPattern***	SERVICE JOURNEY PATTERN ➔STOP POINT IN JOURNEY PATTERN
		ServicePattern ➔***StopPointInJourneyPattern*** *::::*➔FlexiblePointProperties*	SERVICE PATTERN ➔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 '*EU_PI_NETWORK'* shall contain schedule data for only a single LINE.		

Table 129 — TypeOfFrame: *EU_PI_TIMETABLE*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*'epip:EU_PI_TIMETABLE'*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***TimetableFrame***	TIMETABLE FRAME
classes	*ClassInContextRef*	***GroupOfServices*** ***→GroupOfServicesMember***	GROUP OF SERVICEs [NeTEx only]
		ServiceJourney ***→JourneyPart*** ***→TimetabledPassingTime*** ***→FlexibleServiceProperties*** ***TemplateServiceJourney*** ***→HeadwayJourneyGroup***	SERVICE JOURNEY →JOURNEY PART →TIMETABLED PASSING TIME FLEXIBLE SERVICE PROPERTIES TEMPLATE SERVICE JOURNEY →HEADWAY 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 '*EU_PI_TIMETABLE* 'shall contain schedule data for only a single LINE.		

8.10.2 Classes for COMPOSITE FRAMEs

Table 130 — ***TypeOfFrame for*** *EU_PI_LINE_OFFER* - *(“EPIP-LN”)*

Name	Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*‘epip:EU_PI_LINE_OFFER’*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***CompositeFrame***	COMPOSITE FRAME
includes	*TypeOfFrameRef*	• *‘epip:EU_PI_COMMON’* • *‘epip:EU_PI_STOP’* • *‘epip:EU_PI_NETWORK’* • *‘epip:EU_PI_TIMETABLE’* • *‘epip:EU_PI_CALENDAR’*	TYPE OF FRAME (RESOURCE FRAME) TYPE OF FRAME (SITE FRAME) TYPE OF FRAME (SERVICE FRAME TYPE OF FRAME (TIMETABLE FRAME TYPE OF FRAME (SERVICE CALENDAR FRAME
Data Scope: CONTAINER	A ***CompositeFrame*** instance conforming to ***EU_PI_LINE_OFFER*** shall contain data for only a single LINE.		

Table 131 — ***TypeOfFrame for*** *EU_PI_NETWORK_OFFER* **–** *(“EPIP-NT****”)***

Name	Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*‘epip:EU_PI_NETWORK_OFFER’*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***CompositeFrame***	COMPOSITE FRAME
includes	*TypeOfFrameRef*	• *‘epip:EU_PI_COMMON’* • *‘epip:EU_PI_STOP’* • *’epip:EU_PI_NETWORK’* • *‘epip:EU_PI_TIMETABLE’* • *‘epip:EU_PI_CALENDAR’*	TYPE OF FRAME (RESOURCE FRAME) TYPE OF FRAME (SITE FRAME) TYPE OF FRAME (SERVICE FRAME) TYPE OF FRAME (TIMETABLE FRAME) TYPE OF FRAME (SERVICE CALENDAR FRAME)
Data Scope: CONTAINER	A ***CompositeFrame*** instance conforming to ‘*EU_PI_NETWORK_OFFER*’ shall include information for two or more LINEs that make up a transport network.		

Table 132 —

Table 133 — ***TypeOfFrame for*** *EU_PI_STOP_OFFER* **–** *(“EPIP-SP****”)***

Name	Type	Content XML Classes & Nesting	TM Entities & Composition
id	*TypeOfFrameIdType*	*‘epip:EU_PI_STOP_OFFER’*	TYPE OF FRAME
FrameClassRef	*NameOfClass*	***CompositeFrame***	COMPOSITE FRAME

includes	*TypeOfFrameRef*	- *'epip:EU_PI_COMMON'* - *'epip:EU_PI_STOP'*	TYPE OF FRAME (RESOURCE FRAME) TYPE OF FRAME (SITE FRAME)
Data Scope: CONTAINER	A ***CompositeFrame*** instance conforming to '*EU_PI_STOP_OFFER*' shall include one or more SiteFrame instances of type '*epip:EU_PI_STOP'* each giving information about STOP PLACEs for one or more MODEs, OPERATORs or ADMINISTRATIVE AREAs. Large data sets may be partitioned into multiple ***SiteFrame*** instances. an operator, region.		

8.10.3 Classes for EPIP Metadata FRAMEs

Table 134 — TypeOfFrame: *EU_PI_METADATA*

Name	XML Type	Content XML Classes & Nesting	TM Entities & Nesting
id	*TypeOfFrameIdType*	*'epip:EU_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 '*EU_PI_METADATA'* shall contain definitions of the fixed ***TypeOfFrame*** instances and ***TypeOfPlace*** instances defined by the EPIP.		

8.11 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 shall be included too so as to give a context).

> For simplicity, in the EPIP 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 EPIP for each element should be populated. A producer making an export shall therefore handle four different cases:

1. Elements and attributes which are **mandatory in the NeTEx schema** (so shall always be present).

2. Elements and attributes which are **optional in the NeTEx schema** but **mandatory in the EPIP** (so shall always be populated).
3. Elements and attributes which are **optional in the EPIP**; 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 EPIP**; these should follow the same principles as used for (3).

8.11.1 Empty 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 EPIP).
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 EPIP) - rather than say, just a delta of changes) - a consumer system shall 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 word 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.

8.12 Data exchange

EPIP, data exchange will be implemented through the exchange of files that are XML documents. Each file will contain a single **PublicationDelivery, itself** containing a single version frame of one of the permitted EPIP types).

Three different types of EPIP conformant XML document will be accepted as files:

- **Line** XML documents containing a single '*EU_PI_LINE_OFFER' **CompositeFrame*** (describing the stops, topology and timetable of a single line).
- **Network** XML documents containing a single '*EU_PI_NETWORK_OFFER' **CompositeFrame*** (providing the stops, topology and timetable of a multiple lines) Within the ***CompositeFrame*** there may multiple instances of the different EPIP specific frame types, for example a separate ***TimetableFrame*** for each LINE.

- **Stop** XML documents containing a single '*EU_PI_STOP_OFFER*' ***CompositeFrame*** (describing all the stops of a country, an authority or an operator). It will contain one or more '*EU_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.

8.13 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 accepted by the EPIP.

Thus, an EPIP conformant document wraps a single ***PublicationDelivery*** element around one or more "payload" ***CompositeFrame*** instances of the required EPIP ***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 non-profile NeTEx elements to augment or extend the content may also be present (see discussion in ***8.15-Extending the profile***). It is also acceptable to include a ***ResourceFrame*** with EPIP metadata instances that can be used to validate EPIP references.

Table 135 — *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 EPIP 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»				

	Description	*xsd:normalizedString*	0:1	Description of contents.
«cntd»	***dataObjects***	*dataObjects_RelStructure*	0:1	NeTEx VERSION FRAMES making up publication.

8.13.1 Publication 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 EPIP the ***PublicationRequest*** states the ***Line*** or ***Network*** whose data is contained in the frame***.***

Table 136 — *PublicationRequest* – Service Element

Classification	Name	Type	Cardinality	Description
«cntd»	***topics***	*NetworkFrameTopic*	0:*	One or more request filters that describe selection criteria for data to be included in output.

8.13.2 NetworkFrameTopic (Subcomponent of PublicationDelivery)

The ***NetworkFrameTopic*** specifies the selection criteria for including NeTEx objects in the publication.

For EPIP only the ***NeteorkFilterByValue*** 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 ***TograhicPlace***

Table 137 — *NetworkFrameTopic* – Service Element

Classification	Name		Type	Cardinality	Description
	a	***Current***	*EmptyType*	1:1	Return just the values from the producer current at the time the response is created.
«cntd»	b	***NetworkFilterByValue***	*NetworkFilterByValueStructure*	1:0	Values for selecting by value

8.13.3 *NetworkFilterByValue* (Subcomponent of PublicationDelivery)

The ***NetworkFilterByValue*** specifies the selection criteria for including NeTEx objects in the publication.

Table 138 — *NetworkFilterByValue – XML Element*

Classification	Name	Type	Cardinality	Description
«cntd»	***objectReferences***	*(VersionOfObjectRef)*	0:*	Reference to ENTITIES to be returned. In EPIP use to state the LINE if type of CompositeFrrame is for LINE_OFFER
cntd»	***NetworkRef***	*NetworkRefStructure*	0:1	Reference to a NETWORK.
cntd»	***places***	*(PlaceRef)*	0:*	Return all elements conerned with given place. EPIP can state Topographical Palce eg Country, region or araea

8.13.4 PublicationDelivery example

The following fictional example shows a ***PublicationDelivery*** header for a ‘EU_PI_LINE_OFFER’ for RATP Line 1. The frames making up the delivery are also indicated but their contents are omitted.

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="epip:EU_PI_LINE_OFFER" versionRef="1.0" />
                        <LineRef="FR:RATP:Line:1"/>"
                    </objectReferences>
                    <places>
                        <TopographicPlaceRef ref="iso3055:fr”/>France
                    </places>
                    <NetworkRef="FR:RATP:Network:Noctilien"/>
                </NetworkFilterByValue>
            </NetworkFrameTopic>
        </topics>
    </PublicationRequest>
        <dataObjects>
            <CompositeFrame version="3.0" id="ep:FR:RATP:CompositeFrame-EU_PI_LINE_OFFER:Noctilien-1”>
                <ValidBetween>
```

```
                    <FromDate>2019-01-01T09:02:30.0Z</FromDate>
                    <ToDate>2020-01:01T00:02:30.0Z</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"/>
                </FrameDefaults
                <frames>
                    <SiteFrame version="2.1" id="FR:RATP:SiteFrame-EU_PI_STOP:Noctilien-1">
                        <TypeOfFrameRef ref="epip:EU_PI_STOP" versionRef="1.0" />
                            <prerequisites>
            <ResourceFrameRef version="any" ref=" FR:RATP:ResourceFrame-EU_PI_COMMON:Noctilien-1"/>
                            </prerequisites>

                            <Etc., etc., Specific Frame contents>
                    </SiteFrame>
                    <ServiceFrame version="2.1" id="FR:RATP:ServiceFrame-EU_PI_NETWORK:Noctilien-1">
                        <TypeOfFrameRef ref="epip:EU_PI_NETWORK" versionRef="1.0" />
                            <prerequisites>
            <ResourceFrameRef version="any" ref=" FR:RATP:SiteFrame-EU_PI_COMMON:Noctilien-1"/>
                            </prerequisites>

                            <Etc., etc., Specific Frame contents>
                    </ServiceFrame>
            <TimetableFrame version="2.4" id="FR:RATP:TimetableFrame-EU_PI_TIMETABLE:Noctilien-1”>
                        <TypeOfFrameRef ref="epip:EU_PI_TIMETABLE" versionRef="1.0" />
                            <prerequisites>
            <ServiceFrameRef version="any" ref=" FR:RATP:ServiceFrame-EU_PI_COMMON:Noctilien-1"/>
            <ServiceCalendarFrameRef version="any" ref=" FR:RATP:ServiceCalendarFrame-
EU_PI_CALENDAR:Noctilien-1"/>
                            </prerequisites>

                            <Etc., etc., Specific Frame contents>
                    </TimetableFrame>
    <ServiceCalendarFrame version="1.6" id="FR:RATP:ServiceCalendarFrame-EU_PI_CALENDAR:Noctilien-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:Noctilien-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">
                        <TypeOfFrameRef ref="epip:EU_PI_METADATA" versionRef="1.0" />
                            <Etc., etc., Specific Frame Fixed contents>
                    </ResourceFrame>
                </frames>
            </CompositeFrame>
    </dataObjects>
```

```
</PublicationDelivery>
```

8.14 Names of files for EPIP documents

It is helpful to developers and automated workflow applications if the names used for XML document given an indication of their contents and are uniform (and unique).

The following file name structure is recommended for any file dissemination of EPIP conformant documents:

[prefix] - [epip-version] _ [country-code] _ [provider-code] _ [profile-type] _ [doc-topic] _[creation-date]

Where:

- **[prefix]** A fixed prefix agreed upon by the exchange partners, here ‘NX-PI ‘
- **[epip-version]** Major version number of the EPIP; currently ‘01’
- **[country-code]** Country ISO code.
- **[provider-code]** Producer code (use ‘NAP’ if coming from a national access point). Variable length
- **[profile-type]** EPIP 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 EPIP [doc-topic] should correspond to the [frame-topic] specified in 8.4.3 and the publication header (8.13)
  - For ‘*EU_PI_NETWORK_OFFER*’, short name;
  - For ‘*EU_PI_LINE_OFFER*’: LINE number or name if no number is available
  - For ‘*EU_PI_*STOP_*OFFER*’, name or code of LINE, ADMINISTRATIVE ZONE or TOPOGRAPHIC PLACE for which the stops are included.
- **[creation-date]** Date of creation of the XML document (YYYMMDD).
- All fields are separated as indicated either by a hyphen “-“ or by an underscore "_".

Example file name:

```
NX-PI-01_GE_BVB_LINE-3_20161201.xml

NX-PI-02_FR_FR1_NETWORK_RATP_20171105.xml

NX-PI-01_UK_NAP_STOP_AA-471_20171212.xml
```

8.15 Extending the profile

The choices made by the EPIP deliberately limit-what can be exchanged to the minimal subset of NeTEx needed for exchanging a timetable. In practice, it may be that the EPIP mostly fits a particular intended use (and EPIP’s 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 EPIP based data sets with local extensions, but still keep the data set usable as an EPIP conformant document that is readable by any EPIP capable consumer system, the extensions should be used in a manner consistent with the EPIP'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 EPIP, (i.e. Attributes available in the NeTEx schema for object types that are already selected by the EPIP):
    - Excluded attributes are allowed, but the object semantic and consistency 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 EPIP part of the data set to be consistent (i.e. the EPIP 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 EPIP 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 (or features not covered by the NeTEx schema):
    - The normal EPIP 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 EPIP 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 EPIP elements), (ii) using additional enumeration values, (iii) adding ***KeyValue*** instances.
3. **Extended conformance:** Adding additional NeTEx object types as well as augmenting the existing EPIP classes.

A producer system can indicate the nature of the intended conformance of a frame using a ***KeyValue***.

XML EXAMPLE:

```
<SiteFrame version="2.3" id=" FR:RATP:SiteFrame-EU_PI_STOP:RATP-1">
    <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="epip:EU_PI_STOP" versionRef="1.0" />
```

9 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 EPIP 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.

9.1 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. An EPIP document shall 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 (either because XML does not support such a check, 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 an 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 EPIP 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.

9.2 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)CE 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*** shall be populated.

9.3 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 EPIP.

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 EPIP the Transmodel conceptual model as interpreted for NeTEx.

   - Many rules are inherent in the meaning of the conceptual model. For example, that a ***ServicePattern*** shall 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 9.3 below).

- 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); for 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.
- 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 9.5. 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", "check for accessibility data", 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 EPIP 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.*

9.4 Metadata for profiles

NeTEx contains several element types and attributes that of are useful for describing some of the additional validation rules of a specific profile – as such, they really describe metadata (data about data) rather payload data content. These 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 5.3.2).
- ***TypeOfValue***: specifies the allowed values for open code lists (TypeOfValue is abstract; it is specialised as ***TypeOfLine***, ***TypeOfPoint***, etc). Used EPIP specify the monomodal and general stop classifications). See 5.4.1
- ***ValueSet***: groups a set of instances of ***TypeOfValue***. See 5.4.2
- ***VersionFrame / prerequisites:*** indicates which other frames are needed to form a coherent set of data. See 5.3.3.
- ***KeyList:*** can indicate optional characteristics for which a validator may check. Use in the EPIP to indicate if conformance is meant to be *strict* or otherwise. See 5.1.5.

9.5 Summary of data quality rules for EPIP

The following table provides an overview of relevant quality checking rules that may be applied to an EPIP conformant document, grouped approximately according to the criteria discussed above and with the appropriate validation mechanism indicated. The list Is not exhaustive.

Table 139 — Primary 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 shall not accidentally correspond to those of another data provider. — All identifiers shall be unique within the document within their codespace and uniqueness scope (see 8.4). — 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 shall also be correct (checkable with a Profile validator) — Eternal references should be valid references. i.e. present in another available dataset. — A ***versionRef*** value should be specified for all external references so that they 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 ***StopPlace*** 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 expected 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, shall uphold the profile principles for use (see ***8.15-Extending the profile***). In particular they shall not overlap with the content of any 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 shall be placed only in the version frame type 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 semantics 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** shall be consistent with the ***TransportMode*** (‘*schoolBus’* is not a submode of ‘*train’*). — A ***ServicePattern*** shall have a minimum of two ***ScheduledStopPoints***; — The type and mode of a ***ScheduledStopPoint*** shall be consistent with that of any ***StopPlace*** to which it is assigned with a ***PassengerStopAssignment***. (a ***ScheduledStopPoint*** with ***StopType=*** ‘*onstreetBus’* should not be assigned to 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; (**_which shall 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 Profile validator)	The populating of data values in an EPIP conformant document shall fully uphold the specific semantics of the EPIP: — The additional EPIP limitations on the normal Transmodel/ NeTEx semantics shall be followed. For example, the restriction of a ***StopPlace*** hierarchy to two levels.

		— A ***StopPlace*** shall be categorised with a ***TypeOfPlace*** as ‘*epip*:*general* ‘ or ‘*epip*:*monomodal’.* — The rules for encoding EPIP identifiers shall be followed. — The specified EPIP rules for organising and nesting the objects in a frame shall be followed. (in particular where EPI require the inline embedding of ancillary elements that the NeTEx schema also allows to be placed separately and cross referenced). — The ‘*epip’* ***Codespace*** prefix is reserved for EPIP metadata. — Etc.
F	**Data Completeness** (EPIP Profile validator)	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 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 EPIP include: — A EU_PI_LINE_OFFER ***CompositeFrame*** shall contain all the information (frames and content objects) required to properly describe a ***Line***. — Definitions shall be present for all the ***StopPlaces*** used by a ***Line*** through its ***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 one 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. (The ***StopPlace*** instances for Paris 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 few minutes. — 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: — 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 EU_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** (Custom profile validator)	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. Thus, it may be necessary to specify further stricter constraints on certain elements and usages the are beyond the scope of the strict EPIP.

The following paragraphs provide a more detailed list of quality checking rules belonging to the above categories. However, this list is not intended to be comprehensive and shall be completed within quality checking tools.

9.6 Referential integrity *(automatic)*

As explained in section 5.2, **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:

```
<Route version="0564" id="NO::Route:985a:ENTUR">
</Route>

<RouteRef version="0564" ref="NO::Route:985a:ENTUR">INTERNAL</RouteRef>
<RouteRef versionRef="0564" ref="NO::Route:985a:ENTUR">EXTERNAL</RouteRef>
```

or

```
<Route version="any" id="NO::Route:985a:ENTUR">
</Route>

<RouteRef version="any" ref="NO::Route:985a:ENTUR">INTERNAL</RouteRef>
<RouteRef versionRef="any" ref="NO::Route:985a:ENTUR">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 incuded in the EPIP. 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.

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.

BookingArrangements 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 it 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 specifc 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 containdata 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 indivudal 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 parirs 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 ocale to be used when procesing 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 dispalys 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 shall 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 ELEMENTCE

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 wellknown 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 shall 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 ENTITIY 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 objeects 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 AccommodationAccessList

other
freeSeating
reservation
standing

B.5 AccommodationFacilityList

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

Annex C
(informative)

Related EU Directive

C.1 ITS Directive Delegated Regulation: EU-wide Multimodal Travel Information Services

The ITS Directive is a framework for the coordinated and effective deployment and use of intelligent transport systems. Its main objectives are:

- Establish a framework for coordinated and effective deployment and use of ITS within MS and across borders.
- Develop specifications necessary to ensure the compatibility, interoperability and continuity for the deployment and operational use of ITS for priority actions.

The ITS directive defines the following Priority Action.

a) the provision of EU-wide multimodal travel information services
b) the provision of EU-wide real-time traffic information services;
c) data and procedures for the provision, where possible, of road safety related minimum universal traffic information free of charge to users;
d) the harmonised provision for an interoperable EU-wide eCall;
e) the provision of information services for safe and secure parking places for trucks and commercial vehicles;
f) the provision of reservation services for safe and secure parking places for trucks and commercial vehicles.

This document is designed to answer to the expectations of Priority Action a) which is expecting the definition of the necessary requirements to make EU-wide multimodal travel information services accurate and available across borders based on one or more of the following types of provisions:

- functional provisions describing the roles of various stakeholders and the information flow between them.
- technical provisions providing for the technical means to fulfil functional provisions.
- organisational provisions describing procedural obligations of the various stakeholders.
- service provisions describing the levels of services and their content.

The follow are two excerpt article of this commission delegated regulation (eu) of 31.5.2017

> *(17) To ensure the optimal use and full interoperability of the aforementioned standards between Member States,* ***a common minimum profile that identifies the different key elements of the standard should be established*** *and used within national access points. Member States' national profiles shall be based on a common minimum European profile when it exists.*
>
> *(25) Through the Connecting Europe Facility the Commission will support different technical requirements established within this Regulation through a programme support action19, notably the*

> *establishment of the national access point, the conversion to prescribed data exchange standards and* ***the use of common minimum profiles*** *within national access points and the linkage of travel information services where relevant.*

References:

— Europe on the Move: Commission takes action for clean, competitive and connected mobility https://ec.europa.eu/transport/modes/road/news/2017-05-31-europe-on-the-move_en

— Commission Delegated Regulation on the provision of EU-wide multimodal travel information service http://ec.europa.eu/info/law/better-regulation/initiatives/c-2017-3574_en

Bibliography

[1] ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

[2] ISO 639/IETF 1766, *Tags for the Identification of Languages*

[3] ISO/IEC 19501-1:2002, *Unified Modelling Language (UML) – Part 1: Specification*

[4] National standards, in particular profile NEPTUNE, TransXChange, BISON and VDV 452, and other standards like NOPTIS

[5] ERA TAP-TSI: Commission Regulation (EU) No 454/2011 of 5 May 2011 on the technical specification for interoperability relating to the subsystem 'telematics applications for passenger services' of the trans-European rail system.

[6] UIC recommendations and leaflets

[7] XML, Extensible Mark-up Language (XML) 1.0 W3C Recommendation 04 February 2004, available at http://www.w3.org/TR/2004/REC-xml-20040204.