**CEN/TC 278**

Date: 2013-03

**TC 278 WI 00278308**

CEN/TC 278

Secretariat: NEN

# Public transport — Network and Timetable Exchange (NeTEx) — Part 2: Public transport scheduled timetables exchange format

*NeTEx — Haupt-Element — Teil 2: Teil-Titel*

*Transport Public — Echanges des informations planifiées (NeTEx) — Partie 2 : Description de l'offre de transport*

ICS:

Descriptors:

Document type: Technical Specification
Document subtype:
Document stage: Formal Vote
Document language: E

C:\Data\NeTEx\Part 2 - V17 Publication\Part 2 - Word document\prCEN TS 278308 FV (E)-v17 UpdatedDoc.doc
STD Version 2.5a

# Contents

Page

# Foreword

This document (TC 278 WI 00278308) has been prepared by Technical Committee CEN/TC 278 “Road transport and traffic telematics”, the secretariat of which is held by NEN.

This document is currently submitted to the Formal Vote.

This document presents Part 2 of the European Technical Specification known as “NeTEx”. NeTEx provides a framework for specifying communications and data exchange protocols for organisations wishing to exchange scheduled Information relating to public transport operations.

This Technical Specification 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 case 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, and parts 2 and 3 rely on part 1.

The XML schema can be downloaded from www.**netex**.org.uk, along with available guidance on its use, example XML files, and case studies of national and local deployments.

This document is higly technical, and a 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, attributes names are in bold/italic style and use camelcase style with no spaces (***JourneyPattern*** for example).

— To avoid confusion with usual wording, attributes 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 Network and Timetable Exchange (NeTEx) about public transport. It is intended to be used to exchange information between PT organisations systems containing scheduled public transport data. It can also be seen as a complement to the SIRI (Service Interface for Real-time Information) standard, as SIRI needs reference data exchange 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 standardised 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 function.

This standard will improve 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; (ii) using common data models and schemas for the messages exchanged for each service; and (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.

# 1 Scope

## 1.1 General

NeTEx is dedicated to the exchange of scheduled data (network, timetable and fare information) based on Transmodel V5.1 (EN 12986), IFOPT (CEN/TS 28701) and SIRI (CEN/TS 15531-4/5 and EN 15531-1/2/3[1]) and supports information exchange of relevance to public transport services for passenger information and AVMS systems.

NOTE Many NeTEx concepts are taken directly from Transmodel and IFOPT; the definitions and explanation of these concepts are extracted directly from the respective standards and reused in NeTEx, sometimes with further adaptions in order to fit the NETEx context.

The data exchanges targeted by NeTEx are predominantly oriented towards passenger information and also for data exchange between transit scheduling systems and AVMS (Automated Vehicle Monitoring Systems). However it is not restricted to these purposes, and NeTEx can provide an effective solution to many other use cases for transport exchange.

## 1.2 Transport modes

Most public transport modes are taken into account by NeTEx, including train, bus, coach, metro, tramway, ferry, and their submodes. It is possible to describe airports and air journeys, but there has not been any specific consideration of any additional provisions that apply especially to air transport.

## 1.3 Compatibility with existing standards and recommendations

The concepts covered in NeTEx that relate in particular to long-distance train travel include; rail operators and related organizations; stations and related equipment; journey coupling and journey parts; train composition and facilities; planned passing times; timetable versions and validity conditions.

In the case of long distance train the NeTEx takes into account the requirements formulated by the ERA (European Rail Agency) – TAP/TSI (Telematics Applications for Passenger/ Technical Specification for Interoperability, entered into force on 13 May 2011 as the Commission Regulation (EU) No 454/2011), based on UIC directives.

As regards the other exchange protocols, a formal compatibility is ensured with TransXChange (UK), VDV 452 (Germany), NEPTUNE (France), UIC Leaflet, BISON (Netherland) and NOPTIS (Nordic Public Transport Interface Standard).

The data exchange is possible either through dedicated web services, through data file exchanges, or using the SIRI exchange protocol as described in part 2 of the SIRI documentation.

# 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 15531-1, *Public transport - Service interface for real-time information relating to public transport operations - Part 1: Context and framework*[2]

[1] Under development

[2] Under development (WI 00278340)

EN 15531-2, *Public transport - Service interface for real-time information relating to public transport operations - Part 2: Communications infrastructure*[3]

EN 15531-3, *Public transport - Service interface for real-time information relating to public transport operations - Part 3: Functional service interfaces*[4]

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*

EN 12896, Road transport and traffic telematics - Public transport - Reference data model

EN 28701, Intelligent transport systems - Public transport - Identification of Fixed Objects in Public Transport (IFOPT)

# 3 Terms and definitions

For the purposes of this document, the terms and definitions given in CEN/TS 16614-1 apply.

# 4 Symbols and abbreviations

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

# 5 Use Cases for Journey & Journey Time Exchange

NeTEx Part 2 shares its use cases with NeTEx Part 1 since many use cases involve both part 1 and part 2 entities. Please refer to NeTEx Part 1 for a detailed use case description.

# 6 Generic Physical Model and XSD mapping rules

For consistency, the mapping rules for transforming a Conceptual Model to Physical Model and then to XSD are shared between all parts of NeTEx.

Please refer to NeTEx Part 1 for a detailed description of the Physical Model and XSD mapping rules.

# 7 Timing Information – Conceptual and physical data model

## 7.1 Introducion

NeTEx Part 2 timing information model is splited into four main submodels defined as UML packages.

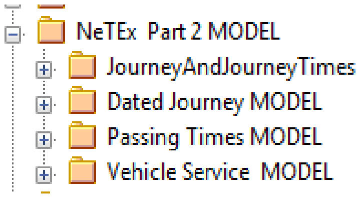


**Figure 1 – NeTEx Part 2 main packages**

[3] Under development (WI 00278341)

[4] Under development (WI 00278342)

- The Journey and Journey Times model: describes the model planned services and dead runs and their timings

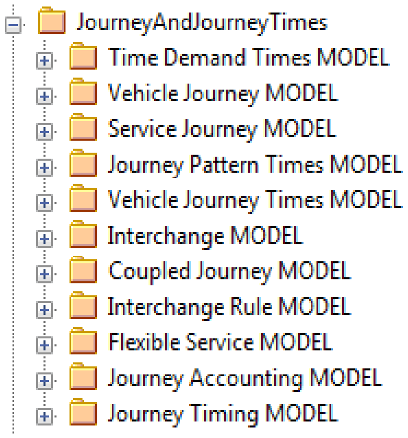


**Figure 2 – JourneyAndJourneyTimes packages**

- The dated journey model: describes the services for a single operating day

- The passing time model: describes all the different types of passing times

- The vehicle service model: describes the informations related to vehicles and their services

### 7.1.1 Journey and Journey Times – Model dependencies

The JOURNEY AND JOURNEY TIMES Model describes the VEHICLE JOURNEYs and other components making up a timetable and is itself divided into a number of separate submodels covering different aspects of VEHICLE JOURNEYs. For ease of understanding, the submodels are presented one at a time, each describing only a small set of related concepts.

The submodels depend on a number of general NeTEx framework models and reusable components described elsewhere (for example, the GENERIC POINT AND LINK model, NOTICE model, etc.,) – See NeTEx Part 1 for further details.

The following figure shows the dependencies between the JOURNEY AND JOURNEY TIMES physical submodels. The terminal packages contain the SERVICE FRAME and the TIMETABLE FRAME. These two VERSION FRAMEs are containers that organise the other payload elements into a coherent set of elements suitable for exchange as a serialised file. The payload elements are contained in the following packages:

TIMETABLE FRAME

- VEHICLE JOURNEY: models journeys that vehicles make**.**

- SERVICE JOURNEY: additionally models the properties of journeys that carry passengers.

- TIME DEMAND TIMEs: models the times of the different demand levels found during a day.

- PASSING TIMEs: describes the times of vehicles at points in their journey.

- JOURNEY TIMINGs: describes the common timing properties for journeys.

— JOURNEY PATTERN TIMEs: describes the timings of JOURNEY PATTERNs.

— VEHICLE JOURNEY TIMEs: describes the timings of VEHICLE JOURNEYs.

— INTERCHANGE: describes interchanges between journeys.

— COUPLED JOURNEY: describes multipart journeys which join and split.

— FLEXIBLE SERVICE: additional describes demand responsive transport services.

— JOURNEY ACCOUNTING: assigns a cost basis for journeys.

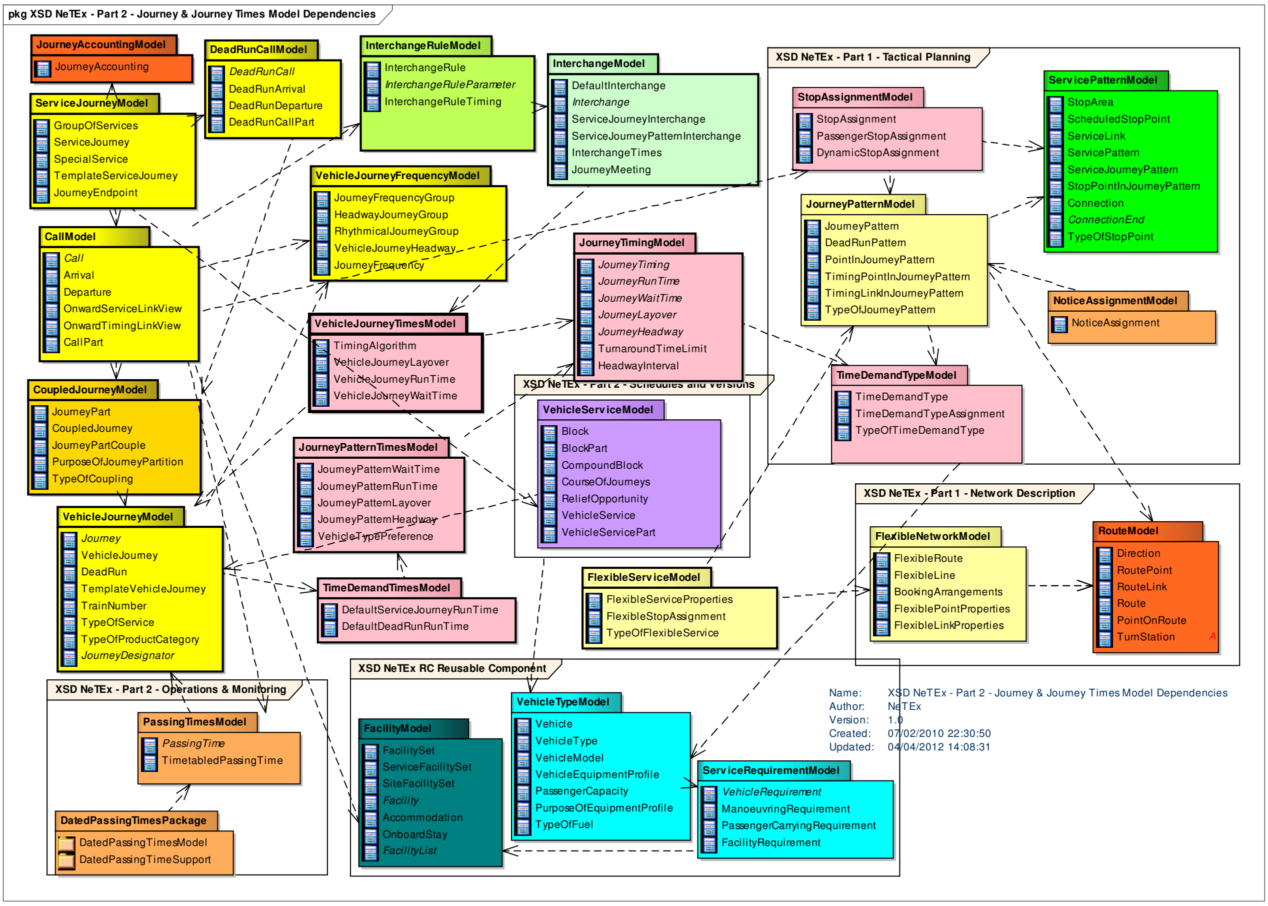


**Figure 3 — Journey – Model Dependencies**

### 7.1.1.1 Timetable Frame

#### 7.1.1.1.1 TIMETABLE FRAME – Conceptual MODEL

The elements of the JOURNEY & JOURNEY TIMES model can be grouped with a TIMETABLE FRAME which holds a coherent set of timetable related elements for data exchange (see VERSION FRAME in the NeTEx Framework section for general concepts relating to version frames).

The primary component exchanged by a TIMETABLE FRAME is a SERVICE JOURNEY, which describes an individual journey. This and other components of a TIMETABLE FRAME are described in detail in the following sections.

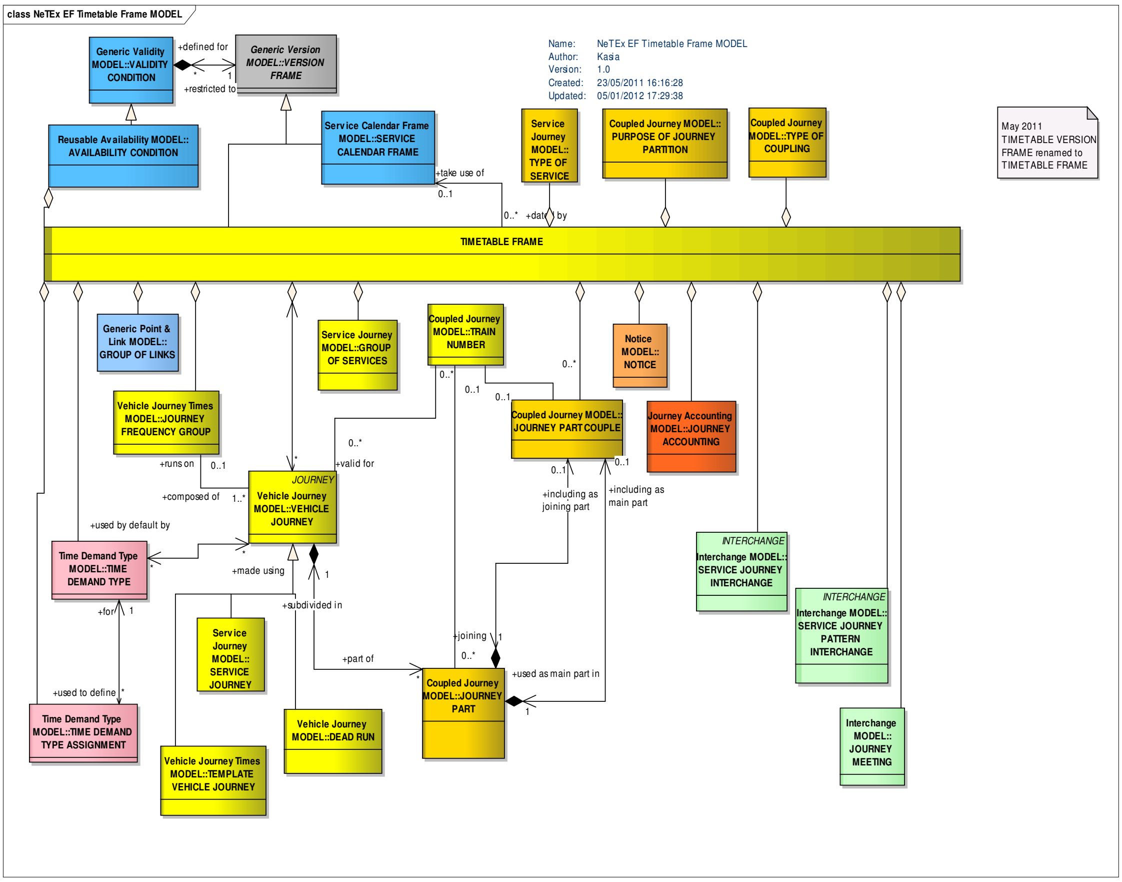

class NeTEx EF Timetable Frame MODEL
Name: NeTEx EF Timetable Frame MODEL
Author: Kasia
Version: 1.0
Created: 23/05/2011 16:16:28
Updated: 05/01/2012 17:29:38
May 2011 TIMETABLE VERSION FRAME renamed to TIMETABLE FRAME
Generic Validity MODEL::VALIDITY CONDITION
+defined for
*
1
+restricted to
Generic Version MODEL::VERSION FRAME
Reusable Availability MODEL:: AVAILABILITY CONDITION
Service Calendar Frame MODEL::SERVICE CALENDAR FRAME
+take use of
0..1
0..*
+dated by
Service Journey MODEL:: TYPE OF SERVICE
Coupled Journey MODEL:: PURPOSE OF JOURNEY PARTITION
Coupled Journey MODEL::TYPE OF COUPLING
TIMETABLE FRAME
Generic Point & Link MODEL:: GROUP OF LINKS
Service Journey MODEL::GROUP OF SERVICES
Coupled Journey MODEL::TRAIN NUMBER
0..*
0..1
0..1
0..*
Notice MODEL:: NOTICE
Vehicle Journey Times MODEL::JOURNEY FREQUENCY GROUP
+runs on
0..1
+composed of
1..*
*
0..*
+valid for
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
Coupled Journey MODEL:: JOURNEY PART COUPLE
0..1
+including as joining part
0..1
+including as main part
Journey Accounting MODEL::JOURNEY ACCOUNTING
+used by default by
Time Demand Type MODEL::TIME DEMAND TYPE
*
*
+made using
1
+subdivided in
+for
1
+used to define
*
Time Demand Type MODEL::TIME DEMAND TYPE ASSIGNMENT
Service Journey MODEL:: SERVICE JOURNEY
Vehicle Journey Times MODEL::TEMPLATE VEHICLE JOURNEY
Vehicle Journey MODEL::DEAD RUN
+part of
*
+joining
1
0..*
Coupled Journey MODEL::JOURNEY PART
+used as main part in
1
INTERCHANGE
Interchange MODEL:: SERVICE JOURNEY INTERCHANGE
INTERCHANGE
Interchange MODEL:: SERVICE JOURNEY PATTERN INTERCHANGE
Interchange MODEL:: JOURNEY MEETING

Figure 4 — Timetable Frame – Conceptual MODEL (UML)

#### 7.1.1.1.2 Timetable Frame – Physical Model

The following diagram shows the Physical model for a TIMETABLE FRAME.

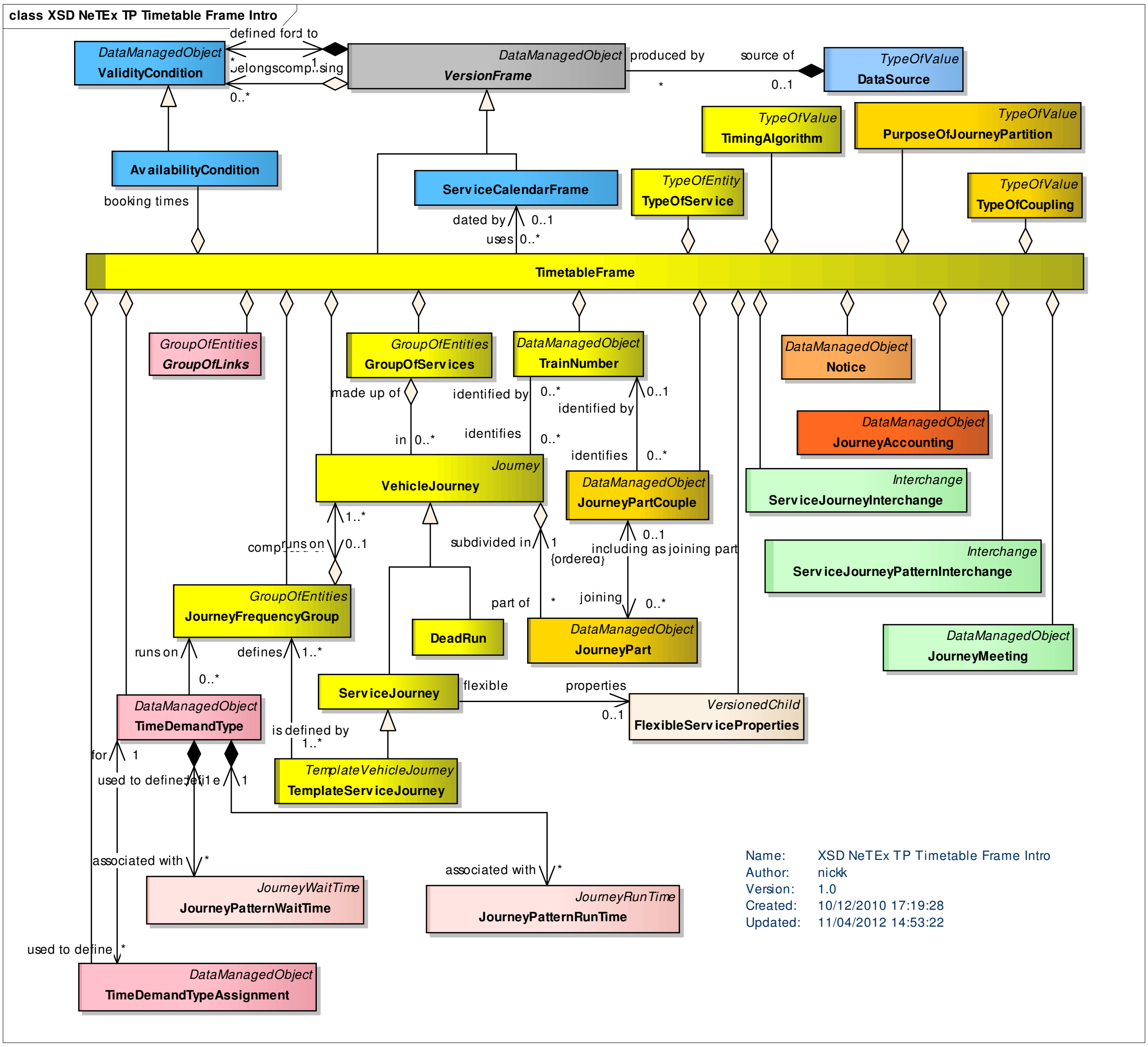


**Figure 5 — Timetable Frame Contents – Physical Model (UML)**

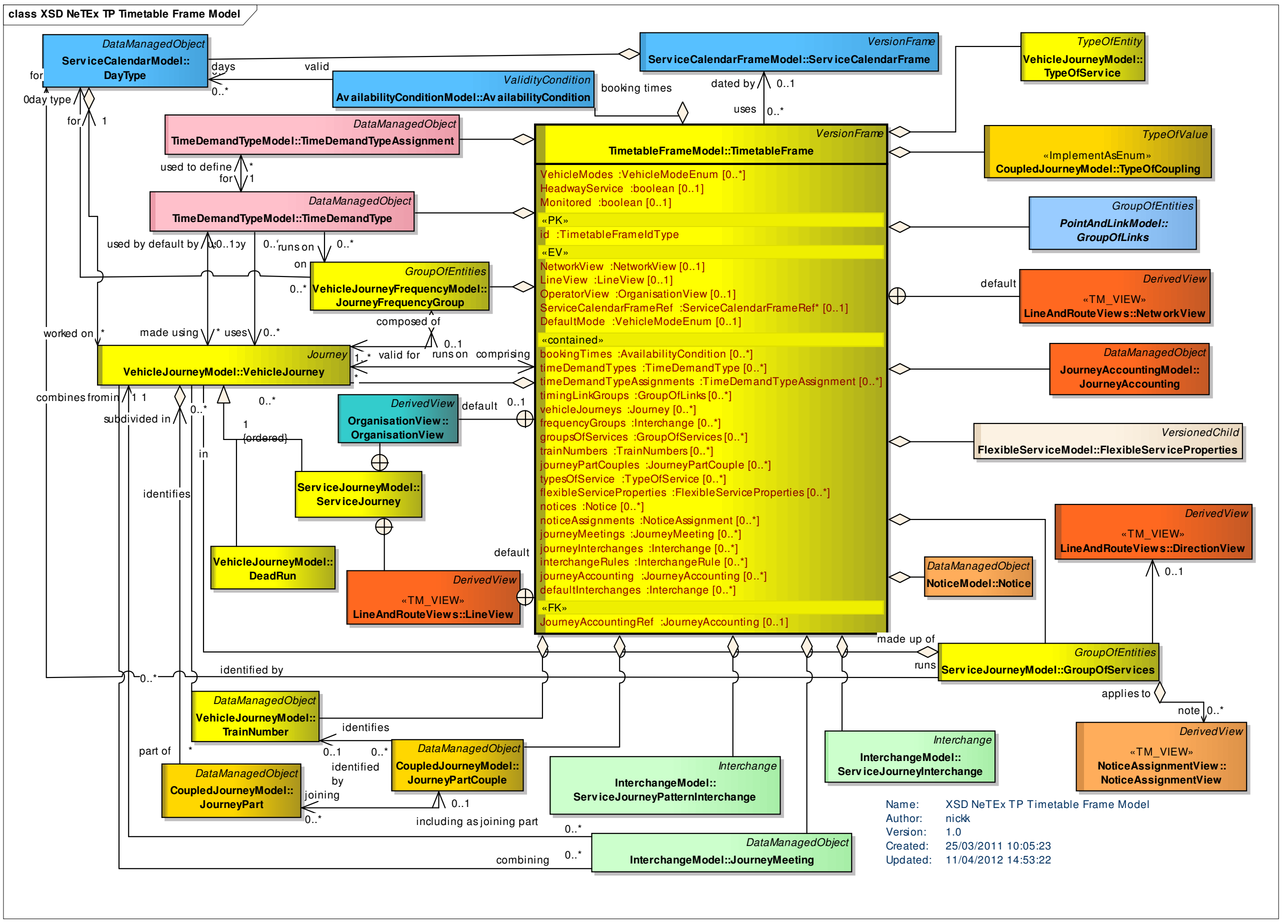

class XSD NeTEx TP Timetable Frame Model
DataManagedObject
ServiceCalendarModel::DayType
VersionFrame
ServiceCalendarFrameModel::ServiceCalendarFrame
TypeOfEntity
VehicleJourneyModel::TypeOfService
ValidityCondition
AvailabilityConditionModel::AvailabilityCondition
days
valid
0..*
booking times
dated by
0..1
uses
0..*
for
0day type
for
1
DataManagedObject
TimeDemandTypeModel::TimeDemandTypeAssignment
used to define
*
for
1
DataManagedObject
TimeDemandTypeModel::TimeDemandType
used by default by
0..1
0..*
runs on
on
0..*
GroupOfEntities
VehicleJourneyFrequencyModel::JourneyFrequencyGroup
composed of
0..1
worked on
*
made using
*
uses
0..*
Journey
VehicleJourneyModel::VehicleJourney
1..*
valid for
runs on
comprising
*
combines fromin
1 1
subdivided in
0..*
0..*
1
{ordered}
in
identifies
DerivedView
OrganisationView::OrganisationView
default
0..1
ServiceJourneyModel::ServiceJourney
VehicleJourneyModel::DeadRun
default
DerivedView
«TM_VIEW»
LineAndRouteViews::LineView
VersionFrame
TimetableFrameModel::TimetableFrame
VehicleModes :VehicleModeEnum [0..*]
HeadwayService :boolean [0..1]
Monitored :boolean [0..1]
«PK»
id :TimetableFrameIdType
«EV»
NetworkView :NetworkView [0..1]
LineView :LineView [0..1]
OperatorView :OrganisationView [0..1]
ServiceCalendarFrameRef :ServiceCalendarFrameRef* [0..1]
DefaultMode :VehicleModeEnum [0..1]
«contained»
bookingTimes :AvailabilityCondition [0..*]
timeDemandTypes :TimeDemandType [0..*]
timeDemandTypeAssignments :TimeDemandTypeAssignment [0..*]
timingLinkGroups :GroupOfLinks [0..*]
vehicleJourneys :Journey [0..*]
frequencyGroups :Interchange [0..*]
groupsOfServices :GroupOfServices [0..*]
trainNumbers :TrainNumbers [0..*]
journeyPartCouples :JourneyPartCouple [0..*]
typesOfService :TypeOfService [0..*]
flexibleServiceProperties :FlexibleServiceProperties [0..*]
notices :Notice [0..*]
noticeAssignments :NoticeAssignment [0..*]
journeyMeetings :JourneyMeeting [0..*]
journeyInterchanges :Interchange [0..*]
interchangeRules :InterchangeRule [0..*]
journeyAccounting :JourneyAccounting [0..*]
defaultInterchanges :Interchange [0..*]
«FK»
JourneyAccountingRef :JourneyAccounting [0..1]
TypeOfValue
«ImplementAsEnum»
CoupledJourneyModel::TypeOfCoupling
GroupOfEntities
PointAndLinkModel::GroupOfLinks
default
DerivedView
«TM_VIEW»
LineAndRouteViews::NetworkView
DataManagedObject
JourneyAccountingModel::JourneyAccounting
VersionedChild
FlexibleServiceModel::FlexibleServiceProperties
DerivedView
«TM_VIEW»
LineAndRouteViews::DirectionView
0..1
DataManagedObject
NoticeModel::Notice
made up of
runs
GroupOfEntities
ServiceJourneyModel::GroupOfServices
applies to
note
0..*
DerivedView
«TM_VIEW»
NoticeAssignmentView::NoticeAssignmentView
identified by
0..*
DataManagedObject
VehicleJourneyModel::TrainNumber
identifies
0..1
0..*
identified by
DataManagedObject
CoupledJourneyModel::JourneyPartCouple
part of
*
DataManagedObject
CoupledJourneyModel::JourneyPart
joining
0..*
0..1
including as joining part
Interchange
InterchangeModel::ServiceJourneyPatternInterchange
Interchange
InterchangeModel::ServiceJourneyInterchange
0..*
combining
0..*
DataManagedObject
InterchangeModel::JourneyMeeting
Name: XSD NeTEx TP Timetable Frame Model
Author: nickk
Version: 1.0
Created: 25/03/2011 10:05:23
Updated: 11/04/2012 14:53:22

**Figure 6 — Timetable Frame – Physical Model Detail (UML)**

### 7.1.1.1.3 Timetable Frame — XSD and attributes

#### 7.1.1.1.3.1 TimetableFrame – Model Element

A set of timetable data (VEHICLE JOURNEYs, etc.) to which the same VALIDITY CONDITIONs have been assigned.

**Table 1 – *TimetableFrame* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *VersionFrame* | ::> | TIMETABLE FRAME inherits from VERSION FRAME. |
| «PK» | ***id*** | *TimetableFrame-IdType* | 1:1 | Identifier of TIMETABLE FRAME. |
| | ***VehicleModes*** | *VehicleModeEnum* | 0:* | Reference to vehicle transport MODEs TIMETABLE |
| | ***HeadwayService*** | *xsd:xsd:boolean* | 0:1 | Whether services of TIMETABLE are operated a headway services. |
| | ***Monitored*** | *xsd:xsd:boolean* | 0:1 | Whether services of TIMETABLE are monitored in real time. |
| «EV» | ***NetworkView*** | *NetworkView* | 0:1 | Reference to default NETWORK for TIMETABLE and derived values of NETWORK. |

| | | | | |
|---|---|---|---|---|
| «EV» | ***LineView*** | *LineView* | 0:1 | Reference to default LINE for TIMETABLE and derived values of LINE. |
| «EV» | ***OperatorView*** | *OrganisationView* | 0:1 | Reference to default OPERATOR for TIMETABLE and derived values of OPERATOR. |
| «EV» | ***ServiceCalendarFrameRef*** | *ServiceCalendarFrameRef* | 0:1 | Reference to default Service CALENDAR for TIMETABLE. |
| «EV» | ***DefaultMode*** | *VehicleModeEnum* | 0:1 | Reference to default Transport MODE for TIMETABLE. |
| «cntd» | ***JourneyAccounting***Ref | *JourneyAccountingRef* | 0:1* | Default JOURNEY ACCOUNTING values for JOURNEYs in frame. |
| «cntd» | ***bookingTimes*** | *AvailabilityCondition* | 0:* | Times at which bookings can be made for the services in the Timetable. |
| «cntd» | ***timeDemandTypes*** | *TimeDemandType* | 0:* | TIME DEMAND TYPEs in the frame. |
| «cntd» | ***timeDemandTypeAssignments*** | *TimeDemandTypeAssignment* | 0:* | TIME DEMAND TYPE ASSIGNMENTS in the frame. |
| «cntd» | ***timingLinkGroups*** | *GroupOfLinks* | 0:* | TIMING LINK GROUPs in the frame. |
| «cntd» | ***vehicleJourneys*** | *Journey* | 0:* | VEHICLE JOURNEYs & SERVICE JOURNEYS s in the frame. |
| *«cntd»* | ***frequencyGroups*** | *FrequencyGroup* | 0:* | FREQUENCY GROUPs in the frame. |
| «cntd» | ***groupsOfServices*** | *GroupOfServices* | 0:* | GROUP OF SERVICEs in the frame. |
| «cntd» | ***journeyPartCouples*** | *JourneyPartCouple* | 0:* | JOURNEY PART COUPLEs in the frame. |
| «cntd» | ***coupledJourneys*** | *CoupledJourney* | 0:* | COUPLED JOURNEYs in the frame. |
| «cntd» | ***serviceFacilitySets*** | *ServiceFacilitySet* | 0:* | SERVICE FACILITY SETs in the frame. |
| «cntd» | ***typesOfService*** | *TypeOfService* | 0:* | TYPEs OF SERVICE in the frame. |
| «cntd» | ***flexibleServiceProperties*** | *FlexibleServiceProperties* | 0:* | FLEXIBLE SERVICE PROPERTIES in the frame. |
| «cntd» | ***notices*** | *Notice* | 0:* | NOTICEs in the frame. |
| «cntd» | ***noticeAssignments*** | *NoticeAssignment* | 0:* | NOTICE ASSIGNMENTs in the frame. |
| «cntd» | ***journeyMeetings*** | *JourneyMeeting* | 0:* | JOURNEY MEETINGs in the frame. |
| «cntd» | ***journeyInterchanges*** | *Interchange* | 0:* | JOURNEY FREQUENCY GROUPs in the frame. |
| «cntd» | ***interchangeRules*** | *InterchangeRule* | 0:* | INTERCHANGE RULEs in the frame. |
| «cntd» | ***journeyAccounting***s | *JourneyAccounting* | 0:* | Default JOURNEY ACCOUNTING values for JOURNEYs in frame. |

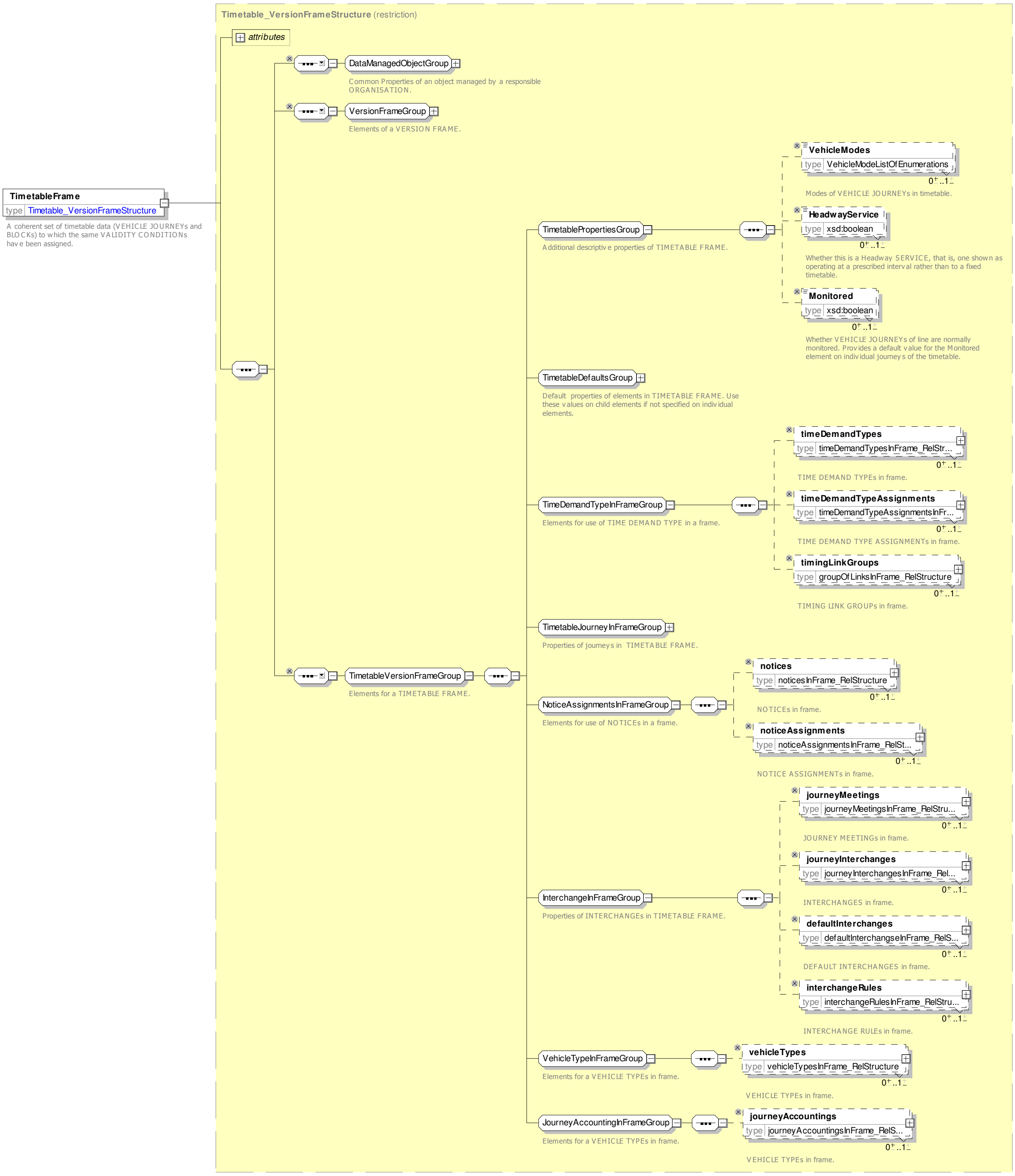
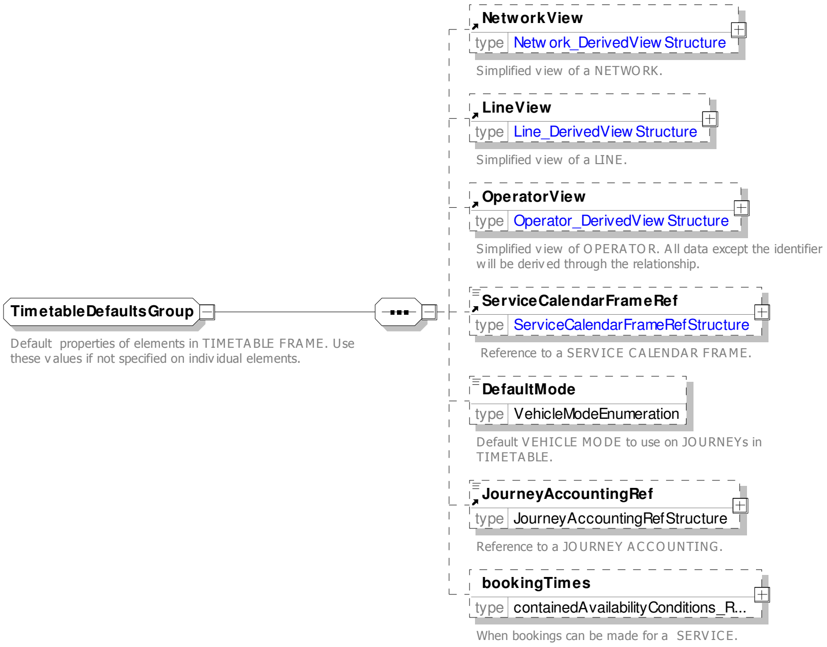

Timetable_VersionFrameStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
VersionFrameGroup
Elements of a VERSION FRAME.
TimetableFrame
type Timetable_VersionFrameStructure
A coherent set of timetable data (VEHICLE JOURNEYs and BLOCKs) to which the same VALIDITY CONDITIONs have been assigned.
TimetablePropertiesGroup
Additional descriptive properties of TIMETABLE FRAME.
VehicleModes
type VehicleModeListOfEnumerations
0..1
Modes of VEHICLE JOURNEYs in timetable.
HeadwayService
type xsd:boolean
0..1
Whether this is a Headway SERVICE, that is, one shown as operating at a prescribed interval rather than to a fixed timetable.
Monitored
type xsd:boolean
0..1
Whether VEHICLE JOURNEYs of line are normally monitored. Provides a default value for the Monitored element on individual journeys of the timetable.
TimetableDefaultsGroup
Default properties of elements in TIMETABLE FRAME. Use these values on child elements if not specified on individual elements.
timeDemandTypes
type timeDemandTypesInFrame_RelStr...
0..1
TIME DEMAND TYPEs in frame.
TimeDemandTypeInFrameGroup
Elements for use of TIME DEMAND TYPE in a frame.
timeDemandTypeAssignments
type timeDemandTypeAssignmentsInFr...
0..1
TIME DEMAND TYPE ASSIGNMENTs in frame.
timingLinkGroups
type groupOfLinksInFrame_RelStructure
0..1
TIMING LINK GROUPs in frame.
TimetableJourneyInFrameGroup
Properties of journeys in TIMETABLE FRAME.
TimetableVersionFrameGroup
Elements for a TIMETABLE FRAME.
notices
type noticesInFrame_RelStructure
0..1
NOTICEs in frame.
NoticeAssignmentsInFrameGroup
Elements for use of NOTICEs in a frame.
noticeAssignments
type noticeAssignmentsInFrame_RelSt...
0..1
NOTICE ASSIGNMENTs in frame.
journeyMeetings
type journeyMeetingsInFrame_RelStru...
0..1
JOURNEY MEETINGs in frame.
journeyInterchanges
type journeyInterchangesInFrame_Rel...
0..1
INTERCHANGEs in frame.
InterchangeInFrameGroup
Properties of INTERCHANGEs in TIMETABLE FRAME.
defaultInterchanges
type defaultInterchangesInFrame_RelS...
0..1
DEFAULT INTERCHANGES in frame.
interchangeRules
type interchangeRulesInFrame_RelStru...
0..1
INTERCHANGE RULEs in frame.
vehicleTypes
type vehicleTypesInFrame_RelStructure
0..1
VehicleTypeInFrameGroup
Elements for a VEHICLE TYPEs in frame.
VEHICLE TYPEs in frame.
JourneyAccountingInFrameGroup
Elements for a VEHICLE TYPEs in frame.
journeyAccountings
type journeyAccountingsInFrame_RelS...
0..1
VEHICLE TYPEs in frame.

Figure 7 — *TimetableFrame* — XSD

### *TimetableFrameDefaultsGroup - Group*



TimetableDefaultsGroup
Default properties of elements in TIMETABLE FRAME. Use these values if not specified on individual elements.
NetworkView
type Network_DerivedViewStructure
Simplified view of a NETWORK.
LineView
type Line_DerivedViewStructure
Simplified view of a LINE.
OperatorView
type Operator_DerivedViewStructure
Simplified view of OPERATOR. All data except the identifier will be derived through the relationship.
ServiceCalendarFrameRef
type ServiceCalendarFrameRefStructure
Reference to a SERVICE CALENDAR FRAME.
DefaultMode
type VehicleModeEnumeration
Default VEHICLE MODE to use on JOURNEYs in TIMETABLE.
JourneyAccountingRef
type JourneyAccountingRefStructure
Reference to a JOURNEY ACCOUNTING.
bookingTimes
type containedAvailabilityConditions_R...
When bookings can be made for a SERVICE.

**Figure 8 — *TimetableFrameDefaultsGroup* — XSD**

***TimetableJourneyInFrameGroup - Group***

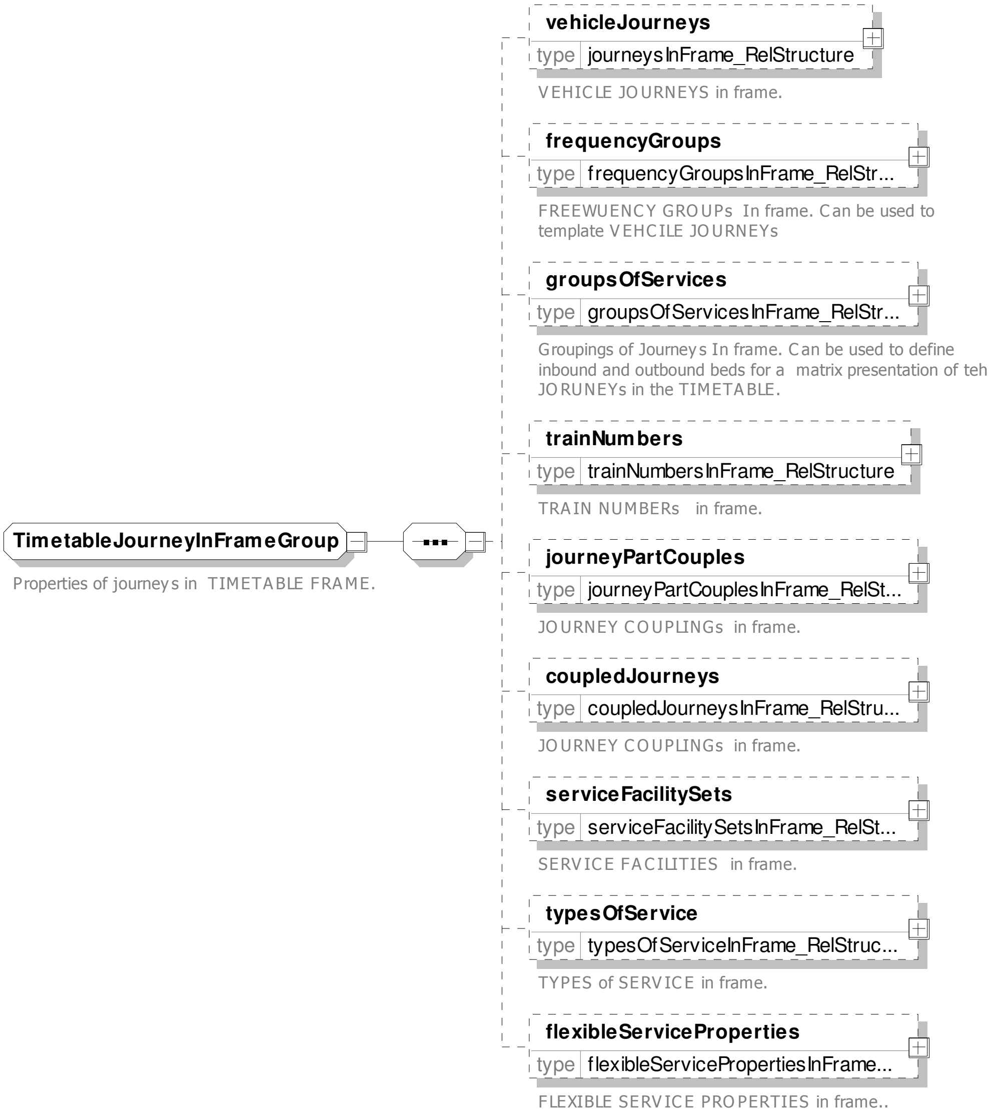

TimetableJourneyInFrameGroup
Properties of journeys in TIMETABLE FRAME.
vehicleJourneys
type journeysInFrame_RelStructure
VEHICLE JOURNEYS in frame.
frequencyGroups
type frequencyGroupsInFrame_RelStr...
FREEWUENCY GROUPs In frame. Can be used to template VEHCILE JOURNEYs
groupsOfServices
type groupsOfServicesInFrame_RelStr...
Groupings of Journeys In frame. Can be used to define inbound and outbound beds for a matrix presentation of teh JORUNEYs in the TIMETABLE.
trainNumbers
type trainNumbersInFrame_RelStructure
TRAIN NUMBERs in frame.
journeyPartCouples
type journeyPartCouplesInFrame_RelSt...
JOURNEY COUPLINGs in frame.
coupledJourneys
type coupledJourneysInFrame_RelStru...
JOURNEY COUPLINGs in frame.
serviceFacilitySets
type serviceFacilitySetsInFrame_RelSt...
SERVICE FACILITIES in frame.
typesOfService
type typesOfServiceInFrame_RelStruc...
TYPES of SERVICE in frame.
flexibleServiceProperties
type flexibleServicePropertiesInFrame...
FLEXIBLE SERVICE PROPERTIES in frame..

**Figure 9 — *TimetableJourneyInFrameGroup* — XSD**

## 7.2 Journey and Journey Times

The JOURNEY and JOURNEY TIMEs model exchanges planned services and dead runs and their timings.

### 7.2.1 Vehicle Journey

#### 7.2.1.1 VEHICLE JOURNEY – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

The daily operation of a vehicle is described by VEHICLE JOURNEYs. A VEHICLE JOURNEY is the defined movement of a vehicle using a specified JOURNEY PATTERN on a particular ROUTE. This movement is hence made between the first and the last POINTs IN JOURNEY PATTERN. 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.

#### 7.2.1.1.1 Basic Vehicle Journey – Conceptual MODEL

There are two different main types of VEHICLE JOURNEYs: passenger-carrying SERVICE JOURNEYs and non-service DEAD RUNs.

— A SERVICE JOURNEY is a VEHICLE JOURNEY on which passengers will be allowed to board or alight from vehicles at stops. These journeys are usually published and known by passengers.

— A DEAD RUN may be necessary for the vehicle to proceed, from the PARKING POINT it was parked at, to the first STOP POINT of the JOURNEY PATTERN where it will start its service operation. On the opposite direction, a DEAD RUN may relate the last STOP POINT the vehicle has stopped at (finishing its service) to the PARKING POINT where it will be parked. A DEAD RUN may also occur when a vehicle changes from one ROUTE to another one in order to continue its service there, or for other various reasons.

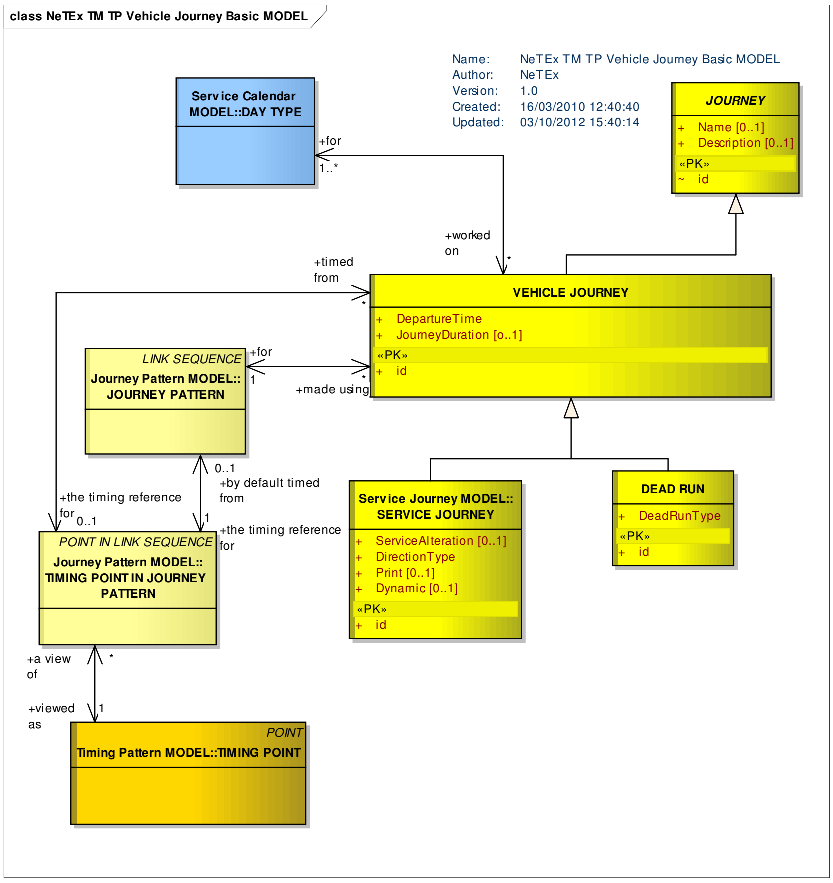

class NeTEx TM TP Vehicle Journey Basic MODEL
Name: NeTEx TM TP Vehicle Journey Basic MODEL
Author: NeTEx
Version: 1.0
Created: 16/03/2010 12:40:40
Updated: 03/10/2012 15:40:14
Service Calendar MODEL::DAY TYPE
JOURNEY
+ Name [0..1]
+ Description [0..1]
«PK»
~ id
+for
1..*
+worked on
*
+timed from
*
VEHICLE JOURNEY
+ DepartureTime
+ JourneyDuration [0..1]
«PK»
+ id
LINK SEQUENCE
Journey Pattern MODEL:: JOURNEY PATTERN
+for
1
*
+made using
0..1
+by default timed from
+the timing reference for
0..1
1
+the timing reference for
POINT IN LINK SEQUENCE
Journey Pattern MODEL:: TIMING POINT IN JOURNEY PATTERN
Service Journey MODEL:: SERVICE JOURNEY
+ ServiceAlteration [0..1]
+ DirectionType
+ Print [0..1]
+ Dynamic [0..1]
«PK»
+ id
DEAD RUN
+ DeadRunType
«PK»
+ id
+a view of
*
+viewed as
1
POINT
Timing Pattern MODEL::TIMING POINT

**Figure 10 — Basic Vehicle Journey – Conceptual MODEL (UML)**

#### 7.2.1.1.2 Vehicle Journey Details – Conceptual MODEL

A VEHICLE JOURNEY may be further defined by a number of other elements, as shown in the following figure. These include interactions with other journeys (JOURNEY PART, JOURNEY MEETING, etc.); temporal and other conditions (DAY TYPE, VALIDITY CONDITION); further descriptive and classification information (TRAIN NUMBER, PRODUCT CATEGORY, TYPE OF SERVICE, stops etc., ); and operational data (BLOCK).

A TEMPLATE JOURNEY allows a set of VEHICLE JOURNEYs to be defined that follow a common temporal pattern – See FREQUENCY GROUPs below.

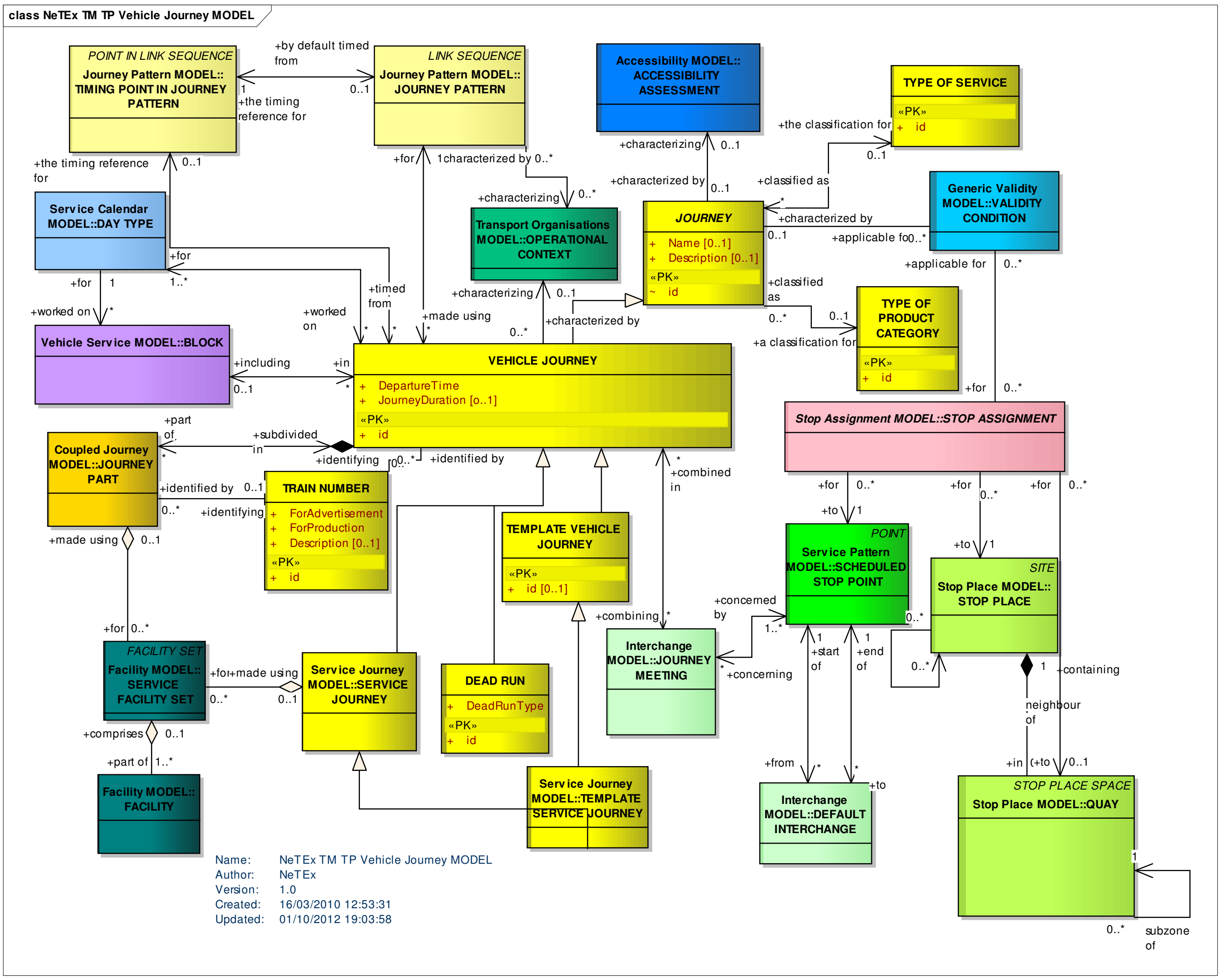


Figure 11 — Vehicle Journey – Conceptual MODEL (UML)

### 7.2.1.2 Vehicle Journey – Physical Model

The VEHICLE JOURNEY physical model adds detailed attributes for VEHICLE JOURNEYs.

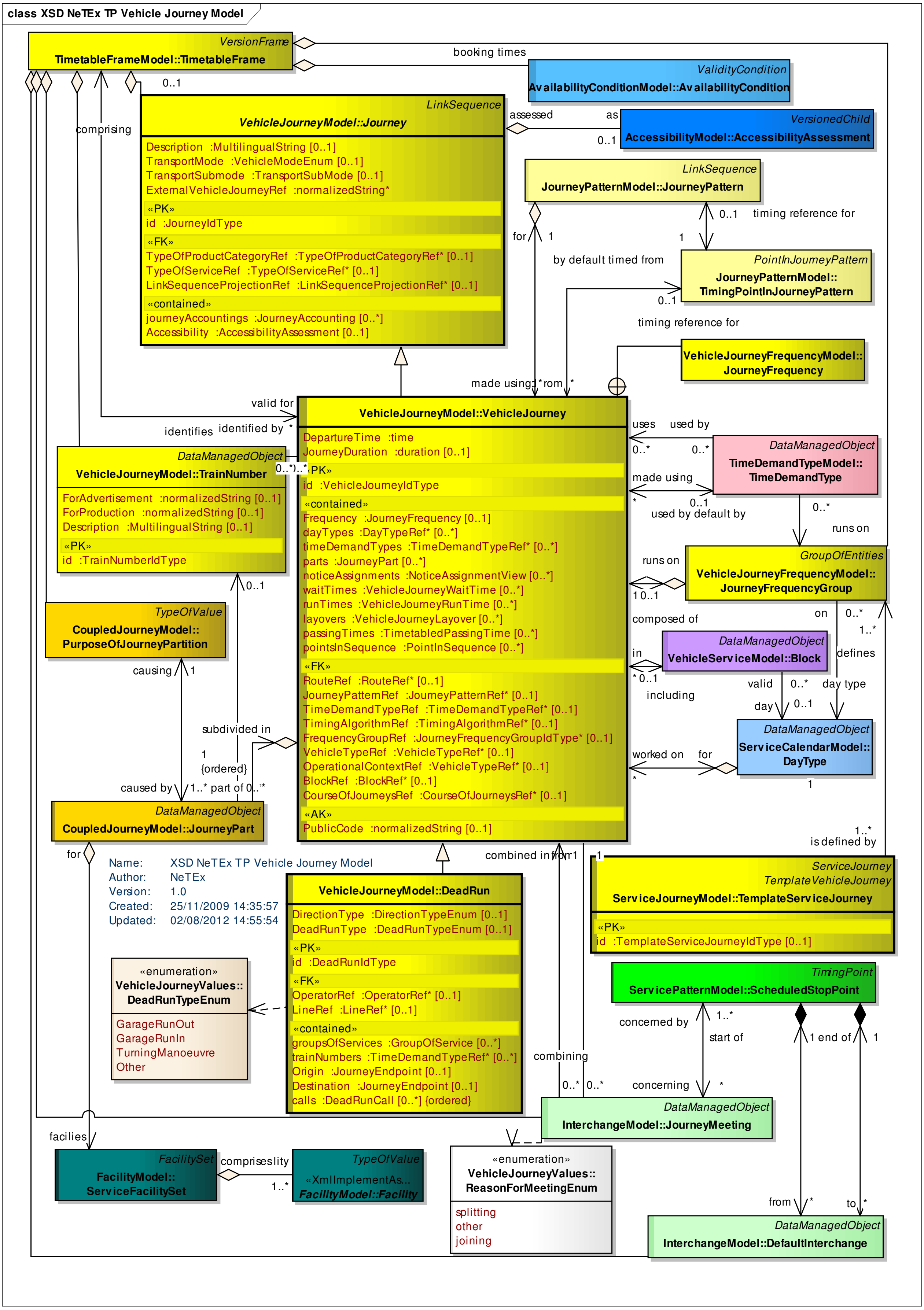

class XSD NeTEx TP Vehicle Journey Model
VersionFrame
TimetableFrameModel::TimetableFrame
booking times
ValidityCondition
AvailabilityConditionModel::AvailabilityCondition
0..1
comprising
LinkSequence
VehicleJourneyModel::Journey
Description :MultilingualString [0..1]
TransportMode :VehicleModeEnum [0..1]
TransportSubmode :TransportSubMode [0..1]
ExternalVehicleJourneyRef :normalizedString*
«PK»
id :JourneyIdType
«FK»
TypeOfProductCategoryRef :TypeOfProductCategoryRef* [0..1]
TypeOfServiceRef :TypeOfServiceRef* [0..1]
LinkSequenceProjectionRef :LinkSequenceProjectionRef* [0..1]
«contained»
journeyAccountings :JourneyAccounting [0..*]
Accessibility :AccessibilityAssessment [0..1]
assessed
as
0..1
VersionedChild
AccessibilityModel::AccessibilityAssessment
LinkSequence
JourneyPatternModel::JourneyPattern
0..1
timing reference for
for
1
1
by default timed from
PointInJourneyPattern
JourneyPatternModel::TimingPointInJourneyPattern
0..1
timing reference for
VehicleJourneyFrequencyModel::JourneyFrequency
made using
*rom
*
valid for
identifies
identified by
*
VehicleJourneyModel::VehicleJourney
DepartureTime :time
JourneyDuration :duration [0..1]
«PK»
id :VehicleJourneyIdType
«contained»
Frequency :JourneyFrequency [0..1]
dayTypes :DayTypeRef* [0..*]
timeDemandTypes :TimeDemandTypeRef* [0..*]
parts :JourneyPart [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
waitTimes :VehicleJourneyWaitTime [0..*]
runTimes :VehicleJourneyRunTime [0..*]
layovers :VehicleJourneyLayover [0..*]
passingTimes :TimetabledPassingTime [0..*]
pointsInSequence :PointInSequence [0..*]
«FK»
RouteRef :RouteRef* [0..1]
JourneyPatternRef :JourneyPatternRef* [0..1]
TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
TimingAlgorithmRef :TimingAlgorithmRef* [0..1]
FrequencyGroupRef :JourneyFrequencyGroupIdType* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
OperationalContextRef :VehicleTypeRef* [0..1]
BlockRef :BlockRef* [0..1]
CourseOfJourneysRef :CourseOfJourneysRef* [0..1]
«AK»
PublicCode :normalizedString [0..1]
DataManagedObject
VehicleJourneyModel::TrainNumber
0..*)..*
ForAdvertisement :normalizedString [0..1]
ForProduction :normalizedString [0..1]
Description :MultilingualString [0..1]
«PK»
id :TrainNumberIdType
uses
used by
0..*
0..*
DataManagedObject
TimeDemandTypeModel::TimeDemandType
made using
*
0..1
used by default by
0..*
runs on
runs on
GroupOfEntities
VehicleJourneyFrequencyModel::JourneyFrequencyGroup
1 0..1
composed of
on
0..*
1..*
defines
DataManagedObject
VehicleServiceModel::Block
in
* 0..1
including
valid
day
0..*
day type
0..1
worked on
for
*
DataManagedObject
ServiceCalendarModel::DayType
1
1..*
is defined by
0..1
TypeOfValue
CoupledJourneyModel::PurposeOfJourneyPartition
causing
1
subdivided in
1
{ordered}
caused by
1..* part of
0..*
DataManagedObject
CoupledJourneyModel::JourneyPart
for
combined in
from
1
1
ServiceJourney
TemplateVehicleJourney
ServiceJourneyModel::TemplateServiceJourney
«PK»
id :TemplateServiceJourneyIdType [0..1]
Name: XSD NeTEx TP Vehicle Journey Model
Author: NeTEx
Version: 1.0
Created: 25/11/2009 14:35:57
Updated: 02/08/2012 14:55:54
VehicleJourneyModel::DeadRun
DirectionType :DirectionTypeEnum [0..1]
DeadRunType :DeadRunTypeEnum [0..1]
«PK»
id :DeadRunIdType
«FK»
OperatorRef :OperatorRef* [0..1]
LineRef :LineRef* [0..1]
«contained»
groupsOfServices :GroupOfService [0..*]
trainNumbers :TimeDemandTypeRef* [0..*]
Origin :JourneyEndpoint [0..1]
Destination :JourneyEndpoint [0..1]
calls :DeadRunCall [0..*] {ordered}
«enumeration»
VehicleJourneyValues::DeadRunTypeEnum
GarageRunOut
GarageRunIn
TurningManoeuvre
Other
TimingPoint
ServicePatternModel::ScheduledStopPoint
concerned by
1..*
start of
1 end of
1
combining
0..*
0..*
concerning
*
DataManagedObject
InterchangeModel::JourneyMeeting
facilies
FacilitySet
FacilityModel::ServiceFacilitySet
comprisesllity
1..*
TypeOfValue
«XmlImplementAs...
FacilityModel::Facility
«enumeration»
VehicleJourneyValues::ReasonForMeetingEnum
splitting
other
joining
from
*
to
*
DataManagedObject
InterchangeModel::DefaultInterchange

Figure 12 — Vehicle Journey – Physical Model (UML)

### 7.2.1.3 Vehicle Journey – Attributes and XSD

#### 7.2.1.3.1 Journey – Model Element

JOURNEY defines common properties of a journey, such as TYPE OF SERVICE, JOURNEY ACCOUNTING, etc.

**Table 2 – *Journey* – Element**

| Classification | Name | | Type | Cardin-ality | Description |
|---|---|---|---|---|---|
| *::>* | *::>* | | *LinkSequence* | *::>* | JOURNEY inherits from LINK SEQUENCE. |
| «PK» | ***id*** | | *JourneyIdType* | 1:1 | Identifier of JOURNEY. |
| | ***Description*** | | *MultilingualString* | 0:1 | Description of JOURNEY. |
| | ***TransportMode*** | | *VehicleModeEnum* | 0:1 | Transport MODE of JOURNEY. |
| | ***TransportSubmode*** | | *TransportSubmode* | 0:1 | Transport Sub MODE of JOURNEY. |
| «AK» | ***ExternalJourney-Ref*** | | *ExternalObjectRef* | 0:1 | An alternative code that uniquely identifies the JOURNEY, specifically for use in AVMS systems.<br>*NOTE For VDV compatibility.* |
| ***«FK»*** | ***TypeOfProduct-CategoryRef*** | | *TypeOfProductCategory*Ref | 0:1 | PRODUCT CATEGORY of a JOURNEY. |
| «FK» | ***TypeOfServiceRef*** | | *TypeOfServiceRef* | 0:1 | TYPE OF SERVICE of JOURNEY. |
| «FK» | **a** | ***LinkSequence ProjectionRef*** | *LinkSequenceProjectionRef* | 0:1 | Reference to LINK SEQUENCE PROJECTION to use to PROJECT JOURNEY onto map, etc. |
| | **b** | ***LinkSequence Projection*** | *LinkSequenceProjection* | 0:1 | LINK SEQUENCE PROJECTION to use to PROJECT JOURNEY onto map, etc. |
| «cntd» | ***journeyAccountings*** | | *JourneyAccounting* | 0:* | JOURNEY ACCOUNTINGs that apply of JOURNEY. |

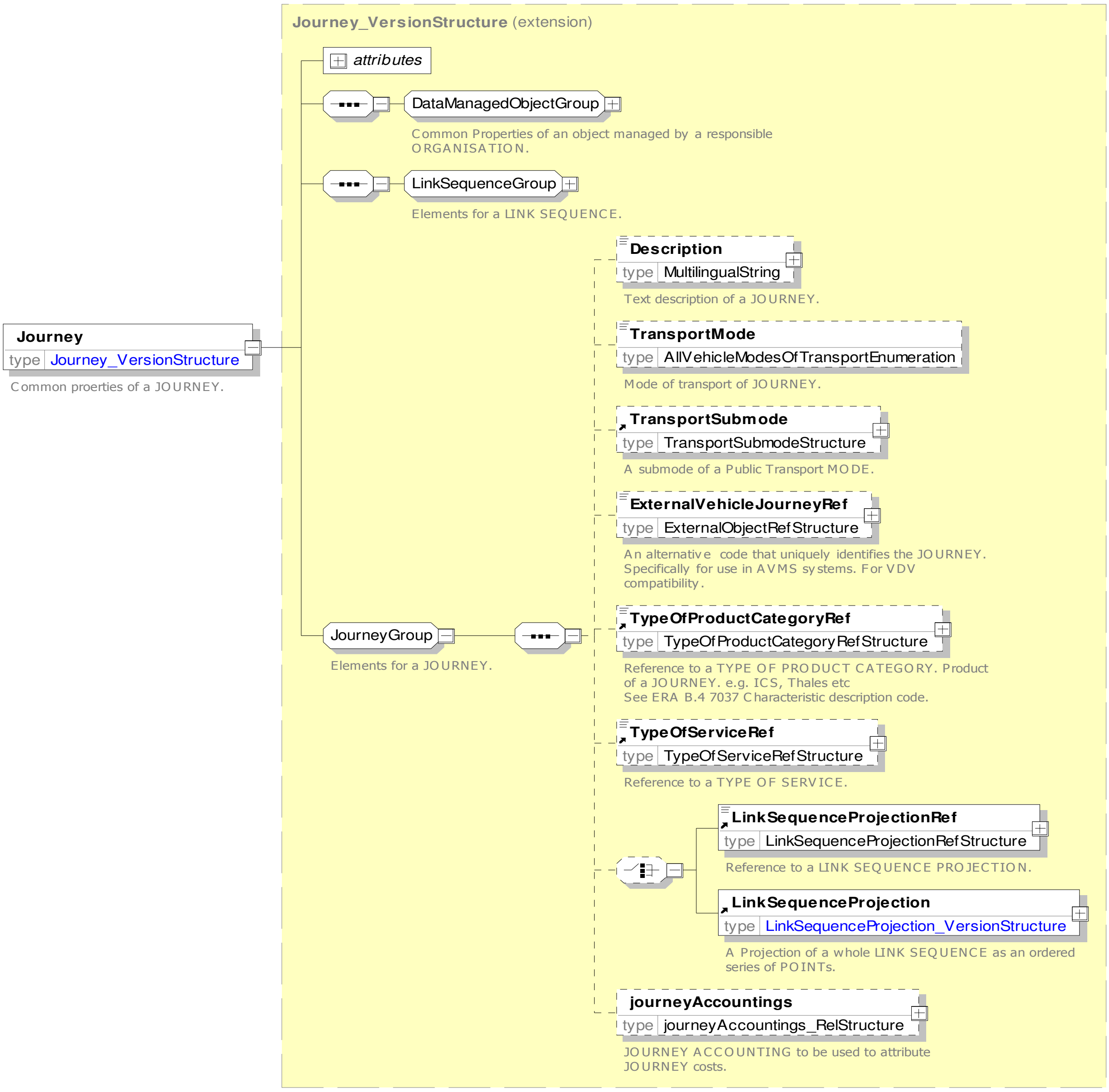

Journey_VersionStructure (extension)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
Description
type MultilingualString
Text description of a JOURNEY.
Journey
type Journey_VersionStructure
Common proerties of a JOURNEY.
TransportMode
type AllVehicleModesOfTransportEnumeration
Mode of transport of JOURNEY.
TransportSubmode
type TransportSubmodeStructure
A submode of a Public Transport MODE.
ExternalVehicleJourneyRef
type ExternalObjectRefStructure
An alternative code that uniquely identifies the JOURNEY. Specifically for use in AVMS systems. For VDV compatibility.
JourneyGroup
Elements for a JOURNEY.
TypeOfProductCategoryRef
type TypeOfProductCategoryRefStructure
Reference to a TYPE OF PRODUCT CATEGORY. Product of a JOURNEY. e.g. ICS, Thales etc
See ERA B.4 7037 Characteristic description code.
TypeOfServiceRef
type TypeOfServiceRefStructure
Reference to a TYPE OF SERVICE.
LinkSequenceProjectionRef
type LinkSequenceProjectionRefStructure
Reference to a LINK SEQUENCE PROJECTION.
LinkSequenceProjection
type LinkSequenceProjection_VersionStructure
A Projection of a whole LINK SEQUENCE as an ordered series of POINTs.
journeyAccountings
type journeyAccountings_RelStructure
JOURNEY ACCOUNTING to be used to attribute JOURNEY costs.

**Figure 13 — *Journey* — XSD**

#### 7.2.1.3.2 VehicleJourney – Model Element

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.

**Table 3 – *VehicleJourney* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *Journey* | ::> | VEHICLE JOURNEY inherits from JOURNEY. |
| «PK» | ***id*** | *VehicleJourney IdType* | 1:1 | Identifier of VEHICLE JOURNEY. |
| | ***DepartureTime*** | *xsd:time* | 1:1 | Departure time of VEHICLE JOURNEY. |
| | ***JourneyDuration*** | *xsd:duration* | 0:1 | Duration of VEHICLE JOURNEY. |
| «cntd» | ***Frequency*** | *JourneyFrequency* | 0:* | Headway frequencies for VEHICLE JOURNEY. |

| | | | | |
|---|---|---|---|---|
| «FK» | ***Frequency-GroupRef*** | *JourneyFrequency-GroupRef* | 0:1 | FREQUENCY GROUP of VEHICLE JOURNEY. |
| «FK» | ***DayTypeRef*** | *DayTypeRef* | 1:* | Reference to DAY TYPE on which VEHICLE JOURNEY runs. |
| «FK» | ***RouteRef*** | *RouteRef* | 0:1 | Reference to ROUTE on which VEHICLE JOURNEY runs. |
| «FK» | ***JourneyPattern-Ref*** | *JourneyPatternRef* | 0:1 | Reference to JOURNEY PATTERN over which VEHICLE JOURNEY runs. |
| «FK» | ***TimeDemandRef*** | *TimeDemandTypeRef* | 0:1 | Reference to TIME DEMAND TYPE during which VEHICLE JOURNEY starts. |
| «FK» | ***Timing-AlgorithmRef*** | *TimingAlgorithmRef* | 0:1 | Reference to TIMING ALGORITHM to use to compute passing times for VEHICLE JOURNEY. |
| «FK» | ***BlockRef*** | *BlockRef* | 0:1 | Reference to BLOCK under which VEHICLE JOURNEY runs. |
| «AK» | ***PublicCode*** | *xsd:normalizedString* | 0:1 | Public Code for VEHICLE JOURNEY. |
| «cntd» | ***layovers*** | *VehicleJourneyLayover* | 0:* | LAYOVERs for VEHICLE JOURNEY. |
| «cntd» | ***waitTimes*** | *VehicleJourneyWaitTime* | 0:* | WAIT TIMEs for VEHICLE JOURNEY. |
| «cntd» | ***runTimes*** | *VehicleJourneyRunTime* | 0:* | RUN TIMEs for VEHICLE JOURNEY. |
| «cntd» | ***passingTimes*** | *TimetabledPassingTime* | 0:* | TIMETABLED PASSING TIMEs for VEHICLE JOURNEY. |
| «cntd» | ***timeDemand-Types*** | *TimeDemandTypeRef* | 0:* | TIME DEMAND TYPEs referenced by VEHICLE JOURNEY. |
| «cntd» | ***parts*** | *JourneyPart* | 0:* | JOURNEY PARTs for VEHICLE JOURNEY. |
| «cntd» | ***notice-Assignments*** | *NoticeAssignmentView* | 0:* | NOTICE ASSIGNMENTs that apply to JOURNEY. |
| «cntd» | ***pointsInSequence*** | *PointInSequence* | 0:* | POINTs in SEQUENCE for VEHICLE JOURNEY. |

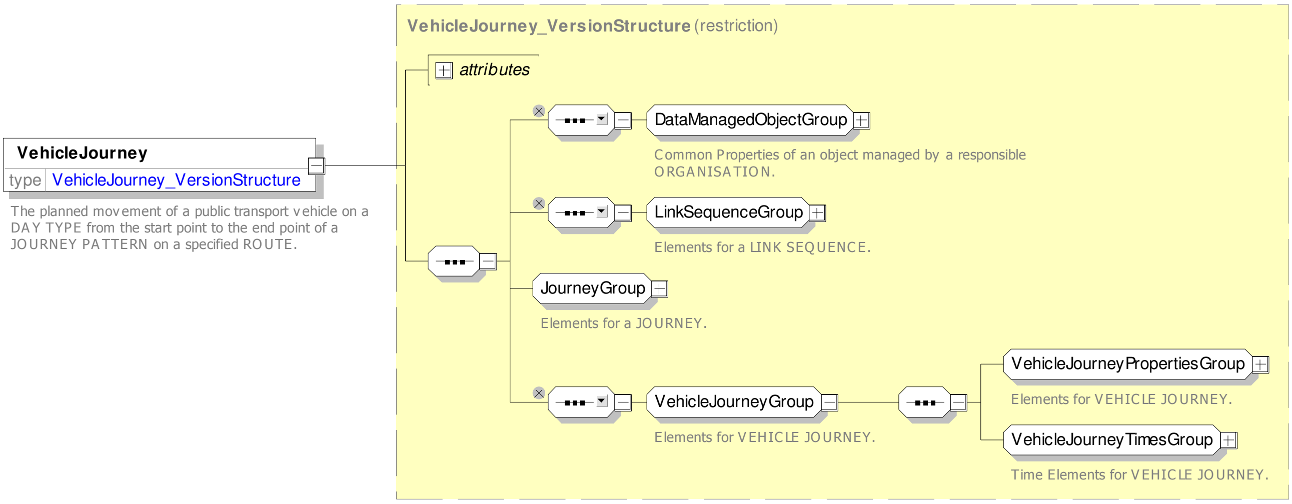

VehicleJourney_VersionStructure (restriction)
attributes
VehicleJourney
type VehicleJourney_VersionStructure
The planned movement of a public transport vehicle on a DAY TYPE from the start point to the end point of a JOURNEY PATTERN on a specified ROUTE.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
JourneyGroup
Elements for a JOURNEY.
VehicleJourneyGroup
Elements for VEHICLE JOURNEY.
VehicleJourneyPropertiesGroup
Elements for VEHICLE JOURNEY.
VehicleJourneyTimesGroup
Time Elements for VEHICLE JOURNEY.

Figure 14 — *VehicleJourney* — XSD

### 7.2.1.3.2.1 *VehicleJourneyReferencesGroup - Group*

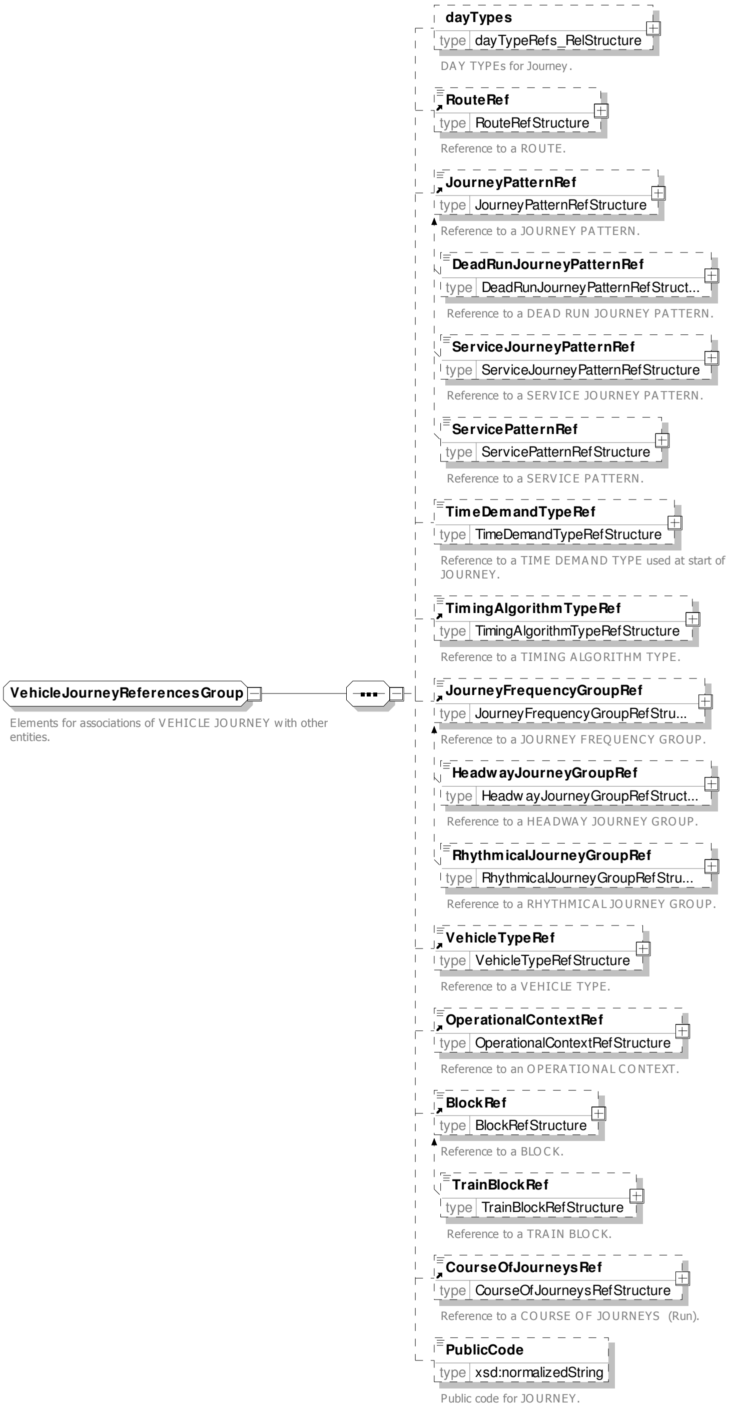

VehicleJourneyReferencesGroup
Elements for associations of VEHICLE JOURNEY with other entities.
dayTypes
type dayTypeRefs_RelStructure
DAY TYPEs for Journey.
RouteRef
type RouteRefStructure
Reference to a ROUTE.
JourneyPatternRef
type JourneyPatternRefStructure
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.
ServiceJourneyPatternRef
type ServiceJourneyPatternRefStructure
Reference to a SERVICE JOURNEY PATTERN.
ServicePatternRef
type ServicePatternRefStructure
Reference to a SERVICE PATTERN.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE used at start of JOURNEY.
TimingAlgorithmTypeRef
type TimingAlgorithmTypeRefStructure
Reference to a TIMING ALGORITHM TYPE.
JourneyFrequencyGroupRef
type JourneyFrequencyGroupRefStru...
Reference to a JOURNEY FREQUENCY GROUP.
HeadwayJourneyGroupRef
type HeadwayJourneyGroupRefStruct...
Reference to a HEADWAY JOURNEY GROUP.
RhythmicalJourneyGroupRef
type RhythmicalJourneyGroupRefStru...
Reference to a RHYTHMICAL JOURNEY GROUP.
VehicleTypeRef
type VehicleTypeRefStructure
Reference to a VEHICLE TYPE.
OperationalContextRef
type OperationalContextRefStructure
Reference to an OPERATIONAL CONTEXT.
BlockRef
type BlockRefStructure
Reference to a BLOCK.
TrainBlockRef
type TrainBlockRefStructure
Reference to a TRAIN BLOCK.
CourseOfJourneysRef
type CourseOfJourneysRefStructure
Reference to a COURSE OF JOURNEYS (Run).
PublicCode
type xsd:normalizedString
Public code for JOURNEY.

**Figure 15 — *VehicleJourneyReferencesGroup* — XSD**

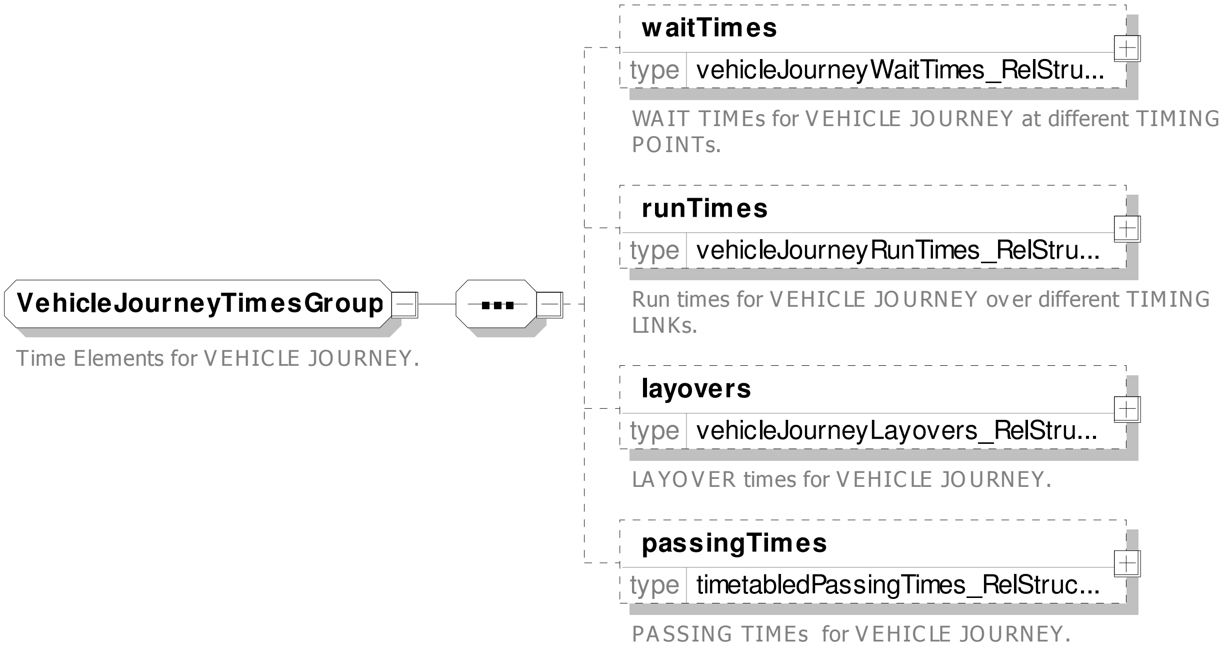

VehicleJourneyTimesGroup
Time Elements for VEHICLE JOURNEY.
waitTimes
type vehicleJourneyWaitTimes_RelStru...
WAIT TIMEs for VEHICLE JOURNEY at different TIMING POINTs.
runTimes
type vehicleJourneyRunTimes_RelStru...
Run times for VEHICLE JOURNEY over different TIMING LINKs.
layovers
type vehicleJourneyLayovers_RelStru...
LAYOVER times for VEHICLE JOURNEY.
passingTimes
type timetabledPassingTimes_RelStruc...
PASSING TIMEs for VEHICLE JOURNEY.

Figure 16 — *VehicleJourneyTimesGroup* — XSD

### 7.2.1.3.3 DeadRun – Model Element

A non-service VEHICLE JOURNEY.

Table 4 – *DeadRun* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *VehicleJourney* | *::>* | DEAD RUN inherits from VEHICLE JOURNEY. |
| «PK» | ***id*** | *DeadRunIdType* | 1:1 | Identifier of DEAD RUN. |
| «FK» | ***VehicleTypeRef*** | *VehicleTypeRef* | 0:1 | Reference to TYPE OF VEHICLE to use on of SERVICE JOURNEY. |
| «EV» | ***OperatorRef*** | *OrganisationRef* | 0:1 | Reference to OPERATOR of SERVICE JOURNEY. |
| «EV» | ***LineRef*** | *LineRef* | 0:1 | Reference to LINE of SERVICE JOURNEY. |
| | ***DirectionType*** | *DirectionTypeEnum* | 0:1 | Type of DIRECTION. |
| «cntd» | ***groupsOfServices*** | *GroupOfServices* | 0:* | GROUP OF SERVICEs to which DEAD RUN belongs. |
| «cntd» | ***trainNumbers*** | *TrainNumber* | 0:* | TRAIN NUMBERs associated with DEAD RUN. |
| «cntd» | ***Origin*** | *JourneyEndpoint* | 0:1 | Origin of DEAD RUN. |
| «cntd» | ***Destination*** | *JourneyEndpoint* | 0:1 | Destination of DEAD RUN. |
| | ***DeadRunType*** | *DeadRunTypeEnum* | 1:1 | Type of DEAD RUN. |
| «cntd» | ***calls*** | *DeadRunCall* | 0:* | CALLs making up DEAD RUN. |

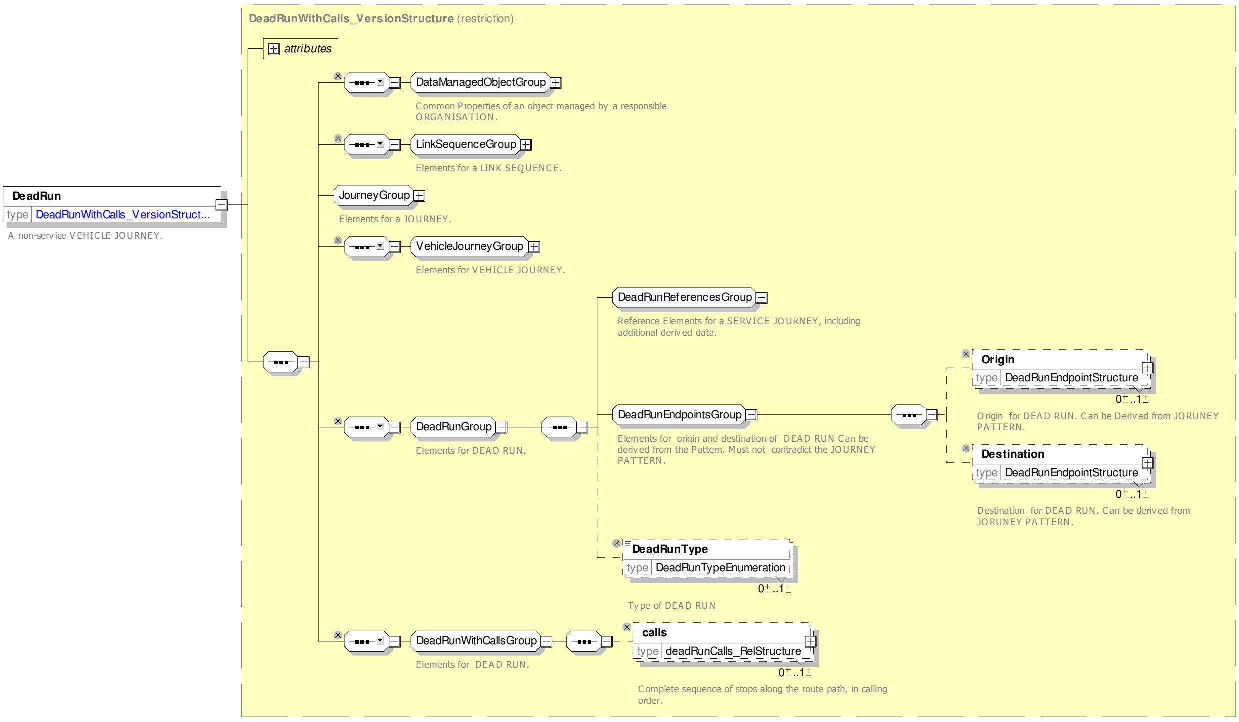

DeadRunWithCalls_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
DeadRun
type DeadRunWithCalls_VersionStruct...
A non-service VEHICLE JOURNEY.
JourneyGroup
Elements for a JOURNEY.
VehicleJourneyGroup
Elements for VEHICLE JOURNEY.
DeadRunReferencesGroup
Reference Elements for a SERVICE JOURNEY, including additional derived data.
Origin
type DeadRunEndpointStructure
0..1
Origin for DEAD RUN. Can be Derived from JORUNEY PATTERN.
DeadRunGroup
Elements for DEAD RUN.
DeadRunEndpointsGroup
Elements for origin and destination of DEAD RUN Can be derived from the Pattern. Must not contradict the JOURNEY PATTERN.
Destination
type DeadRunEndpointStructure
0..1
Destination for DEAD RUN. Can be derived from JORUNEY PATTERN.
DeadRunType
type DeadRunTypeEnumeration
0..1
Type of DEAD RUN
DeadRunWithCallsGroup
Elements for DEAD RUN.
calls
type deadRunCalls_RelStructure
0..1
Complete sequence of stops along the route path, in calling order.

**Figure 17 — *DeadRun* — XSD**

### 7.2.1.3.3.1 *DeadRunReferencesGroup — Group*

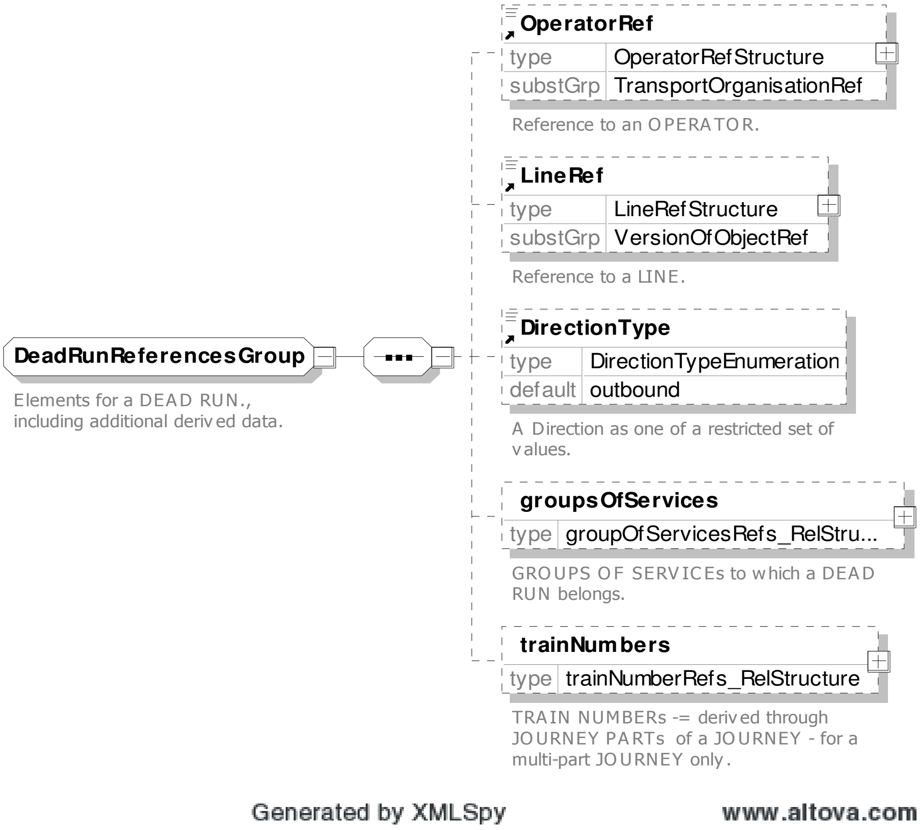

OperatorRef
type OperatorRefStructure
substGrp TransportOrganisationRef
Reference to an OPERATOR.
LineRef
type LineRefStructure
substGrp VersionOfObjectRef
Reference to a LINE.
DeadRunReferencesGroup
Elements for a DEAD RUN., including additional derived data.
DirectionType
type DirectionTypeEnumeration
default outbound
A Direction as one of a restricted set of values.
groupsOfServices
type groupOfServicesRefs_RelStru...
GROUPS OF SERVICEs to which a DEAD RUN belongs.
trainNumbers
type trainNumberRefs_RelStructure
TRAIN NUMBERs -= derived through JOURNEY PARTs of a JOURNEY - for a multi-part JOURNEY only.
Generated by XMLSpy
www.altova.com

**Figure 18 — *DeadRunReferencesGroup* — XSD**

#### 7.2.1.3.3.1 DeadRunType – Allowed values

The following figure shows the allowed values for ***DeadRunType** (DeadRunTypeEnum).*

**Table 5 – *DeadRunType* – Allowed values**

| Value | Description |
|---|---|
| *garageRunOut* | Garage run out |
| *garageRunIn* | Garage run in |
| *turningManoeuvre* | Turning manoeuvre |
| *other* | Other |

#### 7.2.1.3.4 TemplateVehicleJourney – Model Element

A repeating VEHICLE JOURNEY for which a frequency has been specified, either as a HEADWAY JOURNEY GROUP (e.g. every 20 minutes) or a RYTHMICAL JOURNEY GROUP (e.g. at 15, 27 and 40 minutes past the hour) has been specified. It may thus represent multiple journeys or it could also be used simply as a template for adding single extra DATED VEHICLE JOURNEYs after the planning phase.

The frequency of the journey could be defined by a FREQUENCY GROUP*. See later.*

The TEMPLATE VEHICLE JOURNEY of a RYTHMICAL JOURNEY GROUP needs passing times, where only the minute part is meaningful (passing times are repeated every hour with the same minutes), therefore the hour part of passing times will be ignored (a "00" value is recommended)

**Table 6 – *TemplateVehicleJourney* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *Journey* | ::> | TEMPLATE VEHICLE JOURNEY inherits from JOURNEY. |
| «PK» | ***id*** | *TemplateVehicle-JourneyIdType* | 0:1 | Identifier of TEMPLATE VEHICLE JOURNEY. |
| | ***TemplateVehicle-JourneyType*** | *TemplateVehicle-JourneyTypeEnum* | 0:1 | Type of TEMPLATE VEHICLE JOURNEY. |
| «cntd» | ***frequencyGroups*** | *JourneyFrequencyGroup* | 0:* | FREQUENCY GROUPs defining TEMPLATE VEHICLE JOURNEY. |

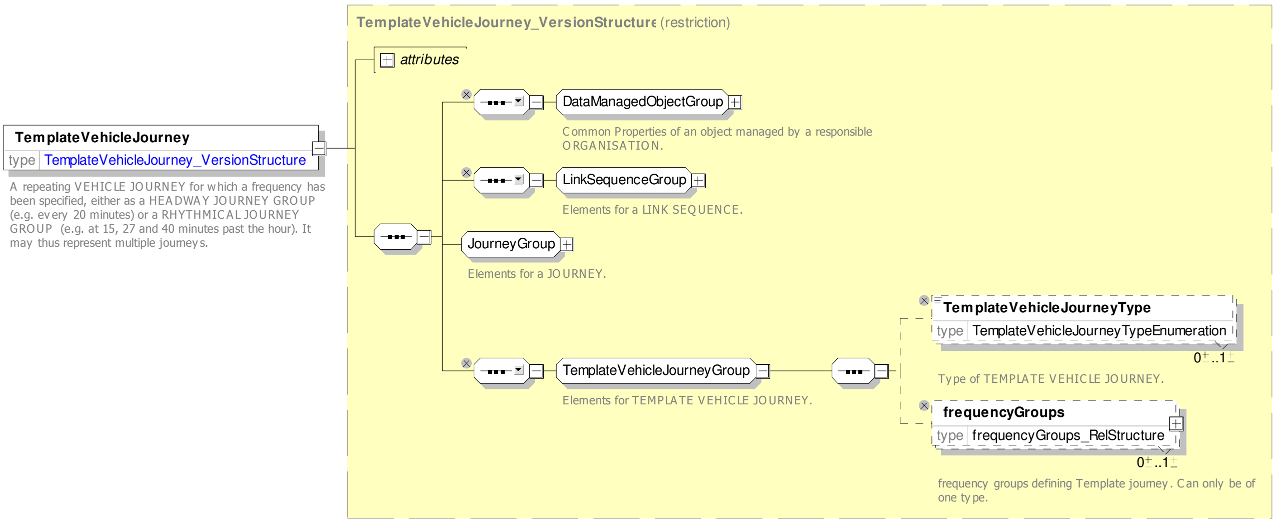

TemplateVehicleJourney_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
JourneyGroup
Elements for a JOURNEY.
TemplateVehicleJourneyGroup
Elements for TEMPLATE VEHICLE JOURNEY.
TemplateVehicleJourneyType
type TemplateVehicleJourneyTypeEnumeration
0..1
Type of TEMPLATE VEHICLE JOURNEY.
frequencyGroups
type frequencyGroups_RelStructure
0..1
frequency groups defining Template journey. Can only be of one type.
TemplateVehicleJourney
type TemplateVehicleJourney_VersionStructure
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.

**Figure 19 — *TemplateVehicleJourney* — XSD**

#### 7.2.1.3.4.1 TemplateVehicleJourneyType – Allowed values

The following table shows the allowed values for **TemplateVehicleJourneyType (**TemplateVehicleJourneyTypeEnum).

**Table 7 – TemplateVehicleJourneyType – Allowed values**

| Value | Description |
|---|---|
| *headway* | Journeys run at intervals |
| *rhythmic* | Journeys run according to a pattern of start times. |
| *other* | Other |

### 7.2.1.3.5 TypeOfService – Model Element

A classification for VEHICLE JOURNEYs and SPECIAL SERVICEs to express some common properties of journeys to be taken into account in the scheduling and/or operations control process.

**Table 8 – *TypeOfService* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *TypeOfEntity* | *::>* | TYPE OF SERVICE inherits from TYPE OF ENTITY. |
| «PK» | ***id*** | *TypeOfServiceIdType* | 1:1 | Identifier of TYPE OF SERVICE. |

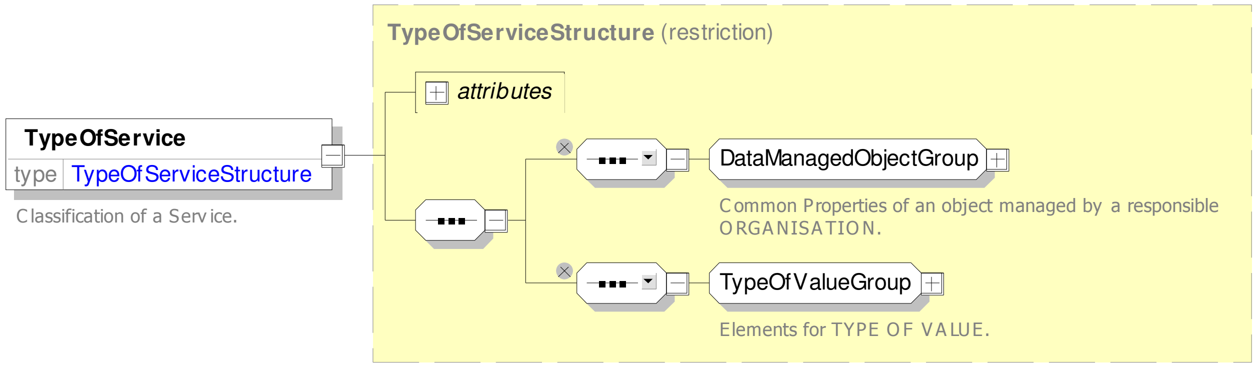

TypeOfServiceStructure (restriction)
attributes
TypeOfService
type TypeOfServiceStructure
Classification of a Service.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
TypeOfValueGroup
Elements for TYPE OF VALUE.

**Figure 20 — *TypeOfService* — XSD**

### 7.2.1.3.6 TypeOfProductCategory – Model Element

A classification for VEHICLE JOURNEYs to express some common properties of journeys for marketing and fare products.

**Table 9 – TypeOfProductCategory – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *TypeOfValue* | ::> | TYPE OF PRODUCT CATEGORY inherits from TYPE OF VALUE. |
| «PK» | ***id*** | *TypeOfProductCategoryIdType* | 1:1 | Identifier of TYPE OF PRODUCT CATEGORY. |

### 7.2.1.3.7 TrainNumber – Model Element

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 10 – *TrainNumber* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | TIME DEMAND PROFILE inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *TrainNumberIdType* | 1:1 | Identifier of TRAIN NUMBER. |
| | ***ForAdvertisement*** | *xsd:normalizedString* | 0:1 | TRAIN NUMBER to use for advertisement to public if different from ID. |
| | ***ForProduction*** | *xsd:normalizedString* | 0:1 | TRAIN NUMBER to use for production purposes, for instance towards technical systems that require an odd or even value according to safety regulations, if different from ID. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of TRAIN NUMBER. |

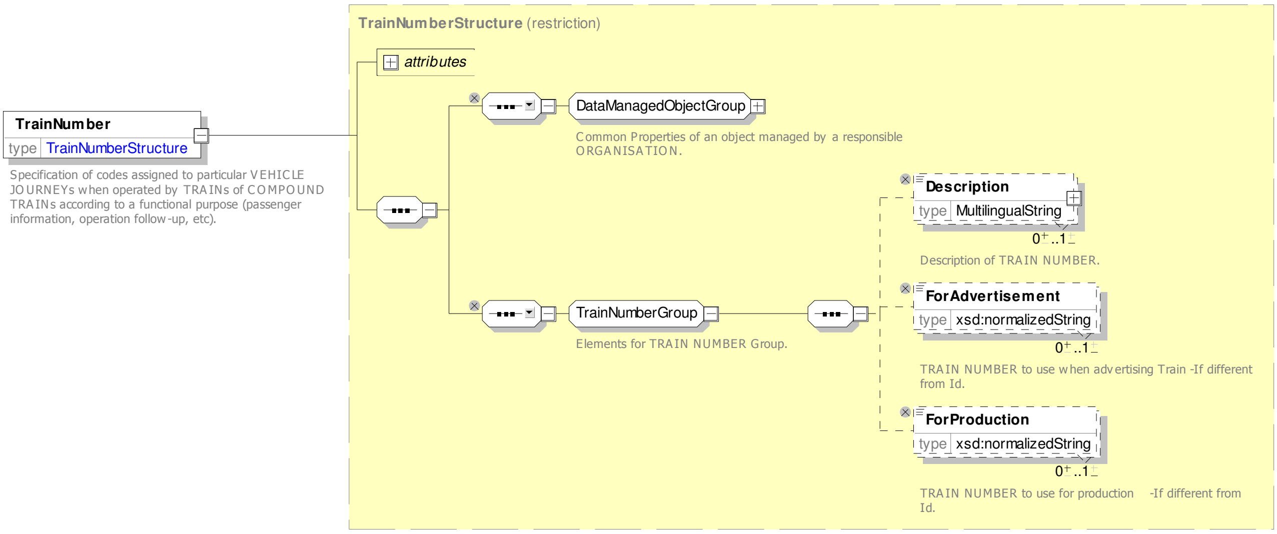

TrainNumberStructure (restriction)
attributes
TrainNumber
type TrainNumberStructure
Specification of codes assigned to particular VEHICLE JOURNEYs when operated by TRAINs of COMPOUND TRAINs according to a functional purpose (passenger information, operation follow-up, etc).
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
TrainNumberGroup
Elements for TRAIN NUMBER Group.
Description
type MultilingualString
0..1
Description of TRAIN NUMBER.
ForAdvertisement
type xsd:normalizedString
0..1
TRAIN NUMBER to use when advertising Train -If different from Id.
ForProduction
type xsd:normalizedString
0..1
TRAIN NUMBER to use for production -If different from Id.

**Figure 21 — *TrainNumber* — XSD**

### 7.2.1.3.8 JourneyDesignator – Model Element

A set of descriptive attributes of a JOURNEY other than its identifier that may be used to identify it. This is of use when describing journeys from a foreign system for which the identifiers are not known

**Table 11 – *JourneyDesignator* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| «FK» | ***FromPointRef*** | *TimingPointRef* | 0:1 | Origin TIMING POINT to use to identify a JOURNEY. |
| «FK» | ***ToPointRef*** | *TimingPointRef* | 0:1 | Destination TIMING POINT to use to identify JOURNEY. |
| | ***DepartureTime*** | *xsd:time* | 0:1 | Departure time of VEHICLE JOURNEY from origin point. |
| | ***ArrivalTime*** | *xsd:time* | 0:1 | Arrival time of VEHICLE JOURNEY at destination point. |
| «AK» | ***Alternative-JourneyRef*** | *xsd:normalizedString* | 0:1 | Alternative identifier of JOURNEY. May possibly be unique only in a limited context. |
| «FK» | ***DayTypeRef*** | *DayTypeRef* | 0:1 | DAY TYPE to use to identify JOURNEY. |

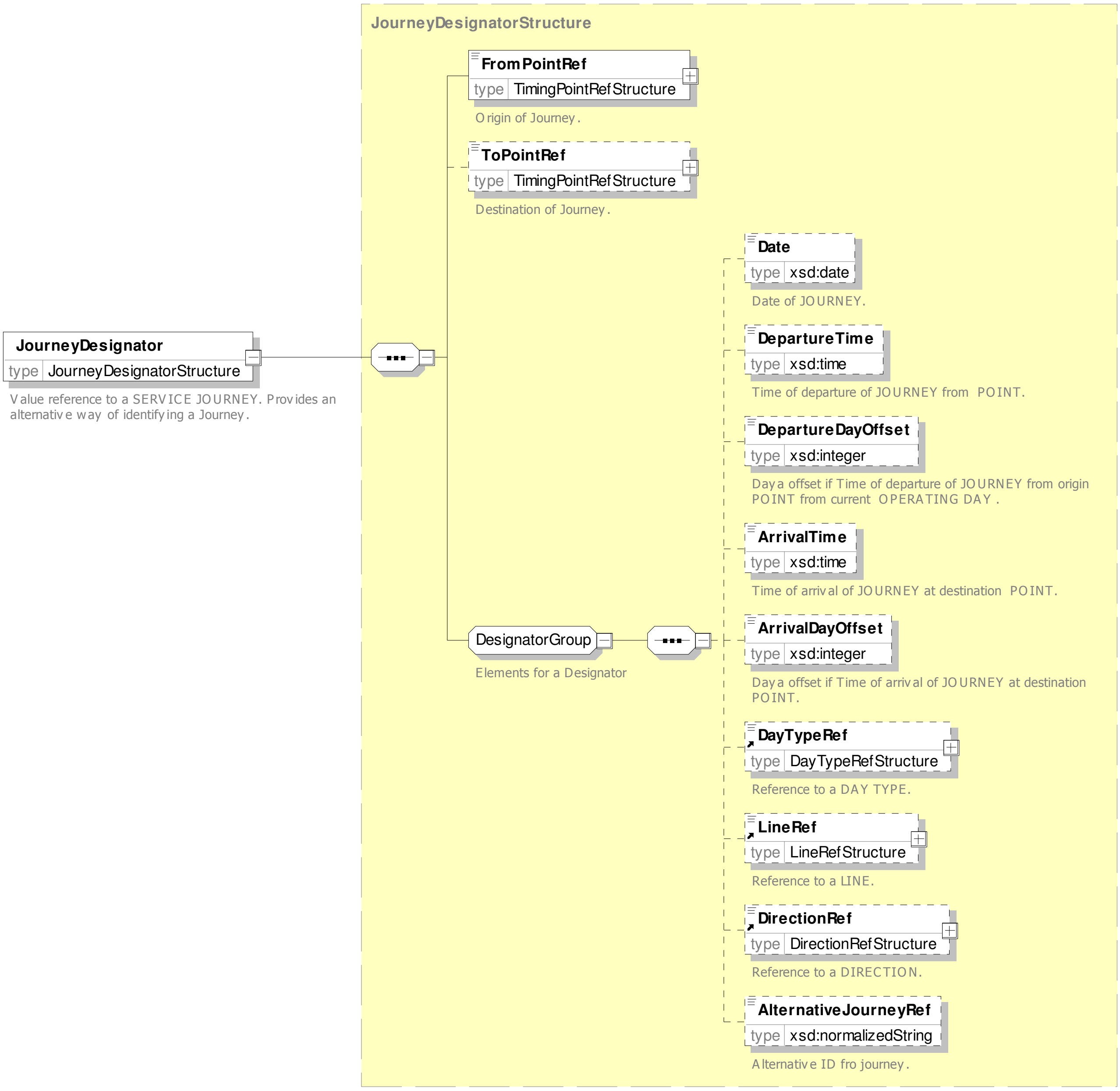

JourneyDesignatorStructure
FromPointRef
type TimingPointRefStructure
Origin of Journey.
ToPointRef
type TimingPointRefStructure
Destination of Journey.
Date
type xsd:date
Date of JOURNEY.
DepartureTime
type xsd:time
Time of departure of JOURNEY from POINT.
DepartureDayOffset
type xsd:integer
Daya offset if Time of departure of JOURNEY from origin POINT from current OPERATING DAY.
ArrivalTime
type xsd:time
Time of arrival of JOURNEY at destination POINT.
ArrivalDayOffset
type xsd:integer
Daya offset if Time of arrival of JOURNEY at destination POINT.
DayTypeRef
type DayTypeRefStructure
Reference to a DAY TYPE.
LineRef
type LineRefStructure
Reference to a LINE.
DirectionRef
type DirectionRefStructure
Reference to a DIRECTION.
AlternativeJourneyRef
type xsd:normalizedString
Alternative ID fro journey.
DesignatorGroup
Elements for a Designator
JourneyDesignator
type JourneyDesignatorStructure
Value reference to a SERVICE JOURNEY. Provides an alternative way of identifying a Journey.

Figure 22 — *JourneyDesignator* — XSD

### 7.2.1.4 XML Examples of Vehicle Journey

See SERVICE JOURNEY for examples of a VEHICLE JOURNEY.

#### 7.2.1.4.1 Dead Run XML Example with passing times and calls

The following code fragment shows a DEAD RUN to position a vehicle at the first stop of a service.

For EXAMPLE

```
<DeadRun version="any" id="xj:dr_L24_01o">
    <Name>1st Dead Run out from Garage in Day service</Name>
    <DepartureTime>11:00:00.0Z</DepartureTime>
    <dayTypes>
        <DayTypeRef version="any" ref="xj:DT_01"/>
    </dayTypes>
    <RouteRef version="any" ref="kx:rt_24o"/>
```

```
        <DeadRunJourneyPatternRef version="any" ref="kx:drjp_L24_01o"/>
        <BlockRef version="any" ref="ops:blk_L24_01"/>
        <passingTimes>
            <TimetabledPassingTime id="kx:tpt_dr_L24_01o_001">
                <TimingPointInJourneyPatternRef version="any" ref="kx:pijp_drjp_L24_01o_001"/>
                <DepartureTime>11:00:00.0Z</DepartureTime>
            </TimetabledPassingTime>
            <TimetabledPassingTime id="kx:tpt_dr_L24_01o_002">
                <StopPointInJourneyPatternRef version="any" ref="kx:pijp_drjp_L24_01o_002"/>
                <ArrivalTime>11:10:00.0Z</ArrivalTime>
                <WaitingTime>PT5M</WaitingTime>
            </TimetabledPassingTime>
        </passingTimes>
        <DirectionType>outbound</DirectionType>
        <Origin>
            <PointRef version="any" ref="mybus:grgpt1"/>
        </Origin>
        <Destination>
            <PointRef version="any" ref="mybus:ssp_001"/>
        </Destination>
        <DeadRunType>garageRunOut</DeadRunType>
    </DeadRun>
```

### 7.2.2 Service Journey

#### 7.2.2.1 SERVICE JOURNEY – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

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.

In addition to the distinction between SERVICE JOURNEYs and DEAD RUNs, operators may wish to classify VEHICLE JOURNEYs by further criteria. For these purposes a TYPE OF SERVICE may be assigned to a VEHICLE JOURNEY, which would express other common properties (e.g. “journey at the maximum load period”).

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. The proposed VEHICLE TYPE will preferably be taken into account by the scheduling algorithm used to compile blocks of vehicle operation. The choice of such a preference may take into account a ranked list of VEHICLE TYPEs for each SERVICE JOURNEY PATTERN. This is described by the entity VEHICLE TYPE PREFERENCE, depending on a particular SERVICE JOURNEY PATTERN, for which a priority ‘rank’ is given for each VEHICLE TYPE, for each DAY TYPE and TIME DEMAND TYPE.

SERVICE JOURNEY INTERCHANGE (the scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs) occurring on the SERVICE JOURNEY and facilities (grouped in a SERVICE FACILITY SET) available for the SERVICE JOURNEY, are also important related information, especially when advertising to the public.

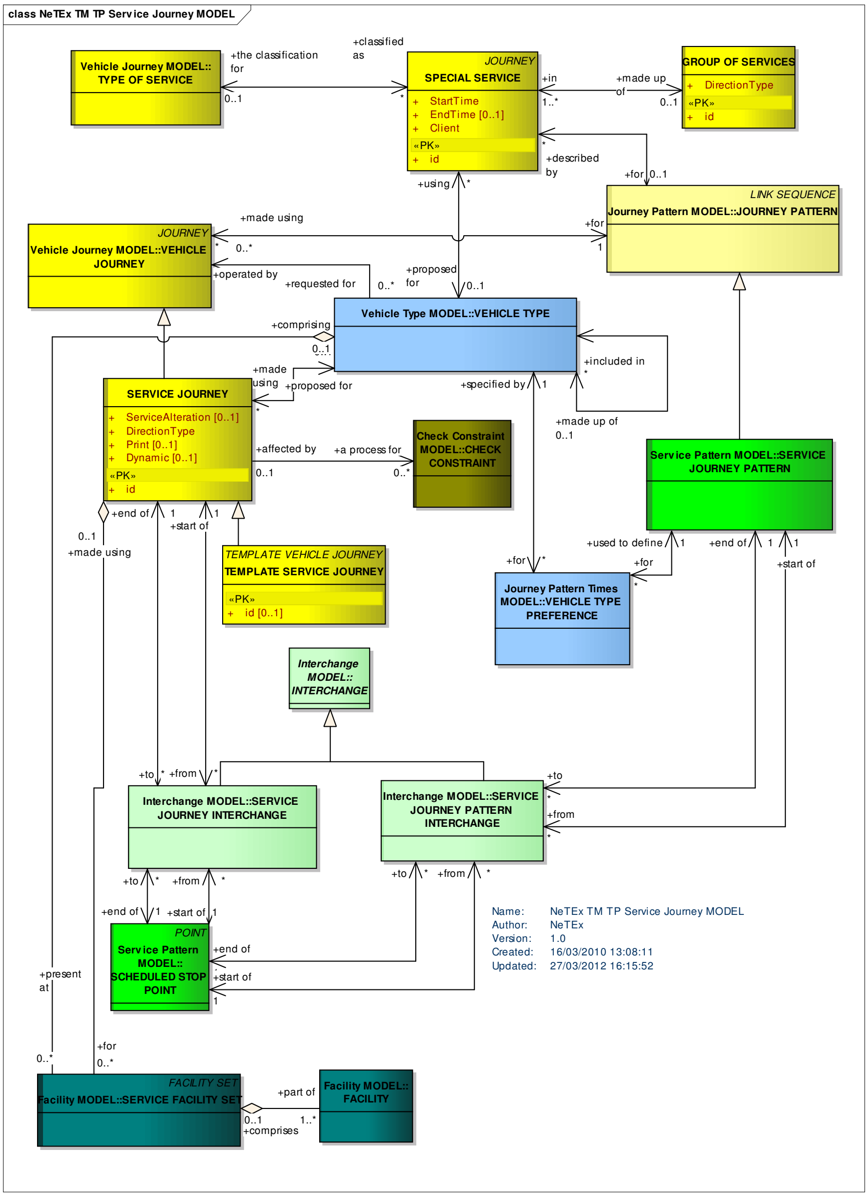


Figure 23 — Service Journey – Conceptual MODEL (UML)

### 7.2.2.2 Special Services

Most of public transport services are operated in a classical way, on a LINE grouping two or more SERVICE JOURNEY PATTERNs, along which passengers may board or alight at fixed stop points, paying fares according to the fare system in use. However, some other types of service may be offered (school services, occasional services, demand-responsive services, etc.). They are usually named “special” services.

The differences between classical services and special services may refer to a number of different aspects, the most important being that the access rights to SPECIAL SERVICEs may differ from the others. Besides occasional services for which the usual fare system applies (e.g. a football match), there are other services using a different system: special fares, access restricted to certain population groups (e.g. pupils from a particular school), etc. In some cases, the service is not directly ordered by the authority in charge of the classical services, but by another authority or by a particular client (which for instance books a bus for a trip or a whole day). Special services are generally not planned in a schedule designed for a DAY TYPE, though it may be the case for very regular services (e.g. school service planned between two SERVICE

JOURNEYs). More often, they are added to the production plan for each particular operating day, according to the requirements for that day. The mission for a special service is usually not described using classical ROUTEs and JOURNEY PATTERNs. Regular special services may have only a rough indication of their origin and destination, which is the case for most occasional services. Some places may play the role of STOP POINT for a special service but are not registered as such by the operator, because no equipment (post or shelter) is installed, etc.

The entity SPECIAL SERVICE describes a service that is not operated in the classical way, i.e. differing from a VEHICLE JOURNEY by some characteristics. A SPECIAL SERVICE is a regular service planned for a DAY TYPE. It has as main attributes a ‘start time’ and an ‘end time’.

A SPECIAL SERVICE is characterised by a TYPE OF SERVICE, which allows distinguishing various categories of services, according to the needs of the user. This classification allows as well distinguishing the DEAD RUNs necessary to perform the sold services. SPECIAL SERVICEs are usually grouped into GROUPs OF SERVICES, which may be sometimes advertised (e.g. the school services may have a published number).

The mission corresponding to a special service may be described by a JOURNEY PATTERN, as classical VEHICLE JOURNEYs. This is the case for some relatively frequent services, using normal STOP POINTs (e.g. service for football match). More frequently, they are only described by a ROUTE, often reduced to origin and a destination POINTs. However, as these end points shall be TIMING POINTs, the mission of a SPECIAL SERVICE is described by a simplified JOURNEY PATTERN.

### 7.2.2.3 Service Journey – Physical Model

SERVICE JOURNEY is used extensively to provide passenger information about journeys. As well as having a number of immediate attributes it is also used to assemble a number of derived attributes that are useful for presentation and that are available through relationships to related elements such as LINE, OPERATOR and JOURNEY PATTERN. As such the SERVICE JOURNEY in the physical model is in effect a view.

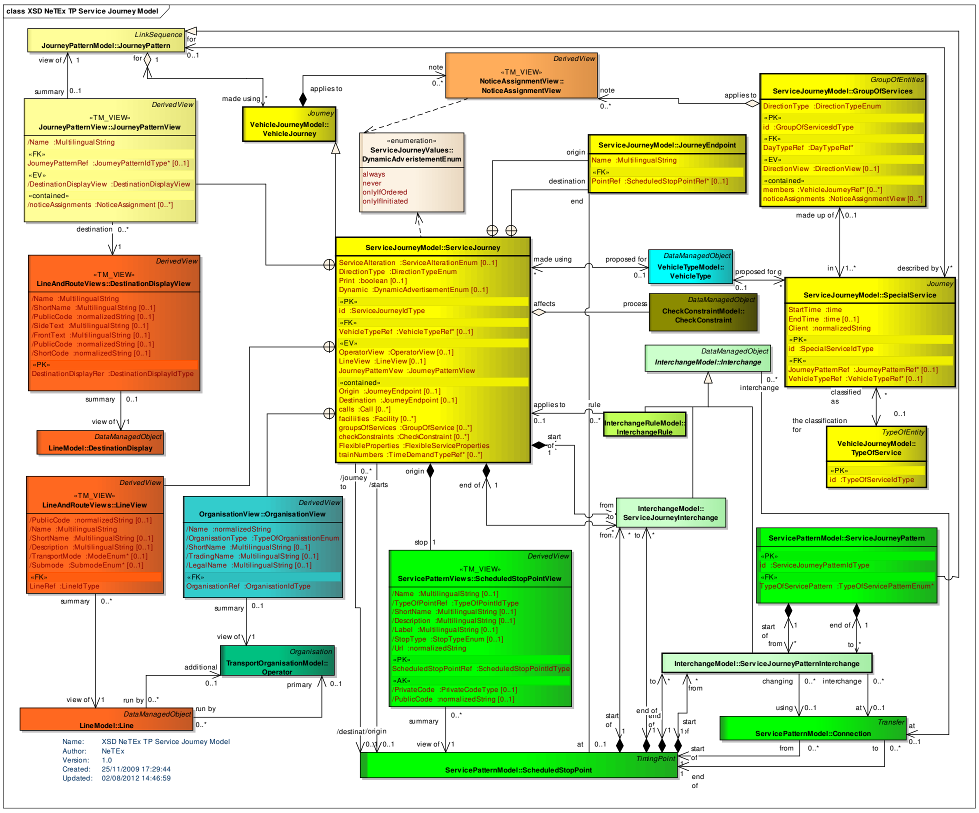

class XSD NeTEx TP Service Journey Model
LinkSequence
JourneyPatternModel::JourneyPattern
view of 1
for 1
for 0..1
summary 0..1
DerivedView
«TM_VIEW»
JourneyPatternView::JourneyPatternView
/Name :MultilingualString
«FK»
JourneyPatternRef :JourneyPatternIdType* [0..1]
«EV»
/DestinationDisplayView :DestinationDisplayView
«contained»
/noticeAssignments :NoticeAssignment [0..*]
destination 0..*
1
made using *
applies to
Journey
VehicleJourneyModel::
VehicleJourney
note 0..*
DerivedView
«TM_VIEW»
NoticeAssignmentView::
NoticeAssignmentView
note 0..*
«enumeration»
ServiceJourneyValues::
DynamicAdveristementEnum
always
never
onlyIfOrdered
onlyIfInitiated
origin
destination
end
ServiceJourneyModel::JourneyEndpoint
Name :MultilingualString
«FK»
PointRef :ScheduledStopPointRef* [0..1]
applies to
GroupOfEntities
ServiceJourneyModel::GroupOfServices
DirectionType :DirectionTypeEnum
«PK»
id :GroupOfServicesIdType
«FK»
DayTypeRef :DayTypeRef*
«EV»
DirectionView :DirectionView [0..1]
«contained»
members :VehicleJourneyRef* [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
made up of 0..1
DerivedView
«TM_VIEW»
LineAndRouteViews::DestinationDisplayView
/Name :MultilingualString
/ShortName :MultilingualString [0..1]
/PublicCode :normalizedString [0..1]
/SideText :MultilingualString [0..1]
/FrontText :MultilingualString [0..1]
/PublicCode :normalizedString [0..1]
/ShortCode :normalizedString [0..1]
«PK»
DestinationDisplayRer :DestinationDisplayIdType
summary 0..1
view of 1
DataManagedObject
LineModel::DestinationDisplay
ServiceJourneyModel::ServiceJourney
ServiceAlteration :ServiceAlterationEnum [0..1]
DirectionType :DirectionTypeEnum
Print :boolean [0..1]
Dynamic :DynamicAdvertisementEnum [0..1]
«PK»
id :ServiceJourneyIdType
«FK»
VehicleTypeRef :VehicleTypeRef* [0..1]
«EV»
OperatorView :OperatorView [0..1]
LineView :LineView [0..1]
JourneyPatternVew :JourneyPatternView
«contained»
Origin :JourneyEndpoint [0..1]
Destination :JourneyEndpoint [0..1]
calls :Call [0..*]
faciliities :Facility [0..*]
groupsOfServices :GroupOfService [0..*]
checkConstraints :CheckConstraint [0..*]
FlexibleProperties :FlexibleServiceProperties
trainNumbers :TimeDemandTypeRef* [0..*]
made using *
proposed for 0..1
DataManagedObject
VehicleTypeModel::
VehicleType
proposed for g 0..1
*
affects
process
DataManagedObject
CheckConstraintModel::
CheckConstraint
DataManagedObject
InterchangeModel::Interchange
0..* interchange
in 1..*
described by *
Journey
ServiceJourneyModel::SpecialService
StartTime :time
EndTime :time [0..1]
Client :normalizedString
«PK»
id :SpecialServiceIdType
«FK»
JourneyPatternRef :JourneyPatternRef* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
classified as *
0..1
the classification for
TypeOfEntity
VehicleJourneyModel::
TypeOfService
«PK»
id :TypeOfServiceIdType
applies to 0..1
rule 0..*
InterchangeRuleModel::
InterchangeRule
start of 1
0..* /journey to
/starts
origin
end of 1
from
to
InterchangeModel::
ServiceJourneyInterchange
from * * to *
DerivedView
«TM_VIEW»
LineAndRouteViews::LineView
/PublicCode :normalizedString [0..1]
/Name :MultilingualString
/ShortName :MultilingualString [0..1]
/Description :MultilingualString [0..1]
/TransportMode :ModeEnum* [0..1]
/Submode :SubmodeEnum* [0..1]
«FK»
LineRef :LineIdType
summary 0..*
view of 1
DerivedView
OrganisationView::OrganisationView
/Name :normalizedString
/OrganisationType :TypeOfOrganisationEnum
/ShortName :MultilingualString [0..1]
/TradingName :MultilingualString [0..1]
/LegalName :MultilingualString [0..1]
«FK»
OrganisationRef :OrganisationIdType
summary 0..1
view of 1
Organisation
TransportOrganisationModel::
Operator
additional 0..1
primary 0..1
run by 0..*
run by 0..*
DataManagedObject
LineModel::Line
stop 1
DerivedView
«TM_VIEW»
ServicePatternViews::ScheduledStopPointView
/Name :MultilingualString [0..1]
/TypeOfPointRef :TypeOfPointIdType
/ShortName :MultilingualString [0..1]
/Description :MultilingualString [0..1]
/Label :MultilingualString [0..1]
/StopType :StopTypeEnum [0..1]
/Url :normalizedString
«PK»
ScheduledStopPointRef :ScheduledStopPointIdType
«AK»
/PrivateCode :PrivateCodeType [0..1]
/PublicCode :normalizedString [0..1]
summary 0..*
view of 1
/destinat/origin
0..1 0..1
at 0..1
start of 1
end of 1
end of 1
start of 1
TimingPoint
ServicePatternModel::ScheduledStopPoint
start of 1
end of 1
ServicePatternModel::ServiceJourneyPattern
«PK»
id :ServiceJourneyPatternIdType
«FK»
TypeOfServicePattern :TypeOfServicePatternEnum*
start of 1
from *
end of 1
to *
InterchangeModel::ServiceJourneyPatternInterchange
to *
* from
changing 0..*
interchange 0..*
using 0..1
at 0..1
Transfer
ServicePatternModel::Connection
at 0..1
from 0..*
to 0..*
Name: XSD NeTEx TP Service Journey Model
Author: NeTEx
Version: 1.0
Created: 25/11/2009 17:29:44
Updated: 02/08/2012 14:46:59

**Figure 24 — Service Journey – Physical Model (UML)**

#### 7.2.2.4 Service Journey – Attributes and XSD

##### 7.2.2.4.1 GroupOfServices – Model Element

A group of SERVICEs, often known to its users by a name or a number.

Table 12 – *GroupOfServices* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *GroupOfEntities* | *::>* | GROUP OF SERVICES inherits from GROUP OF ENTITIES. |
| «PK» | ***id*** | *GroupOfServicesIdType* | 1:1 | Identifier of GROUP OF SERVICEs. |
| «FK» | ***dayTypes*** | *DayTypeRef* | 0:* | Reference DAY TYPEs applying to the GROUP OF SERVICEs. |
|  | ***DirectionType*** | *DirectionTypeEnum* | 1:1 | Type of DIRECTION. See Part 1. |
| «EV» | ***DirectionView*** | *DirectionView* | 0:1 | Reference to common DIRECTION of SERVICE JOURNEYs in the GROUP OF SERVICE, with derived values. |
| «cntd» | ***destination-Displays*** | *DestinationDisplay* | 0:* | DESTINATION DISPLAYs associated with the GROUP OF SERVICEs. |
| «cntd» | ***members*** | *VehicleJourneyRef* | 0:* | Members of a GROUP OF SERVICEs. |
| «cntd» | ***notice-Assignments*** | *NoticeAssignmentView* | 0:* | NOTICE ASSIGNMENTs that apply to a GROUP OF SERVICE. |

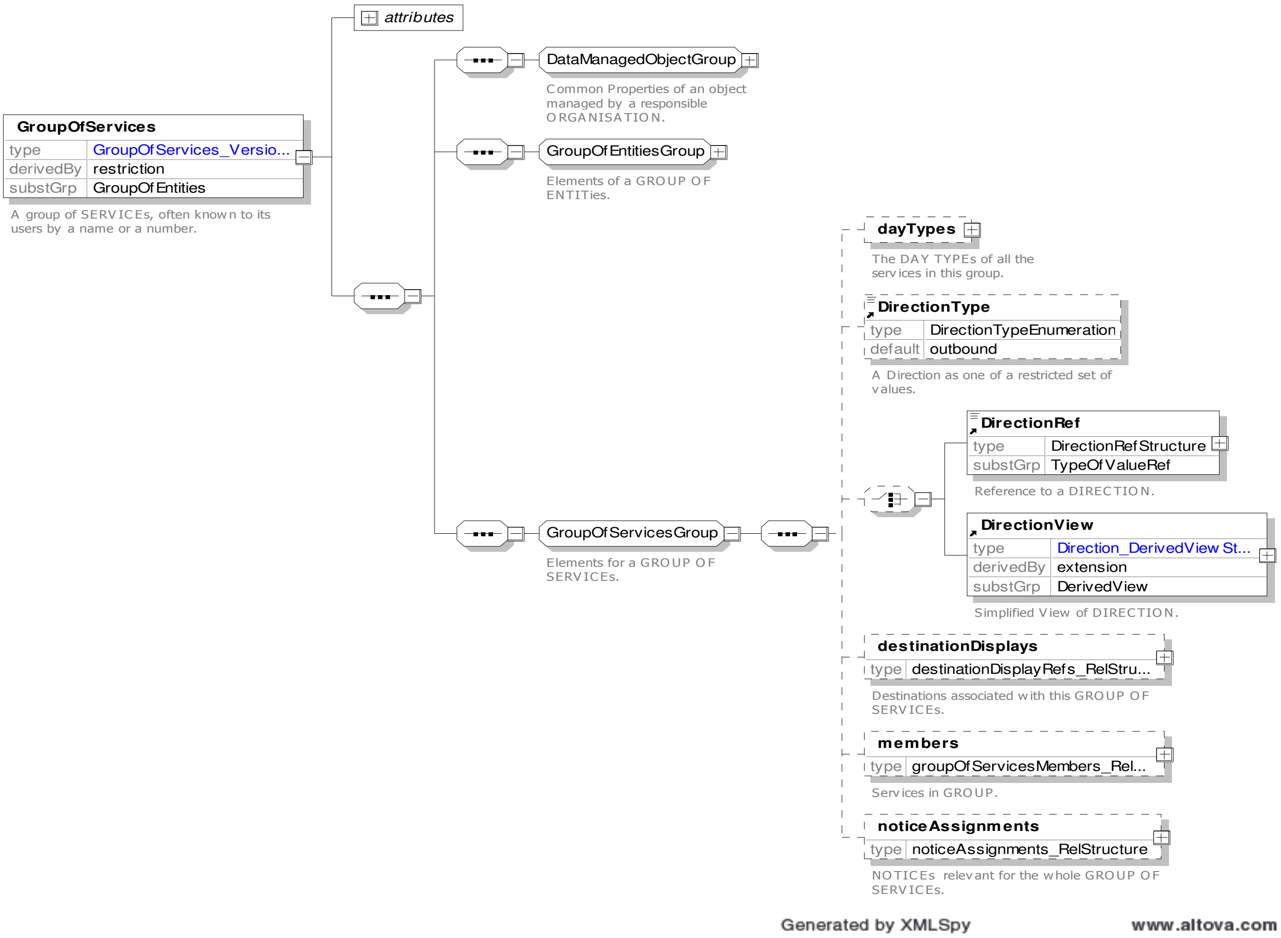

attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
GroupOfServices
type GroupOfServices_Versio...
derivedBy restriction
substGrp GroupOfEntities
A group of SERVICEs, often known to its users by a name or a number.
GroupOfEntitiesGroup
Elements of a GROUP OF ENTITIes.
dayTypes
The DAY TYPEs of all the services in this group.
DirectionType
type DirectionTypeEnumeration
default outbound
A Direction as one of a restricted set of values.
DirectionRef
type DirectionRefStructure
substGrp TypeOfValueRef
Reference to a DIRECTION.
GroupOfServicesGroup
Elements for a GROUP OF SERVICEs.
DirectionView
type Direction_DerivedView St...
derivedBy extension
substGrp DerivedView
Simplified View of DIRECTION.
destinationDisplays
type destinationDisplayRefs_RelStru...
Destinations associated with this GROUP OF SERVICEs.
members
type groupOfServicesMembers_Rel...
Services in GROUP.
noticeAssignments
type noticeAssignments_RelStructure
NOTICEs relevant for the whole GROUP OF SERVICEs.
Generated by XMLSpy
www.altova.com

Figure 25 — *GroupOfServices* — XSD

### 7.2.2.4.2 ServiceJourney – Model Element

A passenger carrying VEHICLE JOURNEY for one specified DAY TYPE. The pattern of working is in principle defined by a SERVICE JOURNEY PATTERN.

Table 13 – ServiceJourney – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *Journey* | *::>* | SERVICE JOURNEY inherits from JOURNEY. It includes elements from VEHICLE JOURNEY |
| «PK» | ***id*** | *ServiceJourneyIdType* | 1:1 | Identifier of SERVICE JOURNEY. |
|  | ***ServiceAlteration*** | *ServiceAlterationEnum* | 0:1 | Status to journey - planned, cancelled, etc. |
| «FK» | ***VehicleTypeRef*** | *VehicleTypeRef* | 0:1 | Reference to TYPE OF VEHICLE to use on of SERVICE JOURNEY. |
| «EV» | ***OperatorView*** | *OrganisationView* | 0:1 | Reference to OPERATOR of SERVICE JOURNEY. May include OPERATOR's derived values. |
| «EV» | ***LineView*** | *LineView* | 0:1 | Reference to LINE of SERVICE JOURNEY. May include derived values of LINE. |

| | | | | |
|---|---|---|---|---|
| | ***DirectionType*** | *DirectionTypeEnum* | 1:1 | Type of DIRECTION. |
| | ***Print*** | *xsd:boolean* | 0:1 | Whether this journey should be visible to public in print channels. |
| | ***Dynamic*** | *Dynamic-AdvertisementEnum* | 0:1 | When this journey should be visible to public in dynamic channels. |
| «EV» | ***JourneyPattern-View*** | *JourneyPatternView* | 1:1 | Reference to JOURNEY PATTERN of SERVICE JOURNEY and derived values of PATTERN. |
| «cntd» | ***Origin*** | *JourneyEndpoint* | 0:1 | Origin of SERVICE JOURNEY. |
| «cntd» | ***Destination*** | *JourneyEndpoint* | 0:1 | Destination of SERVICE JOURNEY. |
| «cntd» | ***calls*** | *Call* | 0:* | CALLs made by SERVICE JOURNEY. |
| «cntd» | ***facilities*** | *Facility* | 0:* | Facilities on SERVICE JOURNEY. |
| «cntd» | ***groupsOfServices*** | *GroupOfServices* | 0:* | GROUP OF SERVICEs to which SERVICE JOURNEY belongs. |
| «cntd» | ***checkConstraints*** | *CheckConstraint* | 0:* | CHECK CONSTRAINTs which apply to SERVICE JOURNEY, check in time, security time. These are advisory and not for journey planning. |
| «cntd» | ***FlexibleService-Properties*** | *FlexibleServiceProperties* | 1:1 | Flexible properties of SERVICE JOURNEY. |

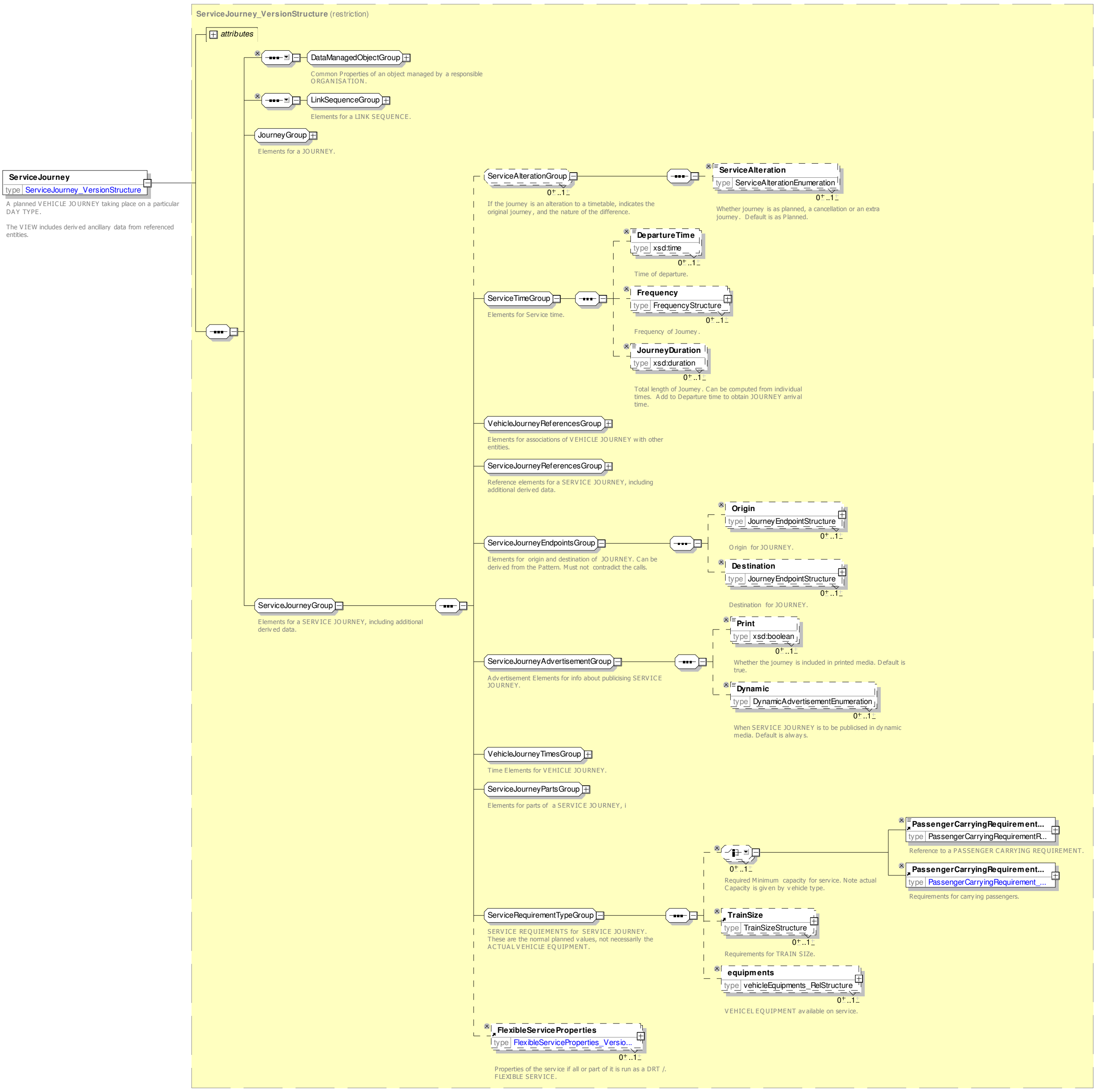

ServiceJourney_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
JourneyGroup
Elements for a JOURNEY.
ServiceJourney
type ServiceJourney_VersionStructure
A planned VEHICLE JOURNEY taking place on a particular DAY TYPE.
The VIEW includes derived ancillary data from referenced entities.
ServiceAlterationGroup
0..1
If the journey is an alteration to a timetable, indicates the original journey, and the nature of the difference.
ServiceAlteration
type ServiceAlterationEnumeration
0..1
Whether journey is as planned, a cancellation or an extra journey. Default is as Planned.
DepartureTime
type xsd:time
0..1
Time of departure.
ServiceTimeGroup
Elements for Service time.
Frequency
type FrequencyStructure
0..1
Frequency of Journey.
JourneyDuration
type xsd:duration
0..1
Total length of Journey. Can be computed from individual times. Add to Departure time to obtain JOURNEY arrival time.
VehicleJourneyReferencesGroup
Elements for associations of VEHICLE JOURNEY with other entities.
ServiceJourneyReferencesGroup
Reference elements for a SERVICE JOURNEY, including additional derived data.
Origin
type JourneyEndpointStructure
0..1
Origin for JOURNEY.
ServiceJourneyEndpointsGroup
Elements for origin and destination of JOURNEY. Can be derived from the Pattern. Must not contradict the calls.
Destination
type JourneyEndpointStructure
0..1
Destination for JOURNEY.
ServiceJourneyGroup
Elements for a SERVICE JOURNEY, including additional derived data.
Print
type xsd:boolean
0..1
Whether the journey is included in printed media. Default is true.
ServiceJourneyAdvertisementGroup
Advertisement Elements for info about publicising SERVICE JOURNEY.
Dynamic
type DynamicAdvertisementEnumeration
0..1
When SERVICE JOURNEY is to be publicised in dynamic media. Default is always.
VehicleJourneyTimesGroup
Time Elements for VEHICLE JOURNEY.
ServiceJourneyPartsGroup
Elements for parts of a SERVICE JOURNEY, i
PassengerCarryingRequirement...
type PassengerCarryingRequirementR...
Reference to a PASSENGER CARRYING REQUIREMENT.
0..1
Required Minimum capacity for service. Note actual Capacity is given by vehicle type.
PassengerCarryingRequirement...
type PassengerCarryingRequirement_...
Requirements for carrying passengers.
TrainSize
type TrainSizeStructure
0..1
Requirements for TRAIN SIZe.
ServiceRequirementTypeGroup
SERVICE REQUIREMENTS for SERVICE JOURNEY. These are the normal planned values, not necessarily the ACTUAL VEHICLE EQUIPMENT.
equipments
type vehicleEquipments_RelStructure
0..1
VEHICEL EQUIPMENT available on service.
FlexibleServiceProperties
type FlexibleServiceProperties_Versio...
0..1
Properties of the service if all or part of it is run as a DRT /. FLEXIBLE SERVICE.

Figure 26 — *ServiceJourney* — XSD

### 7.2.2.4.2.1 ServiceAlteration – Allowed values

The following figure shows the allowed values for ServiceAlteration *(ServiceAlterationEnum).*

Table 14 – ServiceAlteration – Allowed values

| Value | Description |
|---|---|
| *extraJourney* | Journey is a short term addition to the original planned schedule. |
| *cancellation* | Journey is a cancellation of a journey in the planned schedule. |
| *planned* | Journey is as planned. |

### 7.2.2.4.2.2 VehicleJourneyReferencesGroup – Model Element

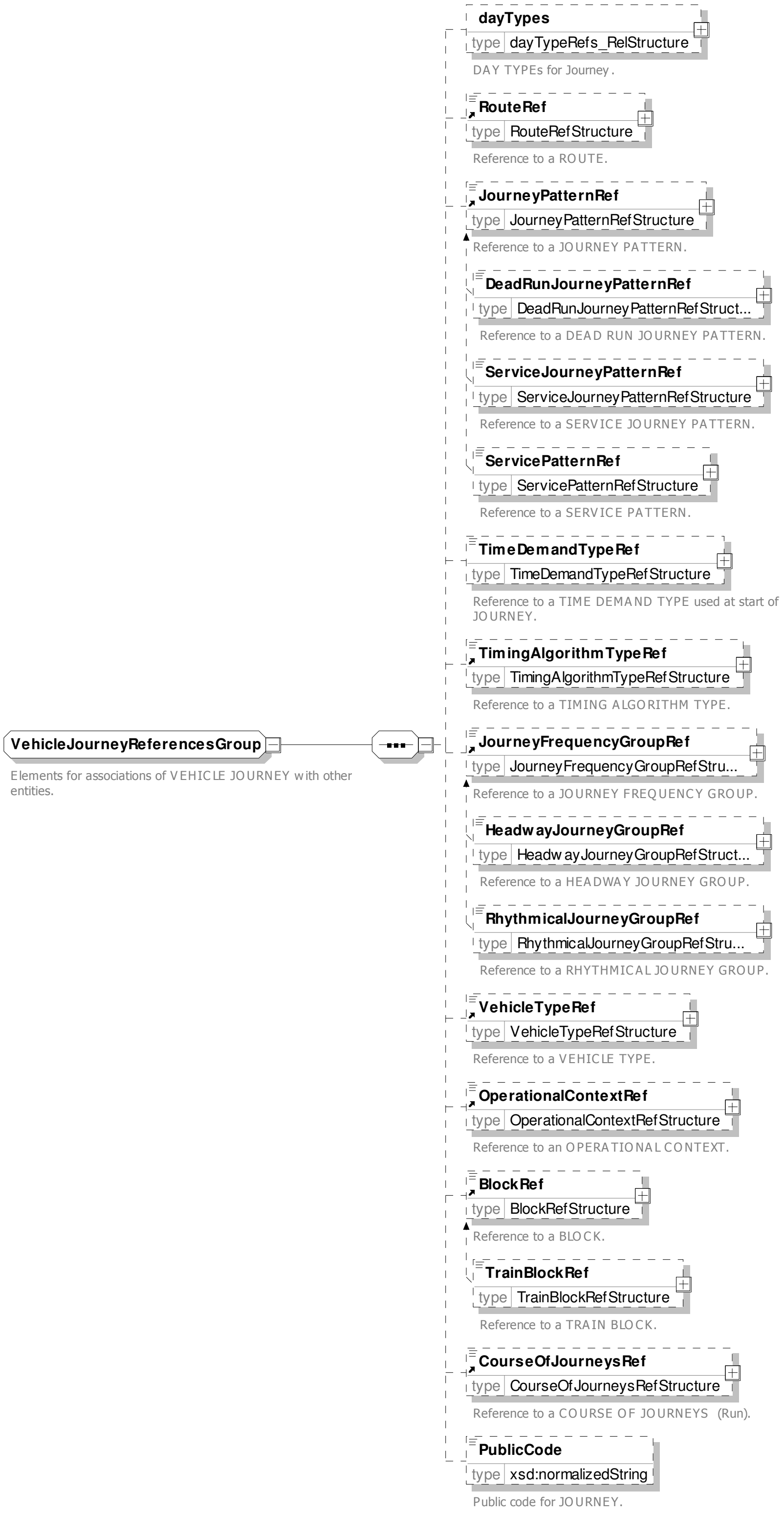

dayTypes
type dayTypeRefs_RelStructure
DAY TYPEs for Journey.
RouteRef
type RouteRefStructure
Reference to a ROUTE.
JourneyPatternRef
type JourneyPatternRefStructure
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.
ServiceJourneyPatternRef
type ServiceJourneyPatternRefStructure
Reference to a SERVICE JOURNEY PATTERN.
ServicePatternRef
type ServicePatternRefStructure
Reference to a SERVICE PATTERN.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE used at start of JOURNEY.
TimingAlgorithmTypeRef
type TimingAlgorithmTypeRefStructure
Reference to a TIMING ALGORITHM TYPE.
VehicleJourneyReferencesGroup
Elements for associations of VEHICLE JOURNEY with other entities.
JourneyFrequencyGroupRef
type JourneyFrequencyGroupRefStru...
Reference to a JOURNEY FREQUENCY GROUP.
HeadwayJourneyGroupRef
type HeadwayJourneyGroupRefStruct...
Reference to a HEADWAY JOURNEY GROUP.
RhythmicalJourneyGroupRef
type RhythmicalJourneyGroupRefStru...
Reference to a RHYTHMICAL JOURNEY GROUP.
VehicleTypeRef
type VehicleTypeRefStructure
Reference to a VEHICLE TYPE.
OperationalContextRef
type OperationalContextRefStructure
Reference to an OPERATIONAL CONTEXT.
BlockRef
type BlockRefStructure
Reference to a BLOCK.
TrainBlockRef
type TrainBlockRefStructure
Reference to a TRAIN BLOCK.
CourseOfJourneysRef
type CourseOfJourneysRefStructure
Reference to a COURSE OF JOURNEYS (Run).
PublicCode
type xsd:normalizedString
Public code for JOURNEY.

Figure 27 — *VehicleJourneyReferencesGroup* — XSD

### 7.2.2.4.2.3 ServiceJourneyReferencesGroup – Model Element

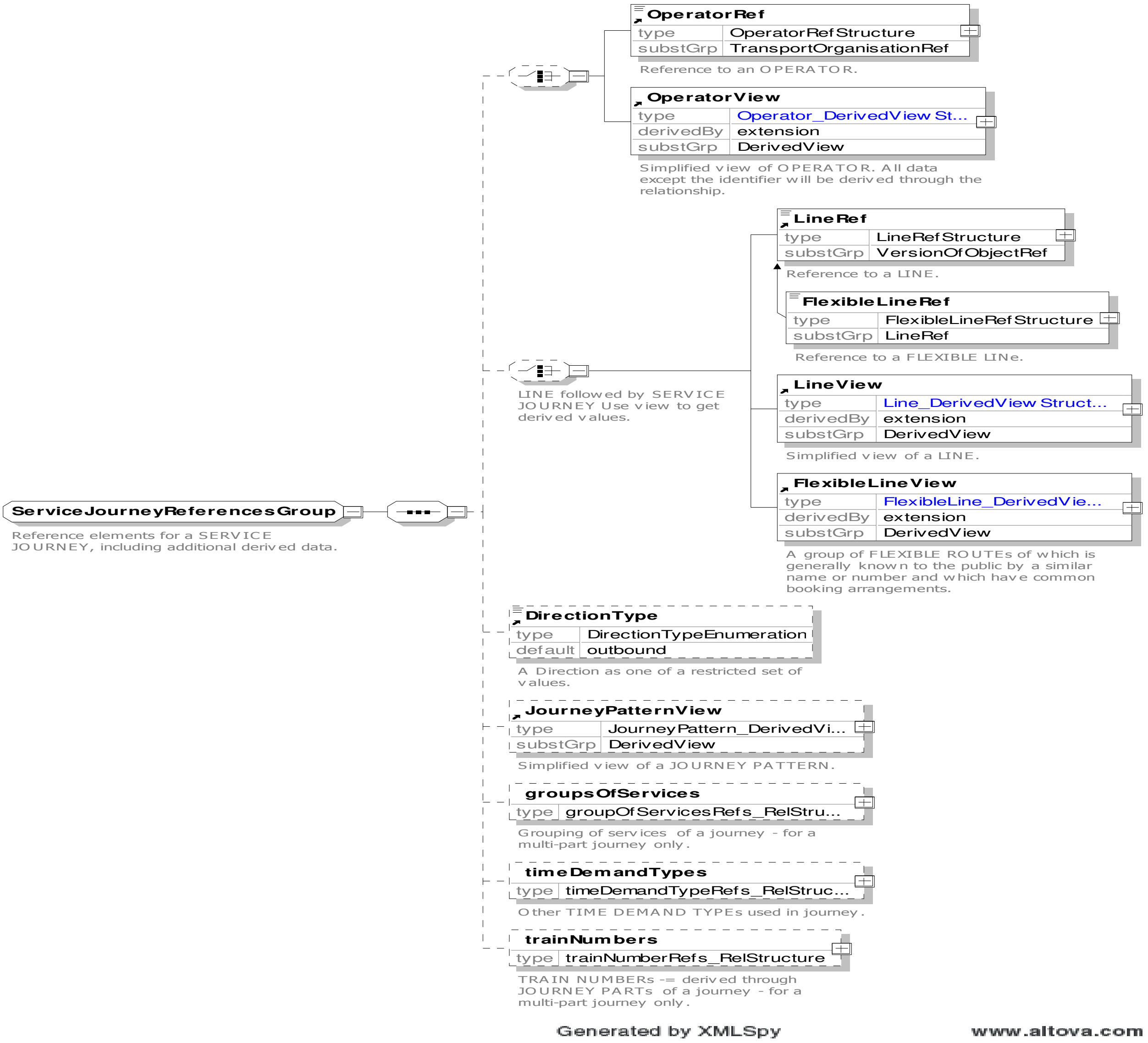

OperatorRef
type OperatorRefStructure
substGrp TransportOrganisationRef
Reference to an OPERATOR.
OperatorView
type Operator_DerivedViewSt...
derivedBy extension
substGrp DerivedView
Simplified view of OPERATOR. All data except the identifier will be derived through the relationship.
LineRef
type LineRefStructure
substGrp VersionOfObjectRef
Reference to a LINE.
FlexibleLineRef
type FlexibleLineRefStructure
substGrp LineRef
Reference to a FLEXIBLE LINe.
LINE followed by SERVICE JOURNEY Use view to get derived values.
LineView
type Line_DerivedViewStruct...
derivedBy extension
substGrp DerivedView
Simplified view of a LINE.
ServiceJourneyReferencesGroup
Reference elements for a SERVICE JOURNEY, including additional derived data.
FlexibleLineView
type FlexibleLine_DerivedVie...
derivedBy extension
substGrp DerivedView
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.
DirectionType
type DirectionTypeEnumeration
default outbound
A Direction as one of a restricted set of values.
JourneyPatternView
type JourneyPattern_DerivedVi...
substGrp DerivedView
Simplified view of a JOURNEY PATTERN.
groupsOfServices
type groupOfServicesRefs_RelStru...
Grouping of services of a journey - for a multi-part journey only.
timeDemandTypes
type timeDemandTypeRefs_RelStruc...
Other TIME DEMAND TYPEs used in journey.
trainNumbers
type trainNumberRefs_RelStructure
TRAIN NUMBERs -= derived through JOURNEY PARTs of a journey - for a multi-part journey only.
Generated by XMLSpy
www.altova.com

**Figure 28 — *ServiceJourneyReferencesGroup* — XSD**

### 7.2.2.4.2.4 DynamicAdvertisement – Allowed values

The following table shows the allowed values for ***DynamicAdvertisement*** *(DynamicAdvertisementEnum).*

**Table 15 – *DynamicAdvertisement* – Allowed values**

| Value | Description |
|---|---|
| *onlyIfSignedOn* | Only show journey in public displays if driver has signed in |
| *OnlyIfOrdered* | Only show journey in public displays if journey has been ordered |
| *always* | Always show journey in public displays |
| *never* | Never show journey in public displays |

### 7.2.2.4.2.5 VehicleJourneyTimesGroup – Model Element

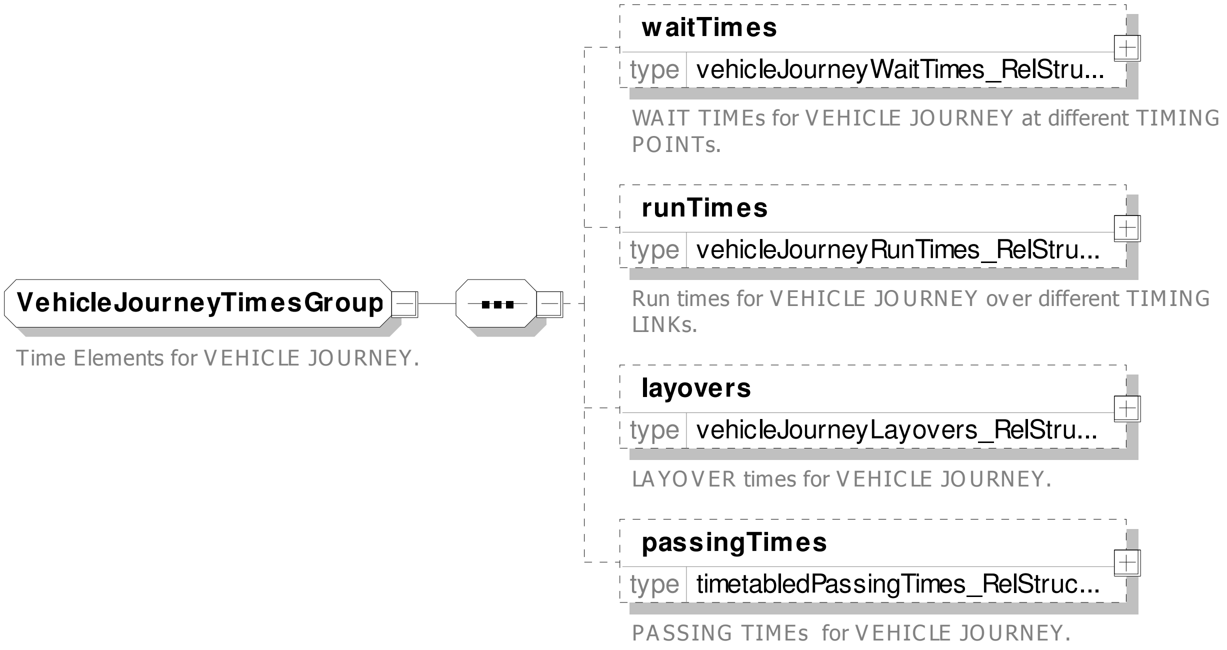

VehicleJourneyTimesGroup
Time Elements for VEHICLE JOURNEY.
waitTimes
type vehicleJourneyWaitTimes_RelStru...
WAIT TIMEs for VEHICLE JOURNEY at different TIMING POINTs.
runTimes
type vehicleJourneyRunTimes_RelStru...
Run times for VEHICLE JOURNEY over different TIMING LINKs.
layovers
type vehicleJourneyLayovers_RelStru...
LAYOVER times for VEHICLE JOURNEY.
passingTimes
type timetabledPassingTimes_RelStruc...
PASSING TIMEs for VEHICLE JOURNEY.

**Figure 29 — *VehicleJourneyTimesGroup* — XSD**

### 7.2.2.4.2.6 ServiceJourneyPartsGroup – Model Element

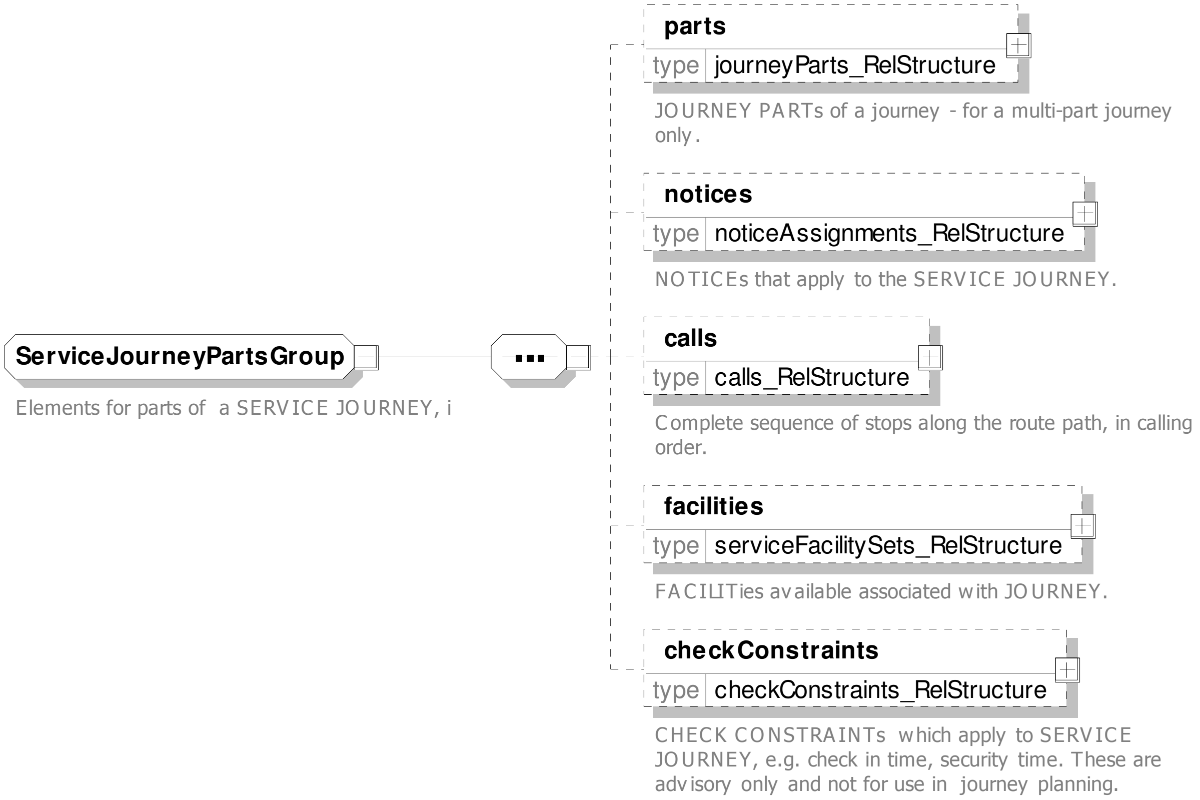

ServiceJourneyPartsGroup
Elements for parts of a SERVICE JOURNEY, i
parts
type journeyParts_RelStructure
JOURNEY PARTs of a journey - for a multi-part journey only.
notices
type noticeAssignments_RelStructure
NOTICEs that apply to the SERVICE JOURNEY.
calls
type calls_RelStructure
Complete sequence of stops along the route path, in calling order.
facilities
type serviceFacilitySets_RelStructure
FACILITies available associated with JOURNEY.
checkConstraints
type checkConstraints_RelStructure
CHECK CONSTRAINTs which apply to SERVICE JOURNEY, e.g. check in time, security time. These are advisory only and not for use in journey planning.

**Figure 30 — *ServiceJourneyPartsGroup* — XSD**

#### 7.2.2.4.2.7 ServiceRequirementTypeGroup – Model Element

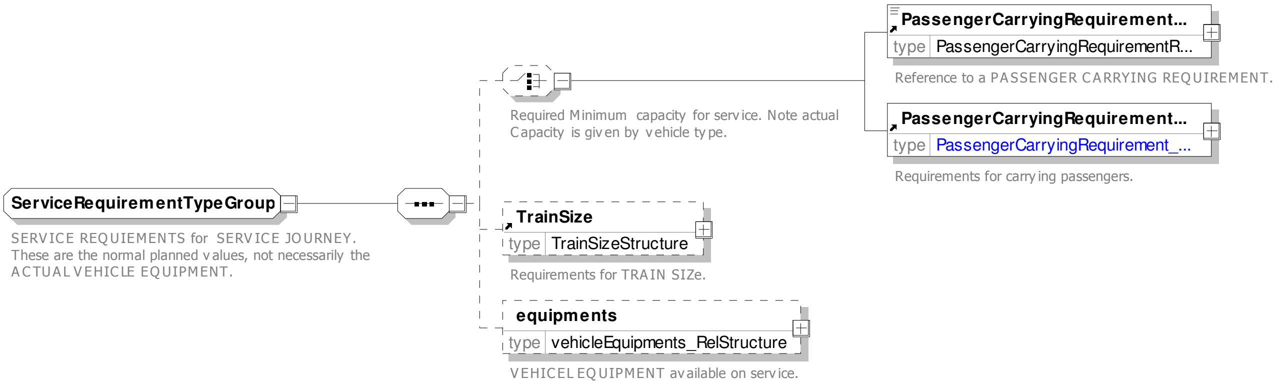

PassengerCarryingRequirement...
type PassengerCarryingRequirementR...
Reference to a PASSENGER CARRYING REQUIREMENT.
Required Minimum capacity for service. Note actual Capacity is given by vehicle type.
PassengerCarryingRequirement...
type PassengerCarryingRequirement_...
Requirements for carrying passengers.
ServiceRequirementTypeGroup
SERVICE REQUIEMENTS for SERVICE JOURNEY. These are the normal planned values, not necessarily the ACTUAL VEHICLE EQUIPMENT.
TrainSize
type TrainSizeStructure
Requirements for TRAIN SIZe.
equipments
type vehicleEquipments_RelStructure
VEHICEL EQUIPMENT available on service.

Figure 31 — *ServiceRequirementTypeGroup* — XSD

### 7.2.2.4.3 SpecialService – Model Element

A work of a vehicle that is not planned in a classical way, i.e. that is generally not based on VEHICLE JOURNEYs using JOURNEY PATTERNs. It involves specific characteristics (such as specific access rights) and/or may be operated under specific circumstances.

**Table 16 – SpecialService – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *Journey* | *::>* | SPECIAL SERVICE inherits from JOURNEY. |
| «PK» | ***id*** | *SpecialServiceIdType* | 1:1 | Identifier of SPECIAL SERVICE. |
| | ***StartTime*** | *xsd:time* | 1:1 | Start time for SPECIAL SERVICE. |
| | ***EndTime*** | *xsd:time* | 0:1 | End time for SPECIAL SERVICE. |
| | ***Client*** | *xsd:normalizedString* | 1:1 | Client for SPECIAL SERVICE. |
| | ***DayTypeRef*** | *DayTypeRef* | 0:1 | DAY TYPE for SPECIAL SERVICE. |
| «FK» | ***JourneyPattern-Ref*** | *JourneyPatternRef* | 0:1 | JOURNEY PATTERN for SPECIAL SERVICE. |
| «FK» | ***VehicleTypeRef*** | *VehicleTypeRef* | 0:1 | VEHICLE TYPE for SPECIAL SERVICE. |
| «cntd» | ***Origin*** | *JourneyEndpoint* | 0:1 | Origin of SERVICE JOURNEY. |
| «cntd» | ***Destination*** | *JourneyEndpoint* | 0:1 | Destination of SERVICE JOURNEY. |
| | ***Print*** | *xsd:boolean* | 0:1 | Whether this journey should be visible to public in print channels. |
| | ***Dynamic*** | *Dynamic-AdvertisementEnum* | 0:1 | When this journey should be visible to public in dynamic channels. |
| «FK» | ***VehicleTypeRef*** | *VehicleTypeRef* | 0:1 | VEHICLE TYPE for SPECIAL SERVICE. |
| :: | ***Booking-Arrangements-Group*** | BookingArrangements | 0:1 | BOOKING ARRANGEMENTs for SPECIAL SERVICE |

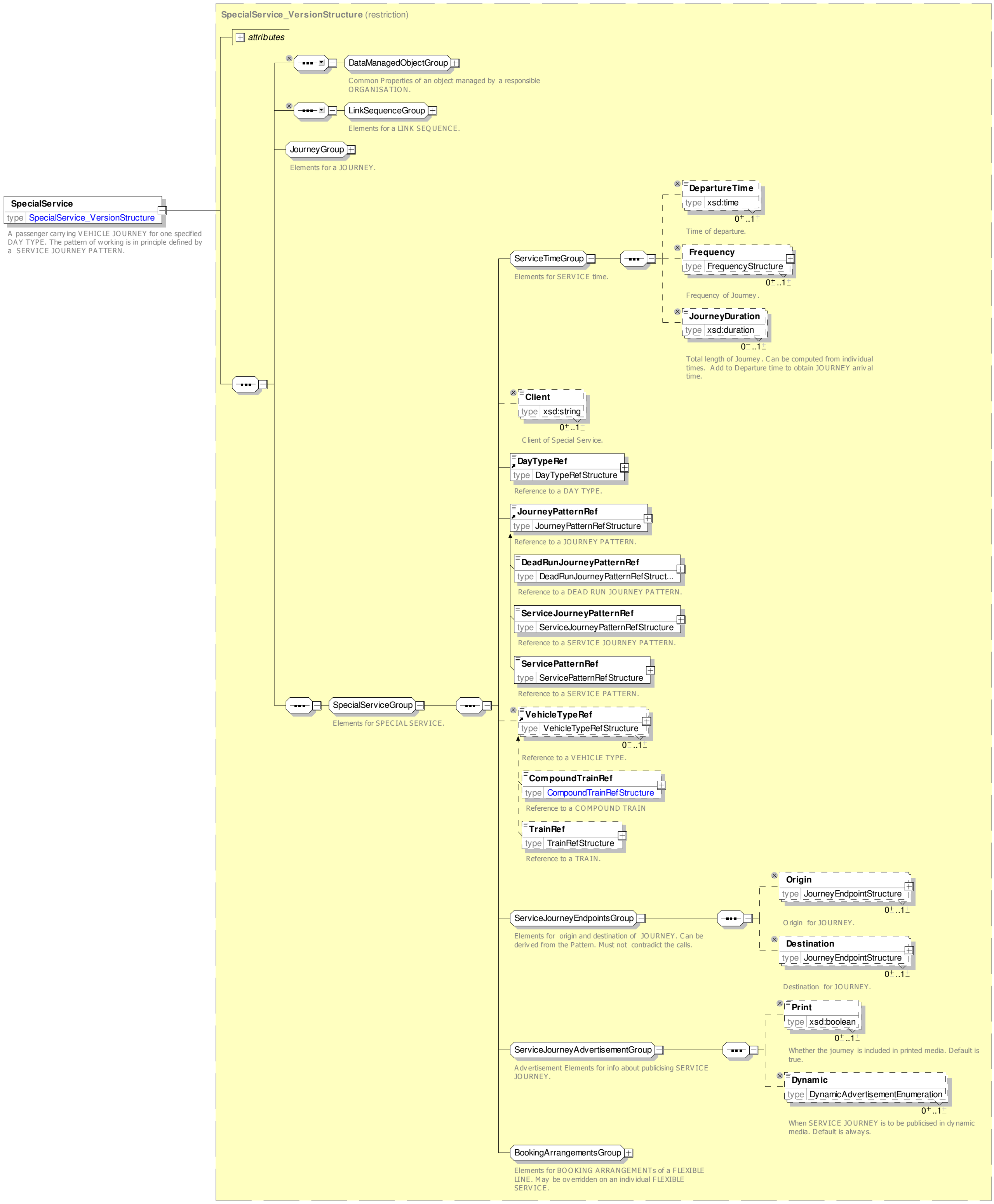

SpecialService_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
JourneyGroup
Elements for a JOURNEY.
SpecialService
type SpecialService_VersionStructure
A passenger carrying VEHICLE JOURNEY for one specified DAY TYPE. The pattern of working is in principle defined by a SERVICE JOURNEY PATTERN.
DepartureTime
type xsd:time
0..1
Time of departure.
ServiceTimeGroup
Elements for SERVICE time.
Frequency
type FrequencyStructure
0..1
Frequency of Journey.
JourneyDuration
type xsd:duration
0..1
Total length of Journey. Can be computed from individual times. Add to Departure time to obtain JOURNEY arrival time.
Client
type xsd:string
0..1
Client of Special Service.
DayTypeRef
type DayTypeRefStructure
Reference to a DAY TYPE.
JourneyPatternRef
type JourneyPatternRefStructure
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.
ServiceJourneyPatternRef
type ServiceJourneyPatternRefStructure
Reference to a SERVICE JOURNEY PATTERN.
ServicePatternRef
type ServicePatternRefStructure
Reference to a SERVICE PATTERN.
SpecialServiceGroup
Elements for SPECIAL SERVICE.
VehicleTypeRef
type VehicleTypeRefStructure
0..1
Reference to a VEHICLE TYPE.
CompoundTrainRef
type CompoundTrainRefStructure
Reference to a COMPOUND TRAIN
TrainRef
type TrainRefStructure
Reference to a TRAIN.
Origin
type JourneyEndpointStructure
0..1
Origin for JOURNEY.
ServiceJourneyEndpointsGroup
Elements for origin and destination of JOURNEY. Can be derived from the Pattern. Must not contradict the calls.
Destination
type JourneyEndpointStructure
0..1
Destination for JOURNEY.
Print
type xsd:boolean
0..1
Whether the journey is included in printed media. Default is true.
ServiceJourneyAdvertisementGroup
Advertisement Elements for info about publicising SERVICE JOURNEY.
Dynamic
type DynamicAdvertisementEnumeration
0..1
When SERVICE JOURNEY is to be publicised in dynamic media. Default is always.
BookingArrangementsGroup
Elements for BOOKING ARRANGEMENTs of a FLEXIBLE LINE. May be overridden on an individual FLEXIBLE SERVICE.

Figure 32 — *SpecialService* — XSD

### 7.2.2.4.3.1 BookingArrangements – *Group*

Information about booking a flexible journey.

**Table 17 – *BookingArrangements* Group– Group**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| | ***BookingMethods*** | *BookingMethodEnum* | 0:* | Booking method for FLEXIBLE LINE. |
| | ***BookingAccess*** | *BookingAccessEnum* | 0:1 | Who can make a booking. |
| | ***LatestBooking-Time*** | *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. |

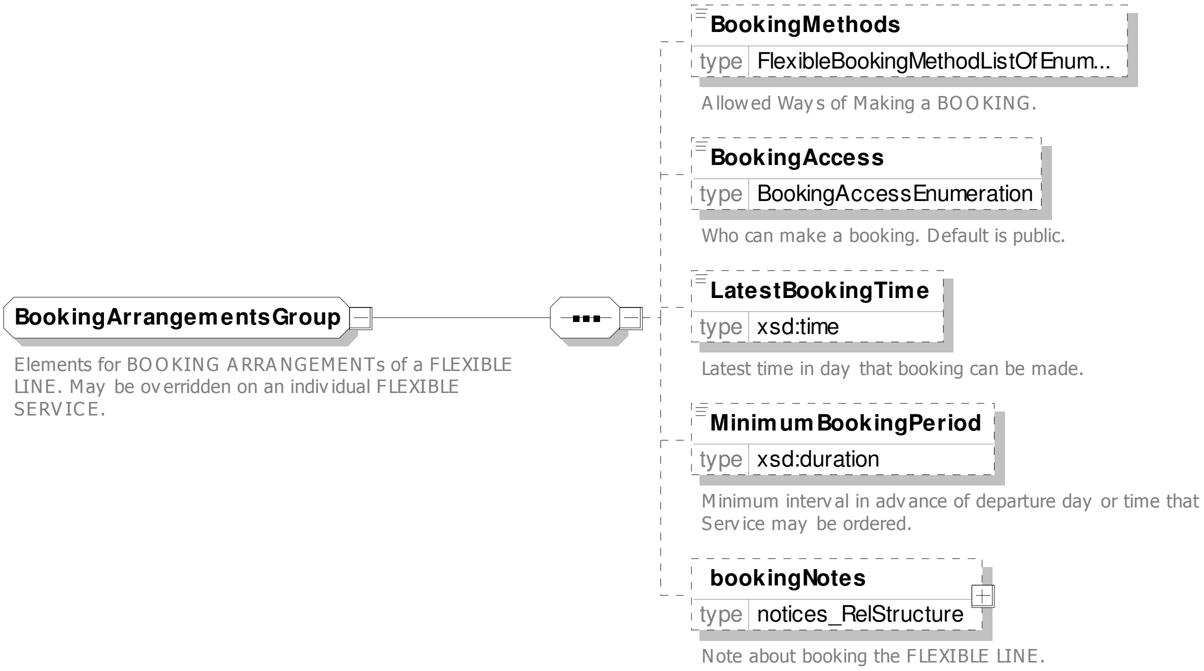

BookingArrangementsGroup
Elements for BOOKING ARRANGEMENTs of a FLEXIBLE LINE. May be overridden on an individual FLEXIBLE SERVICE.
BookingMethods
type FlexibleBookingMethodListOfEnum...
Allowed Ways of Making a BOOKING.
BookingAccess
type BookingAccessEnumeration
Who can make a booking. Default is public.
LatestBookingTime
type xsd:time
Latest time in day that booking can be made.
MinimumBookingPeriod
type xsd:duration
Minimum interval in advance of departure day or time that Service may be ordered.
bookingNotes
type notices_RelStructure
Note about booking the FLEXIBLE LINE.

**Figure 33 — *BookingArrangementsGroup* — XSD**

### 7.2.2.4.4 TemplateServiceJourney – Model Element

A repeating SERVICE JOURNEY for which a frequency has been specified, either as a HEADWAY JOURNEY GROUP (e.g. every 20 minutes) or a RYTHMICAL JOURNEY GROUP (e.g. at 15, 27 and 40 minutes past the hour). It may thus represent multiple journeys or it could be used simply as a template for adding single extra DATED VEHICLE JOURNEYs after the planning phase.

The frequency of the journey could be defined by a *FREQUENCY GROUP. See later*

The TEMPLATE SERVICE JOURNEY of a RYTHMICAL JOURNEY GROUP needs passing times, where only the minute part is meaningful (passing times are repeated every hour with the same minutes), therefore the hour part of passing times will be ignored (a "00" value is recommended).

**Table 18 – *TemplateServiceJourney* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *TemplateVehicleJourney* | ::> | TEMPLATE SERVICE JOURNEY inherits from |

| | | | | |
|---|---|---|---|---|
| | | | | TEMPLATE VEHICLE JOURNEY. |
| «PK» | ***id*** | *TemplateService-JourneyIdType* | 0:1 | Identifier of TEMPLATE SERVICE JOURNEY. |

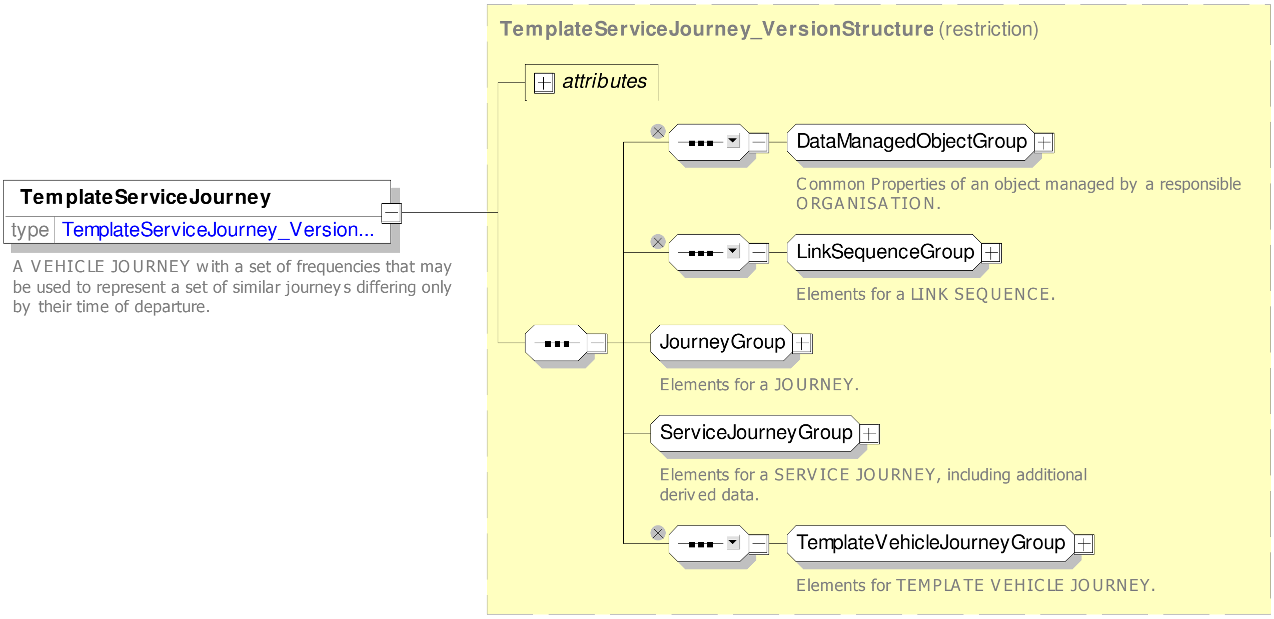

TemplateServiceJourney_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
JourneyGroup
Elements for a JOURNEY.
ServiceJourneyGroup
Elements for a SERVICE JOURNEY, including additional derived data.
TemplateVehicleJourneyGroup
Elements for TEMPLATE VEHICLE JOURNEY.
TemplateServiceJourney
type TemplateServiceJourney_Version...
A VEHICLE JOURNEY with a set of frequencies that may be used to represent a set of similar journeys differing only by their time of departure.

**Figure 34 — TemplateServiceJourney — XSD**

### 7.2.2.5 XML Examples of Service Journey

#### 7.2.2.5.1 Service Journey XML Example with passing times and calls

The following code fragment shows a SERVICE JOURNEY with just three stops – but five timing points.

For EXAMPLE:

```
<ServiceJourney version="any" id="hde:sj_24o_02_AxxD">
    <DepartureTime>14:20:00.0Z</DepartureTime>
    <dayTypes>
        <DayTypeRef version="any" ref="hde:Daype:DT_01-MF-NH"/>
    </dayTypes>
    <ServicePatternRef version="any" ref="hde:svp_24o_AxxD"/>
    <TimeDemandTypeRef version="any" ref="mybus:td_01"/>
    <LineRef version="any" ref="mybus:LN_24"/>
    <JourneyPatternView>
        <RouteRef version="any" ref="mybus:RT_24o"/>
        <DirectionType>outbound</DirectionType>
        <DestinationDisplayRef version="any" ref="mybus:DST_Delta"/>
    </JourneyPatternView>
    <passingTimes>
        <TimetabledPassingTime version="any" id="mybus:tpt_24o_02_AxxD_SSP_001">
            <StopPointInJourneyPatternRef version="any" ref="hde:spijp_24o_AxxD_01"/>
            <DepartureTime>14:20:00.0Z</DepartureTime>
        </TimetabledPassingTime>
        <TimetabledPassingTime version="any" id="hde:tpt_24o_02_AxxD_SSP_001_t1">
            <TimingPointInJourneyPatternRef version="any" ref="hde:tpijp_24o_AxxD_02"/>
            <DepartureTime>14:30:00.0Z</DepartureTime>
        </TimetabledPassingTime>
        <TimetabledPassingTime version="any" id="hde:tpt_24o_02_AxxD_SSP_001_t2">
            <TimingPointInJourneyPatternRef version="any" ref="hde:tpijp_24o_AxxD_03"/>
            <DepartureTime>14:40:00.0Z</DepartureTime>
        </TimetabledPassingTime>
        <TimetabledPassingTime version="any" id="mybus:tpt_24o_02_AxxD_SSP_002">
            <TimingPointInJourneyPatternRef version="any" ref="hde:tpijp_24o_AxxD_04"/>
            <DepartureTime>14:50:00.0Z</DepartureTime>
        </TimetabledPassingTime>
        <TimetabledPassingTime version="any" id="mybus:tpt_24o_02_AxxD_SSP_078">
            <StopPointInJourneyPatternRef version="any" ref="hde:spijp_24o_AxxD_04"/>
            <ArrivalTime>15:40:00.0Z</ArrivalTime>
        </TimetabledPassingTime>
    </passingTimes>
    <calls>
        <Call id="hde:Call:sj_24o_02_001" order="1">
            <ScheduledStopPointRef version="any" ref="mybus:SSP_001"/>
            <Arrival>
                <ForAlighting>false</ForAlighting>
            </Arrival>
            <Departure>
                <Time>14:20:00.0Z</Time>
            </Departure>
        </Call>
        <Call id="hde:Call:sj_24o_02_002" order="2">
            <ScheduledStopPointRef version="any" ref="mybus:SSP_002"/>
                <ForAlighting>false</ForAlighting>
            </Arrival>
            <Departure>
                <Time>14:50:00.0Z</Time>
            </Departure>
        </Call>

        <Call id="hde:Call:sj_24o_02_003" order="3">
            <ScheduledStopPointRef version="any" ref="mybus:SSP_078"/>
            <Arrival>
                <Time>15:40:00.0Z</Time>
            </Arrival>
            <Departure>
                <ForBoarding>false</ForBoarding>
            </Departure>
        </Call>
    </calls>
</ServiceJourney>
```

#### 7.2.2.5.2 Service Journey XML Example with passing times and calls

The following code fragment shows a SERVICE JOURNEY.

For EXAMPLE

```
<ServiceJourney version="any" id="hde:sj_24o_01">
    <DepartureTime>14:20:00.0Z</DepartureTime>
    <dayTypes>
        <DayTypeRef version="any" ref="hde:DT_02-Everyday-NotHoliday"/>
        <DayTypeRef version="any" ref="hde:DT_ChristmasEve"/>
        <DayTypeRef version="any" ref="hde:DT_ChristmasDay"/>
        <DayTypeRef version="any" ref="hde:DT_ChristmasDayDisplacement"/>
        <DayTypeRef version="any" ref="hde:DT_NewYearsEve"/>
        <DayTypeRef version="any" ref="hde:DT_NewYearsDay"/>
        <DayTypeRef version="any" ref="hde:DT_2ndJanuary"/>
        <DayTypeRef version="any" ref="hde:DT_GoodFriday"/>
        <DayTypeRef version="any" ref="hde:DT_EasterSunday"/>
        <DayTypeRef version="any" ref="hde:DT_EasterMonday"/>
    </dayTypes>
    <ServicePatternRef version="any" ref="hde:svp_24o"/>
    <TimeDemandTypeRef version="any" ref="mybus:tdt_01"/>
    <BlockRef ref="mybus:Block:BLK_24o5">EXTERNAL</BlockRef>
    <LineRef version="any" ref="mybus:LN_24"/>
    <JourneyPatternView>
        <RouteRef version="any" ref="mybus:RT_24o"/>
        <DirectionType>outbound</DirectionType>
        <DestinationDisplayRef version="any" ref="mybus:DST_Charley"/>
    </JourneyPatternView>
    <runTimes>
        <VehicleJourneyRunTime id="hde:vjrt_sj_24o_01">
            <Name>Overall run time</Name>
            <TimeDemandTypeRef version="any" ref="mybus:tdt_01"/>
            <TimingLinkRef version="any" ref="mybus:SSP_001_to_SSP_077"/>
            <RunTime>PT70M</RunTime>
        </VehicleJourneyRunTime>
        </runTimes>
```

### 7.2.3 Time Demand Times

#### 7.2.3.1 TIME DEMAND Times – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

Run times and wait times vary during the day, depending in particular on traffic conditions and on the number of passengers boarding or alighting from vehicles at stops. A classification of these conditions into arbitrary levels of demand is defined by the TIME DEMAND TYPE entity.

The TIME DEMAND TYPEs mainly indicate situations like “peak hour traffic conditions”, “off-peak traffic”, “night-owl traffic” etc. In bus operation for instance, the duration of run times usually differs between these situations. Wait times at stops rather depend on the passenger demand, which also varies with the time of day, but in a very similar pattern to the traffic conditions on the roads. Therefore, the TIME DEMAND TYPE serves as an indicator to classify standard run times as well as wait times, depending on specific conditions.

Each VEHICLE JOURNEY takes place at a defined time which can be characterised by specific traffic conditions and a certain passenger demand level. To express these characteristics, a TIME DEMAND TYPE may be assigned to a VEHICLE JOURNEY, in order to choose easily the appropriate run and wait times.

TIME DEMAND TYPEs, built from a CALENDAR DAY and a TIME BAND through an ASSIGNMENT mechanism, have already been introduced in NeTEx Part 1, as part of the tactical planning component: the following figure represents the associated timing information, which are RUN TIMEs, WAIT TIMEs, and a few other timing information like HEADWAY frequency, TURNAROUND TIME LIMIT and JOURNEY PATTERN LAYOVER.

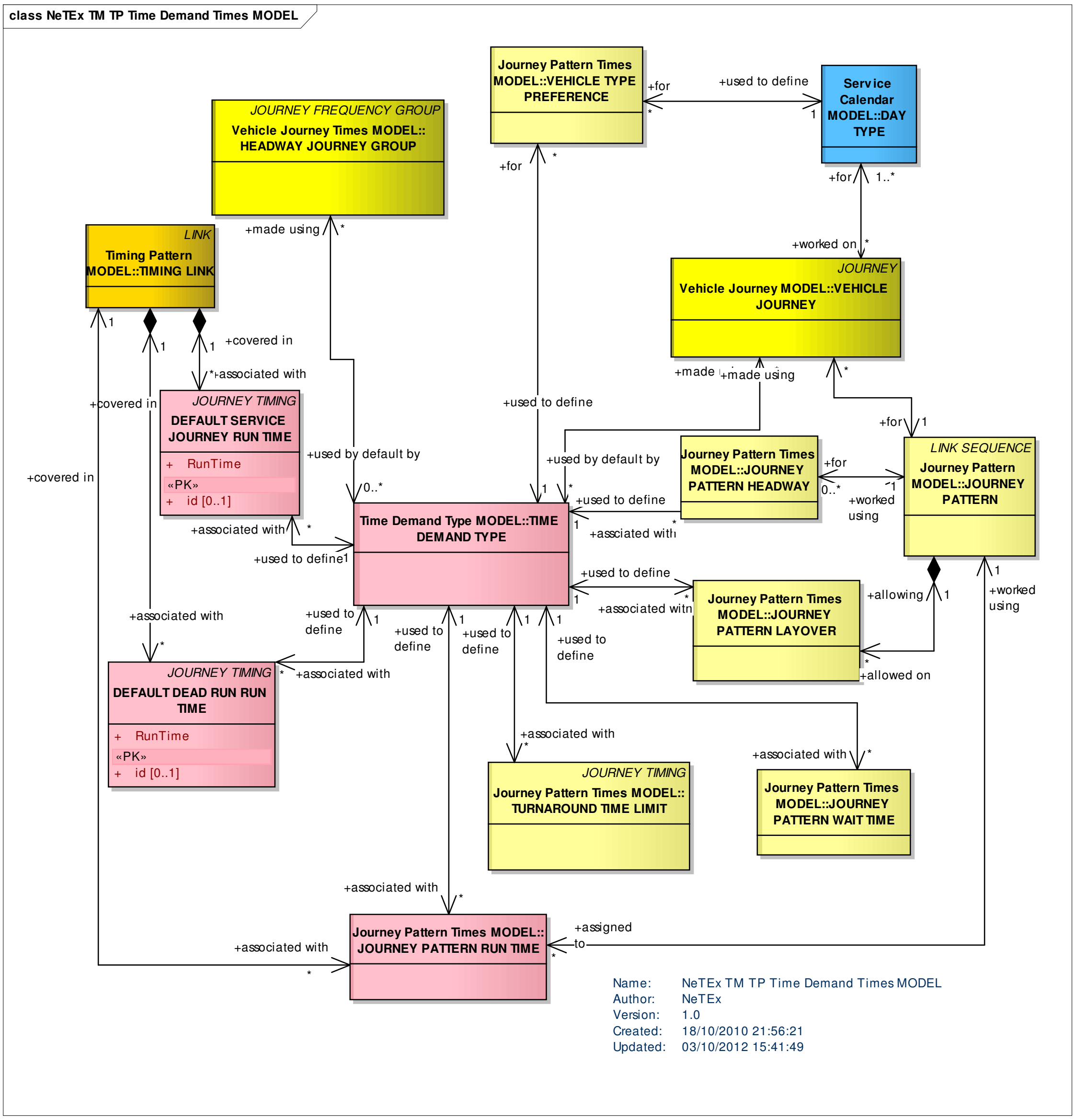

class NeTEx TM TP Time Demand Times MODEL
Journey Pattern Times MODEL::VEHICLE TYPE PREFERENCE
+for
*
+used to define
1
Service Calendar MODEL::DAY TYPE
JOURNEY FREQUENCY GROUP
Vehicle Journey Times MODEL:: HEADWAY JOURNEY GROUP
+for
*
+for
1..*
+made using
*
LINK
Timing Pattern MODEL::TIMING LINK
+worked on
*
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
1
1
1
+covered in
*+associated with
+made
+made using
*
+covered in
JOURNEY TIMING
DEFAULT SERVICE JOURNEY RUN TIME
+ RunTime
«PK»
+ id [0..1]
+used to define
+for
1
+covered in
+used by default by
+used by default by
Journey Pattern Times MODEL::JOURNEY PATTERN HEADWAY
+for
LINK SEQUENCE
Journey Pattern MODEL::JOURNEY PATTERN
0..*
1
*
0..*
+used to define
+worked using
+associated with
*
Time Demand Type MODEL::TIME DEMAND TYPE
1
+assciated with
*
+used to define1
+used to define
1
+worked using
+associated with
+used to define
1
*
+associated with
Journey Pattern Times MODEL::JOURNEY PATTERN LAYOVER
+allowing
1
+used to define
1
1
+used to define
1
+used to define
*
*
+allowed on
JOURNEY TIMING
DEFAULT DEAD RUN RUN TIME
*
+associated with
+ RunTime
«PK»
+ id [0..1]
+associated with
*
+associated with
*
JOURNEY TIMING
Journey Pattern Times MODEL:: TURNAROUND TIME LIMIT
Journey Pattern Times MODEL::JOURNEY PATTERN WAIT TIME
+associated with
*
+assigned to
Journey Pattern Times MODEL:: JOURNEY PATTERN RUN TIME
*
+associated with
*
Name: NeTEx TM TP Time Demand Times MODEL
Author: NeTEx
Version: 1.0
Created: 18/10/2010 21:56:21
Updated: 03/10/2012 15:41:49

**Figure 35 —Time Demand Times – Conceptual MODEL (UML)**

A set of DEFAULT RUN TIMEs (for SERVICE JOURNEYs and DEAD RUNs) may be recorded for any TIMING LINK, one run time corresponding to one TIME DEMAND TYPE. If the TIMING LINK is used by several JOURNEY PATTERNs, the default times may be applied any time it is covered by a VEHICLE JOURNEY, regardless the JOURNEY PATTERN.

A more precise control of run times is possible: a JOURNEY PATTERN RUN TIME is a run time for a TIMING LINK that will be valid only for VEHICLE JOURNEYs made on the specified JOURNEY PATTERN. It will override the default run times for this TIMING LINK.

JOURNEY PATTERN RUN TIMEs are sets of run times for different TIME DEMAND TYPEs. The TIME DEMAND TYPE for a particular VEHICLE JOURNEY is used to select the appropriate time, from the set recorded for a particular TIMING LINK belonging to the JOURNEY PATTERN covered.

A JOURNEY PATTERN WAIT TIME may be recorded to define the time a vehicle will have to wait at a specified TIMING POINT, e.g. to allow a large number of passengers to board or alight, or to wait for a connecting vehicle on another LINE. These wait times depend on the JOURNEY PATTERN that the VEHICLE JOURNEY covers, and on the TIME DEMAND TYPE.

JOURNEY PATTERN WAIT TIMEs may occur on DEAD RUNs also, e.g. at a certain TIMING POINT in a DEAD RUN PATTERN where the driver will be relieved.

A minimum layover time may be defined separately at the end of each JOURNEY PATTERN. This will be stored in the JOURNEY PATTERN LAYOVER entity, depending on a TIME DEMAND TYPE.

A turnaround time is the time taken by a vehicle to proceed from the end of a ROUTE to the start of another. The TURNAROUND TIME LIMIT is dependent on a TIME DEMAND TYPE and is stored against a pair of TIMING POINTs.

The VEHICLE TYPE PREFERENCE, depending on a particular SERVICE JOURNEY PATTERN, defines a priority 'rank' given for each VEHICLE TYPE, for each DAY TYPE and TIME DEMAND TYPE.

Lastly, HEADWAY JOURNEY GROUP and JOURNEY PATTERN HEADWAY, used to define services based on headway frequencies, are both potentially depending on TIME DEMAND TYPE.

#### 7.2.3.2 Time Demand Times – Physical Model

The following figure shows the TIME DEMAND TIMEs physical model.

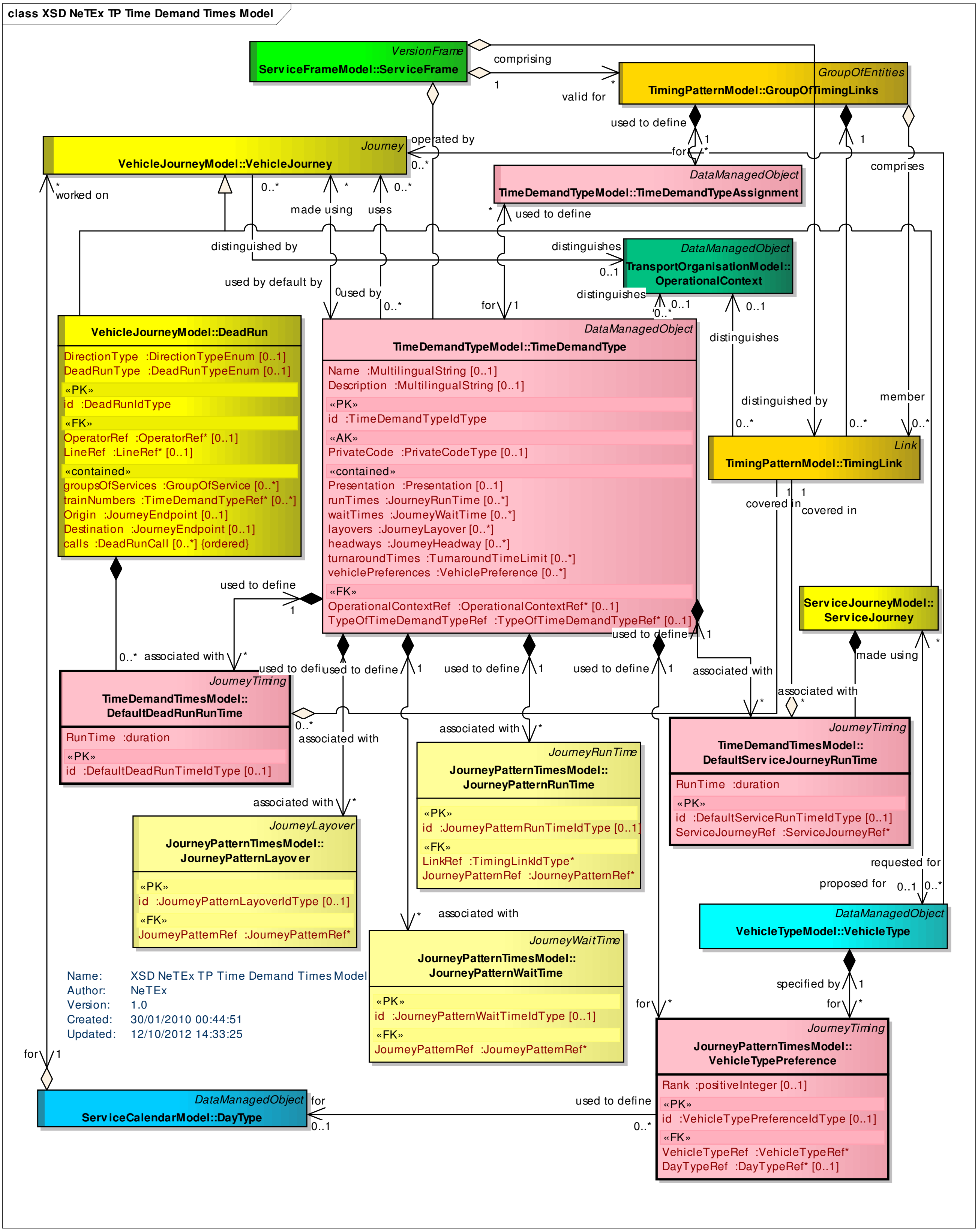

class XSD NeTEx TP Time Demand Times Model
VersionFrame
ServiceFrameModel::ServiceFrame
GroupOfEntities
TimingPatternModel::GroupOfTimingLinks
Journey
VehicleJourneyModel::VehicleJourney
DataManagedObject
TimeDemandTypeModel::TimeDemandTypeAssignment
DataManagedObject
TransportOrganisationModel::OperationalContext
VehicleJourneyModel::DeadRun
DirectionType :DirectionTypeEnum [0..1]
DeadRunType :DeadRunTypeEnum [0..1]
«PK»
id :DeadRunIdType
«FK»
OperatorRef :OperatorRef* [0..1]
LineRef :LineRef* [0..1]
«contained»
groupsOfServices :GroupOfService [0..*]
trainNumbers :TimeDemandTypeRef* [0..*]
Origin :JourneyEndpoint [0..1]
Destination :JourneyEndpoint [0..1]
calls :DeadRunCall [0..*] {ordered}
DataManagedObject
TimeDemandTypeModel::TimeDemandType
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
«PK»
id :TimeDemandTypeIdType
«AK»
PrivateCode :PrivateCodeType [0..1]
«contained»
Presentation :Presentation [0..1]
runTimes :JourneyRunTime [0..*]
waitTimes :JourneyWaitTime [0..*]
layovers :JourneyLayover [0..*]
headways :JourneyHeadway [0..*]
turnaroundTimes :TurnaroundTimeLimit [0..*]
vehiclePreferences :VehiclePreference [0..*]
«FK»
OperationalContextRef :OperationalContextRef* [0..1]
TypeOfTimeDemandTypeRef :TypeOfTimeDemandTypeRef* [0..1]
Link
TimingPatternModel::TimingLink
ServiceJourneyModel::ServiceJourney
JourneyTiming
TimeDemandTimesModel::DefaultDeadRunRunTime
RunTime :duration
«PK»
id :DefaultDeadRunTimeIdType [0..1]
JourneyRunTime
JourneyPatternTimesModel::JourneyPatternRunTime
«PK»
id :JourneyPatternRunTimeIdType [0..1]
«FK»
LinkRef :TimingLinkIdType*
JourneyPatternRef :JourneyPatternRef*
JourneyTiming
TimeDemandTimesModel::DefaultServiceJourneyRunTime
RunTime :duration
«PK»
id :DefaultServiceRunTimeIdType [0..1]
ServiceJourneyRef :ServiceJourneyRef*
JourneyLayover
JourneyPatternTimesModel::JourneyPatternLayover
«PK»
id :JourneyPatternLayoverIdType [0..1]
«FK»
JourneyPatternRef :JourneyPatternRef*
JourneyWaitTime
JourneyPatternTimesModel::JourneyPatternWaitTime
«PK»
id :JourneyPatternWaitTimeIdType [0..1]
«FK»
JourneyPatternRef :JourneyPatternRef*
DataManagedObject
VehicleTypeModel::VehicleType
JourneyTiming
JourneyPatternTimesModel::VehicleTypePreference
Rank :positiveInteger [0..1]
«PK»
id :VehicleTypePreferenceIdType [0..1]
«FK»
VehicleTypeRef :VehicleTypeRef*
DayTypeRef :DayTypeRef* [0..1]
DataManagedObject
ServiceCalendarModel::DayType
comprising
valid for
used to define
for
operated by
comprises
worked on
made using
uses
used to define
distinguished by
distinguishes
used by default by
used by
distinguishes
distinguishes
distinguished by
member
covered in
covered in
used to define
associated with
used to define
used to define
used to define
used to define
associated with
associated with
associated with
made using
associated with
associated with
requested for
proposed for
associated with
specified by
for
for
for
used to define
Name: XSD NeTEx TP Time Demand Times Model
Author: NeTEx
Version: 1.0
Created: 30/01/2010 00:44:51
Updated: 12/10/2012 14:33:25

Figure 36 — Time Demand Times – Physical Model (UML)

#### 7.2.3.2.1 Using Time Demand Times – Physical Model

The following figure shows how timing information is associated with TIMING POINTs and TIMING LINKs for a given TIME DEMAND TYPE. The ***JourneyTiming*** element holds common association properties: it are specialised to create a number of different types of timing.

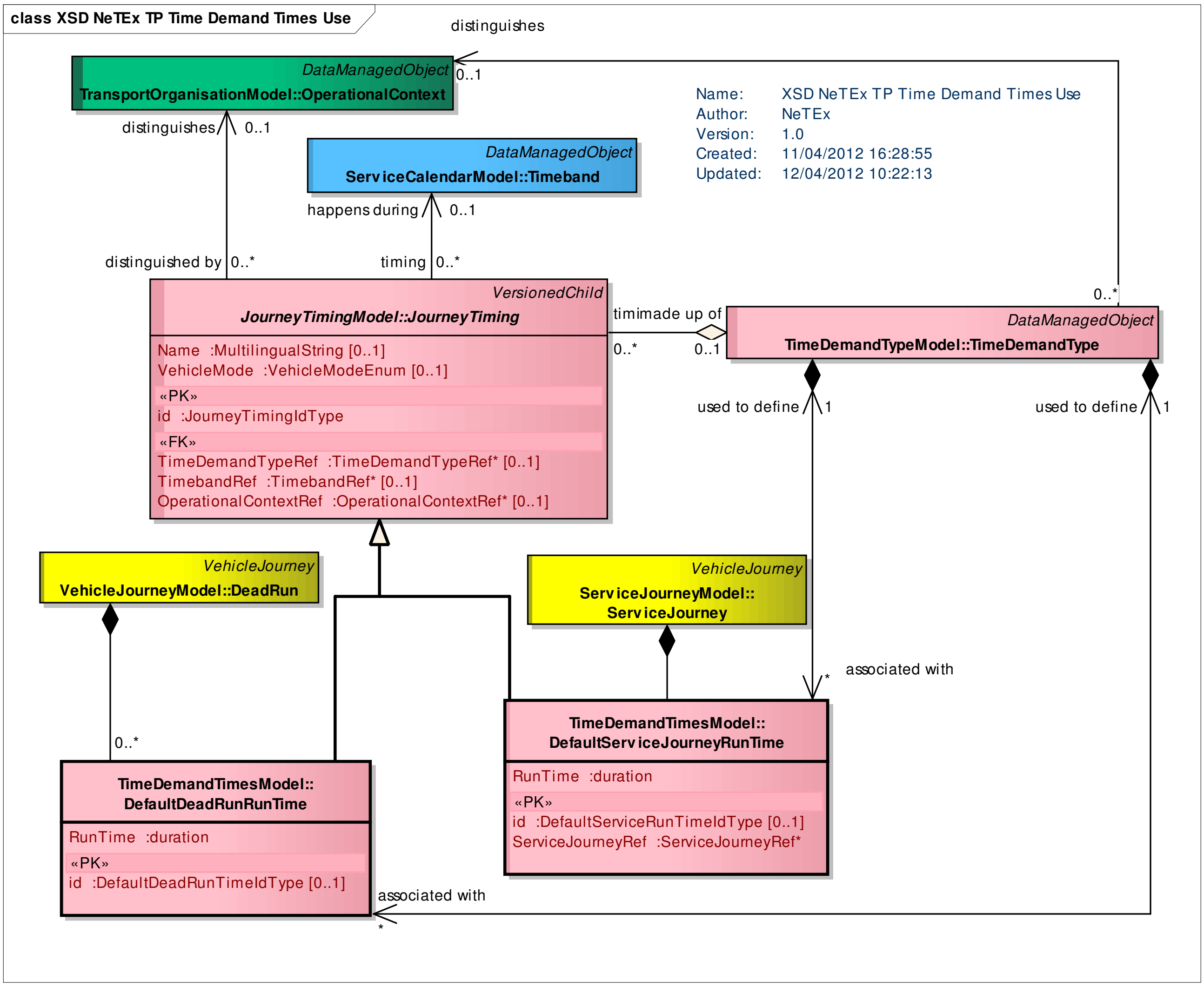

class XSD NeTEx TP Time Demand Times Use
distinguishes
DataManagedObject
TransportOrganisationModel::OperationalContext
0..1
Name: XSD NeTEx TP Time Demand Times Use
Author: NeTEx
Version: 1.0
Created: 11/04/2012 16:28:55
Updated: 12/04/2012 10:22:13
distinguishes
0..1
DataManagedObject
ServiceCalendarModel::Timeband
happens during
0..1
distinguished by
0..*
timing
0..*
VersionedChild
JourneyTimingModel::JourneyTiming
Name :MultilingualString [0..1]
VehicleMode :VehicleModeEnum [0..1]
«PK»
id :JourneyTimingIdType
«FK»
TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
TimebandRef :TimebandRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
timimade up of
0..*
0..1
0..*
DataManagedObject
TimeDemandTypeModel::TimeDemandType
used to define
1
used to define
1
VehicleJourney
VehicleJourneyModel::DeadRun
VehicleJourney
ServiceJourneyModel::ServiceJourney
associated with
*
0..*
TimeDemandTimesModel::DefaultServiceJourneyRunTime
RunTime :duration
«PK»
id :DefaultServiceRunTimeIdType [0..1]
ServiceJourneyRef :ServiceJourneyRef*
TimeDemandTimesModel::DefaultDeadRunRunTime
RunTime :duration
«PK»
id :DefaultDeadRunTimeIdType [0..1]
associated with
*

**Figure 37 — Time Demand Times Use – Physical Model (UML)**

### 7.2.3.3 Time Demand Times – Attributes and XSD

#### 7.2.3.3.1 DefaultServiceJourneyRunTime – Model Element

The default time taken by a vehicle to traverse a TIMING LINK during a SERVICE JOURNEY, for a specified TIME DEMAND TYPE. This time may be superseded by the JOURNEY PATTERN RUN TIME or VEHICLE JOURNEY RUN TIME if these exist.

Table 19 – *DefaultServiceJourneyRunTime* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyTiming* | ::> | DEFAULT SERVICE JOURNEY RUN TIME inherits from JOURNEY TIMING. |
| «PK» | ***id*** | *DefaultService-RunTimeIdType* | 0:1 | Identifier of DEFAULT SERVICE JOURNEY RUN TIME. |
| | ***RunTime*** | *xsd:duration* | 1:1 | Run time as a duration. |
| «PK» | ***ServiceJourneyRef*** | *ServiceJourneyRef* | 1:1 | SERVICE JOURNEY with which DEFAULT RUN TIME is associated. |

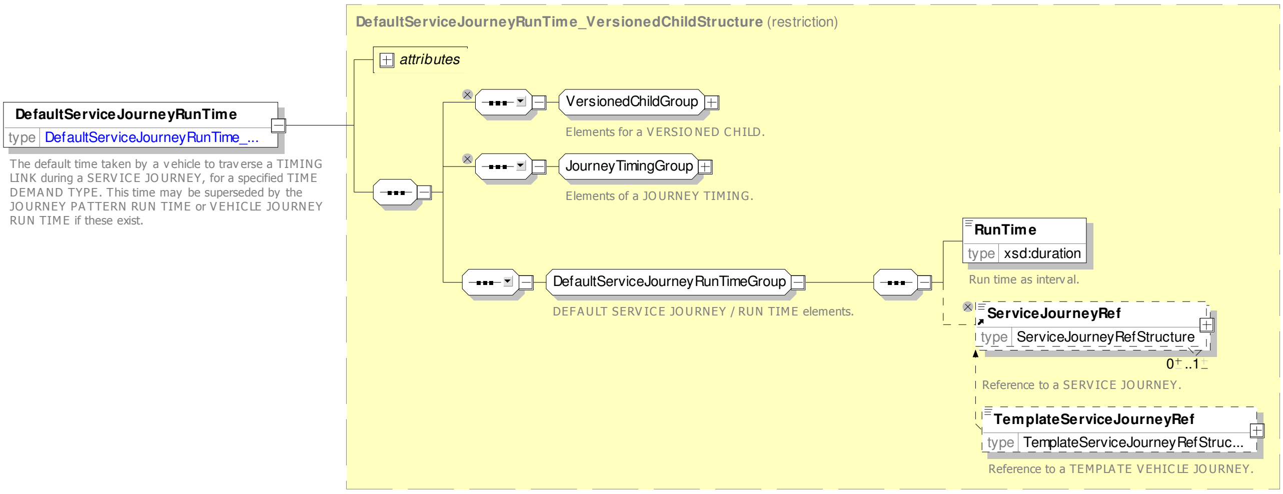

DefaultServiceJourneyRunTime_VersionedChildStructure (restriction)
attributes
DefaultServiceJourneyRunTime
type DefaultServiceJourneyRunTime_...
The default time taken by a vehicle to traverse a TIMING LINK during a SERVICE JOURNEY, for a specified TIME DEMAND TYPE. This time may be superseded by the JOURNEY PATTERN RUN TIME or VEHICLE JOURNEY RUN TIME if these exist.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
DefaultServiceJourneyRunTimeGroup
DEFAULT SERVICE JOURNEY / RUN TIME elements.
RunTime
type xsd:duration
Run time as interval.
ServiceJourneyRef
type ServiceJourneyRefStructure
0..1
Reference to a SERVICE JOURNEY.
TemplateServiceJourneyRef
type TemplateServiceJourneyRefStruc...
Reference to a TEMPLATE VEHICLE JOURNEY.

**Figure 38 — *DefaultServiceJourneyRunTime* — XSD**

### 7.2.3.3.2 DefaultDeadRunRunTime – Model Element

The time taken to traverse a TIMING LINK during a DEAD RUN, for a specified TIME DEMAND TYPE. This time may be superseded by the JOURNEY PATTERN RUN TIME or VEHICLE JOURNEY RUN TIME if these exist.

Table 20 – *DefaultDeadRunRunTime* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyTiming* | ::> | DEFAULT DEAD RUN RUN TIME inherits from JOURNEY TIMING. |
| «PK» | ***id*** | *DefaultDeadRunTimeIdType* | 0:1 | Identifier of DEFAULT DEAD RUN RUN TIME. |
| | ***RunTime*** | *xsd:duration* | 1:1 | Default time to make a DEAD RUN. |

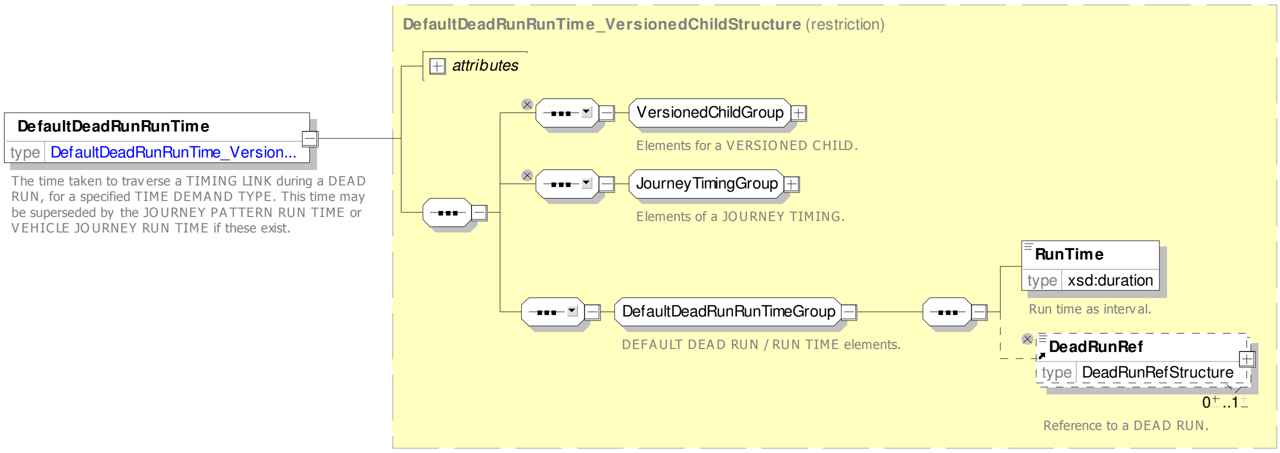

DefaultDeadRunRunTime_VersionedChildStructure (restriction)
attributes
DefaultDeadRunRunTime
type DefaultDeadRunRunTime_Version...
The time taken to traverse a TIMING LINK during a DEAD RUN, for a specified TIME DEMAND TYPE. This time may be superseded by the JOURNEY PATTERN RUN TIME or VEHICLE JOURNEY RUN TIME if these exist.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
RunTime
type xsd:duration
Run time as interval.
DefaultDeadRunRunTimeGroup
DEFAULT DEAD RUN / RUN TIME elements.
DeadRunRef
type DeadRunRefStructure
0..1
Reference to a DEAD RUN.

Figure 39 — *DefaultDeadRunRunTimeTime* — XSD

## 7.2.3.4 Time Demand Profile —Physical Model

The following figure introduces the Physical Model for a TIME DEMAND PROFILE. It provides a convenient way of handling a set of JOURNEY PATTERN RUN TIMEs using the same TIME DEMAND TYPE.

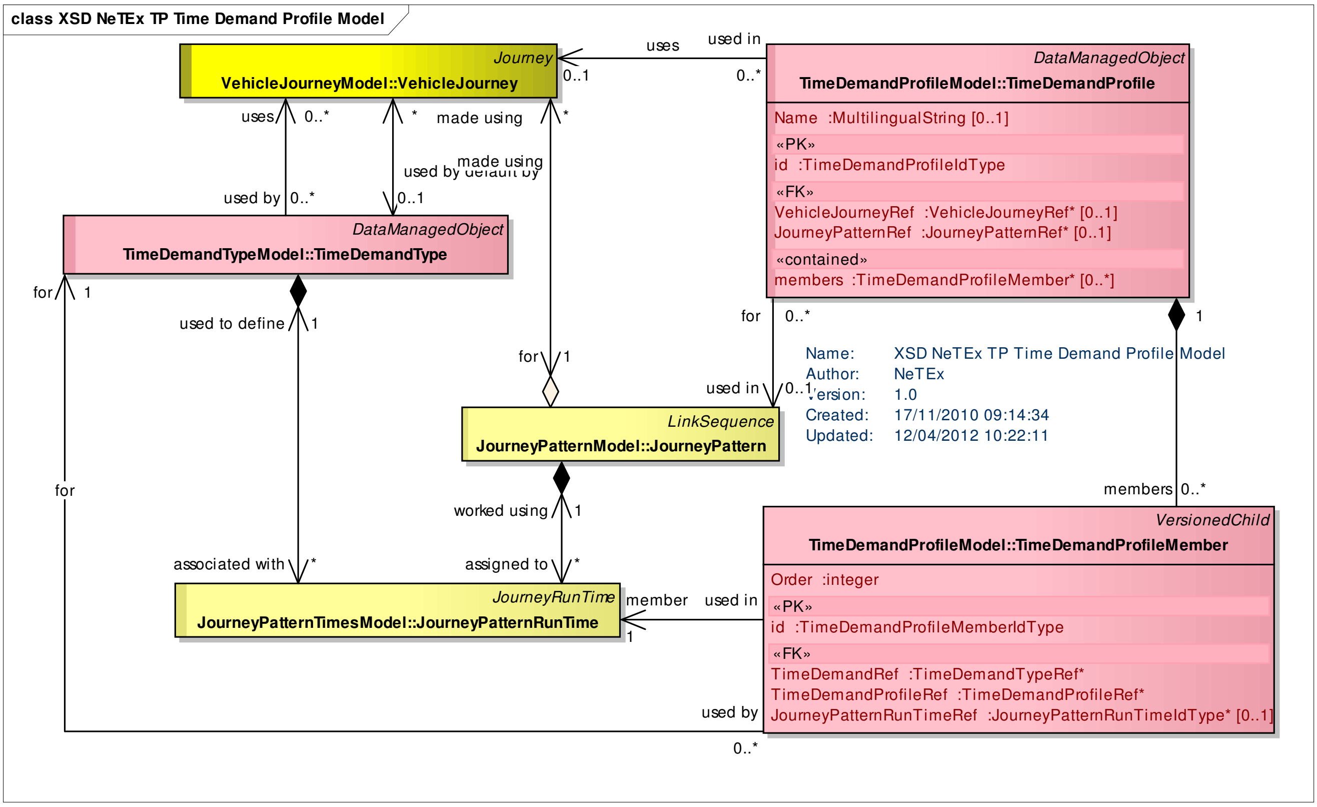

class XSD NeTEx TP Time Demand Profile Model
Journey
VehicleJourneyModel::VehicleJourney
DataManagedObject
TimeDemandProfileModel::TimeDemandProfile
Name :MultilingualString [0..1]
«PK»
id :TimeDemandProfileIdType
«FK»
VehicleJourneyRef :VehicleJourneyRef* [0..1]
JourneyPatternRef :JourneyPatternRef* [0..1]
«contained»
members :TimeDemandProfileMember* [0..*]
uses
used in
0..1
0..*
uses
0..*
*
made using
*
used by
made using
used by default by
0..*
0..1
DataManagedObject
TimeDemandTypeModel::TimeDemandType
for
1
used to define
1
for
1
for
0..*
1
used in
0..1
Name: XSD NeTEx TP Time Demand Profile Model
Author: NeTEx
Version: 1.0
Created: 17/11/2010 09:14:34
Updated: 12/04/2012 10:22:11
LinkSequence
JourneyPatternModel::JourneyPattern
for
worked using
1
members
0..*
associated with
*
assigned to
*
VersionedChild
TimeDemandProfileModel::TimeDemandProfileMember
Order :integer
«PK»
id :TimeDemandProfileMemberIdType
«FK»
TimeDemandRef :TimeDemandTypeRef*
TimeDemandProfileRef :TimeDemandProfileRef*
JourneyPatternRunTimeRef :JourneyPatternRunTimeIdType* [0..1]
JourneyRunTime
JourneyPatternTimesModel::JourneyPatternRunTime
member
used in
1
used by
0..*

Figure 40 — Time Demand Profile – Physical Model (UML)

## 7.2.3.5 Time Demand Profile – Attributes and XSD

### 7.2.3.5.1 TimeDemandProfile – Model Element

An indicator of traffic conditions or other factors which may affect vehicle run or wait times. It may be entered directly by the scheduler or defined by the use of TIME BANDs.

**Table 21 – *TimeDemandProfile* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| *::>* | ***::>*** | *DataManagedObject* | *::>* | TIME DEMAND PROFILE inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *TimeDemandProfileIdType* | 1:1 | Identifier of TIME DEMAND PROFILE. |
| | ***Name*** | *MultilingualString* | 0:1 | VEHICLE JOURNEY associated with TIME DEMAND PROFILE. |
| «FK» | ***TimeDemandTypeRef*** | *TimeDemandTypeRef* | 0:1 | TIME DEMAND TYPE associated with TIME DEMAND PROFILE. |
| «FK» | ***TimebandRef*** | *TimebandRef* | 0:1 | Reference to a TIME BAND. |
| «FK» | ***JourneyPatternRef*** | *JourneyPatternRef* | 0:1 | JOURNEY PATTERN associated with TIME DEMAND PROFILE. |
| «FK» | ***VehicleJourneyRef*** | *VehicleJourneyRef* | 0:1 | Identifier of TIME DEMAND PROFILE. |
| «cntd» | ***members*** | *TimeDemand-ProfileMember* | 0:* | Members of TIME DEMAND PROFILE. |

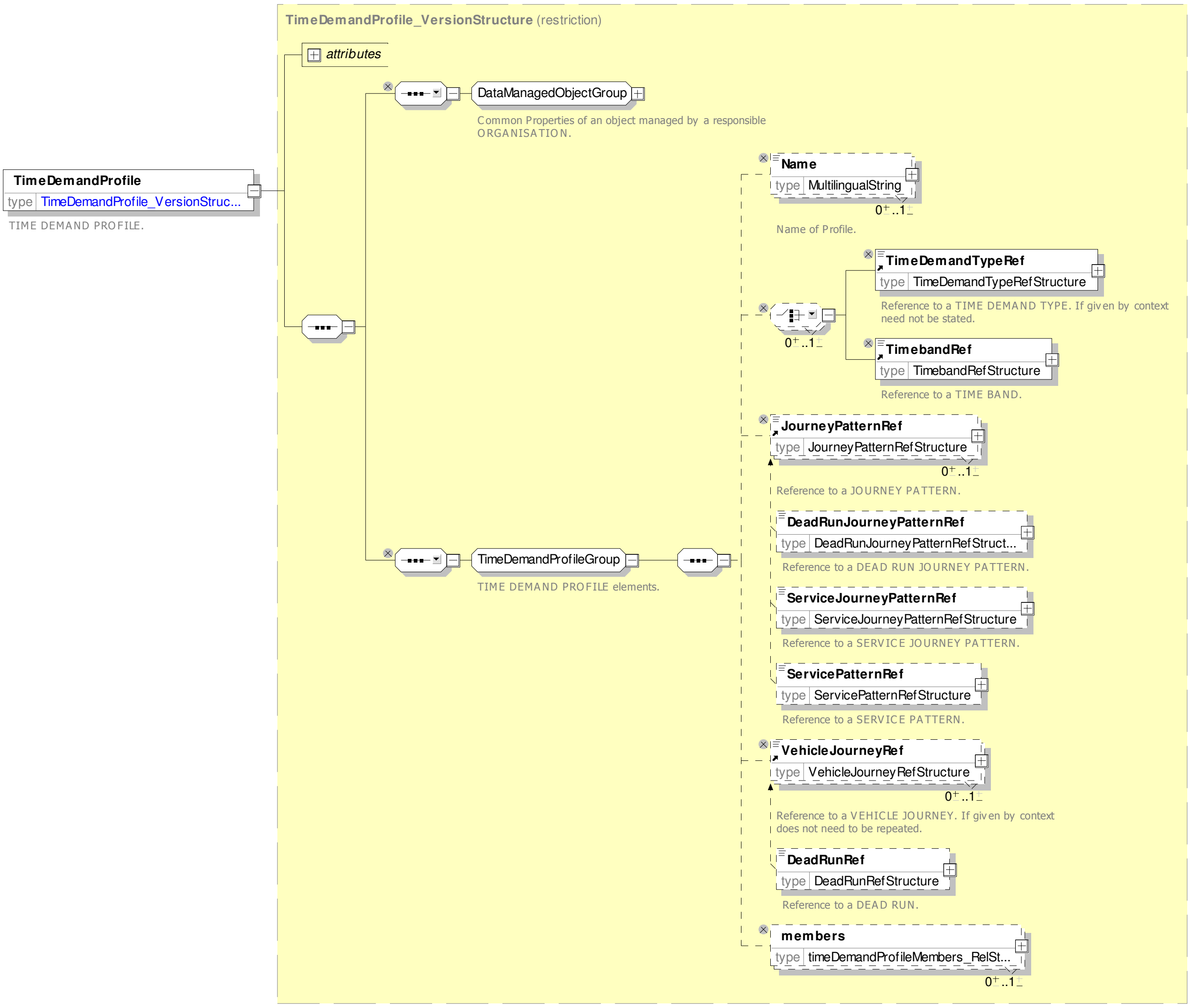

TimeDemandProfile_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
TimeDemandProfile
type TimeDemandProfile_VersionStruc...
TIME DEMAND PROFILE.
Name
type MultilingualString
0..1
Name of Profile.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
0..1
TimebandRef
type TimebandRefStructure
Reference to a TIME BAND.
JourneyPatternRef
type JourneyPatternRefStructure
0..1
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.
ServiceJourneyPatternRef
type ServiceJourneyPatternRefStructure
Reference to a SERVICE JOURNEY PATTERN.
ServicePatternRef
type ServicePatternRefStructure
Reference to a SERVICE PATTERN.
TimeDemandProfileGroup
TIME DEMAND PROFILE elements.
VehicleJourneyRef
type VehicleJourneyRefStructure
0..1
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.
members
type timeDemandProfileMembers_RelSt...
0..1

Figure 41 — *TimeDemandProfile* — XSD

### 7.2.3.5.2 TimeDemandProfileMember – Model Element

A members of the TIME DEMAND PROFILE described above.

Table 22 – *TimeDemandProfileMember* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *GroupMember* | *::>* | TIME DEMAND PROFILE inherits from GROUP MEMBER. |
| «PK» | ***id*** | *TimeDemandProfile-MemberIdType* | 1:1 | Identifier of TIME DEMAND PROFILE MEMBER. |
| | ***order*** | *xsd:integer* | 1:1 | Order of member within profile. |
| «FK» | ***TimeDemand-ProfileRef*** | *TimeDemandProfileRef* | 1:1 | TIME DEMAND PROFILE of PROFILE MEMBER. |
| «PK» | ***TimeDemandRef*** | *TimeDemandTypeRef* | 1:1 | TIME DEMAND TYPE of PROFILE MEMBER. |
| «FK» | ***JourneyPattern-*** | *JourneyPatternRunTime-* | 0:1 | JOURNEY PATTERN RUN TIME of PROFILE |

| | | | | |
|---|---|---|---|---|
| | ***RunTimeRef*** | *Ref* | | MEMBER. |

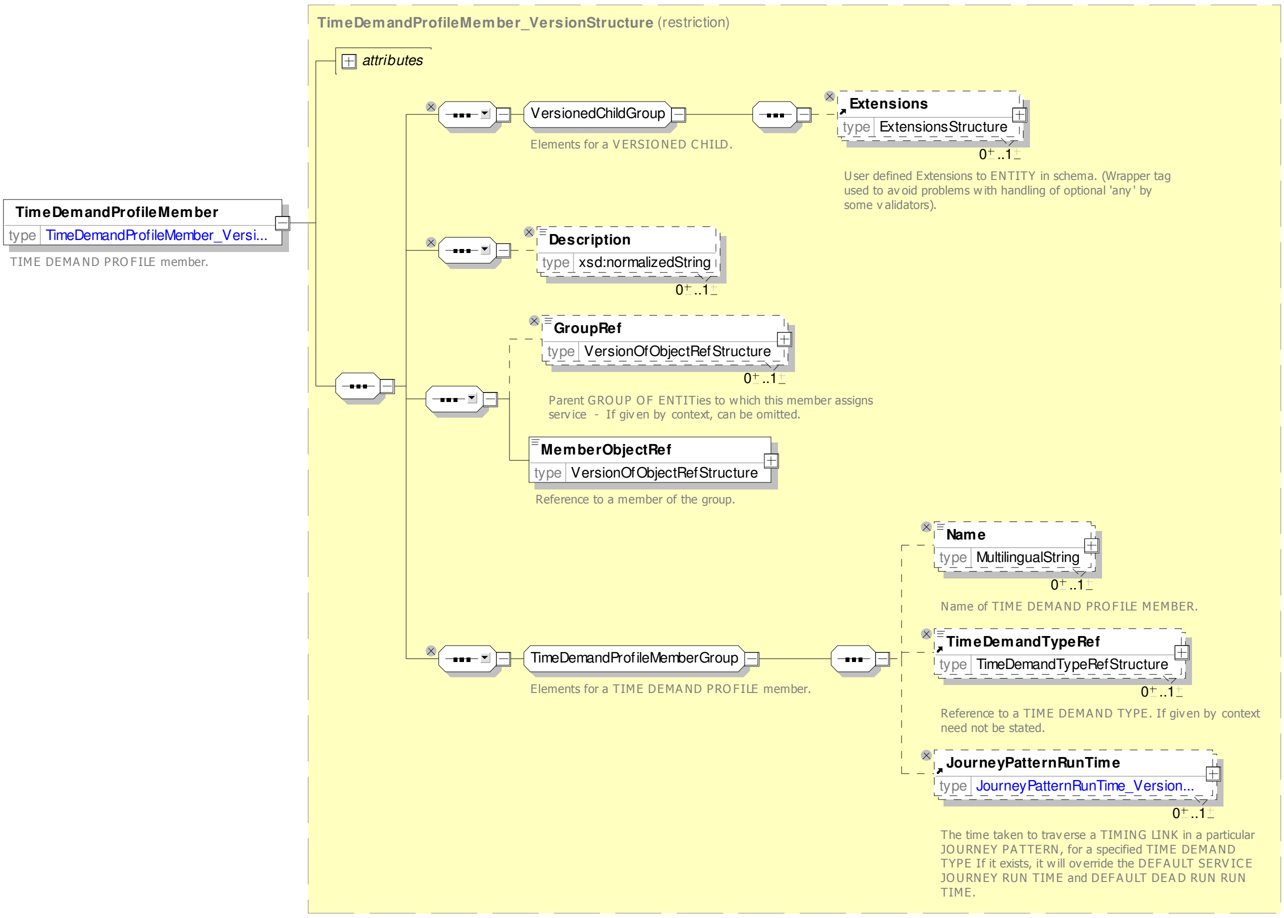

TimeDemandProfileMember_VersionStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
Extensions
type ExtensionsStructure
0..1
User defined Extensions to ENTITY in schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
TimeDemandProfileMember
type TimeDemandProfileMember_Versi...
TIME DEMAND PROFILE member.
Description
type xsd:normalizedString
0..1
GroupRef
type VersionOfObjectRefStructure
0..1
Parent GROUP OF ENTITies to which this member assigns service - If given by context, can be omitted.
MemberObjectRef
type VersionOfObjectRefStructure
Reference to a member of the group.
Name
type MultilingualString
0..1
Name of TIME DEMAND PROFILE MEMBER.
TimeDemandProfileMemberGroup
Elements for a TIME DEMAND PROFILE member.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
0..1
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
JourneyPatternRunTime
type JourneyPatternRunTime_Version...
0..1
The time taken to traverse a TIMING LINK in a particular JOURNEY PATTERN, for a specified TIME DEMAND TYPE If it exists, it will override the DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.

**Figure 42 — TimeDemandProfileMember — XSD**

### 7.2.3.6 XML Example of Time Demand Timings

The following code fragment shows a TIME DEMAND TYPE with several RUN TIMEs associated with it for particular links.

For EXAMPLE

```
<timeDemandTypes>
    <TimeDemandType version="any" id="kx:tdt_45_Day">
        <Name>Normal Time demand </Name>
        <runTimes>
            <!--- =====Outbound=== -->
            <JourneyRunTime version="any" id="kx::jprt_sjp_L24o_ssp_001-ssp_002_tdt_45">
                <Name>Run Time from 001 to 002: 10 minutes</Name>
                <TimingLinkRef version="any" ref="kx:tl_ssp_001_to_ssp_002"/>
                <RunTime>PT10M</RunTime>
            </JourneyRunTime>
            <JourneyRunTime version="any" id="kx:jprt_sjp_L24o_ssp_002-ssp_077_tdt_45">
                <Name>Run Time from 002 to 007: 20 minutes</Name>
                <TimingLinkRef version="any" ref="kx:tl_ssp_002_to_ssp_077"/>
                <RunTime>PT20M</RunTime>
            </JourneyRunTime>
            <JourneyRunTime version="any" id="kx:jprt_sjp_L24o_ssp_077-ssp_002_tdt_45">
                <Name>Time from 077 to 002: 20 minutes</Name>
                <TimeDemandTypeRef version="any" ref="kx:tdt_45_Day"/>
                <TimingLinkRef version="any" ref="kx:tl_ssp_077_to_ssp_002"/>
```

```xml
                <RunTime>PT20M</RunTime>
            </JourneyRunTime>
            <JourneyRunTime version="any" id="kx:jprt_sjp_L24o_ssp_002-ssp_001_tdt_45">
                <Name>Time from 002 to 001: 10 minutes</Name>
                <TimeDemandTypeRef version="any" ref="kx:tdt_45_Day"/>
                <TimingLinkRef version="any" ref="kx:tl_ssp_002_to_ssp_001"/>
                <RunTime>PT10M</RunTime>
            </JourneyRunTime>
        </runTimes>
    </TimeDemandType>
    <TimeDemandType version="any" id="kx:tdt_46_Night">
        <Name>Night Time demand </Name>
    </TimeDemandType>
</timeDemandTypes>
```

### 7.2.4 Journey Timing

#### 7.2.4.1 JOURNEY TIMING – Conceptual MODEL

The JOURNEY TIMING model defines common properties for timing elements that can be specialised in the VEHICLE JOURNEY and JOURNEY PATTERN timing models.

A JOURNEY TIMING provides an abstract type for a number of different specialised types of timing information:

- JOURNEY LAYOVER;
- JOURNEY WAIT TIME;
- JOURNEY HEADWAY;
- JOURNEY RUN TIME;
- TURNAROUND TIME LIMIT;
- DEFAULT DEAD RUN TIME;
- DEFAULT SERVICE JOURNEY RUN TIME.

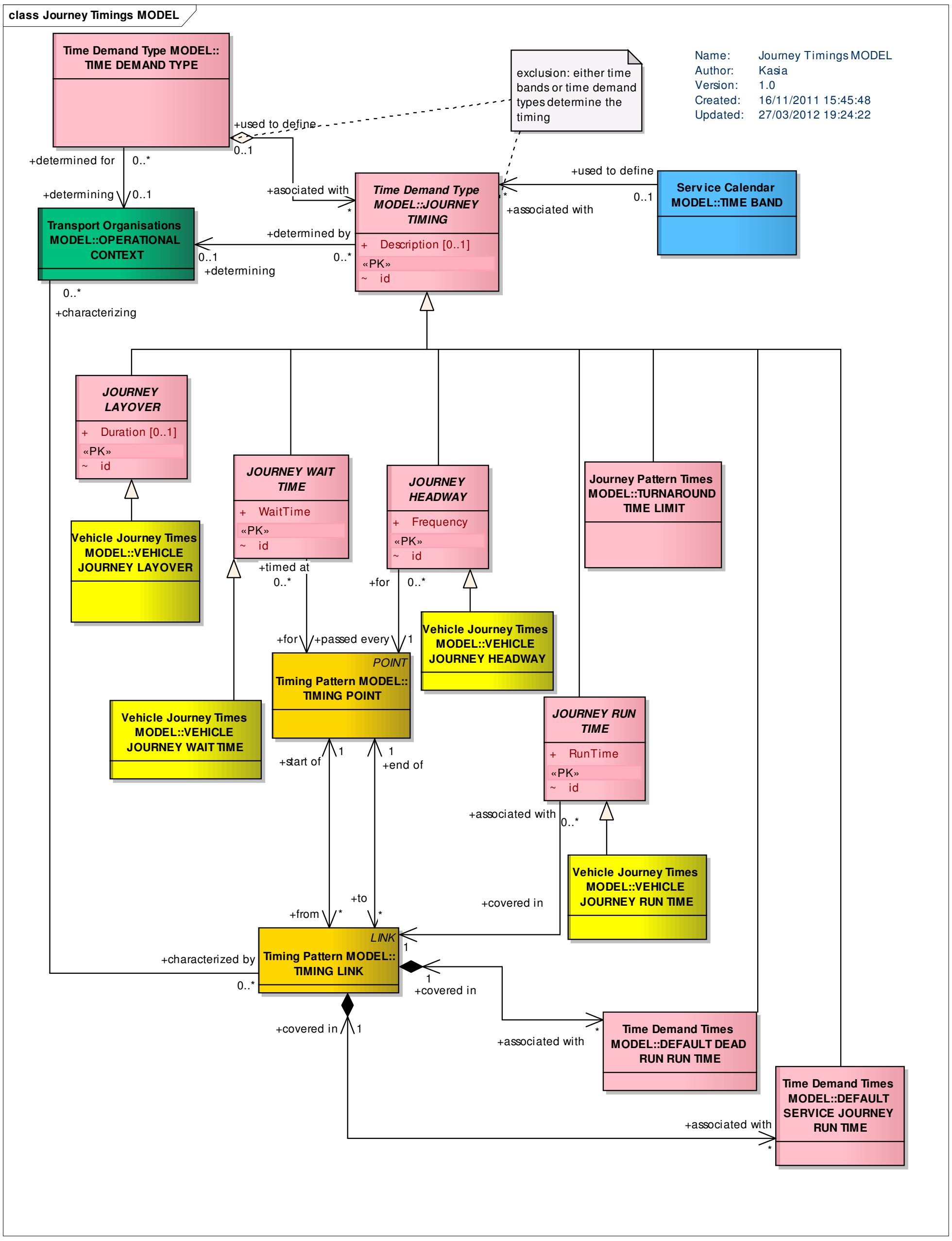

class Journey Timings MODEL
Time Demand Type MODEL::TIME DEMAND TYPE
exclusion: either time bands or time demand types determine the timing
Name: Journey Timings MODEL
Author: Kasia
Version: 1.0
Created: 16/11/2011 15:45:48
Updated: 27/03/2012 19:24:22
+used to define
0..1
+determined for
0..*
+determining
0..1
+associated with
*
+used to define
0..1
+associated with
*
Time Demand Type MODEL::JOURNEY TIMING
+ Description [0..1]
«PK»
~ id
Service Calendar MODEL::TIME BAND
Transport Organisations MODEL::OPERATIONAL CONTEXT
+determined by
0..1
+determining
0..*
0..*
+characterizing
JOURNEY LAYOVER
+ Duration [0..1]
«PK»
~ id
JOURNEY WAIT TIME
+ WaitTime
«PK»
~ id
JOURNEY HEADWAY
+ Frequency
«PK»
~ id
Journey Pattern Times MODEL::TURNAROUND TIME LIMIT
Vehicle Journey Times MODEL::VEHICLE JOURNEY LAYOVER
+timed at
0..*
+for
0..*
Vehicle Journey Times MODEL::VEHICLE JOURNEY HEADWAY
+for
+passed every
1
POINT
Timing Pattern MODEL::TIMING POINT
Vehicle Journey Times MODEL::VEHICLE JOURNEY WAIT TIME
JOURNEY RUN TIME
+ RunTime
«PK»
~ id
+start of
1
1
+end of
+associated with
0..*
Vehicle Journey Times MODEL::VEHICLE JOURNEY RUN TIME
+from
*
+to
*
+covered in
LINK
Timing Pattern MODEL::TIMING LINK
+characterized by
0..*
1
1
+covered in
+covered in
1
+associated with
*
Time Demand Times MODEL::DEFAULT DEAD RUN RUN TIME
Time Demand Times MODEL::DEFAULT SERVICE JOURNEY RUN TIME
+associated with
*

**Figure 43 — Journey Timing – Conceptual MODEL (UML)**

### 7.2.4.1.1 Layover times

LAYOVER TIMEs describe a certain time allowance that may be given at the end of each VEHICLE JOURNEY, before starting the next one, to compensate delays or for other purposes (e.g. rest time for the driver). This “layover time” can be regarded as a buffer time, which may or may not be actually consumed in real time operation.

A minimum layover time may be defined separately at the end of each JOURNEY PATTERN. This will be stored in the JOURNEY PATTERN LAYOVER entity, depending on a TIME DEMAND TYPE.

Such standard layover times may be superseded by a layover time defined for a specific VEHICLE JOURNEY.

#### 7.2.4.1.2 Wait times

A WAIT TIME may be recorded to define the time a vehicle will have to wait at a specified TIMING POINT, e.g. to allow a large number of passengers to board or alight, or to wait for a connecting vehicle on another LINE. These wait times depend on the JOURNEY PATTERN that the VEHICLE JOURNEY covers, and on the TIME DEMAND TYPE. JOURNEY PATTERN WAIT TIMEs may occur on DEAD RUNs also, e.g. at a certain TIMING POINT in a DEAD RUN PATTERN where the driver will be relieved.

A JOURNEY WAIT TIME may be stored for each individual VEHICLE JOURNEY, at any TIMING POINT IN JOURNEY PATTERN, in the covered JOURNEY PATTERN.

A VEHICLE JOURNEY WAIT TIME, if it exists, overrides any JOURNEY PATTERN WAIT TIMEs that may have been stored for the TIMING POINT in question.

#### 7.2.4.1.3 Headway times

Headway frequency information that is available for a VEHICLE JOURNEY supersedes the JOURNEY PATTERN HEADWAY. It has to be understood as the delay between the previous and the next VEHICLE JOURNEY. This information shall be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services).

#### 7.2.4.1.4 Run times

A precise control of run times is possible by using the JOURNEY RUN TIME. It defines a run time for a TIMING LINK that will be valid only for specific VEHICLE JOURNEYs and overrides the default run times for this TIMING LINK.

JOURNEY RUN TIMEs are sets of run times for different TIME DEMAND TYPEs. The TIME DEMAND TYPE for a particular VEHICLE JOURNEY is used to select the appropriate time, from the set recorded for a particular TIMING LINK belonging to the JOURNEY PATTERN covered.

A VEHICLE JOURNEY RUN TIME is specific to one VEHICLE JOURNEY and applies to a particular TIMING LINK. If it exists, it overrides any run time that may have been stored for the TIMING LINK in question.

#### 7.2.4.1.5 Turnaround times

Another constraint to be taken into account in fixing layover times may be a maximal or minimal turnaround time. A turnaround time is the time taken by a vehicle to proceed from the end of a ROUTE to the start of another. There are some limitations for turnaround times, which are used as parameters by scheduling systems for the vehicle scheduling procedure.

The TURNAROUND TIME LIMIT is dependent on a TIME DEMAND TYPE and is stored against a pair of TIMING POINTs. These points represent a TIMING POINT where a vehicle may end a SERVICE JOURNEY and a TIMING POINT where it may start from, for a subsequent SERVICE JOURNEY. The duration of a DEAD RUN (relating the two TIMING POINTs) possibly scheduled between these two SERVICE JOURNEYs is included in the turnaround time.

#### 7.2.4.1.6 Times for Dead Runs

The path to proceed from the end point of a SERVICE JOURNEY to the start point of the following is normally described by a DEAD RUN PATTERN. However, it is often not worth to model explicitly a short movement as a DEAD RUN, covering an explicit DEAD RUN PATTERN. In such a case, a ‘minimum duration’ may be stored in the TURNAROUND TIME LIMIT, as the minimum time needed by a vehicle to cover this short path. This minimum duration will of course be superseded by the run times (plus wait times, if any) associated with an explicitly modelled DEAD RUN between the two TIMING POINTs concerned.

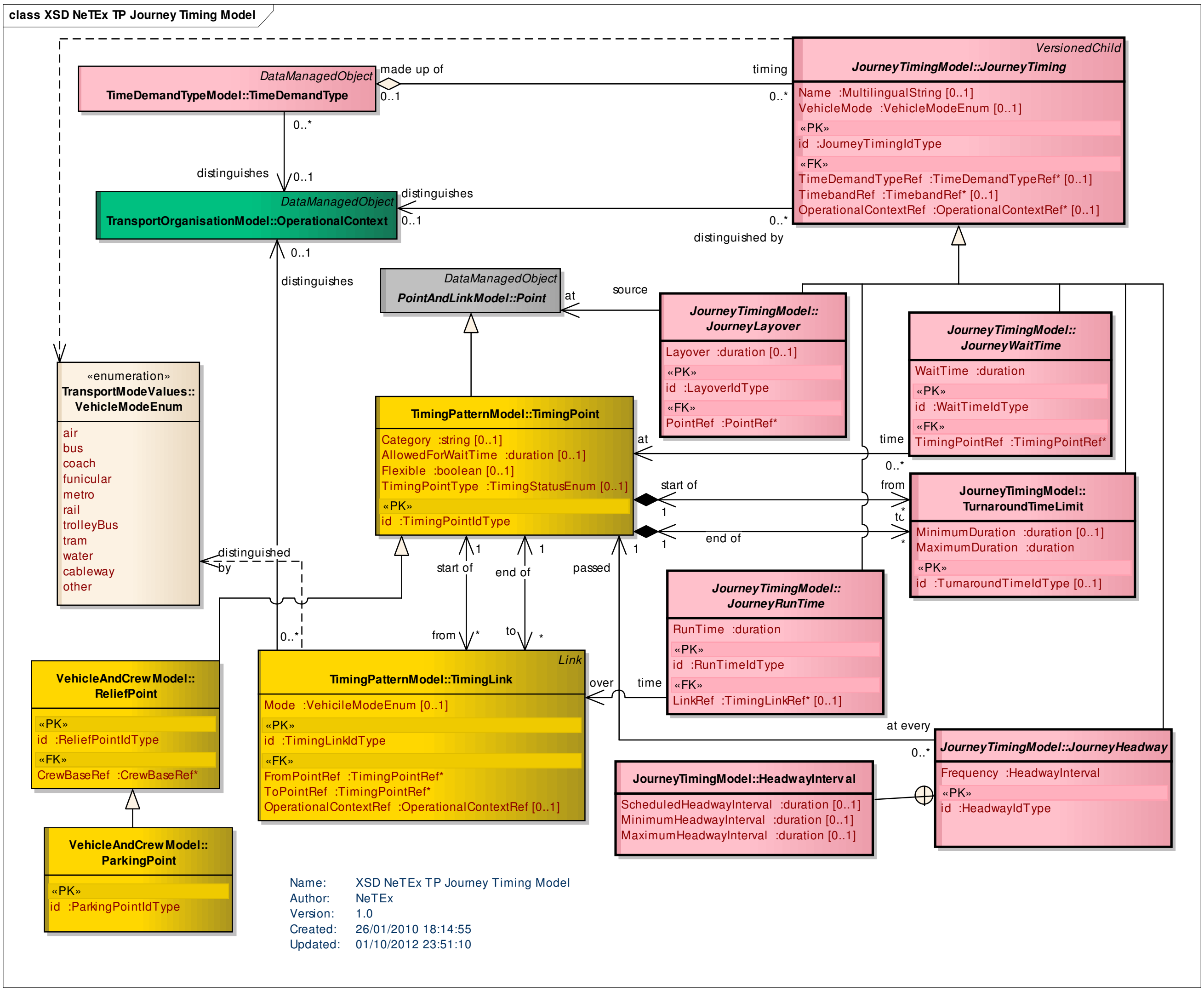
In the case of a SERVICE JOURNEY, there are STOP POINTs in the JOURNEY PATTERN where passengers can board or alight from the vehicle. Therefore, run times will probably be different if a vehicle crosses a TIMING LINK during a SERVICE JOURNEY or a DEAD RUN. Default run times are hence recorded in two different entities: DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.

Using these default run times, the timing information for each VEHICLE JOURNEY can be derived by looking for the TIMING POINTs in the associated JOURNEY PATTERN, accessing the TIMING LINKs between these TIMING POINTs and choosing the appropriate run time. These times will be found, for each TIMING LINK, in the DEFAULT SERVICE JOURNEY RUN TIME or DEFAULT DEAD RUN RUN TIME entity, according to the type of journey.

The choice among the run times recorded for one TIMING LINK is determined by the TIME DEMAND TYPE which has been assigned to the VEHICLE JOURNEY.

### 7.2.4.2 Journey Timing – Physical Model

The following figure shows the JOURNEY TIMING physical model and its attributes.



class XSD NeTEx TP Journey Timing Model
DataManagedObject
TimeDemandTypeModel::TimeDemandType
made up of
0..1
timing
0..*
VersionedChild
JourneyTimingModel::JourneyTiming
Name :MultilingualString [0..1]
VehicleMode :VehicleModeEnum [0..1]
«PK»
id :JourneyTimingIdType
«FK»
TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
TimebandRef :TimebandRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
0..*
distinguishes
0..1
DataManagedObject
TransportOrganisationModel::OperationalContext
distinguishes
0..1
0..*
distinguished by
0..1
distinguishes
DataManagedObject
PointAndLinkModel::Point
at
source
JourneyTimingModel::JourneyLayover
Layover :duration [0..1]
«PK»
id :LayoverIdType
«FK»
PointRef :PointRef*
JourneyTimingModel::JourneyWaitTime
WaitTime :duration
«PK»
id :WaitTimeIdType
«FK»
TimingPointRef :TimingPointRef*
«enumeration»
TransportModeValues::VehicleModeEnum
air
bus
coach
funicular
metro
rail
trolleyBus
tram
water
cableway
other
TimingPatternModel::TimingPoint
Category :string [0..1]
AllowedForWaitTime :duration [0..1]
Flexible :boolean [0..1]
TimingPointType :TimingStatusEnum [0..1]
«PK»
id :TimingPointIdType
at
time
0..*
start of
1
from
tc
*
JourneyTimingModel::TurnaroundTimeLimit
MinimumDuration :duration [0..1]
MaximumDuration :duration
«PK»
id :TurnaroundTimeIdType [0..1]
end of
1
*
distinguished by
1
start of
1
end of
passed
1
JourneyTimingModel::JourneyRunTime
RunTime :duration
«PK»
id :RunTimeIdType
«FK»
LinkRef :TimingLinkRef* [0..1]
0..*
from
*
to
*
VehicleAndCrewModel::ReliefPoint
«PK»
id :ReliefPointIdType
«FK»
CrewBaseRef :CrewBaseRef*
Link
TimingPatternModel::TimingLink
Mode :VehicileModeEnum [0..1]
«PK»
id :TimingLinkIdType
«FK»
FromPointRef :TimingPointRef*
ToPointRef :TimingPointRef*
OperationalContextRef :OperationalContextRef [0..1]
over
time
at every
0..*
JourneyTimingModel::JourneyHeadway
Frequency :HeadwayInterval
«PK»
id :HeadwayIdType
JourneyTimingModel::HeadwayInterval
ScheduledHeadwayInterval :duration [0..1]
MinimumHeadwayInterval :duration [0..1]
MaximumHeadwayInterval :duration [0..1]
VehicleAndCrewModel::ParkingPoint
«PK»
id :ParkingPointIdType
Name: XSD NeTEx TP Journey Timing Model
Author: NeTEx
Version: 1.0
Created: 26/01/2010 18:14:55
Updated: 01/10/2012 23:51:10

**Figure 44 — Journey Timing – Physical Model (UML)**

#### 7.2.4.3 Journey Timing – Attributes and XSD

##### 7.2.4.3.1 JourneyWaitTime – Model Element

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

**Table 23 – *JourneyWaitTime* – Element**

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

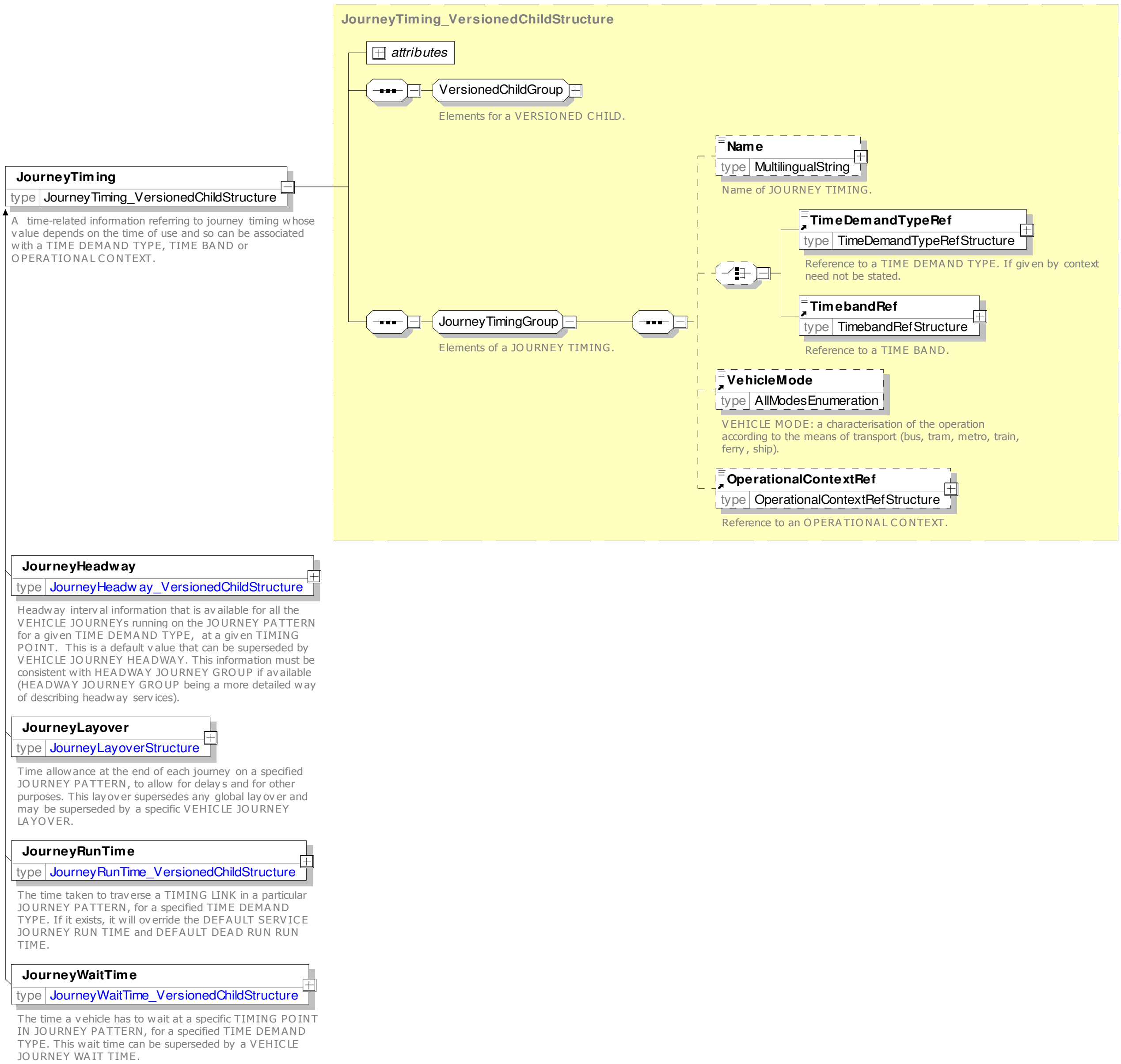

JourneyTiming_VersionedChildStructure
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTiming
type JourneyTiming_VersionedChildStructure
A time-related information referring to journey timing whose value depends on the time of use and so can be associated with a TIME DEMAND TYPE, TIME BAND or OPERATIONAL CONTEXT.
Name
type MultilingualString
Name of JOURNEY TIMING.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
TimebandRef
type TimebandRefStructure
Reference to a TIME BAND.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
VehicleMode
type AllModesEnumeration
VEHICLE MODE: a characterisation of the operation according to the means of transport (bus, tram, metro, train, ferry, ship).
OperationalContextRef
type OperationalContextRefStructure
Reference to an OPERATIONAL CONTEXT.
JourneyHeadway
type JourneyHeadway_VersionedChildStructure
Headway interval information that is available for all the VEHICLE JOURNEYs running on the JOURNEY PATTERN for a given TIME DEMAND TYPE, at a given TIMING POINT. This is a default value that can be superseded by VEHICLE JOURNEY HEADWAY. This information must be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services).
JourneyLayover
type JourneyLayoverStructure
Time allowance at the end of each journey on a specified JOURNEY PATTERN, to allow for delays and for other purposes. This layover supersedes any global layover and may be superseded by a specific VEHICLE JOURNEY LAYOVER.
JourneyRunTime
type JourneyRunTime_VersionedChildStructure
The time taken to traverse a TIMING LINK in a particular JOURNEY PATTERN, for a specified TIME DEMAND TYPE. If it exists, it will override the DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.
JourneyWaitTime
type JourneyWaitTime_VersionedChildStructure
The time a vehicle has to wait at a specific TIMING POINT IN JOURNEY PATTERN, for a specified TIME DEMAND TYPE. This wait time can be superseded by a VEHICLE JOURNEY WAIT TIME.

**Figure 45 — *JourneyWaitTime* — XSD**

### 7.2.4.3.2 JourneyRunTime – Model Element

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

**Table 24 – *JourneyRunTime* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyTiming* | ::> | JOURNEY RUN TIME inherits from JOURNEY TIMING. |
| «PK» | ***id*** | *RunTimeIdType* | 1:1 | Identifier of JOURNEY RUN TIME. |
| «FK» | ***TimingLinkRef*** | *TimingLinkRef* | 0:1 | TIMING LINK associated with timing. |

| | | | | |
|---|---|---|---|---|
| | ***RunTime*** | *xsd:duration* | 1:1 | Run time of JOURNEY RUN TIME. |

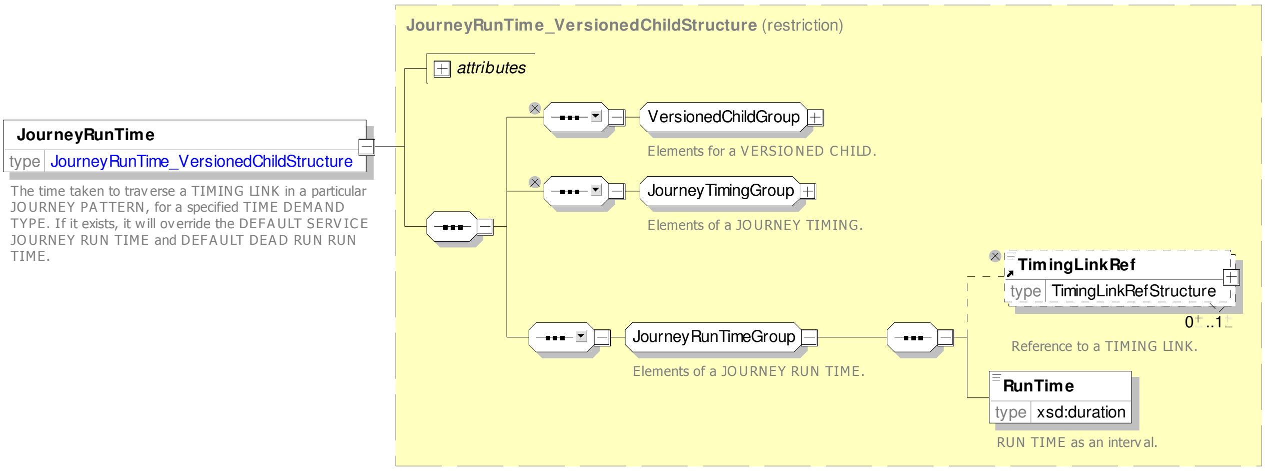

JourneyRunTime_VersionedChildStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyRunTime
type JourneyRunTime_VersionedChildStructure
The time taken to traverse a TIMING LINK in a particular JOURNEY PATTERN, for a specified TIME DEMAND TYPE. If it exists, it will override the DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.
TimingLinkRef
type TimingLinkRefStructure
0..1
Reference to a TIMING LINK.
JourneyRunTimeGroup
Elements of a JOURNEY RUN TIME.
RunTime
type xsd:duration
RUN TIME as an interval.

Figure 46 — *JourneyRunTime* — XSD

### 7.2.4.3.3 JourneyLayover – Model Element

Time allowance at the end of each journey on a specified JOURNEY PATTERN, to allow for delays and for other purposes. This layover supersedes any global layover and may be superseded by a specific VEHICLE JOURNEY LAYOVER.

Table 25 – *JourneyLayover* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyTiming* | ::> | JOURNEY LAYOVER inherits from JOURNEY TIMING. |
| «PK» | ***id*** | *LayoverIdType* | 1:1 | Identifier of JOURNEY LAYOVER. |
| | ***Layover*** | *xsd:duration* | 0:1 | Time spent on JOURNEY LAYOVER. |
| «FK» | ***PointRef*** | *PointRef* | 1:1 | POINT associated with of JOURNEY LAYOVER. |

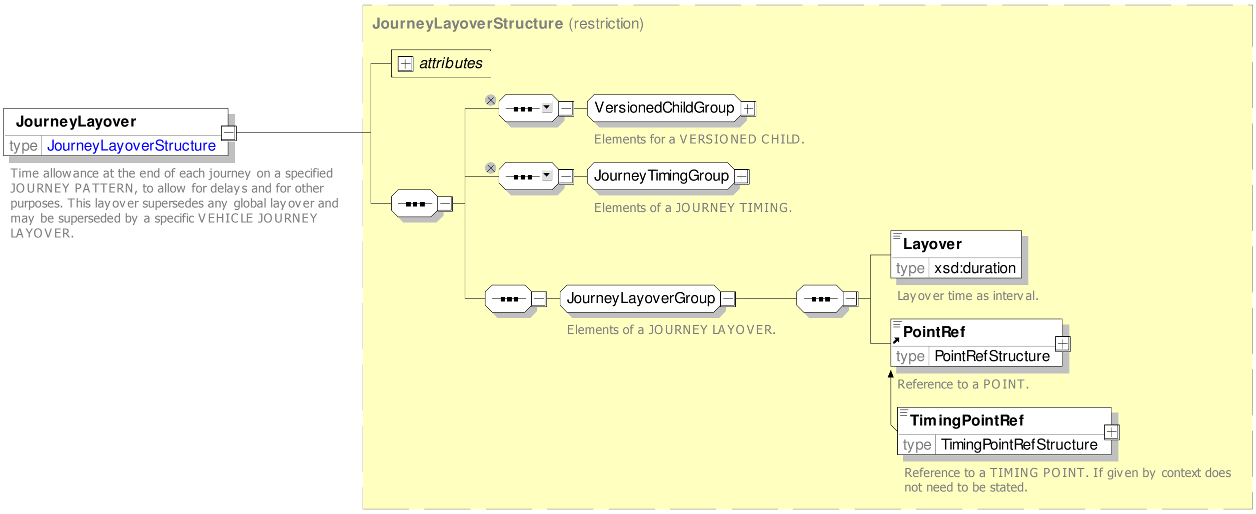

JourneyLayoverStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyLayover
type JourneyLayoverStructure
Time allowance at the end of each journey on a specified JOURNEY PATTERN, to allow for delays and for other purposes. This layover supersedes any global layover and may be superseded by a specific VEHICLE JOURNEY LAYOVER.
Layover
type xsd:duration
Layover time as interval.
JourneyLayoverGroup
Elements of a JOURNEY LAYOVER.
PointRef
type PointRefStructure
Reference to a POINT.
TimingPointRef
type TimingPointRefStructure
Reference to a TIMING POINT. If given by context does not need to be stated.

Figure 47 — *JourneyLayover* — XSD

### 7.2.4.3.4 JourneyHeadway – Model Element

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

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

Table 26 – *JourneyHeadway* – Element

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

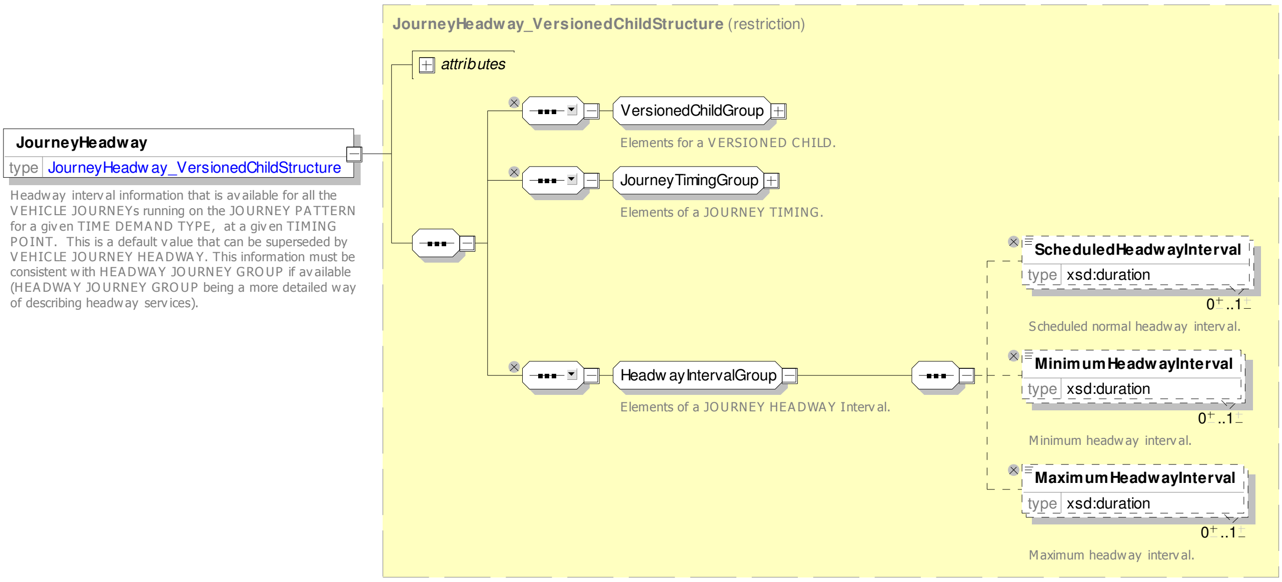

JourneyHeadway_VersionedChildStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyHeadway
type JourneyHeadway_VersionedChildStructure
Headway interval information that is available for all the VEHICLE JOURNEYs running on the JOURNEY PATTERN for a given TIME DEMAND TYPE, at a given TIMING POINT. This is a default value that can be superseded by VEHICLE JOURNEY HEADWAY. This information must be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services).
HeadwayIntervalGroup
Elements of a JOURNEY HEADWAY Interval.
ScheduledHeadwayInterval
type xsd:duration
0..1
Scheduled normal headway interval.
MinimumHeadwayInterval
type xsd:duration
0..1
Minimum headway interval.
MaximumHeadwayInterval
type xsd:duration
0..1
Maximum headway interval.

Figure 48 — *JourneyHeadway* — XSD

### 7.2.4.3.5 HeadwayInterval – Model Group

Set of Headway intervals to use to define a headway period.

Table 27 – *HeadwayInterval* – Group

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***ScheduledHeadwayInterval*** | *xsd:duration* | 0:1 | Scheduled HEADWAY INTERVAL. |
| | ***MinimumHeadwayInterval*** | *xsd:duration* | 0:1 | Minimum HEADWAY INTERVAL. |
| | ***MaximumHeadwayInterval*** | *xsd:duration* | 0:1 | Maximum HEADWAY INTERVAL. |

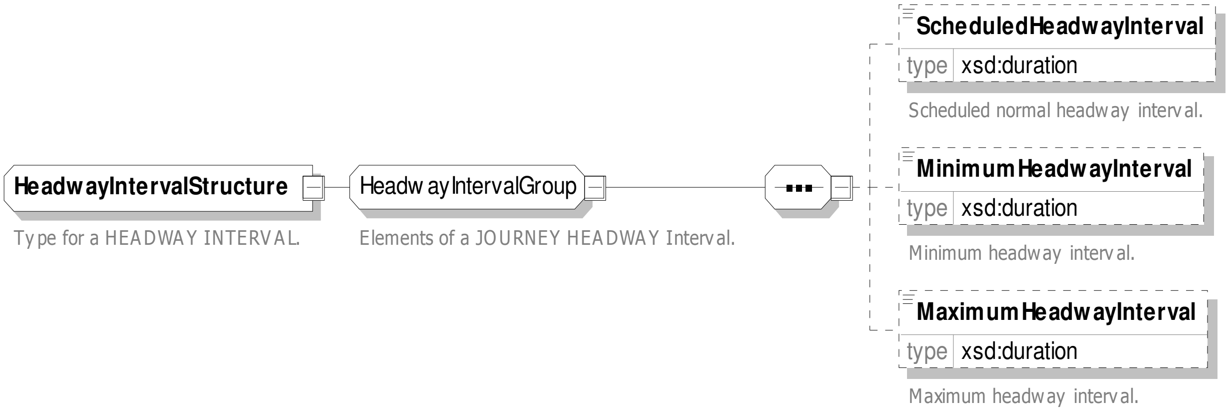


Figure 49 — *HeadwayInterval* — XSD

#### 7.2.4.3.6 TurnaroundTimeLimit – Model Element

The maximum time for which a vehicle may be scheduled to wait at a particular TIMING POINT (often included in a TURN STATION) without being returned to a PARKING POINT. A minimum time for a vehicle to turn its direction may also be recorded. This may be superseded by a DEAD RUN.

**Table 28 – *TurnaroundTimeLimitTime* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyTiming* | ::> | TURNAROUND TIME LIMIT inherits from JOURNEY TIMING. |
| «PK» | ***id*** | *TurnaroundTimeLimit-IdType* | 0:1 | Identifier of TURNAROUND TIME LIMIT. |
| | ***MinimumDuration*** | *xsd:duration* | 0:1 | Minimum time needed for turnaround. |
| | ***MaximumDuration*** | *xsd:duration* | 1:1 | Maximum time needed for turnaround. |

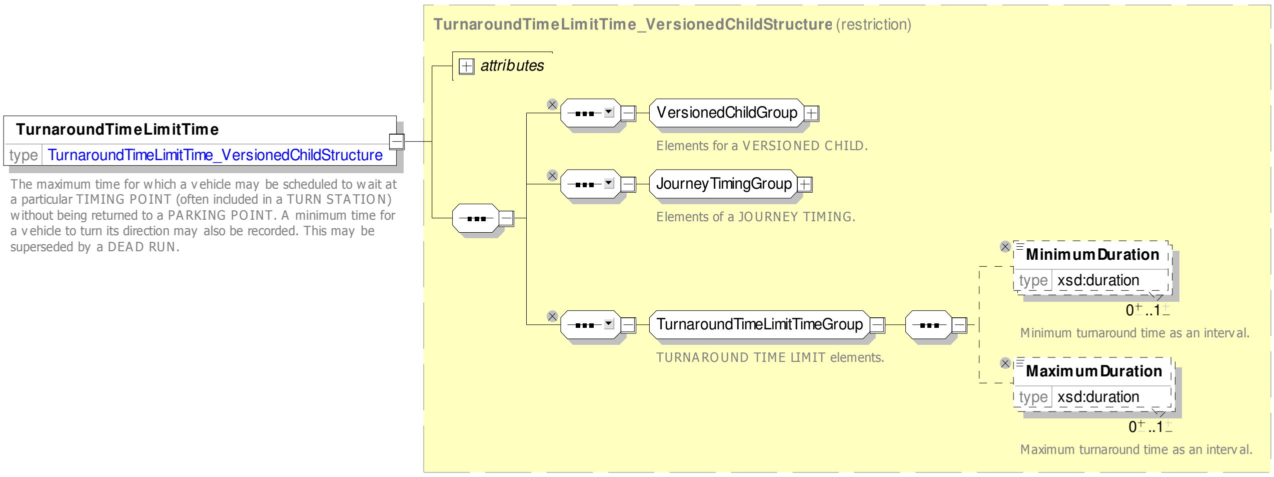

TurnaroundTimeLimitTime_VersionedChildStructure (restriction)
attributes
TurnaroundTimeLimitTime
type TurnaroundTimeLimitTime_VersionedChildStructure
The maximum time for which a vehicle may be scheduled to wait at a particular TIMING POINT (often included in a TURN STATION) without being returned to a PARKING POINT. A minimum time for a vehicle to turn its direction may also be recorded. This may be superseded by a DEAD RUN.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
TurnaroundTimeLimitTimeGroup
TURNAROUND TIME LIMIT elements.
MinimumDuration
type xsd:duration
0..1
Minimum turnaround time as an interval.
MaximumDuration
type xsd:duration
0..1
Maximum turnaround time as an interval.

**Figure 50 — *TurnaroundTimeLimitTime* — XSD**

### 7.2.5 Journey Pattern Times

#### 7.2.5.1 JOURNEY PATTERN TIMES – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

For any TIMING LINK, run times corresponding to one TIME DEMAND TYPE may be recorded. In some cases, such run times are default run times and if the TIMING LINK is used by several JOURNEY PATTERNs, the default times may be applied any time it is covered by a VEHICLE JOURNEY, regardless the JOURNEY PATTERN.

However, NeTEx allows for a more precise control of run times than is possible by using default times at TIMING LINK level, namely JOURNEY PATTERN RUN TIMEs. A JOURNEY PATTERN RUN TIME is a run time for a TIMING LINK that will be valid only for VEHICLE JOURNEYs made on the specified JOURNEY PATTERN. It overrides the default run times that might have been defined for this TIMING LINK.

JOURNEY PATTERN RUN TIMEs are sets of run times for different TIME DEMAND TYPEs. The TIME DEMAND TYPE for a particular VEHICLE JOURNEY is used to select the appropriate time, from the set recorded for a particular TIMING LINK belonging to the JOURNEY PATTERN covered.

A JOURNEY PATTERN WAIT TIME may be recorded to define the time a vehicle will have to wait at a specified TIMING POINT, e.g. to allow a large number of passengers to board or alight, or to wait for a connecting vehicle on another LINE. These wait times depend on the JOURNEY PATTERN that the VEHICLE JOURNEY covers, and on the TIME DEMAND TYPE.

A certain time allowance may be given at the end of each VEHICLE JOURNEY, before starting the next one, to compensate delays or for other purposes (e.g. rest time for the driver). This “layover time” can be regarded as a buffer time on a specified JOURNEY PATTERN for the different TIME DEMAND TYPEs. This layover may be superseded by a specific VEHICLE JOURNEY LAYOVER (cf. 5.1.5).

In the case of frequency-based services, another type of information is needed, namely headway duration information. It is given by JOURNEY PATTERN HEADWAY that is available for all the VEHICLE JOURNEYs running on the JOURNEY PATTERN at the TIMING POINTs IN JOURNEY PATTERN depending upon the different TIME DEMAND TYPEs. A more detailed modelling of frequency-based services is described in 5.1.5.

The VEHICLE TYPE PREFERENCE represents a default VEHICLE TYPE proposed for the journeys, depending on the JOURNEY PATTERN covered and the TIME DEMAND TYPE. It is not a truly temporal concept but has to be understood as a proposal to be taken into account for BLOCK compilation. It may be a ranked list of VEHICLE TYPEs for each SERVICE JOURNEY PATTERN and for each DAY TYPE and TIME DEMAND TYPE.

The diagram reminds that all this timing information is to be considered in a specific OPERATIONAL CONTEXT, that expresses the characterization of a set of operational objects, such as timing or links determined either by a DEPARTMENT or by an ORGANISATIONAL UNIT.

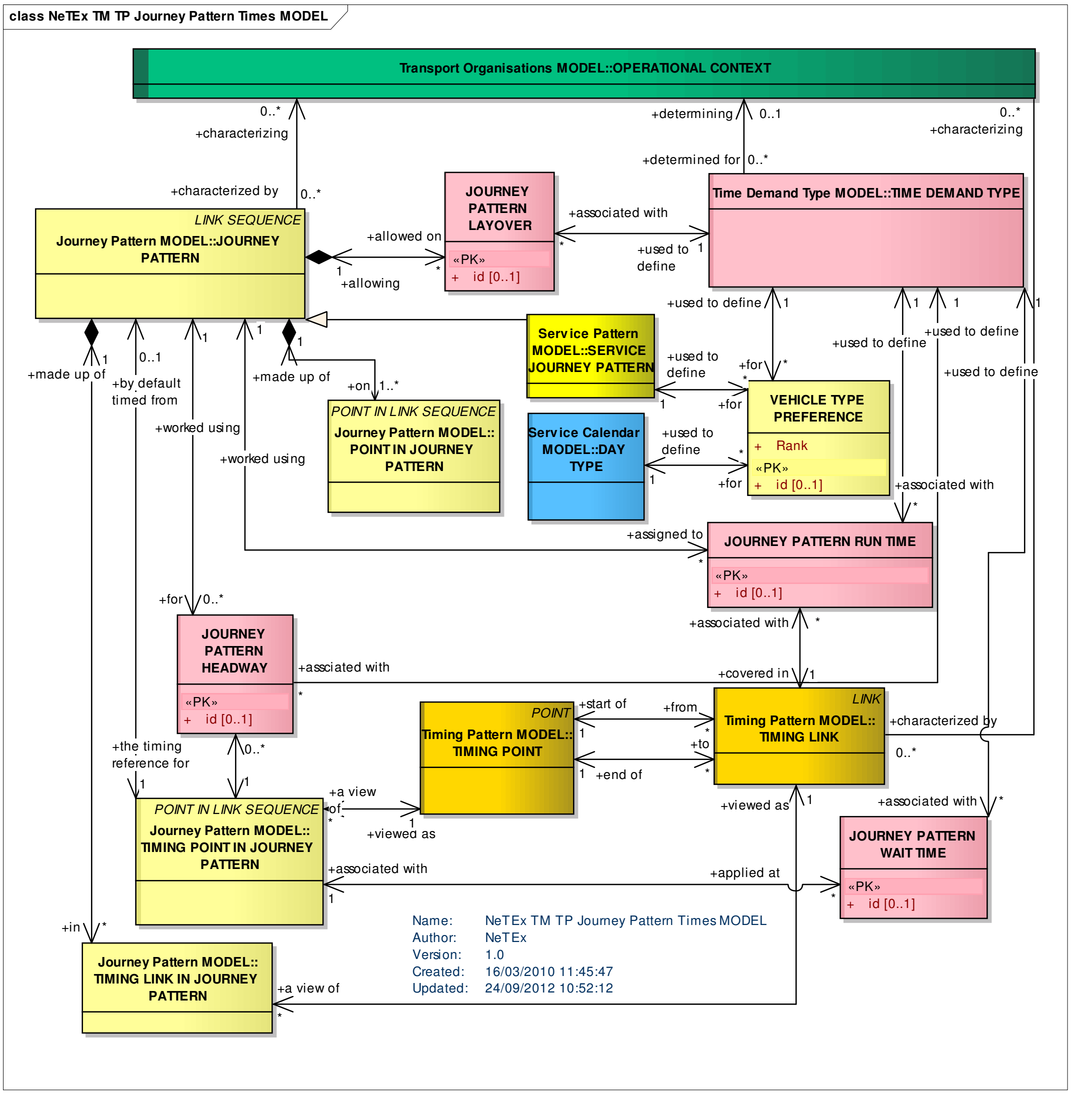
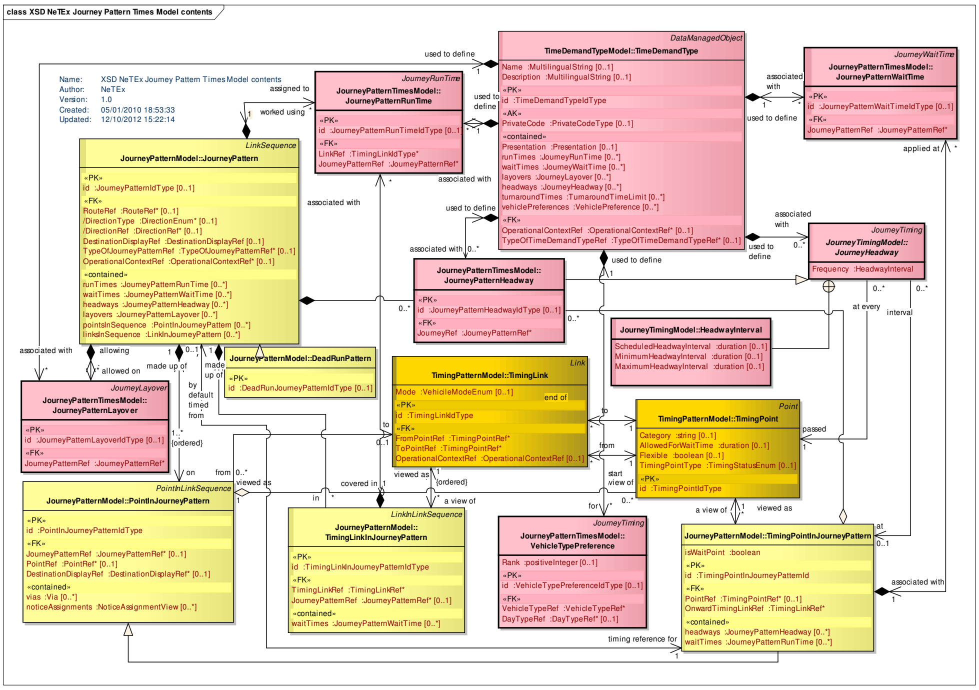


**Figure 51 — Journey Pattern Times – Conceptual MODEL (UML)**

### 7.2.5.2 Journey Pattern Times – Physical Model

The physical model is derived from the conceptual model and specialises elements of the JOURNEY TIMING model.

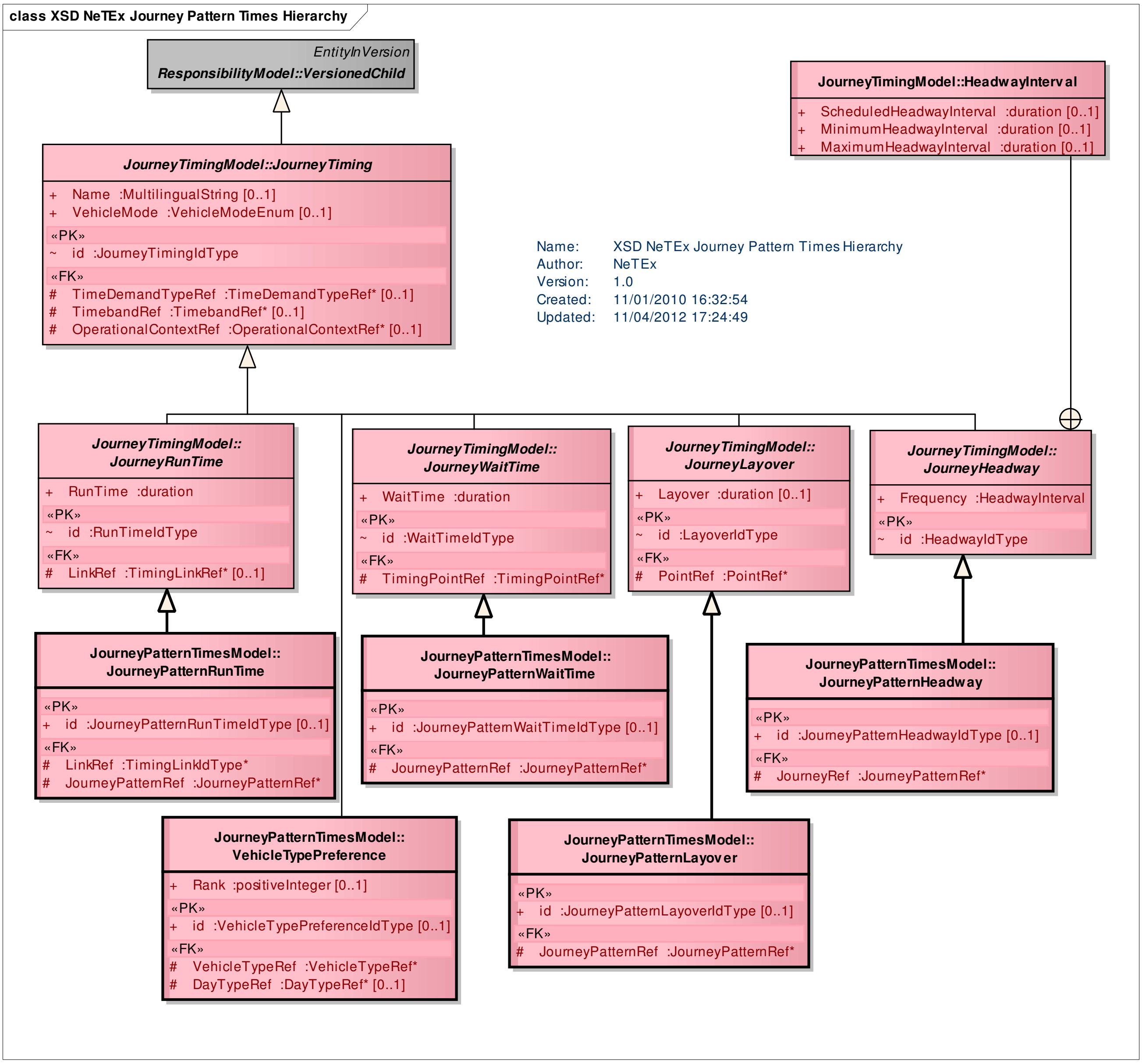
In this case, the value 'duration' represents the duration of the different times for the whole JOURNEY PATTERN. In other words, the different 'durations' represent time elements (run times, wait times, etc.) for all the VEHICLE JOURNEYs of a certain JOURNEY PATTERN, depending on the time of use, i.e. associated with a TIME DEMAND TYPE or, for users working with TIME BANDs, associated to a particular TIME BAND.


class XSD NeTEx Journey Pattern Times Model contents
Name: XSD NeTEx Journey Pattern Times Model contents
Author: NeTEx
Version: 1.0
Created: 05/01/2010 18:53:33
Updated: 12/10/2012 15:22:14
DataManagedObject
TimeDemandTypeModel::TimeDemandType
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
«PK»
id :TimeDemandTypeIdType
«AK»
PrivateCode :PrivateCodeType [0..1]
«contained»
Presentation :Presentation [0..1]
runTimes :JourneyRunTime [0..*]
waitTimes :JourneyWaitTime [0..*]
layovers :JourneyLayover [0..*]
headways :JourneyHeadway [0..*]
turnaroundTimes :TurnaroundTimeLimit [0..*]
vehiclePreferences :VehiclePreference [0..*]
«FK»
OperationalContextRef :OperationalContextRef* [0..1]
TypeOfTimeDemandTypeRef :TypeOfTimeDemandTypeRef* [0..1]
JourneyRunTime
JourneyPatternTimesModel::JourneyPatternRunTime
«PK»
id :JourneyPatternRunTimeIdType [0..1]
«FK»
LinkRef :TimingLinkIdType*
JourneyPatternRef :JourneyPatternRef*
JourneyWaitTime
JourneyPatternTimesModel::JourneyPatternWaitTime
«PK»
id :JourneyPatternWaitTimeIdType [0..1]
«FK»
JourneyPatternRef :JourneyPatternRef*
LinkSequence
JourneyPatternModel::JourneyPattern
«PK»
id :JourneyPatternIdType [0..1]
«FK»
RouteRef :RouteRef* [0..1]
/DirectionType :DirectionEnum* [0..1]
/DirectionRef :DirectionRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef [0..1]
TypeOfJourneyPatternRef :TypeOfJourneyPatternRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
«contained»
runTimes :JourneyPatternRunTime [0..*]
waitTimes :JourneyPatternWaitTime [0..*]
headways :JourneyPatternHeadway [0..*]
layovers :JourneyPatternLayover [0..*]
pointsInSequence :PointInJourneyPattern [0..*]
linksInSequence :LinkInJourneyPattern [0..*]
JourneyPatternTimesModel::JourneyPatternHeadway
«PK»
id :JourneyPatternHeadwayIdType [0..1]
«FK»
JourneyRef :JourneyPatternRef*
JourneyTiming
JourneyTimingModel::JourneyHeadway
Frequency :HeadwayInterval
JourneyTimingModel::HeadwayInterval
ScheduledHeadwayInterval :duration [0..1]
MinimumHeadwayInterval :duration [0..1]
MaximumHeadwayInterval :duration [0..1]
JourneyPatternModel::DeadRunPattern
«PK»
id :DeadRunJourneyPatternIdType [0..1]
JourneyLayover
JourneyPatternTimesModel::JourneyPatternLayover
«PK»
id :JourneyPatternLayoverIdType [0..1]
«FK»
JourneyPatternRef :JourneyPatternRef*
Link
TimingPatternModel::TimingLink
Mode :VehicileModeEnum [0..1]
«PK»
id :TimingLinkIdType
«FK»
FromPointRef :TimingPointRef*
ToPointRef :TimingPointRef*
OperationalContextRef :OperationalContextRef [0..1]
Point
TimingPatternModel::TimingPoint
Category :string [0..1]
AllowedForWaitTime :duration [0..1]
Flexible :boolean [0..1]
TimingPointType :TimingStatusEnum [0..1]
«PK»
id :TimingPointIdType
PointInLinkSequence
JourneyPatternModel::PointInJourneyPattern
«PK»
id :PointInJourneyPatternIdType
«FK»
JourneyPatternRef :JourneyPatternRef* [0..1]
PointRef :PointRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef* [0..1]
«contained»
vias :Via [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
LinkInLinkSequence
JourneyPatternModel::TimingLinkInJourneyPattern
«PK»
id :TimingLinkInJourneyPatternIdType
«FK»
TimingLinkRef :TimingLinkRef*
JourneyPatternRef :JourneyPatternRef* [0..1]
«contained»
waitTimes :JourneyPatternWaitTime [0..*]
JourneyTiming
JourneyPatternTimesModel::VehicleTypePreference
Rank :positiveInteger [0..1]
«PK»
id :VehicleTypePreferenceIdType [0..1]
«FK»
VehicleTypeRef :VehicleTypeRef*
DayTypeRef :DayTypeRef* [0..1]
JourneyPatternModel::TimingPointInJourneyPattern
isWaitPoint :boolean
«PK»
id :TimingPointInJourneyPatternId
«FK»
PointRef :TimingPointRef* [0..1]
OnwardTimingLinkRef :TimingLinkRef*
«contained»
headways :JourneyPatternHeadway [0..*]
waitTimes :JourneyPatternRunTime [0..*]
used to define
assigned to
worked using
used to define
associated with
associated with
used to define
associated with
associated with
used to define
applied at
associated with
used to define
used to define
at every
interval
associated with
allowing
allowed on
made up of
made up of
by default timed from
{ordered}
on
from
viewed as
covered in
in
to
end of
viewed as
{ordered}
a view of
to
from
start
view of
for
passed
a view of
viewed as
at
associated with
timing reference for
1
*
0..*
0..1
1..*

## Figure 52 — Journey Pattern Times – Physical Model (UML)

### 7.2.5.2.1 Journey Pattern Time hierarchies – Physical Model

The following figures shows the inherited attributes of the various journey timings.



class XSD NeTEx Journey Pattern Times Hierarchy
EntityInVersion
ResponsibilityModel::VersionedChild
JourneyTimingModel::JourneyTiming
+ Name :MultilingualString [0..1]
+ VehicleMode :VehicleModeEnum [0..1]
«PK»
~ id :JourneyTimingIdType
«FK»
# TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
# TimebandRef :TimebandRef* [0..1]
# OperationalContextRef :OperationalContextRef* [0..1]
JourneyTimingModel::HeadwayInterval
+ ScheduledHeadwayInterval :duration [0..1]
+ MinimumHeadwayInterval :duration [0..1]
+ MaximumHeadwayInterval :duration [0..1]
Name: XSD NeTEx Journey Pattern Times Hierarchy
Author: NeTEx
Version: 1.0
Created: 11/01/2010 16:32:54
Updated: 11/04/2012 17:24:49
JourneyTimingModel::JourneyRunTime
+ RunTime :duration
«PK»
~ id :RunTimeIdType
«FK»
# LinkRef :TimingLinkRef* [0..1]
JourneyTimingModel::JourneyWaitTime
+ WaitTime :duration
«PK»
~ id :WaitTimeIdType
«FK»
# TimingPointRef :TimingPointRef*
JourneyTimingModel::JourneyLayover
+ Layover :duration [0..1]
«PK»
~ id :LayoverIdType
«FK»
# PointRef :PointRef*
JourneyTimingModel::JourneyHeadway
+ Frequency :HeadwayInterval
«PK»
~ id :HeadwayIdType
JourneyPatternTimesModel::JourneyPatternRunTime
«PK»
+ id :JourneyPatternRunTimeIdType [0..1]
«FK»
# LinkRef :TimingLinkIdType*
# JourneyPatternRef :JourneyPatternRef*
JourneyPatternTimesModel::JourneyPatternWaitTime
«PK»
+ id :JourneyPatternWaitTimeIdType [0..1]
«FK»
# JourneyPatternRef :JourneyPatternRef*
JourneyPatternTimesModel::JourneyPatternHeadway
«PK»
+ id :JourneyPatternHeadwayIdType [0..1]
«FK»
# JourneyRef :JourneyPatternRef*
JourneyPatternTimesModel::VehicleTypePreference
+ Rank :positiveInteger [0..1]
«PK»
+ id :VehicleTypePreferenceIdType [0..1]
«FK»
# VehicleTypeRef :VehicleTypeRef*
# DayTypeRef :DayTypeRef* [0..1]
JourneyPatternTimesModel::JourneyPatternLayover
«PK»
+ id :JourneyPatternLayoverIdType [0..1]
«FK»
# JourneyPatternRef :JourneyPatternRef*

**Figure 53 — Journey Pattern Times – Physical Model (UML)**

### 7.2.5.3 Journey Pattern Times – Attributes and XSD

#### 7.2.5.3.1 JourneyPatternWaitTime – Model Element

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

**Table 29 – *JourneyPatternWaitTime* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyWaitTime* | ::> | JOURNEY PATTERN WAIT TIME inherits from JOURNEY WAIT TIME. |
| «PK» | ***id*** | *JourneyPatternWaitTimeIdType* | 0:1 | Identifier of JOURNEY PATTERN WAIT TIME. |

| | | | | |
|---|---|---|---|---|
| «FK» | ***JourneyRef*** | *JourneyPatternRef* | 1:1 | JOURNEY PATTERN with which WAIT TIME is associated. |

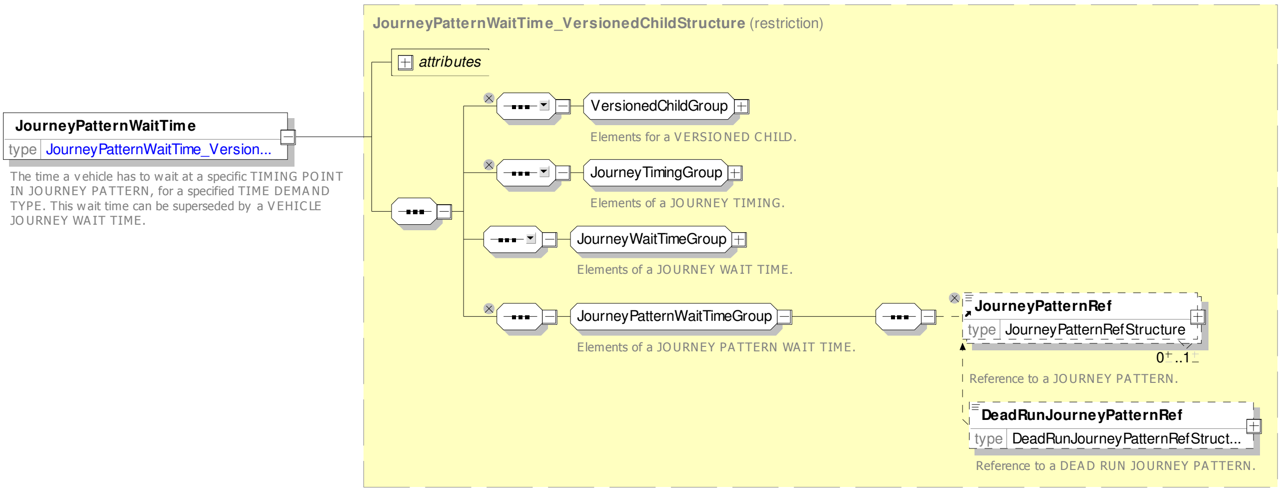

JourneyPatternWaitTime_VersionedChildStructure (restriction)
attributes
JourneyPatternWaitTime
type JourneyPatternWaitTime_Version...
The time a vehicle has to wait at a specific TIMING POINT IN JOURNEY PATTERN, for a specified TIME DEMAND TYPE. This wait time can be superseded by a VEHICLE JOURNEY WAIT TIME.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyWaitTimeGroup
Elements of a JOURNEY WAIT TIME.
JourneyPatternWaitTimeGroup
Elements of a JOURNEY PATTERN WAIT TIME.
JourneyPatternRef
type JourneyPatternRefStructure
0..1
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.

Figure 54 — *JourneyPatternWaitTime* — XSD

### 7.2.5.3.2 JourneyPatternRunTime – Model Element

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

Table 30 – *JourneyPatternRunTime* – Element

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyRunTime* | ::> | JOURNEY PATTERN RUN TIME inherits from JOURNEY RUN TIME. |
| «PK» | ***id*** | *JourneyPatternRunTimeIdType* | 0:1 | Identifier of JOURNEY PATTERN RUN TIME. |
| «FK» | ***LinkRef*** | *TimingLinkRef* | 1:1 | TIMING LINK with which TURNAROUND TIME is associated. |
| «FK» | ***JourneyRef*** | *JourneyPatternRef* | 1:1 | JOURNEY PATTERN with which TURNAROUND TIME is associated. |

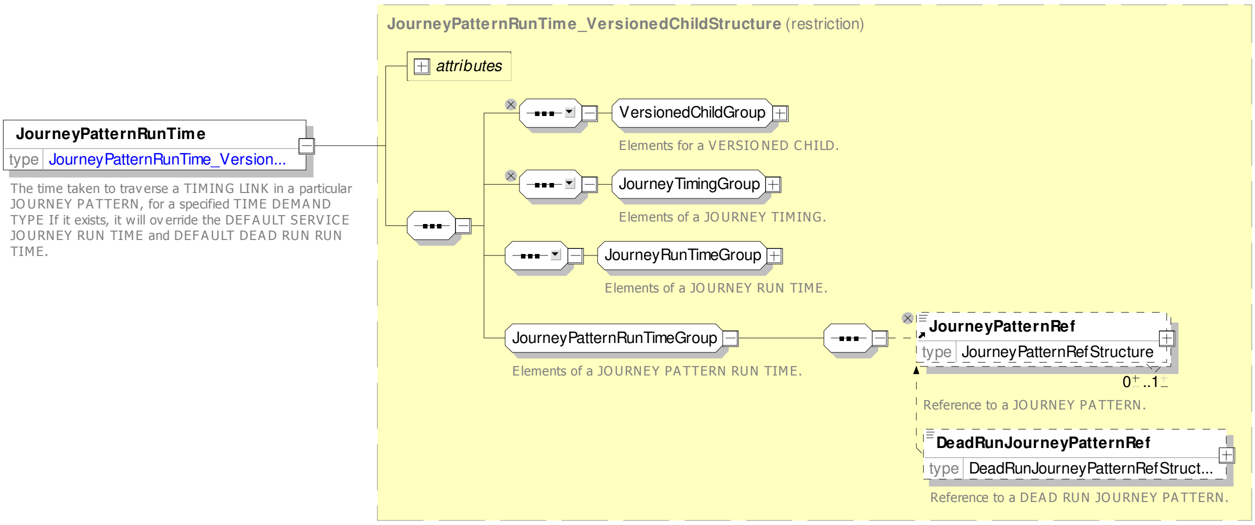

JourneyPatternRunTime_VersionedChildStructure (restriction)
attributes
JourneyPatternRunTime
type JourneyPatternRunTime_Version...
The time taken to traverse a TIMING LINK in a particular JOURNEY PATTERN, for a specified TIME DEMAND TYPE If it exists, it will override the DEFAULT SERVICE JOURNEY RUN TIME and DEFAULT DEAD RUN RUN TIME.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyRunTimeGroup
Elements of a JOURNEY RUN TIME.
JourneyPatternRunTimeGroup
Elements of a JOURNEY PATTERN RUN TIME.
JourneyPatternRef
type JourneyPatternRefStructure
0..1
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.

Figure 55 — *JourneyPatternRunTime* — XSD

### 7.2.5.3.3 JourneyPatternLayover – Model Element

The JOURNEY PATTERN LAYOVER is the time allowance at the end of each journey on a specified JOURNEY PATTERN, to allow for delays and for other purposes. This layover supersedes any global layover and may be superseded by a specific VEHICLE JOURNEY LAYOVER.

Table 31 – *JourneyPatternLayover* – Element

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyLayoverTime* | ::> | JOURNEY PATTERN LAYOVER inherits from JOURNEY LAYOVER. |
| «PK» | ***id*** | *JourneyPatternLayoverIdType* | 0:1 | Identifier of JOURNEY PATTERN LAYOVER. |
| «FK» | ***JourneyPatternRef*** | *JourneyPatternRef* | 1:1 | JOURNEY PATTERN with which LAYOVER is associated. |

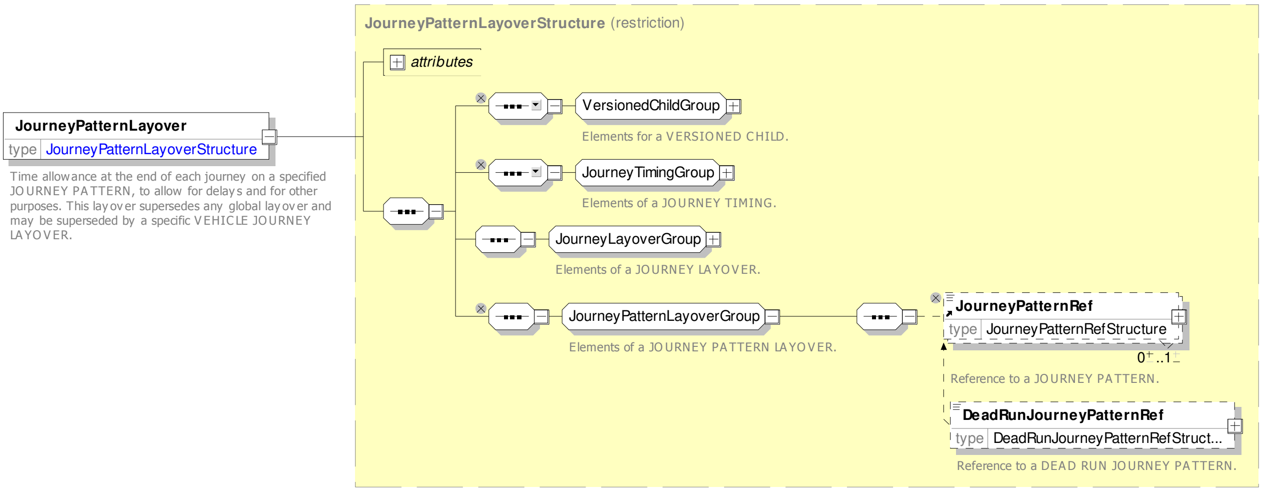

JourneyPatternLayoverStructure (restriction)
attributes
JourneyPatternLayover
type JourneyPatternLayoverStructure
Time allowance at the end of each journey on a specified JOURNEY PATTERN, to allow for delays and for other purposes. This layover supersedes any global layover and may be superseded by a specific VEHICLE JOURNEY LAYOVER.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyLayoverGroup
Elements of a JOURNEY LAYOVER.
JourneyPatternLayoverGroup
Elements of a JOURNEY PATTERN LAYOVER.
JourneyPatternRef
type JourneyPatternRefStructure
0..1
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.

Figure 56 — *JourneyPatternLayover* — XSD

### 7.2.5.3.4 JourneyPatternHeadway – Model Element

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

This information shall be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services, cf. 5.1.5).

**Table 32 – *JourneyPatternHeadway* – Element**

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

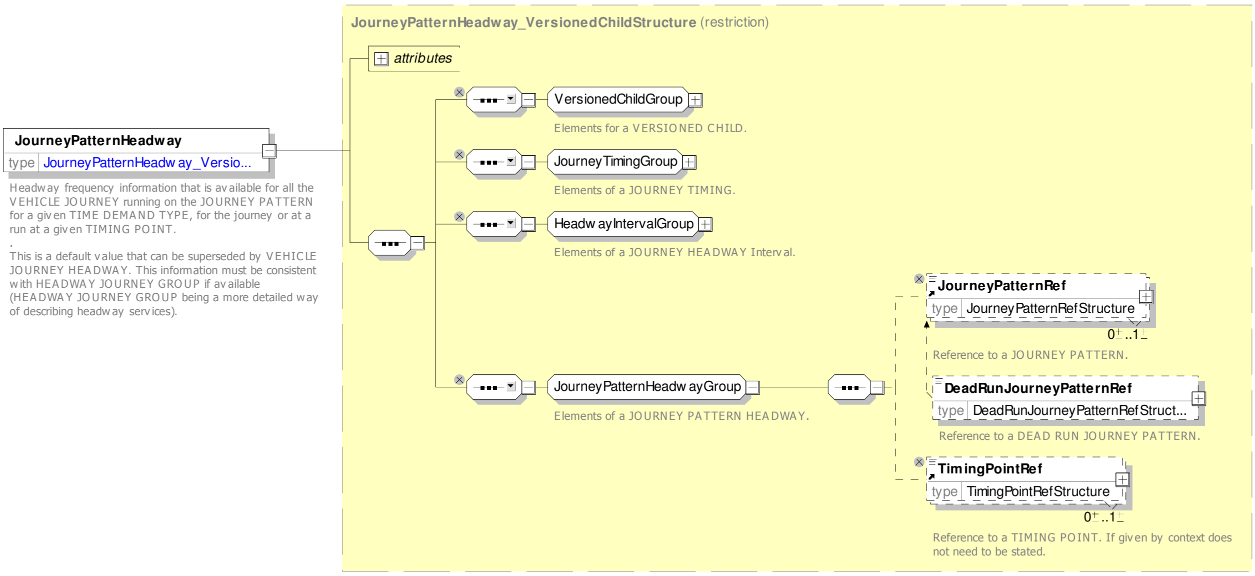

JourneyPatternHeadway_VersionedChildStructure (restriction)
attributes
JourneyPatternHeadway
type JourneyPatternHeadway_Versio...
Headway frequency information that is available for all the VEHICLE JOURNEY running on the JOURNEY PATTERN for a given TIME DEMAND TYPE, for the journey or at a run at a given TIMING POINT.
.
This is a default value that can be superseded by VEHICLE JOURNEY HEADWAY. This information must be consistent with HEADWAY JOURNEY GROUP if available (HEADWAY JOURNEY GROUP being a more detailed way of describing headway services).
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
HeadwayIntervalGroup
Elements of a JOURNEY HEADWAY Interval.
JourneyPatternHeadwayGroup
Elements of a JOURNEY PATTERN HEADWAY.
JourneyPatternRef
type JourneyPatternRefStructure
0..1
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.
TimingPointRef
type TimingPointRefStructure
0..1
Reference to a TIMING POINT. If given by context does not need to be stated.

**Figure 57 — *JourneyPatternHeadway* — XSD**

### 7.2.5.3.5 VehicleTypePreference – Model Element

The preference for the use of a particular VEHICLE TYPE for a SERVICE JOURNEY PATTERN, depending on the DAY TYPE and TIME DEMAND TYPE. The rank of preferences shall be recorded. Different VEHICLE TYPEs may be given the same rank.

**Table 33 – *VehicleTypePreference* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *TimeDemandTiming* | ::> | JOURNEY PATTERN HEADWAY inherits from TIME DEMAND TIMING. |

| | | | | |
|---|---|---|---|---|
| «PK» | ***id*** | *VehicleType-PreferenceIdType* | 0:1 | Identifier of VEHICLE TYPE PREFERENCE. |
| | ***Rank*** | *xsd:positiveInteger* | 1:1 | Ranking of VEHICLE TYPE PREFERENCE. |
| «FK» | ***DayTypeRef*** | *DayTypeRef* | 0:1 | DAY TYPE with which PREFERENCE is associated. |
| «FK» | ***VehicleTypeRef*** | *VehicleTypeRef* | 1:1 | VEHICLE TYPE with which PREFERENCE is associated. |

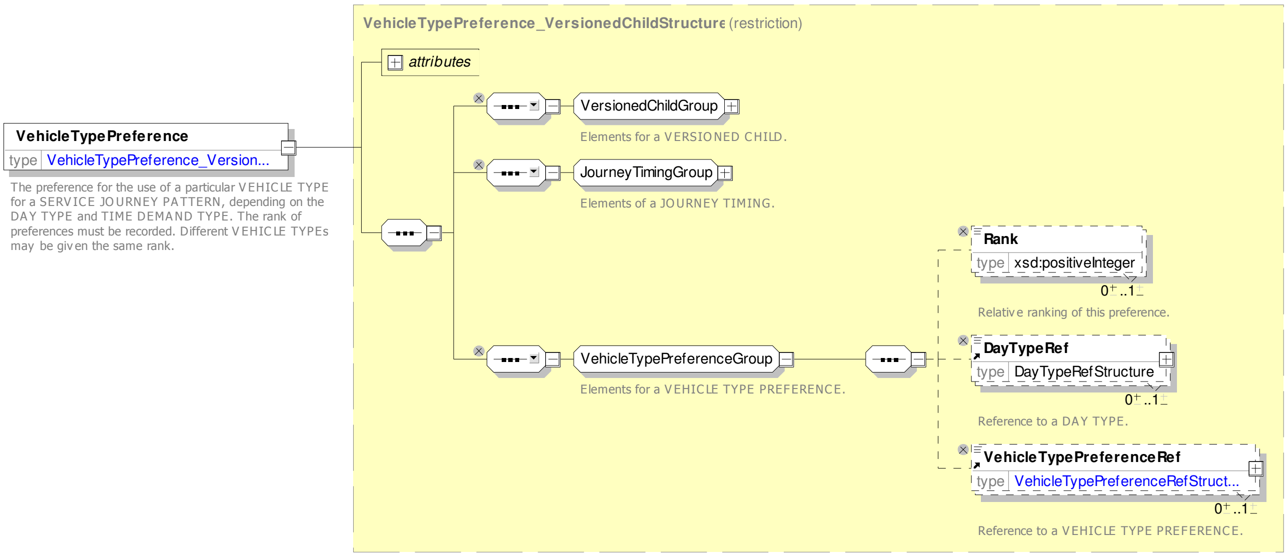

VehicleTypePreference_VersionedChildStructure (restriction)
attributes
VehicleTypePreference
type VehicleTypePreference_Version...
The preference for the use of a particular VEHICLE TYPE for a SERVICE JOURNEY PATTERN, depending on the DAY TYPE and TIME DEMAND TYPE. The rank of preferences must be recorded. Different VEHICLE TYPEs may be given the same rank.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
VehicleTypePreferenceGroup
Elements for a VEHICLE TYPE PREFERENCE.
Rank
type xsd:positiveInteger
0..1
Relative ranking of this preference.
DayTypeRef
type DayTypeRefStructure
0..1
Reference to a DAY TYPE.
VehicleTypePreferenceRef
type VehicleTypePreferenceRefStruct...
0..1
Reference to a VEHICLE TYPE PREFERENCE.

**Figure 58 — *VehicleTypePreference* — XSD**

### 7.2.6 Vehicle Journey Times

#### 7.2.6.1 VEHICLE JOURNEY TIMES – Conceptual MODEL

There may be reasons for needing a more precise control of run and wait times than is possible using the standard times at JOURNEY PATTERN level. Some exception times may be required for a single VEHICLE JOURNEY. For instance, a vehicle may be slowed intentionally on a particular VEHICLE JOURNEY to meet a scheduled interchange. Some companies adjust the times (e.g. wait times, but even run times) to cope with driver scheduling regulations. In the case of a very long VEHICLE JOURNEY, e.g. in a rural area, a single TIME DEMAND TYPE may not apply to the whole journey, because traffic conditions change substantially between its start and end.

It is hence necessary to be able to override the standard times by times for a specific VEHICLE JOURNEY.

A number of specialisations of JOURNEY TIMING are used to provide VEHICLE JOURNEY:

— VEHICLE JOURNEY RUN TIME is the time taken to traverse a specified TIMING LINK IN JOURNEY PATTERN on a specified VEHICLE JOURNEY. This gives the most detailed control over times and overrides the DEFAULT SERVICE JOURNEY RUN TIME and JOURNEY PATTERN RUN TIME and the DEFAULT DEAD RUN RUN TIME.

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

— VEHICLE JOURNEY LAYOVER is time allowance at the end of a specified VEHICLE JOURNEY. This time supersedes any global layover or JOURNEY PATTERN LAYOVER.

— VEHICLE JOURNEY HEADWAY is the headway frequency information that is available for a VEHICLE JOURNEY (to be understood as the delay between the previous and the next VEHICLE JOURNEY).

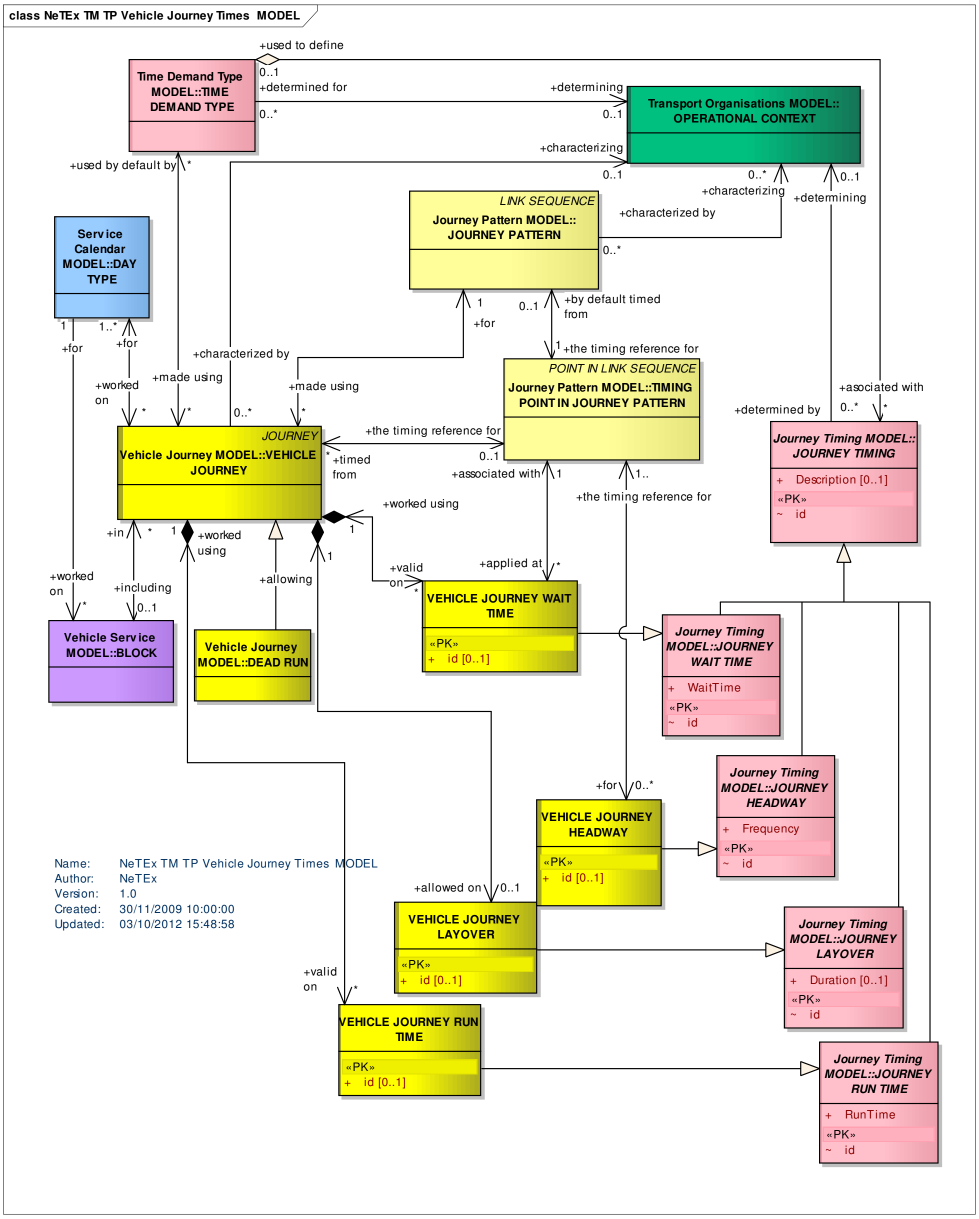

class NeTEx TM TP Vehicle Journey Times MODEL
+used to define
0..1
+determined for
0..*
+determining
0..1
Time Demand Type MODEL::TIME DEMAND TYPE
Transport Organisations MODEL:: OPERATIONAL CONTEXT
+characterizing
0..1
+used by default by
*
0..*
+characterizing
0..1
+determining
+characterized by
LINK SEQUENCE
Journey Pattern MODEL:: JOURNEY PATTERN
0..*
Service Calendar MODEL::DAY TYPE
1
1..*
+for
+for
1
+for
0..1
+by default timed from
1
+the timing reference for
+characterized by
+worked on
+made using
+made using
POINT IN LINK SEQUENCE
Journey Pattern MODEL::TIMING POINT IN JOURNEY PATTERN
+determined by
+asociated with
0..*
*
*
*
0..*
*
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
+the timing reference for
*
+timed from
0..1
+associated with
1
1..
+the timing reference for
Journey Timing MODEL:: JOURNEY TIMING
+ Description [0..1]
«PK»
~ id
+worked using
+in
*
1
+worked using
1
1
+allowing
+valid on
*
+applied at
*
+worked on
*
+including
0..1
Vehicle Service MODEL::BLOCK
Vehicle Journey MODEL::DEAD RUN
VEHICLE JOURNEY WAIT TIME
«PK»
+ id [0..1]
Journey Timing MODEL::JOURNEY WAIT TIME
+ WaitTime
«PK»
~ id
+for
0..*
VEHICLE JOURNEY HEADWAY
«PK»
+ id [0..1]
Journey Timing MODEL::JOURNEY HEADWAY
+ Frequency
«PK»
~ id
Name: NeTEx TM TP Vehicle Journey Times MODEL
Author: NeTEx
Version: 1.0
Created: 30/11/2009 10:00:00
Updated: 03/10/2012 15:48:58
+allowed on
0..1
VEHICLE JOURNEY LAYOVER
«PK»
+ id [0..1]
Journey Timing MODEL::JOURNEY LAYOVER
+ Duration [0..1]
«PK»
~ id
+valid on
*
VEHICLE JOURNEY RUN TIME
«PK»
+ id [0..1]
Journey Timing MODEL::JOURNEY RUN TIME
+ RunTime
«PK»
~ id

**Figure 59 — Vehicle Journey Times — Conceptual MODEL (UML)**

A more detailed model of frequency-based services is represented in the diagram below. Two different types of services are commonly found, representing journeys with a particular behaviour as to the regularity of their timing.

One of them is the common frequency-based service, based on the concept of HEADWAY INTERVAL, i.e. a regular interval duration between the journeys ‘every 10 minutes’ for example. The other possibility is a set of particular ‘rhythm’ services occurring over every hour, for example, services running at ‘xxh10’, ‘xxh25’, ‘xxh45’ past the hour. In either case a TEMPLATE VEHICLE JOURNEY is associated with a JOURNEY FREQUENCY GROUP to represents a group of JOURNEYs running at the given frequency.

Such a behaviour can be conditional upon the TIME DEMAND TYPE and TIME BAND.

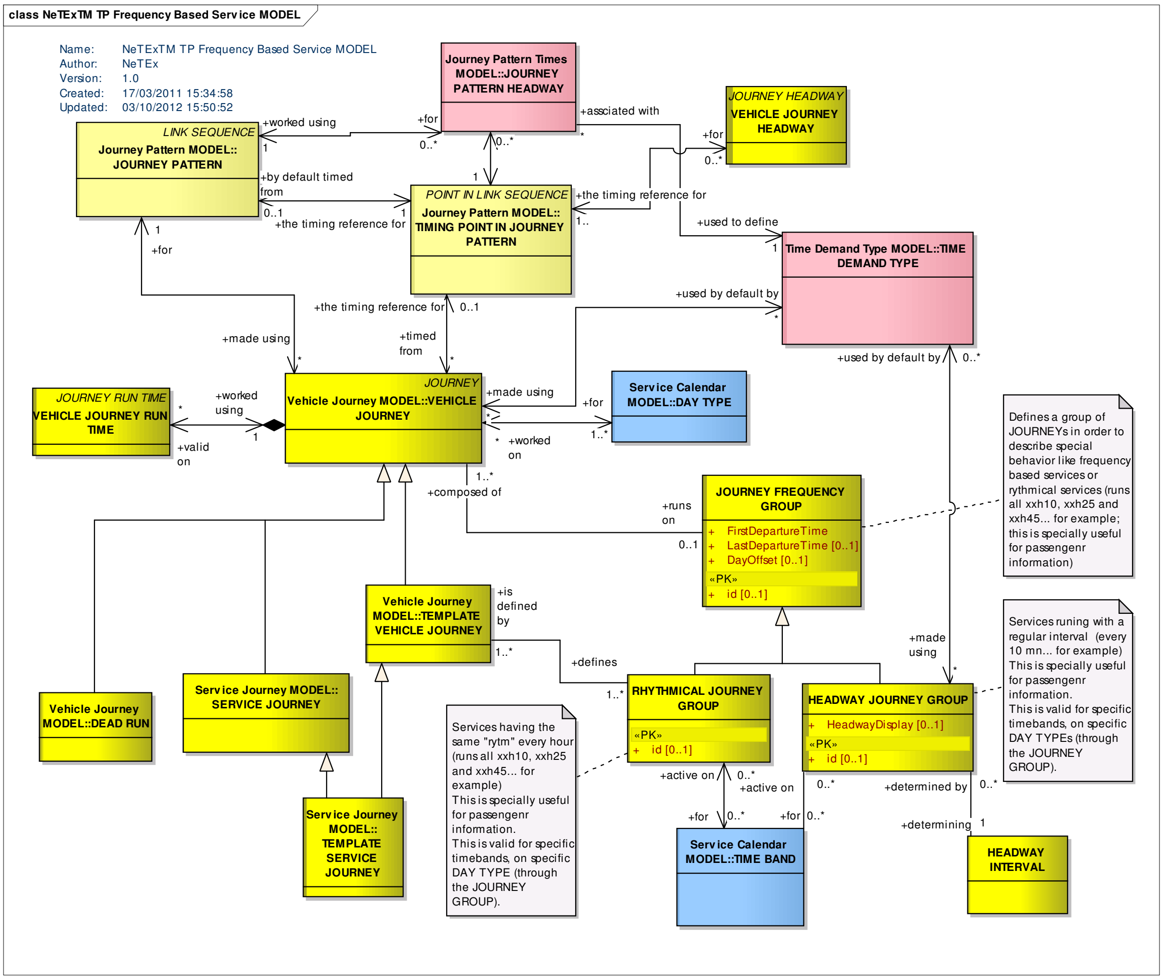
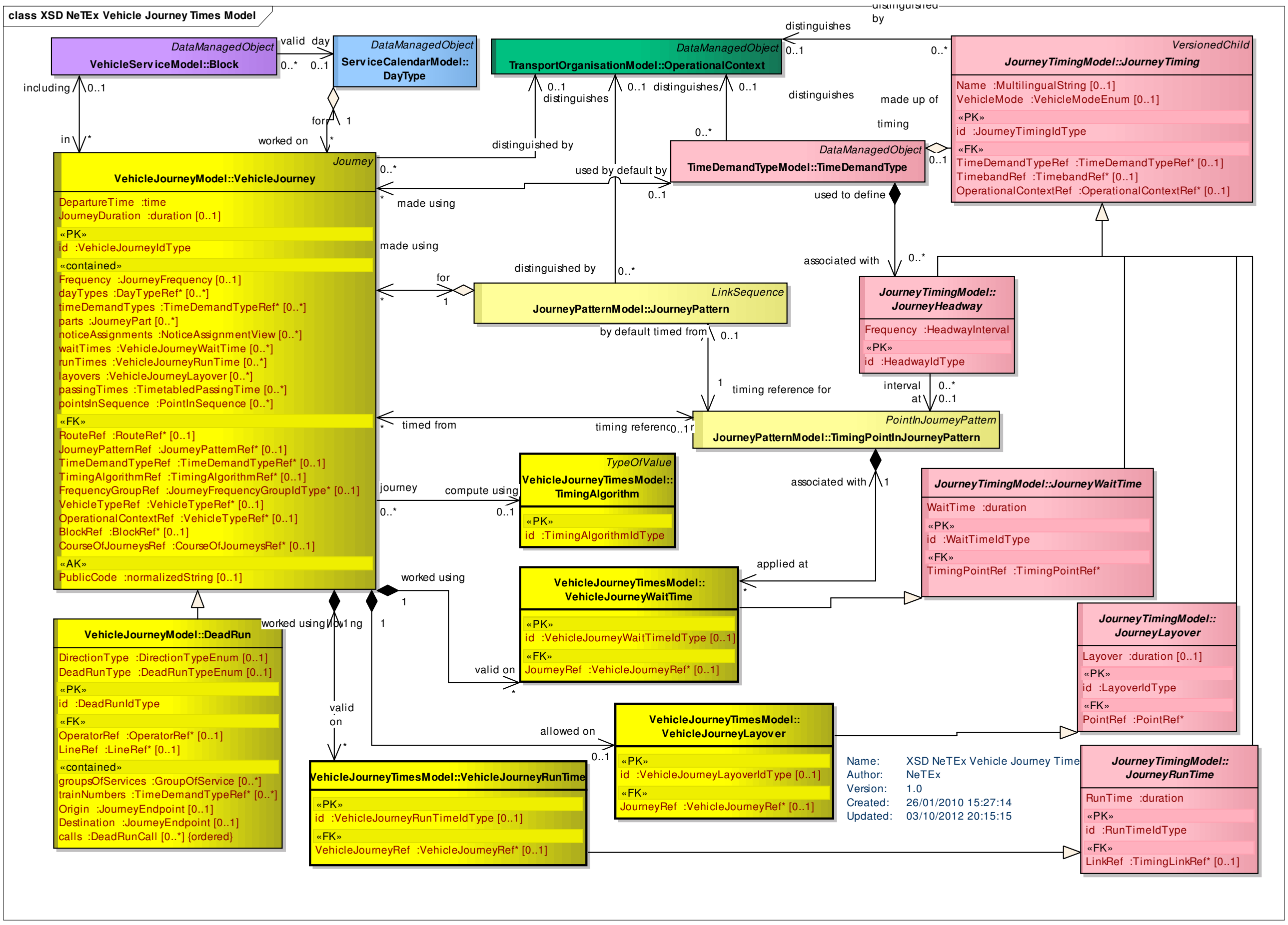

class NeTExTM TP Frequency Based Service MODEL
Name: NeTExTM TP Frequency Based Service MODEL
Author: NeTEx
Version: 1.0
Created: 17/03/2011 15:34:58
Updated: 03/10/2012 15:50:52
LINK SEQUENCE
Journey Pattern MODEL:: JOURNEY PATTERN
Journey Pattern Times MODEL::JOURNEY PATTERN HEADWAY
JOURNEY HEADWAY
VEHICLE JOURNEY HEADWAY
POINT IN LINK SEQUENCE
Journey Pattern MODEL:: TIMING POINT IN JOURNEY PATTERN
Time Demand Type MODEL::TIME DEMAND TYPE
JOURNEY RUN TIME
VEHICLE JOURNEY RUN TIME
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
Service Calendar MODEL::DAY TYPE
JOURNEY FREQUENCY GROUP
+ FirstDepartureTime
+ LastDepartureTime [0..1]
+ DayOffset [0..1]
«PK»
+ id [0..1]
Vehicle Journey MODEL::TEMPLATE VEHICLE JOURNEY
Vehicle Journey MODEL::DEAD RUN
Service Journey MODEL:: SERVICE JOURNEY
Service Journey MODEL:: TEMPLATE SERVICE JOURNEY
RHYTHMICAL JOURNEY GROUP
«PK»
+ id [0..1]
HEADWAY JOURNEY GROUP
+ HeadwayDisplay [0..1]
«PK»
+ id [0..1]
Service Calendar MODEL::TIME BAND
HEADWAY INTERVAL
+worked using
+for
0..*
1
+asciated with
*
+for
0..*
0..*
+by default timed from
0..1
1
+the timing reference for
+the timing reference for
1..
+used to define
1
+for
+used by default by
*
+the timing reference for
0..1
+made using
*
+timed from
*
+used by default by
0..*
+worked using
*
1
+valid on
+made using
*
+for
1..*
* +worked on
1..*
+composed of
+runs on
0..1
+is defined by
1..*
+defines
1..*
+made using
*
+active on
0..*
+active on
0..*
+determined by
0..*
+for
0..*
+for
0..*
+determining
1
Defines a group of JOURNEYs in order to describe special behavior like frequency based services or rythmical services (runs all xxh10, xxh25 and xxh45... for example; this is specially useful for passengenr information)
Services runing with a regular interval (every 10 mn... for example) This is specially useful for passengenr information. This is valid for specific timebands, on specific DAY TYPEs (through the JOURNEY GROUP).
Services having the same "rytm" every hour (runs all xxh10, xxh25 and xxh45... for example) This is specially useful for passengenr information. This is valid for specific timebands, on specific DAY TYPE (through the JOURNEY GROUP).

Figure 60 — Frequency Based Service – Conceptual MODEL (UML)

### 7.2.6.2 Vehicle Journey Times – Physical Model

The physical model is derived from the conceptual model. For the frequency-based services the concept of JOURNEY FREQUENCY is introduced. It is the frequency at which a given VEHICLE will run at a given TIMING POINT for a given TIME DEMAND TYPE. This frequency supersedes any global frequency or JOURNEY PATTERN FREQUENCY. HEADWAY INTERVAL gives a set of headway intervals to use to define a headway duration.

The physical model allows for a link between every VEHICLE JOURNEY with the timing algorithm used to compute times at timing points.


class XSD NeTEx Vehicle Journey Times Model
DataManagedObject
VehicleServiceModel::Block
DataManagedObject
ServiceCalendarModel::DayType
DataManagedObject
TransportOrganisationModel::OperationalContext
VersionedChild
JourneyTimingModel::JourneyTiming
Name :MultilingualString [0..1]
VehicleMode :VehicleModeEnum [0..1]
«PK»
id :JourneyTimingIdType
«FK»
TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
TimebandRef :TimebandRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
DataManagedObject
TimeDemandTypeModel::TimeDemandType
Journey
VehicleJourneyModel::VehicleJourney
DepartureTime :time
JourneyDuration :duration [0..1]
«PK»
id :VehicleJourneyIdType
«contained»
Frequency :JourneyFrequency [0..1]
dayTypes :DayTypeRef* [0..*]
timeDemandTypes :TimeDemandTypeRef* [0..*]
parts :JourneyPart [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
waitTimes :VehicleJourneyWaitTime [0..*]
runTimes :VehicleJourneyRunTime [0..*]
layovers :VehicleJourneyLayover [0..*]
passingTimes :TimetabledPassingTime [0..*]
pointsInSequence :PointInSequence [0..*]
«FK»
RouteRef :RouteRef* [0..1]
JourneyPatternRef :JourneyPatternRef* [0..1]
TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
TimingAlgorithmRef :TimingAlgorithmRef* [0..1]
FrequencyGroupRef :JourneyFrequencyGroupIdType* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
OperationalContextRef :VehicleTypeRef* [0..1]
BlockRef :BlockRef* [0..1]
CourseOfJourneysRef :CourseOfJourneysRef* [0..1]
«AK»
PublicCode :normalizedString [0..1]
LinkSequence
JourneyPatternModel::JourneyPattern
JourneyTimingModel::JourneyHeadway
Frequency :HeadwayInterval
«PK»
id :HeadwayIdType
PointInJourneyPattern
JourneyPatternModel::TimingPointInJourneyPattern
TypeOfValue
VehicleJourneyTimesModel::TimingAlgorithm
«PK»
id :TimingAlgorithmIdType
JourneyTimingModel::JourneyWaitTime
WaitTime :duration
«PK»
id :WaitTimeIdType
«FK»
TimingPointRef :TimingPointRef*
VehicleJourneyTimesModel::VehicleJourneyWaitTime
«PK»
id :VehicleJourneyWaitTimeIdType [0..1]
«FK»
JourneyRef :VehicleJourneyRef* [0..1]
JourneyTimingModel::JourneyLayover
Layover :duration [0..1]
«PK»
id :LayoverIdType
«FK»
PointRef :PointRef*
VehicleJourneyModel::DeadRun
DirectionType :DirectionTypeEnum [0..1]
DeadRunType :DeadRunTypeEnum [0..1]
«PK»
id :DeadRunIdType
«FK»
OperatorRef :OperatorRef* [0..1]
LineRef :LineRef* [0..1]
«contained»
groupsOfServices :GroupOfService [0..*]
trainNumbers :TimeDemandTypeRef* [0..*]
Origin :JourneyEndpoint [0..1]
Destination :JourneyEndpoint [0..1]
calls :DeadRunCall [0..*] {ordered}
VehicleJourneyTimesModel::VehicleJourneyLayover
«PK»
id :VehicleJourneyLayoverIdType [0..1]
«FK»
JourneyRef :VehicleJourneyRef* [0..1]
VehicleJourneyTimesModel::VehicleJourneyRunTime
«PK»
id :VehicleJourneyRunTimeIdType [0..1]
«FK»
VehicleJourneyRef :VehicleJourneyRef* [0..1]
JourneyTimingModel::JourneyRunTime
RunTime :duration
«PK»
id :RunTimeIdType
«FK»
LinkRef :TimingLinkRef* [0..1]
Name: XSD NeTEx Vehicle Journey Time
Author: NeTEx
Version: 1.0
Created: 26/01/2010 15:27:14
Updated: 03/10/2012 20:15:15
valid day
including
in
for
worked on
distinguished by
distinguishes
distinguished by
distinguishes
made up of
timing
used by default by
used to define
made using
made using
for
distinguished by
associated with
by default timed from
timing reference for
interval at
timed from
timing reference
journey
compute using
associated with
applied at
worked using
worked using
valid on
valid on
allowed on
0..1
0..*
1
*

## Figure 61 — Vehicle Journey Times – Physical Model (UML)

The VEHICLE JOURNEY TIMES physical model is directly derived from the conceptual model. However, for convenience of implementation purposes, an additional view element, a CALL is traduced, see later below. This assembles data related to a POINT IN JOURNEY PATTERN, as part of a VEHICLE JOURNEY.

In particular a CALL incorporates information concerning the time a vehicle passes a SCHEDULED STOP POINT or other POINT IN JOURNEY PATTERN. In the case where the PASSING TIMEs are known, this is represented by ‘arrival times’ and ‘departure times’. In the case of frequency-based services, although at an operational level there may still be known arrival and departure times for each journey, they are not necessarily exposed to the user, but rather the timings of vehicles at a stop may be presented as intervals, for example ‘Every 5 to 10 minutes’.

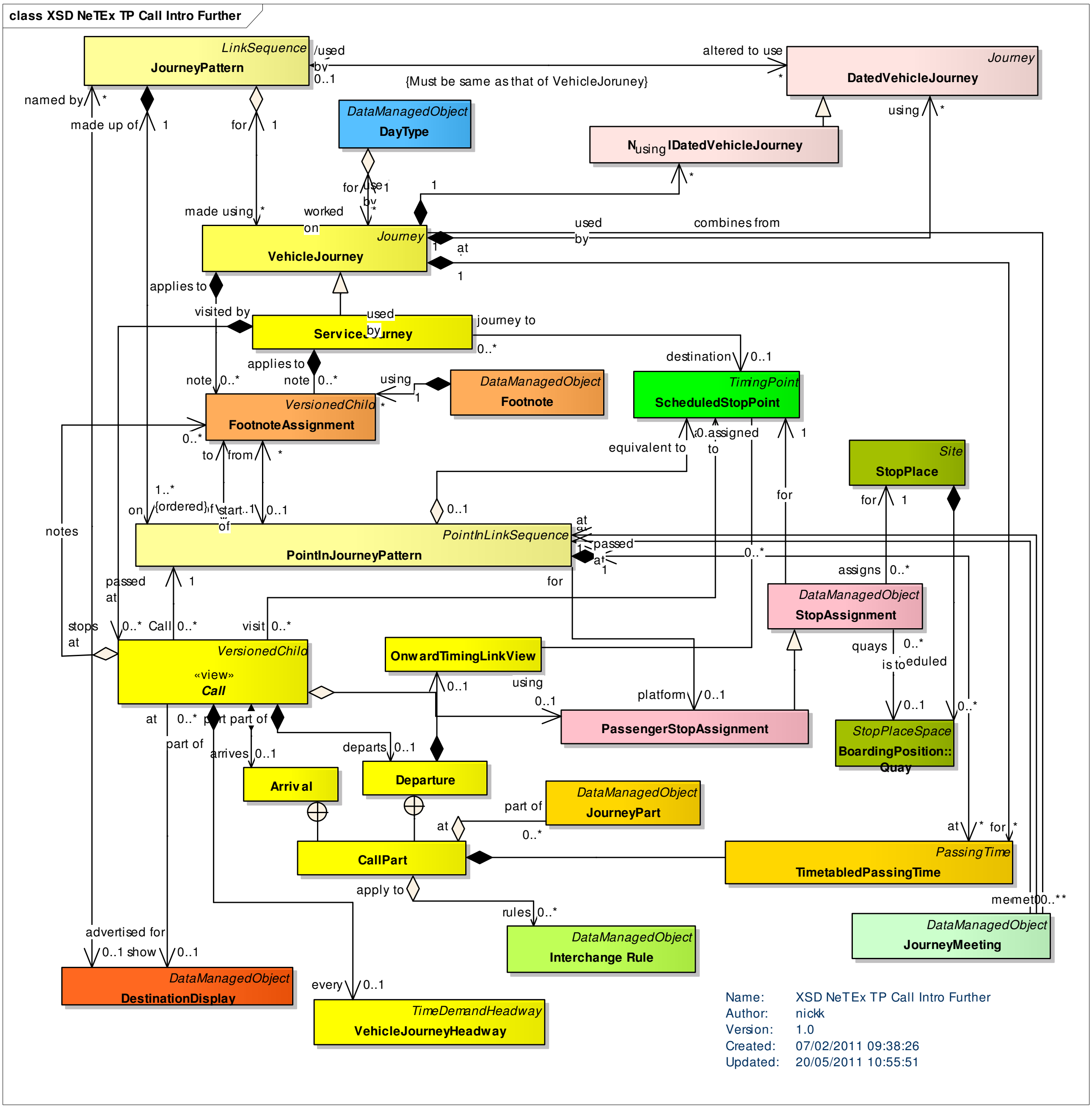


**Figure 62 — Call – Physical Model (UML)**

### 7.2.6.3 Vehicle Journey Times – Attributes and XSD

#### 7.2.6.3.1 VehicleJourneyWaitTime – Model Element

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

**Table 34 – *VehicleJourneyWaitTime* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyWaitTime* | ::> | VEHICLE JOURNEY WAIT TIME inherits from |

| | | | | |
|---|---|---|---|---|
| | | | | JOURNEY WAIT TIME. |
| «PK» | ***id*** | *VehicleJourneyWaitTimeIdType* | 0:1 | Identifier of VEHICLE JOURNEY WAIT TIME. |
| «FK» | ***Vehicle-JourneyRef*** | *VehicleJourneyRef* | 0:1 | Reference to a VEHICLE JOURNEY with which this WAIT TIME is associated. |

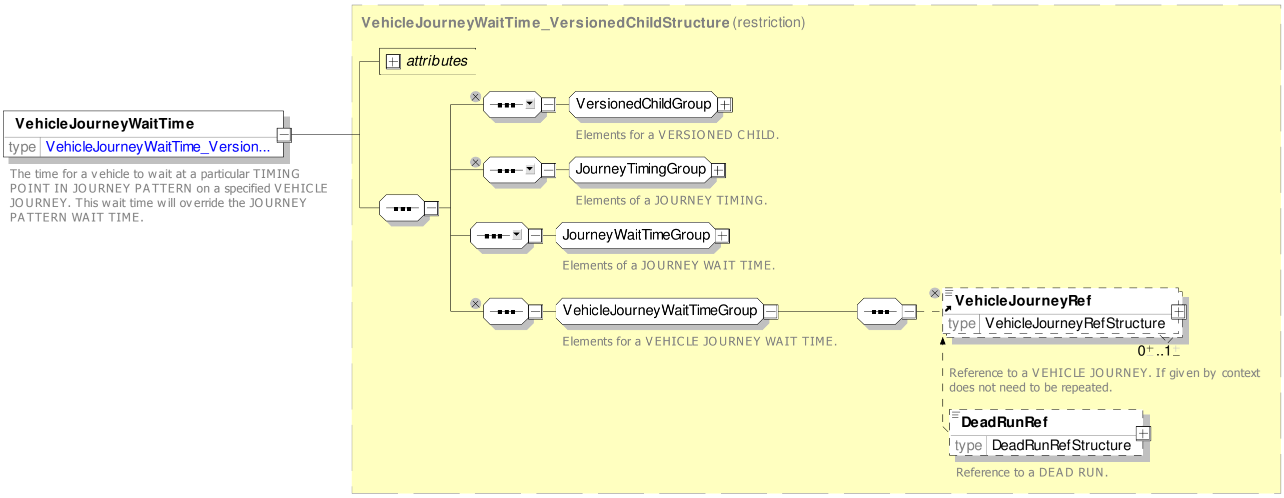

VehicleJourneyWaitTime_VersionedChildStructure (restriction)
attributes
VehicleJourneyWaitTime
type VehicleJourneyWaitTime_Version...
The time for a vehicle to wait at a particular TIMING POINT IN JOURNEY PATTERN on a specified VEHICLE JOURNEY. This wait time will override the JOURNEY PATTERN WAIT TIME.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyWaitTimeGroup
Elements of a JOURNEY WAIT TIME.
VehicleJourneyWaitTimeGroup
Elements for a VEHICLE JOURNEY WAIT TIME.
VehicleJourneyRef
type VehicleJourneyRefStructure
0..1
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.

**Figure 63 — *VehicleJourneyWaitTime* — XSD**

### 7.2.6.3.2 VehicleJourneyRunTime – Model Element

The VEHICLE JOURNEY RUN TIME is the time taken to traverse a specified TIMING LINK IN JOURNEY PATTERN on a specified VEHICLE JOURNEY. This gives the most detailed control over times and overrides the DEFAULT SERVICE JOURNEY RUN TIME and JOURNEY PATTERN RUN TIME and the DEFAULT DEAD RUN RUN TIME.

**Table 35 – *VehicleJourneyRunTime* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyRunTime* | ::> | VEHICLE JOURNEY RUN TIME inherits from JOURNEY RUN TIME. |
| «PK» | ***id*** | *VehicleJourneyRunTimeId Type* | 0:1 | Identifier of VEHICLE JOURNEY RUN TIME. |
| «FK» | ***VehicleJourneyRe f*** | *VehicleJourneyRef* | 0:1 | VEHICLE JOURNEY for which this is a RUN TIME. |

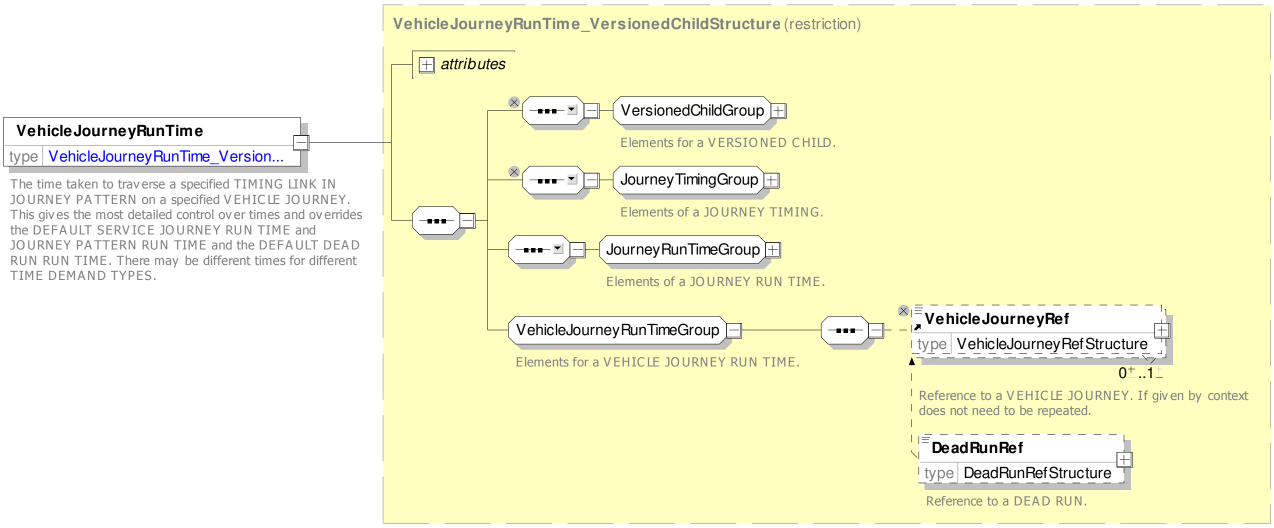

VehicleJourneyRunTime_VersionedChildStructure (restriction)
attributes
VehicleJourneyRunTime
type VehicleJourneyRunTime_Version...
The time taken to traverse a specified TIMING LINK IN JOURNEY PATTERN on a specified VEHICLE JOURNEY. This gives the most detailed control over times and overrides the DEFAULT SERVICE JOURNEY RUN TIME and JOURNEY PATTERN RUN TIME and the DEFAULT DEAD RUN RUN TIME. There may be different times for different TIME DEMAND TYPES.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyRunTimeGroup
Elements of a JOURNEY RUN TIME.
VehicleJourneyRunTimeGroup
Elements for a VEHICLE JOURNEY RUN TIME.
VehicleJourneyRef
type VehicleJourneyRefStructure
0..1
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.

**Figure 64 — *VehicleJourneyRunTime* — XSD**

### 7.2.6.3.3 VehicleJourneyLayover – Model Element

VEHICLE JOURNEY LAYOVER is a time allowance at the end of a specified VEHICLE JOURNEY. This time supersedes any global layover or JOURNEY PATTERN LAYOVER.

**Table 36 – *VehicleJourneyLayover* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *JourneyLayover* | ::> | VEHICLE JOURNEY LAYOVER inherits from JOURNEY LAYOVER. |
| «PK» | ***id*** | *VehicleJourneyLayoverIdType* | 0:1 | Identifier of VEHICLE JOURNEY LAYOVER. |
| «FK» | ***VehicleJourney Ref*** | *VehicleJourneyRef* | 0:1 | Reference to a VEHICLE JOURNEY with which this LAYOVER is associated. |

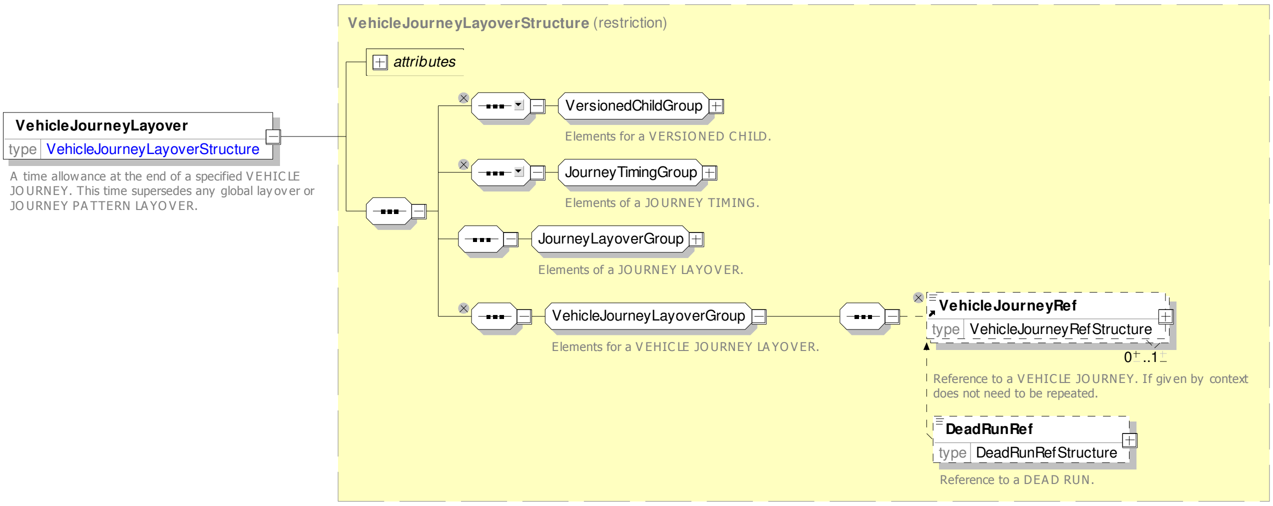

VehicleJourneyLayoverStructure (restriction)
attributes
VehicleJourneyLayover
type VehicleJourneyLayoverStructure
A time allowance at the end of a specified VEHICLE JOURNEY. This time supersedes any global layover or JOURNEY PATTERN LAYOVER.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
JourneyLayoverGroup
Elements of a JOURNEY LAYOVER.
VehicleJourneyLayoverGroup
Elements for a VEHICLE JOURNEY LAYOVER.
VehicleJourneyRef
type VehicleJourneyRefStructure
0..1
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.

**Figure 65 — *VehicleJourneyLayover* — XSD**

#### 7.2.6.3.4 TimingAlgorithm – Model Element

Timing algorithm used to compute times at timing points.

**Table 37 – *TimingAlgorithm* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *TypeOfValue* | ::> | TIMING ALGORITHM inherits from TYPE OF VALUE. |
| «PK» | ***id*** | *TimingAlgorithmIdType* | 1:1 | Identifier of TIMING ALGORITHM. |

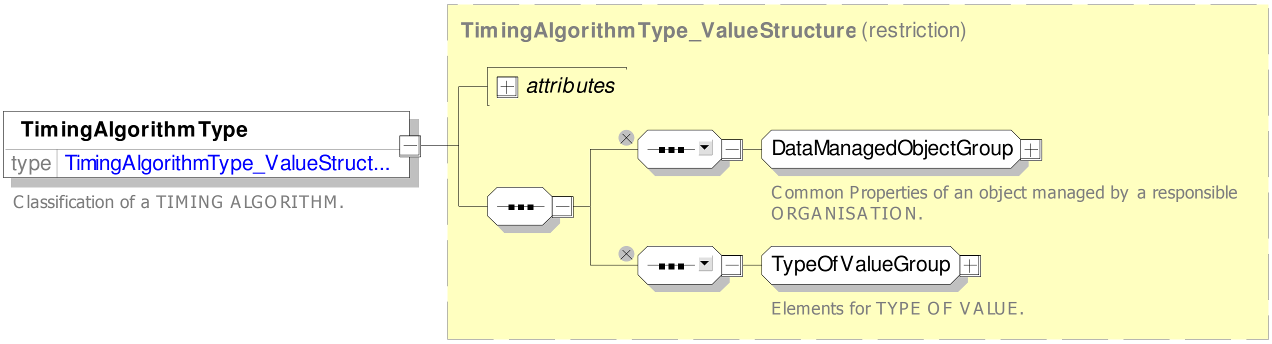

TimingAlgorithmType_ValueStructure (restriction)
attributes
TimingAlgorithmType
type TimingAlgorithmType_ValueStruct...
Classification of a TIMING ALGORITHM.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
TypeOfValueGroup
Elements for TYPE OF VALUE.

**Figure 66 — *TimingAlgorithmType* — XSD**

### 7.2.6.4 Vehicle Journey Frequency

#### 7.2.6.4.1 Vehicle Journey Frequency – Conceptual MODEL

Often in passenger information systems there is a requirement to show multiple journeys as a single column in a timetable for example “And then every 20 minutes until 6 pm”**.** The JOURNEY FREQUENCY model allows groups of journeys to be aggregated using a TEMPLATE JOURNEY and an overall frequency to be specified for them, using either a HEADWAY JOURNEY GROUP (e.g. ‘*every 10 minutes*’) or a RHYTHMICAL JOURNEY GROUP – one that runs at a regular interval past the hour (for example ‘*xxh10’*, ‘*xxh25’* and ‘*xxh45*’).

Even though every scheduled VEHICLE JOURNEY has a specific set of PASSING TIMEs that it will run to, these will not necessarily be revealed to the passenger and a JOURNEY FREQUENCY can be used to present the journey as frequency based.

For passenger information purposes it is possible to exchange a TEMPLATE VEHICLE JOURNEY without any concrete VEHICLE JOURNEYs.

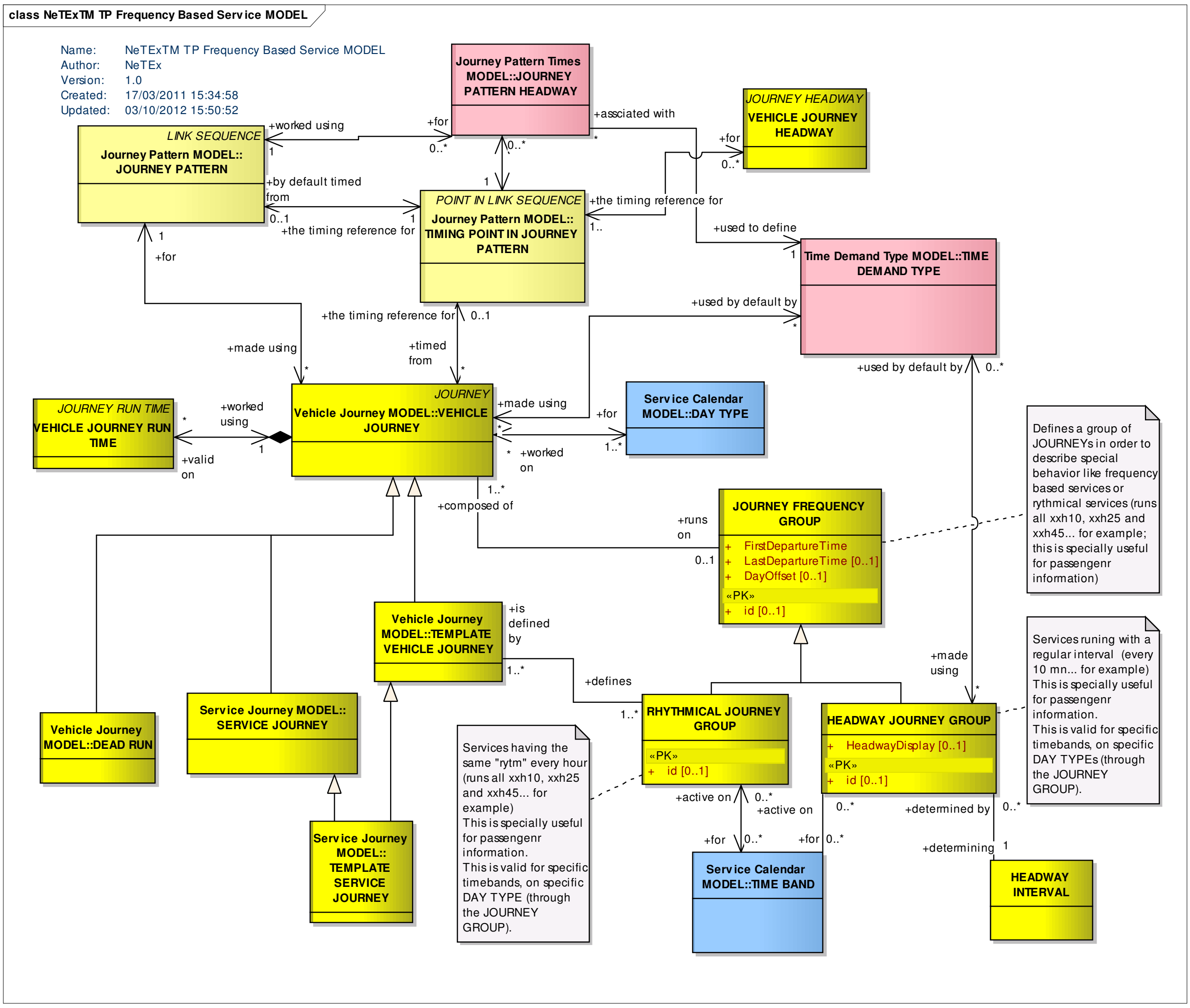

class NeTExTM TP Frequency Based Service MODEL
Name: NeTExTM TP Frequency Based Service MODEL
Author: NeTEx
Version: 1.0
Created: 17/03/2011 15:34:58
Updated: 03/10/2012 15:50:52
LINK SEQUENCE
Journey Pattern MODEL:: JOURNEY PATTERN
Journey Pattern Times MODEL::JOURNEY PATTERN HEADWAY
JOURNEY HEADWAY
VEHICLE JOURNEY HEADWAY
POINT IN LINK SEQUENCE
Journey Pattern MODEL:: TIMING POINT IN JOURNEY PATTERN
Time Demand Type MODEL::TIME DEMAND TYPE
JOURNEY RUN TIME
VEHICLE JOURNEY RUN TIME
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
Service Calendar MODEL::DAY TYPE
JOURNEY FREQUENCY GROUP
+ FirstDepartureTime
+ LastDepartureTime [0..1]
+ DayOffset [0..1]
«PK»
+ id [0..1]
Vehicle Journey MODEL::TEMPLATE VEHICLE JOURNEY
Vehicle Journey MODEL::DEAD RUN
Service Journey MODEL:: SERVICE JOURNEY
Service Journey MODEL:: TEMPLATE SERVICE JOURNEY
RHYTHMICAL JOURNEY GROUP
«PK»
+ id [0..1]
HEADWAY JOURNEY GROUP
+ HeadwayDisplay [0..1]
«PK»
+ id [0..1]
Service Calendar MODEL::TIME BAND
HEADWAY INTERVAL
+worked using
+for
1
0..*
0..*
+asciated with
*
+for
0..*
+by default timed from
0..1
1
+the timing reference for
1
+the timing reference for
1..
+used to define
1
+for
1
+used by default by
*
+the timing reference for
0..1
+made using
*
+timed from
*
+used by default by
0..*
+worked using
*
1
+valid on
+made using
+for
*
+worked on
1..*
1..*
+composed of
+runs on
0..1
+is defined by
1..*
+defines
1..*
+made using
*
+active on
0..*
+active on
+for
0..*
+for
0..*
0..*
+determined by
0..*
+determining
1
Defines a group of JOURNEYs in order to describe special behavior like frequency based services or rythmical services (runs all xxh10, xxh25 and xxh45... for example; this is specially useful for passengenr information)
Services runing with a regular interval (every 10 mn... for example) This is specially useful for passengenr information. This is valid for specific timebands, on specific DAY TYPEs (through the JOURNEY GROUP).
Services having the same "rytm" every hour (runs all xxh10, xxh25 and xxh45... for example) This is specially useful for passengenr information. This is valid for specific timebands, on specific DAY TYPE (through the JOURNEY GROUP).

Figure 67 — Vehicle Journey Frequency – Conceptual Model (UML)

### 7.2.6.4.2 Vehicle Journey Frequency – Physical Model

The following figure shows the Physical model for VEHICLE JOURNEY FREQUENCY with detailed attributes.

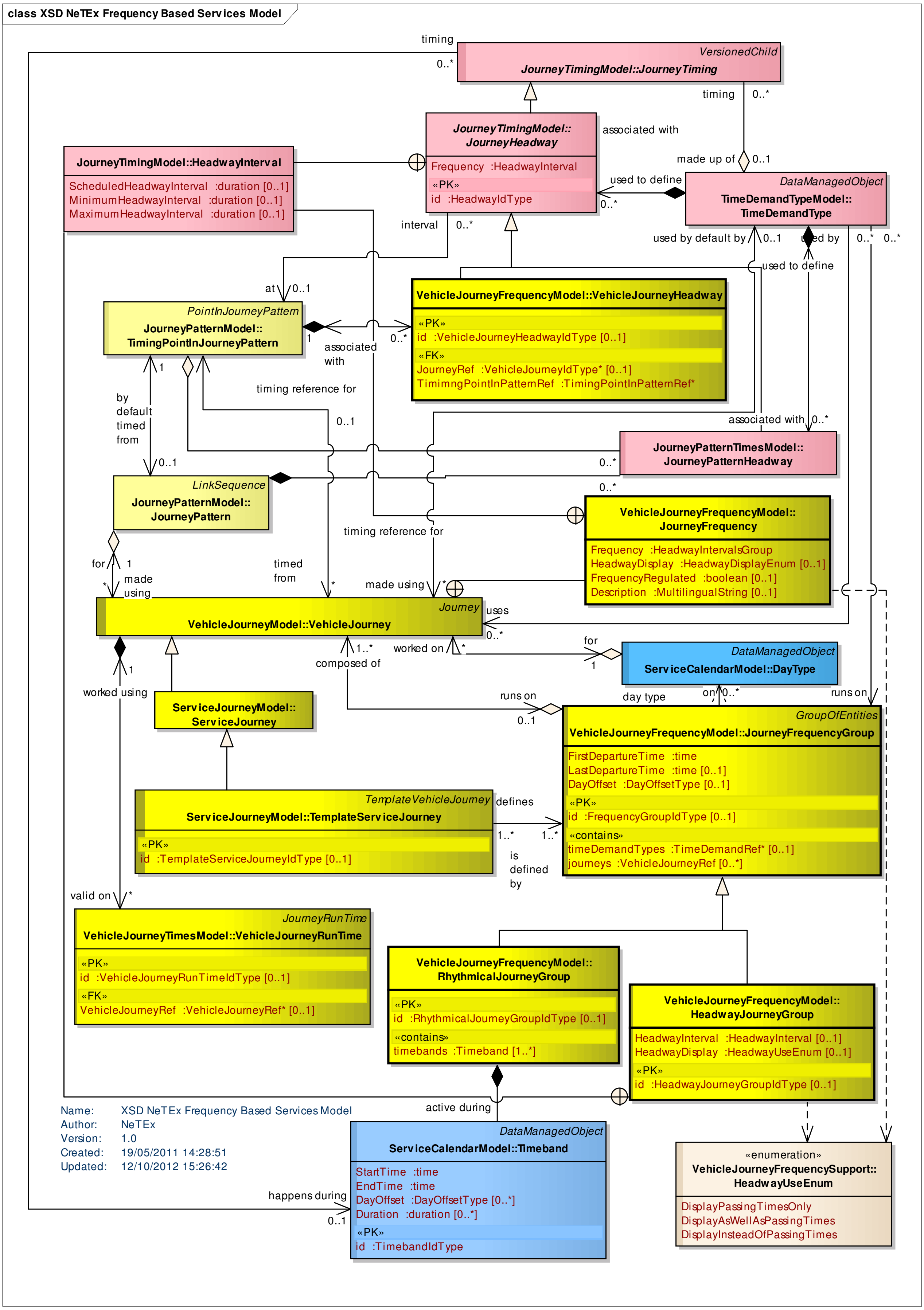

class XSD NeTEx Frequency Based Services Model
VersionedChild
JourneyTimingModel::JourneyTiming
timing
0..*
timing
0..*
JourneyTimingModel::JourneyHeadway
Frequency :HeadwayInterval
«PK»
id :HeadwayIdType
associated with
made up of
0..1
used to define
0..*
DataManagedObject
TimeDemandTypeModel::TimeDemandType
JourneyTimingModel::HeadwayInterval
ScheduledHeadwayInterval :duration [0..1]
MinimumHeadwayInterval :duration [0..1]
MaximumHeadwayInterval :duration [0..1]
interval
0..*
used by default by
0..1
used by
0..*
0..*
used to define
at
0..1
PointInJourneyPattern
JourneyPatternModel::TimingPointInJourneyPattern
1
associated with
0..*
VehicleJourneyFrequencyModel::VehicleJourneyHeadway
«PK»
id :VehicleJourneyHeadwayIdType [0..1]
«FK»
JourneyRef :VehicleJourneyIdType* [0..1]
TimimngPointInPatternRef :TimingPointInPatternRef*
1
by default timed from
0..1
timing reference for
0..1
associated with
0..*
JourneyPatternTimesModel::JourneyPatternHeadway
0..*
LinkSequence
JourneyPatternModel::JourneyPattern
0..*
VehicleJourneyFrequencyModel::JourneyFrequency
Frequency :HeadwayIntervalsGroup
HeadwayDisplay :HeadwayDisplayEnum [0..1]
FrequencyRegulated :boolean [0..1]
Description :MultilingualString [0..1]
timing reference for
for
1
made using
*
timed from
*
made using
*
Journey
VehicleJourneyModel::VehicleJourney
uses
0..*
1..*
composed of
worked on
*
for
1
DataManagedObject
ServiceCalendarModel::DayType
1
worked using
ServiceJourneyModel::ServiceJourney
runs on
0..1
day type
on
0..*
runs on
GroupOfEntities
VehicleJourneyFrequencyModel::JourneyFrequencyGroup
FirstDepartureTime :time
LastDepartureTime :time [0..1]
DayOffset :DayOffsetType [0..1]
«PK»
id :FrequencyGroupIdType [0..1]
«contains»
timeDemandTypes :TimeDemandRef* [0..1]
journeys :VehicleJourneyRef [0..*]
TemplateVehicleJourney
ServiceJourneyModel::TemplateServiceJourney
«PK»
id :TemplateServiceJourneyIdType [0..1]
defines
1..*
1..*
is defined by
valid on
*
JourneyRunTime
VehicleJourneyTimesModel::VehicleJourneyRunTime
«PK»
id :VehicleJourneyRunTimeIdType [0..1]
«FK»
VehicleJourneyRef :VehicleJourneyRef* [0..1]
VehicleJourneyFrequencyModel::RhythmicalJourneyGroup
«PK»
id :RhythmicalJourneyGroupIdType [0..1]
«contains»
timebands :Timeband [1..*]
VehicleJourneyFrequencyModel::HeadwayJourneyGroup
HeadwayInterval :HeadwayInterval [0..1]
HeadwayDisplay :HeadwayUseEnum [0..1]
«PK»
id :HeadwayJourneyGroupIdType [0..1]
active during
DataManagedObject
ServiceCalendarModel::Timeband
StartTime :time
EndTime :time
DayOffset :DayOffsetType [0..*]
Duration :duration [0..*]
«PK»
id :TimebandIdType
happens during
0..1
«enumeration»
VehicleJourneyFrequencySupport::HeadwayUseEnum
DisplayPassingTimesOnly
DisplayAsWellAsPassingTimes
DisplayInsteadOfPassingTimes
Name: XSD NeTEx Frequency Based Services Model
Author: NeTEx
Version: 1.0
Created: 19/05/2011 14:28:51
Updated: 12/10/2012 15:26:42

**Figure 68 — Vehicle Journey Frequency – Physical Model (UML)**

### 7.2.6.5 Vehicle Journey Frequency – Attributes and XSD

#### 7.2.6.5.1 JourneyFrequencyGroup – Model Element

JOURNEY FREQUENCY GROUP defines a set of JOURNEYs in order to describe special frequency behaviour like frequency based services or rhythmical services (one that runs at a regular interval past the hour, for example '*xxh10'*, '*xxh25'* and '*xxh45'*; this is especially useful for passenger information).

**Table 38 – *JourneyFrequencyGroup* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *GroupOfEntities* | *::>* | JOURNEY FREQUENCY GROUP inherits from GROUP OF ENTITies. |
| «PK» | ***id*** | *FrequencyGroupIdType* | 0:1 | Identifier of FREQUENCY GROUP. |
|  | ***FirstDeparture-Time*** | *xsd:time* | 1:1 | Time of first departure in FREQUENCY GROUP. |
|  | ***LastDeparture-Time*** | *xsd:time* | 0:1 | Time of last departure in FREQUENCY GROUP. |
|  | ***DayOffset*** | *xsd:integer* | 0:1 | Offset of end time day from start time. |
| «cntd» | ***timeDemand-Types*** | *TimeDemandRef* | 0:1 | TIME DEMAND TYPEs for which this FREQUENCY GROUP applies. |
| «cntd» | ***journeys*** | *VehicleJourneyRef* | 0:* | Journeys belonging to FREQUENCY GROUP. |

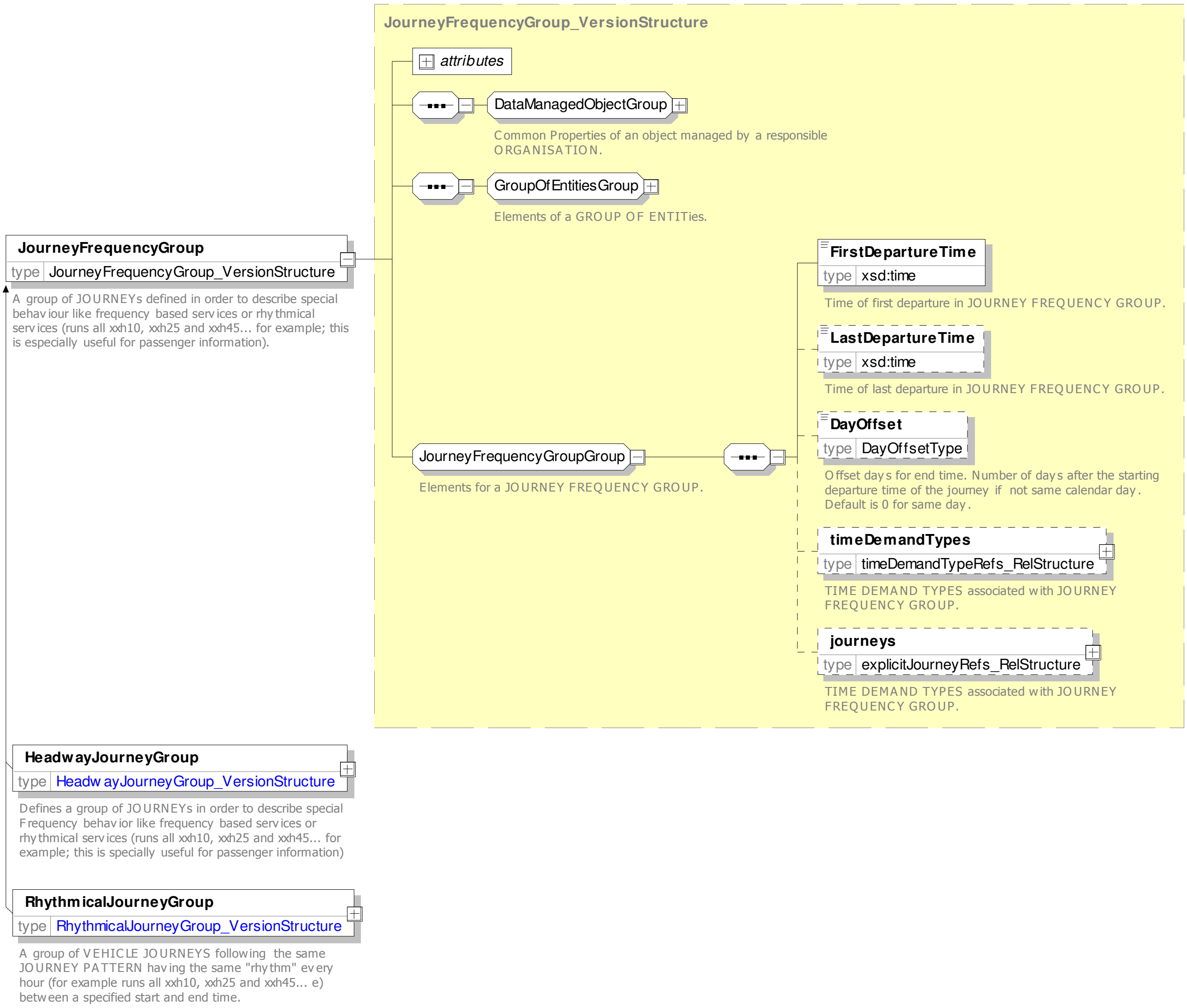

JourneyFrequencyGroup_VersionStructure
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
GroupOfEntitiesGroup
Elements of a GROUP OF ENTITies.
JourneyFrequencyGroup
type JourneyFrequencyGroup_VersionStructure
A group of JOURNEYs defined in order to describe special behaviour like frequency based services or rhythmical services (runs all xxh10, xxh25 and xxh45... for example; this is especially useful for passenger information).
FirstDepartureTime
type xsd:time
Time of first departure in JOURNEY FREQUENCY GROUP.
LastDepartureTime
type xsd:time
Time of last departure in JOURNEY FREQUENCY GROUP.
DayOffset
type DayOffsetType
Offset days for end time. Number of days after the starting departure time of the journey if not same calendar day. Default is 0 for same day.
JourneyFrequencyGroupGroup
Elements for a JOURNEY FREQUENCY GROUP.
timeDemandTypes
type timeDemandTypeRefs_RelStructure
TIME DEMAND TYPES associated with JOURNEY FREQUENCY GROUP.
journeys
type explicitJourneyRefs_RelStructure
TIME DEMAND TYPES associated with JOURNEY FREQUENCY GROUP.
HeadwayJourneyGroup
type HeadwayJourneyGroup_VersionStructure
Defines a group of JOURNEYs in order to describe special Frequency behavior like frequency based services or rhythmical services (runs all xxh10, xxh25 and xxh45... for example; this is specially useful for passenger information)
RhythmicalJourneyGroup
type RhythmicalJourneyGroup_VersionStructure
A group of VEHICLE JOURNEYS following the same JOURNEY PATTERN having the same "rhythm" every hour (for example runs all xxh10, xxh25 and xxh45... e) between a specified start and end time.

Figure 69 — *JourneyFrequencyGroup* — XSD

### 7.2.6.5.2 HeadwayJourneyGroup – Model Element

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*’). This is especially useful for presenting passenger information.

Table 39 – *HeadwayJourneyGroup* – 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. |
| | ***ScheduledHeadwayInterval*** | *xsd:duration* | 0:1 | Scheduled normal HEADWAY INTERVAL. |

| | | | | |
|---|---|---|---|---|
| | ***Minimum-HeadwayInterval*** | *xsd:duration* | 0:1 | Minimum HEADWAY INTERVAL. |
| | ***Maximum-HeadwayInterval*** | *xsd:duration* | 0:1 | Maximum HEADWAY INTERVAL. |
| | ***HeadwayDisplay*** | *HeadwayUseEnum* | 0:1 | How headway is to be displayed to passengers. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of Headway Journey Groups. |

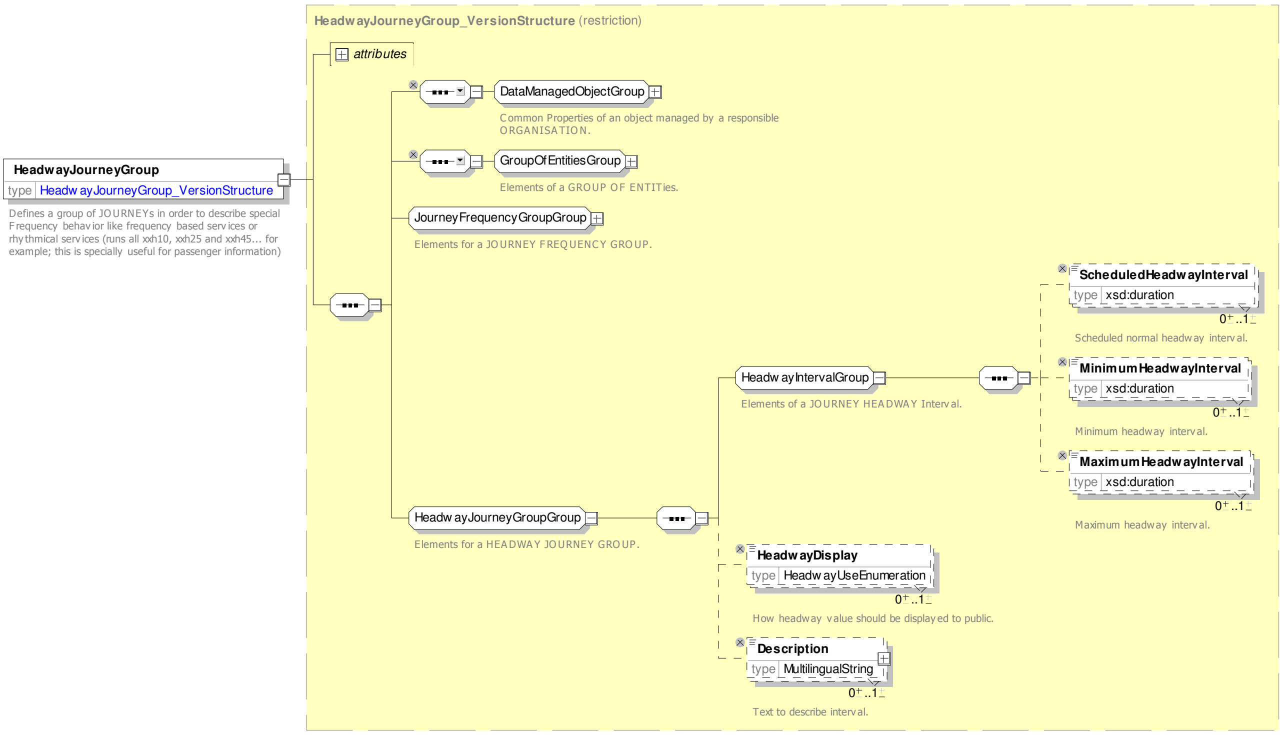

HeadwayJourneyGroup_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
GroupOfEntitiesGroup
Elements of a GROUP OF ENTITies.
JourneyFrequencyGroupGroup
Elements for a JOURNEY FREQUENCY GROUP.
HeadwayJourneyGroup
type HeadwayJourneyGroup_VersionStructure
Defines a group of JOURNEYs in order to describe special Frequency behavior like frequency based services or rhythmical services (runs all xxh10, xxh25 and xxh45... for example; this is specially useful for passenger information)
ScheduledHeadwayInterval
type xsd:duration
0..1
Scheduled normal headway interval.
HeadwayIntervalGroup
Elements of a JOURNEY HEADWAY Interval.
MinimumHeadwayInterval
type xsd:duration
0..1
Minimum headway interval.
MaximumHeadwayInterval
type xsd:duration
0..1
Maximum headway interval.
HeadwayJourneyGroupGroup
Elements for a HEADWAY JOURNEY GROUP.
HeadwayDisplay
type HeadwayUseEnumeration
0..1
How headway value should be displayed to public.
Description
type MultilingualString
0..1
Text to describe interval.

**Figure 70 — *HeadwayJourneyGroup* — XSD**

#### 7.2.6.5.2.1 HeadwayDisplay – Allowed values

The following table shows the allowed values for ***HeadwayDisplay** (HeadwayDisplayEnum).*

**Table 40 – *HeadwayDisplay* – Allowed values**

| Value | Description |
|---|---|
| *displayInsteadOfPassingTimes* | Display intervals instead of passing times. |
| *displayAsWellAsPassingTimes* | Display intervals as well as passing times. |
| *displayPassingTimesOnly* | Display only passing times. |

### 7.2.6.5.3 RhythmicalJourneyGroup – Model Element

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

Table 41 – *RhythmicalJourneyGroup* – Element

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

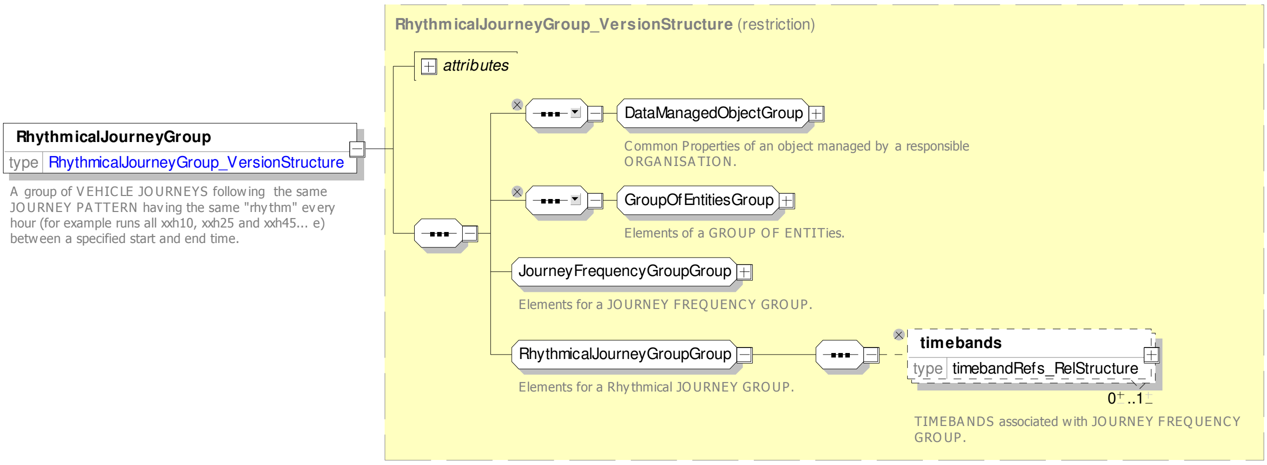

RhythmicalJourneyGroup
type RhythmicalJourneyGroup_VersionStructure
A group of VEHICLE JOURNEYS following the same JOURNEY PATTERN having the same "rhythm" every hour (for example runs all xxh10, xxh25 and xxh45... e) between a specified start and end time.
RhythmicalJourneyGroup_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
GroupOfEntitiesGroup
Elements of a GROUP OF ENTITies.
JourneyFrequencyGroupGroup
Elements for a JOURNEY FREQUENCY GROUP.
RhythmicalJourneyGroupGroup
Elements for a Rhythmical JOURNEY GROUP.
timebands
type timebandRefs_RelStructure
0..1
TIMEBANDS associated with JOURNEY FREQUENCY GROUP.

Figure 71 — *RhythmicalJourneyGroup* — XSD

### 7.2.6.5.4 VehicleJourneyHeadway – Model Element

VEHICLE JOURNEY HEADWAY describes the headway frequency information to use when presenting an individual VEHICLE JOURNEY (to be understood as the delay between the previous and the next VEHICLE JOURNEY). This information shall be consistent with HEADWAY JOURNEY GROUP if the JOURNEY is part of a FREQUENCY GROUP.

Table 42 – *VehicleJourneyHeadway* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *HeadwayInterval* | ::> | VEHICLE JOURNEY HEADWAY inherits from HEADWAY INTERVAL. |
| «PK» | ***id*** | *VehicleJourney-HeadwayIdType* | 0:1 | Identifier of VEHICLE JOURNEY HEADWAY. |
| «FK» | ***JourneyRef*** | *VehicleJourneyRef* | 0:1 | Reference to JOURNEY associated with VEHICLE JOURNEY HEADWAY. |
| «FK» | ***TimingPoint-InPatternRef*** | *TimingPointInPatternRef* | 1:1 | Reference to a TIMING POINT in JOURNEY PATTERN associated with VEHICLE JOURNEY HEADWAY. |

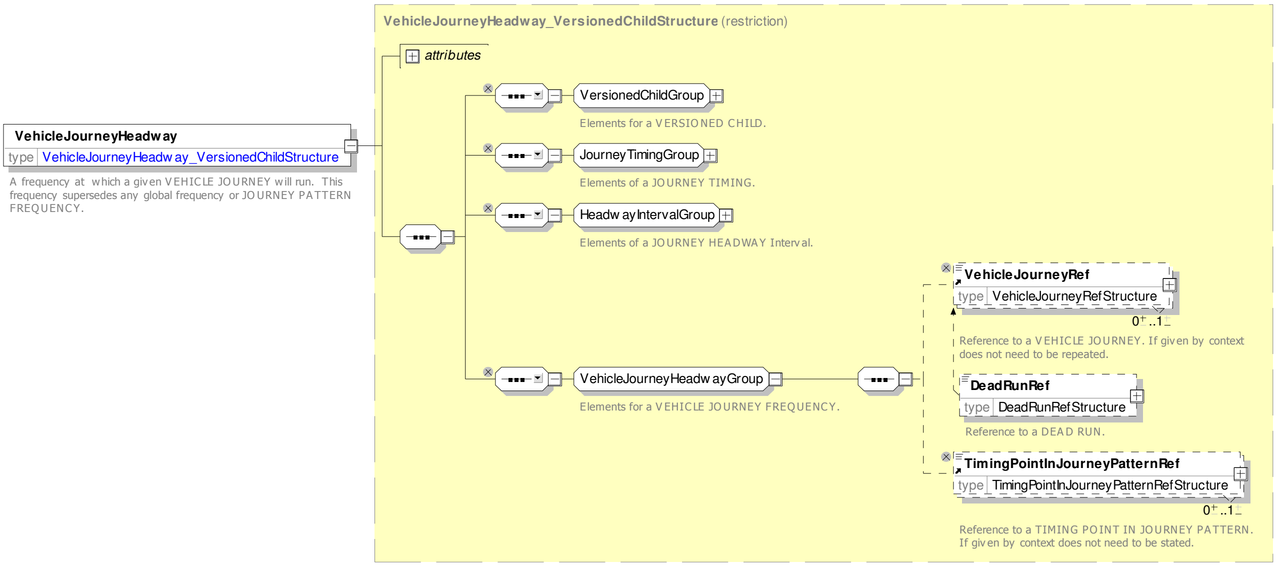

VehicleJourneyHeadway_VersionedChildStructure (restriction)
attributes
VehicleJourneyHeadway
type VehicleJourneyHeadway_VersionedChildStructure
A frequency at which a given VEHICLE JOURNEY will run. This frequency supersedes any global frequency or JOURNEY PATTERN FREQUENCY.
VersionedChildGroup
Elements for a VERSIONED CHILD.
JourneyTimingGroup
Elements of a JOURNEY TIMING.
HeadwayIntervalGroup
Elements of a JOURNEY HEADWAY Interval.
VehicleJourneyHeadwayGroup
Elements for a VEHICLE JOURNEY FREQUENCY.
VehicleJourneyRef
type VehicleJourneyRefStructure
0..1
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.
TimingPointInJourneyPatternRef
type TimingPointInJourneyPatternRefStructure
0..1
Reference to a TIMING POINT IN JOURNEY PATTERN. If given by context does not need to be stated.

Figure 72 — *VehicleJourneyHeadway* — XSD

### 7.2.6.5.5 JourneyFrequency – Model Element

JOURNEY FREQUENCY describes a frequency at which a given VEHICLE will run at a given TIMING POINT for a given TIME DEMAND TYPE. This frequency supersedes any global frequency or JOURNEY PATTERN FREQUENCY.

Table 43 – *JourneyFrequency* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| | ***Description*** | *MultilingualString* | 0:1 | Text description to use for frequency. E.g. *'Every five minutes'.* |
| | ***ScheduledHeadwayInterval*** | *duration* | 0:1 | Scheduled normal HEADWAY INTERVAL. |
| | ***Minimum-HeadwayInterval*** | *duration* | 0:1 | Minimum HEADWAY INTERVAL. |
| | ***Maximum-HeadwayInterval*** | *duration* | 0:1 | Maximum HEADWAY INTERVAL. |
| | ***FrequencyRegulated*** | *boolean* | 0:1 | Whether frequency falls under regulatory agreement or not. |
| | ***HeadwayDisplay*** | *HeadwayUseEnum* | 0:1 | Use to be made of HEADWAY INTERVAL information when displaying to public. Default is 'Display Instead of Passing Times'. |

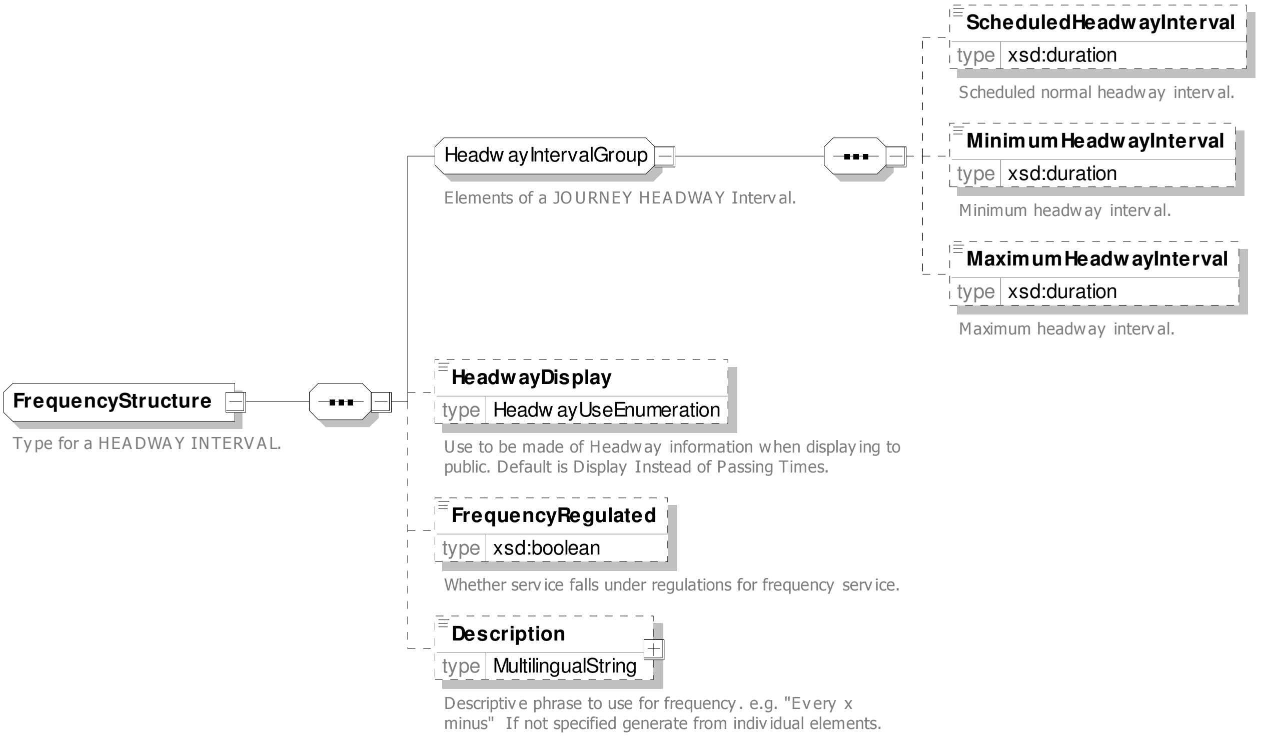

ScheduledHeadwayInterval
type xsd:duration
Scheduled normal headway interval.
MinimumHeadwayInterval
type xsd:duration
Minimum headway interval.
MaximumHeadwayInterval
type xsd:duration
Maximum headway interval.
HeadwayIntervalGroup
Elements of a JOURNEY HEADWAY Interval.
FrequencyStructure
Type for a HEADWAY INTERVAL.
HeadwayDisplay
type HeadwayUseEnumeration
Use to be made of Headway information when displaying to public. Default is Display Instead of Passing Times.
FrequencyRegulated
type xsd:boolean
Whether service falls under regulations for frequency service.
Description
type MultilingualString
Descriptive phrase to use for frequency. e.g. "Every x minus" If not specified generate from individual elements.

Figure 73 — *JourneyFrequency* — XSD

### 7.2.6.6 XML Example of Frequency Journeys

#### 7.2.6.6.1 Frequency Journey Group - XML Example

The following code fragment shows a TEMPLATE VEHICLE JOURNEY two different frequency groups.

For EXAMPLE

```
<TemplateServiceJourney version="any" id="hde:tvjh_24o_01">
            <DepartureTime>10:00:00.0Z</DepartureTime>
            <dayTypes>
                <DayTypeRef version="any" ref="hde:DT_01MTWTFSS"/>
            </dayTypes>
                <ServicePatternRef ref="hde:svp_24o">EXTERNAL</ServicePatternRef>
            <LineRef version="any" ref="mybus:Line:LN_234"/>
            <calls>
                <Call id="hde:tvjh_24o_02_001" version="any" order="1">
                    <ScheduledStopPointRef version="any" ref="mybus:SSP_001"/>
                    <Arrival>
                        <ForAlighting>false</ForAlighting>
                    </Arrival>
                    <Departure>
                        <Time>10:20:00.0Z</Time>
                    </Departure>
                </Call>
                <Call id="hde:tvjh_24o_02_002" version="any" order="2">
                    <ScheduledStopPointRef version="any" ref="mybus:SSP_002"/>
                    <Arrival>
                        <Time>11:20:00.0Z</Time>
                    </Arrival>
                    <Departure>
                        <Time>11:22:00.0Z</Time>
                        <WaitTime>PT2M</WaitTime>
                    </Departure>
                </Call>
                <Call id="hde:tvjh_24o_02_003" version="any" order="3">
                    <ScheduledStopPointRef version="any" ref="mybus:SSP_077"/>
                    <Arrival>
                        <Time>12:20:00.0Z</Time>
            </Arrival>
            <Departure>
                <ForBoarding>false</ForBoarding>
            </Departure>
        </Call>
    </calls>
    <frequencyGroups>
        <HeadwayJourneyGroup version="any" id="hde:hjg_24o_01">
            <Name>Regular Interval service between 10am and 12:00 pm</Name>
            <Description>About every 12 minutes</Description>
            <FirstDepartureTime>10:00:00</FirstDepartureTime>
            <LastDepartureTime>12:00:00</LastDepartureTime>
            <ScheduledHeadwayInterval>PT12M</ScheduledHeadwayInterval>
            <HeadwayDisplay>DisplayInsteadOfPassingTimes</HeadwayDisplay>
        </HeadwayJourneyGroup>
        <HeadwayJourneyGroup version="any" id="hde:hjg_24o_02">
            <Name>Regular Interval service between 12am and 18:00 pm</Name>
            <Description>About every 20 minutes</Description>
            <FirstDepartureTime>12:00:00</FirstDepartureTime>
            <LastDepartureTime>18:00:00</LastDepartureTime>
            <ScheduledHeadwayInterval>PT20M</ScheduledHeadwayInterval>
            <HeadwayDisplay>DisplayInsteadOfPassingTimes</HeadwayDisplay>
        </HeadwayJourneyGroup>
    </frequencyGroups>
</TemplateServiceJourney>
```

#### 7.2.6.6.2 Rhythmical Journey Group - XML Example

The following code fragment shows a TEMPLATE VEHICLE JOURNEY with a RHYTHMICAL JOURNEY GROUP.

For EXAMPLE

```
<TemplateServiceJourney version="any" id="hde:tvj_24o_02">
            <DepartureTime>10:20:00.0Z</DepartureTime>
            <dayTypes>
                <DayTypeRef version="any" ref="hde:DT_01MTWTFSS"/>
            </dayTypes>
            <ServicePatternRef ref="hde::jp_24o"/>
            <LineRef version="any" ref="mybus:LN_234"/>
            <calls>
                <Call id="hde:tvj_24o_02_001" version="any" order="1">
                    <ScheduledStopPointRef version="any" ref="mybus:SSP_001"/>
                    <Arrival>
                        <ForAlighting>false</ForAlighting>
                    </Arrival>
                    <Departure>
                        <Time>10:20:00.0Z</Time>
                    </Departure>
                    <Frequency>
                        <ScheduledHeadwayInterval>PT8M</ScheduledHeadwayInterval>
                    </Frequency>
                </Call>
                <Call id="hde:Ca:tvj_24o_02_002" version="any" order="2">
                    <ScheduledStopPointRef version="any" ref="mybus:SSP_002"/>

                    <Departure>
                        <Time>11:22:00.0Z</Time>
                    </Departure>
                    <Frequency>
                        <ScheduledHeadwayInterval>PT8M</ScheduledHeadwayInterval>
                    </Frequency>
                </Call>
                <Call id="hde:tvj_24o_02_003" version="any" order="3">
                    <ScheduledStopPointRef version="any" ref="mybus:SSP_077"/>
                    <Arrival>
                        <Time>12:20:00.0Z</Time>
                    </Arrival>
                    <Departure>
                        <ForBoarding>false</ForBoarding>
                    </Departure>
                    <Frequency>
                        <ScheduledHeadwayInterval>PT8M</ScheduledHeadwayInterval>
            </Frequency>
        </Call>
    </calls>
    <frequencyGroups>
        <RhythmicalJourneyGroup version="any" id="hde:rjg_24o_01">
            <Name>Regular Interval service between 10am and 17:00 pm</Name>
            <Description>At 20 &amp; 45 Minutes past the hour </Description>
            <FirstDepartureTime>10:00:00</FirstDepartureTime>
            <LastDepartureTime>17:00:00</LastDepartureTime>
            <timebands>
                <TimebandRef version="any" ref="hde:TM_20"/>
                <TimebandRef version="any" ref="hde:TM_45"/>
            </timebands>
        </RhythmicalJourneyGroup>
    </frequencyGroups>
</TemplateServiceJourney>
```

### 7.2.7 Interchange

#### 7.2.7.1 INTERCHANGE – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

In order to reach the destination of a trip if there is no direct service between the origin and the destination SCHEDULED STOP POINTs, a passenger will have to interchange between vehicles. A transfer will be necessary where the passenger will leave the vehicle at a particular SCHEDULED STOP POINT and enter another vehicle (which serves another SERVICE JOURNEY, usually on a different LINE) at the same or another SCHEDULED STOP POINT.

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

Sometimes both SERVICE JOURNEYs are planned to stop at exactly the same SCHEDULED STOP POINT but usually there are two different (but nearby) SCHEDULED STOP POINTs involved. In such a case, these SCHEDULED STOP POINTs will often (but need not necessarily) belong to the same STOP AREA. Passengers will have to walk a certain distance after disembarking from the feeder SERVICE JOURNEY to reach the stop position of the distributor SERVICE JOURNEY.

A CONNECTION expresses that there is a possible walking link that is suitable for a passenger to interchange from one public transport vehicle to another between two specified SCHEDULED STOP POINTs and the time allocated for a passenger to traverse the link.

Software used to control guaranteed interchanges can use the time information given to use a CONNECTION link as to assist calculating how long a distributor SERVICE JOURNEY needs to wait after a fetcher SERVICE JOURNEY has arrived before it can depart. If no specific CONNECTION link is available, timings from a DEFAULT CONNECTION may be used.

The connection time information could also be used for passenger information such as in travel planners, however sometimes more detailed information about timings and suitable paths adapted to the individual traveller’s specific preferences and capabilities is necessary. In that case additional information can be retrieved from attributes related to the physical model, such as PATH LINKs and NAVIGATION PATHs (see NeTEx Part 1).

In some cases 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 be adapted to the fact that the passenger should not change vehicle as the transfer is implicit. In this case it is also important that operation control staff is aware of the consequences to passengers if the operation is altered in such a way that two different vehicles are used for the two involved SERVICE JOURNEYs.

In real-time operation, interchanges that are advertised to the public should be controlled. For instance, vehicles may sometimes have to wait if the arriving vehicle in an interchange is delayed. The company will probably define certain rules and instructions for drivers on how to react in case of deviations from the planned schedule.

The SERVICE JOURNEY INTERCHANGE may be used to store some information on the planned interchange between two SERVICE JOURNEYs such as if 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.

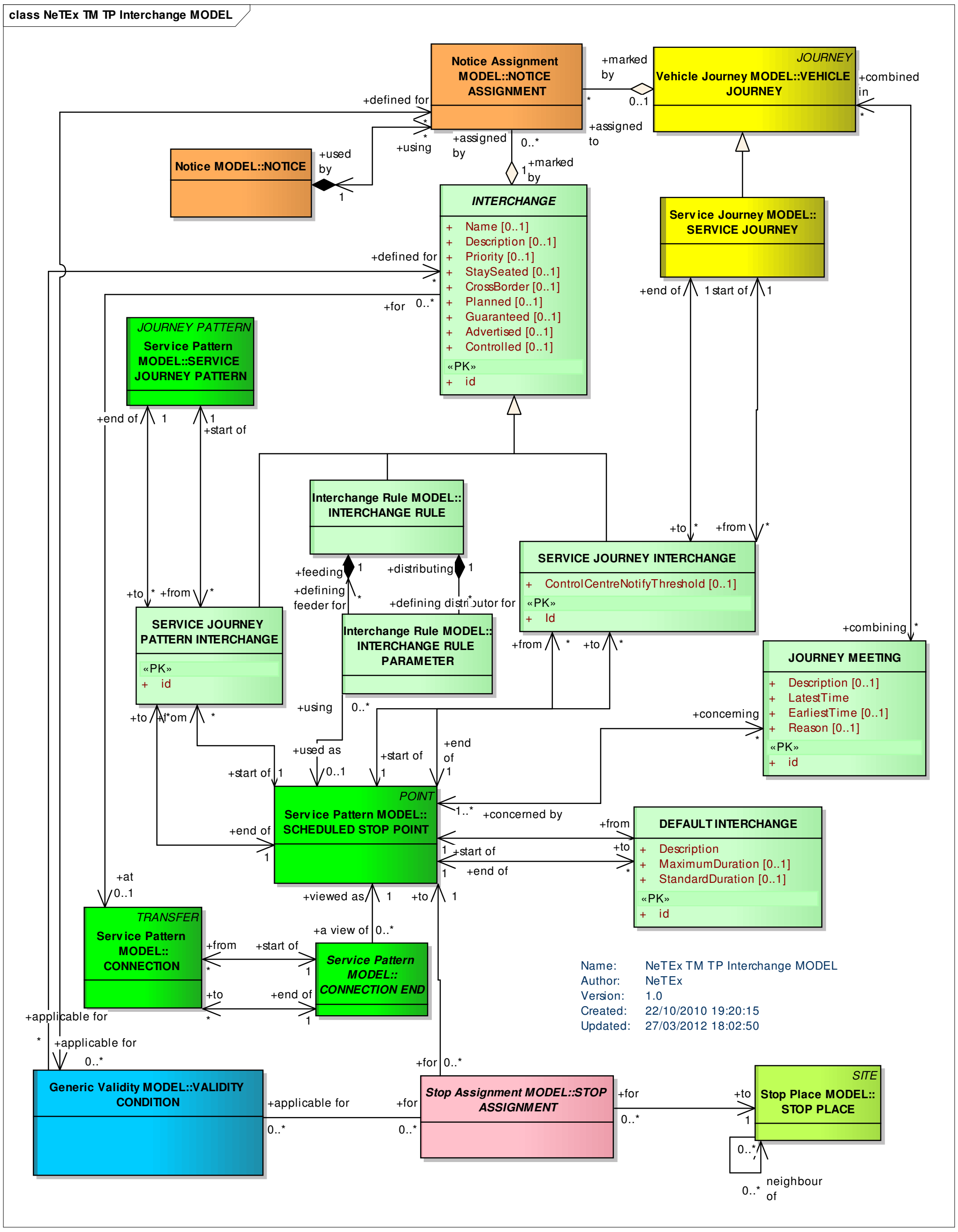

class NeTEx TM TP Interchange MODEL
Notice Assignment MODEL::NOTICE ASSIGNMENT
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
Notice MODEL::NOTICE
INTERCHANGE
+ Name [0..1]
+ Description [0..1]
+ Priority [0..1]
+ StaySeated [0..1]
+ CrossBorder [0..1]
+ Planned [0..1]
+ Guaranteed [0..1]
+ Advertised [0..1]
+ Controlled [0..1]
«PK»
+ id
Service Journey MODEL:: SERVICE JOURNEY
JOURNEY PATTERN
Service Pattern MODEL::SERVICE JOURNEY PATTERN
Interchange Rule MODEL:: INTERCHANGE RULE
SERVICE JOURNEY INTERCHANGE
+ ControlCentreNotifyThreshold [0..1]
«PK»
+ Id
SERVICE JOURNEY PATTERN INTERCHANGE
«PK»
+ id
Interchange Rule MODEL:: INTERCHANGE RULE PARAMETER
JOURNEY MEETING
+ Description [0..1]
+ LatestTime
+ EarliestTime [0..1]
+ Reason [0..1]
«PK»
+ id
POINT
Service Pattern MODEL:: SCHEDULED STOP POINT
DEFAULT INTERCHANGE
+ Description
+ MaximumDuration [0..1]
+ StandardDuration [0..1]
«PK»
+ id
TRANSFER
Service Pattern MODEL:: CONNECTION
Service Pattern MODEL:: CONNECTION END
Generic Validity MODEL::VALIDITY CONDITION
Stop Assignment MODEL::STOP ASSIGNMENT
SITE
Stop Place MODEL:: STOP PLACE
+defined for
+marked by
+combined in
+assigned to
+assigned by
+using
+used by
+marked by
+for
+end of
+start of
1 start of
+to
+from
+feeding
+distributing
+defining feeder for
+defining distributor for
+combining
+using
+concerning
+used as
+concerned by
+viewed as
+a view of
+at
+applicable for
neighbour of
0..1
0..*
1..*
1
*
Name: NeTEx TM TP Interchange MODEL
Author: NeTEx
Version: 1.0
Created: 22/10/2010 19:20:15
Updated: 27/03/2012 18:02:50

**Figure 74 — Interchange – Conceptual MODEL (UML)**

### 7.2.7.1.1 Default Timings for Interchanges

Some timing values for SERVICE JOURNEY INTERCHANGEs may be derived from other NeTEx elements during the planning and scheduling processing.

When optimising the start times of the SERVICE JOURNEYs, the scheduler has to take into account certain parameters affecting service quality, for example the influence of passenger waiting times at interchanges on their total travel times. Ideally the waiting times for passengers at interchanges resulting from the journey schedule should be as short as possible, allowing for necessary transfer time. But on the other hand, the public transport company has to provide its service efficiently, according to a limited number of vehicles available. Thus the waiting times of passengers at interchanges, resulting from the schedule, will be a compromise between what is desirable and what is possible. In order to find the best compromise, certain quality parameters are defined that have to be taken into account by the journey optimisation and synchronisation process, and by journey planning engines when computing possible journeys. The entity DEFAULT INTERCHANGE is used to store such quality parameters. It is recorded against a SCHEDULED STOP POINT where passengers may leave a vehicle and a second SCHEDULED STOP POINT where they can proceed to enter a second vehicle. It may be used for any pair of SERVICE JOURNEYs serving respectively the first and the second SCHEDULED STOP POINT.

One of the stored parameters is the maximum duration that would be allowed for an interchange. If this maximum duration is exceeded for a pair of SERVICE JOURNEYs serving the two SCHEDULED STOP POINTs, this will no longer be regarded as a possible interchange between these journeys. In this situation, the scheduler will perhaps be warned by the system and be asked to retime these journeys; otherwise these journeys may not be advertised as connected at the specified SCHEDULED STOP POINTs. Another parameter is the standard duration that is aimed at, when planning an interchange. This should be taken into account as a reference value when timing the SERVICE JOURNEYs.

#### 7.2.7.1.2 Journey meetings and Interchanges

A JOURNEY MEETING describes the possibility to plan the schedules according to various interchange possibilities:

- Interchange with another service, of which only the arrival or departure time is known.
- More generally, service scheduled according to the time fixed for an external event, which will feed, or be fed by, this service (school, spectacle, etc.).
- Organisation of a meeting (hub) between several services, during a defined time band; this is a simplified specification of several interchanges. If needed this could be described in detail using several INTERCHANGE RULEs or SERVICE JOURNEY INTERCHANGEs.
- Specification of a rendez-vous (time and place) for any journey that can meet the appointment.

A JOURNEY MEETING may be related to one or several SERVICE JOURNEYs, which are planned according to this JOURNEY MEETING. It may be timed by an earliest time (e.g. the arrival time of a feeder line, plus the duration of a possible transfer) or by a latest time (e.g. the opening hour of the school served by the journey), or both (e.g. the time band of a hub).

A JOURNEY MEETING is located at one or several STOP POINTs, which shall be also classified as TIMING POINTs. It is planned in principle for VEHICLE JOURNEYs specified for the same DAY TYPE. The timing reference of these VEHICLE JOURNEYs will probably be chosen according to the JOURNEY MEETING specified.

In NeTEx consequences of any DEFAULT INTERCHANGE or JOURNEY MEETING used in the planning phase that needs to be exchanged should be expressed as the resulting SERVICE JOURNEY timings, INTERCHANGE RULEs and/or SERVICE JOURNEY INTERCHANGEs.

#### 7.2.7.1.3 Service Journey Pattern Interchange

Sometimes, the scheduled possibility to interchange needs to be differentiated, depending on the SERVICE JOURNEY PATTERNs involved. The SERVICE JOURNEY PATTERN INTERCHANGE specifies such an interchange between two particular SERVICE JOURNEY PATTERNs and a corresponding pair of SCHEDULED STOP POINTs, which shall be served by the respective JOURNEY PATTERNs. The quality level to be aimed at will certainly depend on the service frequency on each

SERVICE JOURNEY PATTERN. If these frequencies are more or less similar, the default values can be used as maximum and standard duration of interchange times. If the frequency is very low on one SERVICE JOURNEY PATTERN and considerably higher on another, the situation is different. An interchange waiting time of 13 minutes may be acceptable for passengers if the buses only run every two hours on a particular SERVICE JOURNEY PATTERN, but would not be a good solution if the buses run every 15 minutes.

The SERVICE JOURNEY PATTERN INTERCHANGE may be used to store some information on the planned interchange between two SERVICE JOURNEY PATTERNs (level of priority, advertised or not, guaranteed or not). It allows as well to store the quality parameters (maximum and standard duration) to be used for scheduling of SERVICE JOURNEYs, which will override any default parameter.

Thus the SERVICE JOURNEY PATTERN INTERCHANGE could in some instances be used during the planning phase to decide the actual timings of involved SERVICE JOURNEYs to be exchanged.

It is possible to exchange the SERVICE JOURNEY PATTERN INTERCHANGE through NeTEx using the INTERCHANGE RULE (see later chapter). Alternatively the consequences of the SERVICE JOURNEY PATTERN INTERCHANGE could be applied and exchanged as the resulting SERVICE JOURNEY INTERCHANGEs.

#### 7.2.7.1.3.1 Interchange Rule

There is also a close relation between SERVICE JOURNEY INTERCHANGE and INTERCHANGE RULE**.** In the physical model the SERVICE JOURNEY INTERCHANGE expresses an interchange between two specified SERVICE JOURNEYs**,** while the INTERCHANGE RULE (which is described in a later chapter) expresses interchanges where only one or none of the both SERVICE JOURNEYs are specified explicitly.

### 7.2.7.2 Interchange – Physical Model

The INTERCHANGE physical model describes SERVICE JOURNEY INTERCHANGEs and CONNECTION links and how they relate to the SERVICE JOURNEY and SCHEDULED STOP POINT. It includes detailed attributes:

- StaySeated expresses that the distributor and feeder SERVICE JOURNEYs are the same physical vehicle.
- Planned expresses that the interchange is planned.
- Advertised expresses that the interchange should be advertised to the public.
- Guaranteed expresses that it is guaranteed that the feeder SERVICE JOURNEY will wait for the distributor SERVICE JOURNEY.

#### 7.2.7.2.1 Use of Interchange times

**MaximumWaitTime** expresses the maximum duration a distributor SERVICE JOURNEY should delay its departure time waiting for a feeder **ServiceJourney**.

If specified, the **ExplictTransferTime** attribute of the SERVICE JOURNEY INTERCHANGE overrides the **DefaultDuration** for the CONNECTION. This value expresses the default duration allocated for a passenger to transfer, normally by walking, from the SCHEDULED STOP POINT of the feeder SERVICE JOURNEY to the SCHEDULED STOP POINT of the distributor SERVICE JOURNEY**.** The default transfer time of the passenger should be added to the feeder SERVICE JOURNEY forecasted arrival time to decide if the distributor Service Journey will exceed the **MaximumWaitTime** when waiting for passengers from the delayed distributor.

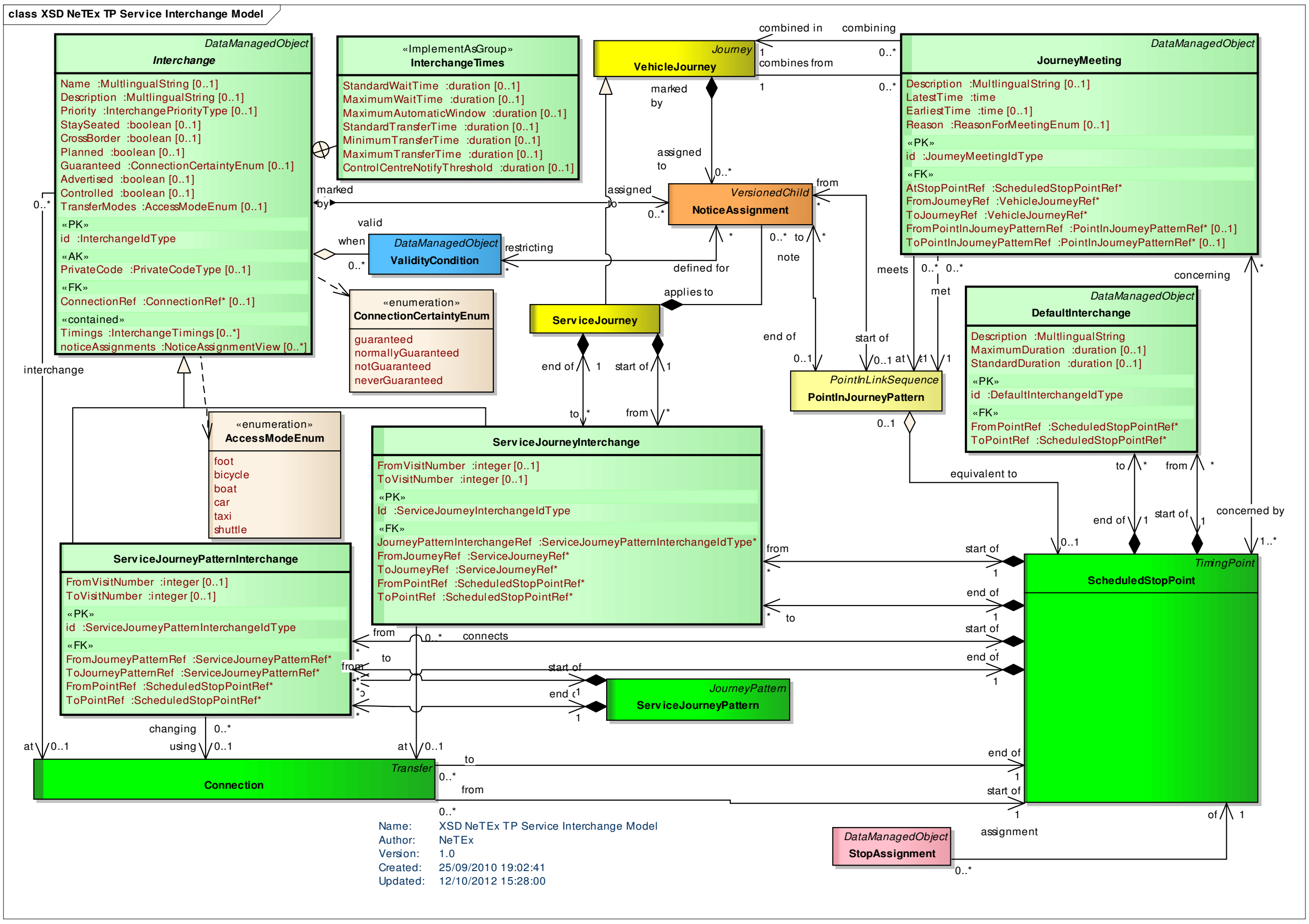

class XSD NeTEx TP Service Interchange Model
DataManagedObject
Interchange
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
Priority :InterchangePriorityType [0..1]
StaySeated :boolean [0..1]
CrossBorder :boolean [0..1]
Planned :boolean [0..1]
Guaranteed :ConnectionCertaintyEnum [0..1]
Advertised :boolean [0..1]
Controlled :boolean [0..1]
TransferModes :AccessModeEnum [0..1]
«PK»
id :InterchangeIdType
«AK»
PrivateCode :PrivateCodeType [0..1]
«FK»
ConnectionRef :ConnectionRef* [0..1]
«contained»
Timings :InterchangeTimings [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
«ImplementAsGroup»
InterchangeTimes
StandardWaitTime :duration [0..1]
MaximumWaitTime :duration [0..1]
MaximumAutomaticWindow :duration [0..1]
StandardTransferTime :duration [0..1]
MinimumTransferTime :duration [0..1]
MaximumTransferTime :duration [0..1]
ControlCentreNotifyThreshold :duration [0..1]
Journey
VehicleJourney
combined in
combining
combines from
marked by
assigned to
assigned
VersionedChild
NoticeAssignment
marked by
valid when
DataManagedObject
ValidityCondition
restricting
defined for
applies to
«enumeration»
ConnectionCertaintyEnum
guaranteed
normallyGuaranteed
notGuaranteed
neverGuaranteed
ServiceJourney
DataManagedObject
JourneyMeeting
Description :MultilingualString [0..1]
LatestTime :time
EarliestTime :time [0..1]
Reason :ReasonForMeetingEnum [0..1]
«PK»
id :JourneyMeetingIdType
«FK»
AtStopPointRef :ScheduledStopPointRef*
FromJourneyRef :VehicleJourneyRef*
ToJourneyRef :VehicleJourneyRef*
FromPointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
ToPointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
meets
met
concerning
DataManagedObject
DefaultInterchange
Description :MultilingualString
MaximumDuration :duration [0..1]
StandardDuration :duration [0..1]
«PK»
id :DefaultInterchangeIdType
«FK»
FromPointRef :ScheduledStopPointRef*
ToPointRef :ScheduledStopPointRef*
PointInLinkSequence
PointInJourneyPattern
equivalent to
interchange
«enumeration»
AccessModeEnum
foot
bicycle
boat
car
taxi
shuttle
ServiceJourneyInterchange
FromVisitNumber :integer [0..1]
ToVisitNumber :integer [0..1]
«PK»
Id :ServiceJourneyInterchangeIdType
«FK»
JourneyPatternInterchangeRef :ServiceJourneyPatternInterchangeIdType*
FromJourneyRef :ServiceJourneyRef*
ToJourneyRef :ServiceJourneyRef*
FromPointRef :ScheduledStopPointRef*
ToPointRef :ScheduledStopPointRef*
ServiceJourneyPatternInterchange
FromVisitNumber :integer [0..1]
ToVisitNumber :integer [0..1]
«PK»
id :ServiceJourneyPatternInterchangeIdType
«FK»
FromJourneyPatternRef :ServiceJourneyPatternRef*
ToJourneyPatternRef :ServiceJourneyPatternRef*
FromPointRef :ScheduledStopPointRef*
ToPointRef :ScheduledStopPointRef*
connects
JourneyPattern
ServiceJourneyPattern
TimingPoint
ScheduledStopPoint
changing
using
Transfer
Connection
DataManagedObject
StopAssignment
assignment
concerned by
Name: XSD NeTEx TP Service Interchange Model
Author: NeTEx
Version: 1.0
Created: 25/09/2010 19:02:41
Updated: 12/10/2012 15:28:00

**Figure 75 — Service Interchange – Physical Model (UML)**

### 7.2.7.3 Interchange – Attributes and XSD

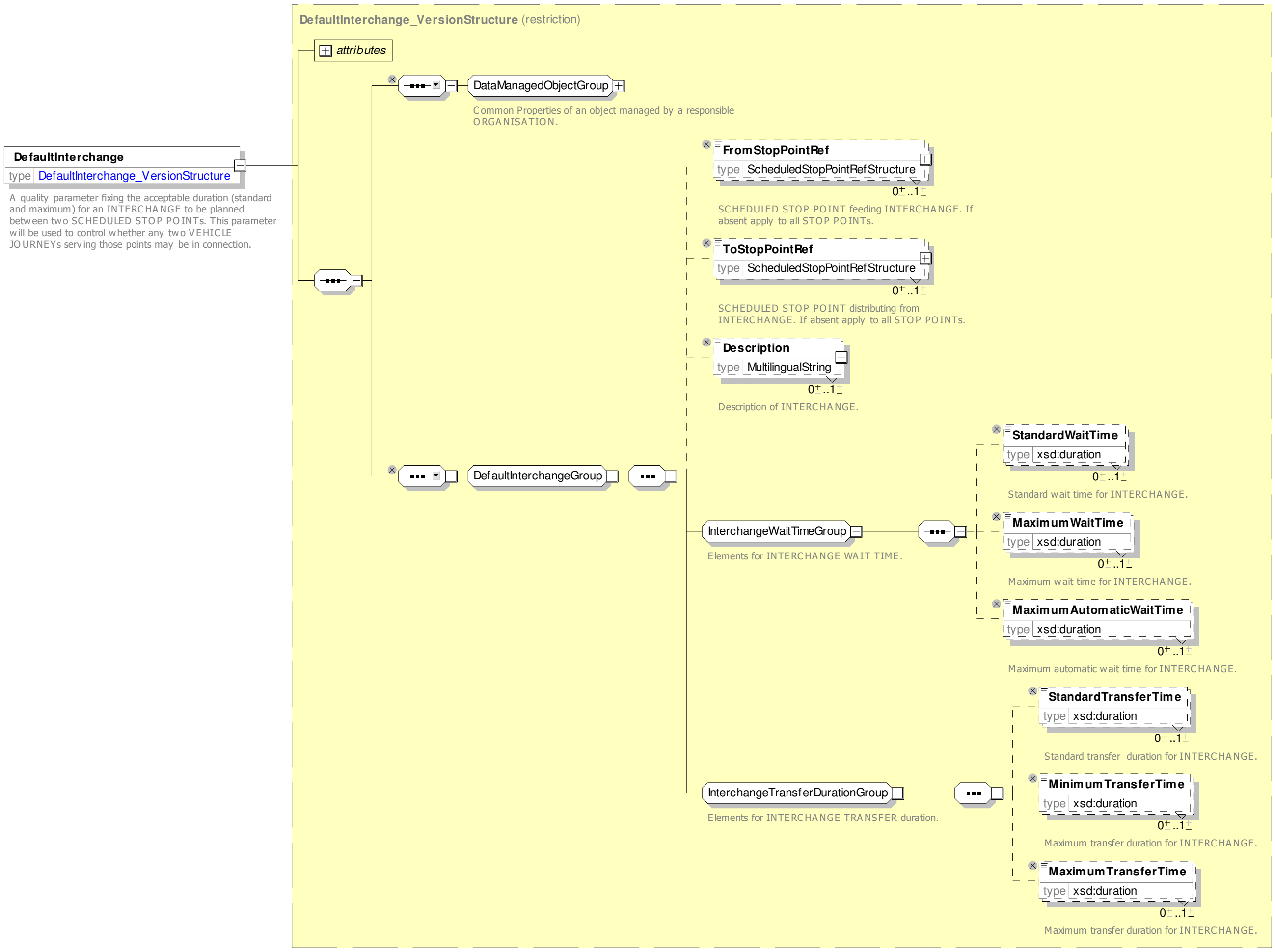
### 7.2.7.3.1 DefaultInterchange – Model Element

A quality parameter fixing the acceptable duration (standard and maximum) for an interchange to be planned between two STOP POINTs. This parameter will be used to control whether any two VEHICLE JOURNEYs serving those points may be in connection.

**Table 44 – *DefaultInterchange* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | DEFAULT INTERCHANGE inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *DefaultInterchangeIdType* | 1:1 | Identifier of DEFAULT INTERCHANGE. |
| | ***Description*** | *MultilingualString* | 1:1 | Description of JOURNEY MEETING. |
| | ***MaximumDuration*** | *xsd:duration* | 0:1 | Maximum wait time for DEFAULT INTERCHANGE. |
| | ***StandardDuration*** | *xsd:duration* | 0:1 | Standard wait time for DEFAULT INTERCHANGE. |
| «FK» | ***FromPointRef*** | *ScheduledStopPointRef* | 1:1 | SCHEDULED STOP POINT from which the |

| | | | | |
|---|---|---|---|---|
| | | | | DEFAULT INTERCHANGE starts. |
| «FK» | ***ToPointRef*** | *ScheduledStopPointRef* | 1:1 | SCHEDULED STOP POINT at which the DEFAULT INTERCHANGE ends. |



DefaultInterchange_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
DefaultInterchange
type DefaultInterchange_VersionStructure
A quality parameter fixing the acceptable duration (standard and maximum) for an INTERCHANGE to be planned between two SCHEDULED STOP POINTs. This parameter will be used to control whether any two VEHICLE JOURNEYs serving those points may be in connection.
FromStopPointRef
type ScheduledStopPointRefStructure
0..1
SCHEDULED STOP POINT feeding INTERCHANGE. If absent apply to all STOP POINTs.
ToStopPointRef
type ScheduledStopPointRefStructure
0..1
SCHEDULED STOP POINT distributing from INTERCHANGE. If absent apply to all STOP POINTs.
Description
type MultilingualString
0..1
Description of INTERCHANGE.
DefaultInterchangeGroup
StandardWaitTime
type xsd:duration
0..1
Standard wait time for INTERCHANGE.
InterchangeWaitTimeGroup
Elements for INTERCHANGE WAIT TIME.
MaximumWaitTime
type xsd:duration
0..1
Maximum wait time for INTERCHANGE.
MaximumAutomaticWaitTime
type xsd:duration
0..1
Maximum automatic wait time for INTERCHANGE.
StandardTransferTime
type xsd:duration
0..1
Standard transfer duration for INTERCHANGE.
InterchangeTransferDurationGroup
Elements for INTERCHANGE TRANSFER duration.
MinimumTransferTime
type xsd:duration
0..1
Maximum transfer duration for INTERCHANGE.
MaximumTransferTime
type xsd:duration
0..1
Maximum transfer duration for INTERCHANGE.

Figure 76 — *DefaultInterchange* — XSD

### 7.2.7.3.2 Interchange – Model Element

The scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs.

Table 45 – *Interchange* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | INTERCHANGE inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *InterchangeIdType* | 1:1 | Identifier of INTERCHANGE. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of INTERCHANGE. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of INTERCHANGE. |

| | | | | |
|---|---|---|---|---|
| «AK» | ***PrivateCode*** | *xsd:normalizedString* | 0:1 | Alternative key for INTERCHANGE. |
| «AK» | ***External-InterchangeRef*** | *ExternalObjectRef* | 0:1 | An alternative code that uniquely identifies the INTERCHANGE specifically for use in AVMS systems.<br>*NOTE For VDV compatibility.* |
| «FK» | ***ConnectionRef*** | *ConnectionRef* | 0:1 | CONNECTION link over which the interchange takes place. |
| | ***Priority*** | *InterchangePriorityType* | 0:1 | Priority assigned to INTERCHANGE. |
| | ***StaySeated*** | *xsd:boolean* | 0:1 | Whether passengers can stay seated to make INTERCHANGE. |
| | ***CrossBorder*** | *xsd:boolean* | 0:1 | Whether INTERCHANGE involves crossing a national border. |
| | ***Planned*** | *xsd:boolean* | 0:1 | Whether INTERCHANGE is planned. |
| | ***Guaranteed*** | *ConnectionCertaintyEnum* | 0:1 | Whether INTERCHANGE is guaranteed, that is distributor services may be held in order to ensure the connection. |
| | ***Advertised*** | *xsd:boolean* | 0:1 | Whether INTERCHANGE is advertised to the public. |
| | ***Controlled*** | *xsd:boolean* | 0:1 | Whether INTERCHANGE is controlled. |
| «cntd» | ***Interchange-TimesGroup*** | *InterchangeTimesGroup* | 0:* | Timings that apply to the INTERCHANGE. |
| | ***TransferModes*** | *AccessModeEnum* | 0:1 | Out of vehicle TRANSPORT MODEs by which transfer at the interchange can be made. See Reusable components. |
| «cntd» | ***notice-Assignments*** | *NoticeAssignmentView* | 0:* | NOTICE ASSIGNMENTs that apply to the INTERCHANGE. |

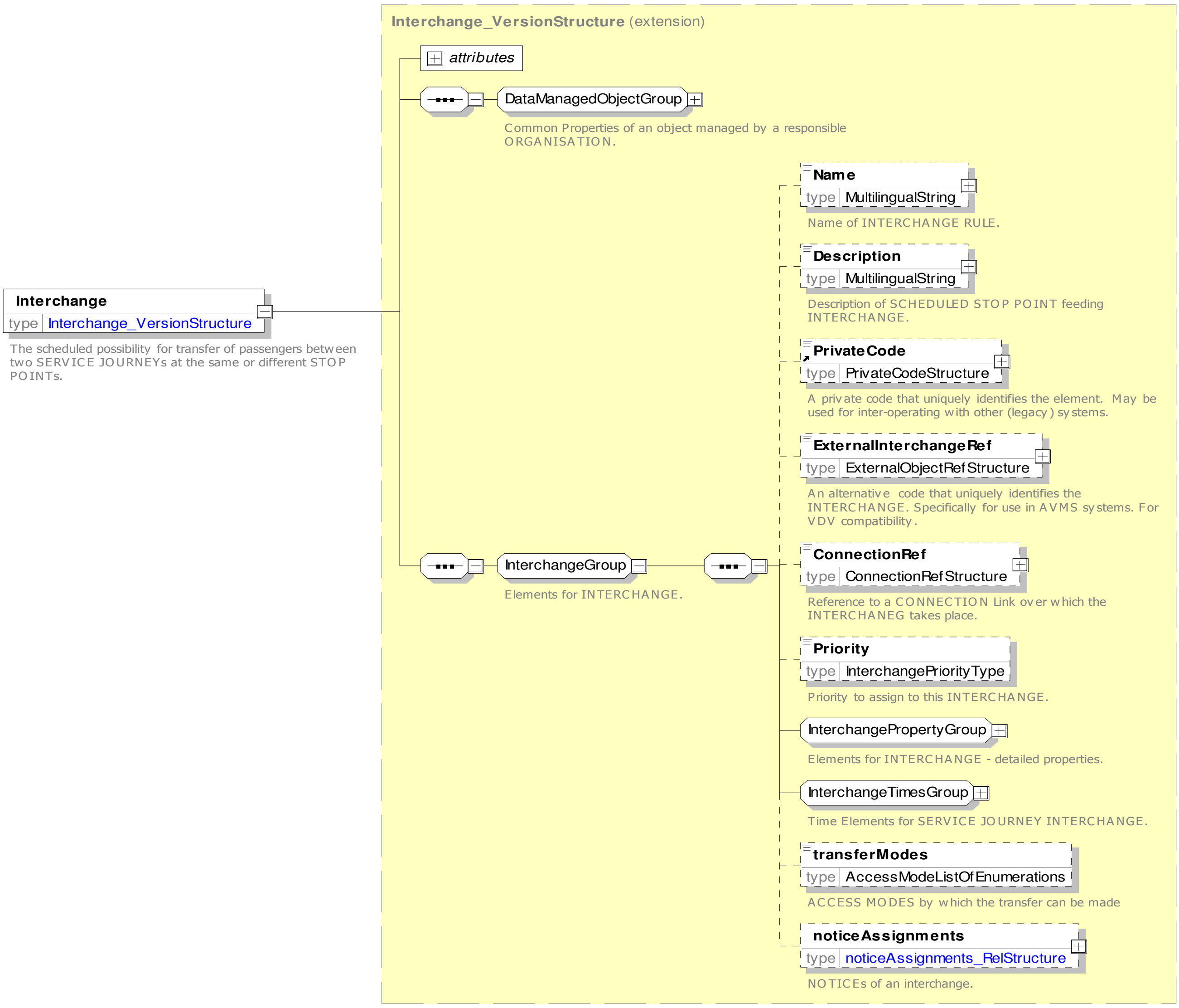

Interchange_VersionStructure (extension)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
Name
type MultilingualString
Name of INTERCHANGE RULE.
Description
type MultilingualString
Description of SCHEDULED STOP POINT feeding INTERCHANGE.
PrivateCode
type PrivateCodeStructure
A private code that uniquely identifies the element. May be used for inter-operating with other (legacy) systems.
ExternalInterchangeRef
type ExternalObjectRefStructure
An alternative code that uniquely identifies the INTERCHANGE. Specifically for use in AVMS systems. For VDV compatibility.
ConnectionRef
type ConnectionRefStructure
Reference to a CONNECTION Link over which the INTERCHANEG takes place.
Priority
type InterchangePriorityType
Priority to assign to this INTERCHANGE.
InterchangePropertyGroup
Elements for INTERCHANGE - detailed properties.
InterchangeTimesGroup
Time Elements for SERVICE JOURNEY INTERCHANGE.
transferModes
type AccessModeListOfEnumerations
ACCESS MODES by which the transfer can be made
noticeAssignments
type noticeAssignments_RelStructure
NOTICEs of an interchange.
InterchangeGroup
Elements for INTERCHANGE.
Interchange
type Interchange_VersionStructure
The scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs.

**Figure 77 — *Interchange* — XSD**

## 7.2.7.3.2.1 *InterchangePropertyGroup* — Group

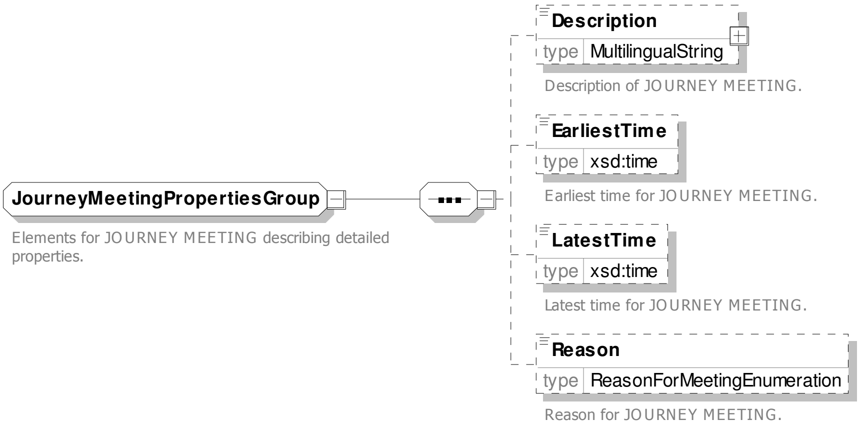

Description
type MultilingualString
Description of JOURNEY MEETING.
EarliestTime
type xsd:time
Earliest time for JOURNEY MEETING.
LatestTime
type xsd:time
Latest time for JOURNEY MEETING.
Reason
type ReasonForMeetingEnumeration
Reason for JOURNEY MEETING.
JourneyMeetingPropertiesGroup
Elements for JOURNEY MEETING describing detailed properties.

**Figure 78 — *InterchangePropertyGroup* — XSD**

Table 46 – *InterchangeTimesGroup* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| | ***StandardWaitTime*** | *xsd:duration* | 0:1 | Standard Time to wait at Interchange. |
| | ***MaximumWaitTime*** | *xsd:duration* | 0:1 | Maximum time that DISTRIBUTOR will wait after its planned departure time. |
| | ***MaximumAutomatic-Window*** | *xsd:duration* | 0:1 | Maximum window for holding DISTRIBUTOR will wait. |
| | ***StandardTransferTime*** | *xsd:duration* | 0:1 | Standard Time needed for transfer. |
| | ***MinimumTransferTime*** | *xsd:duration* | 0:1 | Minimum Time needed for transfer. |
| | ***MaximumTransferTime*** | *xsd:duration* | 0:1 | Maximum Time needed for transfer. |
| | ***ControlCentre-NotifyThreshold*** | *xsd:duration* | 0:1 | Interval before CONTROL CENTRE should be notified associated with SERVICE JOURNEY INTERCHANGE. |

### 7.2.7.3.2.2 *InterchangeTimesGroup* — Group

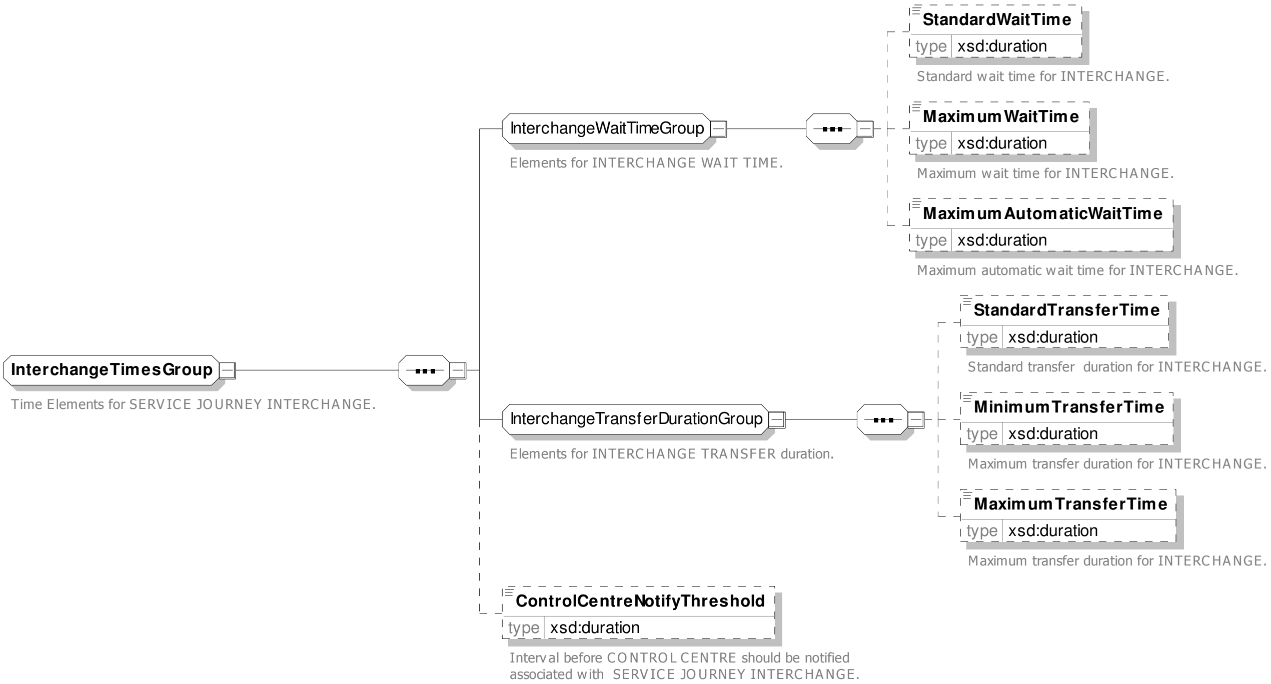

StandardWaitTime
type xsd:duration
Standard wait time for INTERCHANGE.
MaximumWaitTime
type xsd:duration
Maximum wait time for INTERCHANGE.
MaximumAutomaticWaitTime
type xsd:duration
Maximum automatic wait time for INTERCHANGE.
InterchangeWaitTimeGroup
Elements for INTERCHANGE WAIT TIME.
InterchangeTimesGroup
Time Elements for SERVICE JOURNEY INTERCHANGE.
StandardTransferTime
type xsd:duration
Standard transfer duration for INTERCHANGE.
InterchangeTransferDurationGroup
Elements for INTERCHANGE TRANSFER duration.
MinimumTransferTime
type xsd:duration
Maximum transfer duration for INTERCHANGE.
MaximumTransferTime
type xsd:duration
Maximum transfer duration for INTERCHANGE.
ControlCentreNotifyThreshold
type xsd:duration
Interval before CONTROL CENTRE should be notified associated with SERVICE JOURNEY INTERCHANGE.

Figure 79 — *InterchangeTimesGroup* — XSD

### 7.2.7.3.2.3 ConnectionCertainty – Allowed values

The following table shows the allowed values for ***ConnectionCertainty*** ***(****ConnectionCertaintyEnum)*.

**Table 47 – *ConnectionCertainty* – Allowed values**

| Value | Description |
|---|---|
| *guaranteed* | The connection is guaranteed under any circumstances. |
| *normallyGuaranteed* | The connection is normally guaranteed, although the connection time available is shorter than the location connection time. |
| *notGuaranteed* | The connection is not guaranteed, although the connection time available is longer than the location connection time. |
| *neverGuaranteed* | The connection is never guaranteed, although the connection time available is longer than the location connection time |

#### 7.2.7.3.3 ServiceJourneyInterchange – Model Element

The scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs.

**Table 48 – *ServiceJourneyInterchange* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *Interchange* | *::>* | SERVICE JOURNEY INTERCHANGE inherits from INTERCHANGE. |
| «PK» | id | *ServiceJourney-InterchangeIdType* | 1:1 | Identifier of SERVICE JOURNEY INTERCHANGE. |
| «FK» | ***FromPointRef*** | *ScheduledStopPointRef* | 1:1 | SCHEDULED STOP POINT from which the interchange starts. |
|  | ***FromVisitNumber*** | *xsd:integer* | 0:1 | Visit number of feeder journey (only needed if multiple visits). |
| «FK» | ***ToPointRef*** | *ScheduledStopPointRef* | 1:1 | SCHEDULED STOP POINT to which the interchange goes. |
|  | ***ToVisitNumber*** | *xsd:integer* | 0:1 | Visit number of distributor journey (only needed if multiple visits). |
| «FK» | ***FromJourneyRef*** | *ServiceJourneyRef* | 1:1 | Reference to feeder SERVICE JOURNEY from which interchange is made. |
| «FK» | ***ToJourneyRef*** | *ServiceJourneyRef* | 1:1 | Reference to distributor SERVICE JOURNEY to which interchange is made. |
| «FK» | ***ServiceJourneyPattern-InterchangeRef*** | *ServiceJourneyPattern-InterchangeRef* | 1:1 | Reference to SERVICE JOURNEY PATTERN INTERCHANGE for which interchange is made. |

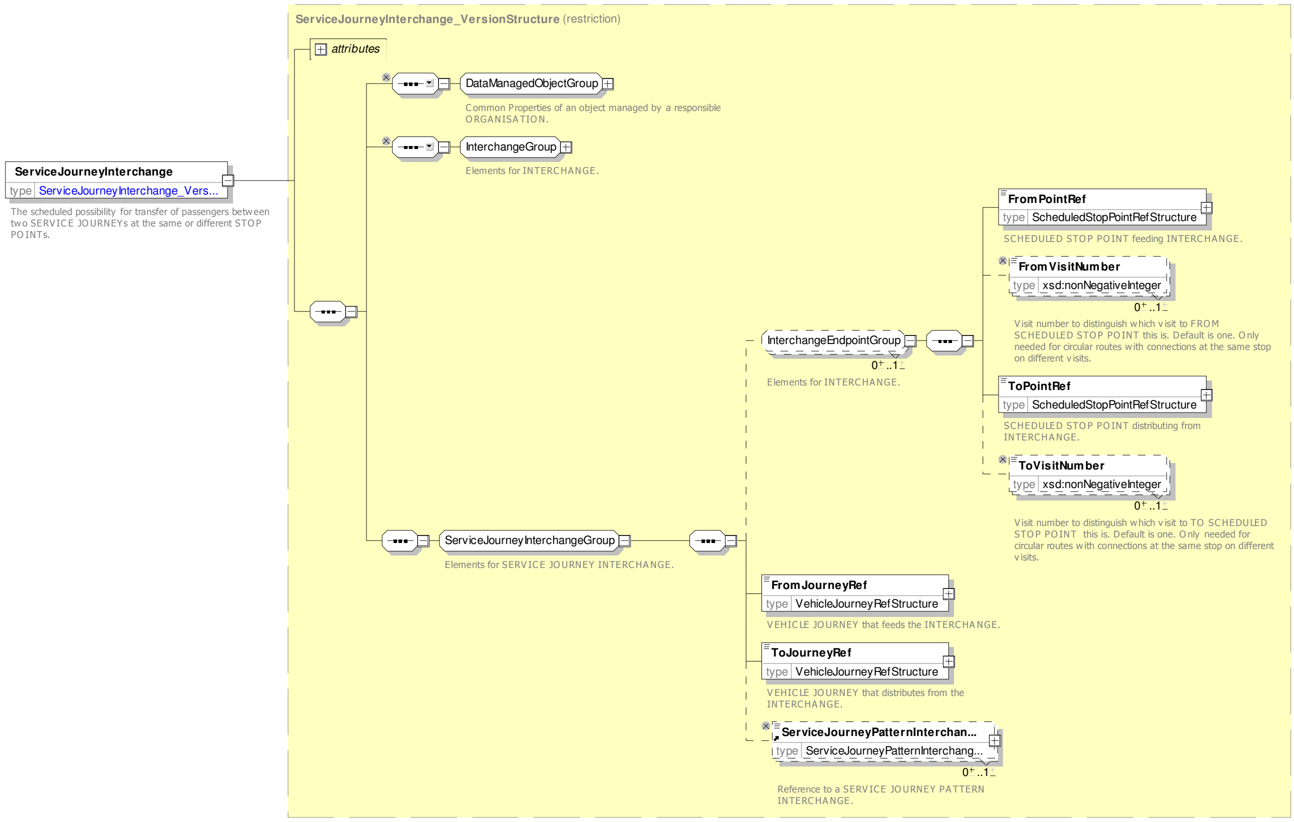

ServiceJourneyInterchange_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
InterchangeGroup
Elements for INTERCHANGE.
ServiceJourneyInterchange
type ServiceJourneyInterchange_Vers...
The scheduled possibility for transfer of passengers between two SERVICE JOURNEYs at the same or different STOP POINTs.
FromPointRef
type ScheduledStopPointRefStructure
SCHEDULED STOP POINT feeding INTERCHANGE.
FromVisitNumber
type xsd:nonNegativeInteger
0..1
Visit number to distinguish which visit to FROM SCHEDULED STOP POINT this is. Default is one. Only needed for circular routes with connections at the same stop on different visits.
InterchangeEndpointGroup
0..1
Elements for INTERCHANGE.
ToPointRef
type ScheduledStopPointRefStructure
SCHEDULED STOP POINT distributing from INTERCHANGE.
ToVisitNumber
type xsd:nonNegativeInteger
0..1
Visit number to distinguish which visit to TO SCHEDULED STOP POINT this is. Default is one. Only needed for circular routes with connections at the same stop on different visits.
ServiceJourneyInterchangeGroup
Elements for SERVICE JOURNEY INTERCHANGE.
FromJourneyRef
type VehicleJourneyRefStructure
VEHICLE JOURNEY that feeds the INTERCHANGE.
ToJourneyRef
type VehicleJourneyRefStructure
VEHICLE JOURNEY that distributes from the INTERCHANGE.
ServiceJourneyPatternInterchan...
type ServiceJourneyPatternInterchang...
0..1
Reference to a SERVICE JOURNEY PATTERN INTERCHANGE.

Figure 80 — *ServiceJourneyInterchange* — XSD

### 7.2.7.3.4 ServiceJourneyPatternInterchange – Model Element

A recognised/organised possibility for passengers to change public transport vehicles using two STOP POINTs (which may be identical) on two particular SERVICE JOURNEY PATTERNs, including the maximum wait duration allowed and the standard to be aimed at. These may supersede the times given for the DEFAULT INTERCHANGE. Schedulers may use this entity for synchronisation of journeys.

Table 49 – *ServiceJourneyPatternInterchange* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *Interchange* | ::> | SERVICE JOURNEY PATTERN INTERCHANGE inherits from INTERCHANGE. |
| «PK» | ***id*** | *ServiceJourneyPattern-InterchangeIdType* | 1:1 | Identifier of SERVICE JOURNEY PATTERN INTERCHANGE. |
| | ***MaximumDuration*** | *xsd:duration* | 0:1 | Maximum duration of a transfer for SERVICE JOURNEY PATTERN INTERCHANGE. |
| | ***StandardDuration*** | *xsd:duration* | 0:1 | Standard duration of a transfer for SERVICE JOURNEY PATTERN 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» | ***FromJourney-PatternRef*** | *ServiceJourneyPatternRef* | 1:1 | Feeder SERVICE JOURNEY PATTERN from which SERVICE JOURNEY PATTERN INTERCHANGE is made. |
| «FK» | ***ToJourneyPattern Ref*** | *ServiceJourneyPatternRef* | 1:1 | Distributor SERVICE JOURNEY PATTERN to which SERVICE JOURNEY PATTERN INTERCHANGE is made. |

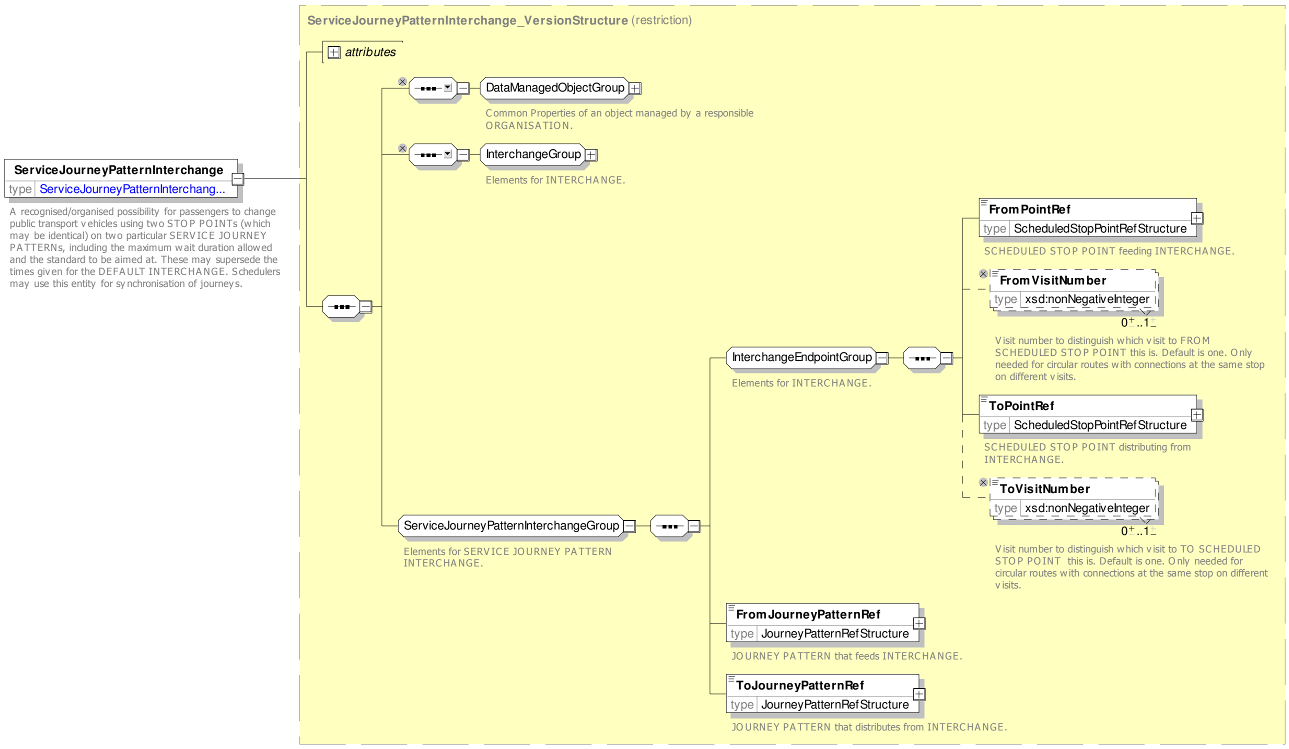

ServiceJourneyPatternInterchange_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
InterchangeGroup
Elements for INTERCHANGE.
ServiceJourneyPatternInterchange
type ServiceJourneyPatternInterchang...
A recognised/organised possibility for passengers to change public transport vehicles using two STOP POINTs (which may be identical) on two particular SERVICE JOURNEY PATTERNs, including the maximum wait duration allowed and the standard to be aimed at. These may supersede the times given for the DEFAULT INTERCHANGE. Schedulers may use this entity for synchronisation of journeys.
FromPointRef
type ScheduledStopPointRefStructure
SCHEDULED STOP POINT feeding INTERCHANGE.
FromVisitNumber
type xsd:nonNegativeInteger
0..1
Visit number to distinguish which visit to FROM SCHEDULED STOP POINT this is. Default is one. Only needed for circular routes with connections at the same stop on different visits.
InterchangeEndpointGroup
Elements for INTERCHANGE.
ToPointRef
type ScheduledStopPointRefStructure
SCHEDULED STOP POINT distributing from INTERCHANGE.
ToVisitNumber
type xsd:nonNegativeInteger
0..1
Visit number to distinguish which visit to TO SCHEDULED STOP POINT this is. Default is one. Only needed for circular routes with connections at the same stop on different visits.
ServiceJourneyPatternInterchangeGroup
Elements for SERVICE JOURNEY PATTERN INTERCHANGE.
FromJourneyPatternRef
type JourneyPatternRefStructure
JOURNEY PATTERN that feeds INTERCHANGE.
ToJourneyPatternRef
type JourneyPatternRefStructure
JOURNEY PATTERN that distributes from INTERCHANGE.

**Figure 81 — *ServiceJourneyPatternInterchange* — XSD**

### 7.2.7.3.5 JourneyMeeting – Model Element

A time constraint for one or several SERVICE JOURNEYs fixing interchanges between them and/or an external event (e.g. arrival or departure of a feeder line, opening time of the theatre, etc.).

**Table 50 – *JourneyMeeting* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | JOURNEY MEETING inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *JourneyMeetingIdType* | 1:1 | Identifier of JOURNEY MEETING. |
| «FK» | ***AtPointRef*** | *ScheduledStopPointRef* | 1:1 | POINT at which MEETING can take place. |

| | | | | |
|---|---|---|---|---|
| «FK» | ***FromJourneyRef*** | *VehicleJourneyRef* | 1:1 | Inbound feeder JOURNEY that meets. |
| «FK» | ***ToJourneyRef*** | *VehicleJourneyRef* | 1:1 | Outbound distributor JOURNEY that meets. |
| | ***Description*** | *MultiLingualString* | 0:1 | Description of JOURNEY MEETING. |
| | ***EarliestTime*** | *xsd:time* | 0:1 | Latest time at which MEETING can take place. |
| | ***LatestTime*** | *xsd:time* | 1:1 | Earliest time at which MEETING can take place. |
| | ***Reason*** | *ReasonForMeetingEnum* | 0:1 | Reason for JOURNEY MEETING. |
| «FK» | ***FromPointIn-JourneyPatternRef*** | *PointInJourneyPatternRef* | 0:1 | POINT IN JOURNEY PATTERN of Inbound feeder JOURNEY that meets. |
| «FK» | ***ToPointInJourneyPatternRef*** | *PointInJourneyPatternRef* | 0:1 | POINT IN JOURNEY PATTERN of outbound distributor JOURNEY that meets. |

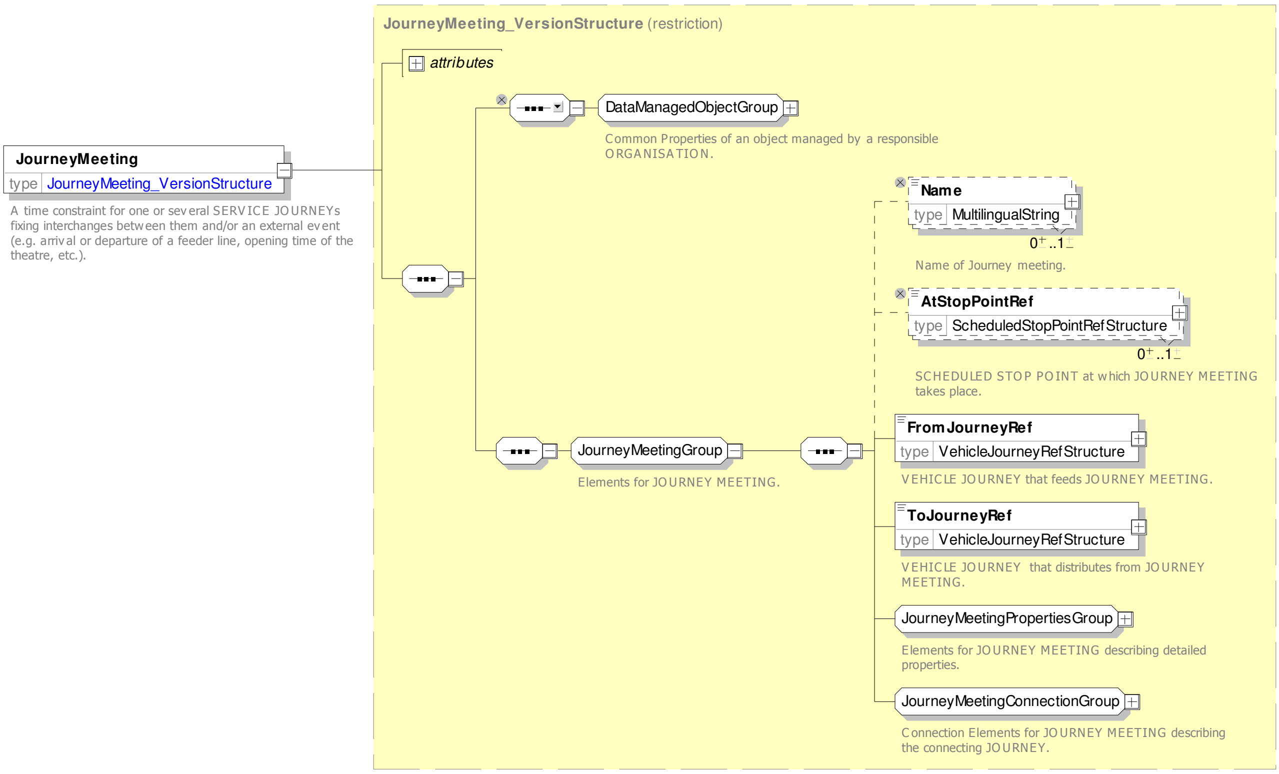

JourneyMeeting_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
JourneyMeeting
type JourneyMeeting_VersionStructure
A time constraint for one or several SERVICE JOURNEYs fixing interchanges between them and/or an external event (e.g. arrival or departure of a feeder line, opening time of the theatre, etc.).
Name
type MultilingualString
0..1
Name of Journey meeting.
AtStopPointRef
type ScheduledStopPointRefStructure
0..1
SCHEDULED STOP POINT at which JOURNEY MEETING takes place.
JourneyMeetingGroup
Elements for JOURNEY MEETING.
FromJourneyRef
type VehicleJourneyRefStructure
VEHICLE JOURNEY that feeds JOURNEY MEETING.
ToJourneyRef
type VehicleJourneyRefStructure
VEHICLE JOURNEY that distributes from JOURNEY MEETING.
JourneyMeetingPropertiesGroup
Elements for JOURNEY MEETING describing detailed properties.
JourneyMeetingConnectionGroup
Connection Elements for JOURNEY MEETING describing the connecting JOURNEY.

Figure 82 — *JourneyMeeting* — XSD

#### 7.2.7.3.5.1 *JourneyMeetingPropertiesGroup* — Group

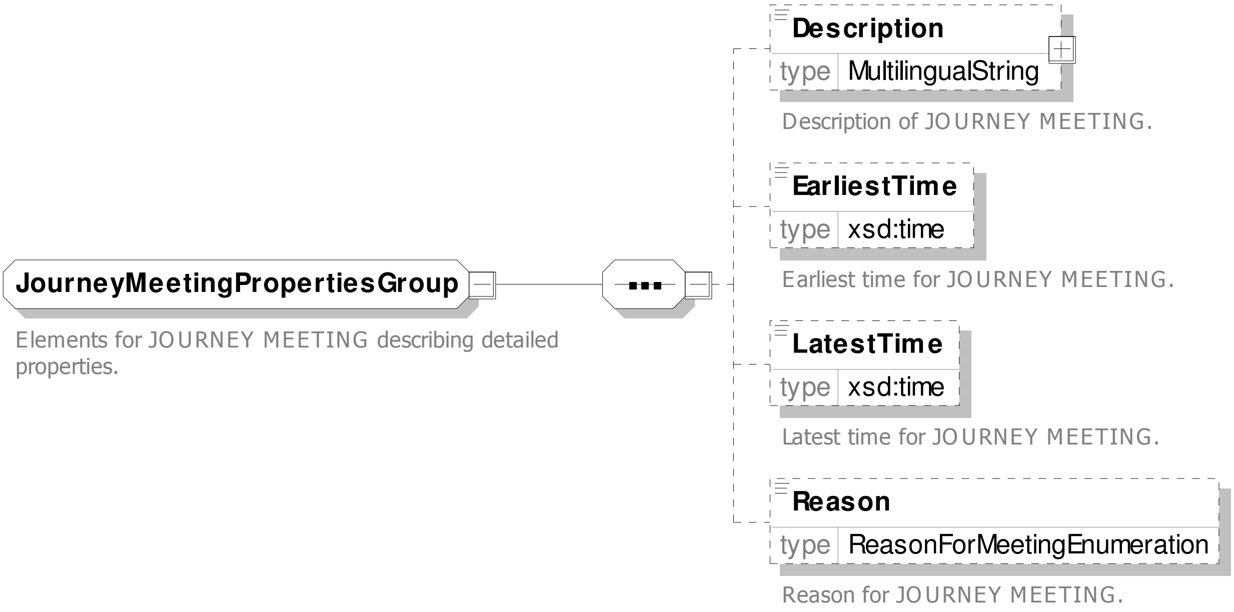

JourneyMeetingPropertiesGroup
Elements for JOURNEY MEETING describing detailed properties.
Description
type MultilingualString
Description of JOURNEY MEETING.
EarliestTime
type xsd:time
Earliest time for JOURNEY MEETING.
LatestTime
type xsd:time
Latest time for JOURNEY MEETING.
Reason
type ReasonForMeetingEnumeration
Reason for JOURNEY MEETING.

**Figure 83 — *JourneyMeetingPropertiesGroup* — XSD**

### 7.2.7.3.5.2 *JourneyMeetingConnectionGroup*— Group

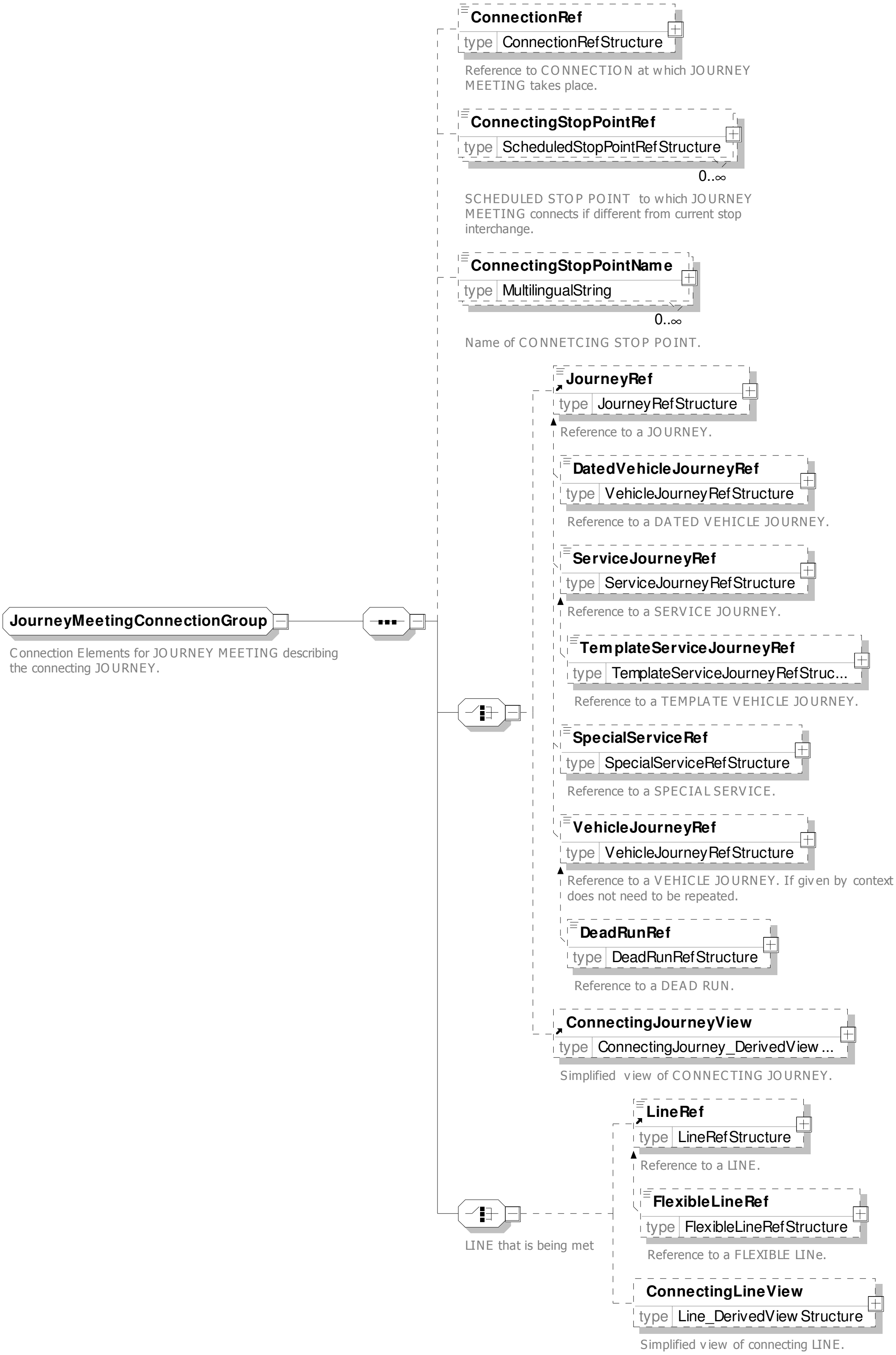

ConnectionRef
type ConnectionRefStructure
Reference to CONNECTION at which JOURNEY MEETING takes place.
ConnectingStopPointRef
type ScheduledStopPointRefStructure
0..∞
SCHEDULED STOP POINT to which JOURNEY MEETING connects if different from current stop interchange.
ConnectingStopPointName
type MultilingualString
0..∞
Name of CONNETCING STOP POINT.
JourneyRef
type JourneyRefStructure
Reference to a JOURNEY.
DatedVehicleJourneyRef
type VehicleJourneyRefStructure
Reference to a DATED VEHICLE JOURNEY.
ServiceJourneyRef
type ServiceJourneyRefStructure
Reference to a SERVICE JOURNEY.
JourneyMeetingConnectionGroup
Connection Elements for JOURNEY MEETING describing the connecting JOURNEY.
TemplateServiceJourneyRef
type TemplateServiceJourneyRefStruc...
Reference to a TEMPLATE VEHICLE JOURNEY.
SpecialServiceRef
type SpecialServiceRefStructure
Reference to a SPECIAL SERVICE.
VehicleJourneyRef
type VehicleJourneyRefStructure
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.
ConnectingJourneyView
type ConnectingJourney_DerivedView ...
Simplified view of CONNECTING JOURNEY.
LineRef
type LineRefStructure
Reference to a LINE.
FlexibleLineRef
type FlexibleLineRefStructure
Reference to a FLEXIBLE LINe.
LINE that is being met
ConnectingLineView
type Line_DerivedView Structure
Simplified view of connecting LINE.

**Figure 84 — *JourneyMeetingConnectionGroup*— XSD**

#### 7.2.7.3.5.3 ReasonForMeeting – Allowed values

The following table shows the allowed values for ***ReasonForMeeting*** *(ReasonForMeetingEnum).*

**Table 51 – *ReasonForMeeting* – Allowed values**

| Value | Description |
|---|---|
| *joining* | Meeting is joining of two services. |
| *splitting* | Meeting is splitting of two services. |
| *tariffSection* | Meeting is for start of a Tariff Section |
| *serviceFacility* | Meeting is for start of service Facilities. |

### 7.2.7.3.6 JourneyMeetingView – Model Element

A ***JourneyMeetingView*** provides a view of a JOURNEY MEETING including derived attributes. See JOURNEY MEETING and INTERCHANGE for a description of attributes.

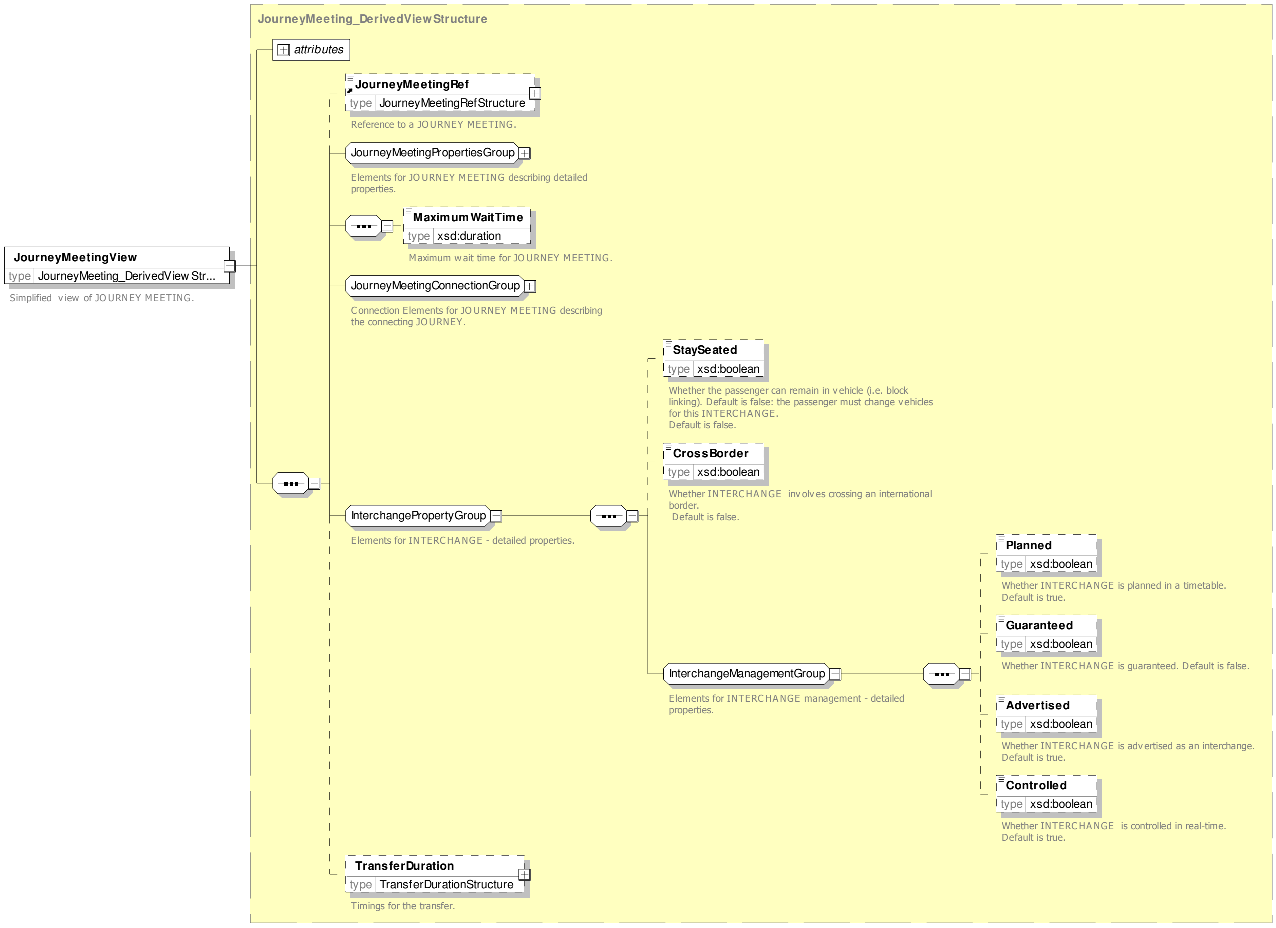

JourneyMeeting_DerivedViewStructure
attributes
JourneyMeetingRef
type JourneyMeetingRefStructure
Reference to a JOURNEY MEETING.
JourneyMeetingPropertiesGroup
Elements for JOURNEY MEETING describing detailed properties.
MaximumWaitTime
type xsd:duration
Maximum wait time for JOURNEY MEETING.
JourneyMeetingView
type JourneyMeeting_DerivedView Str...
Simplified view of JOURNEY MEETING.
JourneyMeetingConnectionGroup
Connection Elements for JOURNEY MEETING describing the connecting JOURNEY.
StaySeated
type xsd:boolean
Whether the passenger can remain in vehicle (i.e. block linking). Default is false: the passenger must change vehicles for this INTERCHANGE. Default is false.
CrossBorder
type xsd:boolean
Whether INTERCHANGE involves crossing an international border. Default is false.
InterchangePropertyGroup
Elements for INTERCHANGE - detailed properties.
Planned
type xsd:boolean
Whether INTERCHANGE is planned in a timetable. Default is true.
Guaranteed
type xsd:boolean
Whether INTERCHANGE is guaranteed. Default is false.
InterchangeManagementGroup
Elements for INTERCHANGE management - detailed properties.
Advertised
type xsd:boolean
Whether INTERCHANGE is advertised as an interchange. Default is true.
Controlled
type xsd:boolean
Whether INTERCHANGE is controlled in real-time. Default is true.
TransferDuration
type TransferDurationStructure
Timings for the transfer.

**Figure 85 — *JourneyMeetingView* — XSD**

### 7.2.7.3.7 ConnectingJourneyView – Model Element

***ConnectingJourneyView*** provides a view of a CONNECTING JOURNEY including derived attributes.

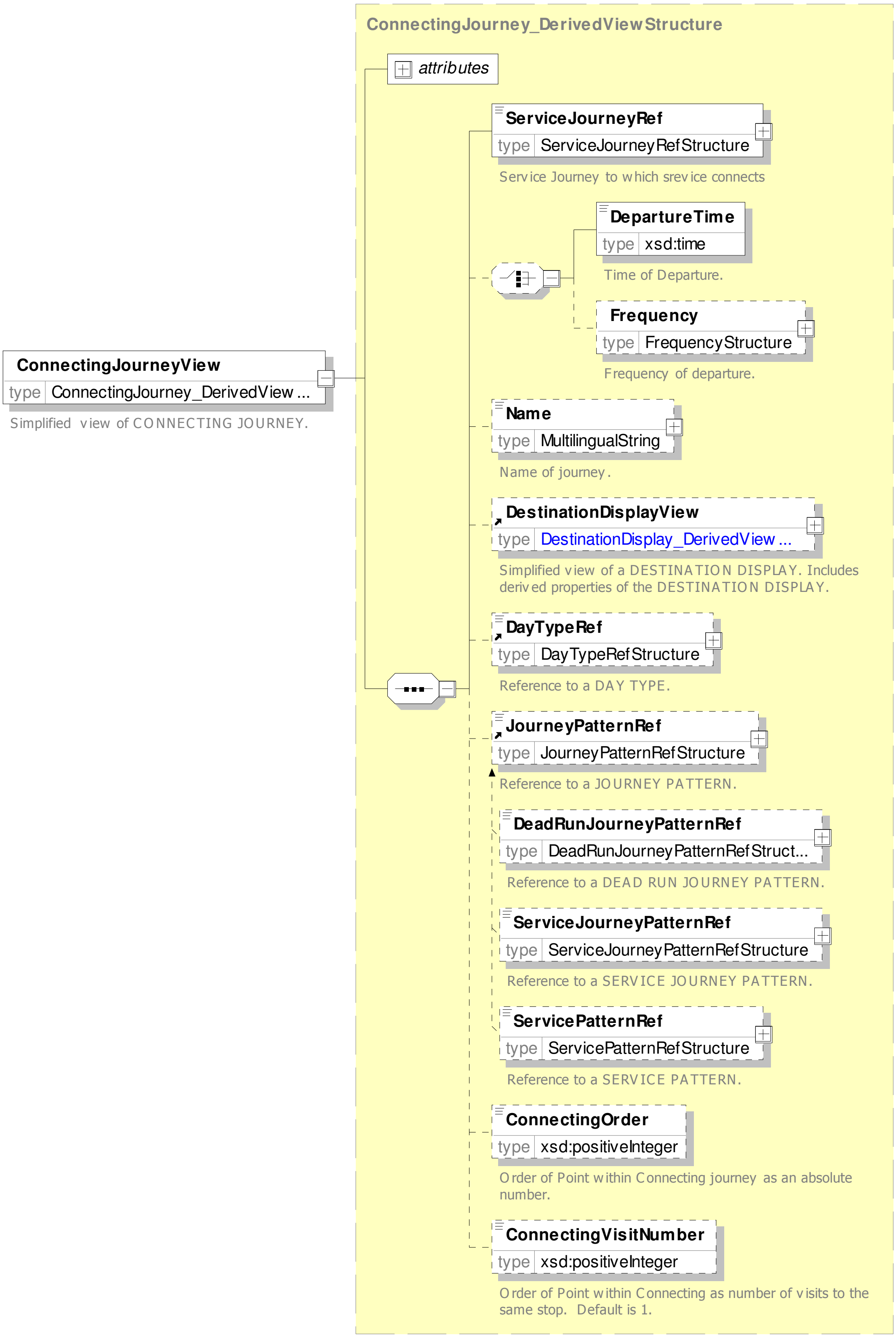

ConnectingJourney_DerivedViewStructure
attributes
ServiceJourneyRef
type ServiceJourneyRefStructure
Service Journey to which srevice connects
DepartureTime
type xsd:time
Time of Departure.
Frequency
type FrequencyStructure
Frequency of departure.
ConnectingJourneyView
type ConnectingJourney_DerivedView ...
Simplified view of CONNECTING JOURNEY.
Name
type MultilingualString
Name of journey.
DestinationDisplayView
type DestinationDisplay_DerivedView ...
Simplified view of a DESTINATION DISPLAY. Includes derived properties of the DESTINATION DISPLAY.
DayTypeRef
type DayTypeRefStructure
Reference to a DAY TYPE.
JourneyPatternRef
type JourneyPatternRefStructure
Reference to a JOURNEY PATTERN.
DeadRunJourneyPatternRef
type DeadRunJourneyPatternRefStruct...
Reference to a DEAD RUN JOURNEY PATTERN.
ServiceJourneyPatternRef
type ServiceJourneyPatternRefStructure
Reference to a SERVICE JOURNEY PATTERN.
ServicePatternRef
type ServicePatternRefStructure
Reference to a SERVICE PATTERN.
ConnectingOrder
type xsd:positiveInteger
Order of Point within Connecting journey as an absolute number.
ConnectingVisitNumber
type xsd:positiveInteger
Order of Point within Connecting as number of visits to the same stop. Default is 1.

**Figure 86 — *ConnectingJourneyView* — XSD**

#### 7.2.7.4 XML Example of Interchanges

##### 7.2.7.4.1 Service Journey Interchange - XML Example

The following code fragment shows two SERVICE JOURNEY INTERCHANGEs.

For EXAMPLE

```
<journeyInterchanges>
    <ServiceJourneyInterchange version="any" id="hde:sji_24o_01_to_46o_01">
<Description>15:00 Change for Romeo    </Description>
        <ConnectionRef version="any" ref="mybus:cx_SSP_002"/>
        <StaySeated>false</StaySeated>
        <Planned>true</Planned>
        <Guaranteed>false</Guaranteed>
        <Advertised>true</Advertised>
        <Controlled>false</Controlled>
        <StandardWaitTime>PT30M</StandardWaitTime>
        <MaximumWaitTime>PT35M</MaximumWaitTime>
        <StandardTransferTime>PT1M</StandardTransferTime>
        <MinimumTransferTime>PT1M</MinimumTransferTime>
        <FromPointRef version="any" ref="mybus:SSP_002"/>
        <ToPointRef version="any" ref="mybus:SSP_002"/>
        <FromJourneyRef version="any" ref="hde:sj_24o_01"/>
        <ToJourneyRef version="any" ref="hde:sj_46o_01"/>
    </ServiceJourneyInterchange>
    <ServiceJourneyInterchange version="any" id="hde:sji_24o_02_to_46o_02">
        <Description>16:00 Change for Romeo    </Description>
        <ConnectionRef version="any" ref="mybus:cx_SSP_002"/>
        <StaySeated>false</StaySeated>
        <Planned>true</Planned>
        <Guaranteed>false</Guaranteed>
        <Advertised>true</Advertised>
        <Controlled>false</Controlled>
        <StandardWaitTime>PT30M</StandardWaitTime>
        <MaximumWaitTime>PT35M</MaximumWaitTime>
        <StandardTransferTime>PT1M</StandardTransferTime>
        <MinimumTransferTime>PT1M</MinimumTransferTime>
        <FromPointRef version="any" ref="mybus:SSP_002"/>
        <ToPointRef version="any" ref="mybus:SSP_002"/>
        <FromJourneyRef version="any" ref="hde:sj_24o_02"/>
        <ToJourneyRef version="any" ref="hde:sj_46o_02"/>
    </ServiceJourneyInterchange>
```

##### 7.2.7.4.2 Journey Meeting - XML Example

The following code fragment shows tow SERVICE JOURNEY MEETINGs.

For EXAMPLE

```
<journeyMeetings>
    <JourneyMeeting version="any" id="hde:jm_24o_46o_01">
        <AtStopPointRef version="any" ref="mybus:SSP_002"/>
        <FromJourneyRef version="any" ref="hde:sj_24o_01"/>
        <ToJourneyRef version="any" ref="hde:sj_46o_01"/>
        <Description>Transfer to route 46</>
        <EarliestTime>14:30:00.0Z</EarliestTime>
        <LatestTime>14:35:00.0Z</LatestTime>
    </JourneyMeeting>
    <JourneyMeeting version="any" id="hde:jm_24o_46o_02">
        <AtStopPointRef version="any" ref="mybus:SSP_002"/>
        <FromJourneyRef version="any" ref="hde:sj_24o_01"/>
        <ToJourneyRef version="any" ref="hde:sj_46o_01"/>
        <Description>Transfer to route 46</Description>
        <EarliestTime>15:30:00.0Z</EarliestTime>
        <LatestTime>15:35:00.0Z</LatestTime>
    </JourneyMeeting>
</journeyMeetings>
```

### 7.2.8 Interchange Rule

#### 7.2.8.1 INTERCHANGE RULE – Conceptual MODEL

An INTERCHANGE RULE allows an intended interchange to be recorded in the schedule without having to specify the exact details of both SERVICE JOURNEYs involved in the interchange.

The INTERCHANGE RULE instead provides enough details so that eligible SERVICE JOURNEY INTERCHANGEs can be determined at a later stage after the scheduling is completed. This later stage could be when two schedules from different sources are coordinated in an integrator system, or in real time in a system considering late re-scheduling and delays of involved LINEs.

As described earlier, a SERVICE JOURNEY INTERCHANGE involves two different SERVICE JOURNEYs. The passenger has the possibility to transfer from a *feeder* SERVICE JOURNEY at a SCHEDULED STOP POINT to a *distributor* SERVICE JOURNEY at the same or nearby SCHEDULED STOP POINT.

Often the two involved SERVICE JOURNEYs in the interchange are planned separately. Sometimes the SERVICE JOURNEYs are worked by different companies.

This means that it is not an easy task for a scheduler to keep track of the exact details of the SERVICE JOURNEY at the other end of an intended interchange as the SERVICE JOURNEY at the other end might be altered, deleted or replaced at any time.

Instead of trying to keep track of such external changes and continuously readjust the schedule with the details of the external SERVICE JOURNEY, it is easier in this case to use an INTERCHANGE RULE.

The INTERCHANGE RULE uses INTERCHANGE RULE PARAMETERs which will remain stable over time to identify the involved SERVICE JOURNEYs.

The INTERCHANGE RULE PARAMETER specifies criteria that a candidate SERVICE JOURNEY shall fulfil to be considered. Examples of such criteria are working on a certain LINE in a specific DIRECTION. There could also be criteria relating to time of day or matching a specified SERVICE JOURNEY identifier. Which criteria or combination of criteria that are used will differ for different use cases.

In the same way it is possible to filter on which SCHEDULED STOP POINTs the INTERCHANGE RULE applies. The SCHEDULED STOP POINT of the feeder and distributor SERVICE JOURNEY are defined separately. Instead of specifying a SCHEDULED STOP POINT, it is possible to specify a STOP AREA. This is interpreted as that any SCHEDULED STOP POINT within that STOP AREA is accepted for the related SERVICE JOURNEY. The STOP AREA construction is useful when the exact details of where the vehicle is planned to stop are not known in advance, or might change at a late stage.

Finally there is also a final matching criterion, the maximum interchange window duration. This is used to filter out pairs of feeder and distributor SERVICE JOURNEYs where the feeder is planned to arrive so much earlier than the distributor is planned to depart, that the combination is not of interest as a SERVICE JOURNEY INTERCHANGE.

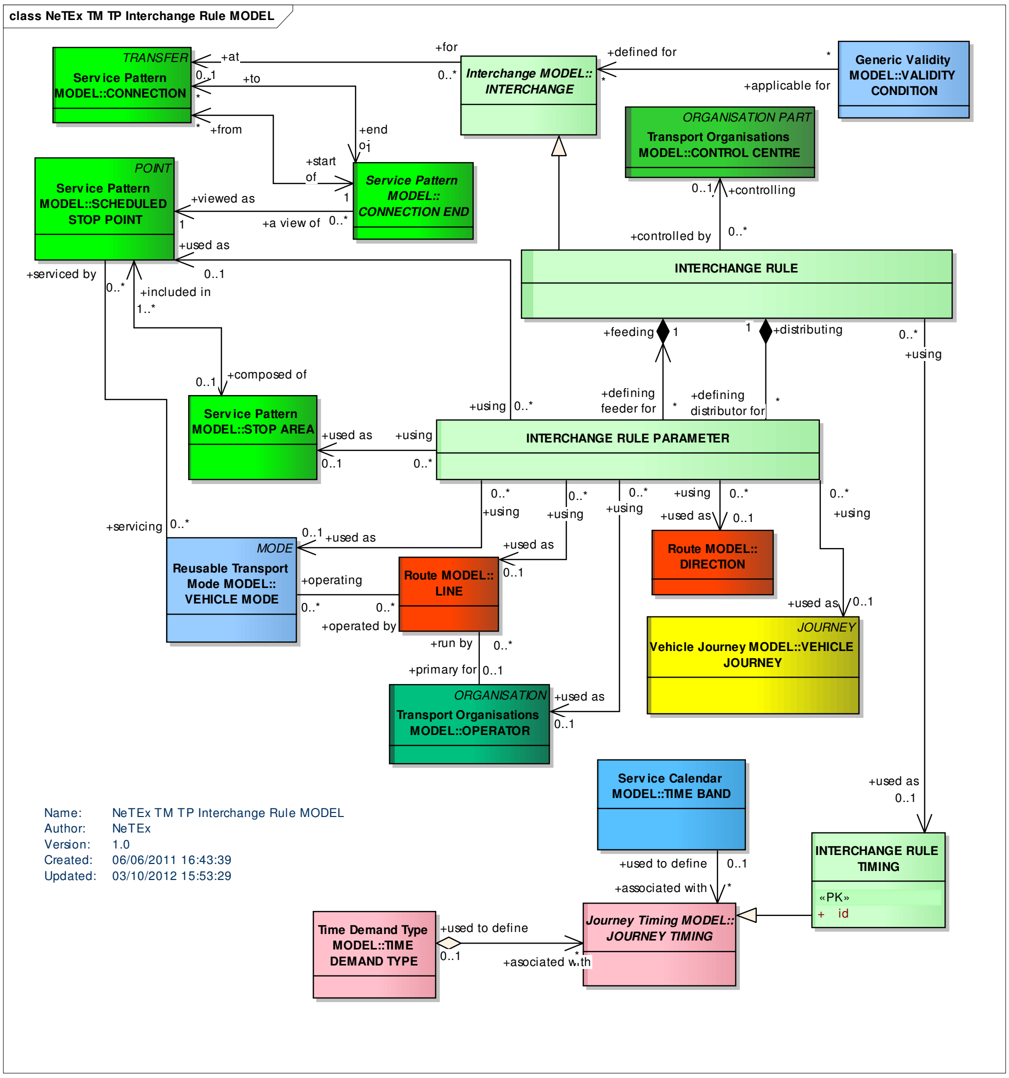

class NeTEx TM TP Interchange Rule MODEL
TRANSFER
Service Pattern MODEL::CONNECTION
+at
+to
+from
+for
+end
+start of
Interchange MODEL::INTERCHANGE
+defined for
+applicable for
Generic Validity MODEL::VALIDITY CONDITION
ORGANISATION PART
Transport Organisations MODEL::CONTROL CENTRE
+controlling
+controlled by
POINT
Service Pattern MODEL::SCHEDULED STOP POINT
+viewed as
+a view of
Service Pattern MODEL:: CONNECTION END
+used as
+serviced by
+included in
INTERCHANGE RULE
+feeding
+distributing
+using
+composed of
Service Pattern MODEL::STOP AREA
+defining feeder for
+defining distributor for
INTERCHANGE RULE PARAMETER
+used as
+using
+servicing
MODE
Reusable Transport Mode MODEL:: VEHICLE MODE
+operating
+operated by
Route MODEL:: LINE
Route MODEL:: DIRECTION
+run by
+primary for
ORGANISATION
Transport Organisations MODEL::OPERATOR
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
Service Calendar MODEL::TIME BAND
+used to define
+associated with
INTERCHANGE RULE TIMING
«PK»
+ id
Journey Timing MODEL:: JOURNEY TIMING
Time Demand Type MODEL::TIME DEMAND TYPE
+asociated with
Name: NeTEx TM TP Interchange Rule MODEL
Author: NeTEx
Version: 1.0
Created: 06/06/2011 16:43:39
Updated: 03/10/2012 15:53:29

**Figure 87 — Interchange Rule – Conceptual MODEL (UML)**

### 7.2.8.2 Interchange Rule – Physical Model

The following figure shows the physical model for an INTERCHANGE RULE.

Each INTERCHANGE RULE may have a feeder and distributor filter: that is, an INTERCHANGE RULE PARAMETER that constrains the rule only to apply to a specified set of journeys. Constraints include the LINE, DIRECTION, OPERATOR, connecting STOP POINT, connection STOP AREA, origin STOP POINT, and penultimate STOP POINT.

There may be different INTERCHANGE RULE TIMINGs for different times of day.

— MaximumWindow expresses the maximum duration between the feeder SERVICE JOURNEYs planned arrival time and the distributor SERVICE JOURNEYs planned departure time.

— MaximumWaitTime expresses the maximum duration a distributor SERVICE JOURNEY should delay its departure time waiting for a feeder SERVICE JOURNEY.

The default transfer time of the passenger should be added to the forecasted arrival time of the feeder SERVICE JOURNEY to decide if the distributor SERVICE JOURNEY will exceed the **MaximumWaitTime** when waiting for passengers from the delayed distributor.

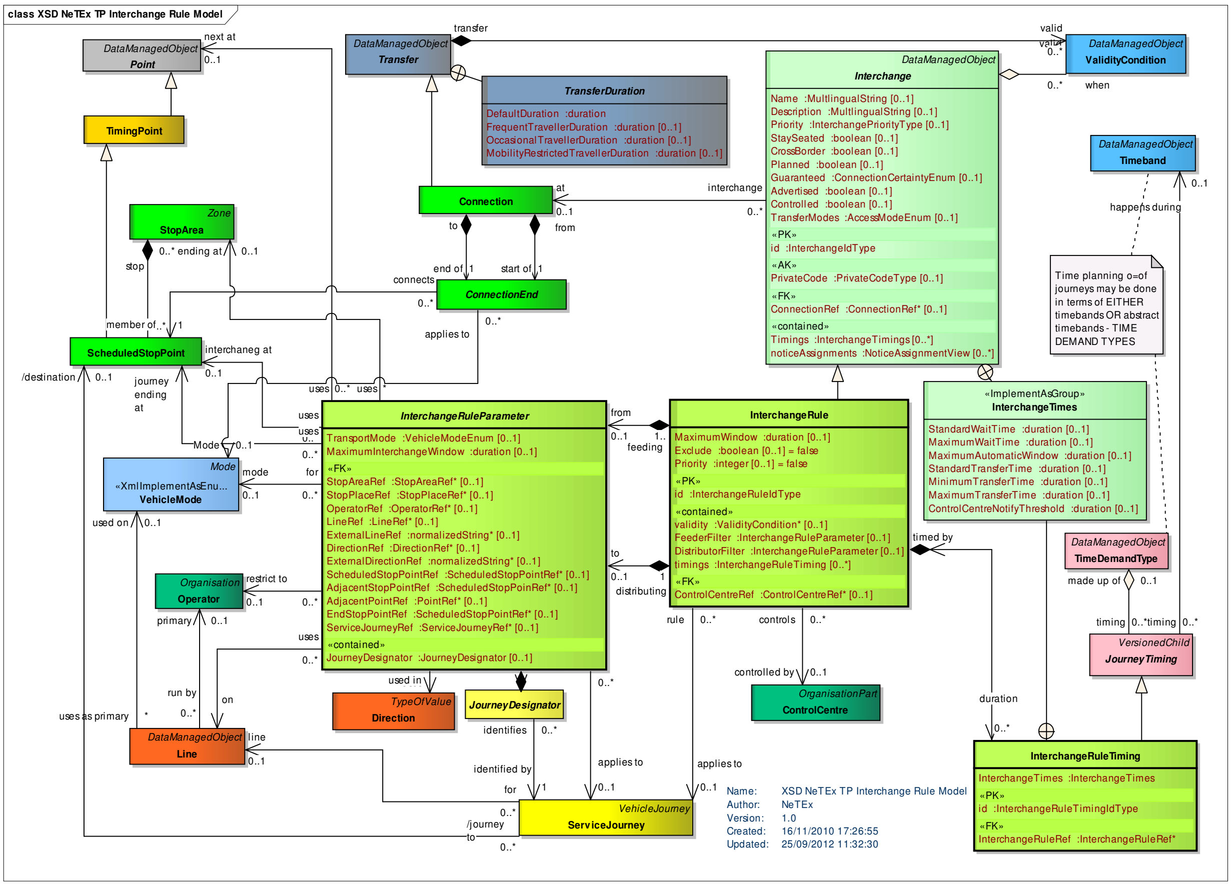

class XSD NeTEx TP Interchange Rule Model
DataManagedObject
Point
next at
0..1
TimingPoint
Zone
StopArea
stop
0..* ending at
0..1
member of ..*
1
ScheduledStopPoint
interchaneg at
0..1
/destination
0..1
journey
ending
at
DataManagedObject
Transfer
transfer
TransferDuration
DefaultDuration :duration
FrequentTravellerDuration :duration [0..1]
OccasionalTravellerDuration :duration [0..1]
MobilityRestrictedTravellerDuration :duration [0..1]
Connection
at
0..1
interchange
0..*
to
from
end of 1
start of 1
connects
0..*
ConnectionEnd
0..*
applies to
uses 0..* uses *
DataManagedObject
Interchange
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
Priority :InterchangePriorityType [0..1]
StaySeated :boolean [0..1]
CrossBorder :boolean [0..1]
Planned :boolean [0..1]
Guaranteed :ConnectionCertaintyEnum [0..1]
Advertised :boolean [0..1]
Controlled :boolean [0..1]
TransferModes :AccessModeEnum [0..1]
«PK»
id :InterchangeIdType
«AK»
PrivateCode :PrivateCodeType [0..1]
«FK»
ConnectionRef :ConnectionRef* [0..1]
«contained»
Timings :InterchangeTimings [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
valid
valid
0..*
DataManagedObject
ValidityCondition
0..*
when
DataManagedObject
Timeband
0..1
happens during
Time planning o=of journeys may be done in terms of EITHER timebands OR abstract timebands - TIME DEMAND TYPES
InterchangeRuleParameter
TransportMode :VehicleModeEnum [0..1]
MaximumInterchangeWindow :duration [0..1]
«FK»
StopAreaRef :StopAreaRef* [0..1]
StopPlaceRef :StopPlaceRef* [0..1]
OperatorRef :OperatorRef* [0..1]
LineRef :LineRef* [0..1]
ExternalLineRef :normalizedString* [0..1]
DirectionRef :DirectionRef* [0..1]
ExternalDirectionRef :normalizedString* [0..1]
ScheduledStopPointRef :ScheduledStopPointRef* [0..1]
AdjacentStopPointRef :ScheduledStopPoinRef* [0..1]
AdjacentPointRef :PointRef* [0..1]
EndStopPointRef :ScheduledStopPointRef* [0..1]
ServiceJourneyRef :ServiceJourneyRef* [0..1]
«contained»
JourneyDesignator :JourneyDesignator [0..1]
uses
uses
Mode
0..1
0..*
Mode
«XmlImplementAsEnu...
VehicleMode
mode
for
0..1
0..*
used on
0..1
Organisation
Operator
restrict to
0..1
0..*
primary
0..1
uses
0..*
run by
0..*
on
uses as primary
*
DataManagedObject
Line
line
0..1
used in
TypeOfValue
Direction
JourneyDesignator
identifies
0..*
identified by
for
1
0..*
applies to
0..1
0..*
/journey
to
0..*
VehicleJourney
ServiceJourney
from
0..1
1..
feeding
InterchangeRule
MaximumWindow :duration [0..1]
Exclude :boolean [0..1] = false
Priority :integer [0..1] = false
«PK»
id :InterchangeRuleIdType
«contained»
validity :ValidityCondition* [0..1]
FeederFilter :InterchangeRuleParameter [0..1]
DistributorFilter :InterchangeRuleParameter [0..1]
timings :InterchangeRuleTiming [0..*]
«FK»
ControlCentreRef :ControlCentreRef* [0..1]
to
0..1
1
distributing
rule
0..*
controls
0..*
controlled by
0..1
OrganisationPart
ControlCentre
applies to
0..1
timed by
duration
0..*
«ImplementAsGroup»
InterchangeTimes
StandardWaitTime :duration [0..1]
MaximumWaitTime :duration [0..1]
MaximumAutomaticWindow :duration [0..1]
StandardTransferTime :duration [0..1]
MinimumTransferTime :duration [0..1]
MaximumTransferTime :duration [0..1]
ControlCentreNotifyThreshold :duration [0..1]
DataManagedObject
TimeDemandType
made up of
0..1
timing
0..*timing
0..*
VersionedChild
JourneyTiming
InterchangeRuleTiming
InterchangeTimes :InterchangeTimes
«PK»
id :InterchangeRuleTimingIdType
«FK»
InterchangeRuleRef :InterchangeRuleRef*
Name: XSD NeTEx TP Interchange Rule Model
Author: NeTEx
Version: 1.0
Created: 16/11/2010 17:26:55
Updated: 25/09/2012 11:32:30

Figure 88 — Interchange Rule – Physical Model (UML)

### 7.2.8.3 Interchange Rule – Attributes and XSD

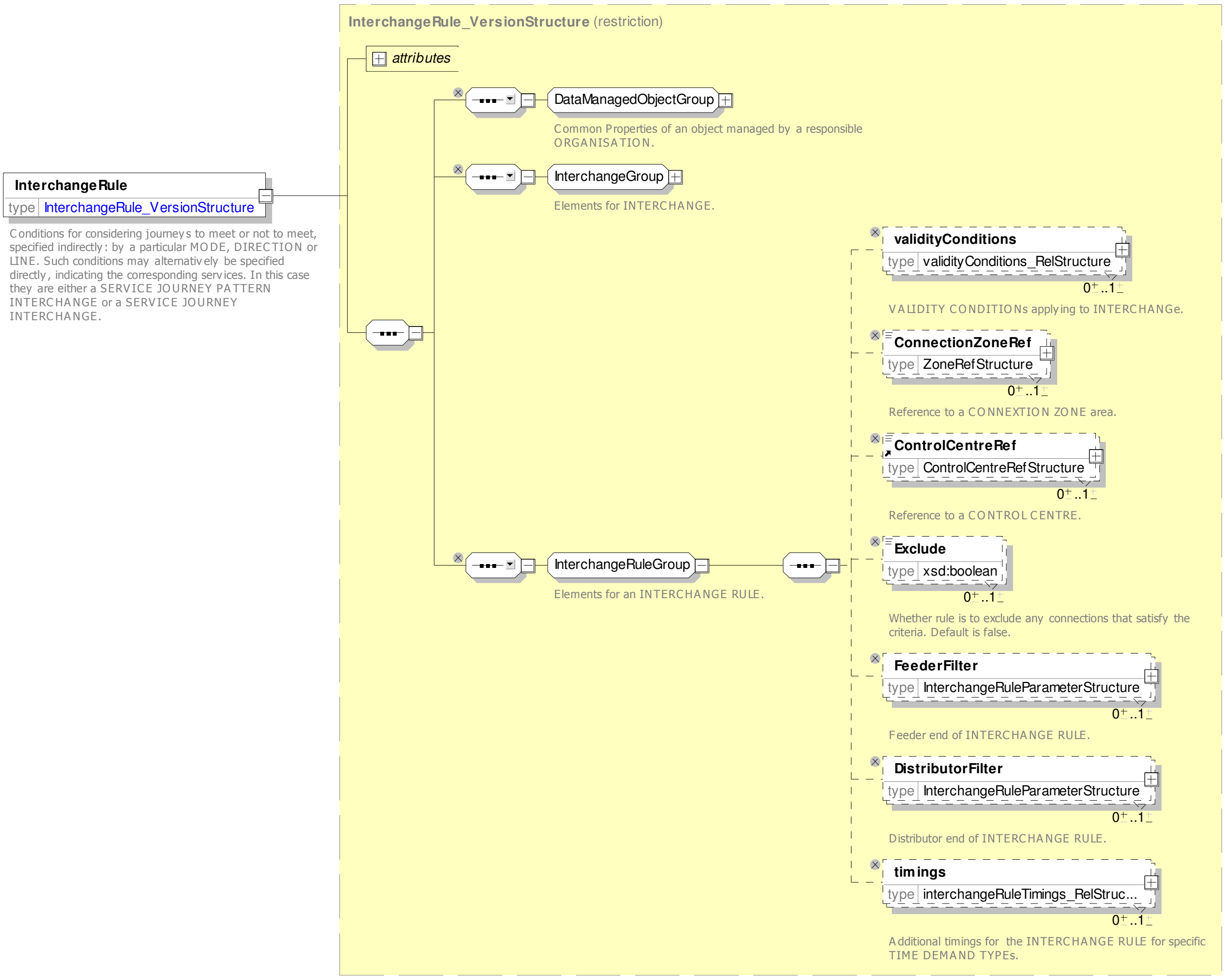
#### 7.2.8.3.1 InterchangeRule – Model Element

INTERCHANGE RULE specifies conditions for considering JOURNEYs to meet or not to meet, specified indirectly: by a particular MODE, DIRECTION or LINE. Such conditions may alternatively be specified directly, indicating the corresponding services. In this case they are either a SERVICE JOURNEY PATTERN INTERCHANGE or a SERVICE JOURNEY INTERCHANGE.

Table 52 – *InterchangeRule* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | INTERCHANGE RULE inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *InterchangeRuleIdType* | 1:1 | Identifier of INTERCHANGE RULE. |
| | ***MaximumWindow*** | *xsd:duration* | 0:1 | Maximum window for holding DISTRIBUTOR will wait. |
| «cntd» | ***validity*** | *ValidityCondition* | 0:1 | VALIDITY CONDITION governing INTERCHANGE RULE. |
| | ***Exclude*** | *xsd:boolean* | 0:1 | Whether rule is to exclude interchanges of journeys that match the filter criteria. |

|  |  |  |  |  |
|---|---|---|---|---|
| «FK» | ***ControlCentreRef*** | *ControlCentreRef* | 0:1 | Reference to a CONTROL CENTRE for which RULE applies. |
| «cntd» | ***FeederFilter*** | *InterchangeRule-Parameter* | 0:1 | Feeder journey INTERCHANGE RULE FILTER associated with INTERCHANGE RULE. |
| «cntd» | ***DistributorFilter*** | *InterchangeRule-Parameter* | 0:1 | Distributor journey INTERCHANGE RULE FILTER associated with INTERCHANGE RULE. |
| «cntd» | ***timings*** | *InterchangeRuleTiming* | 0:* | Timings for INTERCHANGE RULE. |



InterchangeRule_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
InterchangeGroup
Elements for INTERCHANGE.
InterchangeRule
type InterchangeRule_VersionStructure
Conditions for considering journeys to meet or not to meet, specified indirectly: by a particular MODE, DIRECTION or LINE. Such conditions may alternatively be specified directly, indicating the corresponding services. In this case they are either a SERVICE JOURNEY PATTERN INTERCHANGE or a SERVICE JOURNEY INTERCHANGE.
InterchangeRuleGroup
Elements for an INTERCHANGE RULE.
validityConditions
type validityConditions_RelStructure
0..1
VALIDITY CONDITIONs applying to INTERCHANGE.
ConnectionZoneRef
type ZoneRefStructure
0..1
Reference to a CONNEXTION ZONE area.
ControlCentreRef
type ControlCentreRefStructure
0..1
Reference to a CONTROL CENTRE.
Exclude
type xsd:boolean
0..1
Whether rule is to exclude any connections that satisfy the criteria. Default is false.
FeederFilter
type InterchangeRuleParameterStructure
0..1
Feeder end of INTERCHANGE RULE.
DistributorFilter
type InterchangeRuleParameterStructure
0..1
Distributor end of INTERCHANGE RULE.
timings
type interchangeRuleTimings_RelStruc...
0..1
Additional timings for the INTERCHANGE RULE for specific TIME DEMAND TYPEs.

**Figure 89 — *InterchangeRule* — XSD**

#### 7.2.8.3.2 InterchangeRuleParameter – Model Element

Assignment of parameters characterising an INTERCHANGE RULE.

A specification of the filtering criteria for the feeder or distributor end of an INTERCHANGE RULE.

**Table 53 – *InterchangeRuleParameter* – Element**

| Classifi | Name | Type | Cardin- | Description |
|---|---|---|---|---|

| cation | | | ality | |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | INTERCHANGE RULE PARAMETER inherits from DATA MANAGED OBJECT |
| | ***TransportMode*** | *VehicleModeEnum* | 0:1 | Mode of CONNECTING JOURNEYs to which INTERCHANGE RULE applies. |
| «FK» | ***StopAreaRef*** | *StopAreaRef* | 0:1 | STOP AREA to which INTERCHANGE RULE applies. |
| «FK» | ***StopPlaceRef*** | *StopPlaceRef* | 0:1 | STOP PLACE feeding or distributing JOURNEY to which INTERCHANGE RULE applies. |
| «FK» | ***OperatorRef*** | *OperatorRef* | 0:1 | OPERATOR of JOURNEYs to which INTERCHANGE RULE applies. |
| «FK» | ***LineRef*** | *LineRef* | 0:1 | LINE of JOURNEYs to which INTERCHANGE RULE applies. |
| «FK» | ***ExternalLineRef*** | *ExternalLineRef* | 0:1 | External reference to LINE of JOURNEYs to which INTERCHANGE RULE applies. |
| «FK» | ***DirectionRef*** | *DirectionRef* | 0:1 | Reference to DIRECTION of TRANSFER to which INTERCHANGE RULE applies |
| «FK» | ***ExternalDirectionRef*** | *ExternalDirectionRef* | 0:1 | External reference to DIRECTION of TRANSFER to which INTERCHANGE RULE applies |
| «FK» | ***ScheduledStopPointRef*** | *ScheduledStopPointRef* | 0:1 | Start or end SCHEDULED STOP POINT in INTERCHANGE of connecting JOURNEY to which INTERCHANGE RULE applies. |
| «FK» | ***AdjacentStopPointRef*** | *ScheduledStopPointRef* | 0:1 | Prior (feeder) or onwards (distributor) STOP POINT before of journeys to which INTERCHANGE RULE applies. |
| «FK» | ***AdjacentStopPlaceRef*** | *StopPlaceRef* | 0:1 | Prior (feeder) or onwards (distributor) STOP PLACE before of journeys to which INTERCHANGE RULE applies. |
| «FK» | ***AdjacentPointRef*** | *PointRef* | 0:1 | Prior (feeder) or onwards (distributor) POINT before connection used by journey to which INTERCHANGE RULE applies. |
| «FK» | ***EndStopPointRef*** | *ScheduledStopPointRef* | 0:1 | Origin (for feed journeys) or Destination (for distributor journeys) SCHEDULED STOP POINT of connecting JOURNEY to which INTERCHANGE RULE applies. |
| «FK» | ***ServiceJourneyRef*** | *ServiceJourneyRef* | 0:1 | SERVICE JOURNEY to which INTERCHANGE RULE applies. |
| | ***MaximumInterchangeWindow*** | *xsd:duration* | 0:1 | Maximum interval for making interchange. |
| «cntd» | ***JourneyDesignator*** | *JourneyDesignator* | 0:1 | Means of identifying a JOURNEY whose ID is not known. |

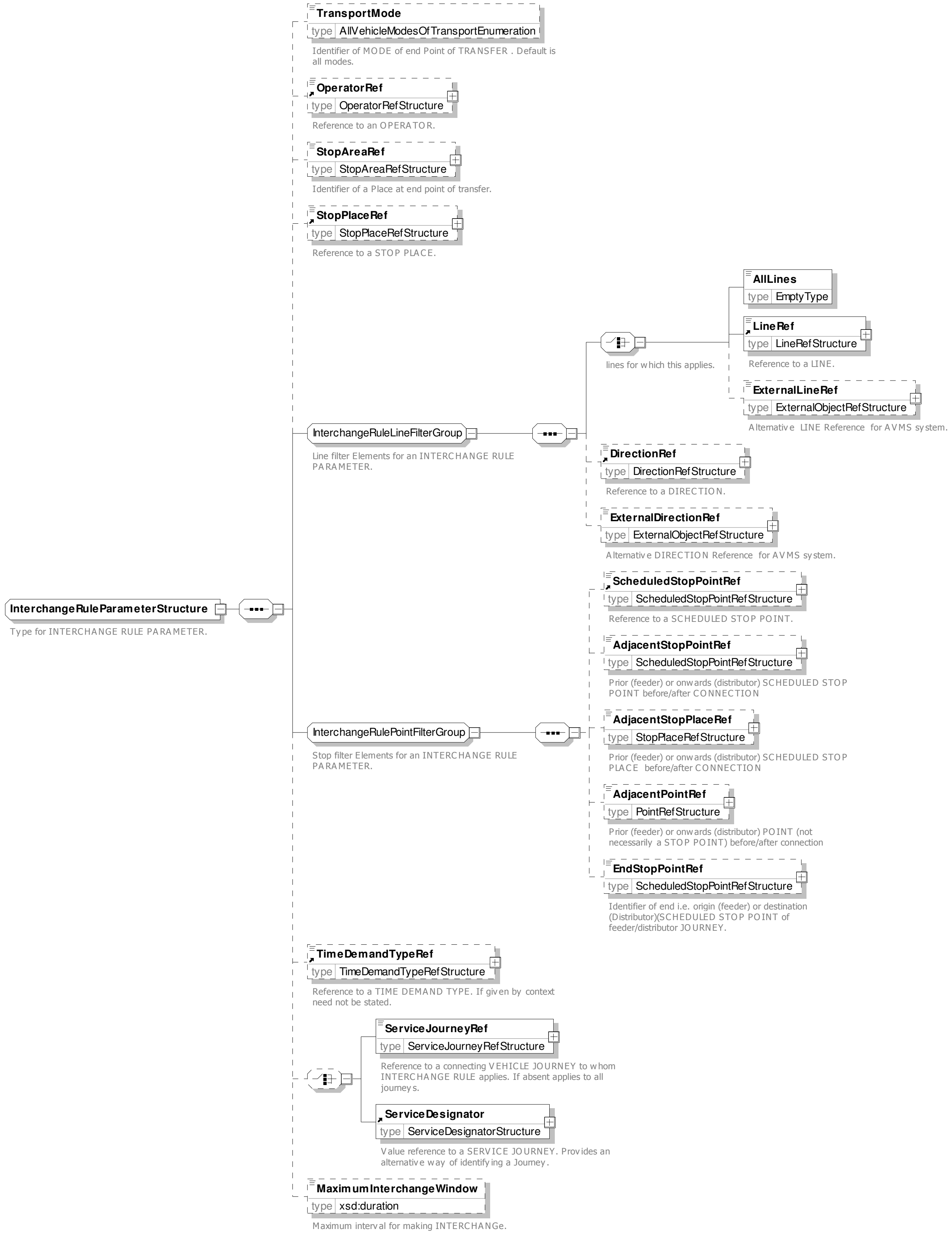

TransportMode
type AllVehicleModesOfTransportEnumeration
Identifier of MODE of end Point of TRANSFER . Default is all modes.
OperatorRef
type OperatorRefStructure
Reference to an OPERATOR.
StopAreaRef
type StopAreaRefStructure
Identifier of a Place at end point of transfer.
StopPlaceRef
type StopPlaceRefStructure
Reference to a STOP PLACE.
AllLines
type EmptyType
LineRef
type LineRefStructure
Reference to a LINE.
lines for which this applies.
ExternalLineRef
type ExternalObjectRefStructure
Alternative LINE Reference for AVMS system.
InterchangeRuleLineFilterGroup
Line filter Elements for an INTERCHANGE RULE PARAMETER.
DirectionRef
type DirectionRefStructure
Reference to a DIRECTION.
ExternalDirectionRef
type ExternalObjectRefStructure
Alternative DIRECTION Reference for AVMS system.
InterchangeRuleParameterStructure
Type for INTERCHANGE RULE PARAMETER.
ScheduledStopPointRef
type ScheduledStopPointRefStructure
Reference to a SCHEDULED STOP POINT.
AdjacentStopPointRef
type ScheduledStopPointRefStructure
Prior (feeder) or onwards (distributor) SCHEDULED STOP POINT before/after CONNECTION
AdjacentStopPlaceRef
type StopPlaceRefStructure
Prior (feeder) or onwards (distributor) SCHEDULED STOP PLACE before/after CONNECTION
InterchangeRulePointFilterGroup
Stop filter Elements for an INTERCHANGE RULE PARAMETER.
AdjacentPointRef
type PointRefStructure
Prior (feeder) or onwards (distributor) POINT (not necessarily a STOP POINT) before/after connection
EndStopPointRef
type ScheduledStopPointRefStructure
Identifier of end i.e. origin (feeder) or destination (Distributor)(SCHEDULED STOP POINT of feeder/distributor JOURNEY.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
ServiceJourneyRef
type ServiceJourneyRefStructure
Reference to a connecting VEHICLE JOURNEY to whom INTERCHANGE RULE applies. If absent applies to all journeys.
ServiceDesignator
type ServiceDesignatorStructure
Value reference to a SERVICE JOURNEY. Provides an alternative way of identifying a Journey.
MaximumInterchangeWindow
type xsd:duration
Maximum interval for making INTERCHANGe.

**Figure 90 — *InterchangeRuleParameter* — XSD**

### 7.2.8.3.3 InterchangeRuleTiming – Model Element

Conditions for considering JOURNEYs to meet or not to meet, specified indirectly: by a particular MODE, DIRECTION or LINE. Such conditions may alternatively be specified directly, indicating the corresponding services. In this case they are either a SERVICE JOURNEY PATTERN INTERCHANGE or a SERVICE JOURNEY INTERCHANGE.

Table 54 – *InterchangeRuleTiming* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *VersionedChild* | ::> | INTERCHANGE RULE TIMING inherits from VERSIONED CHILD. |
| «PK» | ***id*** | *InterchangeRule-TimingIdType* | 1:1 | Identifier of INTERCHANGE RULE TIMING. |
| «FK» | ***InterchangeRuleRef*** | *InterchangeRuleRef* | 1:1 | Identifier of INTERCHANGE RULE to which this timing belongs. |
| | ***MaximumWindow*** | *xsd:duration* | 0:1 | Maximum window for holding DISTRIBUTOR will wait. |
| | ***StandardWaitTime*** | *xsd:duration* | 0:1 | Standard Time to wait at Interchange. |
| | ***MaximumWaitTime*** | *xsd:duration* | 0:1 | Maximum time that DISTRIBUTOR will wait. |
| | ***StandardTransferTime*** | *xsd:duration* | 0:1 | Standard Time needed for transfer. |
| | ***MinimumTransferTime*** | *xsd:duration* | 0:1 | Minimum Time needed for transfer. |
| | ***MaximumTransferTime*** | *xsd:duration* | 0:1 | Maximum Time needed for transfer. |

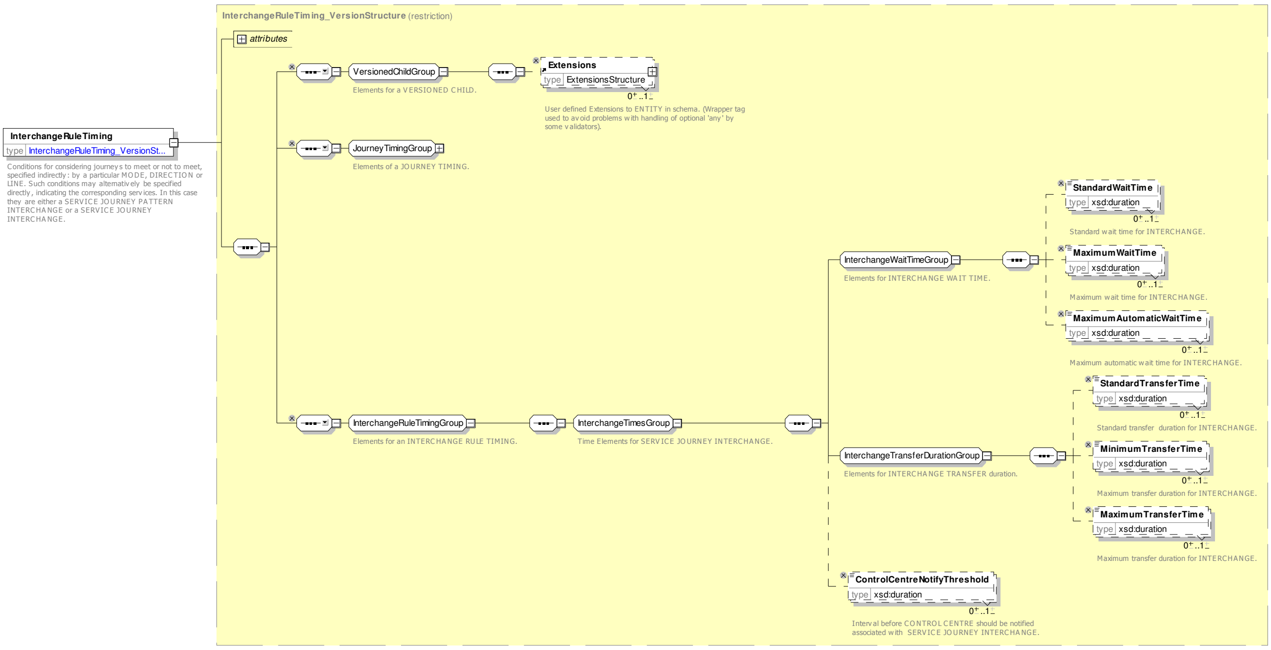

InterchangeRuleTiming_VersionStructure (restriction)
attributes
InterchangeRuleTiming
type InterchangeRuleTiming_VersionSt...
Conditions for considering journeys to meet or not to meet, specified indirectly: by a particular MODE, DIRECTION or LINE. Such conditions may alternatively be specified directly, indicating the corresponding services. In this case they are either a SERVICE JOURNEY PATTERN INTERCHANGE or a SERVICE JOURNEY INTERCHANGE.
VersionedChildGroup
Elements for a VERSIONED CHILD.
Extensions
type ExtensionsStructure
0..1
User defined Extensions to ENTITY in schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
JourneyTimingGroup
Elements of a JOURNEY TIMING.
InterchangeRuleTimingGroup
Elements for an INTERCHANGE RULE TIMING.
InterchangeTimesGroup
Time Elements for SERVICE JOURNEY INTERCHANGE.
InterchangeWaitTimeGroup
Elements for INTERCHANGE WAIT TIME.
StandardWaitTime
type xsd:duration
0..1
Standard wait time for INTERCHANGE.
MaximumWaitTime
type xsd:duration
0..1
Maximum wait time for INTERCHANGE.
MaximumAutomaticWaitTime
type xsd:duration
0..1
Maximum automatic wait time for INTERCHANGE.
InterchangeTransferDurationGroup
Elements for INTERCHANGE TRANSFER duration.
StandardTransferTime
type xsd:duration
0..1
Standard transfer duration for INTERCHANGE.
MinimumTransferTime
type xsd:duration
0..1
Maximum transfer duration for INTERCHANGE.
MaximumTransferTime
type xsd:duration
0..1
Maximum transfer duration for INTERCHANGE.
ControlCentreNotifyThreshold
type xsd:duration
0..1
Interval before CONTROL CENTRE should be notified associated with SERVICE JOURNEY INTERCHANGE.

Figure 91 — *InterchangeRuleTiming* — XSD

### 7.2.8.4 XML Example of Interchange Rule

The following code fragment shows three INTERCHANGE RULEs.

For EXAMPLE

```
<interchangeRules>
    <InterchangeRule version="any" id="hde:IR_25_all_f">
        <Name>Main Station - All inbound</Name>
        <ConnectionRef version="any" ref="mybus:Connection:CX_25"/>
        <Priority>5</Priority>
        <Planned>true</Planned>
        <Guaranteed>false</Guaranteed>
        <Advertised>true</Advertised>
        <Controlled>true</Controlled>
        <MaximumWaitTime>PT2M</MaximumWaitTime>
        <MaximumAutomaticWaitTime>PT2M</MaximumAutomaticWaitTime>
        <StandardTransferTime>PT5M</StandardTransferTime>
        <MinimumTransferTime>PT3M</MinimumTransferTime>
        <MaximumTransferTime>PT8M</MaximumTransferTime>
        <ControlCentreRef ref="hde:123"/>
        <FeederFilter>
            <TransportMode>bus</TransportMode>
            <AllLines/>
            <MaximumInterchangeWindow>PT5M</MaximumInterchangeWindow>
        </FeederFilter>
    </InterchangeRule>
    <InterchangeRule version="any" id="hde:IR_25_LN1_f">
        <Name>Main Station - Line 1 inbound Special rule</Name>
        <ConnectionRef version="any" ref="mybus:CX_25"/>
        <Priority>1</Priority>
        <Planned>true</Planned>
        <Guaranteed>true</Guaranteed>
        <Advertised>true</Advertised>
        <Controlled>true</Controlled>
        <StandardWaitTime>PT3M</StandardWaitTime>
        <MaximumWaitTime>PT3M</MaximumWaitTime>
        <MaximumAutomaticWaitTime>PT1M</MaximumAutomaticWaitTime>
        <ControlCentreRef ref="hde:123"/>
        <FeederFilter>
            <TransportMode>bus</TransportMode>
            <LineRef version="any" ref="mybus:Line:LN_1"/>
            <DirectionRef version="any" ref="mybus:DR_hbf"/>
            <ScheduledStopPointRef version="any" ref="mybus:SSP_025"/>
            <NextStopPointRef version="any" ref="mybus:SSP_003"/>
            <EndStopPointRef version="any" ref="mybus:SSP_002"/>
            <MaximumInterchangeWindow>PT4M</MaximumInterchangeWindow>
        </FeederFilter>
    </InterchangeRule>
    <InterchangeRule version="any" id="hde:IR_25_LN6_d">
        <Name>Main Station - Line 6 outbound </Name>
        <ConnectionRef version="any" ref="mybus:CX_25"/>
        <Priority>5</Priority>
        <Planned>true</Planned>
        <Guaranteed>true</Guaranteed>
        <Advertised>true</Advertised>
        <Controlled>true</Controlled>
        <StandardWaitTime>PT4M</StandardWaitTime>
        <MaximumWaitTime>PT4M</MaximumWaitTime>
        <MaximumAutomaticWaitTime>PT4M</MaximumAutomaticWaitTime>
        <StandardTransferTime>PT5M</StandardTransferTime>
        <MinimumTransferTime>PT3M</MinimumTransferTime>
        <MaximumTransferTime>PT8M</MaximumTransferTime>
        <ControlCentreRef ref="hde:ControlCentre:123"/>
        <DistributorFilter>
            <TransportMode>bus</TransportMode>
            <LineRef version="any" ref="mybus:Line:LN_6"/>
            <DirectionRef version="any" ref="mybus:DR_Hospital"/>
            <ScheduledStopPointRef version="any" ref="mybus:SSP_025"/>
            <NextStopPointRef version="any" ref="mybus:SSP_011"/>
            <EndStopPointRef version="any" ref="mybus:SSP_012"/>
            <MaximumInterchangeWindow>PT5M</MaximumInterchangeWindow>
        </DistributorFilter>
    </InterchangeRule>
```

```
</interchangeRules>
```

### 7.2.9 Coupled Journey

#### 7.2.9.1 COUPLED JOURNEY – Conceptual MODEL

An important additional issue for rail systems compared to conventional bus systems, is the operation of vehicles coupled in trains, rather than running as separate autonomous vehicles. One consequence is the additional possibility to adjust the service supply to the demand as regards the types of vehicle. Trains may be shortened or prolonged during the day, or even within one service journey.

Furthermore the description of journeys to the passenger is more complex as train may be separated in two (or more) parts at a particular branching point, where both train parts continue their journey on different routes towards separate destinations. Conversely, two short trains coming from different feeding routes may be scheduled to meet at a joining point, where they are coupled to continue their service as one long train on a common route.

NeTEx separately represents the coupling of vehicles in a TRAIN and the linking of JOURNEY PARTs of VEHICLE JOURNEYs.

The concept of a COUPLED JOURNEY is related to the action of vehicle coupling.

‘Vehicle’ is understood as a unit remaining stable all along a VEHICLE JOURNEY. Therefore, a ‘vehicle’ may be either a single vehicle (e.g. a single tramway vehicle) or a TRAIN composed of several TRAIN ELEMENTs or a COMPOUND TRAIN composed of elementary TRAINs.

The considerations related to the vehicle coupling actions are based on the concept of VEHICLE TYPE.

The entity TRAIN describes an elementary train and is thus a VEHICLE TYPE. A TRAIN consists of TRAIN ELEMENTs assembled together. The composition of the TRAIN is provided by a TRAIN COMPONENT, giving the order of the TRAIN ELEMENT in the TRAIN.

Like any vehicle, a TRAIN operates VEHICLE JOURNEYs. If there is no coupling action during a VEHICLE JOURNEY, there is only one TRAIN for the journey. Two or more TRAINs may be coupled together for a part of a VEHICLE JOURNEY, or for longer periods, during which the composition of the compound train changes (thus the VEHICLE TYPE). Such coupled vehicles are represented by the concept of COMPOUND TRAINs.

Two distinct points of view of vehicle coupling need to be taken into account:

- For operational management, which will typically manage work periods for each VEHICLE TYPE. Such periods are described as BLOCKs, worked from a PARKING POINT to another, composed of sets of VEHICLE JOURNEYs. BLOCKs may be coupled (building COMPOUND BLOCKs, representing the work of a vehicle during the time it is coupled to another vehicle) or separated for a while, building BLOCK PARTs, i.e. the parts of a BLOCK corresponding to the different JOURNEY PARTs of the VEHICLE JOURNEYs in a BLOCK. In this modelling TYPE OF COUPLING allows for a classification of BLOCK PARTs, for example to indicate whether the coupling or the separation may occur at the start or at the end of a BLOCK.

- For passenger information, which is not concerned by the BLOCK description but by the fact that VEHICLE JOURNEYs are either coupled or not; these actions are occurring during a journey (at an intermediate point, e.g. where two routes meet, or separation where they diverge), JOURNEY PARTs are considered to describe the different journey parts. Any part of a VEHICLE JOURNEY that is coupled with another is qualified with a JOURNEY PART. In such a case, the PURPOSE OF JOURNEY PARTITION is “coupling”. One of the coupled JOURNEY PARTs is considered to be the main part of the compound vehicle formed. The entity JOURNEY PART COUPLE represents the coupling of one JOURNEY PART to the main one. In the case of the separation of vehicles, JOURNEY PARTs are created and the PURPOSE OF JOURNEY PARTITION is “separation”.

#### 7.2.9.1.1 Train coupling – Conceptual MODEL

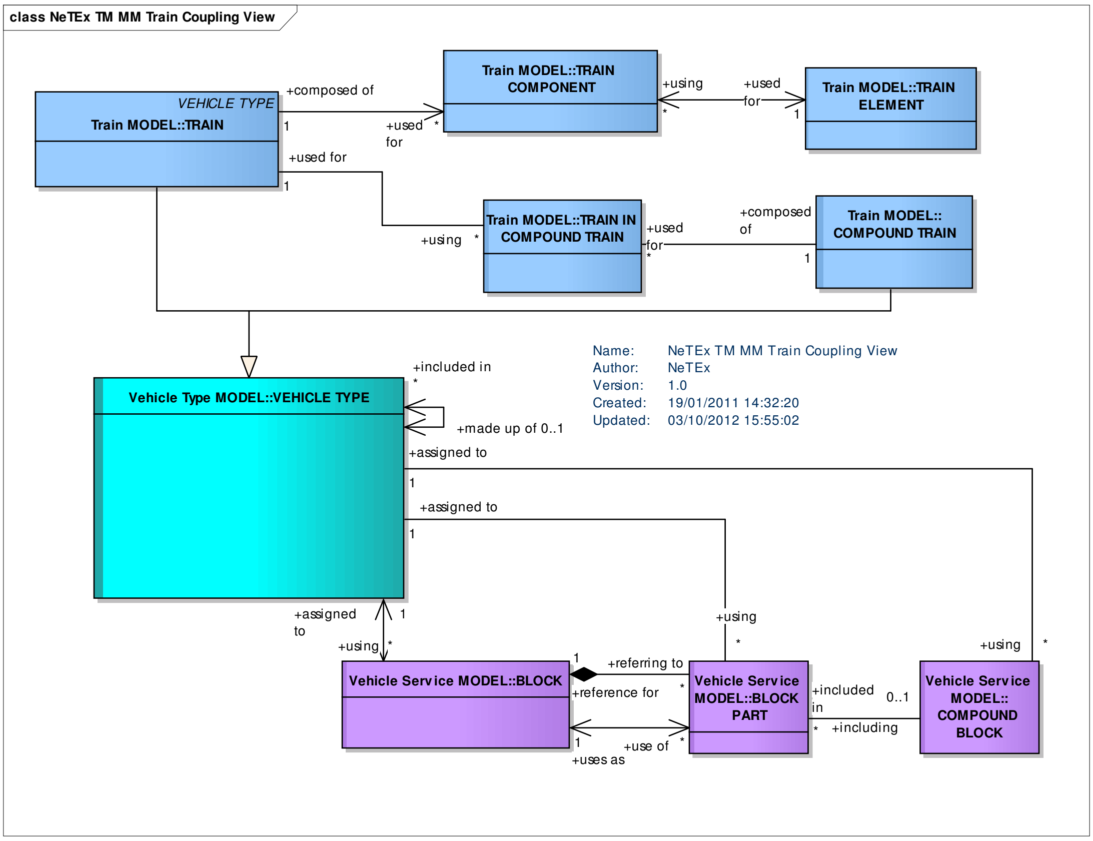

class NeTEx TM MM Train Coupling View
VEHICLE TYPE
Train MODEL::TRAIN
+composed of
1
+used for
*
Train MODEL::TRAIN COMPONENT
+using
*
+used for
1
Train MODEL::TRAIN ELEMENT
+used for
1
+using
*
Train MODEL::TRAIN IN COMPOUND TRAIN
+used for
*
+composed of
1
Train MODEL:: COMPOUND TRAIN
Name: NeTEx TM MM Train Coupling View
Author: NeTEx
Version: 1.0
Created: 19/01/2011 14:32:20
Updated: 03/10/2012 15:55:02
Vehicle Type MODEL::VEHICLE TYPE
+included in
*
+made up of 0..1
+assigned to
1
+assigned to
1
+assigned to
1
+using
*
Vehicle Service MODEL::BLOCK
1
+referring to
+reference for
*
Vehicle Service MODEL::BLOCK PART
1
+uses as
+use of
*
+using
*
+included in
0..1
*
+including
+using
*
Vehicle Service MODEL:: COMPOUND BLOCK

**Figure 92 — Train Coupling – Conceptual MODEL (UML)**

#### 7.2.9.1.2 Journey Parts & Train Numbers

A COUPLED JOURNEY is a complete journey operated by a COMPOUND TRAIN, composed of two or more VEHICLE JOURNEYs remaining coupled together all along a JOURNEY PATTERN. A COUPLED JOURNEY may be viewed as a single VEHICLE JOURNEY. This view of a COUPLED JOURNEY as a single VEHICLE JOURNEY is used for passenger information, as the passenger does not care whether this journey is composite from the vehicle management point of view.

Both operational and passenger information points of view are linked to a concept of TRAIN NUMBER, which gives a specification of codes assigned to particular VEHICLE JOURNEYs, JOURNEY PARTs and JOURNEY PART COUPLEs when operated by TRAINs or COMPOUND TRAINs according to a functional purpose (passenger information, operation follow-up, etc.). Two attributes control the exposing of numbers for use: ***ForAdvertisement*** and ***ForProduction***.

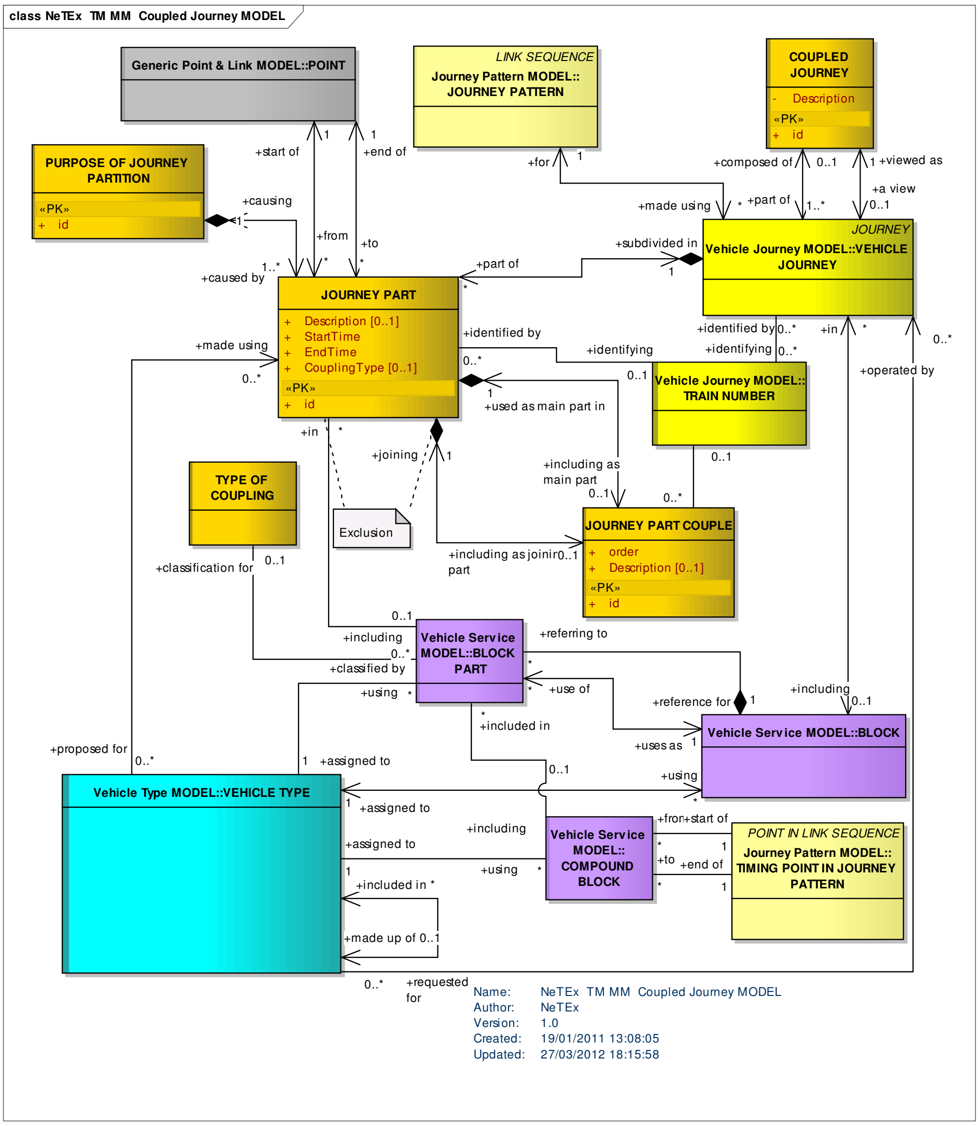


Figure 93 — Coupled Journey – Conceptual MODEL (UML)

### 7.2.9.2 Coupled Journey – Example

Both the joining and splitting of train journeys can be represented in NeTEx, with exact information about which carriage goes to which destination.

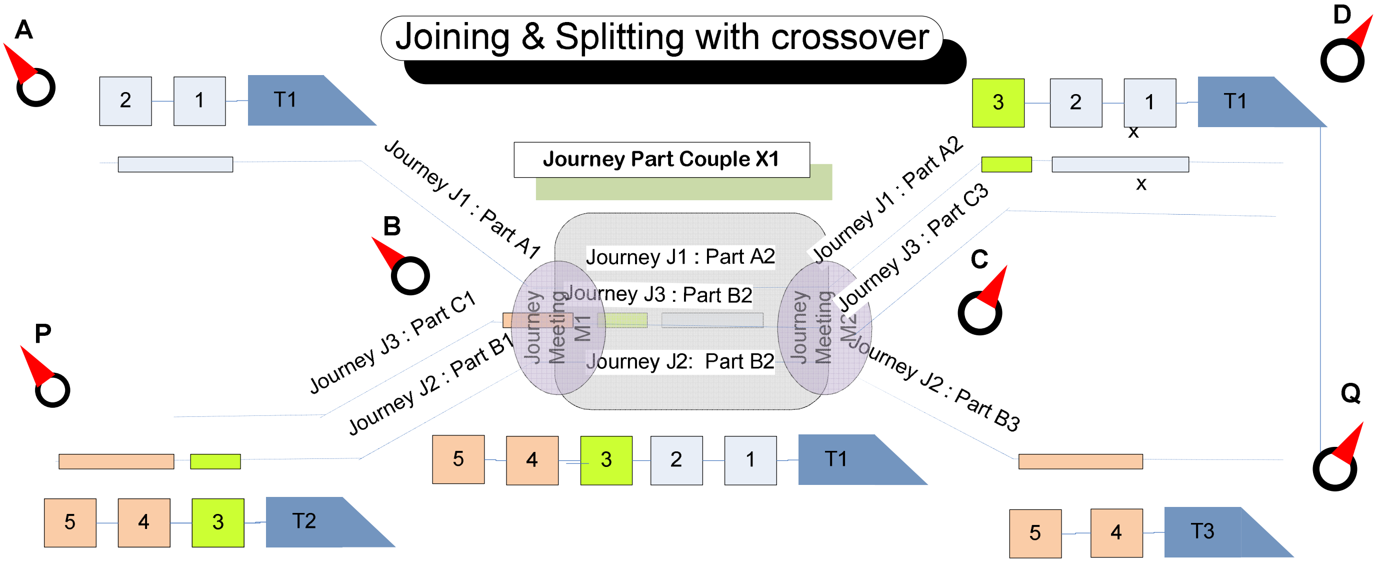


Figure 94 — Example – Joining and splitting Trains

### 7.2.9.2.1 Example of Joining and splitting: Amsterdam to Warsaw

In this example, we consider a train that starts in *Amsterdam* that has parts that run to three separate destinations, *Warsaw*, *Copenhagen* and *Prague*. *Hannover* and *Berlin* are the splitting points and another train (60467) joins the Berlin to Prague leg. There are multiple TRAIN NUMBERs associated with the respective trains

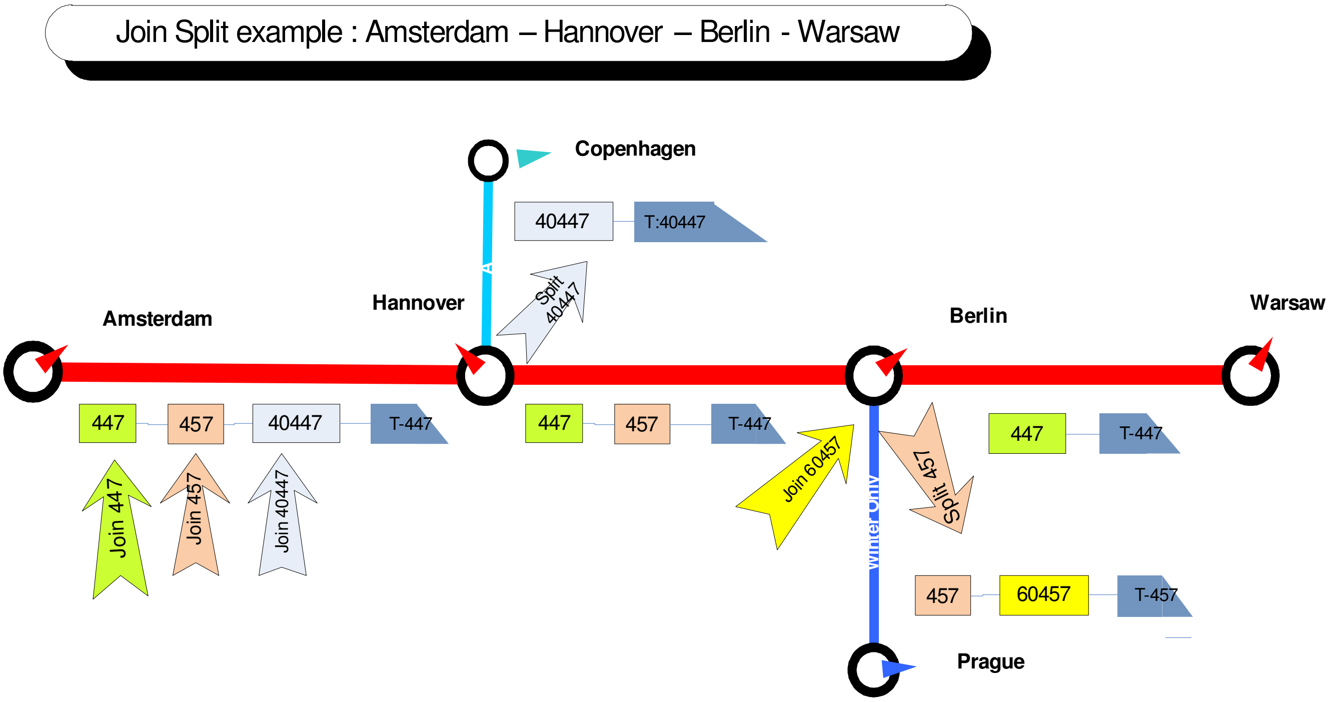


Figure 95 — Example – Joining and splitting: Amsterdam to Warsaw

This can be represented by three SERVICE JOURNEYs, each made up of several JOURNEY PARTs.

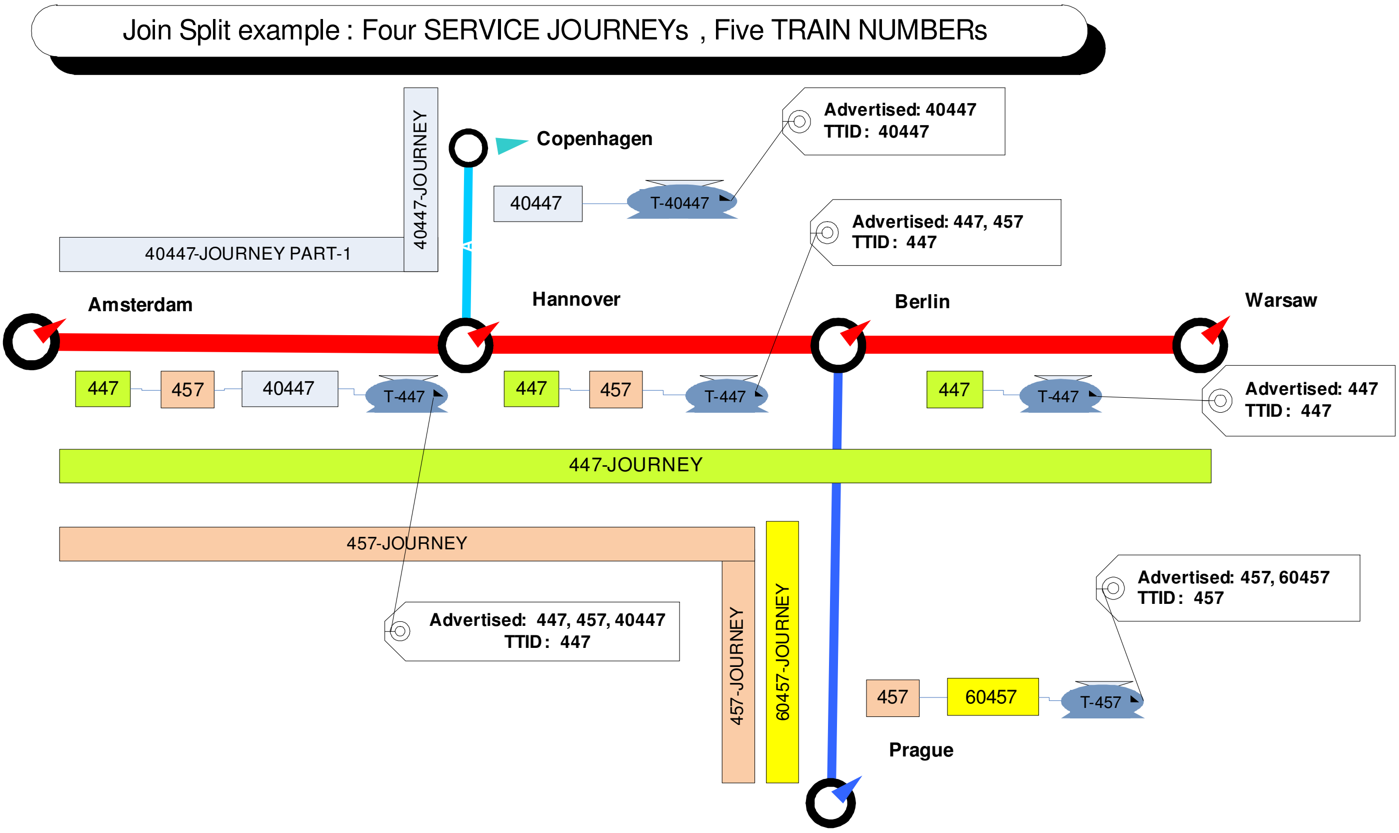


**Figure 96 — Example – Journeys: Amsterdam to Warsaw**

The JOURNEY PARTs are shown in the following diagram.

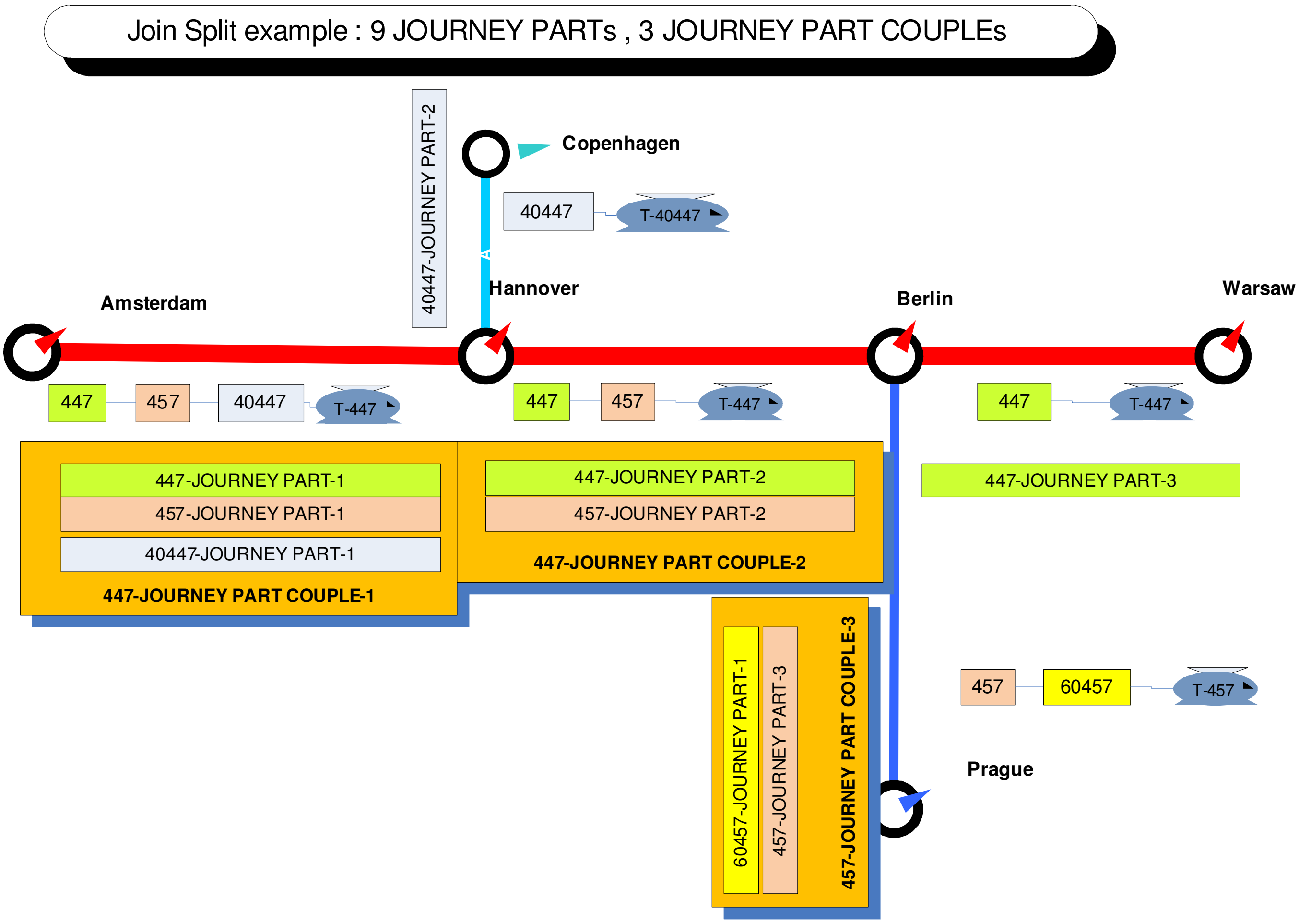


**Figure 97 — Example – Journey Parts: Amsterdam to Warsaw**

The points at which the journeys join or split can be represented by JOURNEY MEETINGs.

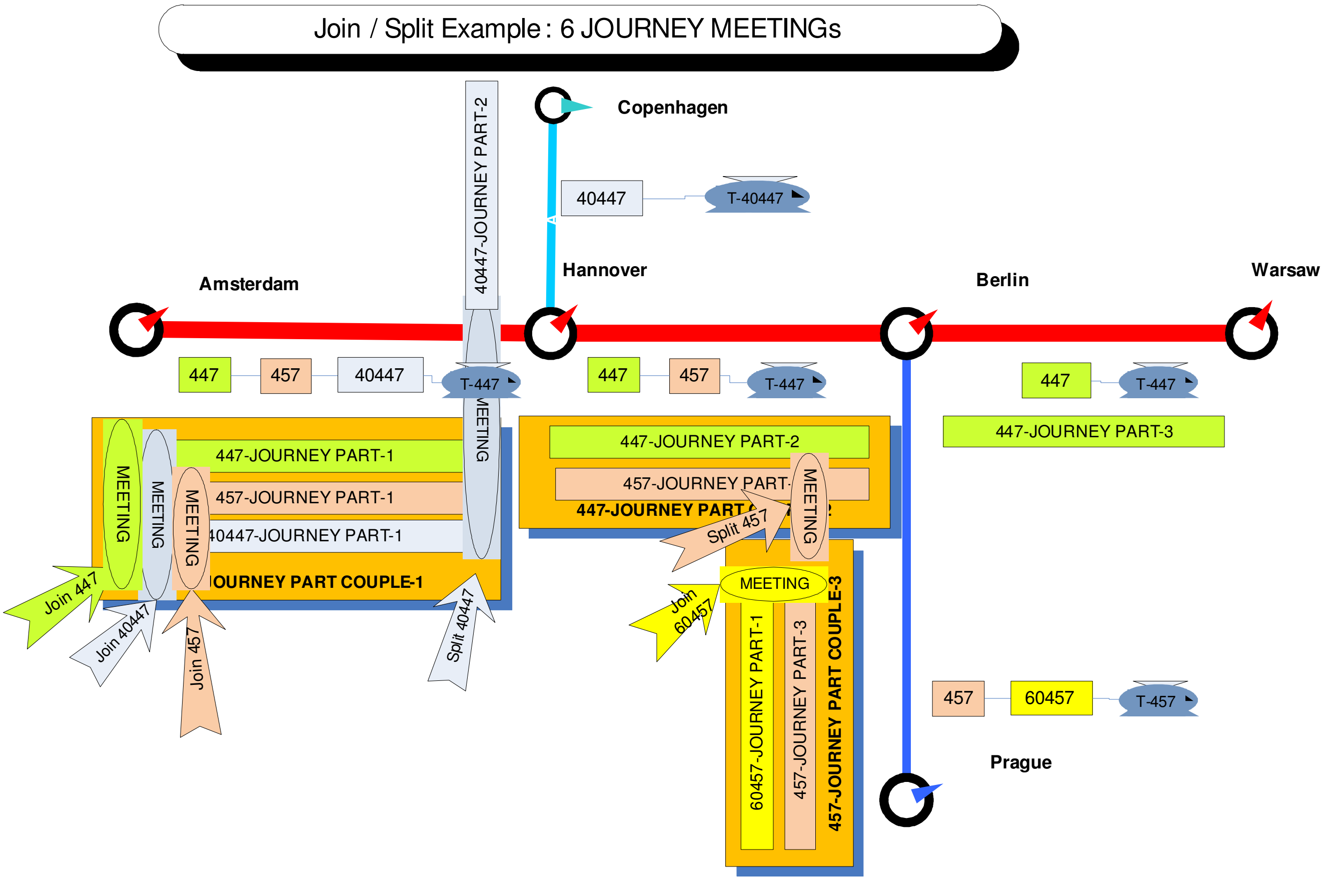


**Figure 98 — Example – Journey Meetings: Amsterdam to Warsaw**

The detailed train makeup of which carriages go to which destinations can be represented with TRAIN, TRAIN COMPONENT and TRAIN ELEMENTs.

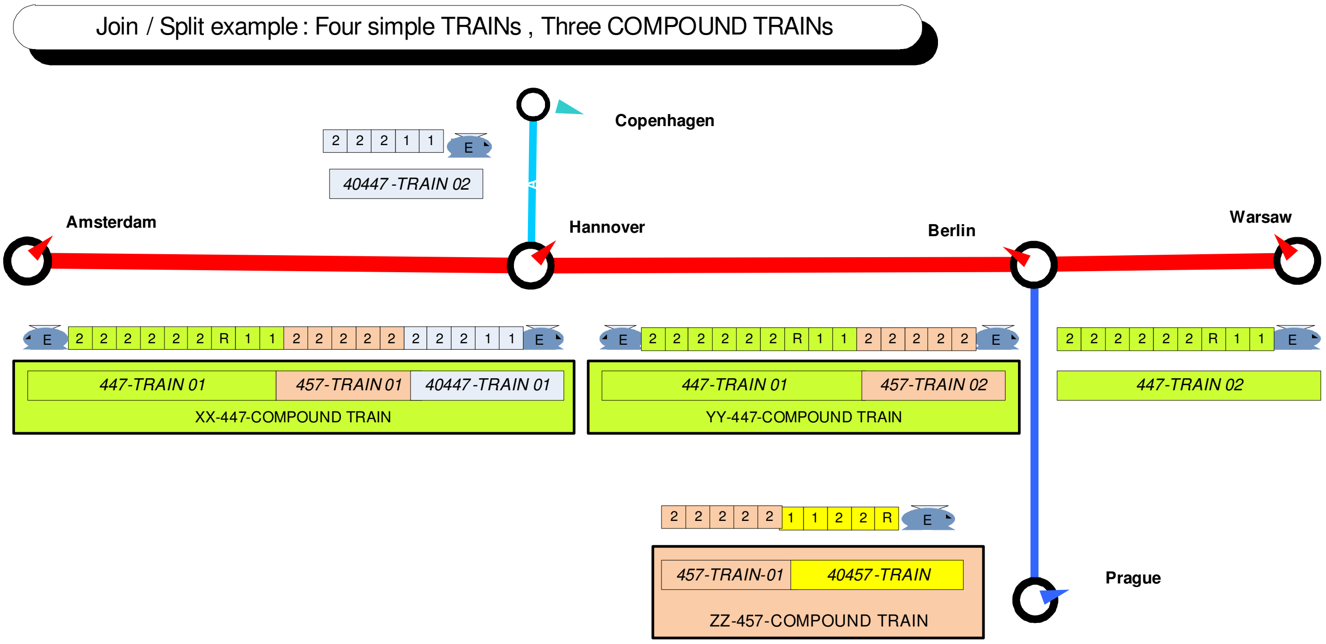


**Figure 99 — Example – Journey Meetings: Amsterdam to Warsaw**

The detailed train makeup can be related to the JOURNEY PARTs using JOURNEY PART COUPLEs and BOCK PARTs. Here we show the Amsterdam to Warsaw elements.

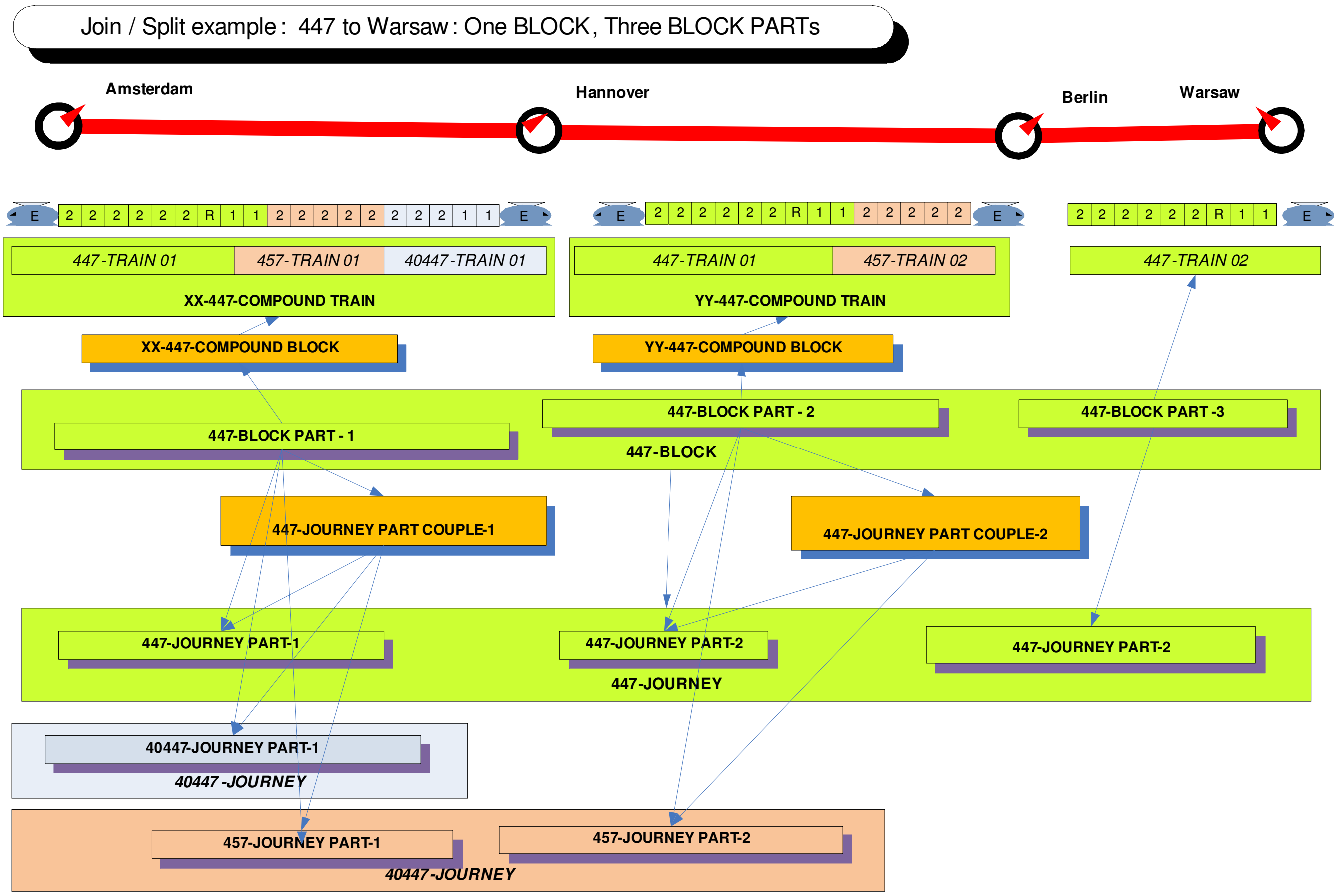


Figure 100 — Example – Train Block Parts and Journey Part Couples: Amsterdam to Warsaw

The detailed train makeup can be related to the JOURNEY PARTs using JOURNEY PART COUPLEs and BOCK PARTs. Here we show the Amsterdam to Prague elements.

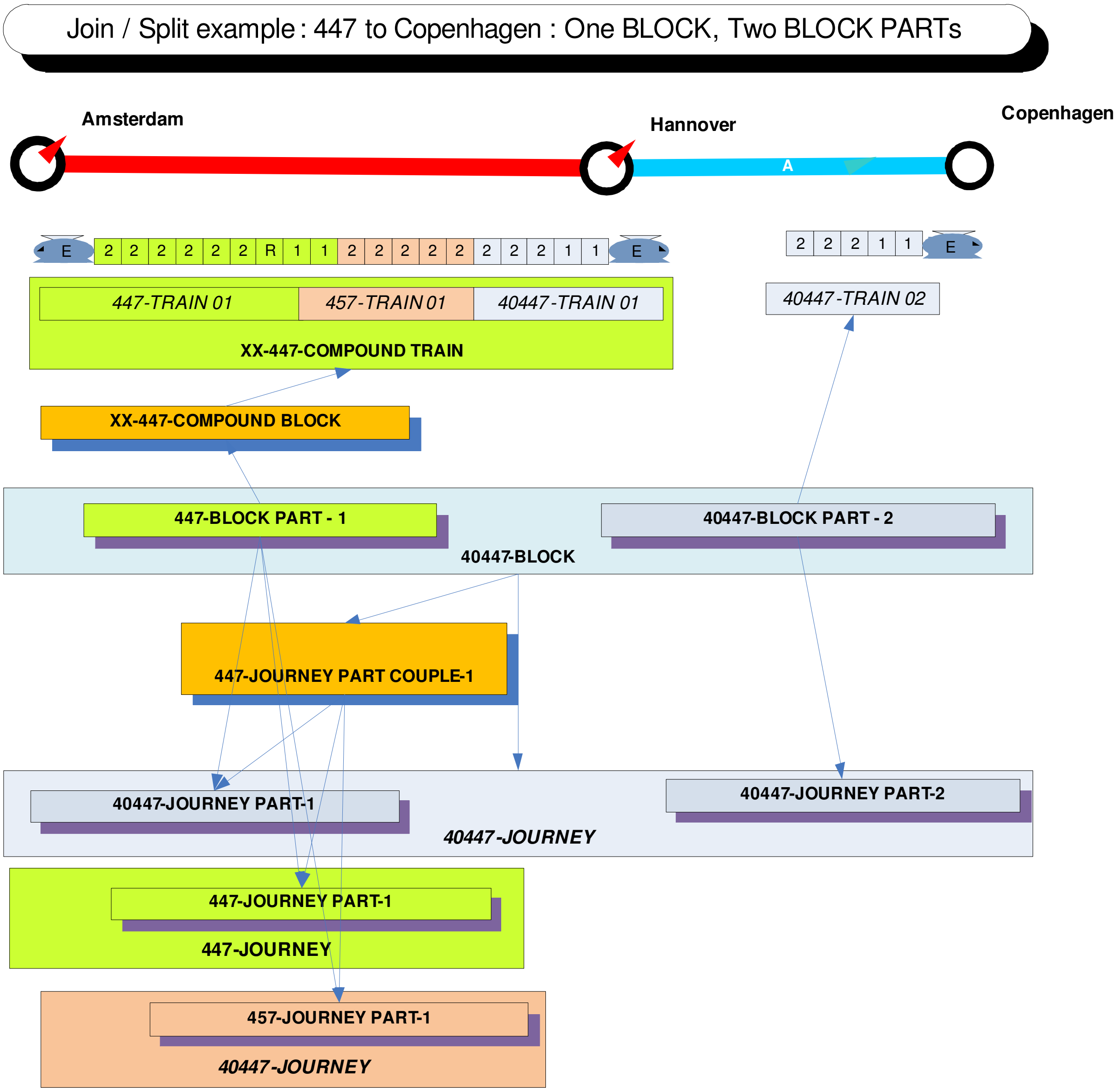


Figure 101 — Example – Train Block Parts and Journey Part Couples: Amsterdam to Copenhagen

### 7.2.9.3 Coupled Journey – Physical Model

The following figure shows the physical model for COUPLED JOURNEYs.

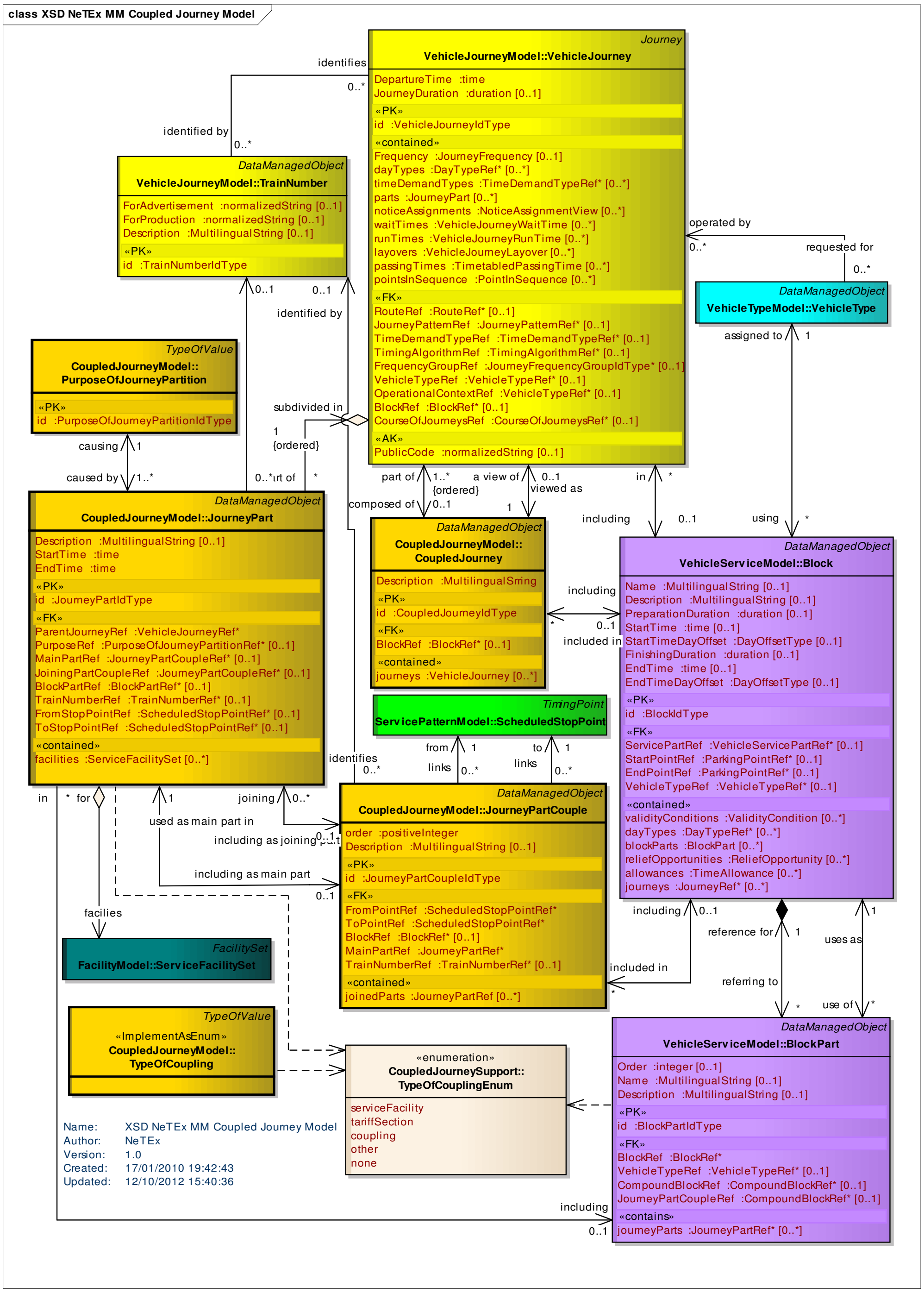

class XSD NeTEx MM Coupled Journey Model
Journey
VehicleJourneyModel::VehicleJourney
DepartureTime :time
JourneyDuration :duration [0..1]
«PK»
id :VehicleJourneyIdType
«contained»
Frequency :JourneyFrequency [0..1]
dayTypes :DayTypeRef* [0..*]
timeDemandTypes :TimeDemandTypeRef* [0..*]
parts :JourneyPart [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
waitTimes :VehicleJourneyWaitTime [0..*]
runTimes :VehicleJourneyRunTime [0..*]
layovers :VehicleJourneyLayover [0..*]
passingTimes :TimetabledPassingTime [0..*]
pointsInSequence :PointInSequence [0..*]
«FK»
RouteRef :RouteRef* [0..1]
JourneyPatternRef :JourneyPatternRef* [0..1]
TimeDemandTypeRef :TimeDemandTypeRef* [0..1]
TimingAlgorithmRef :TimingAlgorithmRef* [0..1]
FrequencyGroupRef :JourneyFrequencyGroupIdType* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
OperationalContextRef :VehicleTypeRef* [0..1]
BlockRef :BlockRef* [0..1]
CourseOfJourneysRef :CourseOfJourneysRef* [0..1]
«AK»
PublicCode :normalizedString [0..1]
identifies
0..*
identified by
0..*
DataManagedObject
VehicleJourneyModel::TrainNumber
ForAdvertisement :normalizedString [0..1]
ForProduction :normalizedString [0..1]
Description :MultilingualString [0..1]
«PK»
id :TrainNumberIdType
0..1
0..1
identified by
operated by
0..*
requested for
0..*
DataManagedObject
VehicleTypeModel::VehicleType
assigned to
1
TypeOfValue
CoupledJourneyModel::
PurposeOfJourneyPartition
«PK»
id :PurposeOfJourneyPartitionIdType
subdivided in
1
{ordered}
causing
1
caused by
1..*
0..*art of
*
part of
1..*
{ordered}
composed of
0..1
a view of
0..1
viewed as
1
in
*
including
0..1
using
*
DataManagedObject
CoupledJourneyModel::JourneyPart
Description :MultilingualString [0..1]
StartTime :time
EndTime :time
«PK»
id :JourneyPartIdType
«FK»
ParentJourneyRef :VehicleJourneyRef*
PurposeRef :PurposeOfJourneyPartitionRef* [0..1]
MainPartRef :JourneyPartCoupleRef* [0..1]
JoiningPartCoupleRef :JourneyPartCoupleRef* [0..1]
BlockPartRef :BlockPartRef* [0..1]
TrainNumberRef :TrainNumberRef* [0..1]
FromStopPointRef :ScheduledStopPointRef* [0..1]
ToStopPointRef :ScheduledStopPointRef* [0..1]
«contained»
facilities :ServiceFacilitySet [0..*]
DataManagedObject
CoupledJourneyModel::
CoupledJourney
Description :MultilingualString
«PK»
id :CoupledJourneyIdType
«FK»
BlockRef :BlockRef* [0..1]
«contained»
journeys :VehicleJourney [0..*]
including
*
0..1
included in
DataManagedObject
VehicleServiceModel::Block
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
PreparationDuration :duration [0..1]
StartTime :time [0..1]
StartTimeDayOffset :DayOffsetType [0..1]
FinishingDuration :duration [0..1]
EndTime :time [0..1]
EndTimeDayOffset :DayOffsetType [0..1]
«PK»
id :BlockIdType
«FK»
ServicePartRef :VehicleServicePartRef* [0..1]
StartPointRef :ParkingPointRef* [0..1]
EndPointRef :ParkingPointRef* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
«contained»
validityConditions :ValidityCondition [0..*]
dayTypes :DayTypeRef* [0..*]
blockParts :BlockPart [0..*]
reliefOpportunities :ReliefOpportunity [0..*]
allowances :TimeAllowance [0..*]
journeys :JourneyRef* [0..*]
TimingPoint
ServicePatternModel::ScheduledStopPoint
from
1
links
0..*
to
1
links
0..*
identifies
0..*
in
* for
1
joining
0..*
used as main part in
including as joining part
0..1
including as main part
0..1
DataManagedObject
CoupledJourneyModel::JourneyPartCouple
order :positiveInteger
Description :MultilingualString [0..1]
«PK»
id :JourneyPartCoupleIdType
«FK»
FromPointRef :ScheduledStopPointRef*
ToPointRef :ScheduledStopPointRef*
BlockRef :BlockRef* [0..1]
MainPartRef :JourneyPartRef*
TrainNumberRef :TrainNumberRef* [0..1]
«contained»
joinedParts :JourneyPartRef [0..*]
facilies
FacilitySet
FacilityModel::ServiceFacilitySet
including
0..1
reference for
1
uses as
1
included in
*
referring to
*
use of
*
TypeOfValue
«ImplementAsEnum»
CoupledJourneyModel::
TypeOfCoupling
«enumeration»
CoupledJourneySupport::
TypeOfCouplingEnum
serviceFacility
tariffSection
coupling
other
none
DataManagedObject
VehicleServiceModel::BlockPart
Order :integer [0..1]
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
«PK»
id :BlockPartIdType
«FK»
BlockRef :BlockRef*
VehicleTypeRef :VehicleTypeRef* [0..1]
CompoundBlockRef :CompoundBlockRef* [0..1]
JourneyPartCoupleRef :CompoundBlockRef* [0..1]
«contains»
journeyParts :JourneyPartRef* [0..*]
including
0..1
Name: XSD NeTEx MM Coupled Journey Model
Author: NeTEx
Version: 1.0
Created: 17/01/2010 19:42:43
Updated: 12/10/2012 15:40:36

Figure 102 — Coupled Journey – Physical Model (UML)

### 7.2.9.3.1 Train Coupling – Physical Model

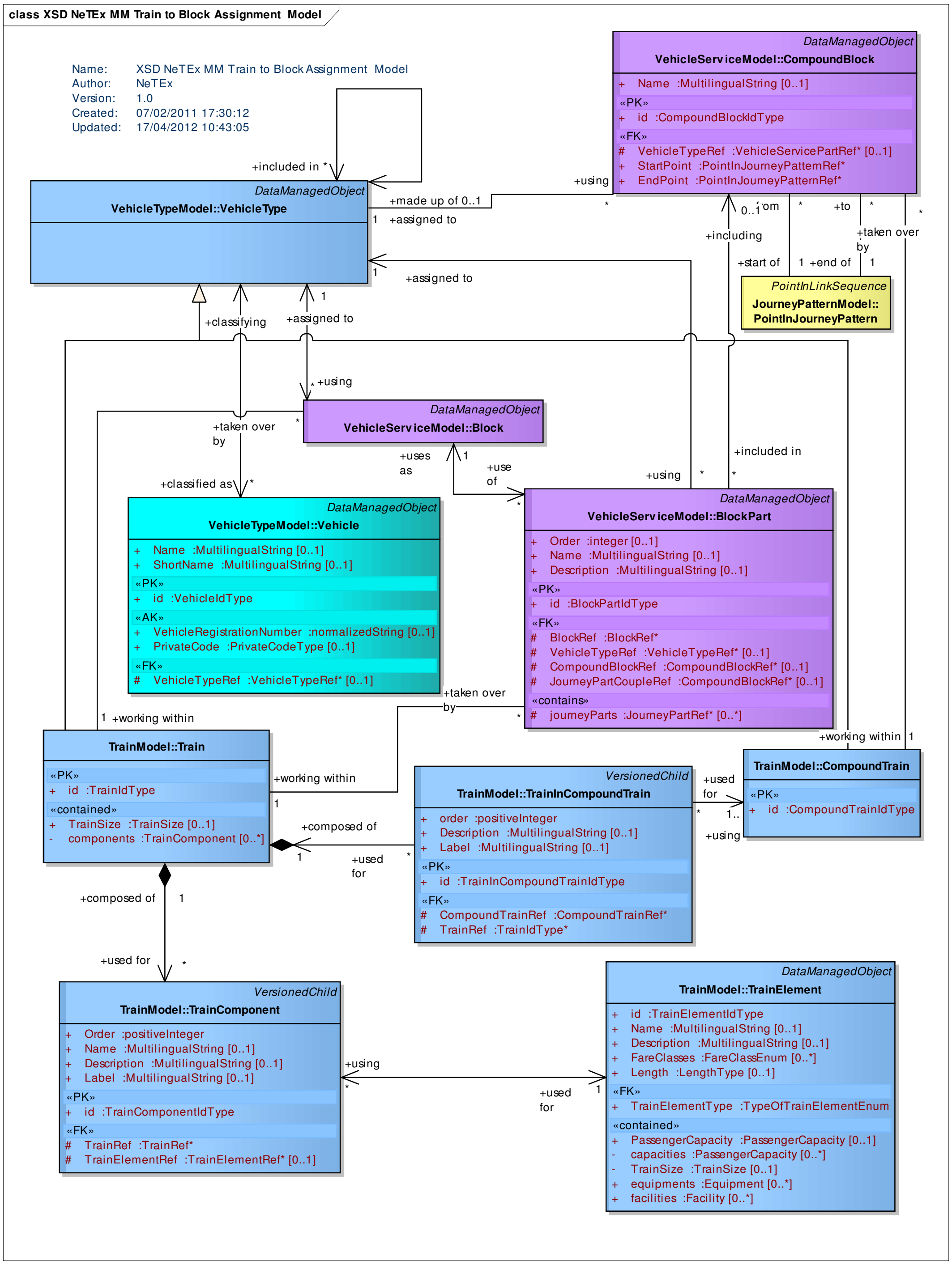

class XSD NeTEx MM Train to Block Assignment Model
Name: XSD NeTEx MM Train to Block Assignment Model
Author: NeTEx
Version: 1.0
Created: 07/02/2011 17:30:12
Updated: 17/04/2012 10:43:05
DataManagedObject
VehicleServiceModel::CompoundBlock
+ Name :MultilingualString [0..1]
«PK»
+ id :CompoundBlockIdType
«FK»
# VehicleTypeRef :VehicleServicePartRef* [0..1]
+ StartPoint :PointInJourneyPatternRef*
+ EndPoint :PointInJourneyPatternRef*
DataManagedObject
VehicleTypeModel::VehicleType
+included in *
+made up of 0..1
1 +assigned to
+using
*
0..1
+from
*
+to
*
+taken over by
*
+including
+start of
1 +end of
1
PointInLinkSequence
JourneyPatternModel::PointInJourneyPattern
1
+assigned to
+classifying
+assigned to
1
* +using
DataManagedObject
VehicleServiceModel::Block
+taken over by
*
+classified as
*
+uses as
1
+use of
+using
*
+included in
*
DataManagedObject
VehicleTypeModel::Vehicle
+ Name :MultilingualString [0..1]
+ ShortName :MultilingualString [0..1]
«PK»
+ id :VehicleIdType
«AK»
+ VehicleRegistrationNumber :normalizedString [0..1]
+ PrivateCode :PrivateCodeType [0..1]
«FK»
# VehicleTypeRef :VehicleTypeRef* [0..1]
DataManagedObject
VehicleServiceModel::BlockPart
+ Order :integer [0..1]
+ Name :MultilingualString [0..1]
+ Description :MultilingualString [0..1]
«PK»
+ id :BlockPartIdType
«FK»
# BlockRef :BlockRef*
# VehicleTypeRef :VehicleTypeRef* [0..1]
# CompoundBlockRef :CompoundBlockRef* [0..1]
# JourneyPartCoupleRef :CompoundBlockRef* [0..1]
«contains»
# journeyParts :JourneyPartRef* [0..*]
+taken over by
*
1 +working within
+working within 1
TrainModel::Train
«PK»
+ id :TrainIdType
«contained»
+ TrainSize :TrainSize [0..1]
- components :TrainComponent [0..*]
+working within
1
+composed of
1
+used for
*
VersionedChild
TrainModel::TrainInCompoundTrain
+ order :positiveInteger
+ Description :MultilingualString [0..1]
+ Label :MultilingualString [0..1]
«PK»
+ id :TrainInCompoundTrainIdType
«FK»
# CompoundTrainRef :CompoundTrainRef*
# TrainRef :TrainIdType*
+used for
*
1..
+using
TrainModel::CompoundTrain
«PK»
+ id :CompoundTrainIdType
+composed of
1
+used for
*
VersionedChild
TrainModel::TrainComponent
+ Order :positiveInteger
+ Name :MultilingualString [0..1]
+ Description :MultilingualString [0..1]
+ Label :MultilingualString [0..1]
«PK»
+ id :TrainComponentIdType
«FK»
# TrainRef :TrainRef*
# TrainElementRef :TrainElementRef* [0..1]
+using
*
+used for
1
DataManagedObject
TrainModel::TrainElement
+ id :TrainElementIdType
+ Name :MultilingualString [0..1]
+ Description :MultilingualString [0..1]
+ FareClasses :FareClassEnum [0..*]
+ Length :LengthType [0..1]
«FK»
+ TrainElementType :TypeOfTrainElementEnum
«contained»
+ PassengerCapacity :PassengerCapacity [0..1]
- capacities :PassengerCapacity [0..*]
- TrainSize :TrainSize [0..1]
+ equipments :Equipment [0..*]
+ facilities :Facility [0..*]

**Figure 103 — Train to Block Assignment – Physical Model (UML)**

## 7.2.9.4 Coupled Journey – Attributes and XSD

### 7.2.9.4.1 JourneyPart – Model Element

A part of a VEHICLE JOURNEY created according to a specific functional purpose, for instance in situations when vehicle coupling or separating occurs.

**Table 55 – *JourneyPart* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | JOURNEY PART inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *JourneyPartIdType* | 1:1 | Identifier of JOURNEY PART. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of a JOURNEY PART. |
| «FK» | ***ParentJourneyRef*** | *VehicleJourneyRef* | 0:1 | Parent VEHICLE JOURNEY to which JOURNEY PART belongs. |
| «FK» | ***MainPartRef*** | *JourneyPartCoupleRef* | 1:1 | REFERENCE to JOURNEY PART COUPLE of main JOURNEY of a JOURNEY PART. |
| «FK» | ***JourneyPart-CoupleRef*** | *JourneyPartCoupleRef* | 0:1 | REFERENCE to JOURNEY PART COUPLE of JOURNEY that JOURNEY PART joins. |
| «FK» | ***TrainNumberRef*** | *TrainNumberRef* | 0:1 | REFERENCE to TRAIN NUMBER of JOURNEY PART |
| «FK» | ***BlockPartRef*** | *BlockPartRef* | 0:1 | REFERENCE to BLOCK PART of JOURNEY that JOURNEY PART joins. |
| «FK» | ***FromStopPointRef*** | *ScheduledStopPointRef* | 0:1 | REFERENCE to from start SCHEDULED STOP POINT for JOURNEY PART. |
| «FK» | ***ToStopPointRef*** | *ScheduledStopPointRef* | 0:1 | REFERENCE to end SCHEDULED STOP POINT for JOURNEY PART. |
| | ***StartTime*** | *xsd:time* | 1:1 | Start time of a JOURNEY PART. |
| | ***EndTime*** | *xsd:time* | 1:1 | End time of a JOURNEY PART. |
| «FK» | ***PurposeOf-JourneyPartition-Ref*** | *PurposeOfJourneyPartitionRef* | 0:1 | PURPOSE of JOURNEY PARTITION. |
| «cntd» | ***facilities*** | *ServiceFacilitySet* | 0:* | Facilities available during a JOURNEY PART. |

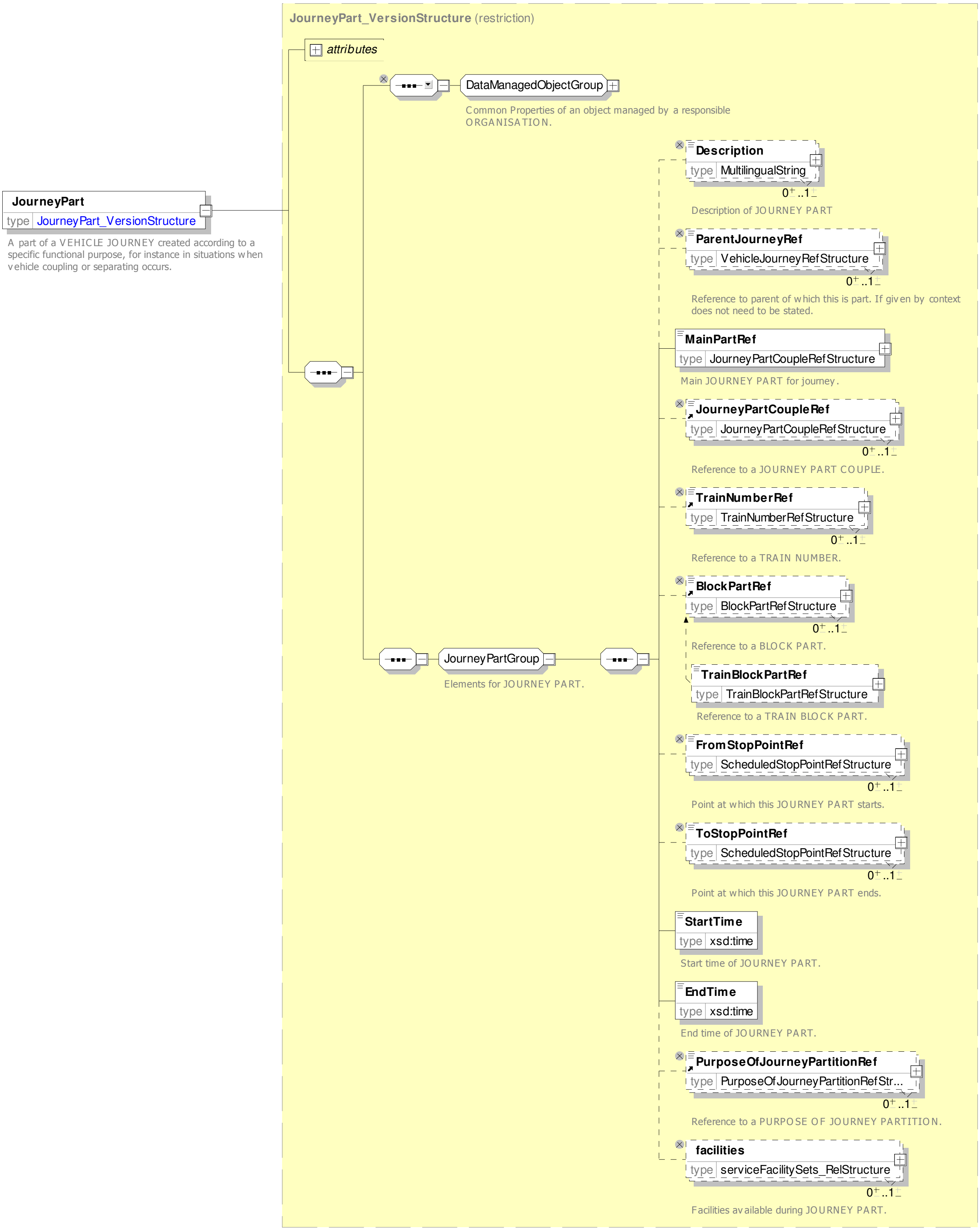

JourneyPart
type JourneyPart_VersionStructure
A part of a VEHICLE JOURNEY created according to a specific functional purpose, for instance in situations when vehicle coupling or separating occurs.
JourneyPart_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
Description
type MultilingualString
0..1
Description of JOURNEY PART
ParentJourneyRef
type VehicleJourneyRefStructure
0..1
Reference to parent of which this is part. If given by context does not need to be stated.
MainPartRef
type JourneyPartCoupleRefStructure
Main JOURNEY PART for journey.
JourneyPartCoupleRef
type JourneyPartCoupleRefStructure
0..1
Reference to a JOURNEY PART COUPLE.
TrainNumberRef
type TrainNumberRefStructure
0..1
Reference to a TRAIN NUMBER.
BlockPartRef
type BlockPartRefStructure
0..1
Reference to a BLOCK PART.
JourneyPartGroup
Elements for JOURNEY PART.
TrainBlockPartRef
type TrainBlockPartRefStructure
Reference to a TRAIN BLOCK PART.
FromStopPointRef
type ScheduledStopPointRefStructure
0..1
Point at which this JOURNEY PART starts.
ToStopPointRef
type ScheduledStopPointRefStructure
0..1
Point at which this JOURNEY PART ends.
StartTime
type xsd:time
Start time of JOURNEY PART.
EndTime
type xsd:time
End time of JOURNEY PART.
PurposeOfJourneyPartitionRef
type PurposeOfJourneyPartitionRefStr...
0..1
Reference to a PURPOSE OF JOURNEY PARTITION.
facilities
type serviceFacilitySets_RelStructure
0..1
Facilities available during JOURNEY PART.

Figure 104 — *JourneyPart* — XSD

### 7.2.9.4.2 CoupledJourney – Model Element

A complete journey operated by a coupled train, composed of two or more VEHICLE JOURNEYs remaining coupled together all along a JOURNEY PATTERN. A COUPLED JOURNEY may be viewed as a single VEHICLE JOURNEY.

**Table 56 – *CoupledJourney* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | COUPLED JOURNEY inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *CoupledJourneyIdType* | 1:1 | Identifier of COUPLED JOURNEY. |
| | ***Description*** | *MultilingualString* | 1:1 | Description of COUPLED JOURNEY. |
| «FK» | ***TrainBlockRef*** | *BlockRef* | 0:1 | BLOCK supplying that COUPLE JOURNEY. |
| «cntd» | ***journeys*** | *VehicleJourney* | 0:* | JOURNEYs linked by of COUPLED JOURNEY. |

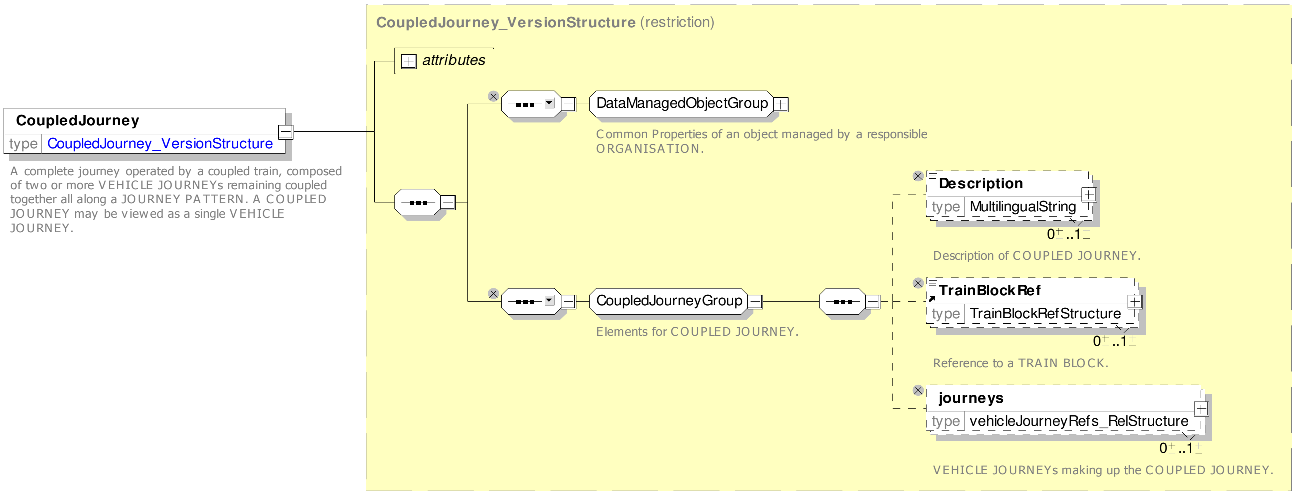

CoupledJourney_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
CoupledJourney
type CoupledJourney_VersionStructure
A complete journey operated by a coupled train, composed of two or more VEHICLE JOURNEYs remaining coupled together all along a JOURNEY PATTERN. A COUPLED JOURNEY may be viewed as a single VEHICLE JOURNEY.
Description
type MultilingualString
0..1
Description of COUPLED JOURNEY.
TrainBlockRef
type TrainBlockRefStructure
0..1
Reference to a TRAIN BLOCK.
CoupledJourneyGroup
Elements for COUPLED JOURNEY.
journeys
type vehicleJourneyRefs_RelStructure
0..1
VEHICLE JOURNEYs making up the COUPLED JOURNEY.

**Figure 105 — *CoupledJourney* — XSD**

### 7.2.9.4.3 JourneyPartCouple – Model Element

Two or more JOURNEY PARTs of different VEHICLE JOURNEYs served simultaneously by a train set up by coupling their single vehicles.

**Table 57 – *JourneyPartCouple* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | JOURNEY PART COUPLE Inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *JourneyPartCoupleIdType* | 1:1 | Identifier of JOURNEY PART COUPLE. |
| | ***order*** | *xsd:positiveInteger* | 1:1 | Order of JOURNEY PART COUPLE within |

| | | | | |
|---|---|---|---|---|
| | | | | JOURNEY. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of JOURNEY PART COUPLE. |
| «FK» | ***FromPointRef*** | *ScheduledStopPointRef* | 1:1 | Reference to POINT at which JOURNEY PART COUPLE starts. |
| «FK» | ***ToPointRef*** | *ScheduledStopPointRef* | 1:1 | Reference to POINT at which JOURNEY PART COUPLE ends. |
| «FK» | ***BlockRef*** | *BlockRef* | 0:1 | Reference to BLOCK associated with JOURNEY PART COUPLE. |
| «FK» | ***MainPartRef*** | *JourneyPartRef* | 1:1 | Main JOURNEY PART associated with JOURNEY PART COUPLE. |
| «cntd» | ***joinedParts*** | *JourneyPartRef* | 0:* | JOURNEY PARTs that JOURNEY PART COUPLE joins. |
| «FK» | ***TrainNumberRef*** | *TrainNumberRef* | 0:1 | REFERENCE to TRAIN NUMBER of JOURNEY PART COUPLE. |

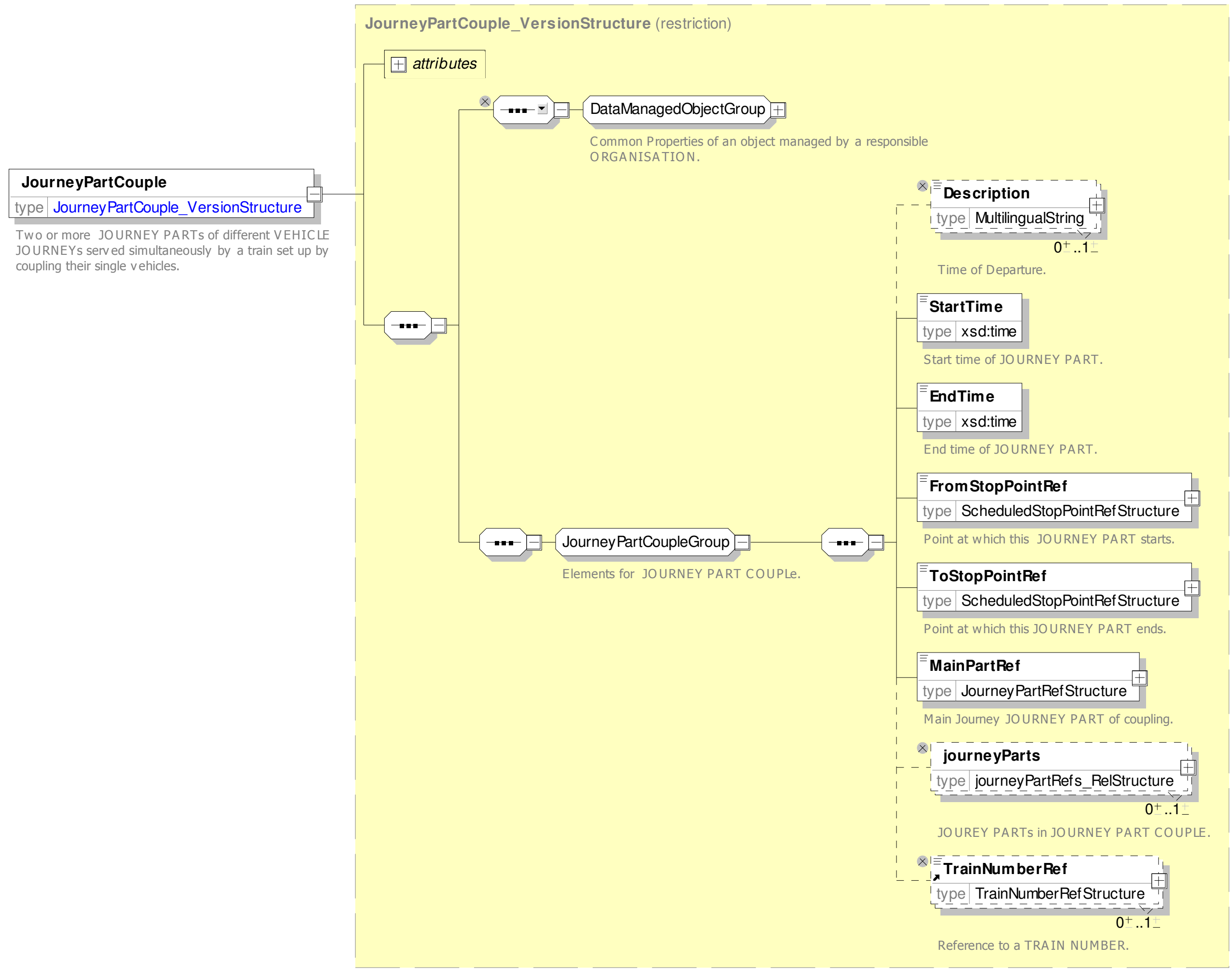

JourneyPartCouple_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
JourneyPartCouple
type JourneyPartCouple_VersionStructure
Two or more JOURNEY PARTs of different VEHICLE JOURNEYs served simultaneously by a train set up by coupling their single vehicles.
Description
type MultilingualString
0..1
Time of Departure.
StartTime
type xsd:time
Start time of JOURNEY PART.
EndTime
type xsd:time
End time of JOURNEY PART.
FromStopPointRef
type ScheduledStopPointRefStructure
Point at which this JOURNEY PART starts.
JourneyPartCoupleGroup
Elements for JOURNEY PART COUPLe.
ToStopPointRef
type ScheduledStopPointRefStructure
Point at which this JOURNEY PART ends.
MainPartRef
type JourneyPartRefStructure
Main Journey JOURNEY PART of coupling.
journeyParts
type journeyPartRefs_RelStructure
0..1
JOUREY PARTs in JOURNEY PART COUPLE.
TrainNumberRef
type TrainNumberRefStructure
0..1
Reference to a TRAIN NUMBER.

Figure 106 — *JourneyPartCouple* — XSD

### 7.2.9.4.4 *PurposeOfJourneyPartition* – Model Element

A classification of JOURNEY PARTITIONs according to their functional purpose.

**Table 58 – *PurposeOfJourneyPartition* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *TypeOfValue* | ::> | PURPOSE OF JOURNEY PARTITION inherits from TYPE OF VALUE |
| «PK» | ***id*** | *PurposeOfJourney-PartitionIdType* | 1:1 | Identifier of PURPOSE OF JOURNEY PARTITION. |

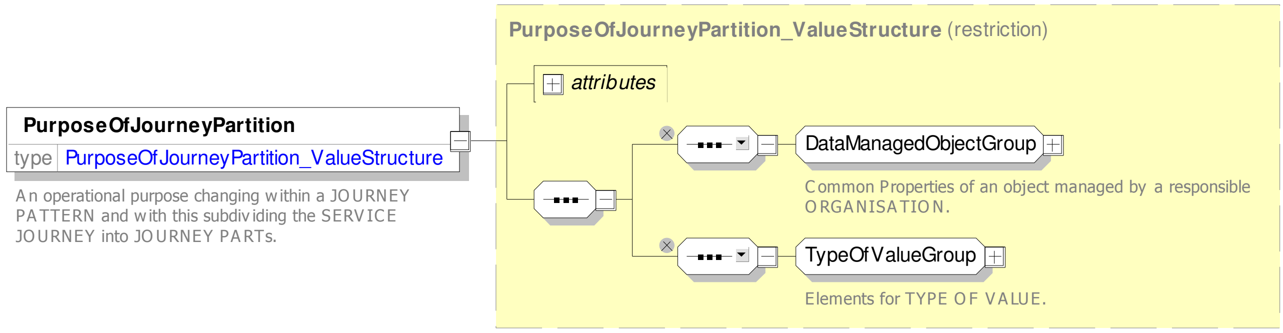

PurposeOfJourneyPartition_ValueStructure (restriction)
attributes
PurposeOfJourneyPartition
type PurposeOfJourneyPartition_ValueStructure
An operational purpose changing within a JOURNEY PATTERN and with this subdividing the SERVICE JOURNEY into JOURNEY PARTs.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
TypeOfValueGroup
Elements for TYPE OF VALUE.

**Figure 107 — *PurposeOfJourneyPartition* — XSD**

### 7.2.9.5 XML Example of Journey Parts & Couple Journeys

The following code fragment shows a SERVICE JOURNEY with three parts**.**

For EXAMPLE

```
<ServiceJourney version="any" id="hde:sj_447">
    <Description>447 Amsterdam to Warsaw</Description>
    <DepartureTime>09:00:00Z</DepartureTime>
    <ServiceJourneyPatternRef version="any" ref="hde:JP_447_amsterdam_warsaw"/>
    <TrainRef version="any" ref="hde:trn_447"/>
    <BlockRef version="any" ref="ops:blk_447_amsterdam-warsaw"/>
    <LineView>
        <PublicCode>447</PublicCode>
        <Name>Amsterdam to Warsaw Express</Name>
        <TransportMode>rail</TransportMode>
    </LineView>
    <JourneyPatternView>
        <RouteRef version="any" ref="myrail:RT_447"/>
        <DirectionType>outbound</DirectionType>
        <DestinationDisplayRef version="any" ref="hde:DST_warsaw"/>
    </JourneyPatternView>
    <parts>
        <JourneyPart version="any" id="hde:jpt_447_01">
            <ParentJourneyRef version="any" ref="hde:sj_447"/>
            <MainPartRef version="any" ref="hde:vjp_447_01"/>
            <JourneyPartCoupleRef version="any" ref="hde:jpc_01_amsterdam-hannover"/>
            <TrainNumberRef version="any" ref="hde:tn_447"/>
            <BlockPartRef version="any" ref="ops:blkpt_447_01_amsterdam-hannover"/>
            <FromStopPointRef version="any" ref="uic:nl_amsterdam"/>
            <ToStopPointRef version="any" ref="uic:de_hannover"/>
            <StartTime>09:00:00Z</StartTime>
            <EndTime>12:00:00Z</EndTime>
            <Description>Amsterdam to Hannover</Description>
            <PurposeOfJourneyPartitionRef
ref="hde:coupling">coupling</PurposeOfJourneyPartitionRef>
```

```
        </JourneyPart>
        <JourneyPart version="any" id="hde:jpt_447_02">
            <ParentJourneyRef version="any" ref="hde:sj_447"/>
            <MainPartRef version="any" ref="hde:JourneyPart:jpt_447_02"/>
            <JourneyPartCoupleRef version="any" ref="hde:jpc_02_hannover-berlin"/>
            <TrainNumberRef version="any" ref="hde:tn_447"/>
            <BlockPartRef version="any" ref="ops:BlockPart:blkpt_447_02_hannover-berlin"/>
            <FromStopPointRef version="any" ref="uic:de_hannover"/>
            <ToStopPointRef version="any" ref="uic:de_berlin"/>
            <StartTime>12:00:00Z</StartTime>
            <EndTime>15:00:00Z</EndTime>
            <Description>Hannover to Berlin</Description>
            <PurposeOfJourneyPartitionRef
ref="hde:coupling">coupling</PurposeOfJourneyPartitionRef>
        </JourneyPart>
        <JourneyPart version="any" id="hde:jpt_447_03">
            <ParentJourneyRef version="any" ref="hde:sj_447"/>
            <MainPartRef version="any" ref="hde:JourneyPart:jpt_447_03"/>
            <TrainNumberRef version="any" ref="hde:tn_447"/>
            <BlockPartRef version="any" ref="ops:blkpt_447_03_berlin-warsaw"/>
            <FromStopPointRef version="any" ref="uic:de_berlin"/>
            <ToStopPointRef version="any" ref="uic:pl_warsaw"/>
            <StartTime>15:10:00.0Z</StartTime>
            <EndTime>19:00:00Z</EndTime>
            <Description>Berlin to Warsaw</Description>
            <PurposeOfJourneyPartitionRef
ref="hde:coupling">coupling</PurposeOfJourneyPartitionRef>
        </JourneyPart>
    </parts>
    <calls>
        <Call id="hde:sj_447_001" version="any" order="1">
            <ScheduledStopPointRef version="any" ref="uic:nl_amsterdam"/>
            <Arrival>
                <ForAlighting>false</ForAlighting>
            </Arrival>
            <Departure>
                <Time>09:00:00.0Z</Time>
                <ForBoarding>true</ForBoarding>
                <journeyMeetings>
                    <JourneyMeetingView>
                        <JourneyMeetingRef version="any"
ref="hde:jm_457_joinTo_447_dep_amsterdam"/>
                        <Description>Join with 457 </Description>
                        <EarliestTime>09:00:00Z</EarliestTime>
                        <Reason>joining</Reason>
                    </JourneyMeetingView>
                    <JourneyMeetingView>
                    <JourneyMeetingRef version="any" ref="hde:jm_40447_joinTo_477_dep_amsterdam"/>
                        <Description>Join with 40447</Description>
                        <EarliestTime>09:00:00Z</EarliestTime>
                        <Reason>joining</Reason>
                    </JourneyMeetingView>
                </journeyMeetings>
                    <ServiceJourneyInterchangeView>
                        <ServiceJourneyInterchangeRef version="any"
ref="hde:sji_sj_447_to_sj_457"/>
                        <Description>In Amsterdam Trains join </Description>
                        <StaySeated>true</StaySeated>
                        <Planned>true</Planned>
                        <Guaranteed>true</Guaranteed>
                        <Advertised>true</Advertised>
                        <Controlled>true</Controlled>
                        <ConnectingJourneyView>
                            <ServiceJourneyRef version="any" ref="hde:sj_457"/>
                            <ConnectingVisitNumber>1</ConnectingVisitNumber>
                        </ConnectingJourneyView>
                        <StandardTransferTime>PT0M</StandardTransferTime>
                    </ServiceJourneyInterchangeView>
                </interchanges>          -->

                <QuayAssignmentView>
                    <StopPlaceRef version="any" ref="uic:nl_amsterdam"/>
                    <QuayRef version="any" ref="uic:nl_amsterdam_1"/>
                    <QuayName>Platform 1</QuayName>
                </QuayAssignmentView>
            </Departure>
```

```xml
            <DestinationDisplayRef version="any" ref="hde:DST_hannover"/>
        </Call>
        <Call id="hde:sj_447_002" version="any" order="2">
            <ScheduledStopPointRef version="any" ref="uic:de_hannover"/>
            <Arrival>
                <Time>12:00:00.0Z</Time>
                    <ServiceJourneyInterchange id="hde:sji_sj_447_to_sj_40447">
                        <Description>In Amsterdam Carriages split to Copenhagen </Description>
                        <StaySeated>true</StaySeated>
                        <Planned>true</Planned>
                        <Guaranteed>true</Guaranteed>
                        <Advertised>true</Advertised>
                        <Controlled>true</Controlled>
                        <ConnectingJourneyView>
                            <ServiceJourneyRef version="any" ref="hde:sj_457"/>
                            <ConnectingVisitNumber>1</ConnectingVisitNumber>
                        </ConnectingJourneyView>
                        <StandardTransferTime>PT0M</StandardTransferTime>
                    </ServiceJourneyInterchange>
            </Arrival>
            <Departure>
                <Time>12:05:00.0Z</Time>
                <ForBoarding>true</ForBoarding>
                <WaitTime>PT5M</WaitTime>
                <QuayAssignmentView>
                    <QuayRef version="any" ref="uic:de_hannover_5"/>
                    <QuayName>Platform 5</QuayName>
                </QuayAssignmentView>
            </Departure>
            <DestinationDisplayRef version="any" ref="hde:DST_warsaw"/>
        </Call>
        <Call id="hde:sj_447_003" version="any" order="3">
            <ScheduledStopPointRef version="any" ref="uic:de_berlin"/>
            <Arrival>
                <Time>15:00:00.0Z</Time>
            </Arrival>
            <Departure>
                <Time>15:10:00.0Z</Time>
                <ForBoarding>true</ForBoarding>
                <WaitTime>PT10M</WaitTime>
                <QuayAssignmentView>
                    <QuayRef version="any" ref="uic::de_berlin_3"/>
                    <QuayName>Platform 3</QuayName>
                </QuayAssignmentView>
            </Departure>
            <DestinationDisplayRef version="any" ref="hde:DST_warsaw"/>
        </Call>
        <Call id="hde:sj_447_004" version="any" order="4">
            <ScheduledStopPointRef version="any" ref="uic:pl_warsaw"/>
            <Arrival>
                <Time>19:10:00.0Z</Time>
                <QuayAssignmentView>
                    <QuayRef version="any" ref="uic:pl_warsaw_8"/>
                    <QuayName>Platform 8</QuayName>
                </QuayAssignmentView>
            </Arrival>
            <Departure>
                <ForBoarding>false</ForBoarding>
            </Departure>
        </Call>
    </calls>
</ServiceJourney>
```

#### 7.2.9.5.1 PurposeOfJourneyPartition – Model Element

An operational purpose to change the characteristic within a JOURNEY PATTERN with subdividing the SERVICE JOURNEY into JOURNEY PARTs.

#### 7.2.9.5.2 TypeOfCoupling – Model Element

An operational purpose to change the characteristic within a JOURNEY PATTERN with subdividing the SERVICE JOURNEY into JOURNEY PARTs.

### 7.2.10 Flexible Service

#### 7.2.10.1 FLEXIBLE SERVICE – Conceptual MODEL

Flexible transport (i.e. Demand Responsive Transport) is characterised by flexible routing and scheduling. Flexible routing is described in the NeTEx Part 1 network topology chapter.

Flexible services can operated on regular line topologies or on a flexible topology. They are defined by extending the regular services with additional flexibility information, through the TYPE OF FLEXIBLE JOURNEY object. If this information is attached to a SERVICE JOURNEY (through FLEXIBLE JOURNEY PROPERTIEs) the all service is flexible. But if it is attached to POINT IN JOURNEY PATTERN, the SERVICE JOURNEY having a FLEXIBLE JOURNEY PROPERTIES will only be flexible on the indicated point (usually these POINTs IN JOURNEY PATTERN will also be SCHEDULED STOP POINTs). This can be used either for partially flexible service or for "mixed types of flexible service" inside the same JOURNEY.

Note that NeTEx point of view on flexible services is a passenger information point of view, not an operator point of view.

Several types of flexible services are available:

- Fixed PASSING TIMEs (meaning schedules passing time: there is a timetable, but the service will only run under condition, mainly sufficient demand).

- Dynamic PASSING TIMEs.

- Fixed HEADWAY FREQUENCY (in this case, a maximum waiting time is available through a HEADWAY JOURNEY GROUP, but no passing times are defined, all is done dynamically depending on the demand).

A ***NotFlexible*** value indicates whether a Stop (i.e. POINT IN JOURNEY PATTERN) is flexible or not.

Two additional properties can also be supplied:

1) Whether cancellation is possible or not, even after booking (meaning that the Operator can decide to cancel a service or a stop, usually because there is not enough demand, or the service is too busy.

2) Whether the PASSING TIME and place may be updated or not, even after booking (usually passing times update to optimise the service).

When a service is flexible, there may be no VEHICLE SERVICE scheduled, for example when they are only "on demand". In this situation a NeTEx dataset will provide a "ghost" VEHICLE SERVICE, being a regular one with FLEXIBLE JOURNEY PROPERTIES and no passing times (i.e. no attached CALLs). This is required in order to provide the possible VALIDITY CONDITIONS, DAY TYPEs, CALENDAR, etc. attached to this FLEXIBLE SERVICE.

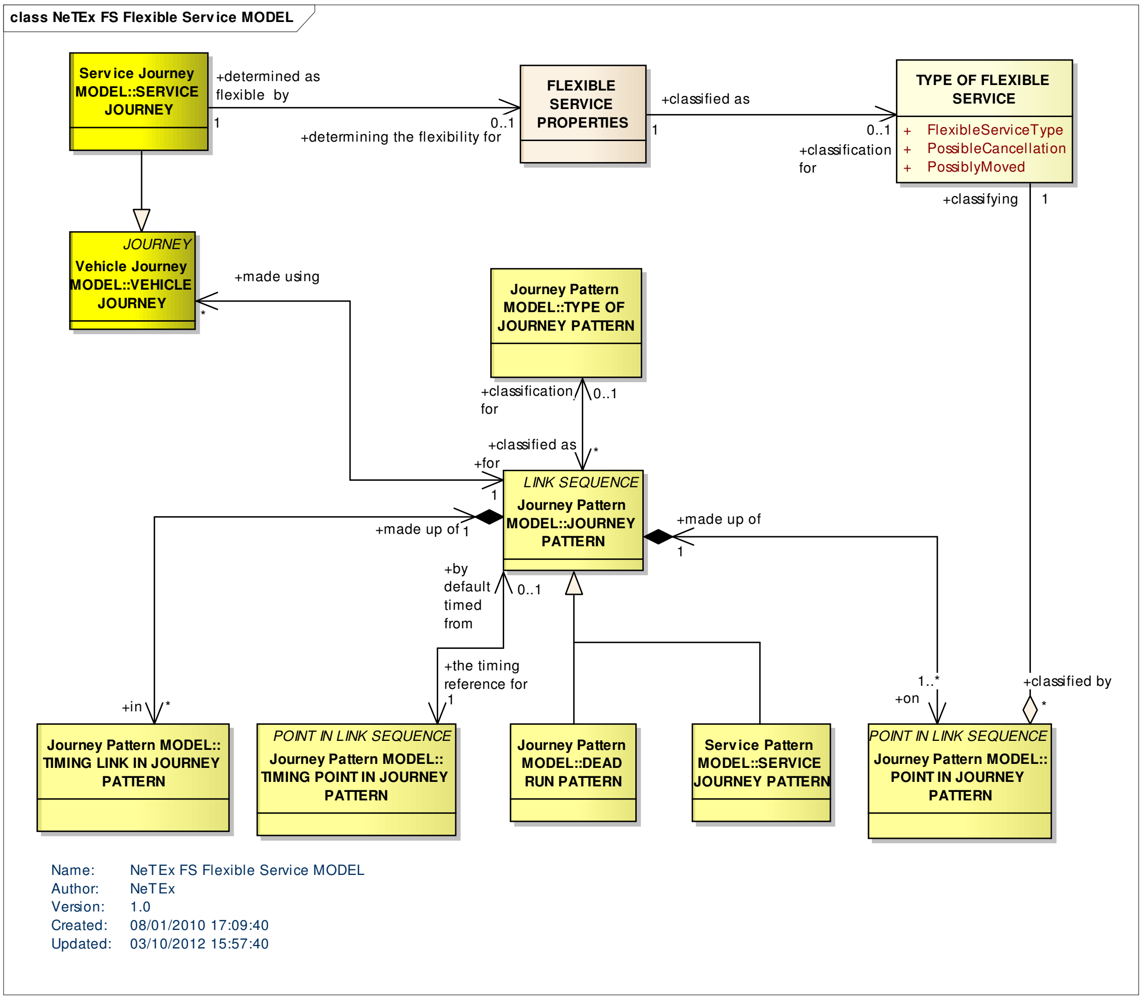

class NeTEx FS Flexible Service MODEL
Service Journey MODEL::SERVICE JOURNEY
+determined as flexible by
1
0..1
+determining the flexibility for
FLEXIBLE SERVICE PROPERTIES
+classified as
1
0..1
+classification for
TYPE OF FLEXIBLE SERVICE
+ FlexibleServiceType
+ PossibleCancellation
+ PossiblyMoved
+classifying
1
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
+made using
*
Journey Pattern MODEL::TYPE OF JOURNEY PATTERN
+classification for
0..1
+classified as
*
+for
1
LINK SEQUENCE
Journey Pattern MODEL::JOURNEY PATTERN
+made up of
1
+made up of
1
+by default timed from
0..1
+the timing reference for
1
+in
*
1..*
+on
+classified by
*
Journey Pattern MODEL:: TIMING LINK IN JOURNEY PATTERN
POINT IN LINK SEQUENCE
Journey Pattern MODEL:: TIMING POINT IN JOURNEY PATTERN
Journey Pattern MODEL::DEAD RUN PATTERN
Service Pattern MODEL::SERVICE JOURNEY PATTERN
POINT IN LINK SEQUENCE
Journey Pattern MODEL:: POINT IN JOURNEY PATTERN
Name: NeTEx FS Flexible Service MODEL
Author: NeTEx
Version: 1.0
Created: 08/01/2010 17:09:40
Updated: 03/10/2012 15:57:40

**Figure 108 — Flexible Service – Conceptual MODEL (UML)**

### 7.2.10.2 Flexible Service Example

There are numerous examples of flexible services. One of them is the TAD 118 (Tisséo, Toulouse area, France). The figure below provides a schematic map of this flexible service: the red points show the possible stops of the TAD 118 (depending on the demand) and the orange and green lines show the regular service (not flexible) lines.

The TAD 118 services start at 6h20 and end at 20h50 with a journey starting every 30 minutes at Colomiers Gare SNCF (if ther is sufficient demand). This is an example of FLEXIBLE SERVICE with fixed HEADWAY FREQUENCY (the line topology being a Flexible zone with fixed stops, see NeTEx Part 1 - 8.4.8.1 Flexible Network Introduction). In this example cancellation is not possible and PASSING TIME and stop place are not changed after booking.

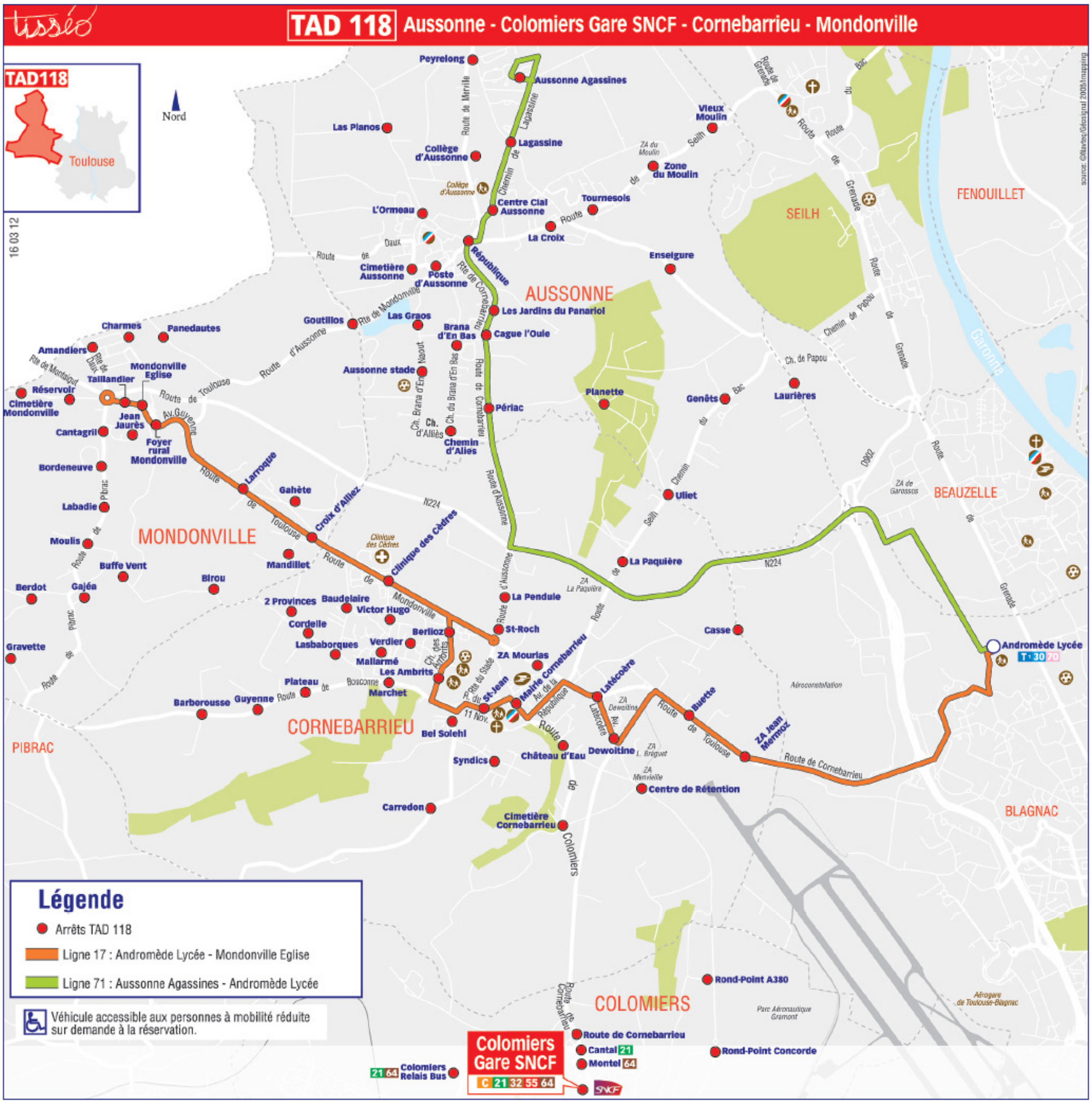


**Figure 109 —Flexible service example: Tisseo TAD 118** *(with courtesy of Tisseo, data from August 2012)*

### 7.2.10.2.1 Flexible Service – Type of Flexible Service

The type of Flexible service may be specified on the Journey Pattern.

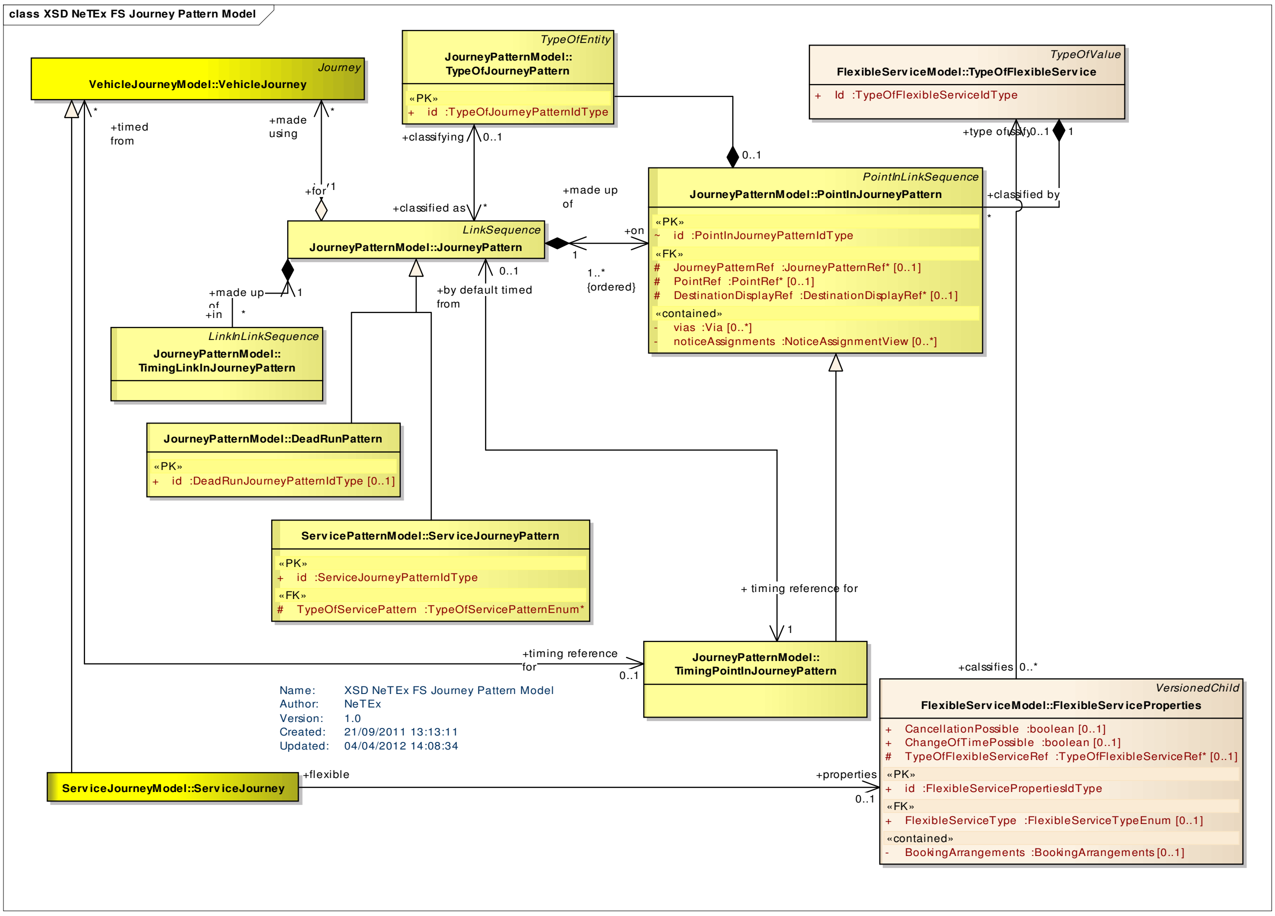

class XSD NeTEx FS Journey Pattern Model
Journey
VehicleJourneyModel::VehicleJourney
TypeOfEntity
JourneyPatternModel::
TypeOfJourneyPattern
«PK»
+ id :TypeOfJourneyPatternIdType
TypeOfValue
FlexibleServiceModel::TypeOfFlexibleService
+ Id :TypeOfFlexibleServiceIdType
+timed
from
+made
using
*
*
+classifying
0..1
+type ofassify
0..1
1
0..1
+for
1
+classified as
*
+made up
of
PointInLinkSequence
JourneyPatternModel::PointInJourneyPattern
+classified by
*
«PK»
~ id :PointInJourneyPatternIdType
«FK»
# JourneyPatternRef :JourneyPatternRef* [0..1]
# PointRef :PointRef* [0..1]
# DestinationDisplayRef :DestinationDisplayRef* [0..1]
«contained»
- vias :Via [0..*]
- noticeAssignments :NoticeAssignmentView [0..*]
LinkSequence
JourneyPatternModel::JourneyPattern
+on
1
1..*
{ordered}
0..1
+by default timed
from
+made up
of
1
+in
*
LinkInLinkSequence
JourneyPatternModel::
TimingLinkInJourneyPattern
JourneyPatternModel::DeadRunPattern
«PK»
+ id :DeadRunJourneyPatternIdType [0..1]
ServicePatternModel::ServiceJourneyPattern
«PK»
+ id :ServiceJourneyPatternIdType
«FK»
# TypeOfServicePattern :TypeOfServicePatternEnum*
+ timing reference for
1
+timing reference
for
0..1
JourneyPatternModel::
TimingPointInJourneyPattern
+calssifies
0..*
VersionedChild
FlexibleServiceModel::FlexibleServiceProperties
+ CancellationPossible :boolean [0..1]
+ ChangeOfTimePossible :boolean [0..1]
# TypeOfFlexibleServiceRef :TypeOfFlexibleServiceRef* [0..1]
«PK»
+ id :FlexibleServicePropertiesIdType
«FK»
+ FlexibleServiceType :FlexibleServiceTypeEnum [0..1]
«contained»
- BookingArrangements :BookingArrangements [0..1]
Name: XSD NeTEx FS Journey Pattern Model
Author: NeTEx
Version: 1.0
Created: 21/09/2011 13:13:11
Updated: 04/04/2012 14:08:34
ServiceJourneyModel::ServiceJourney
+flexible
+properties
0..1

**Figure 110 — Type of Flexible Service – Physical Model (UML)**

### 7.2.10.2.2 Flexible Service – Physical Model details

The physical model introduces a set of attributes through two main nested elements:

- BOOKING ARRANGEMENT describing, for on demand service, the type of booking (online, CallOffice, CallDriver, etc.), how long in advance of departure day or time a service may be ordered, etc.

- CONTACT DETAILs providing URL, phone number, mail, etc. in order to book the service or to get information.

The following figure shows service and network elements relating to FLEXIBLE services.

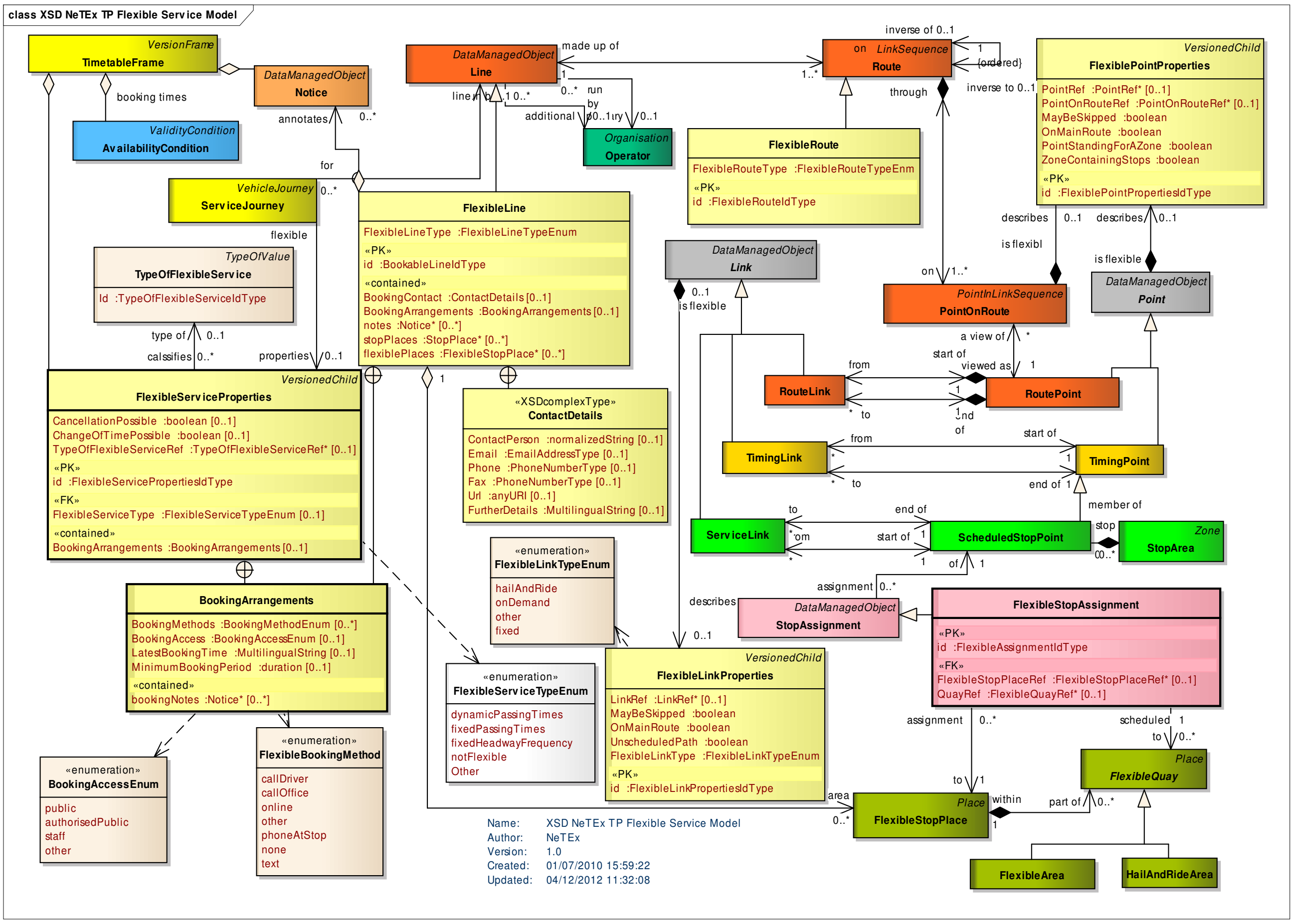


Figure 111 — Flexible Service – Physical Model (UML)

### 7.2.10.3 Flexible Service – Attributes and XSD

#### 7.2.10.3.1 FlexibleServiceProperties – Model Element

Additional Properties of a FLEXIBLE or demand responsive (DRT) service. A service may be part fixed, part flexible or completely flexible.

**Table 59 — FlexibleServiceProperties – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | FLEXIBLE SERVICE PROPERTIES inherits from DATA MANAGED OBJECT |
| «PK» | ***id*** | *FlexibleService-PropertiesIdType* | 1:1 | Identifier of FLEXIBLE SERVICE PROPERTIES. |
| «FK» | ***TypeOfFlexible-ServiceRef*** | TypeOf*FlexibleServiceRef* | 1:1 | TYPE OF FLEXIBLE SERVICE for which these are the properties. |
| «cntd» | ***Booking-Arrangements*** | *BookingArrangements* | 0:1 | Booking contact for FLEXIBLE SERVICE. |
| | ***FlexibleService-Type*** | *FlexibleServiceTypeEnum* | 0:1 | Flexible service type is FixedPassingTimes/DynamicPassingTimes/FixedHeadwayFrequency (in the last value, this provides a maximum waiting time, but no passing time is defined, all is done dynamically depending on the demand). A NotFlexible value is probably also required to clearly state that a Stop (i.e. Point in JOURNEY PATTERN) is not flexible when others are. |
| | ***Cancellation-Possible*** | *xsd:boolean* | 0:1 | Whether cancellation is always possible (meaning the Operator can decided to cancel a journey, usually because there are not enough passenger, or they are too busy to run the service.) |
| | ***ChangeOfTime-Possible*** | *xsd:boolean* | 0:1 | PossiblyMoved flag tells that it may be moved (usually passing time update to optimise the service) (need for both PossibleCancelation and PossiblyMoved since both can be true... these 2 attributes being really need only forDynamicPassingTimes ). |

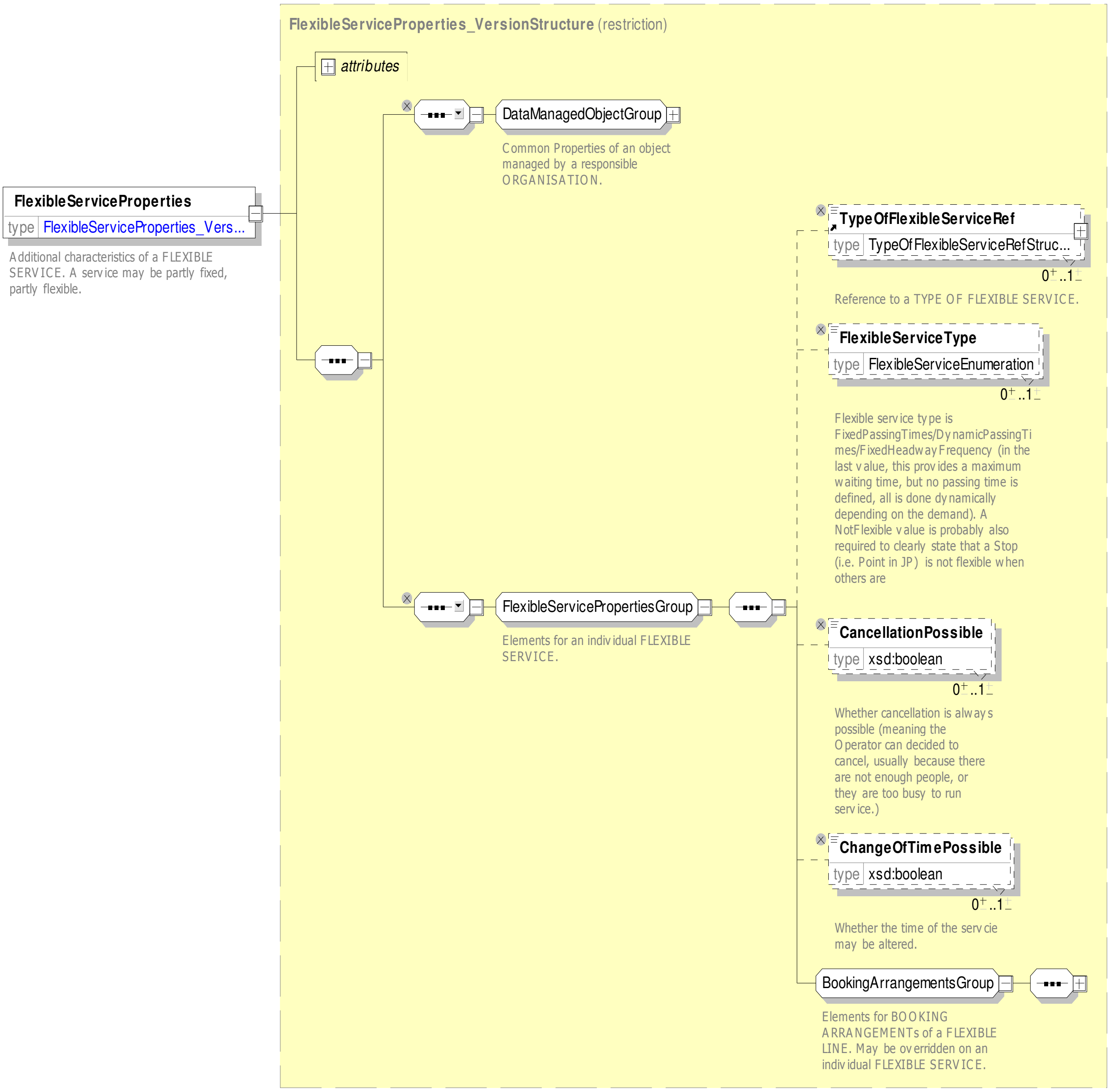

FlexibleServiceProperties_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
FlexibleServiceProperties
type FlexibleServiceProperties_Vers...
Additional characteristics of a FLEXIBLE SERVICE. A service may be partly fixed, partly flexible.
TypeOfFlexibleServiceRef
type TypeOfFlexibleServiceRefStruc...
0..1
Reference to a TYPE OF FLEXIBLE SERVICE.
FlexibleServiceType
type FlexibleServiceEnumeration
0..1
Flexible service type is FixedPassingTimes/DynamicPassingTimes/FixedHeadwayFrequency (in the last value, this provides a maximum waiting time, but no passing time is defined, all is done dynamically depending on the demand). A NotFlexible value is probably also required to clearly state that a Stop (i.e. Point in JP) is not flexible when others are
FlexibleServicePropertiesGroup
Elements for an individual FLEXIBLE SERVICE.
CancellationPossible
type xsd:boolean
0..1
Whether cancellation is always possible (meaning the Operator can decided to cancel, usually because there are not enough people, or they are too busy to run service.)
ChangeOfTimePossible
type xsd:boolean
0..1
Whether the time of the servcie may be altered.
BookingArrangementsGroup
Elements for BOOKING ARRANGEMENTs of a FLEXIBLE LINE. May be overridden on an individual FLEXIBLE SERVICE.

**Figure 112 — *FlexibleServiceProperties* — XSD**

### 7.2.10.3.2 FlexibleStopAssignment – Model Element

The allocation of a SCHEDULED STOP POINT (i.e. a STOP POINT of a SERVICE PATTERN or JOURNEY PATTERN) to a specific FLEXIBLE STOP PLACE, and also possibly a FLEXIBLE AREA or HAIL AND RIDE AREA. May be subject to a VALIDITY CONDITION.

**Table 60 – *FlexibleStopAssignment* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *StopAssignment* | ::> | FLEXIBLE STOP ASSIGNMENT inherits from STOP ASSIGNMENT. |

|  |  |  |  |  |
|---|---|---|---|---|
| «PK» | ***id*** | *FlexibleAssignmentIdType* | 1:1 | Identifier of a FLEXIBLE STOP ASSIGNMENT. |
| «FK» | ***ScheduledStop-PointRef*** | *ScheduledStopPointRef* | 0:1 | Scheduled STOP POINT that is assigned to a FLEXIBLE STOP PLACE by this FLEXIBLE STOP ASSIGNMENT. |
| «FK» | ***FlexibleStopPlace Ref*** | *FlexibleStopPlaceRef* | 0:1 | FLEXIBLE STOP PLACE that is assigned to a SCHEDULED STOP POINT by this FLEXIBLE STOP ASSIGNMENT. |
| «FK» | ***FlexibleQuayRef*** | *FlexibleQuayRef* | 0:1 | FLEXIBLE QUAY that is assigned to a SCHEDULED STOP POINT by this FLEXIBLE STOP ASSIGNMENT. |

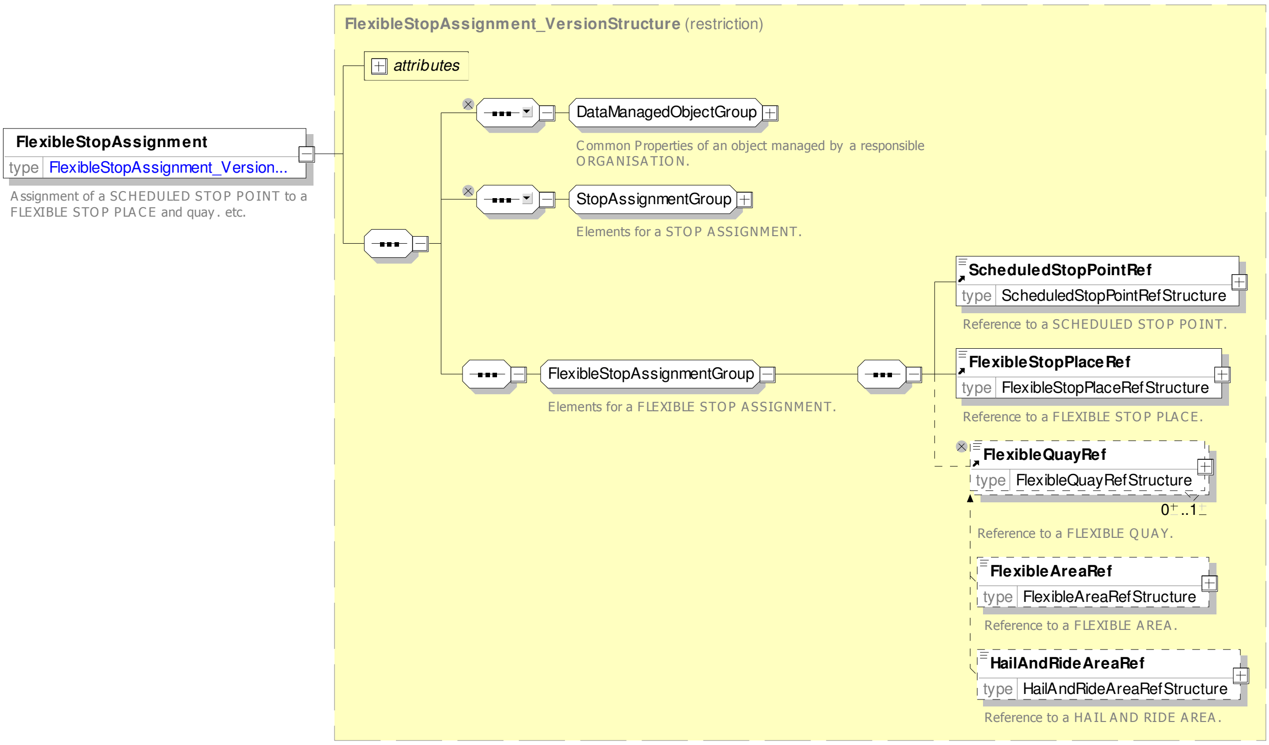

FlexibleStopAssignment_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
StopAssignmentGroup
Elements for a STOP ASSIGNMENT.
FlexibleStopAssignment
type FlexibleStopAssignment_Version...
Assignment of a SCHEDULED STOP POINT to a FLEXIBLE STOP PLACE and quay. etc.
FlexibleStopAssignmentGroup
Elements for a FLEXIBLE STOP ASSIGNMENT.
ScheduledStopPointRef
type ScheduledStopPointRefStructure
Reference to a SCHEDULED STOP POINT.
FlexibleStopPlaceRef
type FlexibleStopPlaceRefStructure
Reference to a FLEXIBLE STOP PLACE.
FlexibleQuayRef
type FlexibleQuayRefStructure
0..1
Reference to a FLEXIBLE QUAY.
FlexibleAreaRef
type FlexibleAreaRefStructure
Reference to a FLEXIBLE AREA.
HailAndRideAreaRef
type HailAndRideAreaRefStructure
Reference to a HAIL AND RIDE AREA.

**Figure 113 — *FlexibleStopAssignment* — XSD**

### 7.2.10.3.3 TypeOfFlexibleService – Model Element

Classification of a FLEXIBLE JOURNEY PATTERN. All the services on this type of JOURNEY PATTERN are flexible services.

Types of flexible services include :

- - Virtual line service.
- - Flexible service with main route.
- - Corridor service.
- - Fixed stop area-wide flexible service.
- - Free area-wide flexible service.
- - Mixed types of flexible service (not at POINT level).
- The type of flexibility can be defined at JOURNEY PATTERN LEVEL or at POINT IN JOURNEY PATTERN LEVEL in case of "mixed types of flexible service" inside the same JOURNEY PATTERN

Table 61 – *TypeOfFlexibleService* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *TypeOfValue* | ::> | TYPE OF FLEXIBLE SERVICE inherits from TYPE OF VALUE |
| | **id** | *TypeOfFlexibleServiceIdType* | 1:1 | Type of FLEXIBLE SERVICE.<br>— - Virtual line service.<br>— - Flexible service with main route.<br>— - Corridor service.<br>— - Fixed stop area-wide flexible service.<br>— - Free area-wide flexible service.<br>— - Mixed types of flexible service |

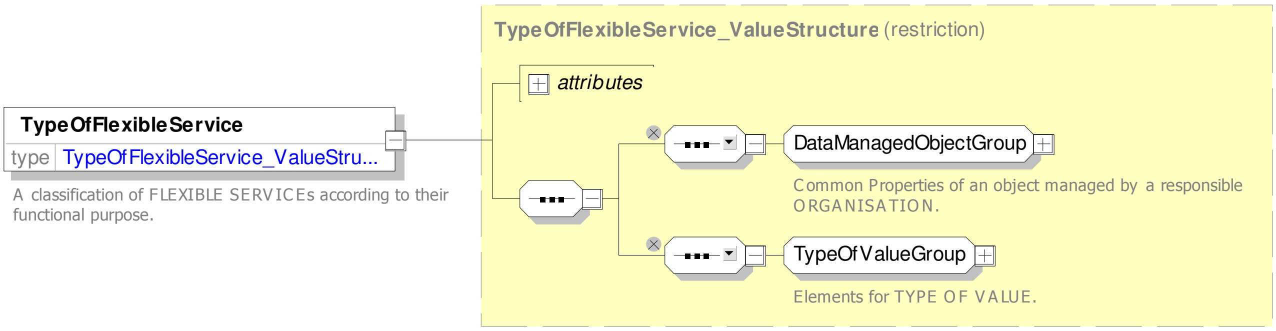

TypeOfFlexibleService_ValueStructure (restriction)
attributes
TypeOfFlexibleService
type TypeOfFlexibleService_ValueStru...
A classification of FLEXIBLE SERVICEs according to their functional purpose.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
TypeOfValueGroup
Elements for TYPE OF VALUE.

Figure 114 — *TypeOfFlexibleService* — XSD

### 7.2.10.4 XML Example of Flexible Service

The following code fragment shows a TEMPLATE VEHICLE JOURNEY with two different frequency groups.

For EXAMPLE

```
<ServiceJourney version="any" id="hde:sj_24o_01">
    <DepartureTime>14:20:00.0Z</DepartureTime>
    <ServicePatternRef version="any" ref="hde:svp_24o"/>
    <LineRef version="any" ref="mybus:FlexibleLine:FL_24"/>
    <JourneyPatternView>
        <RouteRef version="any" ref="mybus:RT_24o"/>
        <DirectionType>outbound</DirectionType>
        <DestinationDisplayRef version="any" ref="mybus:DST_Charley"/>
    </JourneyPatternView>
    <calls>
        <Call id="hde:sj_24o_01_001" order="1">
            <ScheduledStopPointRef version="any" ref="mybus:SSP_001"/>
            <Arrival>
                <ForAlighting>false</ForAlighting>
            </Arrival>
            <Departure>
                <Time>14:00:00.0Z</Time>
            </Departure>
        </Call>
        <Call id="hde:sj_24o_01_002" order="2">
            <ScheduledStopPointRef version="any" ref="mybus:SSP_002"/>
            <Arrival>
                <Time>14:30:00.0Z</Time>
            </Arrival>
            <Departure>
                <Time>14:32:00.0Z</Time>
            </Departure>
        </Call>
        <Call id="hde:sj_24o_01_003" order="3">
            <ScheduledStopPointRef version="any" ref="mybus:SSP_077"/>
            <Arrival>
                <Time>15:10:00.0Z</Time>
            </Arrival>
            <Departure>
                <ForBoarding>false</ForBoarding>
            </Departure>
        </Call>
    </calls>
    <FlexibleServiceProperties version="1" id="hde:sj_24o_02">
        <TypeOfFlexibleServiceRef ref="myfs" version="any"/>
        <BookingMethods>online callOffice</BookingMethods>
        <BookingAccess>public</BookingAccess>
        <LatestBookingTime>16:00:00Z</LatestBookingTime>
        <MinimumBookingPeriod>PT2H</MinimumBookingPeriod>
        <bookingNotes>
            <NoticeRef version="1" ref="hde:sj_24o_02_BN"/>
        </bookingNotes>
    </FlexibleServiceProperties>
</ServiceJourney>
```

### 7.2.11 Journey Accounting

The JOURNEY ACCOUNTING provides a set of parameters characterizing VEHICLE JOURNEYs or SPECIAL SERVICEs used for accounting purposes in particular in contracts between ORGANISATIONs.

The accounting can be done based on the distance run by the JOURNEY, on its duration or on both.

#### 7.2.11.1 JOURNEY ACCOUNTING – Conceptual MODEL

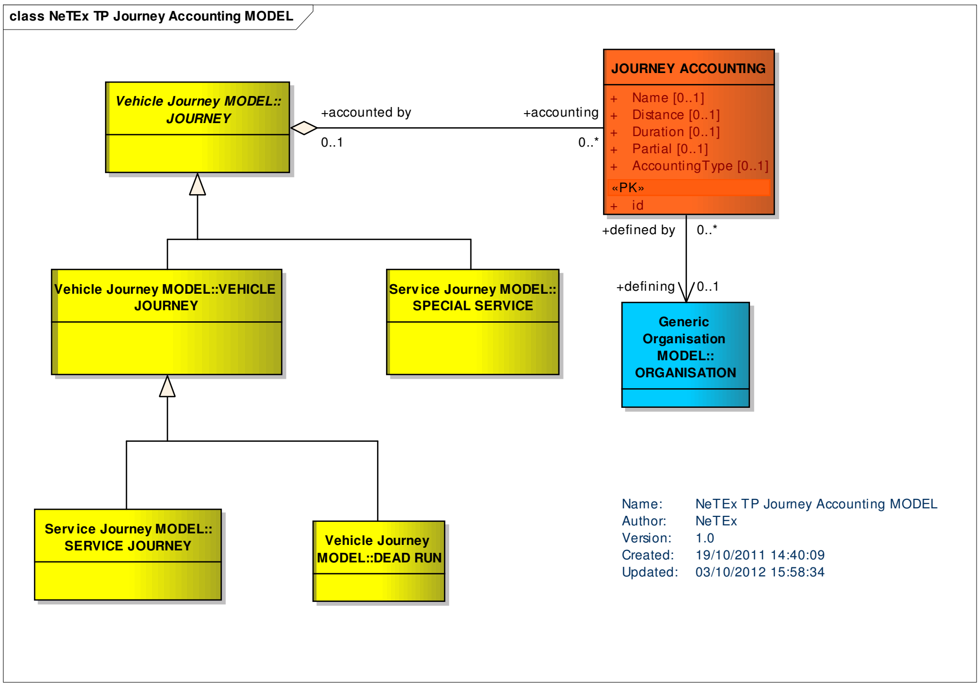

class NeTEx TP Journey Accounting MODEL
Vehicle Journey MODEL::JOURNEY
+accounted by
0..1
+accounting
0..*
JOURNEY ACCOUNTING
+ Name [0..1]
+ Distance [0..1]
+ Duration [0..1]
+ Partial [0..1]
+ AccountingType [0..1]
«PK»
+ id
+defined by
0..*
+defining
0..1
Generic Organisation MODEL:: ORGANISATION
Vehicle Journey MODEL::VEHICLE JOURNEY
Service Journey MODEL:: SPECIAL SERVICE
Service Journey MODEL:: SERVICE JOURNEY
Vehicle Journey MODEL::DEAD RUN
Name: NeTEx TP Journey Accounting MODEL
Author: NeTEx
Version: 1.0
Created: 19/10/2011 14:40:09
Updated: 03/10/2012 15:58:34

Figure 115 — Journey Accounting – Conceptual MODEL (UML)

### 7.2.11.2 Journey Accounting – Physical Model

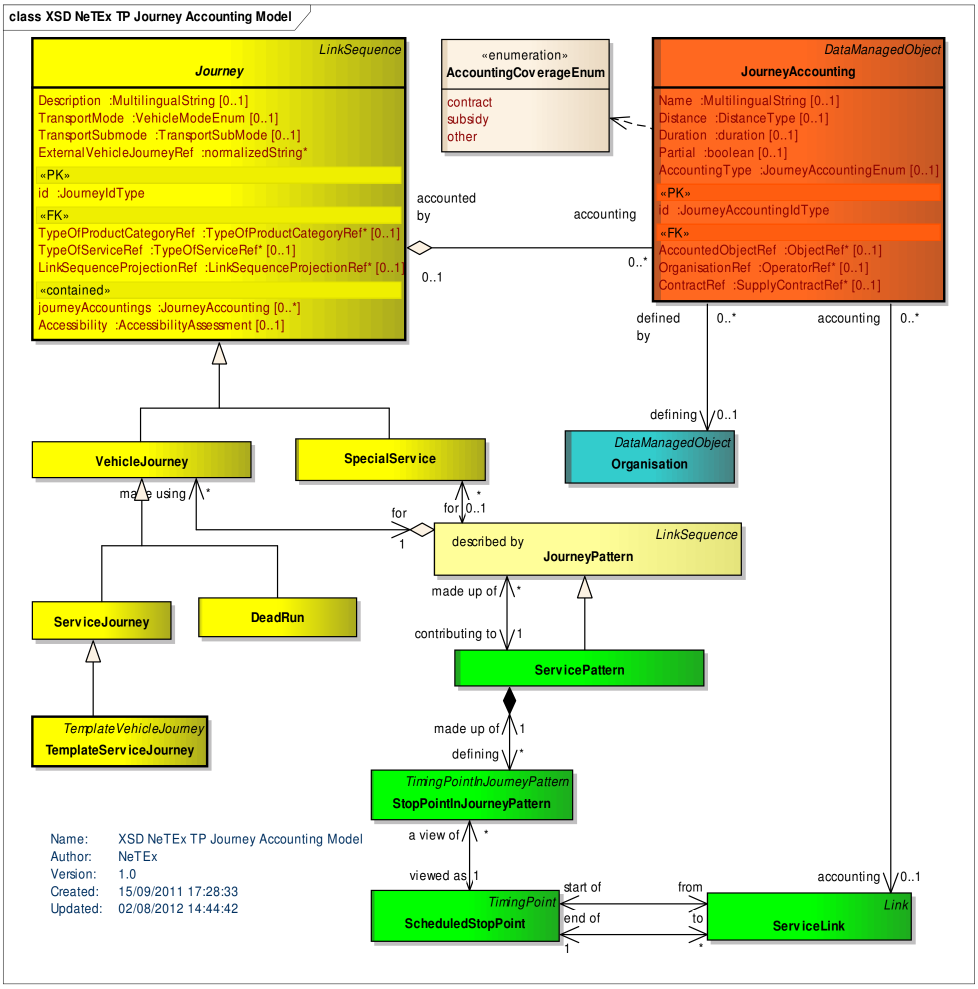

class XSD NeTEx TP Journey Accounting Model
LinkSequence
Journey
Description :MultilingualString [0..1]
TransportMode :VehicleModeEnum [0..1]
TransportSubmode :TransportSubMode [0..1]
ExternalVehicleJourneyRef :normalizedString*
«PK»
id :JourneyIdType
«FK»
TypeOfProductCategoryRef :TypeOfProductCategoryRef* [0..1]
TypeOfServiceRef :TypeOfServiceRef* [0..1]
LinkSequenceProjectionRef :LinkSequenceProjectionRef* [0..1]
«contained»
journeyAccountings :JourneyAccounting [0..*]
Accessibility :AccessibilityAssessment [0..1]
«enumeration»
AccountingCoverageEnum
contract
subsidy
other
DataManagedObject
JourneyAccounting
Name :MultilingualString [0..1]
Distance :DistanceType [0..1]
Duration :duration [0..1]
Partial :boolean [0..1]
AccountingType :JourneyAccountingEnum [0..1]
«PK»
id :JourneyAccountingIdType
«FK»
AccountedObjectRef :ObjectRef* [0..1]
OrganisationRef :OperatorRef* [0..1]
ContractRef :SupplyContractRef* [0..1]
accounted by
accounting
0..*
0..1
defined by
0..*
accounting
0..*
defining
0..1
DataManagedObject
Organisation
VehicleJourney
SpecialService
made using
*
for
for
*
0..1
1
described by
LinkSequence
JourneyPattern
made up of
*
contributing to
1
ServiceJourney
DeadRun
ServicePattern
TemplateVehicleJourney
TemplateServiceJourney
made up of
1
defining
*
TimingPointInJourneyPattern
StopPointInJourneyPattern
a view of
*
viewed as
1
TimingPoint
ScheduledStopPoint
start of
end of
1
from
to
*
accounting
0..1
Link
ServiceLink
Name: XSD NeTEx TP Journey Accounting Model
Author: NeTEx
Version: 1.0
Created: 15/09/2011 17:28:33
Updated: 02/08/2012 14:44:42

Figure 116 — Journey Accounting – Physical Model (UML)

### 7.2.11.3 Journey Accounting – Attributes and XSD

#### 7.2.11.3.1 JourneyAccounting – Model Element

Parameters characterizing VEHICLE JOURNEYs or SPECIAL SERVICEs used for accounting purposes in particular in contracts between ORGANISATIONs.

Table 62 – *JourneyAccounting* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | JOURNEY ACCOUNTING inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *JourneyAccountingIdType* | 1:1 | Identifier of JOURNEY ACCOUNTING. |
|  | ***Name*** | *MultilingualString* | 0:1 | Name of JOURNEY ACCOUNTING. |
|  | ***Distance*** | *DistanceType* | 0:1 | Distance to us in for JOURNEY ACCOUNTING. |

|  |  |  |  |  |
|---|---|---|---|---|
|  | ***Duration*** | *xsd:duration* | 0:1 | Duration to use for JOURNEY ACCOUNTING. |
|  | ***Partial*** | *xsd:boolean* | 0:1 | Whether all or some of journey is subject to ACCOUNTING. |
|  | ***AccountingType*** | *JourneyAccountingEnum* | 0:1 | Nature of JOURNEY ACCOUNTING. |
| «FK» | ***Accounted-ObjectRef*** | *ObjectRef* | 0:1 | Object for which this provides the accounting. |
| «FK» | ***OrganisationRef*** | *OperatorRef* | 0:1 | Reference to ORGANISATION providing contract or subsidy. |
| «FK» | ***ContractRef*** | *SupplyContractRef* | 0:1 | Supply Contract under which JOURNEY ACCOUNTING is provided. |

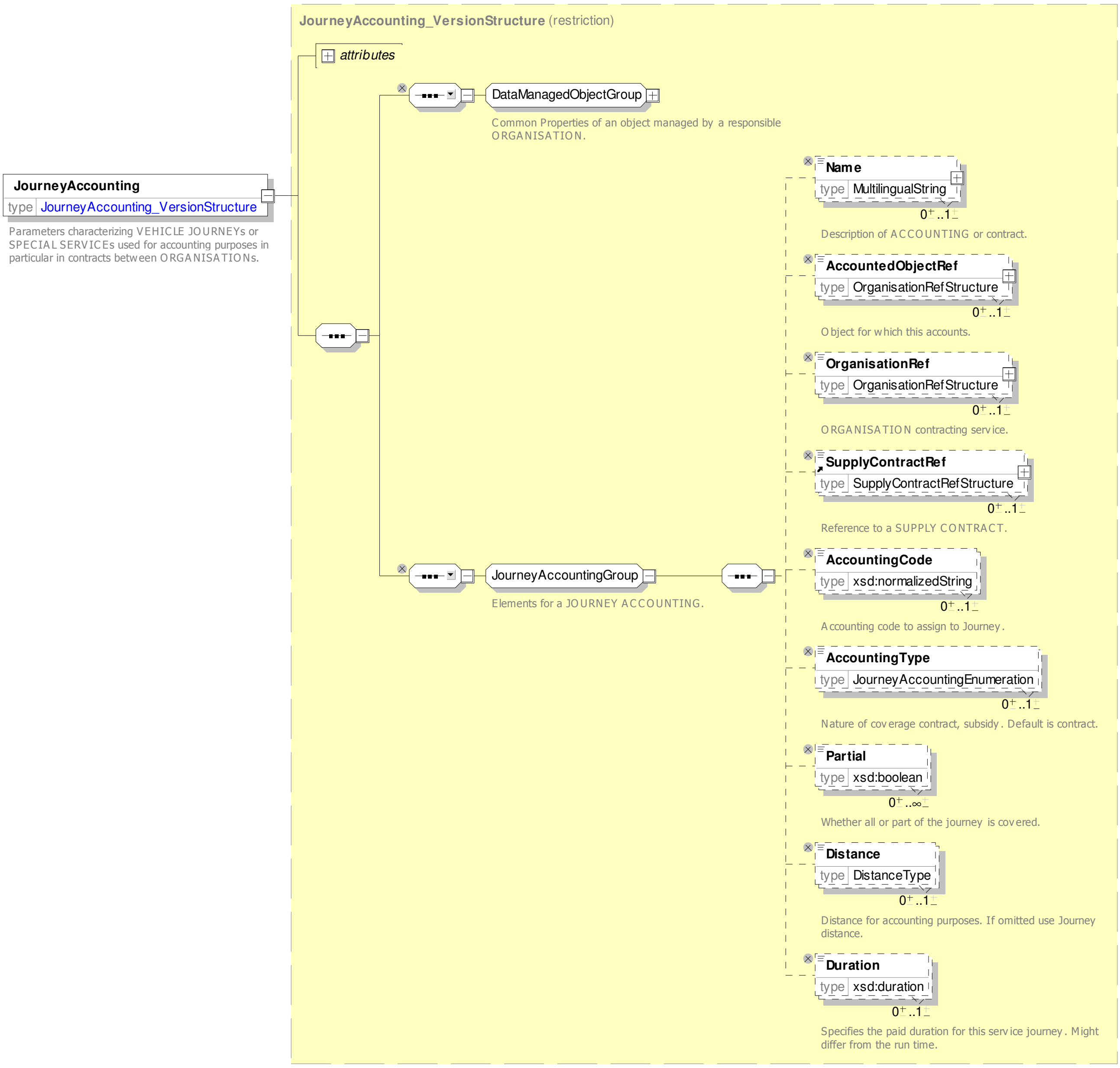

JourneyAccounting_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
JourneyAccounting
type JourneyAccounting_VersionStructure
Parameters characterizing VEHICLE JOURNEYs or SPECIAL SERVICEs used for accounting purposes in particular in contracts between ORGANISATIONs.
Name
type MultilingualString
0..1
Description of ACCOUNTING or contract.
AccountedObjectRef
type OrganisationRefStructure
0..1
Object for which this accounts.
OrganisationRef
type OrganisationRefStructure
0..1
ORGANISATION contracting service.
SupplyContractRef
type SupplyContractRefStructure
0..1
Reference to a SUPPLY CONTRACT.
JourneyAccountingGroup
Elements for a JOURNEY ACCOUNTING.
AccountingCode
type xsd:normalizedString
0..1
Accounting code to assign to Journey.
AccountingType
type JourneyAccountingEnumeration
0..1
Nature of coverage contract, subsidy. Default is contract.
Partial
type xsd:boolean
0..∞
Whether all or part of the journey is covered.
Distance
type DistanceType
0..1
Distance for accounting purposes. If omitted use Journey distance.
Duration
type xsd:duration
0..1
Specifies the paid duration for this service journey. Might differ from the run time.

Figure 117 — *JourneyAccounting* — XSD

#### 7.2.11.3.1.1 JourneyAccountingType – Allowed values

The following table shows the allowed values for ***JourneyAccountingType*** *(JourneyAccountingType*E*num).*

**Table 63 – JourneyAccountingType – Allowed values**

| Value | Description |
|---|---|
| *contract* | Journey is undertaken under contract. |
| *subsidy* | Journey is undertaken under subsidy. |
| *other* | Other |

### 7.2.11.4 XML Example of Journey Accounting

The following code fragment shows Journey ACCOUNTINGs on a SERVICE JOURNEYs**.**

For EXAMPLE

```
<ServiceJourney version="any" id="hde:sj_24o_01">
        <journeyAccountings>
            <JourneyAccounting id="mytim:ac02">
                <OrganisationRef xsi:type="AuthorityRefStructure" ref="txc:xshire"/>
                <AccountingType>subsidy</AccountingType>
                <Partial>false</Partial>
                <Distance>20</Distance>
            </JourneyAccounting>
            <JourneyAccounting id="mytim:ac02">
                <OrganisationRef xsi:type="AuthorityRefStructure" ref="txc:xshire”/>
                <AccountingType>contract</AccountingType>
                <Partial>false</Partial>
                <Distance>20</Distance>
            </JourneyAccounting>
        </journeyAccountings>
        <DepartureTime>14:20:00.0Z</DepartureTime>
        <ServicePatternRef version="any" ref="hde:ServicePattern:svp_24o"/>
        <TimeDemandTypeRef version="any" ref="mybus:TimeDemandType:tdt_01"/>
        <LineRef version="any" ref="mybus:Line:LN_24"/>
```

### 7.2.12 Dated Journey

### 7.2.12.1 Dated Journey Package dependencies

The Dated Journey model extends the Vehicle Journey Model.

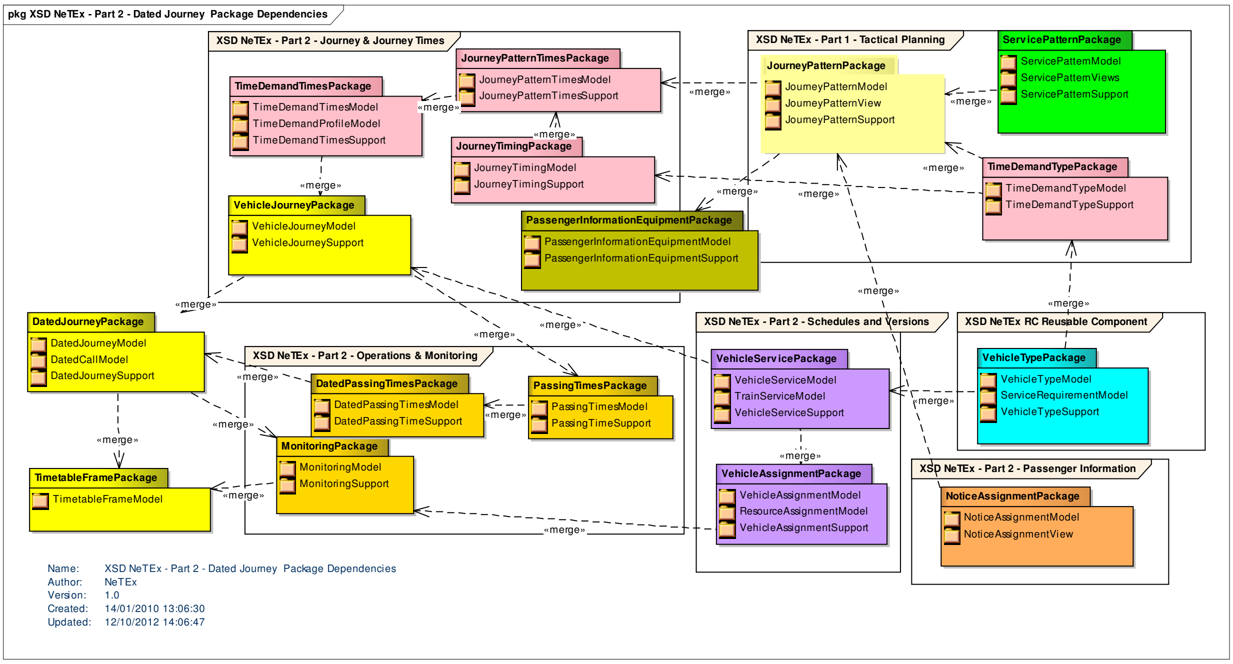


**Figure 118 — Dated Journey Package Dependencies**

### 7.2.12.2 DATED JOURNEY – Conceptual MODEL

Many reasons lead to modify the operational plan in the short term: special events, changes in the road infrastructure, incidents, etc. Some VEHICLE JOURNEYs may be added or deleted, may use alternative or shortened ROUTEs and JOURNEY PATTERNs, occasional services may be added, etc. If these changes are only valid for one or a few days, the reference schedule for a DAY TYPE is maybe not modified, but the changes are only stored for the appropriate OPERATING DAYs: the DATED VEHICLE JOURNEY describes a vehicle journey planned for one specific OPERATING DAY.

The NORMAL DATED VEHICLE JOURNEYs are based upon a VEHICLE JOURNEY, as produced for a DAY TYPE by the scheduling process. If there is no disturbance in the service, these normal journeys will be an exact image of the theoretical plan, applied to the specific OPERATING DAY. However, short-term modifications may be applied to these journeys, for instance when the controller decides to let a vehicle turn before the terminus. Such a control action updates the latest valid plan and produces a modified version of the DATED VEHICLE JOURNEY *(provided as explanation but outside of NeTEx scope)*.

A DATED VEHICLE JOURNEY serves one JOURNEY PATTERN. The classification of JOURNEY PATTERNs through TYPE OF JOURNEY PATTERN may include some categories of occasional JOURNEY PATTERNs, to be used only by extra or modified DATED VEHICLE JOURNEYs (e.g. for occasionally shortened or deviated routes). In this situation, NeTEx allows the exchange of such of occasional JOURNEY PATTERNs without any VEHICLE JOURNEY operating it (for example, a scheduling system sends these occasional JOURNEY PATTERNs to an AVMS who will use some for operational control to create extra VEHICLE JOURNEY serving them).

The following schema also provides some PASSING TIMEs description: detailed explanations of PASSING TIMEs are provided in following sections.

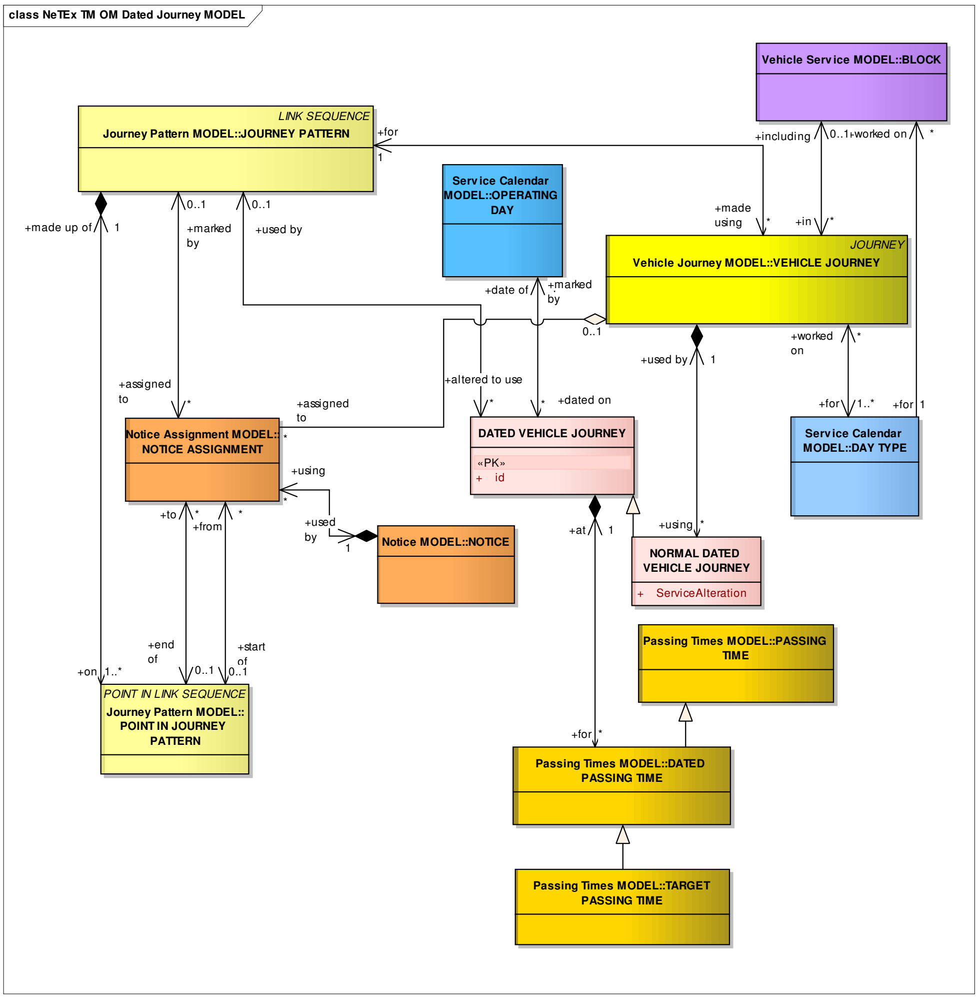

class NeTEx TM OM Dated Journey MODEL
Vehicle Service MODEL::BLOCK
LINK SEQUENCE
Journey Pattern MODEL::JOURNEY PATTERN
+for
1
+including
0..1
+worked on
*
Service Calendar MODEL::OPERATING DAY
+made using
*
+in
*
0..1
0..1
+made up of
1
+marked by
+used by
JOURNEY
Vehicle Journey MODEL::VEHICLE JOURNEY
+date of
+marked by
0..1
+worked on
*
+used by
1
+assigned to
*
+altered to use
*
*
+dated on
+assigned to
*
+for
1..*
+for
1
Notice Assignment MODEL::NOTICE ASSIGNMENT
DATED VEHICLE JOURNEY
«PK»
+ id
Service Calendar MODEL::DAY TYPE
+using
*
+to
*
+from
*
+used by
1
Notice MODEL::NOTICE
+at
1
+using
*
NORMAL DATED VEHICLE JOURNEY
+ ServiceAlteration
Passing Times MODEL::PASSING TIME
+end of
+start of
+on
1..*
0..1
0..1
POINT IN LINK SEQUENCE
Journey Pattern MODEL::POINT IN JOURNEY PATTERN
+for
*
Passing Times MODEL::DATED PASSING TIME
Passing Times MODEL::TARGET PASSING TIME

Figure 119 — Dated Journey — Conceptual MODEL (UML)

#### 7.2.12.2.1 NeTEx vs. SIRI Scope of Dated Vehicle Journeys

Additional DATED VEHICLE JOURNEYs may be included in the production plan for an OPERATING DAY. Such extra DATED VEHICLE JOURNEYs may be created de novo, or by duplication of an existing journey, with modification of the departure time and other properties. They may for instance correspond to a reinforcement of the service, or result from a complete rescheduling.

The exchange of DATED VEHICLE JOURNEYs for operational purposes (e.g. between an AVMS and a passenger information system) is considered out of scope for NeTEx since this is already covered by the SIRI-PT Publication Timetable Service. The SIRI-PT service in effect uses a view of the NeTEx DATED VEHICLE JOURNEY optimised for its purposes.

However, the NORMAL DATED VEHICLE JOURNEY is an important object for exchanges between scheduling systems and AVMS, and as the NORMAL DATED VEHICLE JOURNEY inherits from the DATED VEHICLE JOURNEY, these both objects are available in NeTEx.

### 7.2.12.3 Dated Journey – Physical Model

The following diagram shows the DATED JOURNEY physical model.

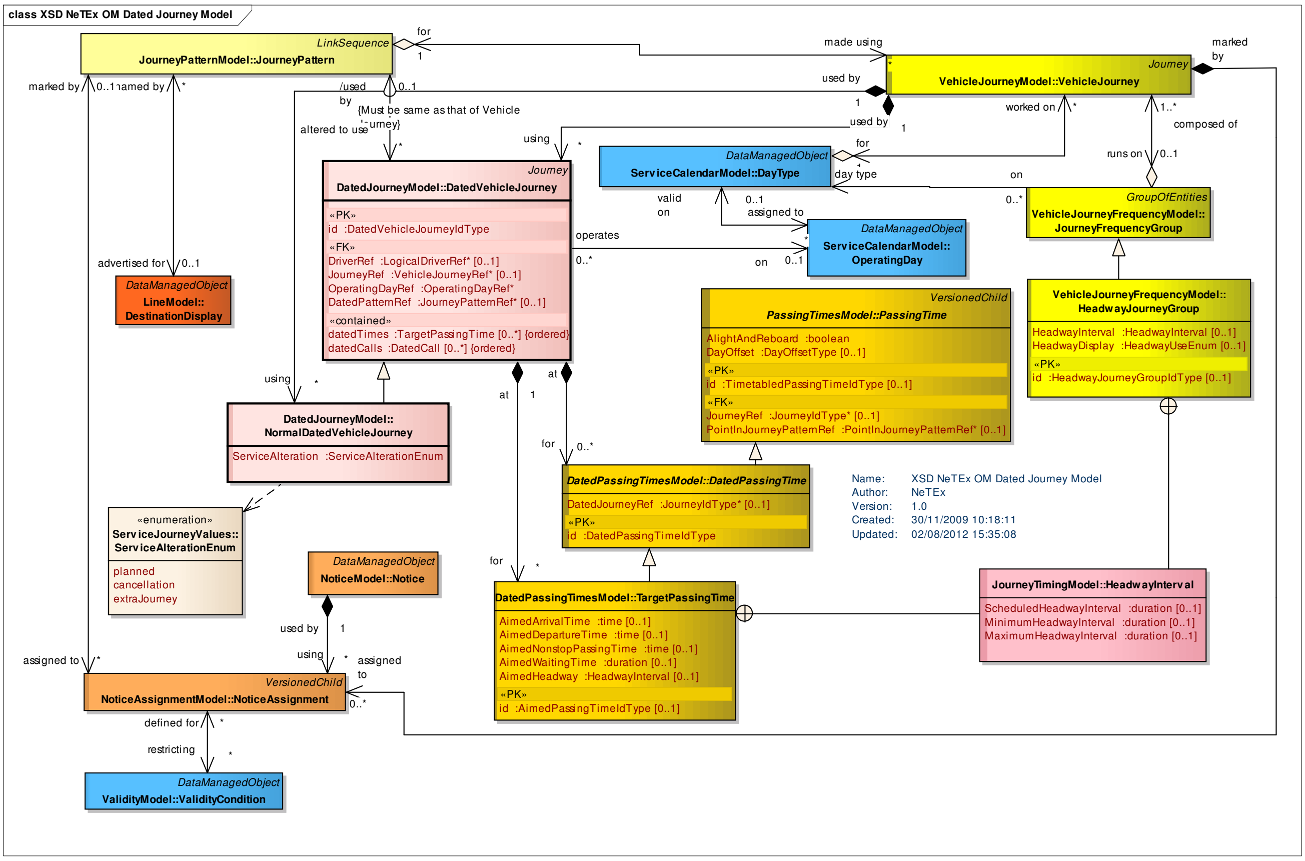

class XSD NeTEx OM Dated Journey Model
LinkSequence
JourneyPatternModel::JourneyPattern
for
1
made using
marked by
Journey
VehicleJourneyModel::VehicleJourney
used by
1
used by
1
worked on
*
1..*
composed of
runs on
0..1
marked by
0..1
named by
*
/used by
0..1
{Must be same as that of Vehicle Journey}
altered to use
*
using
*
DataManagedObject
ServiceCalendarModel::DayType
for
day type
on
0..*
GroupOfEntities
VehicleJourneyFrequencyModel::JourneyFrequencyGroup
valid on
0..1
assigned to
DataManagedObject
ServiceCalendarModel::OperatingDay
operates
0..*
on
0..1
Journey
DatedJourneyModel::DatedVehicleJourney
«PK»
id :DatedVehicleJourneyIdType
«FK»
DriverRef :LogicalDriverRef* [0..1]
JourneyRef :VehicleJourneyRef* [0..1]
OperatingDayRef :OperatingDayRef*
DatedPatternRef :JourneyPatternRef* [0..1]
«contained»
datedTimes :TargetPassingTime [0..*] {ordered}
datedCalls :DatedCall [0..*] {ordered}
advertised for
0..1
DataManagedObject
LineModel::DestinationDisplay
VersionedChild
PassingTimesModel::PassingTime
AlightAndReboard :boolean
DayOffset :DayOffsetType [0..1]
«PK»
id :TimetabledPassingTimeIdType [0..1]
«FK»
JourneyRef :JourneyIdType* [0..1]
PointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
VehicleJourneyFrequencyModel::HeadwayJourneyGroup
HeadwayInterval :HeadwayInterval [0..1]
HeadwayDisplay :HeadwayUseEnum [0..1]
«PK»
id :HeadwayJourneyGroupIdType [0..1]
using
*
at
1
at
for
0..*
DatedJourneyModel::NormalDatedVehicleJourney
ServiceAlteration :ServiceAlterationEnum
DatedPassingTimesModel::DatedPassingTime
DatedJourneyRef :JourneyIdType* [0..1]
«PK»
id :DatedPassingTimeIdType
Name: XSD NeTEx OM Dated Journey Model
Author: NeTEx
Version: 1.0
Created: 30/11/2009 10:18:11
Updated: 02/08/2012 15:35:08
«enumeration»
ServiceJourneyValues::ServiceAlterationEnum
planned
cancellation
extraJourney
DataManagedObject
NoticeModel::Notice
for
*
DatedPassingTimesModel::TargetPassingTime
AimedArrivalTime :time [0..1]
AimedDepartureTime :time [0..1]
AimedNonstopPassingTime :time [0..1]
AimedWaitingTime :duration [0..1]
AimedHeadway :HeadwayInterval [0..1]
«PK»
id :AimedPassingTimeIdType [0..1]
JourneyTimingModel::HeadwayInterval
ScheduledHeadwayInterval :duration [0..1]
MinimumHeadwayInterval :duration [0..1]
MaximumHeadwayInterval :duration [0..1]
used by
1
using
*
assigned to
assigned to
*
VersionedChild
NoticeAssignmentModel::NoticeAssignment
0..*
defined for
*
restricting
*
DataManagedObject
ValidityModel::ValidityCondition

Figure 120 — Dated Journey – Physical Model (UML)

## 7.2.12.4 Dated Journey – Attributes and XSD

### 7.2.12.4.1 DatedVehicleJourney – Model Element

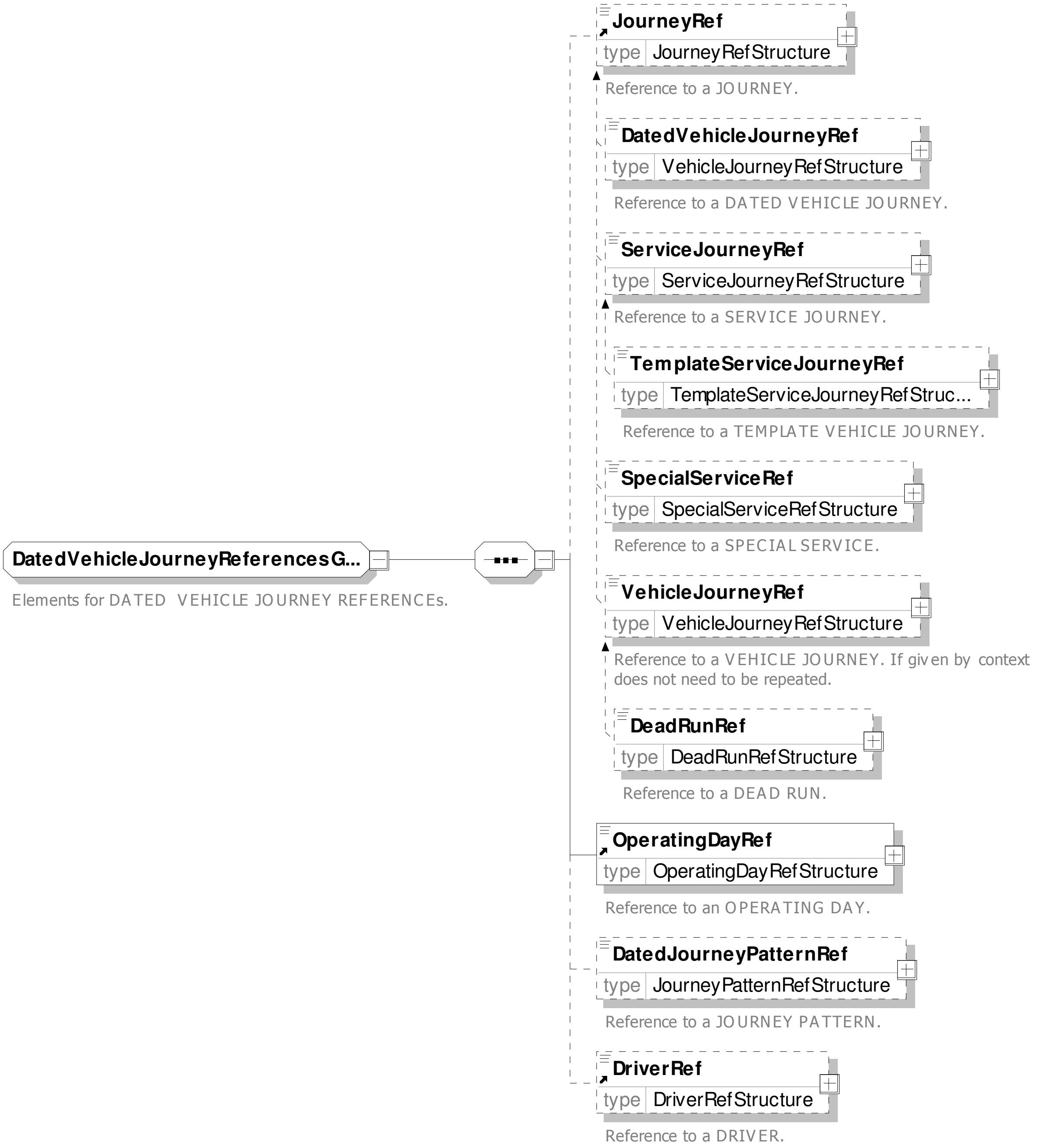
A particular journey of a vehicle on a particular OPERATING DAY.

Table 64 – *DatedVehicleJourney* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *VehicleJourney* | *::>* | DATED VEHICLE JOURNEY inherits from VEHICLE JOURNEY. |
| «PK» | ***id*** | *DatedVehicleJourney-IdType* | 1:1 | Identifier of DATED VEHICLE JOURNEY. |
| «FK» | ***JourneyRef*** | *VehicleJourneyRef* | 0:1 | VEHICLE JOURNEY for which this is a DATED VEHICLE JOURNEY. |
| «FK» | ***OperatingDayRef*** | *OperatingDayRef* | 1:1 | OPERATING DAY for DATED VEHICLE JOURNEY. |
| «AK» | ***ExternalDated-JourneyRef*** | *ExternalObjectRef* | 0:1 | An alternative code that uniquely identifies the DATED VEHICLE JOURNEY, specifically for use in AVMS systems.<br>*NOTE For VDV compatibility.* |

| | | | | |
|---|---|---|---|---|
| «FK» | ***DatedJourney-PatternRef*** | *JourneyPatternRef* | 0:1 | DATED JOURNEY PATTERN of DATED VEHICLE JOURNEY. |
| «FK» | ***DriverRef*** | *DriverRef* | 0:* | DRIVER of DATED VEHICLE JOURNEY. |
| «cntd» | ***datedPassing-Times*** | *TargetPassingTime* | 0:* | Passing times of DATED VEHICLE JOURNEY. |
| «cntd» | ***datedCalls*** | *DatedCall* | 0:* | CALLs of DATED VEHICLE JOURNEY. |

### 7.2.12.4.1.1 DatedVehicleJourneyReferencesGroup — Group



DatedVehicleJourneyReferencesG...
Elements for DATED VEHICLE JOURNEY REFERENCEs.
JourneyRef
type JourneyRefStructure
Reference to a JOURNEY.
DatedVehicleJourneyRef
type VehicleJourneyRefStructure
Reference to a DATED VEHICLE JOURNEY.
ServiceJourneyRef
type ServiceJourneyRefStructure
Reference to a SERVICE JOURNEY.
TemplateServiceJourneyRef
type TemplateServiceJourneyRefStruc...
Reference to a TEMPLATE VEHICLE JOURNEY.
SpecialServiceRef
type SpecialServiceRefStructure
Reference to a SPECIAL SERVICE.
VehicleJourneyRef
type VehicleJourneyRefStructure
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.
OperatingDayRef
type OperatingDayRefStructure
Reference to an OPERATING DAY.
DatedJourneyPatternRef
type JourneyPatternRefStructure
Reference to a JOURNEY PATTERN.
DriverRef
type DriverRefStructure
Reference to a DRIVER.

Figure 121 — *DatedVehicleJourneyReferencesGroup* — XSD

#### 7.2.12.4.2 NormalDatedVehicleJourney – Model Element

A DATED VEHICLE JOURNEY identical to a long-term planned VEHICLE JOURNEY.

**Table 65 – *NormalDatedVehicleJourney* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***ServiceAlteration*** | *ServiceAlterationEnum* | 1:1 | Service alteration type for NORMAL JOURNEY. See SERVICE JOURNEY. |

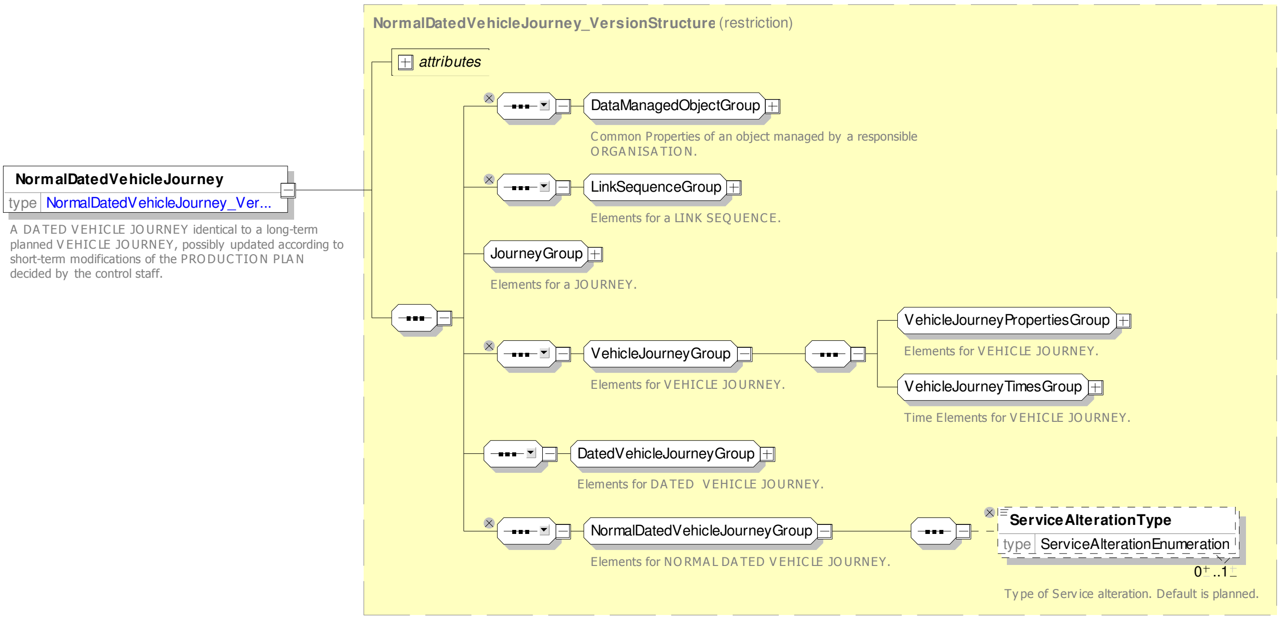

NormalDatedVehicleJourney_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
NormalDatedVehicleJourney
type NormalDatedVehicleJourney_Ver...
A DATED VEHICLE JOURNEY identical to a long-term planned VEHICLE JOURNEY, possibly updated according to short-term modifications of the PRODUCTION PLAN decided by the control staff.
LinkSequenceGroup
Elements for a LINK SEQUENCE.
JourneyGroup
Elements for a JOURNEY.
VehicleJourneyPropertiesGroup
Elements for VEHICLE JOURNEY.
VehicleJourneyGroup
Elements for VEHICLE JOURNEY.
VehicleJourneyTimesGroup
Time Elements for VEHICLE JOURNEY.
DatedVehicleJourneyGroup
Elements for DATED VEHICLE JOURNEY.
ServiceAlterationType
type ServiceAlterationEnumeration
NormalDatedVehicleJourneyGroup
Elements for NORMAL DATED VEHICLE JOURNEY.
0..1
Type of Service alteration. Default is planned.

**Figure 122 — *NormalDatedVehicleJourney* — XSD**

### 7.2.13 Passing Times

NOTE The following explanations use excerpts from Transmodel.PASSING TIMES – Conceptual MODEL

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 it. 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.)

In the Physical model PASSING TIMEs may also be exchanged in context using CALLs, which in effect provide a slightly denormalised view of a POINT IN JOURNEY PATTERN combined with one or more PASSING TIMEs.

#### 7.2.13.1 Timing Computation of a Journey

The computation of timings is not in the scope of NeTEx (NeTEx allows only to exchange the source information, or the result of the computation). But an understanding of this process gives an insight into the role of the timing elements of the NeTEx model.

To compute the timing of a VEHICLE JOURNEY an initial TIMING POINT belonging to the JOURNEY PATTERN for the journey is chosen as a timing reference and a 'departure time' is specified for the journey at this point. An initial TIME DEMAND TYPE is chosen that applies for that moment time. The standard run and wait times for the TIMING LINK to be applied for that TIME DEMAND TYPE are used to compute the time to the next TIMING POINT. The standard times for the TIMING LINKs may be overridden by specific times from the journey pattern, if they exist. The process is repeated for each TIMING POINT and each TIMING LINK in the JOURNEY PATTERN, if necessary changing the TIME DEMAND TYPE in effect at that time in the journey (and hence the wait and run times in effect). The VEHICLE JOURNEY is hence timed at all TIMING POINTs served by the JOURNEY PATTERN.

A journey that acts as a distributor journey for an incoming feeder journey may be further constrained as to its possible departure times by the JOURNEY MEETING or INTERCHANGE RULEs in effect for the meeting.

Not all the SCHEDULED STOP POINTs served by the JOURNEY PATTERN are necessarily TIMING POINTs. It is still useful to compute passing times at such intermediate SCHEDULED STOP POINTs. This is made by interpolation, taking into account for instance the distance between STOP POINTs.

#### 7.2.13.2 Different types of passing times

In the design of the operational plan for a specific OPERATING DAY, planned passing times may be amended by changes made on the schedule. This will produce “target passing times” for this day, supporting the monitoring of operations. In addition, forecasts may be made on “estimated passing times”, mainly for passenger information, and actually “observed passing times” may be recorded by a monitoring system. All these concepts are described by the generic entity PASSING TIME, which has several sub-types.

Only TIMETABLED PASSING TIME and to some extent TARGET PASSING TIME from a NORMAL DATED VEHICLE JOURNEY are really scheduled information and therefore in the scope of NeTEx Part 2.

A first calculation of DATED PASSING TIMEs may be carried out as soon as the DATED VEHICLE JOURNEYs for a particular OPERATING DAY are defined (being NORMAL DATED VEHICLE JOURNEYs). Such passing times are called TARGET PASSING TIMEs.

Later on, modifications of the plan for a specific OPERATING DAY, in particular changes in journeys, are likely to generate the need to modify the passing times (usually in the operation control system or the AVMS).

The TARGET PASSING TIMEs are normally generated for the first time a few days before the OPERATING DAY, and are from that moment on constantly updated, according to the modifications operated to the schedule. The TARGET PASSING TIME can be seen as the latest official plan for vehicle operation.

The other PASSING TIMEs concepts are described below for information, but are relevant for SIRI (real-time information exchange), not for NeTEx.

The drivers will certainly try to meet the TARGET PASSING TIMEs, but in practice the vehicle may deviate from this plan due to the traffic circumstances. Various systems (such as AVM) can monitor these deviations and provide an estimation of PASSING TIMEs at the next stop points, for passenger information or for service control. Such ESTIMATED PASSING TIMEs will be displayed at stops and will be used for trip planning queries. They may be used as well by the AVM system, in order for instance to alert the controller on a forecasted delay for a relief.

The OBSERVED PASSING TIME describes the actual PASSING TIMEs that have been recorded at a specific POINT. This information is used to compute further ESTIMATED PASSING TIMEs and, of course, for statistical purposes. OBSERVED PASSING TIMEs are resulting from the monitoring process and are therefore related to a MONITORED VEHICLE JOURNEY.

### 7.2.13.3 Passing Time details

#### 7.2.13.3.1 Timetabled Passing Times

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 for a long period of time (e.g. summer, winter timetable). They are calculated for the planned NORMAL DATED VEHICLE JOURNEYs that were defined during the scheduling process. They may be computed as well for any extra DATED VEHICLE JOURNEY created by a CONTROL ACTION.

The TIMETABLED PASSING TIMEs can be calculated for the TIMING POINTs that were used for scheduling, but they are often also generated through interpolation, for POINTs that were not defined as TIMING POINTs (for instance for all STOP POINTs of a JOURNEY PATTERN). Therefore, a TIMETABLED PASSING TIME is defined for a POINT IN the JOURNEY PATTERN that the journey in question is serving.

#### 7.2.13.3.2 Actual Passing Times (informative)

The PASSING TIMEs that are computed on a specific OPERATING DAY are called DATED PASSING TIMEs. DATED PASSING TIME has several subtypes:

- TARGET PASSING TIME, the latest official plan for a DATED VEHICLE JOURNEY, on a POINT IN JOURNEY PATTERN;
- ESTIMATED PASSING TIME, a forecast for a MONITORED VEHICLE JOURNEY, on a POINT IN JOURNEY PATTERN;
- OBSERVED PASSING TIME, recorded for a MONITORED VEHICLE JOURNEY, on a particular POINT.

#### 7.2.13.3.3 Passing Times Attributes

The sub-types of PASSING TIME have several optional attributes, expressing the time value of a possible:

- Arrival time.
- Departure time.
- Waiting time (in general associated to one of the previous two).
- Non-stop passing time when the vehicle is not obliged to stop.
- Latest arrival time, being the operator or authority commitment, and very useful for dynamic interchange management (e.g. for a journey planner), it also provides some accuracy information on the arrival time.
- Earliest departure time, being the operator or authority commitment, and very useful for dynamic interchange management (e.g. for a journey planner), it also provides some accuracy information on the departure time.

In operations of transport modes such as railway, ferry, etc., the functions carried out when a vehicle passes a stop point are much more complex than, for instance, in bus operations. Passengers may be allowed, when the stopping time is sufficient, to alight for a while (e.g. for refreshment) before resuming their trip on the same vehicle. Some facilities or some specific passenger information may be provided on this occasion.

A PASSING TIME may be a simple passage (e.g. of a bus at a stop point) or a longer stay (e.g. in a maritime port of call). The attribute ***AlightAndReboard*** expresses the possibility for the passenger to alight for a while, during the PASSING TIME of a VEHICLE JOURNEY at a particular STOP POINT.

### 7.2.13.4 Passing Times – Physical Model

#### 7.2.13.4.1 Timetabled Passing Times – Physical Model

The PASSING TIMEs physical model is shown in the following figure and introduces detailed attributes. These include HEADWAY INTERVAL attributes that may be useful to present frequencies of a journey at the stop.

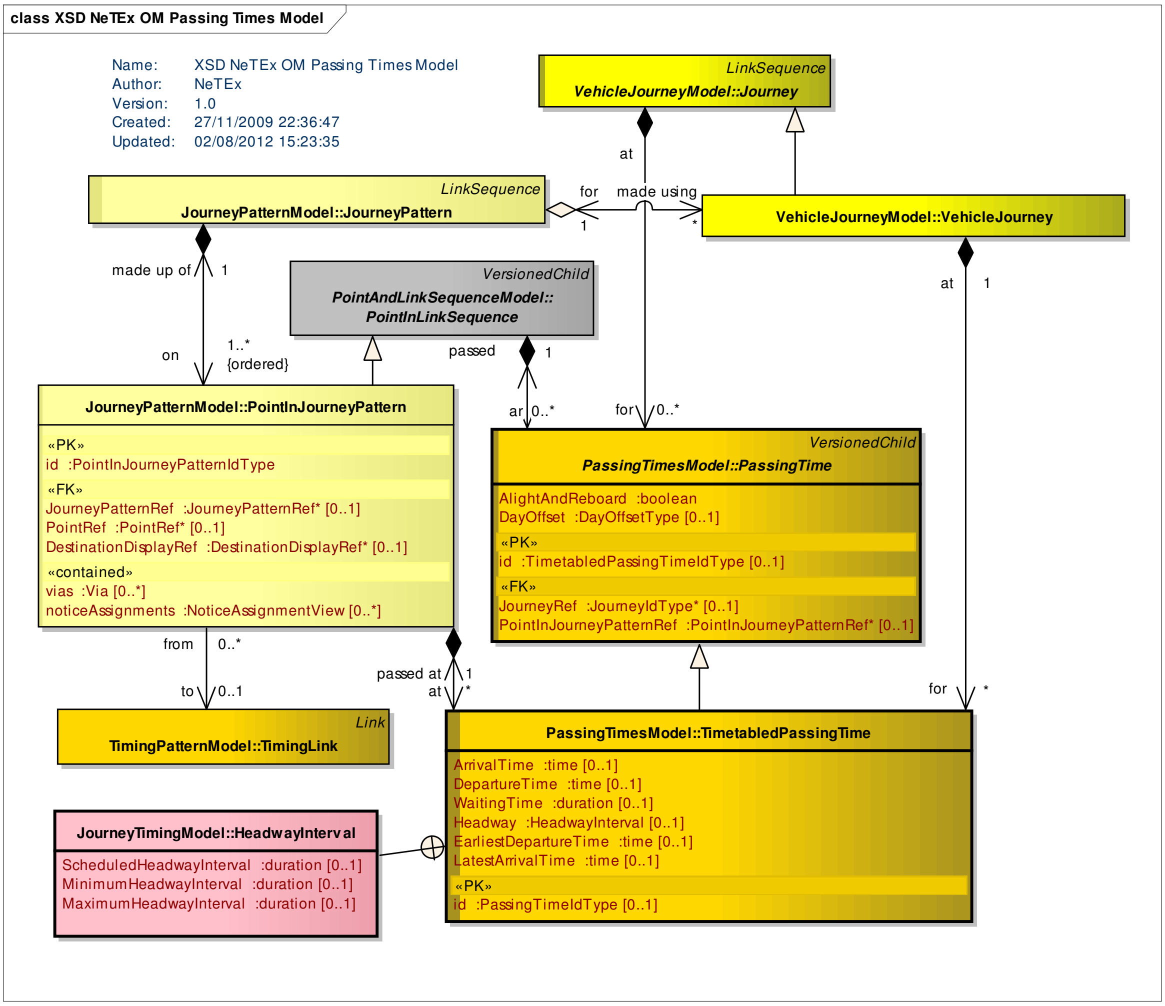

class XSD NeTEx OM Passing Times Model
Name: XSD NeTEx OM Passing Times Model
Author: NeTEx
Version: 1.0
Created: 27/11/2009 22:36:47
Updated: 02/08/2012 15:23:35
LinkSequence
VehicleJourneyModel::Journey
LinkSequence
JourneyPatternModel::JourneyPattern
VehicleJourneyModel::VehicleJourney
at
for
made using
1
*
made up of
1
on
1..*
{ordered}
VersionedChild
PointAndLinkSequenceModel::PointInLinkSequence
passed
1
at
1
JourneyPatternModel::PointInJourneyPattern
«PK»
id :PointInJourneyPatternIdType
«FK»
JourneyPatternRef :JourneyPatternRef* [0..1]
PointRef :PointRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef* [0..1]
«contained»
vias :Via [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
ar 0..*
for 0..*
VersionedChild
PassingTimesModel::PassingTime
AlightAndReboard :boolean
DayOffset :DayOffsetType [0..1]
«PK»
id :TimetabledPassingTimeIdType [0..1]
«FK»
JourneyRef :JourneyIdType* [0..1]
PointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
from 0..*
to 0..1
passed at 1
at *
for *
Link
TimingPatternModel::TimingLink
PassingTimesModel::TimetabledPassingTime
ArrivalTime :time [0..1]
DepartureTime :time [0..1]
WaitingTime :duration [0..1]
Headway :HeadwayInterval [0..1]
EarliestDepartureTime :time [0..1]
LatestArrivalTime :time [0..1]
«PK»
id :PassingTimeIdType [0..1]
JourneyTimingModel::HeadwayInterval
ScheduledHeadwayInterval :duration [0..1]
MinimumHeadwayInterval :duration [0..1]
MaximumHeadwayInterval :duration [0..1]

**Figure 123 — Timetabled Passing Times – Physical Model (UML)**

#### 7.2.13.4.2 Dated Passing Times – Physical Model

The DATED PASSING TIMEs physical model is shown below.

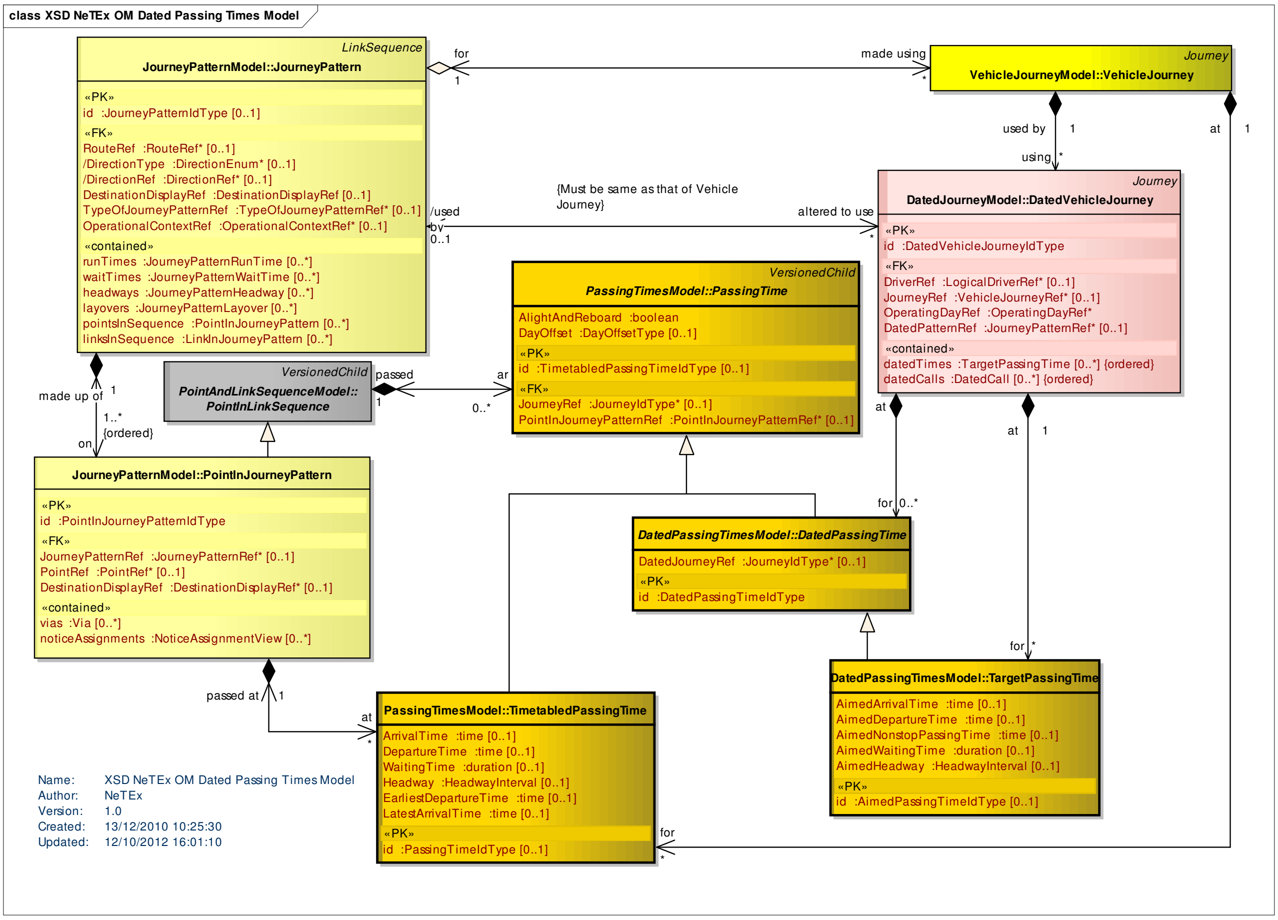

class XSD NeTEx OM Dated Passing Times Model
LinkSequence
JourneyPatternModel::JourneyPattern
«PK»
id :JourneyPatternIdType [0..1]
«FK»
RouteRef :RouteRef* [0..1]
/DirectionType :DirectionEnum* [0..1]
/DirectionRef :DirectionRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef [0..1]
TypeOfJourneyPatternRef :TypeOfJourneyPatternRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
«contained»
runTimes :JourneyPatternRunTime [0..*]
waitTimes :JourneyPatternWaitTime [0..*]
headways :JourneyPatternHeadway [0..*]
layovers :JourneyPatternLayover [0..*]
pointsInSequence :PointInJourneyPattern [0..*]
linksInSequence :LinkInJourneyPattern [0..*]
for
1
made using
*
Journey
VehicleJourneyModel::VehicleJourney
used by
1
at
1
using
*
{Must be same as that of Vehicle Journey}
/used by
0..1
altered to use
*
Journey
DatedJourneyModel::DatedVehicleJourney
«PK»
id :DatedVehicleJourneyIdType
«FK»
DriverRef :LogicalDriverRef* [0..1]
JourneyRef :VehicleJourneyRef* [0..1]
OperatingDayRef :OperatingDayRef*
DatedPatternRef :JourneyPatternRef* [0..1]
«contained»
datedTimes :TargetPassingTime [0..*] {ordered}
datedCalls :DatedCall [0..*] {ordered}
VersionedChild
PassingTimesModel::PassingTime
AlightAndReboard :boolean
DayOffset :DayOffsetType [0..1]
«PK»
id :TimetabledPassingTimeIdType [0..1]
«FK»
JourneyRef :JourneyIdType* [0..1]
PointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
VersionedChild
PointAndLinkSequenceModel::
PointInLinkSequence
passed
1
ar
0..*
made up of
1
1..*
{ordered}
on
at
at
1
for
0..*
JourneyPatternModel::PointInJourneyPattern
«PK»
id :PointInJourneyPatternIdType
«FK»
JourneyPatternRef :JourneyPatternRef* [0..1]
PointRef :PointRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef* [0..1]
«contained»
vias :Via [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
DatedPassingTimesModel::DatedPassingTime
DatedJourneyRef :JourneyIdType* [0..1]
«PK»
id :DatedPassingTimeIdType
for
*
DatedPassingTimesModel::TargetPassingTime
AimedArrivalTime :time [0..1]
AimedDepartureTime :time [0..1]
AimedNonstopPassingTime :time [0..1]
AimedWaitingTime :duration [0..1]
AimedHeadway :HeadwayInterval [0..1]
«PK»
id :AimedPassingTimeIdType [0..1]
passed at
1
at
*
PassingTimesModel::TimetabledPassingTime
ArrivalTime :time [0..1]
DepartureTime :time [0..1]
WaitingTime :duration [0..1]
Headway :HeadwayInterval [0..1]
EarliestDepartureTime :time [0..1]
LatestArrivalTime :time [0..1]
«PK»
id :PassingTimeIdType [0..1]
for
*
Name: XSD NeTEx OM Dated Passing Times Model
Author: NeTEx
Version: 1.0
Created: 13/12/2010 10:25:30
Updated: 12/10/2012 16:01:10

Figure 124 — Dated Passing Times – Physical Model (UML)

### 7.2.13.5 Passing times – Attributes and XSD

#### 7.2.13.5.1 PassingTime – Model Element

Time data concerning public transport vehicles passing a particular POINT; e.g. arrival time, departure time, waiting time.

Table 66 – *PassingTime* – Element

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

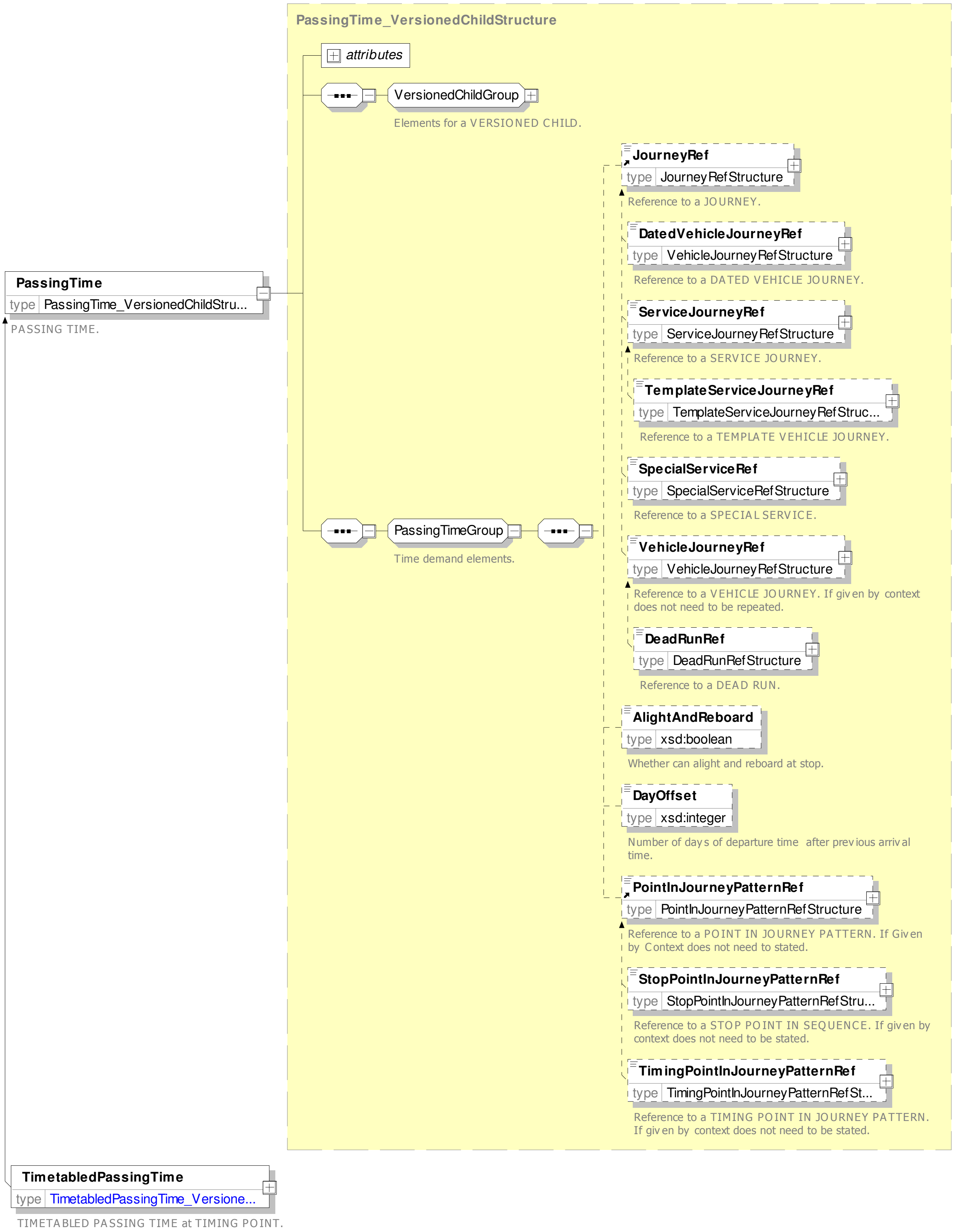

PassingTime_VersionedChildStructure
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
PassingTime
type PassingTime_VersionedChildStru...
PASSING TIME.
JourneyRef
type JourneyRefStructure
Reference to a JOURNEY.
DatedVehicleJourneyRef
type VehicleJourneyRefStructure
Reference to a DATED VEHICLE JOURNEY.
ServiceJourneyRef
type ServiceJourneyRefStructure
Reference to a SERVICE JOURNEY.
TemplateServiceJourneyRef
type TemplateServiceJourneyRefStruc...
Reference to a TEMPLATE VEHICLE JOURNEY.
SpecialServiceRef
type SpecialServiceRefStructure
Reference to a SPECIAL SERVICE.
PassingTimeGroup
Time demand elements.
VehicleJourneyRef
type VehicleJourneyRefStructure
Reference to a VEHICLE JOURNEY. If given by context does not need to be repeated.
DeadRunRef
type DeadRunRefStructure
Reference to a DEAD RUN.
AlightAndReboard
type xsd:boolean
Whether can alight and reboard at stop.
DayOffset
type xsd:integer
Number of days of departure time after previous arrival time.
PointInJourneyPatternRef
type PointInJourneyPatternRefStructure
Reference to a POINT IN JOURNEY PATTERN. If Given by Context does not need to stated.
StopPointInJourneyPatternRef
type StopPointInJourneyPatternRefStru...
Reference to a STOP POINT IN SEQUENCE. If given by context does not need to be stated.
TimingPointInJourneyPatternRef
type TimingPointInJourneyPatternRefSt...
Reference to a TIMING POINT IN JOURNEY PATTERN. If given by context does not need to be stated.
TimetabledPassingTime
type TimetabledPassingTime_Versione...
TIMETABLED PASSING TIME at TIMING POINT.

Figure 125 — *PassingTime* — XSD

### 7.2.13.5.2 TimetabledPassingTime – Model Element

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. Note that for Journey lasting more than one day, a possible ***DayOffset*** is available in ***passingTimeGroup*** (see above). If ***DepartureTime*** is not in the same day as ***ArrivalTime*** the information will be provided thanks to the ***WaitingTime***.

**Table 67 – *TimetabledPassingTime* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *PassingTime* | *::>* | TIMETABLED PASSING TIME inherits from PASSING TIME. |
| «PK» | ***id*** | *PassingTimeIdType* | 0:1 | Identifier of PASSING TIME. |
| | ***ArrivalTime*** | *xsd:time* | 0:1 | Arrival time at POINT IN PATTERN. |
| | ***DepartureTime*** | *xsd:time* | 0:1 | Departure time at POINT IN PATTERN. |
| | ***WaitingTime*** | *xsd:duration* | 0:1 | Waiting time at POINT IN PATTERN. |
| | ***Headway*** | *HeadwayInterval* | 0:1 | Headway interval at POINT IN PATTERN. |
| | ***EarliestDepartureTime*** | *xsd:time* | 0:1 | Earliest Departure time at POINT IN PATTERN. |
| | ***LatestArrivalTime*** | *xsd:time* | 0:1 | Latest Arrival time at POINT IN PATTERN. |

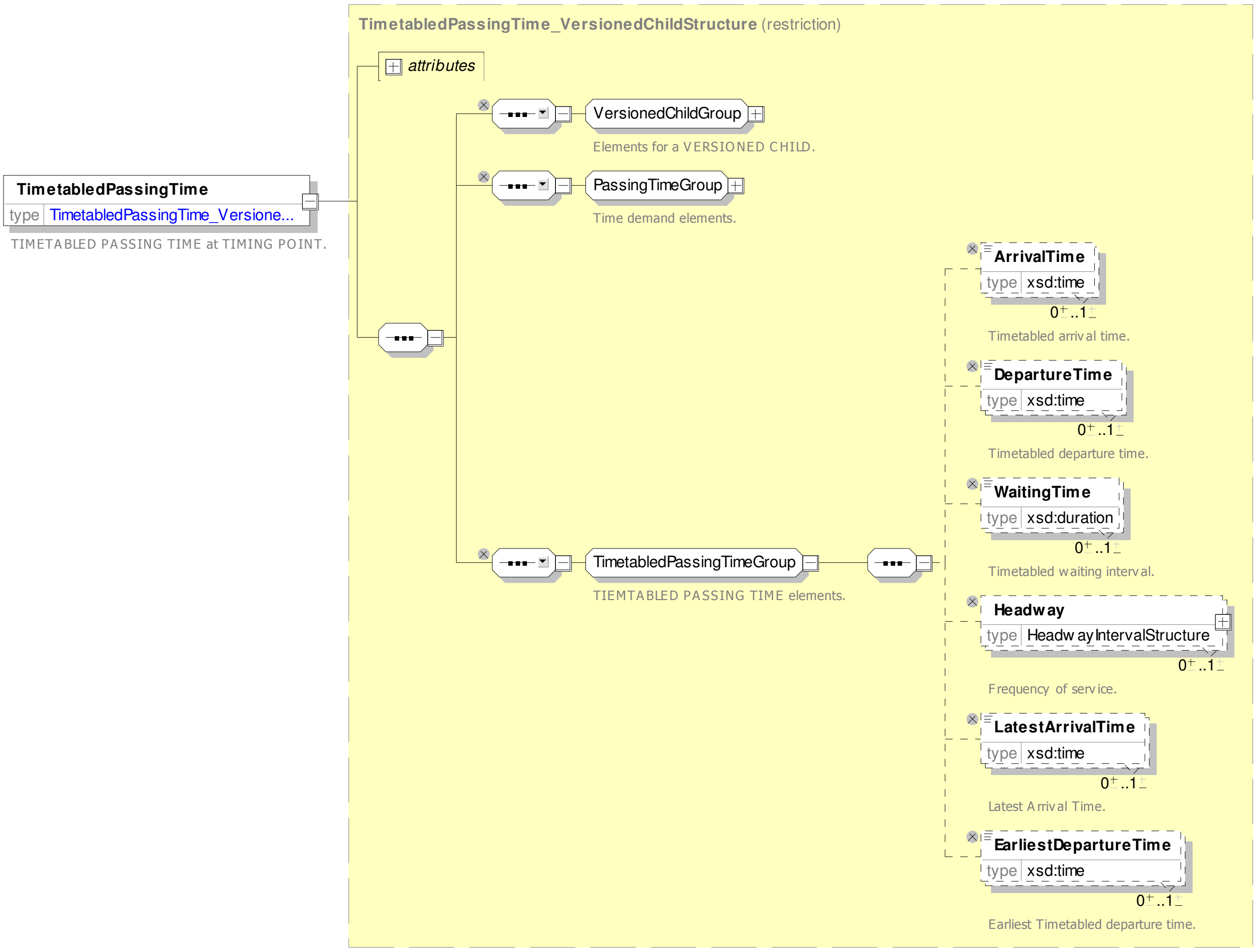

TimetabledPassingTime_VersionedChildStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
PassingTimeGroup
Time demand elements.
TimetabledPassingTime
type TimetabledPassingTime_Versione...
TIMETABLED PASSING TIME at TIMING POINT.
ArrivalTime
type xsd:time
0..1
Timetabled arrival time.
DepartureTime
type xsd:time
0..1
Timetabled departure time.
WaitingTime
type xsd:duration
0..1
Timetabled waiting interval.
TimetabledPassingTimeGroup
TIEMTABLED PASSING TIME elements.
Headway
type HeadwayIntervalStructure
0..1
Frequency of service.
LatestArrivalTime
type xsd:time
0..1
Latest Arrival Time.
EarliestDepartureTime
type xsd:time
0..1
Earliest Timetabled departure time.

**Figure 126 — *TimetabledPassingTime* — XSD**

### 7.2.13.6 Dated Passing times – Attributes and XSD

### 7.2.13.6.1 DatedPassingTime – Model Element

A PASSING TIME on a particular OPERATING DAY.

**Table 68 – *DatedPassingTime* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *PassingTime* | *::>* | DATED PASSING TIME inherits from PASSING TIME. |
| «PK» | ***id*** | *DatedPassingTimeIdType* | 1:1 | Identifier of DATED PASSING TIME. |
| | ***DatedJourneyRef*** | *JourneyRef* | 0:1 | DATED VEHICLE JOURNEY for which this is a PASSING TIME. |

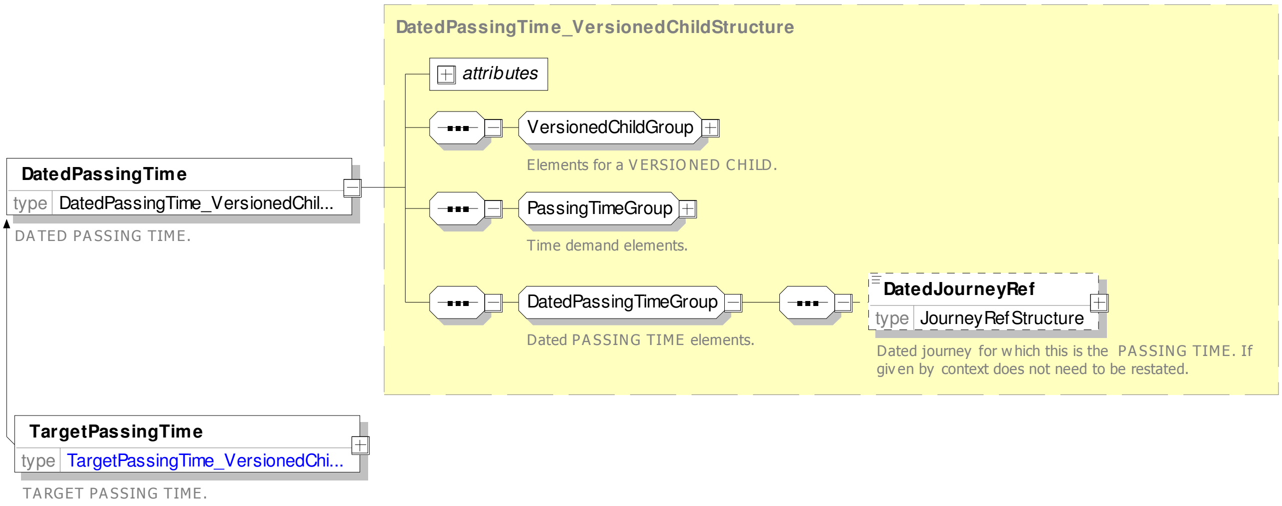

DatedPassingTime_VersionedChildStructure
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
PassingTimeGroup
Time demand elements.
DatedPassingTimeGroup
Dated PASSING TIME elements.
DatedJourneyRef
type JourneyRefStructure
Dated journey for which this is the PASSING TIME. If given by context does not need to be restated.
DatedPassingTime
type DatedPassingTime_VersionedChil...
DATED PASSING TIME.
TargetPassingTime
type TargetPassingTime_VersionedChi...
TARGET PASSING TIME.

Figure 127 — *DatedPassingTime* — XSD

#### 7.2.13.6.2 EstimatedPassingTime – Model Element

Out of NeTEx scope: see A.2.2.1.1

#### 7.2.13.6.3 ObservedPassingTime – Model Element

Out of NeTEx scope: see A.2.2.1.2

#### 7.2.13.6.4 TargetPassingTime – Model Element

Out of NeTEx scope: see A.2.2.1.3

## 7.2.14 Call

### 7.2.14.1 CALL – Conceptual MODEL

Ordered collections of CALLs may be included in SERVICE JOURNEYs and DEAD RUNs exchanged with NeTEx.

A CALL provides a view of a POINT IN JOURNEY PATTERN that assembles data related to the visit to a stop of a VEHICLE JOURNEY at each SCHEDULED STOP POINT of the journey's SERVICE PATTERN and possibly other POINT IN JOURNEY PATTERNs (TIMING POINTS). It additionally can include derived data in a form convenient for data exchange and processing of journeys by displays, including :

— PASSING TIMEs, grouped by arrival and departure;

— Stop usage information (passthrough, no boarding, etc.);

— DESTINATION DISPLAY and VIA information;

— STOP ASSIGNMENTs to specific QUAYs i.e. platforms;

— Visit number (order within the VEHICLE JOURNEY, including repeated visits);

- Referenced entities and their derived properties, such as SCHEDULED STOP POINT, JOURNEY PARTs, SERVICE JOURNEY INTERCHANGES, SERVICE LINKs, etc;
- NOTICEs relating to the stop.

The use of a CALL simplifies the manipulation of data for passenger information delivery. The concept of a CALL is shared with SIRI.

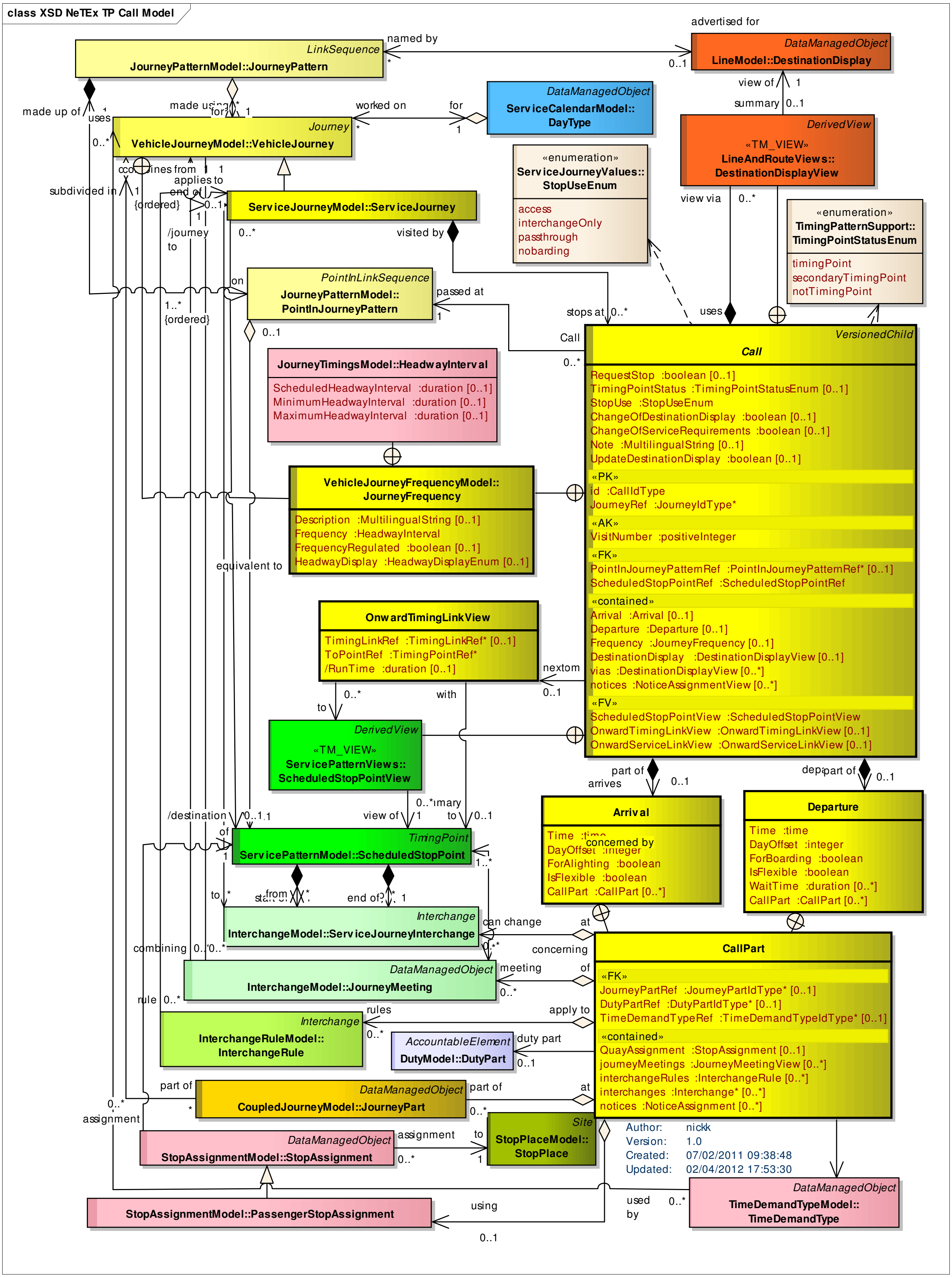

class XSD NeTEx TP Call Model
LinkSequence
JourneyPatternModel::JourneyPattern
named by
advertised for
DataManagedObject
LineModel::DestinationDisplay
0..1
view of
1
summary
0..1
DataManagedObject
ServiceCalendarModel::
DayType
worked on
for
1
made up of
uses
made using
for
1
Journey
VehicleJourneyModel::VehicleJourney
0..*
DerivedView
«TM_VIEW»
LineAndRouteViews::
DestinationDisplayView
«enumeration»
ServiceJourneyValues::
StopUseEnum
access
interchangeOnly
passthrough
nobarding
subdivided in
1
{ordered}
applies to
end of
0..1
ServiceJourneyModel::ServiceJourney
/journey
to
0..*
visited by
view via
0..*
«enumeration»
TimingPatternSupport::
TimingPointStatusEnum
timingPoint
secondaryTimingPoint
notTimingPoint
on
PointInLinkSequence
JourneyPatternModel::
PointInJourneyPattern
passed at
1
1..*
{ordered}
0..1
stops at
0..*
uses
Call
0..*
VersionedChild
Call
RequestStop :boolean [0..1]
TimingPointStatus :TimingPointStatusEnum [0..1]
StopUse :StopUseEnum
ChangeOfDestinationDisplay :boolean [0..1]
ChangeOfServiceRequirements :boolean [0..1]
Note :MultilingualString [0..1]
UpdateDestinationDisplay :boolean [0..1]
«PK»
id :CallIdType
JourneyRef :JourneyIdType*
«AK»
VisitNumber :positiveInteger
«FK»
PointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
ScheduledStopPointRef :ScheduledStopPointRef
«contained»
Arrival :Arrival [0..1]
Departure :Departure [0..1]
Frequency :JourneyFrequency [0..1]
DestinationDisplay :DestinationDisplayView [0..1]
vias :DestinationDisplayView [0..*]
notices :NoticeAssignmentView [0..*]
«FV»
ScheduledStopPointView :ScheduledStopPointView
OnwardTimingLinkView :OnwardTimingLinkView [0..1]
OnwardServiceLinkView :OnwardServiceLinkView [0..1]
JourneyTimingsModel::HeadwayInterval
ScheduledHeadwayInterval :duration [0..1]
MinimumHeadwayInterval :duration [0..1]
MaximumHeadwayInterval :duration [0..1]
VehicleJourneyFrequencyModel::
JourneyFrequency
Description :MultilingualString [0..1]
Frequency :HeadwayInterval
FrequencyRegulated :boolean [0..1]
HeadwayDisplay :HeadwayDisplayEnum [0..1]
equivalent to
OnwardTimingLinkView
TimingLinkRef :TimingLinkRef* [0..1]
ToPointRef :TimingPointRef*
/RunTime :duration [0..1]
nextom
0..1
to
0..*
with
DerivedView
«TM_VIEW»
ServicePatternViews::
ScheduledStopPointView
part of
arrives
0..1
dep part of
0..1
/destination
0..1
1
0..*
primary
view of
1
to
0..1
of
1
TimingPoint
ServicePatternModel::ScheduledStopPoint
1..*
Arrival
Time :time
DayOffset :integer
concerned by
ForAlighting :boolean
IsFlexible :boolean
CallPart :CallPart [0..*]
Departure
Time :time
DayOffset :integer
ForBoarding :boolean
IsFlexible :boolean
WaitTime :duration [0..*]
CallPart :CallPart [0..*]
to
start
from
end of
1
Interchange
InterchangeModel::ServiceJourneyInterchange
can change
at
combining
0..*
0..*
concerning
DataManagedObject
InterchangeModel::JourneyMeeting
meeting
of
0..*
CallPart
«FK»
JourneyPartRef :JourneyPartIdType* [0..1]
DutyPartRef :DutyPartIdType* [0..1]
TimeDemandTypeRef :TimeDemandTypeIdType* [0..1]
«contained»
QuayAssignment :StopAssignment [0..1]
journeyMeetings :JourneyMeetingView [0..*]
interchangeRules :InterchangeRule [0..*]
interchanges :Interchange* [0..*]
notices :NoticeAssignment [0..*]
rule
0..*
rules
apply to
Interchange
InterchangeRuleModel::
InterchangeRule
0..*
AccountableElement
DutyModel::DutyPart
duty part
0..1
part of
DataManagedObject
CoupledJourneyModel::JourneyPart
part of
at
0..*
0..*
assignment
Site
StopPlaceModel::
StopPlace
DataManagedObject
StopAssignmentModel::StopAssignment
assignment
to
0..*
1
Author: nickk
Version: 1.0
Created: 07/02/2011 09:38:48
Updated: 02/04/2012 17:53:30
StopAssignmentModel::PassengerStopAssignment
using
0..1
used
by
0..*
DataManagedObject
TimeDemandTypeModel::
TimeDemandType

**Figure 128 —Call – Physical Model**

#### 7.2.14.2 Call – Model Element

A visit to a SCHEDULED STOP POINT (or other POINT IN JOURNEY PATTERN) as part of a SERVICE JOURNEY. The CALL is a view that brings together data relating to the individual visit.

**Table 69 – *Call* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *VersionedChild* | *::>* | CALL inherits from VERSIONED CHILD |
| «PK» | ***id*** | *CallIdType* | 1:1 | Identifier of CALL. |
| «AK» | ***VisitNumber*** | *xsd:positiveInteger* | 1:1 | Repeat count of visit to the same stop with the journey. Default is 1. Will be higher for routes that visit the same stop twice. |
| «PK» | ***JourneyRef*** | *JourneyRef* | 1:1 | Reference to JOURNEY containing CALL. |
| «FK» | ***PointInJourney-PatternRef*** | *PointInJourneyPatternRef* | 0:1 | POINT IN JOURNEY PATTERN corresponding to CALL. |
| | ***RequestStop*** | *xsd:boolean* | 0:1 | Whether the stop is a Request Stop. |
| | ***TimingPointStatus*** | *TimingPointStatusEnum* | 0:1 | Timing Status of CALL. |
| | ***StopUse*** | *StopUseEnum* | 1:1 | Activity at stop. |
| «cntd» | ***Arrival*** | *Arrival* | 0:1 | Arrival part of CALL. |
| «cntd» | ***Departure*** | *Departure* | 0:1 | Departure part of CALL. |
| «cntd» | ***Frequency*** | *JourneyFrequency* | 0:1 | Frequency information for CALL. |
| «FV» | ***ScheduledStop-PointView*** | *ScheduledStopPointView* | 0:1 | Reference to the SCHEDULED STOP POINT visited by CALL. May include derived data. |
| «cntd» | ***Destination-Display*** | *DestinationDisplayView* | 0:1 | DESTINATION DISPLAY associated with the CALL. To be used from this point onwards. |
| | ***ChangeOf-Destination-Display*** | *xsd:boolean* | 0:1 | Whether DESTINATION DISPLAY changes at this point. |
| «cntd» | ***vias*** | *DestinationDisplayView* | 0:* | Set of DESTINATION DISPLAY to use to show as onward VIA points associated with the CALL. |
| | ***ChangeOfService-Requirements*** | *xsd:boolean* | 0:1 | Whether DESTINATION DISPLAY changes at this point. |
| | ***Note*** | *MultilingualString* | 0:1 | Arbitrary note associated with the CALL. For internal use. Footnotes etc., should be specified with NOTICEs. |
| «cntd» | ***noticeAssignments*** | *NoticeAssignmentView* | 0:* | NOTICE ASSIGNMENTs, footnotes etc. associated with the CALL. May include derived data. |
| «FV» | ***OnwardTiming-LinkView*** | *OnwardTimingLinkView* | 0:1 | Reference to the next TIMING LINK followed after this CALL. May include derived data. |
| «FV» | ***OnwardService-LinkView*** | *OnwardServiceLinkView* | 0:1 | Reference to the next SERVICE LINK followed after this CALL. May include derived data. |

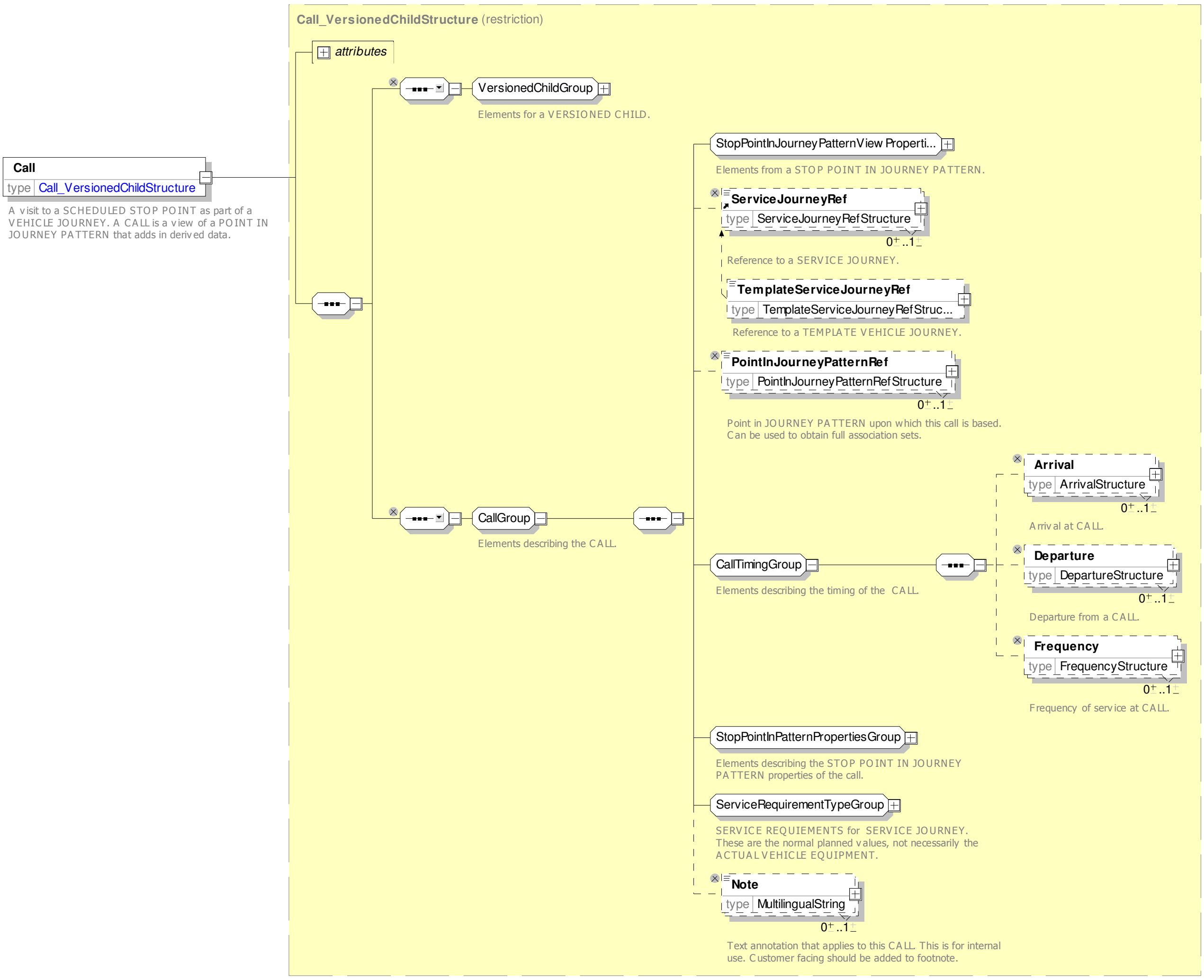

Call_VersionedChildStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
Call
type Call_VersionedChildStructure
A visit to a SCHEDULED STOP POINT as part of a VEHICLE JOURNEY. A CALL is a view of a POINT IN JOURNEY PATTERN that adds in derived data.
StopPointInJourneyPatternViewProperti...
Elements from a STOP POINT IN JOURNEY PATTERN.
ServiceJourneyRef
type ServiceJourneyRefStructure
0..1
Reference to a SERVICE JOURNEY.
TemplateServiceJourneyRef
type TemplateServiceJourneyRefStruc...
Reference to a TEMPLATE VEHICLE JOURNEY.
PointInJourneyPatternRef
type PointInJourneyPatternRefStructure
0..1
Point in JOURNEY PATTERN upon which this call is based. Can be used to obtain full association sets.
Arrival
type ArrivalStructure
0..1
Arrival at CALL.
CallGroup
Elements describing the CALL.
CallTimingGroup
Elements describing the timing of the CALL.
Departure
type DepartureStructure
0..1
Departure from a CALL.
Frequency
type FrequencyStructure
0..1
Frequency of service at CALL.
StopPointInPatternPropertiesGroup
Elements describing the STOP POINT IN JOURNEY PATTERN properties of the call.
ServiceRequirementTypeGroup
SERVICE REQUIEMENTS for SERVICE JOURNEY. These are the normal planned values, not necessarily the ACTUAL VEHICLE EQUIPMENT.
Note
type MultilingualString
0..1
Text annotation that applies to this CALL. This is for internal use. Customer facing should be added to footnote.

Figure 129 — *Call* — XSD

### 7.2.14.2.1.1 StopPointInJourneyPatternViewPropertiesGroup – Group

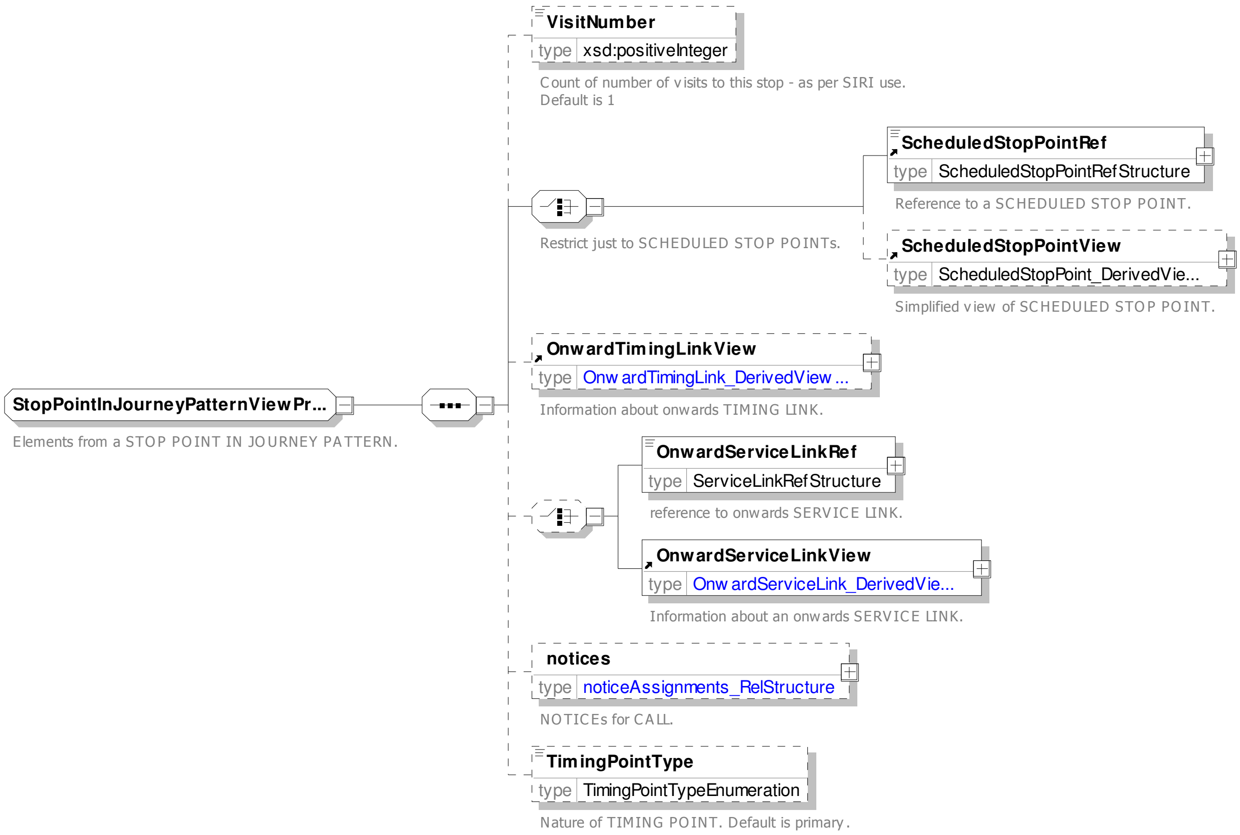

VisitNumber
type xsd:positiveInteger
Count of number of visits to this stop - as per SIRI use. Default is 1
ScheduledStopPointRef
type ScheduledStopPointRefStructure
Reference to a SCHEDULED STOP POINT.
Restrict just to SCHEDULED STOP POINTs.
ScheduledStopPointView
type ScheduledStopPoint_DerivedVie...
Simplified view of SCHEDULED STOP POINT.
OnwardTimingLinkView
type OnwardTimingLink_DerivedView ...
Information about onwards TIMING LINK.
StopPointInJourneyPatternViewPr...
Elements from a STOP POINT IN JOURNEY PATTERN.
OnwardServiceLinkRef
type ServiceLinkRefStructure
reference to onwards SERVICE LINK.
OnwardServiceLinkView
type OnwardServiceLink_DerivedVie...
Information about an onwards SERVICE LINK.
notices
type noticeAssignments_RelStructure
NOTICEs for CALL.
TimingPointType
type TimingPointTypeEnumeration
Nature of TIMING POINT. Default is primary.

**Figure 130 — *StopPointInJourneyPatternViewPropertiesGroup* — XSD**

### 7.2.14.2.1.2 TimingPointStatus – Allowed values

The following table shows the allowed values for *TimingPointStatus* **(***TimingPointStatus*E*num).*

**Table 70 – TimingPointStatus – Allowed values**

| Value | Description |
|---|---|
| *timingPoint* | Timing Point |
| *secondaryTimingPoint* | Secondary Timing Point |
| *notTimingPoint* | Not Timing Point |

#### 7.2.14.2.1.3 StopPointInPatternViewPropertiesGroup – Group

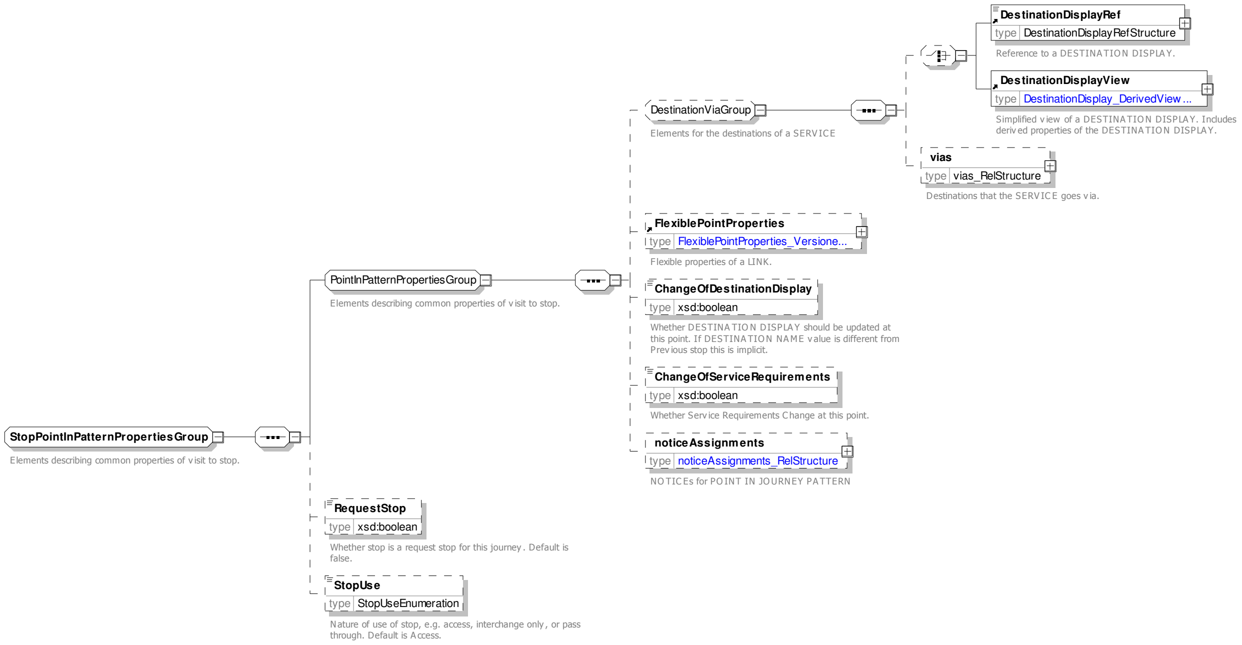

DestinationDisplayRef
type DestinationDisplayRefStructure
Reference to a DESTINATION DISPLAY.
DestinationDisplayView
type DestinationDisplay_DerivedView ...
Simplified view of a DESTINATION DISPLAY. Includes derived properties of the DESTINATION DISPLAY.
DestinationViaGroup
Elements for the destinations of a SERVICE
vias
type vias_RelStructure
Destinations that the SERVICE goes via.
FlexiblePointProperties
type FlexiblePointProperties_Versione...
Flexible properties of a LINK.
PointInPatternPropertiesGroup
Elements describing common properties of visit to stop.
ChangeOfDestinationDisplay
type xsd:boolean
Whether DESTINATION DISPLAY should be updated at this point. If DESTINATION NAME value is different from Previous stop this is implicit.
ChangeOfServiceRequirements
type xsd:boolean
Whether Service Requirements Change at this point.
StopPointInPatternPropertiesGroup
Elements describing common properties of visit to stop.
noticeAssignments
type noticeAssignments_RelStructure
NOTICEs for POINT IN JOURNEY PATTERN
RequestStop
type xsd:boolean
Whether stop is a request stop for this journey. Default is false.
StopUse
type StopUseEnumeration
Nature of use of stop, e.g. access, interchange only, or pass through. Default is Access.

Figure 131 — *StopPointInPatternPropertiesGroup* — XSD

### 7.2.14.3 Arrival – Model Element

The ARRIVAL of a SERVICE JOURNEY to make a CALL at a SCHEDULED STOP POINT.

Table 71 – *Arrival* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***Time*** | *xsd:time* | 1:1 | Latest Time of Arrival. |
| | ***DayOffset*** | *xsd:integer* | 1:1 | Day offset from SERVICE JOURNEY start day. 0 = Same day. |
| | ***ForAlighting*** | *xsd:boolean* | 1:1 | Whether alighting is allowed for CALL. |
| | ***IsFlexible*** | *xsd:boolean* | 1:1 | Whether use of stop is flexible. |
| | ***CallPart*** | *CallPart* | 0:* | Common elements, e.g. NOTICEs, QUAY ASSIGNMENT**,** INTERCHANGEs and JOURNEY PART associated with Arrival part. of CALL |

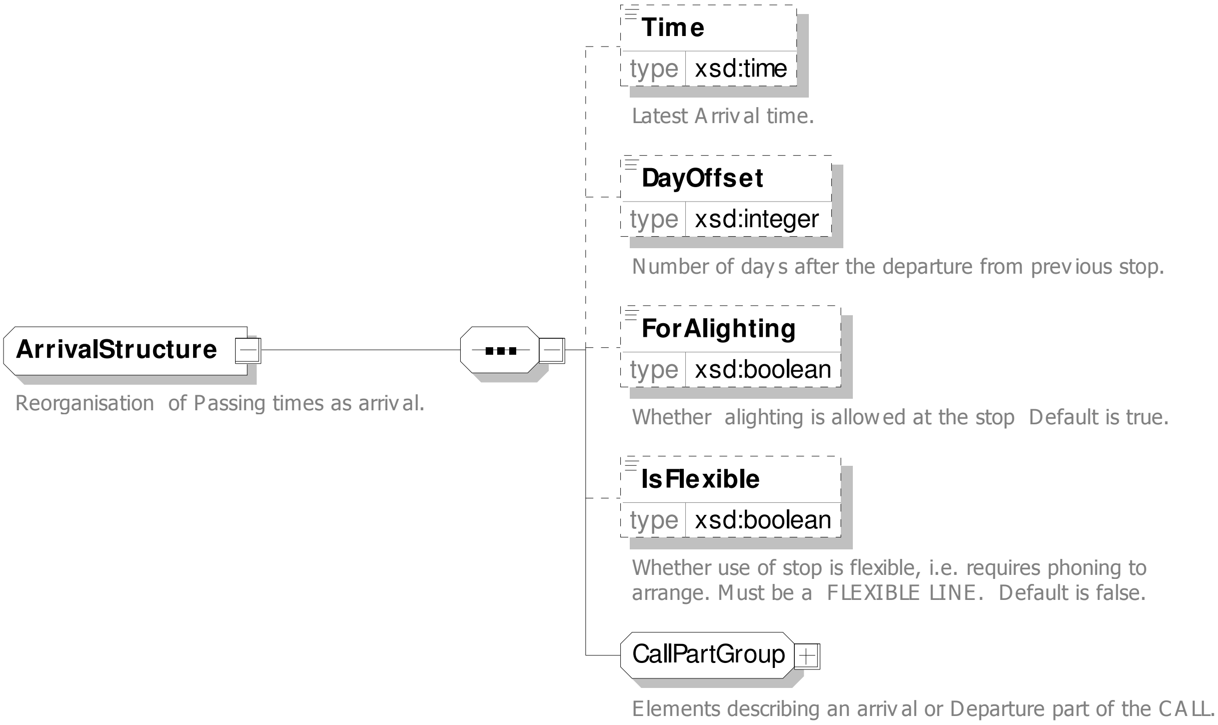

ArrivalStructure
Reorganisation of Passing times as arrival.
Time
type xsd:time
Latest Arrival time.
DayOffset
type xsd:integer
Number of days after the departure from previous stop.
ForAlighting
type xsd:boolean
Whether alighting is allowed at the stop Default is true.
IsFlexible
type xsd:boolean
Whether use of stop is flexible, i.e. requires phoning to arrange. Must be a FLEXIBLE LINE. Default is false.
CallPartGroup
Elements describing an arrival or Departure part of the CALL.

Figure 132 — *ArrivalStructure* — XSD

#### 7.2.14.4 Departure – Model Element

The DEPARTURE of a SERVICE JOURNEY from making a CALL at a SCHEDULED STOP POINT.

Table 72 – *Departure* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***Time*** | *xsd:time* | 1:1 | Earliest time of departure. |
| | ***DayOffset*** | *xsd:integer* | 1:1 | Day offset from SERVICE JOURNEY start day. 0 = Same day. |
| | ***ForBoarding*** | *xsd:boolean* | 1:1 | Whether boarding is allowed for CALL. |
| | ***IsFlexible*** | *xsd:boolean* | 1:1 | Whether use of stop is flexible. |
| | ***WaitTime*** | *xsd:duration* | 0:* | Time to wait at stop after arrival before departure. |
| | ***CallPart*** | *CallPart* | 0:* | Common elements, e.g. NOTICEs, QUAY ASSIGNMENT, INTERCHANGEs and JOURNEY PART associated with departure part of CALL. |

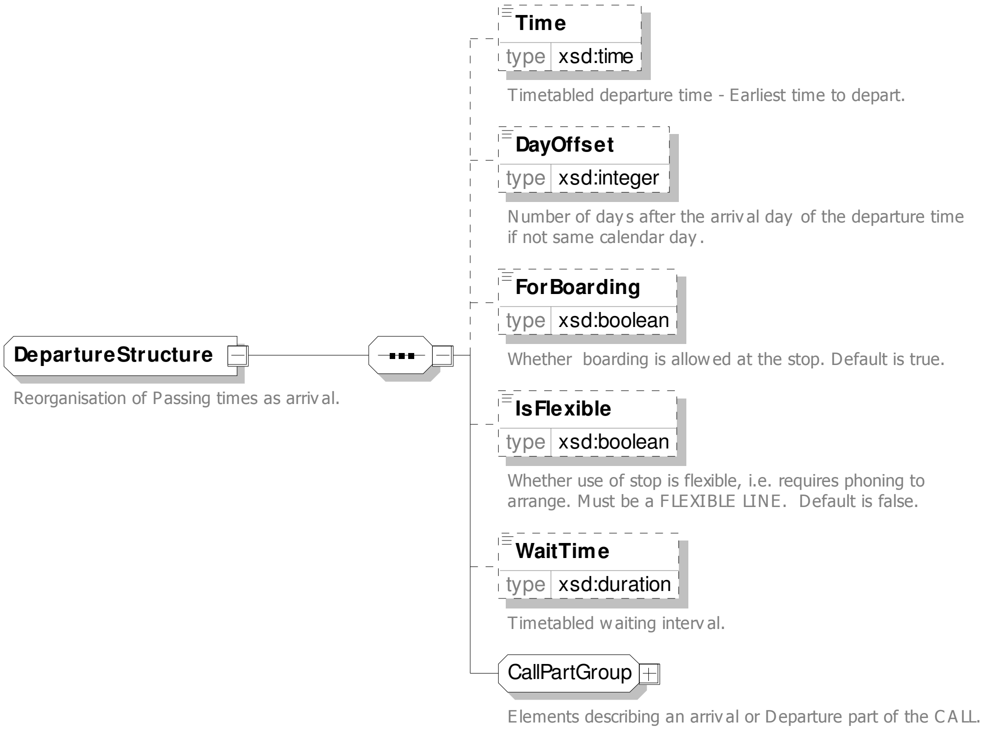

Time
type xsd:time
Timetabled departure time - Earliest time to depart.
DayOffset
type xsd:integer
Number of days after the arrival day of the departure time if not same calendar day.
ForBoarding
type xsd:boolean
Whether boarding is allowed at the stop. Default is true.
DepartureStructure
Reorganisation of Passing times as arrival.
IsFlexible
type xsd:boolean
Whether use of stop is flexible, i.e. requires phoning to arrange. Must be a FLEXIBLE LINE. Default is false.
WaitTime
type xsd:duration
Timetabled waiting interval.
CallPartGroup
Elements describing an arrival or Departure part of the CALL.

Figure 133 — *DepartureStructure* — XSD

### 7.2.14.5 OnwardServiceLinkView – Model Element

The ***OnwardServiceLinkView*** can be used to provide information about the onward SERVICE LINK from the CALL.

Table 73 – *OnwardServiceLinkView* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***ServiceLinkRef*** | *ServiceLinkRef* | 0:1 | Reference to onwards SERVICE LINK. |
| | ***ToPointRef*** | *ScheduledStopPointRef* | 1:1 | Reference to next SCHEDULED STOP POINT in pattern. |
| | ***Distance*** | *DistanceType* | 0:1 | DISTANCE for onward SERVICE LINK. |
| | ***RunTime*** | *xsd:duration* | 0:1 | RUN TIME for onward SERVICE LINK. |

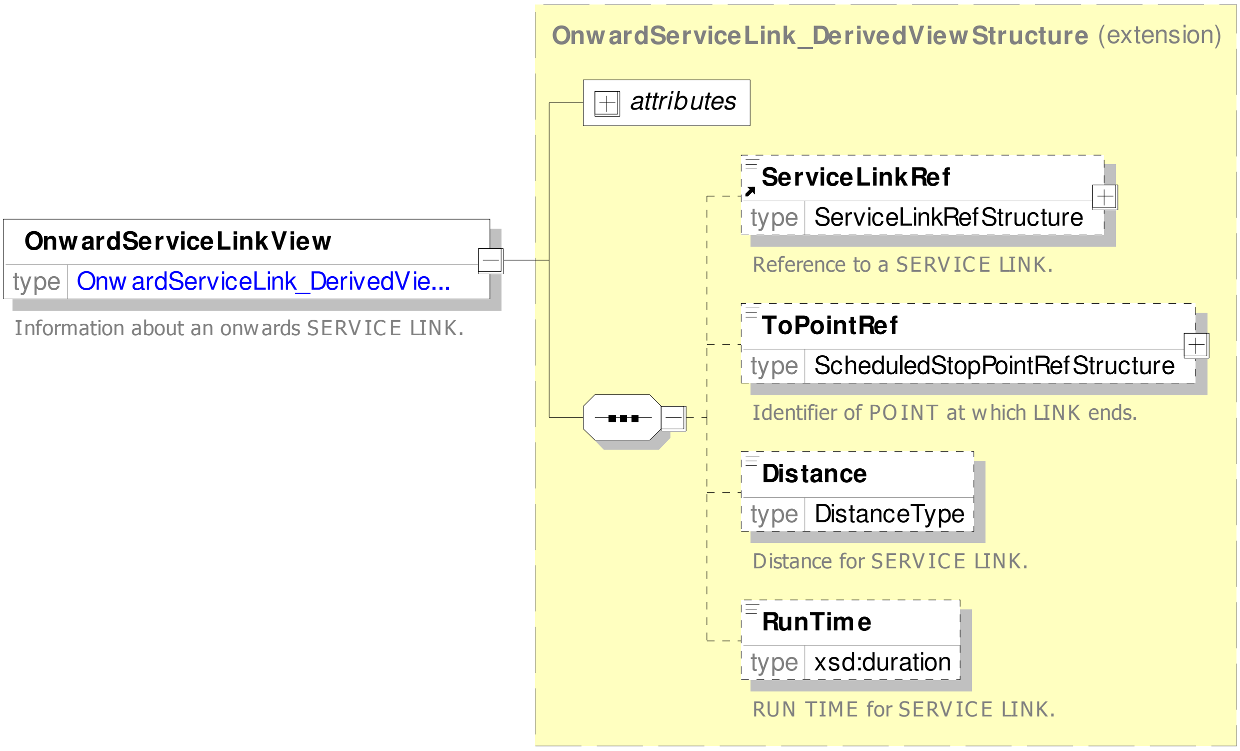

OnwardServiceLink_DerivedViewStructure (extension)
attributes
ServiceLinkRef
type ServiceLinkRefStructure
Reference to a SERVICE LINK.
OnwardServiceLinkView
type OnwardServiceLink_DerivedVie...
Information about an onwards SERVICE LINK.
ToPointRef
type ScheduledStopPointRefStructure
Identifier of POINT at which LINK ends.
Distance
type DistanceType
Distance for SERVICE LINK.
RunTime
type xsd:duration
RUN TIME for SERVICE LINK.

Figure 134 — *OnwardServiceLinkView* — XSD

### 7.2.14.6 OnwardTimingLinkView – Model Element

The OnwardServiceLinkView can be used to provide information about the onward TIMING LINK from the CALL.

Table 74 – *OnwardTimingLinkView* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***TimingLinkRef*** | *TimingLinkRef* | 0:1 | Reference to onwards TIMING LINK. |
| | ***ToPointRef*** | *TimingPointRef* | 1:1 | Reference to next SCHEDULED STOP POINT in pattern. |
| | ***RunTime*** | *xsd:duration* | 0:1 | RUN TIME for onward TIMING LINK. |

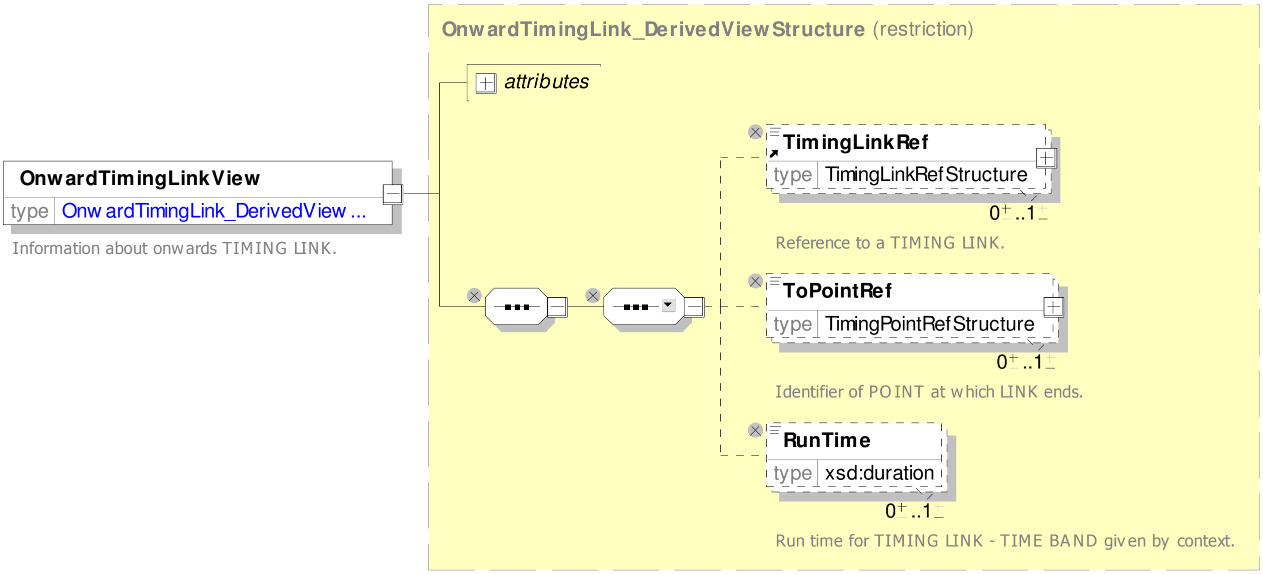

OnwardTimingLink_DerivedViewStructure (restriction)
attributes
OnwardTimingLinkView
type OnwardTimingLink_DerivedView ...
Information about onwards TIMING LINK.
TimingLinkRef
type TimingLinkRefStructure
0..1
Reference to a TIMING LINK.
ToPointRef
type TimingPointRefStructure
0..1
Identifier of POINT at which LINK ends.
RunTime
type xsd:duration
0..1
Run time for TIMING LINK - TIME BAND given by context.

Figure 135 — *OnwardTimingLinkView* — XSD

### 7.2.14.7 CallPartGroup – Model Element

A ***CallPart*** describes common properties of an ***Arrival*** or ***Departure*** of a CALL.

Table 75 – *CallPartGroup* – Group

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| «FK» | ***JourneyPartRef*** | *JourneyPartRef* | 0:1 | JOURNEY PART to which CALL belongs. |
| «cntd» | ***journeyMeetings*** | *JourneyMeetingView* | 0:* | JOURNEY MEETINGs of this journey at this specific CALL PART. |
| «cntd» | ***interchanges*** | *Interchange* | 0:* | INTERCHANGE of this journey at this specific CALL PART. |
| «cntd» | ***interchangeRules*** | *InterchangeRule* | 0:* | INTERCHANGE RULEs of this journey at this specific CALL PART. |
| «FK» | ***TimeDemandTypeRef*** | *TimeDemandTypeRef* | 0:1 | TIME DEMAND TYPE of this journey at this specific CALL. |
| «FK» | ***DutyPartRef*** | *DutyPartRef* | 0:1 | DUTY that applies from this CALL onwards. |
| «cntd» | ***QuayAssignment*** | *StopAssignment* | 0:1 | Assignment to a QUAY for this CALL PART. |
| «cntd» | ***notice-Assignments*** | *NoticeAssignment* | 0:* | NOTICEs that apply at this specific CALL PART. |

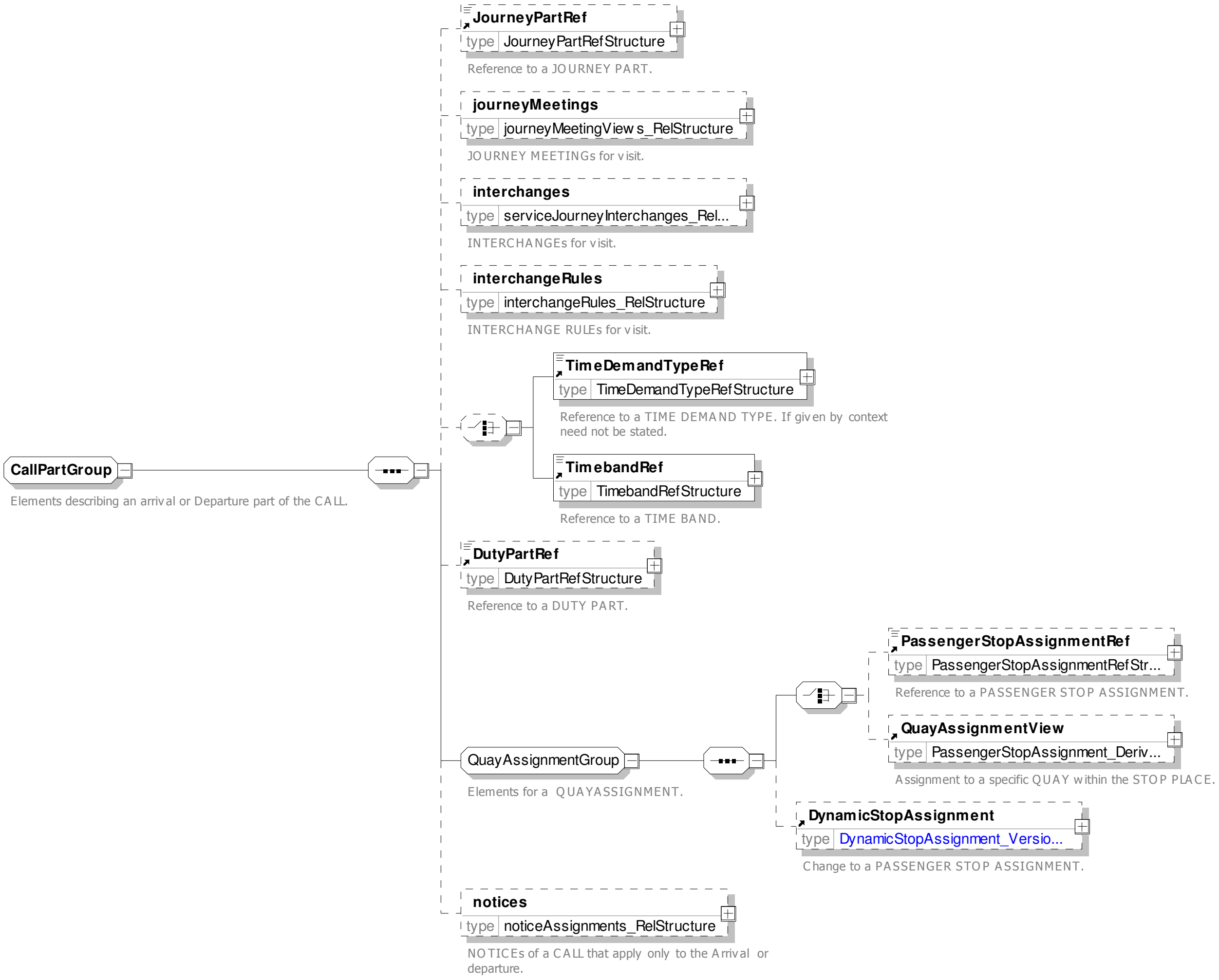

JourneyPartRef
type JourneyPartRefStructure
Reference to a JOURNEY PART.
journeyMeetings
type journeyMeetingViews_RelStructure
JOURNEY MEETINGs for visit.
interchanges
type serviceJourneyInterchanges_Rel...
INTERCHANGEs for visit.
interchangeRules
type interchangeRules_RelStructure
INTERCHANGE RULEs for visit.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
TimebandRef
type TimebandRefStructure
Reference to a TIME BAND.
CallPartGroup
Elements describing an arrival or Departure part of the CALL.
DutyPartRef
type DutyPartRefStructure
Reference to a DUTY PART.
PassengerStopAssignmentRef
type PassengerStopAssignmentRefStr...
Reference to a PASSENGER STOP ASSIGNMENT.
QuayAssignmentView
type PassengerStopAssignment_Deriv...
Assignment to a specific QUAY within the STOP PLACE.
QuayAssignmentGroup
Elements for a QUAYASSIGNMENT.
DynamicStopAssignment
type DynamicStopAssignment_Versio...
Change to a PASSENGER STOP ASSIGNMENT.
notices
type noticeAssignments_RelStructure
NOTICEs of a CALL that apply only to the Arrival or departure.

Figure 136 — *CallPartGroup* — XSD

### 7.2.14.8 DeadRunCall – Model Element

A visit to a POINT as part of a DEAD RUN. The DEAD RUN CALL is a view that brings together data relating to the visit to the individual point.

Table 76 – *DeadRunCall* – Element

| Classification | Name | | Type | Cardin-ality | Description |
|---|---|---|---|---|---|
| ::> | ::> | | *VersionedChild* | ::> | DEAD RUN CALL inherits from VERSIONED CHILD. |
| «PK» | ***id*** | | *DeadRunCallIdType* | 1:1 | Identifier of DEAD RUN. |
| | ***order*** | | *xsd:integer* | 0:1 | Order of DEAD RUN CALL. |
| | ***Point*** | | *choice* | 0:1 | POINT visited by call. |
| «FK» | ***a*** | ***RoutePointRef*** | ***RoutePointRef*** | 0:1 | Reference to the ROUTE STOP POINT visited by DEAD RUN CALL. |
| «FK» | ***b*** | ***TimingPointRef*** | ***TimingPointRef*** | 0:1 | Reference to the TIMING STOP POINT visited by |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | DEAD RUN CALL. |
| «FK» | ***c*** | ***TimingPointRef*** | | 0:1 | Reference to the SCHEDULED STOP POINT visited by CALL. |
| «FV» | ***d*** | ***Scheduled-StopPointView*** | *ScheduledStopPointView* | 0:1 | Reference to the SCHEDULED STOP POINT visited by CALL. May include derived data. |
| | ***TimingPointStatus*** | | *TimingPointStattsEnum* | 0:1 | Nature of TIMING POINT. |
| | ***JourneyRef*** | | *DeadRunRef* | 1:1 | Reference to JOURNEY containing CALL. |
| «FK» | ***PointInJourney-PatternRef*** | | *PointInJourneyPatternRef* | 0:1 | Reference to POINT IN JOURNEY PATTERN for CALL. |
| «cntd» | ***Arrival*** | | *DeadRunArrival* | 0:1 | Arrival part of DEAD RUN CALL. |
| «cntd» | ***Departure*** | | *DeadRunDeparture* | 0:1 | Departure part of DEAD RUN CALL. |
| | ***Note*** | | *MultilingualString* | 0:1 | Note on CALL. |
| «cntd» | ***Destination-Display*** | | *DestinationDisplayView* | 0:1 | DESTINATION DISPLAY associated with the CALL. |
| «cntd» | ***vias*** | | *DestinationDisplayView* | 0:* | Set of via names to use to show as onward via points associated with the CALL. |
| «FV» | ***OnwardTiming-LinkView*** | | *OnwardTimingLinkView* | 0:1 | Onward TIMING LINK for next link from CALL stop. |

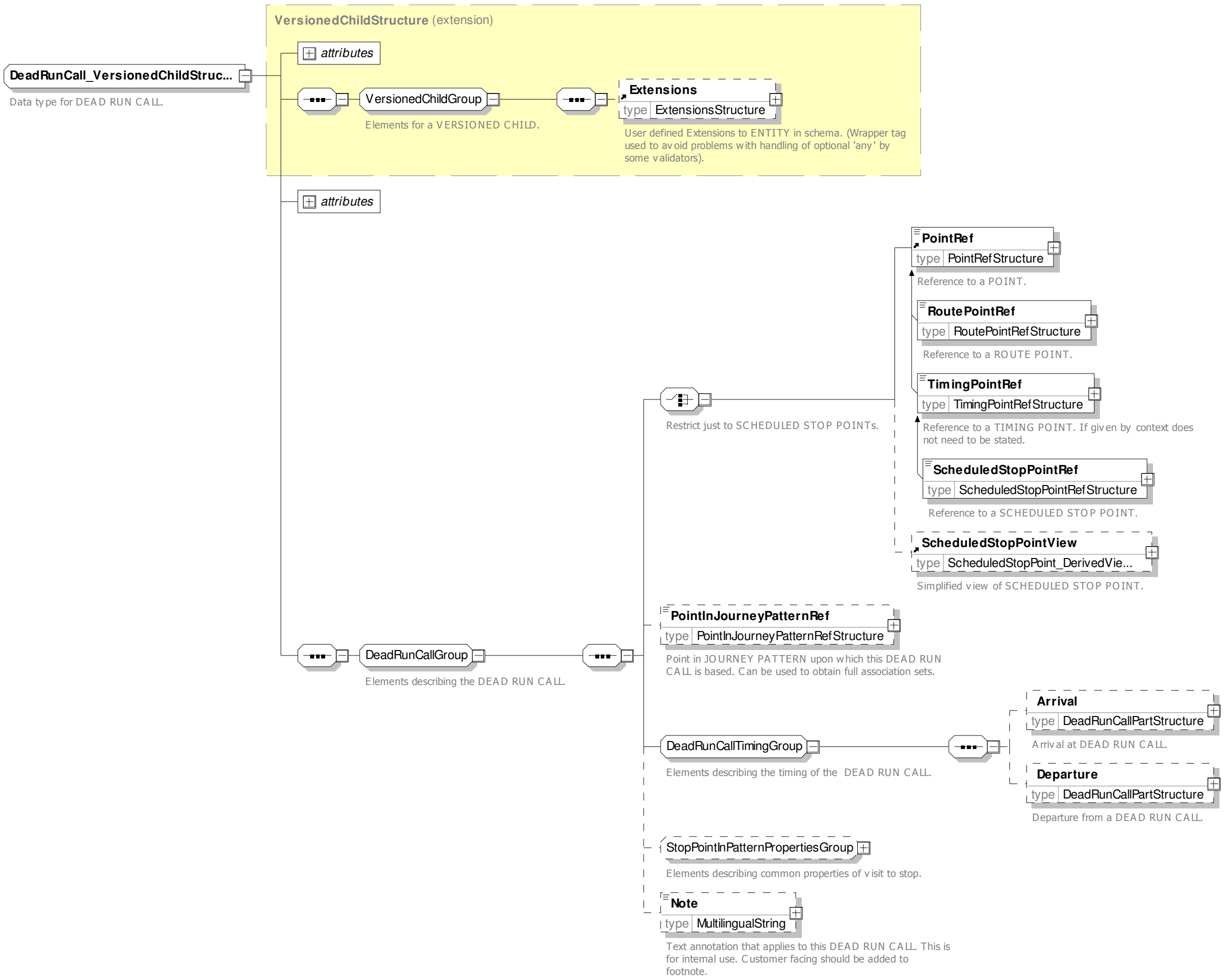

VersionedChildStructure (extension)
attributes
DeadRunCall_VersionedChildStruc...
Data type for DEAD RUN CALL.
VersionedChildGroup
Elements for a VERSIONED CHILD.
Extensions
type ExtensionsStructure
User defined Extensions to ENTITY in schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
attributes
PointRef
type PointRefStructure
Reference to a POINT.
RoutePointRef
type RoutePointRefStructure
Reference to a ROUTE POINT.
TimingPointRef
type TimingPointRefStructure
Reference to a TIMING POINT. If given by context does not need to be stated.
Restrict just to SCHEDULED STOP POINTs.
ScheduledStopPointRef
type ScheduledStopPointRefStructure
Reference to a SCHEDULED STOP POINT.
ScheduledStopPointView
type ScheduledStopPoint_DerivedVie...
Simplified view of SCHEDULED STOP POINT.
PointInJourneyPatternRef
type PointInJourneyPatternRefStructure
Point in JOURNEY PATTERN upon which this DEAD RUN CALL is based. Can be used to obtain full association sets.
DeadRunCallGroup
Elements describing the DEAD RUN CALL.
Arrival
type DeadRunCallPartStructure
Arrival at DEAD RUN CALL.
DeadRunCallTimingGroup
Elements describing the timing of the DEAD RUN CALL.
Departure
type DeadRunCallPartStructure
Departure from a DEAD RUN CALL.
StopPointInPatternPropertiesGroup
Elements describing common properties of visit to stop.
Note
type MultilingualString
Text annotation that applies to this DEAD RUN CALL. This is for internal use. Customer facing should be added to footnote.

Figure 137 — *DeadRunCall* — XSD

### 7.2.14.9 DeadRunArrival – Model Element

The ARRIVAL of a SERVICE JOURNEY DEAD RUN to make a CALL at a POINT IN JOURNEY PATTERN SCHEDULED STOP POINT.

Table 77 – *DeadRunArrival* – Element

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| | ***Time*** | *time* | 1:1 | Latest time of arrival. |
| | ***DayOffset*** | *integer* | 1:1 | Day offset from DEAD RUN start day. 0 = Same day. |
| | ***CallPart*** | *DeadRunCallPart*Group | 0:* | Common properties of a DEAD RUN CALL arrival. |

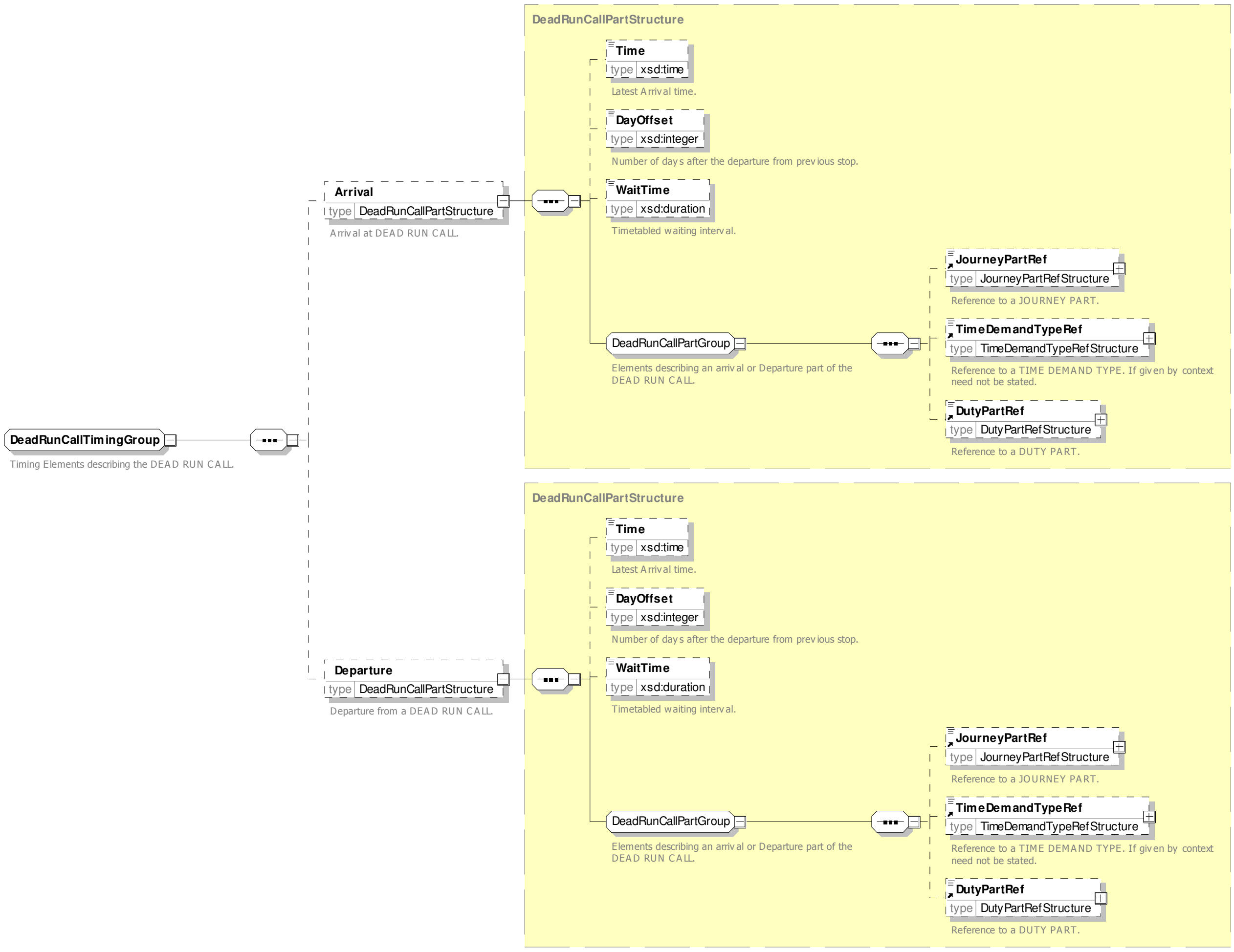

DeadRunCallTimingGroup
Timing Elements describing the DEAD RUN CALL.
Arrival
type DeadRunCallPartStructure
Arrival at DEAD RUN CALL.
DeadRunCallPartStructure
Time
type xsd:time
Latest Arrival time.
DayOffset
type xsd:integer
Number of days after the departure from previous stop.
WaitTime
type xsd:duration
Timetabled waiting interval.
DeadRunCallPartGroup
Elements describing an arrival or Departure part of the DEAD RUN CALL.
JourneyPartRef
type JourneyPartRefStructure
Reference to a JOURNEY PART.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
DutyPartRef
type DutyPartRefStructure
Reference to a DUTY PART.
Departure
type DeadRunCallPartStructure
Departure from a DEAD RUN CALL.
DeadRunCallPartStructure
Time
type xsd:time
Latest Arrival time.
DayOffset
type xsd:integer
Number of days after the departure from previous stop.
WaitTime
type xsd:duration
Timetabled waiting interval.
DeadRunCallPartGroup
Elements describing an arrival or Departure part of the DEAD RUN CALL.
JourneyPartRef
type JourneyPartRefStructure
Reference to a JOURNEY PART.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
DutyPartRef
type DutyPartRefStructure
Reference to a DUTY PART.

**Figure 138 — *DeadRunTimingGroup* — XSD**

### 7.2.14.10 DeadRunDeparture – Model Element

The DEPARTURE of a DEAD RUN from making a CALL at a POINT IN JOURNEY PATTERN.

**Table 78 – *DeadRunDeparture* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| | ***Time*** | *xsd:time* | 1:1 | Earliest Time of Departure. |
| | ***DayOffset*** | *xsd:integer* | 1:1 | Day offset from DEAD RUN start day. 0 = Same day. |
| | ***WaitTime*** | *xsd:duration* | 0:* | Time to wait at stop after arrival before departure. |
| | ***CallPart*** | *DeadRunCallPartGroup* | 0:* | Common properties of a DEAD RUN CALL Departure. |

### 7.2.14.11 DeadRunCallPart – Model Element

The dead run call part describes common properties of a DEAD RUN CALL Arrival and Departure.

Table 79 – *DeadRunCallPart* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| «FK» | ***JourneyPartRef*** | *JourneyPartRef* | 0:1 | JOURNEY PART to which CALL belongs. |
| «FK» | ***DutyPartRef*** | *DutyPartRef* | 0:1 | DUTY that applies from this CALL onwards. |
| «FK» | ***TimeDemandTypeRef*** | *TimeDemandTypeRef* | 0:1 | JOURNEY MEETINGs of this journey at this specific DEAD RUN CALL. |

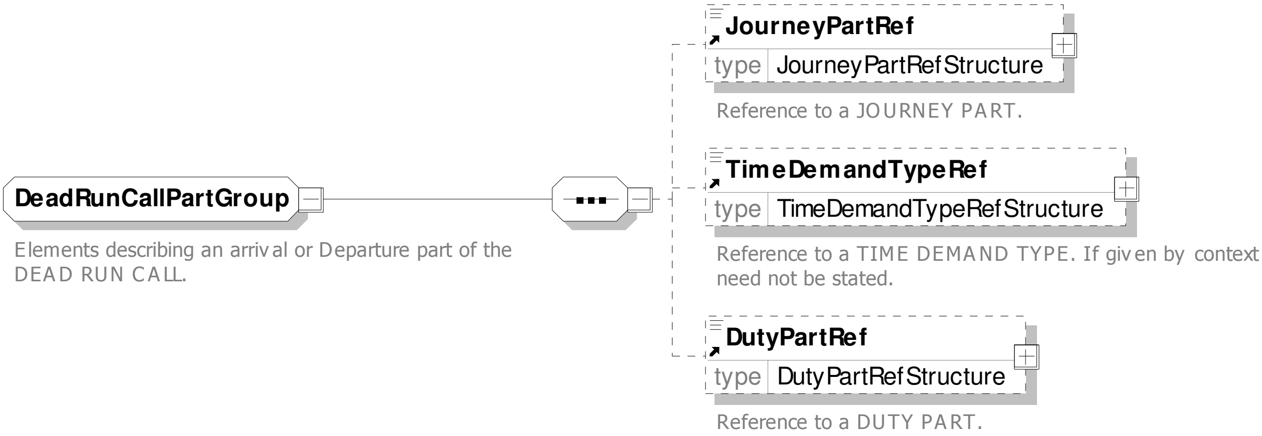

DeadRunCallPartGroup
Elements describing an arrival or Departure part of the DEAD RUN CALL.
JourneyPartRef
type JourneyPartRefStructure
Reference to a JOURNEY PART.
TimeDemandTypeRef
type TimeDemandTypeRefStructure
Reference to a TIME DEMAND TYPE. If given by context need not be stated.
DutyPartRef
type DutyPartRefStructure
Reference to a DUTY PART.

Figure 139 — *DeadRunCallPart* — XSD

### 7.2.14.12 XML Examples of Calls

Some examples of CALLs can be found in the examples for SERVICE JOURNEY and COUPLED JOURNEY.

#### 7.2.14.12.1 Calls for a circular route — XML Example

The following code fragment shows a SERVICE JOURNEY with five CALLs (*'A-B-C-D-A')* forming a circular route. The DESTINATION DISPLAY changes as it moves around the circle. The second visit to stop Alpha has a visit number of 2.

For EXAMPLE

```
<ServiceJourney version="any" id="hde:sj_42c_AE_01">
    <DepartureTime>14:00:00.0Z</DepartureTime>
    <dayTypes>
        <DayTypeRef version="any" ref="hde:DT_01-MF-NH"/>
    </dayTypes>
    <LineRef version="any" ref="mybus:LN_42"/>
    <runTimes>
        <VehicleJourneyRunTime id="hde:vjrt_sj_42c_AE_01">
            <Name>Overall run time</Name>
            <RunTime>PT75M</RunTime>
        </VehicleJourneyRunTime>
    </runTimes>
    <calls>
        <Call id="hde:sj_42c_AE_01_Alpha1" order="1">
            <VisitNumber>1</VisitNumber>
            <Arrival>
                <ForAlighting>false</ForAlighting>
            </Arrival>
            <Departure>
```

```
                <Time>14:00:00.0Z</Time>
            </Departure>
            <DestinationDisplayRef version="any" ref="mybus:DST_Charley"/>
            <ChangeOfDestinationDisplay>true</ChangeOfDestinationDisplay>
        </Call>
        <Call id="hde:sj_42c_AE_01_Bravo1" order="2">
            <VisitNumber>1</VisitNumber>
            <ScheduledStopPointRef version="any" ref="mybus:Bravo"/>
            <Departure>
                <Time>14:15:00.0Z</Time>
            </Departure>
            <DestinationDisplayRef version="any" ref="mybus:DST_Charley"/>
        </Call>
        <Call id="hde:sj_42c_AE_01_Charley1" order="3">
            <VisitNumber>1</VisitNumber>
            <ScheduledStopPointRef version="any" ref="mybus:Charley"/>
            <Departure>
                <Time>14:30:00.0Z</Time>
            </Departure>
            <DestinationDisplayRef version="any" ref="mybus:DST_Alpha"/>
            <ChangeOfDestinationDisplay>true</ChangeOfDestinationDisplay>
        </Call>
        <Call id="hde:sj_42c_AE_01_Delta1" order="4">
            <VisitNumber>1</VisitNumber>
            <ScheduledStopPointRef version="any" ref="mybus:Delta"/>
            <Departure>
                <Time>14:45:00.0Z</Time>
            </Departure>
            <DestinationDisplayRef version="any" ref="mybus:DST_Alpha"/>
        </Call>
        <Call id="hde:sj_42c_AE_01_Alpha2" order="6">
            <VisitNumber>2</VisitNumber>
            <ScheduledStopPointRef version="any" ref="mybus:Alpha"/>
            <Arrival>
                <Time>15:15:00.0Z</Time>
            </Arrival>
            <Departure>
            </Departure>
            <DestinationDisplayRef version="any" ref="mybus:DST_Charley"/>
            <ChangeOfDestinationDisplay>true</ChangeOfDestinationDisplay>
        </Call>
    </calls>
</ServiceJourney>
```

### 7.2.15 Dated Call

#### 7.2.15.1 DATED CALL – Conceptual MODEL

A DATED CALL refines a CALL element and provides a view of a POINT IN JOURNEY PATTERN that assembles data related to the visit to a stop of a VEHICLE JOURNEY at a SCHEDULED STOP POINT.

— PASSING TIMEs, grouped by arrival and departure.

— Stop usage information (passthrough, no boarding, etc.).

— DESTINATION DISPLAY and VIA information.

— STOP ASSIGNMENTs to specific QUAYs i.e. platforms.

— Visit number (order within the VEHICLE JOURNEY, including repeated visits).

— Referenced entities and their derived properties, such as SCHEDULED STOP POINT, JOURNEY PARTs, SERVICE JOURNEY INTERCHANGES, SERVICE LINKs, etc.

— NOTICEs relating to the CALL.

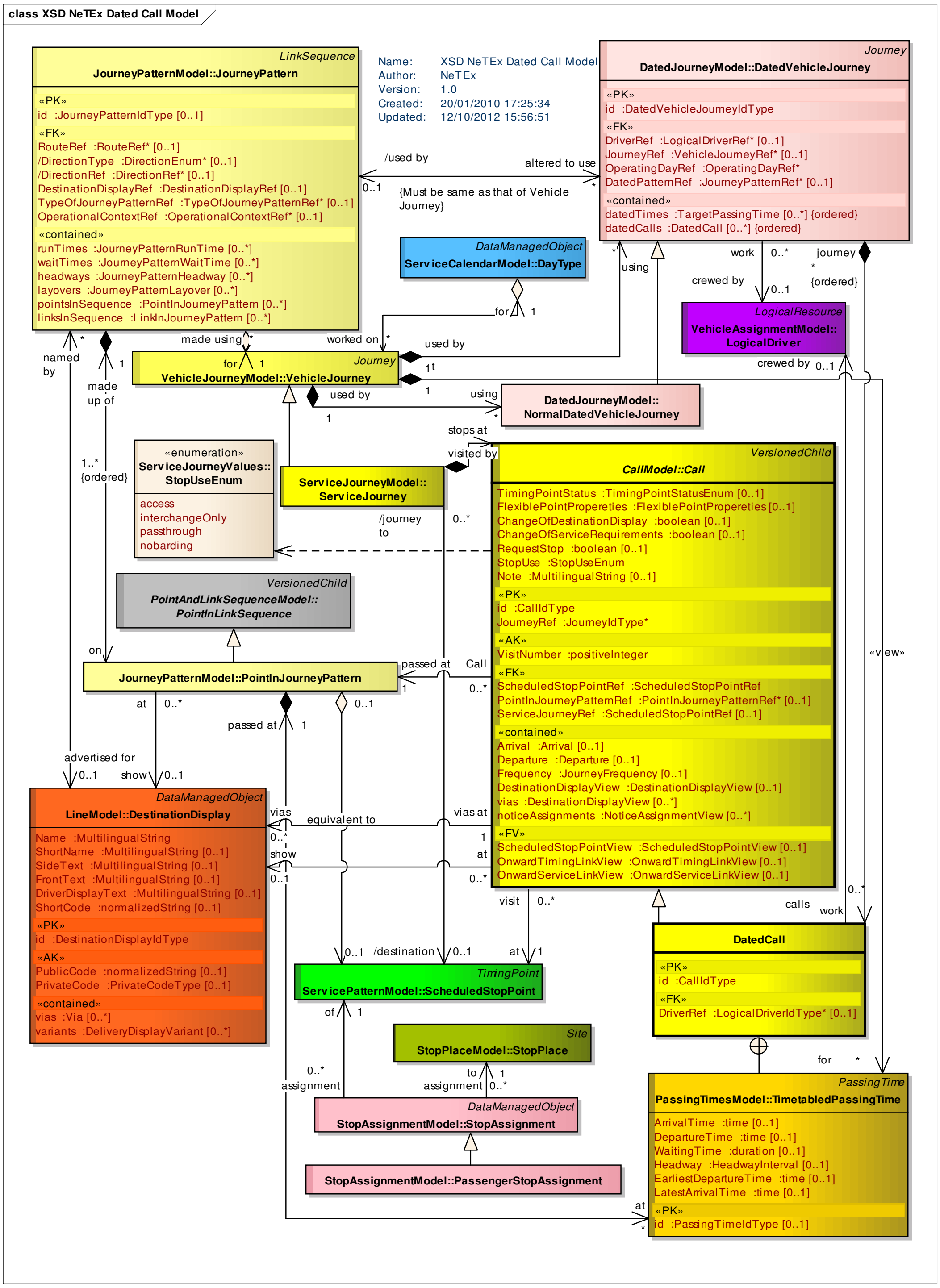

class XSD NeTEx Dated Call Model
Name: XSD NeTEx Dated Call Model
Author: NeTEx
Version: 1.0
Created: 20/01/2010 17:25:34
Updated: 12/10/2012 15:56:51
LinkSequence
JourneyPatternModel::JourneyPattern
«PK»
id :JourneyPatternIdType [0..1]
«FK»
RouteRef :RouteRef* [0..1]
/DirectionType :DirectionEnum* [0..1]
/DirectionRef :DirectionRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef [0..1]
TypeOfJourneyPatternRef :TypeOfJourneyPatternRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
«contained»
runTimes :JourneyPatternRunTime [0..*]
waitTimes :JourneyPatternWaitTime [0..*]
headways :JourneyPatternHeadway [0..*]
layovers :JourneyPatternLayover [0..*]
pointsInSequence :PointInJourneyPattern [0..*]
linksInSequence :LinkInJourneyPattern [0..*]
Journey
DatedJourneyModel::DatedVehicleJourney
«PK»
id :DatedVehicleJourneyIdType
«FK»
DriverRef :LogicalDriverRef* [0..1]
JourneyRef :VehicleJourneyRef* [0..1]
OperatingDayRef :OperatingDayRef*
DatedPatternRef :JourneyPatternRef* [0..1]
«contained»
datedTimes :TargetPassingTime [0..*] {ordered}
datedCalls :DatedCall [0..*] {ordered}
/used by
altered to use
0..1
*
{Must be same as that of Vehicle Journey}
DataManagedObject
ServiceCalendarModel::DayType
for
1
using
work
0..*
journey
*
{ordered}
crewed by
0..1
LogicalResource
VehicleAssignmentModel::LogicalDriver
crewed by
0..1
made using
worked on
used by
named by
made up of
Journey
VehicleJourneyModel::VehicleJourney
for
1
1
1
used by
1
using
*
DatedJourneyModel::NormalDatedVehicleJourney
stops at
visited by
«enumeration»
ServiceJourneyValues::StopUseEnum
access
interchangeOnly
passthrough
nobarding
ServiceJourneyModel::ServiceJourney
/journey to
0..*
1..*
{ordered}
VersionedChild
PointAndLinkSequenceModel::PointInLinkSequence
on
JourneyPatternModel::PointInJourneyPattern
passed at
Call
1
0..*
at
0..*
0..1
passed at
1
advertised for
0..1
show
0..1
VersionedChild
CallModel::Call
TimingPointStatus :TimingPointStatusEnum [0..1]
FlexiblePointProperies :FlexiblePointProperies [0..1]
ChangeOfDestinationDisplay :boolean [0..1]
ChangeOfServiceRequirements :boolean [0..1]
RequestStop :boolean [0..1]
StopUse :StopUseEnum
Note :MultilingualString [0..1]
«PK»
id :CallIdType
JourneyRef :JourneyIdType*
«AK»
VisitNumber :positiveInteger
«FK»
ScheduledStopPointRef :ScheduledStopPointRef
PointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
ServiceJourneyRef :ScheduledStopPointRef [0..1]
«contained»
Arrival :Arrival [0..1]
Departure :Departure [0..1]
Frequency :JourneyFrequency [0..1]
DestinationDisplayView :DestinationDisplayView [0..1]
vias :DestinationDisplayView [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
«FV»
ScheduledStopPointView :ScheduledStopPointView [0..1]
OnwardTimingLinkView :OnwardTimingLinkView [0..1]
OnwardServiceLinkView :OnwardServiceLinkView [0..1]
«view»
DataManagedObject
LineModel::DestinationDisplay
Name :MultilingualString
ShortName :MultilingualString [0..1]
SideText :MultilingualString [0..1]
FrontText :MultilingualString [0..1]
DriverDisplayText :MultilingualString [0..1]
ShortCode :normalizedString [0..1]
«PK»
id :DestinationDisplayIdType
«AK»
PublicCode :normalizedString [0..1]
PrivateCode :PrivateCodeType [0..1]
«contained»
vias :Via [0..*]
variants :DeliveryDisplayVariant [0..*]
vias
0..*
equivalent to
vias at
1
at
show
0..1
0..*
visit
0..*
calls
work
0..*
0..1
/destination
0..1
at
1
TimingPoint
ServicePatternModel::ScheduledStopPoint
of
1
DatedCall
«PK»
id :CallIdType
«FK»
DriverRef :LogicalDriverIdType* [0..1]
Site
StopPlaceModel::StopPlace
to
1
assignment
0..*
0..*
assignment
for
*
DataManagedObject
StopAssignmentModel::StopAssignment
StopAssignmentModel::PassengerStopAssignment
PassingTime
PassingTimesModel::TimetabledPassingTime
ArrivalTime :time [0..1]
DepartureTime :time [0..1]
WaitingTime :duration [0..1]
Headway :HeadwayInterval [0..1]
EarliestDepartureTime :time [0..1]
LatestArrivalTime :time [0..1]
«PK»
id :PassingTimeId Type [0..1]
at
*

Figure 140 — Dated Call – Physical Model (UML)

### 7.2.15.2 DatedCall – Model Element

A CALL for a DATED VEHICLE JOURNEY.

Table 80 – *DatedCall* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *Call* | ::> | DATED CALL inherits from CALL. |
| «PK» | ***id*** | *CallIdType* | 1:1 | Identifier of DATED CALL. |
| «FK» | ***DriverRef*** | *LogicalDriverRef* | 0:1 | LOGICAL DRIVER undertaking CALL. |

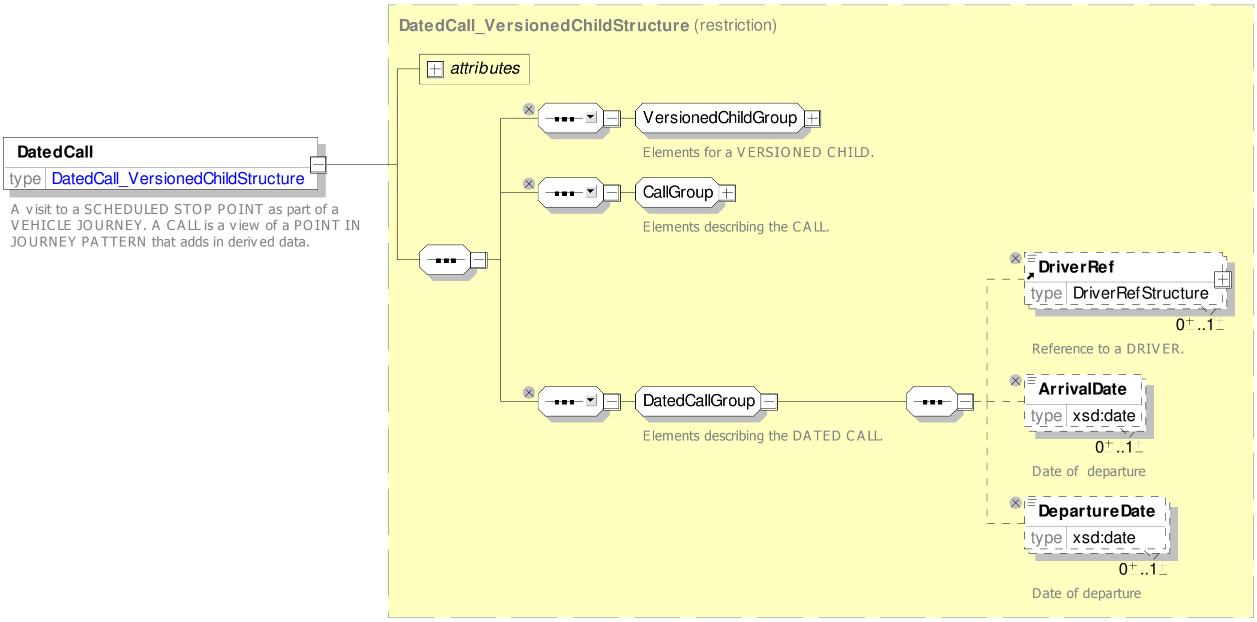

DatedCall_VersionedChildStructure (restriction)
attributes
DatedCall
type DatedCall_VersionedChildStructure
A visit to a SCHEDULED STOP POINT as part of a VEHICLE JOURNEY. A CALL is a view of a POINT IN JOURNEY PATTERN that adds in derived data.
VersionedChildGroup
Elements for a VERSIONED CHILD.
CallGroup
Elements describing the CALL.
DatedCallGroup
Elements describing the DATED CALL.
DriverRef
type DriverRefStructure
0..1
Reference to a DRIVER.
ArrivalDate
type xsd:date
0..1
Date of departure
DepartureDate
type xsd:date
0..1
Date of departure

Figure 141 — *DatedCall* — XSD

# 8 Vehicle Scheduling

## 8.1 Vehicle Scheduling – Model dependencies

The VEHICLE SERVICE Model describes the BLOCKs and VEHICLE SCHEDULEs for vehicle operation and is itself divided into a number of separate submodels covering different aspects of the scheduling. For ease of understanding, the submodels are presented one at a time, each describing only a small set of related concepts.

The submodels depend on a number of general NeTEx framework models and Part 1 and Part 2 described elsewhere.

The following figure shows the dependencies between the Vehicle Service physical submodels. The terminal packages contain the VEHICLE SCHEDULE FRAME and the DRIVER SCHEDULE FRAME. These two VERSION FRAMEs organise the other elements into a coherent set of elements suitable for exchange as a serialised file. The payload elements are contained in the following packages:

— VEHICLE SCHEDULE FRAME

— VEHICLE SERVICE: models the BLOCKs and VEHICLE SERVICEs for operation.

— TRAIN SERVICE: models the TRAIN BLOCKs for train operation.

— DRIVER SCHEDULE FRAME

— DUTY: models the driver DUTies for operations.

— DUTY STRETCH: models the driver work assignments.

NOTE DUTY and DRIVER related issues are out of current NeTEx scope (it may be integrated in later additional Parts). Therefore, information provided on DUTY and DRIVER is only informative (more information Annex A).

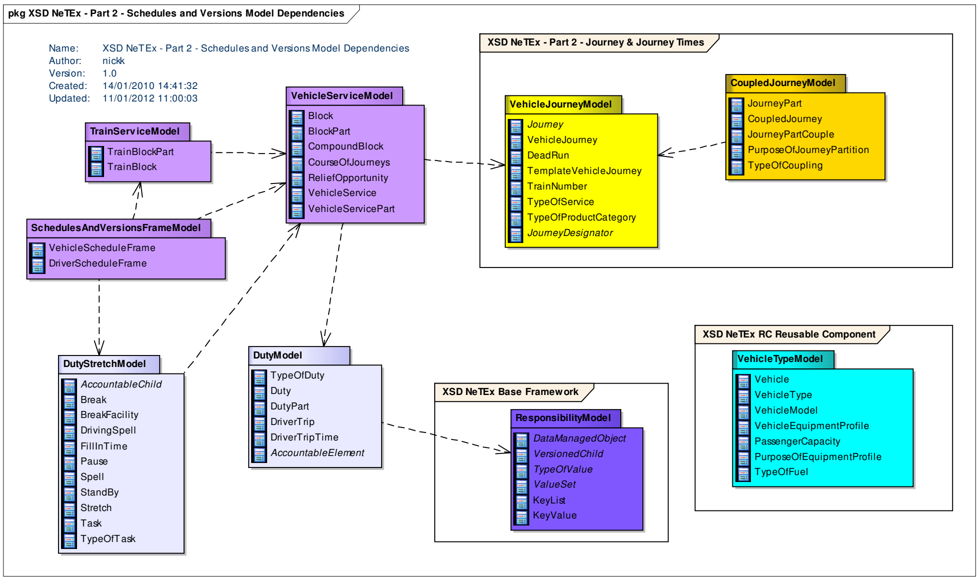


**Figure 142 — Vehicle Schedules – Model Dependencies**

## 8.2 Vehicle Scheduling

### 8.2.1 Vehicle Schedule Frame

#### 8.2.1.1 VEHICLE SCHEDULE FRAME – Conceptual MODEL

The elements of the VEHICLE SCHEDULE model can be grouped within a VEHICLE SCHEDULE FRAME which holds a coherent set of vehicle related elements for data exchange; including BLOCKs, COURSEs of JOURNEY, and VEHICLE SERVICE. See VERSION FRAME in the NeTEx Framework section for general concepts relating to version frames.

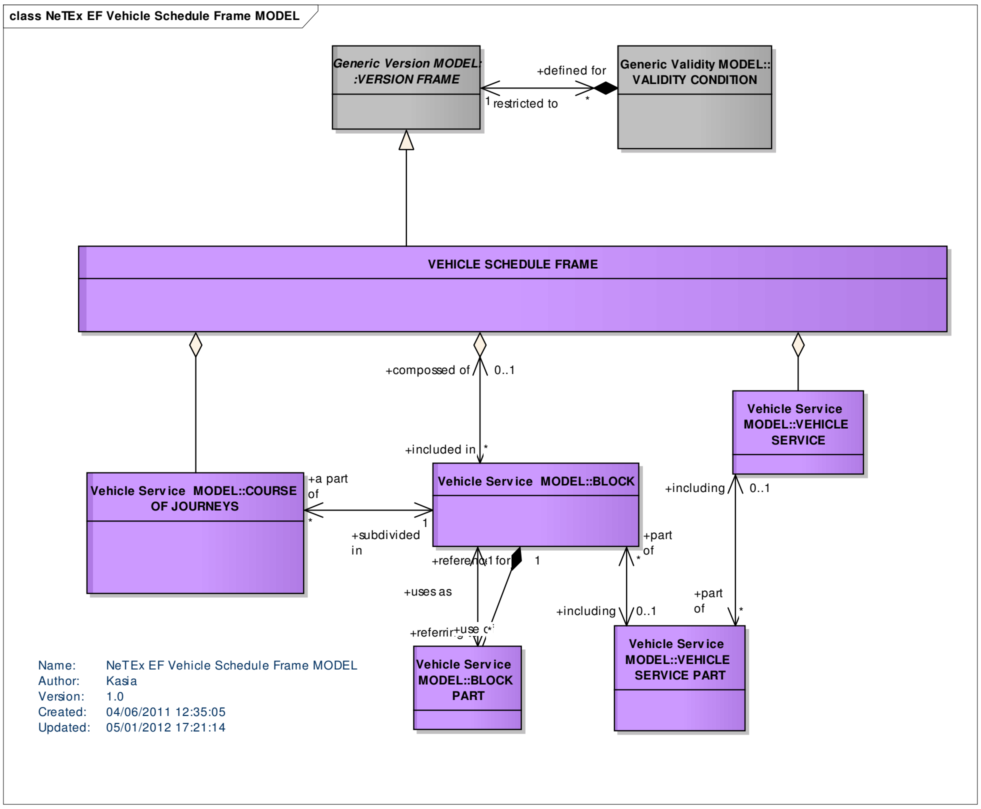
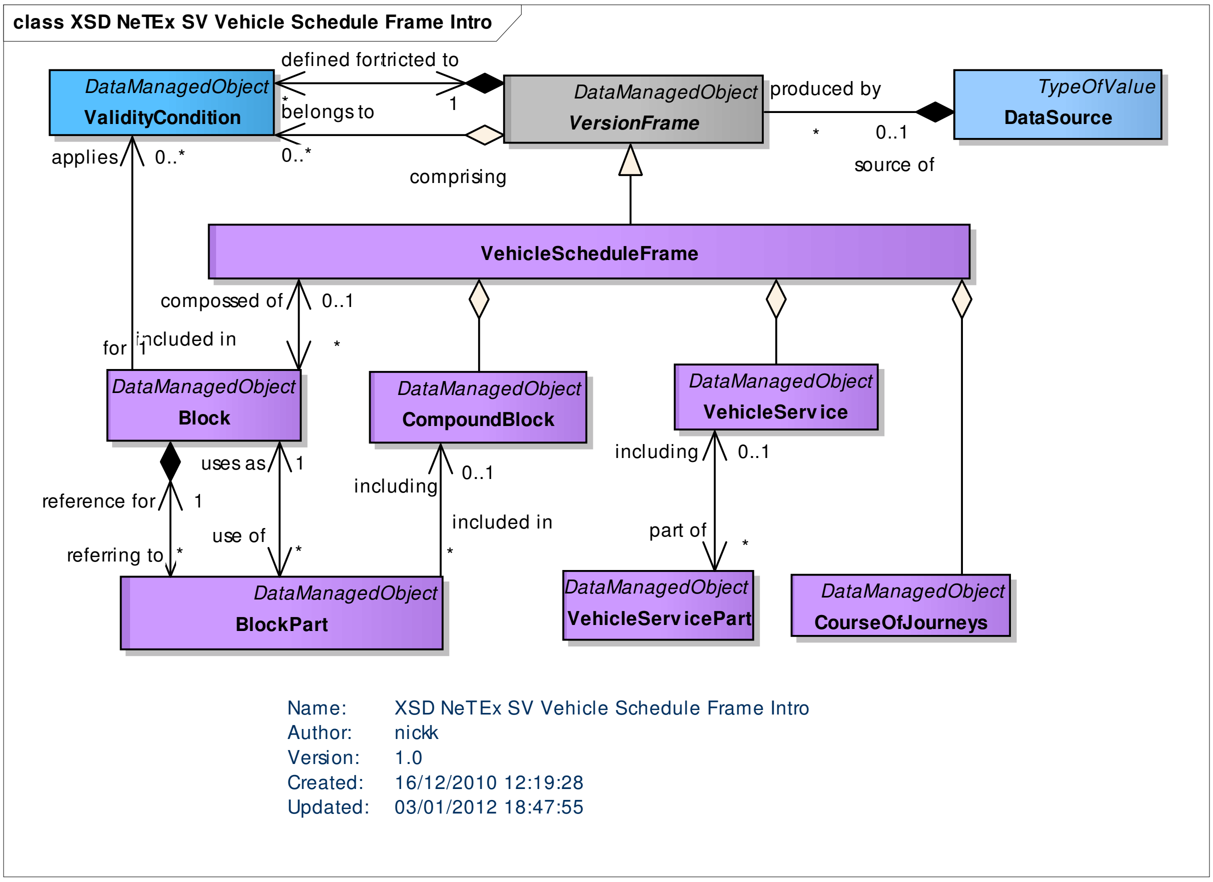

class NeTEx EF Vehicle Schedule Frame MODEL
Generic Version MODEL::VERSION FRAME
Generic Validity MODEL::VALIDITY CONDITION
+defined for
1
restricted to
*
VEHICLE SCHEDULE FRAME
+compossed of
0..1
+included in
*
Vehicle Service MODEL::VEHICLE SERVICE
Vehicle Service MODEL::COURSE OF JOURNEYS
+a part of
*
1
+subdivided in
Vehicle Service MODEL::BLOCK
+including
0..1
+part of
*
+referen
1 for
1
+uses as
+including
0..1
+part of
*
+referring
+use of
*
Vehicle Service MODEL::BLOCK PART
Vehicle Service MODEL::VEHICLE SERVICE PART
Name: NeTEx EF Vehicle Schedule Frame MODEL
Author: Kasia
Version: 1.0
Created: 04/06/2011 12:35:05
Updated: 05/01/2012 17:21:14

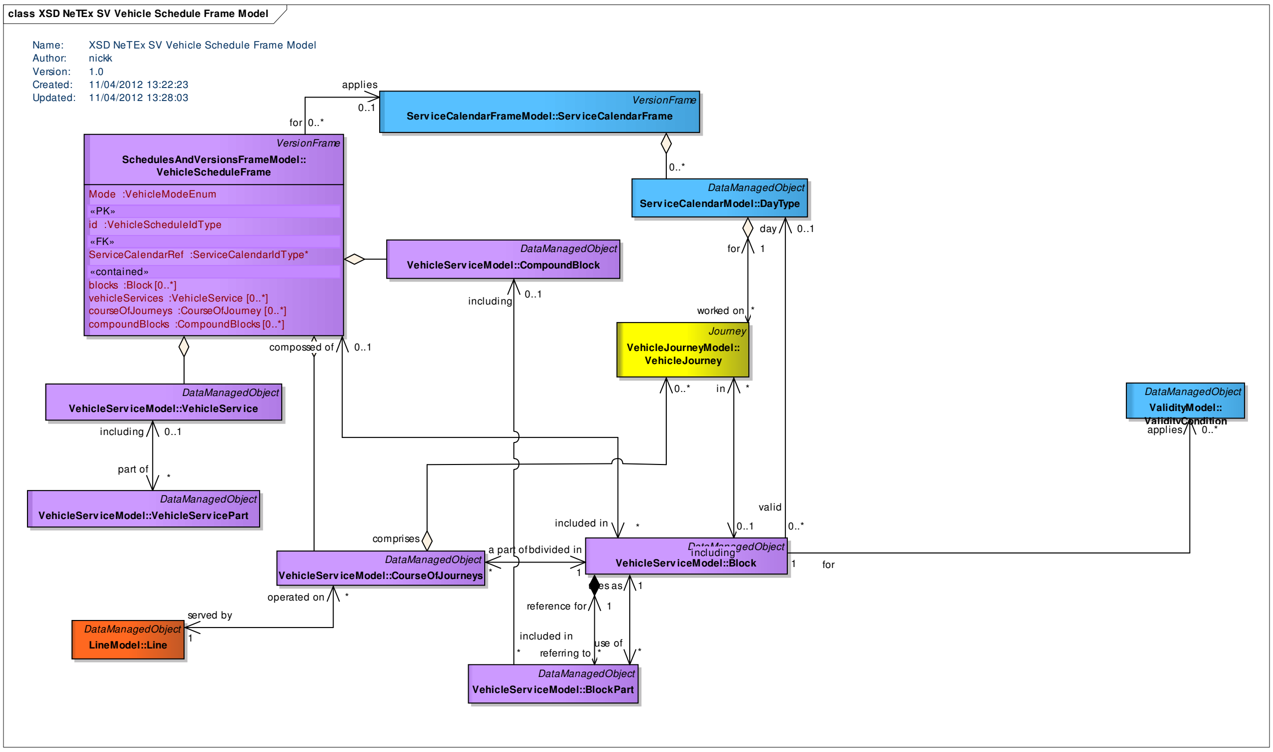
**Figure 143 — Vehicle Schedule Frame – Conceptual MODEL (UML)**

### 8.2.1.2 Vehicle Schedule Frame – Physical model

The following diagram shows the physical model for a VEHICLE SCHEDULE FRAME.



## Figure 144 — Vehicle Schedule Frame Contents– Physical Model (UML)


class XSD NeTEx SV Vehicle Schedule Frame Model
Name: XSD NeTEx SV Vehicle Schedule Frame Model
Author: nickk
Version: 1.0
Created: 11/04/2012 13:22:23
Updated: 11/04/2012 13:28:03
applies
0..1
for 0..*
VersionFrame
ServiceCalendarFrameModel::ServiceCalendarFrame
0..*
DataManagedObject
ServiceCalendarModel::DayType
VersionFrame
SchedulesAndVersionsFrameModel::
VehicleScheduleFrame
Mode :VehicleModeEnum
«PK»
id :VehicleScheduleIdType
«FK»
ServiceCalendarRef :ServiceCalendarIdType*
«contained»
blocks :Block [0..*]
vehicleServices :VehicleService [0..*]
courseOfJourneys :CourseOfJourney [0..*]
compoundBlocks :CompoundBlocks [0..*]
DataManagedObject
VehicleServiceModel::CompoundBlock
including
0..1
day
0..1
for
1
worked on
*
Journey
VehicleJourneyModel::
VehicleJourney
compossed of
0..1
DataManagedObject
VehicleServiceModel::VehicleService
including
0..1
part of
*
DataManagedObject
VehicleServiceModel::VehicleServicePart
0..*
in
*
DataManagedObject
ValidityModel::
ValidityCondition
applies
0..*
valid
included in
*
0..1
0..*
comprises
a part of
bdivided in
DataManagedObject
VehicleServiceModel::CourseOfJourneys
*
1
DataManagedObject
including
VehicleServiceModel::Block
1
for
operated on
*
served by
1
DataManagedObject
LineModel::Line
es as
1
reference for
1
included in
use of
referring to
*
*
*
DataManagedObject
VehicleServiceModel::BlockPart

## Figure 145 — Vehicle Schedule Frame – Physical Model (UML)

### 8.2.1.3 Vehicle Schedule Frame — XSD and attributes

#### 8.2.1.3.1 VehicleScheduleFrame – Model Element

A VEHICLE SCHEDULE FRAME organizes the set of all BLOCKs defined for a specific DAY TYPE to which the same VALIDITY CONDITIONs have been assigned (usually defined for a specific GROUP OF LINEs).

**Table 81 – *VehicleScheduleFrame* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *VersionFrame* | ::> | VEHICLE SCHEDULE FRAME inherits from VERSION FRAME. |
| «PK» | ***id*** | *VehicleScheduleIdType* | 1:1 | Identifier of VEHICLE SCHEDULE FRAME. |
| «FK» | ***Service-CalendarRef*** | *ServiceCalendarRef* | 1:1 | Reference to SERVICE CALENDAR for VEHICLE SCHEDULE FRAME. |
|  | ***Mode*** | *VehicleModeEnum* | 0:1 | Mode of VEHICLE JOURNEYs in VEHICLE SCHEDULE FRAME. |
| «cntd» | ***blocks*** | *Block \| TrainBlock \| CompoundBlock* | 0:* | BLOCKs in VEHICLE SCHEDULE FRAME. |
| «cntd» | ***courseOf-Journeys*** | *CourseOfJourney* | 0:* | COURSEs OF JOURNEYs in VEHICLE SCHEDULE FRAME. |
| «cntd» | ***vehicleServices*** | *VehicleService* | 0:* | VEHICLE SERVICEs in VEHICLE SCHEDULE FRAME. |
| «cntd» | ***relief-Opportunities*** | *ReliefOpportunity* | 0:* | RELIEF OPPORTUNITies in VEHICLE SCHEDULE FRAME. |

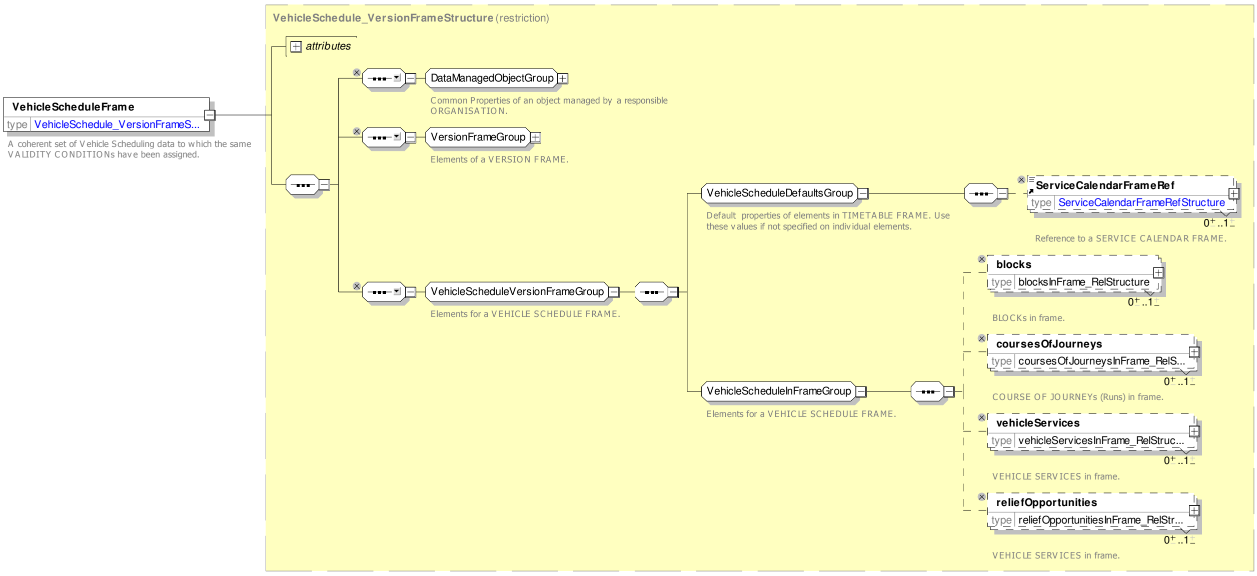

VehicleSchedule_VersionFrameStructure (restriction)
attributes
VehicleScheduleFrame
type VehicleSchedule_VersionFrameS...
A coherent set of Vehicle Scheduling data to which the same VALIDITY CONDITIONs have been assigned.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
VersionFrameGroup
Elements of a VERSION FRAME.
VehicleScheduleVersionFrameGroup
Elements for a VEHICLE SCHEDULE FRAME.
VehicleScheduleDefaultsGroup
Default properties of elements in TIMETABLE FRAME. Use these values if not specified on individual elements.
ServiceCalendarFrameRef
type ServiceCalendarFrameRefStructure
0..1
Reference to a SERVICE CALENDAR FRAME.
VehicleScheduleInFrameGroup
Elements for a VEHICLE SCHEDULE FRAME.
blocks
type blocksInFrame_RelStructure
0..1
BLOCKs in frame.
coursesOfJourneys
type coursesOfJourneysInFrame_RelS...
0..1
COURSE OF JOURNEYs (Runs) in frame.
vehicleServices
type vehicleServicesInFrame_RelStruc...
0..1
VEHICLE SERVICES in frame.
reliefOpportunities
type reliefOpportunitiesInFrame_RelStr...
0..1
VEHICLE SERVICES in frame.

**Figure 146 — *VehicleScheduleFrame* — XSD**

### 8.2.2 Vehicle Service

#### 8.2.2.1 VEHICLE SERVICE – Conceptual MODEL

The whole work of a logical vehicle for a DAY TYPE is described by a VEHICLE SERVICE, composed of one or several VEHICLE SERVICE PARTs.

Three levels of activities (BLOCK, VEHICLE SERVICE PART, and VEHICLE SERVICE) are possible, but will not always be described. The concept of VEHICLE SERVICE is not strictly necessary, once the number of required vehicles has been optimised and, in many situations, the distinction between BLOCK and VEHICLE SERVICE PART is not useful. Therefore quite often only a subpart of this model will be used (e.g. BLOCK).

##### 8.2.2.1.1 Blocks

In vehicle planning for day types, the activities of a logical vehicle are grouped into BLOCKs, which describe continuous periods during which the vehicle constantly requires, in principle, a driver. A BLOCK starts from the time it leaves a PARKING POINT and ends at the time it returns to park at this or another PARKING POINT.

A BLOCK includes in particular all VEHICLE JOURNEYs (SERVICE JOURNEYs and DEAD RUNs) planned for this period. It may include as well SPECIAL SERVICEs. Between two journeys or services, a PAUSE may be planned, during which the driver is still responsible for the vehicle.

A necessary constraint is that all VEHICLE JOURNEYs included in a BLOCK shall refer to the same DAY TYPE. The DAY TYPE of a BLOCK may hence be derived from the DAY TYPE of the VEHICLE JOURNEYs it includes.

Some operators distinguish in addition sequences of BLOCKs starting and ending in a garage (i.e. the end points shall be GARAGE POINTs, but not necessarily a real garage). Such a VEHICLE SERVICE PART may be composed, for instance, of two BLOCKs, separated by a parking period spent, without any driver, at a PARKING POINT which is not a GARAGE POINT (e.g. parking periods at off-peak hours for buses, at a parking point in the city). This concept is useful when no maintenance facility is available at this PARKING POINT or if only a few parking places are available: the physical VEHICLE assigned to this VEHICLE SERVICE PART may not be maintained or exchanged during the parking period without a relatively complex action.

A BLOCK PART is a part of a BLOCK corresponding to the different JOURNEY PARTs of the VEHICLE JOURNEYs in a BLOCK.

One BLOCK for a VEHICLE may last for many hours or even the whole day. Drivers cannot work during such long time and arrangements shall be made to relieve drivers during the BLOCKs. RELIEF POINTs are predefined for this purpose. Every point of time within a BLOCK when the vehicle passes a RELIEF POINT is known as a RELIEF OPPORTUNITY. RELIEF OPPORTUNITies may or may not be taken for actually relieving the driver. Each RELIEF OPPORTUNITY will occur at a RELIEF POINT that is viewed as a TIMING POINT IN JOURNEY PATTERN.

##### 8.2.2.1.2 Course of Journeys

Another classical method of designing the work of logical vehicles uses the concept of COURSE OF JOURNEYS. This method starts with the definition of VEHICLE JOURNEYs for all the LINEs of an ORGANISATIONAL UNIT (e.g. a depot, or even the whole network). All these VEHICLE JOURNEYs are then chained into suitable BLOCKs of vehicle operation. As a result of this method, one BLOCK is likely to be operated on several (or even many) different LINEs. A COURSE OF JOURNEYS is a part of a BLOCK continuously operated on one single LINE, without any interruption. Thus, a BLOCK will include several COURSEs OF JOURNEYS if it consists of VEHICLE JOURNEYs serving more than one LINE.

In some cases BLOCKs are subdivided into COURSEs OF JOURNEYS using additional criteria, e.g. change of the driver or other (arbitrarily chosen) events or situations. An (optional) attribute giving the start time of a COURSE OF JOURNEYS in the BLOCK will then determine which journeys are included in that course.

Any COURSE OF JOURNEYs shall include only VEHICLE JOURNEYs belonging to the same BLOCK, hence defined for the same DAY TYPE. This entity is hence valid for one DAY TYPE only.

#### 8.2.2.1.3 Vehicle Type

A VEHICLE TYPE characterises the common properties of a defined class of public transport vehicles. The most relevant properties for scheduling are the seating and standing capacities.

Depending on the time of day at which a SERVICE JOURNEY takes place, and on the ROUTE and JOURNEY PATTERN it covers, a default VEHICLE TYPE may be proposed for this journey. The most suitable VEHICLE TYPE will be chosen to meet the demand at various periods of the day, whenever possible. For instance, articulated or double-deck buses will be preferred to standard buses on the lines where the demand is the most important, especially at peak hours.

However, practical limitations, in particular of infrastructure, will also orient the choice of a VEHICLE TYPE, depending on its size: minibuses will be chosen for routes passing through narrow streets, while double-deck buses could not be used where the available height in tunnels is not sufficient. In rail systems, the VEHICLE TYPE should for instance be compatible with the length of stations and the height of platforms.

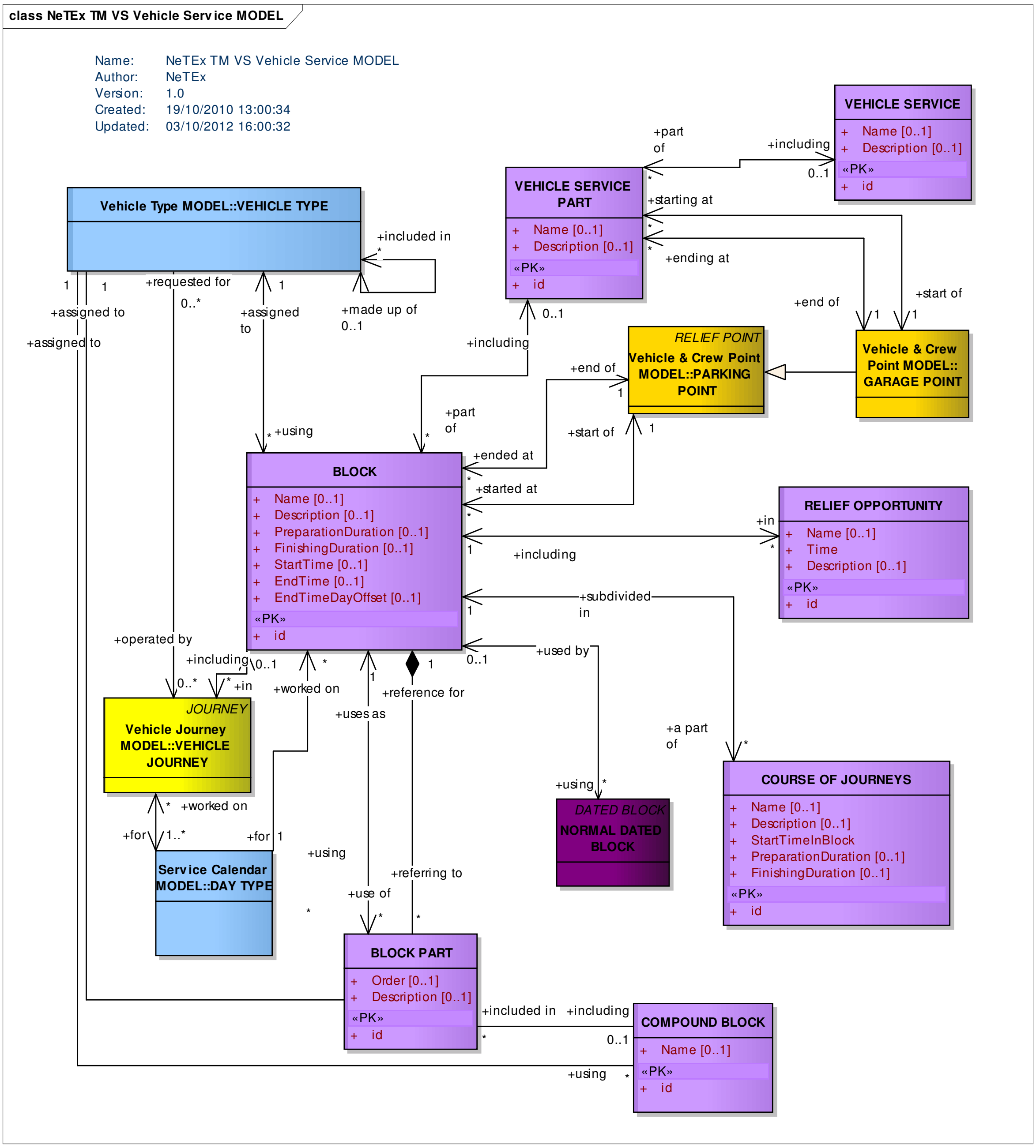
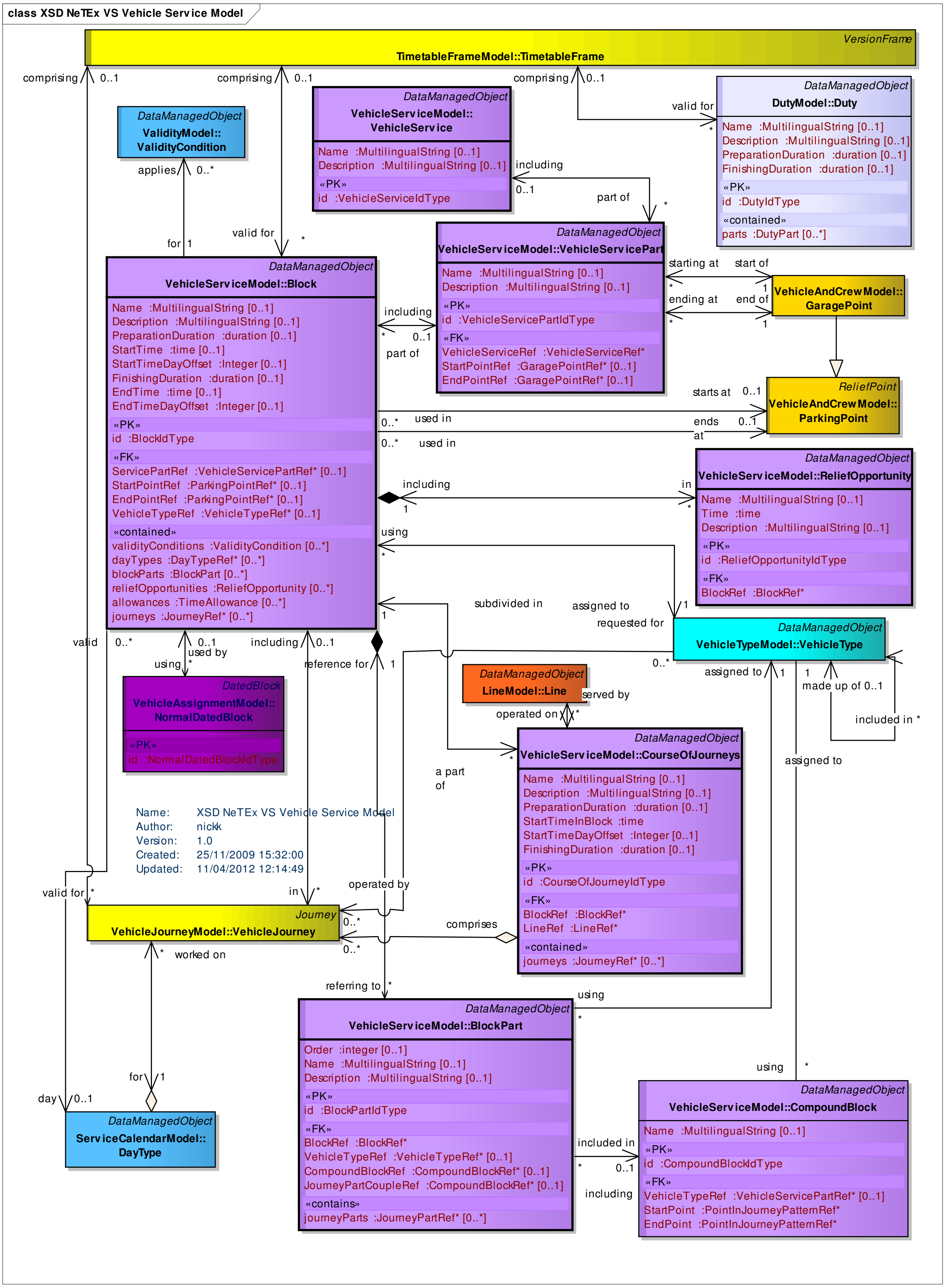


Figure 147 — Vehicle Service – Conceptual MODEL (UML)

### 8.2.2.2 Vehicle Service – Physical Model

The following figure shows the physical model.


class XSD NeTEx VS Vehicle Service Model
VersionFrame
TimetableFrameModel::TimetableFrame
comprising
0..1
comprising
0..1
comprising
0..1
DataManagedObject
ValidityModel::
ValidityCondition
applies
0..*
for
1
DataManagedObject
VehicleServiceModel::
VehicleService
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
«PK»
id :VehicleServiceIdType
including
0..1
part of
*
valid for
*
DataManagedObject
DutyModel::Duty
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
PreparationDuration :duration [0..1]
FinishingDuration :duration [0..1]
«PK»
id :DutyIdType
«contained»
parts :DutyPart [0..*]
valid for
*
DataManagedObject
VehicleServiceModel::VehicleServicePart
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
«PK»
id :VehicleServicePartIdType
«FK»
VehicleServiceRef :VehicleServiceRef*
StartPointRef :GaragePointRef* [0..1]
EndPointRef :GaragePointRef* [0..1]
starting at
start of
*
1
ending at
end of
*
1
VehicleAndCrewModel::
GaragePoint
DataManagedObject
VehicleServiceModel::Block
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
PreparationDuration :duration [0..1]
StartTime :time [0..1]
StartTimeDayOffset :Integer [0..1]
FinishingDuration :duration [0..1]
EndTime :time [0..1]
EndTimeDayOffset :Integer [0..1]
«PK»
id :BlockIdType
«FK»
ServicePartRef :VehicleServicePartRef* [0..1]
StartPointRef :ParkingPointRef* [0..1]
EndPointRef :ParkingPointRef* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
«contained»
validityConditions :ValidityCondition [0..*]
dayTypes :DayTypeRef* [0..*]
blockParts :BlockPart [0..*]
reliefOpportunities :ReliefOpportunity [0..*]
allowances :TimeAllowance [0..*]
journeys :JourneyRef* [0..*]
including
*
0..1
part of
starts at
0..1
ReliefPoint
VehicleAndCrewModel::
ParkingPoint
0..*
used in
ends at
0..1
0..*
used in
DataManagedObject
VehicleServiceModel::ReliefOpportunity
Name :MultilingualString [0..1]
Time :time
Description :MultilingualString [0..1]
«PK»
id :ReliefOpportunityIdType
«FK»
BlockRef :BlockRef*
including
1
in
*
using
*
subdivided in
1
assigned to
requested for
1
DataManagedObject
VehicleTypeModel::VehicleType
0..*
assigned to
1
1
made up of 0..1
included in *
assigned to
valid
0..*
0..1
used by
using
*
including
0..1
reference for
1
DatedBlock
VehicleAssignmentModel::
NormalDatedBlock
«PK»
id :NormalDatedBlockIdType
DataManagedObject
LineModel::Line
served by
operated on
*
DataManagedObject
VehicleServiceModel::CourseOfJourneys
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
PreparationDuration :duration [0..1]
StartTimeInBlock :time
StartTimeDayOffset :Integer [0..1]
FinishingDuration :duration [0..1]
«PK»
id :CourseOfJourneyIdType
«FK»
BlockRef :BlockRef*
LineRef :LineRef*
«contained»
journeys :JourneyRef* [0..*]
a part of
*
Name: XSD NeTEx VS Vehicle Service Model
Author: nickk
Version: 1.0
Created: 25/11/2009 15:32:00
Updated: 11/04/2012 12:14:49
valid for
*
in
*
operated by
Journey
VehicleJourneyModel::VehicleJourney
0..*
comprises
0..*
*
worked on
referring to
*
using
*
DataManagedObject
VehicleServiceModel::BlockPart
Order :integer [0..1]
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
«PK»
id :BlockPartIdType
«FK»
BlockRef :BlockRef*
VehicleTypeRef :VehicleTypeRef* [0..1]
CompoundBlockRef :CompoundBlockRef* [0..1]
JourneyPartCoupleRef :CompoundBlockRef* [0..1]
«contains»
journeyParts :JourneyPartRef* [0..*]
using
*
DataManagedObject
VehicleServiceModel::CompoundBlock
Name :MultilingualString [0..1]
«PK»
id :CompoundBlockIdType
«FK»
VehicleTypeRef :VehicleServicePartRef* [0..1]
StartPoint :PointInJourneyPatternRef*
EndPoint :PointInJourneyPatternRef*
included in
*
0..1
including
for
1
day
0..1
DataManagedObject
ServiceCalendarModel::
DayType

## Figure 148 — Vehicle Service – Physical Model (UML)

#### 8.2.2.3 Vehicle Service – Attributes and XSD

##### 8.2.2.3.1 Block – Model Element

The work of a vehicle from the time it leaves a PARKING POINT after parking until its next return to park at a PARKING POINT. Any subsequent departure from a PARKING POINT after parking marks the start of a new BLOCK.

**Table 82 – *Block* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | BLOCK inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *BlockIdType* | 1:1 | Identifier of BLOCK. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of BLOCK. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of BLOCK. |
| | ***PrivateCode*** | *PrivateCode* | 0:1 | Private code of BLOCK. |
| | ***Preparation-Duration*** | *xsd:duration* | 0:1 | How long needed to prepare to run BLOCK. |
| | ***StartTime*** | *xsd:time* | 0:1 | Start time of BLOCK. In principle this can be derived from the Start time of the first journey and the preparation duration but may be stated explicitly as well. |
| | ***StartTime-DayOffset*** | *xsd:integer* | 0:1 | Day offset of Start time from current OPERATING DAY |
| | ***FinishingDuration*** | *xsd:duration* | 0:1 | How long needed to prepare to complete BLOCK. |
| | ***EndTime*** | *xsd:time* | 0:1 | End time of BLOCK. In principle this can be derived from the Start time of the last journey and the finishing duration but may be stated explicitly as well. |
| | ***EndTime-DayOffset*** | *xsd:integer* | 0:1 | Day offset of end time from current OPERATING DAY |
| «cntd» | ***validityConditions*** | *ValidityCondition* | 0:* | VALIDITY CONDITIONS controlling use of BLOCK. |
| «FK» | ***dayTypes*** | DayType | 0:1 | DAY TYPE associated with BLOCK. |
| «FK» | ***VehicleService-PartRef*** | *VehicleServicePartRef* | 0:1 | SERVICE PART associated with BLOCK. |
| «FK» | ***VehicleTypeRef*** | *VehicleServicePartRef* | 0:1 | VEHICLE TYPE associated with BLOCK. |
| «FK» | ***StartPointRef*** | *PointRef* | 0:1 | POINT from which BLOCK starts. |
| «FK» | ***EndPointRef*** | *PointRef* | 0:1 | POINT at which BLOCK ends. |
| «cntd» | ***journeys*** | *JourneyRef* | 0:* | Journeys in BLOCK. |
| «cntd» | ***coursesOf-Journeys*** | CourseOf*JourneyRef* | 0:* | Runs or Courses of Journey in BLOCK. |
| «cntd» | ***blockParts*** | *BlockPart* | 0:* | Parts of BLOCK. |
| «cntd» | ***reliefOpportunities*** | *ReliefOpportunity* | 0:* | RELIEF OPPORTUNITies in BLOCK. |

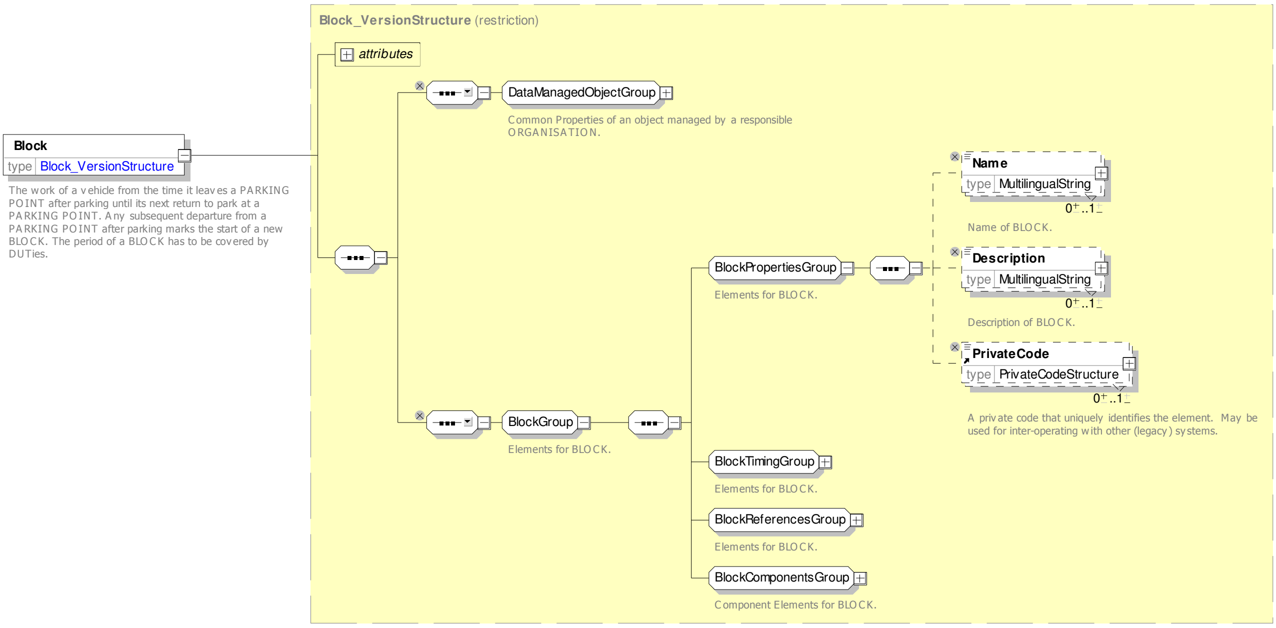

Block_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
Block
type Block_VersionStructure
The work of a vehicle from the time it leaves a PARKING POINT after parking until its next return to park at a PARKING POINT. Any subsequent departure from a PARKING POINT after parking marks the start of a new BLOCK. The period of a BLOCK has to be covered by DUTies.
Name
type MultilingualString
0..1
Name of BLOCK.
BlockPropertiesGroup
Elements for BLOCK.
Description
type MultilingualString
0..1
Description of BLOCK.
PrivateCode
type PrivateCodeStructure
0..1
A private code that uniquely identifies the element. May be used for inter-operating with other (legacy) systems.
BlockGroup
Elements for BLOCK.
BlockTimingGroup
Elements for BLOCK.
BlockReferencesGroup
Elements for BLOCK.
BlockComponentsGroup
Component Elements for BLOCK.

**Figure 149 — *Block* — XSD**

### 8.2.2.3.1.1 BlockTimingGroup – Model Element

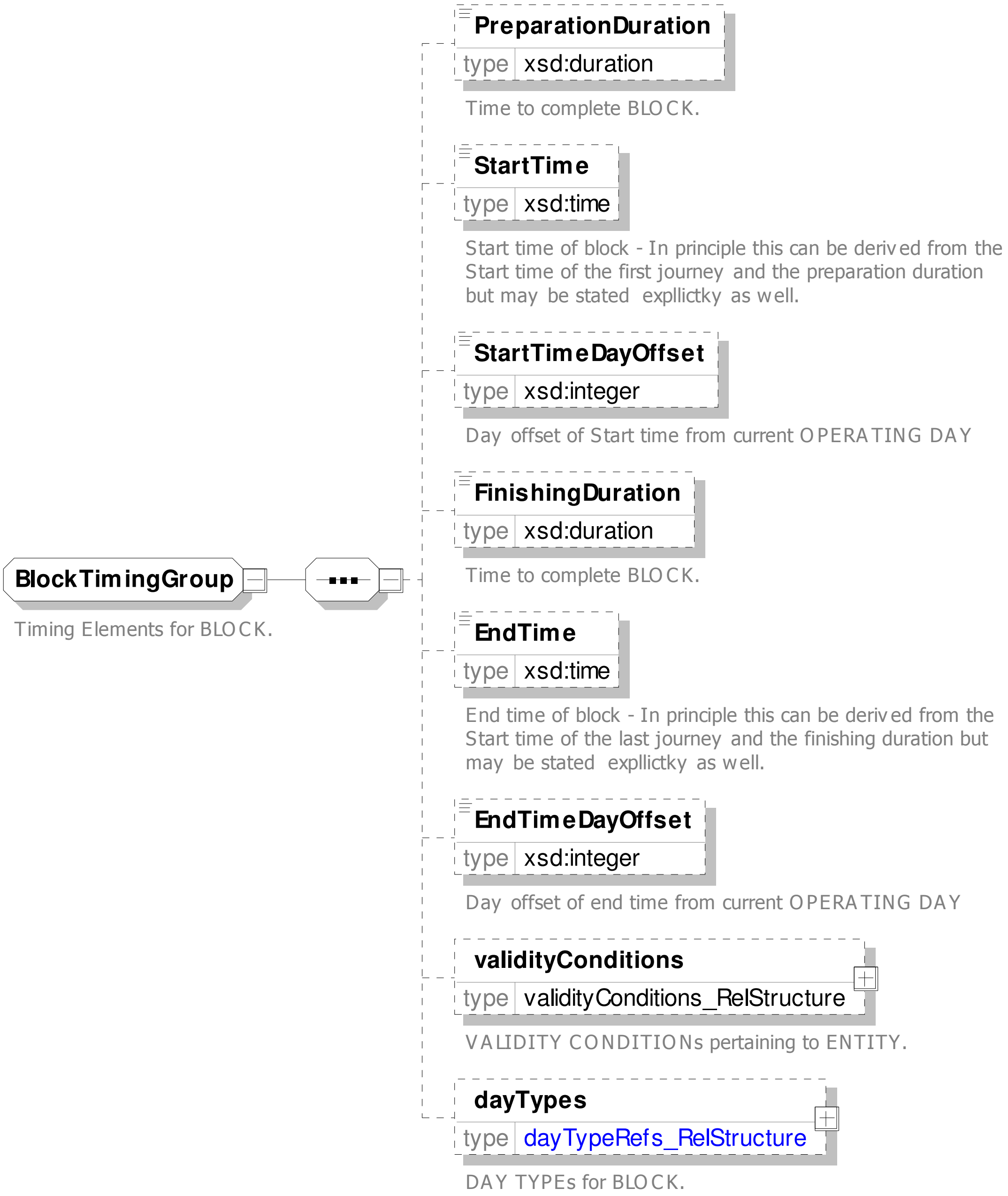

PreparationDuration
type xsd:duration
Time to complete BLOCK.
StartTime
type xsd:time
Start time of block - In principle this can be derived from the Start time of the first journey and the preparation duration but may be stated expllictky as well.
StartTimeDayOffset
type xsd:integer
Day offset of Start time from current OPERATING DAY
FinishingDuration
type xsd:duration
Time to complete BLOCK.
BlockTimingGroup
Timing Elements for BLOCK.
EndTime
type xsd:time
End time of block - In principle this can be derived from the Start time of the last journey and the finishing duration but may be stated expllictky as well.
EndTimeDayOffset
type xsd:integer
Day offset of end time from current OPERATING DAY
validityConditions
type validityConditions_RelStructure
VALIDITY CONDITIONs pertaining to ENTITY.
dayTypes
type dayTypeRefs_RelStructure
DAY TYPEs for BLOCK.

Figure 150 — *BlockTimingGroup* — XSD

#### 8.2.2.3.1.2 BlockReferencesGroup – Model Element

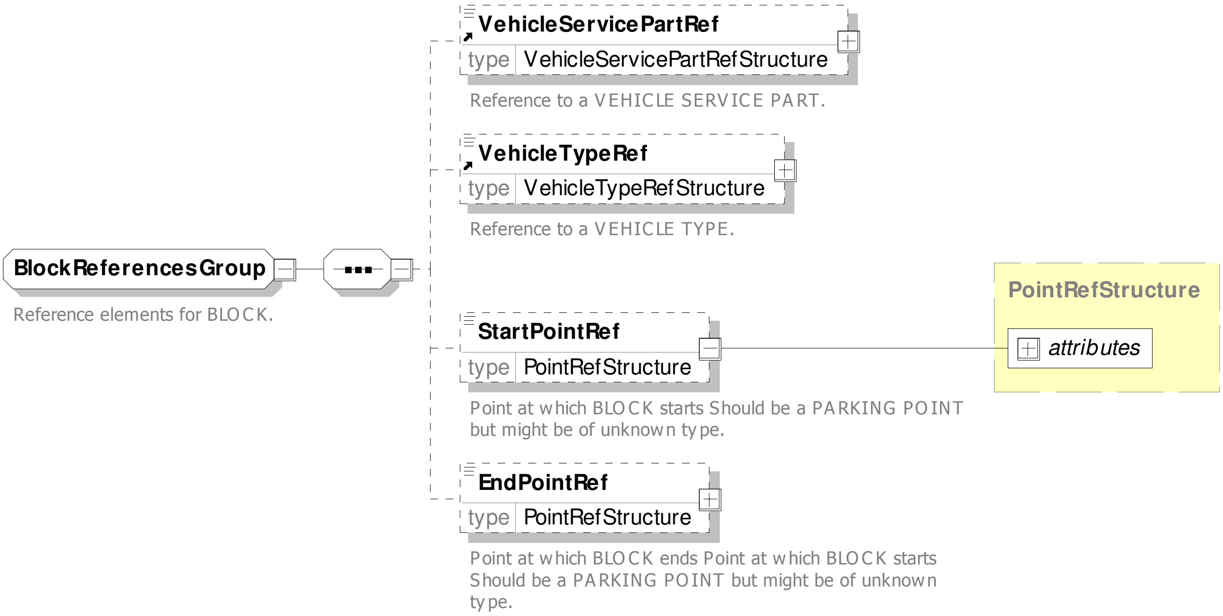

VehicleServicePartRef
type VehicleServicePartRefStructure
Reference to a VEHICLE SERVICE PART.
VehicleTypeRef
type VehicleTypeRefStructure
Reference to a VEHICLE TYPE.
BlockReferencesGroup
Reference elements for BLOCK.
PointRefStructure
attributes
StartPointRef
type PointRefStructure
Point at which BLOCK starts Should be a PARKING POINT but might be of unknown type.
EndPointRef
type PointRefStructure
Point at which BLOCK ends Point at which BLOCK starts Should be a PARKING POINT but might be of unknown type.

Figure 151 — *BlockReferencesGroup* — XSD

#### 8.2.2.3.1.3 BlockComponentsGroup – Model Element

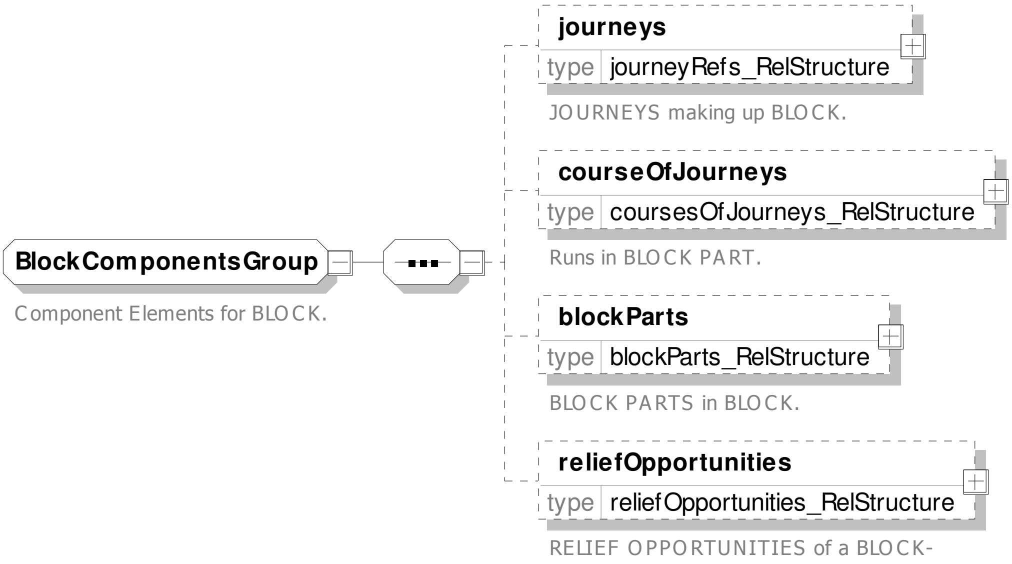

journeys
type journeyRefs_RelStructure
JOURNEYS making up BLOCK.
courseOfJourneys
type coursesOfJourneys_RelStructure
Runs in BLOCK PART.
BlockComponentsGroup
Component Elements for BLOCK.
blockParts
type blockParts_RelStructure
BLOCK PARTS in BLOCK.
reliefOpportunities
type reliefOpportunities_RelStructure
RELIEF OPPORTUNITIES of a BLOCK-

Figure 152 — *BlockComponentsGroup* — XSD

### 8.2.2.3.2 BlockPart – Model Element

A part of position of a vehicle BLOCK.

Table 83 – *BlockPart* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | BLOCK PART inherits from DATA MANAGED |

| | | | | |
|---|---|---|---|---|
| | | | | OBJECT |
| | ***order*** | *xsd:integer* | 0:1 | Order of BLOCK PART within BLOCK. |
| «PK» | ***id*** | *BlockPartIdType* | 1:1 | Identifier of BLOCK PART. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of BLOCK PART. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of BLOCK PART. |
| «FK» | ***BlockRef*** | *BlockRef* | 1:1 | Reference to parent BLOCK of which this is part**.** |
| «FK» | ***VehicleTypeRef*** | *VehicleTypeRef* | 0:1 | VEHICLE TYPE to be used for BLOCK PART. |
| «FK» | ***CompoundBlockRef*** | *CompoundBlockRef* | 0:1 | COMPOUND BLOCK corresponding to BLOCK PART - for TRAINs. |
| | ***Journey parts*** | Choice | | JOURNEY PARTs in BLOCK. |
| «FK» | a ***JourneyPartCoupleRef*** | *JourneyPartCoupleRef* | 0:1 | Reference to a JOURNEY PART COUPLE identifying a set of JOURNEY PARTs. |
| «cntd» | b ***journeyParts*** | *JourneyPartRef* | 0:* | References to JOURNEY PARTs in BLOCK. |

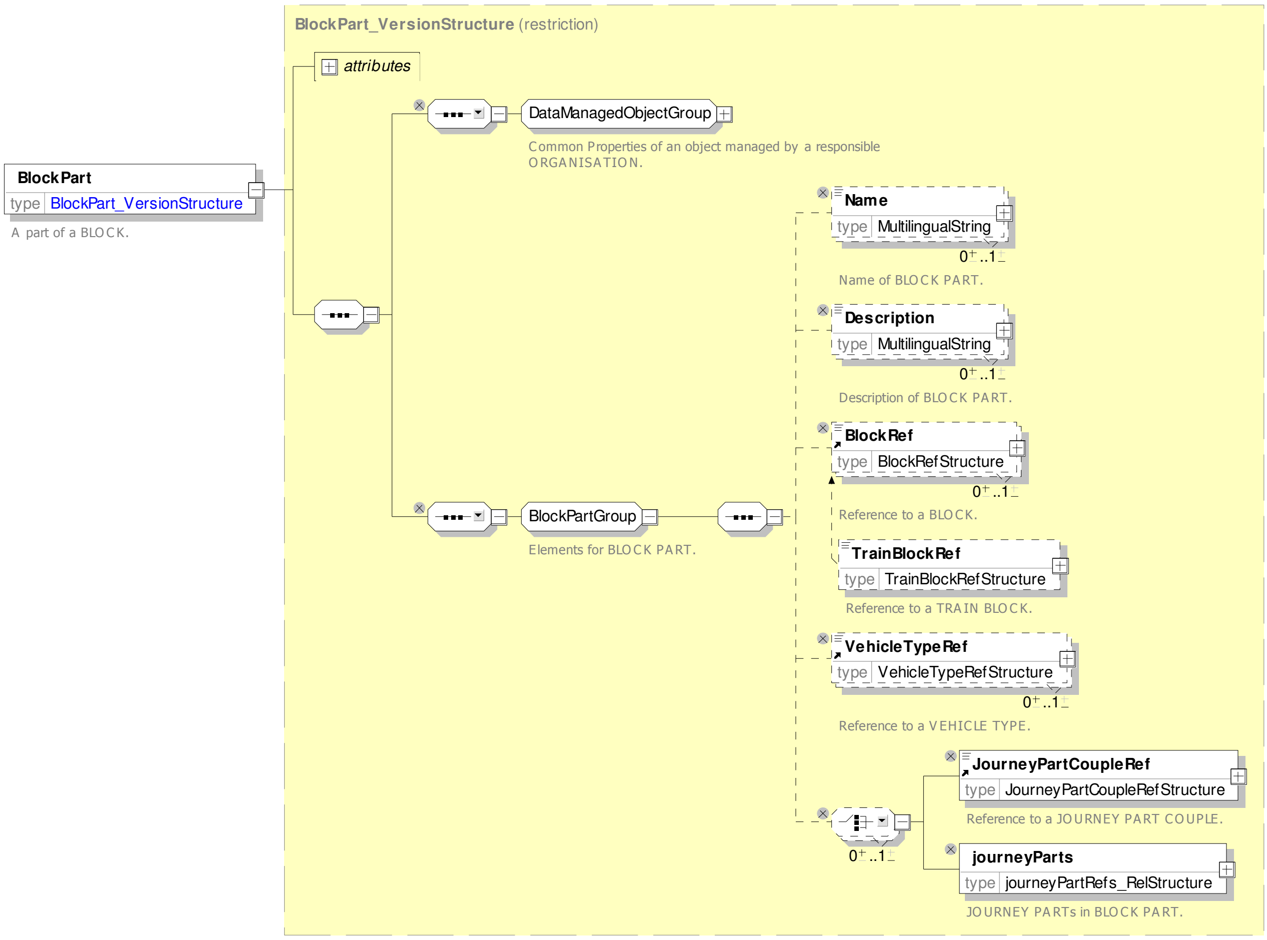

BlockPart_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
BlockPart
type BlockPart_VersionStructure
A part of a BLOCK.
Name
type MultilingualString
0..1
Name of BLOCK PART.
Description
type MultilingualString
0..1
Description of BLOCK PART.
BlockRef
type BlockRefStructure
0..1
Reference to a BLOCK.
TrainBlockRef
type TrainBlockRefStructure
Reference to a TRAIN BLOCK.
BlockPartGroup
Elements for BLOCK PART.
VehicleTypeRef
type VehicleTypeRefStructure
0..1
Reference to a VEHICLE TYPE.
JourneyPartCoupleRef
type JourneyPartCoupleRefStructure
Reference to a JOURNEY PART COUPLE.
0..1
journeyParts
type journeyPartRefs_RelStructure
JOURNEY PARTs in BLOCK PART.

**Figure 153 — *BlockPart* — XSD**

### 8.2.2.3.3 CompoundBlock – Model Element

The work of a vehicle during the time it is coupled to another vehicle.

**Table 84 – *CompoundBlock* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | COMPOUND BLOCK inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *CompoundBlockIdType* | 1:1 | Identifier of COMPOUND BLOCK. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of COMPOUND BLOCK. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of COMPOUND BLOCK. |
| «FK» | ***VehicleTypeRef*** | *VehicleServicePartRef* | 0:1 | VEHICLE TYPE associated with COMPOUND BLOCK. |
| «FK» | ***StartPointRef*** | *PointInJourneyPatternRef* | 1:1 | Start POINT IN JOURNEY PATTERN of COMPOUND BLOCK. |
| «FK» | ***EndPointRef*** | *PointInJourneyPatternRef* | 1:1 | End POINT IN JOURNEY PATTERN of COMPOUND BLOCK. |
| | ***parts*** | BlockPart | 0:* | BLOCK PARTs in COMPOUND BLOCK. |

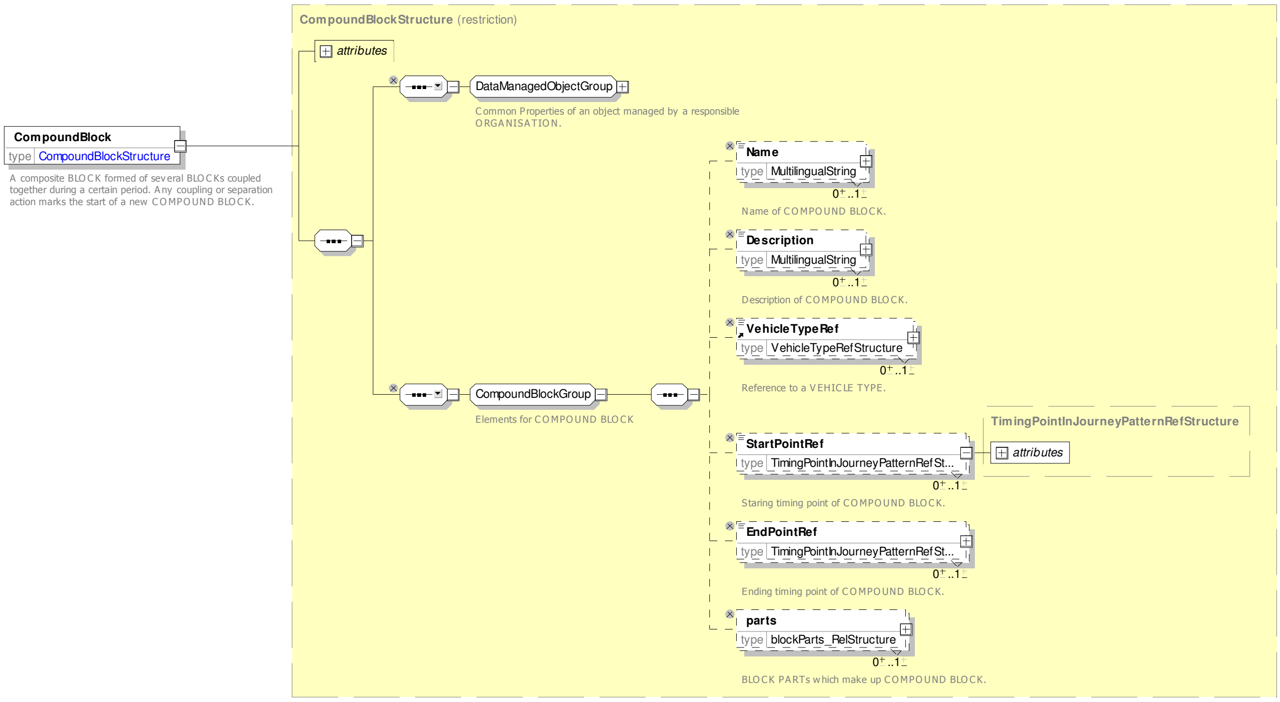

CompoundBlockStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
CompoundBlock
type CompoundBlockStructure
A composite BLOCK formed of several BLOCKs coupled together during a certain period. Any coupling or separation action marks the start of a new COMPOUND BLOCK.
Name
type MultilingualString
0..1
Name of COMPOUND BLOCK.
Description
type MultilingualString
0..1
Description of COMPOUND BLOCK.
VehicleTypeRef
type VehicleTypeRefStructure
0..1
Reference to a VEHICLE TYPE.
CompoundBlockGroup
Elements for COMPOUND BLOCK
TimingPointInJourneyPatternRefStructure
attributes
StartPointRef
type TimingPointInJourneyPatternRefSt...
0..1
Staring timing point of COMPOUND BLOCK.
EndPointRef
type TimingPointInJourneyPatternRefSt...
0..1
Ending timing point of COMPOUND BLOCK.
parts
type blockParts_RelStructure
0..1
BLOCK PARTs which make up COMPOUND BLOCK.

**Figure 154 — *CompoundBlock* — XSD**

#### 8.2.2.3.4 CourseOfJourneys – Model Element

A part of a BLOCK composed of consecutive VEHICLE JOURNEYs defined for the same DAY TYPE, all operated on the same LINE.

**Table 85 – *CourseOfJourney* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | COURSE OF JOURNEY inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *CourseOfJourneyIdType* | 1:1 | Identifier of COURSE OF JOURNEY. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of COURSE OF JOURNEYs. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of COURSE OF JOURNEYs. |
| | ***CourseOfJourneysNumber*** | xsd:integer | 0:1 | Number for COURSE OF JOURNEYs. |
| | ***Preparation-Duration*** | *xsd:duration* | 0:1 | How long needed to prepare for COURSE OF JOURNEY in BLOCK. |
| | ***StartTimeInBlock*** | *xsd:time* | 1:1 | Start time of COURSE OF JOURNEY in BLOCK. |
| | ***StartTime-DayOffset*** | *xsd:integer* | 0:1 | Day offset of start time from OPERATING DAY at start of Journey. |
| | ***FinishingDuration*** | *xsd:duration* | 0:1 | How long COURSE OF JOURNEY in BLOCK. |
| «FK» | ***BlockRef*** | *BlockRef* | 1:1 | BLOCK with which COURSE OF JOURNEY is ASSOCIATED. |
| «FK» | ***LineRef*** | *LineRef* | 1:1 | LINE ASSOCIATED with COURSE OF JOURNEY. |
| «contained» | ***journeys*** | *JourneyRef* | 0:* | VEHICLE JOURNEYs associated with COURSE OF JOURNEY. |

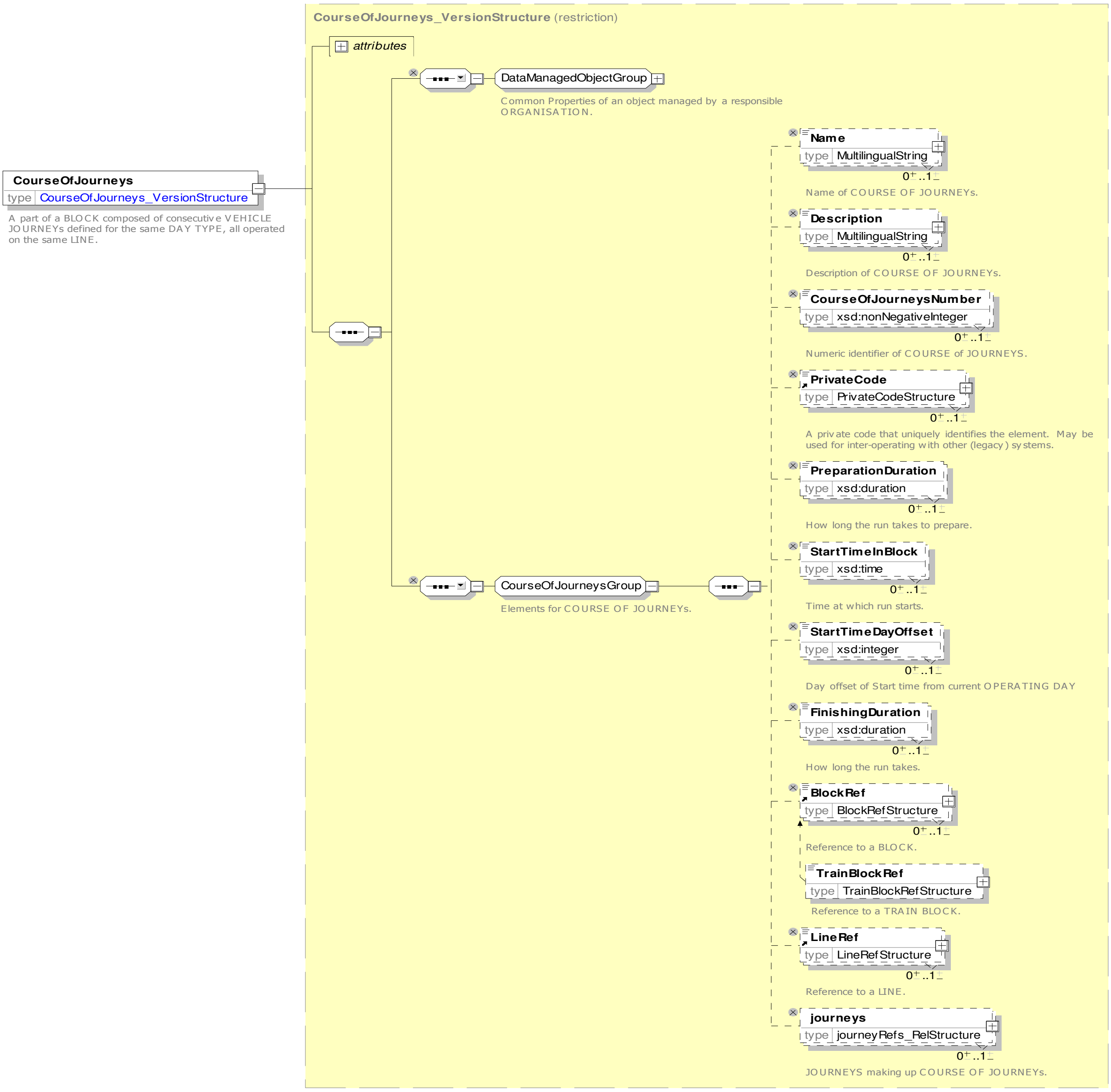

CourseOfJourneys_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
CourseOfJourneys
type CourseOfJourneys_VersionStructure
A part of a BLOCK composed of consecutive VEHICLE JOURNEYs defined for the same DAY TYPE, all operated on the same LINE.
Name
type MultilingualString
0..1
Name of COURSE OF JOURNEYs.
Description
type MultilingualString
0..1
Description of COURSE OF JOURNEYs.
CourseOfJourneysNumber
type xsd:nonNegativeInteger
0..1
Numeric identifier of COURSE of JOURNEYS.
PrivateCode
type PrivateCodeStructure
0..1
A private code that uniquely identifies the element. May be used for inter-operating with other (legacy) systems.
PreparationDuration
type xsd:duration
0..1
How long the run takes to prepare.
StartTimeInBlock
type xsd:time
0..1
Time at which run starts.
CourseOfJourneysGroup
Elements for COURSE OF JOURNEYs.
StartTimeDayOffset
type xsd:integer
0..1
Day offset of Start time from current OPERATING DAY
FinishingDuration
type xsd:duration
0..1
How long the run takes.
BlockRef
type BlockRefStructure
0..1
Reference to a BLOCK.
TrainBlockRef
type TrainBlockRefStructure
Reference to a TRAIN BLOCK.
LineRef
type LineRefStructure
0..1
Reference to a LINE.
journeys
type journeyRefs_RelStructure
0..1
JOURNEYS making up COURSE OF JOURNEYs.

Figure 155 — *CourseOfJourneys* — XSD

### 8.2.2.3.5 ReliefOpportunity – Model Element

A time in a BLOCK where a vehicle passes a RELIEF POINT. This opportunity may or may not be actually used for a relief.

Table 86 – *ReliefOpportunity* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | RELIEF OPPORTUNITY inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *ReliefOpportunityIdType* | 1:1 | Identifier of RELIEF OPPORTUNITY. |
|  | ***Name*** | *MultilingualString* | 0:1 | NAME of RELIEF OPPORTUNITY. |

| | | | | |
|---|---|---|---|---|
| | ***Description*** | *MultilingualString* | 0:1 | DESCRIPTION of RELIEF OPPORTUNITY. |
| | ***Time*** | *xsd:time* | 1:1 | TIME at which RELIEF OPPORTUNITY takes place. |
| «FK» | ***BlockRef*** | *BlockRef* | 1:1 | BLOCK associated with RELIEF OPPORTUNITY. |
| | ***Description*** | *MultilingualString* | 0:1 | DESCRIPTION of RELIEF OPPORTUNITY. |

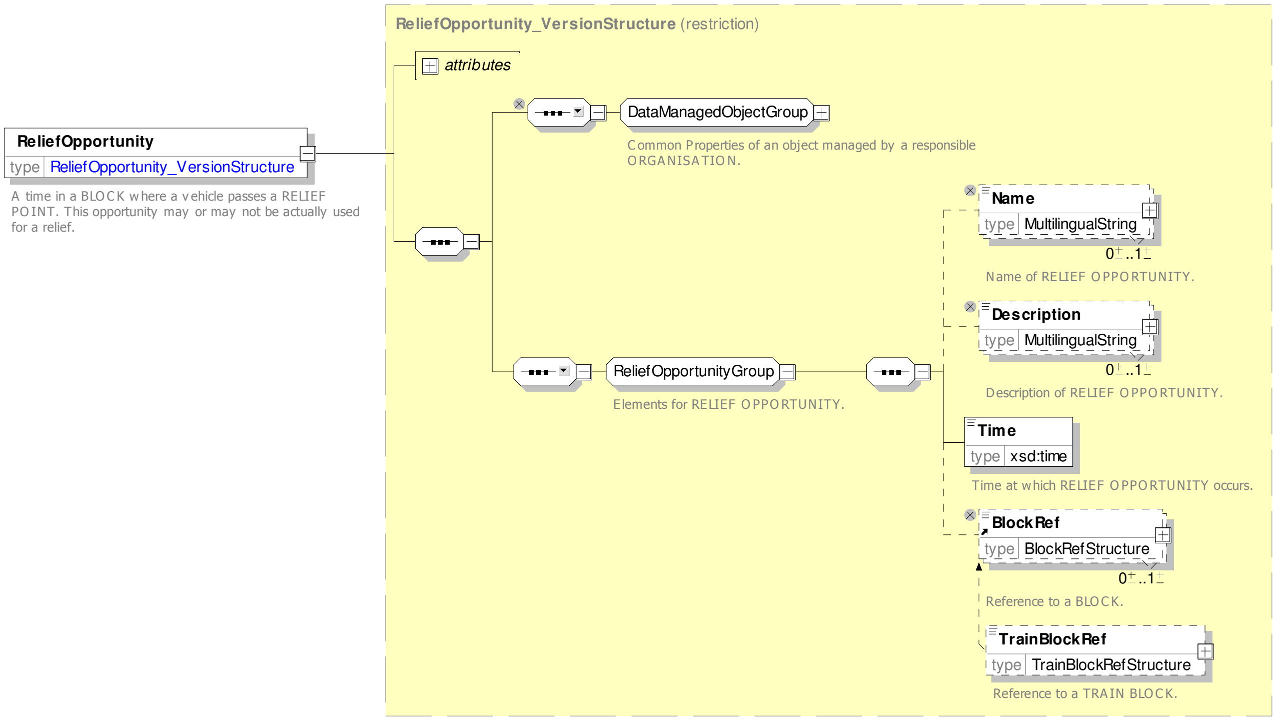

ReliefOpportunity_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
ReliefOpportunity
type ReliefOpportunity_VersionStructure
A time in a BLOCK where a vehicle passes a RELIEF POINT. This opportunity may or may not be actually used for a relief.
Name
type MultilingualString
0..1
Name of RELIEF OPPORTUNITY.
Description
type MultilingualString
0..1
Description of RELIEF OPPORTUNITY.
ReliefOpportunityGroup
Elements for RELIEF OPPORTUNITY.
Time
type xsd:time
Time at which RELIEF OPPORTUNITY occurs.
BlockRef
type BlockRefStructure
0..1
Reference to a BLOCK.
TrainBlockRef
type TrainBlockRefStructure
Reference to a TRAIN BLOCK.

Figure 156 — *ReliefOpportunity* — XSD

### 8.2.2.3.6 VehicleService – Model Element

A work plan for a vehicle for a whole day, planned for a specific DAY TYPE. A VEHICLE SERVICE includes one or several VEHICLE SERVICE PARTs.

Table 87 – *VehicleService* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | VEHICLE SERVICE inherits from DATA MANAGED OBJECT. |
| | ***id*** | *VehicleServiceIdType* | 1:1 | Identifier of VEHICLE SERVICE. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of VEHICLE SERVICE. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of VEHICLE SERVICE. |
| «cntd» | ***vehicleService-Parts*** | *VehicleServicePart* | 0:* | Parts of VEHICLE SERVICE. |

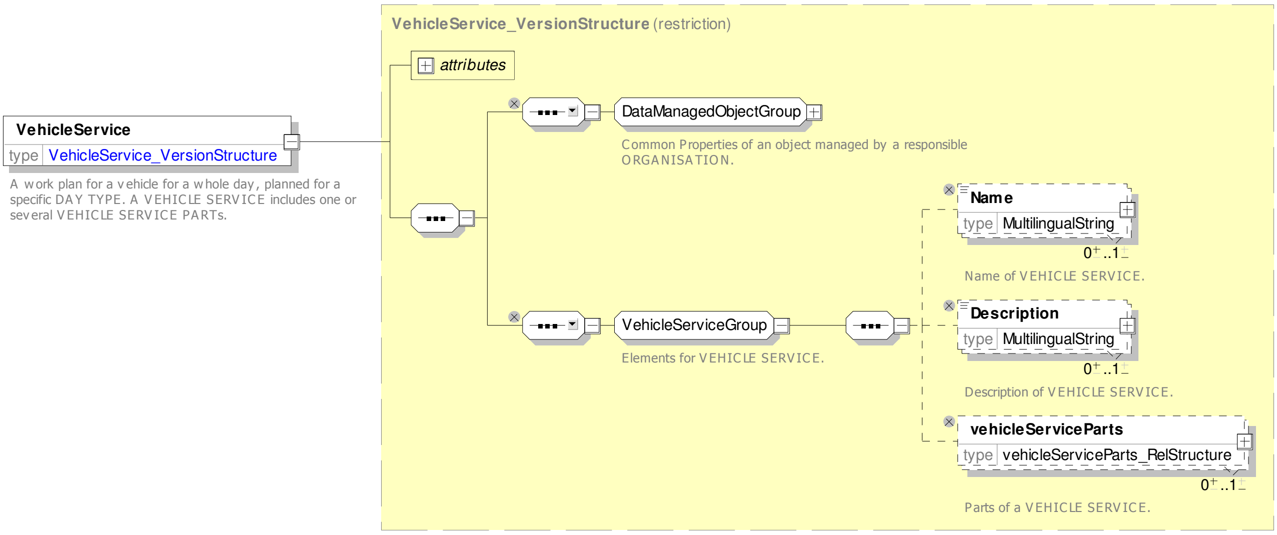

VehicleService_VersionStructure (restriction)
attributes
VehicleService
type VehicleService_VersionStructure
A work plan for a vehicle for a whole day, planned for a specific DAY TYPE. A VEHICLE SERVICE includes one or several VEHICLE SERVICE PARTs.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
Name
type MultilingualString
0..1
Name of VEHICLE SERVICE.
VehicleServiceGroup
Elements for VEHICLE SERVICE.
Description
type MultilingualString
0..1
Description of VEHICLE SERVICE.
vehicleServiceParts
type vehicleServiceParts_RelStructure
0..1
Parts of a VEHICLE SERVICE.

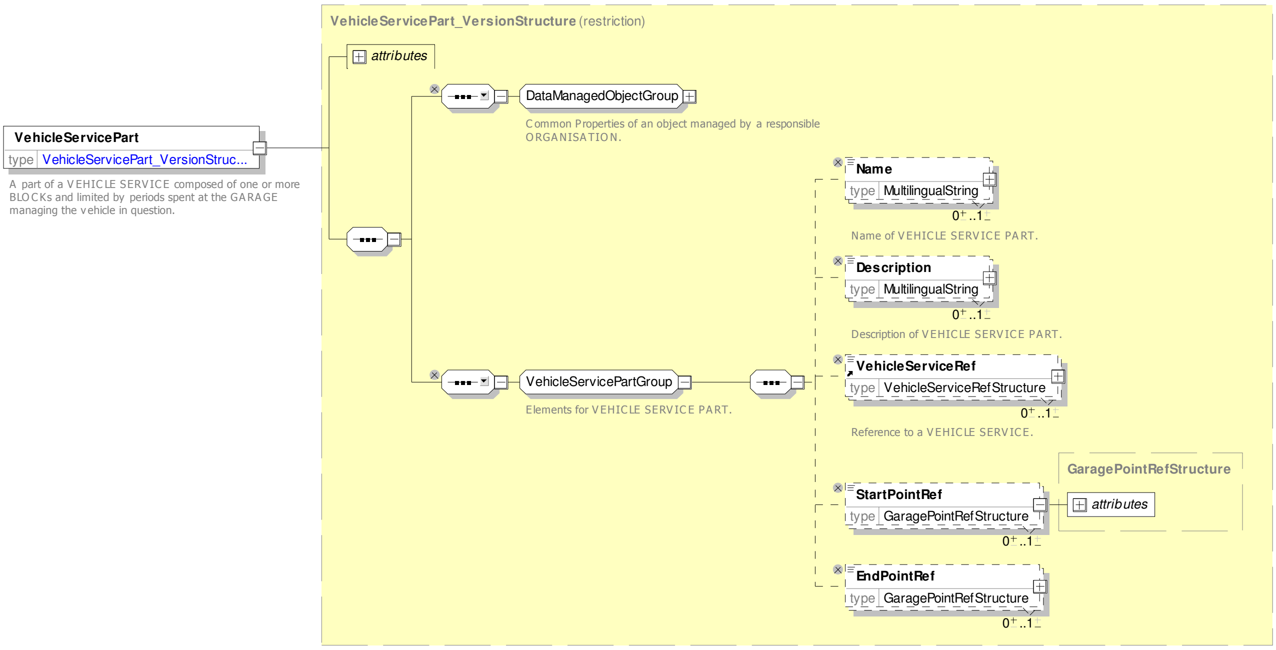
Figure 157 — *VehicleService* — XSD

### 8.2.2.3.7 VehicleServicePart – Model Element

A part of a VEHICLE SERVICE composed of one or more BLOCKs and limited by periods spent at the GARAGE managing the vehicle in question.

Table 88 – *VehicleServicePart* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | VEHICLE SERVICE PART inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *VehicleServicePartIdType* | 1:1 | Identifier of VEHICLE SERVICE PART. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of VEHICLE SERVICE PART. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of VEHICLE SERVICE PART. |
| «FK» | ***VehicleServiceRef*** | *VehicleServiceRef* | 1:1 | VEHICLE SERVICE of which this a VEHICLE SERVICE PART. |
| «FK» | ***StartPointRef*** | *GaragePointRef* | 0:1 | Start GARAGE POINT of VEHICLE SERVICE PART. |
| «FK» | ***EndPointRef*** | *GaragePointRef* | 0:1 | End GARAGE POINT of VEHICLE SERVICE PART. |


VehicleServicePart
type VehicleServicePart_VersionStruc...
A part of a VEHICLE SERVICE composed of one or more BLOCKs and limited by periods spent at the GARAGE managing the vehicle in question.
VehicleServicePart_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
Name
type MultilingualString
0..1
Name of VEHICLE SERVICE PART.
Description
type MultilingualString
0..1
Description of VEHICLE SERVICE PART.
VehicleServicePartGroup
Elements for VEHICLE SERVICE PART.
VehicleServiceRef
type VehicleServiceRefStructure
0..1
Reference to a VEHICLE SERVICE.
GaragePointRefStructure
attributes
StartPointRef
type GaragePointRefStructure
0..1
EndPointRef
type GaragePointRefStructure
0..1

## Figure 158 — *VehicleServicePart* — XSD

## 8.2.3 Train Service

### 8.2.3.1 TRAIN SERVICE – Conceptual MODEL

The TRAIN SERVICE Model allows TRAINs and parts of trains to be related to JOURNEY PARTs.

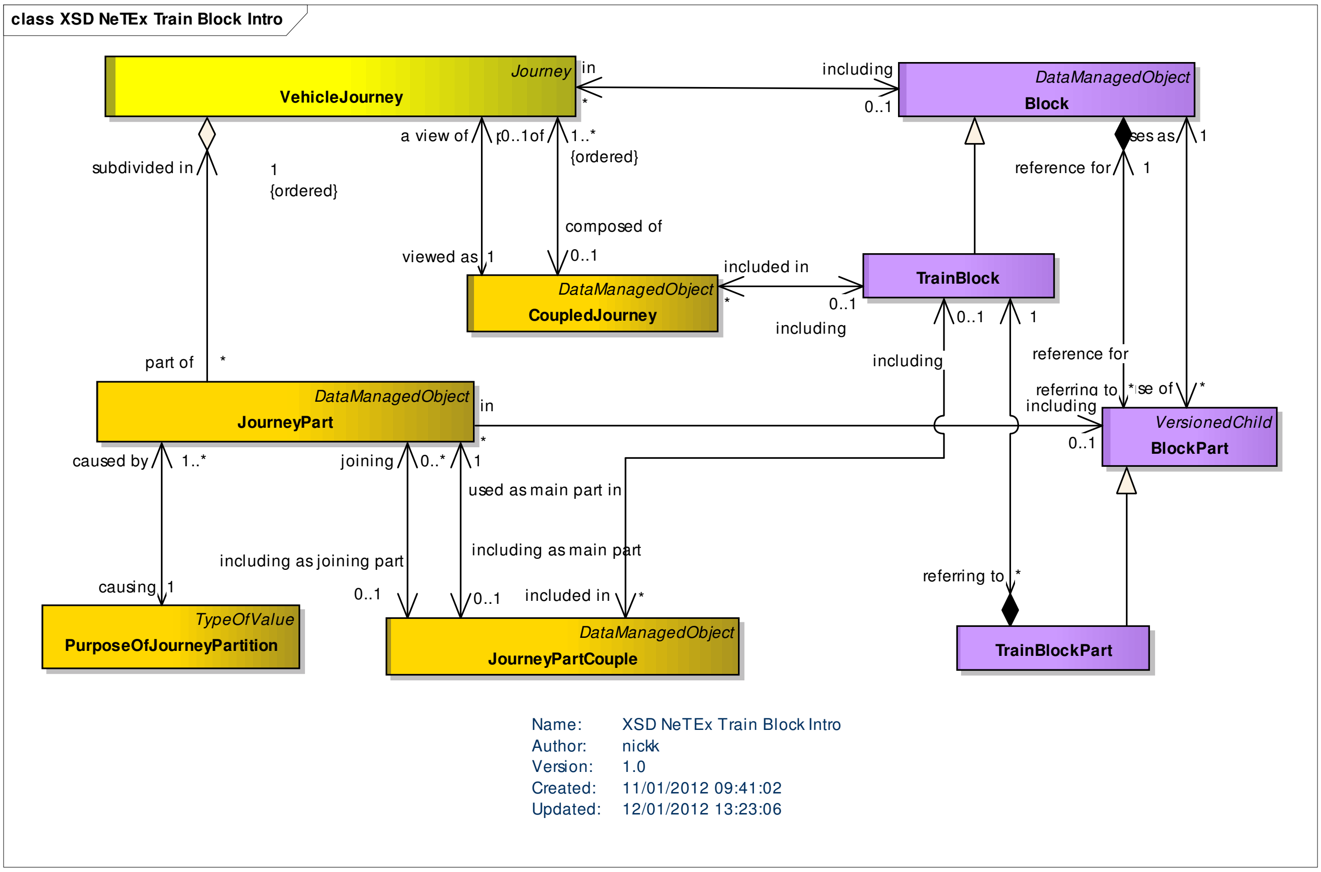
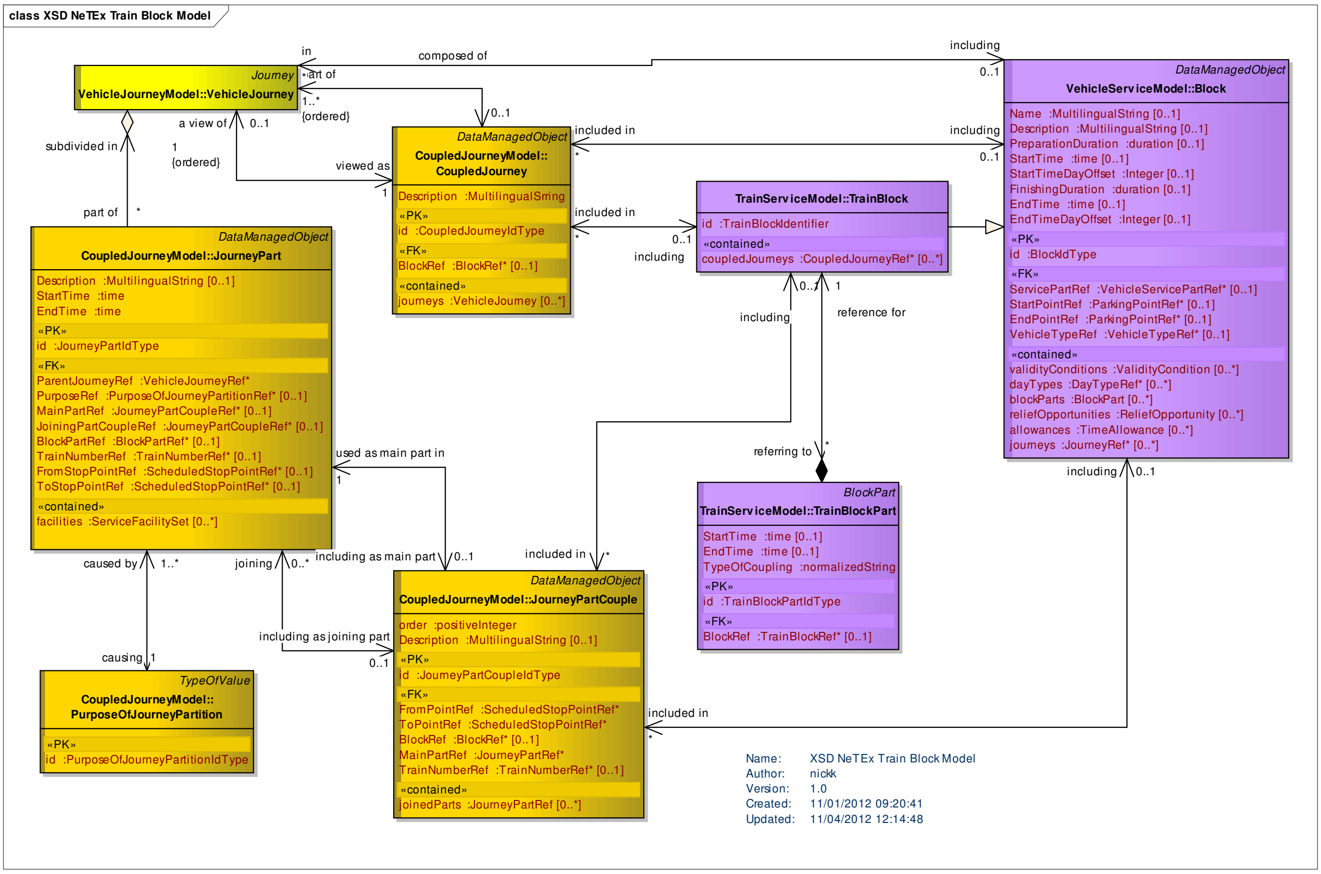


Figure 159 — Train Service Model – Conceptual MODEL (UML)

### 8.2.3.2 Train Service – Physical Model

The following figure shows the Physical model for the TRAIN SERVICE Model.



class XSD NeTEx Train Block Model
Journey
VehicleJourneyModel::VehicleJourney
in
composed of
part of
1..*
{ordered}
including
0..1
DataManagedObject
VehicleServiceModel::Block
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
PreparationDuration :duration [0..1]
StartTime :time [0..1]
StartTimeDayOffset :Integer [0..1]
FinishingDuration :duration [0..1]
EndTime :time [0..1]
EndTimeDayOffset :Integer [0..1]
«PK»
id :BlockIdType
«FK»
ServicePartRef :VehicleServicePartRef* [0..1]
StartPointRef :ParkingPointRef* [0..1]
EndPointRef :ParkingPointRef* [0..1]
VehicleTypeRef :VehicleTypeRef* [0..1]
«contained»
validityConditions :ValidityCondition [0..*]
dayTypes :DayTypeRef* [0..*]
blockParts :BlockPart [0..*]
reliefOpportunities :ReliefOpportunity [0..*]
allowances :TimeAllowance [0..*]
journeys :JourneyRef* [0..*]
subdivided in
a view of
0..1
1
{ordered}
0..1
viewed as
1
DataManagedObject
CoupledJourneyModel::
CoupledJourney
Description :MultilingualSrring
«PK»
id :CoupledJourneyIdType
«FK»
BlockRef :BlockRef* [0..1]
«contained»
journeys :VehicleJourney [0..*]
included in
*
including
0..1
included in
*
0..1
including
TrainServiceModel::TrainBlock
id :TrainBlockIdentifier
«contained»
coupledJourneys :CoupledJourneyRef* [0..*]
part of
*
DataManagedObject
CoupledJourneyModel::JourneyPart
Description :MultilingualString [0..1]
StartTime :time
EndTime :time
«PK»
id :JourneyPartIdType
«FK»
ParentJourneyRef :VehicleJourneyRef*
PurposeRef :PurposeOfJourneyPartitionRef* [0..1]
MainPartRef :JourneyPartCoupleRef* [0..1]
JoiningPartCoupleRef :JourneyPartCoupleRef* [0..1]
BlockPartRef :BlockPartRef* [0..1]
TrainNumberRef :TrainNumberRef* [0..1]
FromStopPointRef :ScheduledStopPointRef* [0..1]
ToStopPointRef :ScheduledStopPointRef* [0..1]
«contained»
facilities :ServiceFacilitySet [0..*]
0..1
1
including
reference for
used as main part in
1
referring to
*
including
0..1
BlockPart
TrainServiceModel::TrainBlockPart
StartTime :time [0..1]
EndTime :time [0..1]
TypeOfCoupling :normalizedString
«PK»
id :TrainBlockPartIdType
«FK»
BlockRef :TrainBlockRef* [0..1]
caused by
1..*
joining
0..*
including as main part
0..1
included in
*
DataManagedObject
CoupledJourneyModel::JourneyPartCouple
order :positiveInteger
Description :MultilingualString [0..1]
«PK»
id :JourneyPartCoupleIdType
«FK»
FromPointRef :ScheduledStopPointRef*
ToPointRef :ScheduledStopPointRef*
BlockRef :BlockRef* [0..1]
MainPartRef :JourneyPartRef*
TrainNumberRef :TrainNumberRef* [0..1]
«contained»
joinedParts :JourneyPartRef [0..*]
including as joining part
0..1
causing
1
TypeOfValue
CoupledJourneyModel::
PurposeOfJourneyPartition
«PK»
id :PurposeOfJourneyPartitionIdType
included in
*
Name: XSD NeTEx Train Block Model
Author: nickk
Version: 1.0
Created: 11/01/2012 09:20:41
Updated: 11/04/2012 12:14:48

Figure 160 — Train Service Model – Physical Model (UML)

### 8.2.3.3 TrainServiceModel – Attributes and XSD

#### 8.2.3.3.1 TrainBlock – Model Element

A composite train formed of several BLOCKs coupled together during a certain period. Any coupling or separation action marks the start of a new TRAIN BLOCK.

Table 89 – *TrainBlock* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *Block* | ::> | TRAIN BLOCK inherits from BLOCK. |
| | ***id*** | *TrainBlockIdentifier* | 1:1 | Identifier of a TRAIN BLOCK. |
| «cntd» | ***coupledJourneys*** | *CoupledJourneyRef* | 0:* | COUPLED JOURNEYs associated with COURSE OF JOURNEY. |

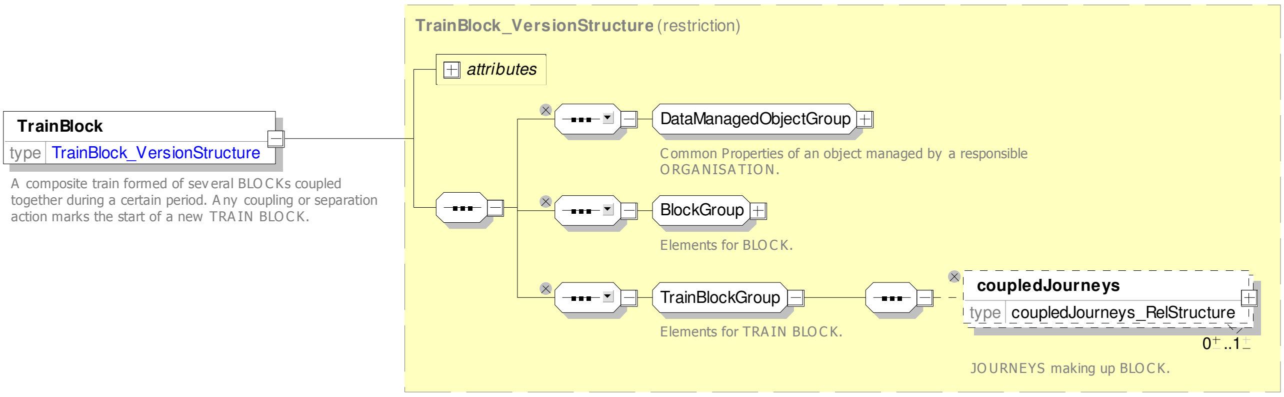

TrainBlock_VersionStructure (restriction)
attributes
TrainBlock
type TrainBlock_VersionStructure
A composite train formed of several BLOCKs coupled together during a certain period. Any coupling or separation action marks the start of a new TRAIN BLOCK.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
BlockGroup
Elements for BLOCK.
TrainBlockGroup
Elements for TRAIN BLOCK.
coupledJourneys
type coupledJourneys_RelStructure
0..1
JOURNEYS making up BLOCK.

Figure 161 — *TrainBlock* — XSD

### 8.2.3.3.2 TrainBlockPart – Model Element

The position of a vehicle BLOCK within a TRAIN BLOCK.

Table 90 – *TrainBlockPart* – Element

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *BlockPart* | ::> | TRAIN BLOCK PART inherits from BLOCK PART. |
| «PK» | ***id*** | *TrainBlockPartIdType* | 1:1 | Identifier of a TRAIN BLOCK PART. |
| | ***StartTime*** | *xsd:time* | 0:1 | Start time of TRAIN BLOCK PART. |
| | ***StartTime-DayOffset*** | *xsd:integer* | 0:1 | Day offset of start time from OPERATING DAY at start of journey. |
| | ***EndTime*** | *xsd:time* | 0:1 | End time of TRAIN BLOCK PART. |
| | ***EndTime-DayOffset*** | *xsd:integer* | 0:1 | Day offset of end time from OPERATING DAY at start of journey. |
| | ***TypeOfCoupling*** | *xsd:normalizedString* | 01 | TYPE OF COUPLING of TRAIN BLOCK PART. |

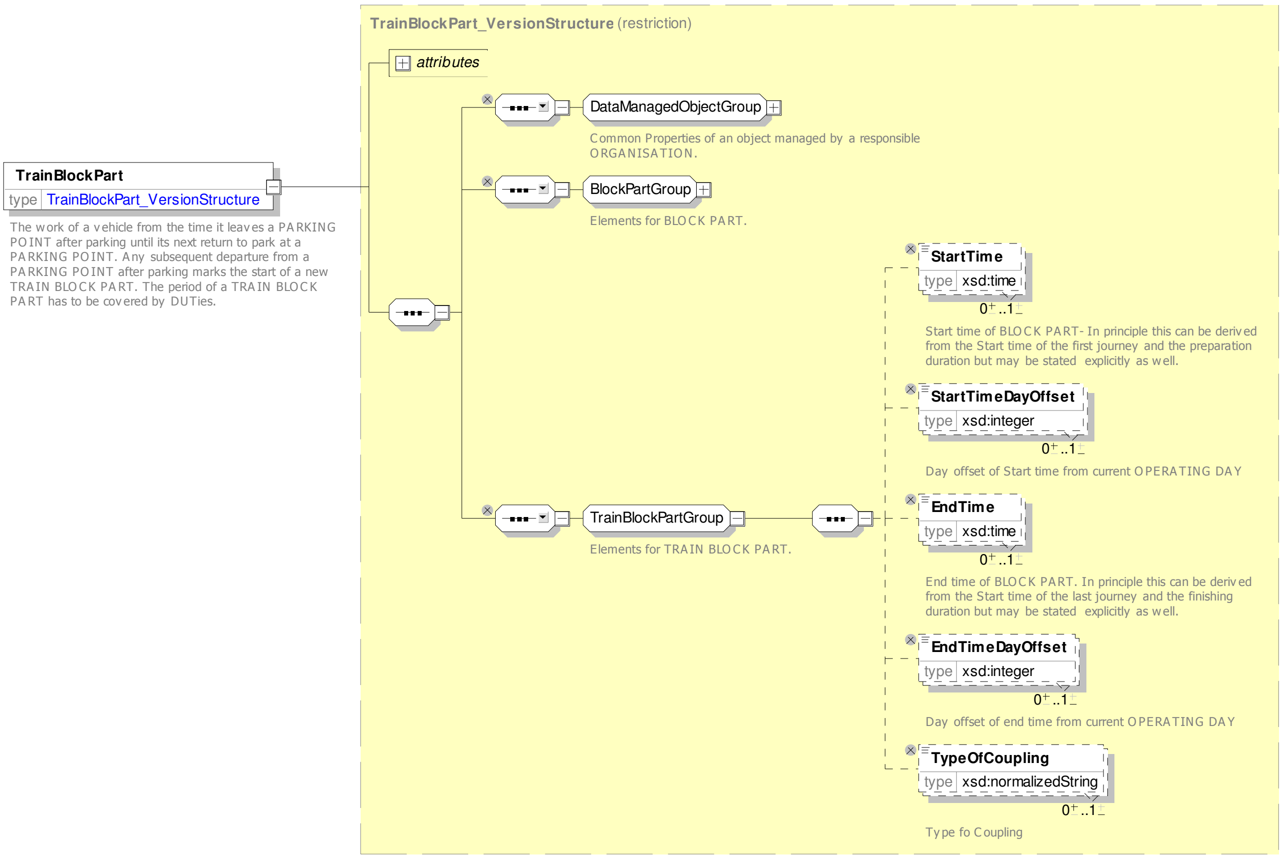

TrainBlockPart_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
BlockPartGroup
Elements for BLOCK PART.
TrainBlockPart
type TrainBlockPart_VersionStructure
The work of a vehicle from the time it leaves a PARKING POINT after parking until its next return to park at a PARKING POINT. Any subsequent departure from a PARKING POINT after parking marks the start of a new TRAIN BLOCK PART. The period of a TRAIN BLOCK PART has to be covered by DUTies.
StartTime
type xsd:time
0..1
Start time of BLOCK PART- In principle this can be derived from the Start time of the first journey and the preparation duration but may be stated explicitly as well.
StartTimeDayOffset
type xsd:integer
0..1
Day offset of Start time from current OPERATING DAY
TrainBlockPartGroup
Elements for TRAIN BLOCK PART.
EndTime
type xsd:time
0..1
End time of BLOCK PART. In principle this can be derived from the Start time of the last journey and the finishing duration but may be stated explicitly as well.
EndTimeDayOffset
type xsd:integer
0..1
Day offset of end time from current OPERATING DAY
TypeOfCoupling
type xsd:normalizedString
0..1
Type fo Coupling

**Figure 162 — *TrainBlockPart* — XSD**

# Annex A
# (informative)

# Monitoring & Control

## A.1 Introduction

Although NeTEx is dedicated to scheduled information exchange, it is important to have a good understanding of how it is related to realtime information. The following chapters provides such an explanation. Therefore the information and model below are mainly out of NeTEx scope (but still Transmodel compliant).

Real-time information exchange has to be manage using SIRI (CEN/TS 15531-1/5). SIRI and NeTEx are both based on Transmodel and IFOPT and are fully compatible.

## A.2 Monitoring & Control

### A.2.1 Monitored Vehicle Journey

#### A.2.1.1 Monitored Vehicle Journey – Conceptual MODEL

The following Monitored Vehicle Journey model is purely informative and is not fully explained here: refer to Transmodel for further details (most of the following explanation is directly extracted from Transmodel).

A MONITORED VEHICLE JOURNEY describes a journey recognised as being in process, at the time of the VEHICLE MONITORING.

Whenever possible, a MONITORED VEHICLE JOURNEY will be related to a DATED VEHICLE JOURNEY. This means that the system states that the journey underway is a DATED VEHICLE JOURNEY present in the latest valid plan.

In normal situations, this DATED VEHICLE JOURNEY is the expected journey, i.e. the DATED VEHICLE JOURNEY planned at that time in the DATED BLOCK assigned as work plan to the monitored LOGICAL VEHICLE. However, there may be some inconsistency with the work plan. It is in particular the case when the service has an important delay and that the LOGICAL VEHICLE is supposed to run a further DATED VEHICLE JOURNEY than the monitored journey. In some rare cases, the system may even monitor a DATED VEHICLE JOURNEY belonging to another DATED BLOCK than assigned to that LOGICAL VEHICLE.

In other situations, the system is unable to identify any DATED VEHICLE JOURNEY, but recognises a particular JOURNEY PATTERN as being served by the MONITORED VEHICLE JOURNEY. In classical AVM systems, this may happen if the vehicle follows an unexpected JOURNEY PATTERN, possibly because a control action has been ordered verbally, but not yet entered by the controller. Some modern systems, coupled to a geographical system, allow to build a ROUTE and a JOURNEY PATTERN unknown so far, when a vehicle serves them..

As long as the MONITORED VEHICLE JOURNEY is not completed, there is some uncertainty as regards the monitoring data. For instance, the vehicle may be monitored as serving a particular JOURNEY PATTERN, but operates an unexpected short turn at an intermediate TURN STATION. The system should be in principle able to declare a further VEHICLE MONITORING, modifying the previous one by indication of the actually served JOURNEY PATTERN (which has its end at the TURN STATION).

It is of importance to distinguish the latest valid plan from the monitored situation on the network. The monitored situation may be sometimes not precise enough (failure, gap in the location process, etc.) or not detailed enough to be related to a work plan. The latest valid plan is not always up to date, for instance when a control action is ordered verbally before having been entered in the monitoring system by the controller. In some systems, the work plan is even not registered in the system (e.g. when the system is only oriented towards passenger information).

#### A.2.1.1.1 Monitored Passing Times

Theoretical passing times at stop points may be computed from the basic timing information (see section 6.4.7). In the design of the operational plan for a specific OPERATING DAY, such planned passing times may be amended by changes made on the schedule. This will produce “target passing times” for this day, supporting the monitoring of operations. In addition, forecasts may be made on “estimated passing times”, mainly for passenger information, and actually “observed passing times” may be recorded by a monitoring system. All these concepts are described by the generic entity PASSING TIME, which has several sub-types.

Modifications of the plan for a specific OPERATING DAY, in particular changes in journeys, are likely to generate the need to recalculate the passing times. These modified passing times are defined as DATED PASSING TIMEs. Several sub-types of DATED PASSING TIME reflect the different status and use of passing times at different moments in time.

A first calculation of DATED PASSING TIMEs may be carried out as soon as the DATED VEHICLE JOURNEYs for a particular OPERATING DAY are defined. Such passing times are called TARGET PASSING TIMEs. They are calculated for all points that are defined as TIMING POINTs IN JOURNEY PATTERN for the corresponding DATED VEHICLE JOURNEYs.

The TARGET PASSING TIMEs are normally generated for the first time a few days before the OPERATING DAY, and are from that moment on constantly updated, according to the modifications operated to the schedule. The TARGET PASSING TIME can be seen as the latest official plan for vehicle operation.

The drivers will certainly try to meet the TARGET PASSING TIMEs, but in practice the vehicle may deviate from this plan due to the traffic circumstances. Various systems (such as AVM) can monitor these deviations and provide an estimation of PASSING TIMEs at the next stop points, for passenger information or for service control. This estimation may be based on planned run times or on mean run times. Such ESTIMATED PASSING TIMEs will be displayed at stops or will be used for trip planning queries. They may be used as well by the AVM system, in order for instance to alert the controller on a forecasted delay for a relief.

An ESTIMATED PASSING TIME is computed for a particular MONITORED VEHICLE JOURNEY, at a specified POINT IN JOURNEY PATTERN. The ESTIMATED PASSING TIMEs are constantly changing with every new information about the progress of the vehicle on its route. It can be useful to store some ESTIMATED PASSING TIMEs for statistical reasons.

The entity OBSERVED PASSING TIME describes the actual PASSING TIMEs that have been recorded at a specific POINT. This information is used to compute further ESTIMATED PASSING TIMEs and, of course, for statistical purposes. OBSERVED PASSING TIMEs are resulting from the monitoring process and are therefore related to a MONITORED VEHICLE JOURNEY. This entity may also be used for manual collection (surveys) of passing times. Any collected passing time that cannot be related to a MONITORED VEHICLE JOURNEY should be described, on the contrary, only as a VEHICLE DETECTING at the measurement POINT.

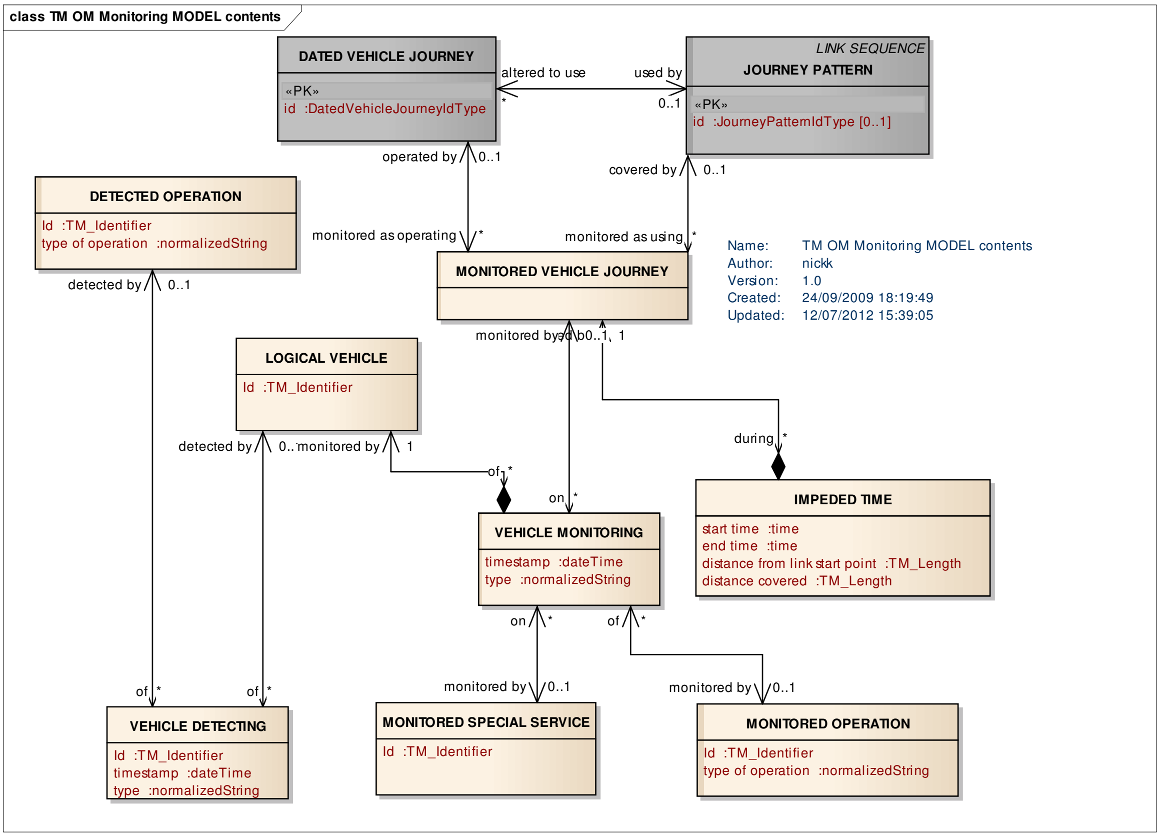


Figure 163 — Monitored Vehicle Journey — Conceptual MODEL (UML)

### A.2.1.2 Monitored Vehicle Journey – Physical Model

The following diagram shows the MONITORED VEHICLE JOURNEY physical model.

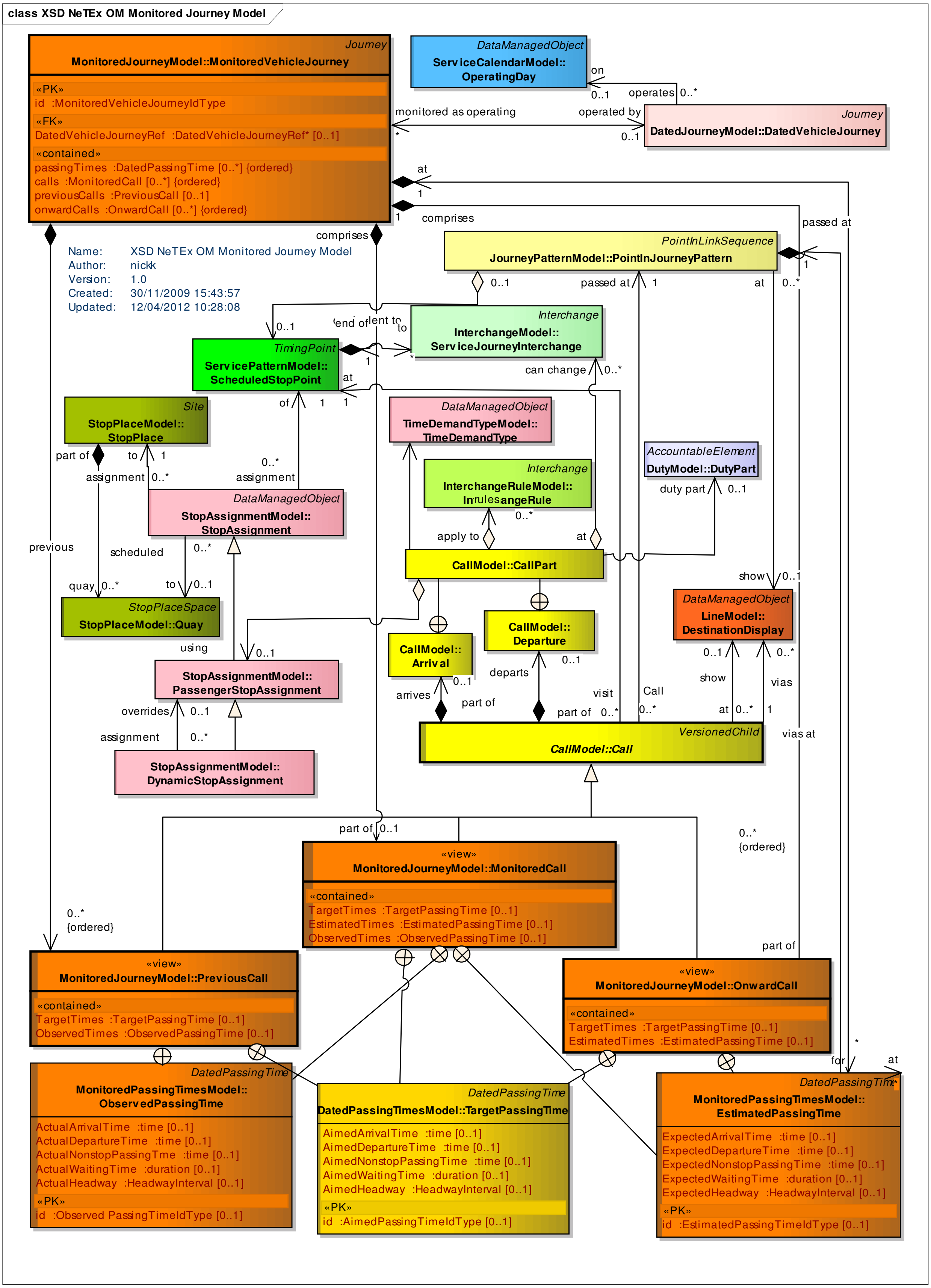

class XSD NeTEx OM Monitored Journey Model
Journey
MonitoredJourneyModel::MonitoredVehicleJourney
«PK»
id :MonitoredVehicleJourneyIdType
«FK»
DatedVehicleJourneyRef :DatedVehicleJourneyRef* [0..1]
«contained»
passingTimes :DatedPassingTime [0..*] {ordered}
calls :MonitoredCall [0..*] {ordered}
previousCalls :PreviousCall [0..1]
onwardCalls :OnwardCall [0..*] {ordered}
DataManagedObject
ServiceCalendarModel::OperatingDay
on
0..1
operates
0..*
monitored as operating
operated by
*
0..1
Journey
DatedJourneyModel::DatedVehicleJourney
at
1
1
comprises
passed at
comprises
Name: XSD NeTEx OM Monitored Journey Model
Author: nickk
Version: 1.0
Created: 30/11/2009 15:43:57
Updated: 12/04/2012 10:28:08
PointInLinkSequence
JourneyPatternModel::PointInJourneyPattern
0..1
passed at
1
at
0..*
1
0..1
end of
lent to
to
Interchange
InterchangeModel::ServiceJourneyInterchange
TimingPoint
ServicePatternModel::ScheduledStopPoint
1
*
can change
0..*
at
Site
StopPlaceModel::StopPlace
of
1
1
DataManagedObject
TimeDemandTypeModel::TimeDemandType
AccountableElement
DutyModel::DutyPart
part of
to
1
0..*
assignment
0..*
assignment
Interchange
InterchangeRuleModel::InterchangeRule
duty part
0..1
DataManagedObject
StopAssignmentModel::StopAssignment
0..*
apply to
at
previous
scheduled
0..*
CallModel::CallPart
show
0..1
quay
0..*
to
0..1
StopPlaceSpace
StopPlaceModel::Quay
DataManagedObject
LineModel::DestinationDisplay
CallModel::Departure
CallModel::Arrival
using
0..1
0..1
0..*
0..1
0..1
departs
show
vias
StopAssignmentModel::PassengerStopAssignment
arrives
part of
visit
Call
overrides
0..1
part of
0..*
0..*
at
0..*
1
assignment
0..*
VersionedChild
CallModel::Call
vias at
StopAssignmentModel::DynamicStopAssignment
part of
0..1
0..*
{ordered}
«view»
MonitoredJourneyModel::MonitoredCall
«contained»
TargetTimes :TargetPassingTime [0..1]
EstimatedTimes :EstimatedPassingTime [0..1]
ObservedTimes :ObservedPassingTime [0..1]
0..*
{ordered}
part of
«view»
MonitoredJourneyModel::PreviousCall
«contained»
TargetTimes :TargetPassingTime [0..1]
ObservedTimes :ObservedPassingTime [0..1]
«view»
MonitoredJourneyModel::OnwardCall
«contained»
TargetTimes :TargetPassingTime [0..1]
EstimatedTimes :EstimatedPassingTime [0..1]
*
for
at
DatedPassingTime
MonitoredPassingTimesModel::ObservedPassingTime
ActualArrivalTime :time [0..1]
ActualDepartureTime :time [0..1]
ActualNonstopPassingTme :time [0..1]
ActualWaitingTime :duration [0..1]
ActualHeadway :HeadwayInterval [0..1]
«PK»
id :Observed PassingTimeIdType [0..1]
DatedPassingTime
DatedPassingTimesModel::TargetPassingTime
AimedArrivalTime :time [0..1]
AimedDepartureTime :time [0..1]
AimedNonstopPassingTime :time [0..1]
AimedWaitingTime :duration [0..1]
AimedHeadway :HeadwayInterval [0..1]
«PK»
id :AimedPassingTimeIdType [0..1]
DatedPassingTime
MonitoredPassingTimesModel::EstimatedPassingTime
ExpectedArrivalTime :time [0..1]
ExpectedDepartureTime :time [0..1]
ExpectedNonstopPassingTime :time [0..1]
ExpectedWaitingTime :duration [0..1]
ExpectedHeadway :HeadwayInterval [0..1]
«PK»
id :EstimatedPassingTimeIdType [0..1]

Figure 164 — Monitored Vehicle Journey – Physical Model (UML)

### A.2.1.3 Monitored JourneyModel – Attributes and XSD

#### A.2.1.3.1 MonitoredVehicleJourney – Model Element

A journey that is monitored as being operated by a LOGICAL VEHICLE. According to the monitoring system capabilities, a MONITORED VEHICLE JOURNEY may be related to a DATED VEHICLE JOURNEY, or only to a JOURNEY PATTERN.

**Table 91 – MonitoredVehicleJourney – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| «PK» | ***id*** | *MonitoredVehicleJourney-IdType* | 1:1 | Identifier of MONITORED VEHICLE JOURNEY. |
| «FK» | ***DatedVehicle-JourneyRef*** | *DatedVehicleJourneyRef* | 0:1 | DATED VEHICLE JOURNEY which this MONITORED VEHICLE JOURNEY follows. |
| «cntd» | ***passingTimes*** | *DatedPassingTime* | 0:* | Passing times for MONITORED VEHICLE JOURNEY. |
| «cntd» | ***calls*** | *MonitoredCall* | 0:* | MONITORED CALLs for MONITORED VEHICLE JOURNEY. |
| «cntd» | ***previousCalls*** | *PreviousCall* | 0:1 | PREVIOUS CALLs before current stop in JOURNEY. |
| «cntd» | ***onwardCalls*** | *OnwardCall* | 0:* | ONWARD CALLs after current stop in journey. |

#### A.2.1.3.2 MonitoredSpecialService – Model Element

A special service that is monitored as being operated by a LOGICAL VEHICLE.

**Table 92 – MonitoredSpecialService – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *Journey* | ::> | MONITORED SPECIAL SERVICE inherits from JOURNEY. |
| «PK» | ***id*** | *Monitored-SpecialServiceIdType* | 1:1 | Identifier of MONITORED SPECIAL SERVICE. |
| «FK» | ***SpecialServiceRef*** | *SpecialServiceRef* | 0:1 | Reference tyo a MONITORED SPECIAL SERVICE. |

#### A.2.1.3.3 MonitoredCall – Model Element

A CALL in a MONITORED VEHICLE JOURNEY for which real-time data is available.

**Table 93 – *MonitoredCall* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *Call* | ::> | MONI\|TORED CALL inherits from CALL. |
| «cntd» | ***TargetTimes*** | *TargetPassingTime* | 0:1 | Intended times at TIMING POINT IN JOURNEY PATTERN,. |
| «cntd» | ***EstimatedTimes*** | *EstimatedPassingTime* | 0:1 | Estimated times at TIMING POINT IN JOURNEY PATTERN. |
| «cntd» | ***ObservedTimes*** | *ObservedPassingTime* | 0:1 | Observed times at TIMING POINT IN JOURNEY PATTERN. |
|  | ***previousCalls*** | *PreviousCall* | 0:* | CALLs previous to this CALL. |
|  | ***onwardsCalls*** | *OnwardCall* | 0:* | CALLs subsequent to this CALL. |

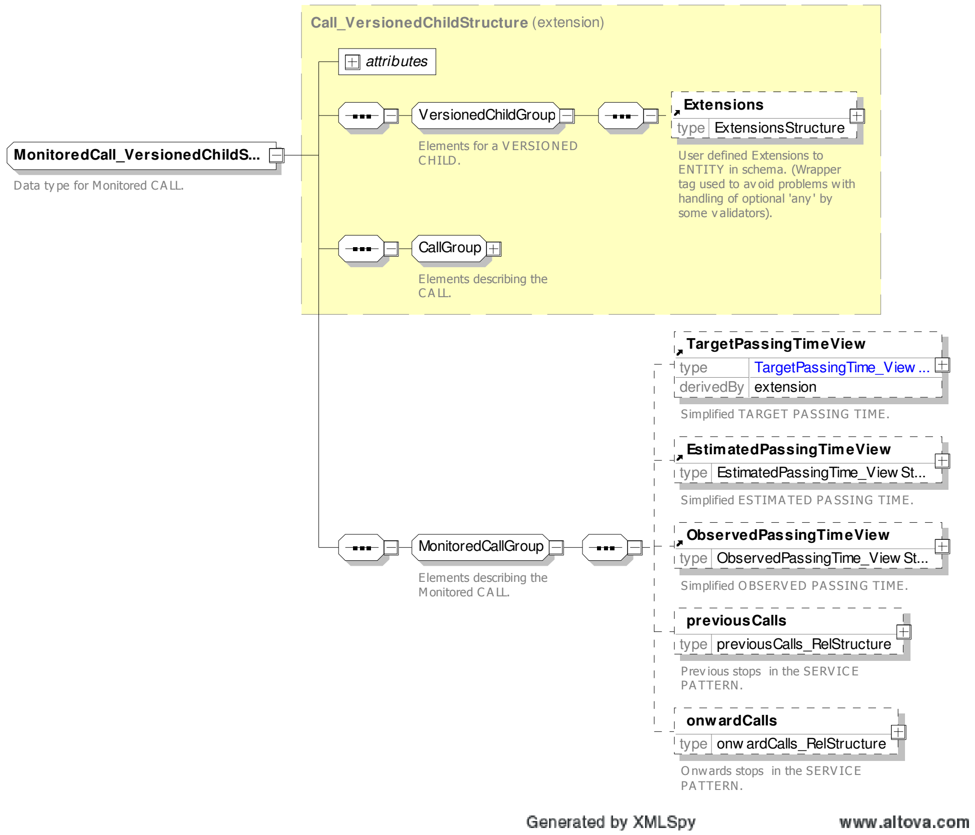

Call_VersionedChildStructure (extension)
attributes
MonitoredCall_VersionedChildS...
Data type for Monitored CALL.
VersionedChildGroup
Elements for a VERSIONED CHILD.
Extensions
type ExtensionsStructure
User defined Extensions to ENTITY in schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
CallGroup
Elements describing the CALL.
TargetPassingTimeView
type TargetPassingTime_View ...
derivedBy extension
Simplified TARGET PASSING TIME.
EstimatedPassingTimeView
type EstimatedPassingTime_View St...
Simplified ESTIMATED PASSING TIME.
ObservedPassingTimeView
type ObservedPassingTime_View St...
Simplified OBSERVED PASSING TIME.
MonitoredCallGroup
Elements describing the Monitored CALL.
previousCalls
type previousCalls_RelStructure
Previous stops in the SERVICE PATTERN.
onwardCalls
type onwardCalls_RelStructure
Onwards stops in the SERVICE PATTERN.
Generated by XMLSpy
www.altova.com

**Figure 165 — MonitoredCall — XSD**

### A.2.1.3.4 PreviousCall – Model Element

An already completed CALL of a MONITORED VEHICLE JOURNEY that occurred earlier in the JOURNEY PATTERN before the current stop.

Table 94 – *PreviousCall* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *Call* | ::> | PREVIOUS CALL inherits from CALL. |
| «cntd» | ***TargetTimes*** | *TargetPassingTime* | 0:1 | Intended times at TIMING POINT IN JOURNEY PATTERN. |
| «cntd» | ***ObservedTimes*** | *ObservedPassingTime* | 0:1 | Observed times at TIMING POINT IN JOURNEY PATTERN. |

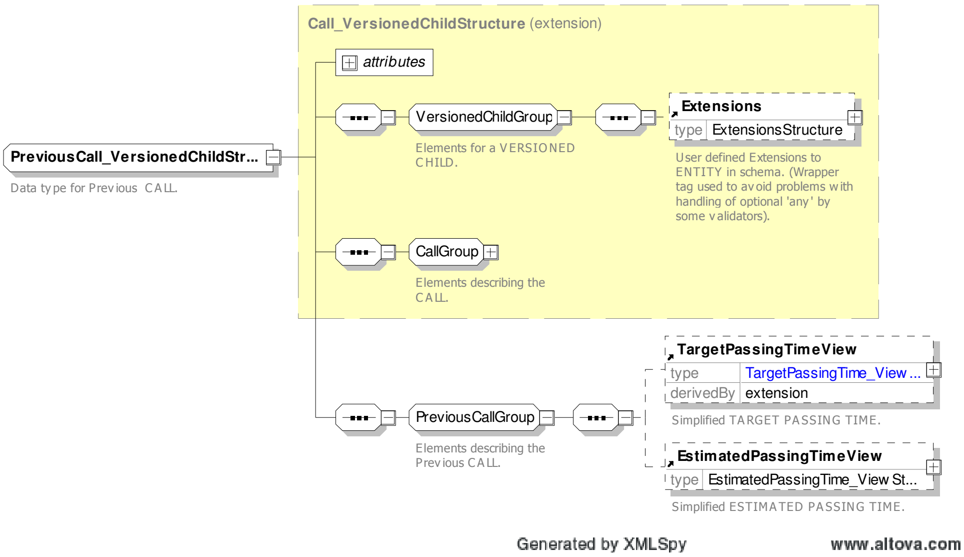

Call_VersionedChildStructure (extension)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
Extensions
type ExtensionsStructure
User defined Extensions to ENTITY in schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
PreviousCall_VersionedChildStr...
Data type for Previous CALL.
CallGroup
Elements describing the CALL.
TargetPassingTimeView
type TargetPassingTime_View...
derivedBy extension
Simplified TARGET PASSING TIME.
PreviousCallGroup
Elements describing the Previous CALL.
EstimatedPassingTimeView
type EstimatedPassingTime_View St...
Simplified ESTIMATED PASSING TIME.
Generated by XMLSpy
www.altova.com

Figure 166 — *PreviousCall* — XSD

### A.2.1.3.5 OnwardCall – Model Element

A further CALL of a MONITORED VEHICLE JOURNEY that will occurred later in the JOURNEY PATTERN after the current stop.

Table 95 – *OnwardCall* – Element

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *Call* | ::> | ONWARDS CALL inherits from CALL. |
| «cntd» | ***TargetTimes*** | *TargetPassingTime* | 0:1 | Intended times at TIMING POINT IN JOURNEY PATTERN. |
| «cntd» | ***EstimatedTimes*** | *EstimatedPassingTime* | 0:1 | Estimated times at TIMING POINT IN JOURNEY PATTERN. |

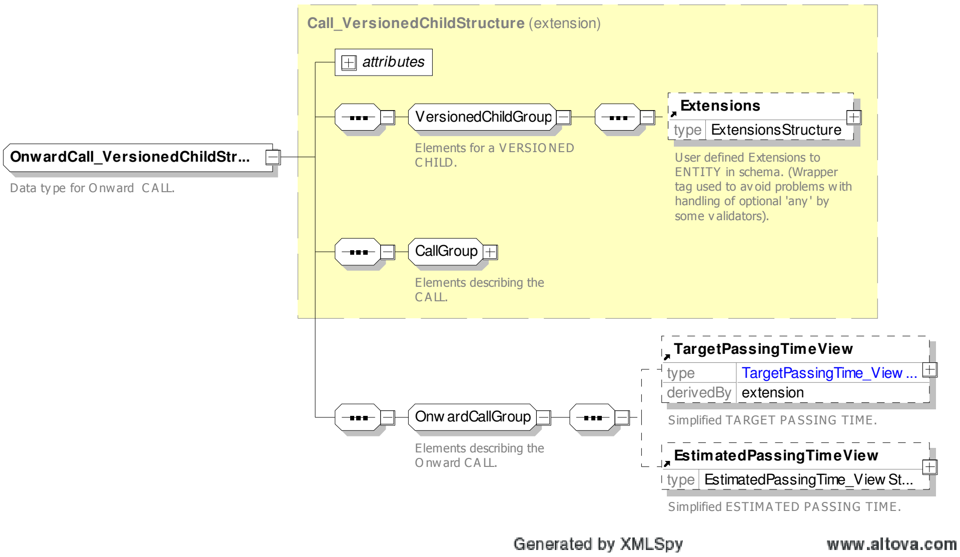

Call_VersionedChildStructure (extension)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
Extensions
type ExtensionsStructure
User defined Extensions to ENTITY in schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
OnwardCall_VersionedChildStr...
Data type for Onward CALL.
CallGroup
Elements describing the CALL.
TargetPassingTimeView
type TargetPassingTime_View ...
derivedBy extension
Simplified TARGET PASSING TIME.
OnwardCallGroup
Elements describing the Onward CALL.
EstimatedPassingTimeView
type EstimatedPassingTime_View St...
Simplified ESTIMATED PASSING TIME.
Generated by XMLSpy
www.altova.com

Figure 167 — OnwardCall — XSD

## A.2.2 Dated Passing Times – Physical Model

Additional information about DATED PASSING TIMEs physical model are provided here: they are providing the link between NeTEx and SIRI concepts. The physical model is reminded below (see 7.2.13.6).

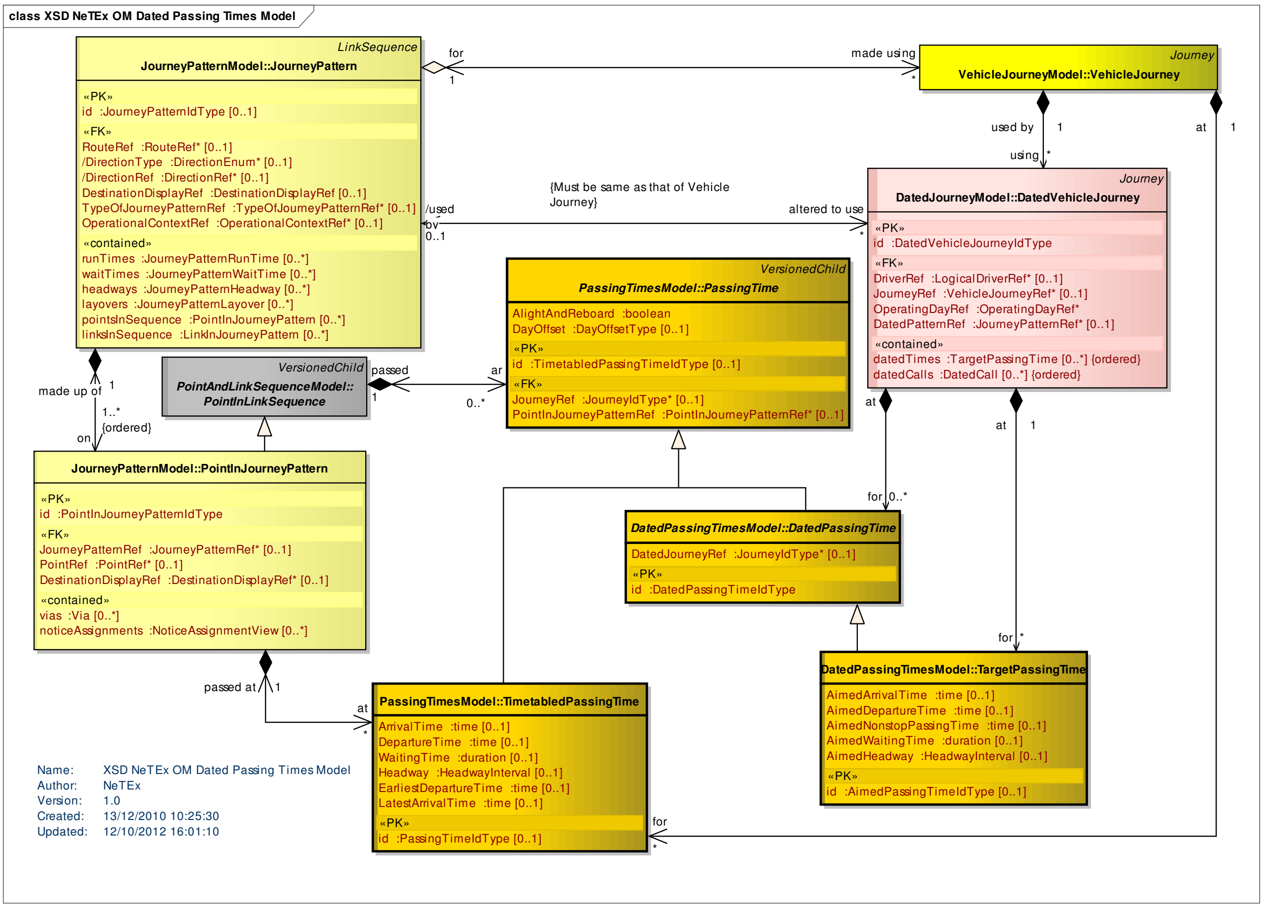

class XSD NeTEx OM Dated Passing Times Model
LinkSequence
JourneyPatternModel::JourneyPattern
«PK»
id :JourneyPatternIdType [0..1]
«FK»
RouteRef :RouteRef* [0..1]
/DirectionType :DirectionEnum* [0..1]
/DirectionRef :DirectionRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef [0..1]
TypeOfJourneyPatternRef :TypeOfJourneyPatternRef* [0..1]
OperationalContextRef :OperationalContextRef* [0..1]
«contained»
runTimes :JourneyPatternRunTime [0..*]
waitTimes :JourneyPatternWaitTime [0..*]
headways :JourneyPatternHeadway [0..*]
layovers :JourneyPatternLayover [0..*]
pointsInSequence :PointInJourneyPattern [0..*]
linksInSequence :LinkInJourneyPattern [0..*]
for
1
made using
*
Journey
VehicleJourneyModel::VehicleJourney
used by
1
using
*
at
1
{Must be same as that of Vehicle Journey}
/used by
0..1
altered to use
*
Journey
DatedJourneyModel::DatedVehicleJourney
«PK»
id :DatedVehicleJourneyIdType
«FK»
DriverRef :LogicalDriverRef* [0..1]
JourneyRef :VehicleJourneyRef* [0..1]
OperatingDayRef :OperatingDayRef*
DatedPatternRef :JourneyPatternRef* [0..1]
«contained»
datedTimes :TargetPassingTime [0..*] {ordered}
datedCalls :DatedCall [0..*] {ordered}
VersionedChild
PassingTimesModel::PassingTime
AlightAndReboard :boolean
DayOffset :DayOffsetType [0..1]
«PK»
id :TimetabledPassingTimeIdType [0..1]
«FK»
JourneyRef :JourneyIdType* [0..1]
PointInJourneyPatternRef :PointInJourneyPatternRef* [0..1]
made up of
1
1..*
{ordered}
on
VersionedChild
PointAndLinkSequenceModel::PointInLinkSequence
passed
1
ar
0..*
at
at
1
for
0..*
JourneyPatternModel::PointInJourneyPattern
«PK»
id :PointInJourneyPatternIdType
«FK»
JourneyPatternRef :JourneyPatternRef* [0..1]
PointRef :PointRef* [0..1]
DestinationDisplayRef :DestinationDisplayRef* [0..1]
«contained»
vias :Via [0..*]
noticeAssignments :NoticeAssignmentView [0..*]
DatedPassingTimesModel::DatedPassingTime
DatedJourneyRef :JourneyIdType* [0..1]
«PK»
id :DatedPassingTimeIdType
for
*
DatedPassingTimesModel::TargetPassingTime
AimedArrivalTime :time [0..1]
AimedDepartureTime :time [0..1]
AimedNonstopPassingTime :time [0..1]
AimedWaitingTime :duration [0..1]
AimedHeadway :HeadwayInterval [0..1]
«PK»
id :AimedPassingTimeIdType [0..1]
passed at
1
at
*
PassingTimesModel::TimetabledPassingTime
ArrivalTime :time [0..1]
DepartureTime :time [0..1]
WaitingTime :duration [0..1]
Headway :HeadwayInterval [0..1]
EarliestDepartureTime :time [0..1]
LatestArrivalTime :time [0..1]
«PK»
id :PassingTimeIdType [0..1]
for
*
Name: XSD NeTEx OM Dated Passing Times Model
Author: NeTEx
Version: 1.0
Created: 13/12/2010 10:25:30
Updated: 12/10/2012 16:01:10

Figure 168 — Dated Passing Times – Physical Model (UML)

### A.2.2.1 Passing times – Attributes and XSD

#### A.2.2.1.1 EstimatedPassingTime – Model Element

Time data, calculated from the latest available input, about when a public transport vehicle will pass a particular POINT IN JOURNEY PATTERN on a specified DATED VEHICLE JOURNEY. These are mainly used to inform passengers about expected times of arrival and/or departure, but may also be used for monitoring and re-planning.

**Table 96 – *EstimatedPassingTime* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DatedPassingTime* | *::>* | ESTIMATED PASSING TIME inherits from DATED PASSING TIME. |
| «PK» | ***id*** | *EstimatedPassing-TimeIdType* | 0:1 | Identifier of ESTIMATED PASSING TIME. |
| | ***ExpectedDeparture-Time*** | *xsd:time* | 0:1 | Expected departure time at POINT IN JOURNEY PATTERN. |
| | ***ExpectedArrivalTime*** | *xsd:time* | 0:1 | Expected arrival time at POINT IN JOURNEY PATTERN. |
| | ***ExpectedWaitingTime*** | *xsd:duration* | 0:1 | Expected waiting time at POINT IN JOURNEY PATTERN. |
| | ***ExpectedNonstop-PassingTime*** | *xsd:time* | 0:1 | Expected nonstop passing time at POINT IN JOURNEY PATTERN. |
| | ***ExpectedHeadway*** | *HeadwayInterval* | 0:1 | Expected headway time at POINT IN JOURNEY PATTERN. |

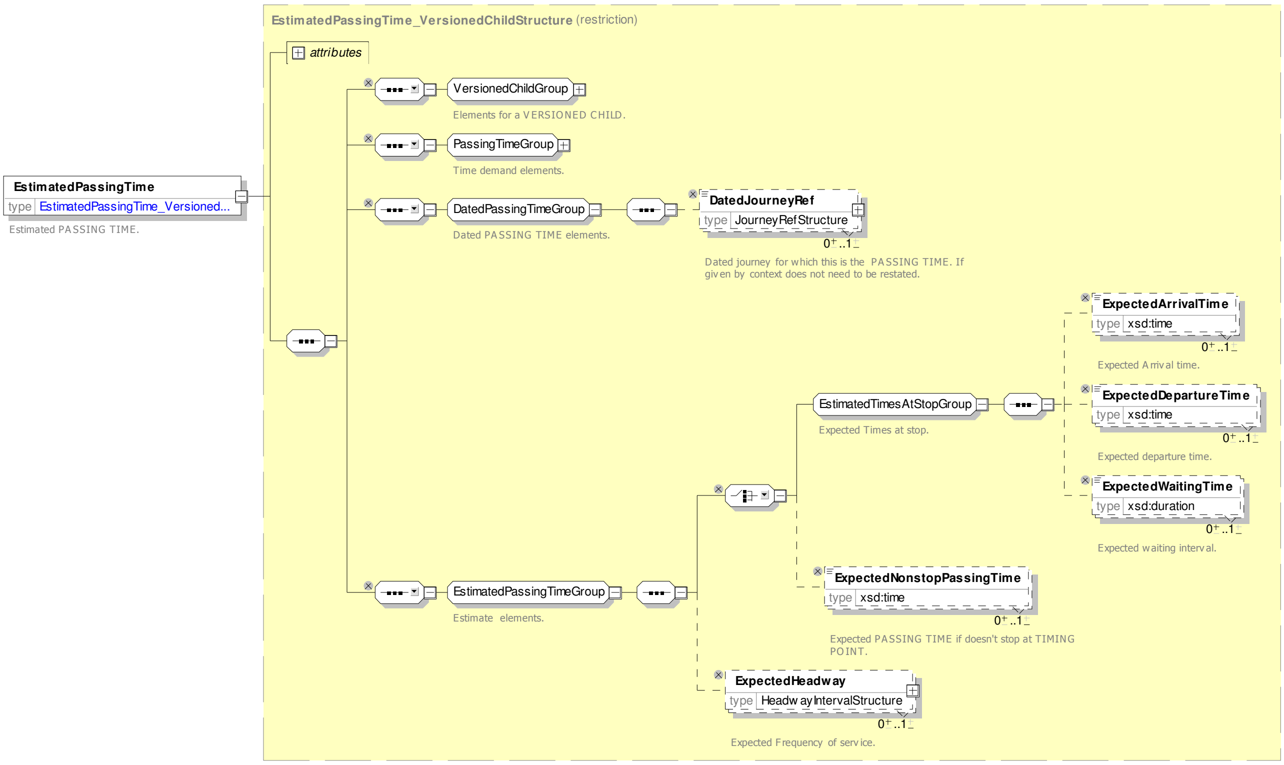

EstimatedPassingTime_VersionedChildStructure (restriction)
attributes
VersionedChildGroup
Elements for a VERSIONED CHILD.
PassingTimeGroup
Time demand elements.
EstimatedPassingTime
type EstimatedPassingTime_Versioned...
Estimated PASSING TIME.
DatedPassingTimeGroup
Dated PASSING TIME elements.
DatedJourneyRef
type JourneyRefStructure
0..1
Dated journey for which this is the PASSING TIME. If given by context does not need to be restated.
ExpectedArrivalTime
type xsd:time
0..1
Expected Arrival time.
EstimatedTimesAtStopGroup
Expected Times at stop.
ExpectedDepartureTime
type xsd:time
0..1
Expected departure time.
ExpectedWaitingTime
type xsd:duration
0..1
Expected waiting interval.
EstimatedPassingTimeGroup
Estimate elements.
ExpectedNonstopPassingTime
type xsd:time
0..1
Expected PASSING TIME if doesn't stop at TIMING POINT.
ExpectedHeadway
type HeadwayIntervalStructure
0..1
Expected Frequency of service.

**Figure 169 — *EstimatedPassingTime* — XSD**

### A.2.2.1.2 ObservedPassingTime – Model Element

The actual passing of a public transport vehicle at a pre-defined POINT during a MONITORED VEHICLE JOURNEY.

**Table 97 – *ObservedPassingTime* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *DatedPassingTime* | ::> | OBSERVED PASSING TIME inherits from DATED PASSING TIME. |
| «PK» | ***id*** | *Observed PassingTimeIdType* | 0:1 | Identifier of OBSERVED PASSING TIME. |
| | ***ActualArrivalTime*** | *xsd:time* | 0:1 | Actual arrival time at POINT IN JOURNEY PATTERN. |
| | ***ActualDepartureTime*** | *xsd:time* | 0:1 | Actual departure time at POINT IN JOURNEY PATTERN. |
| | ***ActualWaitingTime*** | *xsd:duration* | 0:1 | Actual waiting time at POINT IN JOURNEY PATTERN. |
| | ***ActualNonstop-PassingTme*** | *xsd:time* | 0:1 | Actual nonstop passing time at POINT IN JOURNEY PATTERN. |
| | ***ActualHeadway*** | *HeadwayInterval* | 0:1 | Actual headway time at POINT IN JOURNEY PATTERN. |

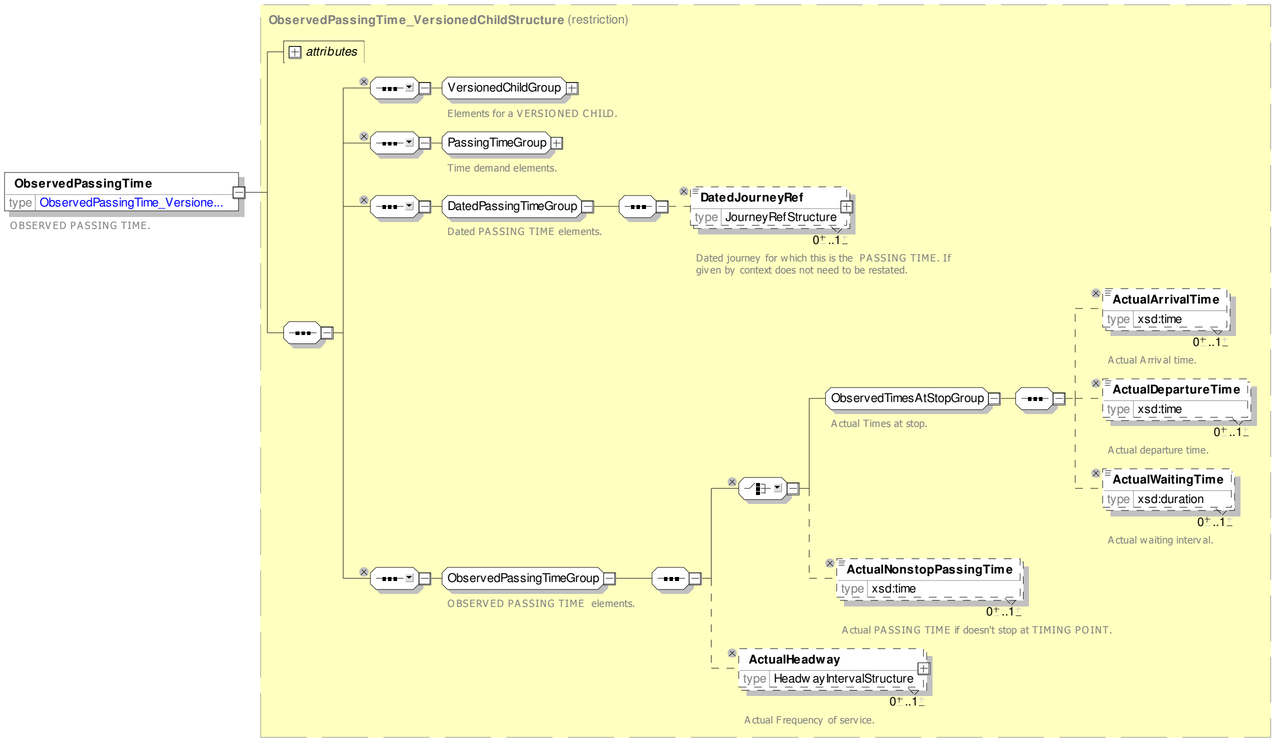


**Figure 170 — *ObservedPassingTime* — XSD**

### A.2.2.1.3 TargetPassingTime – Model Element

Time data about when a public transport vehicle should pass a particular POINT IN JOURNEY PATTERN on a particular DATED VEHICLE JOURNEY, in order to match the latest valid plan.

**Table 98 – *TargetPassingTime* – Element**

| Classification | Name | Type | Cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DatedPassingTime* | *::>* | TARGET PASSING TIME inherits from DATED PASSING TIME. |
| «PK» | ***id*** | *AimedPassingTime-IdType* | 0:1 | Identifier of AIMED PASSING TIME. |
| | ***AimedArrivalTime*** | *xsd:time* | 0:1 | Intended arrival time at TIMING POINT IN JOURNEY PATTERN. |
| | ***AimedDepartureTime*** | *xsd:time* | 0:1 | Intended departure time at TIMING POINT IN JOURNEY PATTERN. |
| | ***AimedWaitingTime*** | *xsd:duration* | 0:1 | Aimed waiting time at TIMING POINT IN JOURNEY PATTERN. |
| | ***AimedNonstop-PassingTime*** | *xsd:time* | 0:1 | Intended passing time at TIMING POINT IN JOURNEY PATTERN if vehicle does not stop. |
| | ***AimedHeadway*** | *HeadwayInterval* | 0:1 | Aimed headway interval at POINT IN JOURNEY PATTERN. |

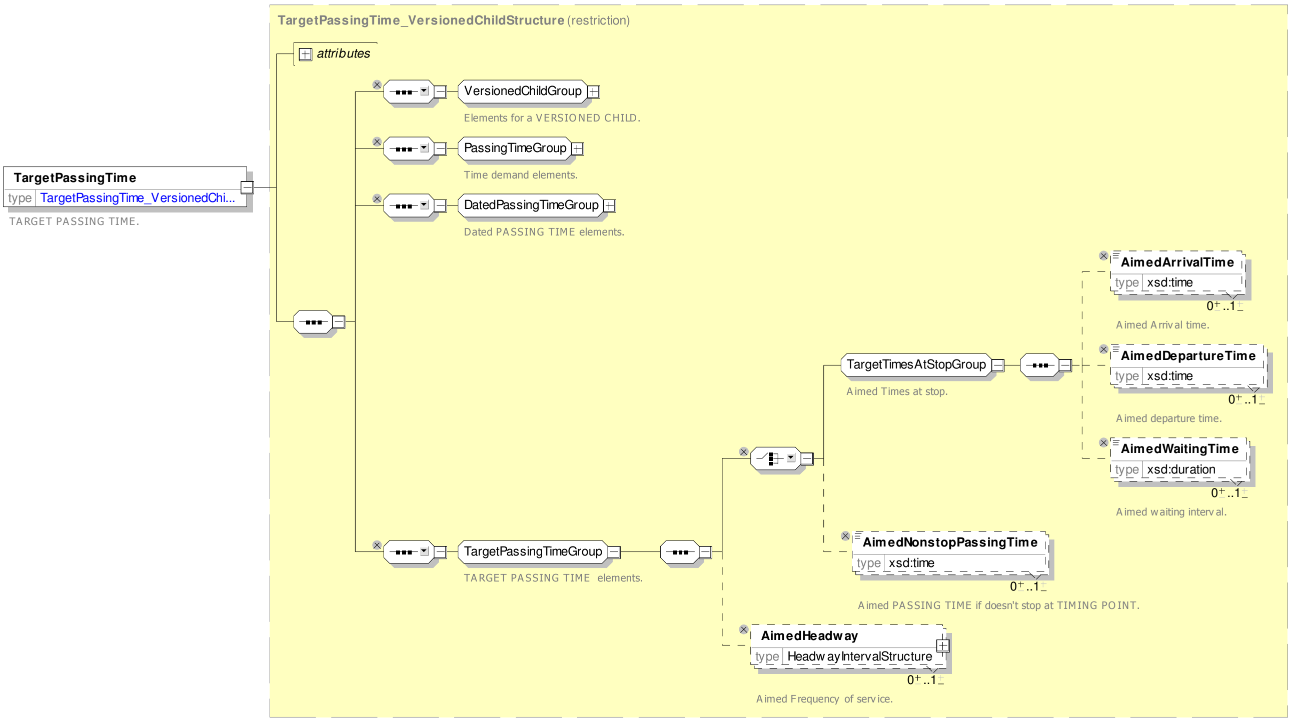


Figure 171 — *TargetPassingTime* — XSD

# Annex B
(informative)

# Driver Scheduling

## B.1 Introduction

Although DRIVERs and DUTY are not in NeTEx's scope, it is important to have a good understanding of how they could possibly be connected together. The following chapters provides such an explanation. Therefore the information and model below are only informative.

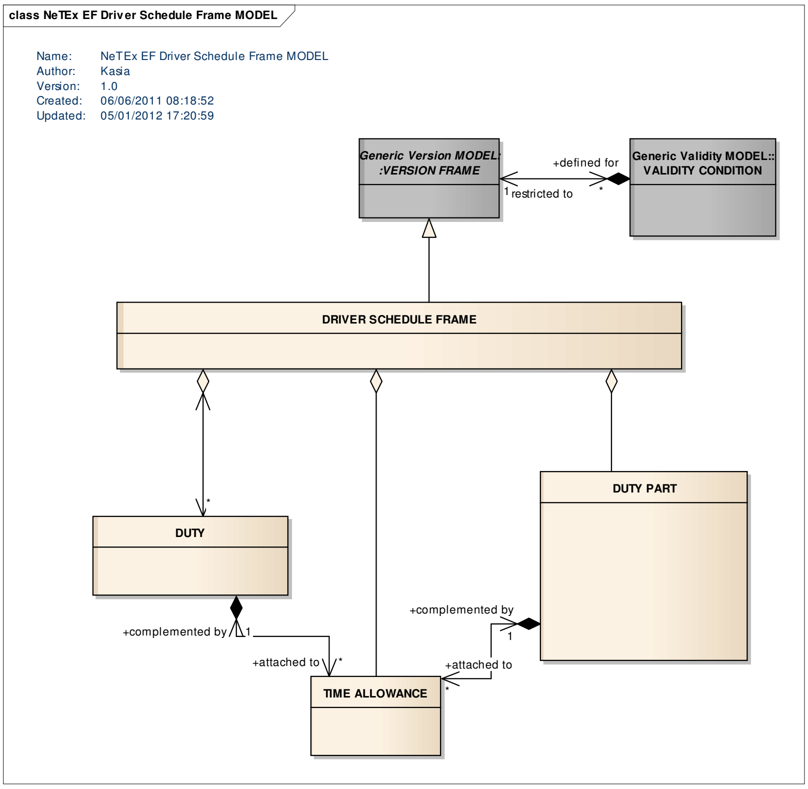
## B.2 Driver Scheduling

### B.2.1 Driver Schedule Frame

#### B.2.1.1 DRIVER SCHEDULE FRAME – Conceptual MODEL

The elements of the Network Description model can be grouped with a DRIVER SCHEDULED FRAME which holds a coherent set of driver related elements for data exchange. See VERSION FRAME in the NeTEx Framework section for general concepts relating to version frames.

The DRIVER SCHEDULE FRAME holds DUTies and DUTY parts, describing a driver's allocated work.



class NeTEx EF Driver Schedule Frame MODEL
Name: NeTEx EF Driver Schedule Frame MODEL
Author: Kasia
Version: 1.0
Created: 06/06/2011 08:18:52
Updated: 05/01/2012 17:20:59
Generic Version MODEL::VERSION FRAME
+defined for
1 restricted to
*
Generic Validity MODEL::VALIDITY CONDITION
DRIVER SCHEDULE FRAME
*
DUTY
DUTY PART
+complemented by
1
+complemented by
1
+attached to
*
+attached to
*
TIME ALLOWANCE

Figure 172 — Driver Schedule Frame – Conceptual MODEL (UML)

### B.2.1.2 Driver Schedule Frame – Physical Model

The following diagram shows the Physical model for a DRIVER SCHEDULE FRAME.

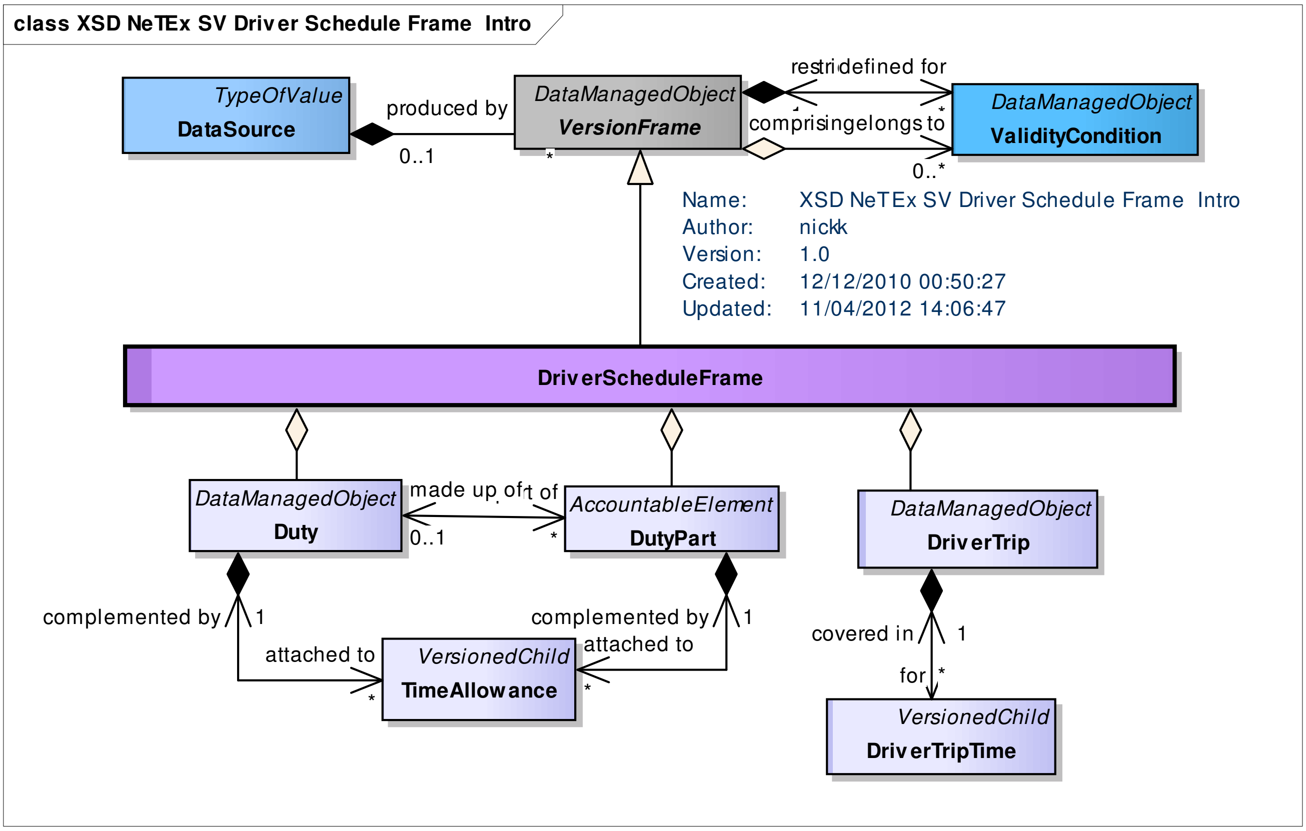


**Figure 173 — Driver Schedule Frame Contents – Physical Model (UML)**

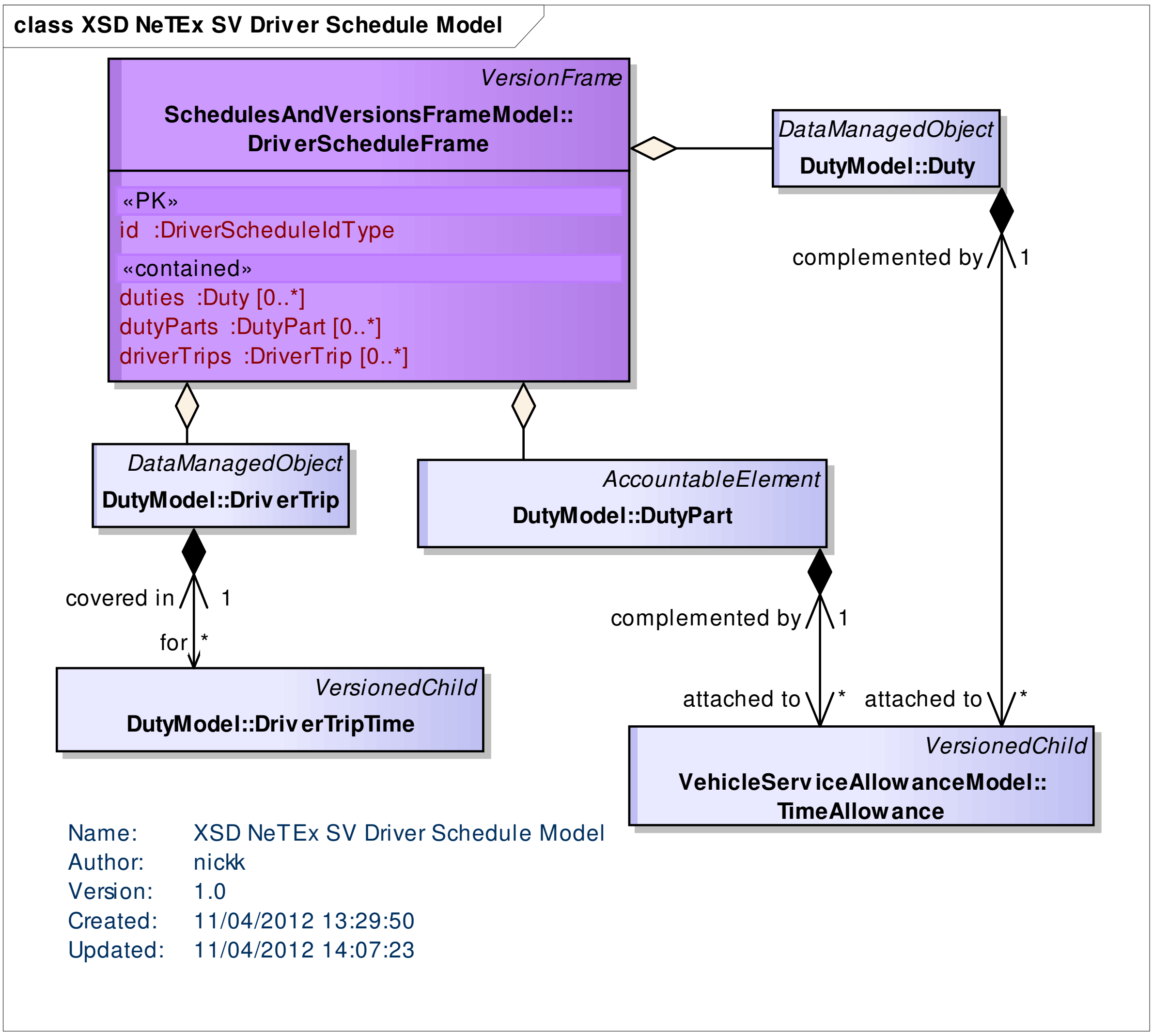


**Figure 174 — Driver Schedule Frame – Physical Model (UML)**

### B.2.1.3 Driver Schedule Frame — XSD and attributes

#### B.2.1.3.1 DriverScheduleFrame – Model Element

A DRIVER SCHEDULE FRAME organizes the set of all DUTies defined for a specific DAY TYPE to which the same VALIDITY CONDITIONs have been assigned.

**Table 99 – *DriverScheduleFrame* – Element**

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *VersionFrame* | *::>* | DRIVER SCHEDULE FRAME inherits from VERSION FRAME. |
| «PK» | ***id*** | *DriverScheduleIdType* | 1:1 | Identifier of DRIVER SCHEDULE FRAME. |
| «cntd» | ***duties*** | *Duty* | 0:* | DUTies in DRIVER SCHEDULE FRAME. |
| «cntd» | ***dutyParts*** | *DutyPart* | 0:* | DUTY PARTs in DRIVER SCHEDULE FRAME. |
| «cntd» | ***driverTrips*** | *DriverTrip* | 0:* | DRIVER TRIPs in DRIVER SCHEDULE FRAME. |

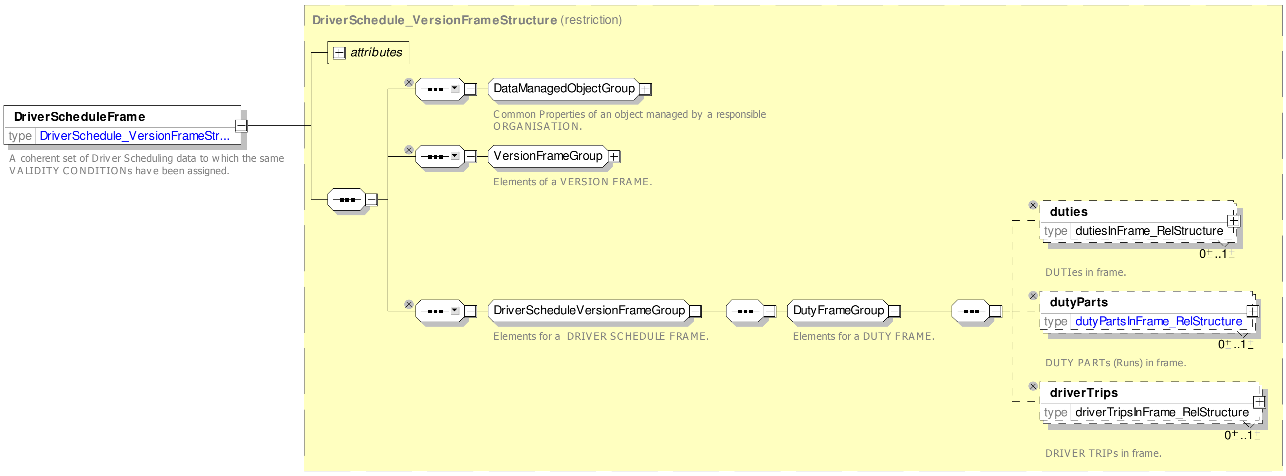

DriverSchedule_VersionFrameStructure (restriction)
attributes
DriverScheduleFrame
type DriverSchedule_VersionFrameStr...
A coherent set of Driver Scheduling data to which the same VALIDITY CONDITIONs have been assigned.
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
VersionFrameGroup
Elements of a VERSION FRAME.
DriverScheduleVersionFrameGroup
Elements for a DRIVER SCHEDULE FRAME.
DutyFrameGroup
Elements for a DUTY FRAME.
duties
type dutiesInFrame_RelStructure
0..1
DUTIes in frame.
dutyParts
type dutyPartsInFrame_RelStructure
0..1
DUTY PARTs (Runs) in frame.
driverTrips
type driverTripsInFrame_RelStructure
0..1
DRIVER TRIPs in frame.

Figure 175 — *DriverScheduleFrame* — XSD

## B.2.2 Duty

### B.2.2.1 DUTY – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

A DUTY describes a day’s work for a logical driver on one DAY TYPE.

An assigned DUTY may be divided into one or two (or sometimes even more) DUTY PARTs, which are each bounded by sign on and sign off. A DUTY consisting of two DUTY PARTs is commonly known as a “split” duty. A DUTY PART is a continuous period during which the driver is under the responsibility of the operator. In many companies, a DUTY PART is therefore a continuous paid period as well.

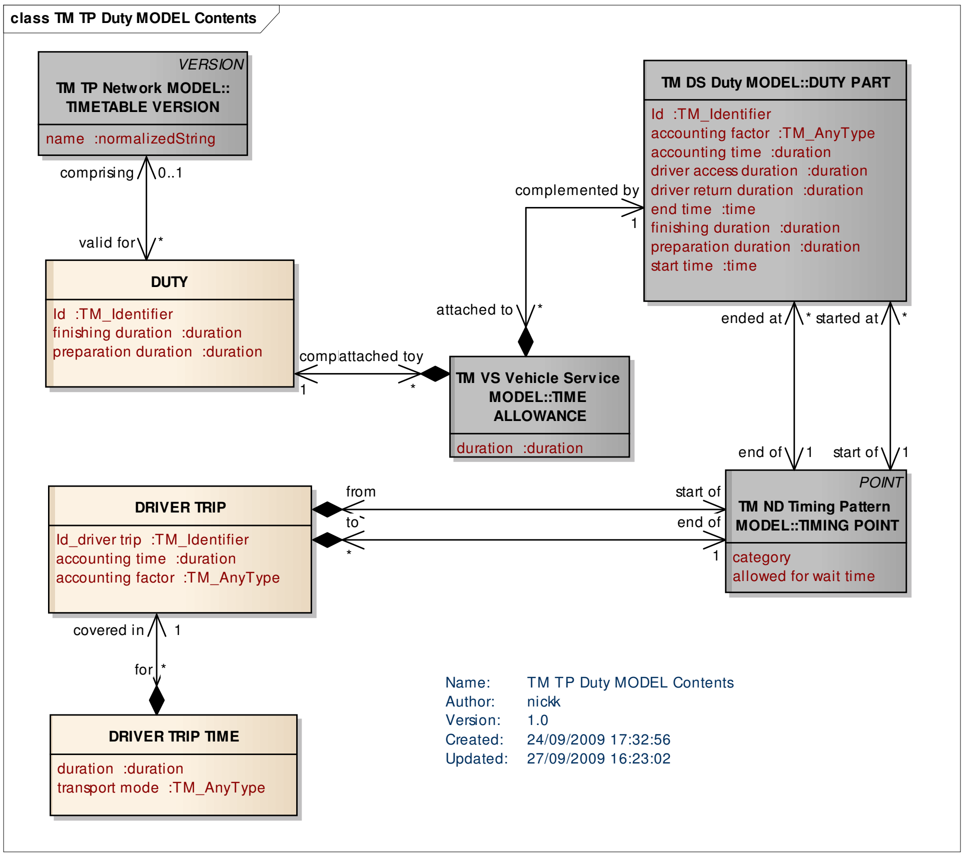

class TM TP Duty MODEL Contents
VERSION
TM TP Network MODEL::
TIMETABLE VERSION
name :normalizedString
comprising
0..1
valid for
*
DUTY
Id :TM_Identifier
finishing duration :duration
preparation duration :duration
comprised of
1
attached to
*
TM VS Vehicle Service
MODEL::TIME
ALLOWANCE
duration :duration
attached to
*
complemented by
1
TM DS Duty MODEL::DUTY PART
Id :TM_Identifier
accounting factor :TM_AnyType
accounting time :duration
driver access duration :duration
driver return duration :duration
end time :time
finishing duration :duration
preparation duration :duration
start time :time
ended at
*
started at
*
end of
1
start of
1
POINT
TM ND Timing Pattern
MODEL::TIMING POINT
category
allowed for wait time
DRIVER TRIP
Id_driver trip :TM_Identifier
accounting time :duration
accounting factor :TM_AnyType
from
to
*
start of
end of
1
covered in
1
for
*
DRIVER TRIP TIME
duration :duration
transport mode :TM_AnyType
Name: TM TP Duty MODEL Contents
Author: nickk
Version: 1.0
Created: 24/09/2009 17:32:56
Updated: 27/09/2009 16:23:02

Figure-176— DUTY — Conceptual MODEL (UML)

### B.2.2.1.1 Duty – Physical Model

The following figure shows the DUTY Physical Model.

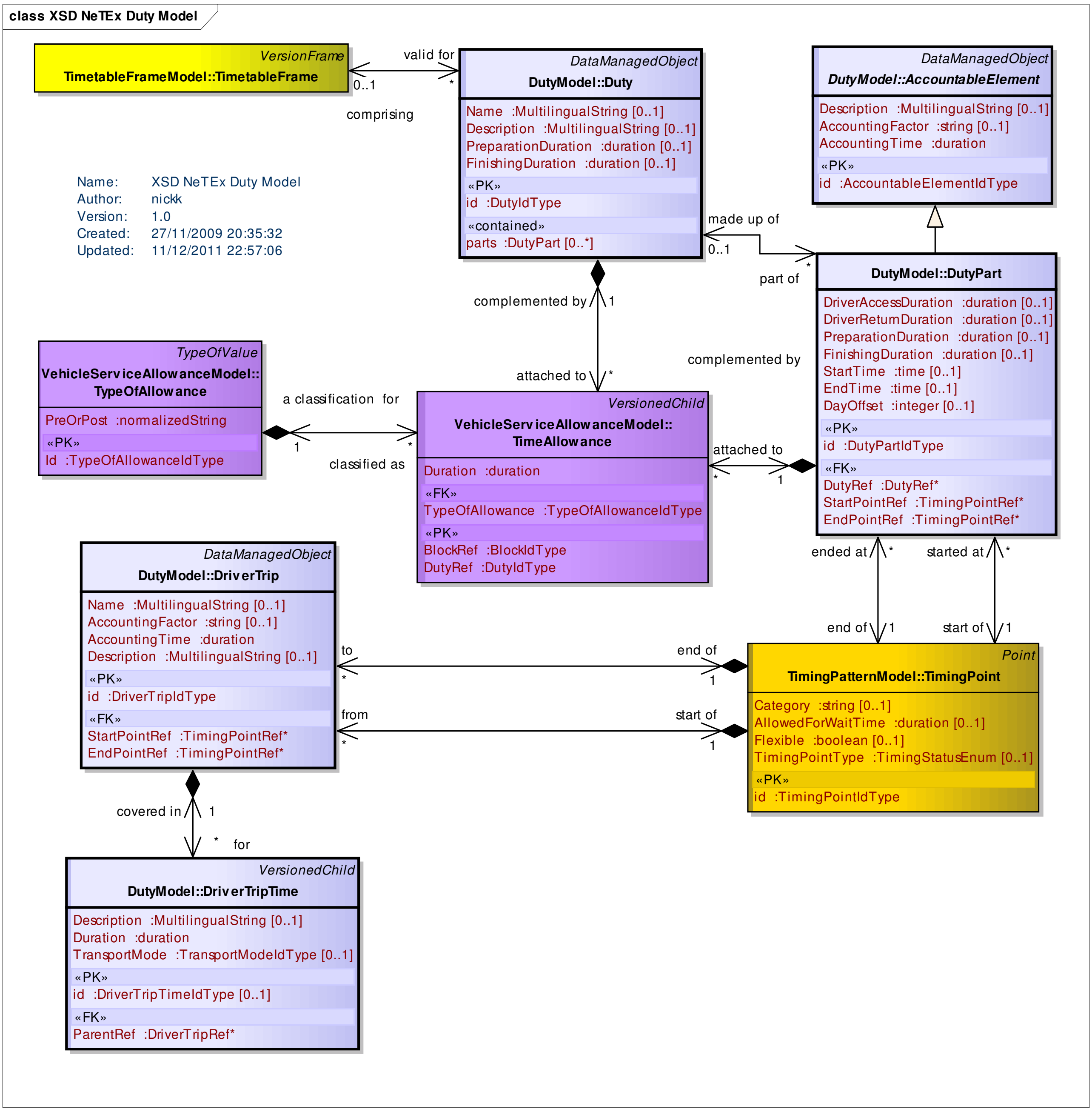

class XSD NeTEx Duty Model
VersionFrame
TimetableFrameModel::TimetableFrame
valid for
0..1
*
comprising
DataManagedObject
DutyModel::Duty
Name :MultilingualString [0..1]
Description :MultilingualString [0..1]
PreparationDuration :duration [0..1]
FinishingDuration :duration [0..1]
«PK»
id :DutyIdType
«contained»
parts :DutyPart [0..*]
DataManagedObject
DutyModel::AccountableElement
Description :MultilingualString [0..1]
AccountingFactor :string [0..1]
AccountingTime :duration
«PK»
id :AccountableElementIdType
Name: XSD NeTEx Duty Model
Author: nickk
Version: 1.0
Created: 27/11/2009 20:35:32
Updated: 11/12/2011 22:57:06
made up of
0..1
part of
*
DutyModel::DutyPart
DriverAccessDuration :duration [0..1]
DriverReturnDuration :duration [0..1]
PreparationDuration :duration [0..1]
FinishingDuration :duration [0..1]
StartTime :time [0..1]
EndTime :time [0..1]
DayOffset :integer [0..1]
«PK»
id :DutyPartIdType
«FK»
DutyRef :DutyRef*
StartPointRef :TimingPointRef*
EndPointRef :TimingPointRef*
complemented by
1
attached to
*
complemented by
TypeOfValue
VehicleServiceAllowanceModel::
TypeOfAllowance
PreOrPost :normalizedString
«PK»
Id :TypeOfAllowanceIdType
a classification for
1
*
classified as
VersionedChild
VehicleServiceAllowanceModel::
TimeAllowance
Duration :duration
«FK»
TypeOfAllowance :TypeOfAllowanceIdType
«PK»
BlockRef :BlockIdType
DutyRef :DutyIdType
attached to
*
1
ended at
*
started at
*
end of
1
start of
1
DataManagedObject
DutyModel::DriverTrip
Name :MultilingualString [0..1]
AccountingFactor :string [0..1]
AccountingTime :duration
Description :MultilingualString [0..1]
«PK»
id :DriverTripIdType
«FK»
StartPointRef :TimingPointRef*
EndPointRef :TimingPointRef*
to
*
end of
1
from
*
start of
1
Point
TimingPatternModel::TimingPoint
Category :string [0..1]
AllowedForWaitTime :duration [0..1]
Flexible :boolean [0..1]
TimingPointType :TimingStatusEnum [0..1]
«PK»
id :TimingPointIdType
covered in
1
*
for
VersionedChild
DutyModel::DriverTripTime
Description :MultilingualString [0..1]
Duration :duration
TransportMode :TransportModeIdType [0..1]
«PK»
id :DriverTripTimeIdType [0..1]
«FK»
ParentRef :DriverTripRef*

Figure 177 — Duty – Physical Model (UML)

### B.2.2.1.2 Duty – Attributes and XSD

#### B.2.2.1.2.1 Duty – Model Element

The work to be performed by a driver on a particular DAY TYPE.

Table 100 – *Duty* – Element

| Classification | Name | Type | cardinality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | DUTY inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *DutyIdType* | 1:1 | Identifier of DUTY. |
| | ***Name*** | *MultilingualString* | 0:1 | Name of DUTY. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of DUTY. |
| | ***Preparation-Duration*** | *xsd:duration* | 0:1 | Time needed to prepare for DUTY. |
| | ***FinishingDuration*** | *xsd:duration* | 0:1 | Time needed to complete DUTY at end. |
| «cntd» | ***parts*** | *DutyPart* | 0:* | List of DUTY Parts. |

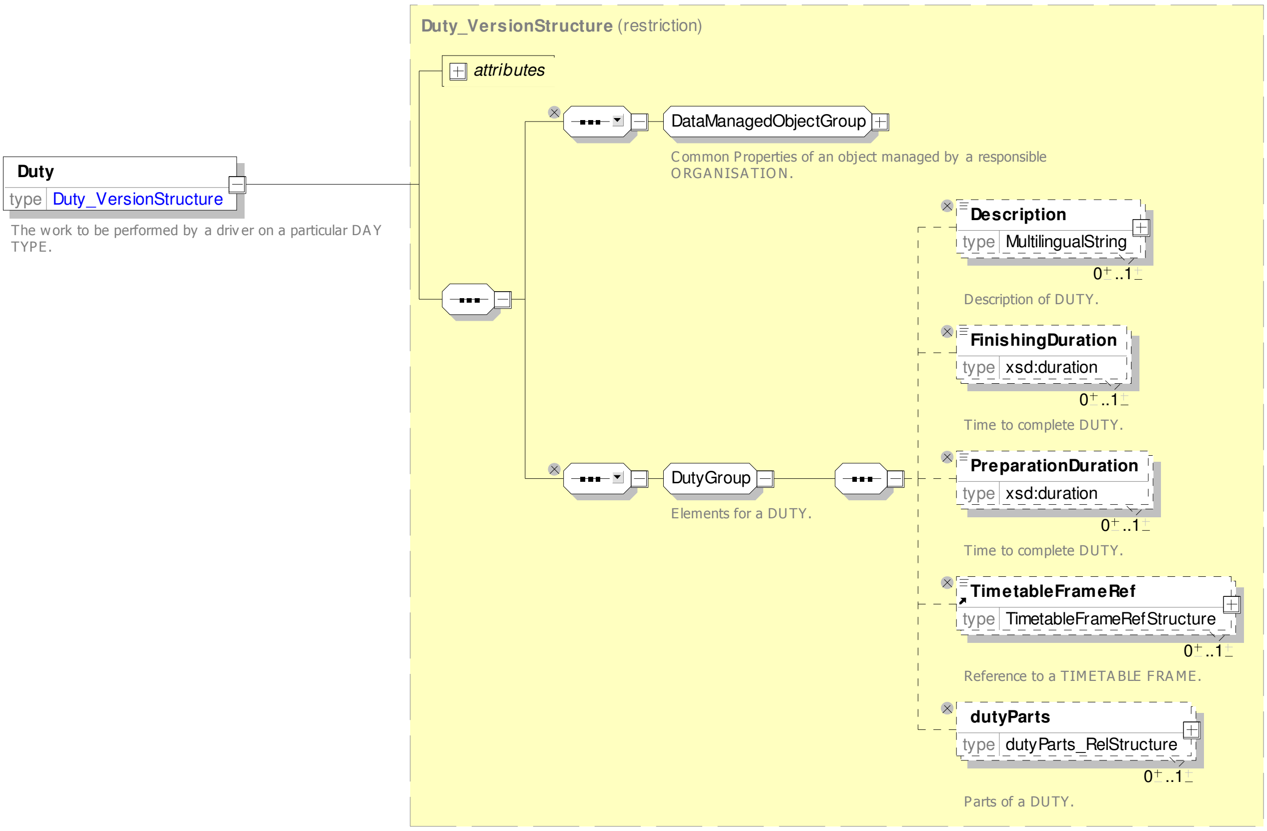


Figure 178 — *Duty* — XSD

### B.2.2.1.2.2 AccountableElement – Model Element

A period of a driver's DUTY during which he or she is continuously working without a BREAK. PAUSEs during which he or she remains responsible for the vehicle may be included.

**Table 101 – *AccountableElement* – Element**

| Classification | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | ACCOUNTABLE ELEMENT inherits from DATA MANAGED OBJECT. |
| «PK» | *id* | *AccountableElement-IdType* | 1:1 | Identifier of ACCOUNTABLE ELEMENT. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of ACCOUNTING ELEMENT. |
| | ***AccountingFactor*** | *xsd:normalizedString* | 0:1 | Accounting code to use on ACCOUNTABLE ELEMENT. |
| | ***AccountingTime*** | *xsd:duration* | 1:1 | Accountable time to use on ACCOUNTABLE ELEMENT. |

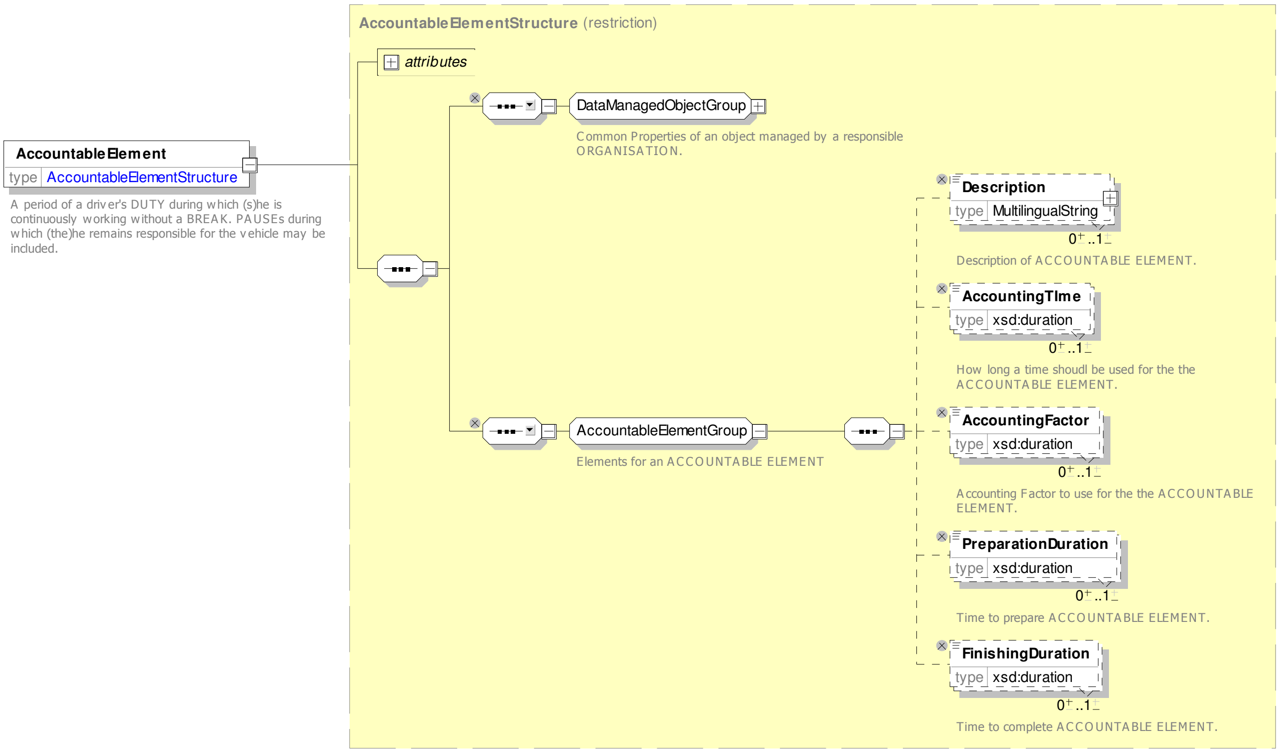

AccountableElementStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
AccountableElement
type AccountableElementStructure
A period of a driver's DUTY during which (s)he is continuously working without a BREAK. PAUSEs during which (the)he remains responsible for the vehicle may be included.
Description
type MultilingualString
0..1
Description of ACCOUNTABLE ELEMENT.
AccountingTIme
type xsd:duration
0..1
How long a time shoudl be used for the the ACCOUNTABLE ELEMENT.
AccountableElementGroup
Elements for an ACCOUNTABLE ELEMENT
AccountingFactor
type xsd:duration
0..1
Accounting Factor to use for the the ACCOUNTABLE ELEMENT.
PreparationDuration
type xsd:duration
0..1
Time to prepare ACCOUNTABLE ELEMENT.
FinishingDuration
type xsd:duration
0..1
Time to complete ACCOUNTABLE ELEMENT.

**Figure 179 — *AccountableElement* — XSD**

### B.2.2.1.2.3 DutyPart – Model Element

A continuous part of a driver DUTY during which (s)he is under the management of the company. A DUTY PART may include BREAKs.

Table 102 – *DutyPart* – Element

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| ::> | ::> | *DataManagedObject* | ::> | DUTY PART inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *DutyPartIdType* | 1:1 | Identifier of DUTY PART. |
| «FK» | ***DutyRef*** | *DutyRef* | 1:1 | Reference to of DUTY of which this is part. |
| | ***DriverAccessDuration*** | *xsd:duration* | 0:1 | Time needed for driver access to DUTY PART. |
| | ***DriverReturnDuration*** | *xsd:duration* | 0:1 | Time needed for driver return from DUTY PART. |
| | ***PreparationDuration*** | *xsd:duration* | 0:1 | Time needed to prepare for DUTY PART. |
| | ***FinishingDuration*** | *xsd:duration* | 0:1 | Time needed to complete DUTY PART. |
| | ***StartTime*** | *xsd:time* | 0:1 | Start time of DUTY PART. |
| «FK» | ***StartPointRef*** | *TimingPointRef* | 1:1 | Start POINT of DUTY PART. |
| | ***EndTime*** | *xsd:time* | 0:1 | End time of DUTY PART. |
| «FK» | ***EndPointRef*** | *TimingPointRef* | 1:1 | End POINT of DUTY PART. |
| | ***DayOffset*** | *xsd:integer* | 0:1 | Day offset of end Time of DUTY PART from atart of JOURNEY. |

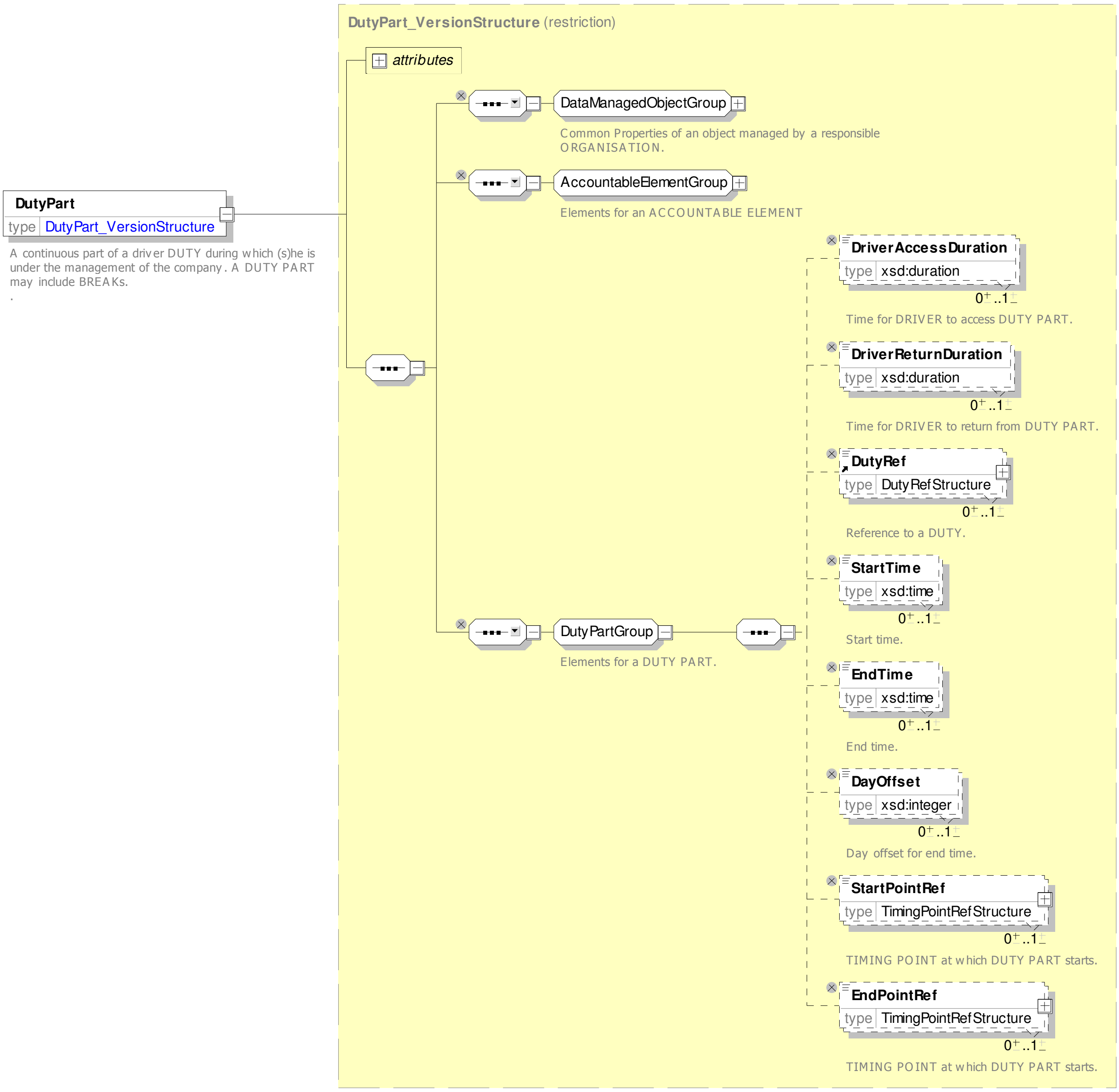

DutyPart_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
AccountableElementGroup
Elements for an ACCOUNTABLE ELEMENT
DutyPart
type DutyPart_VersionStructure
A continuous part of a driver DUTY during which (s)he is under the management of the company. A DUTY PART may include BREAKs.
DriverAccessDuration
type xsd:duration
0..1
Time for DRIVER to access DUTY PART.
DriverReturnDuration
type xsd:duration
0..1
Time for DRIVER to return from DUTY PART.
DutyRef
type DutyRefStructure
0..1
Reference to a DUTY.
StartTime
type xsd:time
0..1
Start time.
DutyPartGroup
Elements for a DUTY PART.
EndTime
type xsd:time
0..1
End time.
DayOffset
type xsd:integer
0..1
Day offset for end time.
StartPointRef
type TimingPointRefStructure
0..1
TIMING POINT at which DUTY PART starts.
EndPointRef
type TimingPointRefStructure
0..1
TIMING POINT at which DUTY PART starts.

**Figure 180 — *DutyPart* — XSD**

### B.2.2.1.2.4 DriverTrip – Model Element

A planned non-driving movement of a driver within a DUTY PART. This may be necessary to reach the first SPELL in a STRETCH, between two SPELLs or after the last SPELL in a STRETCH. It may be entirely on foot or may use a VEHICLE JOURNEY on a vehicle driven by another driver.

**Table 103 – *DriverTrip* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | DRIVER TRIP inherits from DATA MANAGED OBJECT. |
| «PK» | ***id*** | *DriverTripIdType* | 1:1 | Identifier of DRIVER TRIP. |

| | | | | |
|---|---|---|---|---|
| | ***Name*** | *MultilingualString* | 0:1 | Name of DRIVER TRIP. |
| | ***AccountingFactor*** | *xsd:string* | 0:1 | Accounting code to use for DRIVER TRIP. |
| | ***AccountingTime*** | *xsd:duration* | 1:1 | Accountable time to use for DRIVER TRIP. |
| | ***Description*** | *MultilingualString* | 0:1 | Description of DRIVER TRIP. |
| «FK» | ***StartPointRef*** | *TimingPointRef* | 1:1 | Start POINT of DRIVER TRIP. |
| «FK» | ***EndPointRef*** | *TimingPointRef* | 1:1 | End POINT of DRIVER TRIP. |

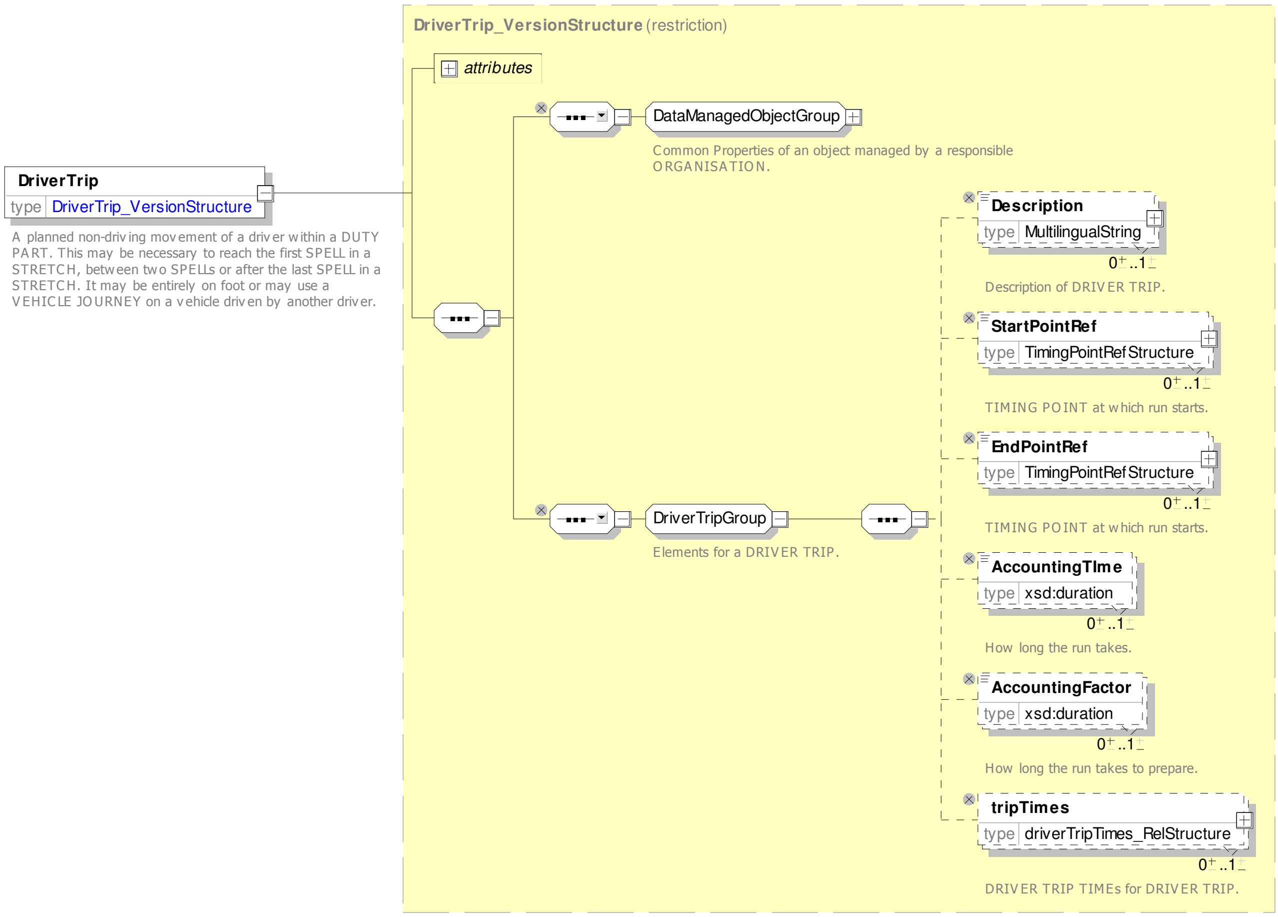

DriverTrip_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
DriverTrip
type DriverTrip_VersionStructure
A planned non-driving movement of a driver within a DUTY PART. This may be necessary to reach the first SPELL in a STRETCH, between two SPELLs or after the last SPELL in a STRETCH. It may be entirely on foot or may use a VEHICLE JOURNEY on a vehicle driven by another driver.
Description
type MultilingualString
0..1
Description of DRIVER TRIP.
StartPointRef
type TimingPointRefStructure
0..1
TIMING POINT at which run starts.
EndPointRef
type TimingPointRefStructure
0..1
TIMING POINT at which run starts.
DriverTripGroup
Elements for a DRIVER TRIP.
AccountingTIme
type xsd:duration
0..1
How long the run takes.
AccountingFactor
type xsd:duration
0..1
How long the run takes to prepare.
tripTimes
type driverTripTimes_RelStructure
0..1
DRIVER TRIP TIMEs for DRIVER TRIP.

**Figure 181 — *DriverTrip* — XSD**

### B.2.2.1.2.5 DriverTripTime – Model Element

The time allowed for a driver to cover a particular DRIVER TRIP during a specified TIME BAND.

**Table 104 – *DriverTripTime* – Element**

| Classifi cation | Name | Type | Cardin-ality | Description |
|---|---|---|---|---|
| *::>* | *::>* | *DataManagedObject* | *::>* | DRIVER TRIP TIME inherits from DATA MANAGED OBJECT. |

|  |  |  |  |  |
|---|---|---|---|---|
| «PK» | ***id*** | *DriverTripTimeIdType* | 0:1 | Identifier of DRIVER TRIP TIME. |
| «FK» | ***DriverTripRef*** | *DriverTripRef* | 1:1 | DRIVER TRIP to which this DRIVER TRIP TIME belongs. |
|  | ***Description*** | *MultilingualString* | 0:1 | Description of DRIVER TRIP TIME. |
|  | ***Duration*** | *xsd:duration* | 1:1 | Duration of DRIVER TRIP TIME. |
|  | ***TransportMode*** | *TransportModeEnum* | 0:1 | Transport mode of DRIVER TRIP TIME. |

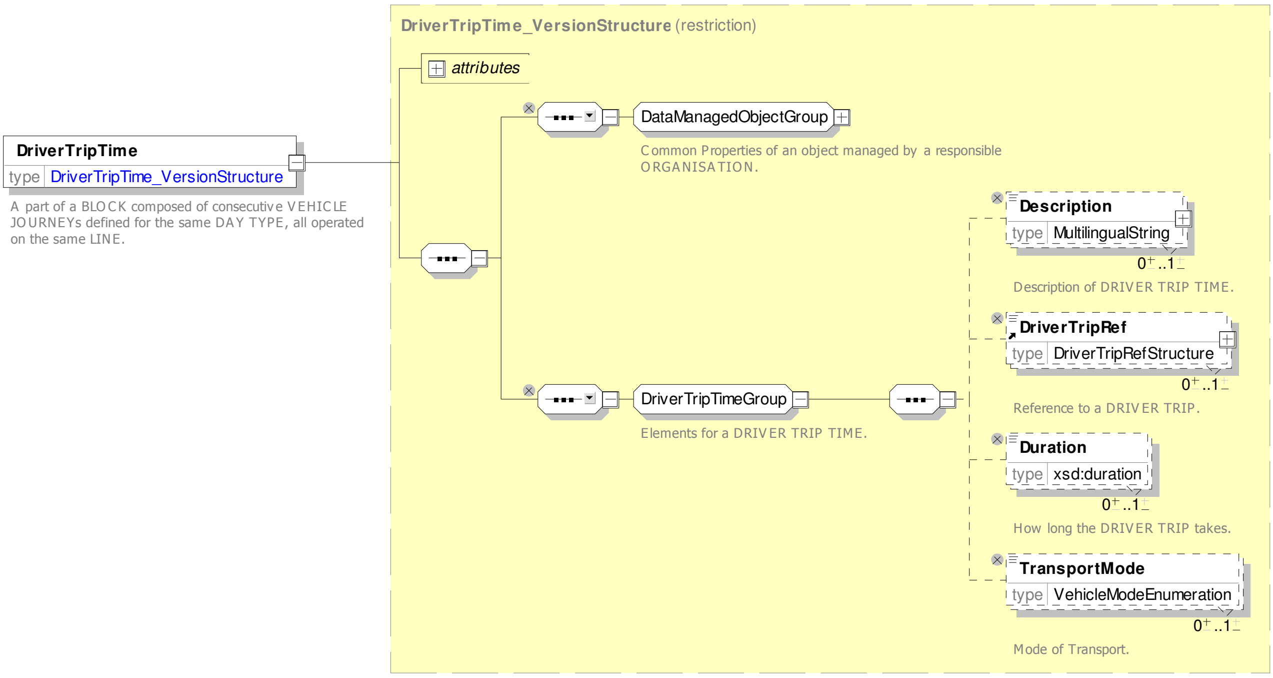

DriverTripTime_VersionStructure (restriction)
attributes
DataManagedObjectGroup
Common Properties of an object managed by a responsible ORGANISATION.
DriverTripTime
type DriverTripTime_VersionStructure
A part of a BLOCK composed of consecutive VEHICLE JOURNEYs defined for the same DAY TYPE, all operated on the same LINE.
Description
type MultilingualString
0..1
Description of DRIVER TRIP TIME.
DriverTripRef
type DriverTripRefStructure
0..1
Reference to a DRIVER TRIP.
DriverTripTimeGroup
Elements for a DRIVER TRIP TIME.
Duration
type xsd:duration
0..1
How long the DRIVER TRIP takes.
TransportMode
type VehicleModeEnumeration
0..1
Mode of Transport.

**Figure 182 — *DriverTripTime* — XSD**

## B.2.3 Duty Stretch

### B.2.3.1 DUTY STRETCH – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

As a DUTY PART may last a relatively long time, they often include breaks planned for driver resting. Each DUTY PART is composed of one or several STRETCHes, which are continuous period of work not interrupted by any BREAK. A BREAK should therefore follow a STRETCH and is included in the corresponding DUTY PART.

Each STRETCH is composed of one or several SPELLs. A SPELL describes a period during which the same main activity is to be performed (e.g. driving a vehicle, stand-by period, etc.). SPELLs are subdivided into DRIVING SPELLs and NON DRIVING SPELLs.

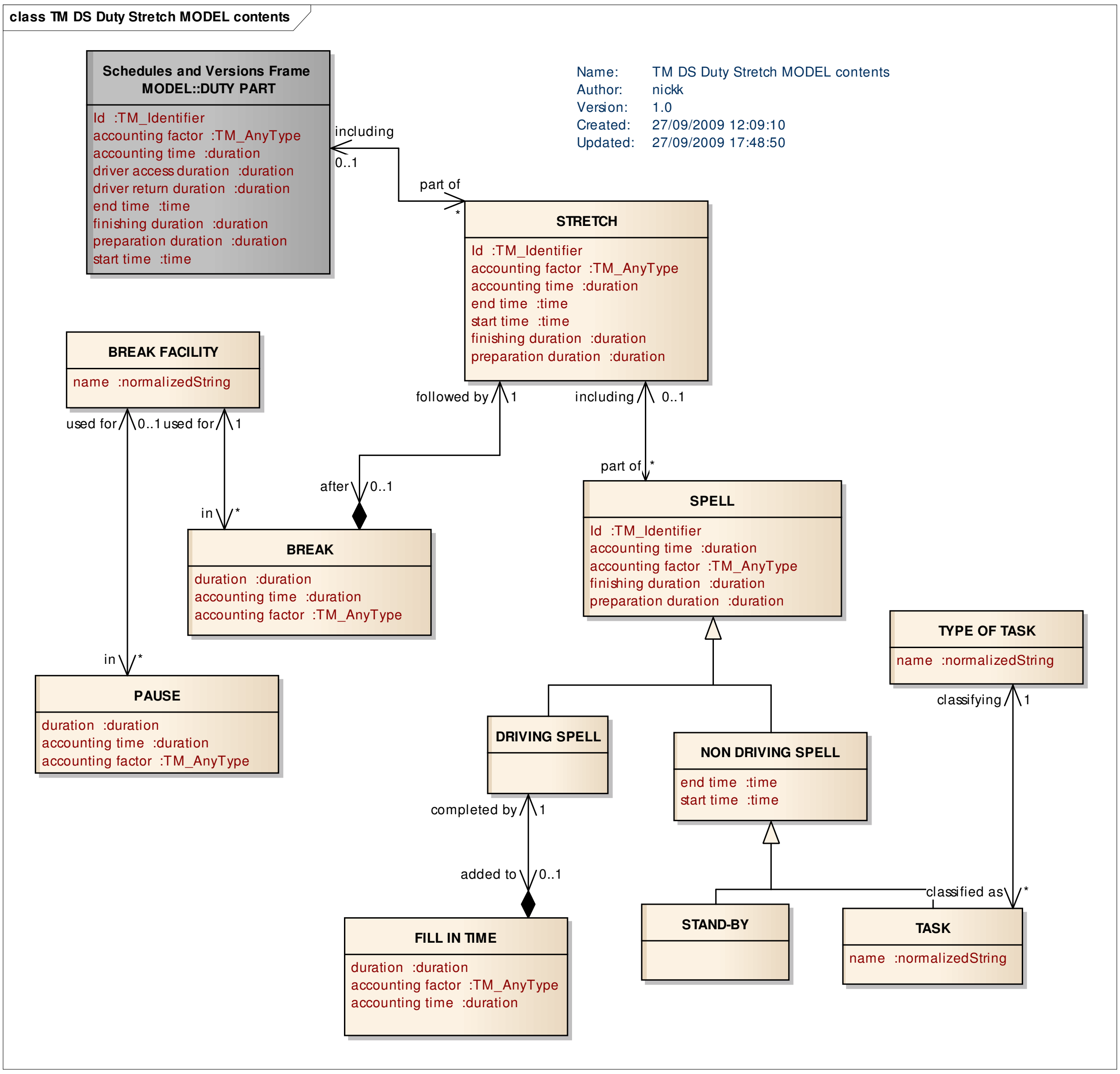

class TM DS Duty Stretch MODEL contents
Name: TM DS Duty Stretch MODEL contents
Author: nickk
Version: 1.0
Created: 27/09/2009 12:09:10
Updated: 27/09/2009 17:48:50
Schedules and Versions Frame MODEL::DUTY PART
Id :TM_Identifier
accounting factor :TM_AnyType
accounting time :duration
driver access duration :duration
driver return duration :duration
end time :time
finishing duration :duration
preparation duration :duration
start time :time
including
0..1
part of
*
STRETCH
Id :TM_Identifier
accounting factor :TM_AnyType
accounting time :duration
end time :time
start time :time
finishing duration :duration
preparation duration :duration
BREAK FACILITY
name :normalizedString
used for
0..1
used for
1
followed by
1
including
0..1
after
0..1
part of
*
in
*
BREAK
duration :duration
accounting time :duration
accounting factor :TM_AnyType
SPELL
Id :TM_Identifier
accounting time :duration
accounting factor :TM_AnyType
finishing duration :duration
preparation duration :duration
TYPE OF TASK
name :normalizedString
in
*
PAUSE
duration :duration
accounting time :duration
accounting factor :TM_AnyType
classifying
1
DRIVING SPELL
NON DRIVING SPELL
end time :time
start time :time
completed by
1
added to
0..1
classified as
*
FILL IN TIME
duration :duration
accounting factor :TM_AnyType
accounting time :duration
STAND-BY
TASK
name :normalizedString

Figure 183 — STRRTCH and SPELL— Conceptual MODEL (UML)

### B.2.3.2 Duty Stretch – Physical Model

The following figure shows the DUTY STRETCH Physical Model.

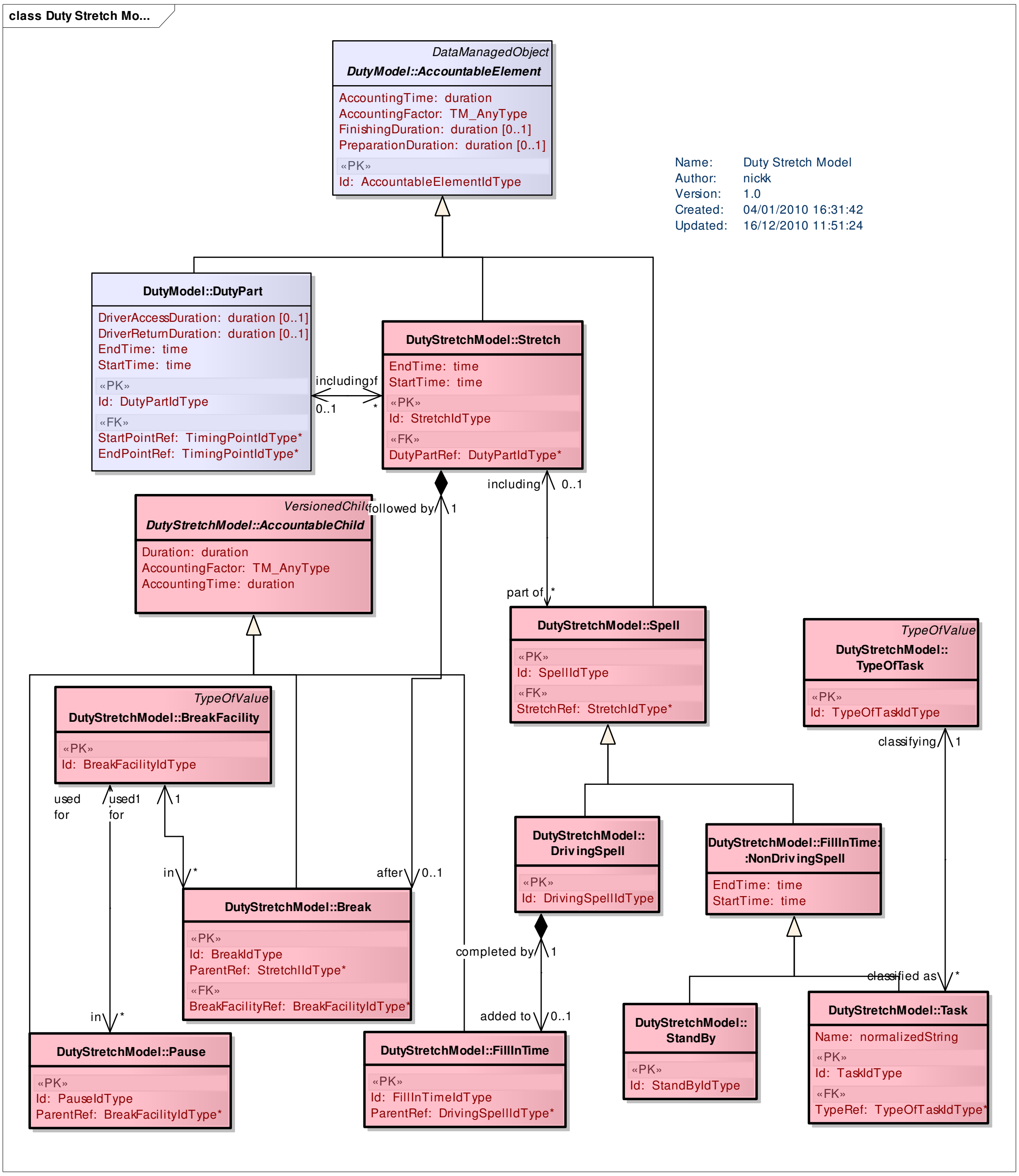


**Figure 184 — Duty Stretch – Physical Model (UML)**

# Bibliography

[1] CEN/TS 15531-1, *Public transport - Service interface for real-time information relating to public transport operations - Part 1: Context and framework*

[2] CEN/TS 15531-2, *Public transport - Service interface for real-time information relating to public transport operations - Part 2: Communications infrastructure*

[3] CEN/TS 15531-3, *Public transport - Service interface for real-time information relating to public transport operations - Part 3: Functional service interfaces*

[4] CEN/TS 15531-4, *Public transport - Service interface for real-time information relating to public transport operations - Part 4: Functional service interfaces: Facility Monitoring*

[5] CEN/TS 15531-5, *Public transport - Service interface for real-time information relating to public transport operations - Part 5: Functional service interfaces - Situation Exchange*

[6] EN 12896, *Road transport and traffic telematics - Public transport - Reference data model*

[7] EN 28701, *Road transport and traffic telematics - Public transport - Identification of fixed objects in public transport*

[8] ISO-8601:2000, *Data elements and interchange formats – Information interchange – Representation of dates and times.*

[9] ISO-639/IETF 1766, *Tags for the Identification of Languages.*

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

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

[12] 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.

[13] UIC recommendations and leaflets.

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