



BSI Standards Publication

Public transport — Network and Timetable Exchange (NeTEx)

Part 2: Public transport scheduled
timetables exchange format

National foreword

This Published Document is the UK implementation of CEN/TS 16614-2:2014.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2014.
Published by BSI Standards Limited 2014

ISBN 978 0 580 82692 4
ICS 35.240.60

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 30 June 2014.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

TECHNICAL SPECIFICATION
SPÉCIFICATION TECHNIQUE
TECHNISCHE SPEZIFIKATION

CEN/TS 16614-2

May 2014

ICS 35.240.60

English Version

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

Transport Public - Échanges des informations planifiées
(NeTEx) - Partie 2: Description de l'offre de transport

Öffentlicher Verkehr - Netzwerk und Fahrplan Austausch
(NeTEx) - Teil 2: Fahrpläne

This Technical Specification (CEN/TS) was approved by CEN on 12 November 2013 for provisional application.

The period of validity of this CEN/TS is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the CEN/TS can be converted into a European Standard.

CEN members are required to announce the existence of this CEN/TS in the same way as for an EN and to make the CEN/TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the CEN/TS) until the final decision about the possible conversion of the CEN/TS into an EN is reached.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Page

Foreword	4
Introduction	5
1 Scope.....	6
1.1 General	6
1.2 Transport modes	6
1.3 Compatibility with existing standards and recommendations.....	6
2 Normative references	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.....	33
7.2.3 Time Demand Times	49
7.2.4 Journey Timing.....	58
7.2.5 Journey Pattern Times	66
7.2.6 Vehicle Journey Times	74
7.2.7 Interchange	92
7.2.8 Interchange Rule	110
7.2.9 Coupled Journey	118
7.2.10 Flexible Service	137
7.2.11 Journey Accounting	145
7.2.12 Dated Journey	150
7.2.13 Passing Times	155
7.2.14 Call	162
7.2.15 Dated Call.....	178
8 Vehicle Scheduling	180
8.1 Vehicle Scheduling – Model dependencies.....	180
8.2 Vehicle Scheduling	181
8.2.1 Vehicle Schedule Frame	181
8.2.2 Vehicle Service	184
8.2.3 Train Service.....	199
Annex A (informative) Monitoring & Control.....	203
A.1 Introduction	203
A.2 Monitoring & Control	203
A.2.1 Monitored Vehicle Journey	203
A.2.1.1 Monitored Vehicle Journey – Conceptual MODEL	203
A.2.1.1.1 Monitored Passing Times	204
A.2.1.2 Monitored Vehicle Journey – Physical Model	205
A.2.1.3 Monitored JourneyModel – Attributes and XSD	207

A.2.1.3.1	MonitoredVehicleJourney – Model Element	207
A.2.1.3.2	MonitoredSpecialService – Model Element	207
A.2.1.3.3	MonitoredCall – Model Element	207
A.2.1.3.4	PreviousCall – Model Element.....	208
A.2.1.3.5	OnwardCall – Model Element	209
A.2.2	Dated Passing Times – Physical Model	210
A.2.2.1	Passing times – Attributes and XSD	211
A.2.2.1.1	EstimatedPassingTime – Model Element.....	211
A.2.2.1.2	ObservedPassingTime – Model Element.....	212
A.2.2.1.3	TargetPassingTime – Model Element	213
Annex B (informative)	Driver Scheduling	215
B.1	Introduction.....	215
B.2	Driver Scheduling.....	215
B.2.1	Driver Schedule Frame	215
B.2.1.1	DRIVER SCHEDULE FRAME – Conceptual MODEL	215
B.2.1.2	Driver Schedule Frame – Physical Model	216
B.2.1.3	Driver Schedule Frame — XSD and attributes	218
B.2.1.3.1	DriverScheduleFrame – Model Element	218
B.2.2	Duty.....	219
B.2.2.1	DUTY – Conceptual MODEL	219
B.2.2.1.1	Duty – Physical Model	220
B.2.2.1.2	Duty – Attributes and XSD	221
B.2.2.1.2.1	Duty – Model Element	221
B.2.2.1.2.2	AccountableElement – Model Element	222
B.2.2.1.2.3	DutyPart – Model Element	223
B.2.2.1.2.4	DriverTrip – Model Element.....	225
B.2.2.1.2.5	DriverTripTime – Model Element	226
B.2.3	Duty Stretch	227
B.2.3.1	DUTY STRETCH – Conceptual MODEL	227
B.2.3.2	Duty Stretch – Physical Model	228
Bibliography		230

Foreword

This document (CEN/TS 16614-2:2014) has been prepared by Technical Committee CEN/TC 278 “Intelligent transport systems”, the secretariat of which is held by NEN.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

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

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

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

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

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

This document is 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).

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this Technical Specification: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

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

1.1 General

NeTEx is dedicated to the exchange of scheduled data (network, timetable and fare information) based on Transmodel V5.1 (EN 12896), IFOPT (EN 28701) and SIRI (CEN/TS 15531-4 and CEN/TS 15531-5 and prEN 15531-1, prEN 15531-2 and prEN 15531-3) 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.

1.2 Transport modes

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

1.3 Compatibility with existing standards and recommendations

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

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

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

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

2 Normative references

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

prEN 15531-1:2013, *Public transport - Service interface for real-time information relating to public transport operations - Part 1: Context and framework*

prEN 15531-2:2013, *Public transport - Service interface for real-time information relating to public transport operations - Part 2: Communications infrastructure*

prEN 15531-3:2013, *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*

CEN/TS 16614-1:2014, *Public transport - Network and Timetable Exchange (NeTEx) - Part 1: Public transport network topology exchange format*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in CEN/TS 16614-1:2014 apply.

4 Symbols and abbreviations

For the purposes of this document, the symbols and abbreviations given in CEN/TS 16614-1:2014 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

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

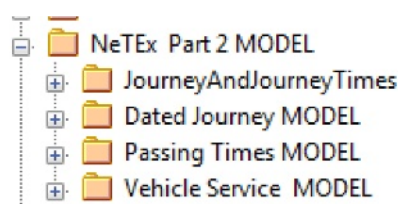


Figure 1 – NeTEx Part 2 main packages

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

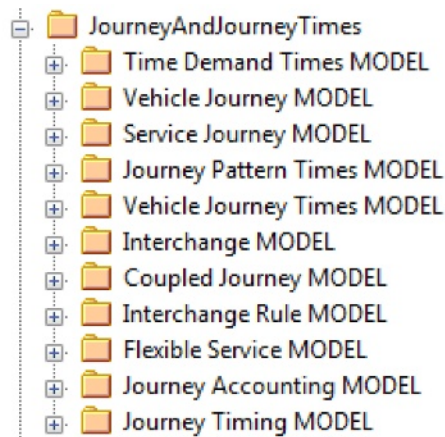


Figure 2 – JourneyAndJourneyTimes packages

- 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 informations related to vehicles and their services

7.1.1 Journey and Journey Times – Model dependencies

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

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

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

TIMETABLE FRAME

- VEHICLE JOURNEY: models journeys that vehicles make.
- SERVICE JOURNEY: additionally models the properties of journeys that carry passengers.
- TIME DEMAND TIMES: models the times of the different demand levels found during a day.
- PASSING TIMES: describes the times of vehicles at points in their journey.
- JOURNEY TIMINGS: describes the common timing properties for journeys.
- JOURNEY PATTERN TIMES: describes the timings of JOURNEY PATTERNS.
- VEHICLE JOURNEY TIMES: describes the timings of VEHICLE JOURNEYS.
- INTERCHANGE: describes interchanges between journeys.

- COUPLED JOURNEY: describes multipart journeys which join and split.
- FLEXIBLE SERVICE: additional describes demand responsive transport services.
- JOURNEY ACCOUNTING: assigns a cost basis for journeys.

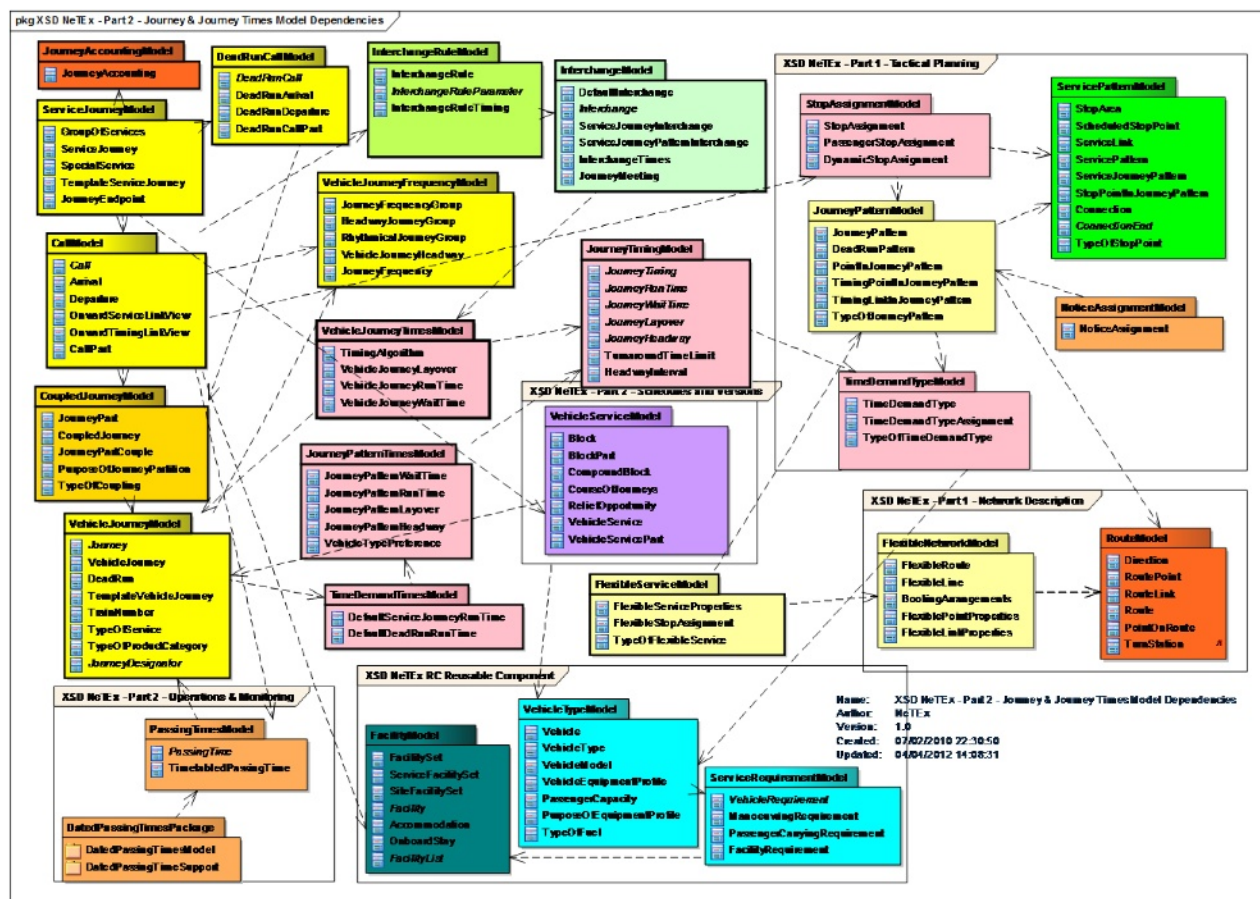


Figure 3 — Journey – Model Dependencies

7.1.1.1 Timetable Frame

7.1.1.1.1 TIMETABLE FRAME – Conceptual MODEL

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

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

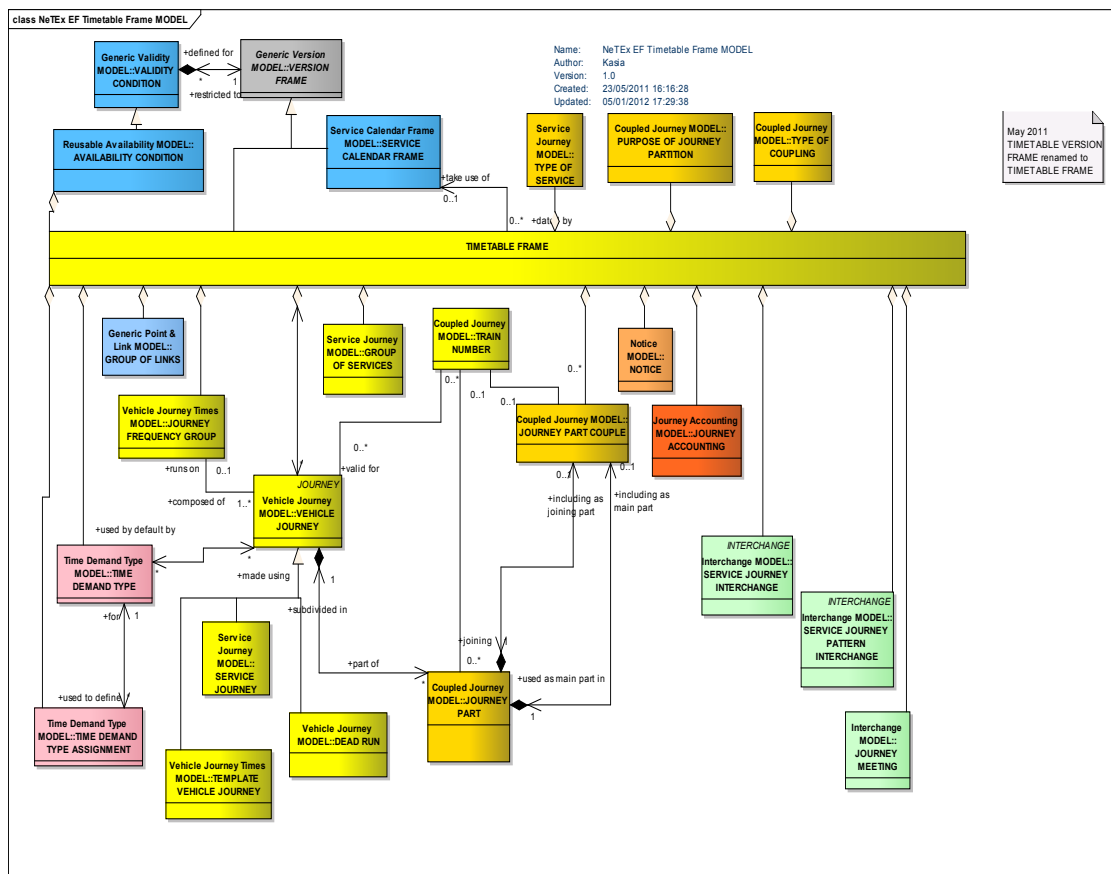


Figure 4 — Timetable Frame – Conceptual MODEL (UML)

7.1.1.1.2 Timetable Frame – Physical Model

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

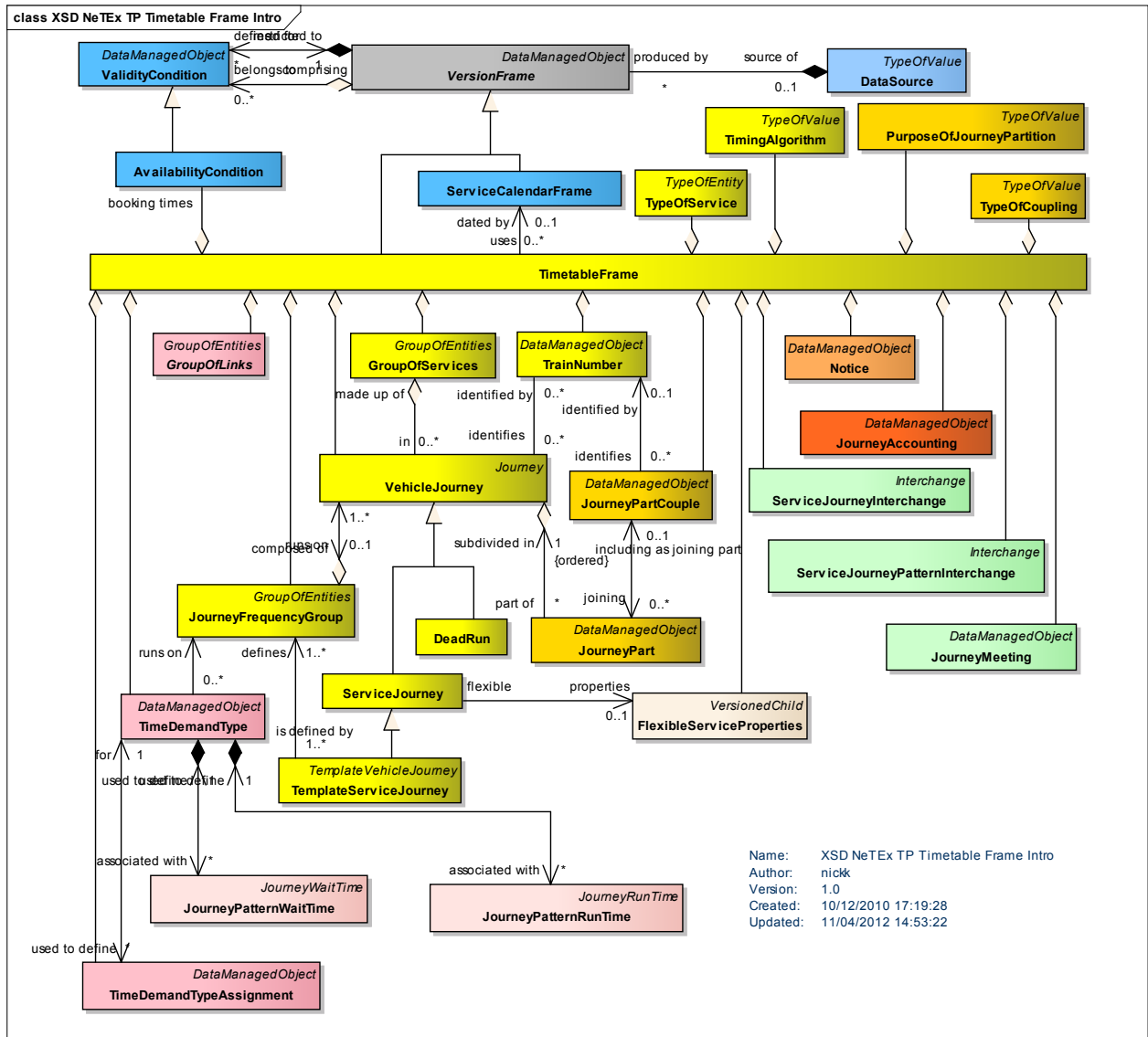


Figure 5 — Timetable Frame Contents – Physical Model (UML)

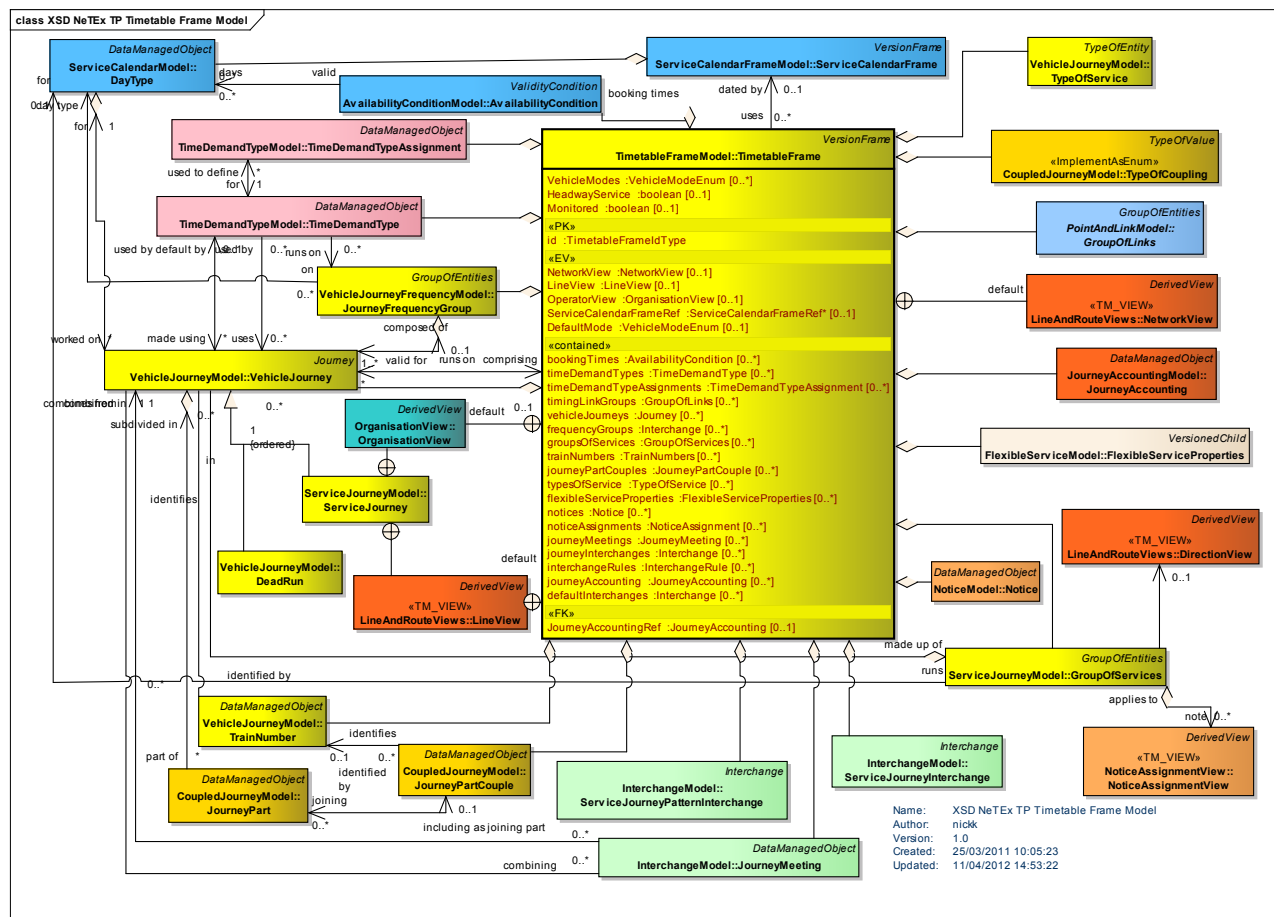


Figure 6 — Timetable Frame – Physical Model Detail (UML)

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:xsd:boolean</i>	0:1	Whether services of TIMETABLE are operated a headway services.
	Monitored	<i>xsd: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>ServiceCalendarFrameRef</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.
«cntd»	timeDemandTypes	<i>TimeDemandType</i>	0:*	TIME DEMAND TYPEs in the frame.
«cntd»	timeDemandTypeAssignments	<i>TimeDemandTypeAssignment</i>	0:*	TIME DEMAND TYPE ASSIGNMENTS in the frame.
«cntd»	timingLinkGroups	<i>GroupOfLinks</i>	0:*	TIMING LINK GROUPs in the frame.
«cntd»	vehicleJourneys	<i>Journey</i>	0:*	VEHICLE JOURNEYS & SERVICE JOURNEYS in the frame.
«cntd»	frequencyGroups	<i>FrequencyGroup</i>	0:*	FREQUENCY GROUPs in the frame.
«cntd»	groupsOfServices	<i>GroupOfServices</i>	0:*	GROUP OF SERVICEs in the frame.
«cntd»	journeyPartCouples	<i>JourneyPartCouple</i>	0:*	JOURNEY PART COUPLES in the frame.
«cntd»	coupledJourneys	<i>CoupledJourney</i>	0:*	COUPLED JOURNEYS in the frame.
«cntd»	serviceFacilitySets	<i>ServiceFacilitySet</i>	0:*	SERVICE FACILITY SETs in the frame.
«cntd»	typesOfService	<i>TypeOfService</i>	0:*	TYPEs OF SERVICE in the frame.
«cntd»	flexibleServiceProperties	<i>FlexibleServiceProperties</i>	0:*	FLEXIBLE SERVICE PROPERTIES in the frame.
«cntd»	notices	<i>Notice</i>	0:*	NOTICEs in the frame.
«cntd»	noticeAssignments	<i>NoticeAssignment</i>	0:*	NOTICE ASSIGNMENTs in the frame.
«cntd»	journeyMeetings	<i>JourneyMeeting</i>	0:*	JOURNEY MEETINGs in the frame.
«cntd»	journeyInterchanges	<i>Interchange</i>	0:*	JOURNEY FREQUENCY GROUPs in the frame.
«cntd»	interchangeRules	<i>InterchangeRule</i>	0:*	INTERCHANGE RULEs in the frame.
«cntd»	journeyAccountings	<i>JourneyAccounting</i>	0:*	Default JOURNEY ACCOUNTING values for JOURNEYS in frame.

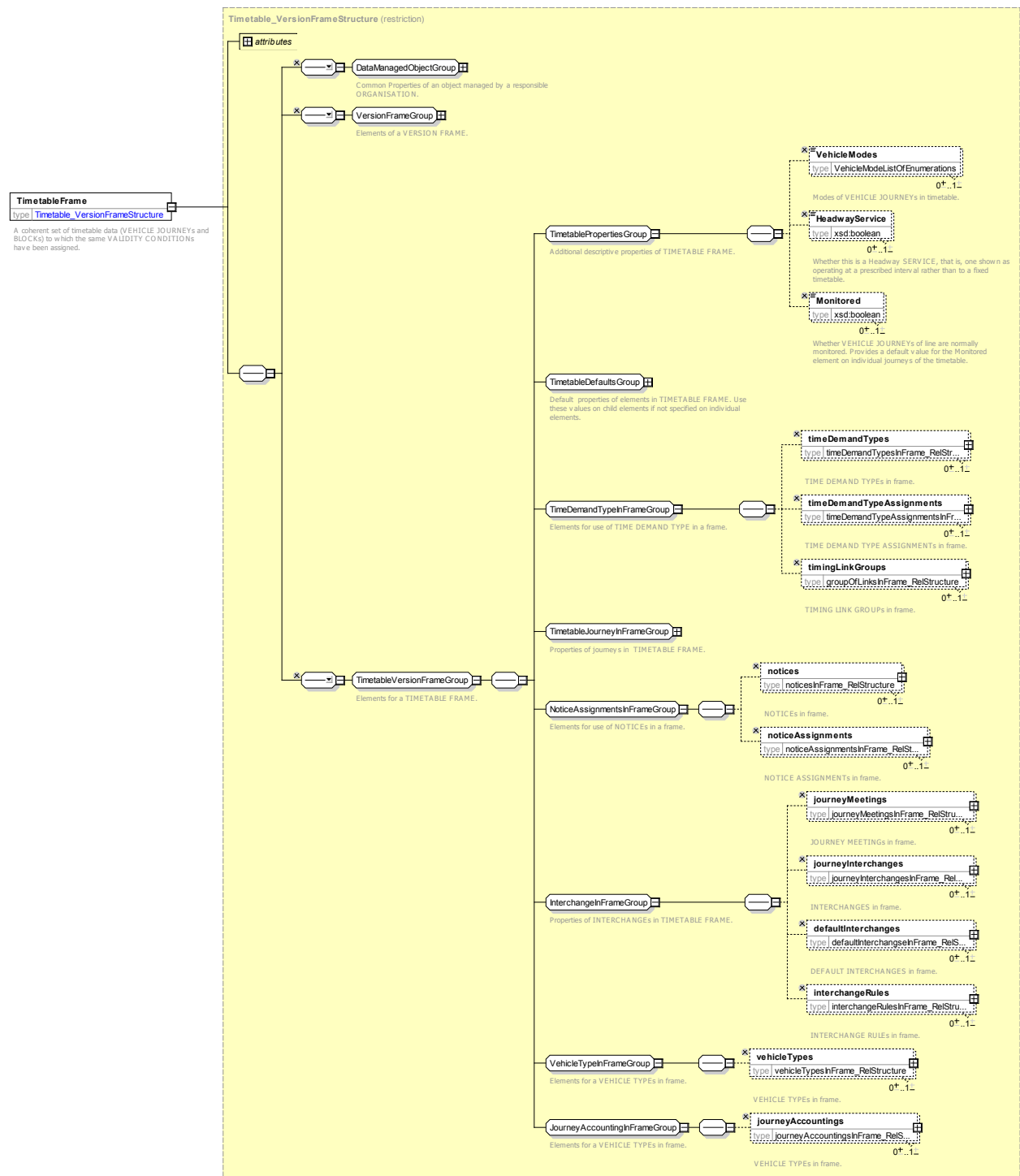


Figure 7 — TimetableFrame — XSD

TimetableFrameDefaultsGroup - Group

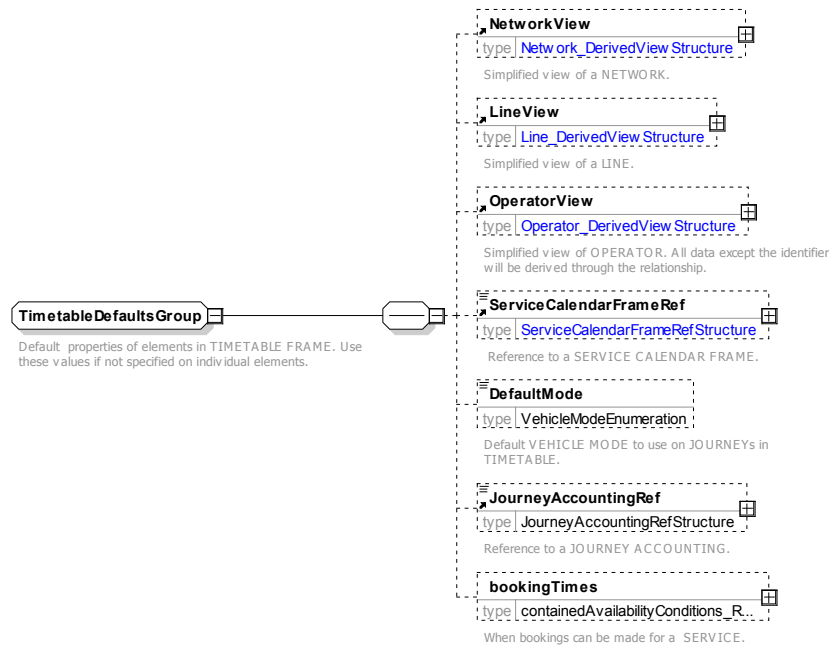


Figure 8 — TimetableFrameDefaultsGroup — XSD

TimetableJourneyInFrameGroup - Group

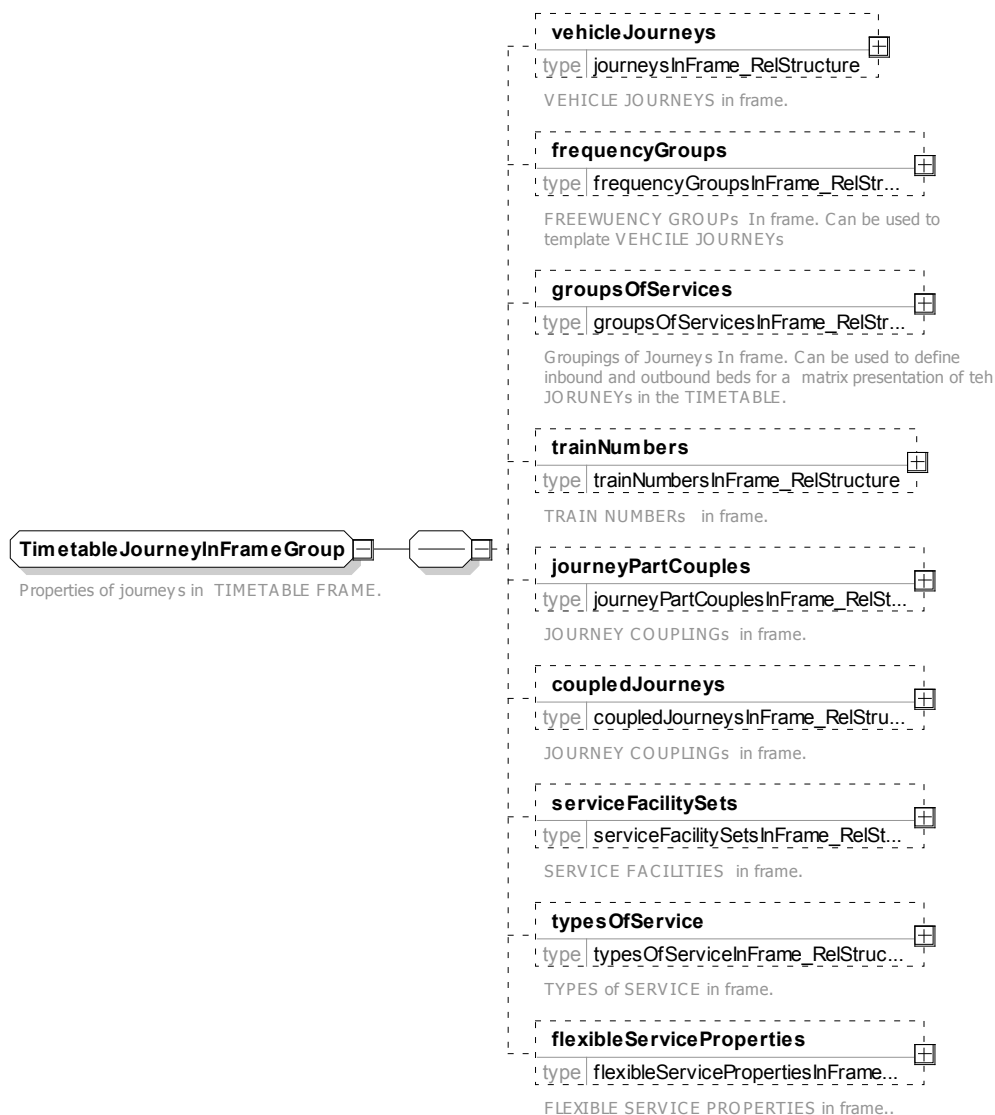


Figure 9 — TimetableJourneyInFrameGroup — XSD

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.

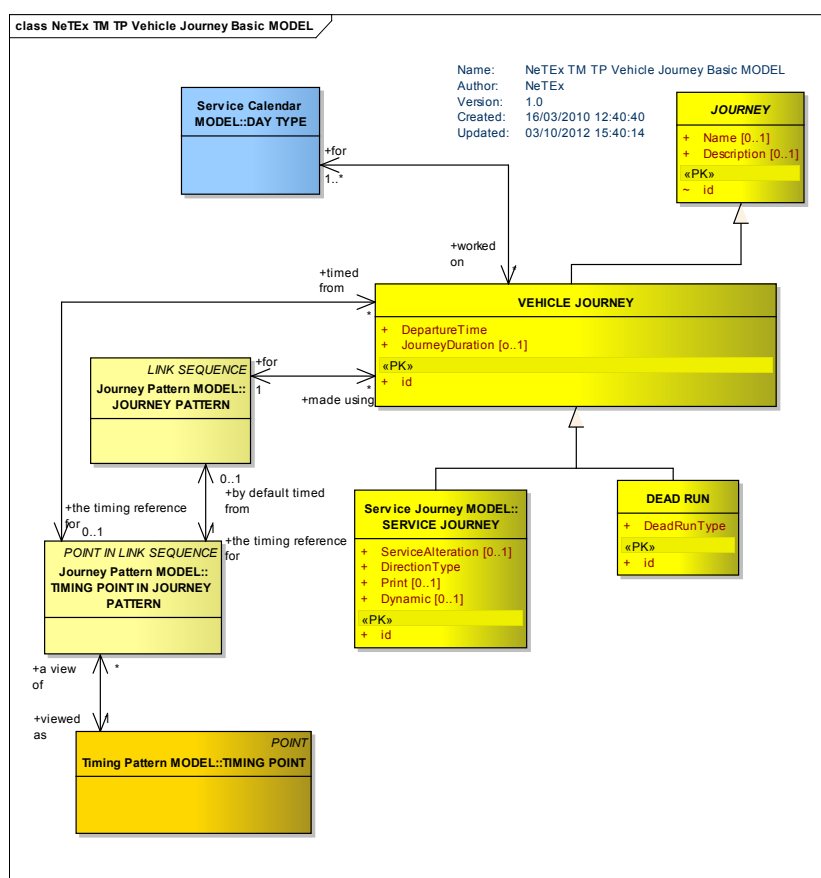


Figure 10 — Basic Vehicle Journey – Conceptual MODEL (UML)

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.

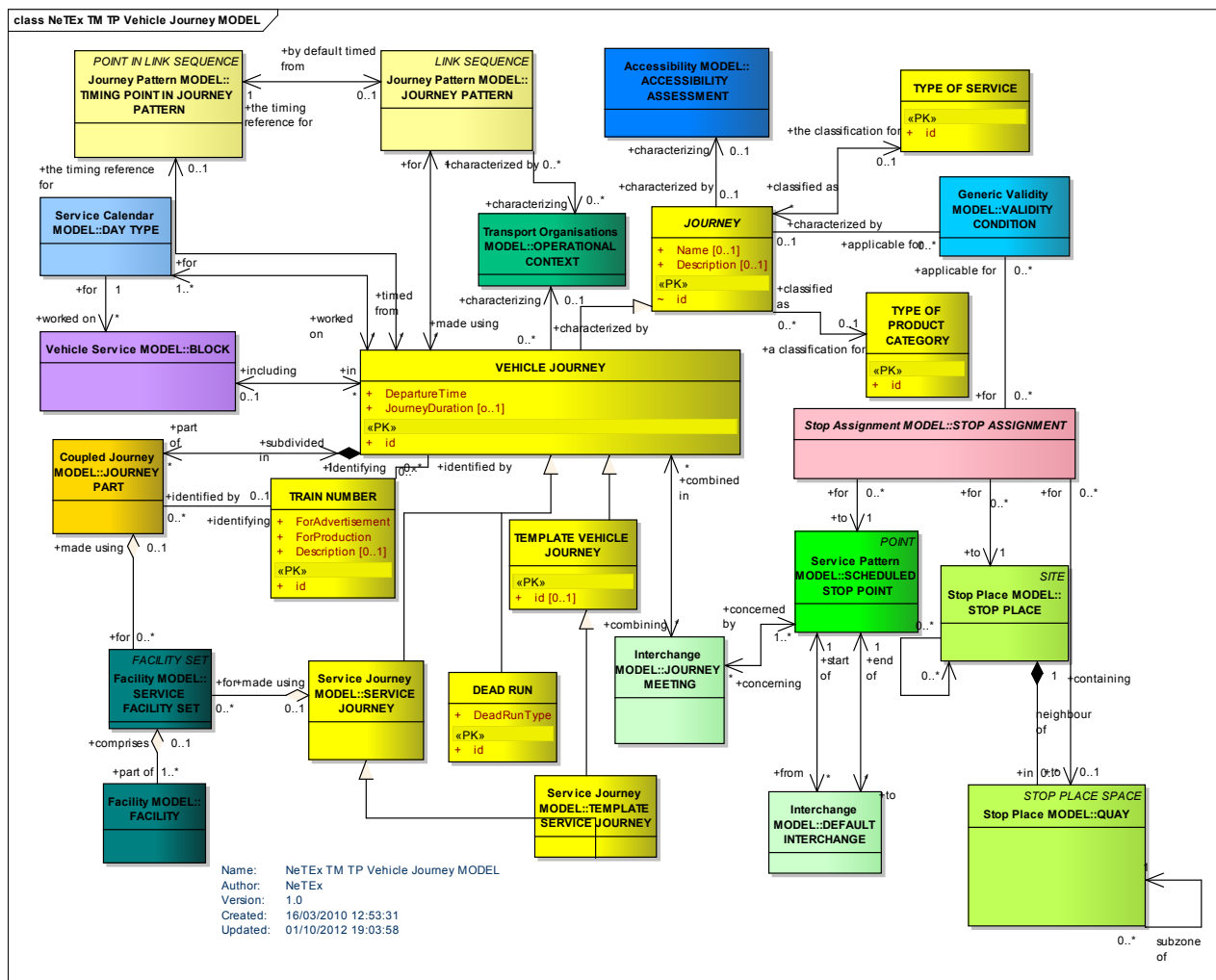


Figure 11 — Vehicle Journey – Conceptual MODEL (UML)

7.2.1.2 Vehicle Journey – Physical Model

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

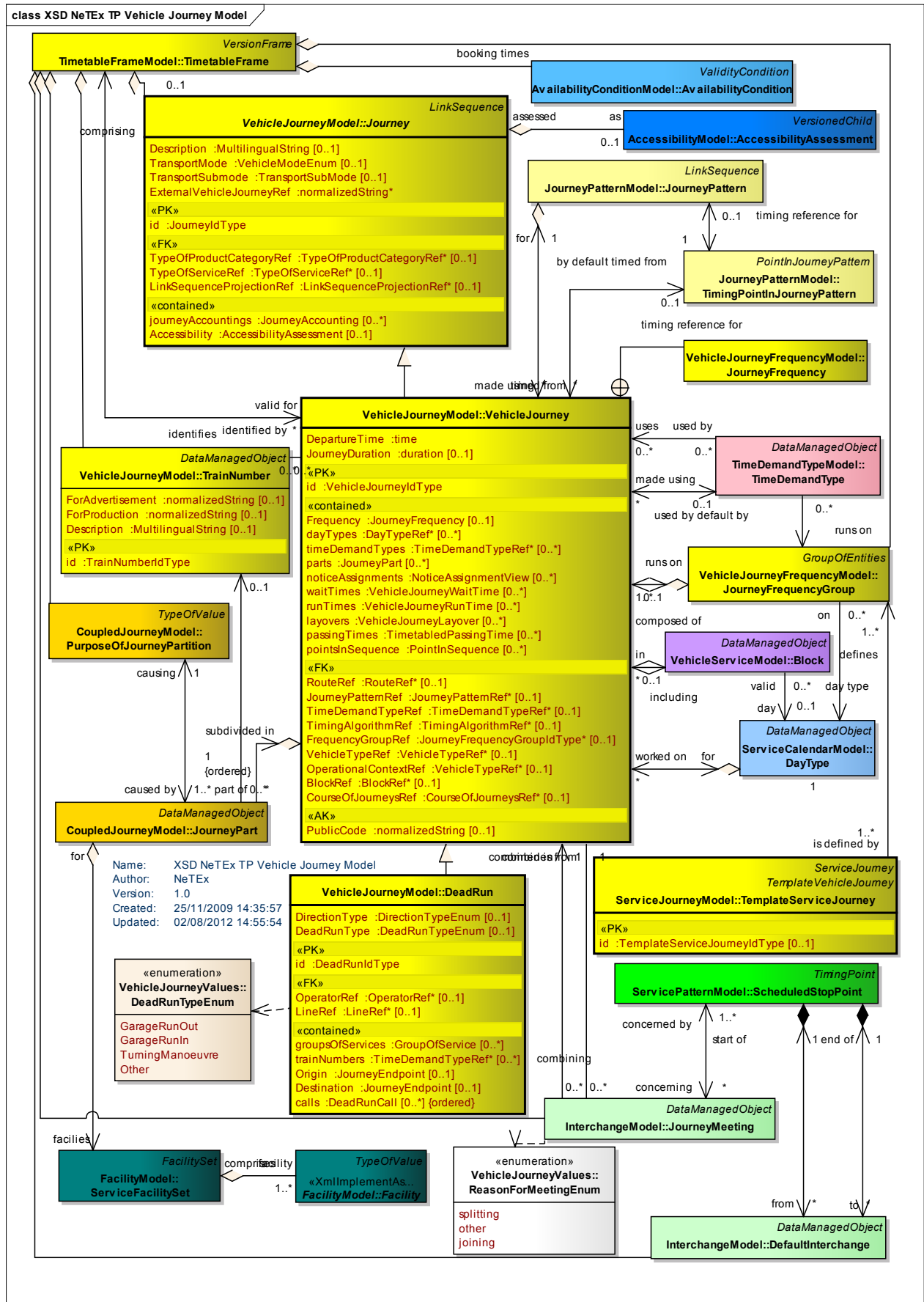


Figure 12 — Vehicle Journey – Physical Model (UML)

7.2.1.3 Vehicle Journey – Attributes and XSD

7.2.1.3.1 Journey – Model Element

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

Table 2 – Journey – Element

Classification	Name	Type	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.
	TransportSubmode	<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.
«cntd»	journeyAccountings	<i>JourneyAccounting</i>	0:*	JOURNEY ACCOUNTINGs that apply of JOURNEY.

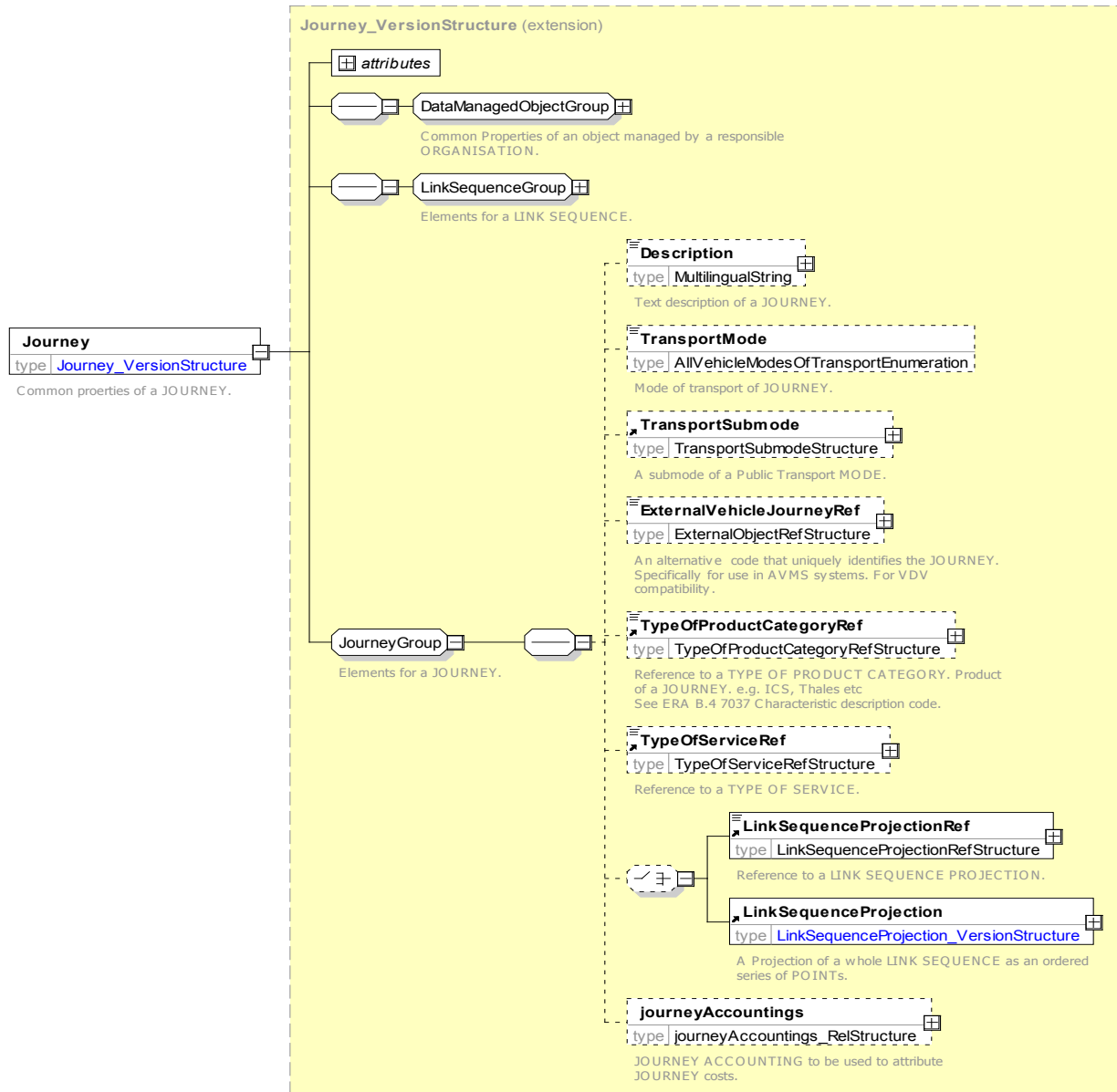


Figure 13 — Journey — XSD

7.2.1.3.2 VehicleJourney – Model Element

The planned movement of a public transport vehicle on a DAY TYPE from the start point to the end point of a JOURNEY PATTERN on a specified ROUTE.

Table 3 – *VehicleJourney* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<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.
	JourneyDuration	<i>xsd:duration</i>	0:1	Duration of VEHICLE JOURNEY.
«cntd»	Frequency	<i>JourneyFrequency</i>	0:*	Headway frequencies for VEHICLE JOURNEY.
«FK»	FrequencyGroupRef	<i>JourneyFrequencyGroupRef</i>	0:1	FREQUENCY GROUP of VEHICLE JOURNEY.
«FK»	DayTypeRef	<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»	BlockRef	<i>BlockRef</i>	0:1	Reference to BLOCK 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»	noticeAssignment	<i>NoticeAssignmentView</i>	0:*	NOTICE ASSIGNMENTs that apply to JOURNEY.
«cntd»	pointsInSequence	<i>PointInSequence</i>	0:*	POINTs in SEQUENCE for VEHICLE JOURNEY.

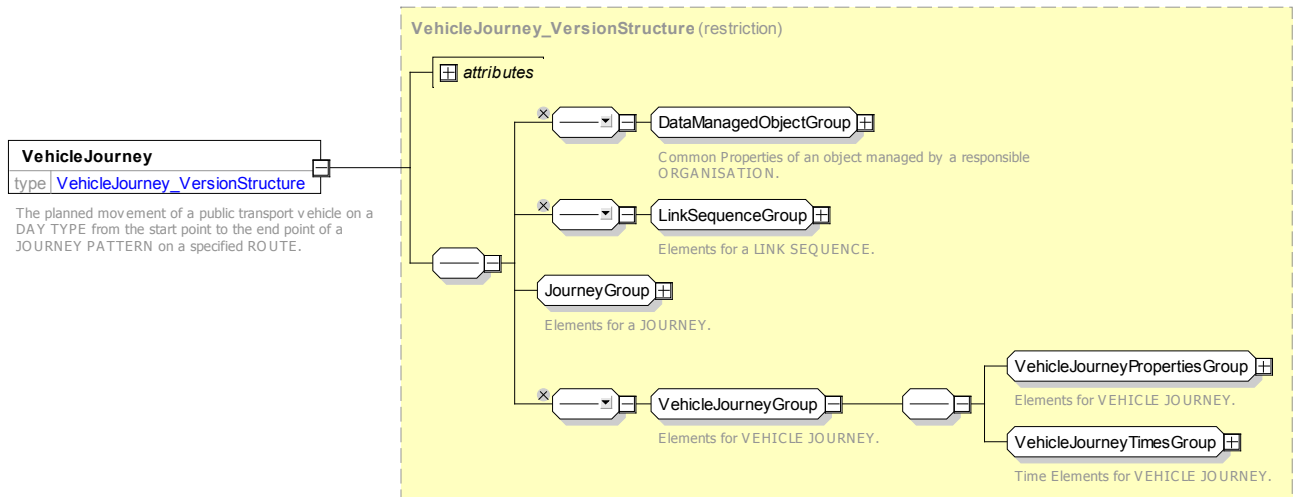


Figure 14 — *VehicleJourney* — XSD

7.2.1.3.2.1 VehicleJourneyReferencesGroup - Group



Figure 15 — *VehicleJourneyReferencesGroup* — XSD

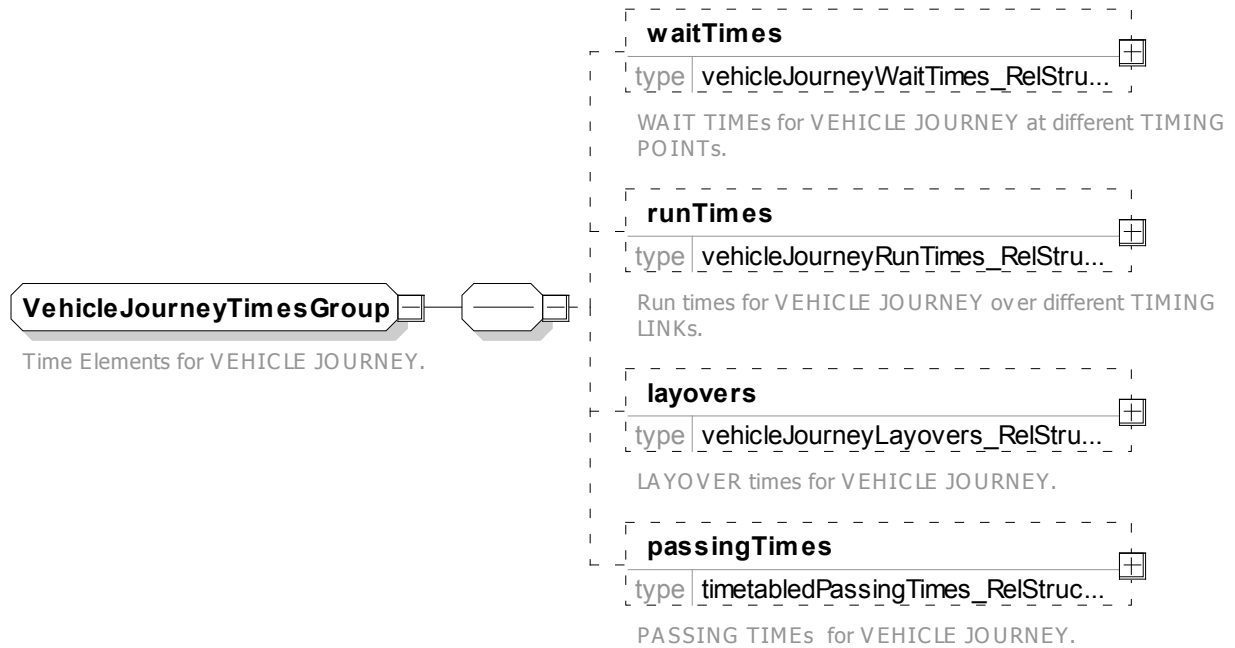


Figure 16 — VehicleJourneyTimesGroup — XSD

7.2.1.3.3 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.
«FK»	VehicleTypeRef	VehicleTypeRef	0:1	Reference to TYPE OF VEHICLE to use on of SERVICE JOURNEY.
«EV»	OperatorRef	OrganisationRef	0:1	Reference to OPERATOR of SERVICE JOURNEY.
«EV»	LineRef	LineRef	0:1	Reference to LINE of SERVICE JOURNEY.
	DirectionType	DirectionTypeEnum	0:1	Type of DIRECTION.
«cntd»	groupsOfServices	GroupOfServices	0:*	GROUP OF SERVICES to which DEAD RUN belongs.
«cntd»	trainNumbers	TrainNumber	0:*	TRAIN NUMBERS associated with DEAD RUN.
«cntd»	Origin	JourneyEndpoint	0:1	Origin of DEAD RUN.
«cntd»	Destination	JourneyEndpoint	0:1	Destination of DEAD RUN.
	DeadRunType	DeadRunTypeEnum	1:1	Type of DEAD RUN.
«cntd»	calls	DeadRunCall	0:*	CALLs making up DEAD RUN.

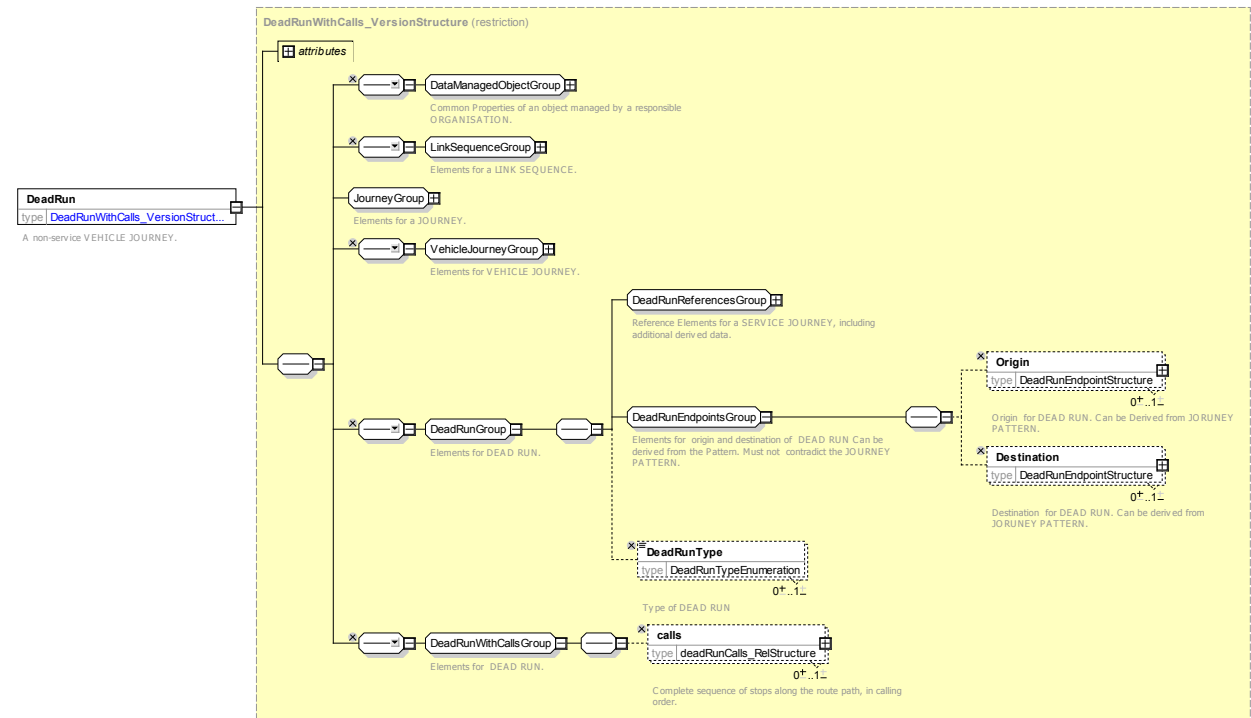
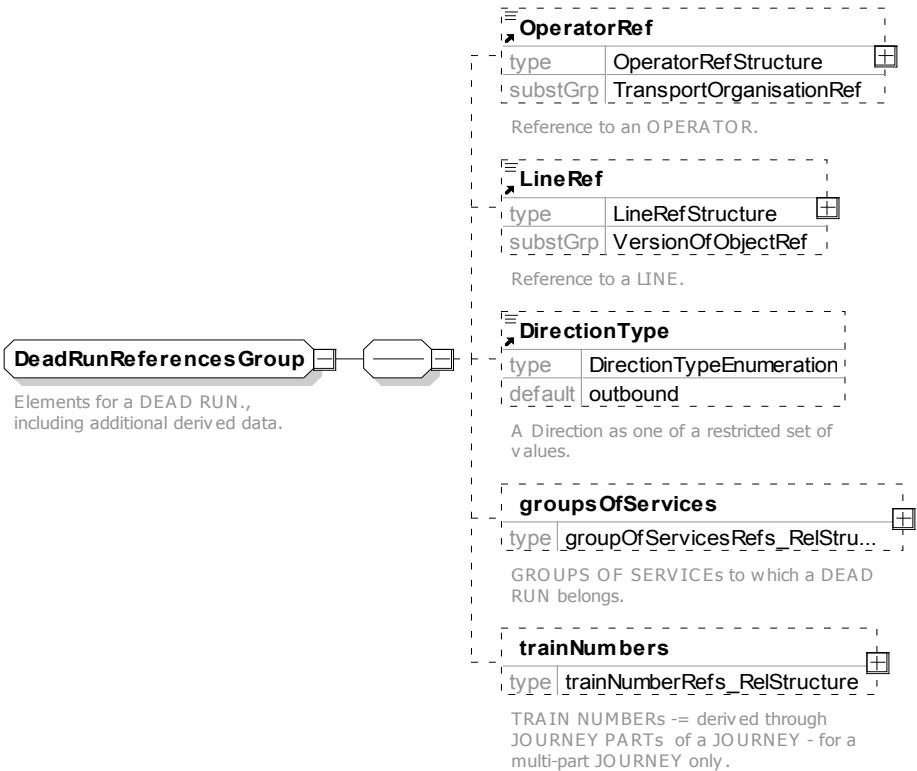


Figure 17 — DeadRun — XSD

7.2.1.3.3.1 DeadRunReferencesGroup — Group



Generated by XMLSpy

www.altova.com

Figure 18 — DeadRunReferencesGroup — XSD

7.2.1.3.3.2 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

7.2.1.3.4 TemplateVehicleJourney – Model Element

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

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

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

Table 6 – TemplateVehicleJourney – Element

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

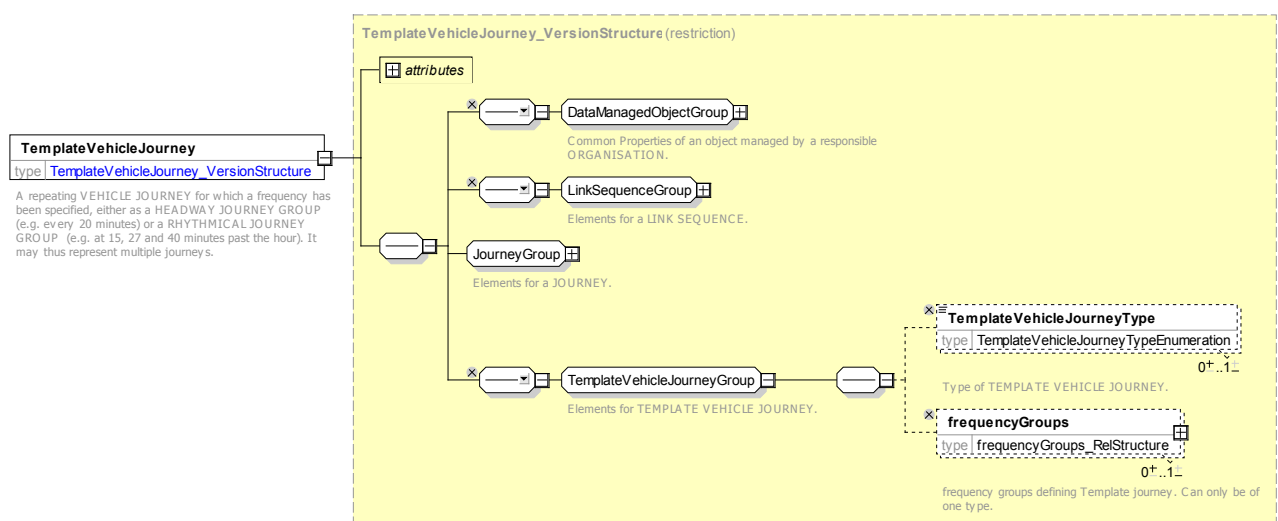


Figure 19 — TemplateVehicleJourney — XSD

7.2.1.3.4.1 TemplateVehicleJourneyType – Allowed values

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

Table 7 – TemplateVehicleJourneyType – Allowed values

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

7.2.1.3.5 TypeOfService – Model Element

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

Table 8 – TypeOfService – Element

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

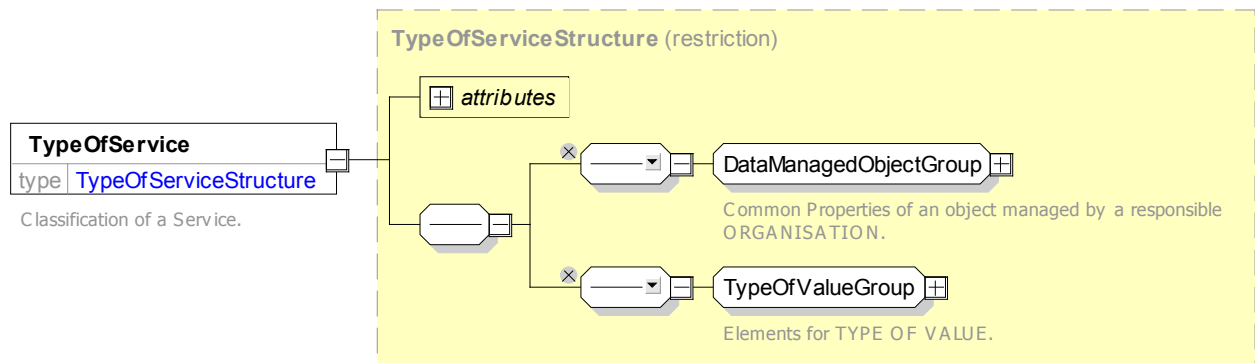


Figure 20 — TypeOfService — XSD

7.2.1.3.6 TypeOfProductCategory – Model Element

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

Table 9 – TypeOfProductCategory – Element

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

7.2.1.3.7 TrainNumber – Model Element

Specification of codes assigned to particular VEHICLE JOURNEYS when operated by TRAINS of COMPOUND TRAINS according to a functional purpose (passenger information, operation follow-up, etc.).

Table 10 – TrainNumber – Element

Classif ication	Name	Type	Cardin ality	Description
::>	::>	<i>DataManagedObject</i>	::>	TIME DEMAND PROFILE inherits from DATA MANAGED OBJECT
«PK»	id	<i>TrainNumberIdType</i>	1:1	Identifier 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.
	Description	<i>MultilingualString</i>	0:1	Description of TRAIN NUMBER.

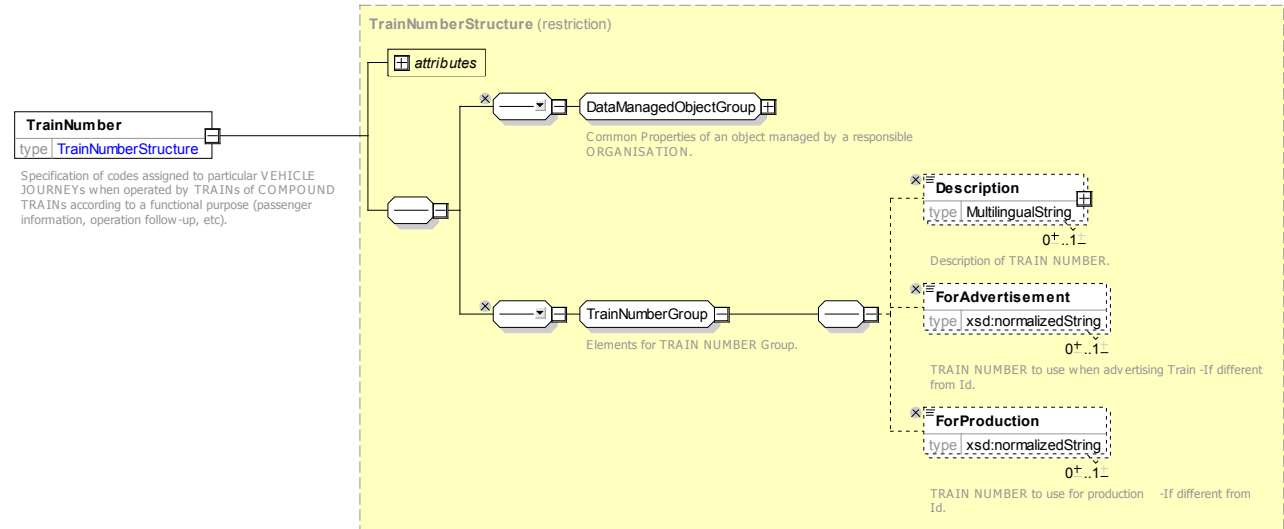


Figure 21 — TrainNumber — XSD

7.2.1.3.8 JourneyDesignator – Model Element

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

Table 11 – *JourneyDesignator* – Element

Classification	Name	Type	Cardinality	Description
«FK»	<i>FromPointRef</i>	<i>TimingPointRef</i>	0:1	Origin TIMING POINT to use to identify a JOURNEY.
«FK»	<i>ToPointRef</i>	<i>TimingPointRef</i>	0:1	Destination TIMING POINT to use to identify JOURNEY.
	<i>DepartureTime</i>	<i>xsd:time</i>	0:1	Departure time of VEHICLE JOURNEY from origin point.
	<i>ArrivalTime</i>	<i>xsd:time</i>	0:1	Arrival time of VEHICLE JOURNEY at destination point.
«AK»	<i>AlternativeJourneyRef</i>	<i>xsd:normalizedString</i>	0:1	Alternative identifier of JOURNEY. May possibly be unique only in a limited context.
«FK»	<i>DayTypeRef</i>	<i>DayTypeRef</i>	0:1	DAY TYPE to use to identify JOURNEY.

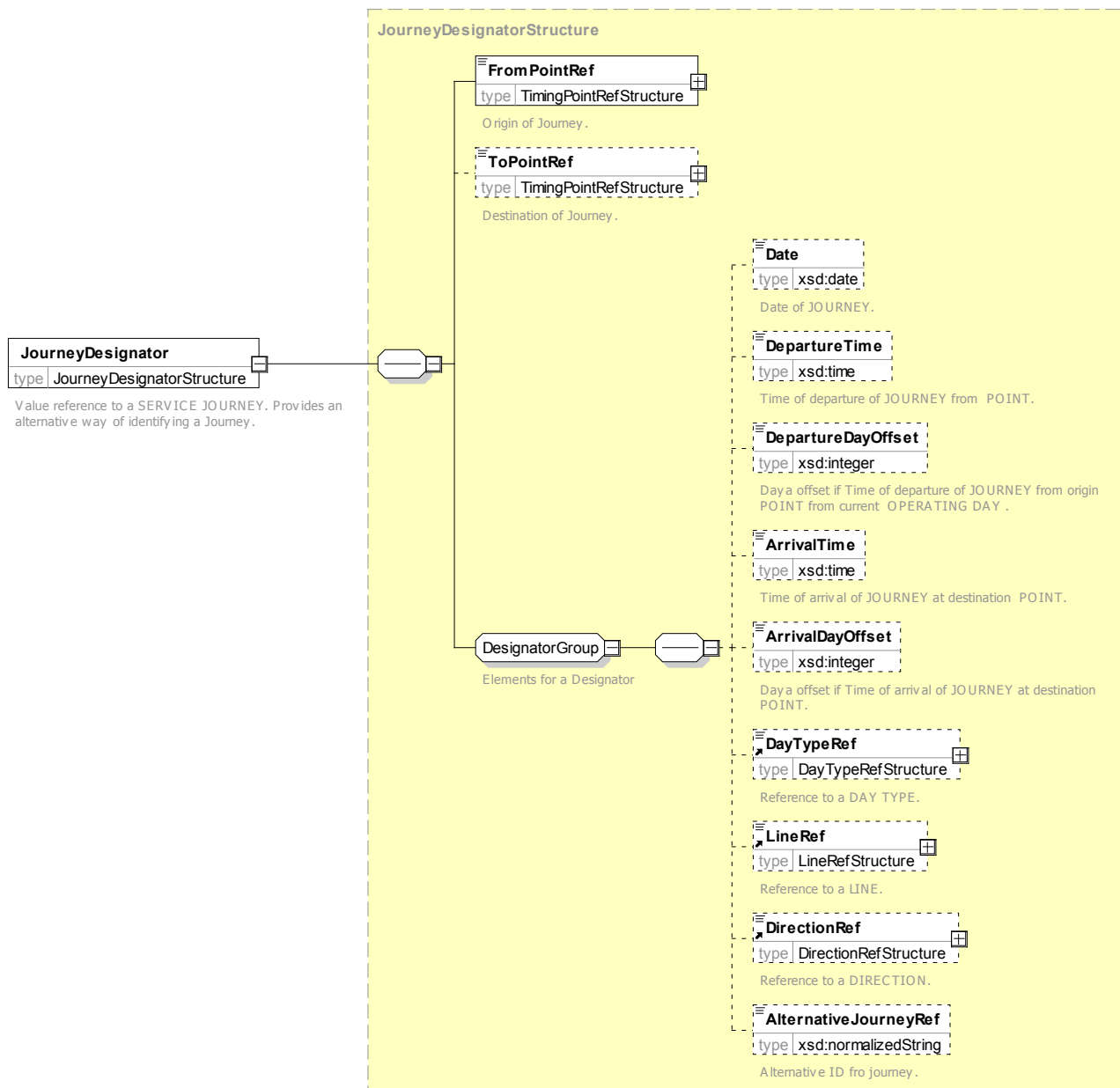


Figure 22 — JourneyDesignator — XSD

7.2.1.4 XML Examples of Vehicle Journey

See SERVICE JOURNEY for examples of a VEHICLE JOURNEY.

7.2.1.4.1 Dead Run XML Example with passing times and calls

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

For EXAMPLE

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

```

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

```

7.2.2 Service Journey

7.2.2.1 SERVICE JOURNEY – Conceptual MODEL

NOTE The following explanations use excerpts from Transmodel.

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

- As the service between an origin and a destination, as advertised to the public.
- As the longest service during which a passenger is allowed to stay on the same vehicle.

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

A default VEHICLE TYPE may be proposed for a journey chosen according to the time of day at which a SERVICE JOURNEY takes place, and the ROUTE and JOURNEY PATTERN it covers. The proposed VEHICLE TYPE will preferably be taken into account by the scheduling algorithm used to compile blocks of vehicle operation. The choice of such a preference may take into account a ranked list of VEHICLE TYPES for each SERVICE JOURNEY PATTERN. This is described by the entity VEHICLE TYPE PREFERENCE, depending on a particular SERVICE JOURNEY PATTERN, for which a priority ‘rank’ is given for each VEHICLE TYPE, for each DAY TYPE and TIME DEMAND TYPE.

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

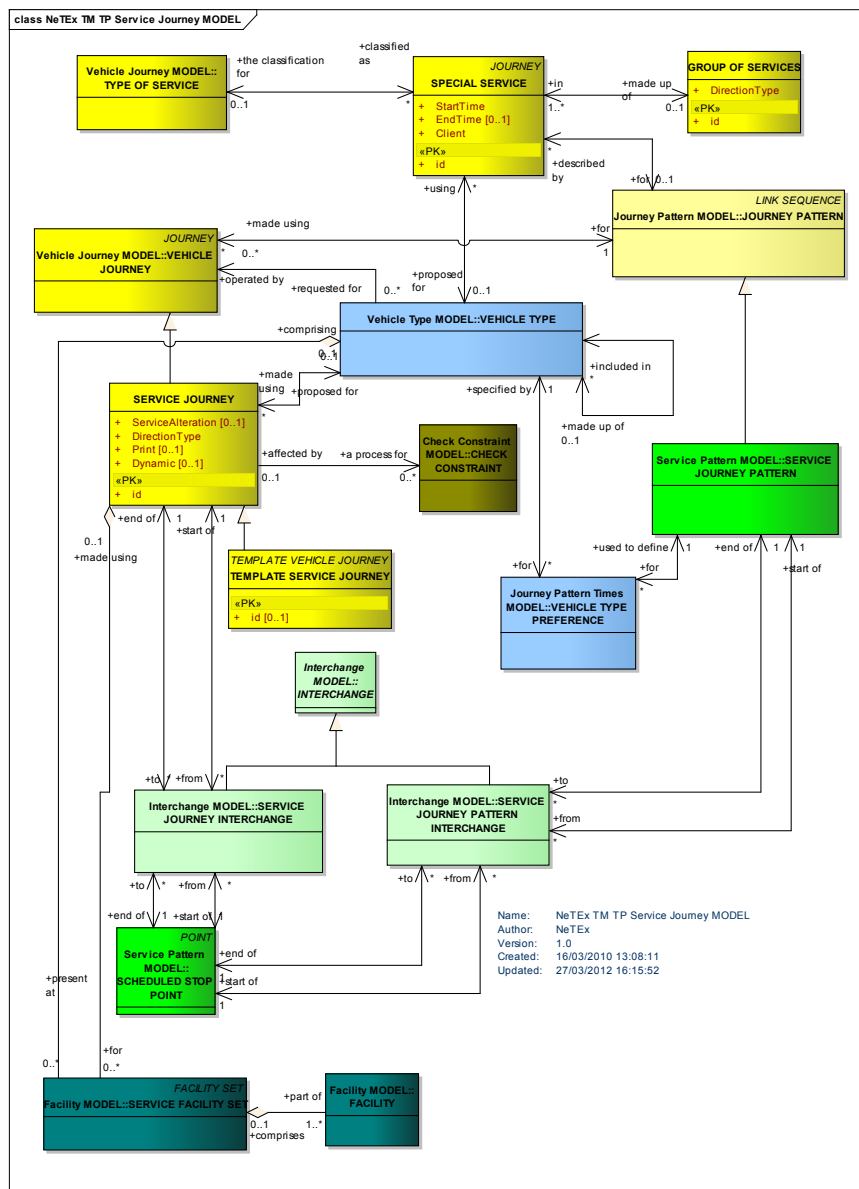


Figure 23 — Service Journey – Conceptual MODEL (UML)

7.2.2.2 Special Services

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

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

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

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

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

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

7.2.2.3 Service Journey – Physical Model

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

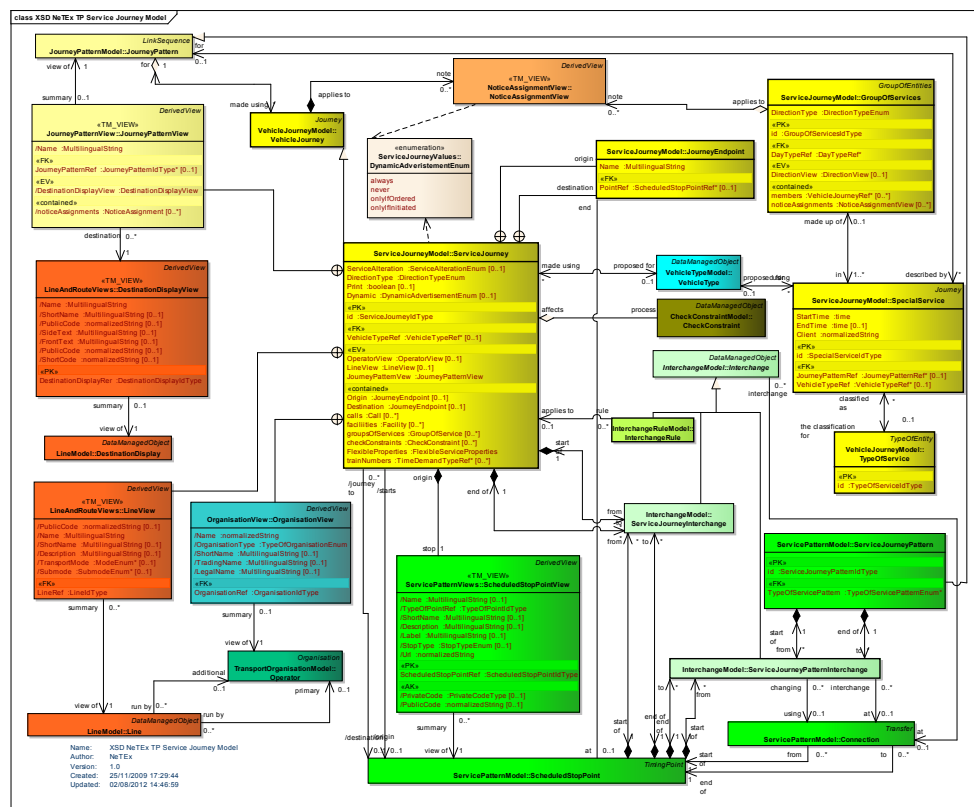


Figure 24 — Service Journey – Physical Model (UML)

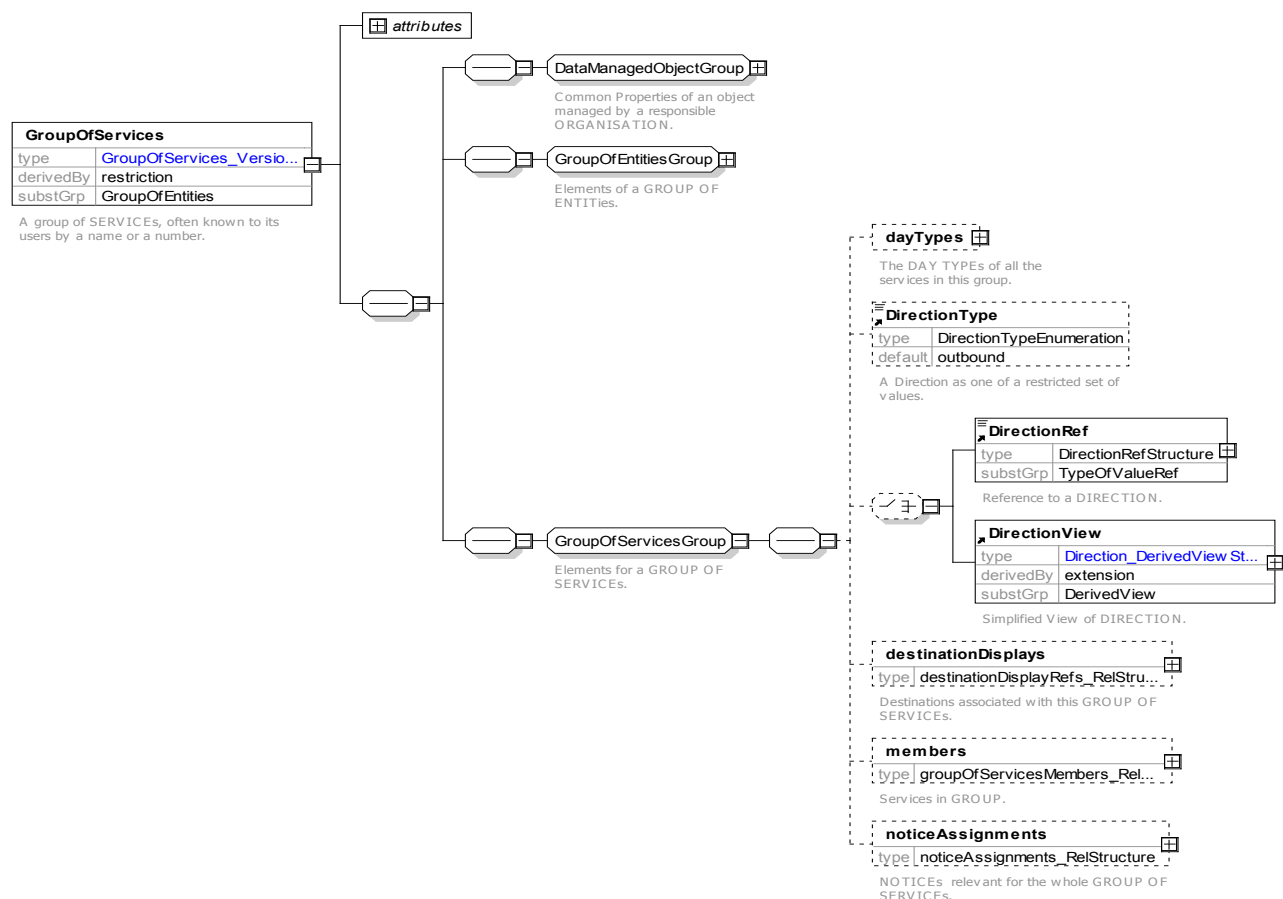
7.2.2.4 Service Journey – Attributes and XSD

7.2.2.4.1 GroupOfServices – Model Element

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

Table 12 – GroupOfServices – Element

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



Generated by XMLSpy

www.altova.com

Figure 25 — GroupOfServices — XSD

7.2.2.4.2 ServiceJourney – Model Element

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

Table 13 – ServiceJourney – Element

Classification	Name	Type	Cardinality	Description
::>	::>	Journey	::>	SERVICE JOURNEY inherits from JOURNEY. It includes elements from VEHICLE JOURNEY
«PK»	id	ServiceJourneyIdType	1:1	Identifier of SERVICE JOURNEY.
	ServiceAlteration	ServiceAlterationEnum	0:1	Status to journey - planned, cancelled, etc.
«FK»	VehicleTypeRef	VehicleTypeRef	0:1	Reference to TYPE OF VEHICLE to use on of SERVICE JOURNEY.
«EV»	OperatorView	OrganisationView	0:1	Reference to OPERATOR of SERVICE JOURNEY. May include OPERATOR's derived values.
«EV»	LineView	LineView	0:1	Reference to LINE of SERVICE JOURNEY. May include derived values of LINE.
	DirectionType	DirectionTypeEnum	1:1	Type of DIRECTION.
	Print	xsd:boolean	0:1	Whether this journey should be visible to public in print channels.
	Dynamic	DynamicAdvertisementEn	0:1	When this journey should be visible to public in

		<i>um</i>		dynamic channels.
«EV»	<i>JourneyPatternView</i>	<i>JourneyPatternView</i>	1:1	Reference to JOURNEY PATTERN of SERVICE JOURNEY and derived values of PATTERN.
«cntd»	<i>Origin</i>	<i>JourneyEndpoint</i>	0:1	Origin of SERVICE JOURNEY.
«cntd»	<i>Destination</i>	<i>JourneyEndpoint</i>	0:1	Destination of SERVICE JOURNEY.
«cntd»	<i>calls</i>	<i>Call</i>	0:*	CALLs made by SERVICE JOURNEY.
«cntd»	<i>facilities</i>	<i>Facility</i>	0:*	Facilities on SERVICE JOURNEY.
«cntd»	<i>groupsOfServices</i>	<i>GroupOfServices</i>	0:*	GROUP OF SERVICES to which SERVICE JOURNEY belongs.
«cntd»	<i>checkConstraints</i>	<i>CheckConstraint</i>	0:*	CHECK CONSTRAINTs which apply to SERVICE JOURNEY, check in time, security time. These are advisory and not for journey planning.
«cntd»	<i>FlexibleServiceProperties</i>	<i>FlexibleServiceProperties</i>	1:1	Flexible properties of SERVICE JOURNEY.

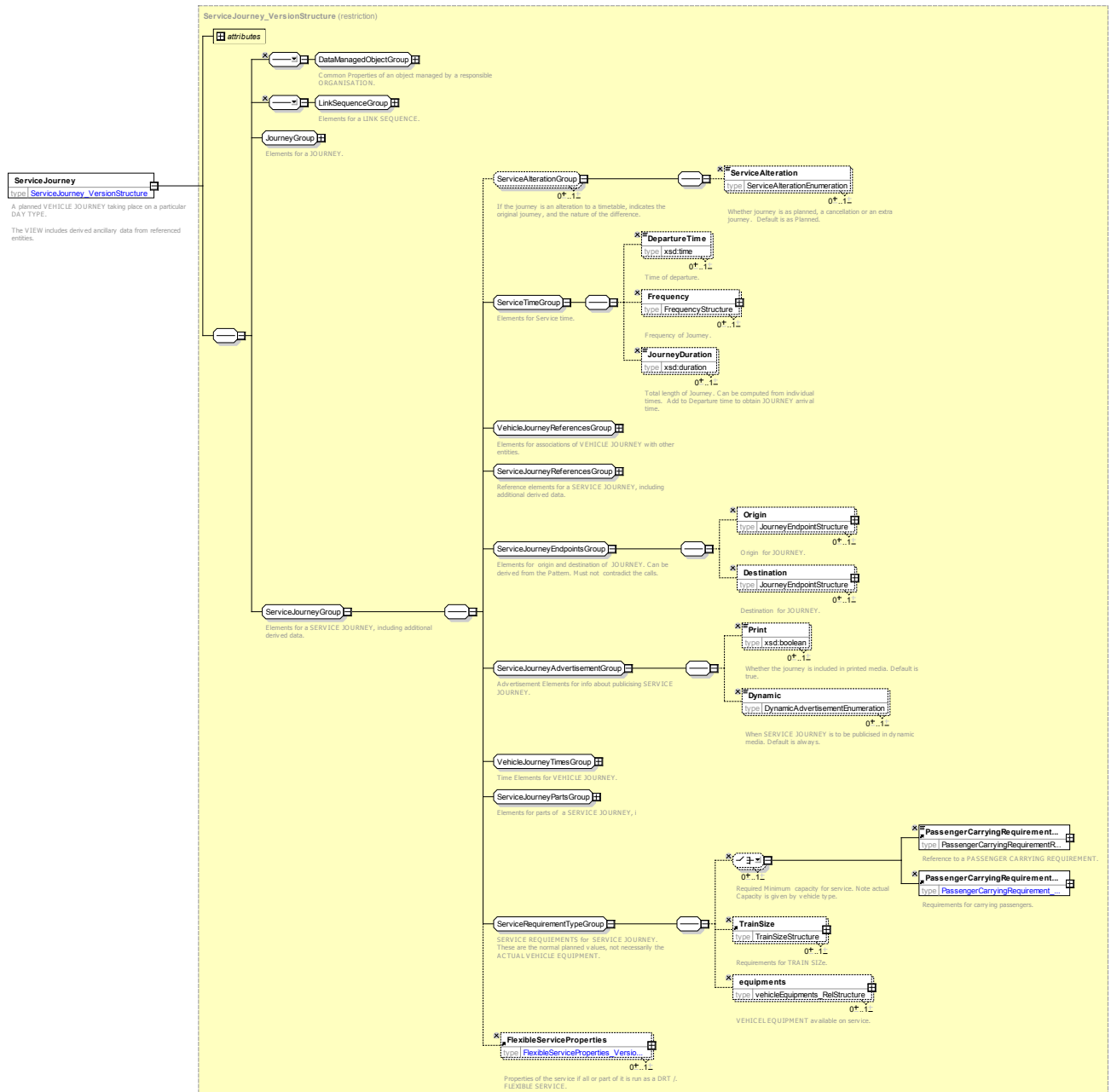


Figure 26 — ServiceJourney — XSD

7.2.2.4.2.1 ServiceAlteration – Allowed values

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

Table 14 – ServiceAlteration – Allowed values

Value	Description
<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

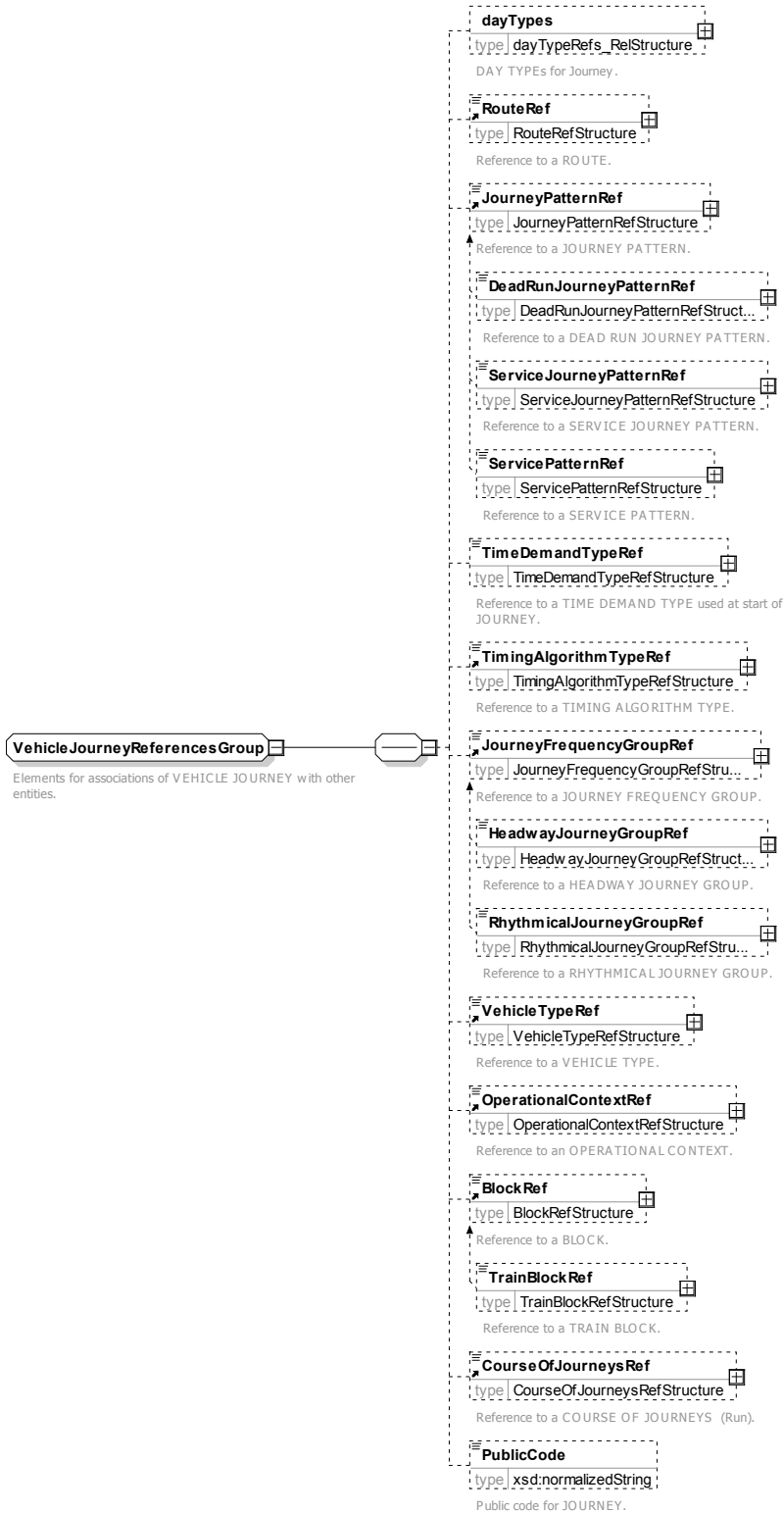


Figure 27 — *VehicleJourneyReferencesGroup* — XSD

7.2.2.4.2.3 ServiceJourneyReferencesGroup – Model Element



Generated by XMLSpy

www.altova.com

Figure 28 — ServiceJourneyReferencesGroup — XSD

7.2.2.4.2.4 DynamicAdvertisement – Allowed values

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

Table 15 – DynamicAdvertisement – Allowed values

Value	Description
<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 VehicleJourneyTimesGroup – Model Element

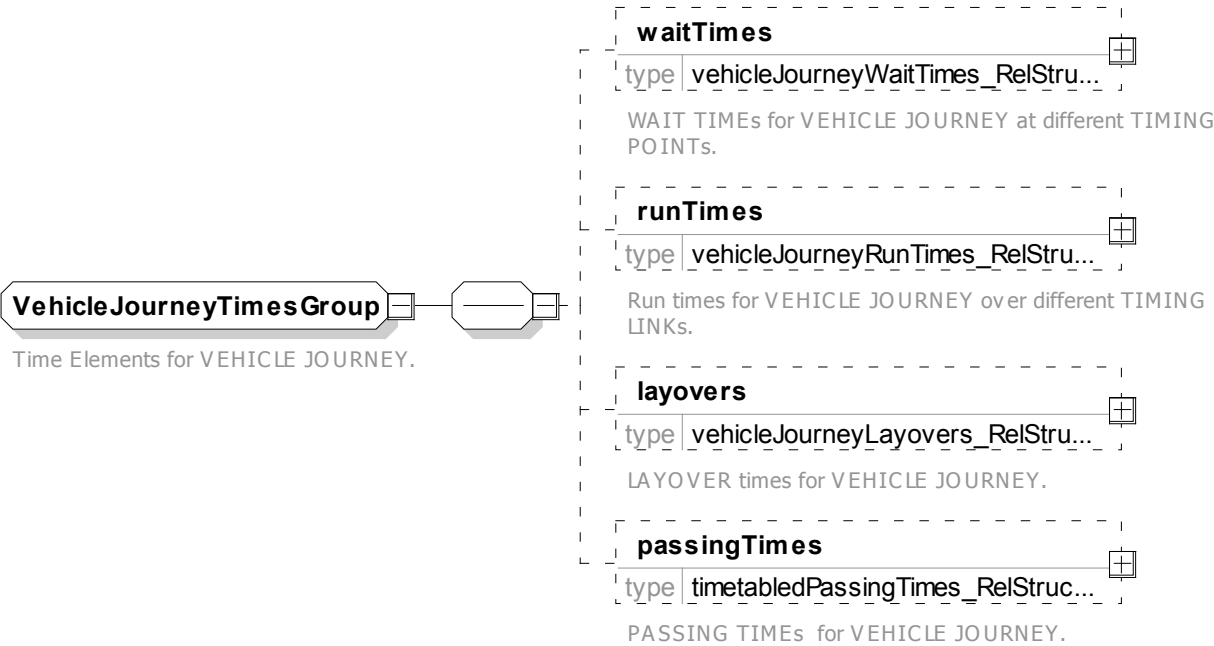


Figure 29 — *VehicleJourneyTimesGroup* — XSD

7.2.2.4.2.6 ServiceJourneyPartsGroup – Model Element

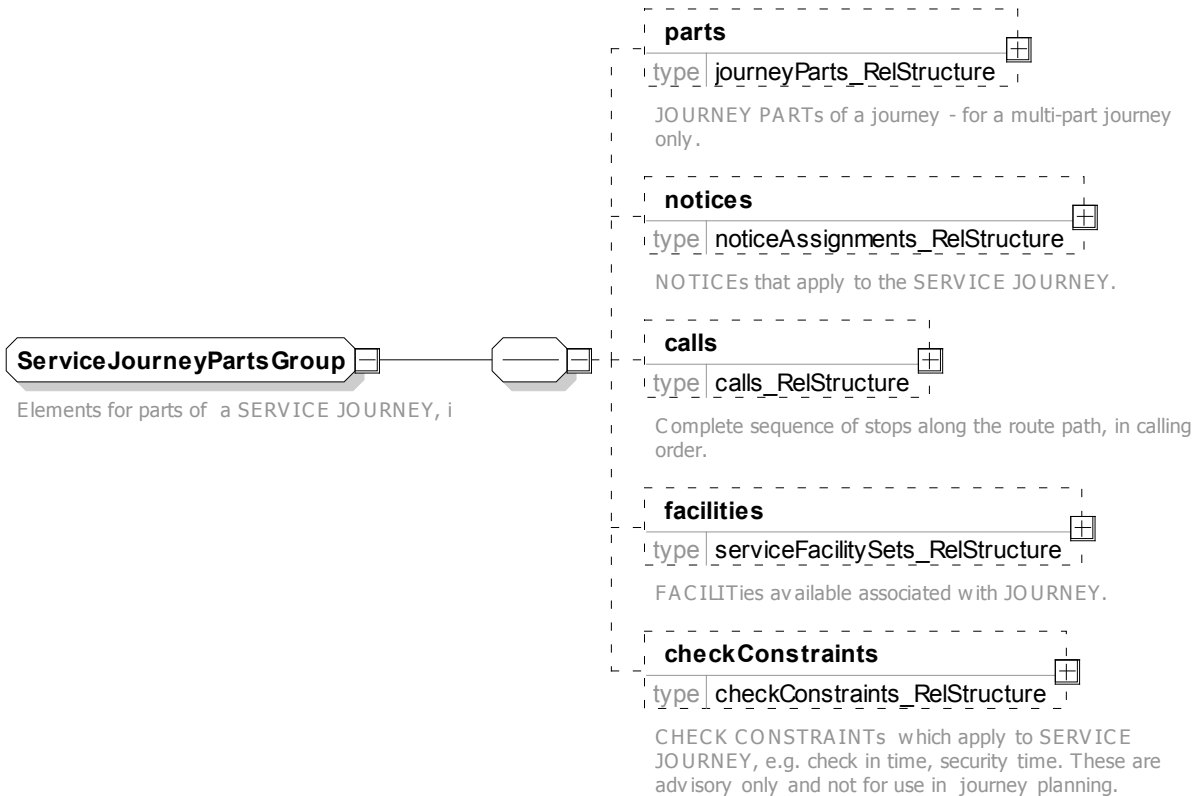


Figure 30 — *ServiceJourneyPartsGroup* — XSD

7.2.2.4.2.7 ServiceRequirementTypeGroup – Model Element

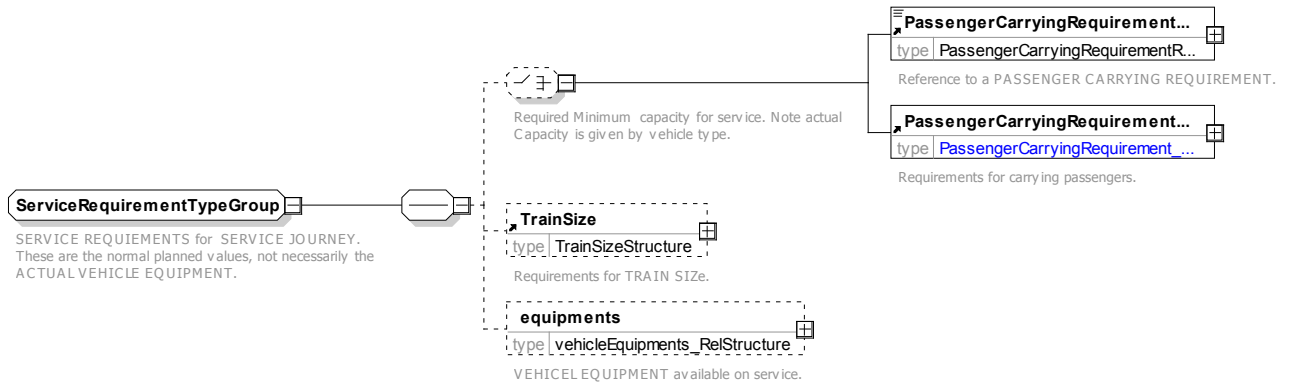


Figure 31 — *ServiceRequirementTypeGroup* — XSD

7.2.2.4.3 SpecialService – Model Element

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

Table 16 – SpecialService – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>Journey</i>	::>	SPECIAL SERVICE inherits from JOURNEY.
«PK»	id	<i>SpecialServiceIdType</i>	1:1	Identifier of SPECIAL SERVICE.
	StartTime	<i>xsd:time</i>	1:1	Start time for SPECIAL SERVICE.
	EndTime	<i>xsd:time</i>	0:1	End time for SPECIAL SERVICE.
	Client	<i>xsd:normalizedString</i>	1:1	Client for SPECIAL SERVICE.
	DayTypeRef	<i>DayTypeRef</i>	0:1	DAY TYPE 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

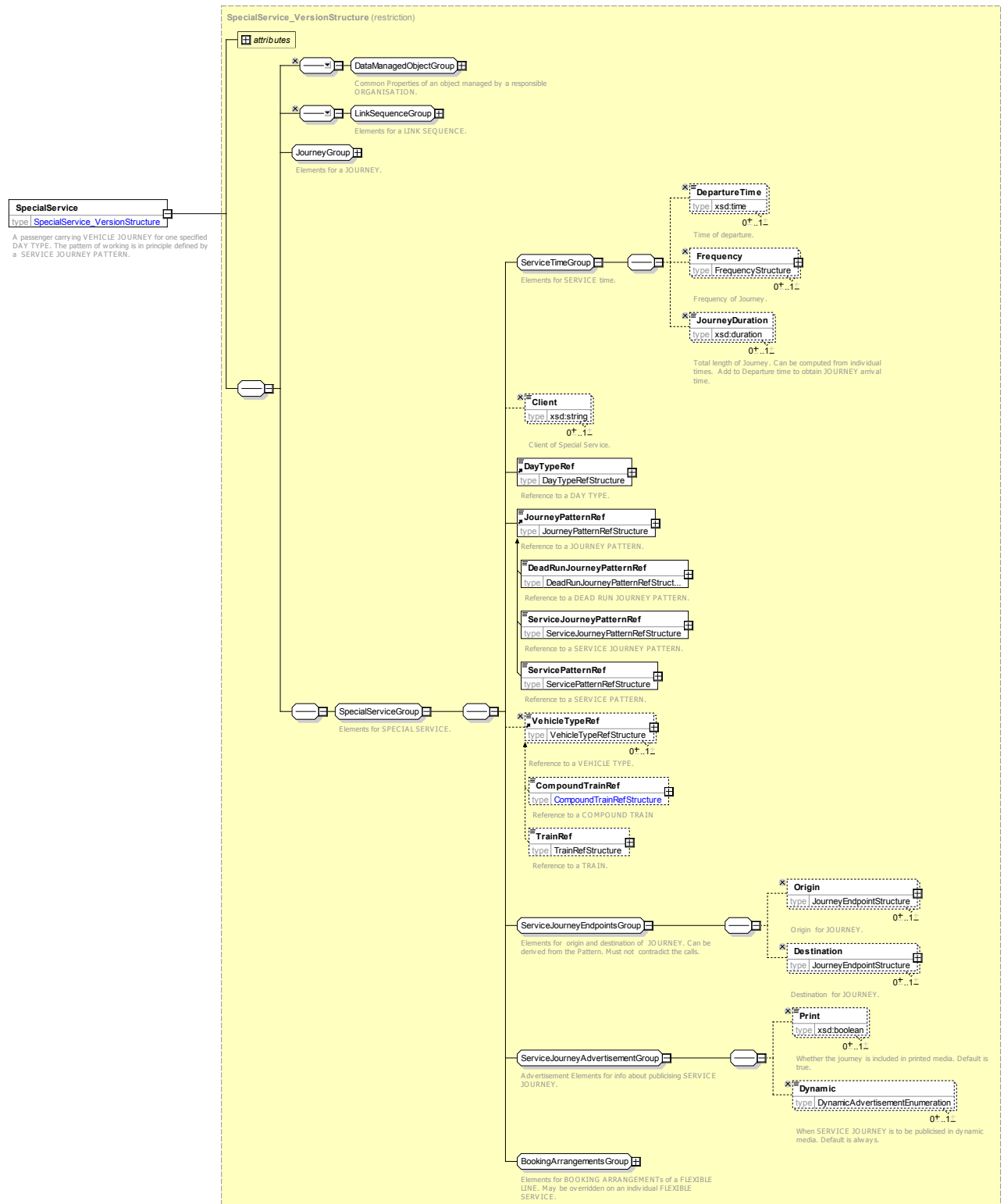


Figure 32 — Special/Service — XSD

7.2.2.4.3.1 BookingArrangements – Group

Information about booking a flexible journey.

Table 17 – *BookingArrangements* Group– Group

Classification	Name	Type	Cardinality	Description
	BookingMethods	<i>BookingMethodEnum</i>	0:*	Booking method for FLEXIBLE LINE.
	BookingAccess	<i>BookingAccessEnum</i>	0:1	Who can make a booking.
	LatestBookingTime	<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.

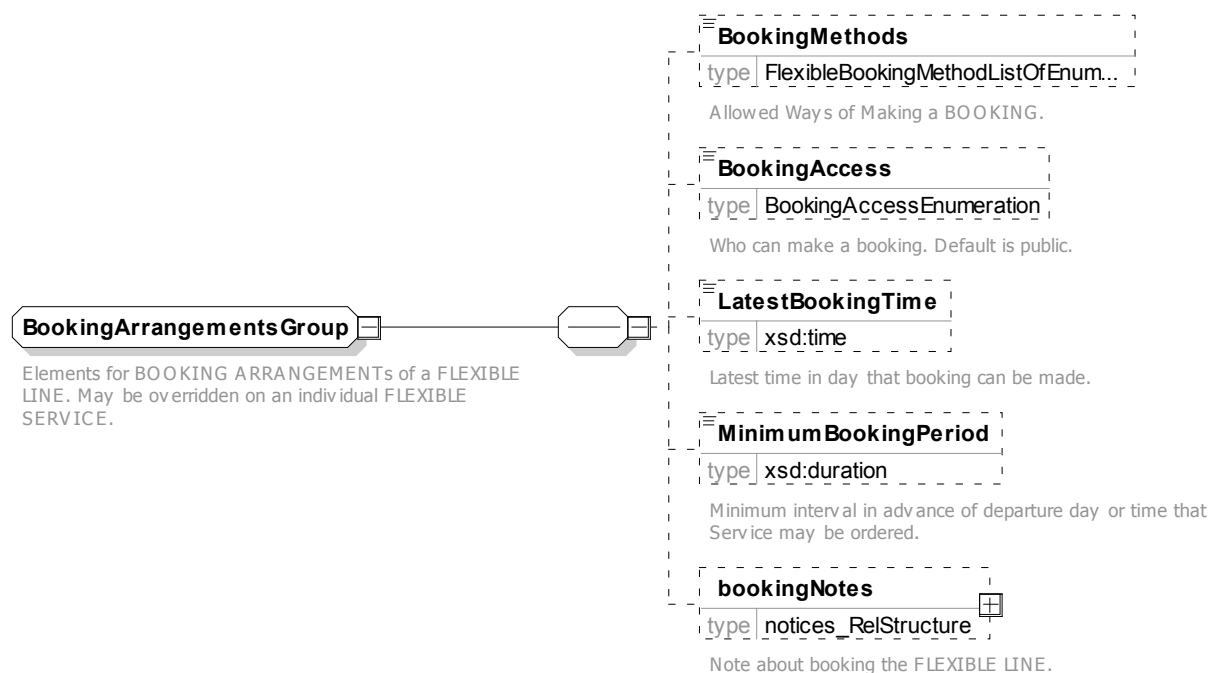


Figure 33 — *BookingArrangementsGroup* — XSD

7.2.2.4.4 TemplateServiceJourney – Model Element

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

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

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

Table 18 – *TemplateServiceJourney* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<i>TemplateVehicleJourney</i>	::>	TEMPLATE SERVICE JOURNEY inherits from TEMPLATE VEHICLE JOURNEY.
«PK»	id	<i>TemplateServiceJourney/IdType</i>	0:1	Identifier of TEMPLATE SERVICE JOURNEY.

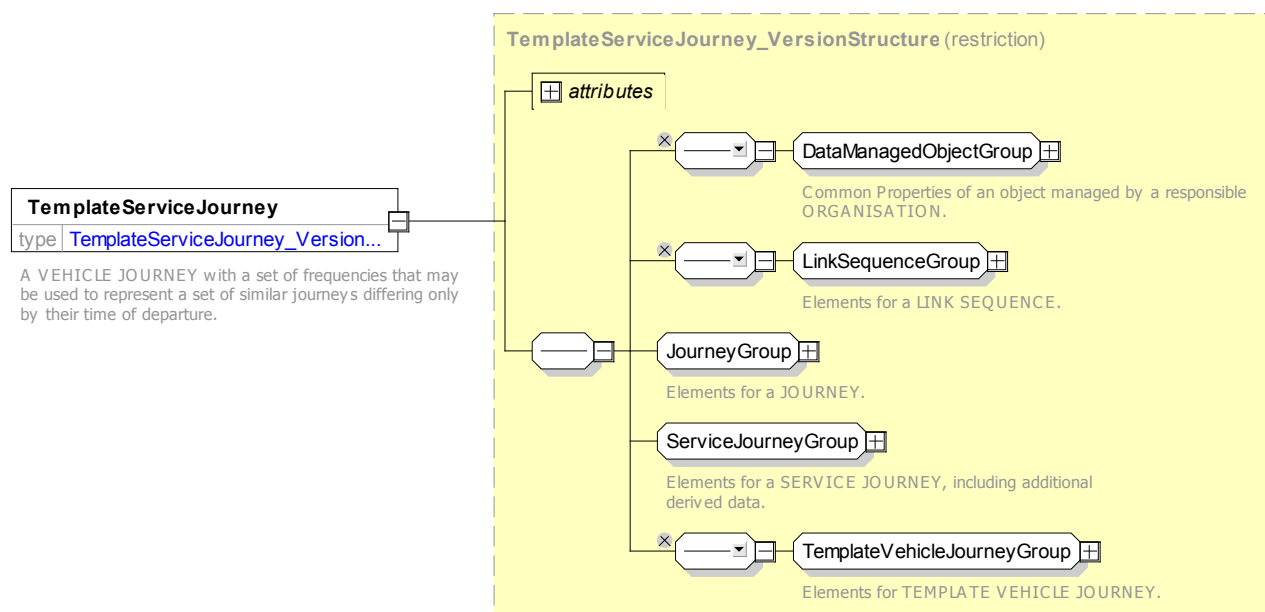


Figure 34 — TemplateServiceJourney — XSD

7.2.2.5 XML Examples of Service Journey

7.2.2.5.1 Service Journey XML Example with passing times and calls

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

For EXAMPLE:

```
<ServiceJourney version="any" id="hde:sj_24o_02_AxxD">
  <DepartureTime>14:20:00.0Z</DepartureTime>
  <dayTypes>
    <DayTypeRef version="any" ref="hde:Daype:DT_01-MF-NH"/>
  </dayTypes>
  <ServicePatternRef version="any" ref="hde:svp_24o_AxxD"/>
  <TimeDemandTypeRef version="any" ref="mybus:td_01"/>
  <LineRef version="any" ref="mybus:LN_24"/>
  <JourneyPatternView>
    <RouteRef version="any" ref="mybus:RT_24o"/>
    <DirectionType>outbound</DirectionType>
    <DestinationDisplayRef version="any" ref="mybus:DST_Delta"/>
  </JourneyPatternView>
  <passingTimes>
    <TimetabledPassingTime version="any"
id="mybus:tpt_24o_02_AxxD_SSP_001">
      <StopPointInJourneyPatternRef version="any"
ref="hde:spijp_24o_AxxD_01"/>
      <DepartureTime>14:20:00.0Z</DepartureTime>
    </TimetabledPassingTime>
    <TimetabledPassingTime version="any"
id="hde:tpt_24o_02_AxxD_SSP_001_t1">
      <TimingPointInJourneyPatternRef version="any"
ref="hde:tpijp_24o_AxxD_02"/>
      <DepartureTime>14:30:00.0Z</DepartureTime>
    </TimetabledPassingTime>
  </passingTimes>
</ServiceJourney>
```

```
<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">
```