

CEN/TC 278

Date: 2017-03

TC 278 TS 16614-2

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:

DRAFT for NeTEx v1.1

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

C:\Data\NeTEx\Part2-2017\Part 2 - Word document\prCEN TS 16614-2 FV (E)-2018-Final1.docx STD Version 2.5a

Contents

Page

Foreword	4
Introduction	5
1 Scope.....	6
2 Normative references	6
2.1 General	6
2.2 Transport modes	6
2.3 Compatibility with existing standards and recommendations.....	6
3 Terms and definitions	7
4 Symbols and abbreviations	7
5 Use Cases for Journey & Journey Time Exchange	7
6 Generic Physical Model and XSD mapping rules	7
7 Timing Information – Conceptual and physical data model	7
7.1 Introduction	7
7.1.1 Journey and Journey Times – Model dependencies.....	8
7.2 Journey and Journey Times	16
7.2.1 Vehicle Journey	16
7.2.2 Service Journey.....	36
7.2.3 Time Demand Times	55
7.2.4 Journey Timing.....	64
7.2.5 Journey Pattern Times	72
7.2.6 Vehicle Journey Times	80
7.2.7 Interchange	97
7.2.8 Interchange Rule	117
7.2.9 Coupled Journey	125
7.2.10 Flexible Service	147
7.2.11 Journey Accounting	156
7.2.12 Dated Journey	160
7.2.13 Passing Times	166
7.2.14 Call.....	174
7.2.15 Dated Call.....	190
8 Vehicle Scheduling	192
8.1 Vehicle Scheduling – Model dependencies.....	192
8.2 Vehicle Scheduling	193
8.2.1 Vehicle Schedule Frame	193
8.2.2 Vehicle Service	196
8.2.3 Train Service.....	211
Annex A (informative) Monitoring & Control.....	215
A.1 Introduction	215
A.2 Monitoring & Control	215
A.2.1 Monitored Vehicle Journey	215
A.2.2 Dated Passing Times – Physical Model.....	222
Annex B (informative) Driver Scheduling.....	227
B.1 Introduction	227
B.2 Driver Scheduling.....	227
B.2.1 Driver Schedule Frame	227
B.2.2 Duty	230
B.2.3 Duty Stretch	237

Annex C (informative) Changes in NeTeX Version 1.1 240
C.1 Introduction..... 240
C.2 General Changes 240
C.3 List of changes 240
Bibliography..... 242

Foreword

This document (TC 278 TS 16614-2) 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 <http://netex-cen.eu> , along with available guidance on its use, example XML files, and case studies of national and local deployments.

This document is highly technical, and 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; (iii) 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

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*

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

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

CEN/TS 15531-4, *Public transport - Service interface for real-time information relating to public transport operations - Part 4: Functional service interfaces: Facility Monitoring*

CEN/TS 15531-5, *Public transport - Service interface for real-time information relating to public transport operations - Part 5: Functional service interfaces - Situation Exchange*

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)

2.1 General

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

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 adaptations 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.

2.2 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.

2.3 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.

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 Introduction

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

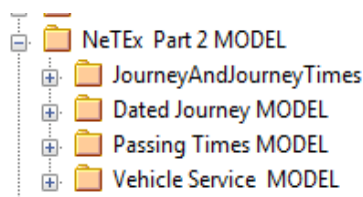


Figure 1 – NeTEx Part 2 main packages

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

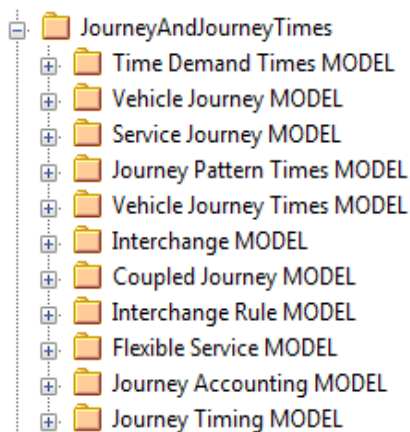


Figure 2 – JourneyAndJourneyTimes packages

- 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 information 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: additionally describes demand responsive transport services.
- JOURNEY ACCOUNTING: assigns a cost basis for journeys.

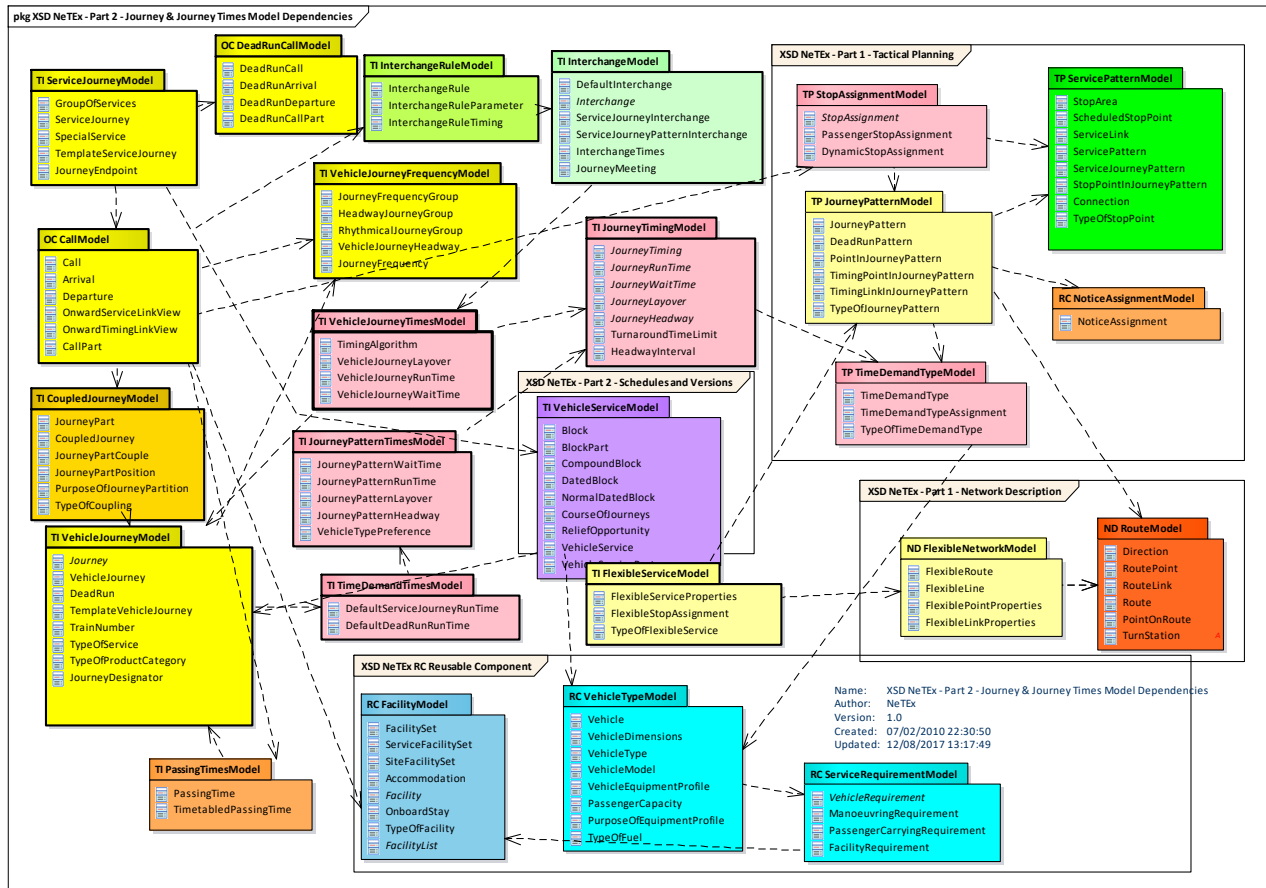


Figure 3 — Journey – Model Dependencies

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.

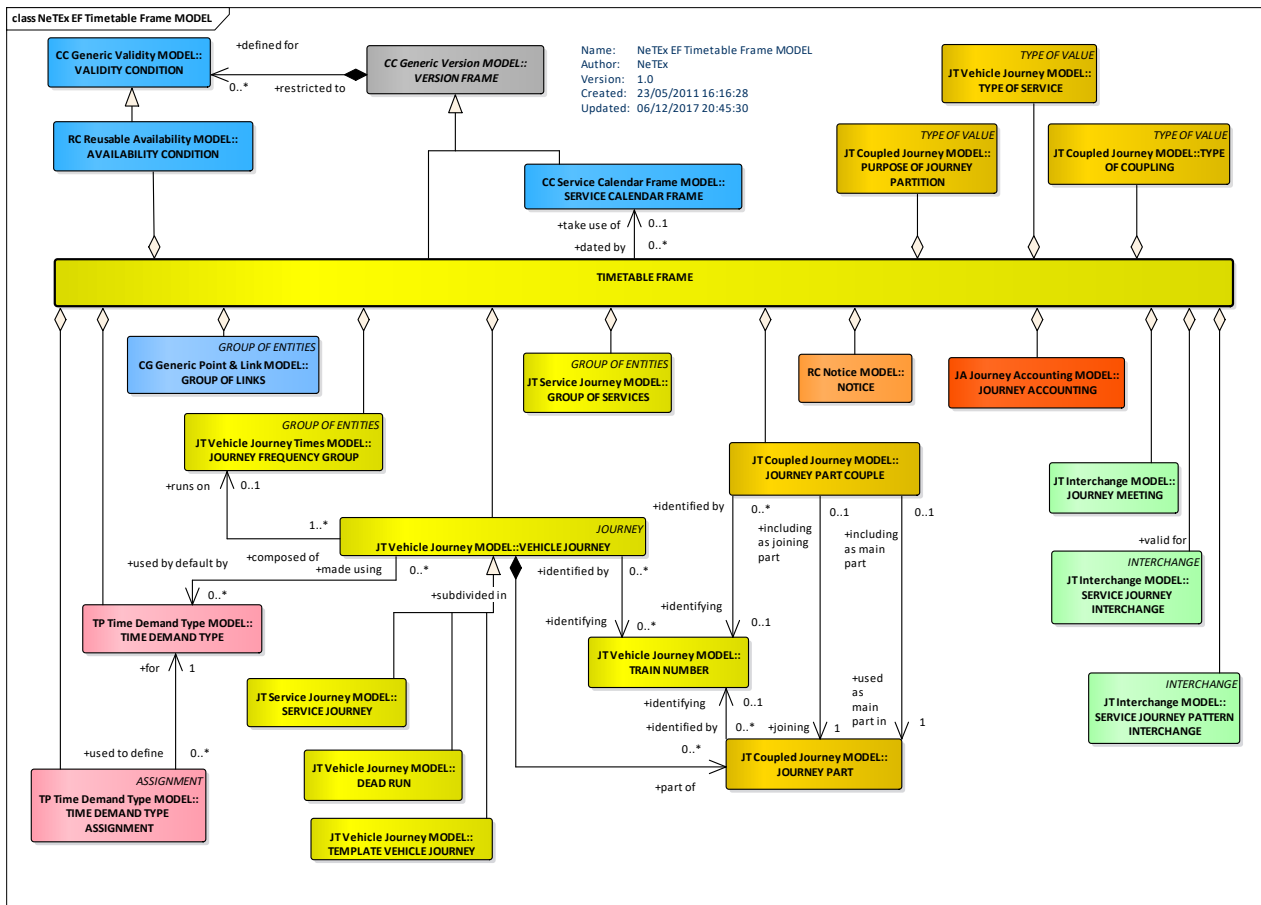


Figure 4 — Timetable Frame – Conceptual MODEL (UML)

7.1.1.1.2 Timetable Frame – Physical Model

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

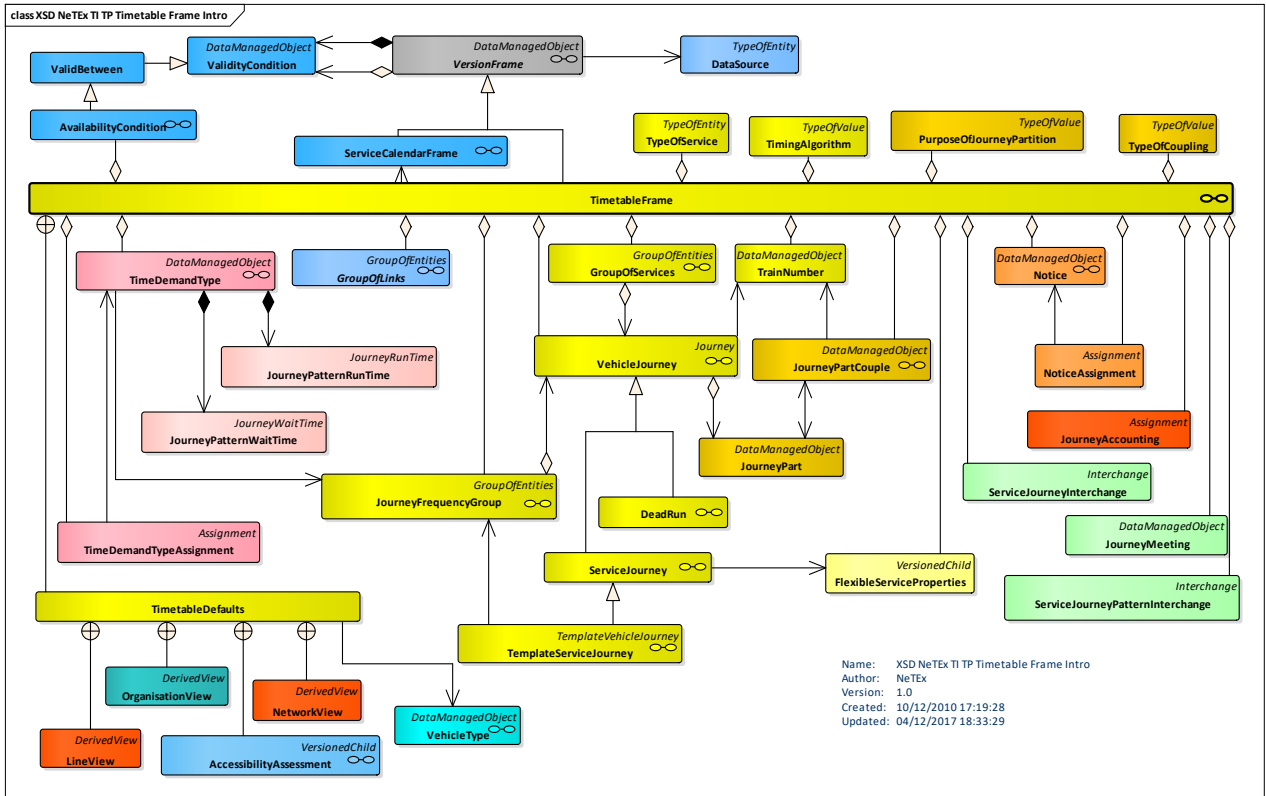


Figure 5 — Timetable Frame Contents – Physical Model (UML)

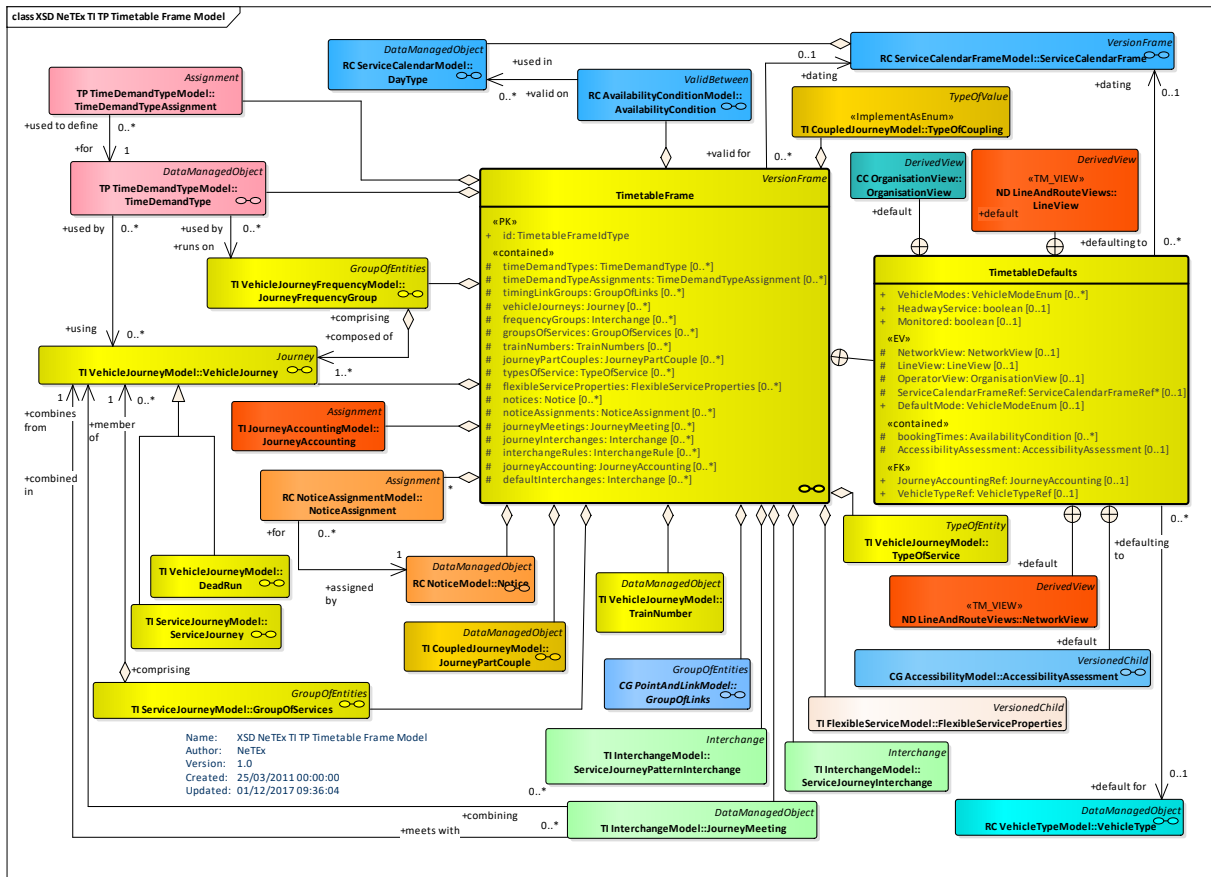


Figure 6 — Timetable Frame – Physical Model Detail (UML)

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
::>	::>	<i>VersionFrame</i>	::>	TIMETABLE FRAME inherits from VERSION FRAME.
«PK»	id	<i>TimetableFrameIdType</i>	1:1	Identifier of TIMETABLE FRAME.
	VehicleModes	<i>VehicleModeEnum</i>	0:*	Reference to vehicle transport MODEs TIMETABLE
	HeadwayService	<i>xsd:boolean</i>	0:1	Whether services of TIMETABLE are operated a headway services.
	Monitored	<i>xsd:boolean</i>	0:1	Whether services of TIMETABLE are monitored in real time.
«EV»	NetworkView	<i>NetworkView</i>	0:1	Reference to default NETWORK for TIMETABLE and derived values of NETWORK.
«EV»	LineView	<i>LineView</i>	0:1	Reference to default LINE for TIMETABLE and derived values of LINE.
«EV»	OperatorView	<i>OrganisationView</i>	0:1	Reference to default OPERATOR for TIMETABLE and derived values of OPERATOR.
«EV»	ServiceCalendarFrameRef	<i>ServiceCalendar-FrameRef</i>	0:1	Reference to default Service CALENDAR for TIMETABLE.
«EV»	DefaultMode	<i>VehicleModeEnum</i>	0:1	Reference to default Transport MODE for TIMETABLE.
«cntd»	JourneyAccountingRef	<i>JourneyAccountingRef</i>	0:1*	Default JOURNEY ACCOUNTING values for JOURNEYS in frame.
«cntd»	bookingTimes	<i>AvailabilityCondition</i>	0:*	Times at which bookings can be made for the services in the Timetable.
	AccessibilityAssessment	<i>AccessibilityAssessment</i>	0:1	Default ACCESSIBILITY ASSESSMENT to assume for journeys in frame. +v1.1

	VehicleTypeRef	VehicleTypeRef	0:1	Default ACCESSIBILITY ASSESSMENT to assume for journeys in frame. +V1.1
«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 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»	journeyAccountings	JourneyAccounting	0:*	Default JOURNEY ACCOUNTING values for JOURNEYs in frame.

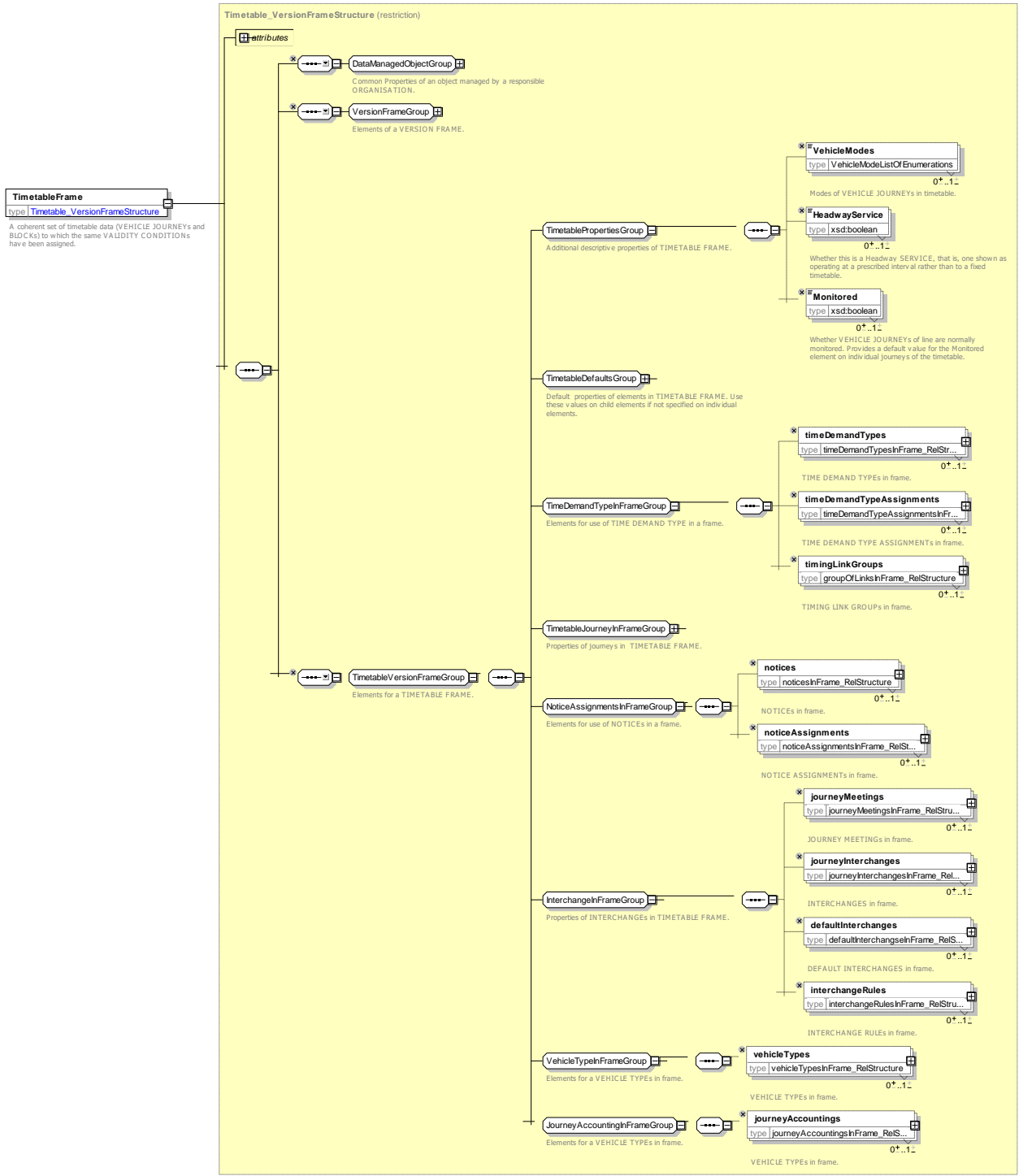


Figure 7 — TimetableFrame — XSD

TimetableFrameDefaultsGroup - Group



Figure 8 — *TimetableFrameDefaultsGroup* — XSD

TimetableJourneyInFrameGroup - Group

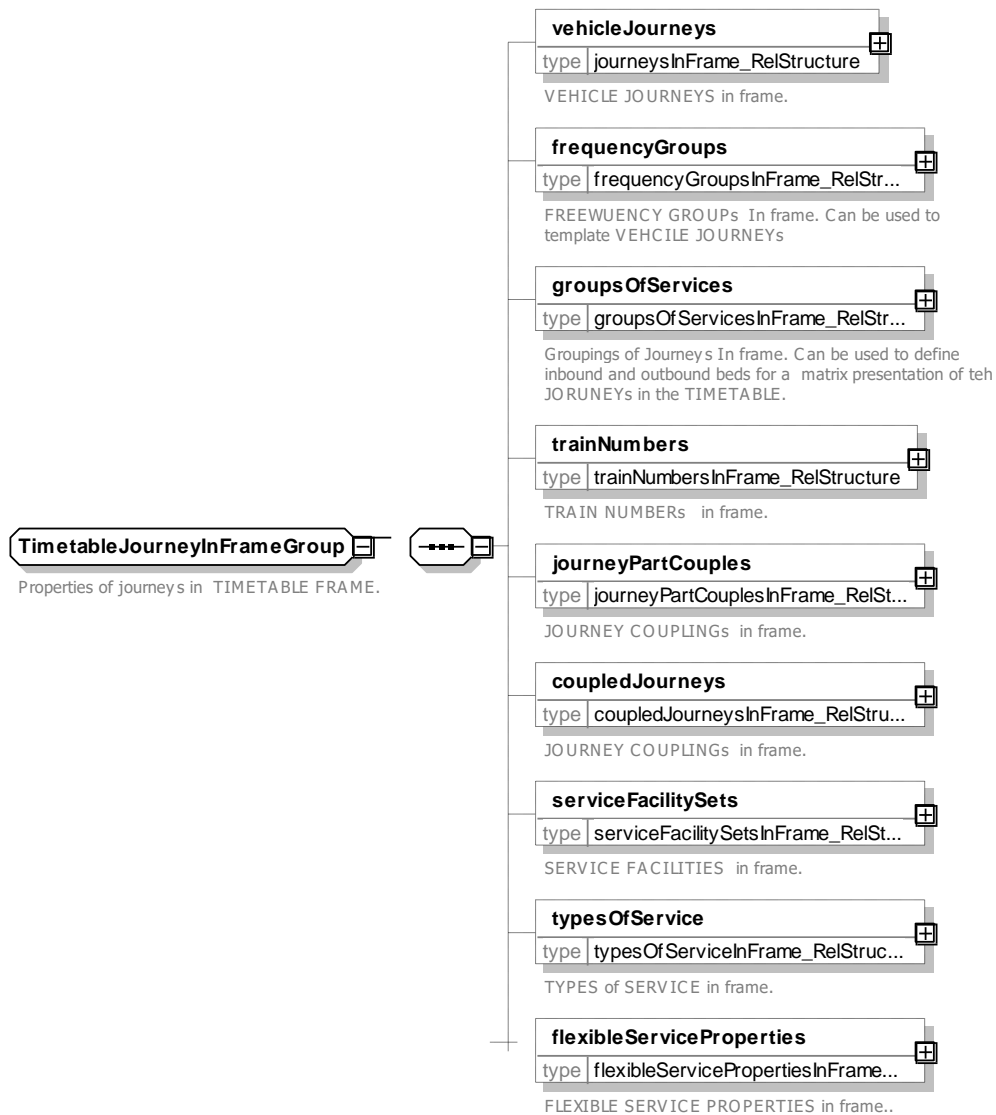


Figure 9 — TimetableJourneyInFrameGroup — XSD

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.

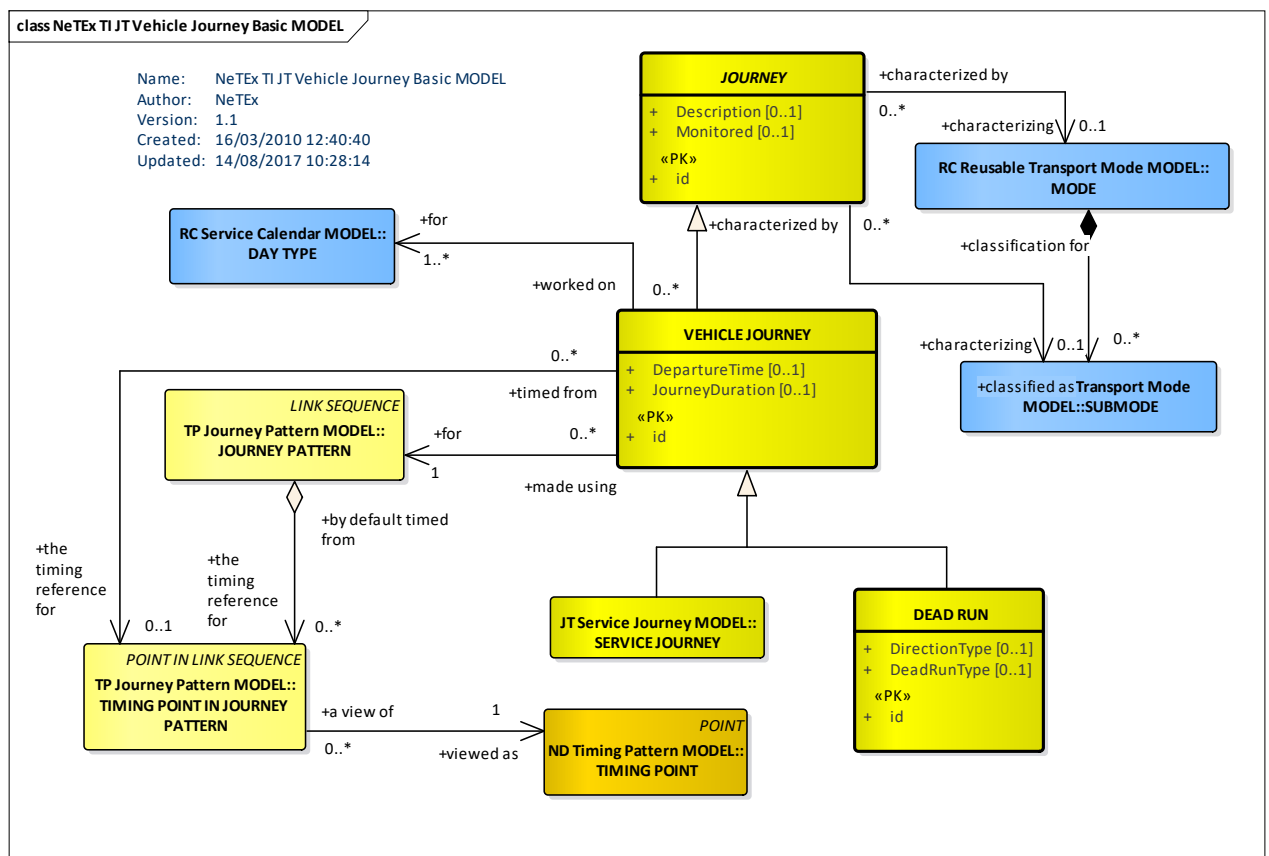


Figure 10 — Basic Vehicle Journey – Conceptual MODEL (UML)

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.

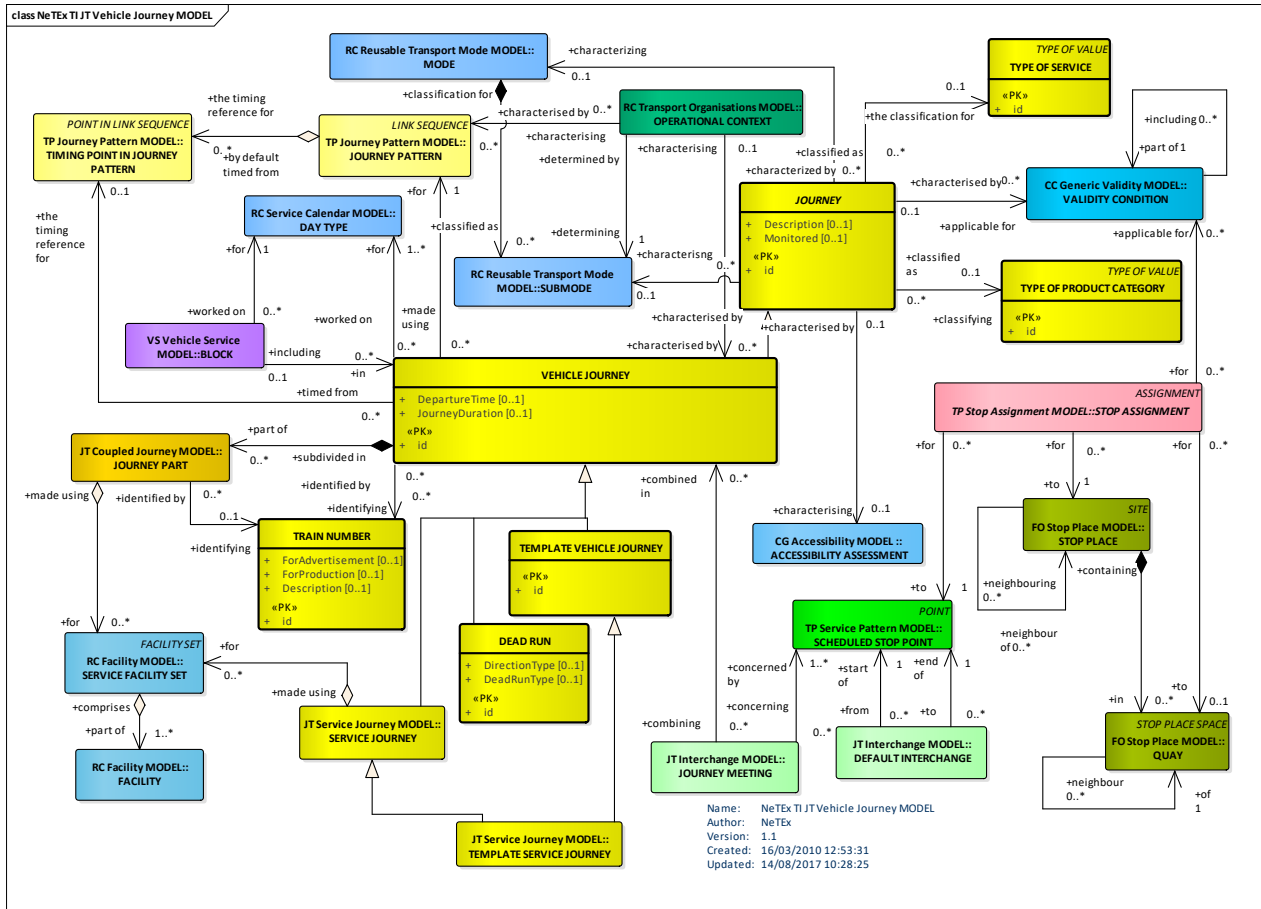


Figure 11 — Vehicle Journey – Conceptual MODEL (UML)

7.2.1.1.3 Vehicle Journey – Physical Model

7.2.1.1.4 Journey – Physical Model

The JOURNEY physical model describes common properties of all types of JOURNEY.

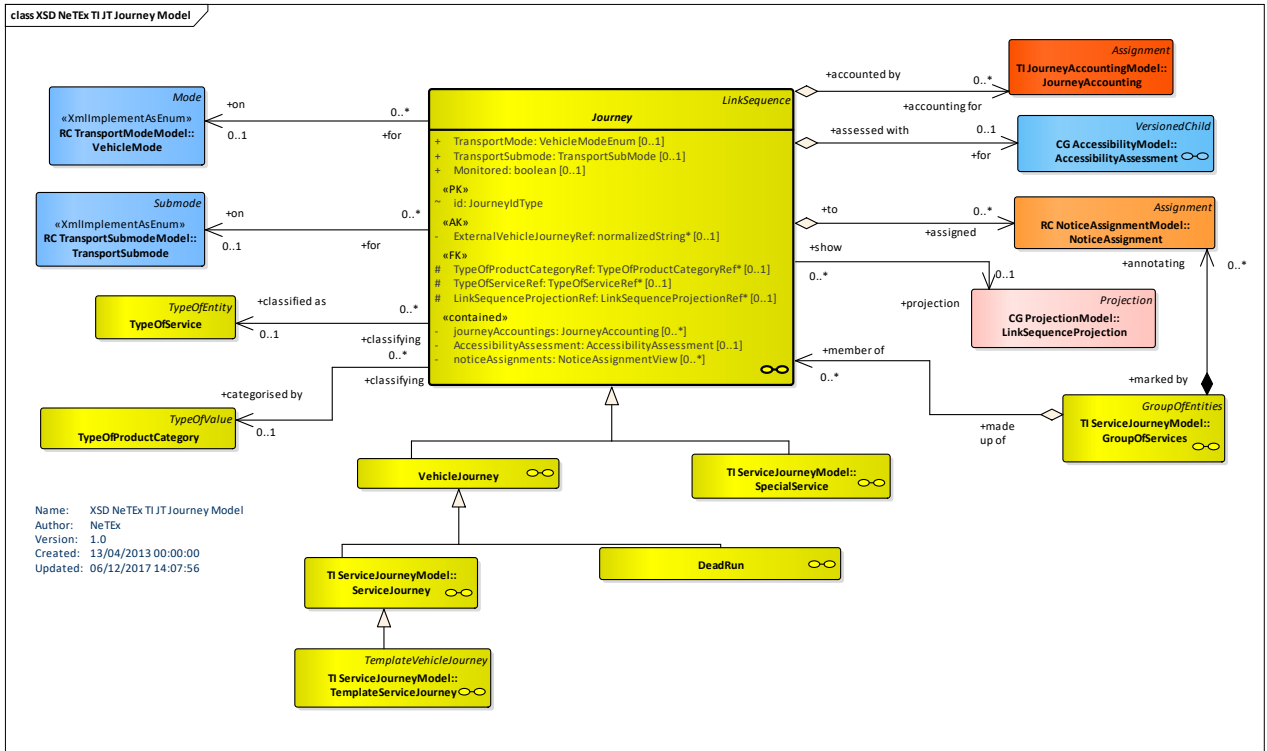


Figure 12 —Journey – Journey MODEL (UML)

7.2.1.1.5 Vehicle Journey Details – Physical Model

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

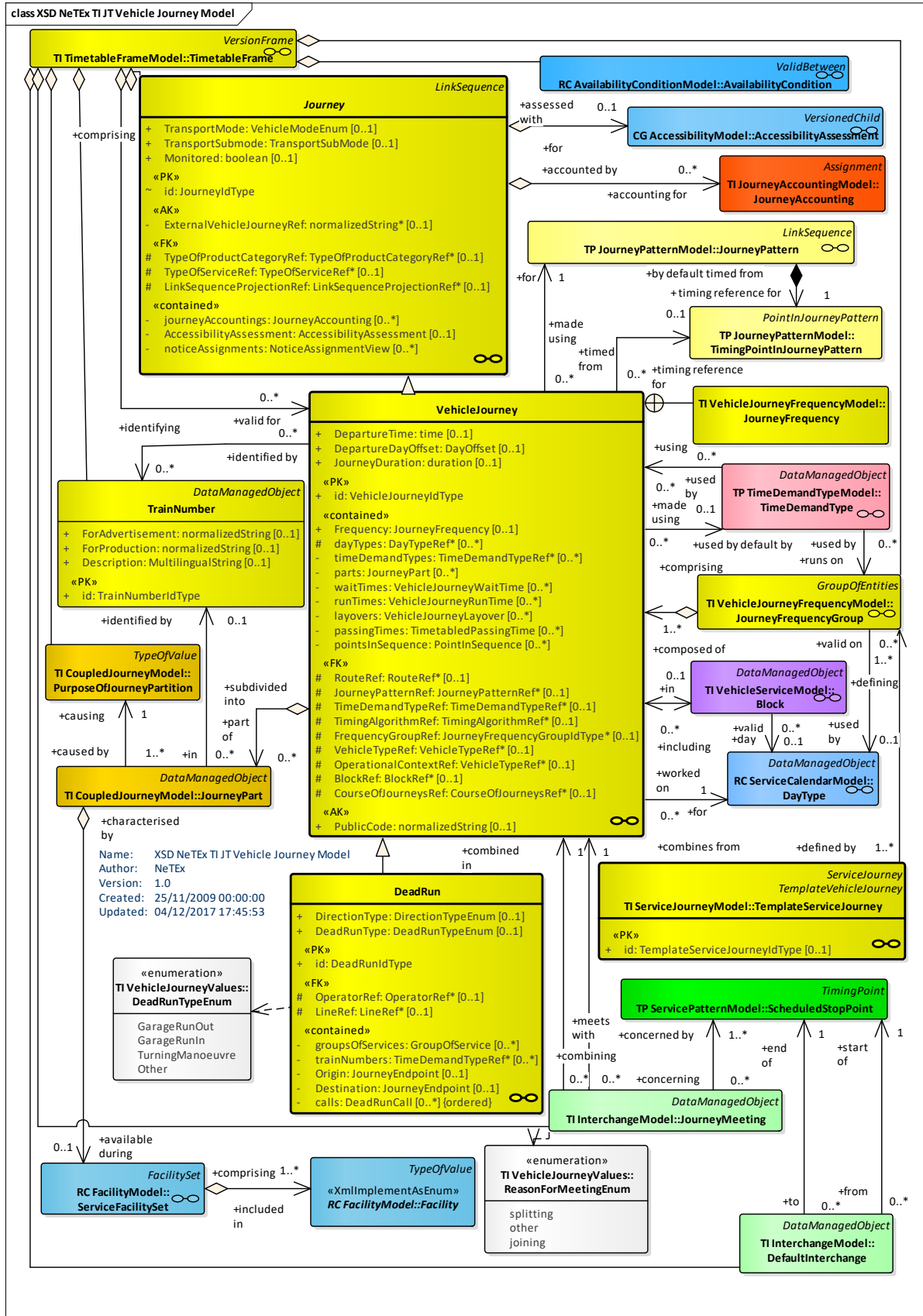


Figure 13 — Vehicle Journey – Physical Model (UML)

7.2.1.1.6 Vehicle Journey Identification – Physical Model

Sometimes it is useful to be able to reference a service journey without specifying an identifier. A *JourneyDesignator* allows a JOURNEY to be identified using a combination of attributes.

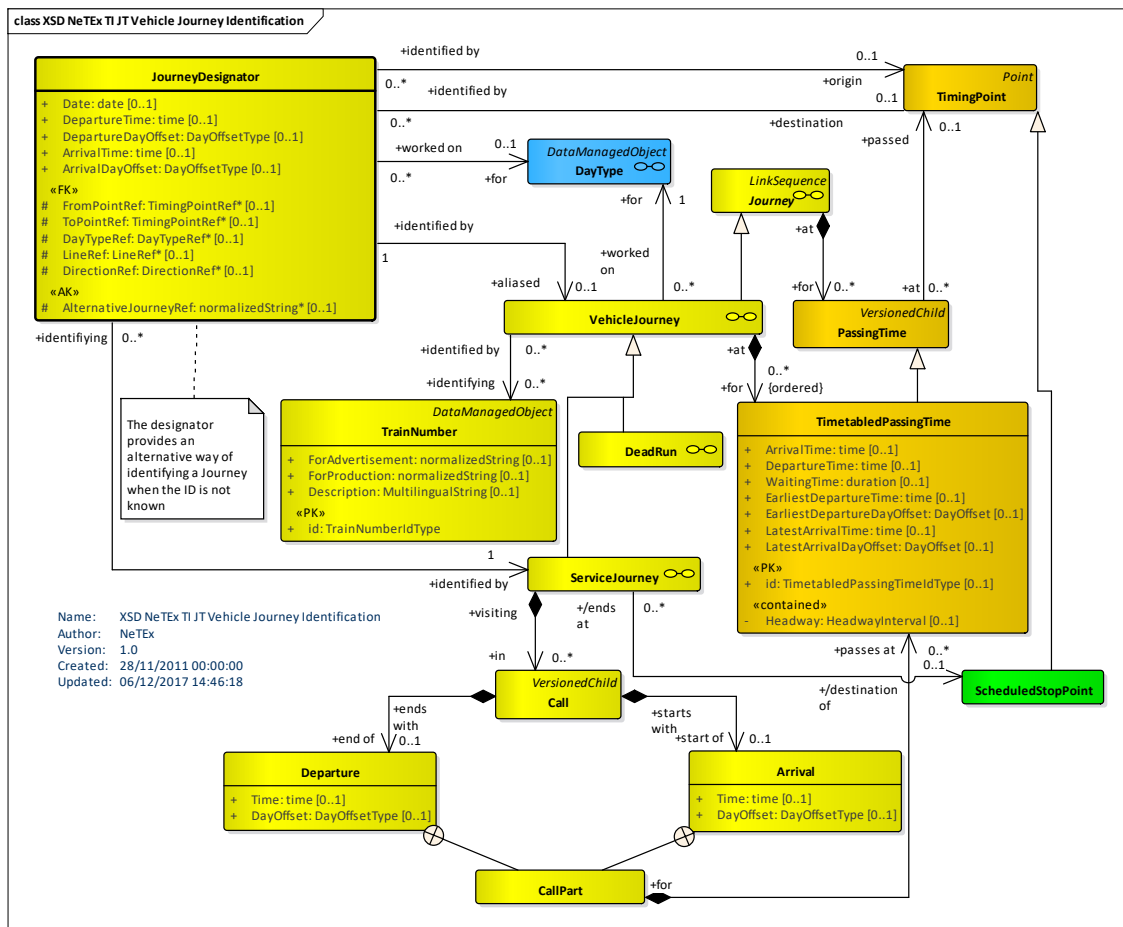


Figure 14 — Vehicle Journey Identification – Physical Model (UML)

7.2.1.2 Vehicle Journey – Attributes and XSD

7.2.1.2.1 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	Cardinality	Description
::>	::>	<i>LinkSequence</i>	::>	JOURNEY inherits from LINK SEQUENCE.
«PK»	id	<i>JourneyIdType</i>	1:1	Identifier of JOURNEY.
	Description	<i>MultilingualString</i>	0:1	Description of JOURNEY.
	TransportMode	<i>VehicleModeEnum</i>	0:1	Transport MODE of JOURNEY.

	TransportSub-mode	<i>TransportSubmode</i>	0:1	Transport Sub MODE of JOURNEY.
«AK»	ExternalJourneyRef	<i>ExternalObjectRef</i>	0:1	An alternative code that uniquely identifies the JOURNEY, specifically for use in AVMS systems. <i>NOTE For VDV compatibility.</i>
«FK»	TypeOfProductCategoryRef	<i>TypeOfProductCategoryRef</i>	0:1	PRODUCT CATEGORY of a JOURNEY.
«FK»	TypeOfServiceRef	<i>TypeOfServiceRef</i>	0:1	TYPE OF SERVICE of JOURNEY.
«FK»	a LinkSequenceProjectionRef	<i>LinkSequenceProjectionRef</i>	0:1	Reference to LINK SEQUENCE PROJECTION to use to PROJECT JOURNEY onto map, etc.
	b LinkSequenceProjection	<i>LinkSequenceProjection</i>	0:1	LINK SEQUENCE PROJECTION to use to PROJECT JOURNEY onto map, etc.
	Monitored	<i>xsd:boolean</i>	0:1	Whether the journey will be monitored in real time.
	AccessibilityAssessment	<i>AccessibilityAssessment</i>	0:*	ACCESSIBILITY ASSESSMENT that apply of JOURNEY.
«cntd»	journeyAccountings	<i>JourneyAccounting</i>	0:*	JOURNEY ACCOUNTINGS that apply of JOURNEY.
	noticeAssignments	<i>noticeAssignment</i>	0:1	NOTICES relevant for the whole GROUP OF SERVICES.

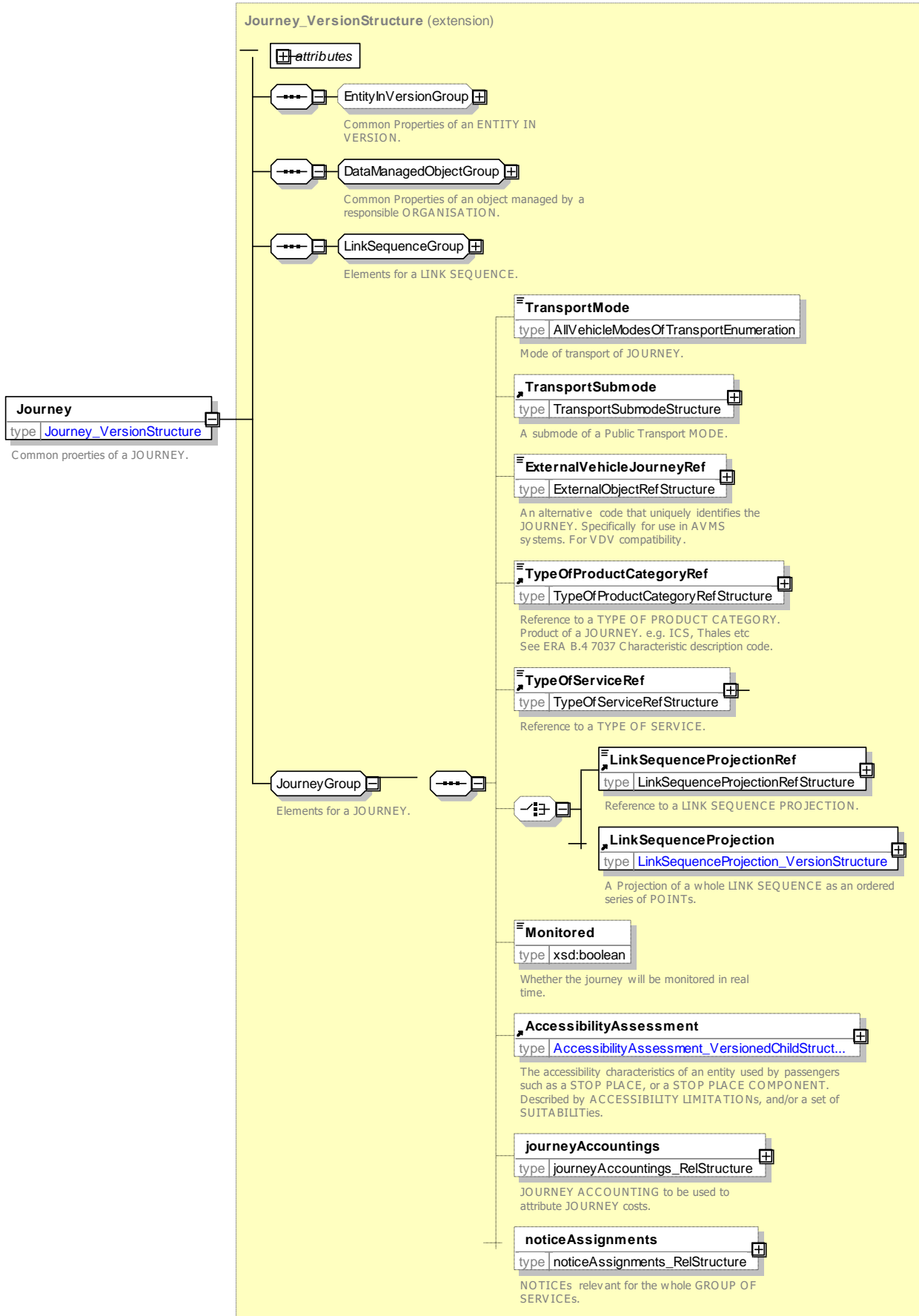


Figure 15 — Journey — XSD

7.2.1.2.2 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
::>	::>	<i>Journey</i>	::>	VEHICLE JOURNEY inherits from JOURNEY.
«PK»	id	<i>VehicleJourney IdType</i>	1:1	Identifier of VEHICLE JOURNEY.
	DepartureTime	<i>xsd:time</i>	1:1	Departure time of VEHICLE JOURNEY.
	DepartureDayOffset	<i>DayOffsetType</i>	0:1	Day offset if day of departure time of VEHICLE JOURNEY differs from the current OPERATING DAY. V1.1
	JourneyDuration	<i>xsd:duration</i>	0:1	Duration of VEHICLE JOURNEY.
«cntd»	Frequency	<i>JourneyFrequency</i>	0:*	Headway frequencies for VEHICLE JOURNEY.
«cntd»	dayTypes	<i>DayTypeRef</i>	1:*	Reference to DAY TYPE on which VEHICLE JOURNEY runs.
«FK»	RouteRef	<i>RouteRef</i>	0:1	Reference to ROUTE on which VEHICLE JOURNEY runs.
«FK»	JourneyPatternRef	<i>JourneyPatternRef</i>	0:1	Reference to JOURNEY PATTERN over which VEHICLE JOURNEY runs.
«FK»	TimeDemandRef	<i>TimeDemandTypeRef</i>	0:1	Reference to TIME DEMAND TYPE during which VEHICLE JOURNEY starts.
«FK»	TimingAlgorithmRef	<i>TimingAlgorithmRef</i>	0:1	Reference to TIMING ALGORITHM to use to compute passing times for VEHICLE JOURNEY.
«FK»	JourneyFrequencyGroupRef	<i>JourneyFrequencyGroupRef</i>	0:1	FREQUENCY GROUP of VEHICLE JOURNEY.
«FK»	VehicleTypeRef	<i>VehicleTypeRef</i>	0:1	Reference to VEHICLE TYPE which VEHICLE JOURNEY runs.
«FK»	<i>OperationalContextRef</i>	<i>OperationalContextRef</i>	0:1	Reference to OPERATIONAL CONTEXT in which VEHICLE JOURNEY runs.
«FK»	BlockRef	<i>BlockRef</i>	0:1	Reference to BLOCK under which VEHICLE JOURNEY runs.
«FK»	CourseOfJourneysRef	CourseOfJourneysRef	0:1	Reference to COURSE OF JOURNEYS under which VEHICLE JOURNEY runs.
«AK»	PublicCode	<i>xsd:normalizedString</i>	0:1	Public Code for VEHICLE JOURNEY.
«cntd»	layovers	<i>VehicleJourneyLayover</i>	0:*	LAYOVERS for VEHICLE JOURNEY.
«cntd»	waitTimes	<i>VehicleJourneyWaitTime</i>	0:*	WAIT TIMES for VEHICLE JOURNEY.
«cntd»	runTimes	<i>VehicleJourneyRunTime</i>	0:*	RUN TIMES for VEHICLE JOURNEY.
«cntd»	passingTimes	<i>TimetabledPassingTime</i>	0:*	TIMETABLED PASSING TIMES for VEHICLE JOURNEY.
«cntd»	timeDemandTypes	<i>TimeDemandTypeRef</i>	0:*	TIME DEMAND TYPEs referenced by VEHICLE JOURNEY.

«cntd»	parts	<i>JourneyPart</i>	0:*	JOURNEY PARTs for VEHICLE JOURNEY.
«cntd»	noticeAssignments	<i>NoticeAssignmentView</i>	0:*	NOTICE ASSIGNMENTs that apply to JOURNEY.
«cntd»	pointsInSequence	<i>PointInSequence</i>	0:*	POINTs in SEQUENCE for VEHICLE JOURNEY.

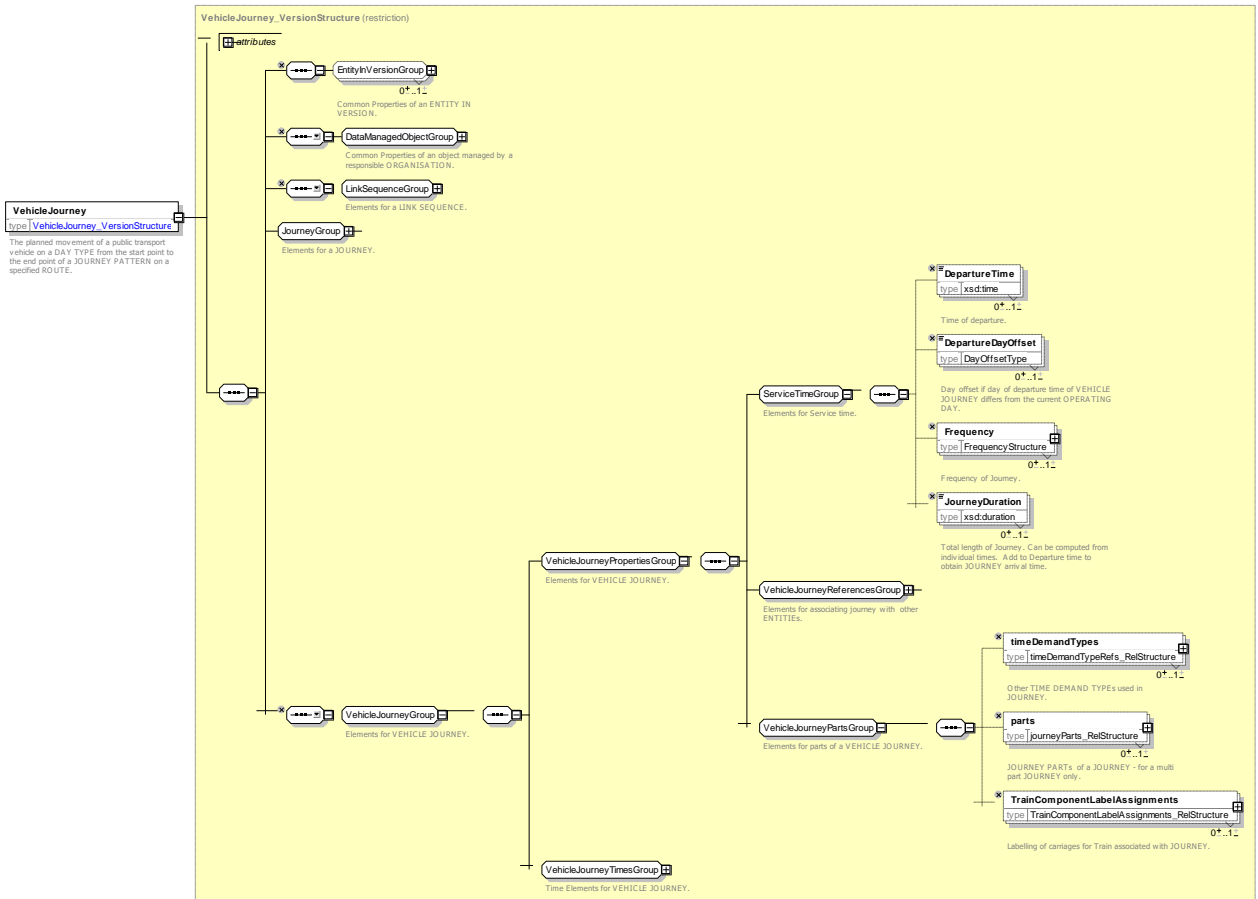


Figure 16 — VehicleJourney — XSD

7.2.1.2.2.1 *VehicleJourneyReferencesGroup - Group*

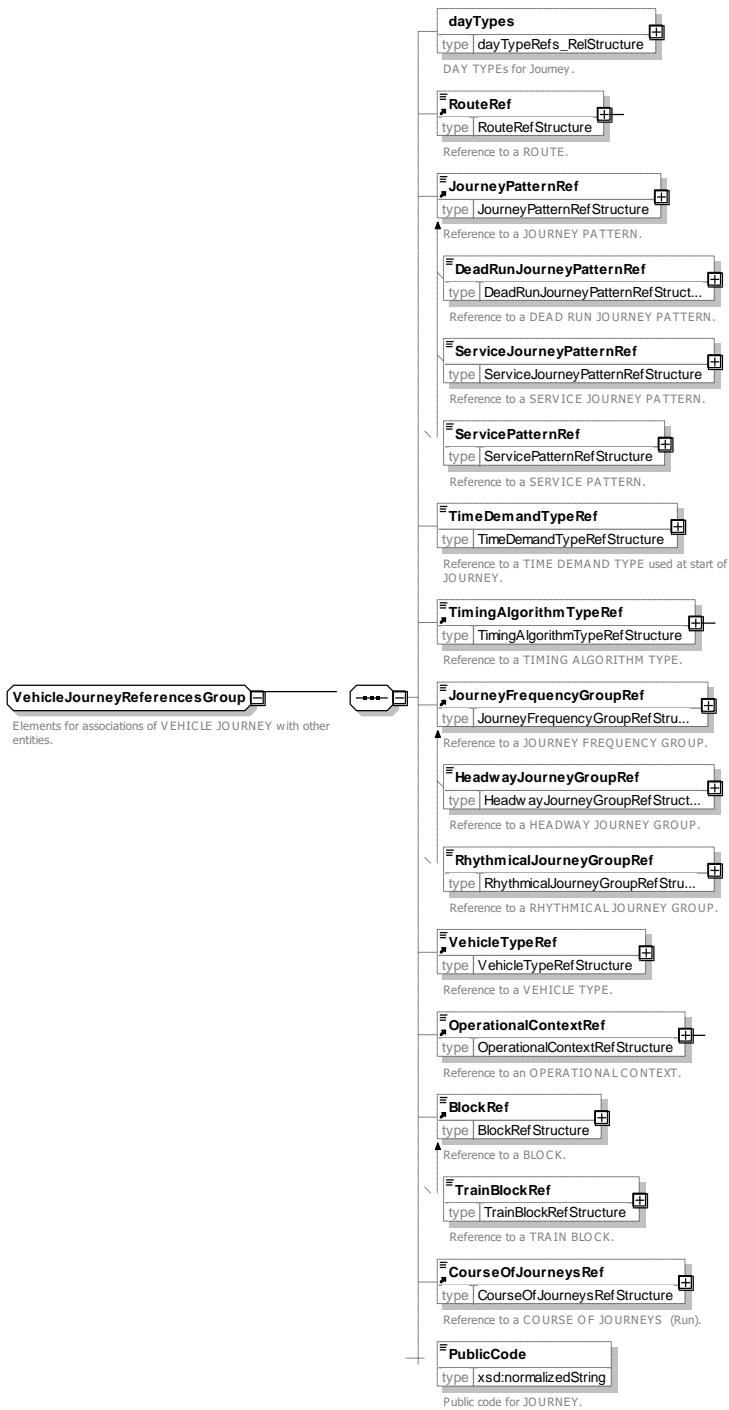


Figure 17 — *VehicleJourneyReferencesGroup* — XSD

7.2.1.2.2 VehicleJourneyTimesGroup - Group

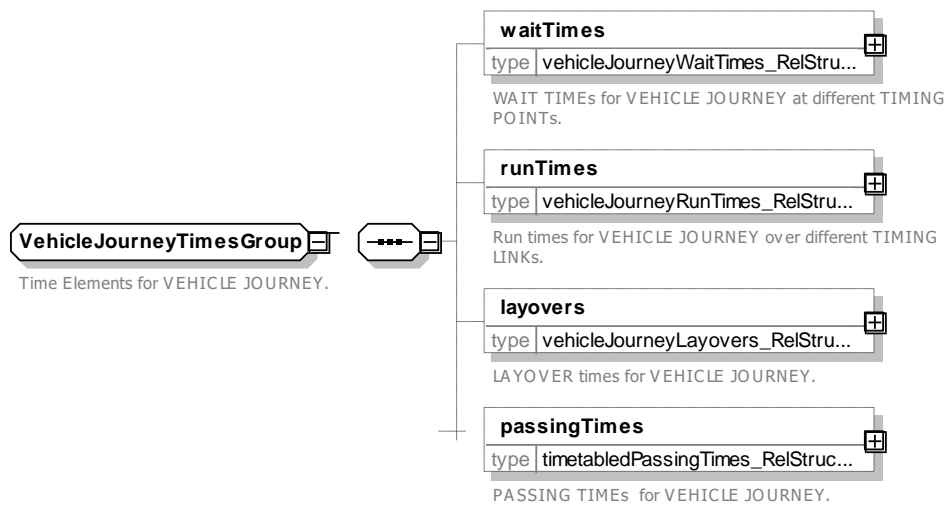


Figure 18 — VehicleJourneyTimesGroup — XSD

7.2.1.2.3 DeadRun – Model Element

A non-service VEHICLE JOURNEY.

Table 4 – DeadRun – Element

Classification	Name	Type	Cardinality	Description
::>	::>	VehicleJourney	::>	DEAD RUN inherits from VEHICLE JOURNEY.
«PK»	id	DeadRunIdType	1:1	Identifier of DEAD RUN.
«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	GroupOfServiceRef	0:*	GROUP OF SERVICEs to which DEAD RUN belongs.
«cntd»	trainNumbers	TrainNumber	0:*	TRAIN NUMBERs associated with DEAD RUN.
«cntd»	Origin	DeadRunEndpoint	0:1	Origin of DEAD RUN.
«cntd»	Destination	DeadRunEndpoint	0:1	Destination of DEAD RUN.
	DeadRunType	DeadRunTypeEnum	1:1	Type of DEAD RUN.
«cntd»	calls	DeadRunCall	0:*	CALLs making up DEAD RUN.

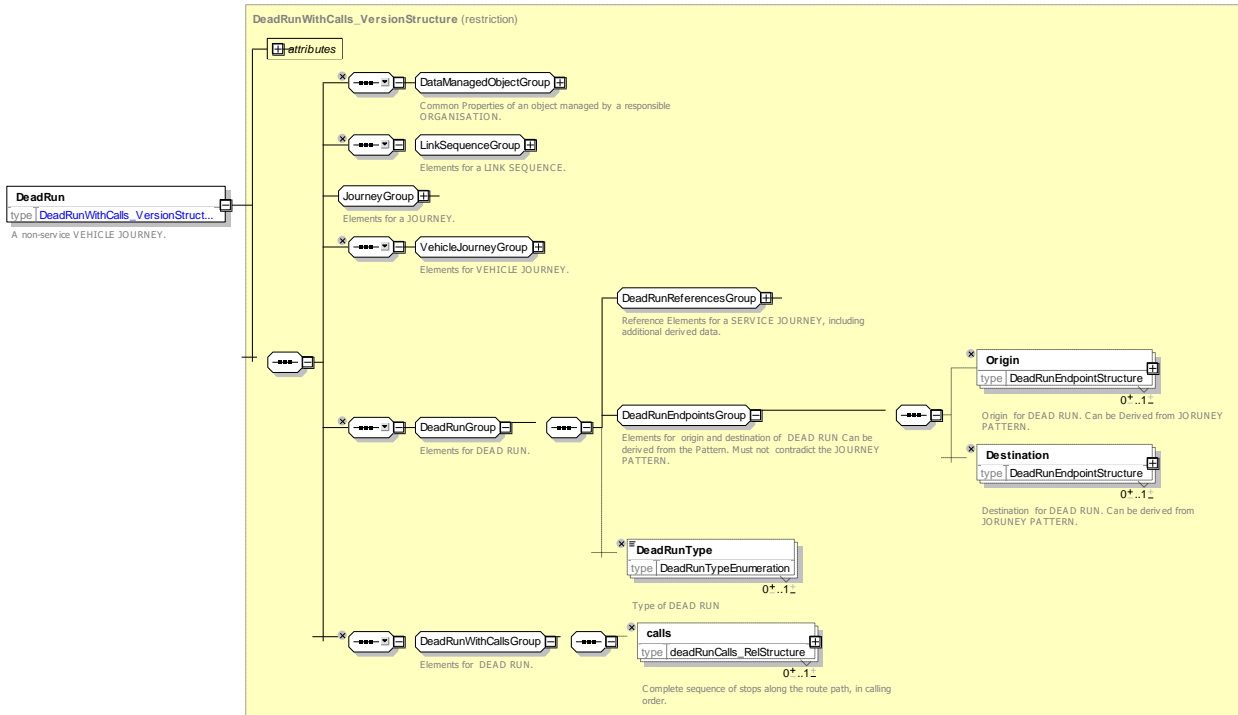
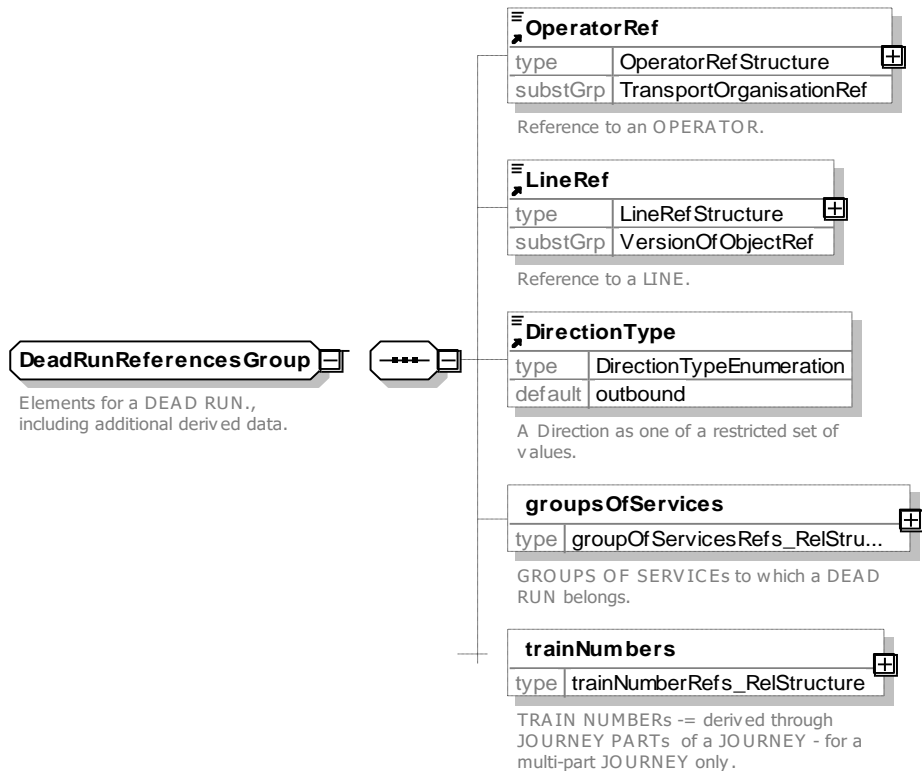


Figure 19 — DeadRun — XSD

7.2.1.2.3.1 DeadRunReferencesGroup — Group



Generated by XMLSpy

www.altova.com

Figure 20 — DeadRunReferencesGroup — XSD

7.2.1.2.3.1 DeadRunType – Allowed values

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

Table 5 – DeadRunType – Allowed values

Value	Description
<i>garageRunOut</i>	Garage run out
<i>garageRunIn</i>	Garage run in
<i>turningManoeuvre</i>	Turning manoeuvre
<i>other</i>	Other

7.2.1.2.4 DeadRunEndPoint – Model Element

Elements for origin and destination of DEAD RUN Can be derived from the Pattern. Must not contradict the JOURNEY PATTERN. .

Table 6 – DeadRunEndPoint – Structure

Classification	Name	Type	Cardinality	Description
«EV»	<i>PointRef</i>	<i>PointRef</i>	0:1	Reference to POINT at end of DEAD RUN..
«EV»	<i>TypeOfPointRef</i>	<i>TypeOfPointRef</i>	0:1	Reference to TYPE OF POINT of POINT at end of DEAD RUN..

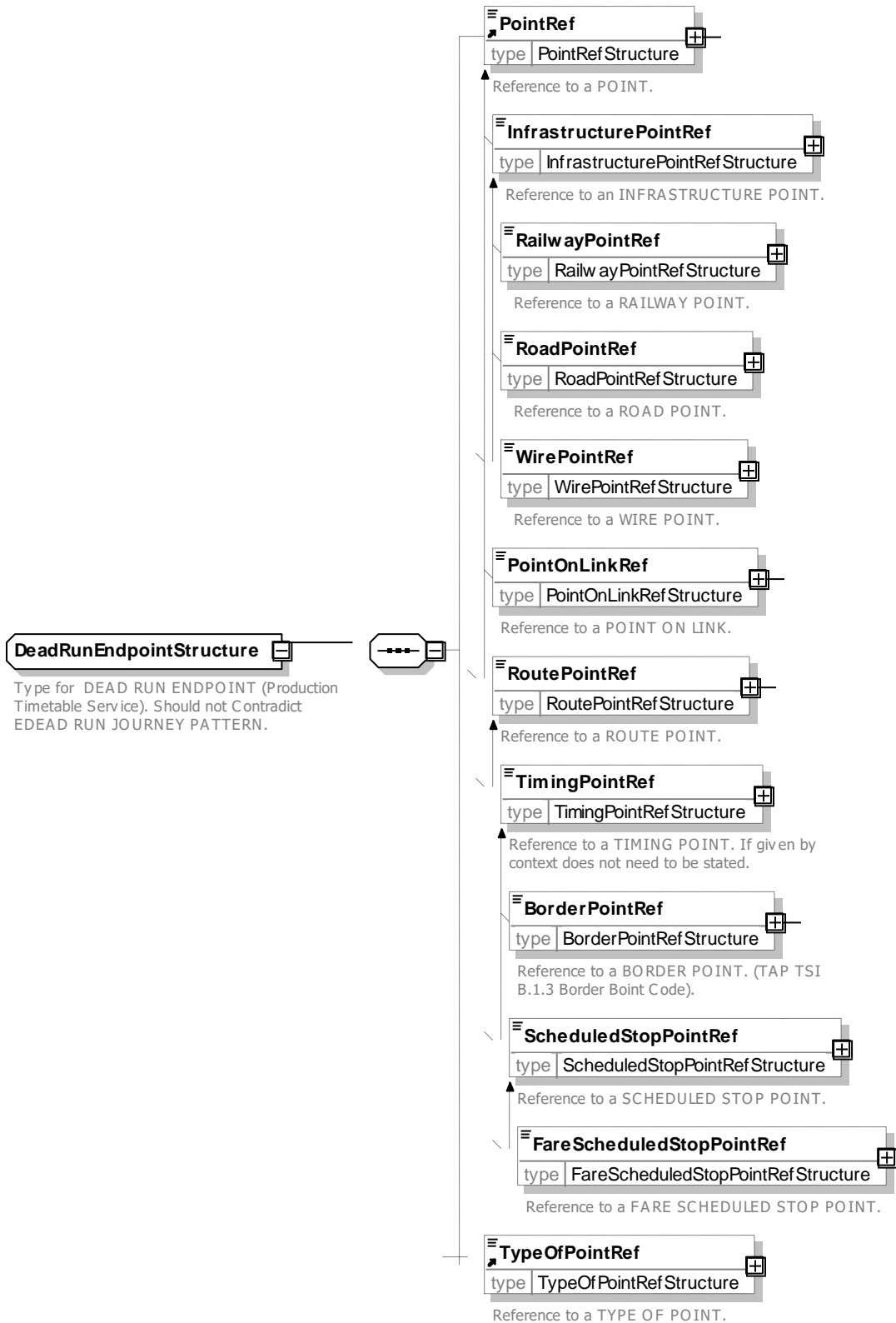


Figure 21 — *DeadRunEndPoint* — XSD

7.2.1.2.5 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 7 – TemplateVehicleJourney – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Journey</i>	::>	TEMPLATE VEHICLE JOURNEY inherits from JOURNEY.
«PK»	id	<i>TemplateVehicleJourney-IdType</i>	0:1	Identifier of TEMPLATE VEHICLE JOURNEY.
	TemplateVehicleJourneyType	<i>TemplateVehicleJourney-TypeEnum</i>	0:1	Type of TEMPLATE VEHICLE JOURNEY.
«cntd»	frequencyGroups	<i>JourneyFrequencyGroup</i>	0:*	FREQUENCY GROUPs defining TEMPLATE VEHICLE JOURNEY.

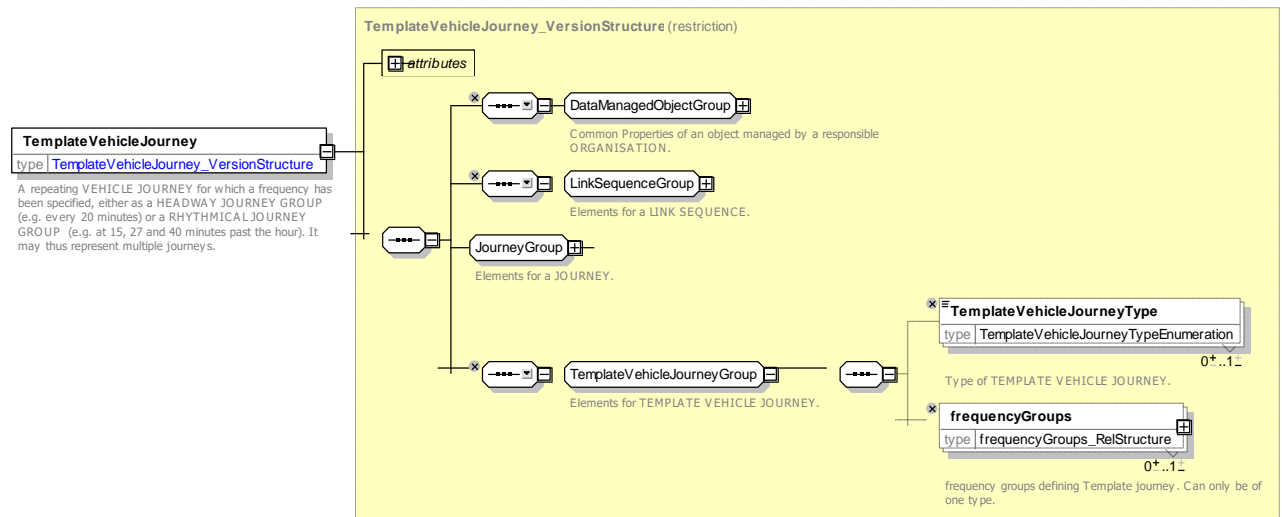


Figure 22 — TemplateVehicleJourney — XSD

7.2.1.2.5.1 TemplateVehicleJourneyType – Allowed values

The following table shows the allowed values for **TemplateVehicleJourneyType** (TemplateVehicleJourney-TypeEnum).

Table 8 – TemplateVehicleJourneyType – Allowed values

Value	Description
<i>headway</i>	Journeys run at intervals
<i>rhythmic</i>	Journeys run according to a pattern of start times.
<i>other</i>	Other

7.2.1.2.6 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 9 – TypeOfService – Element

Classification	Name	Type	Cardinality	Description
::>	::>	TypeOfEntity	::>	TYPE OF SERVICE inherits from TYPE OF ENTITY.
«PK»	<i>id</i>	TypeOfServiceIdType	1:1	Identifier of TYPE OF SERVICE.

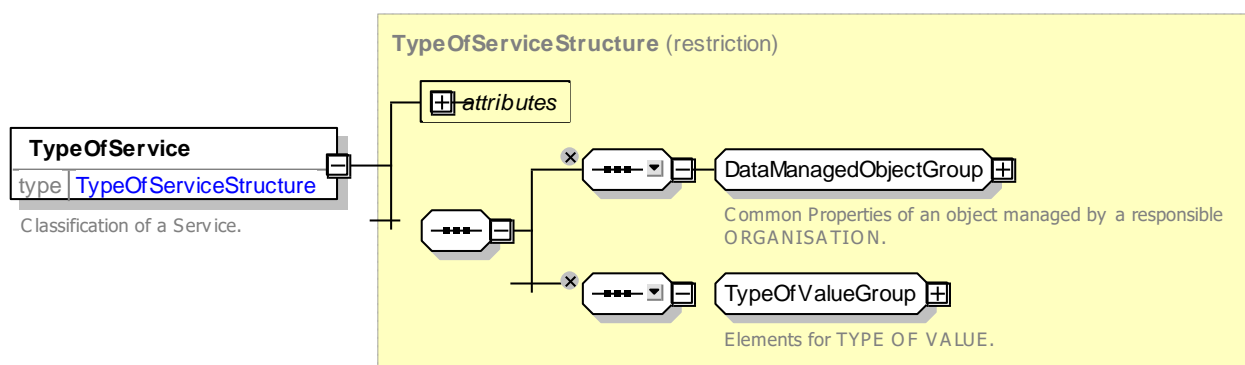


Figure 23 — TypeOfService — XSD

7.2.1.2.7 TypeOfProductCategory – Model Element

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

Table 10 – TypeOfProductCategory – Element

Classification	Name	Type	Cardinality	Description
::>	::>	TypeOfValue	::>	TYPE OF PRODUCT CATEGORY inherits from TYPE OF VALUE.
«PK»	<i>id</i>	TypeOfProductCategoryIdType	1:1	Identifier of TYPE OF PRODUCT CATEGORY.

7.2.1.2.8 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 11 – TrainNumber – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	TRAIN NUMBER inherits from DATA MANAGED OBJECT
«PK»	id	<i>TrainNumberIdType</i>	1:1	Identifier of TRAIN NUMBER.
	Description	<i>MultilingualString</i>	0:1	Description of TRAIN NUMBER.
	ForAdvertisement	<i>xsd:normalizedString</i>	0:1	TRAIN NUMBER to use for advertisement to public if different from ID.
	ForProduction	<i>xsd:normalizedString</i>	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.

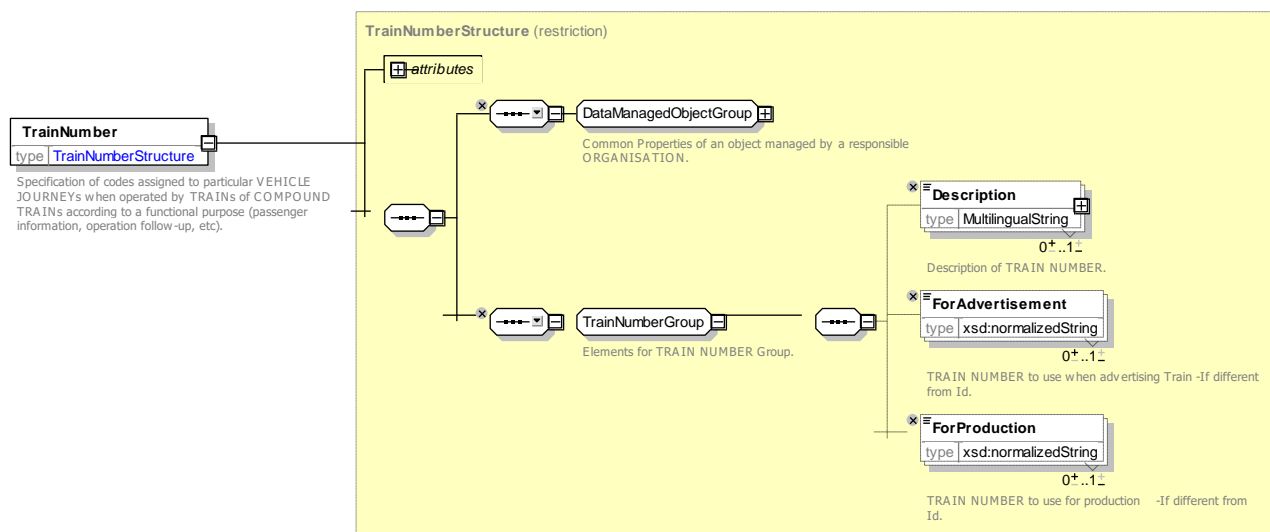


Figure 24 — TrainNumber — XSD

7.2.1.2.9 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 12 – JourneyDesignator – Element

Classification	Name	Type	Cardinality	Description
«FK»	FromPointRef	<i>TimingPointRef</i>	0:1	Origin TIMING POINT to use to identify a JOURNEY.

«FK»	ToPointRef	<i>TimingPointRef</i>	0:1	Destination TIMING POINT to use to identify JOURNEY.
	Date	<i>xsd:date</i>	0:1	Departure date of VEHICLE JOURNEY from origin point.
	DepartureTime	<i>xsd:time</i>	0:1	Departure time of VEHICLE JOURNEY from origin point.
	DepartureDayOffset	<i>DayOffsetType</i>	0:1	Departure day offset from current OPERATING DAY of VEHICLE JOURNEY from origin point.
	ArrivalTime	<i>xsd:time</i>	0:1	Arrival time of VEHICLE JOURNEY at destination point.
	ArrivalDayOffset	<i>ArrivalOffsetType</i>	0:1	Arrival day offset from current OPERATING DAY of VEHICLE JOURNEY from origin point.
«FK»	DayTypeRef	<i>DayTypeRef</i>	0:1	DAY TYPE to use to identify JOURNEY.
«FK»	LineRef	<i>LIneRef</i>	0:1	LINE to use to identify JOURNEY.
«FK»	DirectionRef	<i>LIneRef</i>	0:1	DIRECTION to use to identify JOURNEY.
«AK»	AlternativeJourneyRef	<i>xsd:normalizedString</i>	0:1	Alternative identifier of JOURNEY. May possibly be unique only in a limited context.

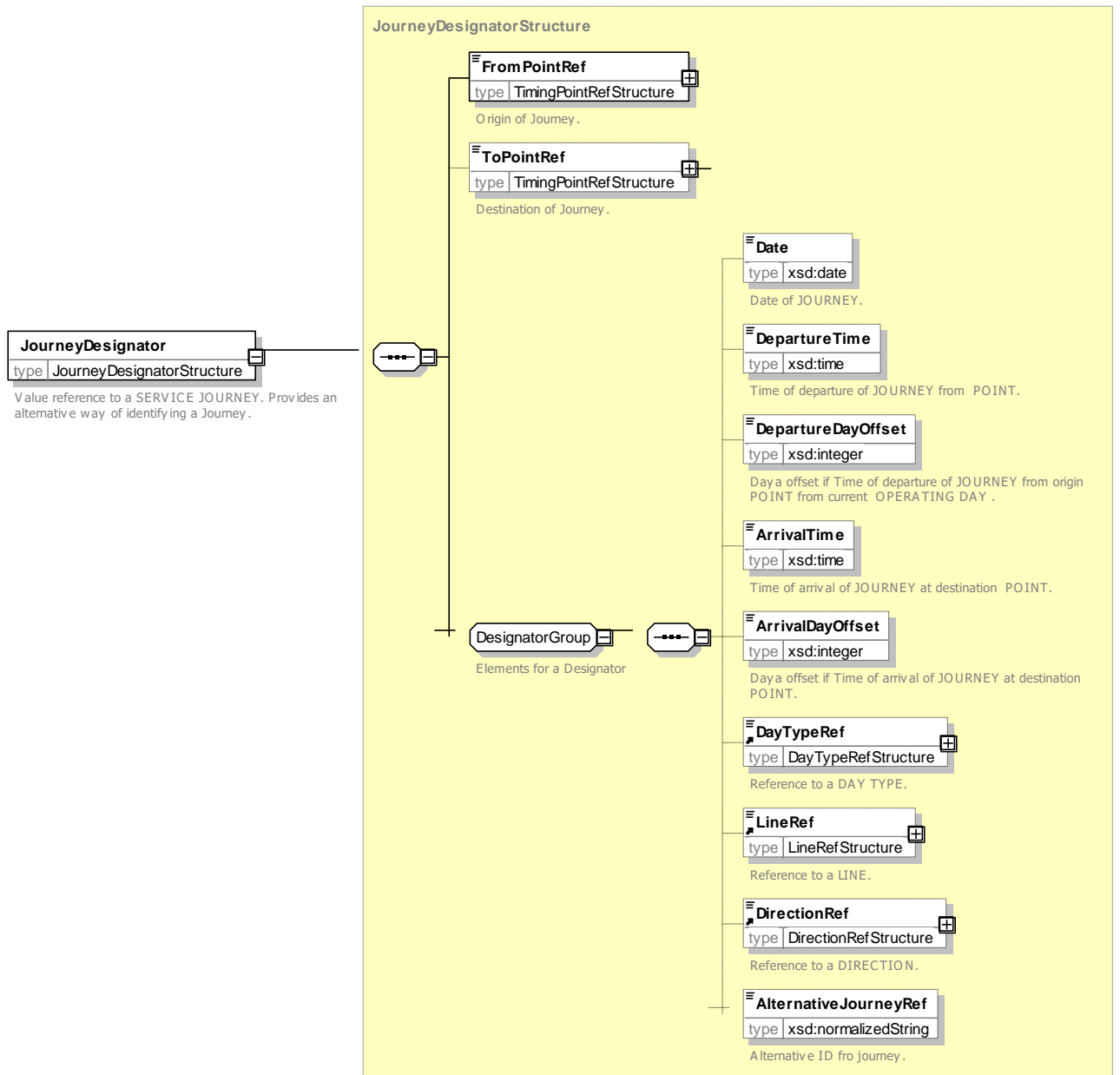


Figure 25 — JourneyDesignator — XSD

7.2.1.3 XML Examples of Vehicle Journey

See SERVICE JOURNEY for examples of a VEHICLE JOURNEY.

7.2.1.3.1 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"/>
</DeadRun>
```

```

<BlockRef version="any" ref="ops:blk_L24_01"/>
<passingTimes>
  <TimetabledPassingTime id="kx:tpt_dr_L24_01o_001">
    <TimingPointInJourneyPatternRef version="any" ref="kx:pjip_drjp_L24_01o_001"/>
    <DepartureTime>11:00:00.0Z</DepartureTime>
  </TimetabledPassingTime>
  <TimetabledPassingTime id="kx:tpt dr L24 01o 002">
    <StopPointInJourneyPatternRef version="any" ref="kx:pjip_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.

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.

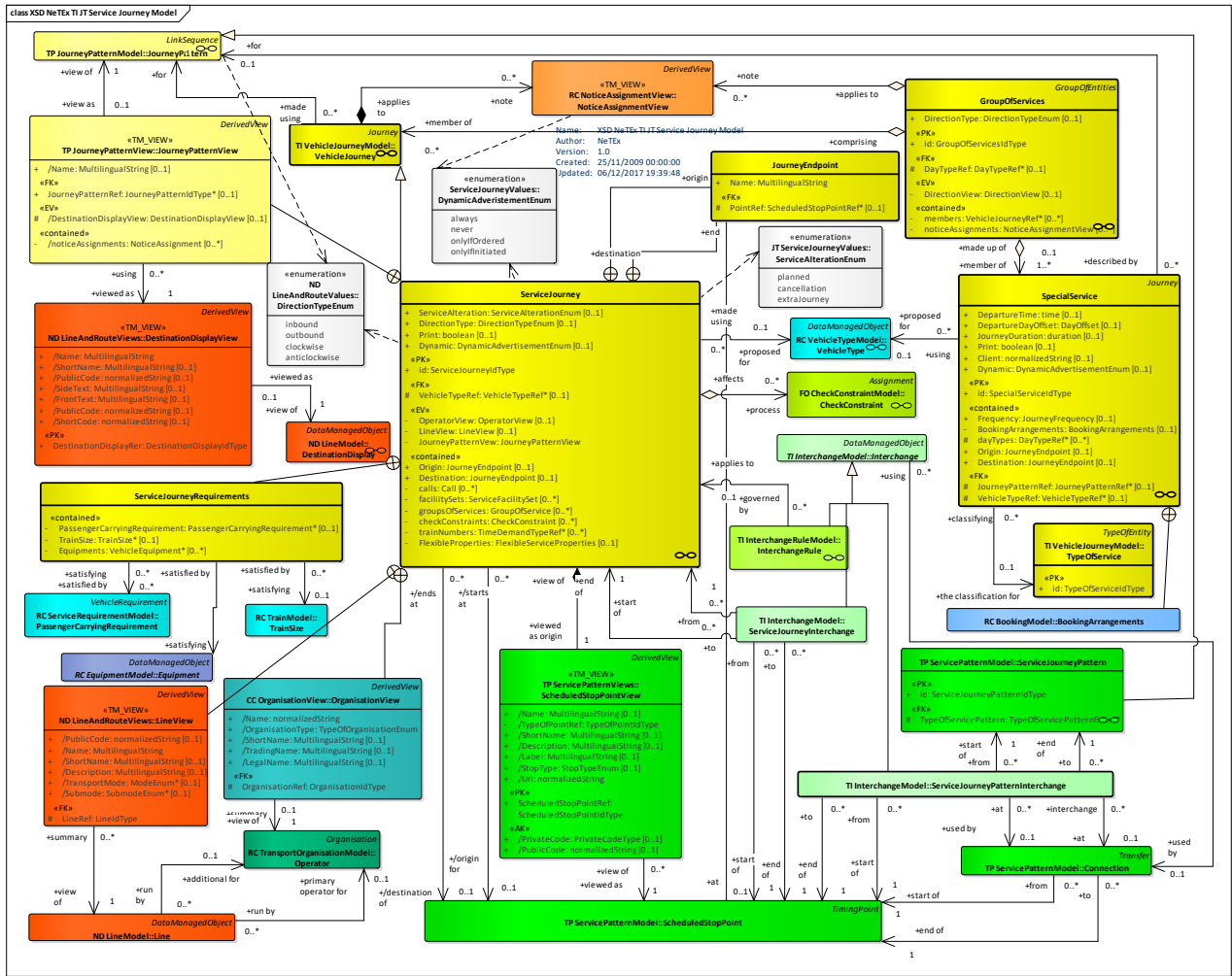


Figure 27 — Service Journey – Physical Model (UML)

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 13 – GroupOfServices – Element

Classification	Name	Type	Cardinality	Description
::>	::>	GroupOfEntities	::>	GROUP OF SERVICES inherits from GROUP OF ENTITIES.
«PK»	id	GroupOfServicesIdType	1:1	Identifier of GROUP OF SERVICES.
«FK»	dayTypes	DayTypeRef	0:*	Reference DAY TYPES applying to the GROUP OF SERVICES.
	DirectionType	DirectionTypeEnum	1:1	Type of DIRECTION. See Part 1.

«EV»	DirectionView	<i>DirectionView</i>	0:1	Reference to common DIRECTION of SERVICE JOURNEYS in the GROUP OF SERVICE, with derived values.
	origin	<i>EndPoint_DerivedView</i>		View of derived data about origin of journey. +V1.1
	destination	<i>EndPoint_DerivedView</i>		View of derived data about destination of journey. +V1.1
«cntd»	destinationDisplays	<i>DestinationDisplay</i>	0:*	DESTINATION DISPLAYS associated with the GROUP OF SERVICES.
«cntd»	members	<i>VehicleJourneyRef</i>	0:*	Members of a GROUP OF SERVICES.
«cntd»	noticeAssignments	<i>NoticeAssignmentView</i>	0:*	NOTICE ASSIGNMENTS that apply to a GROUP OF SERVICE.

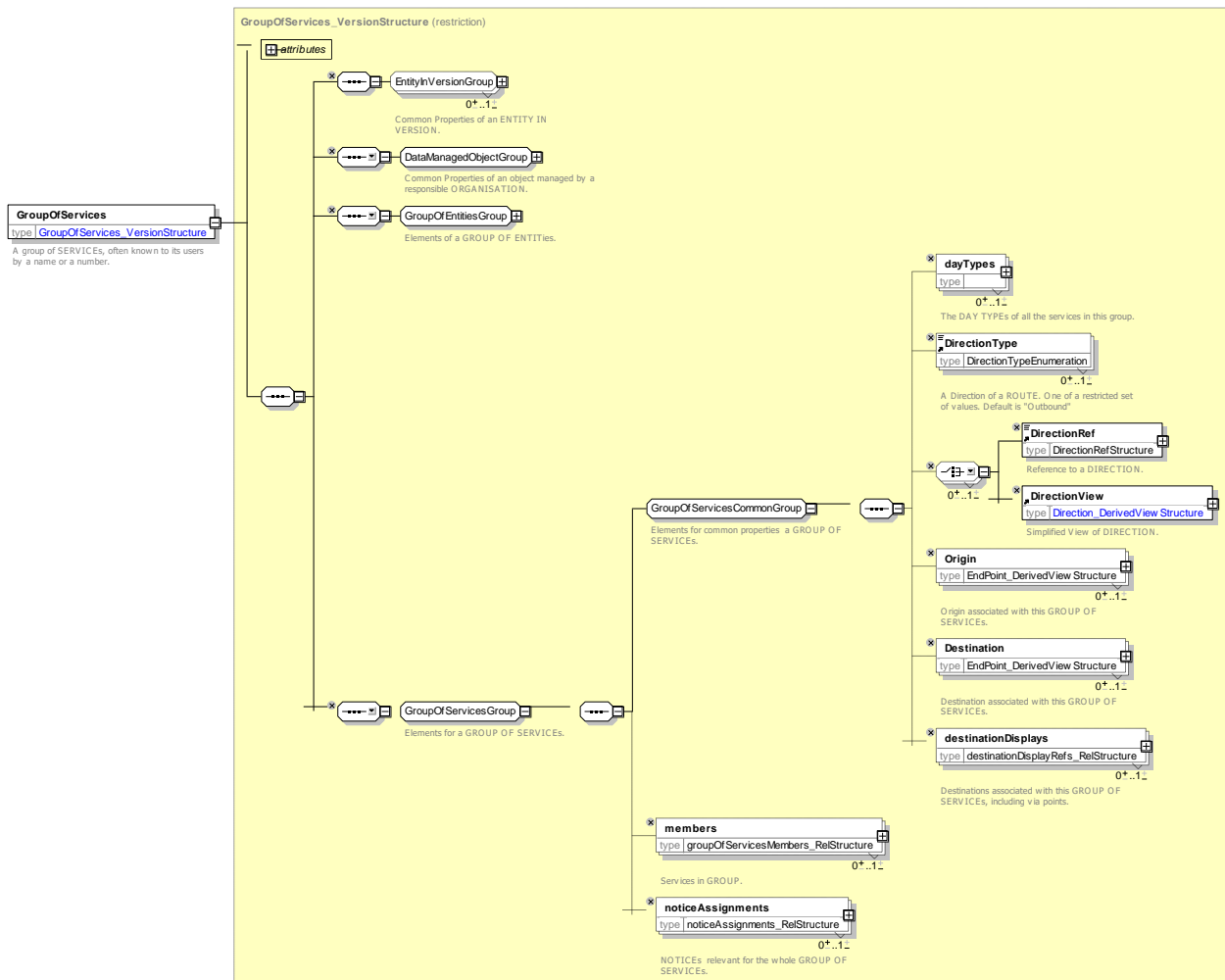


Figure 28 — GroupOfServices — XSD

7.2.2.4.1.1 EndPoint_DerivedViewStructure – Model Structure

The **Origin** and **Destination** elements of a **GroupOfServiceJourneys** allow information about a common origin and destination of the journeys in the group to be given for passenger information purposes. In principle these can be derived from the journey patterns of the journeys in the group.

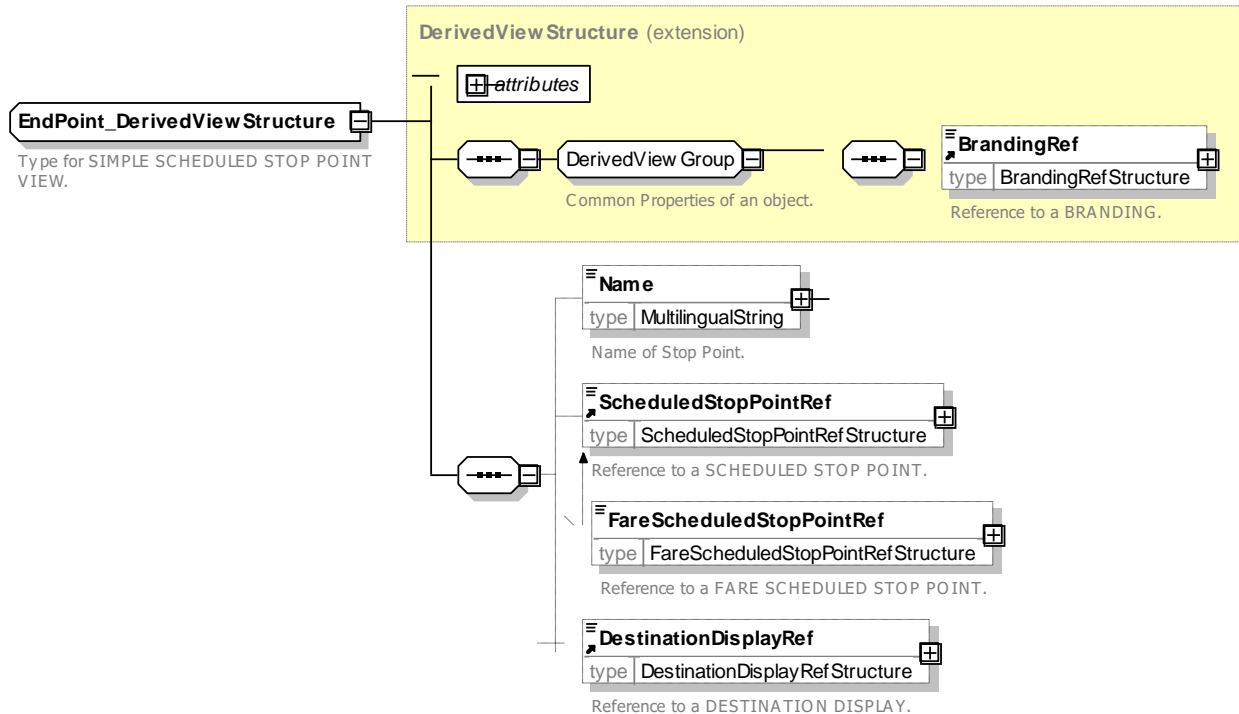


Figure 29 — VehicleJourneyReferencesGroup — XSD

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 14 – ServiceJourney – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Journey</i>	::>	SERVICE JOURNEY inherits from JOURNEY. It includes elements from VEHICLE JOURNEY
«PK»	id	<i>ServiceJourneyIdType</i>	1:1	Identifier of SERVICE JOURNEY.
	ServiceAlteration	<i>ServiceAlterationEnum</i>	0:1	Status to journey - planned, cancelled, etc.
Service Time	DepartureTime	<i>xsd:time</i>	0:1	Time of departure.
	DepartureDayOffset	<i>DayOffsetType</i>	0:1	Day offset if day of departure time of VEHICLE JOURNEY differs from the current OPERATING DAY.
	Frequency	<i>Frequency</i>	0:1	Frequency of Journey.

	JourneyDuration	<i>xsd:duration</i>	0:1	Total length of Journey. Can be computed from individual times. Add to Departure time to obtain JOURNEY arrival time.
«cntd»	dayTypes	<i>dayTypeRefs_RelStructure</i>	0:1	DAY TYPEs for Journey.
«FK»	RouteRef	<i>RouteRef</i>	0:1	Reference to a ROUTE.
«FK»	JourneyPattern-Ref	<i>JourneyPattern-Ref</i>	0:1	Reference to a JOURNEY PATTERN.
«FK»	TimeDemandTypeRef	<i>TimeDemandType</i>	0:1	Reference to a TIME DEMAND TYPE used at start of JOURNEY.
«FK»	TimingAlgorithmTypeRef	<i>TimingAlgorithmTypeRef</i>	0:1	Reference to a TIMING ALGORITHM TYPE.
«FK»	JourneyFrequencyGroupRef	<i>JourneyFrequencyGroupRef</i>	0:1	Reference to a JOURNEY FREQUENCY GROUP.
«FK»	VehicleTypeRef	<i>VehicleTypeRef</i>	0:1	Reference to TYPE OF VEHICLE to use on of SERVICE JOURNEY.
«FK»	OperationalContextRef	<i>OperationalContextRef</i>	0:1	Reference to an OPERATIONAL CONTEXT.
«FK»	BlockRef	<i>BlockRef</i>	0:1	Reference to a BLOCK.
«FK»	CourseOfJourneysRef	<i>CourseOfJourneysRef</i>	0:1	Reference to a COURSE OF JOURNEYS (Run).
	PublicCode	<i>xsd:normalizedString</i>	0:1	Public code for JOURNEY.
«FK»	a OperatorRef	<i>OperatorView</i>	0:1	Reference to an OPERATOR.
«EV»	b OperatorView	<i>OperatorView</i>	0:1	Simplified view of OPERATOR. All data except the identifier will be derived through the relationship.
«EV»	OperatorView	<i>OrganisationView</i>	0:1	Reference to OPERATOR of SERVICE JOURNEY. May include OPERATOR's derived values.
«FK»	a LineRef	<i>LineRef</i>	0:1	Reference to a LINE.
«EV»	b LineView	<i>LineView</i>	0:1	Reference to LINE of SERVICE JOURNEY. May include derived values of LINE.
	FlexibleLineView	<i>FlexibleLine_DerivedView</i>	0:1	A group of FLEXIBLE ROUTEs of which is generally known to the public by a similar name or number and which have common booking arrangements.
	DirectionType	<i>DirectionTypeEnum</i>	0:1	Type of DIRECTION.
«EV»	Journey-PatternView	<i>JourneyPatternView</i>	0:1	Reference to JOURNEY PATTERN of SERVICE JOURNEY and derived values of PATTERN.

				NOTE: To be used as an alternative to <i>JourneyPatternRef</i> above in <i>VehicleJourneyReferencesGroup</i>
	groupsOfServices	groupsOfServiceRef	0:*	Grouping of services of a journey - for a multi-part journey only.
	timeDemandTypes	TimeDemandTypeRef	0:*	Other TIME DEMAND TYPEs used in journey. NOTE: To be used as an alternative to <i>TimeDemandTypeRef</i> above in <i>VehicleJourneyReferencesGroup</i>
«cntd»	trainNumbers	TrainNumberRef	0:*	TRAIN NUMBERS == derived through JOURNEY PARTs of a journey - for a multi-part journey only.
«cntd»	Origin	<i>JourneyEndpoint</i>	0:1	Origin of SERVICE JOURNEY.
«cntd»	Destination	<i>JourneyEndpoint</i>	0:1	Destination of SERVICE JOURNEY.
	Print	<i>xsd:boolean</i>	0:1	Whether this journey should be visible to public in print channels.
	Dynamic	<i>DynamicAdvertisementEnum</i>	0:1	When this journey should be visible to public in dynamic channels.
«cntd»	waitTimes	<i>VehicleJourneyWaitTime</i>	0:*	WAIT TIMEs for VEHICLE JOURNEY at different TIMING POINTs.
«cntd»	runTimes	<i>VehicleJourneyRunTime</i>	0:*	Run times for VEHICLE JOURNEY over different TIMING LINKs.
«cntd»	layovers	<i>VehicleJourneyLayover</i>	0:*	LAYOVER times for VEHICLE JOURNEY.
«cntd»	passingTimes	<i>TimeTabledPassingTime</i>	0:*	PASSING TIMEs for VEHICLE JOURNEY.
«cntd»	parts	<i>JourneyPart</i>	0:*	JOURNEY PARTs of a journey - for a multi-part journey only.
«cntd»	calls	<i>Call</i>	0:*	CALLs made by SERVICE JOURNEY.
«cntd»	facilities	<i>ServiceFacilitySet</i>	0:*	Facilities on SERVICE JOURNEY.
«cntd»	checkConstraints	<i>CheckConstraint</i>	0:*	CHECK CONSTRAINTs which apply to SERVICE JOURNEY, check in time, security time. These are advisory and not for journey planning.
	a PassengerCarryingRequirementRef	<i>PassengerCarryingRequirementRef</i>	0:1	Reference to a PASSENGER CARRYING REQUIREMENT.
«EV»	b PassengerCarryingRequirementsView	<i>PassengerCarryingRequirement</i>	0:1	Requirements for carrying passengers.
	TrainSize	<i>TrainSize</i>	0:1	Requirements for TRAIN SIZE.
«cntd»	equipments	<i>vehicleEquipment</i>	0:*	VEHICLE EQUIPMENT available on service.

«FK»	a	FlexibleServicePropertiesRef	<i>FlexibleServicePropertiesRef</i>	0:1	Reference to a FLEXIBLE SERVICE PROPERTIES.
«cntd»	b	FlexibleServiceProperties	<i>FlexibleServiceProperties</i>	0:1	Flexible properties of SERVICE JOURNEY.

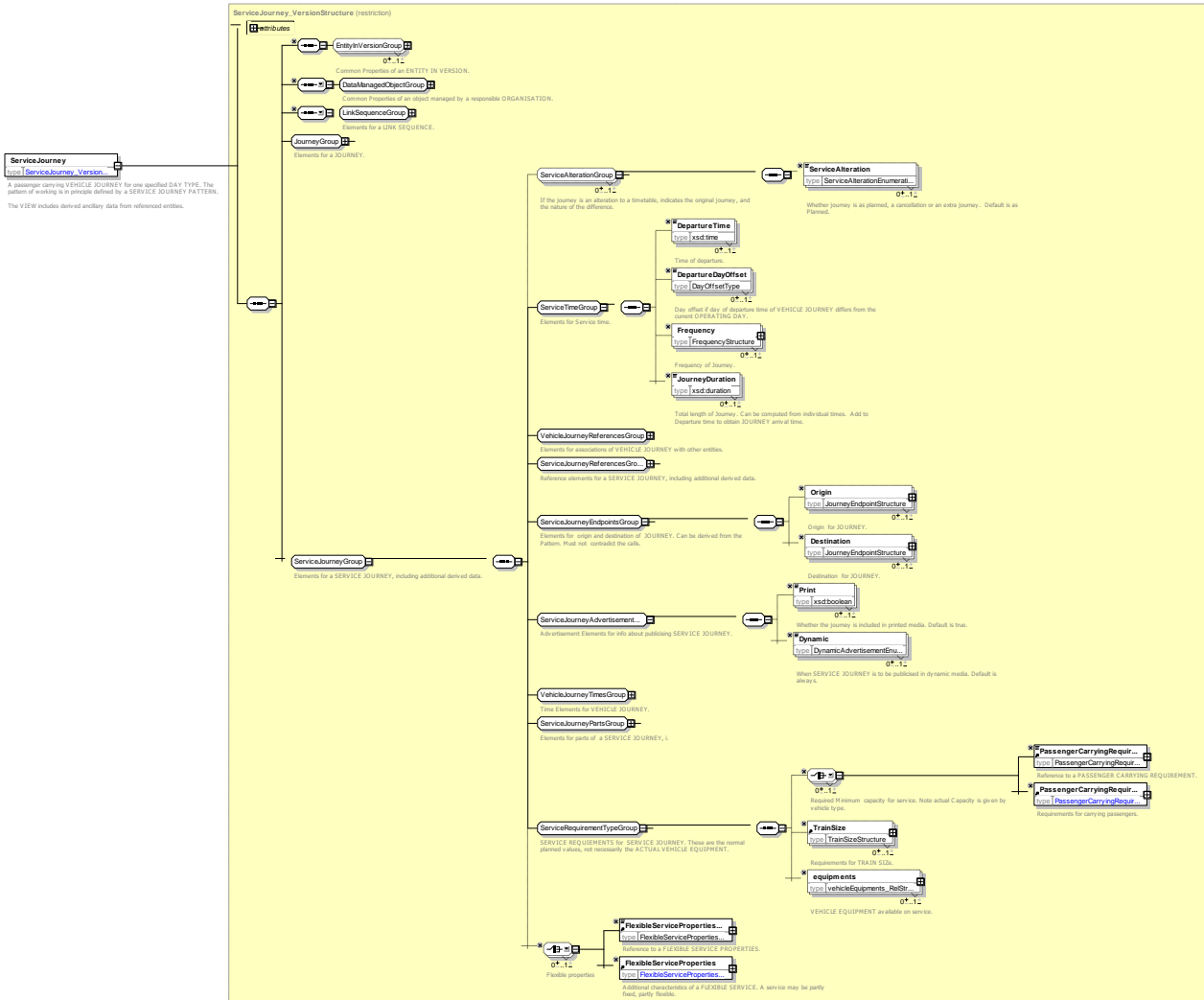


Figure 30 — ServiceJourney — XSD

7.2.2.4.2.1 ServiceAlteration – Allowed values

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

Table 15 – ServiceAlteration – Allowed values

Value	Description
<i>extraJourney</i>	Journey is a short term addition to the original planned schedule.
<i>cancellation</i>	Journey is a cancellation of a journey in the planned schedule.

<i>planned</i>	Journey is as planned.
----------------	------------------------

7.2.2.4.2.2 VehicleJourneyReferencesGroup – Model Element



Figure 31 — VehicleJourneyReferencesGroup — XSD

7.2.2.4.2.3 ServiceJourneyReferencesGroup – Model Element

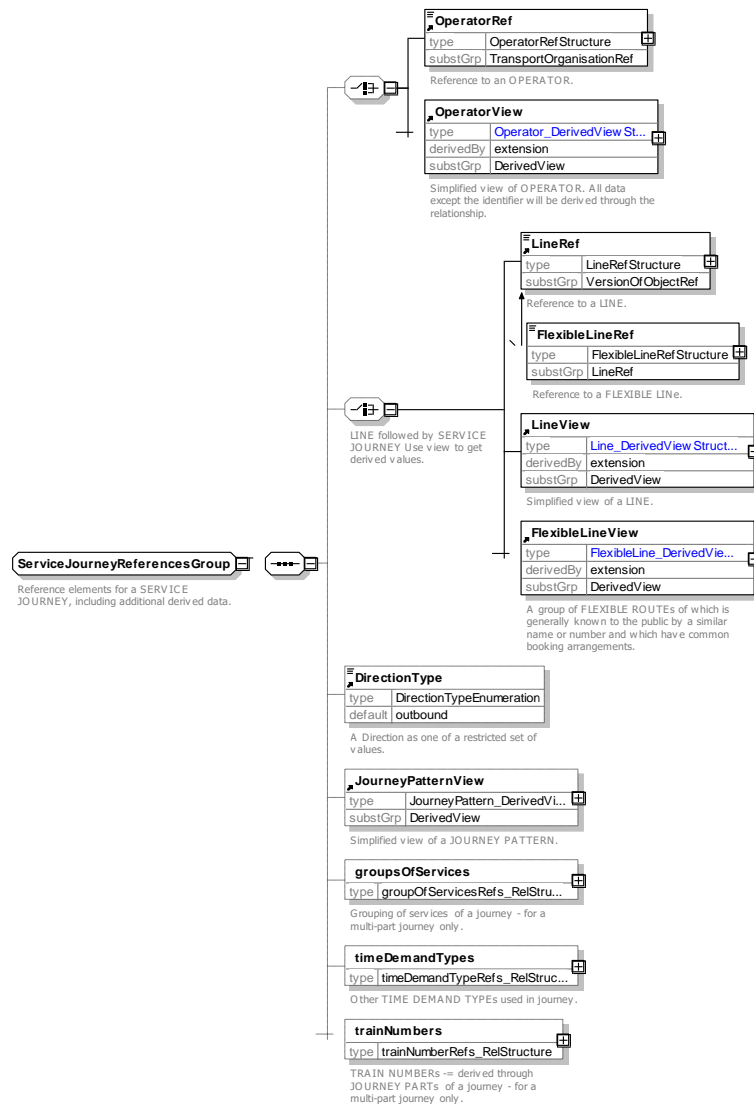


Figure 32 — ServiceJourneyReferencesGroup — XSD

7.2.2.4.2.4 DynamicAdvertisement – Allowed values

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

Table 16 – DynamicAdvertisement – Allowed values

Value	Description
<i>onlyIfSignedOn</i>	Only show journey in public displays if driver has signed in
<i>OnlyIfOrdered</i>	Only show journey in public displays if journey has been ordered
<i>always</i>	Always show journey in public displays
<i>never</i>	Never show journey in public displays

7.2.2.4.2.5 ServiceJourneyPartsGroup – Model Element

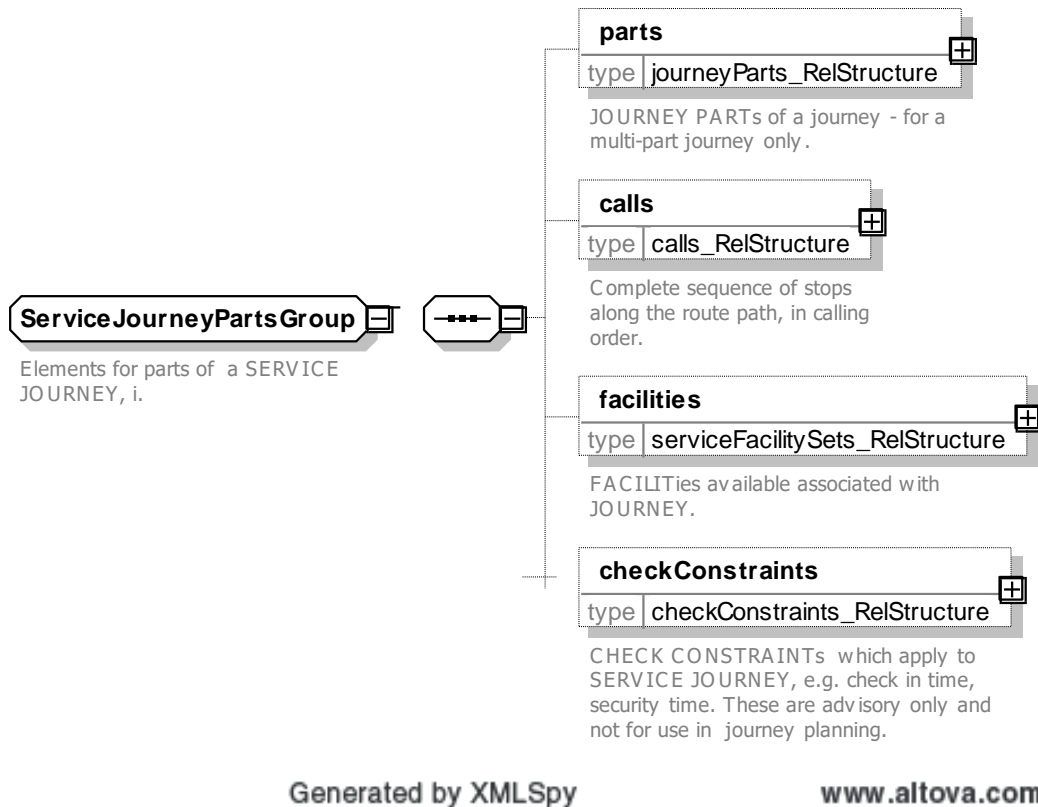


Figure 33 — ServiceJourneyPartsGroup — XSD

7.2.2.4.2.6 ServiceRequirementTypeGroup – Model Element

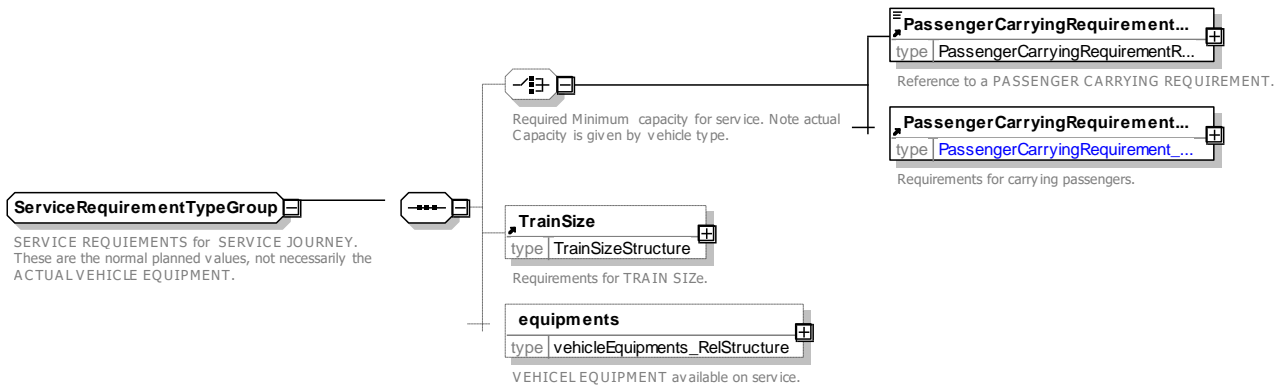


Figure 34 — ServiceRequirementTypeGroup — XSD

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 17 – SpecialService – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Journey</i>	::>	SPECIAL SERVICE inherits from JOURNEY.
«PK»	id	<i>SpecialServiceIdType</i>	1:1	Identifier of SPECIAL SERVICE.
	DepartureTime	<i>xsd:time</i>	1:1	Start time for SPECIAL SERVICE.
	DepartureDayOffset	<i>DayOffsetType</i>	0:1	Day offset if day of departure time of VEHICLE JOURNEY differs from the current OPERATING DAY.
«cntd»	Frequency	<i>FrequencyStructure</i>		Frequency of Journey.
	JourneyDuration	<i>xsd:duration</i>		Total length of Journey. Can be computed from individual times. Add to Departure time to obtain JOURNEY arrival time.
	Client	<i>xsd:normalizedString</i>	1:1	Client for SPECIAL SERVICE.
«cntd»	dayTypes	<i>DayTypeRef</i>	0:*	DAY TYPEs for SPECIAL SERVICE.
«FK»	JourneyPatternRef	<i>JourneyPatternRef</i>	0:1	JOURNEY PATTERN for SPECIAL SERVICE.
«FK»	VehicleTypeRef	<i>VehicleTypeRef</i>	0:1	VEHICLE TYPE for SPECIAL SERVICE.
«cntd»	Origin	<i>JourneyEndpoint</i>	0:1	Origin of SERVICE JOURNEY.
«cntd»	Destination	<i>JourneyEndpoint</i>	0:1	Destination of SERVICE JOURNEY.
	Print	<i>xsd:boolean</i>	0:1	Whether this journey should be visible to public in print channels.
	Dynamic	<i>DynamicAdvertisementEnum</i>	0:1	When this journey should be visible to public in dynamic channels.
«FK»	VehicleTypeRef	<i>VehicleTypeRef</i>	0:1	VEHICLE TYPE for SPECIAL SERVICE.
::	BookingArrangementsGroup	<i>BookingArrangements</i>	0:1	BOOKING ARRANGEMENTs for SPECIAL SERVICE

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.

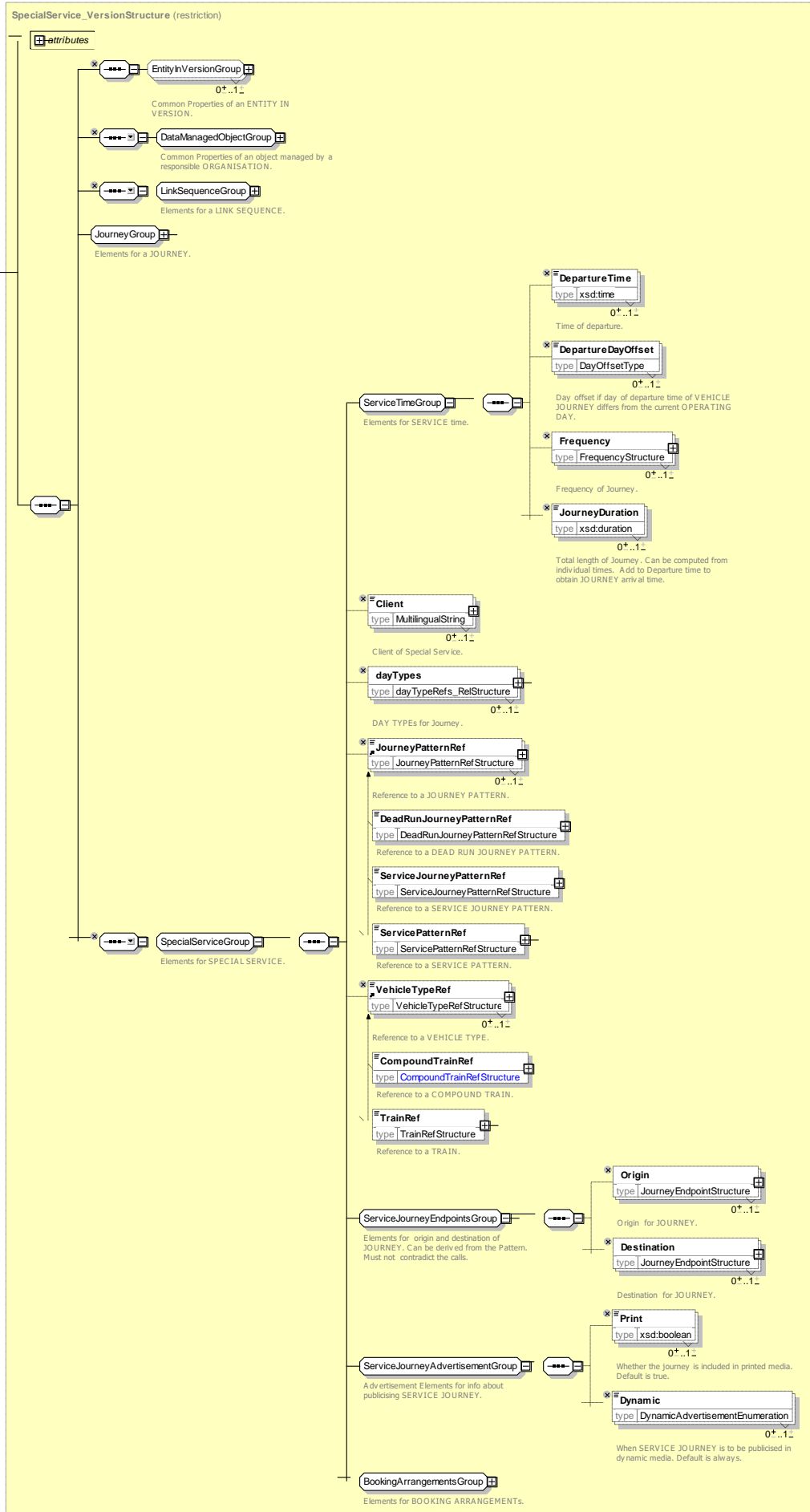


Figure 35 — *SpecialService* — XSD

7.2.2.4.3.1 **BookingArrangements – Group**

Information about booking a flexible journey. See also Part1 for details.

Table 18 – *BookingArrangements Group*– Group

Classification	Name	Type	Cardinality	Description
	BookingContact	<i>Contact</i>	0:1	Contact for Booking.
	BookingMethods	<i>BookingMethodEnum</i>	0:*	Booking method for FLEXIBLE LINE.
	BookingAccess	<i>BookingAccessEnum</i>	0:1	Who can make a booking.
	BookWhen	<i>PurchaseWhenEnumeration</i>	0:1	When Booking can be made.
	BuyWhen	<i>PurchaseMomentEnumeration</i>	0:*	When purchase can be made.
	LatestBooking-Time	<i>MultilingualString</i>	0:1	Latest time in day that booking can be made.
	MinimumBookingPeriod	<i>xsd:duration</i>	0:1	Minimum interval in advance of departure day or time that service may be ordered.
	BookingUrl	<i>xsd:anyURI</i>	0:1	URL for booking.
	BookingNote	<i>MultilingualString</i>	0:1	Note about booking the FLEXIBLE LINE.



Figure 36 — *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 19 – *TemplateServiceJourney* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>TemplateVehicleJourney</i>	::>	TEMPLATE SERVICE JOURNEY inherits from TEMPLATE VEHICLE JOURNEY.

«PK»	<i>id</i>	<i>TemplateServiceJourney-IdType</i>	0:1	Identifier of TEMPLATE SERVICE JOURNEY.
------	-----------	--------------------------------------	-----	---

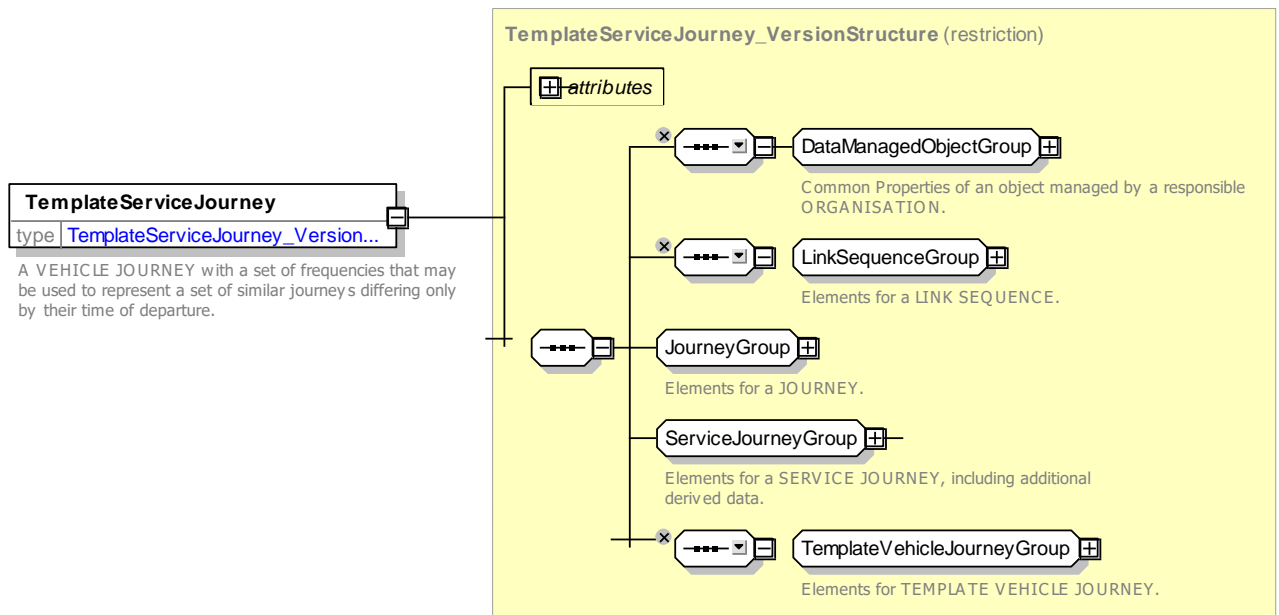


Figure 37 — TemplateServiceJourney — XSD

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 LAY-OVER.

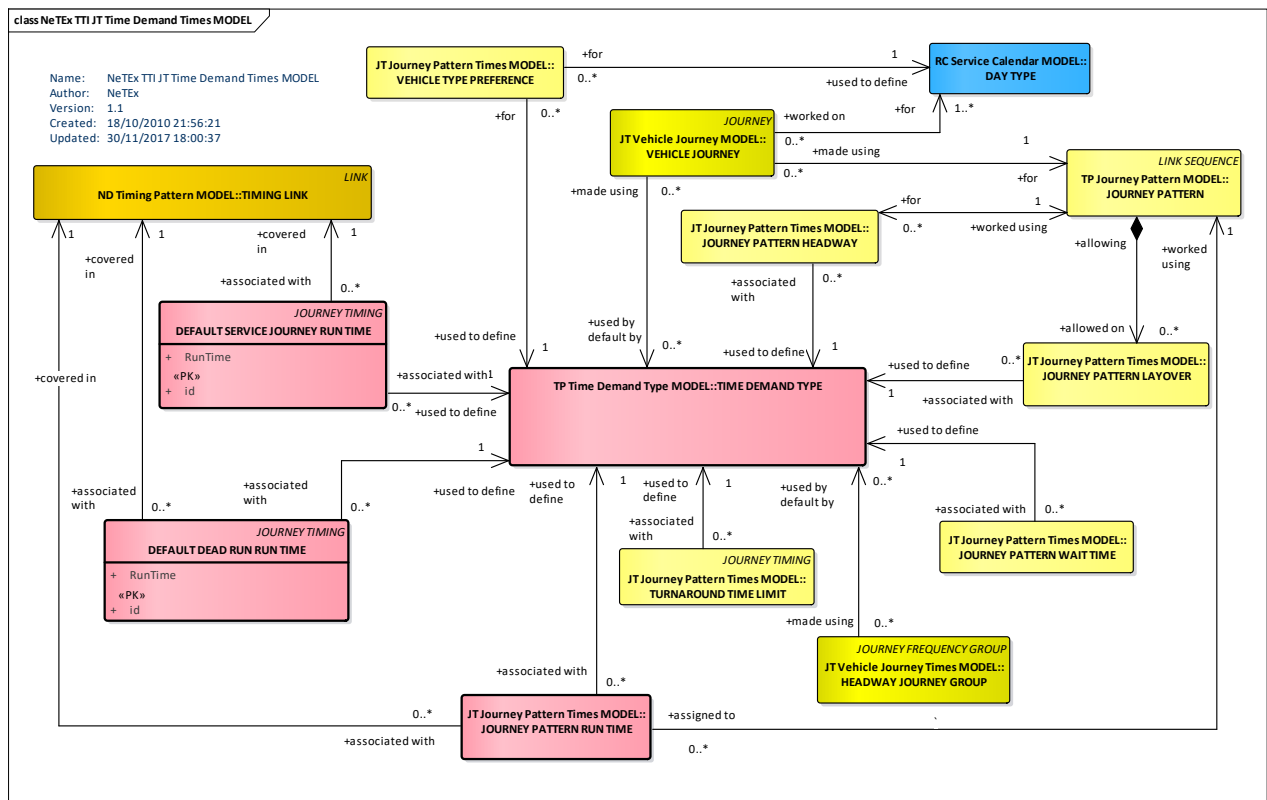


Figure 38 —Time Demand Times – Conceptual MODEL (UML)

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.

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 is specialised to create a number of different types of timing.

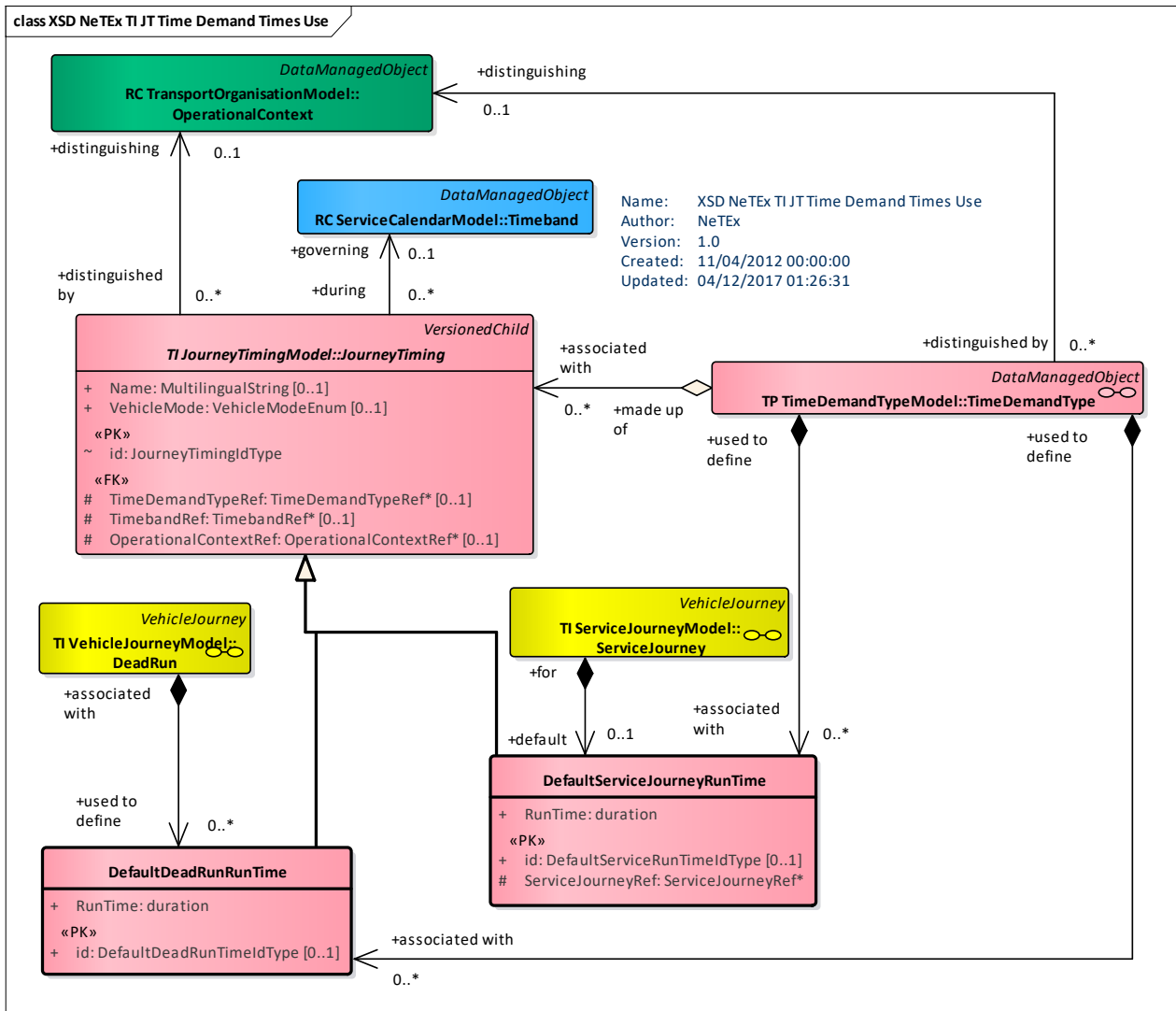


Figure 40 — Time Demand Times Use – Physical Model (UML)

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 20 – DefaultServiceJourneyRunTime – Element

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

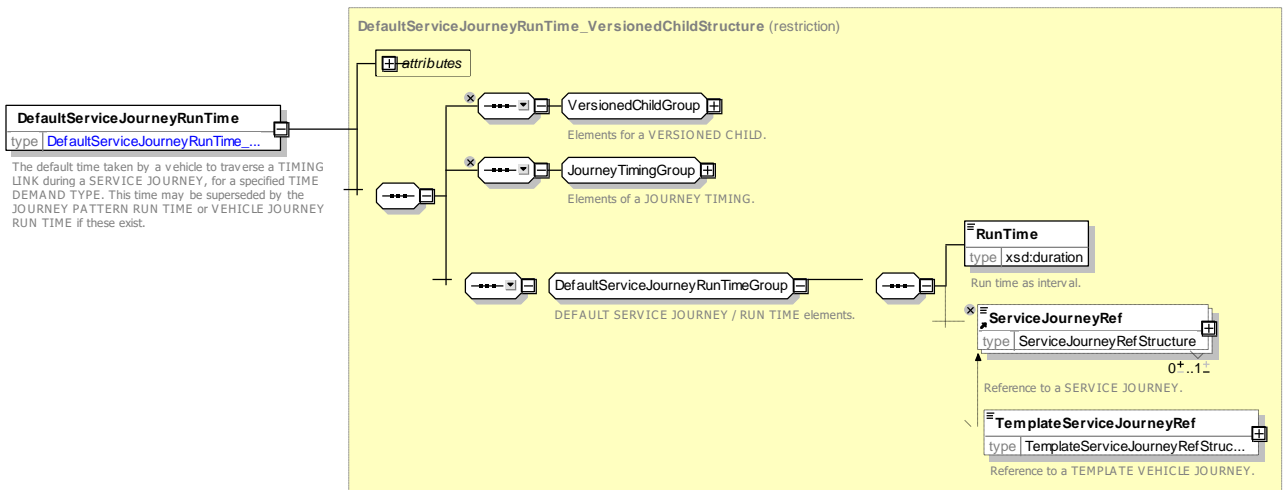


Figure 41 — DefaultServiceJourneyRunTime — XSD

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 21 – DefaultDeadRunRunTime – Element

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

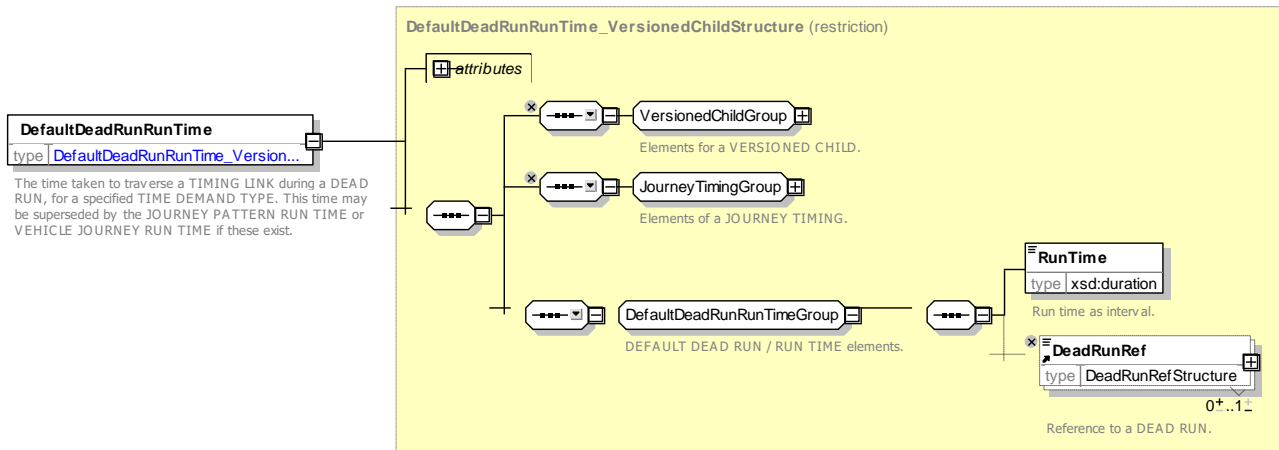


Figure 42 — *DefaultDeadRunRunTimeTime* — XSD

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.

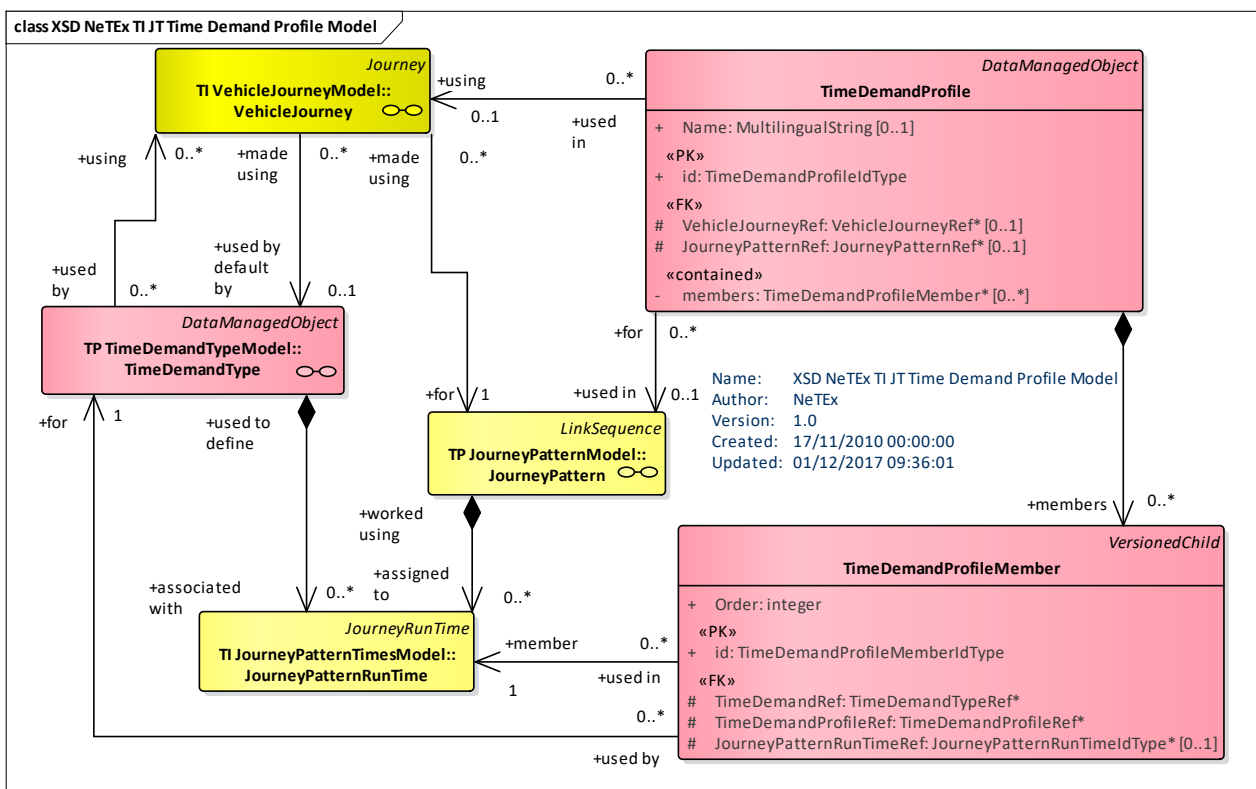


Figure 43 — Time Demand Profile – Physical Model (UML)

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 22 – TimeDemandProfile – Element

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

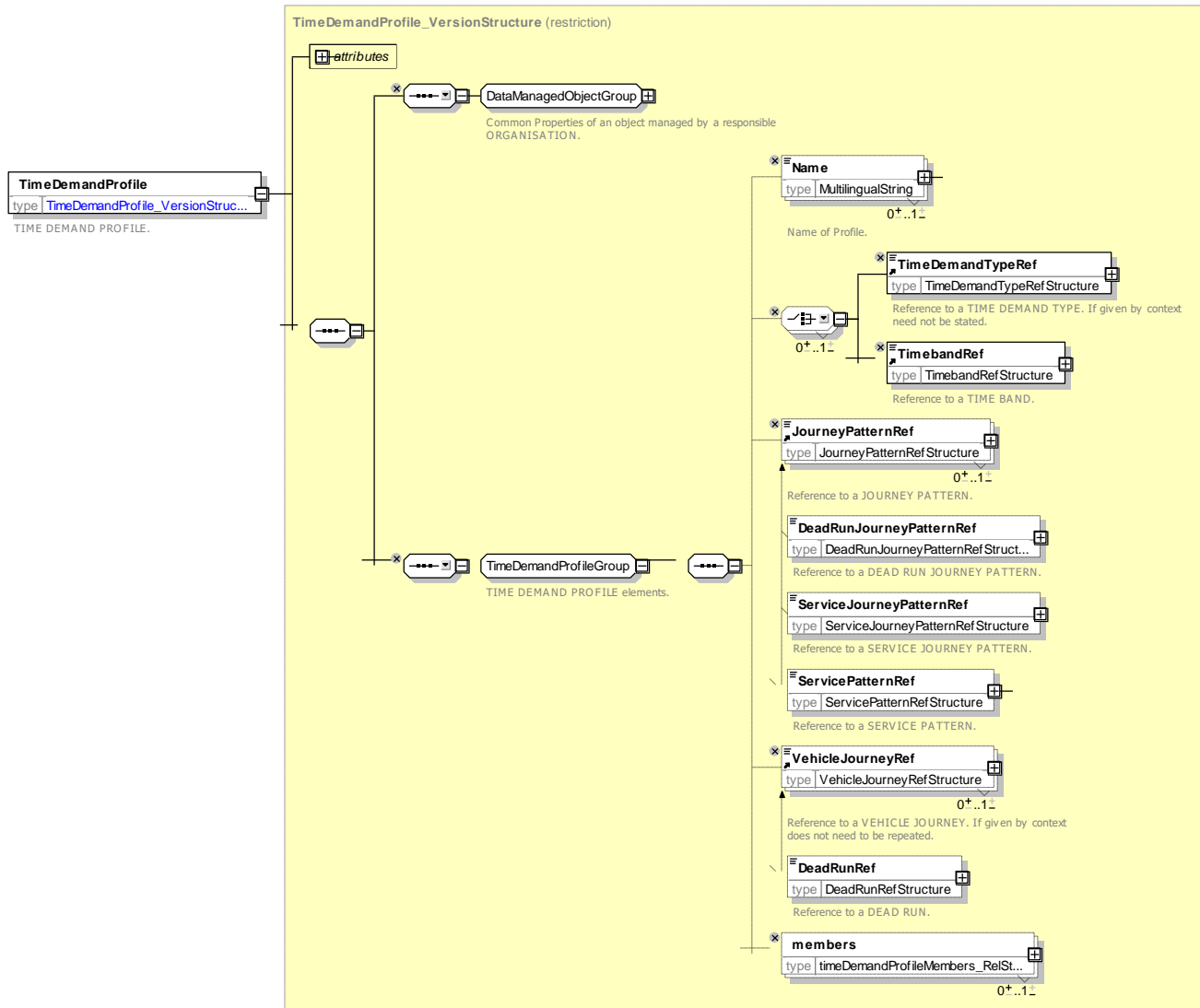


Figure 44 — TimeDemandProfile — XSD

7.2.3.5.2 TimeDemandProfileMember – Model Element

A members of the TIME DEMAND PROFILE described above.

Table 23 – TimeDemandProfileMember – Element

Classification	Name	Type	Cardinality	Description
::>	::>	GroupMember	::>	TIME DEMAND PROFILE inherits from GROUP MEMBER.
«PK»	id	TimeDemandProfileMemberIdType	1:1	Identifier of TIME DEMAND PROFILE MEMBER.
	order	xsd:integer	1:1	Order of member within profile.
«FK»	TimeDemandProfileRef	TimeDemandProfileRef	1:1	TIME DEMAND PROFILE of PROFILE MEMBER.
«PK»	TimeDemandRef	TimeDemandTypeRef	1:1	TIME DEMAND TYPE of PROFILE MEMBER.

«FK»	JourneyPattern-RunTimeRef	<i>JourneyPatternRunTimeRef</i>	0:1	JOURNEY PATTERN RUN TIME of PROFILE MEMBER.
------	----------------------------------	---------------------------------	-----	---

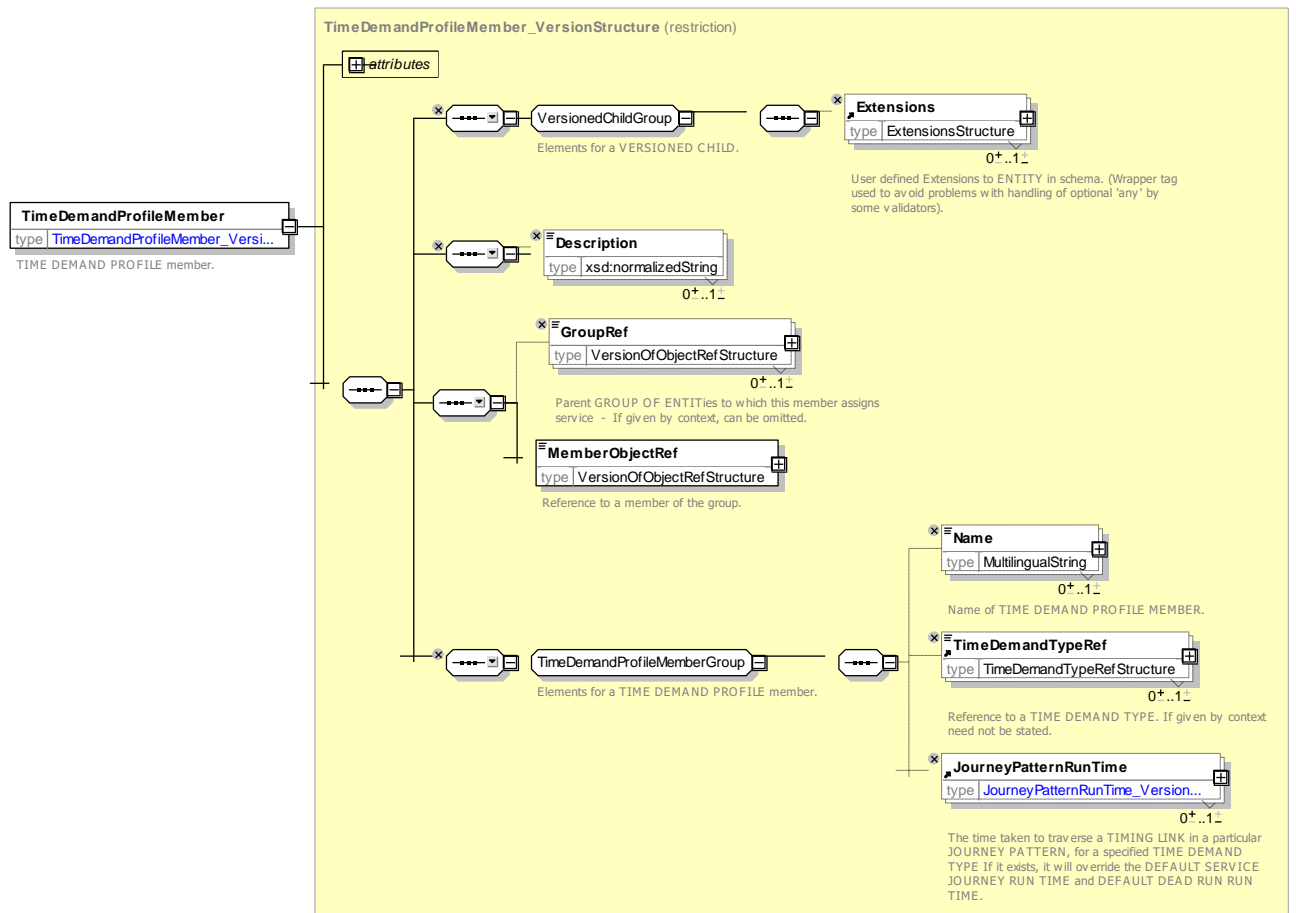


Figure 45 — TimeDemandProfileMember — XSD

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"/>
      </JourneyRunTime>
    </runTimes>
  </TimeDemandType>
</timeDemandTypes>

```

```
<TimingLinkRef version="any" ref="kx:tl_ssp_077_to_ssp_002"/>
<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.

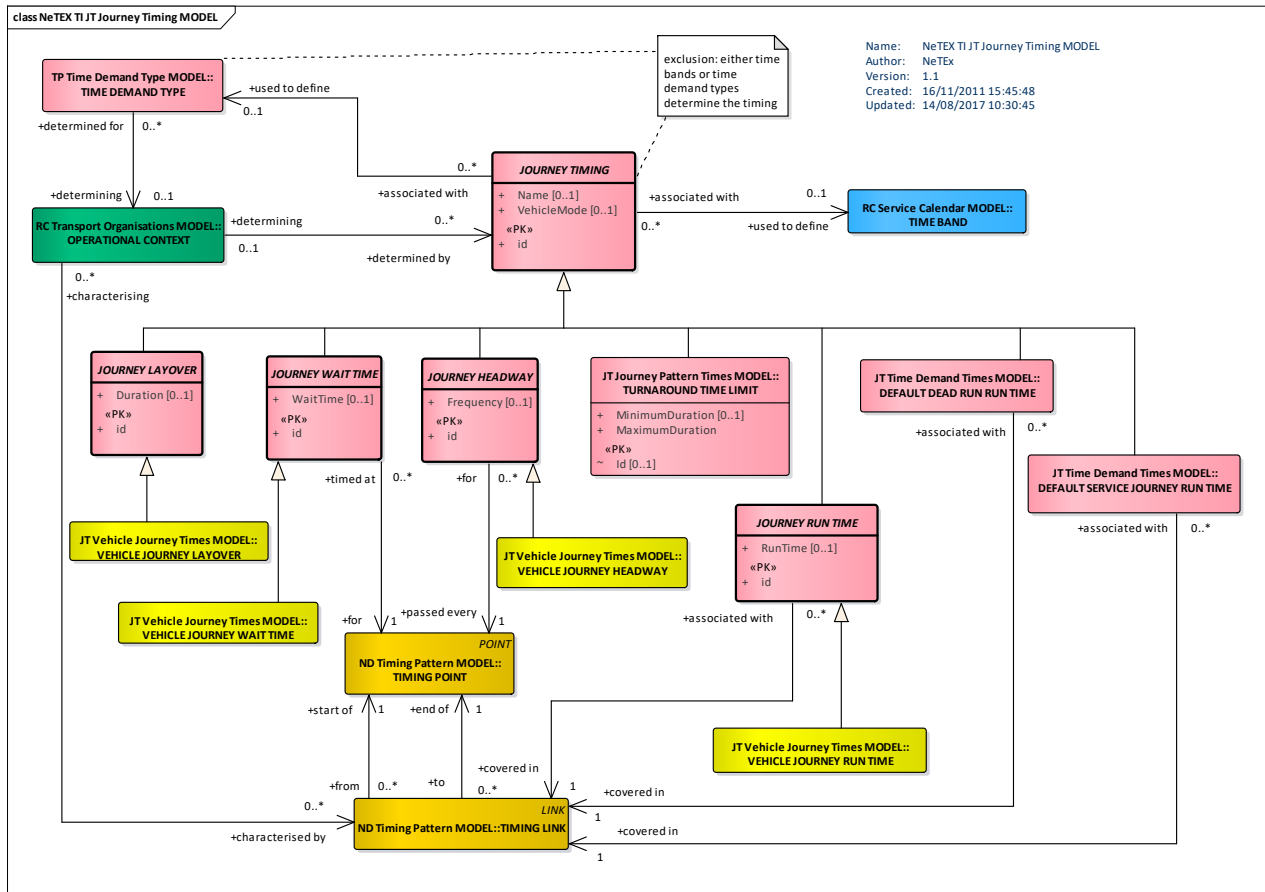


Figure 46 — Journey Timing – Conceptual MODEL (UML)

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.

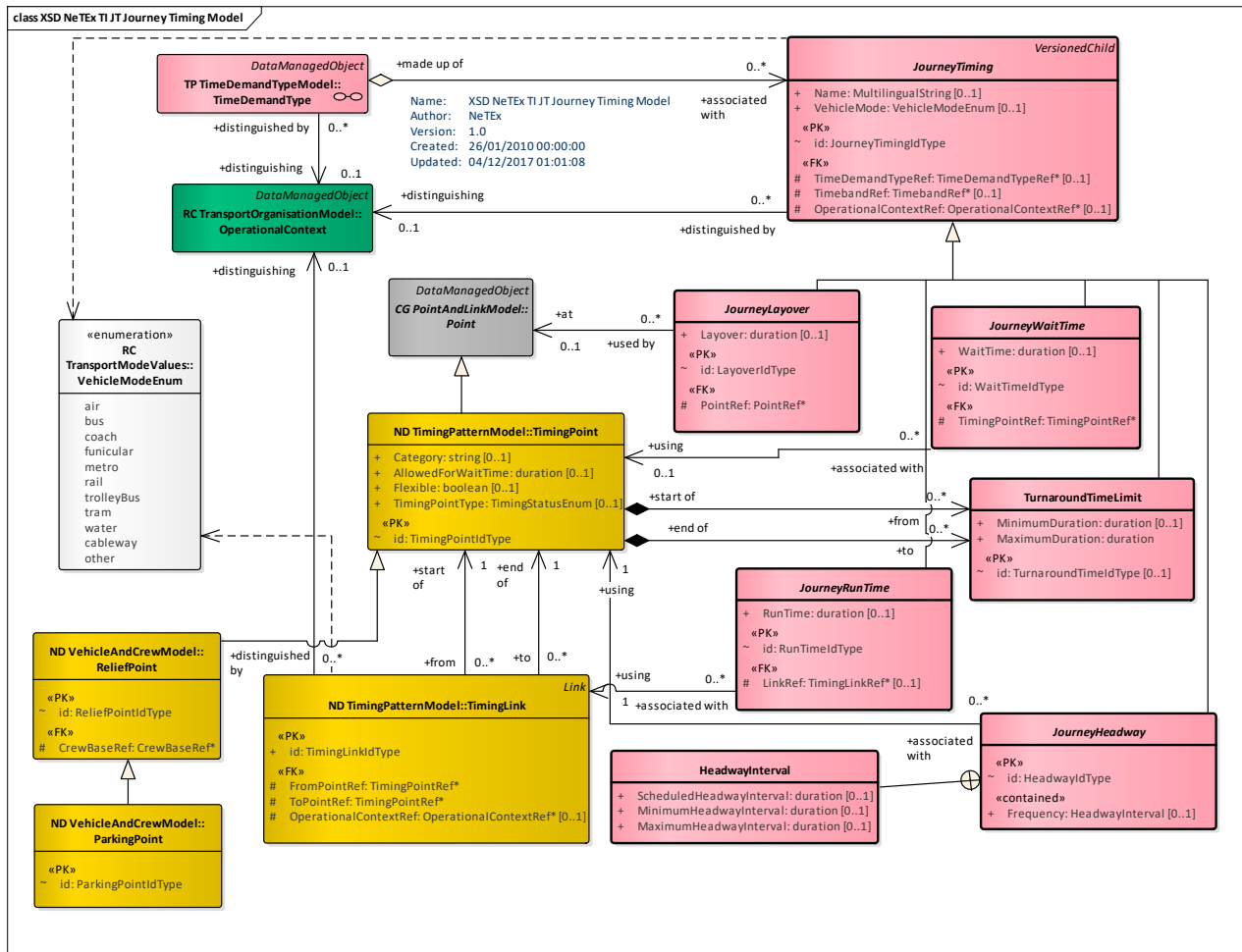


Figure 47 — Journey Timing – Physical Model (UML)

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 24 – JourneyWaitTime – Element

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

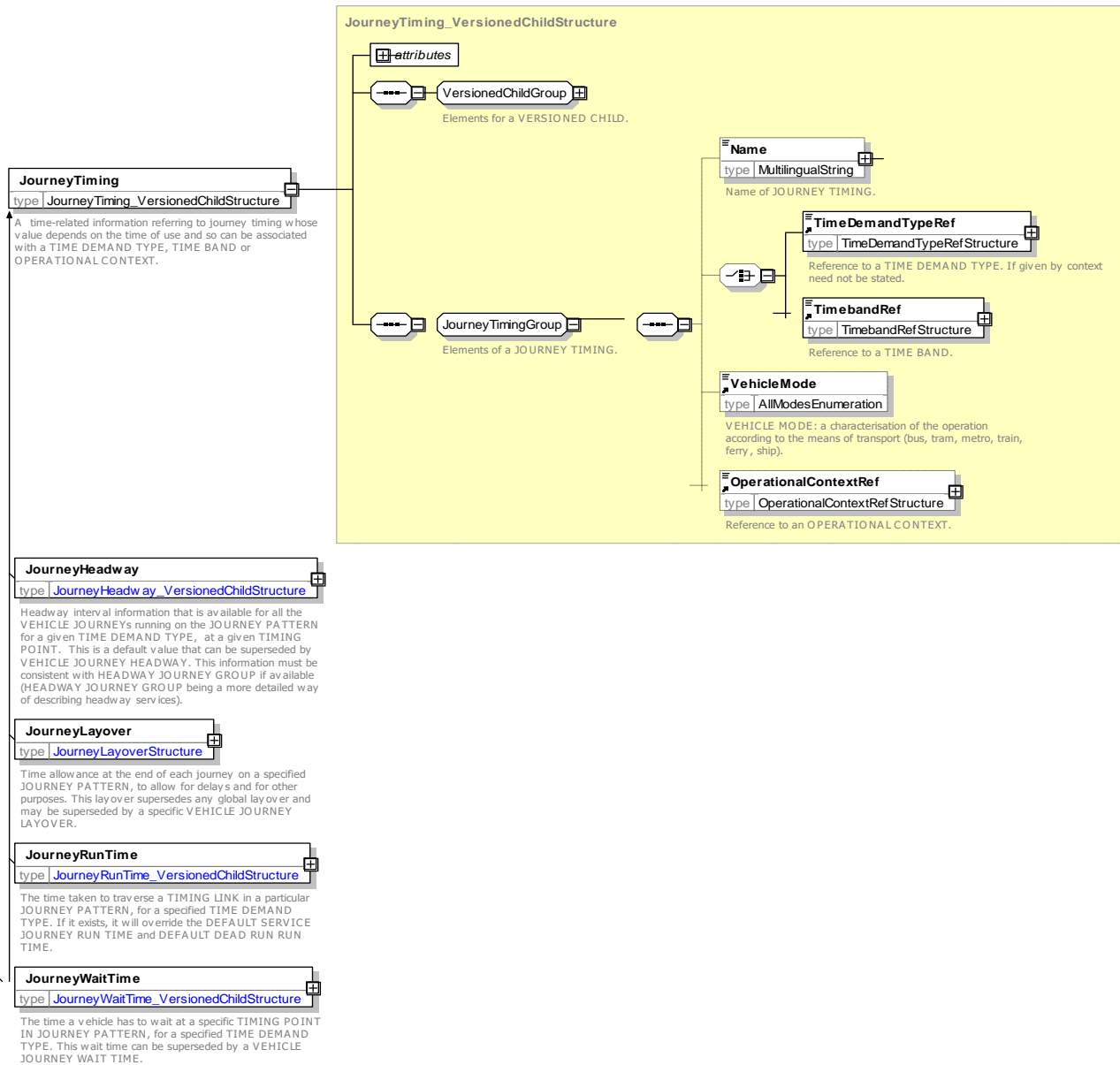


Figure 48 — JourneyWaitTime — XSD

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 25 – JourneyRunTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>JourneyTiming</i>	::>	JOURNEY RUN TIME inherits from JOURNEY TIMING.
«PK»	<i>id</i>	<i>RunTimeIdType</i>	1:1	Identifier of JOURNEY RUN TIME.

«FK»	TimingLinkRef	<i>TimingLinkRef</i>	0:1	TIMING LINK associated with timing.
	RunTime	<i>xsd:duration</i>	1:1	Run time of JOURNEY RUN TIME.

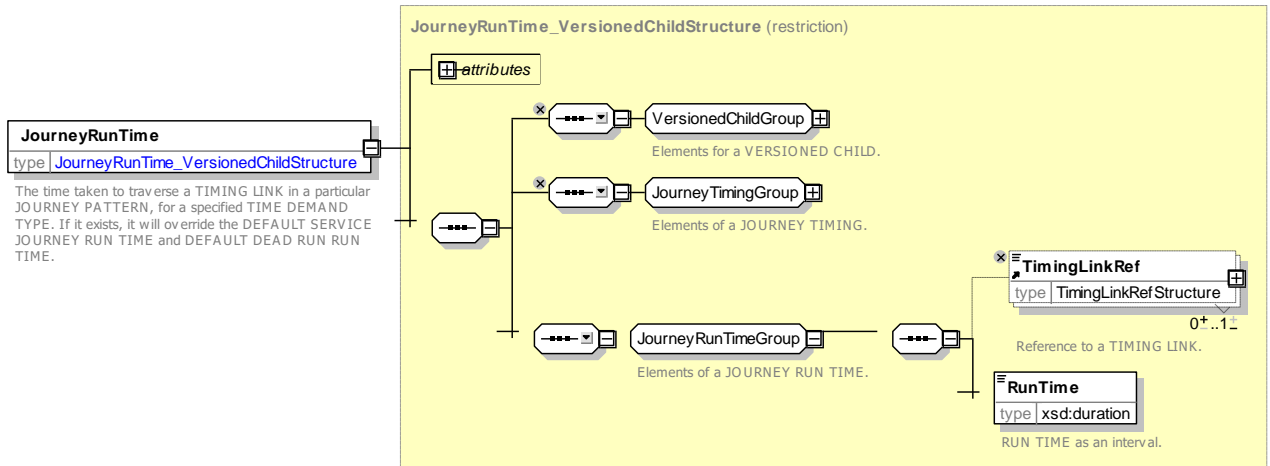


Figure 49 — JourneyRunTime — XSD

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 26 – JourneyLayover – Element

Classi- fication	Name	Type	Cardi- nality	Description
::>	::>	<i>JourneyTiming</i>	::>	JOURNEY LAYOVER inherits from JOURNEY TIMING.
«PK»	id	<i>LayoverIdType</i>	1:1	Identifier of JOURNEY LAYOVER.
	Layover	<i>xsd:duration</i>	0:1	Time spent on JOURNEY LAYOVER.
«FK»	PointRef	<i>PointRef</i>	1:1	POINT associated with of JOURNEY LAYOVER.

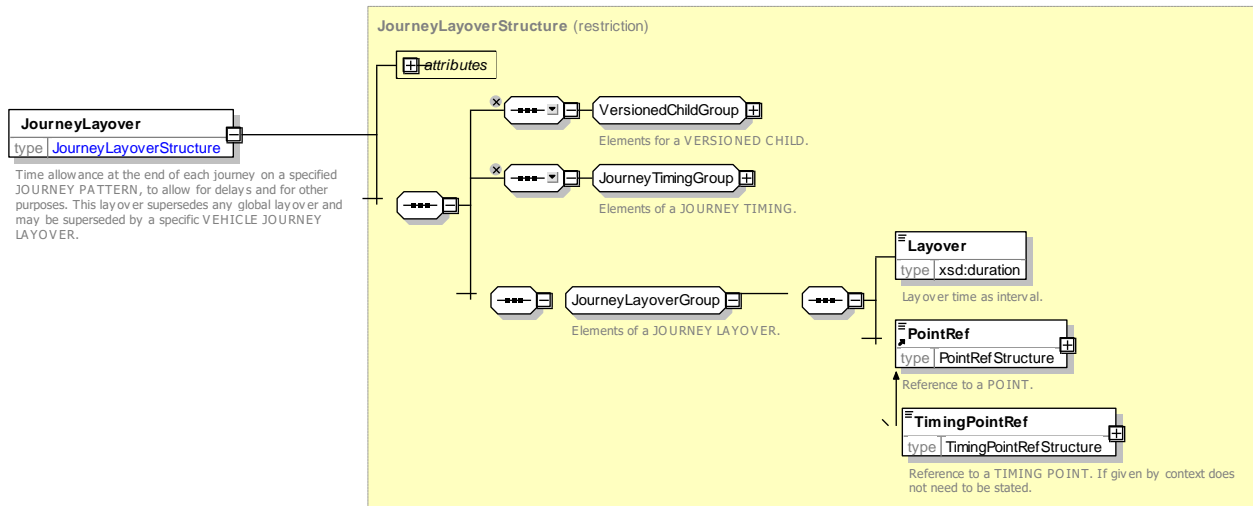


Figure 50 — JourneyLayover — XSD

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 27 – JourneyHeadway – Element

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

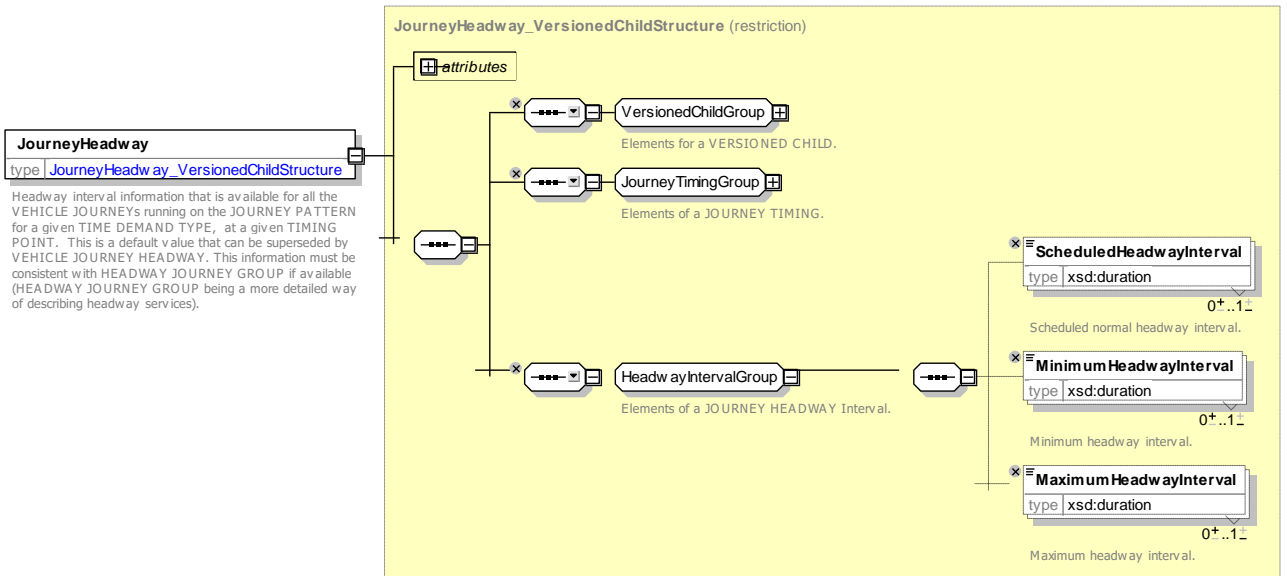


Figure 51 — JourneyHeadway — XSD

7.2.4.3.5 HeadwayInterval – Model Group

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

Table 28 – HeadwayInterval – Group

Classification	Name	Type	Cardinality	Description
	ScheduledHeadwayInterval	<i>xsd:duration</i>	0:1	Scheduled HEADWAY INTERVAL.
	MinimumHeadwayInterval	<i>xsd:duration</i>	0:1	Minimum HEADWAY INTERVAL.
	MaximumHeadwayInterval	<i>xsd:duration</i>	0:1	Maximum HEADWAY INTERVAL.

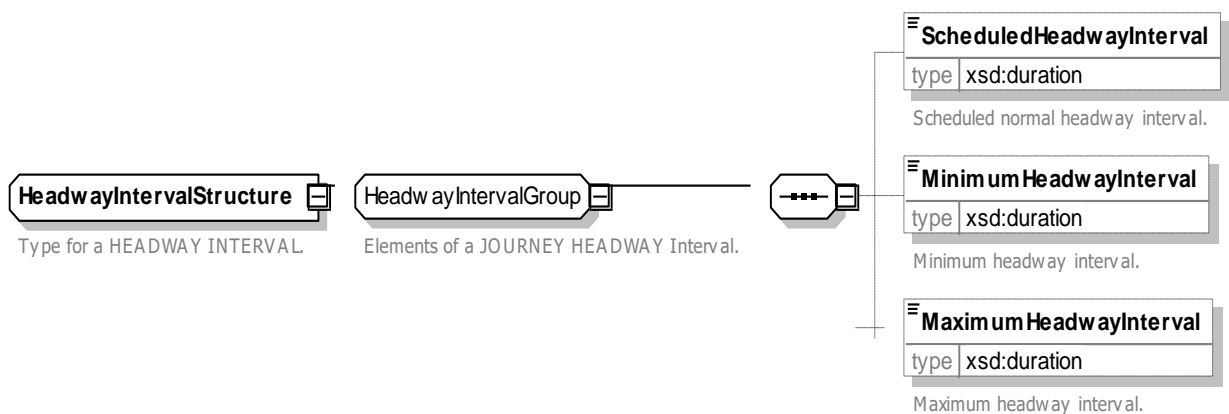


Figure 52 — HeadwayInterval — XSD

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 29 – TurnaroundTimeLimitTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>JourneyTiming</i>	::>	TURNAROUND TIME LIMIT inherits from JOURNEY TIMING.
«PK»	id	<i>TurnaroundTimeLimitIdType</i>	0:1	Identifier of TURNAROUND TIME LIMIT.
	MinimumDuration	<i>xsd:duration</i>	0:1	Minimum time needed for turnaround.
	MaximumDuration	<i>xsd:duration</i>	1:1	Maximum time needed for turnaround.

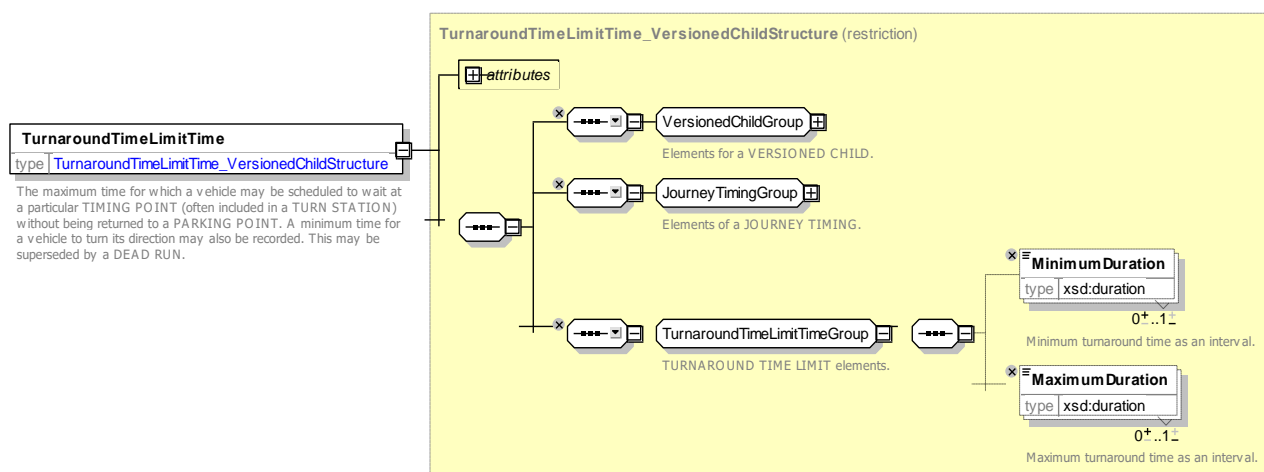


Figure 53 — TurnaroundTimeLimitTime — XSD

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.

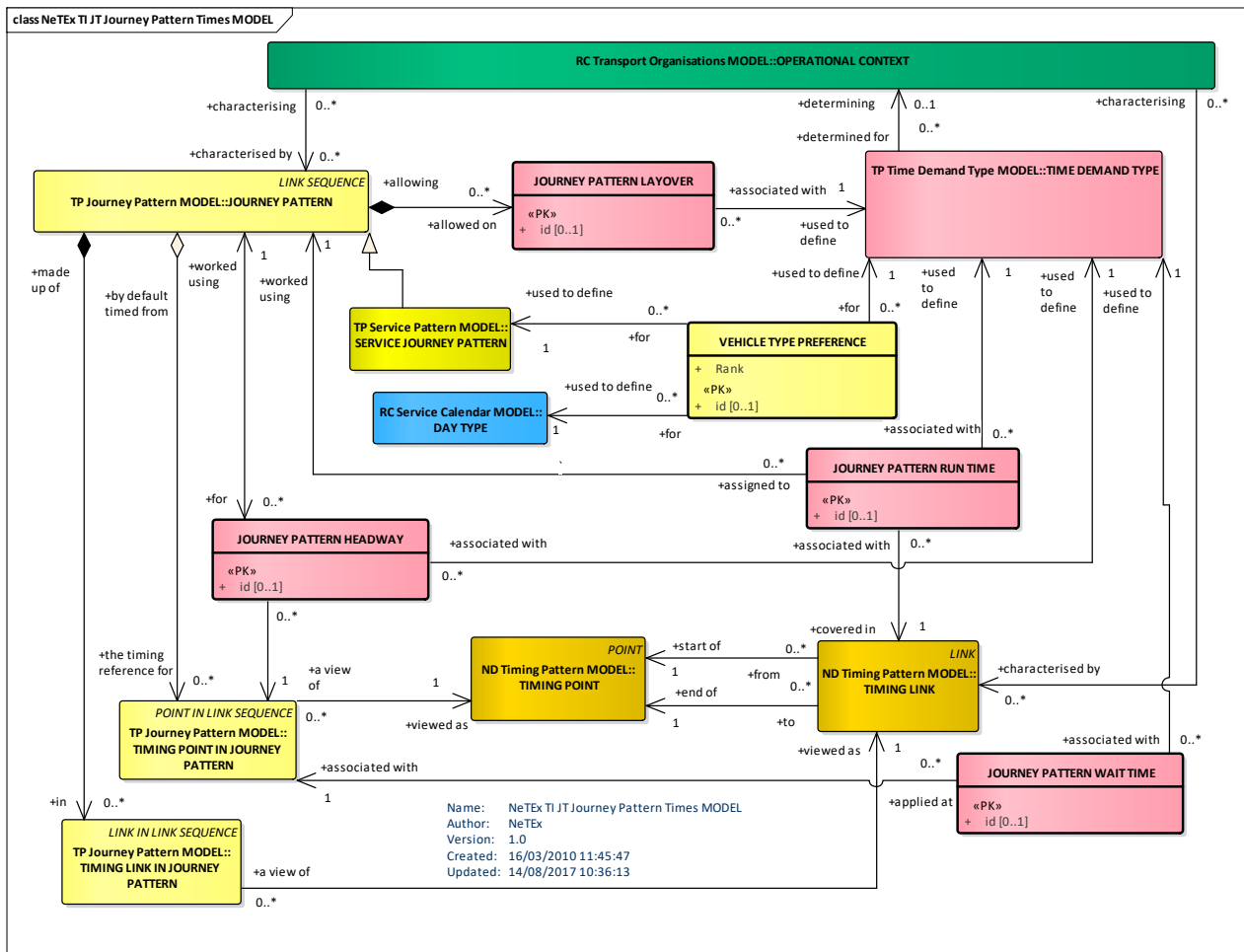


Figure 54 — Journey Pattern Times – Conceptual MODEL (UML)

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.

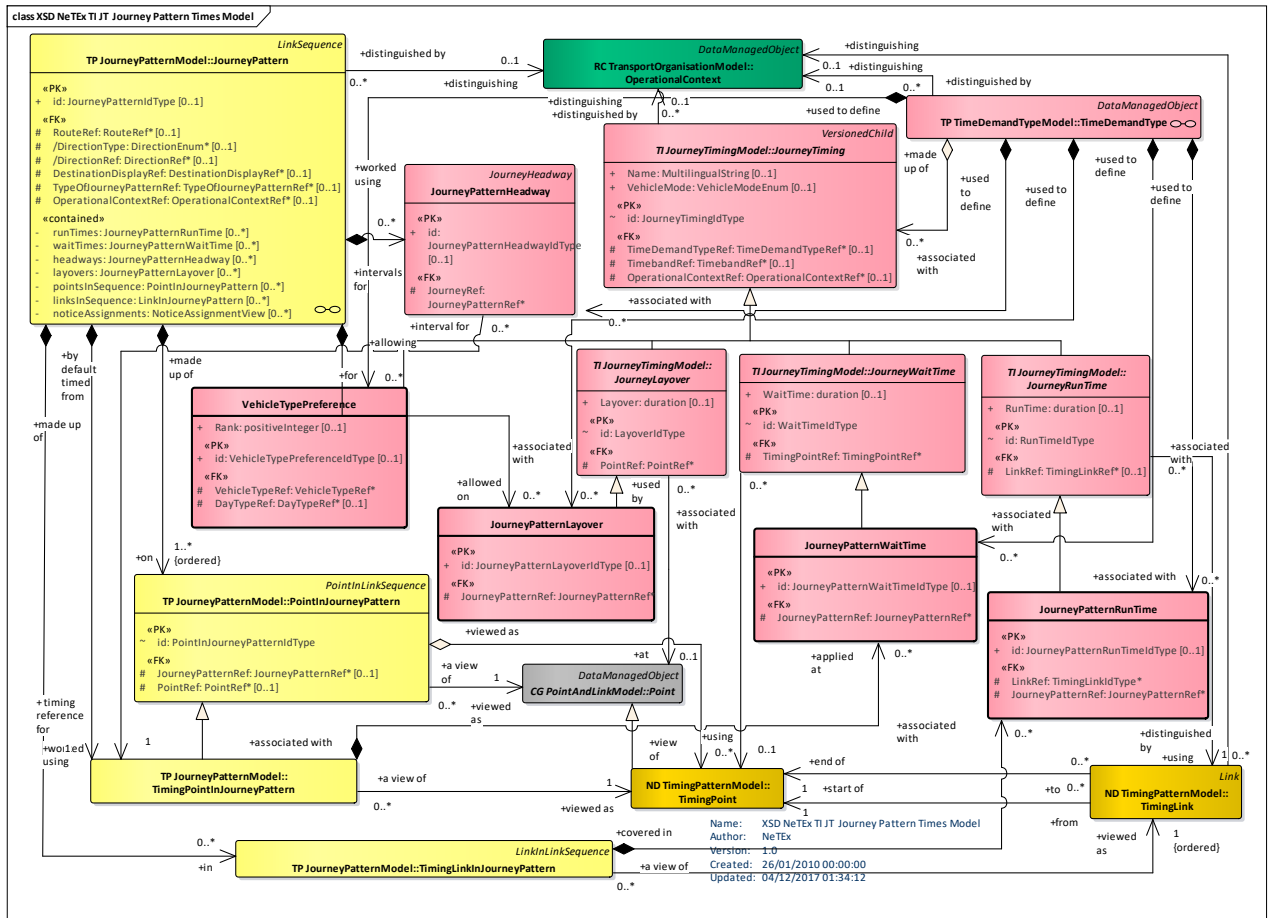


Figure 55 — Journey Pattern Times – Physical Model (UML)

7.2.5.2.1 Journey Pattern Time hierarchies – Physical Model

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

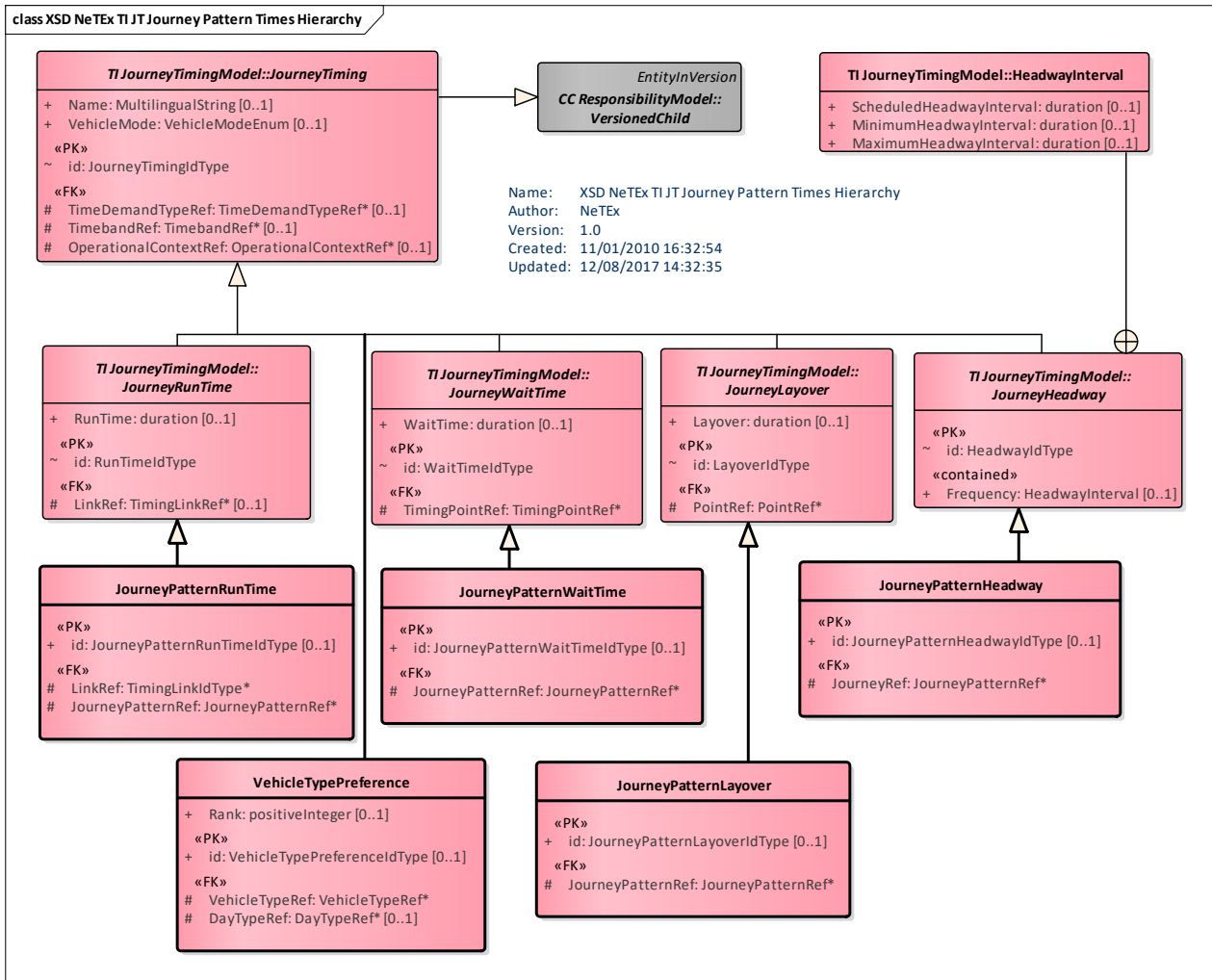


Figure 56 — Journey Pattern Times Hierarchy – Physical Model (UML)

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 30 – JourneyPatternWaitTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>JourneyWaitTime</i>	::>	JOURNEY PATTERN WAIT TIME inherits from JOURNEY WAIT TIME.
«PK»	id	<i>JourneyPatternWaitTimeIdType</i>	0:1	Identifier of JOURNEY PATTERN WAIT TIME.
«FK»	JourneyRef	<i>JourneyPatternRef</i>	1:1	JOURNEY PATTERN with which WAIT TIME is associated.

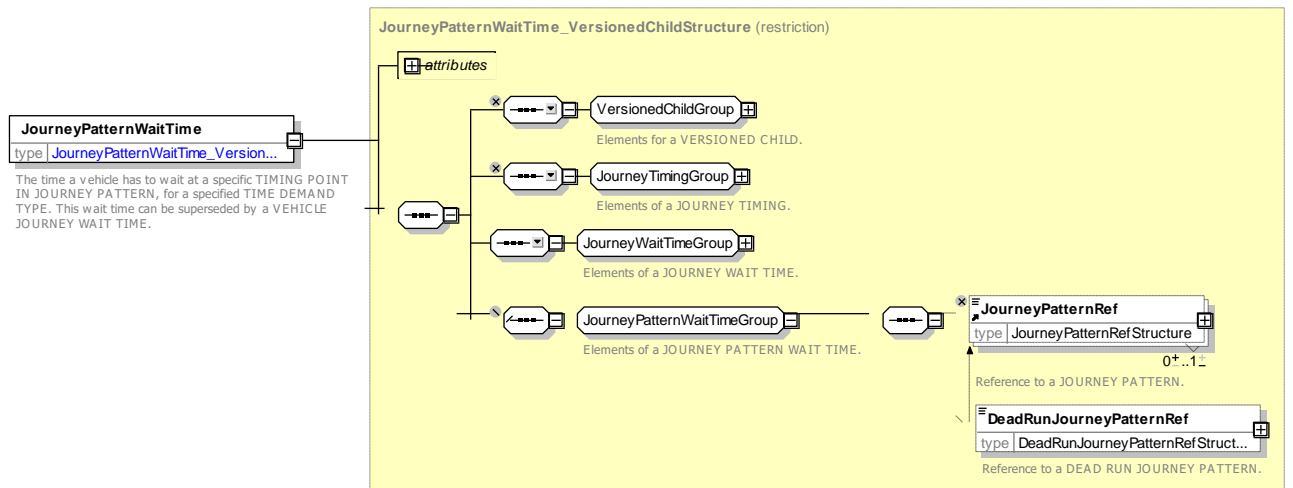


Figure 57 — *JourneyPatternWaitTime* — XSD

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 31 – *JourneyPatternRunTime* – Element

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

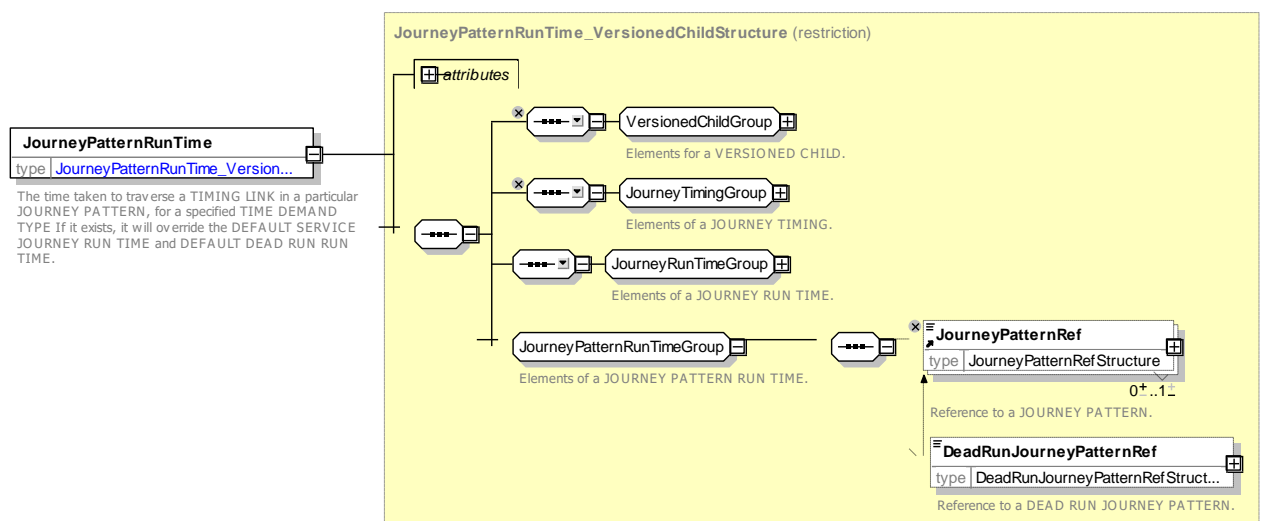


Figure 58 — *JourneyPatternRunTime* — XSD

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 32 – *JourneyPatternLayover* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>JourneyLayoverTime</i>	::>	JOURNEY PATTERN LAYOVER inherits from JOURNEY LAYOVER.
«PK»	<i>id</i>	<i>JourneyPatternLayoverIdType</i>	0:1	Identifier of JOURNEY PATTERN LAYOVER.
«FK»	<i>Journey-PatternRef</i>	<i>JourneyPatternRef</i>	1:1	JOURNEY PATTERN with which LAYOVER is associated.

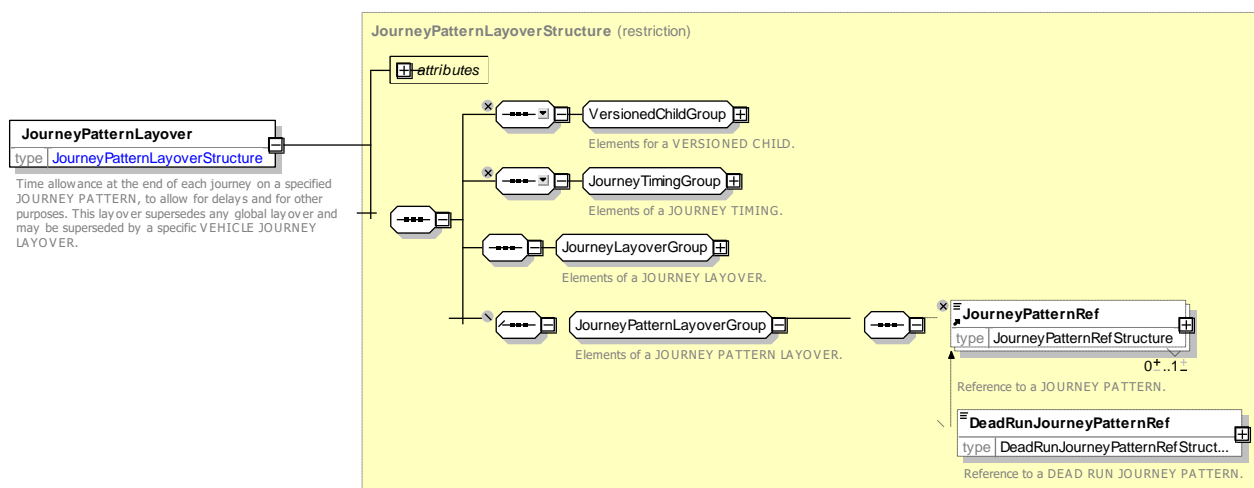


Figure 59 — *JourneyPatternLayover* — XSD

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 33 – *JourneyPatternHeadway* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>TimeDemandHeadway</i>	::>	JOURNEY PATTERN HEADWAY inherits from TIME DEMAND HEADWAY.

«PK»	id	<i>JourneyPatternHeadwayIdType</i>	0:1	Identifier of JOURNEY PATTERN HEADWAY.
«FK»	JourneyRef	<i>JourneyPatternRef</i>	1:1	JOURNEY PATTERN with which HEADWAY is associated.

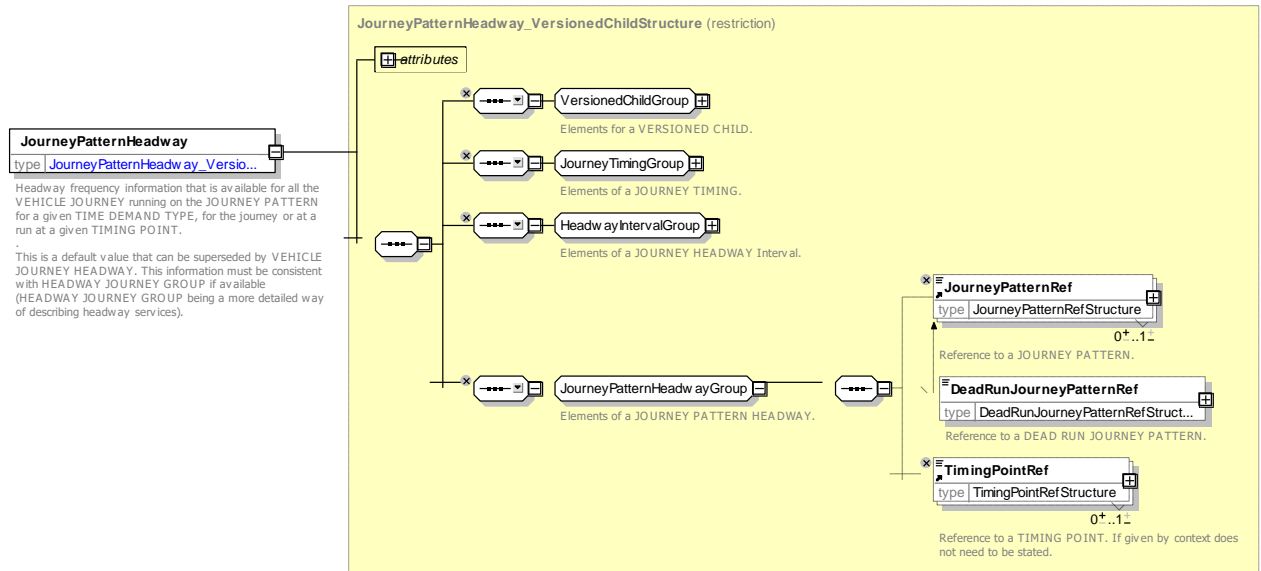


Figure 60 — *JourneyPatternHeadway* — XSD

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 34 – *VehicleTypePreference* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>TimeDemandTiming</i>	::>	JOURNEY PATTERN HEADWAY inherits from TIME DEMAND TIMING.
«PK»	id	<i>VehicleTypePreferenceIdType</i>	0:1	Identifier of VEHICLE TYPE PREFERENCE.
	Rank	<i>xsd:positiveInteger</i>	1:1	Ranking of VEHICLE TYPE PREFERENCE.
«FK»	DayTypeRef	<i>DayTypeRef</i>	0:1	DAY TYPE with which PREFERENCE is associated.
«FK»	VehicleTypeRef	<i>VehicleTypeRef</i>	1:1	VEHICLE TYPE with which PREFERENCE is associated.

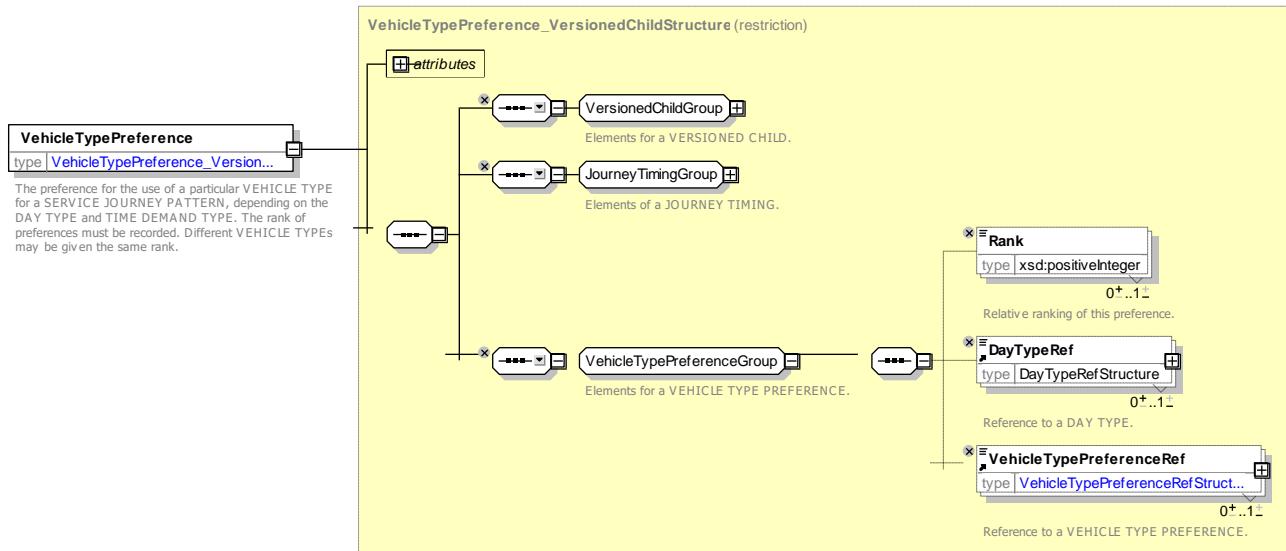


Figure 61 — VehicleTypePreference — XSD

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).

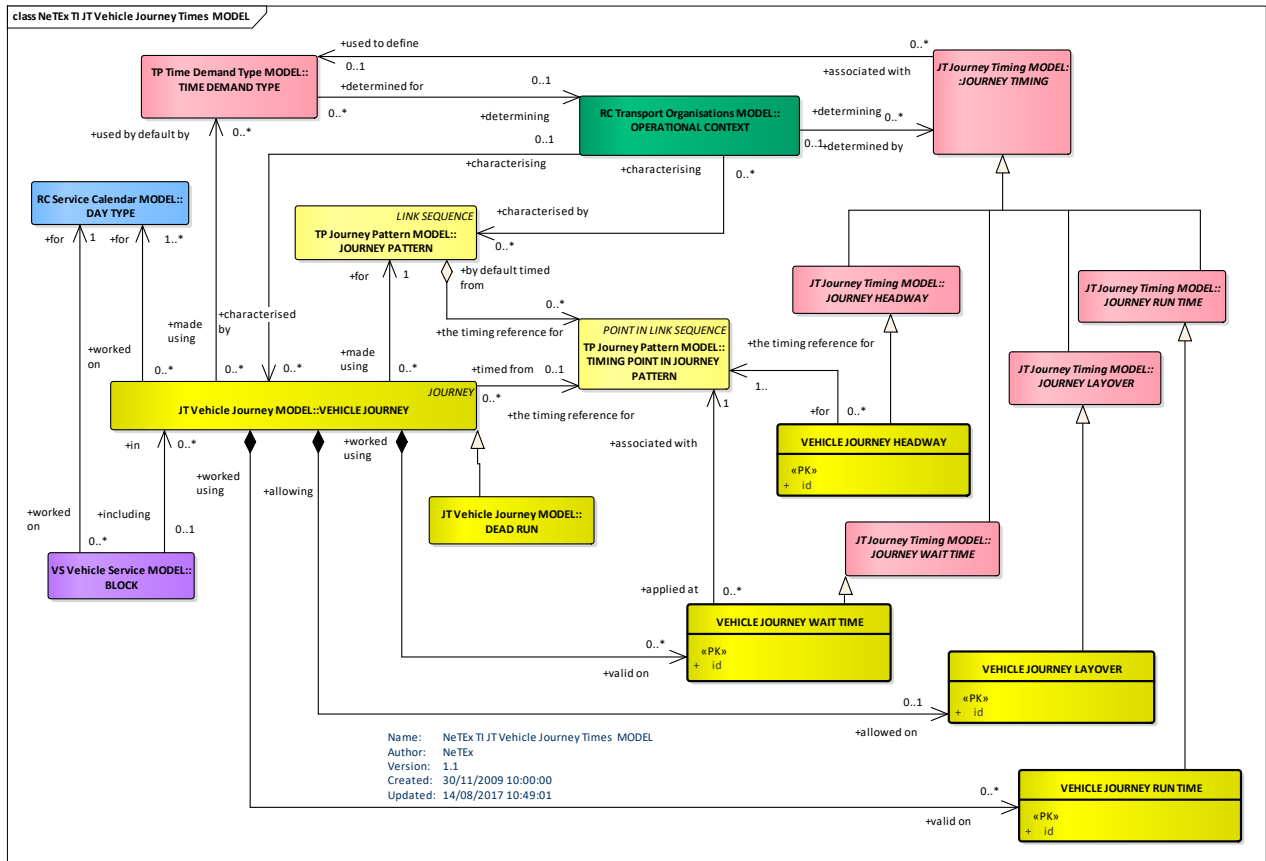


Figure 62 — Vehicle Journey Times — Conceptual MODEL (UML)

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 'xhx10', 'xhx25', 'xhx45' past the hour. In either case a TEMPLATE VEHICLE JOURNEY is associated with a JOURNEY FREQUENCY GROUP to represent a group of JOURNEYS running at the given frequency.

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

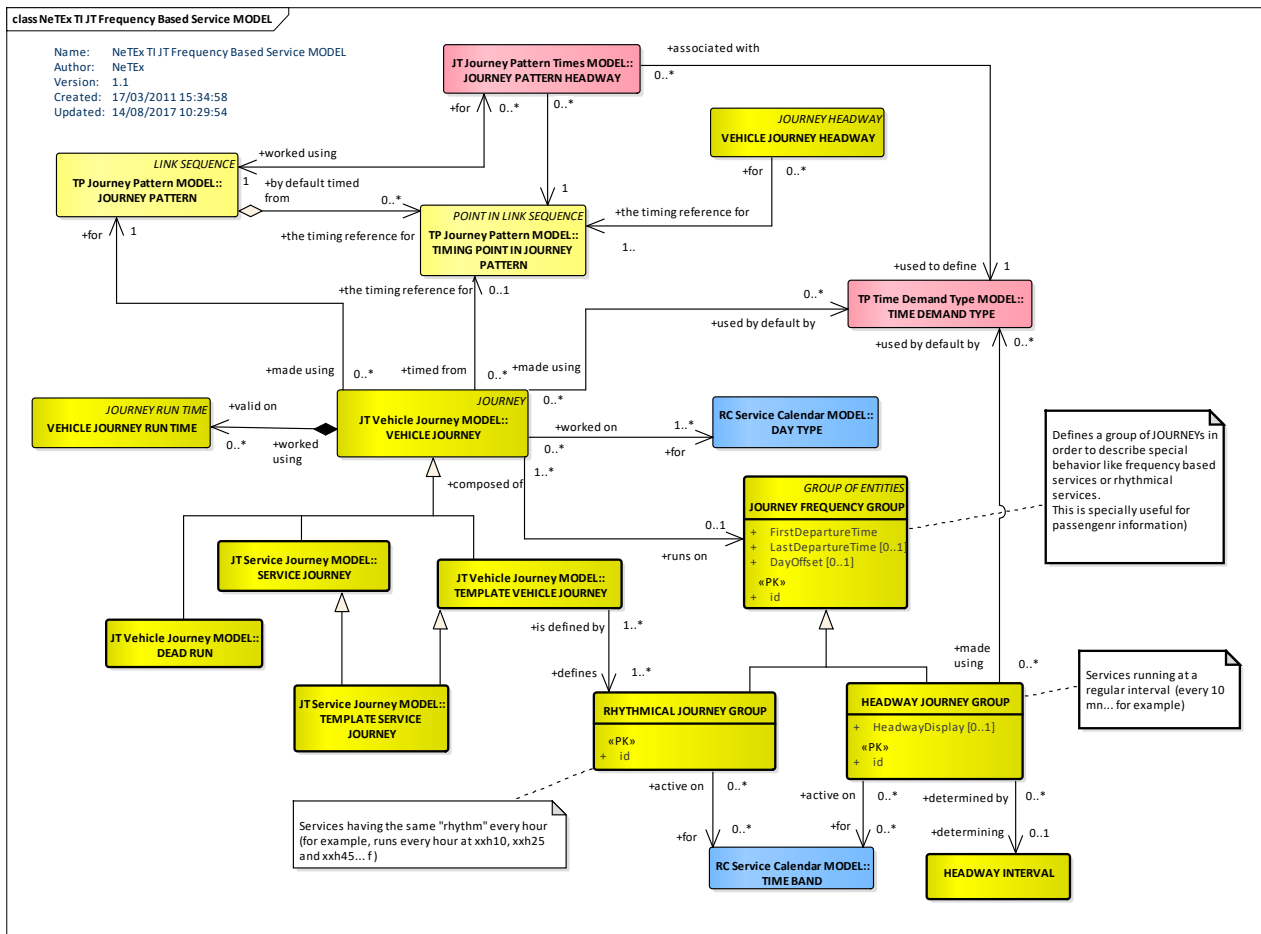


Figure 63 — Frequency Based Service – Conceptual MODEL (UML)

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.

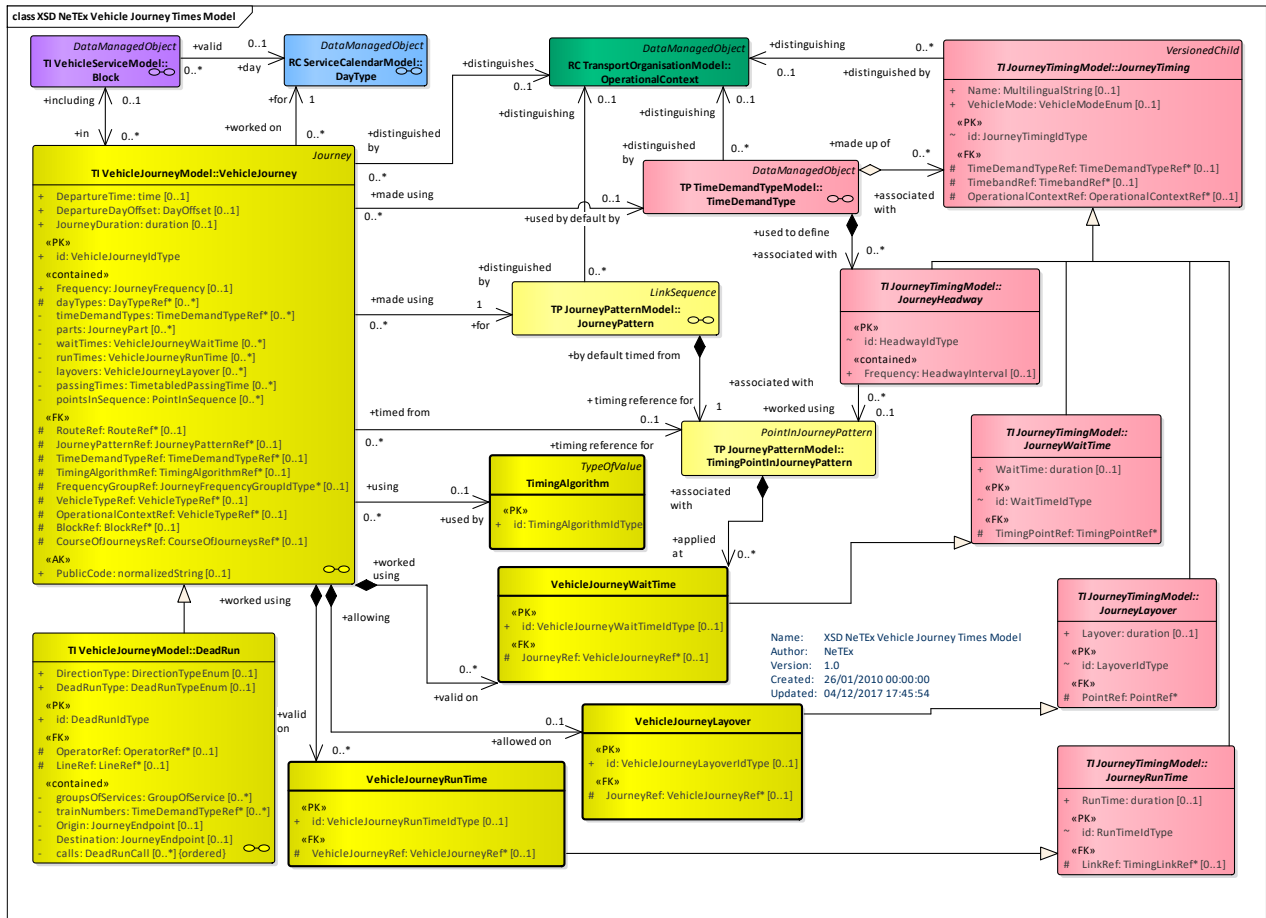


Figure 64 — Vehicle Journey Times – Physical Model (UML)

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 introduced, 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’.

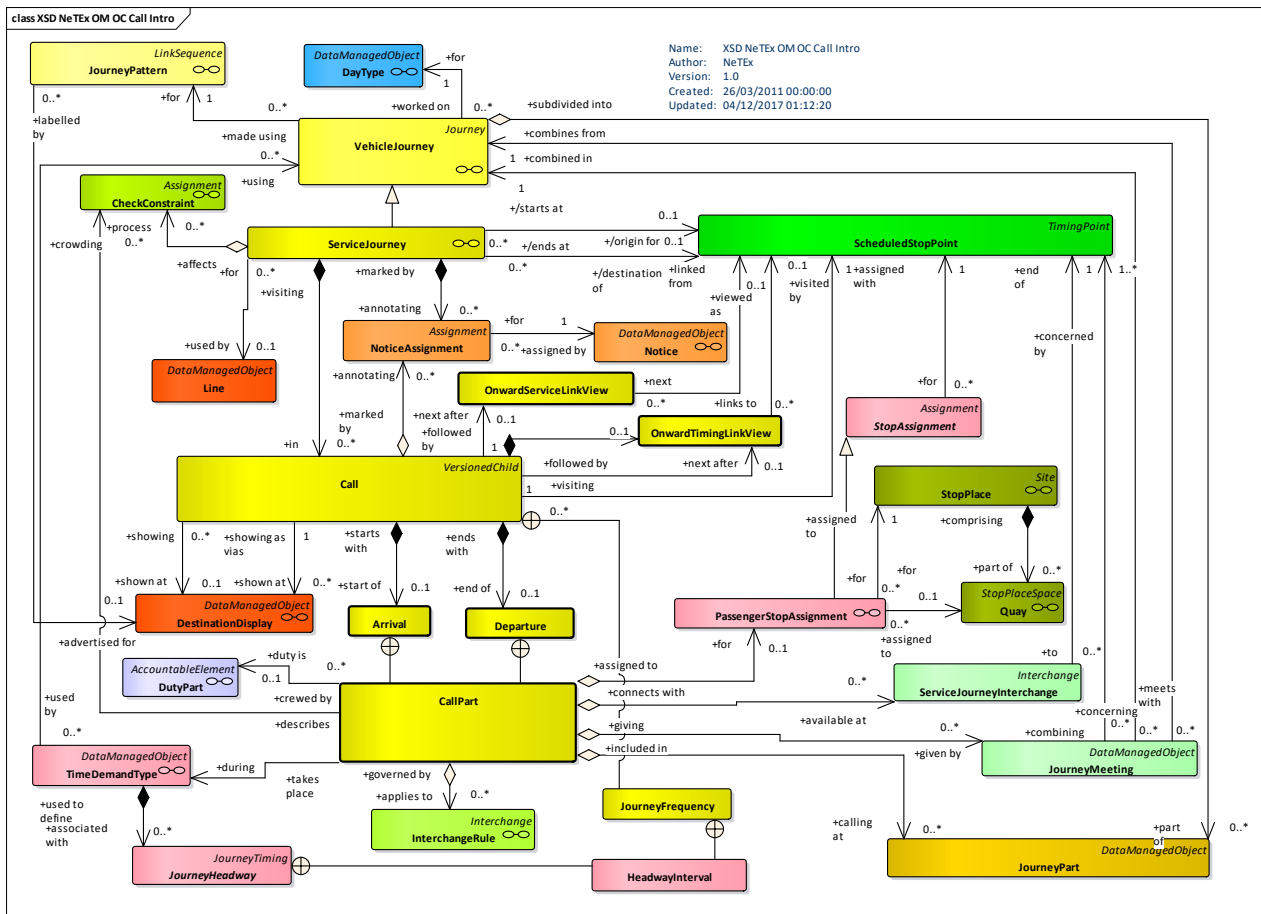


Figure 65 — Call: Overview – Physical Model (UML)

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 35 – VehicleJourneyWaitTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>JourneyWaitTime</i>	::>	VEHICLE JOURNEY WAIT TIME inherits from JOURNEY WAIT TIME.
«PK»	id	<i>VehicleJourneyWaitTimeIdType</i>	0:1	Identifier of VEHICLE JOURNEY WAIT TIME.
«FK»	VehicleJourneyRef	<i>VehicleJourneyRef</i>	0:1	Reference to a VEHICLE JOURNEY with which this WAIT TIME is associated.

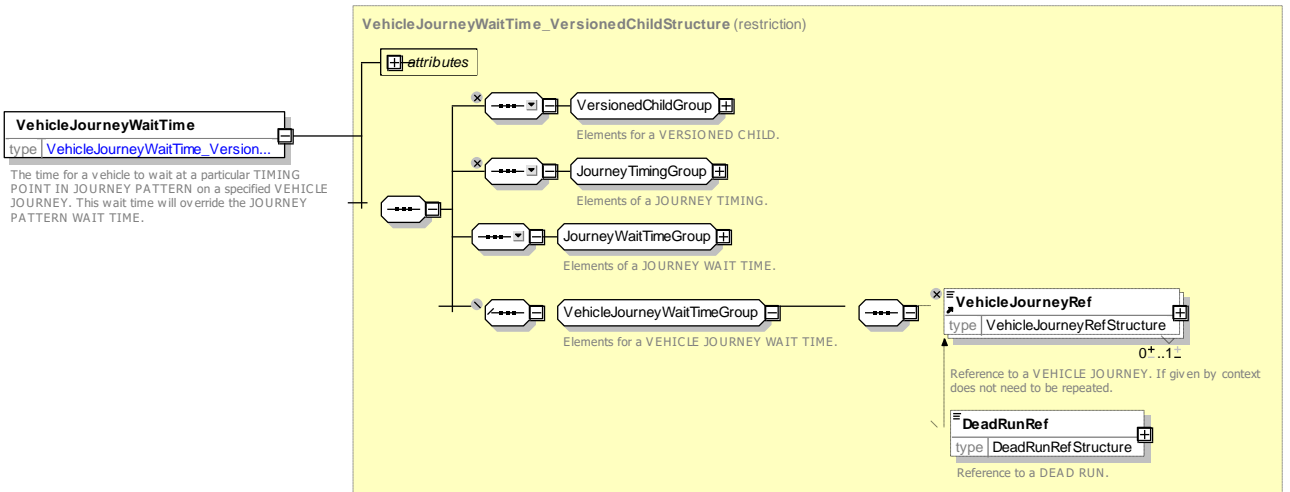


Figure 66 — *VehicleJourneyWaitTime* — XSD

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 36 – *VehicleJourneyRunTime* – Element

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

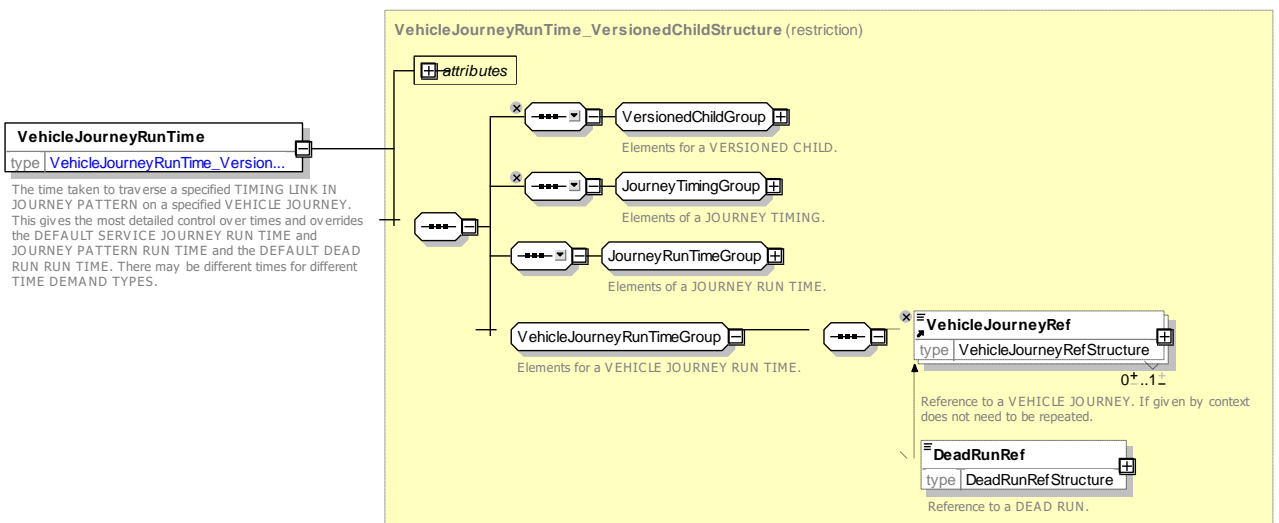


Figure 67 — *VehicleJourneyRunTime* — XSD

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 37 – *VehicleJourneyLayover* – Element

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

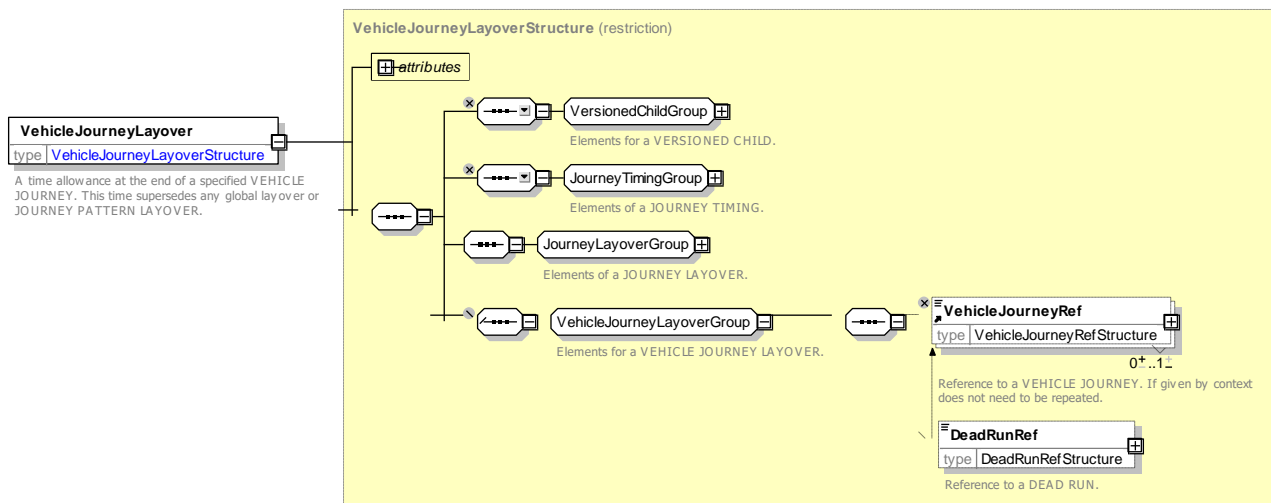


Figure 68 — *VehicleJourneyLayover* — XSD

7.2.6.3.4 *TimingAlgorithm* – Model Element

Timing algorithm used to compute times at timing points.

Table 38 – *TimingAlgorithm* – Element

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

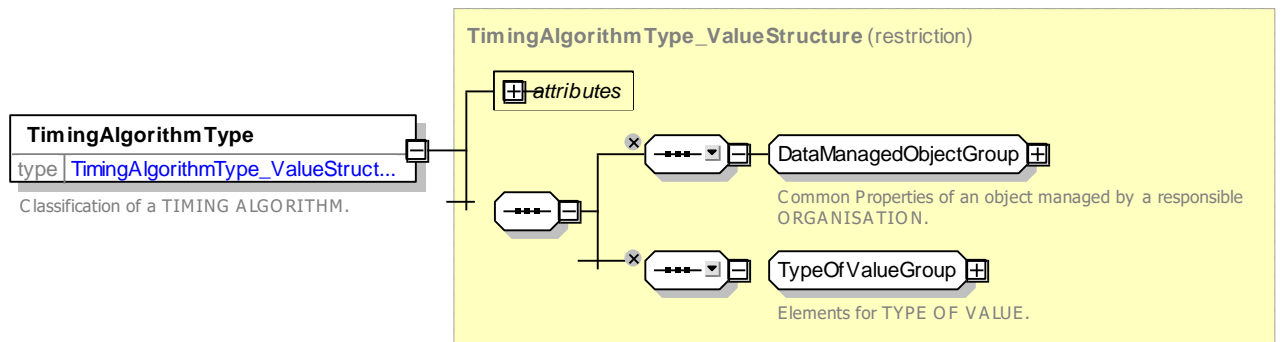


Figure 69 — *TimingAlgorithmType* — XSD

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.

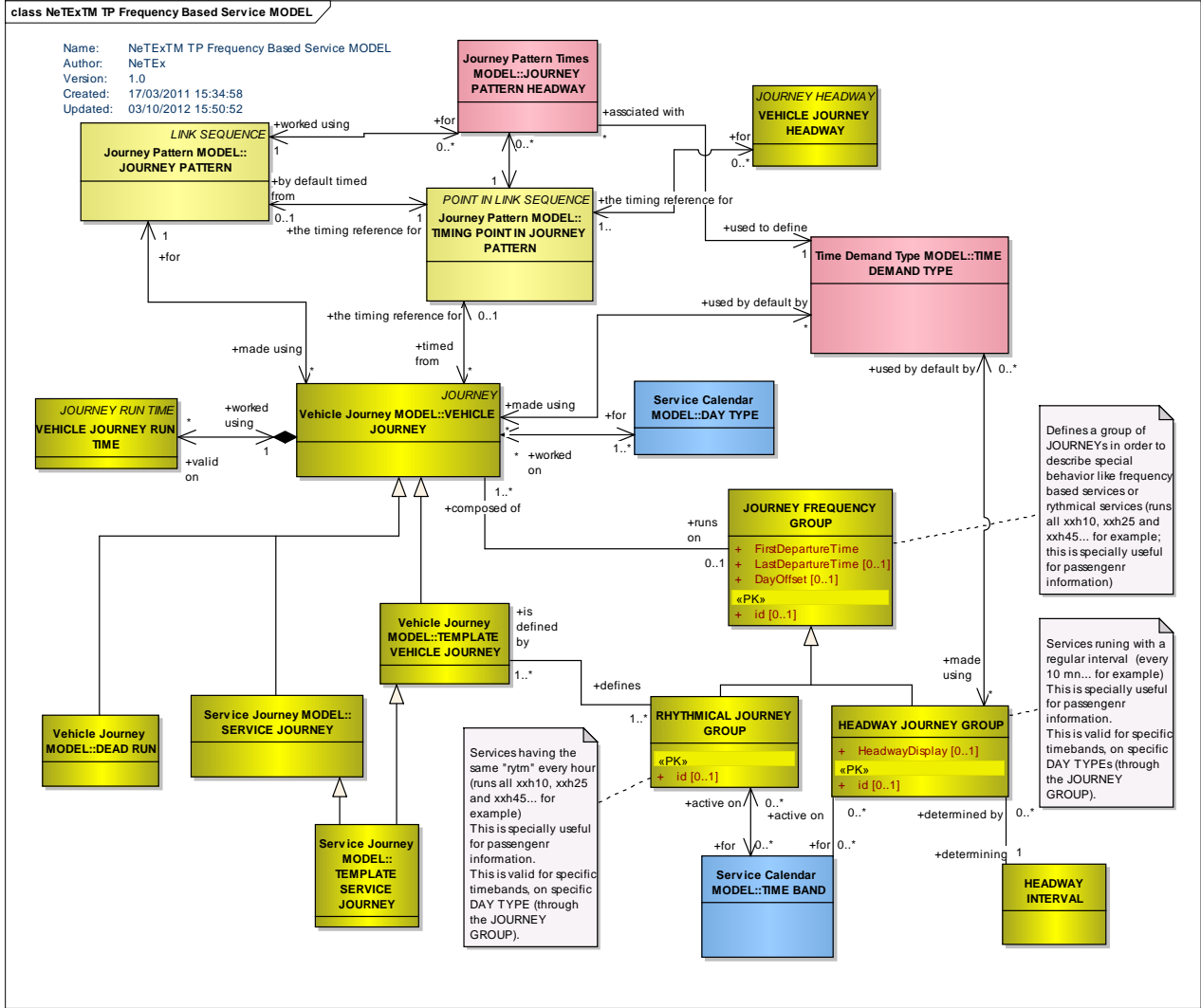
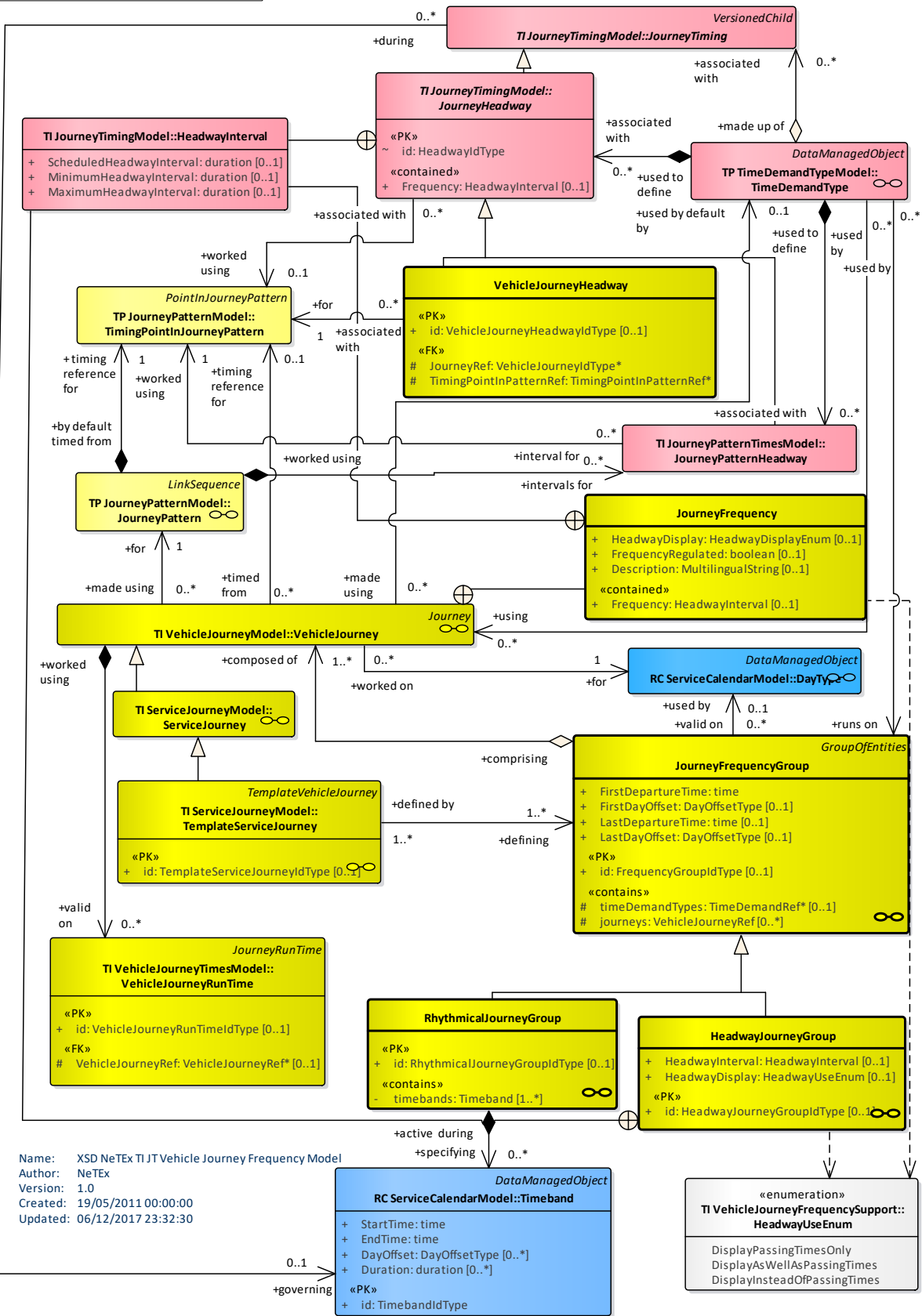


Figure 70 — Vehicle Journey Frequency – Conceptual Model (UML)

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 TI JT Vehicle Journey Frequency Model



Name: XSD NeTeX TI JT Vehicle Journey Frequency Model
 Author: NeTeX
 Version: 1.0
 Created: 19/05/2011 00:00:00
 Updated: 06/12/2017 23:32:30

Figure 71 — 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 39 – JourneyFrequencyGroup – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>GroupOfEntities</i>	::>	JOURNEY FREQUENCY GROUP inherits from GROUP OF ENTITIES.
«PK»	id	<i>FrequencyGroupIdType</i>	0:1	Identifier of FREQUENCY GROUP.
	FirstDepartureTime	<i>xsd:time</i>	1:1	Time of first departure in FREQUENCY GROUP.
	LastDepartureTime	<i>xsd:time</i>	0:1	Time of last departure in FREQUENCY GROUP.
	DayOffset	<i>xsd:integer</i>	0:1	Offset of end time day from start time.
«cntd»	timeDemandTypes	<i>TimeDemandRef</i>	0:1	TIME DEMAND TYPES for which this FREQUENCY GROUP applies.
«cntd»	journeys	<i>VehicleJourneyRef</i>	0:*	Journeys belonging to FREQUENCY GROUP.

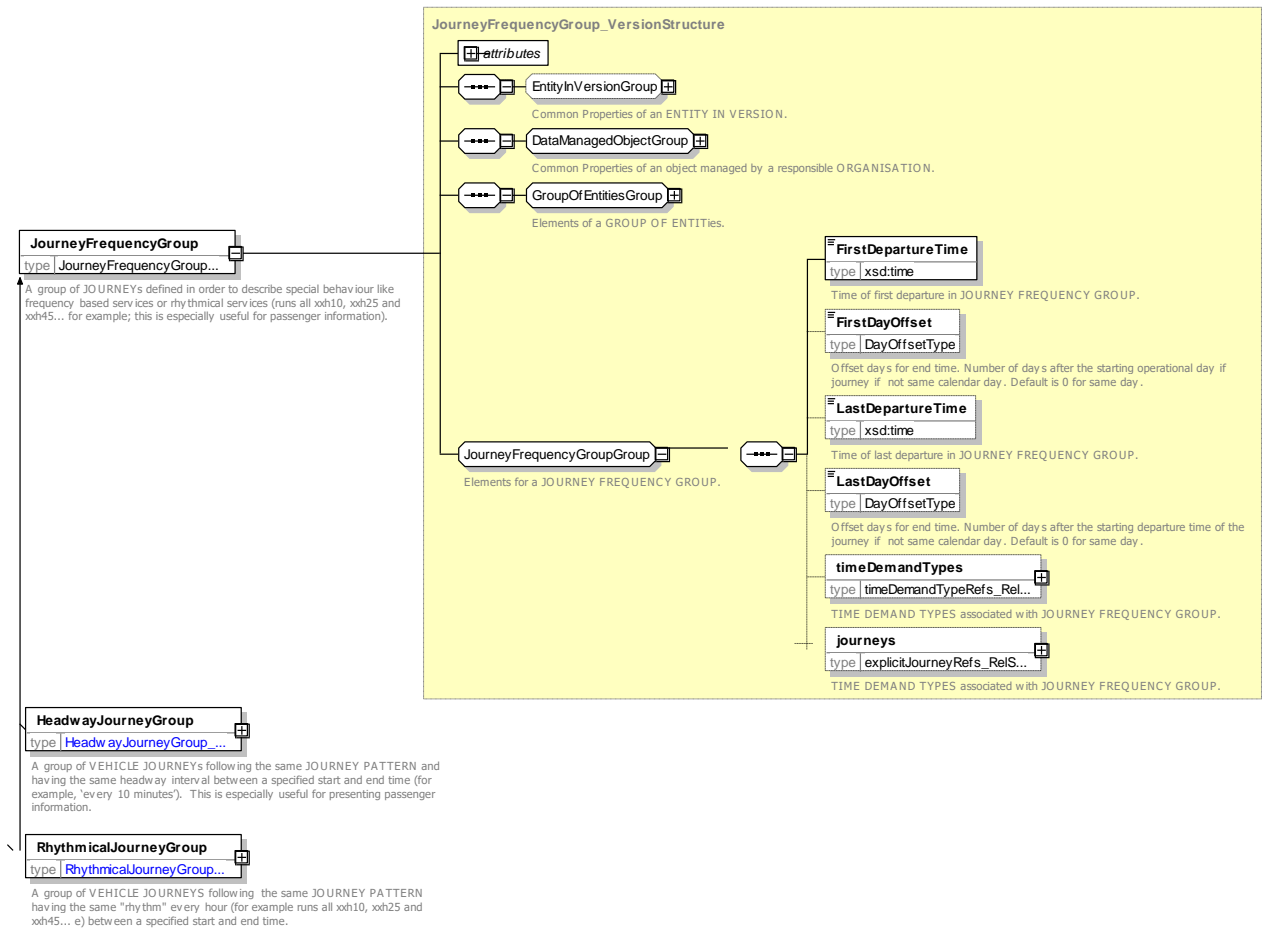


Figure 72 — JourneyFrequencyGroup — XSD

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 40 – 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.
	MinimumHeadwayInterval	xsd:duration	0:1	Minimum HEADWAY INTERVAL.
	MaximumHeadwayInterval	xsd:duration	0:1	Maximum HEADWAY INTERVAL.

	HeadwayDisplay	<i>HeadwayUseEnum</i>	0:1	How headway is to be displayed to passengers.
	Description	<i>MultilingualString</i>	0:1	Description of Headway Journey Groups.

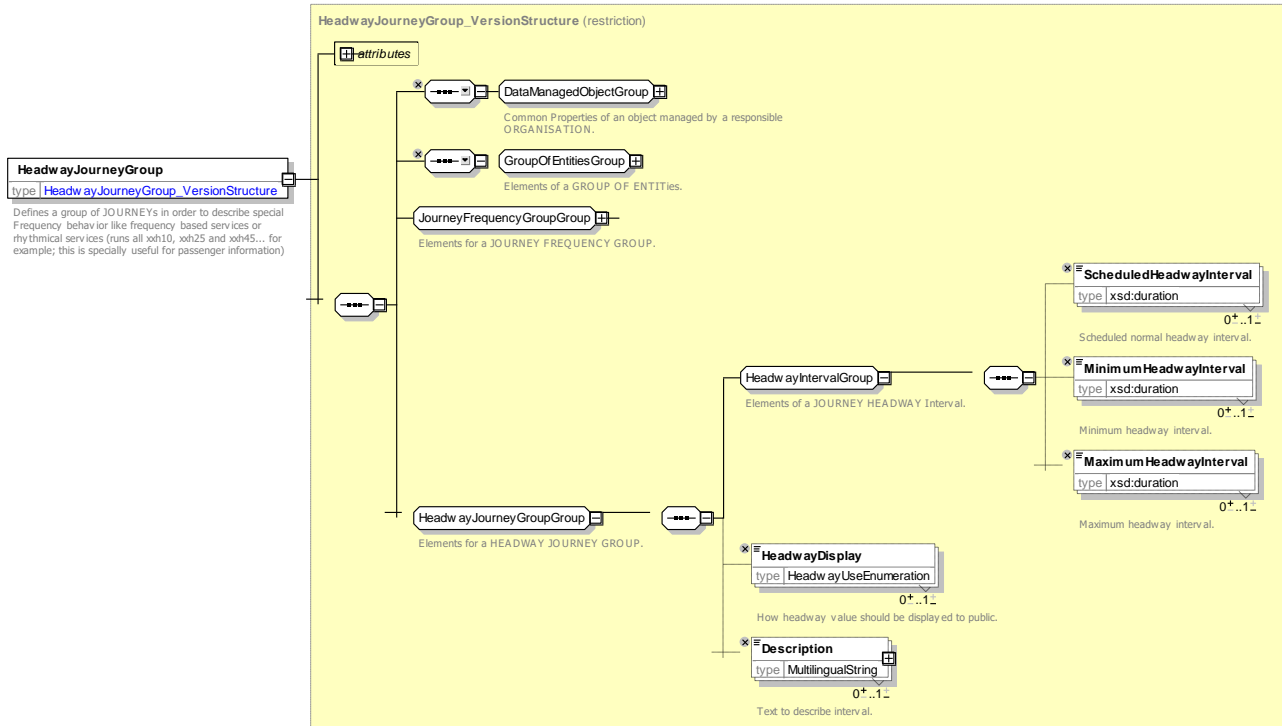


Figure 73 — HeadwayJourneyGroup — XSD

7.2.6.5.2.1 HeadwayDisplay – Allowed values

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

Table 41 – HeadwayDisplay – Allowed values

Value	Description
<i>displayInsteadOfPassingTimes</i>	Display intervals instead of passing times.
<i>displayAsWellAsPassingTimes</i>	Display intervals as well as passing times.
<i>displayPassingTimesOnly</i>	Display only passing times.

7.2.6.5.3 RhythmicJourneyGroup – 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 42 – RhythmicJourneyGroup – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>JourneyFrequencyGroup</i>	::>	RHYTHMICAL JOURNEY GROUP inherits from JOURNEY FREQUENCY GROUP.

«PK»	id	<i>RhythmicalJourneyGroupIdType</i>	0:1	Identifier of RHYTHMICAL JOURNEY GROUP.
«cntd»	timebands	<i>Timeband</i>	1:*	Timebands describing RHYTHMICAL FREQUENCY GROUP. Each specifies some minutes past the hour.

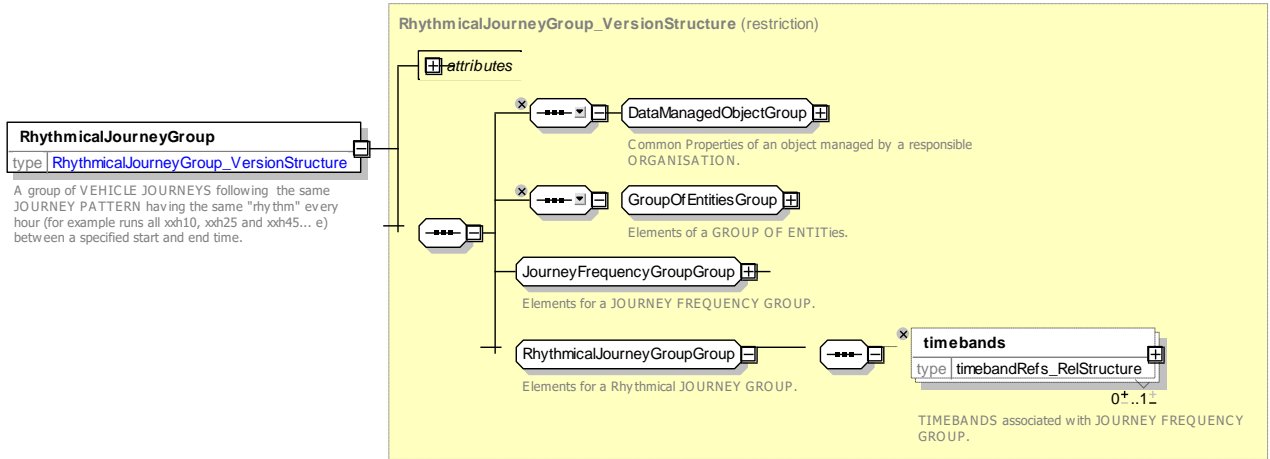


Figure 74 — *RhythmicalJourneyGroup* — XSD

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 43 – *VehicleJourneyHeadway* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>HeadwayInterval</i>	::>	VEHICLE JOURNEY HEADWAY inherits from HEADWAY INTERVAL.
«PK»	id	<i>VehicleJourneyHeadwayIdType</i>	0:1	Identifier of VEHICLE JOURNEY HEADWAY.
«FK»	JourneyRef	<i>VehicleJourneyRef</i>	0:1	Reference to JOURNEY associated with VEHICLE JOURNEY HEADWAY.
«FK»	TimingPointInPatternRef	<i>TimingPointInPatternRef</i>	1:1	Reference to a TIMING POINT in JOURNEY PATTERN associated with VEHICLE JOURNEY HEADWAY.

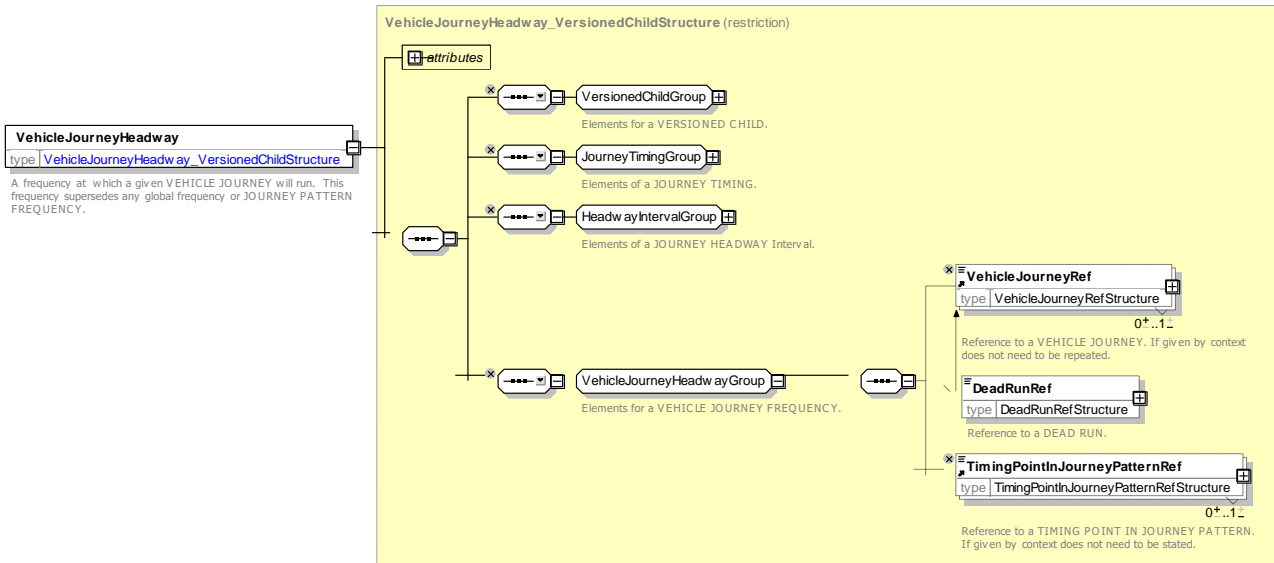


Figure 75 — VehicleJourneyHeadway — XSD

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 44 – JourneyFrequency – Element

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

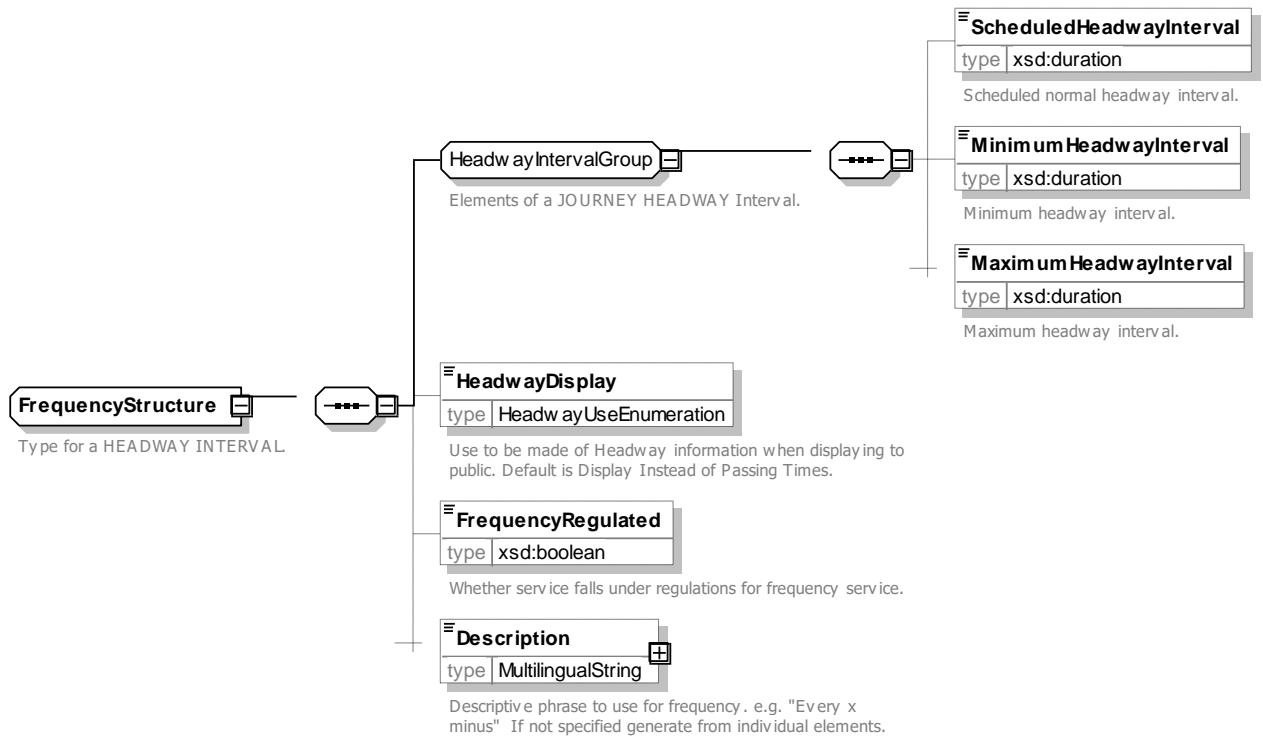


Figure 76 — JourneyFrequency — XSD

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>
    </Call>
  </calls>
</TemplateServiceJourney>

```

```

        <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>
</TemplateServiceJourney>

```



```

        </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.

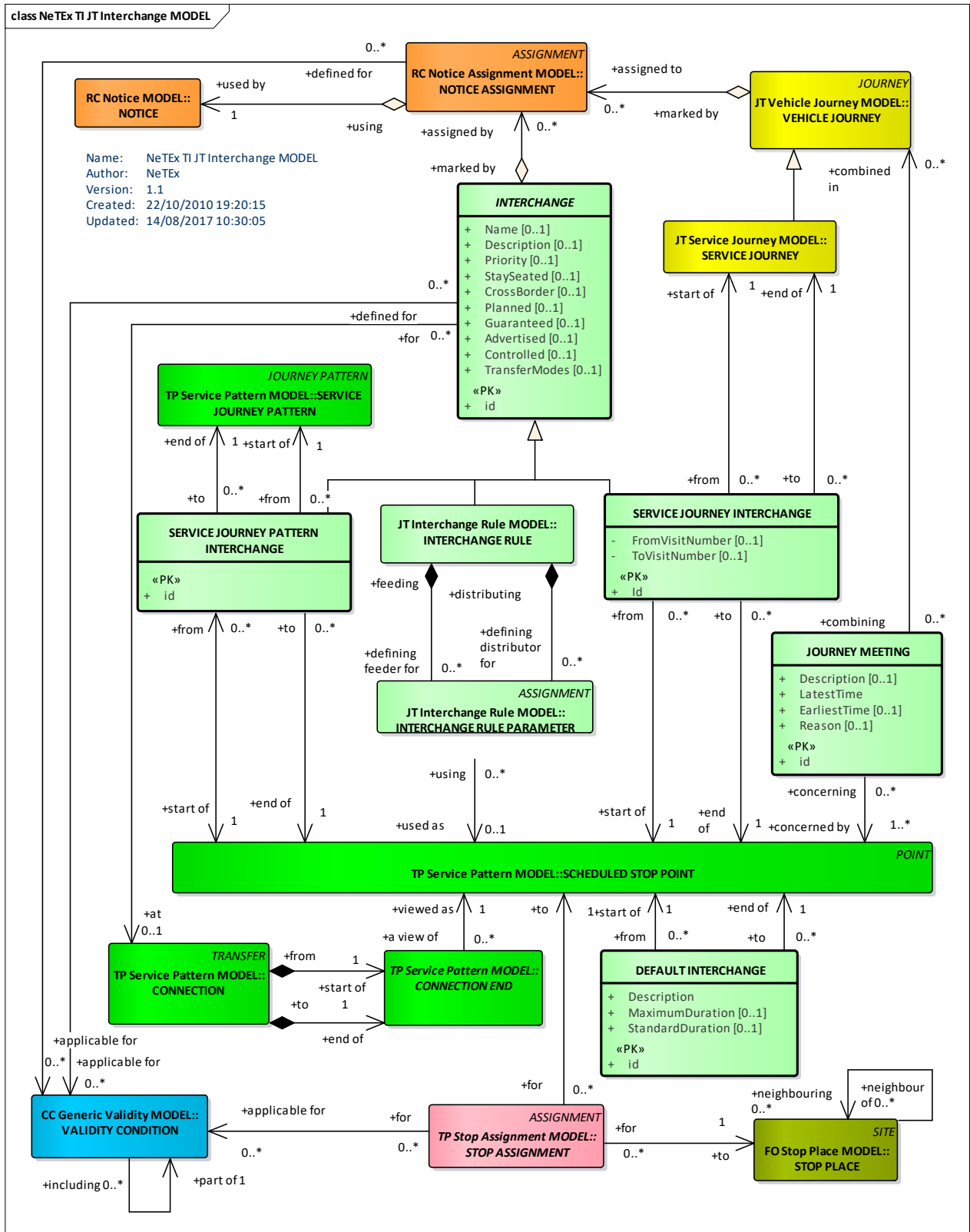


Figure 77 — Interchange – Conceptual MODEL (UML)

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 **ExplicitTransferTime** 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.

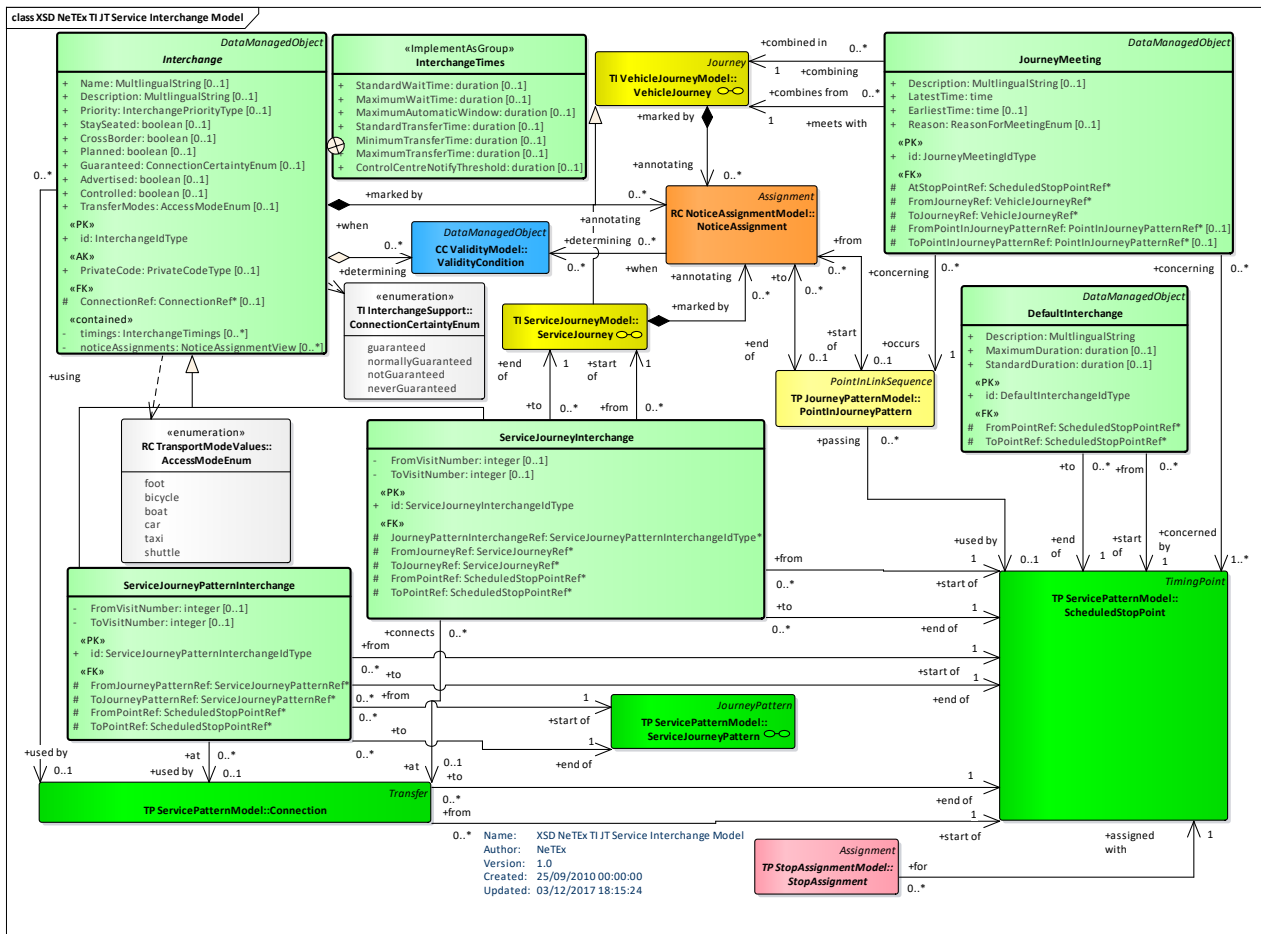


Figure 78 — Service Interchange – Physical Model (UML)

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 45 – DefaultInterchange – Element

Classification	Name	Type	Cardinality	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	<i>ScheduledStopPointRef</i>	1:1	SCHEDULED STOP POINT from which the DEFAULT INTERCHANGE starts.
«FK»	ToPointRef	<i>ScheduledStopPointRef</i>	1:1	SCHEDULED STOP POINT at which the DEFAULT INTERCHANGE ends.

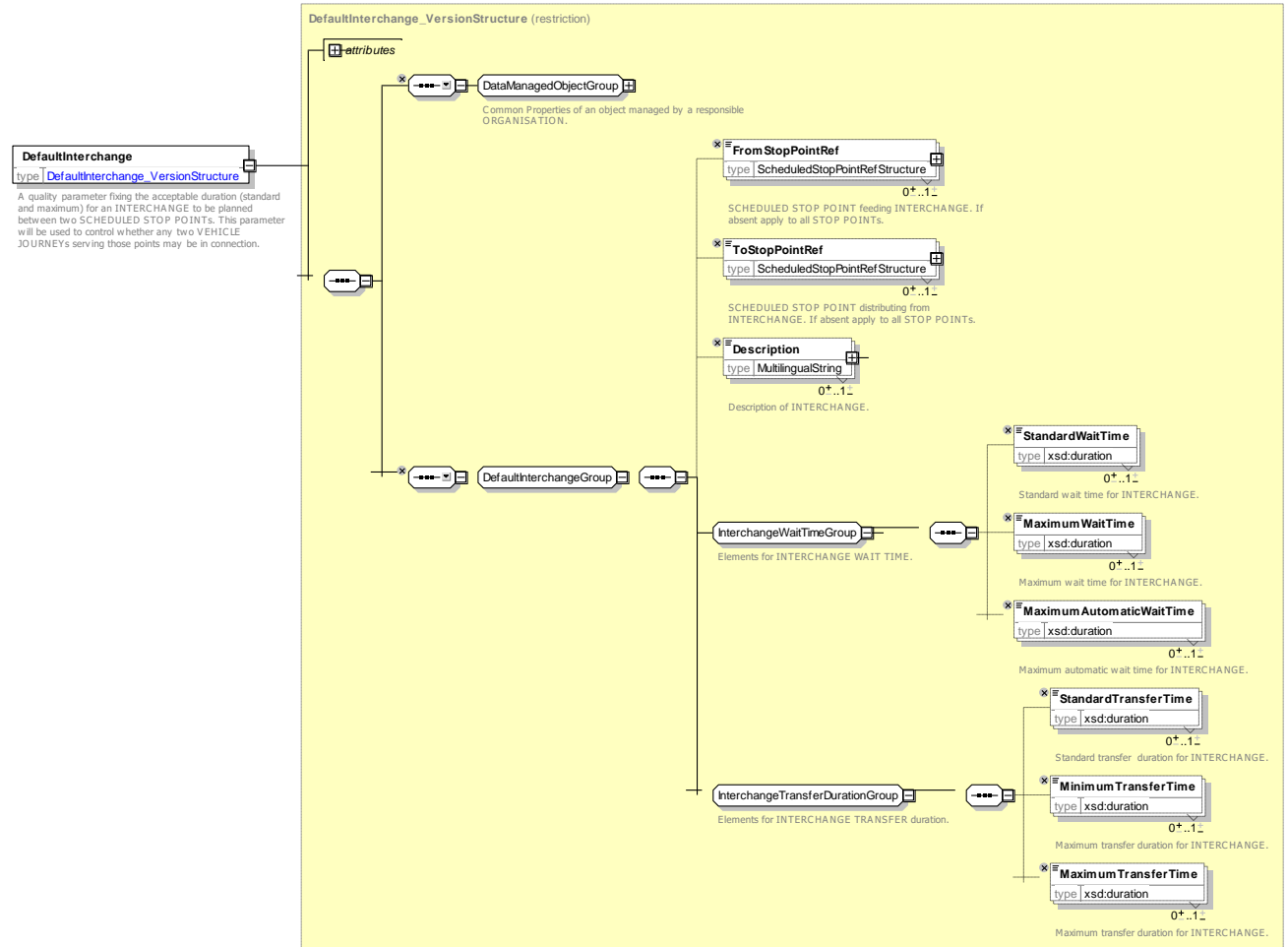


Figure 79 — DefaultInterchange — XSD

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 46 – Interchange – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	INTERCHANGE inherits from DATA MANAGED OBJECT.
«PK»	id	<i>InterchangeIdType</i>	1:1	Identifier of INTERCHANGE.
	Name	<i>MultilingualString</i>	0:1	Name of INTERCHANGE.

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

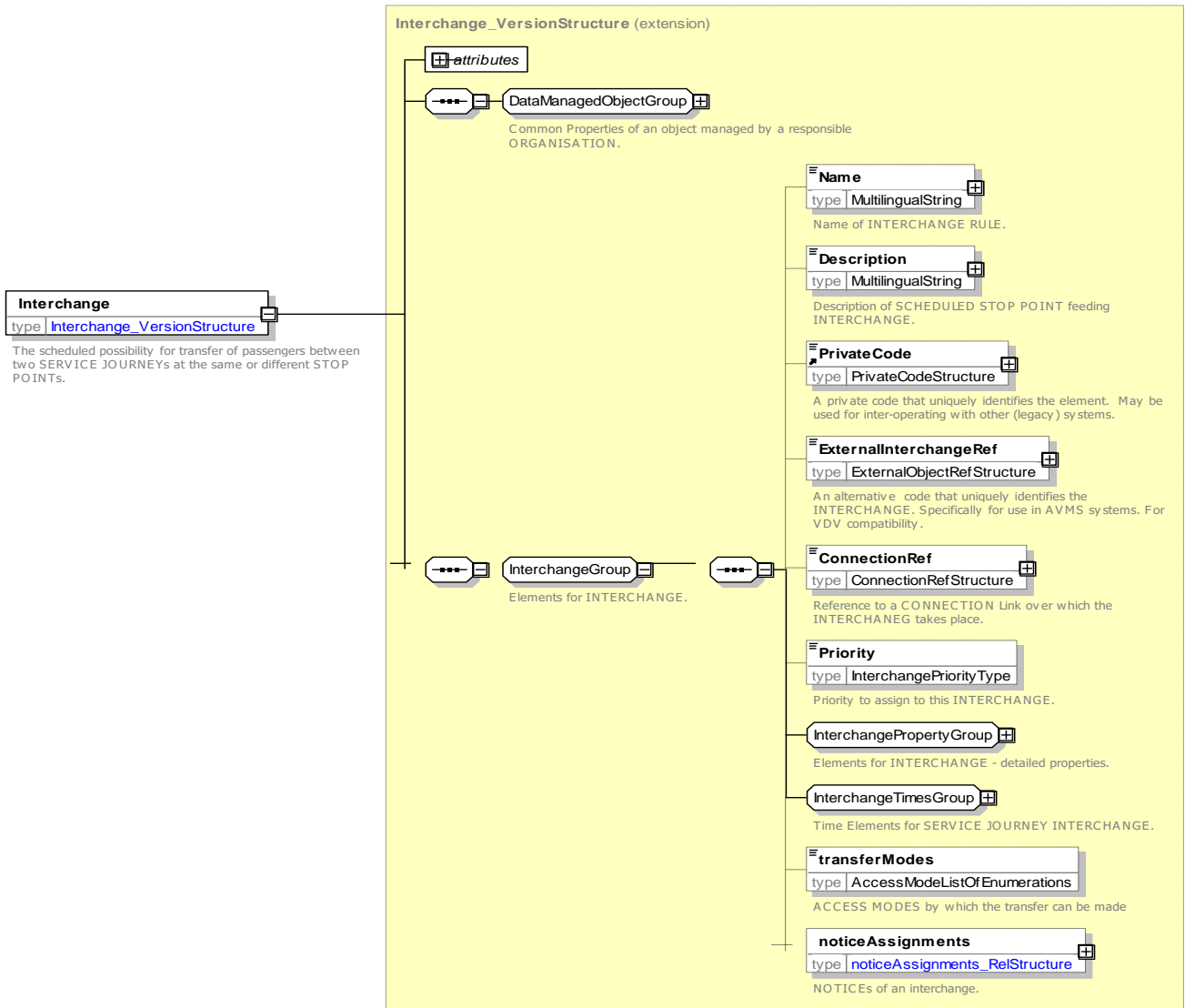


Figure 80 — Interchange — XSD

7.2.7.3.2.1 InterchangePropertyGroup — Group

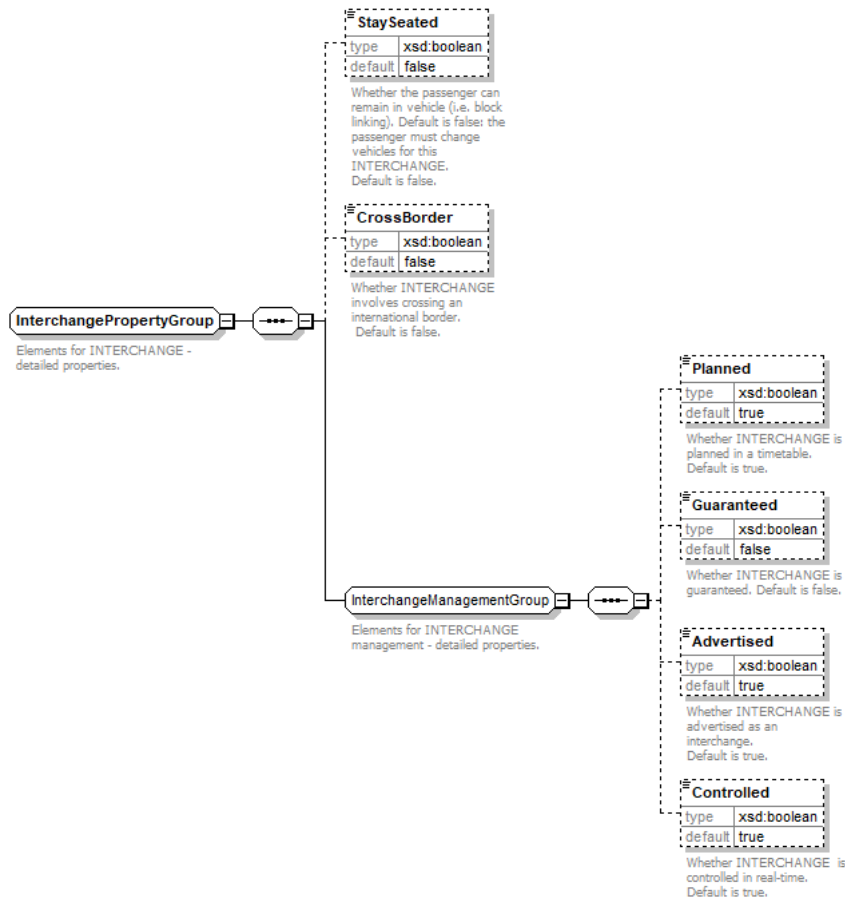


Figure 81 — InterchangePropertyGroup — XSD

Table 47 – InterchangeTimesGroup – Group

Classification	Name	Type	Cardinality	Description
	StandardWaitTime	<i>xsd:duration</i>	0:1	Standard Time to wait at Interchange.
	MaximumWaitTime	<i>xsd:duration</i>	0:1	Maximum time that DISTRIBUTOR will wait after its planned departure time.
	MaximumAutomatic-WaitTime	<i>xsd:duration</i>	0:1	Maximum automatic wait time for INTERCHANGE.
	StandardTransferTime	<i>xsd:duration</i>	0:1	Standard Time needed for transfer.
	MinimumTransferTime	<i>xsd:duration</i>	0:1	Minimum Time needed for transfer.
	MaximumTransferTime	<i>xsd:duration</i>	0:1	Maximum Time needed for transfer.
	ControlCentreNotifyThreshold	<i>xsd:duration</i>	0:1	Interval before CONTROL CENTRE should be notified associated with SERVICE JOURNEY INTERCHANGE.

7.2.7.3.2.2 InterchangeTimesGroup — Group

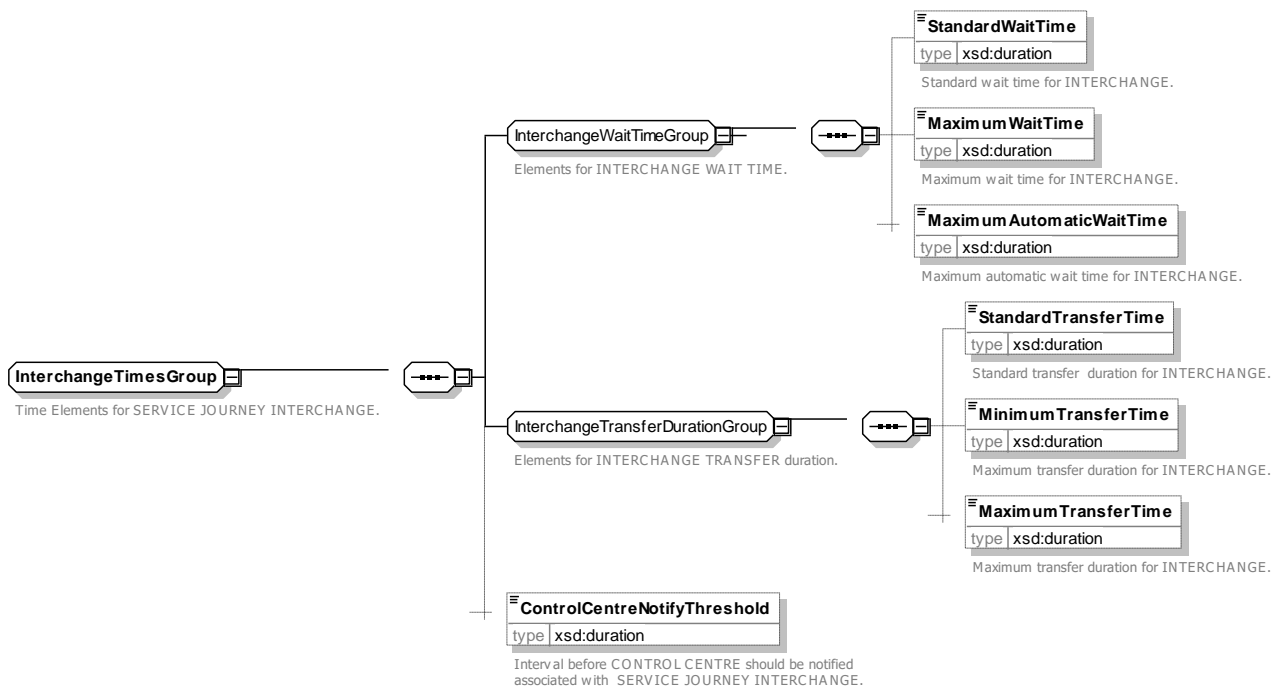


Figure 82 — InterchangeTimesGroup — XSD

7.2.7.3.2.3 ConnectionCertainty – Allowed values

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

Table 48 – ConnectionCertainty – Allowed values

Value	Description
<i>guaranteed</i>	The connection is guaranteed under any circumstances.
<i>normallyGuaranteed</i>	The connection is normally guaranteed, although the connection time available is shorter than the location connection time.
<i>notGuaranteed</i>	The connection is not guaranteed, although the connection time available is longer than the location connection time.
<i>neverGuaranteed</i>	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 49 – ServiceJourneyInterchange – Element

Classification	Name	Type	Cardinality	Description

::>	::>	<i>Interchange</i>	::>	SERVICE JOURNEY INTERCHANGE inherits from INTERCHANGE.
«PK»	id	<i>ServiceJourneyInterchangeIdType</i>	1:1	Identifier of SERVICE JOURNEY INTERCHANGE.
«FK»	FromPointRef	<i>ScheduledStopPointRef</i>	1:1	SCHEDULED STOP POINT from which the interchange starts.
	FromVisitNumber	<i>xsd:integer</i>	0:1	Visit number of feeder journey (only needed if multiple visits).
«FK»	ToPointRef	<i>ScheduledStopPointRef</i>	1:1	SCHEDULED STOP POINT to which the interchange goes.
	ToVisitNumber	<i>xsd:integer</i>	0:1	Visit number of distributor journey (only needed if multiple visits).
«FK»	FromJourneyRef	<i>ServiceJourneyRef</i>	1:1	Reference to feeder SERVICE JOURNEY from which interchange is made.
«FK»	ToJourneyRef	<i>ServiceJourneyRef</i>	1:1	Reference to distributor SERVICE JOURNEY to which interchange is made.
«FK»	ServiceJourneyPatternInterchangeRef	<i>ServiceJourneyPatternInterchangeRef</i>	0:1	Reference to SERVICE JOURNEY PATTERN INTERCHANGE for which interchange is made.

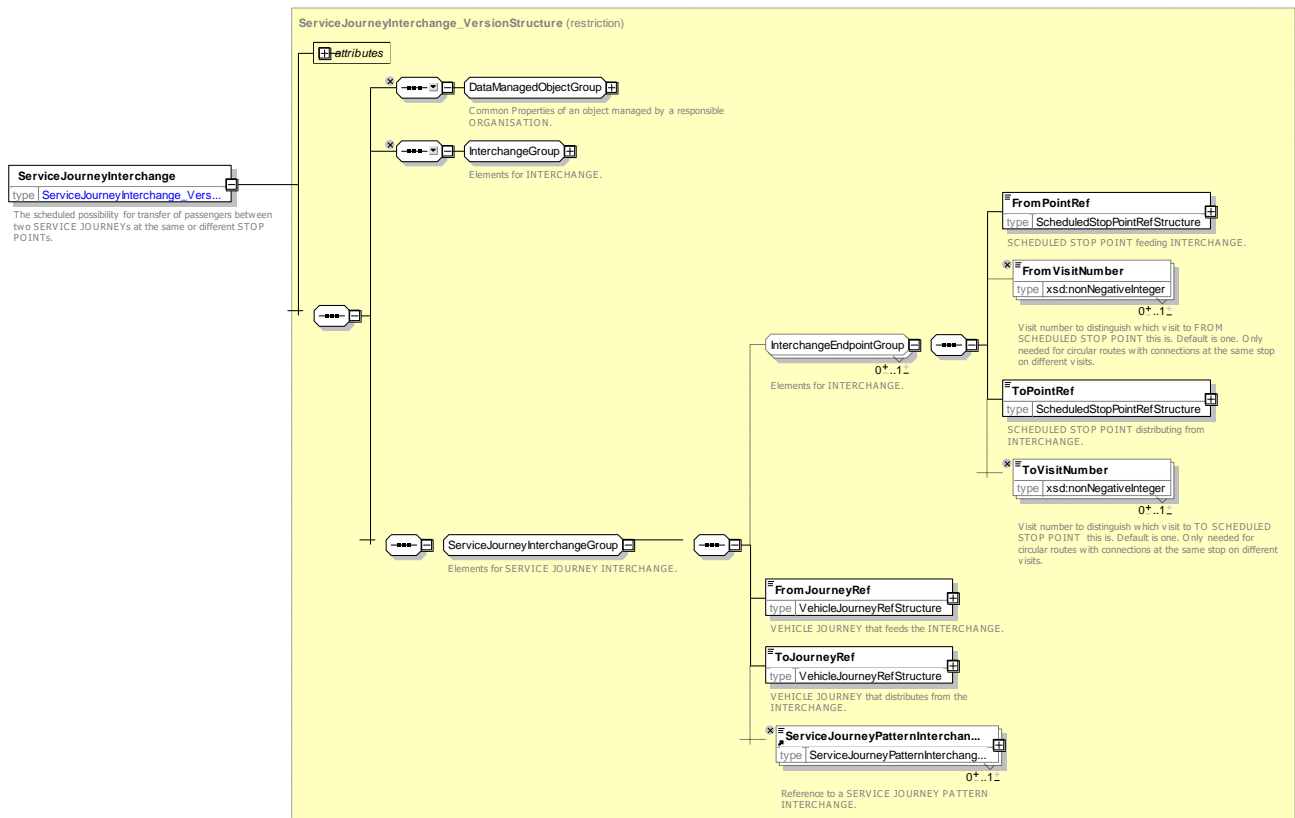


Figure 83 — ServiceJourneyInterchange — XSD

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 50 – ServiceJourneyPatternInterchange – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Interchange</i>	::>	SERVICE JOURNEY PATTERN INTERCHANGE inherits from INTERCHANGE.
«PK»	id	<i>ServiceJourneyPatternInterchangeIdType</i>	1:1	Identifier of SERVICE JOURNEY PATTERN INTERCHANGE.
	MaximumDuration	<i>xsd:duration</i>	0:1	Maximum duration of a transfer for SERVICE JOURNEY PATTERN INTERCHANGE.
	StandardDuration	<i>xsd:duration</i>	0:1	Standard duration of a transfer for SERVICE JOURNEY PATTERN INTERCHANGE.
«FK»	FromPointRef	<i>ScheduledStopPointRef</i>	1:1	SCHEDULED STOP POINT from which the interchange starts.
	FromVisitNumber	<i>xsd:integer</i>	0:1	Visit number of feeder journey (only needed if multiple visits).
«FK»	ToPointRef	<i>ScheduledStopPointRef</i>	1:1	SCHEDULED STOP POINT to which the interchange goes.
	ToVisitNumber	<i>xsd:integer</i>	0:1	Visit number of distributor journey (only needed if multiple visits).
«FK»	FromJourney-PatternRef	<i>ServiceJourneyPatternRef</i>	1:1	Feeder SERVICE JOURNEY PATTERN from which SERVICE JOURNEY PATTERN INTERCHANGE is made.
«FK»	ToJourney-PatternRef	<i>ServiceJourneyPatternRef</i>	1:1	Distributor SERVICE JOURNEY PATTERN to which SERVICE JOURNEY PATTERN INTERCHANGE is made.

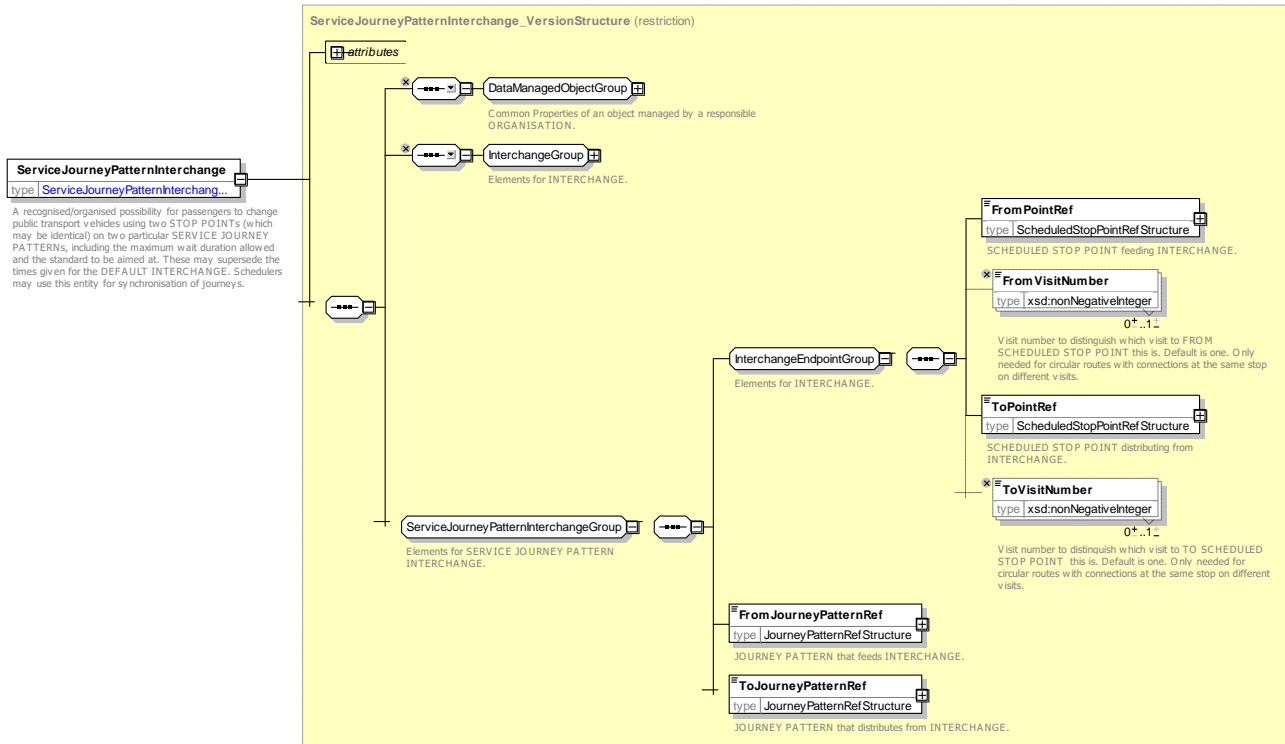


Figure 84 — ServiceJourneyPatternInterchange — XSD

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 51 – JourneyMeeting – Element

Classification	Name	Type	Cardinality	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.
	EarliestTimeDayOffset	DayOffsetType	0:1	Earliest time Day Offset from start of FROM JOURNEY.
	LatestTime	xsd:time	0:1	Earliest time at which MEETING can take place.

	LatestOffset	<i>DayOffsetType</i>	0:1	Latest time Day Offset from start of FROM JOURNEY.
«FK»	ConnectionRef	<i>ScheduledStop ConnectionRef PointRef</i>	0:1	CONNECTION link at which MEETING takes place.
«FK»	ConnectingStop-PointRef	<i>ScheduledStopPointRef</i>	0:1	SCHEDULED STOP POINT at which MEETING takes place.
	ConnectingStop-PointName	<i>MultilingualString</i>	0:1	Name of SCHEDULED STOP POINT at which MEETING takes place.
	ConnectingJourneyView	<i>ConnectingJourney_DerivedViewStructure</i>	0:1	Further details on of Inbound feeder JOURNEY that meets. Can be derived.
	Connecting-LineView	<i>Line_DerivedViewStructure</i>	0:1	urther details on LINE of inbound feeder JOURNEY that meets. Can be derived.
	Reason	<i>ReasonForMeetingEnum</i>	0:1	Reason for JOURNEY MEETING.
«FK»	FromPointInJourneyPatternRef	<i>PointInJourneyPatternRef</i>	0:1	POINT IN JOURNEY PATTERN of Inbound feeder JOURNEY that meets.
«FK»	ToPointInJourney-PatternRef	<i>PointInJourneyPatternRef</i>	0:1	POINT IN JOURNEY PATTERN of outbound distributor JOURNEY that meets.

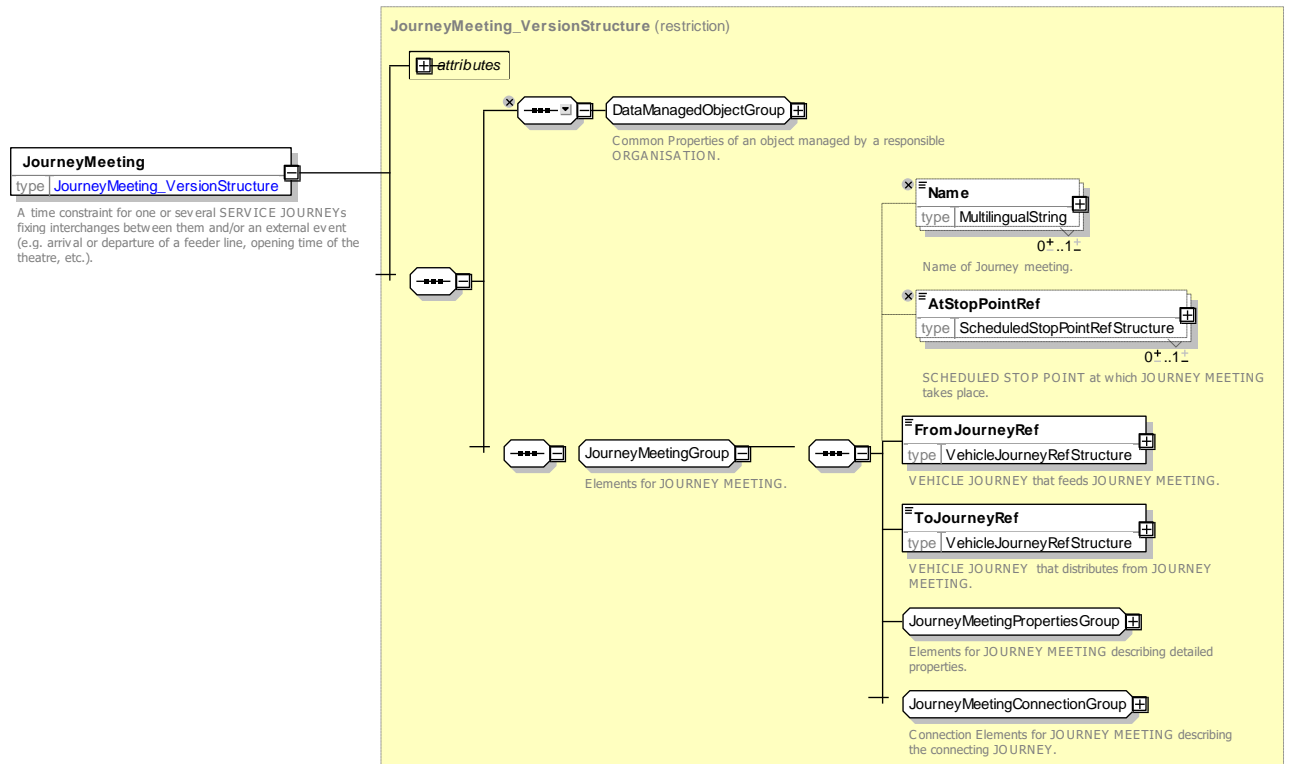


Figure 85 — *JourneyMeeting* — XSD

7.2.7.3.5.1 *JourneyMeetingPropertiesGroup* — Group

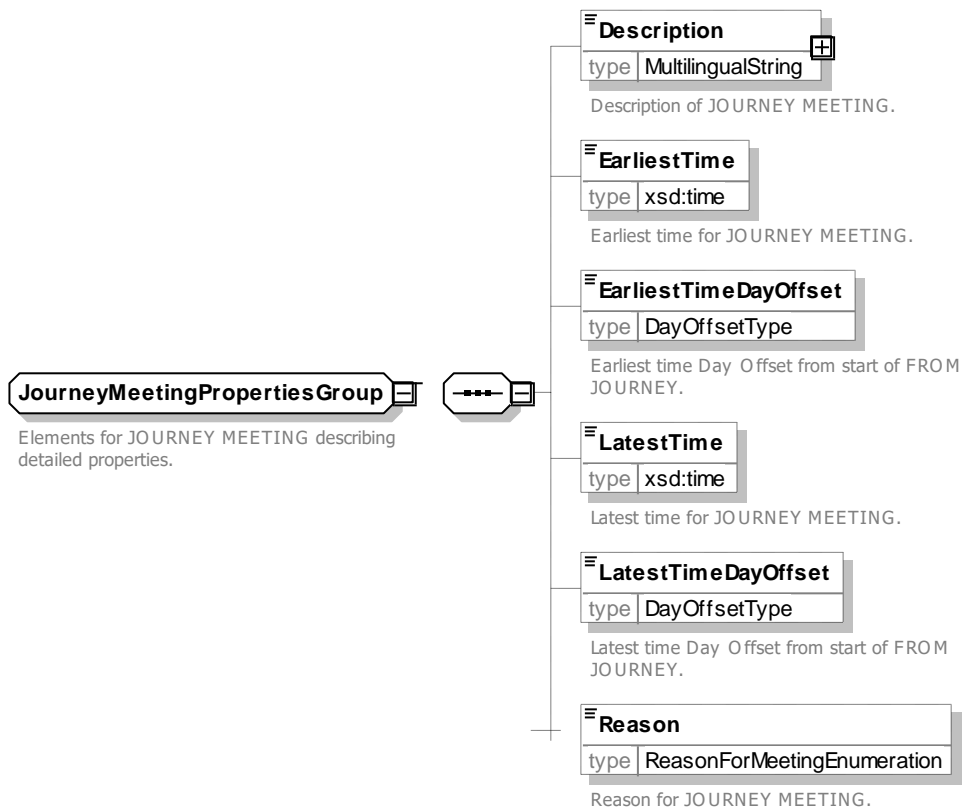


Figure 86 — *JourneyMeetingPropertiesGroup* — XSD

7.2.7.3.5.2 JourneyMeetingConnectionGroup— Group



Figure 87 — JourneyMeetingConnectionGroup— XSD

7.2.7.3.5.3 ReasonForMeeting – Allowed values

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

Table 52 – ReasonForMeeting – Allowed values

Value	Description
<i>joining</i>	Meeting is joining of two services.
<i>splitting</i>	Meeting is splitting of two services.
<i>tariffSection</i>	Meeting is for start of a Tariff Section
<i>serviceFacility</i>	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.

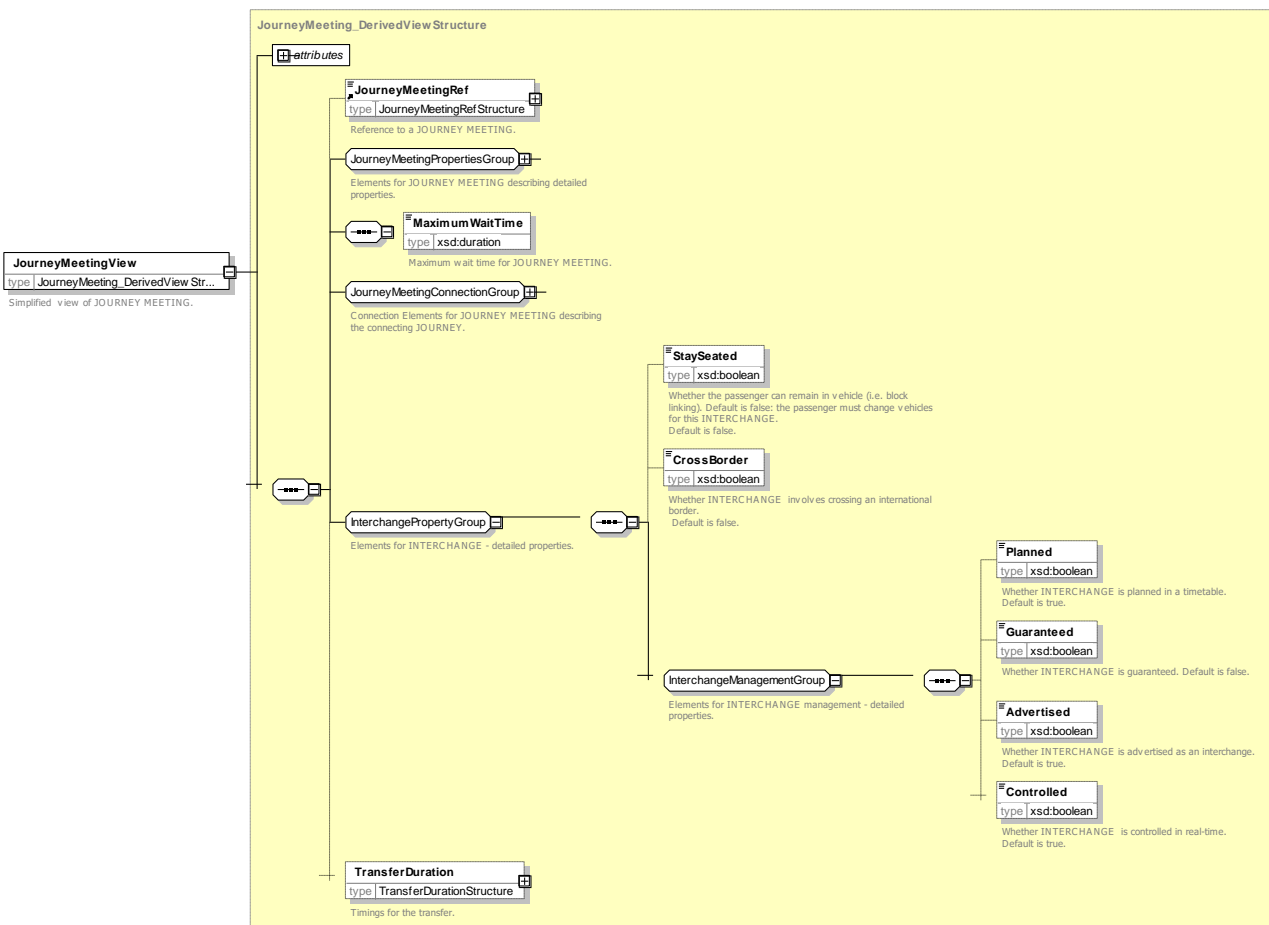


Figure 88 — JourneyMeetingView — XSD

7.2.7.3.7 ConnectingJourneyView – Model Element

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

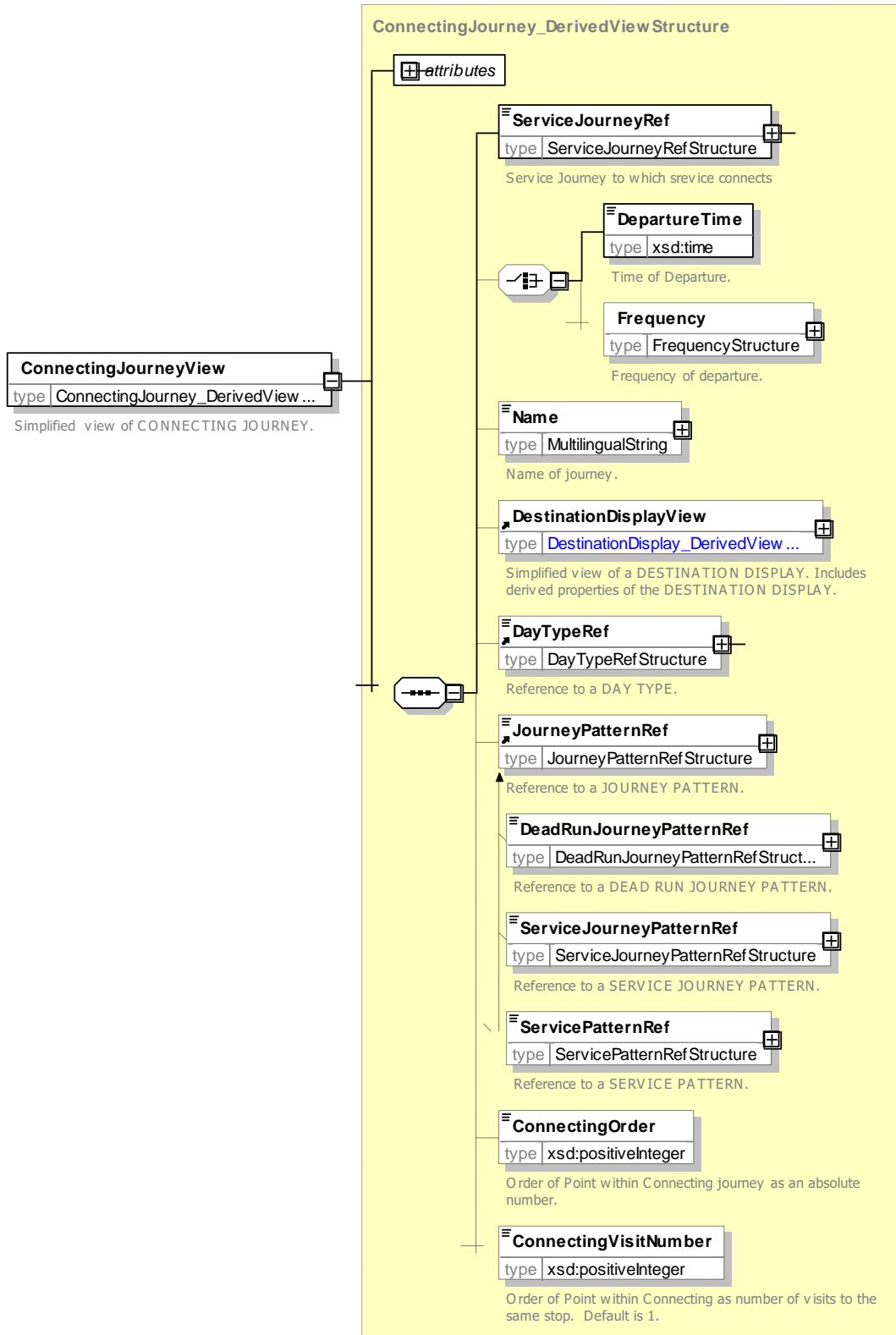


Figure 89 — **ConnectingJourneyView** — XSD

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 two 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</Description>
    <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.

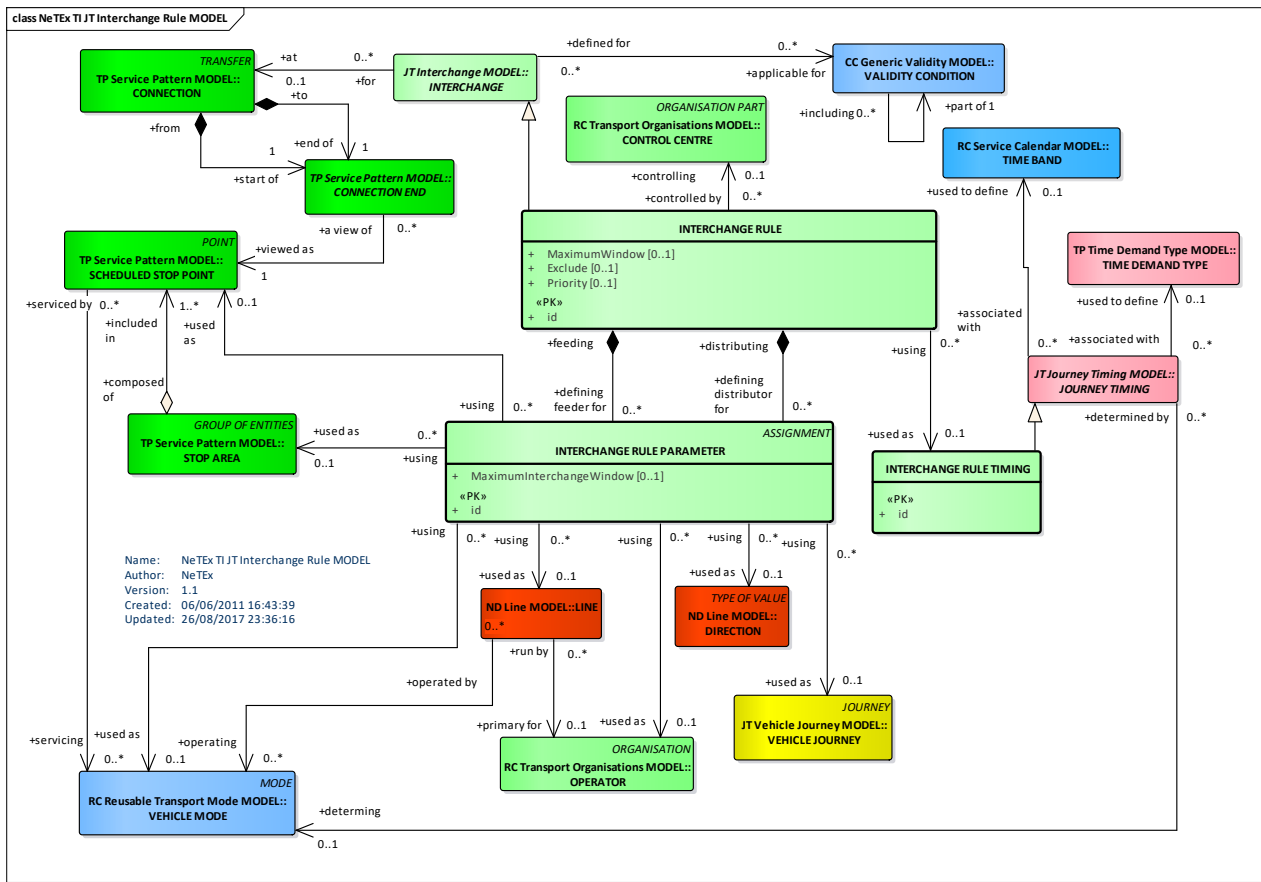


Figure 90 — Interchange Rule – Conceptual MODEL (UML)

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.

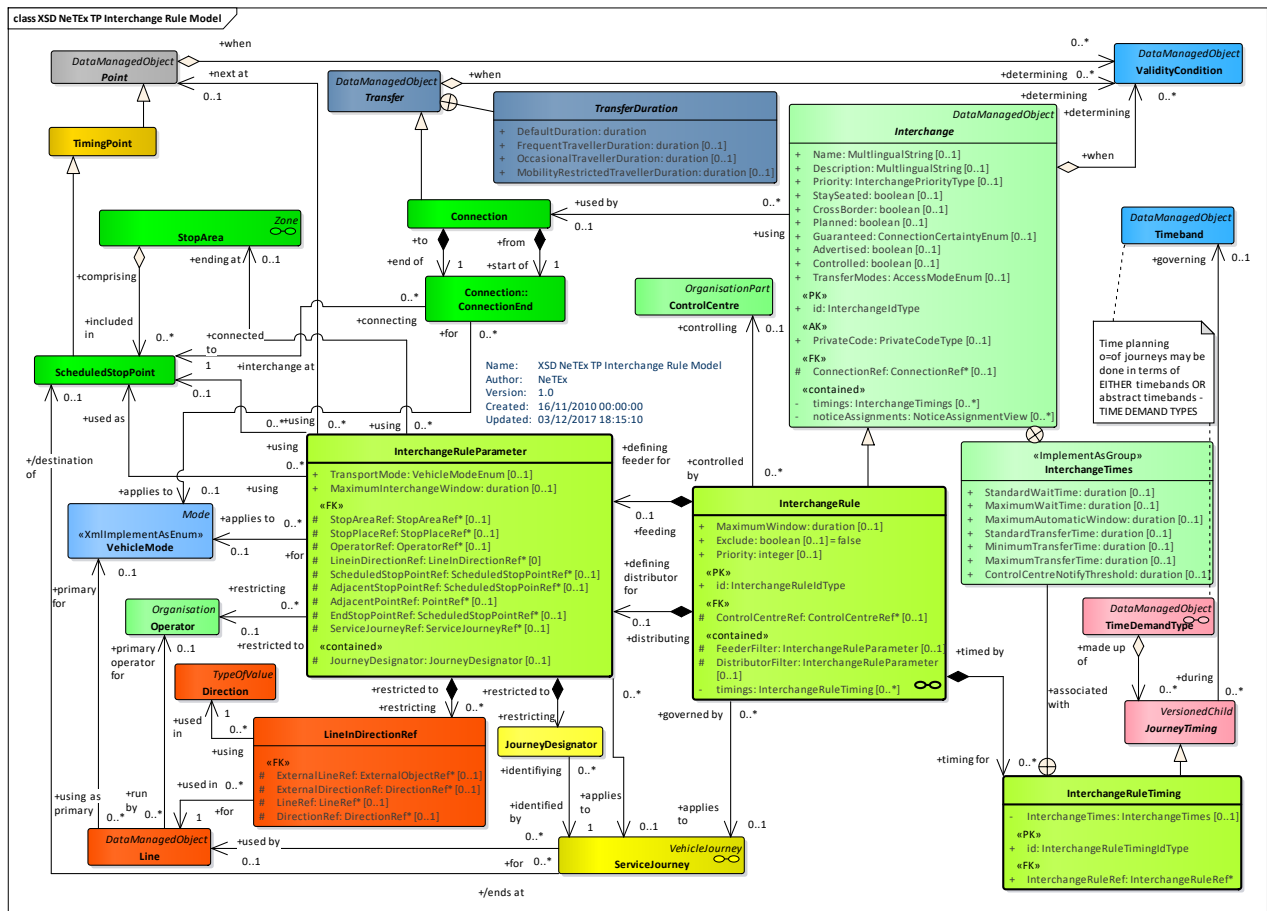


Figure 91 — Interchange Rule – Physical Model (UML)

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 53 – InterchangeRule – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	INTERCHANGE RULE inherits from DATA MANAGED OBJECT
«PK»	id	<i>InterchangeRuleIdType</i>	1:1	Identifier of INTERCHANGE RULE.
	MaximumWindow	<i>xsd:duration</i>	0:1	Maximum window for holding DISTRIBUTOR will wait.
«cntd»	validity	<i>ValidityCondition</i>	0:1	VALIDITY CONDITION governing INTERCHANGE RULE.

	Exclude	<i>xsd:boolean</i>	0:1	Whether rule is to exclude interchanges of journeys that match the filter criteria.
«FK»	ControlCentreRef	<i>ControlCentreRef</i>	0:1	Reference to a CONTROL CENTRE for which RULE applies.
«cntd»	FeederFilter	<i>InterchangeRuleParameter</i>	0:1	Feeder journey INTERCHANGE RULE FILTER associated with INTERCHANGE RULE.
«cntd»	DistributorFilter	<i>InterchangeRuleParameter</i>	0:1	Distributor journey INTERCHANGE RULE FILTER associated with INTERCHANGE RULE.
«cntd»	timings	<i>InterchangeRuleTiming</i>	0:*	Timings for INTERCHANGE RULE.

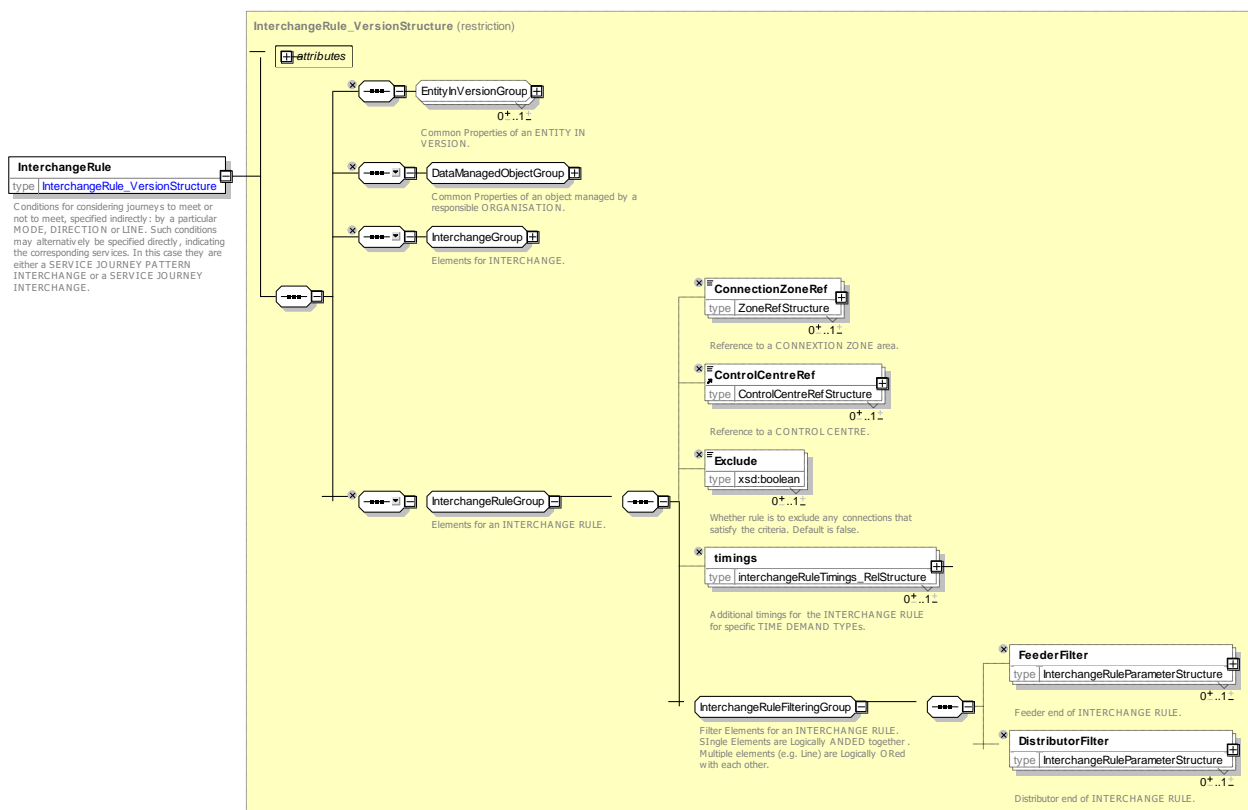


Figure 92 — InterchangeRule — XSD

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 54 – InterchangeRuleParameter – Element

Classification	Name	Type	Cardinality	Description
----------------	------	------	-------------	-------------

::>	::>	<i>DataManagedObject</i>	::>	INTERCHANGE RULE PARAMETER inherits from DATA MANAGED OBJECT
	TransportMode	<i>VehicleModeEnum</i>	0:1	Mode of CONNECTING JOURNEYS to which INTERCHANGE RULE applies.
«FK»	StopAreaRef	<i>StopAreaRef</i>	0:1	STOP AREA to which INTERCHANGE RULE applies.
«FK»	StopPlaceRef	<i>StopPlaceRef</i>	0:1	STOP PLACE feeding or distributing JOURNEY to which INTERCHANGE RULE applies.
«FK»	OperatorRef	<i>OperatorRef</i>	0:1	OPERATOR of JOURNEYS to which INTERCHANGE RULE applies.
	allLines	<i>Alllnes</i>	0:1	Applies to all LINES. v1.1
«cntd»	lineDirectionRefs	<i>LineDirectionRef</i>	0:*	List of LINES in a DIRECTION +v1.1
«FK»	LineDirectionRef	<i>LineDirectionRef</i>	0:1	A LINES in a DIRECTION +v1.1
«FK»	ExternalDirectionRef	<i>ExternalDirectionRef</i>	0:1	External reference to DIRECTION of TRANSFER to which INTERCHANGE RULE applies
«FK»	ScheduledStopPointRef	<i>ScheduledStopPointRef</i>	0:1	Start or end SCHEDULED STOP POINT in INTERCHANGE of connecting JOURNEY to which INTERCHANGE RULE applies.
«FK»	AdjacentStopPointRef	<i>ScheduledStopPointRef</i>	0:1	Prior (feeder) or onwards (distributor) STOP POINT before of journeys to which INTERCHANGE RULE applies.
«FK»	AdjacentStopPlaceRef	<i>StopPlaceRef</i>	0:1	Prior (feeder) or onwards (distributor) STOP PLACE before of journeys to which INTERCHANGE RULE applies.
«FK»	AdjacentPointRef	<i>PointRef</i>	0:1	Prior (feeder) or onwards (distributor) POINT before connection used by journey to which INTERCHANGE RULE applies.
«FK»	EndStopPointRef	<i>ScheduledStopPointRef</i>	0:1	Origin (for feed journeys) or Destination (for distributor journeys) SCHEDULED STOP POINT of connecting JOURNEY to which INTERCHANGE RULE applies.
«FK»	ServiceJourneyRef	<i>ServiceJourneyRef</i>	0:1	SERVICE JOURNEY to which INTERCHANGE RULE applies.
	MaximumInterchangeWindow	<i>xsd:duration</i>	0:1	Maximum interval for making interchange.
«cntd»	JourneyDesignator	<i>JourneyDesignator</i>	0:1	Means of identifying a JOURNEY whose ID is not known.

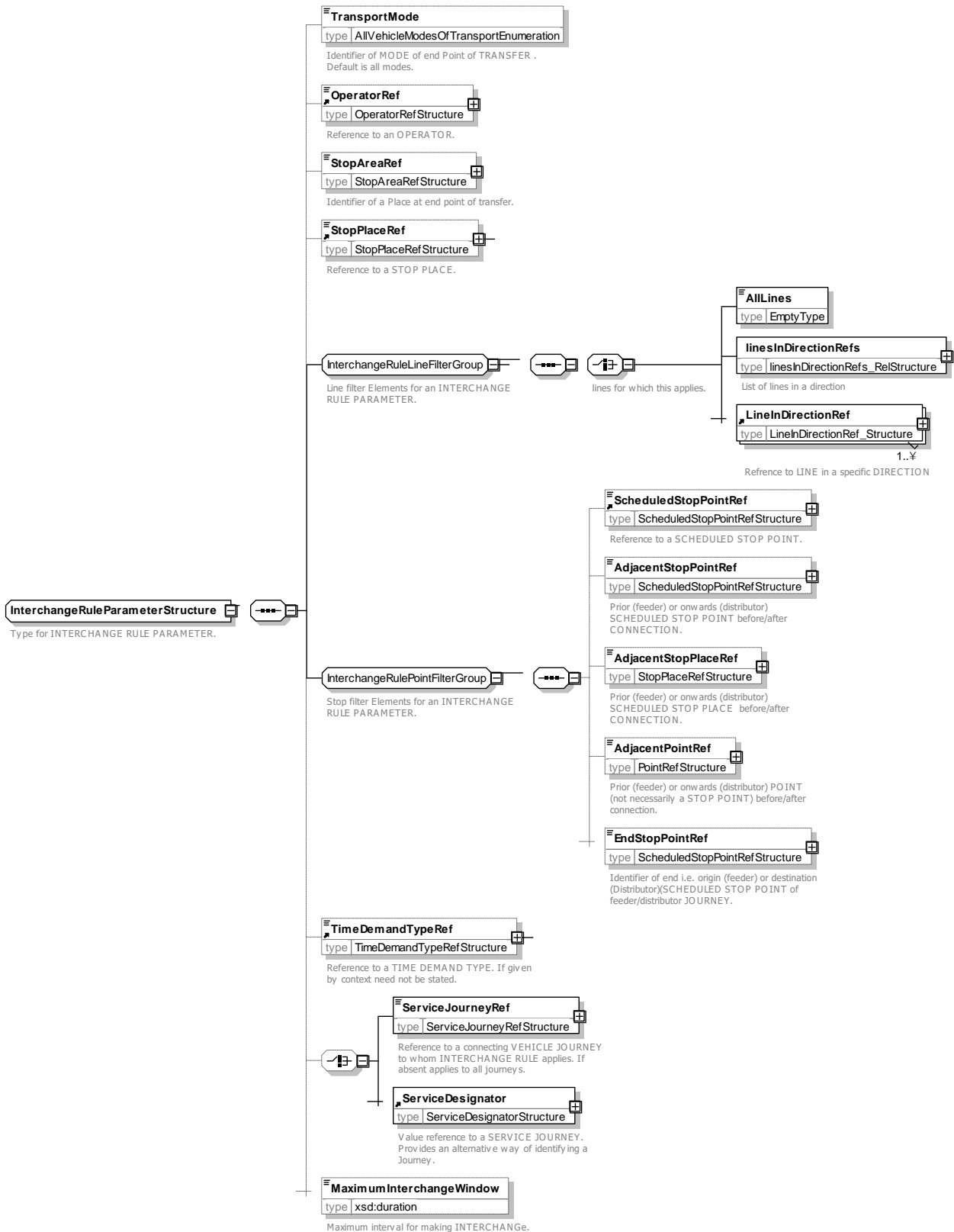


Figure 93 — *InterchangeRuleParameter* — XSD

7.2.8.3.3 LineInDirection – Model Element

A specification of the filtering criteria for LINE and line DIRECTION on the feeder or distributor end of an INTERCHANGE RULE.

Table 55 – LineInDirection – Element

Classification	Name	Type	Cardinality	Description
«FK»	LineRef	<i>LineRef</i>	0:1	LINE of JOURNEYS to which INTERCHANGE RULE applies.
«FK»	DirectionRef	<i>DirectionRef</i>	0:1	Reference to DIRECTION of TRANSFER to which INTERCHANGE RULE applies
«FK»	ExternalLineRef	<i>ExternalLineRef</i>	0:1	External reference to LINE of JOURNEYS to which INTERCHANGE RULE applies.
«FK»	ExternalDirectionRef	<i>ExternalDirectionRef</i>	0:1	External reference to DIRECTION of TRANSFER to which INTERCHANGE RULE applies

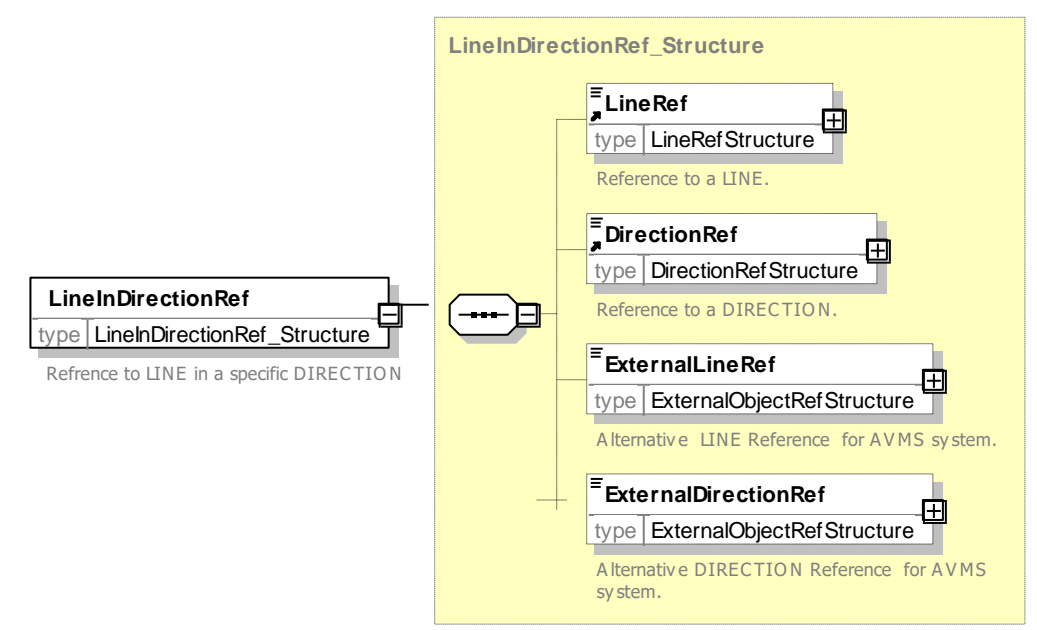


Figure 94 — LineInDirection — XSD

7.2.8.3.4 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 56 – InterchangeRuleTiming – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>VersionedChild</i>	::>	INTERCHANGE RULE TIMING inherits from VERSIONED CHILD.

«PK»	<i>id</i>	<i>InterchangeRuleTimingId Type</i>	1:1	Identifier of INTERCHANGE RULE TIMING.
«FK»	<i>InterchangeRuleRef</i>	<i>InterchangeRuleRef</i>	1:1	Identifier of INTERCHANGE RULE to which this timing belongs.
	<i>MaximumWindow</i>	<i>xsd:duration</i>	0:1	Maximum window for holding DISTRIBUTOR will wait.
	<i>StandardWaitTime</i>	<i>xsd:duration</i>	0:1	Standard Time to wait at Interchange.
	<i>MaximumWaitTime</i>	<i>xsd:duration</i>	0:1	Maximum time that DISTRIBUTOR will wait.
	<i>StandardTransferTime</i>	<i>xsd:duration</i>	0:1	Standard Time needed for transfer.
	<i>MinimumTransferTime</i>	<i>xsd:duration</i>	0:1	Minimum Time needed for transfer.
	<i>MaximumTransferTime</i>	<i>xsd:duration</i>	0:1	Maximum Time needed for transfer.

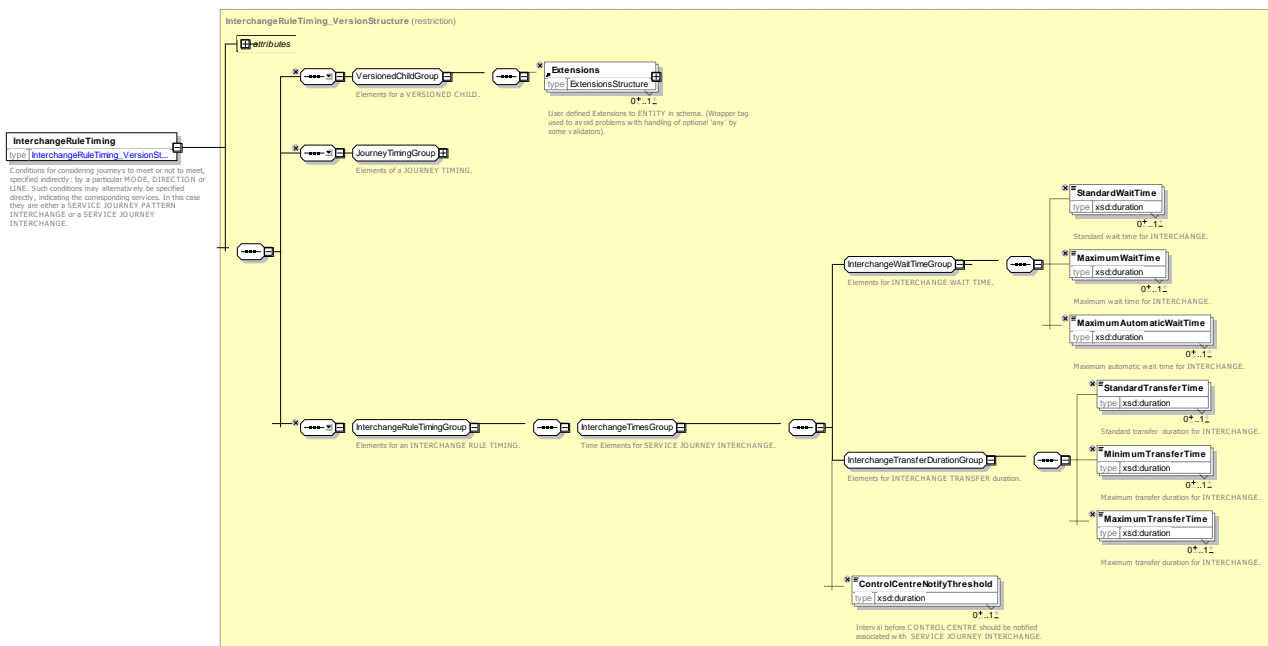


Figure 95 — *InterchangeRuleTiming* — XSD

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>
  </InterchangeRule>
</interchangeRules>

```

```

    <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

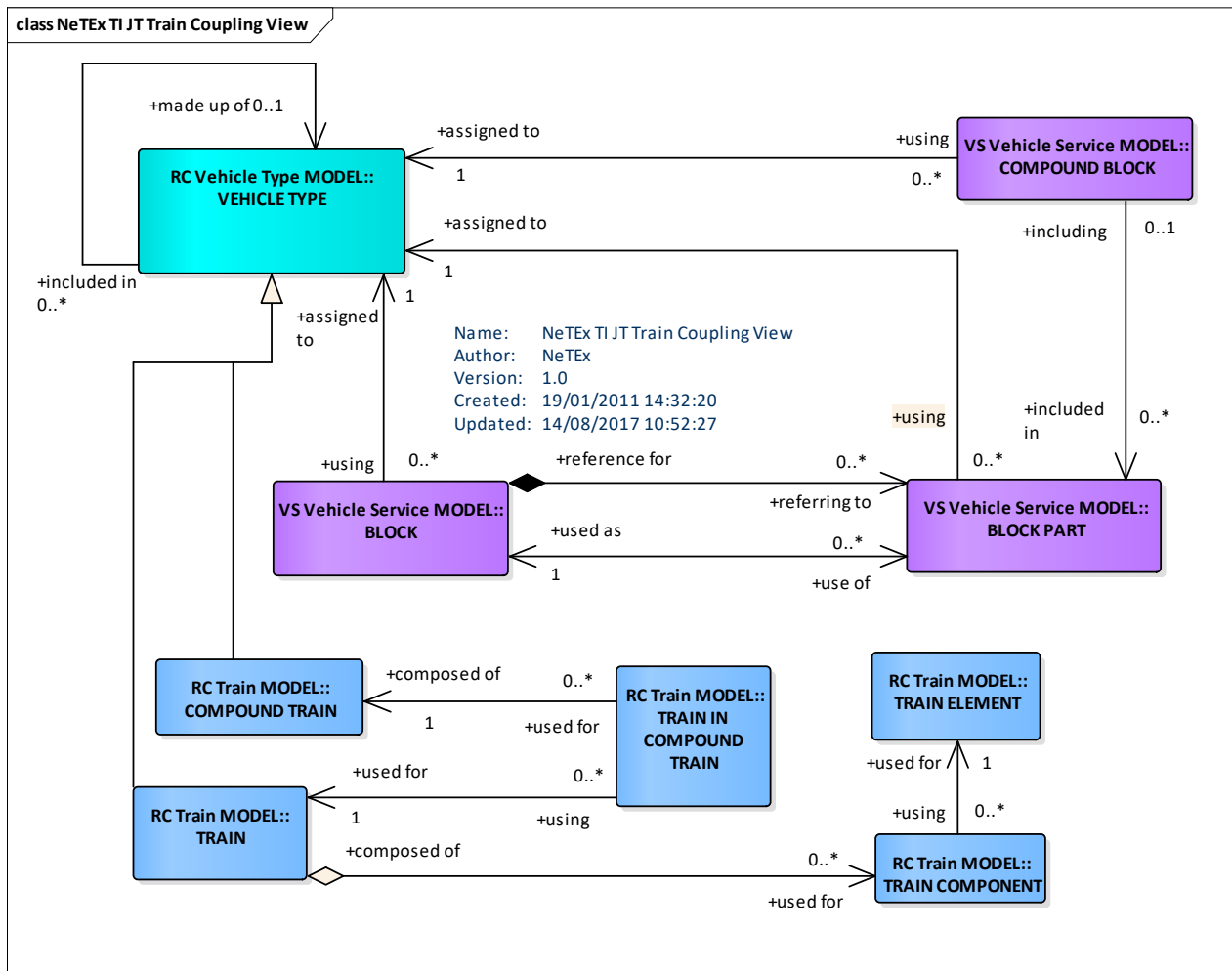


Figure 96 — Train Coupling – Conceptual MODEL (UML)

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**.

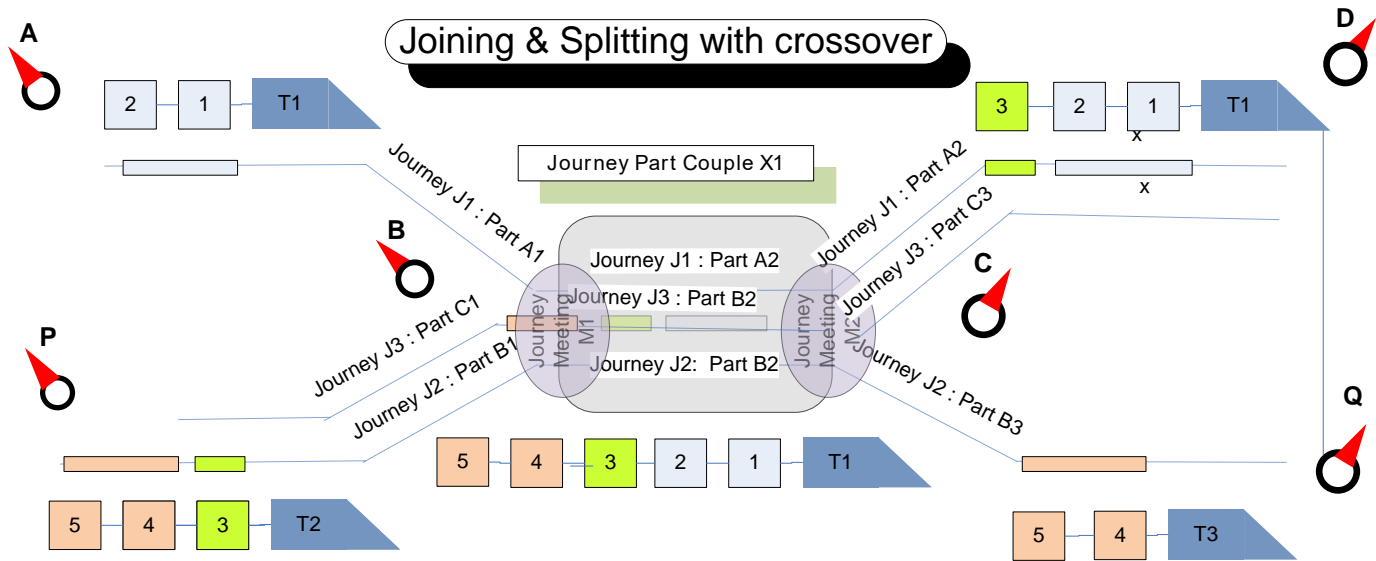


Figure 98 — Example – Joining and splitting Trains

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

Join Split example : Amsterdam – Hannover – Berlin - Warsaw

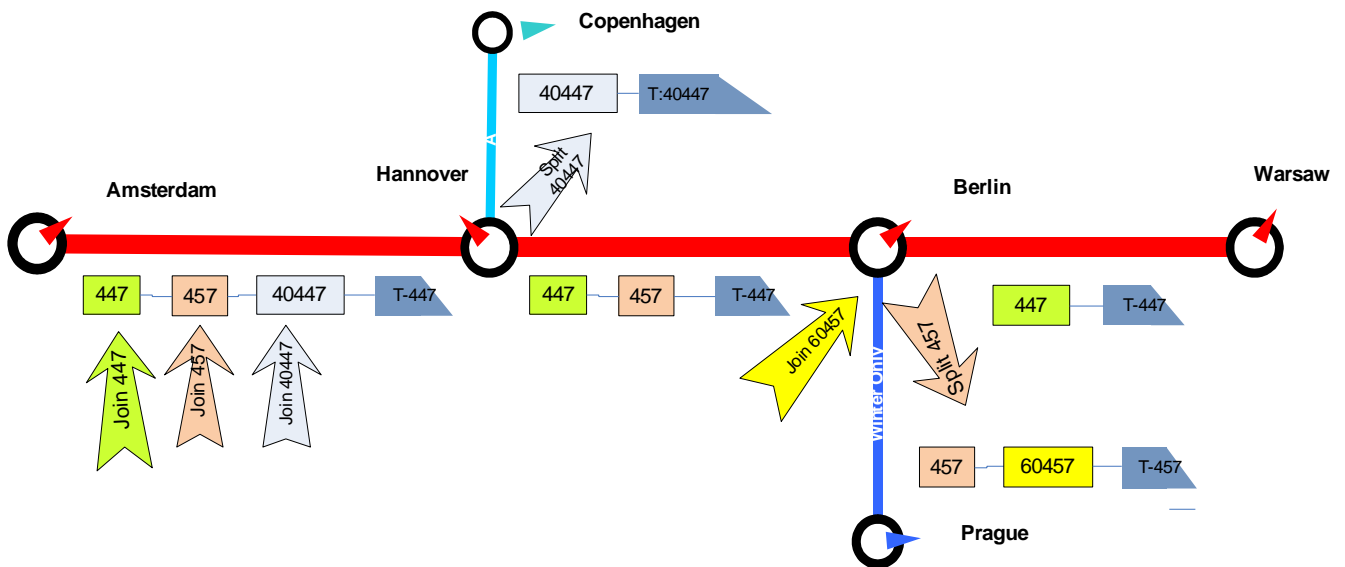


Figure 99 — Example – Joining and splitting: Amsterdam to Warsaw

This can be represented by three SERVICE JOURNEYS, each made up of several JOURNEY PARTS.

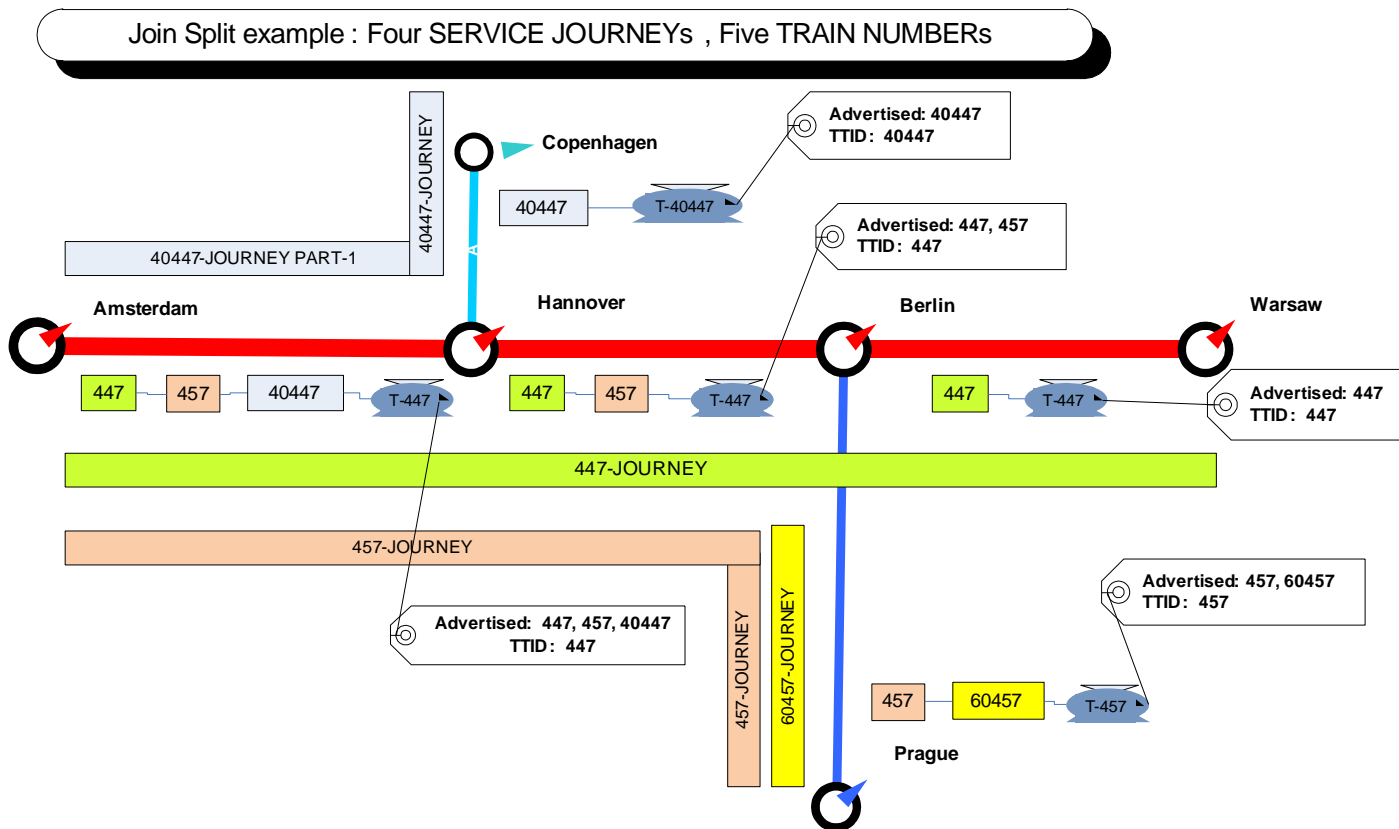


Figure 100 — Example – Journeys: Amsterdam to Warsaw

The JOURNEY PARTs are shown in the following diagram.

Join Split example : 9 JOURNEY PARTS , 3 JOURNEY PART COUPLES

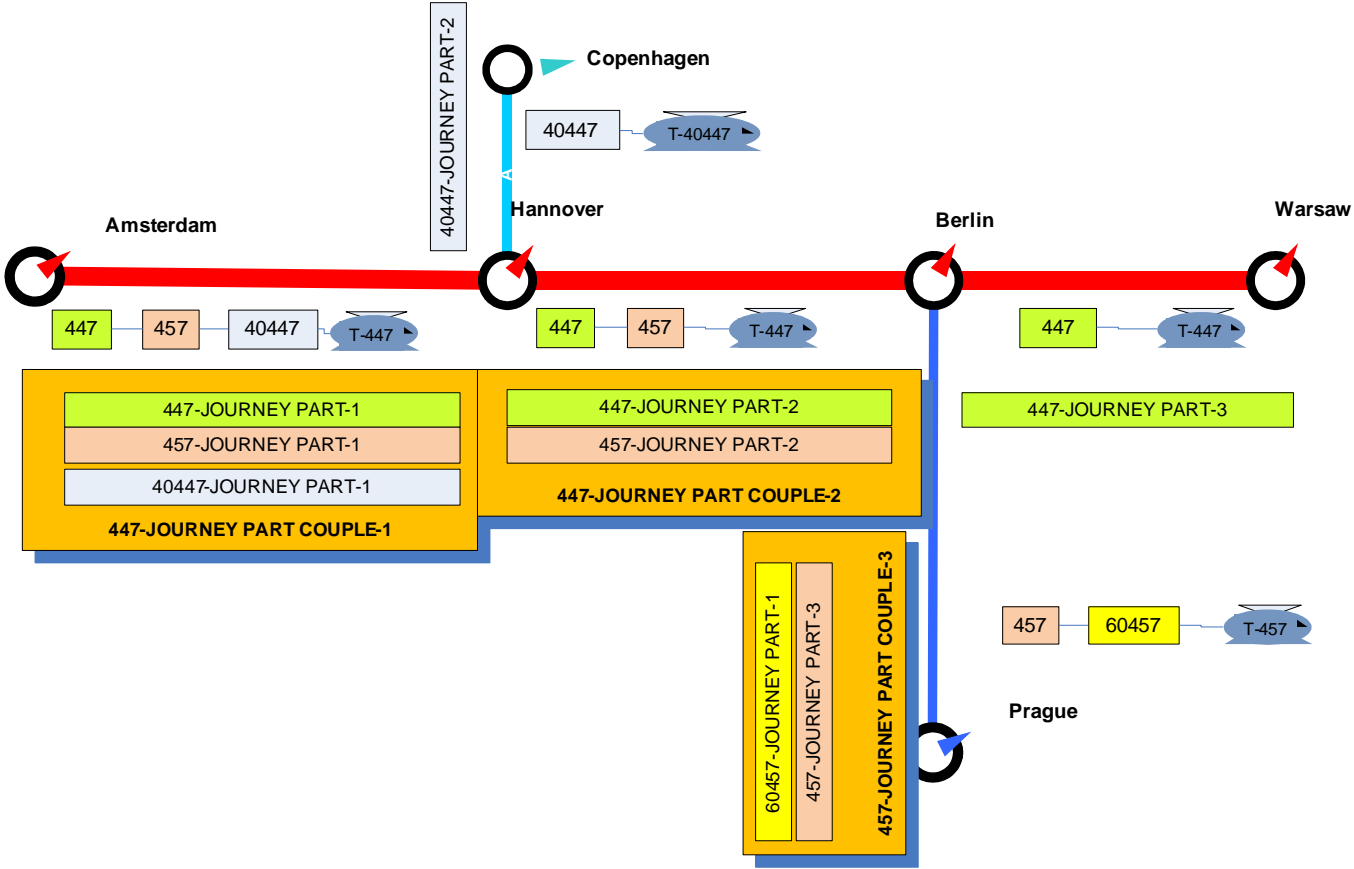


Figure 101 — Example – Journey Parts: Amsterdam to Warsaw

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

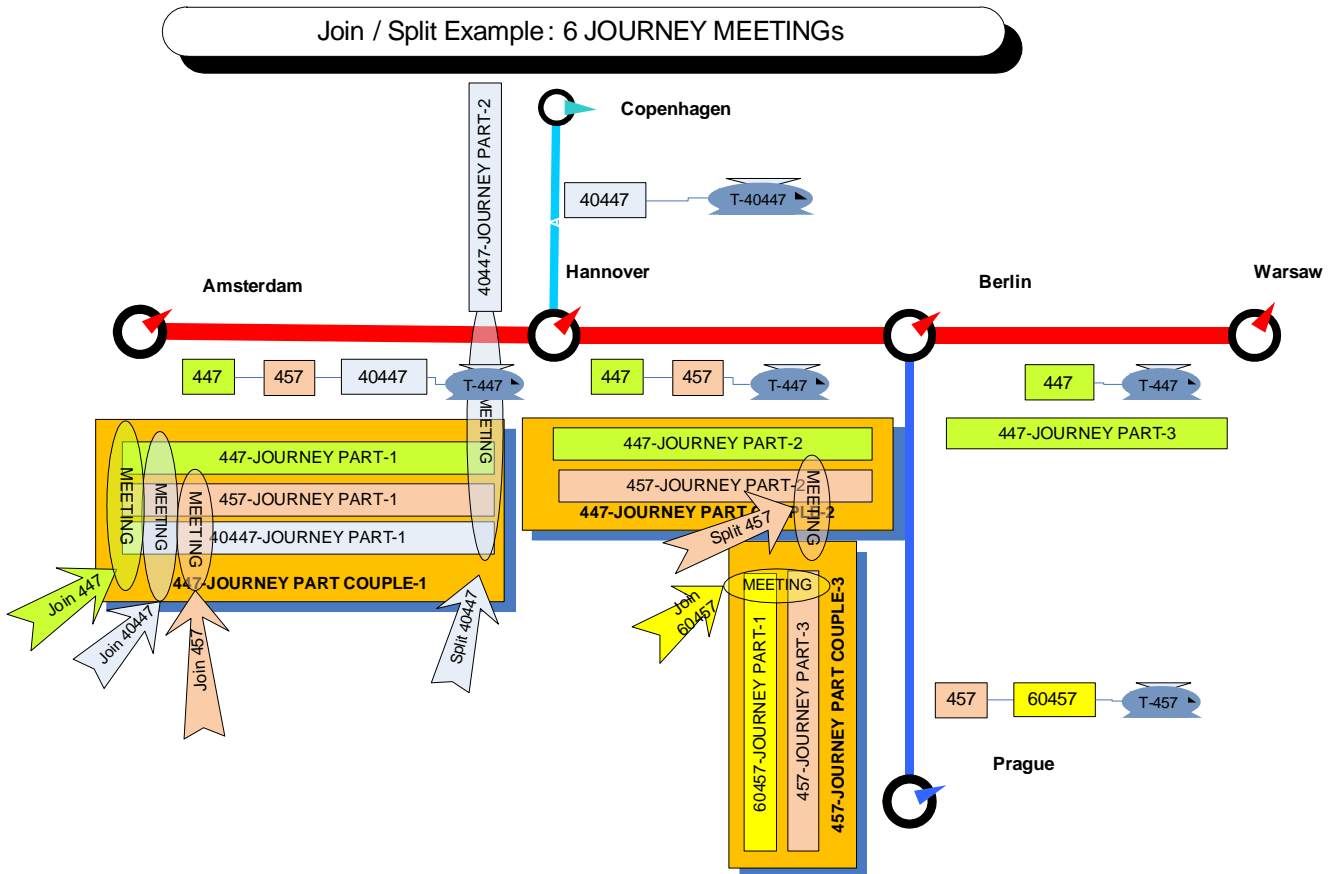


Figure 102 — Example – Journey Meetings: Amsterdam to Warsaw

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

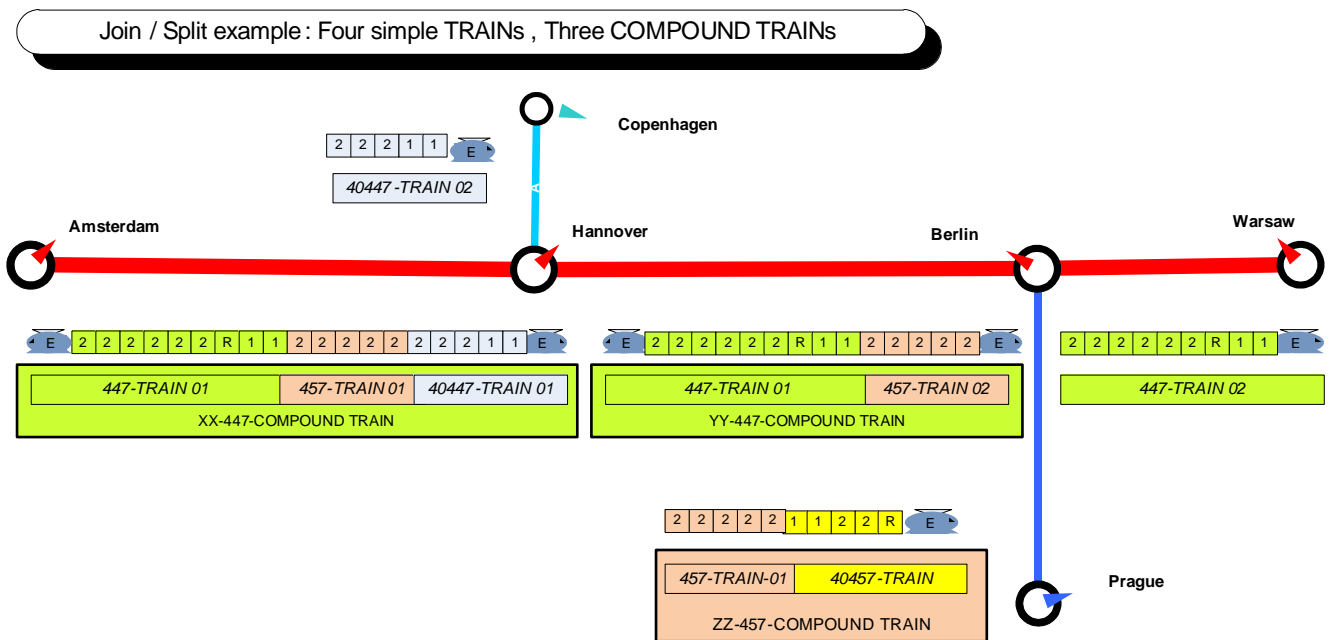


Figure 103 — Example – Journey Meetings: Amsterdam to Warsaw

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.

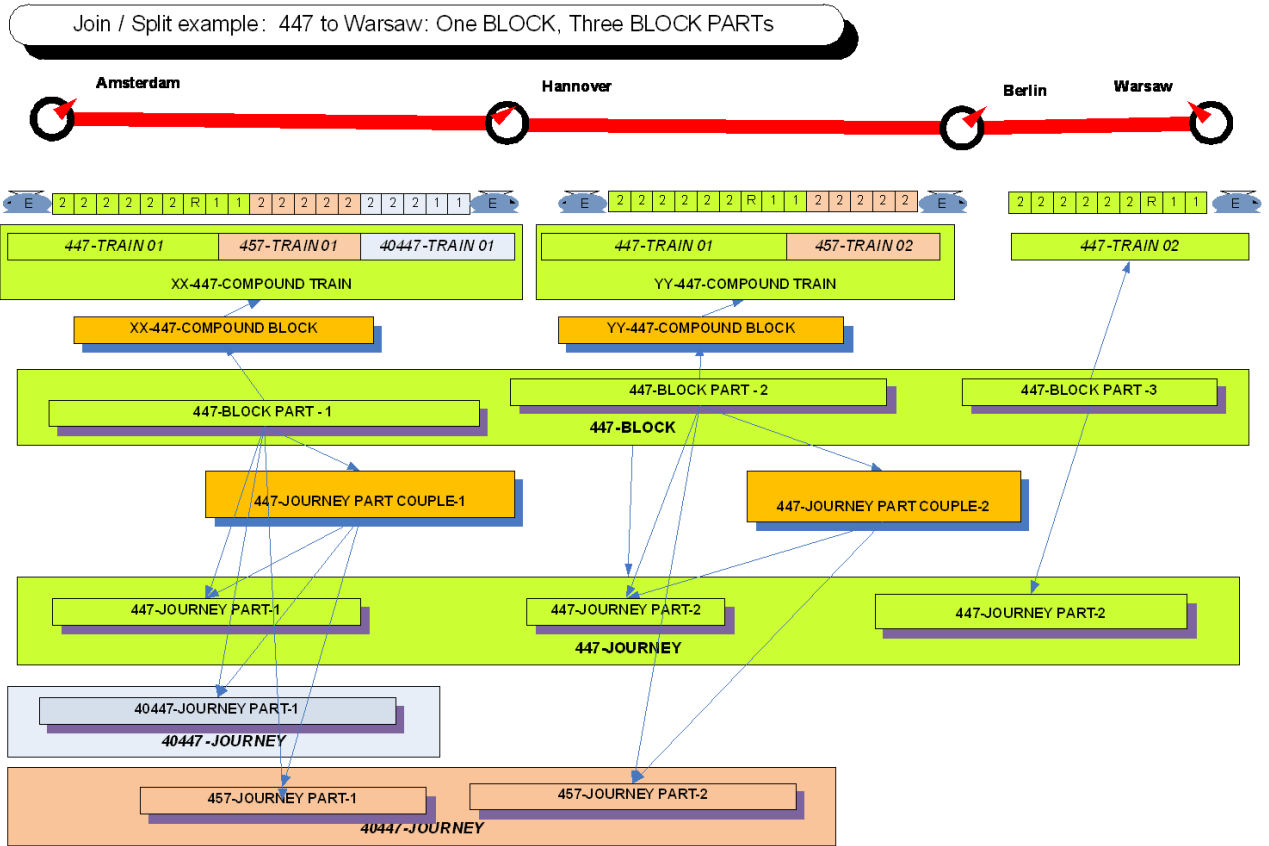


Figure 104 — Example – Train Block Parts and Journey Part Couples: Amsterdam to Warsaw

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.

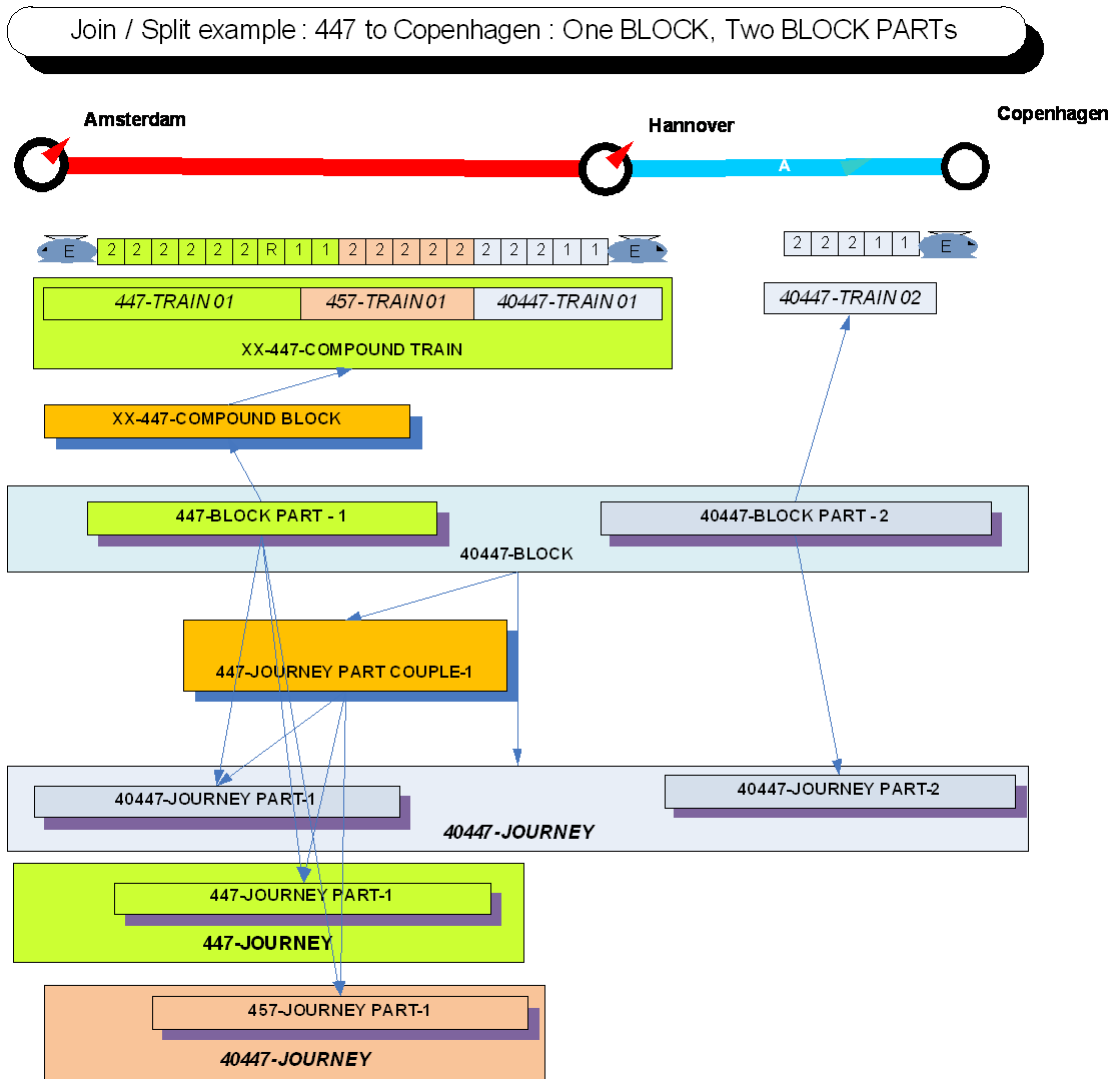


Figure 105 — Example – Train Block Parts and Journey Part Couples: Amsterdam to Copenhagen

7.2.9.3 Coupled Journey – Physical Model

The following figure shows the physical model for COUPLED JOURNEYS.

7.2.9.3.1 Train Coupling – Physical Model

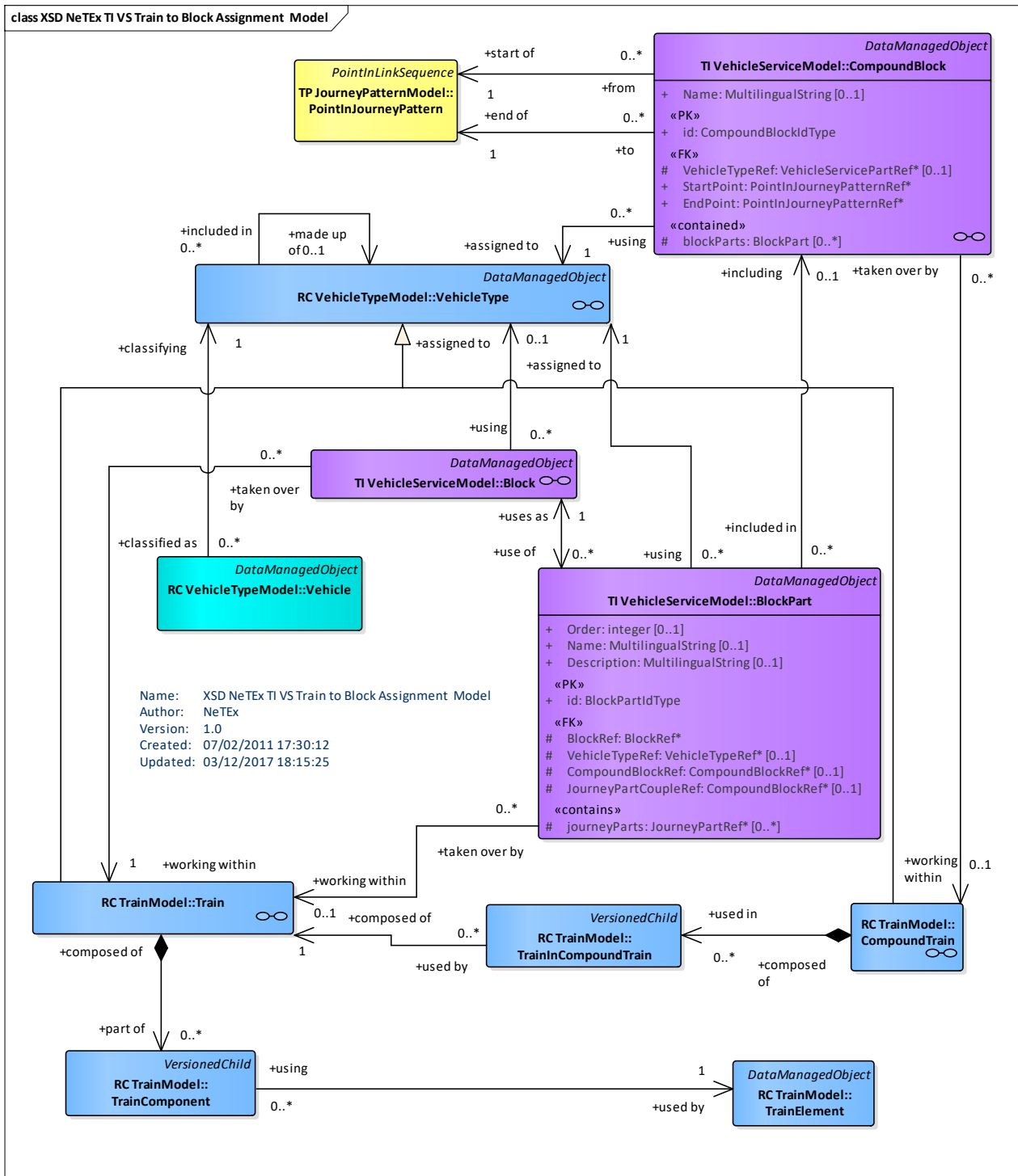


Figure 107 — Train to Block Assignment – Physical Model (UML)

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 57 – JourneyPart – Element Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	JOURNEY PART inherits from DATA MANAGED OBJECT
attr	order	<i>xsd:nonNegativeInteger</i>	0:1	Order of JOURNEY PART within Journey.
«PK»	id	<i>JourneyPartIdType</i>	1:1	Identifier of JOURNEY PART.
	Description	<i>MultilingualString</i>	0:1	Description of a JOURNEY PART.
«FK»	ParentJourneyRef	<i>VehicleJourneyRef</i>	0:1	Parent VEHICLE JOURNEY to which JOURNEY PART belongs.
«FK»	MainPartRef	<i>JourneyPartCoupleRef</i>	1:1	REFERENCE to JOURNEY PART COUPLE of main JOURNEY of a JOURNEY PART.
«FK»	JourneyPartCoupleRef	<i>JourneyPartCoupleRef</i>	0:1	REFERENCE to JOURNEY PART COUPLE of JOURNEY that JOURNEY PART joins.
«FK»	TrainNumberRef	<i>TrainNumberRef</i>	0:1	REFERENCE to TRAIN NUMBER of JOURNEY PART
«FK»	BlockPartRef	<i>BlockPartRef</i>	0:1	REFERENCE to BLOCK PART of JOURNEY that JOURNEY PART joins.
«FK»	FromStopPointRef	<i>ScheduledStopPointRef</i>	0:1	REFERENCE to from start SCHEDULED STOP POINT for JOURNEY PART.
«FK»	ToStopPointRef	<i>ScheduledStopPointRef</i>	0:1	REFERENCE to end SCHEDULED STOP POINT for JOURNEY PART.
	StartTime	<i>xsd:time</i>	1:1	Start time of a JOURNEY PART.
	StartTimeDayOffset	<i>DayOffsetType</i>	0:1	Number of days after journey start time that start time is.
	EndTime	<i>xsd:time</i>	1:1	End time of a JOURNEY PART.
	EndTimeDayOffset	<i>DayOffsetType</i>	0:1	Number of days after journey start time that end time is.
	VehicleOrientation	<i>xsd:boolean</i>	0:1	Orientation of the vehicle carrying out the JOURNEY PART compared to the definition of the corresponding VEHICLE TYPE. true for forward.
«FK»	PurposeOfJourneyPartitionRef	<i>PurposeOfJourneyPartitionRef</i>	0:1	PURPOSE of JOURNEY PARTITION.
«cntd»	facilities	<i>ServiceFacilitySet</i>	0:*	Facilities available during a JOURNEY PART.

«cntd»	<i>journeyPartPositions</i>	<i>journeyPartPositions</i>	0:1	Positions in Train of 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.

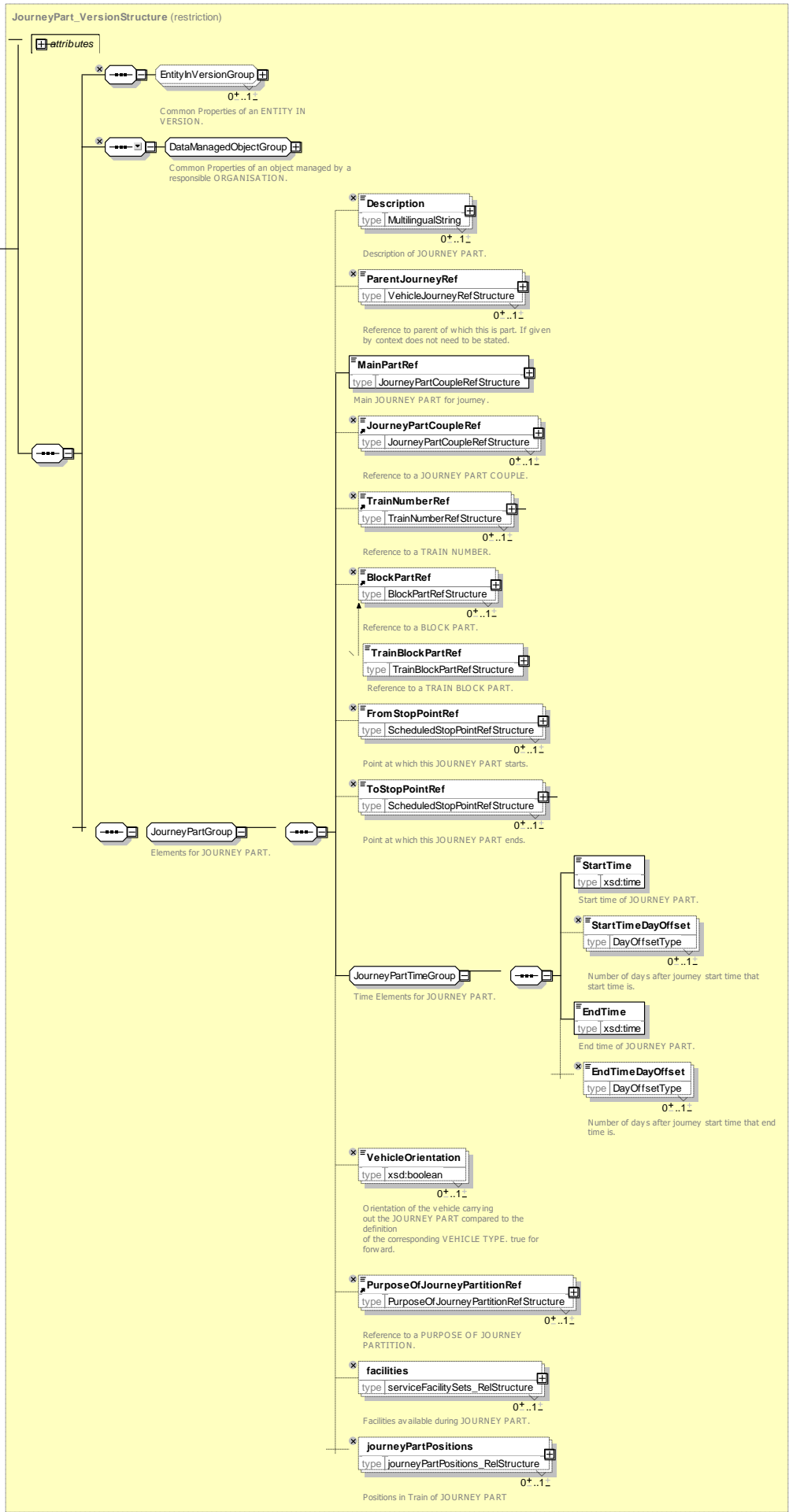


Figure 108 — *JourneyPart* — XSD

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 58 – *CoupledJourney* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	COUPLED JOURNEY inherits from DATA MANAGED OBJECT
«PK»	id	<i>CoupledJourneyIdType</i>	1:1	Identifier of COUPLED JOURNEY.
	Name	<i>MultilingualString</i>	0:1	Name of COUPLED JOURNEY. +v1.1
	Description	<i>MultilingualString</i>	0:1	Description of COUPLED JOURNEY.
«FK»	TrainBlockRef	<i>BlockRef</i>	0:1	BLOCK supplying that COUPLE JOURNEY.
«cntd»	journeys	<i>VehicleJourney</i>	0:*	JOURNEYS linked by of COUPLED JOURNEY.

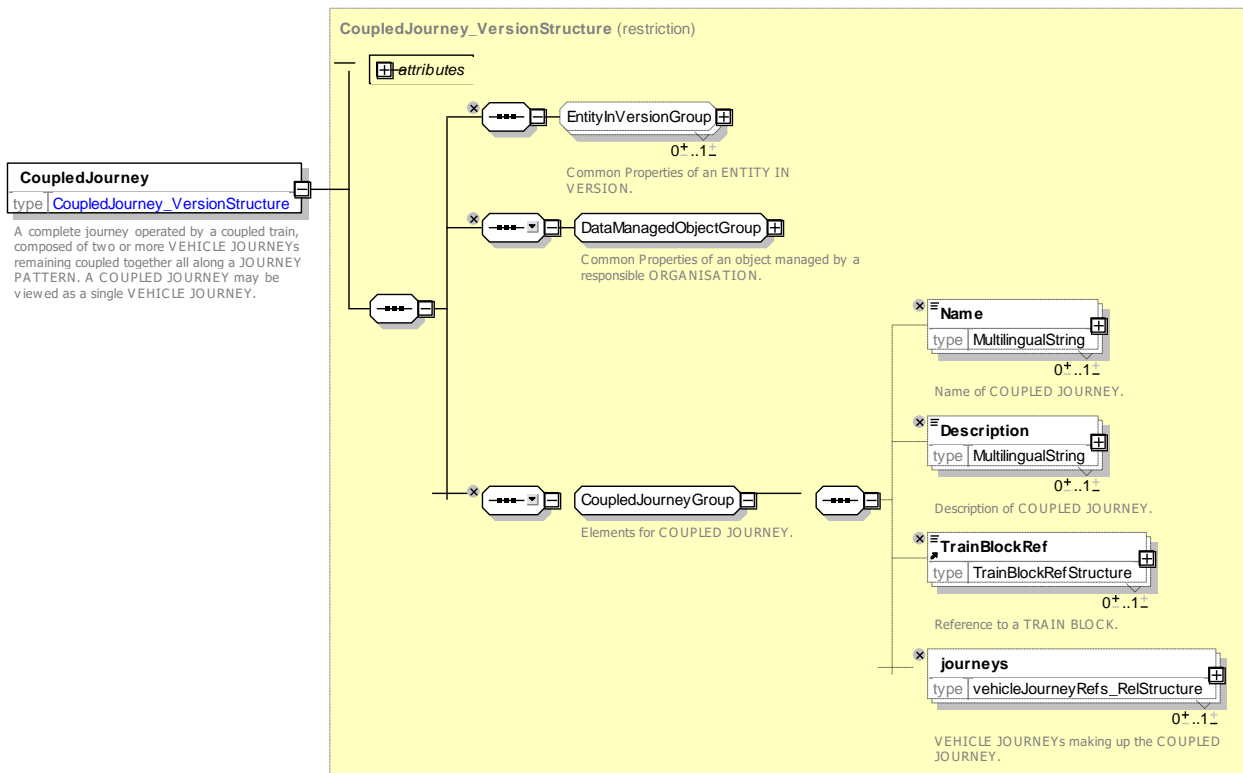


Figure 109 — *CoupledJourney* — XSD

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 59 – JourneyPartCouple – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	JOURNEY PART inherits from DATA MANAGED OBJECT
«PK»	id	<i>JourneyPartIdType</i>	1:1	Identifier of JOURNEY PART.
	Description	<i>MultilingualString</i>	0:1	Description of a JOURNEY PART.
	StartTime	<i>xsd:time</i>	1:1	Start time of a JOURNEY PART.
	StartTimeOffset	<i>DayOffsetType</i>	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day. V1.1
	EndTime	<i>xsd:time</i>	1:1	End time of a JOURNEY PART.
	EndTimeOffset	<i>DayOffsetType</i>	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day. V1.1
«FK»	FromStopPointRef	<i>ScheduledStopPointRef</i>	1:1	REFERENCE to from start SCHEDULED STOP POINT for JOURNEY PART.
«FK»	ToStopPointRef	<i>ScheduledStopPointRef</i>	1:1	REFERENCE to end SCHEDULED STOP POINT for JOURNEY PART.
«FK»	MainPartRef	<i>JourneyPartCoupleRef</i>	1:1	REFERENCE to JOURNEY PART COUPLE of main JOURNEY of a JOURNEY PART.
«FK»	JourneyParts	<i>JourneyParts</i>	0:1	JOURNEY PARTs of JOURNEY that JOURNEY PART joins.
«FK»	TrainNumberRef	<i>TrainNumberRef</i>	0:1	REFERENCE to TRAIN NUMBER of JOURNEY PART

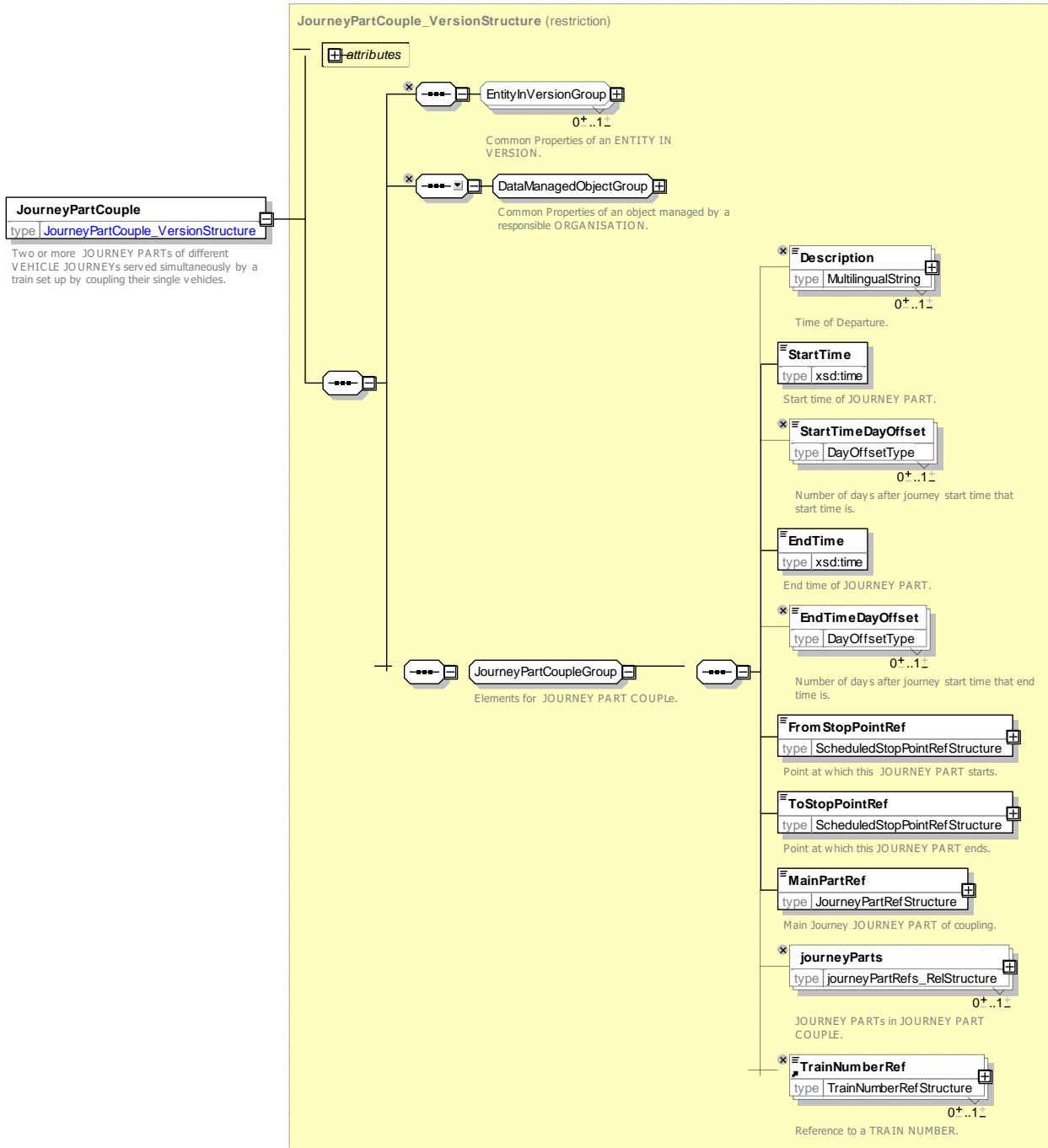


Figure 110 — JourneyPartCouple — XSD

7.2.9.4.4 PurposeOfJourneyPartition – Model Element

A classification of JOURNEY PARTITIONs according to their functional purpose.

Table 60 – JourneyPartPosition – Element

Classification	Name	Type	Cardinality	Description
::>	::>	TypeOfValue	::>	PURPOSE OF JOURNEY PARTITION inherits from TYPE OF VALUE

«PK»	<i>id</i>	<i>PurposeOfJourneyPartitionIdType</i>	1:1	Identifier of PURPOSE OF JOURNEY PARTITION.
------	-----------	--	-----	---

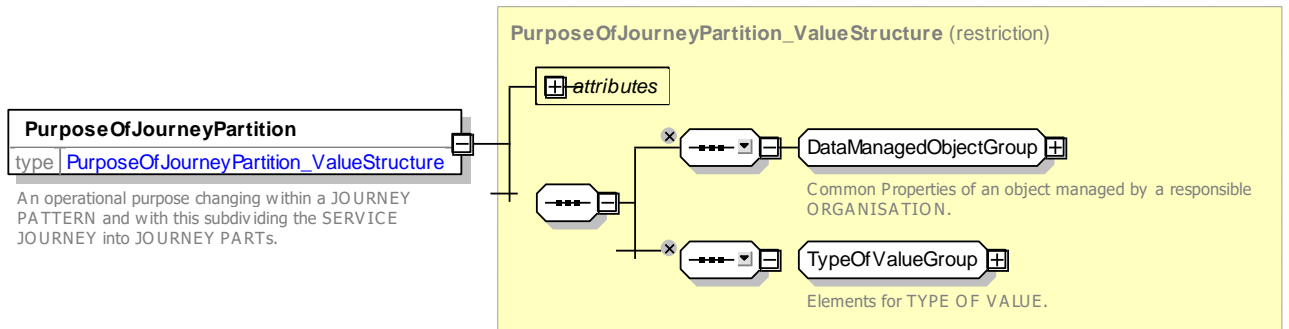


Figure 111 — *PurposeOfJourneyPartition* — XSD

7.2.9.4.5 JourneyPartPosition – Model Element

Position in train of JOURNEY PART from a given stop. May change in the course of the journey as train components are couple and uncoupled. +v1.1..

Table 61 – *JourneyPartPosition*– Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>VersionedChild</i>	::>	JOURNEY PART POSITION inherits from VERSIONED CHILD.
«PK»	<i>id</i>	<i>JourneyPartPositionIdType</i>	1:1	I JOURNEY PART POSITION.
«FK»	<i>ParentJourneyPartRef</i>	<i>ParentJourneyPartRef</i>	0:1	Part JOURNEY PART
«FK»	<i>ScheduledStopPointRef</i>	<i>ScheduledStopPointRef</i>	0:1	SCHEDULED STOP POINT from which the position is valid.
»	<i>PositionInTrain</i>	<i>xsd:integer</i>	0:*	Position of JOURNEY PART in TRAIN - from specified SCHEDULED STOP POINT until otherwise stated.

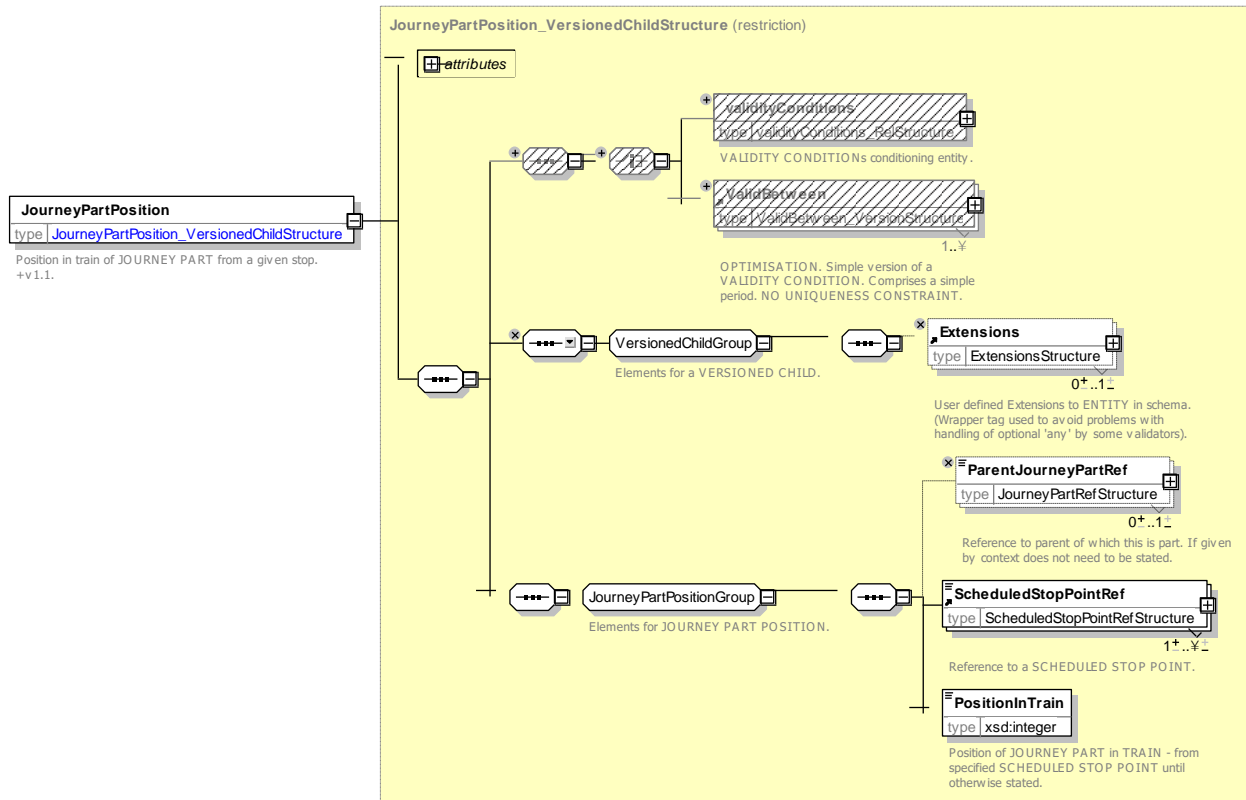


Figure 112 — JourneyPartPosition — XSD

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>
    </JourneyPart>
  </parts>
</ServiceJourney>
```



```

Ref>      <PurposeOfJourneyPartitionRef ref="hde:coupling">coupling</PurposeOfJourneyPartition-
Ref>      </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</PurposeOfJourneyPartition-
Ref>      </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</PurposeOfJourneyPartition-
Ref>      </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_amster-
dam"/>
              <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>
        <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 be 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 whole 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.

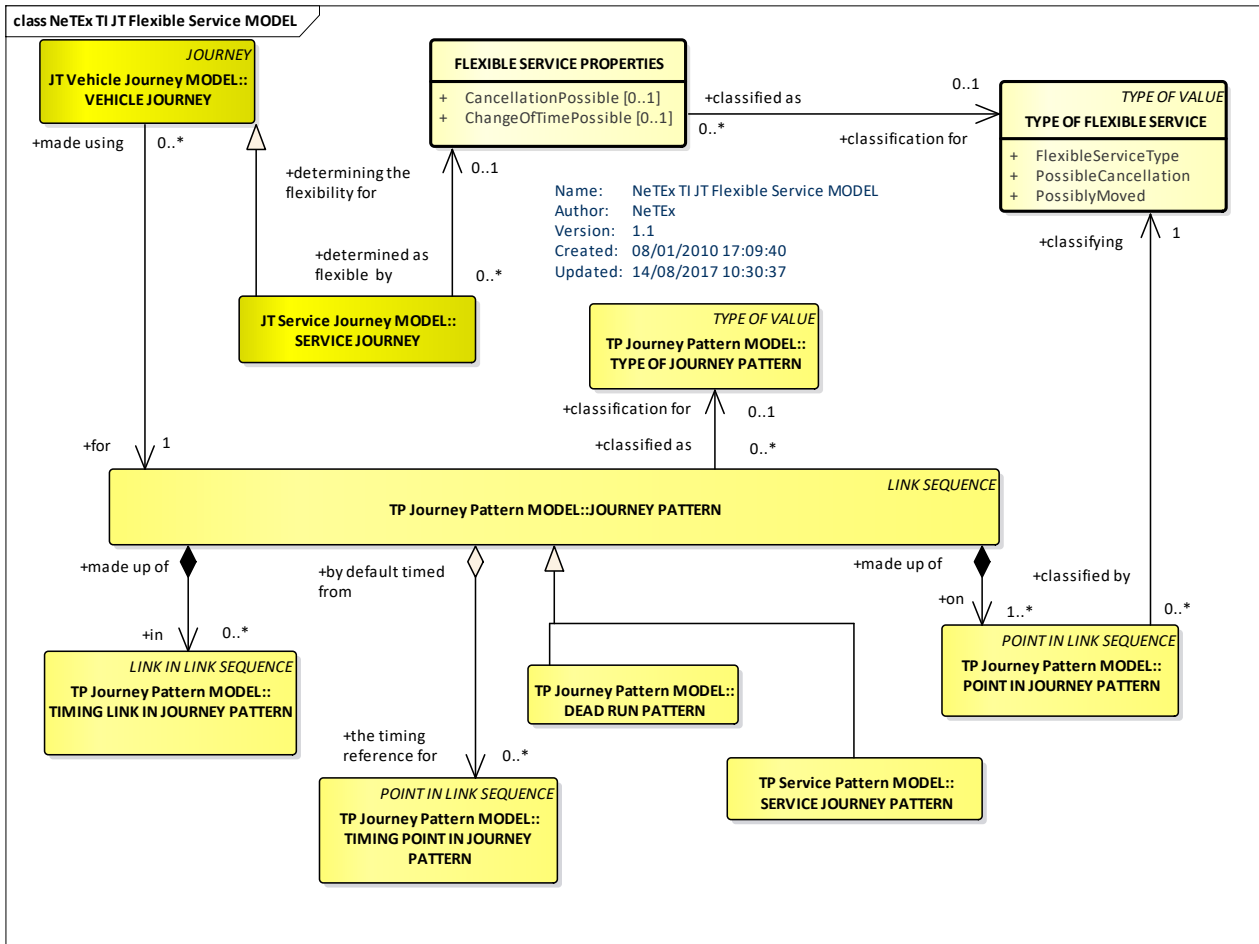


Figure 113 — Flexible Service – Conceptual MODEL (UML)

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 there 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.

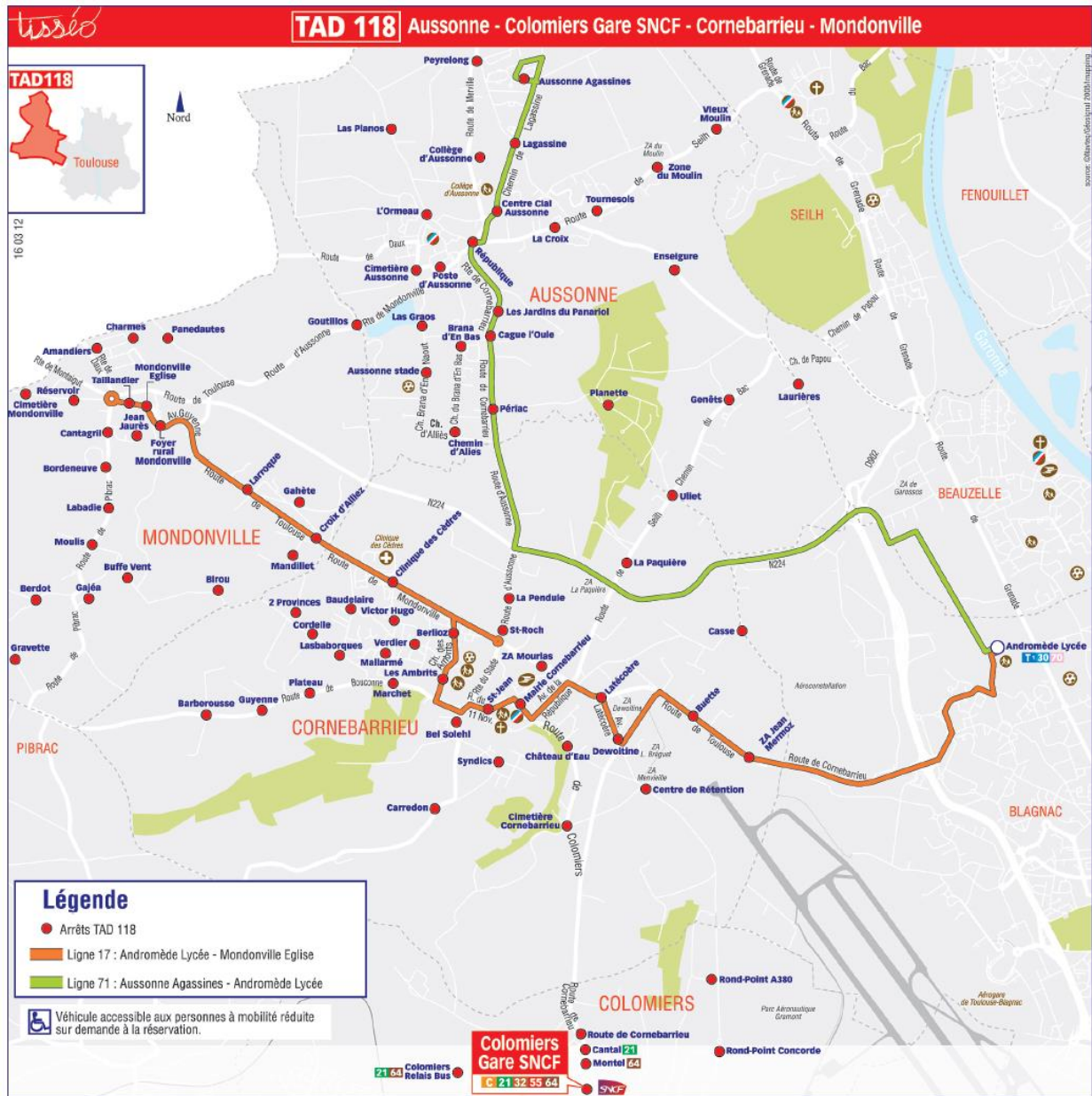


Figure 114 —Flexible service example: Tisseo TAD 118 (with courtesy of Tisseo, data from August 2012)

7.2.10.2.1 Flexible Service – Type of Flexible Service

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

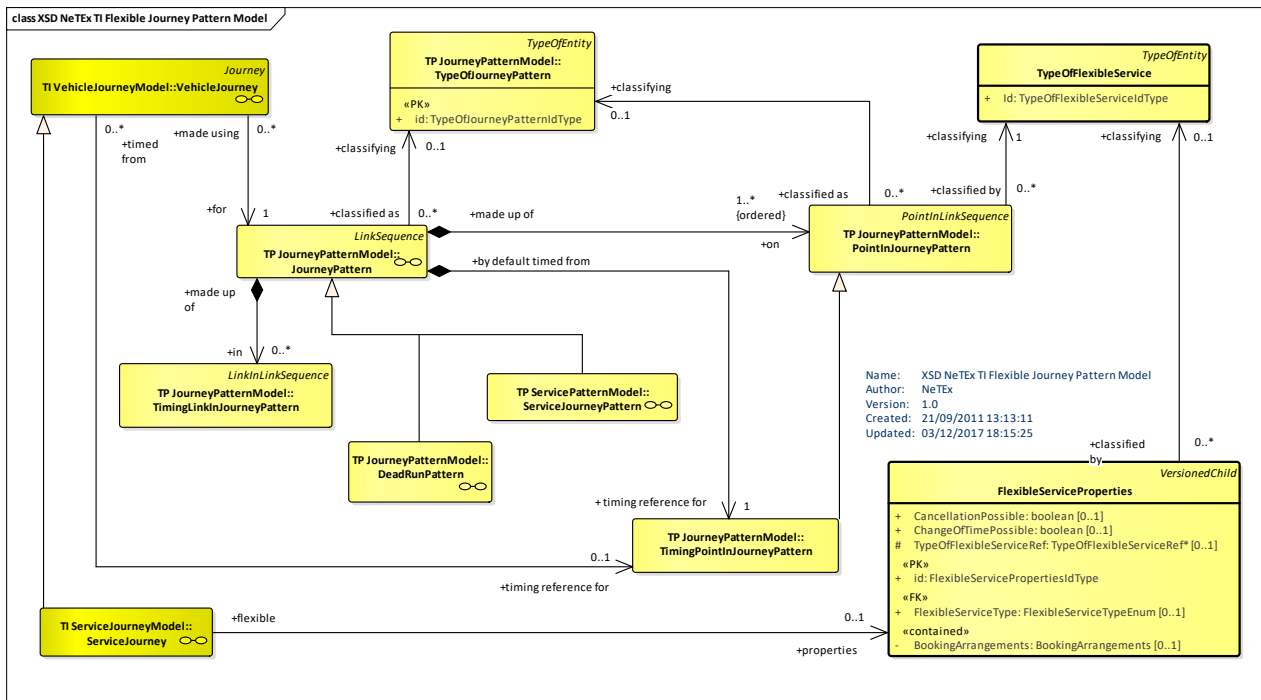


Figure 115 — Type of Flexible Service – Physical Model (UML)

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.

Table 62 — FlexibleServiceProperties – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	FLEXIBLE SERVICE PROPERTIES inherits from DATA MANAGED OBJECT
«PK»	id	<i>FlexibleServicePropertiesIdType</i>	1:1	Identifier of FLEXIBLE SERVICE PROPERTIES.
«FK»	TypeOfFlexibleServiceRef	<i>TypeOfFlexibleServiceRef</i>	1:1	TYPE OF FLEXIBLE SERVICE for which these are the properties.
«cntd»	BookingArrangements	<i>BookingArrangements</i>	0:1	Booking contact for FLEXIBLE SERVICE.
	FlexibleServiceType	<i>FlexibleServiceTypeEnum</i>	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.
	CancellationPossible	<i>xsd:boolean</i>	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.)
	ChangeOfTimePossible	<i>xsd:boolean</i>	0:1	ChangeOfTimePossible flag tells that it may be moved (usually passing time update to optimise the service) (need for both PossibleCancelation and ChangeOfTimePossible since both can be true... these 2 attributes being really need only forDynamicPassingTimes).

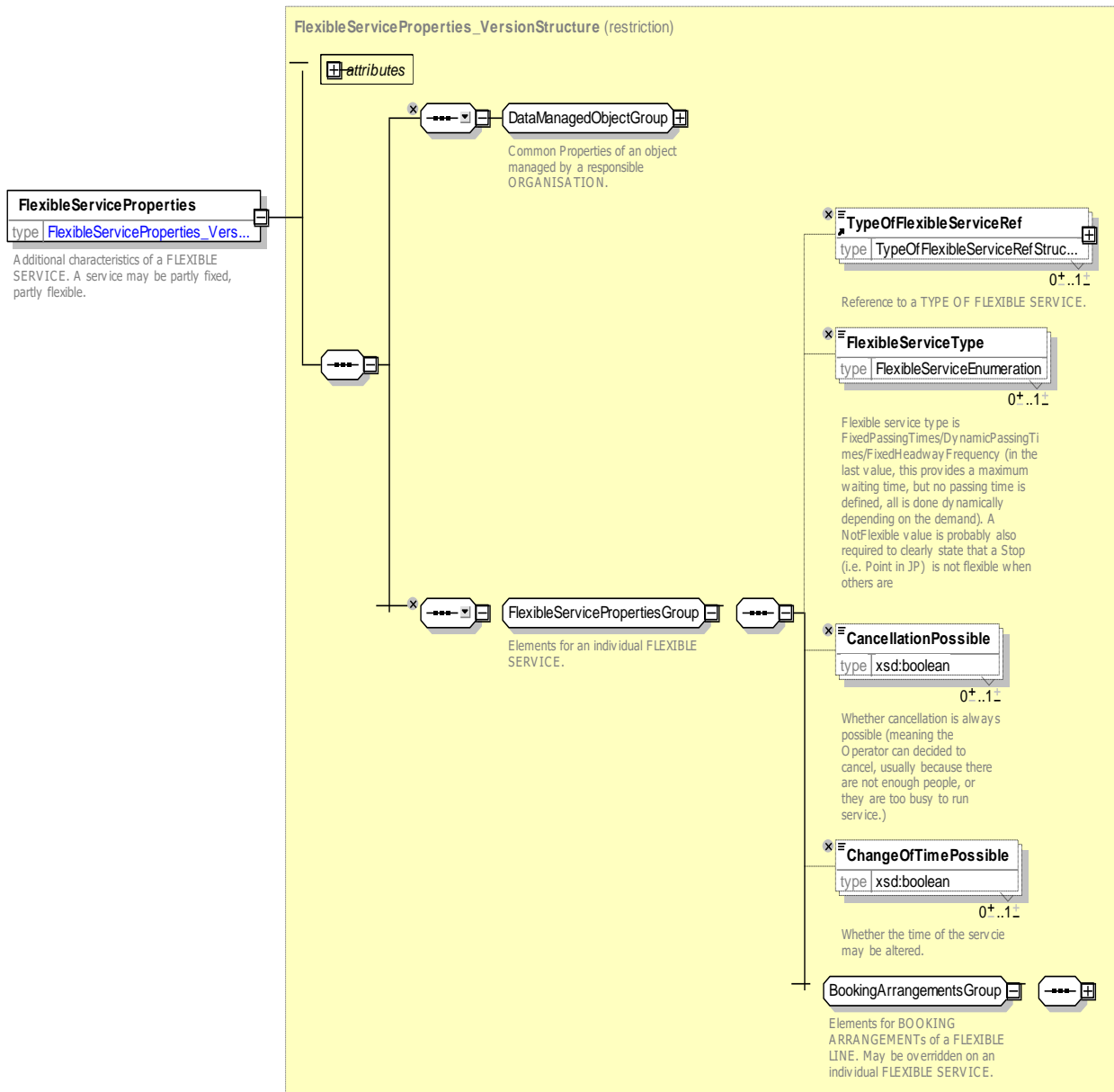


Figure 117 — FlexibleServiceProperties — XSD

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 63 – 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	<i>ScheduledStopPointRef</i>	0:1	Scheduled STOP POINT that is assigned to a FLEXIBLE STOP PLACE by this FLEXIBLE STOP ASSIGNMENT.
«FK»	FlexibleStop-PlaceRef	<i>FlexibleStopPlaceRef</i>	0:1	FLEXIBLE STOP PLACE that is assigned to a SCHEDULED STOP POINT by this FLEXIBLE STOP ASSIGNMENT.
«FK»	FlexibleQuayRef	<i>FlexibleQuayRef</i>	0:1	FLEXIBLE QUAY that is assigned to a SCHEDULED STOP POINT by this FLEXIBLE STOP ASSIGNMENT.

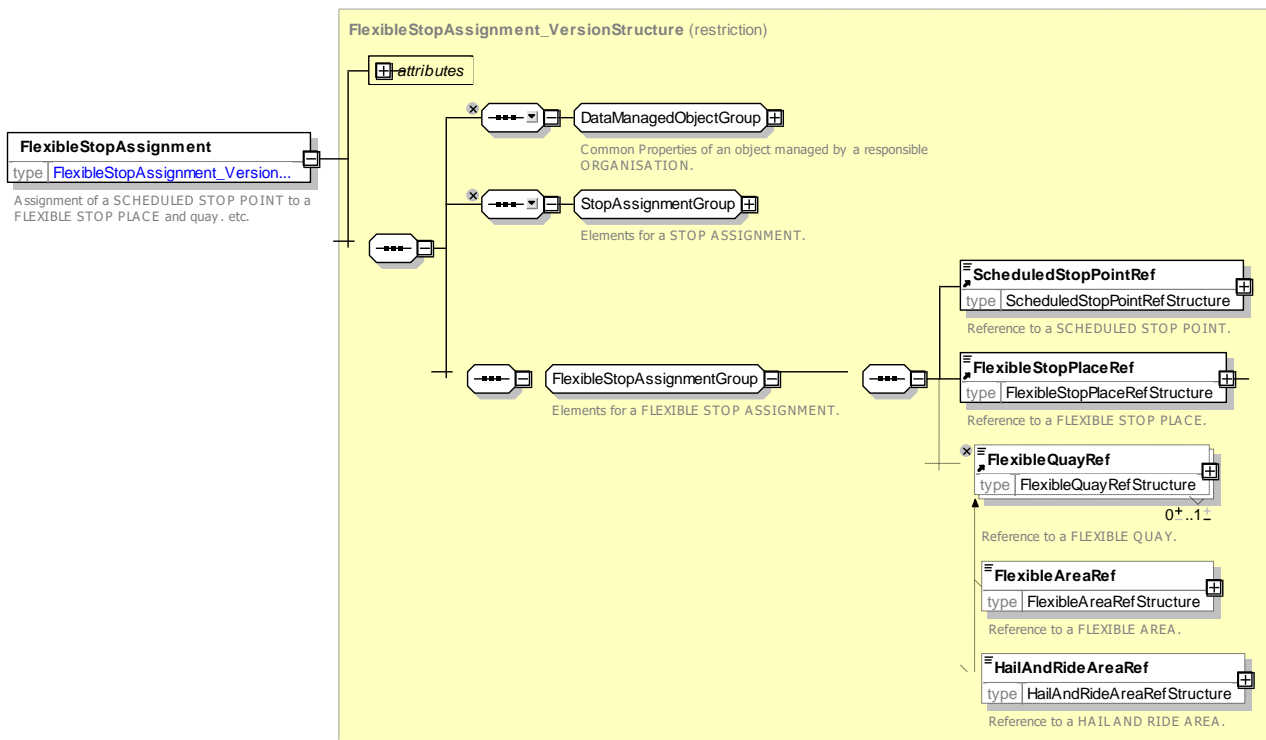


Figure 118 — FlexibleStopAssignment — XSD

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 64 – 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. <ul style="list-style-type: none"> — - 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

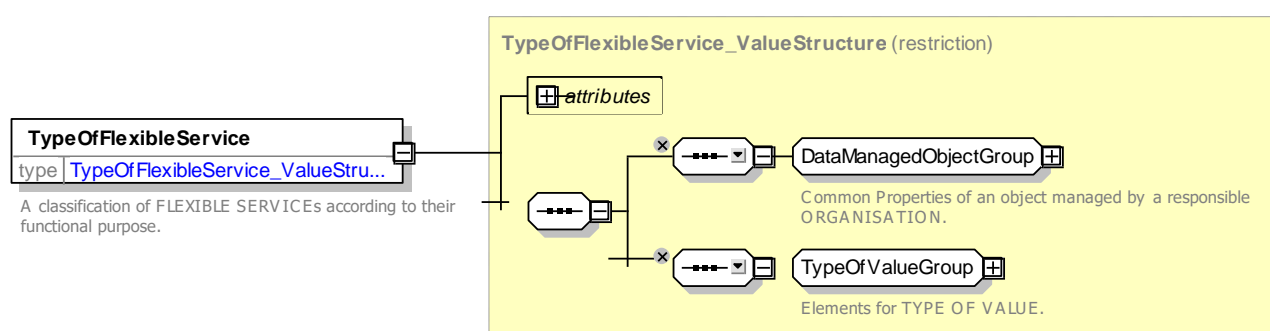


Figure 119 — TypeOfFlexibleService — XSD

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

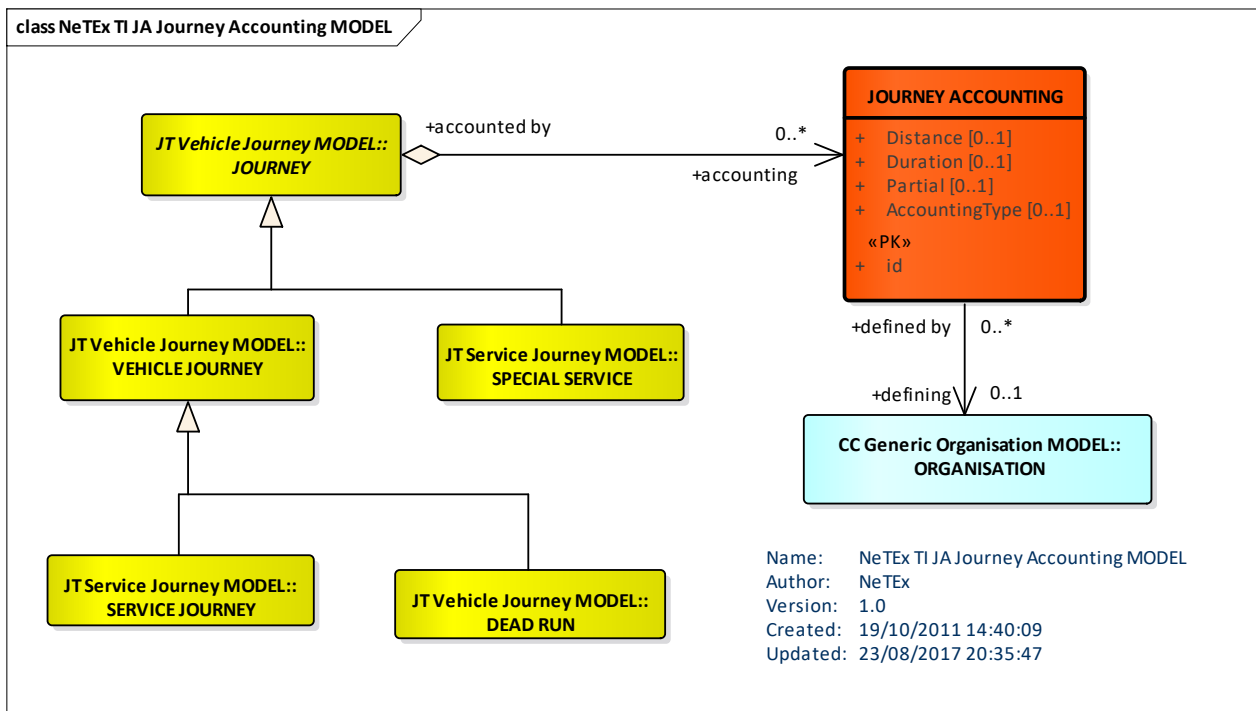


Figure 120 — Journey Accounting – Conceptual MODEL (UML)

7.2.11.2 Journey Accounting – Physical Model

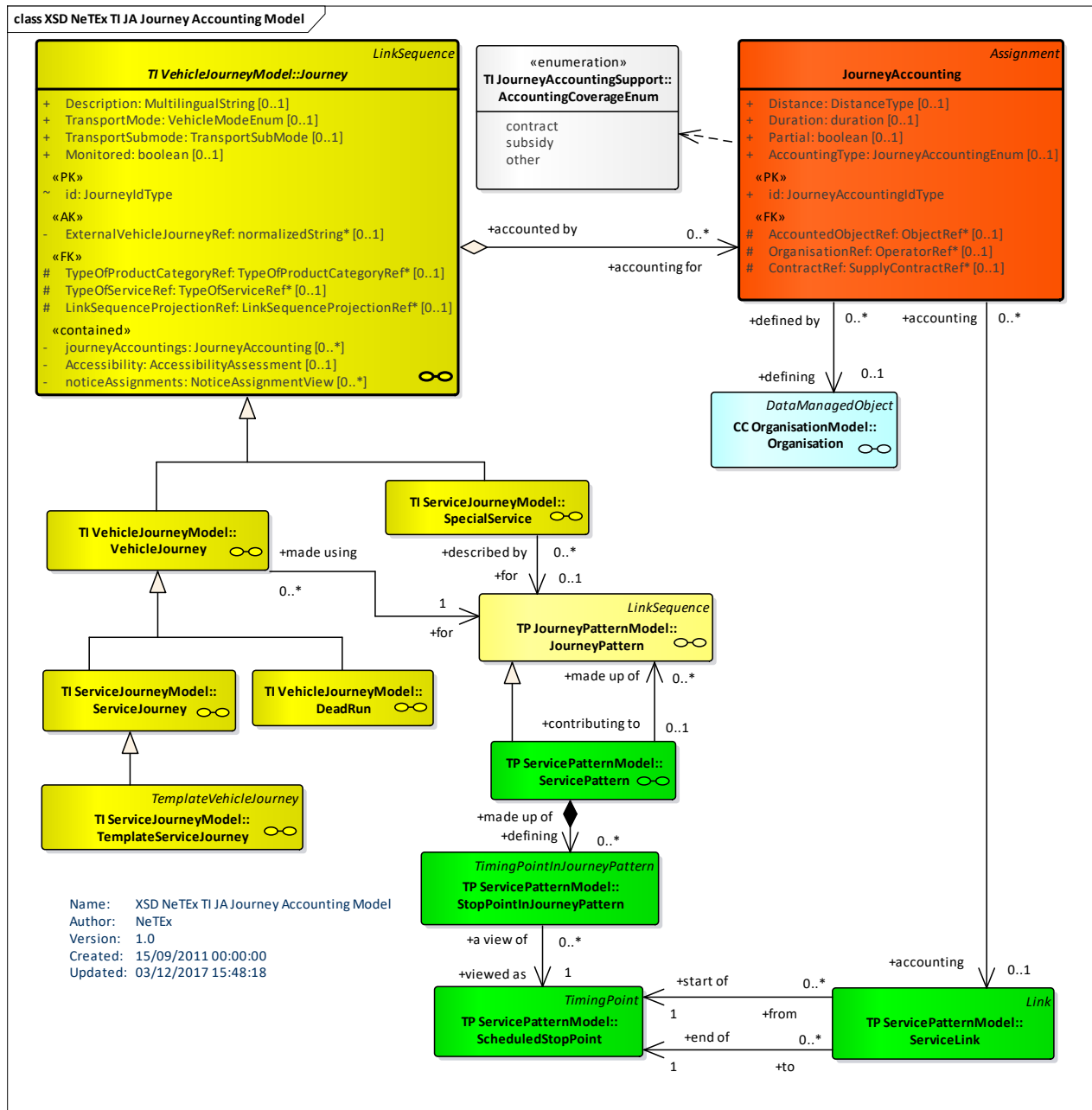


Figure 121 — Journey Accounting – Physical Model (UML)

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 65 – JourneyAccounting – Element

Classification	Name	Type	Cardinality	Description

::>	::>	<i>Assignment</i>	::>	JOURNEY ACCOUNTING inherits from ASSIGNMENT.
«PK»	id	<i>JourneyAccountingIdType</i>	1:1	Identifier of JOURNEY ACCOUNTING.
	Name	<i>MultilingualString</i>	0:1	Name of JOURNEY ACCOUNTING.
	Distance	<i>DistanceType</i>	0:1	Distance to us in for JOURNEY ACCOUNTING.
	Duration	<i>xsd:duration</i>	0:1	Duration to use for JOURNEY ACCOUNTING.
	Partial	<i>xsd:boolean</i>	0:1	Whether all or some of journey is subject to ACCOUNTING.
	AccountingType	<i>JourneyAccountingEnum</i>	0:1	Nature of JOURNEY ACCOUNTING.
«FK»	AccountedObjectRef	<i>ObjectRef</i>	0:1	Object for which this provides the accounting.
«FK»	OrganisationRef	<i>OperatorRef</i>	0:1	Reference to ORGANISATION providing contract or subsidy.
«FK»	ContractRef	<i>SupplyContractRef</i>	0:1	Supply Contract under which JOURNEY ACCOUNTING is provided.

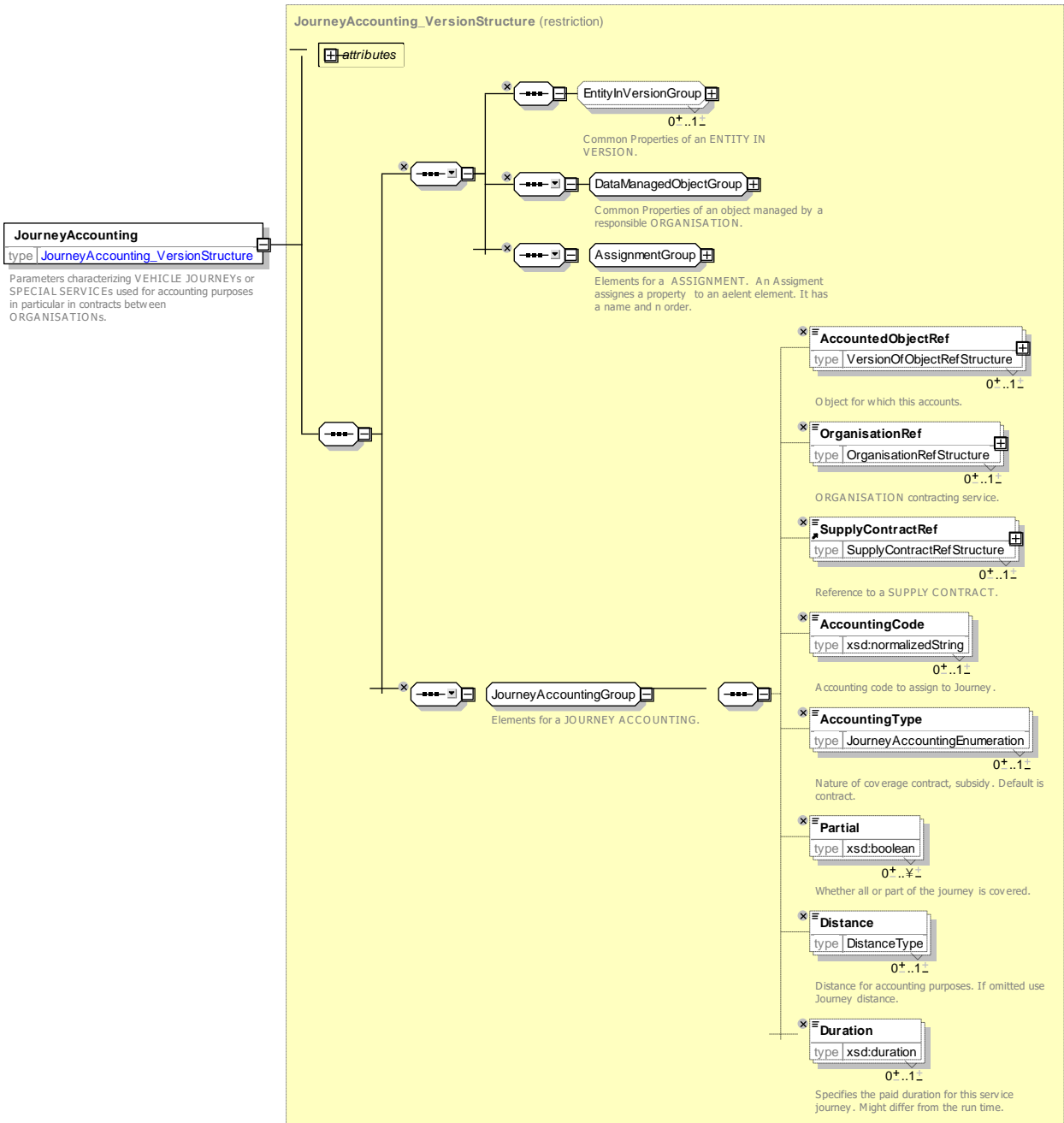


Figure 122 — *JourneyAccounting* — XSD

7.2.11.3.1.1 *JourneyAccountingType* – Allowed values

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

Table 66 – *JourneyAccountingType* – Allowed values

Value	Description
<i>contract</i>	Journey is undertaken under contract.
<i>subsidy</i>	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.

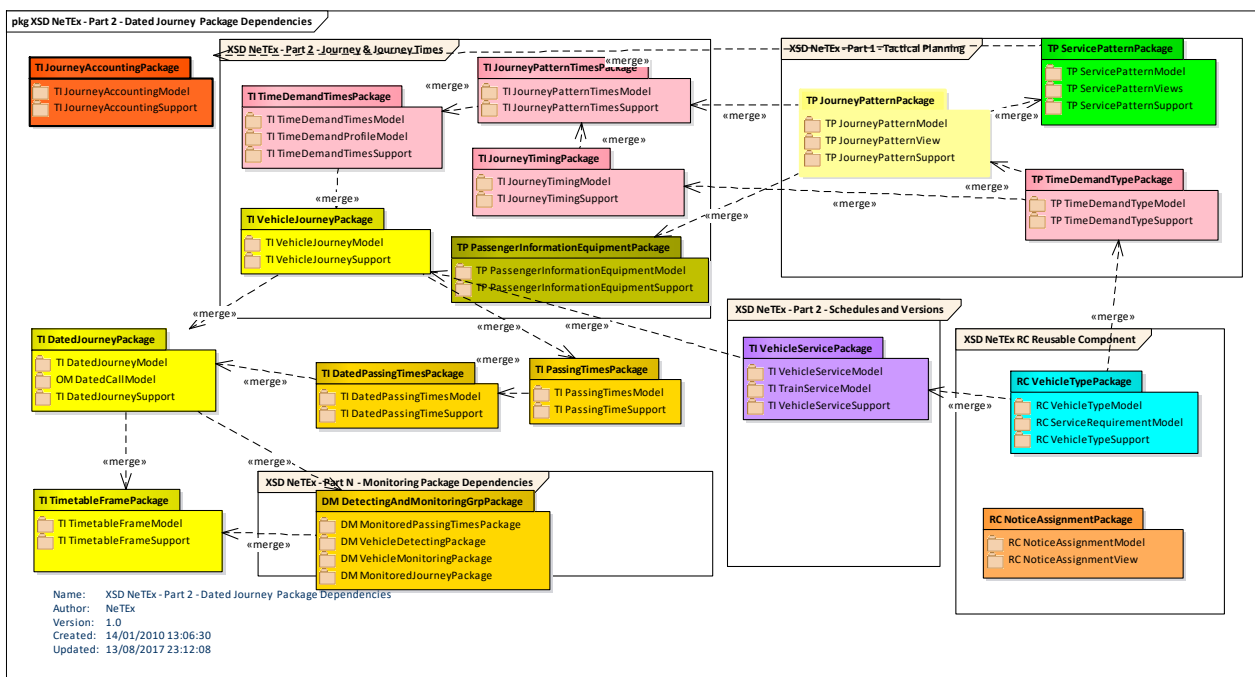


Figure 123 — Dated Journey Package Dependencies

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.

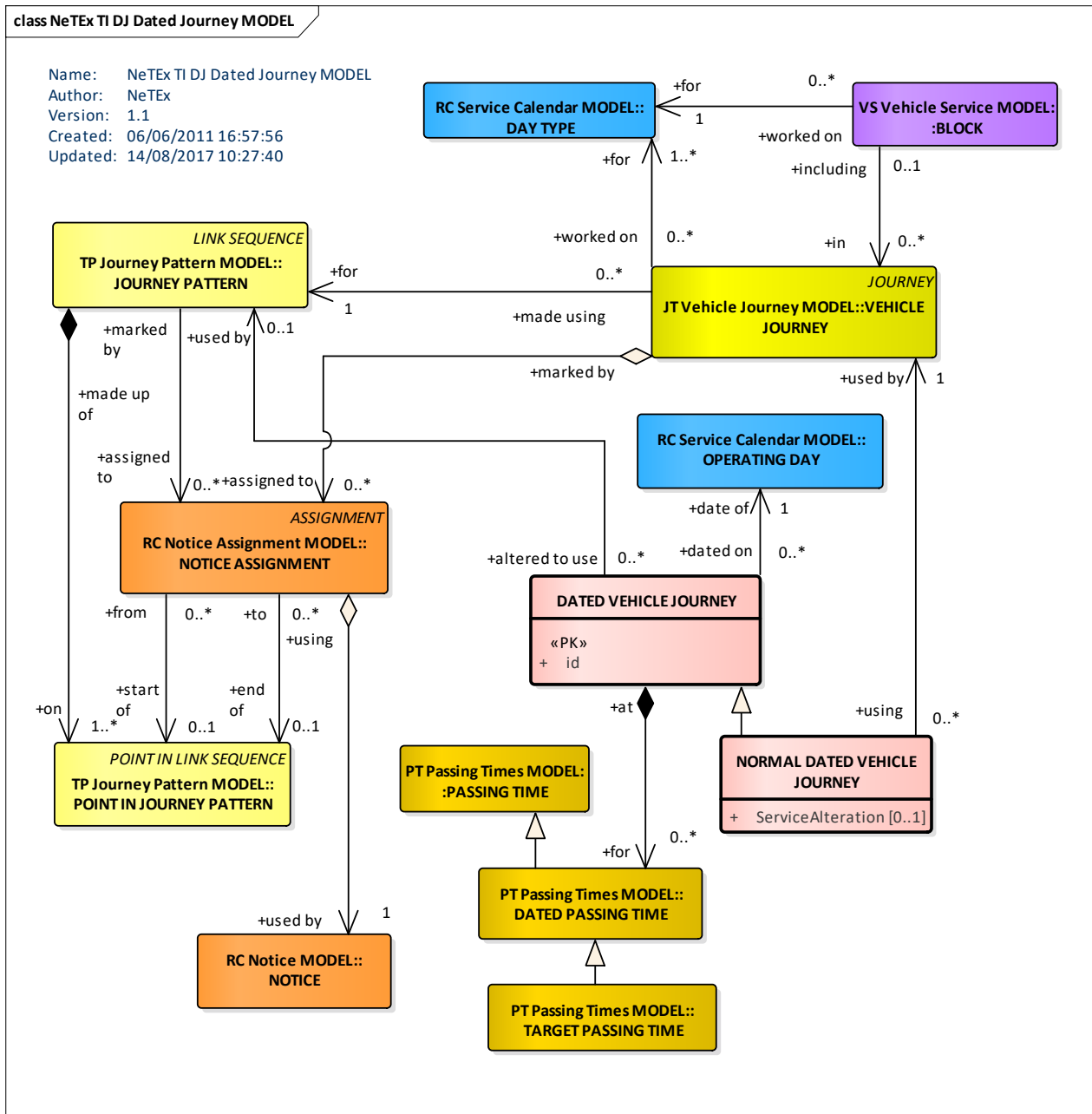


Figure 124 — Dated Journey — Conceptual MODEL (UML)

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.

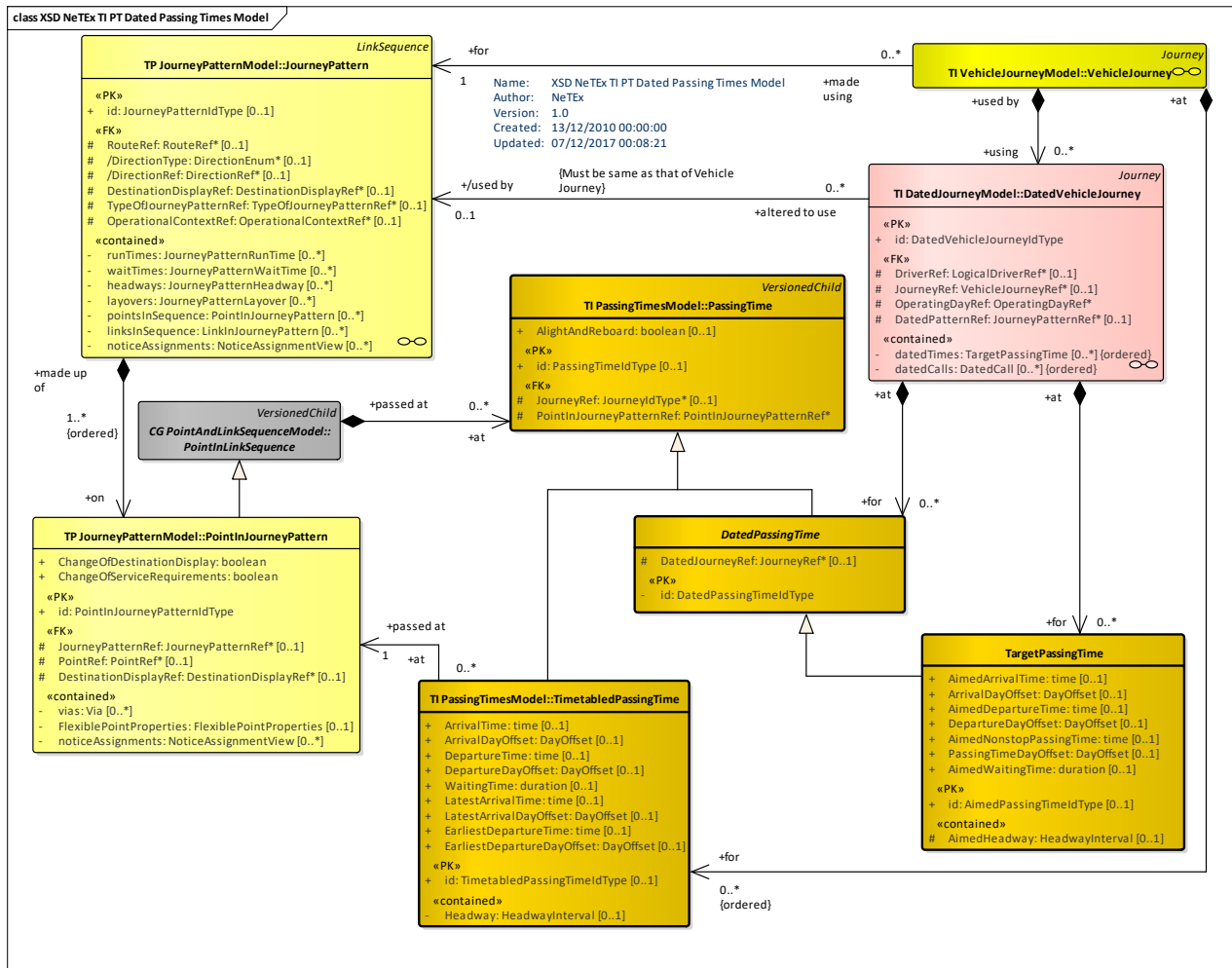


Figure 125 — Dated Journey – Physical Model (UML)

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 67 – DatedVehicleJourney – Element

Classification	Name	Type	Cardinality	Description
::>	::>	VehicleJourney	::>	DATED VEHICLE JOURNEY inherits from VEHICLE JOURNEY.

«PK»	id	<i>DatedVehicleJourney-IdType</i>	1:1	Identifier of DATED VEHICLE JOURNEY.
«FK»	JourneyRef	<i>VehicleJourneyRef</i>	0:1	VEHICLE JOURNEY for which this is a DATED VEHICLE JOURNEY.
«FK»	OperatingDayRef	<i>OperatingDayRef</i>	1:1	OPERATING DAY for DATED VEHICLE JOURNEY.
«AK»	ExternalDate-dJourneyRef	<i>ExternalObjectRef</i>	0:1	An alternative code that uniquely identifies the DATED VEHICLE JOURNEY, specifically for use in AVMS systems. <i>NOTE For VDV compatibility.</i>
«FK»	DatedJourney-PatternRef	<i>JourneyPatternRef</i>	0:1	DATED JOURNEY PATTERN of DATED VEHICLE JOURNEY.
«FK»	DriverRef	<i>DriverRef</i>	0:*	DRIVER of DATED VEHICLE JOURNEY.
«cntd»	datedPassing-Times	<i>TargetPassingTime</i>	0:*	Passing times of DATED VEHICLE JOURNEY.
«cntd»	datedCalls	<i>DatedCall</i>	0:*	CALLs of DATED VEHICLE JOURNEY.

7.2.12.4.1.1 DatedVehicleJourneyReferencesGroup — Group

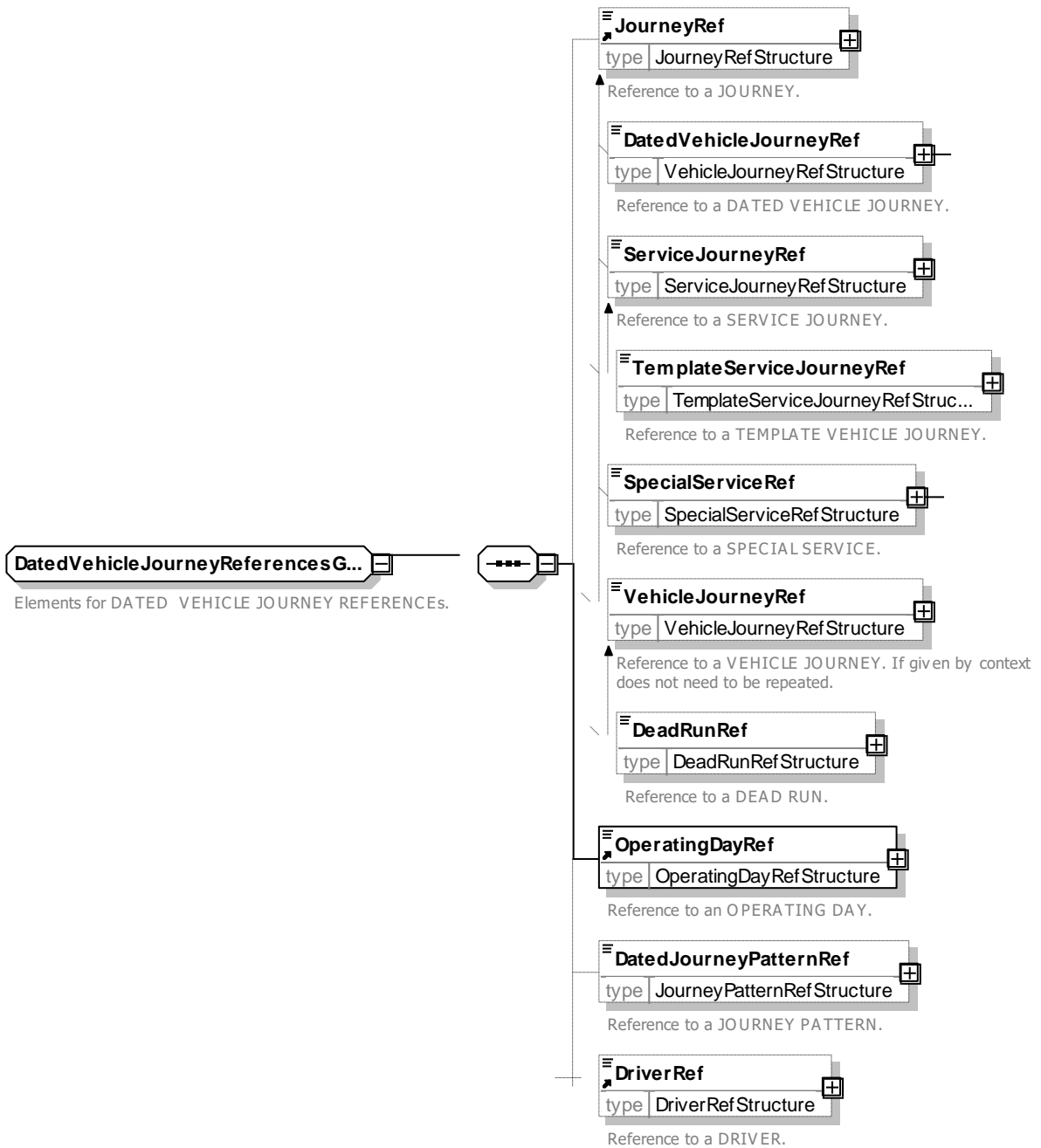


Figure 126 — *DatedVehicleJourneyReferencesGroup* — XSD

7.2.12.4.2 NormalDatedVehicleJourney – Model Element

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

Table 68 – *NormalDatedVehicleJourney* – Element

Classification	Name	Type	Cardinality	Description
	<i>ServiceAlteration</i>	<i>ServiceAlterationEnum</i>	1:1	Service alteration type for NORMAL JOURNEY. See SERVICE JOURNEY.

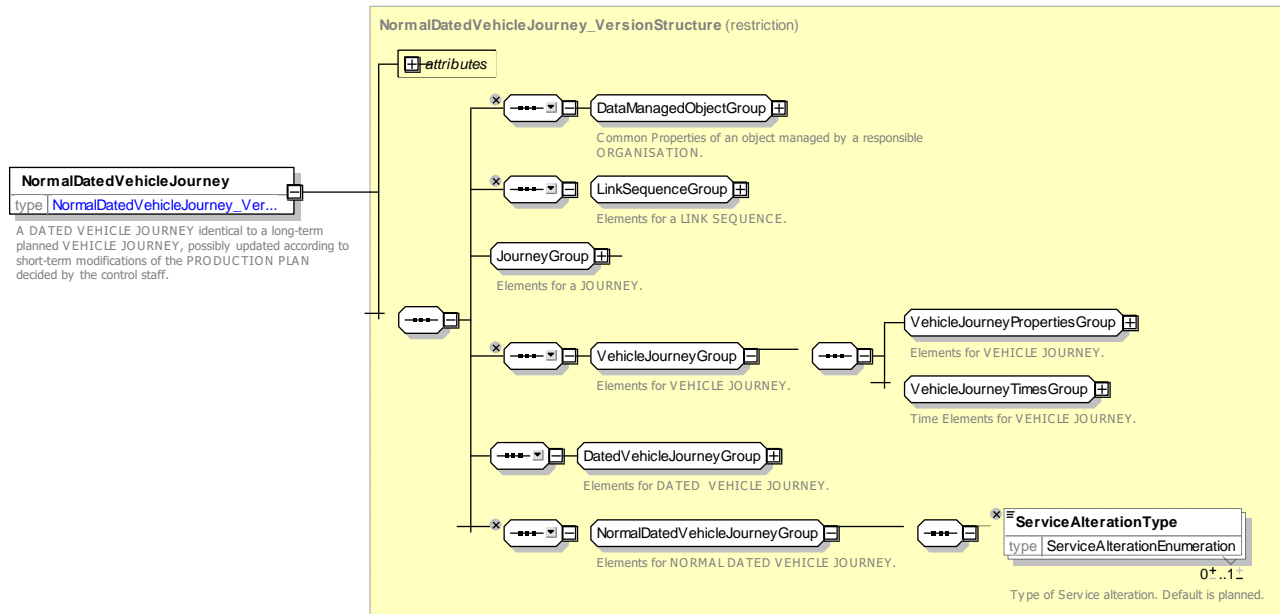


Figure 127 — NormalDatedVehicleJourney — XSD

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.

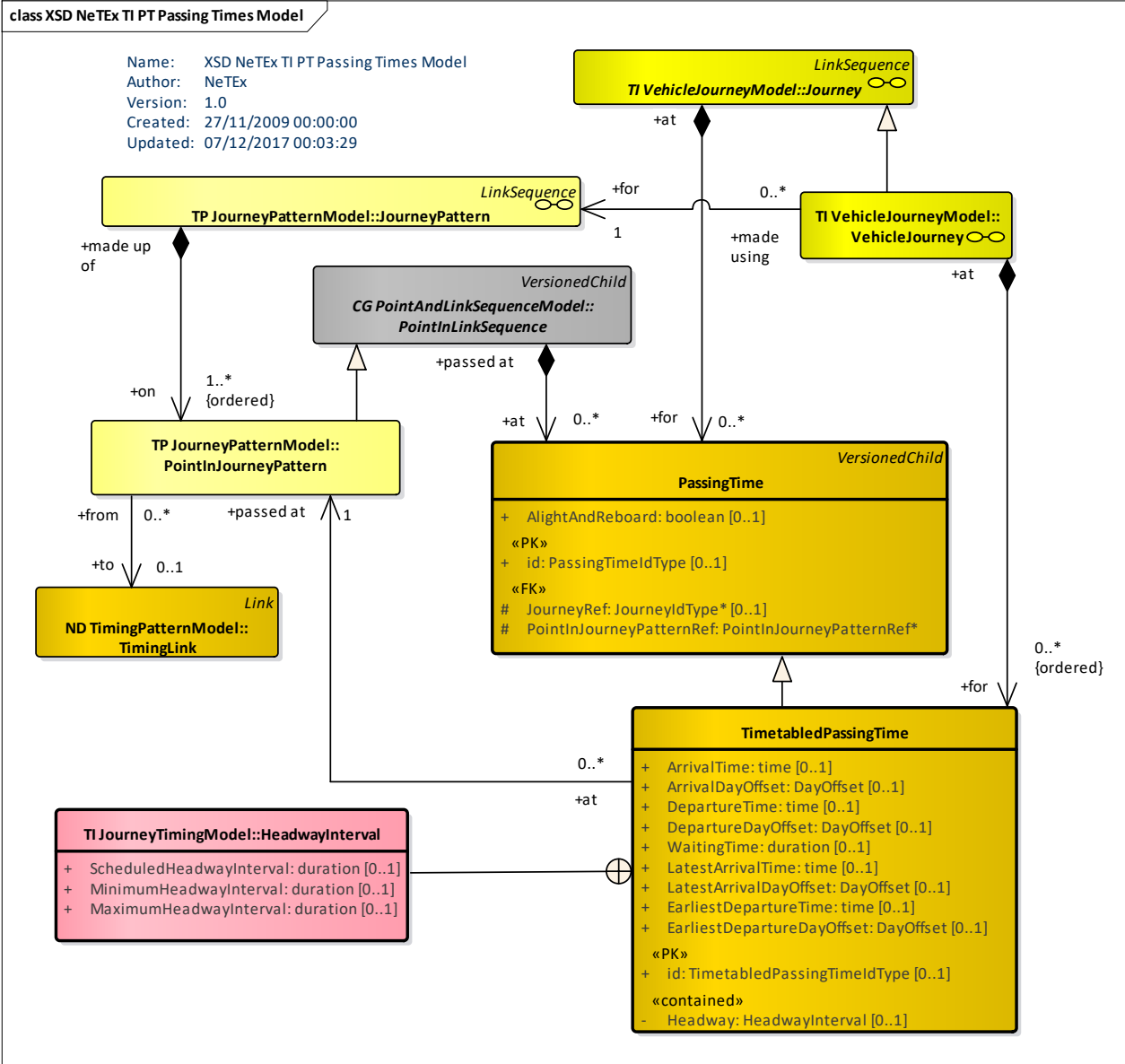


Figure 128 — Timetabled Passing Times – Physical Model (UML)

7.2.13.4.2 Dated Passing Times – Physical Model

The DATED PASSING TIMES physical model is shown below.

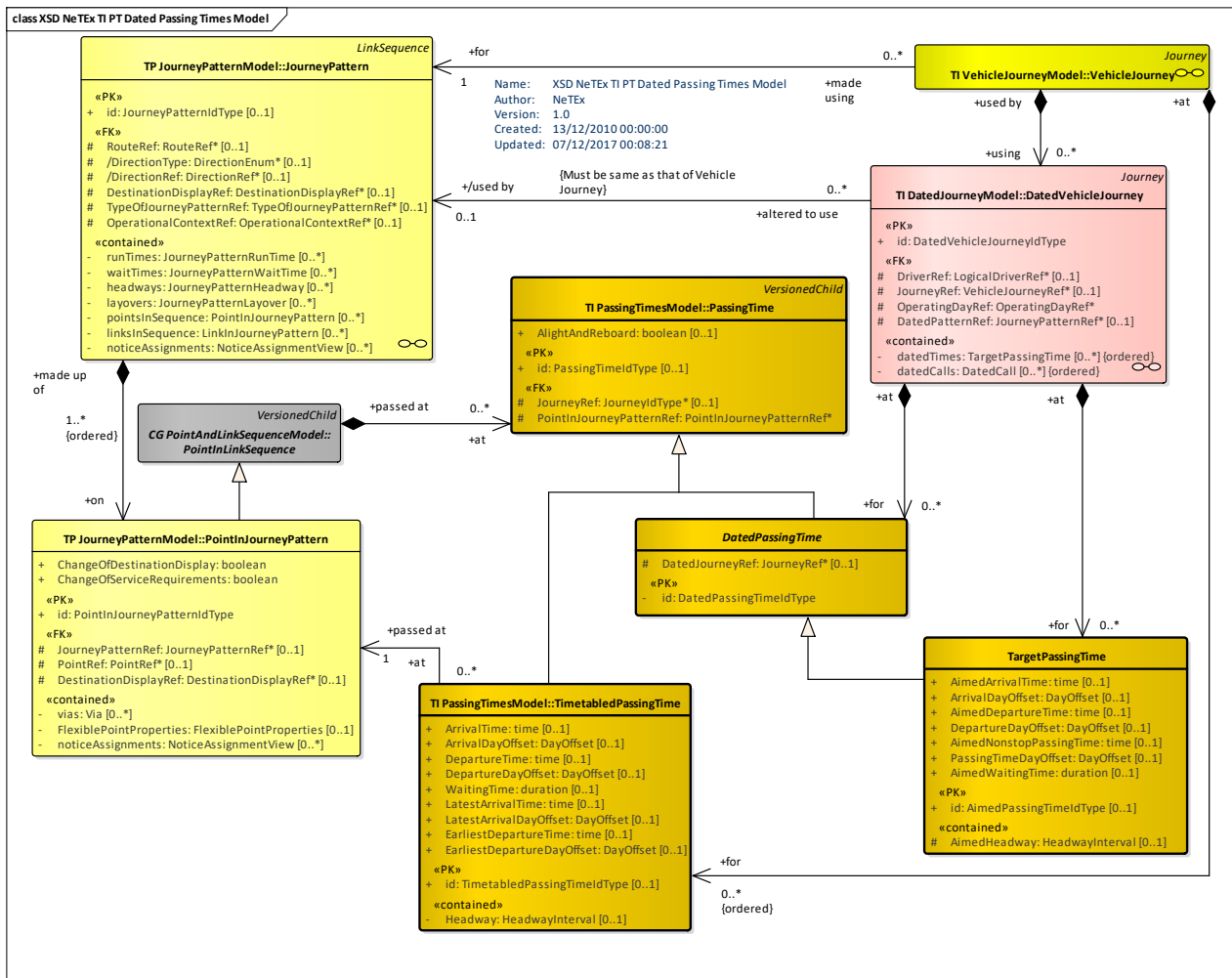


Figure 129 — Dated Passing Times – Physical Model (UML)

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 69 – PassingTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	VersionedChild	::>	PASSING TIME inherits from VERSIONED CHILD.
«PK»	id	TimetabledPassing-TimeIdType	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	Whether can alight and reboard at stop.
«FK»	PointInJourney-PatternRef	PointInLinkSequenceRef	0:1	POINT IN SEQUENCE for which this is a PASSING TIME.

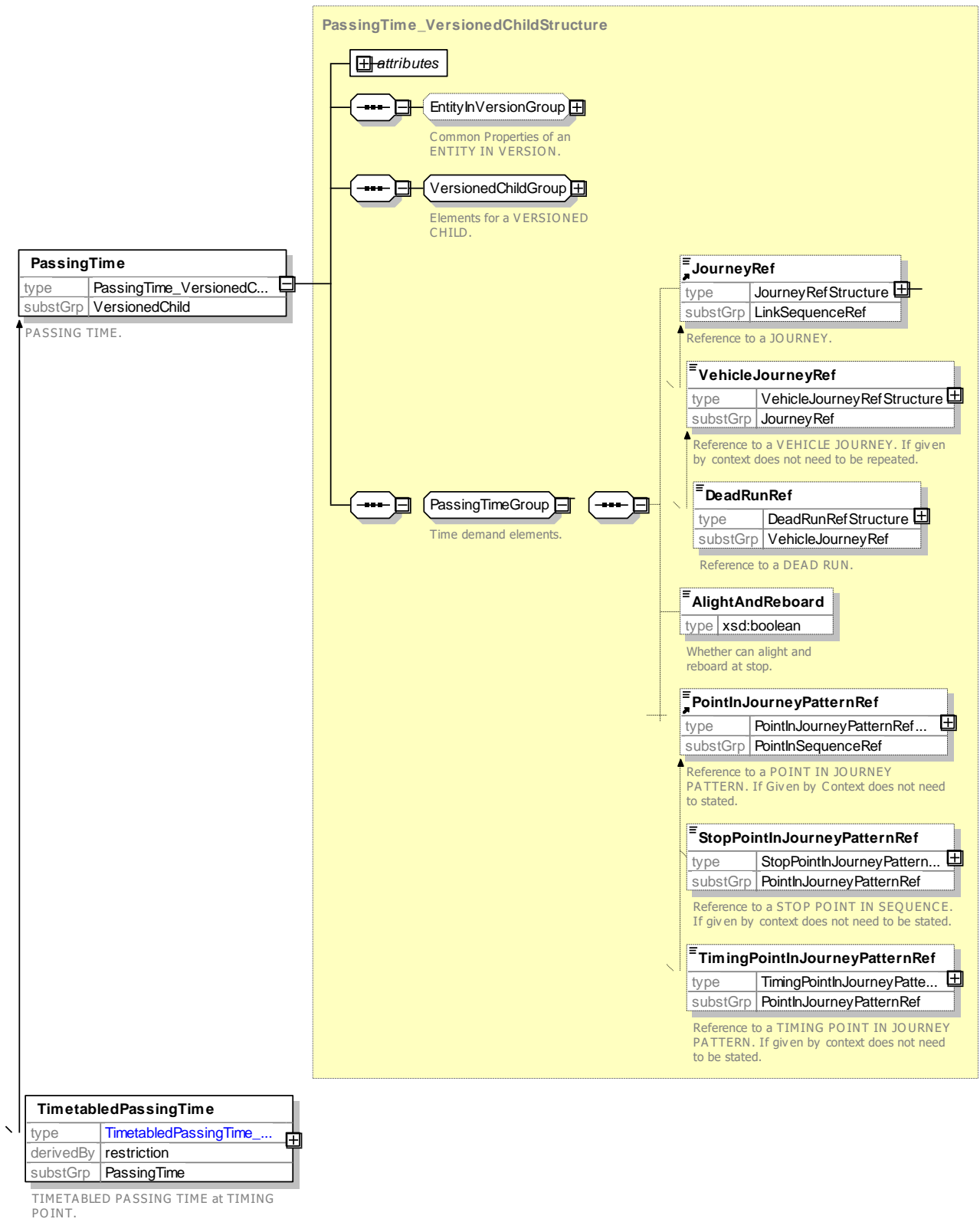


Figure 130 — *PassingTime* — XSD

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 70 – TimetabledPassingTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>PassingTime</i>	::>	TIMETABLED PASSING TIME inherits from PASSING TIME.
«PK»	id	<i>PassingTimeIdType</i>	0:1	Identifier of PASSING TIME.
	ArrivalTime	<i>xsd:time</i>	0:1	Arrival time at POINT IN PATTERN.
	ArrivalDayOffset	<i>DayOffsetType</i>	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.
	DepartureTime	<i>xsd:time</i>	0:1	Departure time at POINT IN PATTERN.
	DepartureDayOffset	<i>DayOffsetType</i>	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.
	WaitingTime	<i>xsd:duration</i>	0:1	Waiting time at POINT IN PATTERN.
	Headway	<i>HeadwayInterval</i>	0:1	Headway interval at POINT IN PATTERN.
	LatestArrivalTime	<i>xsd:time</i>	0:1	Latest Arrival time at POINT IN PATTERN.
	LatestArrivalDayOffset	<i>DayOffsetType</i>	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.
	EarliestDepartureTime	<i>xsd:time</i>	0:1	Earliest Departure time at POINT IN PATTERN.
	EarliestDeparture-DayOffset	<i>DayOffsetType</i>	0:1	Number of days after the starting time of the journey if not same calendar day. Default is 0 for same day.

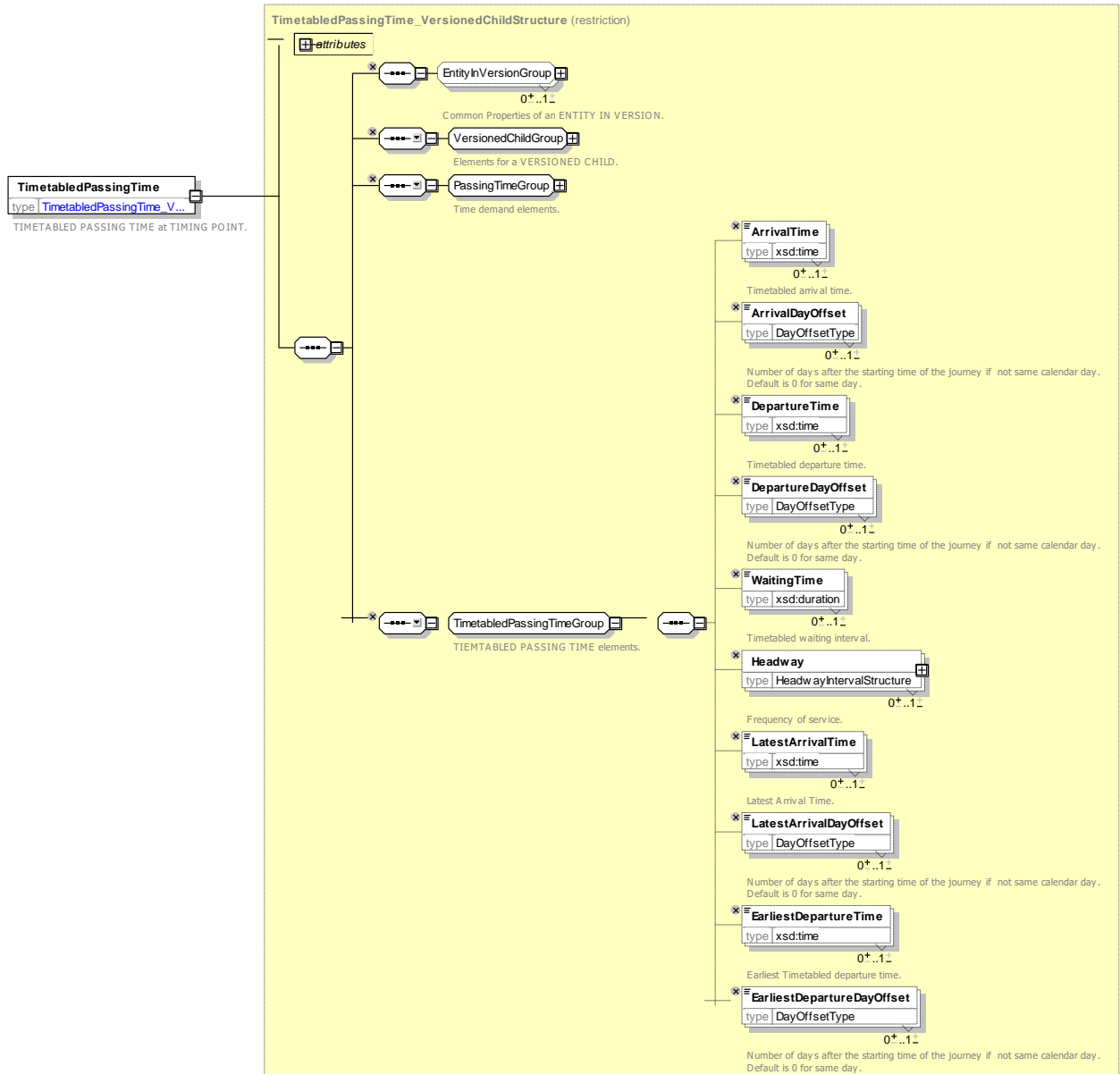


Figure 131 — *TimetabledPassingTime* — XSD

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 71 – *DatedPassingTime* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>PassingTime</i>	::>	DATED PASSING TIME inherits from PASSING TIME.
«PK»	<i>id</i>	<i>DatedPassingTimeIdType</i>	1:1	Identifier of DATED PASSING TIME.

	DatedJourneyRef	<i>JourneyRef</i>	0:1	DATED VEHICLE JOURNEY for which this is a PASSING TIME.
--	------------------------	-------------------	-----	---

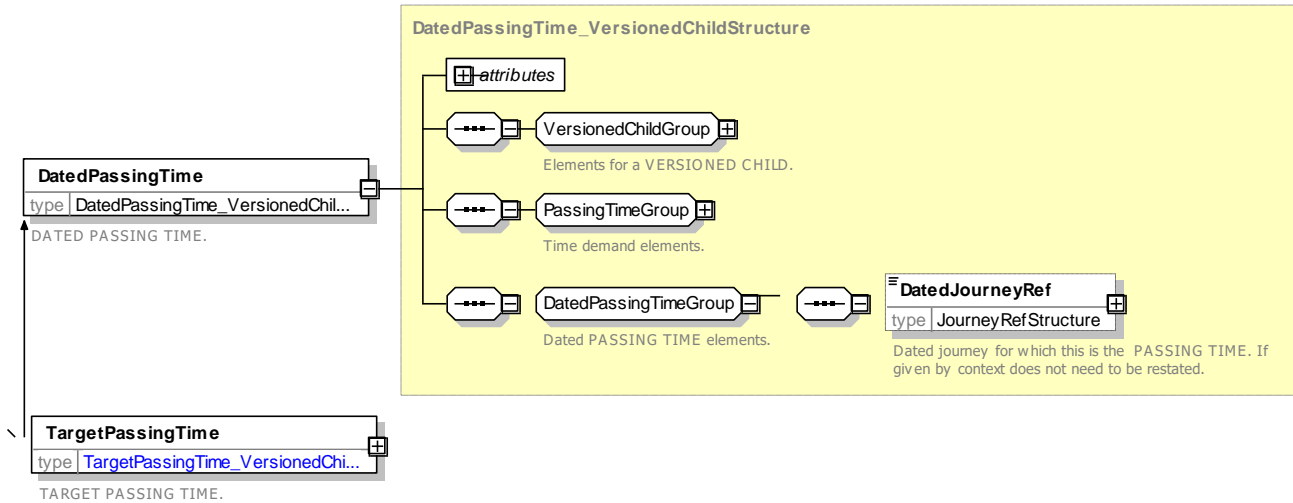


Figure 132 — *DatedPassingTime* — XSD

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.

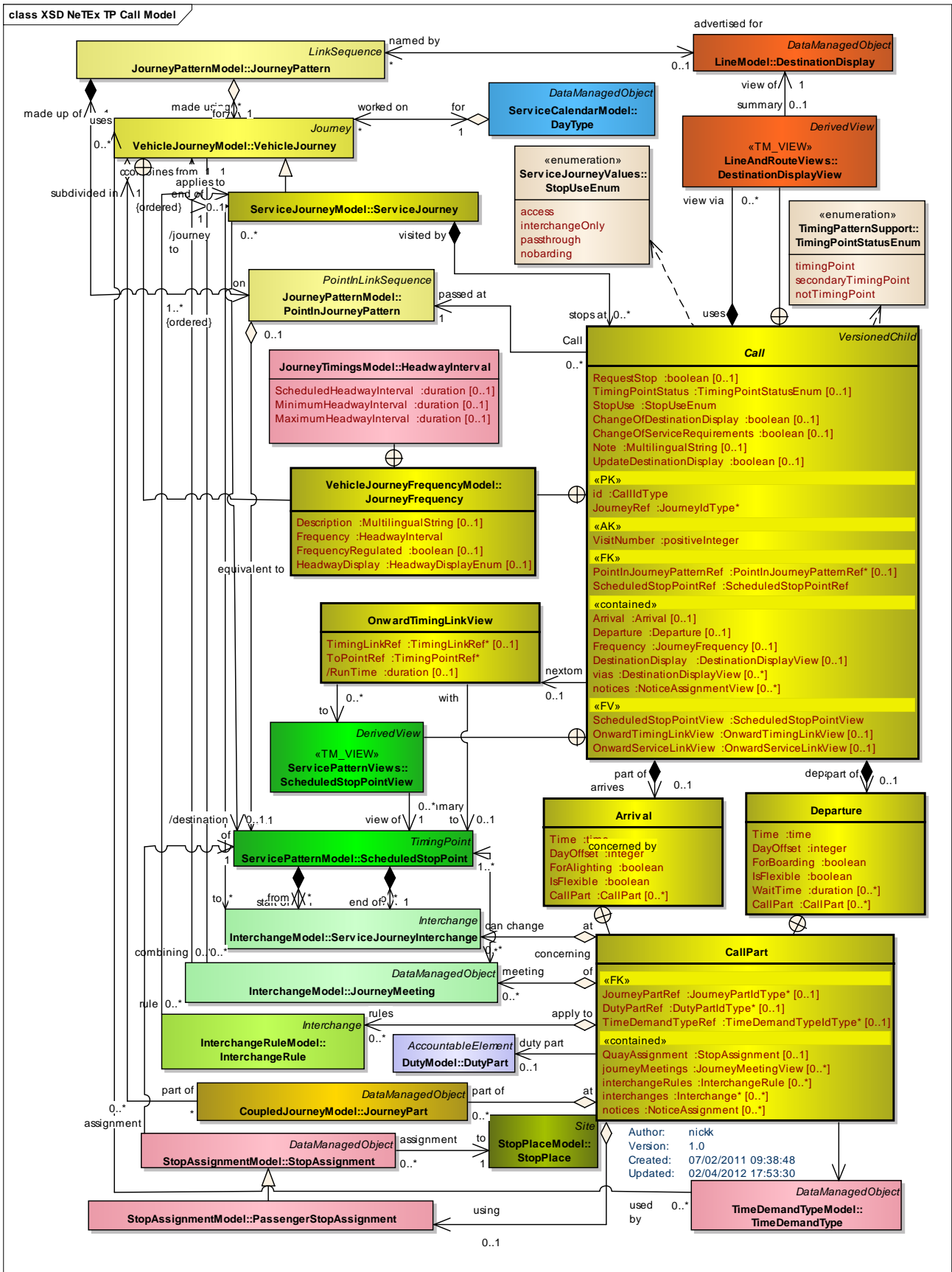


Figure 133 —Call – Physical Model

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 72 – Call – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>VersionedChild</i>	::>	CALL inherits from VERSIONED CHILD
«PK»	id	<i>CallIdType</i>	1:1	Identifier of CALL.
«AK»	VisitNumber	<i>xsd:positiveInteger</i>	0:1	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.
«FK»	a ScheduledStopPointRef	<i>ScheduledStopPointRef</i>	0:1	Reference to a SCHEDULED STOP POINT.
«FV»	b ScheduledStopPointView	<i>ScheduledStopPointView</i>	0:1	Reference to the SCHEDULED STOP POINT visited by CALL. May include derived data.
«FV»	OnwardTimingLinkView	<i>OnwardTimingLinkView</i>	0:1	Reference to the next TIMING LINK followed after this CALL. May include derived data.
	a OnwardServiceLinkRef	<i>ServiceLinkRef</i>	0:1	reference to onwads SERVICE LINK.
«FV»	b OnwardServiceLinkView	<i>OnwardServiceLinkView</i>	0:1	Reference to the next SERVICE LINK followed after this CALL. May include derived data.
	TimingPointStatus	<i>TimingPointStatusEnum</i>	0:1	Timing Status of CALL.
«PK»	ServiceJourneyRef	<i>ServiceJourneyRef</i>	0:1	Reference to SERVICE JOURNEY containing CALL.
«FK»	PointInJourneyPatternRef	<i>PointInJourneyPatternRef</i>	0:1	POINT IN JOURNEY PATTERN corresponding to CALL.
«cntd»	Arrival	<i>Arrival</i>	0:1	Arrival part of CALL.
«cntd»	Departure	<i>Departure</i>	0:1	Departure part of CALL.
«cntd»	Frequency	<i>JourneyFrequency</i>	0:1	Frequency information for CALL.
«cntd»	DestinationDisplay	<i>DestinationDisplayView</i>	0:1	DESTINATION DISPLAY associated with the CALL. To be used from this point onwards.
«cntd»	vias	<i>DestinationDisplayView</i>	0:*	Set of DESTINATION DISPLAY to use to show as onward VIA points associated with the CALL.
	ChangeOfDestinationDisplay	<i>xsd:boolean</i>	0:1	Whether DESTINATION DISPLAY changes at this point.
	ChangeOfServiceRequirements	<i>xsd:boolean</i>	0:1	Whether DESTINATION DISPLAY changes at this point.
«cntd»	noticeAssignments	<i>NoticeAssignmentView</i>	0:*	NOTICE ASSIGNMENTS, footnotes etc. associated with the CALL. May include derived data.
	RequestStop	<i>xsd:boolean</i>	0:1	Whether the stop is a Request Stop.
	RequestMethod	<i>RequestMethodTypeEnumeration</i>	0:1	Method of requesting stop. Default is noneRequired.
	StopUse	<i>StopUseEnum</i>	0:1	Activity at stop.

«cntd»	BookingArrangements	<i>BookingArrangements</i>	0:1	Booking Arrangements for stop if different from thos for SERVICE JOURNEY.
«FK»	a	PassengerCarryingRequirementRef	0:1	Reference to a PASSENGER CARRYING REQUIREMENT.
	b	PassengerCarryingRequirementsView	0:1	Requirements for carrying passengers.
	TrainSize	<i>TrainSize</i>	0:1	Requirements for TRAIN SIZE.
«cntd»	equipments	<i>vehicleEquipment</i>	0:*	VEHICLE EQUIPMENT available on service.
	Note	<i>MultilingualString</i>	0:1	Arbitrary note associated with the CALL. For internal use. Footnotes etc., should be specified with NOTICES.

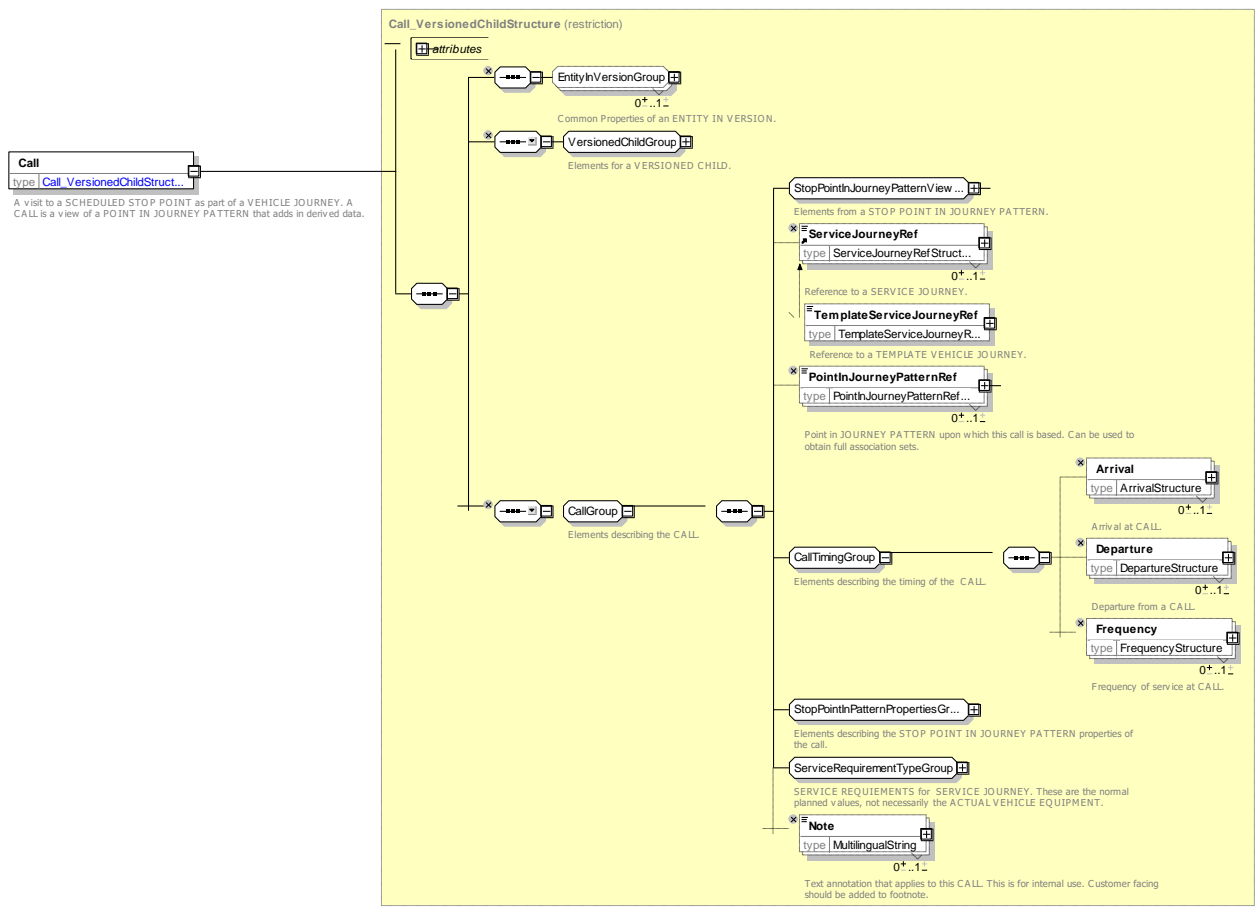


Figure 134 — Call — XSD

7.2.14.2.1.1 StopPointInJourneyPatternViewPropertiesGroup – Group



Figure 135 — StopPointInJourneyPatternViewPropertiesGroup — XSD

7.2.14.2.1.2 TimingPointStatus – Allowed values

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

Table 73 – TimingPointStatus – Allowed values

Value	Description
<i>timingPoint</i>	Timing Point
<i>secondaryTimingPoint</i>	Secondary Timing Point
<i>notTimingPoint</i>	Not Timing Point

7.2.14.2.1.3 StopPointInPatternPropertiesGroup – Group

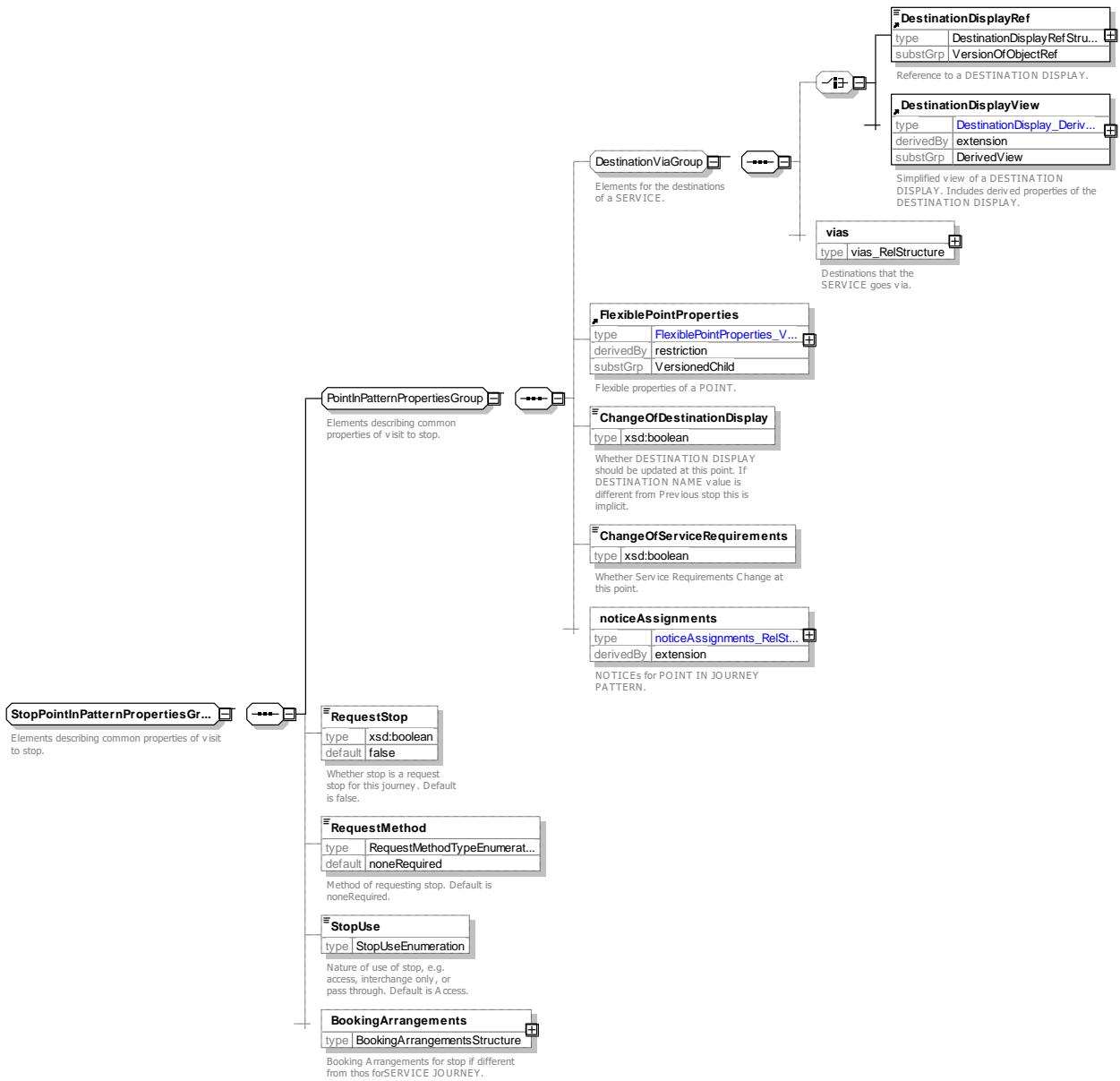


Figure 136 — StopPointInPatternPropertiesGroup — XSD

7.2.14.2.1.4 StopUse – Allowed values

The following table shows the allowed values for StopUde (StopUseEnum).

Table 74 – TimingPointStatus – Allowed values

Value	Description
access	Alighting and boarding allowed.
interchangeOnly	Only interchanging allowed.

passthrough	Does not stophalt at stop.
noBoardingOrAlighting	Halts at stop but no access.

7.2.14.3 Arrival – Model Element

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

Table 75 – Arrival – Element

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

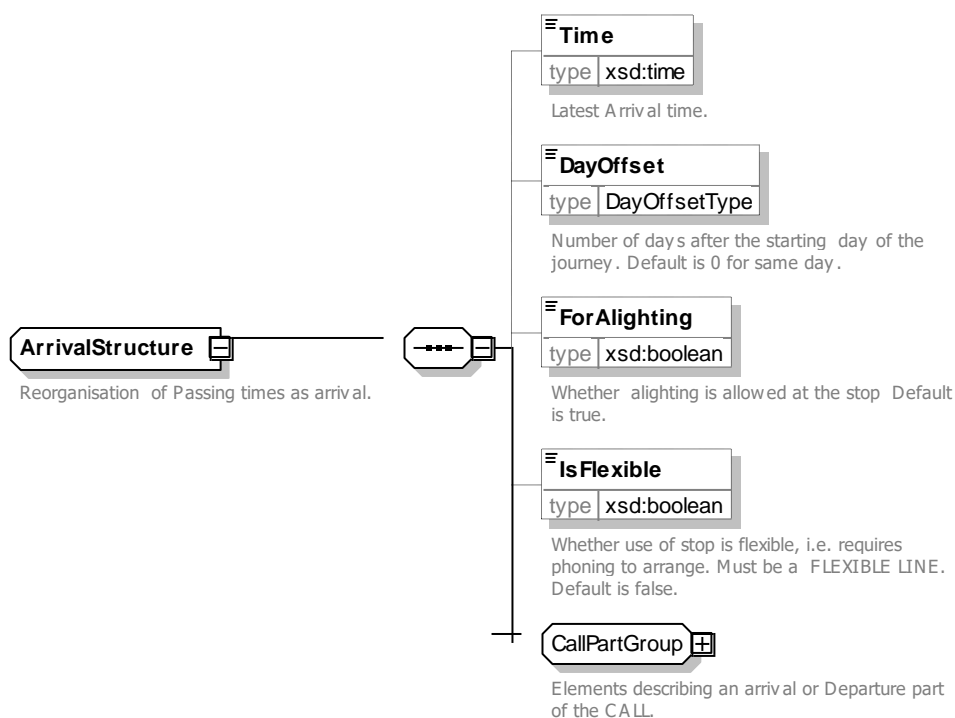


Figure 137 — ArrivalStructure — XSD

7.2.14.4 Departure – Model Element

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

Table 76 – Departure – Element

Classification	Name	Type	Cardinality	Description
----------------	------	------	-------------	-------------

	Time	<i>xsd:time</i>	1:1	Earliest time of departure.
	DayOffset	<i>DayOffsetType</i>	1:1	Day offset from SERVICE JOURNEY start day. 0 = Same day.
	ForBoarding	<i>xsd:boolean</i>	1:1	Whether boarding is allowed for CALL.
	IsFlexible	<i>xsd:boolean</i>	1:1	Whether use of stop is flexible.
	WaitTime	<i>xsd:duration</i>	0:*	Time to wait at stop after arrival before departure.
	CallPart	<i>CallPart</i>	0:*	Common elements, e.g. NOTICEs, QUAY ASSIGNMENT, INTERCHANGES and JOURNEY PART associated with departure part of CALL.

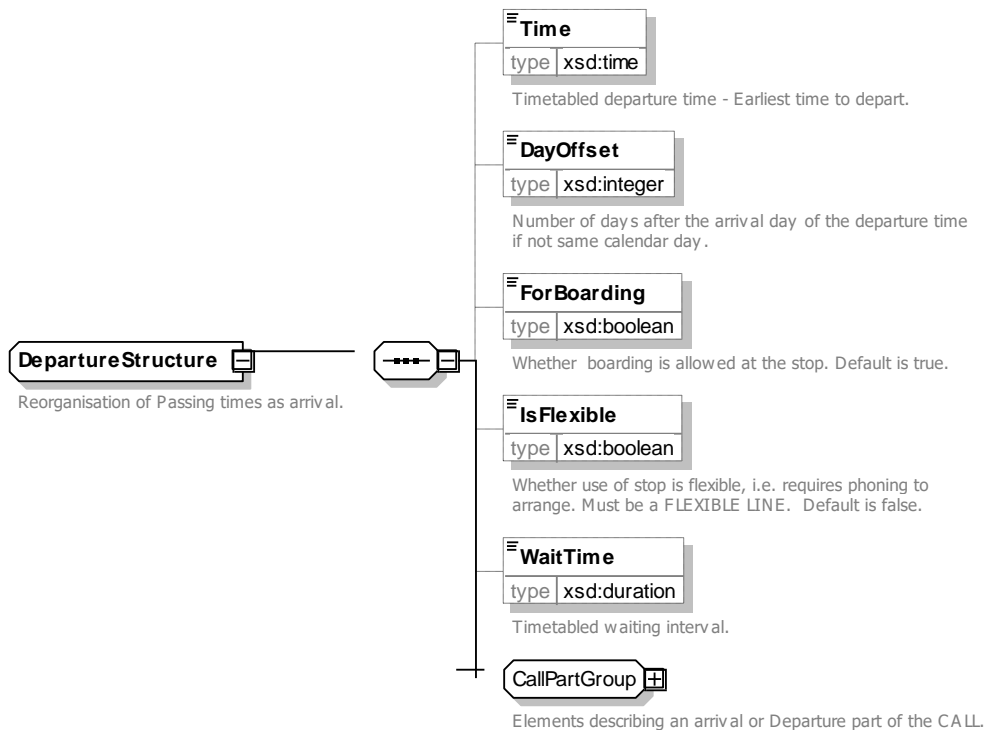


Figure 138 — *DepartureStructure* — XSD

7.2.14.5 OnwardServiceLinkView – Model Element

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

Table 77 – *OnwardServiceLinkView* – Element

Classification	Name	Type	Cardinality	Description
	ServiceLinkRef	<i>ServiceLinkRef</i>	0:1	Reference to onwards SERVICE LINK.
	ToPointRef	<i>ScheduledStopPointRef</i>	1:1	Reference to next SCHEDULED STOP POINT in pattern.

	Distance	<i>DistanceType</i>	0:1	DISTANCE for onward SERVICE LINK.
	RunTime	<i>xsd:duration</i>	0:1	RUN TIME for onward SERVICE LINK.

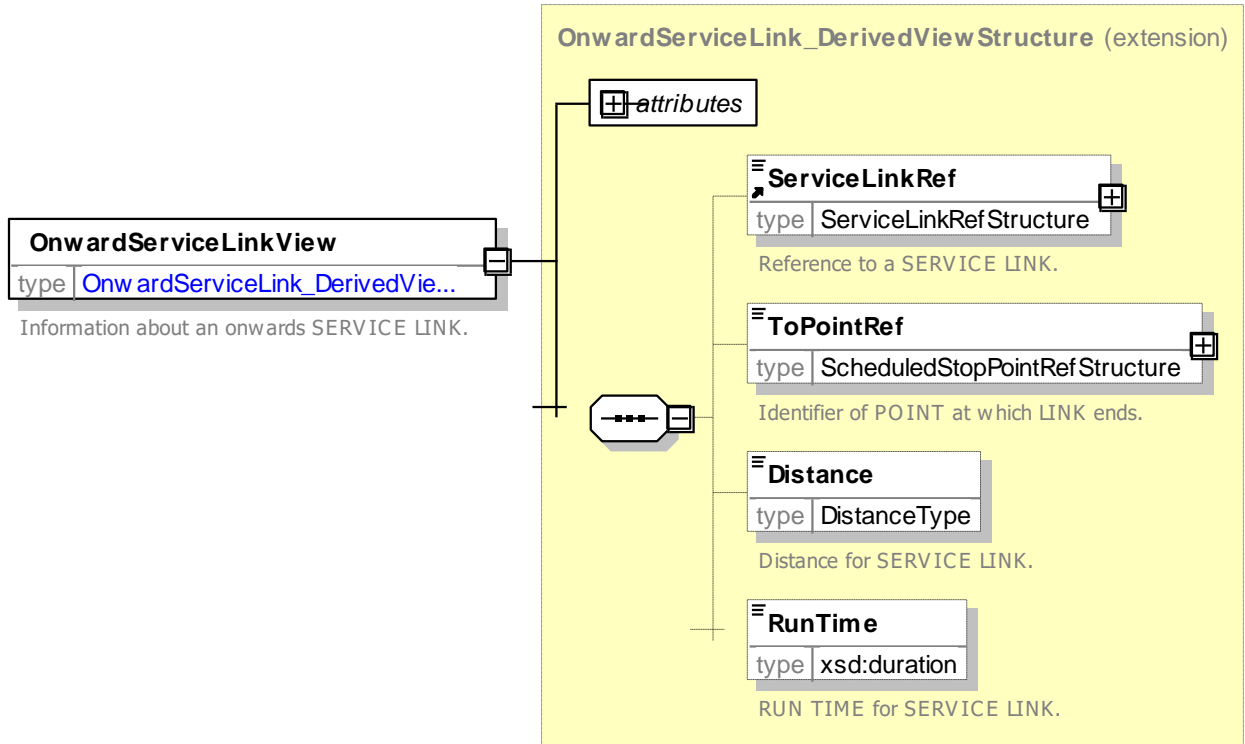


Figure 139 — *OnwardServiceLinkView* — XSD

7.2.14.6 OnwardTimingLinkView – Model Element

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

Table 78 – *OnwardTimingLinkView* – Element

Classification	Name	Type	Cardinality	Description
	TimingLinkRef	<i>TimingLinkRef</i>	0:1	Reference to onwards TIMING LINK.
	ToPointRef	<i>TimingPointRef</i>	1:1	Reference to next SCHEDULED STOP POINT in pattern.
	RunTime	<i>xsd:duration</i>	0:1	RUN TIME for onward TIMING LINK.

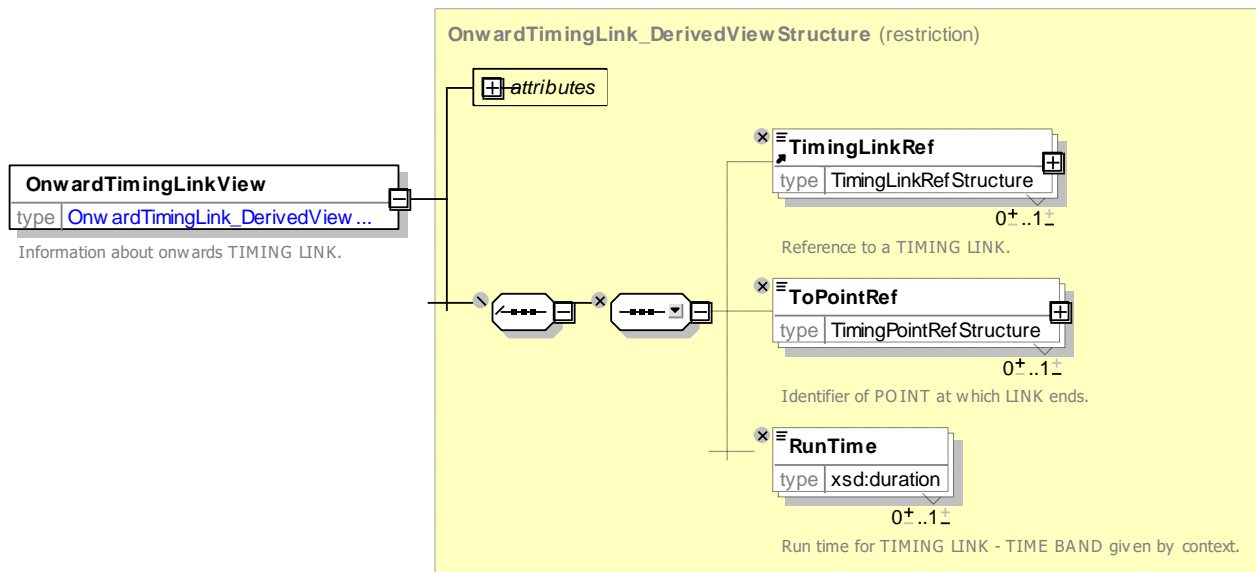


Figure 140 — OnwardTimingLinkView — XSD

7.2.14.7 CallPartGroup – Model Element

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

Table 79 – CallPartGroup – Group

Classification	Name	Type	Cardinality	Description
«FK»	JourneyPartRef	<i>JourneyPartRef</i>	0:1	JOURNEY PART to which CALL belongs.
«cntd»	journeyMeetings	<i>JourneyMeetingView</i>	0:*	JOURNEY MEETINGS of this journey at this specific CALL PART.
«cntd»	interchanges	<i>Interchange</i>	0:*	INTERCHANGE of this journey at this specific CALL PART.
«cntd»	interchangeRules	<i>InterchangeRule</i>	0:*	INTERCHANGE RULES of this journey at this specific CALL PART.
«FK»	a TimeDemandTypeRef	<i>TimeDemandTypeRef</i>	0:1	TIME DEMAND TYPE of this journey at this specific CALL.
«FK»	b TimebandRef	<i>TimebandRef</i>	0:1	Reference to a TIME BAND
«FK»	DutyPartRef	<i>DutyPartRef</i>	0:1	DUTY that applies from this CALL onwards.
«cntd»	QuayAssignment	<i>StopAssignment</i>	0:1	Assignment to a QUAY for this CALL PART.
«cntd»	AccessibilityAssessment	<i>AccessibilityAssessment</i>	0:1	The accessibility characteristics of an entity used by passengers such as a STOP PLACE, or a STOP PLACE COMPONENT. Described by ACCESSIBILITY LIMITATIONS, and/or a set of SUITABILITIES.
«cntd»	CheckConstraint	<i>CheckConstraint</i>	0:1	Characteristics of a SITE COMPONENT representing a process, such as check-in, security screening, ticket control or immigration, that may potentially incur a time penalty that should be allowed for when

				journey planning. Used to mark PATH LINKs to determine transit routes through interchanges.
«cntd»	noticeAssignments	NoticeAssignment	0:*	NOTICES that apply at this specific CALL PART.

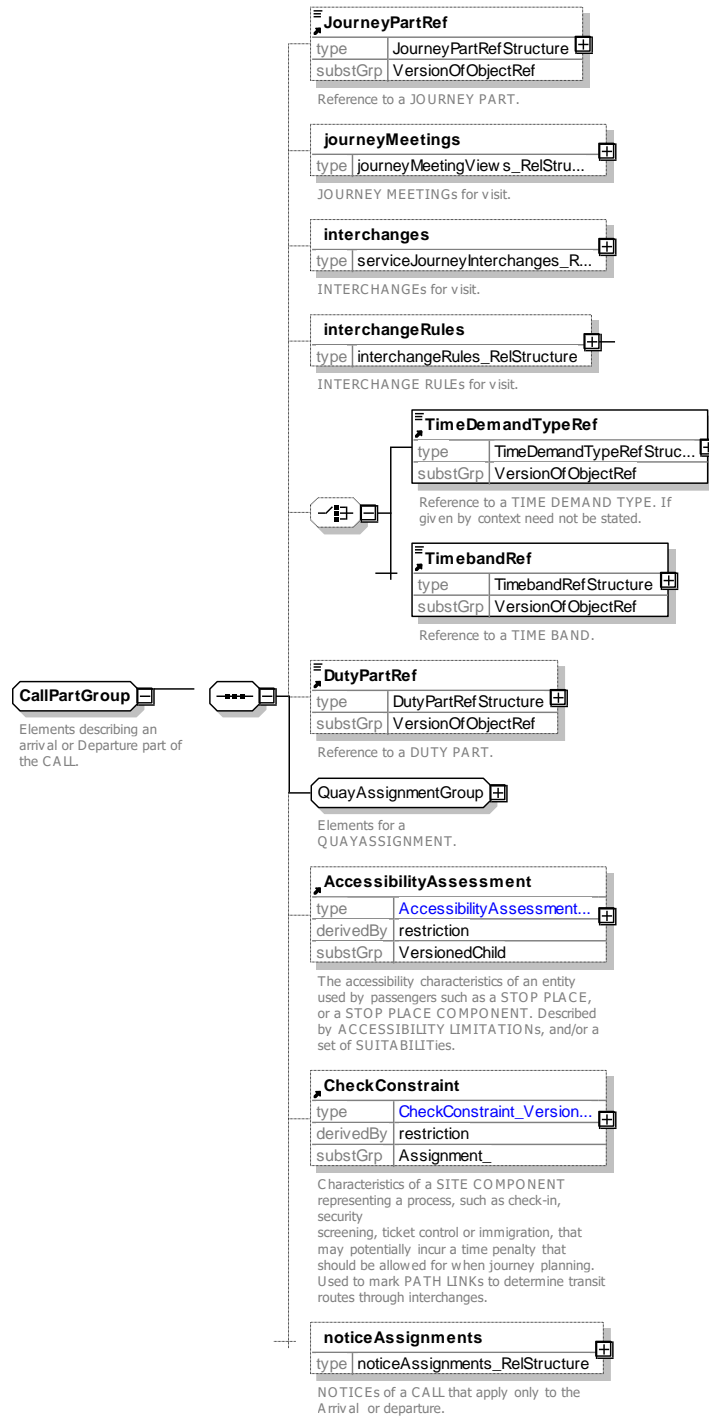


Figure 141 — CallPartGroup — XSD

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 80 – DeadRunCall – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>VersionedChild</i>	::>	DEAD RUN CALL inherits from VERSIONED CHILD.
«PK»	id	<i>DeadRunCallIdType</i>	1:1	Identifier of DEAD RUN.
	order	<i>xsd:integer</i>	0:1	Order of DEAD RUN CALL.
	Point	<i>choice</i>	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 ScheduledStop-PointView	<i>ScheduledStopPointView</i>	0:1	Reference to the SCHEDULED STOP POINT visited by CALL. May include derived data.
	TimingPointStatus	<i>TimingPointStatusEnum</i>	0:1	Nature of TIMING POINT.
	JourneyRef	<i>DeadRunRef</i>	1:1	Reference to JOURNEY containing CALL.
«FK»	PointInJourney-PatternRef	<i>PointInJourneyPatternRef</i>	0:1	Reference to POINT IN JOURNEY PATTERN for CALL.
«cntd»	Arrival	<i>DeadRunArrival</i>	0:1	Arrival part of DEAD RUN CALL.
«cntd»	Departure	<i>DeadRunDeparture</i>	0:1	Departure part of DEAD RUN CALL.
	Note	<i>MultilingualString</i>	0:1	Note on CALL.
«cntd»	DestinationDisplay	<i>DestinationDisplayView</i>	0:1	DESTINATION DISPLAY associated with the CALL.
«cntd»	vias	<i>DestinationDisplayView</i>	0:*	Set of via names to use to show as onward via points associated with the CALL.
«FV»	Onward-TimingLinkView	<i>OnwardTimingLinkView</i>	0:1	Onward TIMING LINK for next link from CALL stop.

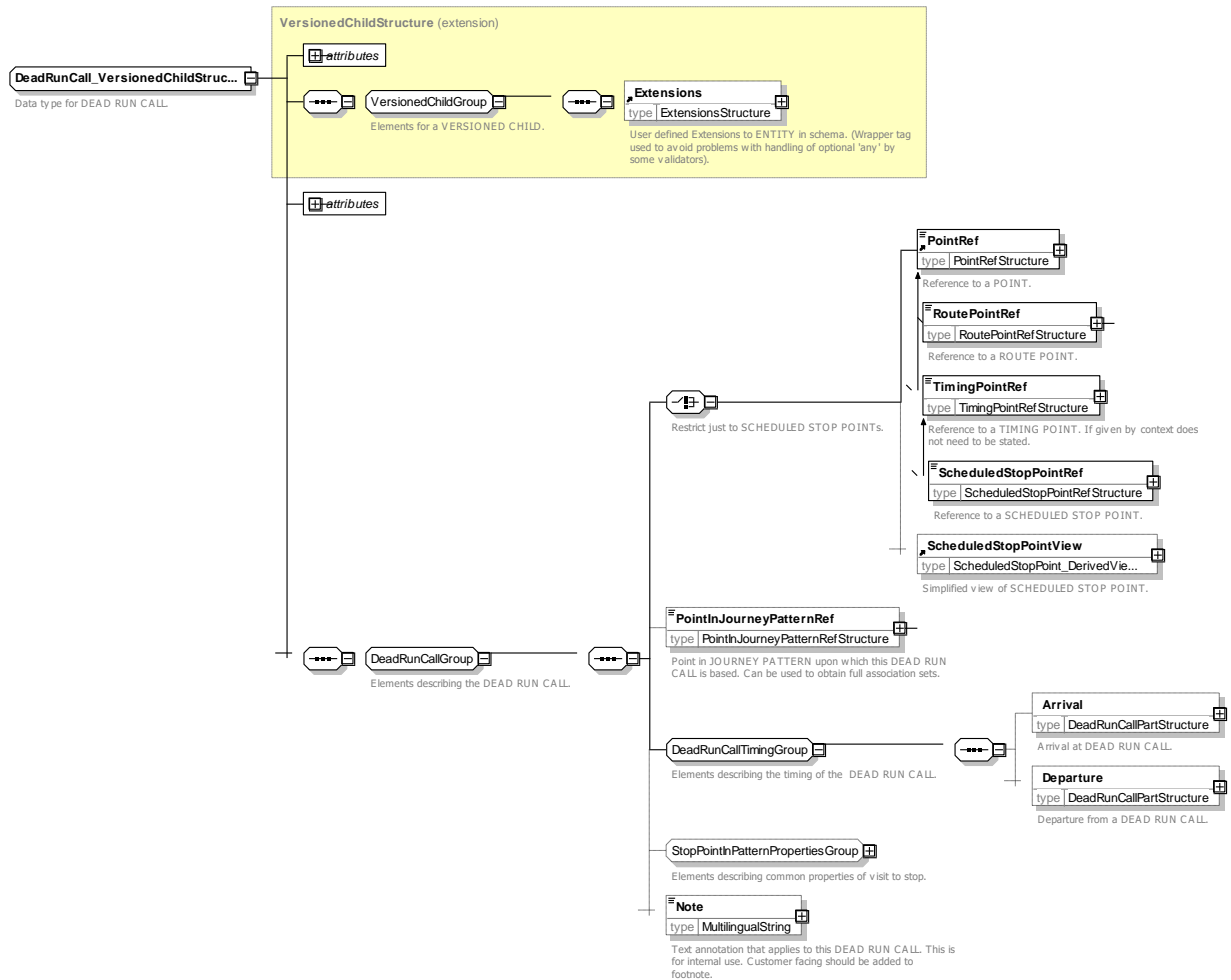


Figure 142 — *DeadRunCall* — XSD

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 81 – *DeadRunArrival* – Element

Classification	Name	Type	Cardinality	Description
	Time	<i>time</i>	1:1	Latest time of arrival.
	DayOffset	<i>integer</i>	1:1	Day offset from DEAD RUN start day. 0 = Same day.
	CallPart	<i>DeadRunCallPartGroup</i>	0:*	Common properties of a DEAD RUN CALL arrival.

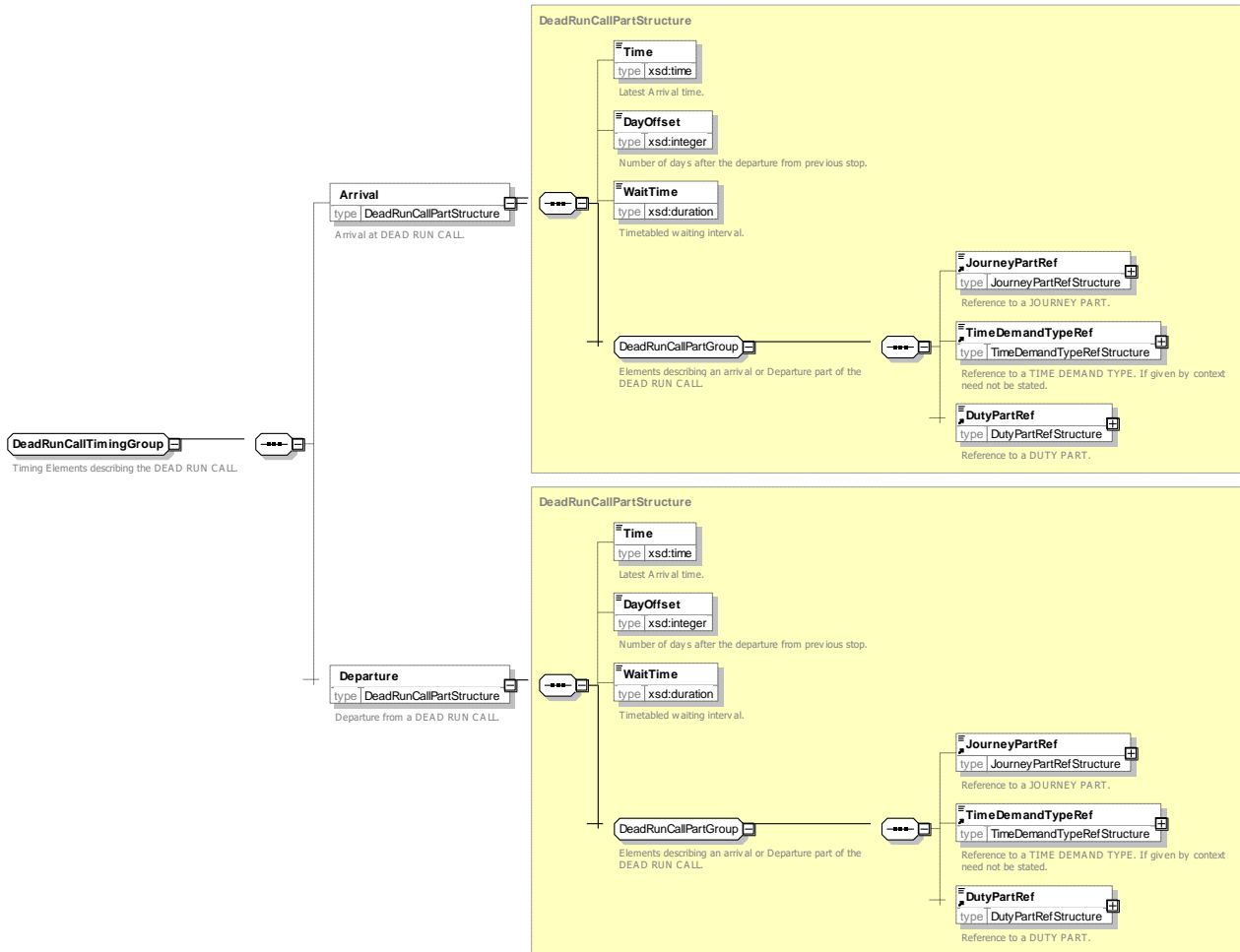


Figure 143 — *DeadRunTimingGroup* — XSD

7.2.14.10 *DeadRunDeparture* – Model Element

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

Table 82 – *DeadRunDeparture* – Element

Classification	Name	Type	Cardinality	Description
	Time	<i>xsd:time</i>	1:1	Earliest Time of Departure.
	DayOffset	<i>DayOffsetType</i>	1:1	Day offset from DEAD RUN start day. 0 = Same day.
	WaitTime	<i>xsd:duration</i>	0:*	Time to wait at stop after arrival before departure.
	CallPart	<i>DeadRunCallPartGroup</i>	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 83 – DeadRunCallPart – Element

Classification	Name	Type	Cardinality	Description
«FK»	JourneyPartRef	<i>JourneyPartRef</i>	0:1	JOURNEY PART to which CALL belongs.
«FK»	DutyPartRef	<i>DutyPartRef</i>	0:1	DUTY that applies from this CALL onwards.
«FK»	TimeDemandTypeRef	<i>TimeDemandTypeRef</i>	0:1	JOURNEY MEETINGS of this journey at this specific DEAD RUN CALL.

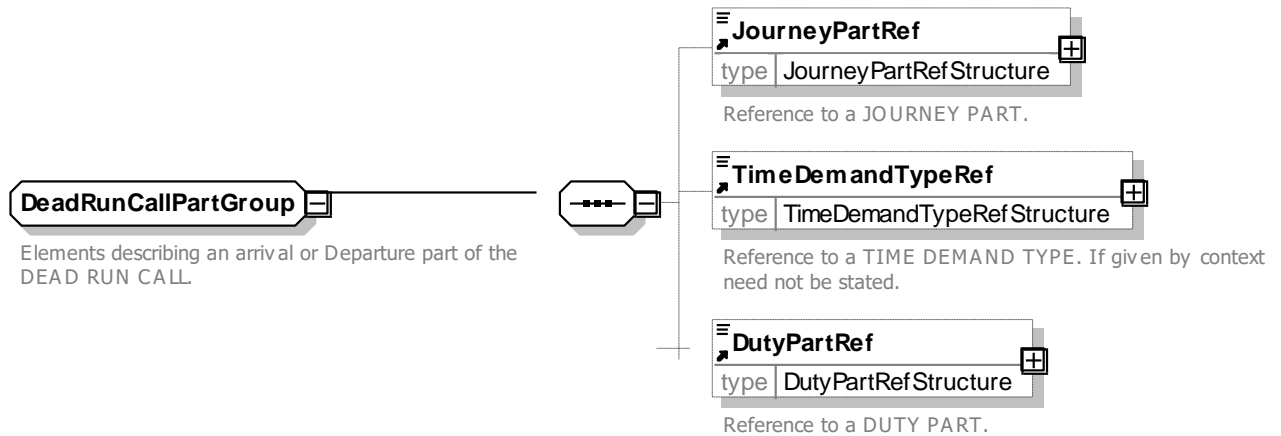


Figure 144 — DeadRunCallPart — XSD

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"/>
    </Call>
  </calls>
</ServiceJourney>
```

```

        <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.

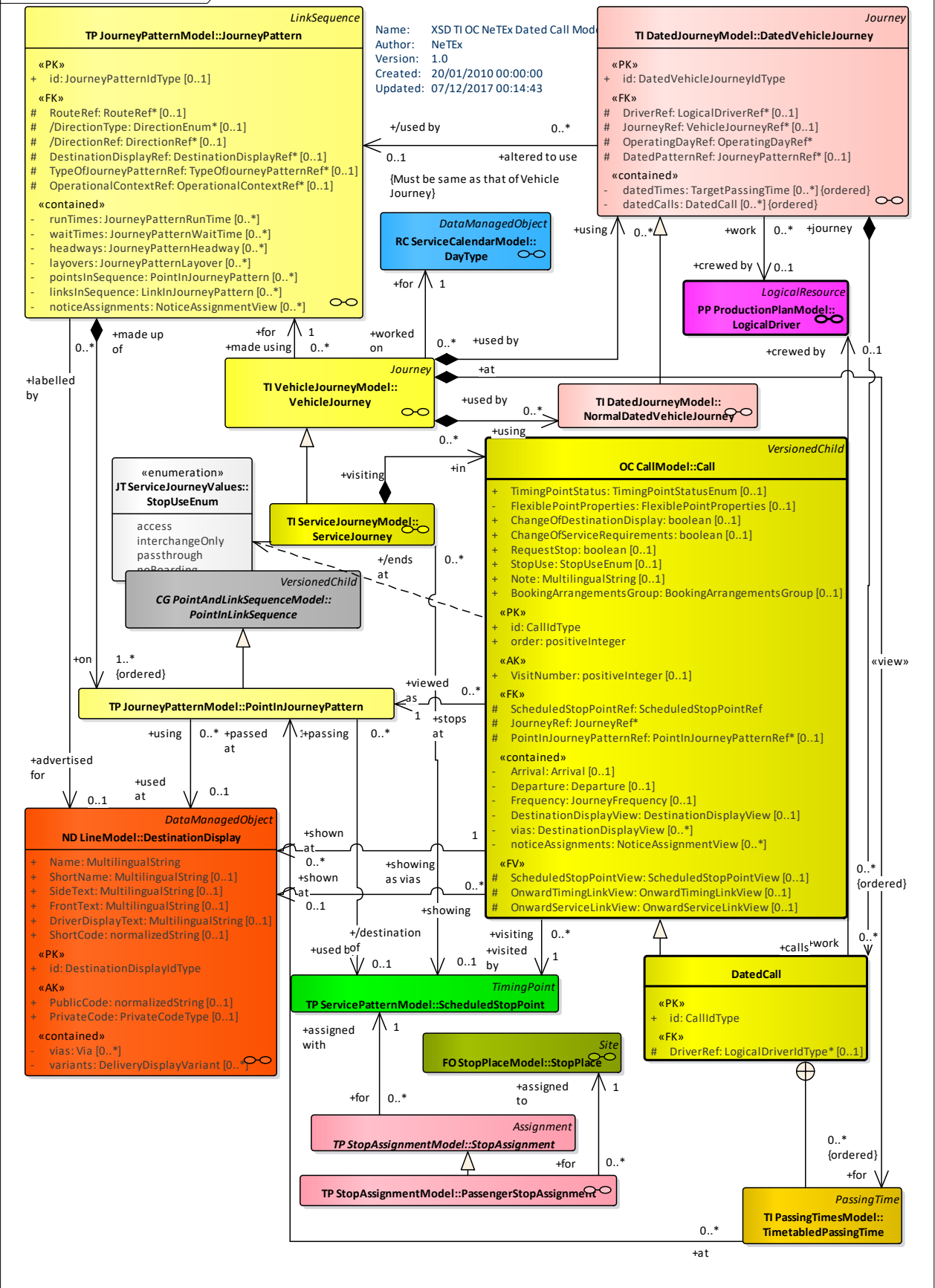


Figure 145 — Dated Call – Physical Model (UML)

7.2.15.2 DatedCall – Model Element

A CALL for a DATED VEHICLE JOURNEY.

Table 84 – DatedCall – Element

Classification	Name	Type	Cardinality	Description
::>	::>	Call	::>	DATED CALL inherits from CALL.
«PK»	<i>id</i>	CallIdType	1:1	Identifier of DATED CALL.
«FK»	<i>DriverRef</i>	LogicalDriverRef	0:1	LOGICAL DRIVER undertaking CALL.

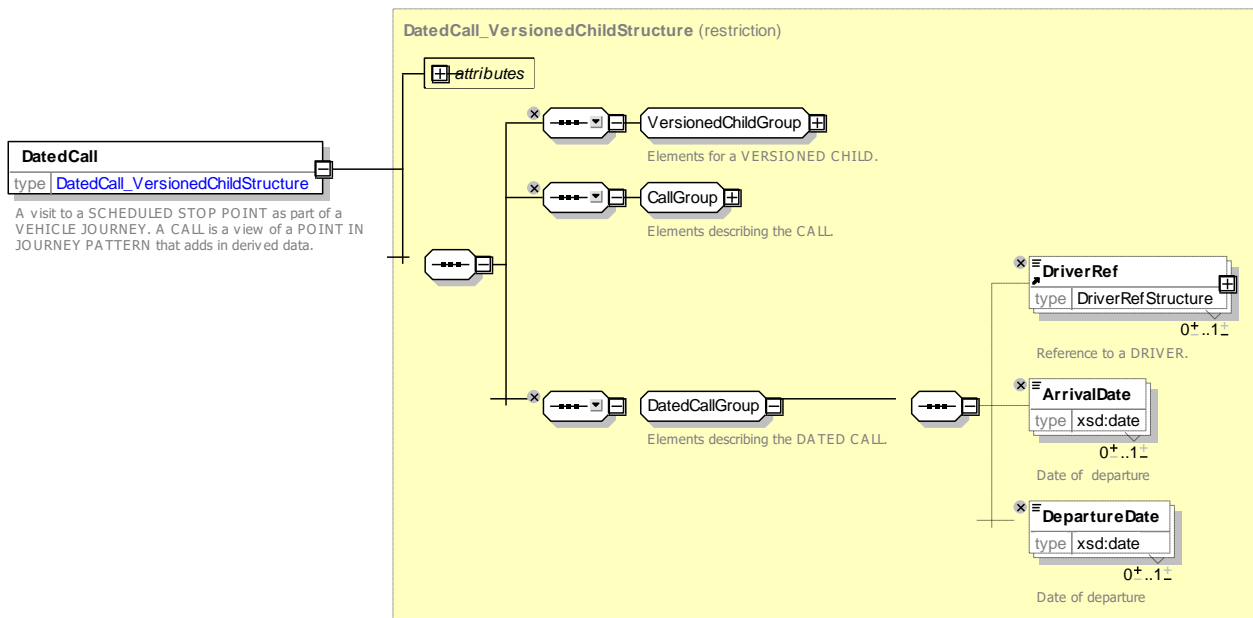


Figure 146 — DatedCall — XSD

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).

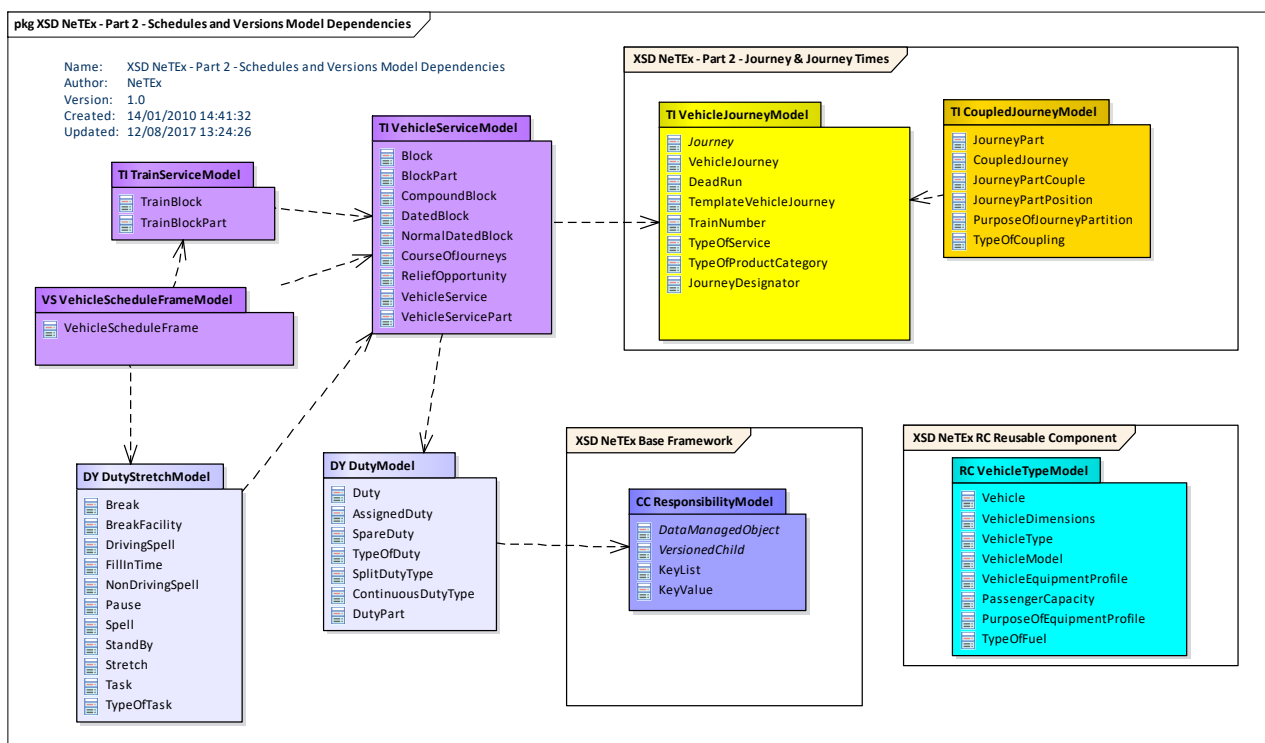


Figure 147 — Vehicle Schedules – Model Dependencies

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.

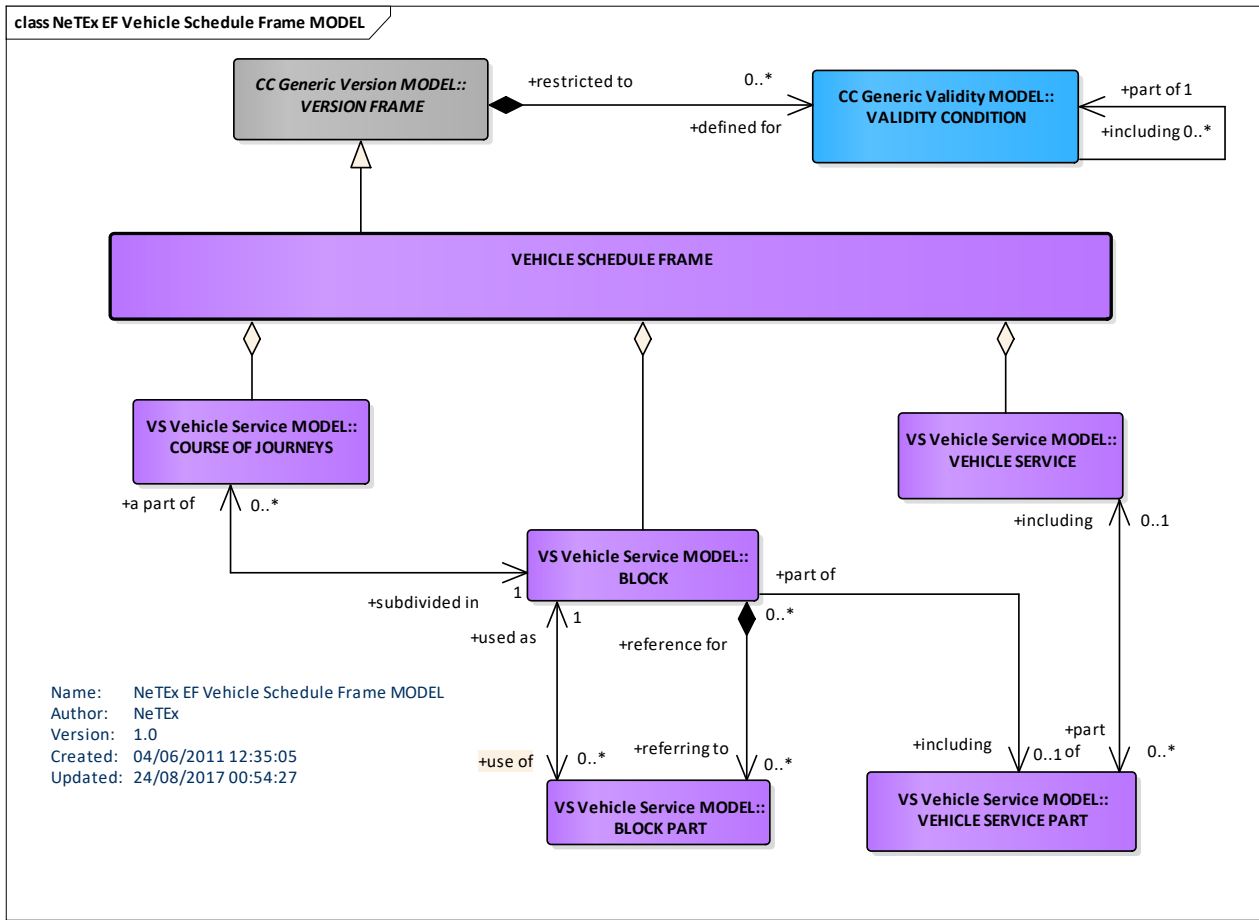


Figure 148 — Vehicle Schedule Frame – Conceptual MODEL (UML)

8.2.1.2 Vehicle Schedule Frame – Physical model

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

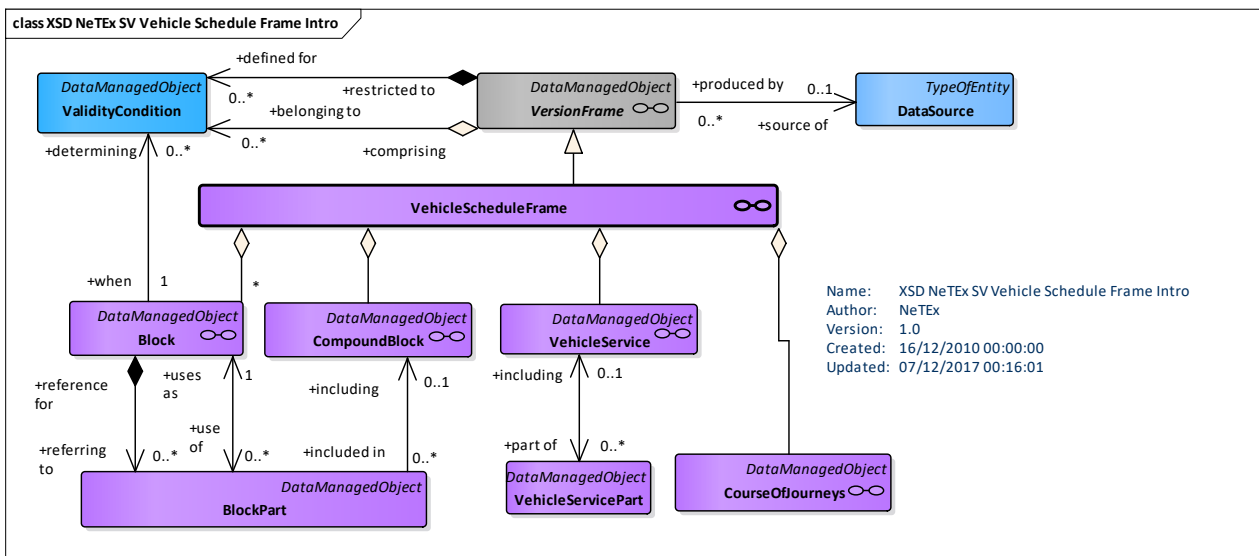


Figure 149 — Vehicle Schedule Frame Contents– Physical Model (UML)

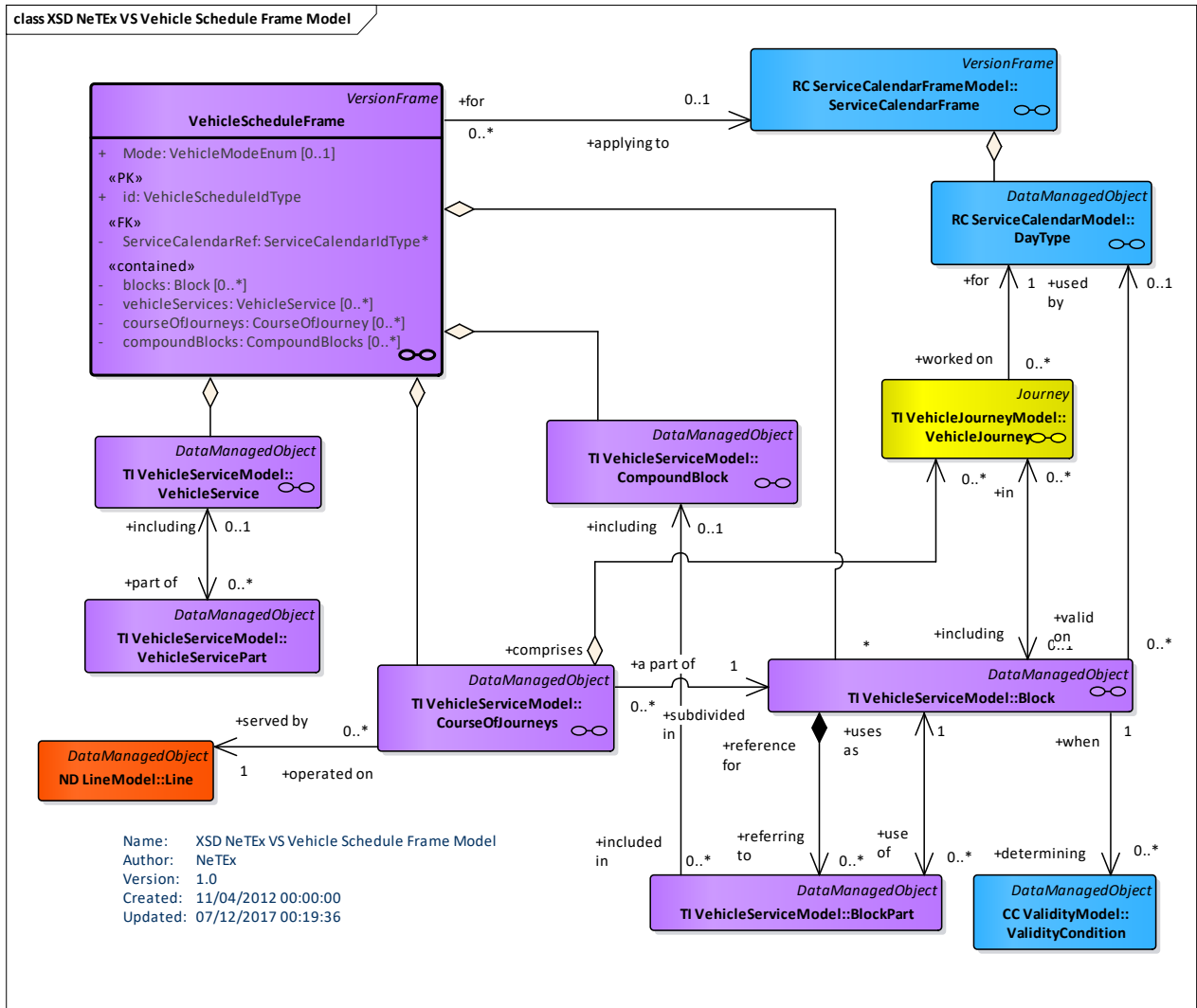


Figure 150 — Vehicle Schedule Frame – Physical Model (UML)

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 85 – 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»	ServiceCalendar-Ref	ServiceCalendarRef	1:1	Reference to SERVICE CALENDAR for VEHICLE SCHEDULE FRAME.

	Mode	<i>VehicleModeEnum</i>	0:1	Mode of VEHICLE JOURNEYS in VEHICLE SCHEDULE FRAME.
«cntd»	blocks	<i>Block TrainBlock CompoundBlock</i>	0:*	BLOCKS in VEHICLE SCHEDULE FRAME.
«cntd»	courseOfJourneys	<i>CourseOfJourney</i>	0:*	COURSES OF JOURNEYS in VEHICLE SCHEDULE FRAME.
«cntd»	vehicleServices	<i>VehicleService</i>	0:*	VEHICLE SERVICES in VEHICLE SCHEDULE FRAME.
«cntd»	reliefOpportunities	<i>ReliefOpportunity</i>	0:*	RELIEF OPPORTUNITIES in VEHICLE SCHEDULE FRAME.

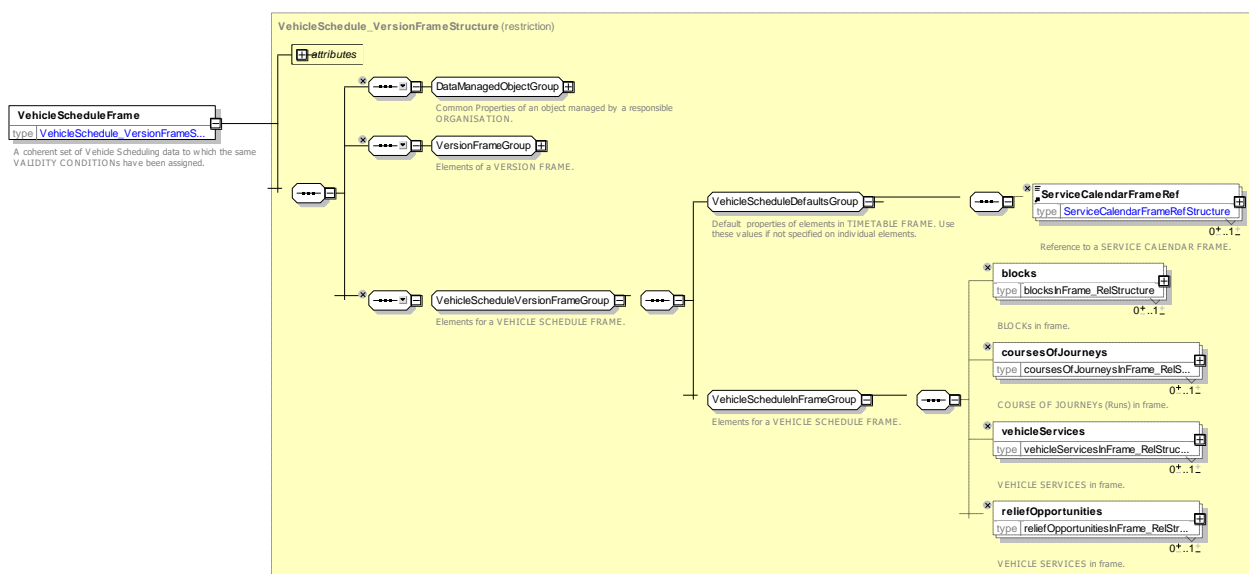


Figure 151 — VehicleScheduleFrame — XSD

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 pre-defined 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.

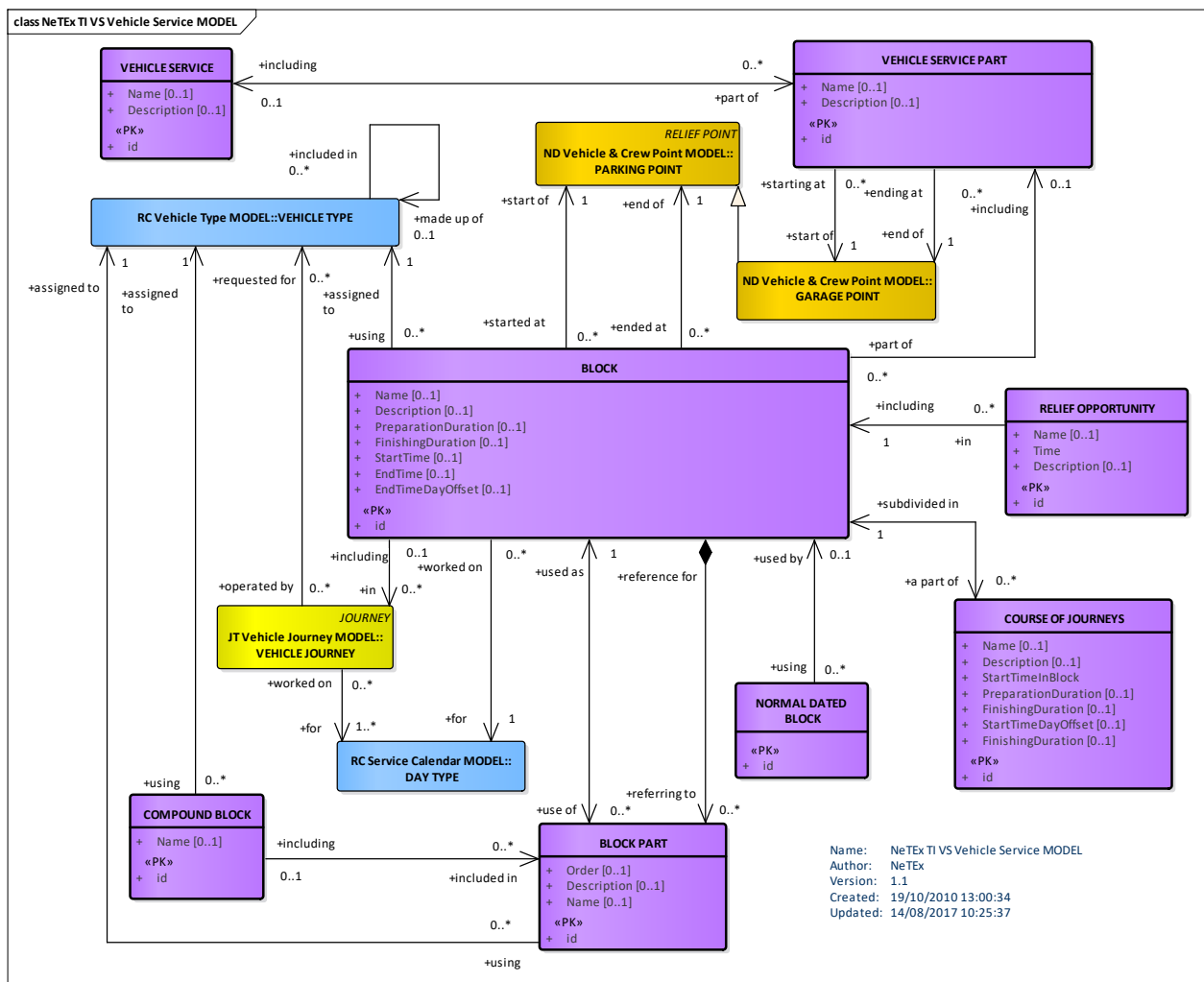


Figure 152 — Vehicle Service – Conceptual MODEL (UML)

8.2.2.2 Vehicle Service – Physical Model

The following figure shows the physical model.

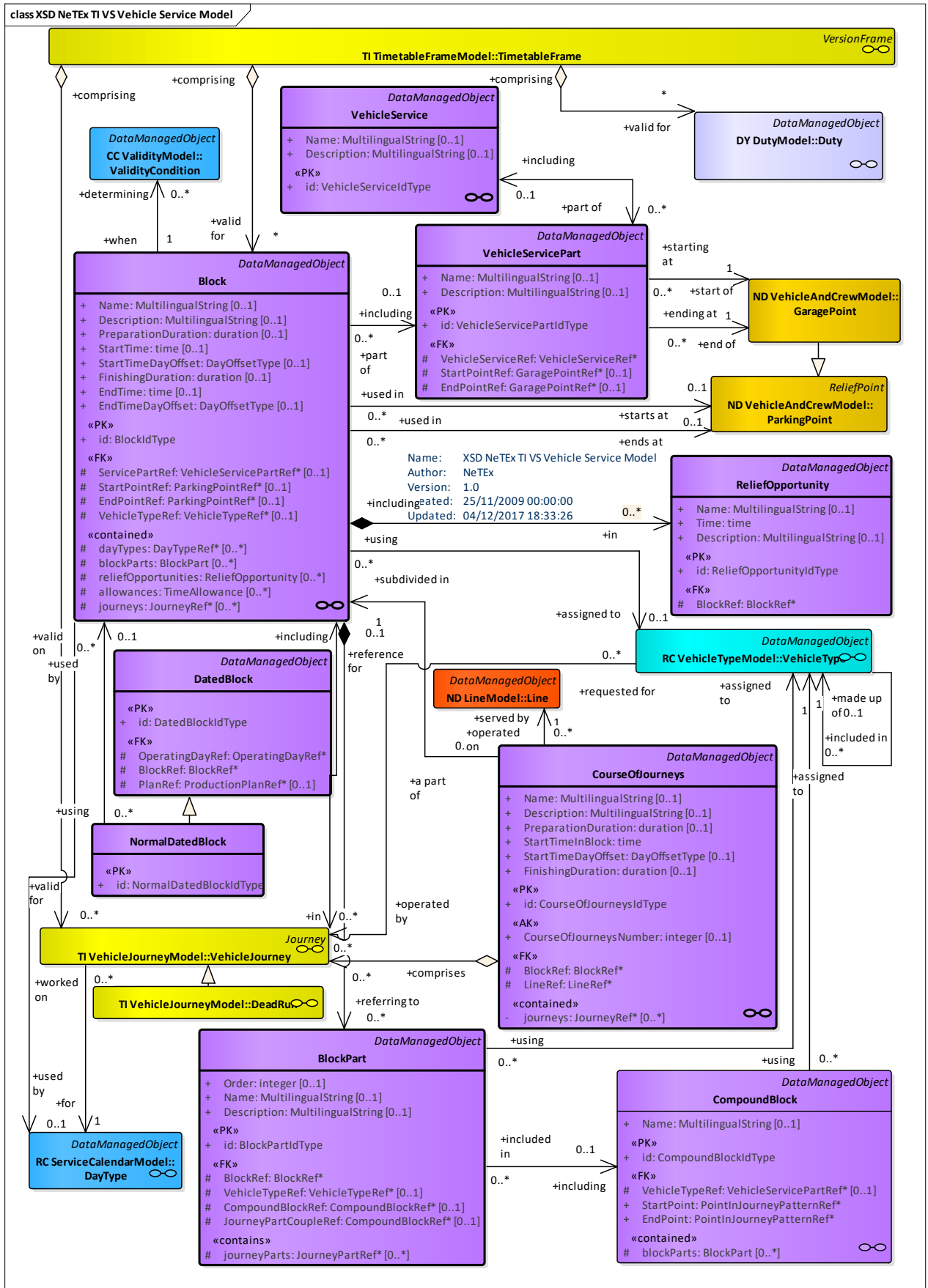


Figure 153 — Vehicle Service – Physical Model (UML)

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 86 – Block – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	BLOCK inherits from DATA MANAGED OBJECT.
«PK»	id	<i>BlockIdType</i>	1:1	Identifier of BLOCK.
	Name	<i>MultilingualString</i>	0:1	Name of BLOCK.
	Description	<i>MultilingualString</i>	0:1	Description of BLOCK.
	PrivateCode	<i>PrivateCode</i>	0:1	Private code of BLOCK.
	PreparationDuration	<i>xsd:duration</i>	0:1	How long needed to prepare to run BLOCK.
	StartTime	<i>xsd:time</i>	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.
	StartTimeDayOffset	<i>xsd:integer</i>	0:1	Day offset of Start time from current OPERATING DAY
	FinishingDuration	<i>xsd:duration</i>	0:1	How long needed to prepare to complete BLOCK.
	EndTime	<i>xsd:time</i>	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.
	EndTimeDayOffset	<i>xsd:integer</i>	0:1	Day offset of end time from current OPERATING DAY
«cntd»	validityConditions	<i>ValidityCondition</i>	0:*	VALIDITY CONDITIONS controlling use of BLOCK.
«FK»	dayTypes	<i>DayType</i>	0:1	DAY TYPE associated with BLOCK.
«FK»	VehicleServicePartRef	<i>VehicleServicePartRef</i>	0:1	SERVICE PART associated with BLOCK.
«FK»	VehicleTypeRef	<i>VehicleServicePartRef</i>	0:1	VEHICLE TYPE associated with BLOCK.
«FK»	StartPointRef	<i>PointRef</i>	0:1	POINT from which BLOCK starts.
«FK»	EndPointRef	<i>PointRef</i>	0:1	POINT at which BLOCK ends.
«cntd»	journeys	<i>JourneyRef</i>	0:*	Journeys in BLOCK.
«cntd»	coursesOfJourneys	<i>CourseOfJourneyRef</i>	0:*	Runs or Courses of Journey in BLOCK.
«cntd»	blockParts	<i>BlockPart</i>	0:*	Parts of BLOCK.
«cntd»	reliefOpportunities	<i>ReliefOpportunity</i>	0:*	RELIEF OPPORTUNITIES in BLOCK.

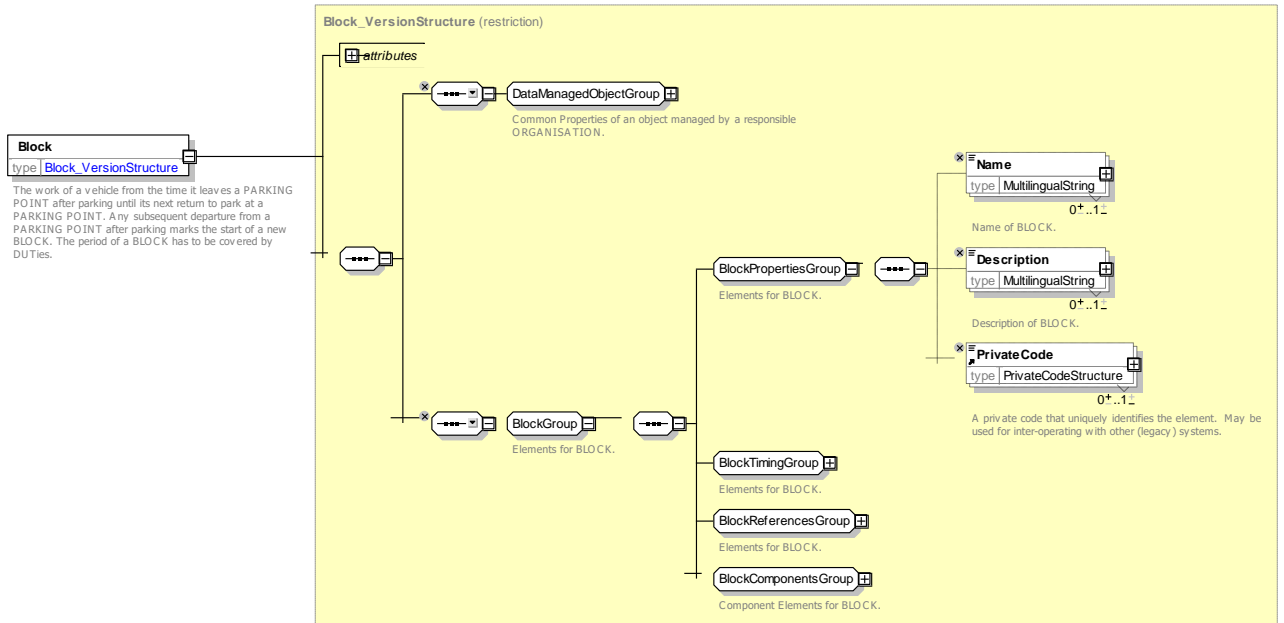


Figure 154 — *Block* — XSD

8.2.2.3.1.1 BlockTimingGroup – Model Element



Figure 155 — BlockTimingGroup — XSD

8.2.2.3.1.2 BlockReferencesGroup – Model Element

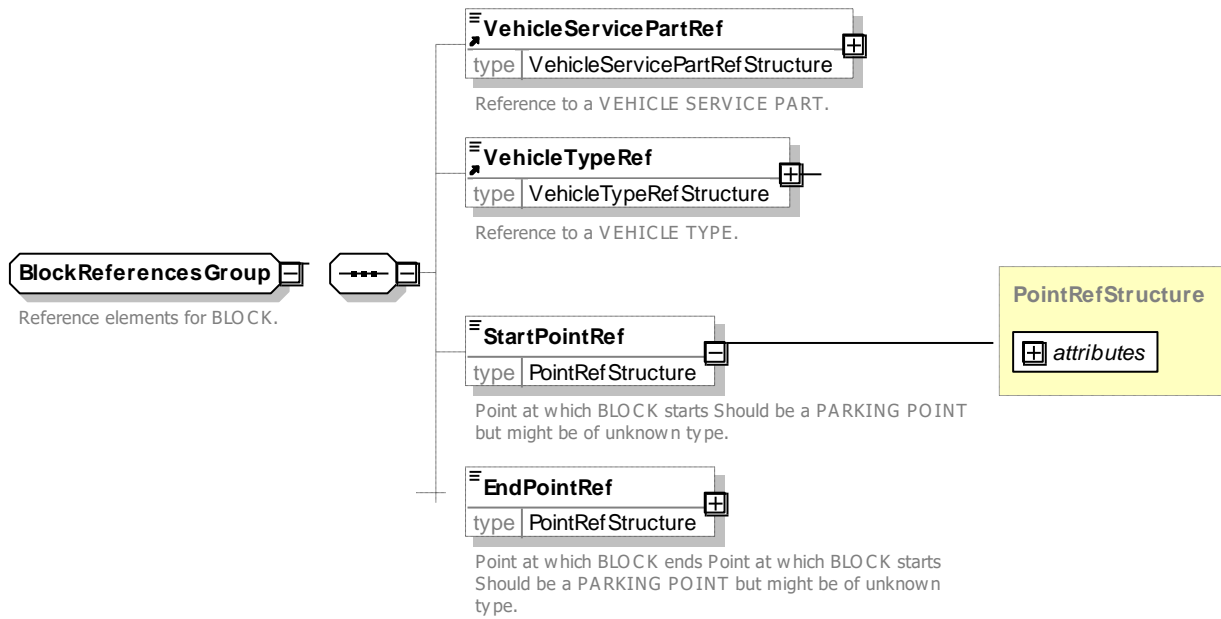


Figure 156 — *BlockReferencesGroup* — XSD

8.2.2.3.1.3 BlockComponentsGroup – Model Element

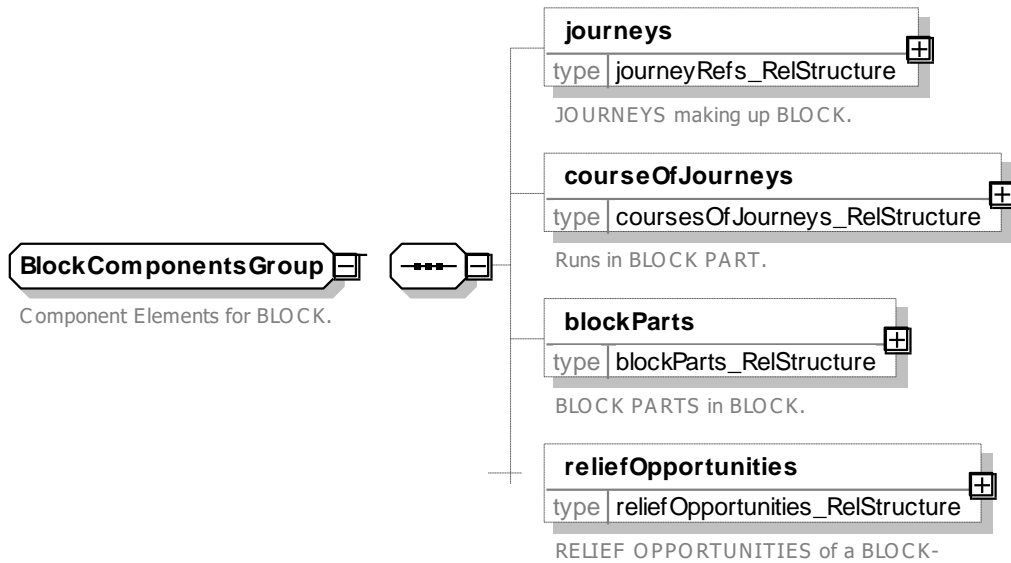


Figure 157 — *BlockComponentsGroup* — XSD

8.2.2.3.2 BlockPart – Model Element

A part of position of a vehicle BLOCK.

Table 87 – *BlockPart* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	BLOCK PART inherits from DATA MANAGED OBJECT

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

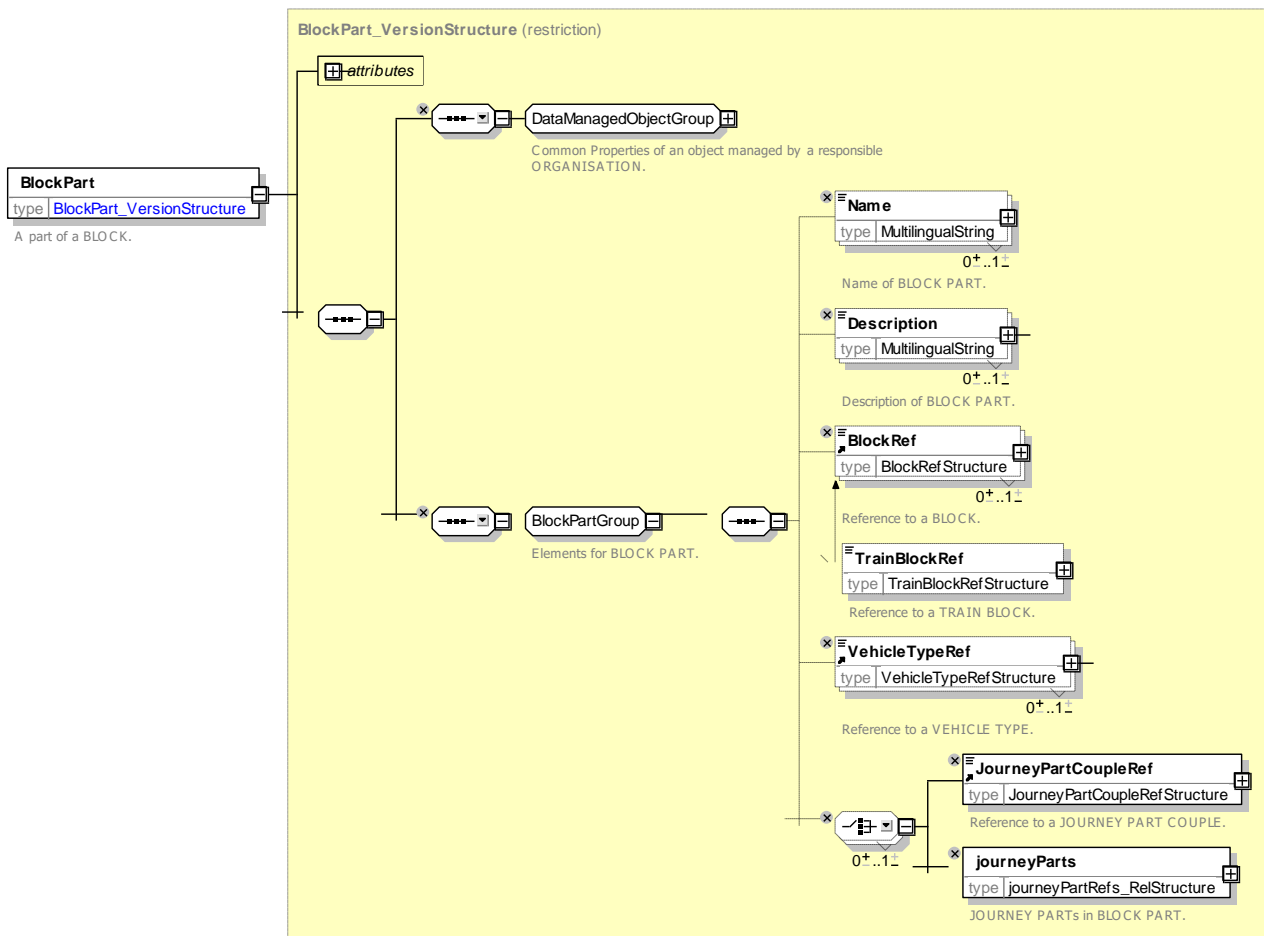


Figure 158 — BlockPart — XSD

8.2.2.3.3 CompoundBlock – Model Element

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

Table 88 – 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.

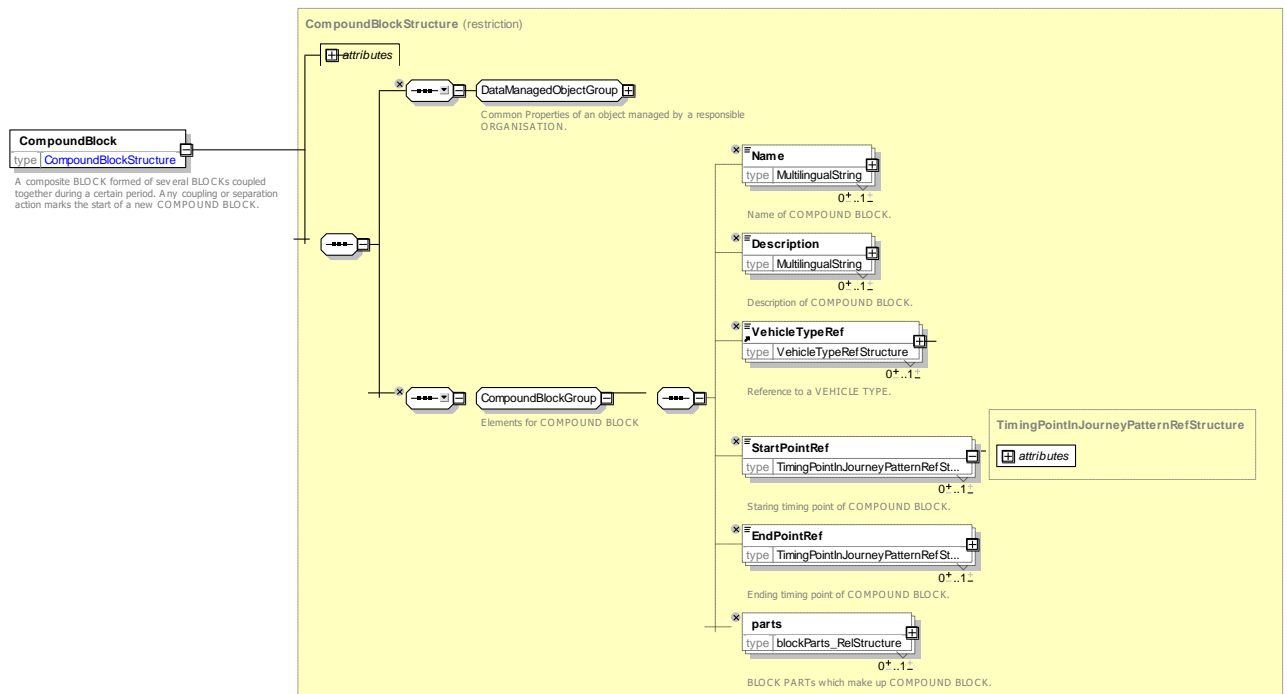


Figure 159 – CompoundBlock – XSD

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 89 – CourseOfJourney – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	COURSE OF JOURNEY inherits from DATA MANAGED OBJECT.
«PK»	id	<i>CourseOfJourneyIdType</i>	1:1	Identifier of COURSE OF JOURNEY.
	Name	<i>MultilingualString</i>	0:1	Name of COURSE OF JOURNEYS.
	Description	<i>MultilingualString</i>	0:1	Description of COURSE OF JOURNEYS.
	CourseOfJourneysNumber	xsd:integer	0:1	Number for COURSE OF JOURNEYS.
	PrivateCode	<i>PrivateCodeStructure</i>	0:1	Private Code for COURSE OF JOURNEYS.
	PreparationDuration	<i>xsd:duration</i>	0:1	How long needed to prepare for COURSE OF JOURNEY in BLOCK.
	StartTimeInBlock	<i>xsd:time</i>	1:1	Start time of COURSE OF JOURNEY in BLOCK.
	StartTimeDayOffset	<i>DayOffsetType</i>	0:1	Day offset of start time from OPERATING DAY at start of Journey.
	FinishingDuration	<i>xsd:duration</i>	0:1	How long COURSE OF JOURNEY in BLOCK.
«FK»	BlockRef	<i>BlockRef</i>	1:1	BLOCK with which COURSE OF JOURNEY is ASSOCIATED.
«FK»	LineRef	<i>LineRef</i>	1:1	LINE ASSOCIATED with COURSE OF JOURNEY.
«contained»	journeys	<i>JourneyRef</i>	0:*	VEHICLE JOURNEYS associated with COURSE OF JOURNEY.

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.

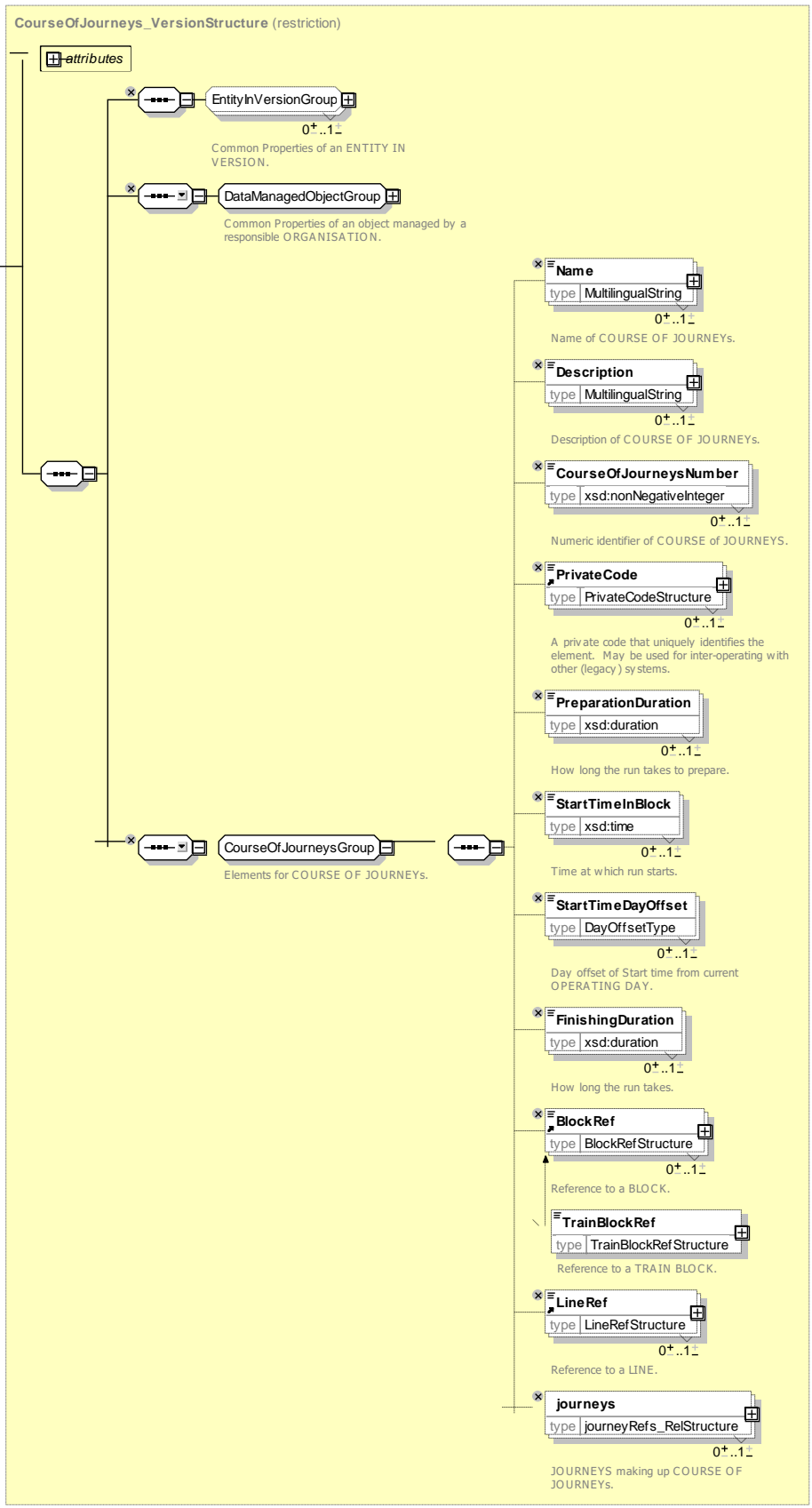


Figure 160 — CourseOfJourneys — XSD

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 90 – ReliefOpportunity – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	RELIEF OPPORTUNITY inherits from DATA MANAGED OBJECT.
«PK»	id	<i>ReliefOpportunityIdType</i>	1:1	Identifier of RELIEF OPPORTUNITY.
	Name	<i>MultilingualString</i>	0:1	NAME of RELIEF OPPORTUNITY.
	Description	<i>MultilingualString</i>	0:1	DESCRIPTION of RELIEF OPPORTUNITY.
	Time	<i>xsd.time</i>	1:1	TIME at which RELIEF OPPORTUNITY takes place.
	DayOffset	<i>DayOffsetType</i>	0:1	Day offset of time from OPERATING DAY at start of Journey. +v1.1
«FK»	BlockRef	<i>BlockRef</i>	1:1	BLOCK associated with RELIEF OPPORTUNITY.
	Description	<i>MultilingualString</i>	0:1	DESCRIPTION of RELIEF OPPORTUNITY.

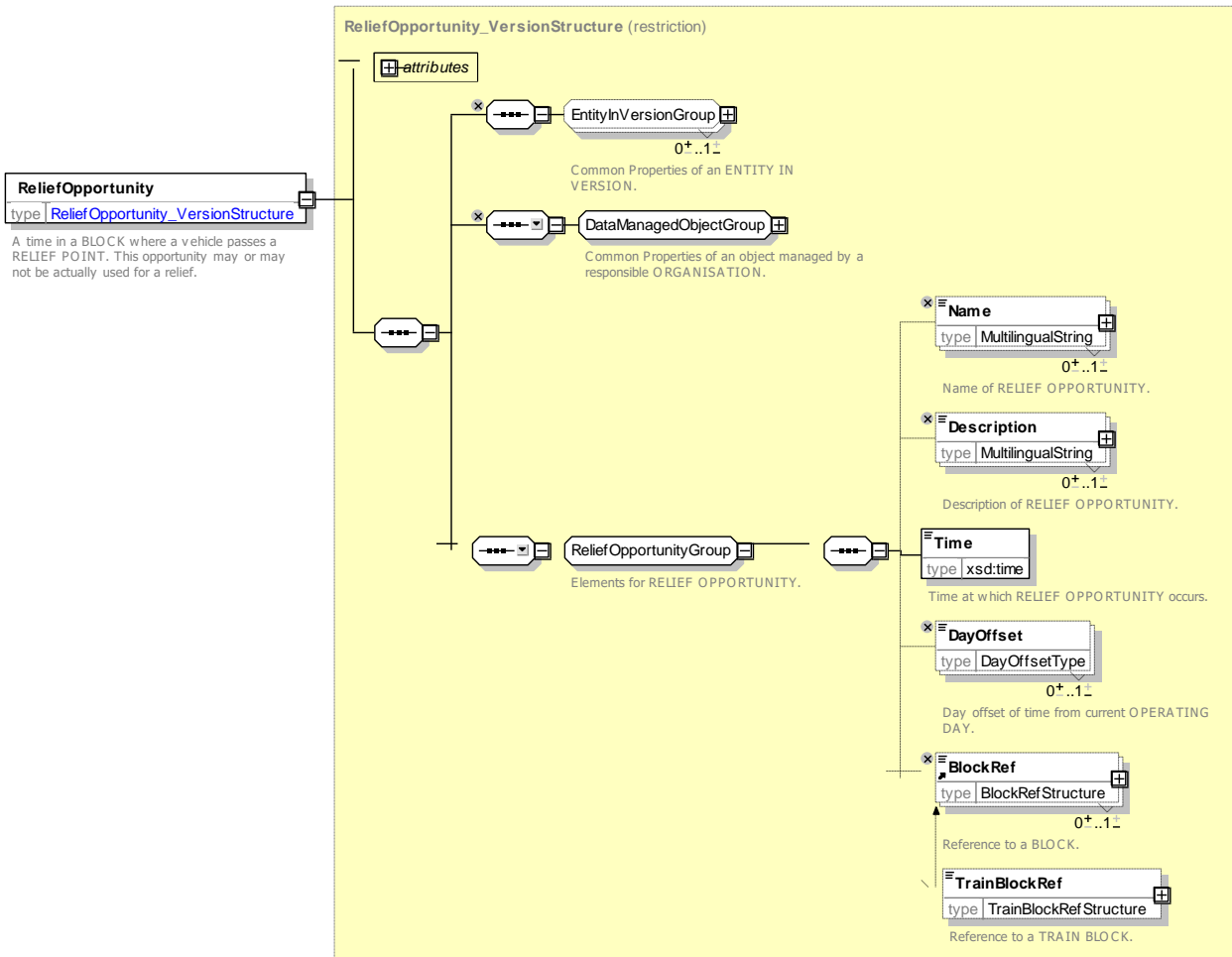


Figure 161 — ReliefOpportunity — XSD

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 91 – 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.

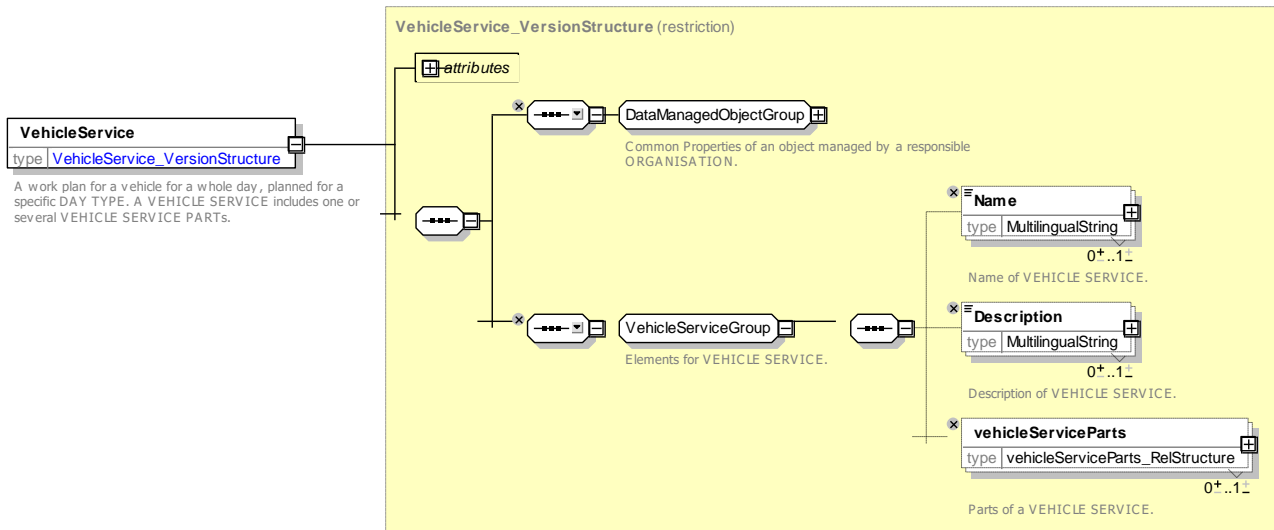


Figure 162 — VehicleService — XSD

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 92 – 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.

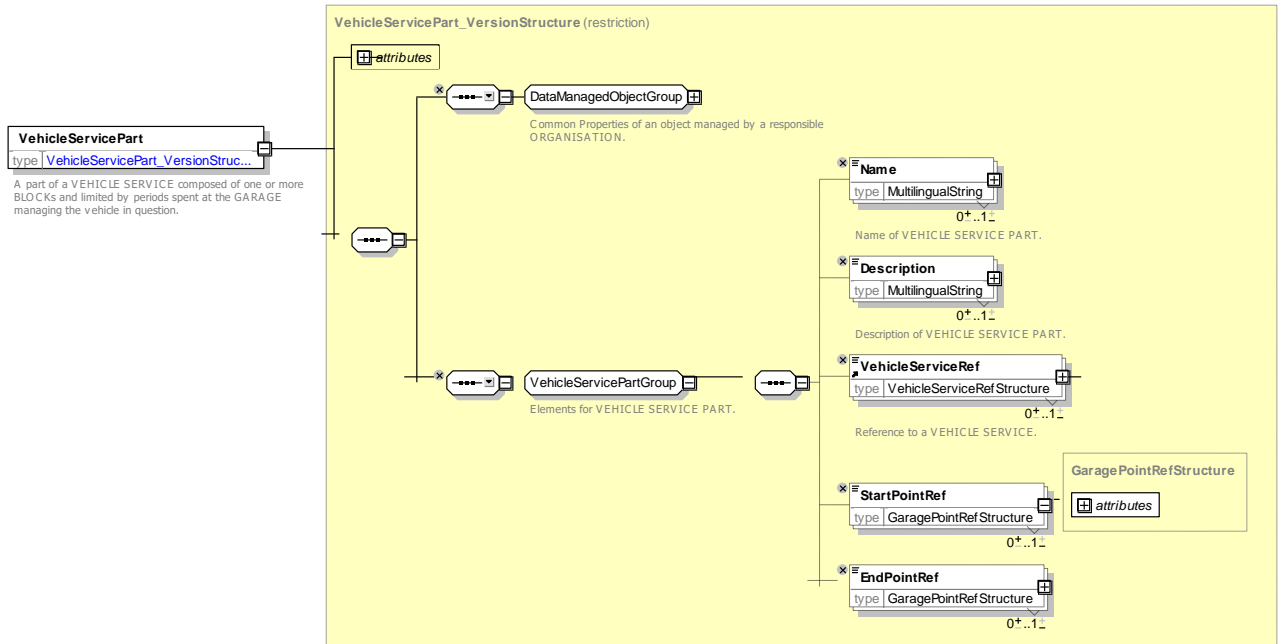


Figure 163 — VehicleServicePart — XSD

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.

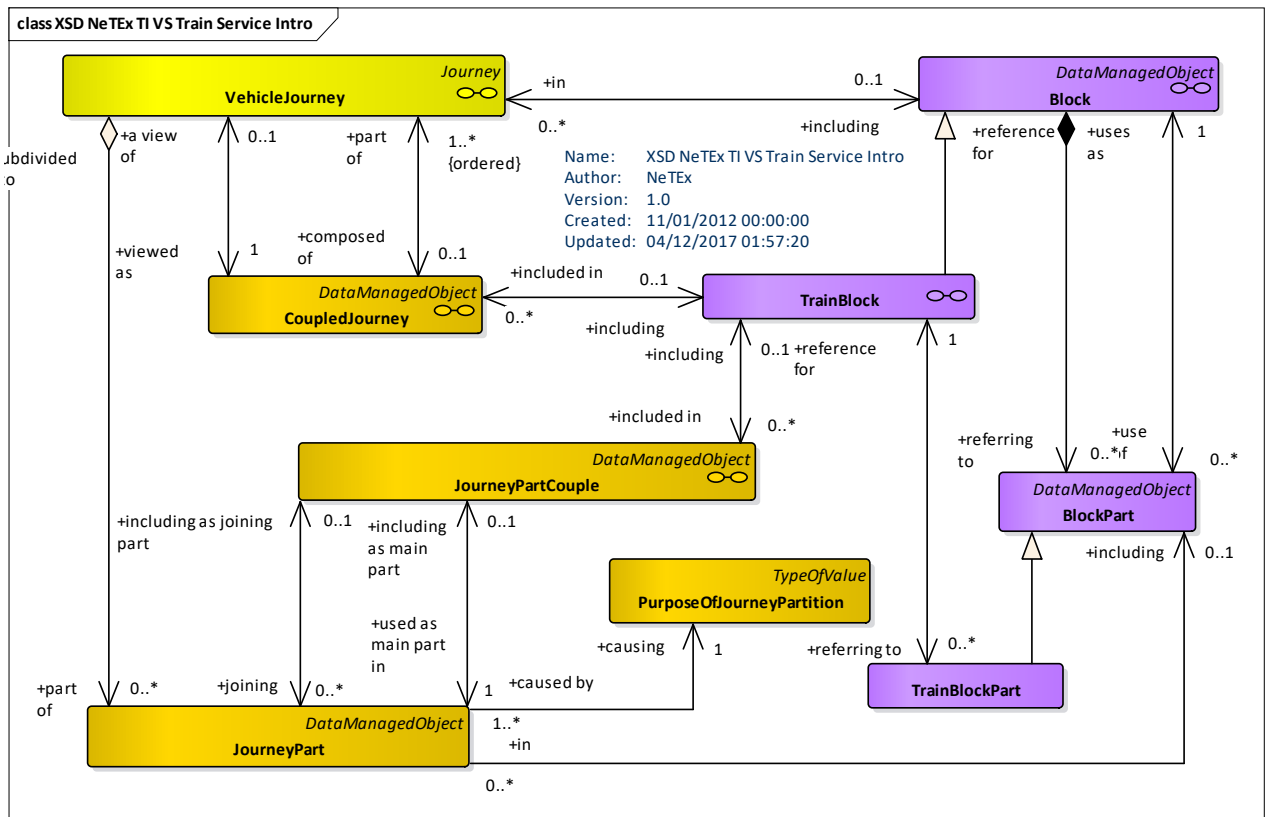


Figure 164 — Train Service Model – Conceptual MODEL (UML)

8.2.3.2 Train Service – Physical Model

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

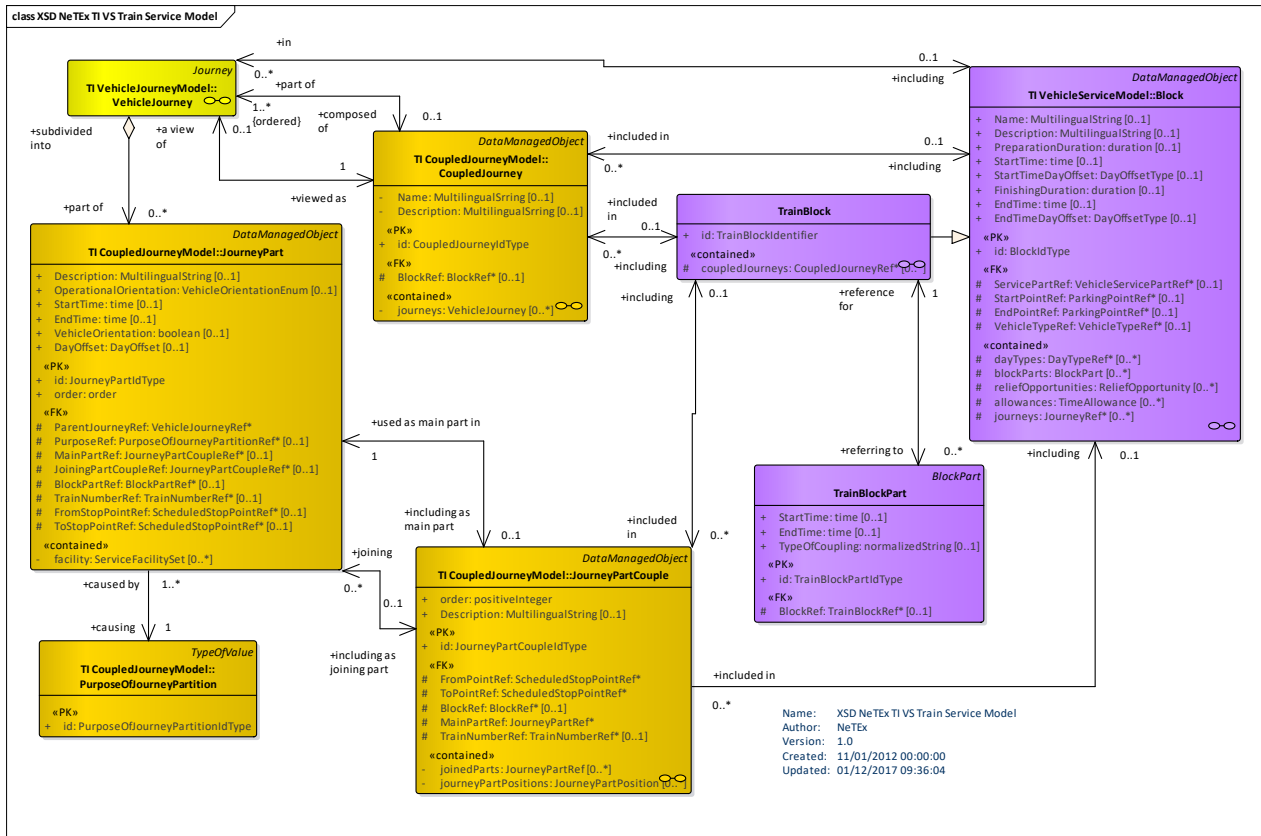


Figure 165 — Train Service Model – Physical Model (UML)

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 93 – TrainBlock – Element

Classification	Name	Type	Cardinality	Description
::>	::>	Block	::>	TRAIN BLOCK inherits from BLOCK.
	<i>id</i>	TrainBlockIdentifier	1:1	Identifier of a TRAIN BLOCK.
«cntd»	<i>coupledJourneys</i>	CoupledJourneyRef	0:*	COUPLED JOURNEYS associated with COURSE OF JOURNEY.

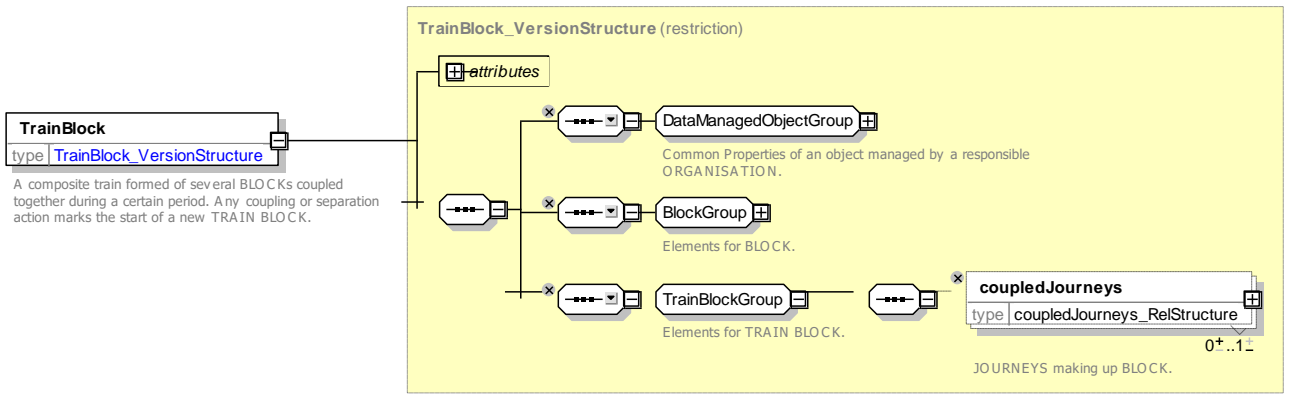


Figure 166 — TrainBlock — XSD

8.2.3.3.2 TrainBlockPart – Model Element

The position of a vehicle BLOCK within a TRAIN BLOCK.

Table 94 – TrainBlockPart – Element

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

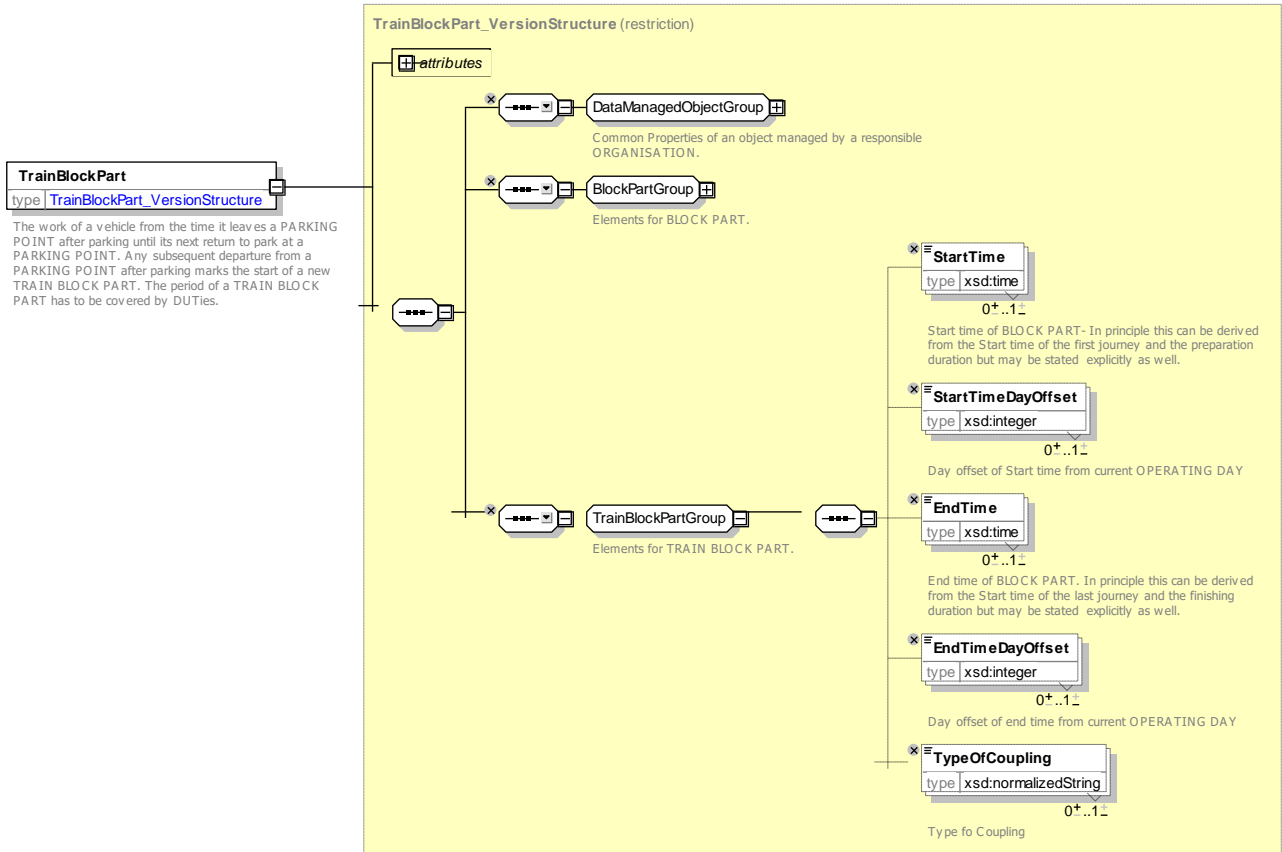


Figure 167 — *TrainBlockPart* — XSD

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.

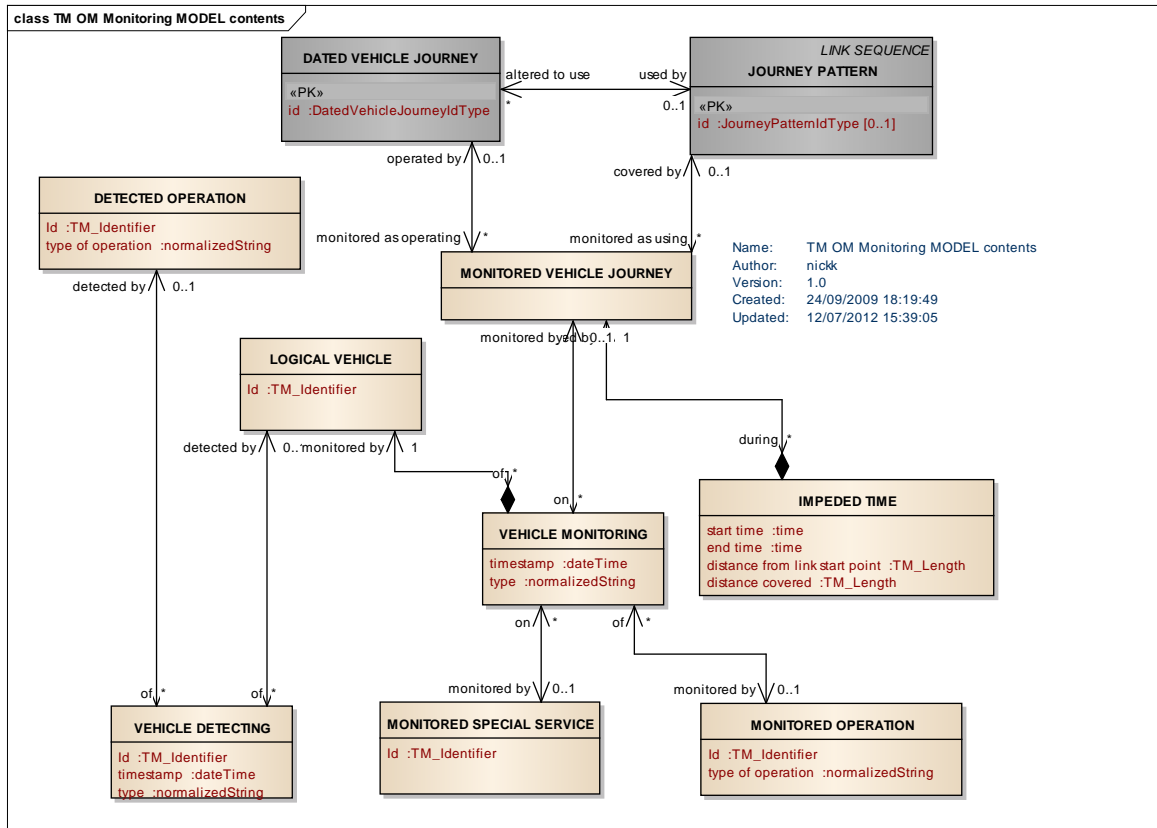


Figure 168 — Monitored Vehicle Journey — Conceptual MODEL (UML)

A.2.1.2 Monitored Vehicle Journey – Physical Model

The following diagram shows the MONITORED VEHICLE JOURNEY physical model.

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 95 – MonitoredVehicleJourney – Element

Classification	Name	Type	Cardinality	Description
«PK»	<i>id</i>	<i>MonitoredVehicleJourney-IdType</i>	1:1	Identifier of MONITORED VEHICLE JOURNEY.
«FK»	<i>DatedVehicleJourneyRef</i>	<i>DatedVehicleJourneyRef</i>	0:1	DATED VEHICLE JOURNEY which this MONITORED VEHICLE JOURNEY follows.
«cntd»	<i>passingTimes</i>	<i>DatedPassingTime</i>	0:*	Passing times for MONITORED VEHICLE JOURNEY.
«cntd»	<i>calls</i>	<i>MonitoredCall</i>	0:*	MONITORED CALLs for MONITORED VEHICLE JOURNEY.
«cntd»	<i>previousCalls</i>	<i>PreviousCall</i>	0:1	PREVIOUS CALLs before current stop in JOURNEY.
«cntd»	<i>onwardCalls</i>	<i>OnwardCall</i>	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 96 – MonitoredSpecialService – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Journey</i>	::>	MONITORED SPECIAL SERVICE inherits from JOURNEY.
«PK»	<i>id</i>	<i>MonitoredSpecialServiceIdType</i>	1:1	Identifier of MONITORED SPECIAL SERVICE.
«FK»	<i>SpecialServiceRef</i>	<i>SpecialServiceRef</i>	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 97 – MonitoredCall – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Call</i>	::>	MONITORED CALL inherits from CALL.

«cntd»	TargetTimes	<i>TargetPassingTime</i>	0:1	Intended times at TIMING POINT IN JOURNEY PATTERN.,
«cntd»	EstimatedTimes	<i>EstimatedPassingTime</i>	0:1	Estimated times at TIMING POINT IN JOURNEY PATTERN.
«cntd»	ObservedTimes	<i>ObservedPassingTime</i>	0:1	Observed times at TIMING POINT IN JOURNEY PATTERN.
	previousCalls	<i>PreviousCall</i>	0:*	CALLs previous to this CALL.
	onwardsCalls	<i>OnwardCall</i>	0:*	CALLs subsequent to this CALL.

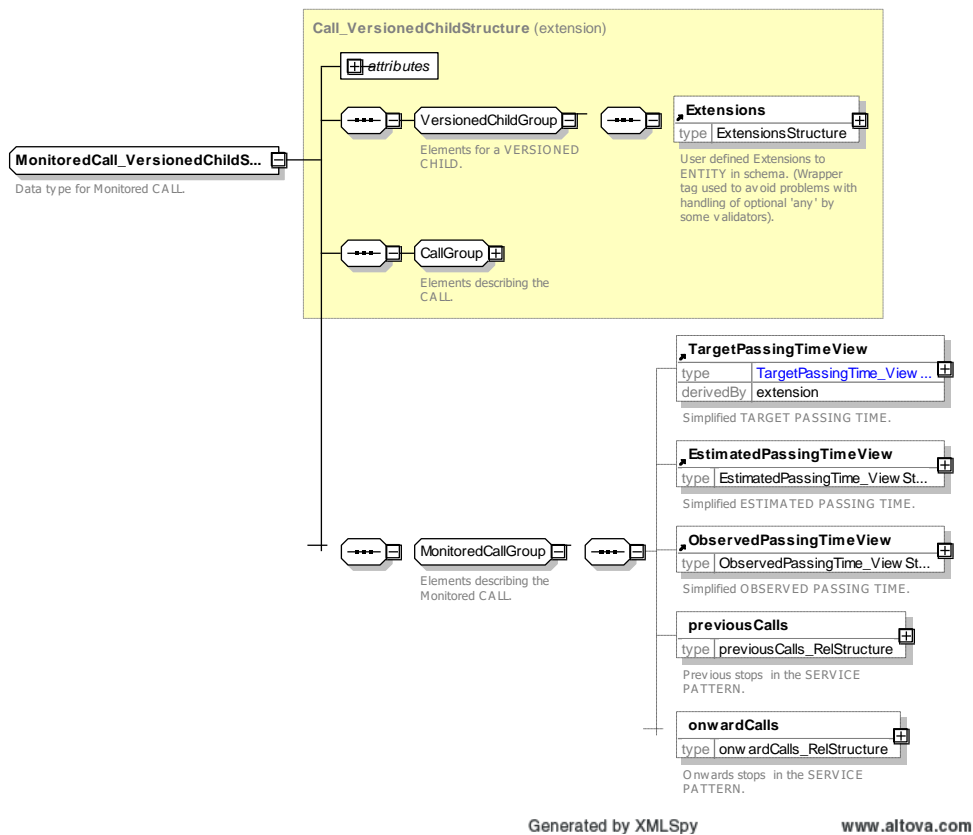


Figure 170 — MonitoredCall — XSD

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 98 – PreviousCall – Element

Classification	Name	Type	Cardinality	Description
::>	::>	Call	::>	PREVIOUS CALL inherits from CALL.

«cntd»	TargetTimes	<i>TargetPassingTime</i>	0:1	Intended times at TIMING POINT IN JOURNEY PATTERN.
«cntd»	ObservedTimes	<i>ObservedPassingTime</i>	0:1	Observed times at TIMING POINT IN JOURNEY PATTERN.

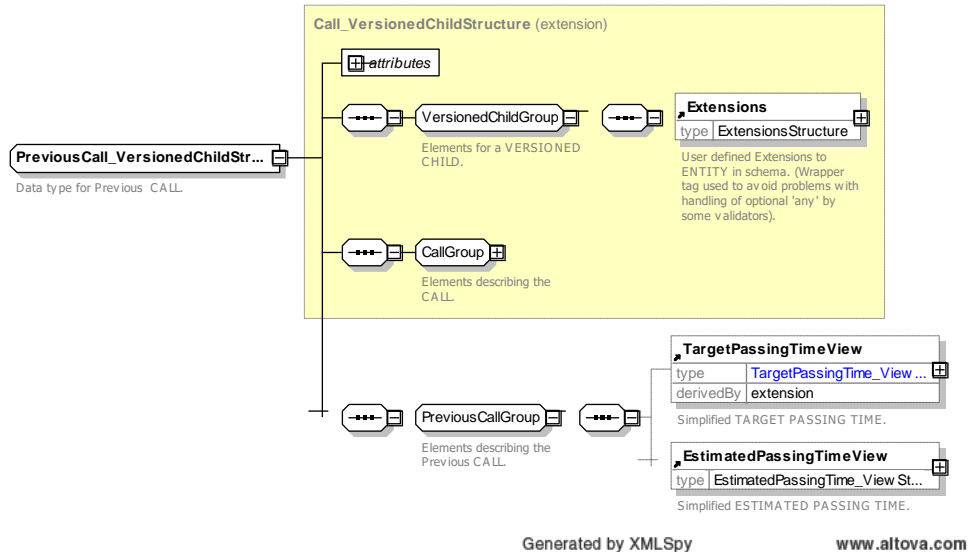


Figure 171 — PreviousCall — XSD

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 99 – OnwardCall – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Call</i>	::>	ONWARDS CALL inherits from CALL.
«cntd»	TargetTimes	<i>TargetPassingTime</i>	0:1	Intended times at TIMING POINT IN JOURNEY PATTERN.
«cntd»	EstimatedTimes	<i>EstimatedPassingTime</i>	0:1	Estimated times at TIMING POINT IN JOURNEY PATTERN.

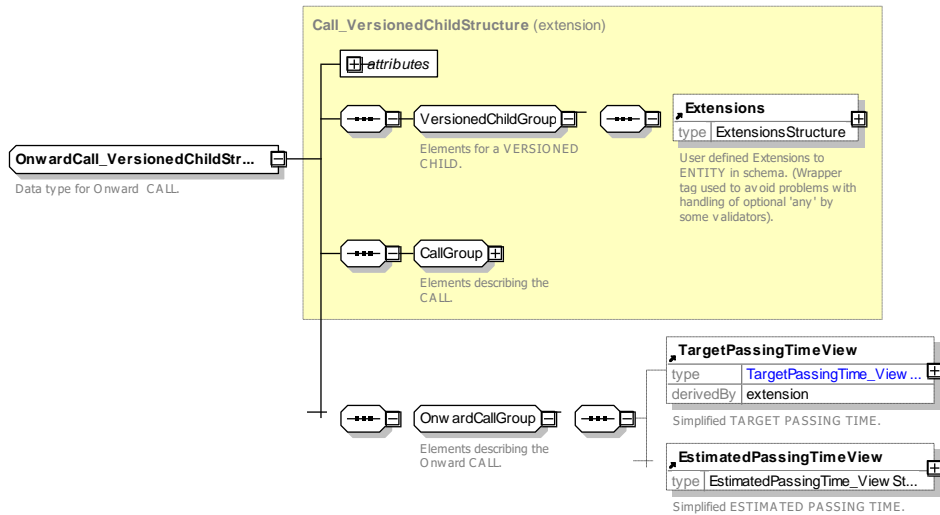


Figure 172 — OnwardCall — XSD

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).

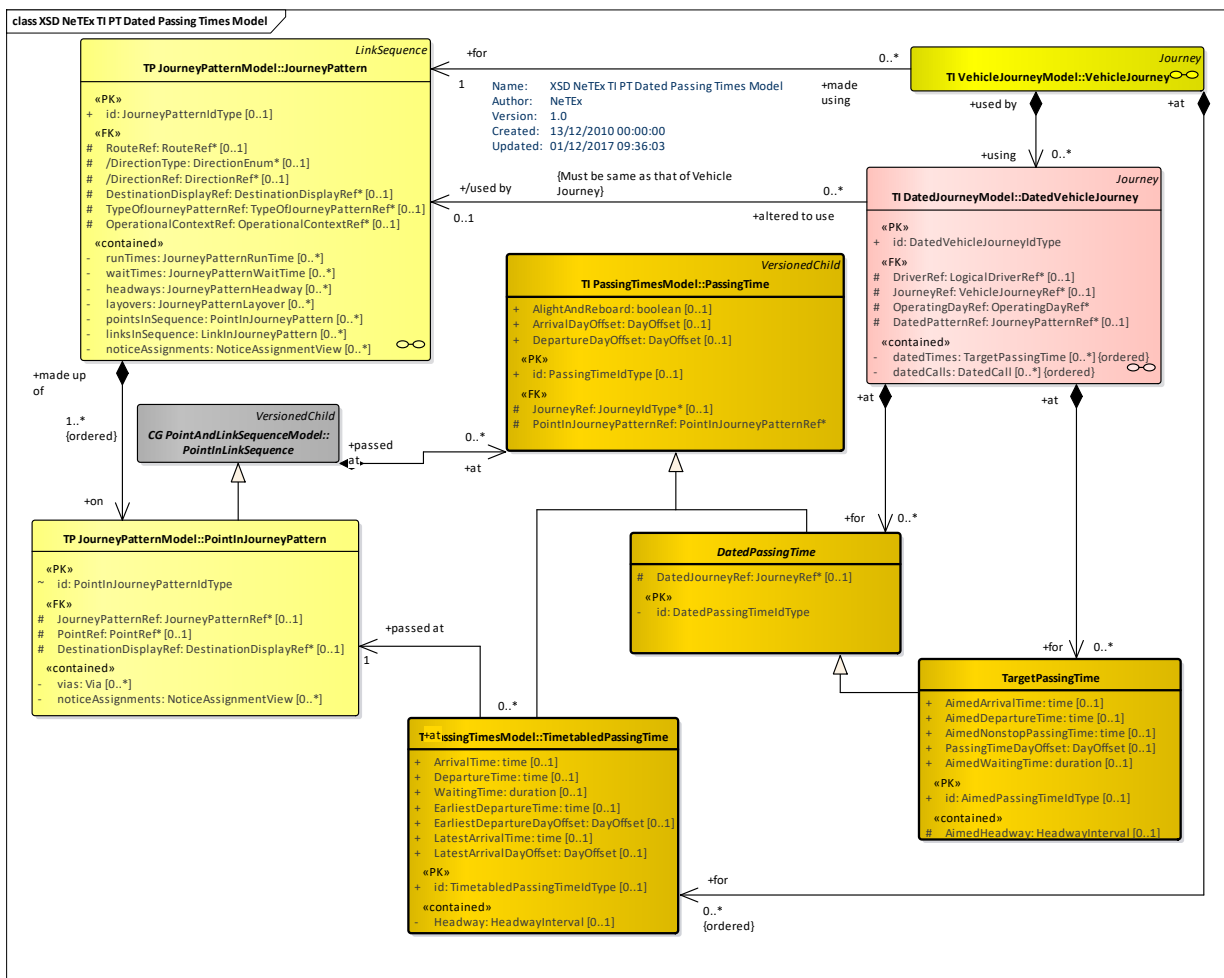


Figure 173 — Dated Passing Times – Physical Model (UML)

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 100 – EstimatedPassingTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DatedPassingTime</i>	::>	ESTIMATED PASSING TIME inherits from DATED PASSING TIME.
«PK»	id	<i>EstimatedPassing-TimeIdType</i>	0:1	Identifier of ESTIMATED PASSING TIME.
	ExpectedArrivalTime	<i>xsd:time</i>	0:1	Expected arrival time at POINT IN JOURNEY PATTERN.
	ArrivalDayOffset	<i>DayOffsetType</i>	0:1	Arrival day offset from start of JOURNEY. +V1.1
	ExpectedDepartureTime	<i>xsd:time</i>	0:1	Expected departure time at POINT IN JOURNEY PATTERN.
	DepartureDayOffset	<i>DayOffsetType</i>	0:1	Departure day offset from start of JOURNEY. +V1.1
	ExpectedWaitingTime	<i>xsd:duration</i>	0:1	Expected waiting time at POINT IN JOURNEY PATTERN.
	ExpectedNonstopPassingTime	<i>xsd:time</i>	0:1	Expected nonstop passing time at POINT IN JOURNEY PATTERN.
	PassingTimeDayOffset	<i>DayOffsetType</i>	0:1	Passing time day offset from start of JOURNEY. +V1.1
	ExpectedHeadway	<i>HeadwayInterval</i>	0:1	Expected headway time at POINT IN JOURNEY PATTERN.

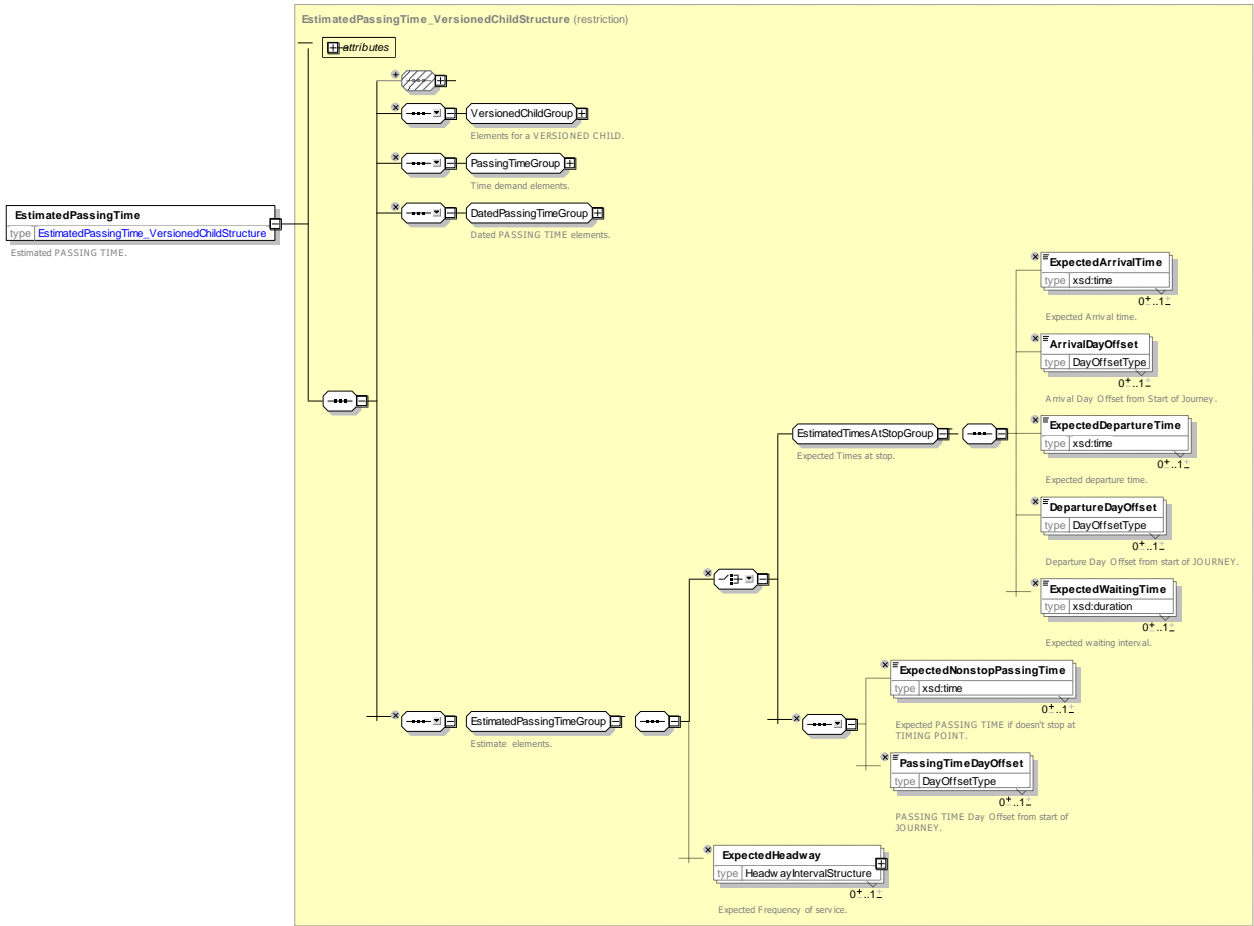


Figure 174 — EstimatedPassingTime — XSD

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 101 – ObservedPassingTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	DatedPassingTime	::>	OBSERVED PASSING TIME inherits from DATED PASSING TIME.
«PK»	id	Observed Passing-TimeIdType	0:1	Identifier of OBSERVED PASSING TIME.
	ActualArrivalTime	xsd:time	0:1	Actual arrival time at POINT IN JOURNEY PATTERN.
	ArrivalDayOffset	DayOffsetType	0:1	Arrival day offset from start of JOURNEY. +V1.1
	ActualDepartureTime	xsd:time	0:1	Actual departure time at POINT IN JOURNEY PATTERN.
	DepartureDayOffset	DayOffsetType	0:1	Departure day offset from start of JOURNEY. +V1.1

	ActualWaitingTime	<i>xsd:duration</i>	0:1	Actual waiting time at POINT IN JOURNEY PATTERN.
	ActualNonstopPassingTme	<i>xsd:time</i>	0:1	Actual nonstop passing time at POINT IN JOURNEY PATTERN.
	PassingTimeDayOffset	<i>DayOffsetType</i>	0:1	Passing time day offset from start of JOURNEY. +V1.1
	ActualHeadway	<i>HeadwayInterval</i>	0:1	Actual headway time at POINT IN JOURNEY PATTERN.

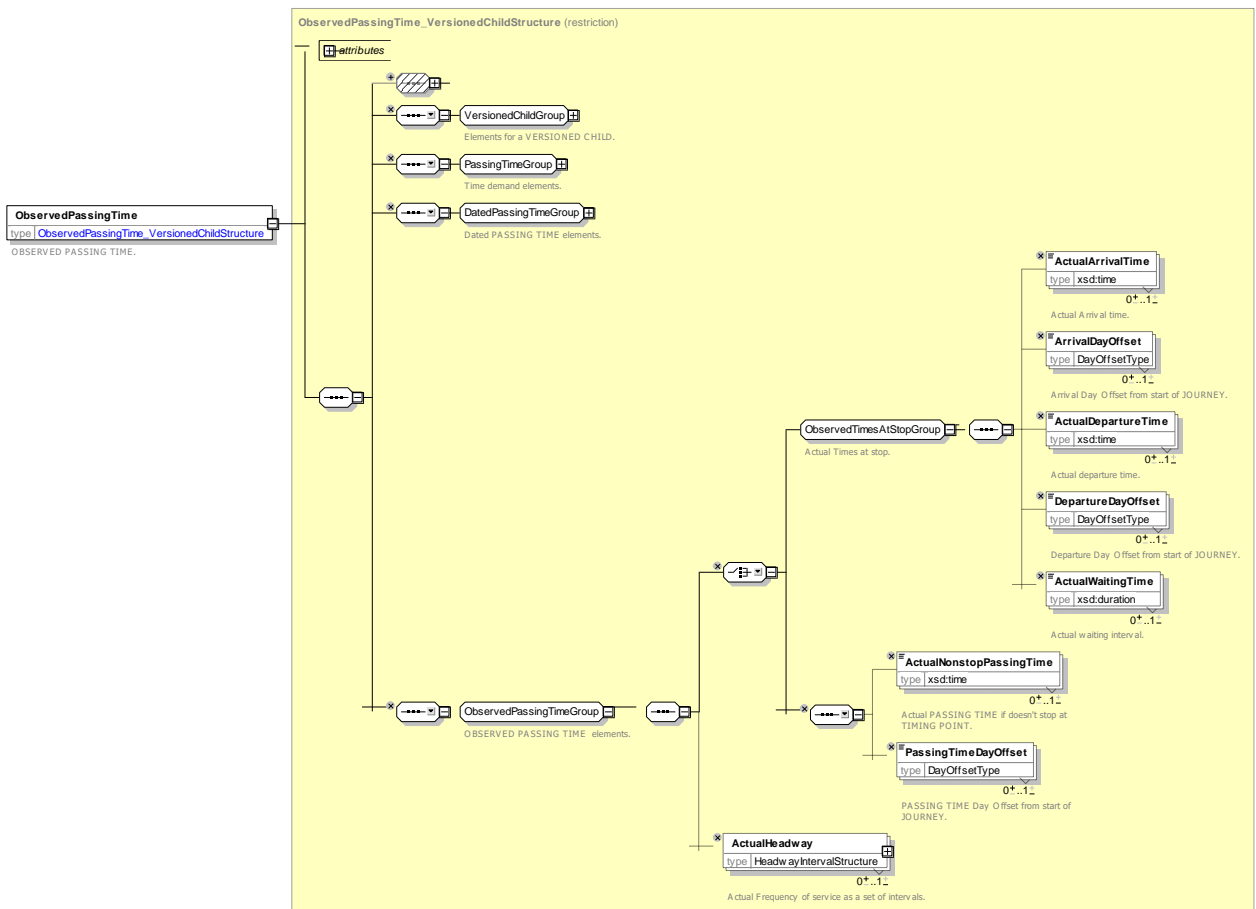


Figure 175 — ObservedPassingTime — XSD

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 102 – TargetPassingTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DatedPassingTime</i>	::>	TARGET PASSING TIME inherits from DATED PASSING TIME.

«PK»	<i>id</i>	<i>AimedPassing-TimeIdType</i>	0:1	Identifier of AIMED PASSING TIME.
	<i>AimedArrivalTime</i>	<i>xsd:time</i>	0:1	Intended arrival time at TIMING POINT IN JOURNEY PATTERN.
	<i>ArrivalDayOffset</i>	<i>DayOffsetType</i>	0:1	Arrival day offset from start of JOURNEY. +V1.1
	<i>AimedDepartureTime</i>	<i>xsd:time</i>	0:1	Intended departure time at TIMING POINT IN JOURNEY PATTERN.
	<i>DepartureDayOffset</i>	<i>DayOffsetType</i>	0:1	Departure day offset from start of JOURNEY. +V1.1
	<i>AimedWaitingTime</i>	<i>xsd:duration</i>	0:1	Aimed waiting time at TIMING POINT IN JOURNEY PATTERN.
	<i>AimedNonstopPassing-Time</i>	<i>xsd:time</i>	0:1	Intended passing time at TIMING POINT IN JOURNEY PATTERN if vehicle does not stop.
	<i>PassingTimeDayOffset</i>	<i>DayOffsetType</i>	0:1	Passing time day offset from start of JOURNEY. +V1.1
	<i>AimedHeadway</i>	<i>HeadwayInterval</i>	0:1	Aimed headway interval at POINT IN JOURNEY PATTERN.

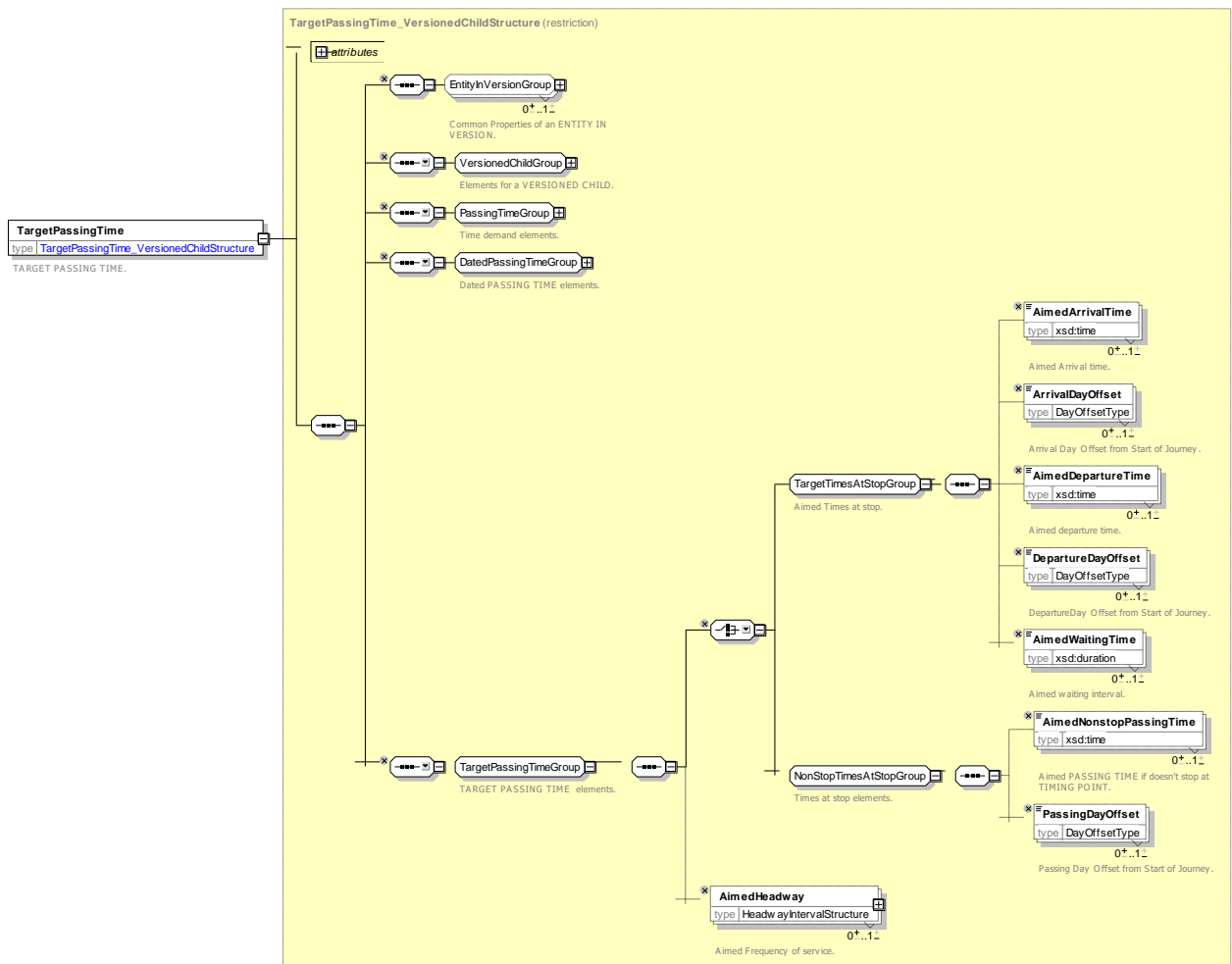


Figure 176 — TargetPassingTime — XSD

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.

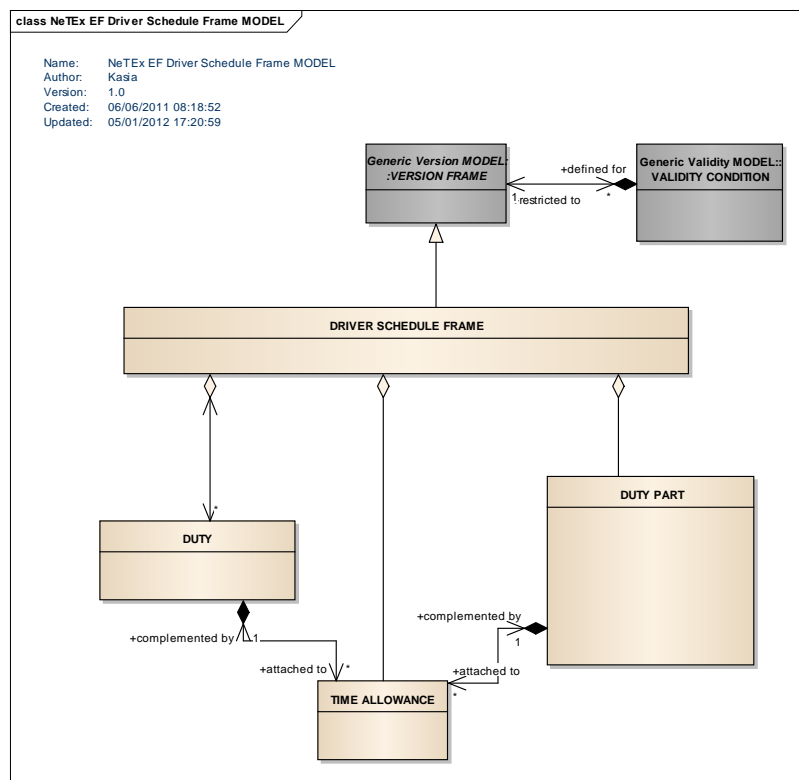


Figure 177 — Driver Schedule Frame – Conceptual MODEL (UML)

B.2.1.2 Driver Schedule Frame – Physical Model

The following diagram shows the Physical model for a DRIVER SCHEDULE FRAME.

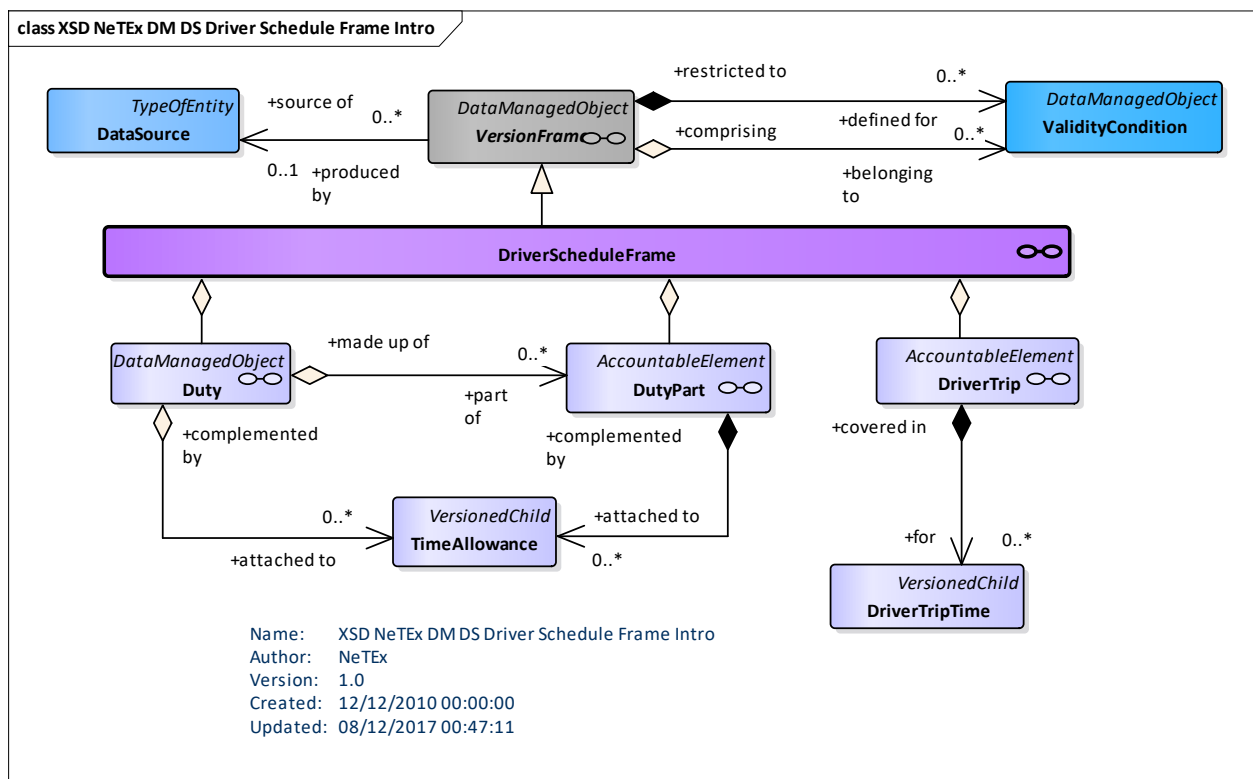


Figure 178 — Driver Schedule Frame Contents – Physical Model (UML)

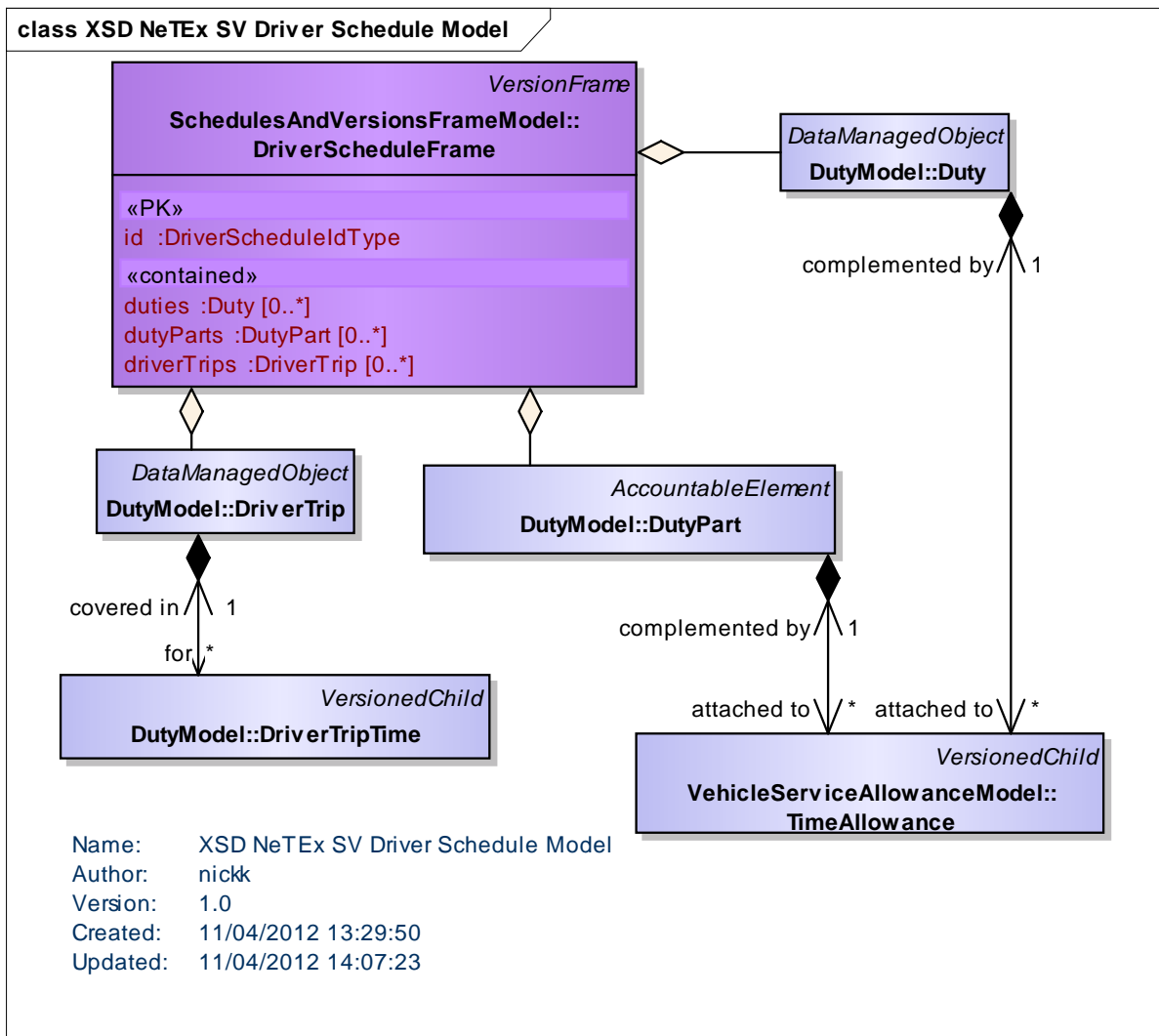


Figure 179 — Driver Schedule Frame – Physical Model (UML)

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 103 – 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.

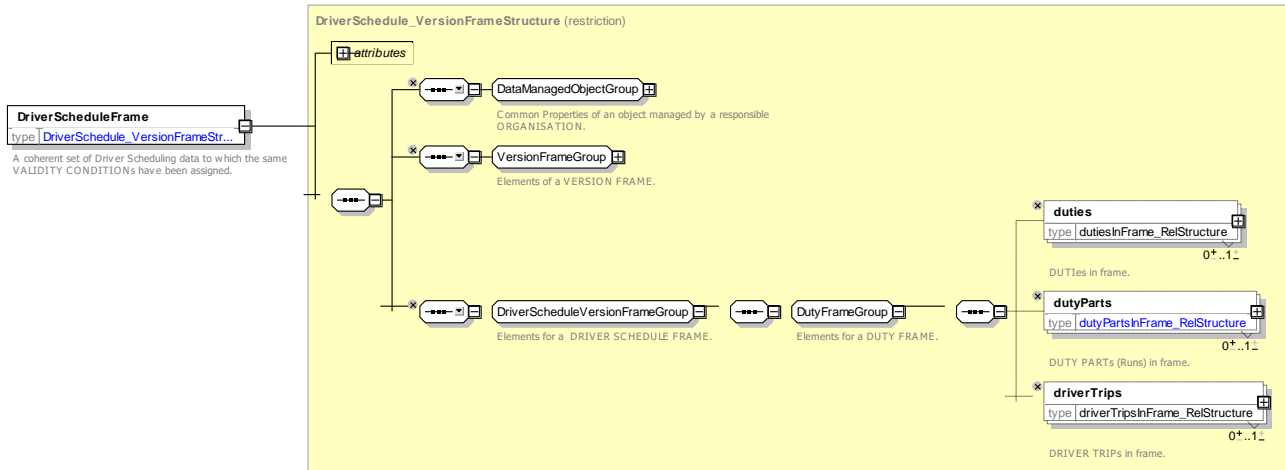


Figure 180 — DriverScheduleFrame — XSD

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.

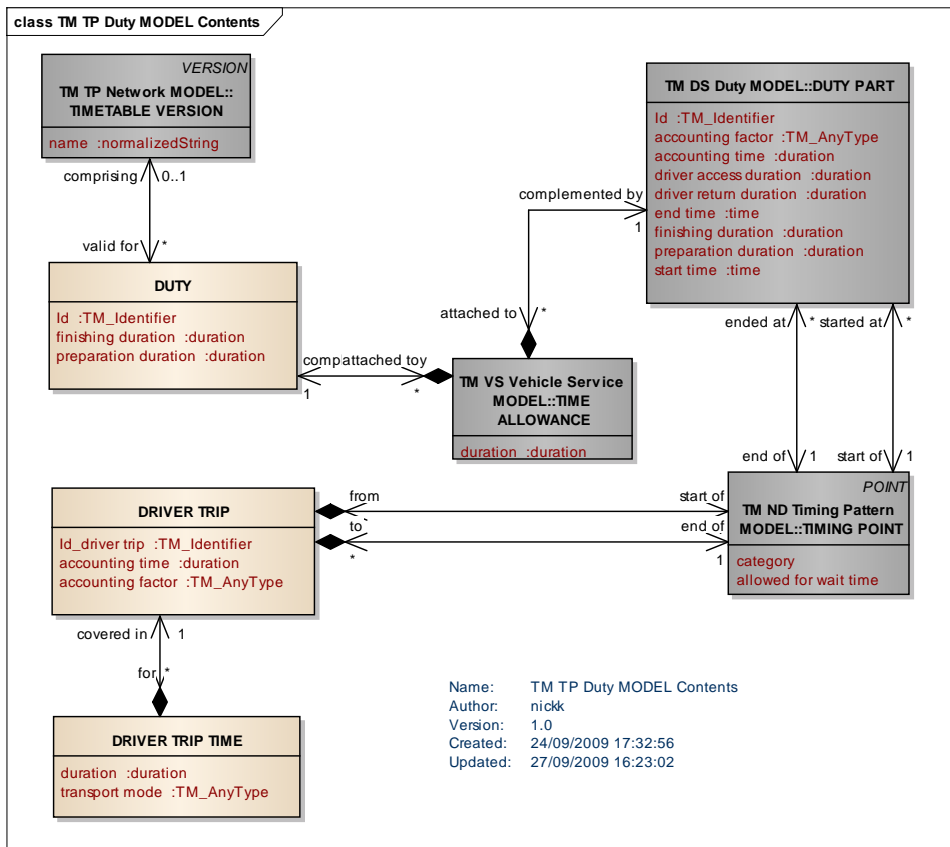


Figure-181— DUTY — Conceptual MODEL (UML)

B.2.2.1.1 Duty – Physical Model

The following figure shows the DUTY Physical Model.

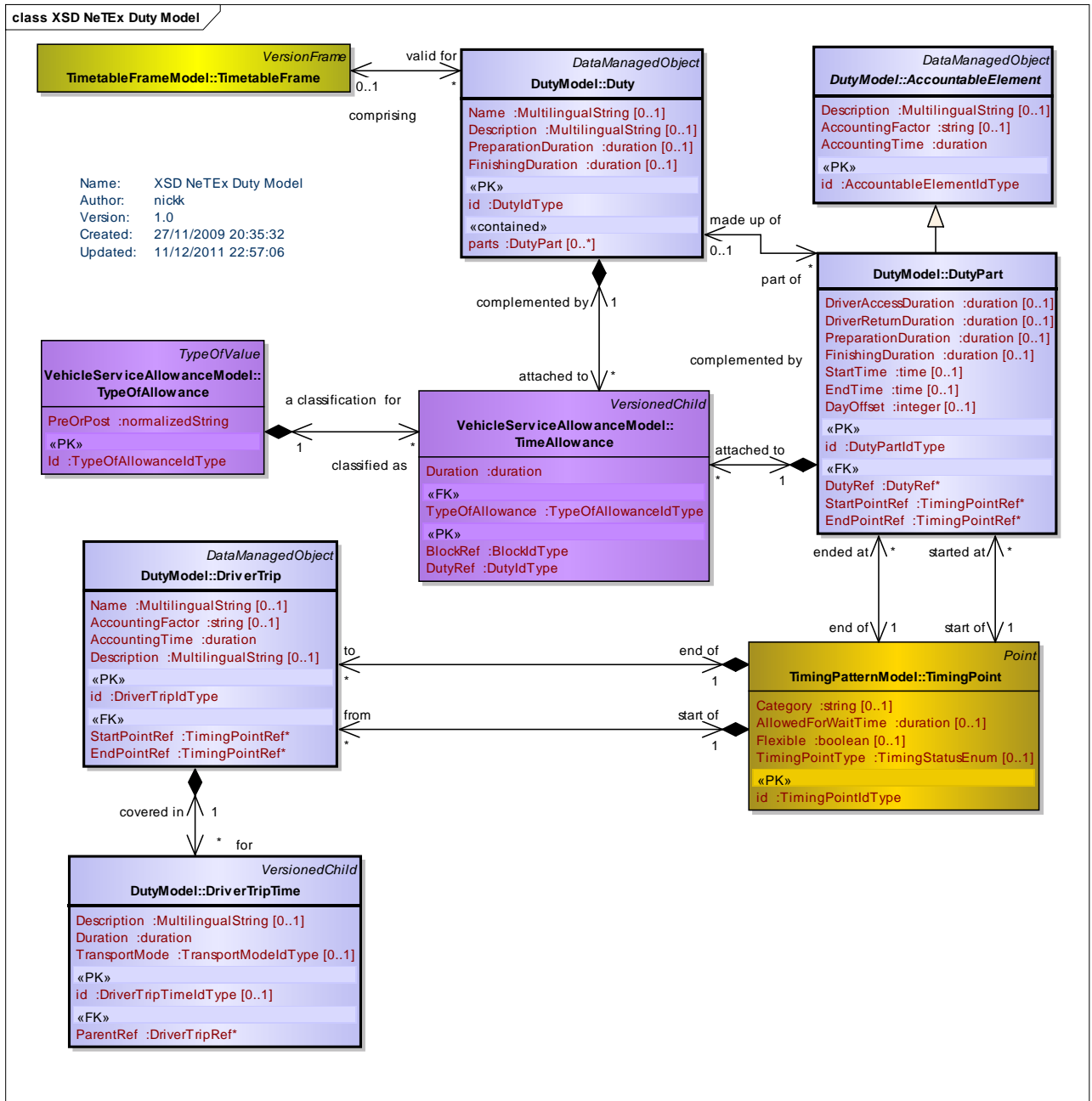


Figure 182 — Duty – Physical Model (UML)

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 104 – Duty – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	DUTY inherits from DATA MANAGED OBJECT.
«PK»	id	<i>DutyIdType</i>	1:1	Identifier of DUTY.
	Name	<i>MultilingualString</i>	0:1	Name of DUTY.
	Description	<i>MultilingualString</i>	0:1	Description of DUTY.
	PreparationDuration	<i>xsd:duration</i>	0:1	Time needed to prepare for DUTY.
	FinishingDuration	<i>xsd:duration</i>	0:1	Time needed to complete DUTY at end.
«cntd»	parts	<i>DutyPart</i>	0:*	List of DUTY Parts.

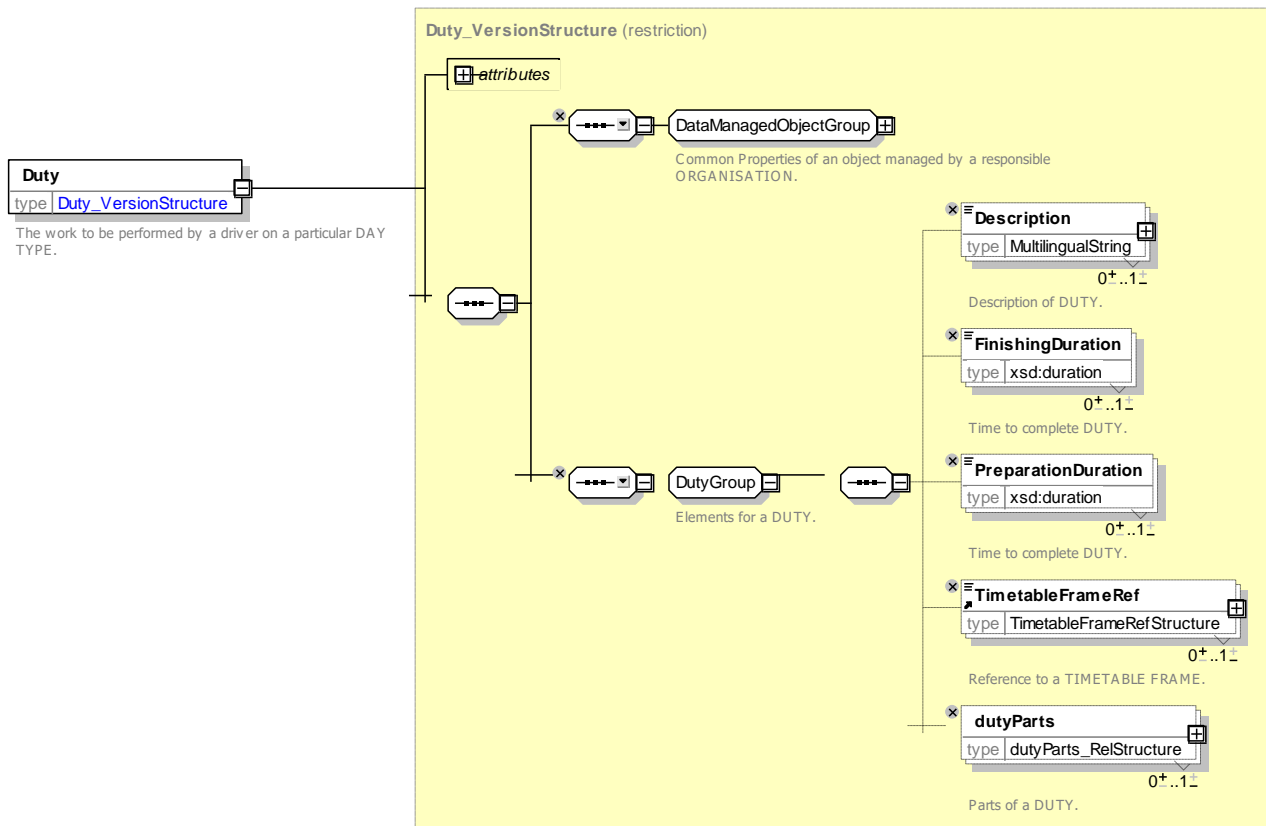


Figure 183 – Duty — XSD

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 105 – AccountableElement – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	ACCOUNTABLE ELEMENT inherits from DATA MANAGED OBJECT.
«PK»	<i>id</i>	<i>AccountableElementIdType</i>	1:1	Identifier of ACCOUNTABLE ELEMENT.
	Description	<i>MultilingualString</i>	0:1	Description of ACCOUNTING ELEMENT.
	AccountingFactor	<i>xsd:normalizedString</i>	0:1	Accounting code to use on ACCOUNTABLE ELEMENT.
	AccountingTime	<i>xsd:duration</i>	1:1	Accountable time to use on ACCOUNTABLE ELEMENT.

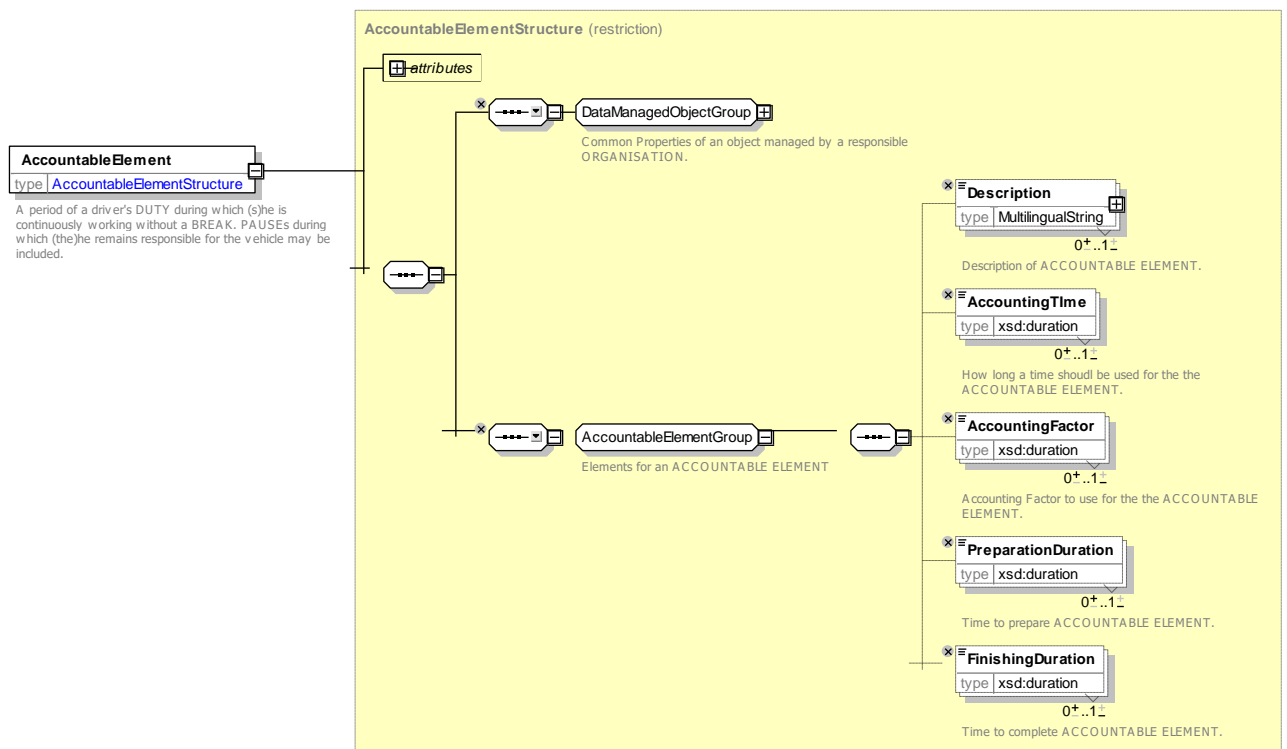


Figure 184 — AccountableElement — XSD

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 106 – *DutyPart* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	DUTY PART inherits from DATA MANAGED OBJECT.
«PK»	id	<i>DutyPartIdType</i>	1:1	Identifier of DUTY PART.
«FK»	DutyRef	<i>DutyRef</i>	1:1	Reference to of DUTY of which this is part.
	DriverAccessDuration	<i>xsd:duration</i>	0:1	Time needed for driver access to DUTY PART.
	DriverReturnDuration	<i>xsd:duration</i>	0:1	Time needed for driver return from DUTY PART.
	PreparationDuration	<i>xsd:duration</i>	0:1	Time needed to prepare for DUTY PART.
	FinishingDuration	<i>xsd:duration</i>	0:1	Time needed to complete DUTY PART.
	StartTime	<i>xsd:time</i>	0:1	Start time of DUTY PART.
«FK»	StartPointRef	<i>TimingPointRef</i>	1:1	Start POINT of DUTY PART.
	EndTime	<i>xsd:time</i>	0:1	End time of DUTY PART.
«FK»	EndPointRef	<i>TimingPointRef</i>	1:1	End POINT of DUTY PART.
	DayOffset	<i>xsd:integer</i>	0:1	Day offset of end Time of DUTY PART from atart of JOURNEY.

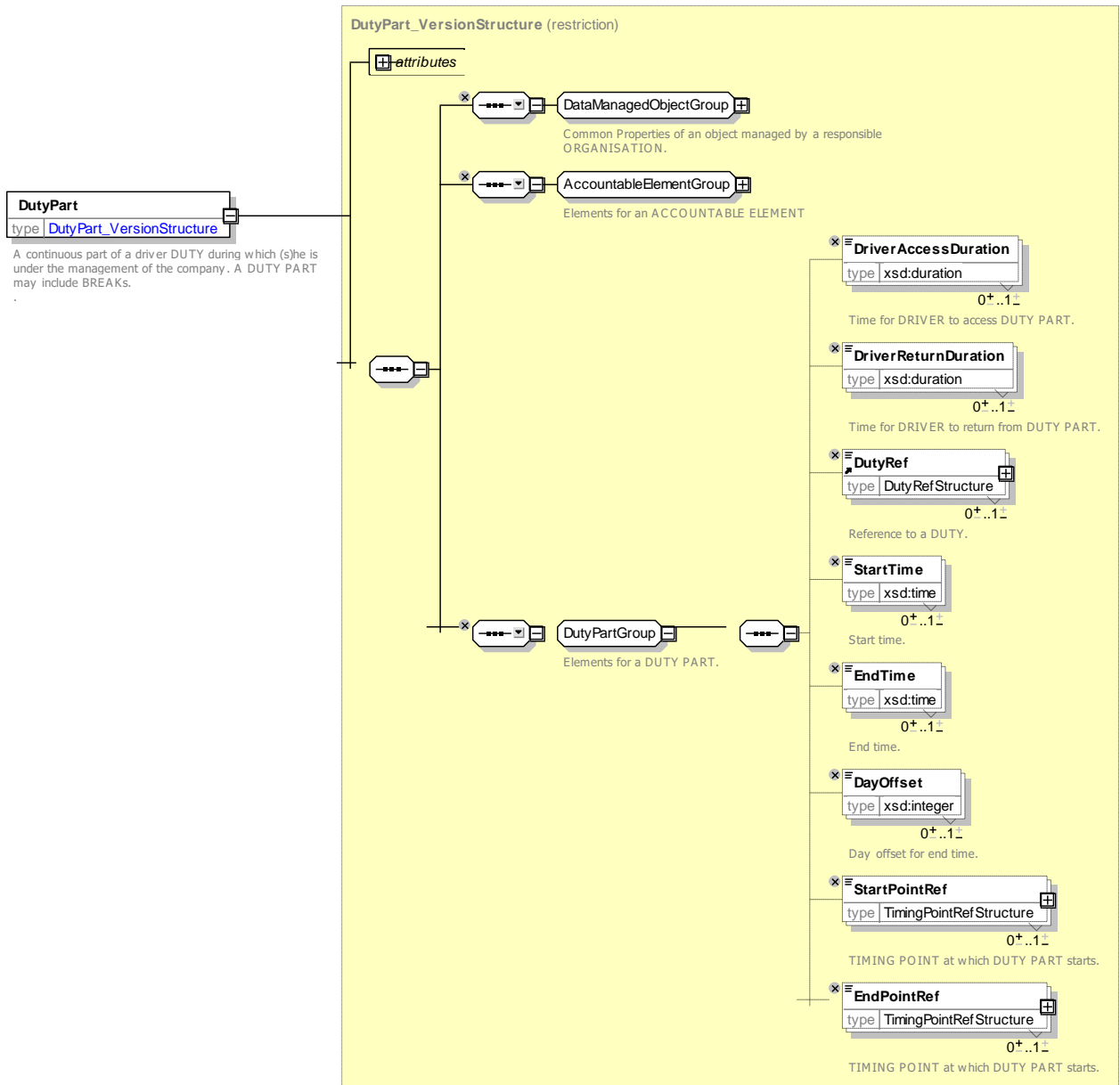


Figure 185 — *DutyPart* — XSD

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 107 – *DriverTrip* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	DRIVER TRIP inherits from DATA MANAGED OBJECT.
«PK»	id	<i>DriverTripIdType</i>	1:1	Identifier of DRIVER TRIP.
	Name	<i>MultilingualString</i>	0:1	Name of DRIVER TRIP.

	AccountingFactor	<i>xsd:string</i>	0:1	Accounting code to use for DRIVER TRIP.
	AccountingTime	<i>xsd:duration</i>	1:1	Accountable time to use for DRIVER TRIP.
	Description	<i>MultilingualString</i>	0:1	Description of DRIVER TRIP.
«FK»	StartPointRef	<i>TimingPointRef</i>	1:1	Start POINT of DRIVER TRIP.
«FK»	EndPointRef	<i>TimingPointRef</i>	1:1	End POINT of DRIVER TRIP.

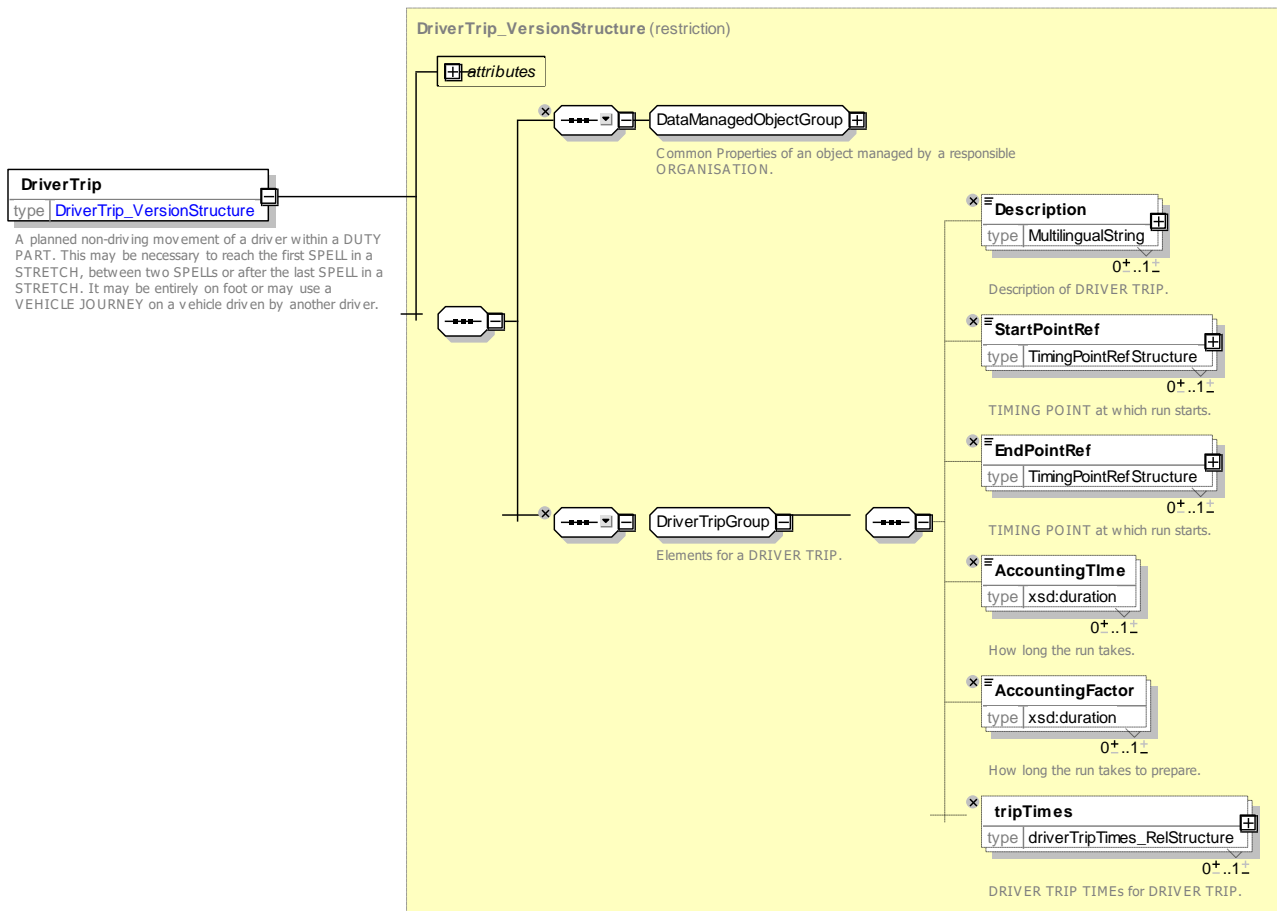


Figure 186 — DriverTrip — XSD

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 108 – DriverTripTime – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>DataManagedObject</i>	::>	DRIVER TRIP TIME inherits from DATA MANAGED OBJECT.
«PK»	id	<i>DriverTripTimeIdType</i>	0:1	Identifier of DRIVER TRIP TIME.

«FK»	DriverTripRef	<i>DriverTripRef</i>	1:1	DRIVER TRIP to which this DRIVER TRIP TIME belongs.
	Description	<i>MultilingualString</i>	0:1	Description of DRIVER TRIP TIME.
	Duration	<i>xsd:duration</i>	1:1	Duration of DRIVER TRIP TIME.
	TransportMode	<i>TransportModeEnum</i>	0:1	Transport mode of DRIVER TRIP TIME.

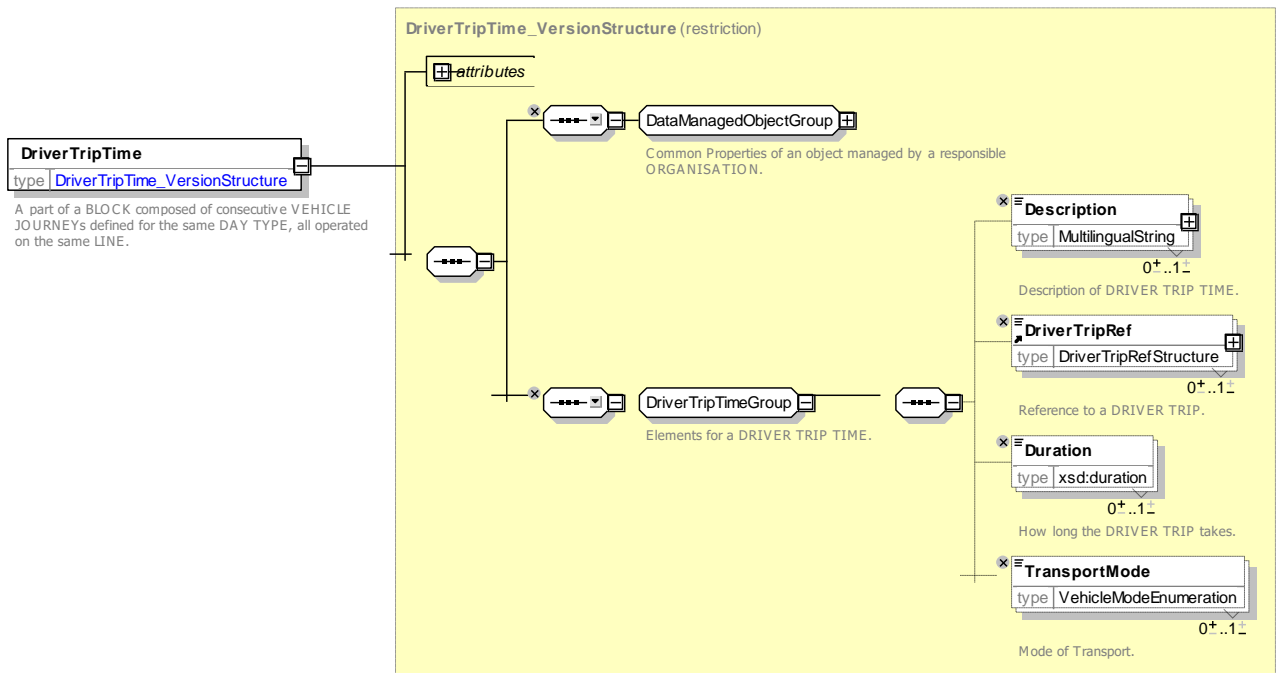


Figure 187 — *DriverTripTime* — XSD

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.

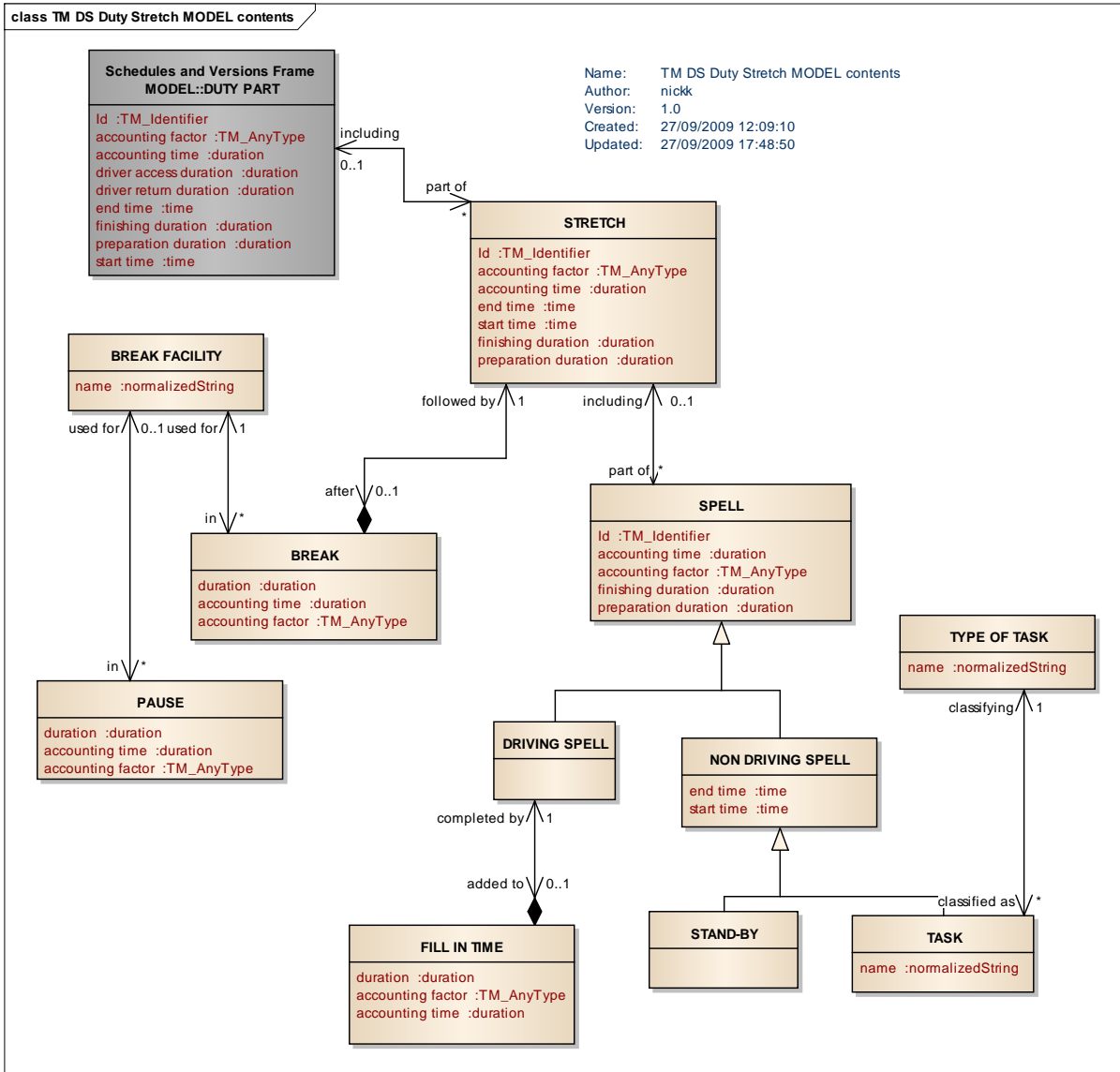


Figure 188 — STRETCH and SPELL— Conceptual MODEL (UML)

B.2.3.2 Duty Stretch – Physical Model

The following figure shows the DUTY STRETCH Physical Model.

Annex C (informative)

Changes in NeTeX Version 1.1

C.1 Introduction

This annex summarises changes made to NeTeX 1.1 over NeTeX 1.0. Note that a number of more general changes are also made to the core framework as described in Part1..

C.2 General Changes

Diagram & Package labels have been modified to use the same system of two letter acronyms used in Transmodel 6.0

Colours of some elements have been altered to lighter tones that are easier to read and also further aligned with Transmodel 6.0

Certain role names have been revised to align closer with Transmodel 6.0 conventions.

Presentation conventions (use of aggregation (black and white diamonds) and navigability arrows) have been revised to align closer with the revised conventions used in Transmodel 6.0,

Organisation of certain packages (especially explicit frames) has been revised to align closer with revised organisation in Transmodel 6.0

C.3 List of changes

CR0045 Use Generic **Assignment**: **JourneyAccounting**, JourneyTimes VehicleTypeSTopAssignment all inherit from **Assignment**

CR0008 **InterchangeRule** Allow multiple LINES and DIRECTIONS on filters.

CR0050 **GroupOfEntities** Allow infoLinks on

CR0010 **JourneyPart** - Add order. and **JourneyPartPosition**.

CR0023 **Call** revise **BookingArrangements**, allow requestMethod etc on

CR0028 **Journey** add **AccessibilityAssessment** (Fix) , Also add default **AccessibilityAssessment** and **vehicle type** to Timetable frame

CR0030 & CR0019 **ServiceJourney/ServiceTimesGroup VehicleJourney, SpecialService**, Service Times: add **DayOffset**., **JourneyPartCouple**, **TargetPassingTime**, **EstimatedPassingTime**, **Observed-PassingTime ReliefOpportunity**, **CourseOfJourney**, **JourneyPartCouple** add DayOffset for *time*.

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>.