CEN/TC 278

Date: 2013-03

TC 278 WI 00278330

CEN/TC 278

Secretariat: NEN

Public transport — Network and Timetable Exchange (NeTEx) — Part 3: Public transport fares exchange format

NeTEx — Haupt-Element — Teil 3: Teil-Titel

Transport Public — Echanges des informations planifiées (NeTEx) — Partie 3 : Echange des informations tarifaires pour le transport public

ICS:

Descriptors:

16614

FINAL DRAFT for NeTEx v1.1

2017.08.08	NK – DRAFT V1 with embedded diagrams	V38
2017.10.10	NJSK –Revised Draft Updated UML Diagrams	V10-nk2
2017.12.03	NJSK –Revised Draft Apply CR changes and correct from CD anomaly list.	V10_nk3
2019.01.02	CD / HW. Take comments from review into account	
2019.04.19	NJSK. Modify for review CRS. Including Norway, EURA and UK Fare CRS. Integrate other changes.	Final Draft_2
2019.04.24	NJSK. Minor corrections.	Final Draft_4

Document type: Technical Specification
Document subtype:
Document stage: Formal Vote
Document language: E
C:\Users\nick.knowles\Google Drive\PROJECTSDRIVE\A_NetEx\2019.02.18-NX -Review\docs\prCEN TS 16614-3 FV (E)-2019-Final-Draft.docx STD Version 2.4a

Contents

Page

Foreword

This document (TC 278 WI 00278330) 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 Part3 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 series, which provides a complete exchange format for public transport networks, timetable description and fare information.

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

— Part 2 is the description of the scheduled timetables exchange format.

— Part 3 is the description of the fare information exchange format.

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

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

NOTE This document is highly technical, and a special care has been taken to keep the text readable. In particular a set of formatting conventions is followed that enhances the usual CEN writing rules in order to distinguish references to elements of the formal models within text:

— Transmodel terms and NeTEx conceptual model elements are in capital letters (JOURNEY PATTERN for example).

— NeTEx physical model names are in bold italic font and use camelcase style with no spaces (***JourneyPattern*** for example).

— NeTEx physical model attribute types are in italic style and use camelcase style with no spaces (*TypeOfEntity* for example).

Introduction

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

The first two parts of the European Technical Specification NeTEx specify a Network and Timetable Exchange for Public Transport. It is intended to be used to exchange data relating to scheduled public transport between the systems of PT organisations. It can also be seen as complementary to the SIRI (Service Interface for Real-time Information) standard, as SIRI needs a prior exchange of reference data from NeTEx's scope to provide the necessary context for the subsequent exchange of a real-time data.

This European Technical Specification (NeTEx – Part 3) specifies exchanges of Public Transport fares between systems and organisations. It is a complement to the Parts 1 & 2 in the sense that it uses a subset of concepts defined there.

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, (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: a modular reusing of 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). It is based on Transmodel V5.1 (EN 12986), IFOPT (CEN/ EN 28701) and SIRI (CEN/TS 15531-4/5 and EN 15531-1/2/3[1]) and supports the exchange of information of relevance for passenger information about public transport services and also for running Automated Vehicle Monitoring Systems (AVMS).

NOTE NeTEx is a refinement and an implementation of Transmodel and IFOPT; the definitions and explanations of these concepts are extracted directly from the respective standard and reused in NeTEx, sometimes with adaptations in order to fit the NeTEx context. Although the data exchanges targeted by NeTEx are predominantly oriented towards provisioning passenger information systems and AVMS with data from transit scheduling systems, it is not restricted to this purpose and NeTEx can also provide an effective solution to many other use cases for transport data exchange.

1.2 Fares scope

This Part3 of NeTEx, is specifically concerned with the exchange of fare structures and fare data, using data models that relate to the underlying network and timetable models defined in Part1 and Part2 and the Fare Collection data model defined in Transmodel V51. See the use cases below for the overall scope of Part3. In summary, it is concerned with data for the following purposes:

(i) To describe the many various possible fare structures that arise in public transport (for example, flat fares, zonal fares, time dependent fares, distance-based fares, stage fares, pay as you go fares, season passes, etc., etc.).

(ii) To describe the fare products that may be purchased having these fare structures and to describe the conditions that may attach to particular fares, for example if restricted to specific groups of users, or subject to temporal restrictions. These conditions may be complex.

(i) To allow actual price data to be exchanged. Note however that NeTEx does not itself specify pricing algorithms or how fares should be calculated. This is the concern of Fare Management Systems. It may be used may be used to exchange various parameters required for pricing calculations that are needed to explain or justify a fare.

(iii) To include the attributes and the text descriptions necessary to present fares and their conditions of sale and use to the public.

NeTEx should be regarded as being ‘upstream’ of retail systems and allows fare data to be managed and integrated with journey planning and network data in public facing information systems. It is complementary to and distinct from the ‘downstream’ ticketing and retail systems that sell fares and of the control systems that validate their use. See ‘Excluded Use Cases’ below for further information on the boundaries of NeTEx with Fare Management Systems.

1.3 Transport modes

All mass 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, air journeys, and air fares, but there has not been any specific consideration of any additional requirements that apply specifically to air transport.

1.4 Compatibility with existing standards and recommendations

The overall approach for the definition of fares within NeTEx Part 3 follows the approach used by Transmodel V5.1, namely the definition of access rights rather than of just products.

[1] Under development

This approach, used in Transmodel V5.1 (Fare Collection data model) to specify the access rights related to the urban public transport (for all urban modes) has been extended to cover access rights for long-distance rail.

NOTE The concepts from Transmodel V5.1 and IFOPT used and/or modified by NeTEx are incorporated into Transmodel V6 to guarantee compatibility and coherence of standards.

Concepts covered in NeTEx Part 1 and 2 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 and train routing restrictions.

In the case of long distance train access rights, 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. These relate in particular to the B1 (Non Reservation Tickets), B2 (Integrated Reservation Tickets) and B3 (Special Fares) along with various UIC Leaflets.

As regards the other exchange protocols for network and timetable exchanges, a formal compatibility is ensured with TransXChange (UK), VDV 452 (Germany), NEPTUNE (France), BISON (The Netherlands) and NOPTIS (Nordic Public Transport Interface Standard).

The exchange of data in NeTEx format can be undertaken using a variety of protocols. For example: through dedicated web services, through data file exchanges, or by using the SIRI exchange protocol as described in part 2 of the SIRI documentation. NeTEx adds additional services using the common SIRI transport mechanism.

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 - Part1: Context and framework*

EN 15531-2, *Public transport - Service interface for real-time information relating to public transport operations - Part2: Communications infrastructure*

EN 15531-3, *Public transport - Service interface for real-time information relating to public transport operations - Part3: 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)

3 Terms and definitions

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

3.1
ACCESS RIGHT IN PRODUCT
(Fare Product MODEL)
A VALIDABLE ELEMENT as a part of a PRE-ASSIGNED FARE PRODUCT, including its possible order in the set of all VALIDABLE ELEMENTs grouped together to define the access right assigned to that PRE-ASSIGNED FARE PRODUCT.

3.2
ACCESS RIGHT PARAMETER ASSIGNMENT
(Validity Parameters MODEL)
The assignment of a fare collection parameter (referring to geography, time, quality or usage) to an element of a fare system (access right, validated access, control mean, etc.).

3.3
AMOUNT OF PRICE UNIT
(Fare Product MODEL)
A FARE PRODUCT consisting in a stored value of PRICE UNITs: an amount of money on an electronic purse, amount of units on a value card etc.

3.4
BLACKLIST
(Fare Contract MODEL)
A list of identified TRAVEL DOCUMENTs or CONTRACTs the validity of which has been cancelled temporarily or permanently, for a specific reason like loss of the document, technical malfunction, no credit on bank account, offences committed by the customer, etc.

3.5
BORDER POINT
(Fare Zone MODEL)
A POINT on the Network marking a boundary for the fare calculation. May or may not be a SCHEDULED STOP POINT.

3.6
CANCELLING
(Cancelling Usage Parameters MODEL)
Parameter giving conditions for cancelling of a purchased access right.

3.7
CAPPED DISCOUNT RIGHT
(Fare Product MODEL)
A specialisation of SALE DISCOUNT RIGHT where the discount is expressed as a rule specifying a ceiling for a given time interval. For example, the London Oyster card fare, which charges for each journey until travel equivalent to a day pass has been consumed after which further travel is free at that day.

3.8
CAPPING RULE
(Fare Product MODEL)
A capping limit for a given time interval, where the capping is expressed by another product. For example, the London Oyster card fare, which charges for each journey until travel equivalent to a day pass for the mode of travel has been consumed.

3.9
CAPPING RULE PRICE
(Fare Product MODEL)
A set of all possible price features of a CAPPING RULE: default total price, discount in value or percentage etc.

3.10
CELL
(Fare Table MODEL)
An unique individual combination of features within a FARE TABLE, used to associate a FARE PRICE with a fare element.

3.11
CHARGING MOMENT
(Fare Product MODEL)
A classification of FARE PRODUCTs according to the payment method and the account location: pre-payment with cancellation (throw-away), pre-payment with debit on a value card, pre-payment without consumption registration (pass), post-payment etc.

3.12
CHARGING POLICY
(Charging Usage Parameters MODEL)
Parameter governing minimum amount and credit allowed when consuming a FARE PRODUCT.

3.13
COMMERCIAL PROFILE
(Eligibility Usage Parameters MODEL)
A category of users depending on their commercial relations with the operator (frequency of use, amount of purchase etc.), often used for allowing discounts.

3.14
COMPANION PROFILE
(Eligibility Usage Parameters MODEL)
The number and characteristics of the persons entitled to travel in a group or as companions to another USER PROFILE.

3.15
CONTROLLABLE ELEMENT
(Validable Element MODEL)
The smallest controllable element of public transport consumption, all along which any VALIDITY PARAMETER ASSIGNMENT remains valid.

3.16
CONTROLLABLE ELEMENT IN SEQUENCE
(Validable Element MODEL)
A CONTROLLABLE ELEMENT as a part of a FARE STRUCTURE ELEMENT, including its possible order in the sequence of CONTROLLABLE ELEMENTs grouped together to form that FARE STRUCTURE ELEMENT, and its possible quantitative limitation.

3.17
CONTROLLABLE ELEMENT PRICE
(Validable Element MODEL)
A set of all possible price features of a CONTROLLABLE ELEMENT: default total price, discount in value or percentage etc.

3.18
CUSTOMER
(Fare Contract MODEL)
An identified person or organisation involved in a fare process. There may be a CONTRACT between the CUSTOMER and the OPERATOR or the AUTHORITY ruling the consumption of services.

3.19
DISCOUNTING RULE
(Fare Calculation Parameters MODEL)
A price calculation rule determined by a set of discounts, depending upon a USAGE PARAMETER, to be applied to a FARE PRICE.

3.20
DISTANCE MATRIX ELEMENT
(Distance Matrix Element MODEL)
A cell of an origin-destination matrix for TARIFF ZONEs or STOP POINTs, expressing a fare distance for the corresponding trip: value in km, number of fare units etc.

3.21
DISTANCE MATRIX ELEMENT PRICE
(Distance Matrix Element MODEL)
A set of all possible price features of a DISTANCE MATRIX ELEMENT: default total price etc.

3.22
DISTRIBUTION ASSIGNMENT
(Sales Offer Package MODEL)
An assignment of the COUNTRY and/or DISTRIBUTION CHANNEL through which a product may or may not be distributed.

3.23
DISTRIBUTION CHANNEL
(Sales Distribution MODEL)
A type of outlet for selling of a product.

Eligibility
(Use Case)
The required characteristics of a customer to be able to purchase a product.

3.24
ELIGIBILITY CHANGE POLICY
(Eligibility Parameters MODEL)
Parameter indicating the action to be taken when a user's eligibility status changes.

3.25
ENTITLEMENT CONSTRAINT
(Entitlement Parameters MODEL)
Constraints on choices for an dependent entitled product relative to the required choices for the prerequisite entitling product.

3.26
ENTITLEMENT GIVEN
(Entitlement Parameters MODEL)
Parameter indicating whether a particular FARE PRODUCT provides an entitlement to buy or use an access right.

3.27
ENTITLEMENT PRODUCT
(Fare Product MODEL)
A precondition to access a service or to purchase a FARE PRODUCT issued by an organisation that may not be a PT operator (e.g. military card).

3.28

ENTITLEMENT REQUIRED
(Entitlement Parameters MODEL)
Parameter indicating whether a particular FARE PRODUCT requires an entitlement to by or use an access right.

3.29
EXCHANGING
(Booking Usage Parameters MODEL)
Whether and how the access right may be exchanged for another access right.

3.30
Fare
(Use Case)
From the customer perspective: the amount that a customer has to pay for a journey or for acquiring a product.

3.31
FARE DAY TYPE
(Fare Calculation Parameters MODEL)
A type of day used in the fare collection domain, characterised by one or more properties which affect the definition of access rights and prices in the fare system.

3.32
FARE DEMAND FACTOR
(Quality Fare Structure MODEL)
A named set of parameters defining a period of travel with a given price, for example off peak, peak, super off peak, etc.

3.33
FARE ELEMENT IN SEQUENCE
(Common Fare Structure MODEL)
A FARE ELEMENT as a part of an ELEMENT, including its possible order in the sequence of FARE ELEMENTs.

3.34
FARE FRAME
(Fare Frame MODEL)
The set of all fare data defined for a specific VEHICLE MODE to which the same VALIDITY CONDITIONs have been assigned.

3.35
FARE FRAME DEFAULTS
(Fare Frame MODEL)
Set of pricing parameters and values to apply to an individual element in the frame if no explicit value is specified on the element.

3.36
FARE INTERVAL
(Common Fare Structure MODEL)
An interval based aspect of the fare structure.

3.37
FARE POINT IN PATTERN
(Fare Zone MODEL)
A POINT IN PATTERN which represents the start or end of a FARE SECTION, or a point used to define a SERIES CONSTRAINT.

3.38
FARE PRICE
(Fare Price MODEL)

Price features DEFINED BY DEFAULT characterizing different PRICE GROUPs.

3.39
FARE PRODUCT
(Fare Product MODEL)
An immaterial marketable element (access rights, discount rights, etc.), specific to a CHARGING MOMENT.

3.40
FARE PRODUCT PRICE
(Fare Product MODEL)
A set of all possible price features of a FARE PRODUCT: default total price, discount in value or percentage etc.

3.41
FARE QUOTA FACTOR
(Quality Fare Structure MODEL)
A named set of parameters defining a number of quota fares available of a given denomination.

3.42
FARE SCHEDULED STOP POINT
(Fare Zone MODEL)
A specialisation of SCHEDULED STOP POINT describing a stop with fare accounting and routing characteristics.

3.43
FARE SECTION
(Fare Zone MODEL)
A subdivision of a JOURNEY PATTERN consisting of consecutive POINTs IN JOURNEY PATTERN, used to define an element of the fare structure.

3.44
Fare structure
(Use Case)
Set of parameters that determine the basic tariffs.

3.45
FARE STRUCTURE ELEMENT
(Fare Structure Element MODEL)
A sequence or set of CONTROLLABLE ELEMENTs to which rules for limitation of access rights and calculation of prices (fare structure) are applied.

3.46
FARE STRUCTURE ELEMENT IN SEQUENCE
(Fare Structure Element MODEL)
A FARE STRUCTURE ELEMENT as a part of a VALIDABLE ELEMENT, including its possible order in the sequence of FARE STRUCTURE ELEMENTs forming that VALIDABLE ELEMENT, and its possible quantitative limitation.

3.47
FARE STRUCTURE ELEMENT PRICE
(Fare Structure Element MODEL)
A set of all possible price features of a FARE STRUCTURE ELEMENT: default total price, discount in value or percentage etc.

3.48
FARE STRUCTURE FACTOR
(Common Fare Structure MODEL)
A factor influencing access rights definition or calculation of prices.

3.49

FARE TABLE
(Fare Table MODEL)
A grouping of prices (specialization of PRICE GROUP) that may be associated with all or any of DISTANCE MATRIX ELEMENT, FARE STRUCTURE ELEMENT GEOGRAPHICAL INTERVAL, GROUP OF ACCESS RIGHT PARAMETER, CLASS OF USE, OPERATOR, VEHICLE MODE, FARE PRODUCT.

3.50
FARE UNIT
(Common Fare Structure MODEL)
A unit associated with a FARE STRUCTURE FACTOR.

3.51
FARE ZONE
(Fare Zone MODEL)
A specialization of TARIFF ZONE to include FARE SECTIONs.

3.52
FREQUENCY OF USE
(Travel Usage Parameters MODEL)
The limits of usage frequency for a FARE PRODUCT (or one of its components) or a SALES OFFER PACKAGE during a specific VALIDITY PERIOD. There may be different tariffs depending on how often the right is consumed during the period.

3.53
FULFILMENT METHOD
(Sales Distribution MODEL)
The means by which the ticket is delivered to the CUSTOMER, e.g. online, collection, etc.

3.54
FULFILMENT METHOD PRICE
(Sales Distribution MODEL)
A set of all possible price features of a FULFILMENT METHOD default total price etc.

3.55
GENERIC PARAMETER ASSIGNMENT
(Validity Parameters MODEL)
A VALIDITY PARAMETER ASSIGNMENT specifying generic access rights for a class of products (e.g. a time band limit - 7 to 10 a.m. - for trips made with a student pass).

3.56
GEOGRAPHICAL INTERVAL
(Geographical Fare Structure MODEL)
A geographical interval specifying access rights for the FARE STRUCTURE ELEMENTs within the range of this interval: 0-5 km, 4-6 zones etc.

3.57
GEOGRAPHICAL INTERVAL PRICE
(Geographical Fare Structure MODEL)
A set of all possible price features of a GEOGRAPHICAL INTERVAL: default total price etc.

3.58
GEOGRAPHICAL STRUCTURE FACTOR
(Geographical Fare Structure MODEL)
The value of a GEOGRAPHICAL INTERVAL or a DISTANCE MATRIX ELEMENT expressed by a GEOGRAPHICAL UNIT.

3.59
GEOGRAPHICAL UNIT
(Geographical Fare Structure MODEL)
A unit for calculating geographical graduated fares.

3.60
GEOGRAPHICAL UNIT PRICE
(Geographical Fare Structure MODEL)
A set of all possible price features of a GEOGRAPHICAL UNIT: default total price etc.

3.61
GROUP OF DISTANCE MATRIX ELEMENTs
(Distance Matrix Element MODEL)
A grouping of DISTANCE MATRIX ELEMENTS. May be used to provide reusable Origin / Destination pairs (and associate them with a PRICE).

3.62
GROUP OF DISTRIBUTION CHANNELs
(Sales Distribution MODEL)
A grouping of DISTRIBUTION CHANNELs.

3.63
GROUP OF SALES OFFER PACKAGEs
(Sales Offer Package MODEL)
A grouping of SALES OFFER PACKAGEs.

3.64
GROUP TICKET
(Eligibility Usage Parameters MODEL)
The number and characteristics of persons entitled to travel in addition to the holder of an access right.

3.65
INTERCHANGING
(Travel Usage Parameters MODEL)
Limitations on making changes within a trip.

3.66
LIMITING RULE
(Fare Calculation Parameters MODEL)
Rule for limiting the results of a price calculation.

3.67
LUGGAGE ALLOWANCE
(Luggage Usage Parameters MODEL)
The number and characteristics (weight, volume) of luggage that a holder of an access right is entitled to carry.

3.68
MINIMUM STAY
(Travel Usage Parameters MODEL)
Details of any minimum stay at the destination required to use the product.

3.69
MONTH VALIDITY OFFSET
(Fare Calculation Parameters MODEL)
Days before (negative) or after (positive) the start of the month that a product with a calendar period driven activation becomes valid.

3.70
NETWORK VALIDITY PARAMETER
(Validity Parameters MODEL)
A type of VALIDITY PARAMETER related to the network structure.

3.71

ORGANISATIONAL VALIDITY PARAMETER
(Validity Parameters MODEL)
A type of VALIDITY PARAMETER related to organisational issues.

3.72
PARKING CHARGE BAND
(Parking Tariff MODEL)
Parking charges that describe the cost of using a PARKING or PARKING AREA for a given period.

3.73
PARKING PRICE
(Parking Tariff MODEL)
A specialisation of FARE PRICE that defines the price of a PARKING CHARGE BAND.

3.74
PARKING TARIFF
(Parking Tariff MODEL)
A set of parking CHARGE BANDs that describe the cost of using a PARKING or PARKING AREA.

3.75
PARKING TAX RATE
(Parking Tariff MODEL)
A DISCOUNTABLE FARE PRICE that may be associated with all or any of a USAGE PARAMETER, DISTANCE MATRIX ELEMENT, FARE STRUCTURE ELEMENT or GROUP OF ACCESS PARAMETERs.

3.76
FARE CONTRACT
(Fare Contract MODEL)
A contract with a particular (but possibly anonymous) customer, ruling the consumption of transport services (and joint services). A FARE CONTRACT may be designed for a fixed SALES OFFER PACKAGE (e.g. ticket) or to allow successive purchases of SALES OFFER PACKAGEs.

3.77
FARE CONTRACT ENTRY
(Fare Contract MODEL)
A log entry describing an event referring to the life of a FARE CONTRACT: initial contracting, sales, validation entries, etc. A subset of a FARE CONTRACT ENTRY is often materialised on a TRAVEL DOCUMENT.

3.78
PENALTY POLICY
(Charging Usage Parameters MODEL)
Policy regarding different aspects of penalty charges, for example repeated entry at the same station, not having a ticket etc.

3.79
Post-paid ticketing
(Use Case)
The user is charged sometime after using the transport service (detailed description of process see below).

3.80
PRE-ASSIGNED FARE PRODUCT
(Fare Product MODEL)
A FARE PRODUCT consisting of one or several VALIDABLE ELEMENTs, specific to a CHARGING MOMENT.

3.81
Prepaid ticketing
(Use Case)
The user is charged for either a fare product (ticket) or a deposit prior to riding (detailed description of process see below).

3.82
Price
(Use Case)
Value of fare or tariff.

3.83
PRICE GROUP
(Fare Price MODEL)
A grouping of prices, allowing the grouping of numerous possible consumption elements into a limited number of price references, or to apply grouped increase, in value or percentage.

3.84
PRICE UNIT
(Fare Calculation Parameters MODEL)
A unit to express prices: amount of currency, abstract fare unit, ticket unit or token etc.

3.85
PRICEABLE OBJECT
(Fare Price MODEL)
An element which may have a FARE PRICE.

3.86
PRICING PARAMETER SET
(Fare Calculation Parameters MODEL)
A set of parameters controlling pricing calculations.

3.87
PRICING RULE
(Fare Calculation Parameters MODEL)
A rule used for the calculation of FARE PRICE, determined either by a set of parameters to be applied to a reference price or by a more complex algorithm.

3.88
PRICING SERVICE
(Fare Calculation Parameters MODEL)
A web service used to provide prices dynamically at time of booking or purchase.

3.89
PRODUCT VALIDITY PARAMETER
(Validity Parameters MODEL)
A type of VALIDITY PARAMETER linked to fare products and/or their distribution.

3.90
PURCHASE WINDOW
(Booking Usage Parameters MODEL)
Period in which the product must be purchased.

3.91
QUALITY STRUCTURE FACTOR
(Quality Fare Structure MODEL)
A factor influencing access rights definition or calculation of prices, based on the quality: traffic congestion threshold, early/late reservation etc.

3.92
QUALITY STRUCTURE FACTOR PRICE
(Quality Fare Structure MODEL)
A set of all possible price features of a QUALITY STRUCTURE FACTOR, e.g. default total price etc.

3.93

REFUNDING
(Booking Usage Parameters MODEL)
Whether and how the product may be refunded.

3.94
REPLACING
(Booking Usage Parameters MODEL)
whether and how the access right may be replaced.

3.95
RESELLING
(Booking Usage Parameters MODEL)
Common resale conditions (i.e. for exchange or refund) attached to the product.

3.96
RESERVING
(Booking Usage Parameters MODEL)
indicating whether the access right requires reservation.

3.97
RESIDENTIAL QUALIFICATION
(Eligibility Usage Parameters MODEL)
A parameter providing an authorisation to consider a user as being characterised by a USER PROFILE.

3.98
RETAIL CONSORTIUM
(Retail MODEL)
A group of ORGANISATIONs formally incorporated as a retailer of fare products.

3.99
RETAIL DEVICE
(Retail MODEL)
A retail device used to sell fare products. Its identity can be used to record fulfilment and support security processes.

3.100
ROUND TRIP
(Travel Usage Parameters MODEL)
Properties relating to single or return trip use of an access right.

3.101
ROUNDING
(Fare Calculation Parameters MODEL)
Parameters directing the rounding of values that are the result of calculations.

3.102
ROUNDING STEP
(Fare Calculation Parameters MODEL)
A rounding step to use to round a range of values. If step stable rounding is used, any value larger than the step key and smaller that the next step key should be rounded to this value.

3.103
ROUTING
(Travel Usage Parameters MODEL)
Limitations on routing of an access right.

3.104
ROUTING VALIDITY PARAMETER
(Validity Parameters MODEL)
A type of VALIDITY PARAMETER linked to specific routing.

3.105
SALE DISCOUNT RIGHT
(Fare Product MODEL)
A FARE PRODUCT allowing a customer to benefit from discounts when purchasing SALES OFFER PACKAGEs.

3.106
SALE TRANSACTION
(Sales Transaction MODEL)
A SALE of a FIXED PACKAGE or a SALE of a RELOADABLE PACKAGE.

3.107
SALES NOTICE ASSIGNMENT
(Sales Offer Package MODEL)
The assignment of a NOTICE to a SALES OFFER PACKAGE or a GROUP OF SALES OFFER PACKAGEs.

3.108
SALES OFFER ENTITLEMENT GIVEN
(Sales Offer Entitlement Parameters MODEL)
Parameter indicating whether a particular SALES OFFER PACKAGE provides an entitlement to buy or use an access right.

3.109
SALES OFFER ENTITLEMENT REQUIRED
(Sales Offer Entitlement Parameters MODEL)
Parameter indicating whether a particular SALES OFFER PACKAGE requires an entitlement to by or use an access right.

3.110
SALES OFFER PACKAGE
(Sales Offer Package MODEL)
A package to be sold as a whole, consisting of one or several FARE PRODUCTs materialised thanks to one or several TRAVEL DOCUMENTs. The FARE PRODUCTs may be either directly attached to the TRAVEL DOCUMENTs, or may be reloadable on the TRAVEL DOCUMENTs.

3.111
SALES OFFER PACKAGE ELEMENT
(Sales Offer Package MODEL)
The assignment of a FARE PRODUCT to a TYPE OF TRAVEL DOCUMENT in order to define a SALES OFFER PACKAGE, realised as a fixed assignment (printing, magnetic storage etc.) or by the possibility for the FARE PRODUCT to be reloaded on the TYPE OF TRAVEL DOCUMENT.

3.112
SALES OFFER PACKAGE PRICE
(Sales Offer Package MODEL)
A set of all possible price features of a SALES OFFER PACKAGE: default total price etc.

3.113
SALES OFFER PACKAGE SUBSTITUTION
(Sales Offer Package MODEL)
Information on the preferred substitution of packages with other packages if a quota restricted product is no longer available.

3.114
SALES TRANSACTION FRAME
(Sales Transaction Frame MODEL)
A set of SALES TRANSACTION data elements (SALES TRANSACTIONs, etc.) to which the same VALIDITY CONDITIONs have been assigned.

3.115
SCOPING VALIDITY PARAMETER
(Validity Parameters MODEL)
Grouping of assignments to elements.

3.116
SERIES CONSTRAINT
(Fare Zone MODEL)
An extension of a DISTANCE MATRIX ELEMENT (a cell of an origin-destination matrix for TARIFF ZONEs or SCHEDULED STOP POINTs) expressing a fare distance for the corresponding trip (value in km, number of fare units etc.), constrained to specific routes.SERIES CONSTRAINTs are mainly used for rail fares.

3.117
SERIES CONSTRAINT PRICE
(Fare Zone MODEL)
A set of all possible price features of a SERIES CONSTRAINT: default total price etc.

3.118
SERVICE ACCESS RIGHT
(Fare Product MODEL)
An immaterial marketable element dedicated to accessing some services.

3.119
SERVICE VALIDITY PARAMETER
(Validity Parameters MODEL)
A type of VALIDITY PARAMETER related to service characteristics (e.g. class).

3.120
SPECIFIC PARAMETER ASSIGNMENT
(Sales Transaction MODEL)
A VALIDITY PARAMETER ASSIGNMENT specifying practical parameters during a TRAVEL SPECIFICATION, within a given fare structure (e.g. the origin or destination zone in a zone-counting system).

3.121
START TIME AT STOP POINT
(Quality Fare Structure MODEL)
A time at which a Fare time band (time band peak, off peak) is deemed to begin for trips starting at a particular station.

3.122
STEP LIMIT
(Travel Usage Parameters MODEL)
Geographical parameter limiting the access rights by counts of stops, sections or zones.

3.123
Subscription
(Ue Case)
Purchase of a product by staged payments made on a regular basis.

3.124
SUBSCRIBING
(Charging Usage Parameters MODEL)
Parameter specifying conditions relating to paying for a product by subscription.

3.125
SUPPLEMENT PRODUCT
(Fare Product MODEL)

A PRE-ASSIGNED FARE PRODUCT that will provide additional right when used with (as a complement of) another (reserved seat, second to first class upgrade, etc.). SUPPLEMENT PRODUCT also usually means supplement price.

3.126
SUSPENDING
(Travel Usage Parameters MODEL)
Parameter specifying conditions relating to suspending use of a season pass.

3.127
TARIFF
(Fare Structure Element MODEL)
A particular tariff, described by a combination of parameters. From a planner perspective: the set of discrete elements to be used according to the fare calculation rules to calculate the fare.

3.128
TEMPORAL VALIDITY PARAMETER
(Validity Parameters MODEL)
Grouping of temporal validity parameters.

3.129
THIRD PARTY PRODUCT
(Fare Product MODEL)
A FARE PRODUCT that is marketed together with a Public Transport FARE PRODUCT.

3.130
TIME INTERVAL
(Time Fare Structure MODEL)
A time-based interval specifying access rights for the FARE STRUCTURE ELEMENTs within the range of this interval: 0-1 hour, 1-3 days etc.

3.131
TIME INTERVAL PRICE
(Time Fare Structure MODEL)
A set of all possible price features of a TIME INTERVAL, e.g. default total price etc.

3.132
TIME STRUCTURE FACTOR
(Time Fare Structure MODEL)
The value of a TIME INTERVAL expressed by a TIME UNIT.

3.133
TIME UNIT
(Time Fare Structure MODEL)
A unit for calculating time-based graduated fares.

3.134
TIME UNIT PRICE
(Time Fare Structure MODEL)
A set of all possible price features of a TIME UNIT: default total price etc.

3.135
TRANSFERABILITY
(Booking Usage Parameters MODEL)
The number and characteristics of persons entitled to use the public transport service instead of the original customer.

3.136
TRAVEL DOCUMENT
(Travel Document MODEL)

A particular physical support (ticket, card, etc.) to be held by a customer, allowing the right to travel or to consume joint-services, to proof a payment (including possible discount rights), to store a subset of the CONTRACT liabilities or a combination of those.

3.137
TRAVEL SPECIFICATION
(Sales Transaction MODEL)
The recording of a specification by a customer of parameters giving details of an intended consumption (e.g. origin and destination of a travel).

3.138
TYPE OF CONCESSION
(Usage Parameters MODEL)
A classification of USER PROFILE by type of person eligible to use it.

3.139
TYPE OF FARE PRODUCT
(Fare Product MODEL)
A classification of FARE PRODUCTs.

3.140
TYPE OF FARE CONTRACT
(Fare Contract MODEL)
A classification of FARE CONTRACT.

3.141
TYPE OF FARE CONTRACT ENTRY
(Fare Contract MODEL)
A classification of FARE CONTRACT ENTRYs.

3.142
TYPE OF RETAIL DEVICE
(Retail MODEL)
A classification of RETAIL DEVICEs.

3.143
TYPE OF SALES OFFER PACKAGE
(Sales Offer Package MODEL)
A classification of SALES OFFER PACKAGEs.

3.144
TYPE OF TARIFF
(Fare Structure Element MODEL)
A classification of TARIFFs to express the different classes of fares.

3.145
TYPE OF TRAVEL DOCUMENT
(Travel Document MODEL)
A classification of TRAVEL DOCUMENTs expressing their general functionalities and local functional characteristics specific to the operator. Types of TRAVEL DOCUMENTs like e.g. throw-away ticket, throw-away ticket unit, value card, electronic purse allowing access, public transport credit card etc. may be used to define these categories.

3.146
TYPE OF USAGE PARAMETER
(Usage Parameters MODEL)
A classification of USAGE PARAMETERs to express the nature of parameters.

3.147

USAGE DISCOUNT RIGHT
(Fare Product MODEL)
A FARE PRODUCT allowing a customer to benefit from discounts when consuming VALIDABLE ELEMENTs.

3.148
USAGE PARAMETER
(Usage Parameters MODEL)
A parameter used to specify the use of a SALES OFFER PACKAGE or a FARE PRODUCT.

3.149
USAGE PARAMETER PRICE
(Usage Parameters MODEL)
A set of all possible price features of a USAGE PARAMETER: discount in value or percentage etc.

3.150
USAGE VALIDITY PERIOD
(Travel Usage Parameters MODEL)
A time limitation for validity of a FARE PRODUCT or a SALES OFFER PACKAGE. It may be composed of a standard duration (e.g. 3 days, 1 month) and/or fixed start/end dates and times.

3.151
USER PROFILE
(Eligibility Usage Parameters MODEL)
The social profile of a passenger, based on age group, education, profession, social status, sex etc., often used for allowing discounts: 18-40 years old, graduates, drivers, unemployed, women etc.

3.152
VALIDABLE ELEMENT
(Validable Element MODEL)
A sequence or set of FARE STRUCTURE ELEMENTs, grouped together to be validated in one go.

3.153
VALIDABLE ELEMENT PRICE
(Validable Element MODEL)
A set of all possible price features of a VALIDABLE ELEMENT: default total price, discount in value or percentage etc.

3.154
VALIDITY PARAMETER ASSIGNMENT
(Validity Parameters MODEL)
An ACCESS RIGHT PARAMETER ASSIGNMENT relating a fare collection parameter to a theoretical FARE PRODUCT (or one of its components) or a SALES OFFER PACKAGE.

4 Symbols and abbreviations

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

5 Use Cases for Fare Exchange

5.1 Purpose

This section describes possible use cases for the application of NeTEx Part 3. The use cases should help a more precise understanding of the scope of the standardization work for fare exchange. Use cases contained in this section depend on the concepts from NeTEx Part 1 & 2.

5.2 Business context

5.2.1 Fare planning process

5.2.1.1 Fare structures and fare products

The fare planning use cases are divided into two main functional areas, depending on the type of business process or the type of information that is involved:

- Provision of **Fare Structure** information:

  This considers the creation of information regarding the Fare Structure, i.e. the rules and their parameters used to determine the qualitative, quantitative and pricing conditions for accessing public transport.

  A distinction is made between distance, geographic unit and time-based fares. Together they can be regarded as the addition of the notion of fares to the PT network, and may involve some network related elements, for example the choice of Tariff Zones or border points. In addition, specific use cases are presented that consider the creation and setting of general concessionary fares.

- Provision of **Fare Product** information:

  This considers the creation of Fare Product information.

  A Fare Product is generic description of a set of marketable access rights, e.g. single way ticket from A to B or free travel in 3 adjacent zones. The generic description could be defined in a form of product templates that are used to market access rights.

  Fare Products are entities that can be marketed and sold to and used by PT travellers to obtain access rights and by PT operators to validate access to PT services. Fare Product determined fares are based on the Fare Structure. Possession of a Fare Product gives a specific right to access a PT service. A Fare Product defines when a traveller is charged, either when buying the product, when enacting the access right or a combination of both. I.e. the possession of a Fare Product enables the traveller to use PT services and to be charged for this use.

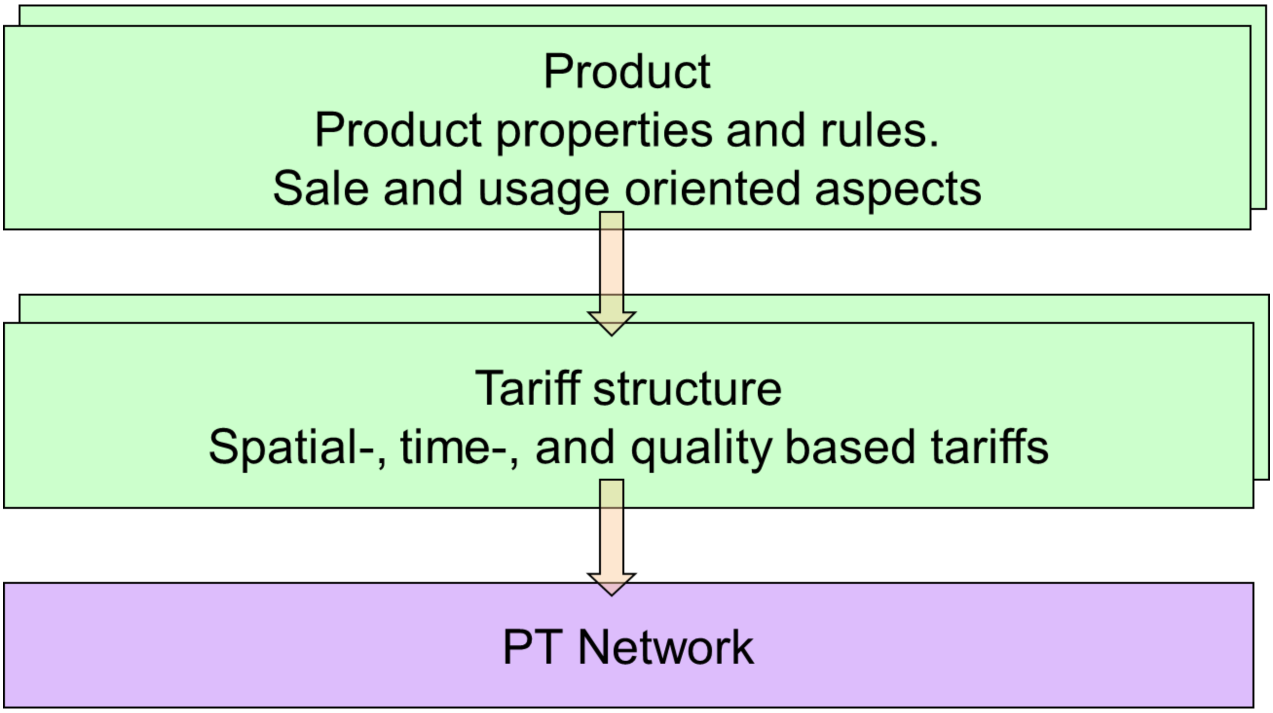


Figure 1 — PT fare overview (Schematic)

Separating the concerns of Fare Structure and Fare Product has significant advantages both for business processes and for technical implementation:

- It enables PT authorities and operators to define and utilise marketable travel products that are based on a reusable Fare Structure, but distinct. It decouples the maintenance of fare products from the maintenance of the fare structure. E.g. annual changes in Fare Structure do not cause a need for drastic changes in products. Special event related products that provide a reduction with respect to the Fare Structure can be defined, sold and used without a need to change the Fare Structure itself at all.

- It enables the definition of simple fare structures and simple fare products that in combination are able to support a wide variety of business requirements.

5.2.1.2 Fare price

In both the Fare Product and the Fare Structure the actual price values are kept logically distinct from the elements that they price. This permits the use of successive revised prices sets with the same fare product and fare structure, and the separate exchange of prices and price updates.

Depending on national or regional law or business models that can be specific to operator or travel mode (e.g. heavy rail), prices can either have a static or dynamic nature.

5.2.1.2.1 **Static prices**

The tariffs and prices have a static nature and are publicised along with the underlying structure. This information has a fixed start and end date validity within which tariffs and prices are static. Depending on the use case, either the discrete prices of underlying features may be exchanged along with the pricing parameters, or a complete set of pre-computed resolved fares for every allowed combination of features (for which the pricing parameters merely provide justification).

5.2.1.2.2 **Dynamic prices**

In situations where PT operator competition is significant, and where the necessary advanced systems are in place, tariffs and prices can be varied dynamically. Such tariffs and prices can depend on market or yield based business rules. Prices are obtained by querying a pricing engine for a given set of criteria (e.g. origin destination, type of user) and only an overall current price is disclosed. The calculation process may be opaque and publication of the full list of current prices and the fare calculation algorithms can be restricted.

In either case, in order for the user to select a product for pricing and understand its applicability to her needs, information still needs to be available about the available products in relation to the journeys they price including their pre and aftersales conditions.

The use cases for definition of a dynamic price based fare policy are outside the scope of the NeTEx standard. Use cases that define the organisation of the use of market prices are within the NeTEx scope. However, in case a fare structure is used as a basis for a market price fare policy, these NeTEx use cases can be used internally. The Fare Query model included provides an informative demonstration of how NeTEx parameters can be mapped to typical dynamic systems.

5.2.1.3 Overall fare planning process

A fare planning process will consider the effects of different possible procurements and prices for tickets. Consequently the process requires modelling of expected traffic levels against the putative fare structure to decide where to place fare boundaries, what products to offer and what prices need to be set to obtain satisfactory returns on investment. This may be an iterative process that must also consider other factors such as network congestion and existing retail infrastructure and ticket validation methods. Sometimes social policy drives the offering of certain products – for example the provision of free passes for veterans or the elderly must also be modelled and costed – since they must be accounted and paid for even if “free” to the user. Although for purposes of exposition the following use cases break the fare policy processes down into discrete steps and separate the development of fare structures from fare products, in reality the two are closely interdependent and development involves the iterative consideration of the overall picture.

The choice of use cases also depends on the degree of transparency of the pricing process being offered. Where only the final resolved prices for every possible combination are exposed (say as a fully populated fare table) it is not necessary to expose all the underlying pricing factors, dependencies and discounts used to compute them. Only a final set of computed fares need be disclosed. In other cases, it is desirable to exchange a set of base fares with the necessary fare structure elements and pricing parameters to derive prices for all the dependant elements. NeTEx can be used for both cases (i.e. “Fare Table Exchange” and “Fare Structure Exchange”).

The fare planning process can be represented by a sequence of workflows (high level use cases):

1. Describe overall fare structure type (rules, parameters determining the access rights).

2. Determine spatial/temporal/quality parameters of the fare structure: (i) whole network, whole line (i.e. flat fares), (ii) point to point (ii) zonal (ii) fare stage (iv) sequence of stages (iv) composite variants, (v) time intervals, etc.

3. Determine sequences of fare structure elements grouped together to be validated in one go (“validable elements”: e.g. bus ride, train ride, metro trip, metro ride, etc.

4. Choose limitation parameters for access rights e.g. round trip allowed, specific user profiles, group tickets, etc.

5. Determine fare products, i.e. “marketable elements” classified according to the “charging moment”, e.g. pre-paid monthly pass, post-paid monthly pass, etc.

6. Add pricing rules and price parameters.

7. Test these against predicted yields and iterate steps 1-6.

8. Choose travel documents and distribution channels.

9. Add deriving prices.

10. Choose sales offer packages to be marketed to customers.

11. Derive final prices.

However, in reality it is unusual to design a whole fare structure from scratch as a new workflow, or even to do this in order to create a new electronic representation. Rather most aspects of the system will be dictated by the existing network and its fare products, fare collection and fare management systems. Processes are required to make updates to the existing representations of the system (stops, routes, services, products, conditions etc) and to manage and update the prices for the available fares.

The process and described above may be used as well to represent the main steps to represent an existing situation (fare structure, fare products, prices and all the linked concepts).

5.2.1.3.1 **Workflow for creation and maintenance of Fare Structures**

The workflow for creation and maintenance of fare structure requires the choice of a fare model that suits the characteristics of the mode and network (e.g. urban mass transit, long distance rail, rural bus etc., etc.); the available channels for ticket retailing and the viable methods of ticket validation, and the required business yields for the expected (and likely fraud) levels. Fare models between different regions, operators and modes will be significantly different.

The definition of a fare structure is based in effect on generic quantitative rules that influence the access rights regulating the consumption, together with the price a passenger has to pay for a specific consumption factor: limitation of the duration or the length of a trip, price based on the number of zones crossed, etc.

These rules describe the use of the transport system in terms of space, time and service quality. Therefore, space-based, temporal and quality factors may need be specified.

A particular version of all the fare structure, i.e. of the rules and associated parameter values will be called tariff in this standard.

A fare structure typically contains base prices optionally enhanced with price differentiations for modality, user profiles, peak / off-peak hours, seat class, etc.

Fares can be flat, i.e. not depending on consumption factors, or progressive, that is increasing in proportion to consumption.

Fare structures differ greatly in their complexity, ranging from a single flat fare to a multi-dimensional matrix of factors connected by a complex object model. The fare model may contain objects such as tariff zones and fare stages that need to be collated with network data such as stop points, and availability restrictions that need to be collated with demand on specific routes. An effective date and (geographical) validity scope will be assigned to the fare structure overall.

The output of all the processes will be fare tables in structured form.

5.2.1.3.2 Workflow for fare calculation and ticket charging

In general, the purchase of a fare product breaks down into several successive steps. This is of particular importance for electronic ticketing systems providing automated fare calculation, but a similar flow for conventional ticketing system processes is observed in the following sections.

5.2.1.3.2.1 Selection of fare product

Whether the selection of a fare product is done prior to a trip, directly after finishing a trip or even later is fundamental to the fare charging process.

In a conventional ticketing system with printed paper tickets or magnetic strip tickets, the fare product, (e.g. a single ride ticket or a season ticket) is selected by the traveller before starting a trip.

The same process may be followed in an electronic ticketing system, where the ticket is provided in electronic form by means of a data set stored on a smartcard or in a back-office system. The traveller may deliberately select a fare product in advance, e.g. a flat rate for a limited region and / or limited validity period, which entitles him to unlimited travelling within the chosen scope (in conventional ticketing systems this is a period pass season ticket).

However, in an electronic ticketing system with automated fare product selection (AFC system), the traveller actually only needs an access right to enter the vehicle and start a trip. No fare product selection need necessarily be done at the start of a journey. The selection of the product used for a specific trip (and its accounting) is done by the AFC system after completion of the trip. For example, the AFC system will automatically distinguish whether the dedicated trip is done within the coverage of a season ticket product held by the user, such as a single trip within the limiting scope, or another product), e.g. a single trip in a different zone, or a mixture of existing and additional access rights.

Moreover, if the tariff includes price capping that limits the total fare cost within a given period to a specified limit, the choice of products used to account for the fares may be modified by subsequent travel. For example, after the third ride with a single ride ticket at the same day the previously selected products (three single ride tickets) may be converted to a day ticket.

5.2.1.3.2.2 Price calculation

Thus, the calculation of a ticket price may be done:

- Either prior to the beginning of a trip (conventional ticketing system)
- At the end of a trip (or part of a trip) (automated fare product selection and price calculation system)
- Or sometime later to incorporate rebates like a fare cap or volume discounts.

5.2.1.3.2.3 Charging of ticket price

After making the price calculation, the price can be charged to the customer in a number of different ways. A general distinction is made between **prepaid ticketing** and **post-paid** ticketing.

Prepaid Ticketing

In a conventional ticketing system with preselected products (paper based or electronic) the price will be charged directly after the price calculation by various means, e.g. a self-service ticket machine or at a sales counter.

Smartcard based electronic ticketing systems often require the user to pay an amount of money as a deposit in advance to riding. This is the case for stored value cards, where the deposit is registered on the smart card, but can also be achieved by linking the smartcard to an online system in the cloud that holds the deposit.

In conventional electronic ticketing systems, deposits are used to pay the preselected tickets, so the process is very similar as paying with cash but without the necessity of cash-handling for every transaction. The smartcard may also be enabled as a payment device so that it links directly to a credit or debit card to make a payment for the prepaid travel at the time of purchase (as is the case of a travel payment application actually in a smart credit card).

A refinement (applicable to both prepaid and post-paid ticketing) is to enable a stored value card for automated top-up so that it will be recharged by a credit from a bank or credit card account at regular intervals or at a predefined credit trigger threshold.

Post-paid Ticketing

Post-paid ticketing systems are characterised by a sales transaction made by the public transport operator on completion of the journey – this may be either a debit for the full amount, or an adjustment against a prepaid deposit charged at the start of the journey. In both cases accounting is only completed after riding.

For stored value cards, the debit may be made against the store. For travel cards that allow credit (including for example the use of nfc credit cards to pay fares) the amount will later be charged to the user by deducting a bank or credit card account.

Stored value cards may be anonymous (i.e. the operator does not know who the owner is other that a passenger can show the card as proof of a ticket) or personal – i.e. registered with the operator, and traceable to an individual. Usually such cards are non-transferable and supported by a photo or other verification mechanism.

To summarise: for prepaid ticketing the user is charged prior to riding for a ticket; for post-paid ticketing the user is charged on completion of the ride (but may also have been required to make a pre-paid deposit when they started the journey).

5.2.1.4 Further evolution of fare systems

Fare structures for complex networks represent a significant investment and once established tend to be relatively stable, with only evolutionary changes to services and routes within an established overall structure., for example the addition of new stops or services on existing routes. There is a continuing need however to periodically re-price existing fares, and it is therefore important to have separate price data with well-defined validity.

There is also a continuing need to introduce new fare products to increase yields from the network; these will typically be constrained to the existing fare collection infrastructure but may products with specific limitations targeted at different users and different travel times, for example promotion passes for tourists. Many operators thus support set two sets of products: a regular set of standard products and fare offerings covering all the network, and an ad hoc set of promotional products intended to generate additional custom at particular times. These will normally have more restrictive commercial conditions.

5.3 Actors and use case types

The following table gives an overview of information technology systems that are likely to use the NeTEx Part3 interface. The “Producer” and “Consumer” columns indicate whether the systems will provide or receive the information content. In the last column examples for organisations are given that might operate such systems. The list in this table is not complete and may be extended.

Table 1 — NeTEx Part3 actors

Systems	Producer	Consumer	Organisations
Service tendering and timetable registration systems: NeTEx Part 3 compliant fare structure information (i.e. tariffs, tariff zones)	x	x	Public transport authorities
Fare Management and Planning Systems: NeTEx Part 3 compliant entities for determination of pre- and post- sale conditions (i.e. generic parameter assignment for attaching conditions for the fare product retail process)	x	x	Public transport authorities, public transport operators
Journey planning systems: NeTEx Part 3 compliant fare product information	x	x	Public transport authorities, public transport operators
Ticket selling systems: NeTEx Part 3 compliant price entities (i.e. sales offer packages, fare products, validable element, usage	x	x	Public transport authorities, public transport operators

parameters, access rights, etc.) and customer entities (i.e. fare contract, customer, user profile) that accompany route information (NeTEx Part 2).			
Automated fare collection (AFC) systems (automatic ticket barriers (gate, turnstile) with fare collection (mechanical, infrared, magnetic)): NeTEx Part 3 compliant travel document, fare product data, access rights in product, access right parameter assignments, border point, blacklists.	x	x	public transport operators
Passenger information systems (i.e. on-board fare displays): NeTEx Part 3 compliant fare price entities for devices (fare prices charged at usage)	x	x	public transport operators
Demand responsive transport (DRT) or Dial-a-Ride Transit (DART) systems: NeTEx Part 3 compliant graduated time-based fare structure entities (i.e. time interval price, time structure factor) and graduated geographically-based fare structure entities (i.e. geographical unit, geographical interval, border point) for (non-)public, (non-)commercial mass transport.	x	x	Demand responsive transport operators
Mapping systems: NeTEx Part 3 compliant price entities for public transport journey routes		x	Commercial and non-commercial services providers like Google Maps, Yellow Pages, etc.
Strategic planning systems: NeTEx Part 3 compliant price entities for planning and decision making in the development of public transport: controllable element price, fare structure element price, geographical interval price, distance matrix price, time interval price, validable element price, usage parameter price, fare product price, sales offer package price.		x	All levels of public transport authorities
General third-party systems		x	Various

The use cases are subdivided into two main groups:

- A set of use-cases that aim to define **a fare policy.** These use cases comprise the definition of the basic features of a **fare structure** and the definition of **fare products** that use the basic fare structure to determine the sale and usage prices.

- A set of use cases that aim to **organise usage** of the **fare policy for passenger information on fares**, fare information to feed sales channels and fare information to feed other operational processes such as validation. These usage processes are partially executed by IT devices. Specific information to configure these devices, such as security sensitive data, are outside the scope of NeTEx.

5.3.1 Use Cases for Fare Policy

The fare policy use cases define the structural elements (fare structures) of a fare (network scope, basic geographical and temporal factor, fare price) and fare products (access rights, entitlements).

5.3.1.1 Provide data for determination of fare network scope (part of fare structure)

FARE-001-NETWORK-X: use cases for determination of relevant PT network scope structures that apply to a fare.

5.3.1.2 Provide data for determination of fare price (part of fare structure)

FARE-001-PRICING-X: use cases for determination of fare price structure parameters.

5.3.1.3 Provide data for determination of basic fare tariffs (part of fare structure)

FARE-001-TARIFF-X: use cases for determination of basic tariff-based fare structure (network) elements.

5.3.1.4 Provide data for determination of fare products

FARE-001-PRODUCT-X: use cases for determination of fare product, which gives the PT user right to access and use PT transport.

5.3.2 Use Cases for Organisation of Fare Policy Usage

5.3.2.1 Provide data for distribution of information about fare products

FARE-002-INFORMATION-X: use cases that describe the organisation and distribution of fare data to downstream systems such as journey planners.

5.3.2.2 Provide data for organisation of fare products sales

FARE-002-SALES-X: use cases for distribution of fare parameters to sale systems.

5.3.2.3 Provide fare data for support of PT operations

FARE-002-OPERATIONS-X: use cases for distribution of fare data for operational support.

5.4 Excluded Use Cases

Examples that illustrate what is out of the scope of NeTEx.

Table 2 — Excluded Use Cases

Excluded use cases or business domains	Reason for exclusion
Provision of fare information to validation devices	Specific provision of fare information to validation devices is out of NeTEx scope. NeTEx Part3 provides entities for

	validation process (i.e. VALIDABLE ELEMENTs like ACCESS RIGHT IN PRODUCT) in the Fare Product MODEL, which may be consumed by validation devices.
Provision of fare information to sale devices	Specific provision of fare information to sale devices is out of NeTEx scope. NeTEx Part 3 provides entities for RETAIL model, which identifies RETAIL CONSORTIUMs, ORGANISATIONs who sell products, and RETAIL DEVICEs used to sell products.
Customer registration and identification	Specific support for the management of registration and identification of customers in public transport is out of NeTEx scope. NeTEx Part 3 provides entities within TRAVEL DOCUMENT MODEL, which enable pairing of a TRAVEL DOCUMENT with a CUSTOMER's identity at the time of a SALE TRANSACTION.
Purchasing and fulfilment	NeTEx supports only basic purchasing and fulfilment parameters like how a product may be purchased (DISTRIBUTION CHANNEL, i.e. online) and how a purchase is subsequently delivered (FULFILMENT METHOD, FULFILMENT METHOD PRICE, i.e. Self-printing). The rest of the purchasing and fulfilment processes is out of NeTEx scope.
Calculation of fare prices	Specification of entities that would model algorithms for fare price calculations are out of NeTEx scope. NeTEx provides only entities for exchange of product price information.

5.5 Use Cases

5.5.1 Collection of Use Cases

The use cases in this section describe exchange of fare-related information. They are compliant with the use case categories for fare exchange identified in the NeTEx Part1 document and subsequently to functional areas in Transmodel:

- FARE-001-X: Definition of fare policy (fare structure, fare products) / (Transmodel: AREA 4)
- FARE-002-X: Organisation of fare policy usage (fare policy implementation, tickets) / (Transmodel: AREA 21)

The use cases are not directly NeTEx use cases. The following tables describe how NeTEx is used to facilitate these use cases and which requirements for NeTEx originate from them.

Use cases number 1-X originate primarily from FareXChange, numbers X - 46 from IFOPT.

The numbering may have gaps because of removal of use cases. (Numbers are currently persistent.)

5.5.1.1 Provide data for definition of fare network scope (part of fare structure)

The use cases are denoted as FARE-001-NETWORK-X where X is the ordered number

5.5.1.1.1 **Use Case: Determination of basic fare structure network scope**

Use Case: FARE-001-NETWORK-001 (#1)	
Name	Determination of basic fare structure network scope.
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	Tariffs will apply to a given network scope that is defined by the PT Operator, domain, line, group of lines, mode, tariff zones, etc., based on the network characteristics Define the scope for the tariff structure in terms of: • **Geography**: network domain, group of lines, line. • **Modes**: the mode or modes the tariff is valid for within the geographical scope. The scope can be defined at alternative abstraction levels: • For the entire tariff structure and implicitly for the tariff information components within. For example, the zones of a metro network. • Explicitly for the information components within the tariff structure, e.g. for point to point fares described in a Distance Matrix.
NeTEx contribution	NeTEx provides a means to exchange a representation of the network and subsequent changes to it. The same elements are used for network and timetable descriptions in NeTEx Part1 and 2, so the fare structure can be related to the timetabled journeys.
Main actors	Fare planning systems.
Main objects	Part1: TRANSPORT MODE, LINE, LINE SECTION, SCHEDULED STOP POINT, TARIFF ZONE, VALIDITY CONDITION, TOPOGRAPHICAL PLACE. Part2: VEHICLE JOURNEY, SERVICE JOURNEY, GROUP OF SERVICEs, TYPE OF SERVICE, TYPE OF PRODUCT CATEGORY. Part3: FARE ZONE, FARE SECTION, BORDER POINT, DISTANCE MATRIX ELEMENT, GROUP OF DISTANCE MATRIX ELEMENTs.

Some examples of validity scope are given below:

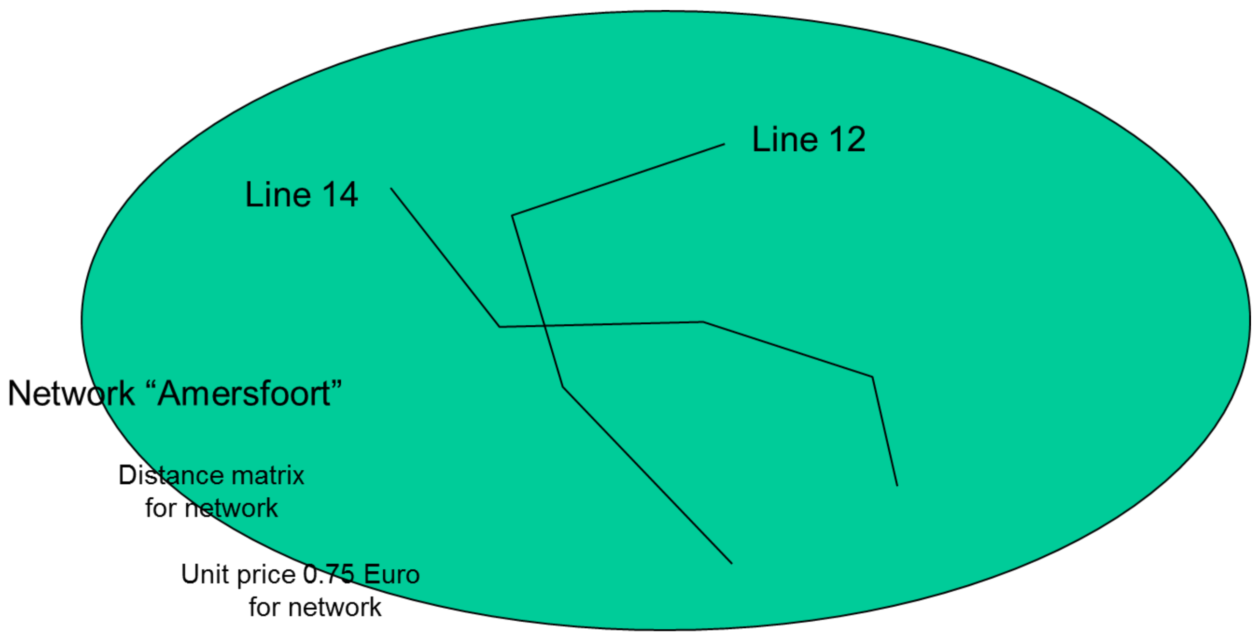


Figure 2 — Example of a fare structure that contains an OD matrix and a unit price (EXM)

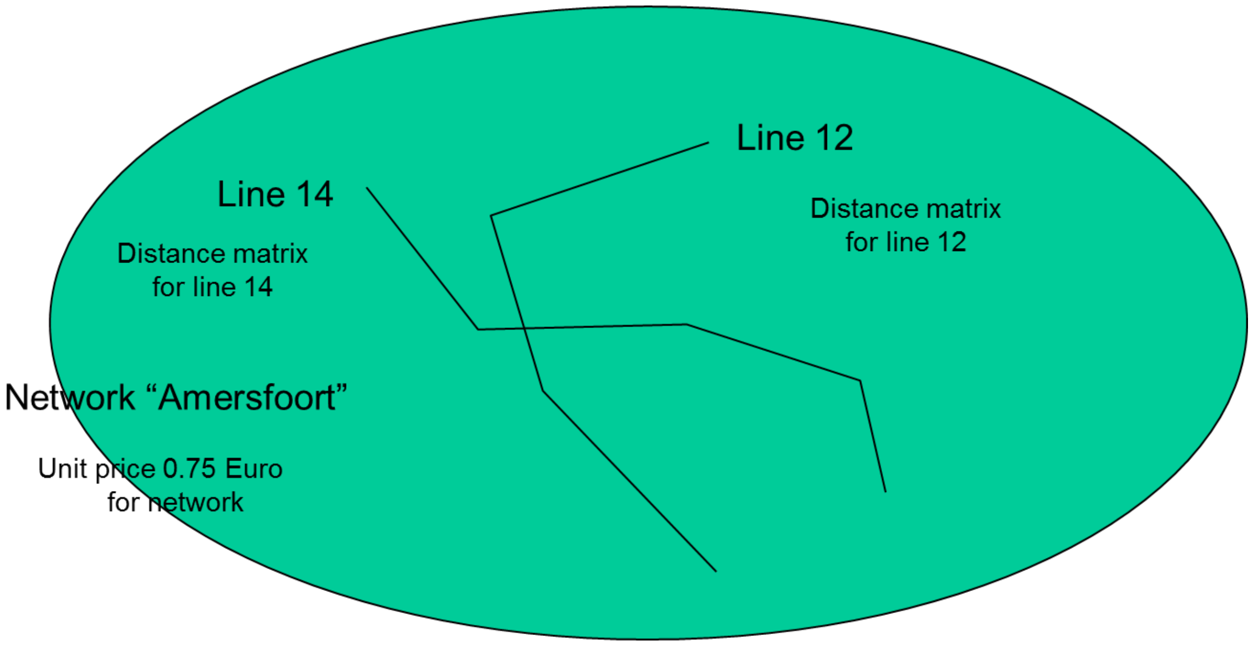


Figure 3 — Example with OD matrices for each line within and a unit price for the entire network (EXM)

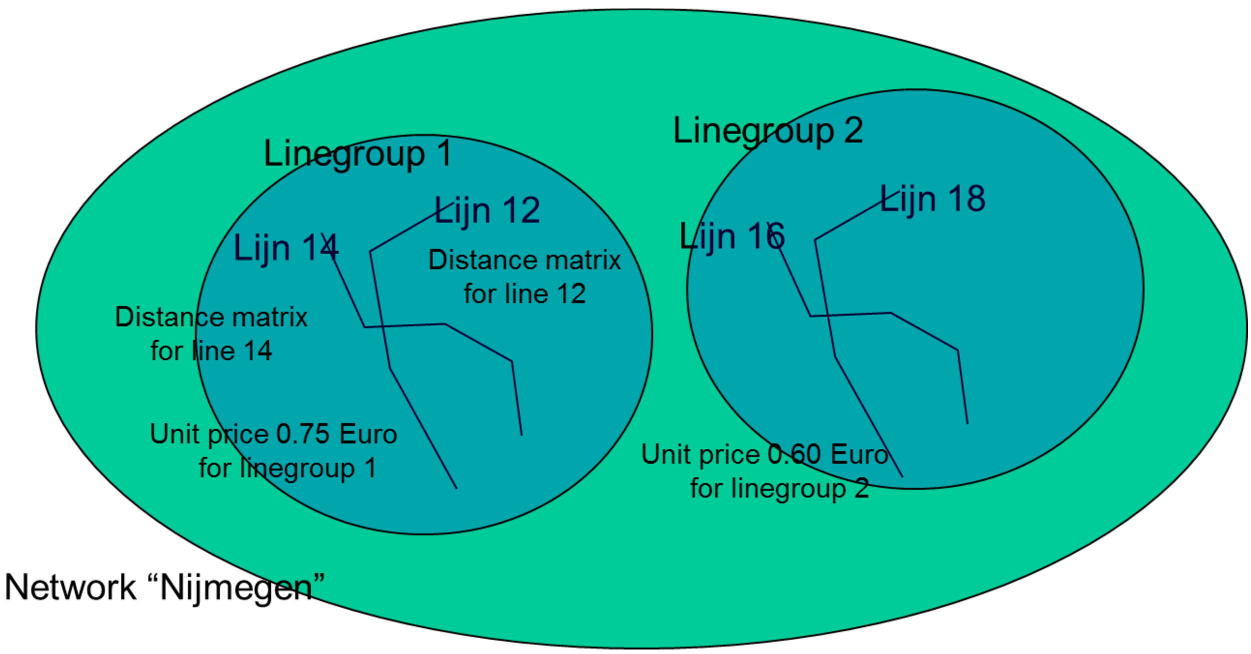


Figure 4 — Example with line groups, OD matrices for each line and a unit price for each line group. (EXM)

5.5.1.1.2 Use Case: Demarcating Fare stages or Fare zones / honeycombs

Use Case: FARE-001-NETWORK-002 (#2)	
Name	Demarcating Fare stages or Fare zones / honeycombs
Source	Transmodel, FareXChange, BISON, NeTEx
Description	In the case where geographically defined space-based factors are used, such as zones or stages, they need to be defined explicitly. A zone or comb can be seen as a layer on stops. Within a stage, zone (or comb) a tariff is flat. A zone or comb can be part of an OD matrix to define tariffs between them. A stage is a layer on a route (or several routes) that defines geographical factors that depend on the start of a journey, e.g. 0 to 3 stops ahead, 4 to 6, etc. ***<u>Fare zones and combs:</u>*** A comb is equivalent to a zone for the level of definition that is required here [2]. Zonal fares require the geographic delineation of a fare zone. Each fare zone will have a different name or number within the transport network. Each zone will be bounded by a polygon or more complex shape e.g. torus. Fare zones may overlap, i.e. a stop may be in two zones. Fare zone can be concentric. Fare zone can form hierarchies containing sub- and super- zones. Define the Fare Zones in terms of: • **Explicitly**: every single stop in the network is directly 'attached' to the named zone. This representation is used for fare calculation purposes. • **Indirectly**: Zone areas are described by a polygon and fares are considered to be in the zone if they reside within that boundary. This representation is used for presentation purposes, e.g. on maps. • **Hierarchy**: a zone can optionally be attached to a sub- or super- zone. E.g. the Berlin combs are part of the bigger zone in which they reside. ***<u>Fare stages:</u>*** Fare stages are geographical points that demarcate a transition boundary for charging a fare. A stage may be at a stop, or between stops.
NeTEx contribution	NeTEx provides a means to exchange the zone and stages and subsequent updates. Common Network Definition elements from Part1 are reused, with fare related properties attached to them.
Main actors	Fare planning systems.
Main objects	Part1: SCHEDULED STOP POINT, TARIFF ZONE, LINE, LINE SECTION, POINT IN JOURNEY PATTERN, Part3: FARE ZONE, FARE SECTION, FARE STRUCTURE ELEMENT, BORDER POINT, FARE SCHEDULED STOP POINT. GEOGRAPHICAL INTERVAL

5.5.1.1.3 Use Case: Define routes and transfer points

Use Case: FARE-001-NETWORK-003 (#3)	
Name	Define routes and transfer points
Source	Transmodel, FareXChange, BISON, TAP/TSI

[2] For instance, the comb structure as used in Berlin is similar to the zone structure that is used in the Netherlands. The zone structure as used in Berlin is quite different geographically to the zones in the Netherlands. However, they can all be defined in terms of groups of points or polygons and the fact that the tariff is flat within a zone or comb is also a common feature.

Description	In some systems, in particular long-distance rail, the fares will depend on the routes taken or the transfers points used. There may be different prices for alternative routes and / or alternative transfer points For the fare structure the available routes may be specified as constraints on point to point fares
NeTEx contribution	NeTEx provides a means to exchange the actual routes and transfer points and subsequent updates.
Main actors	Fare planning systems.
Main objects	SERIES CONSTRAINT, FARE POINT IN PATTERN ROUTING usage parameter

5.5.1.1.4 **Use Case: Projections of Fare elements on maps**

Use Case: FARE-001-NETWORK-004 (#4)	
Name	Projections of Fare elements on maps
Source	Urban fare systems
Description	Visualisations of the zones, fare stages and or routings will be needed for passenger information, for example on maps. These may apply to whole areas, or to specific sections of route Stops and zones may require include projections onto different coordinate systems with colours and symbols.
NeTEx contribution	The NeTEx representation can include projections onto different coordinate systems.
Main actors	Fare planning systems.
Main objects	Part1: SCHEDULED STOP POINT, TARIFF ZONE, LINE, LINE SECTION, POINT, LINK, ZONE, POINT PROJECTION, LINK PROJECTION, ZONE PROJECTION, SCHEMATIC MAP, LOCATION SYSTEM Part3: FARE ZONE, FARE SECTION, FARE SCHEDULED STOP POINT, SERIES CONSTRAINT, FARE POINT IN PATTERN

5.5.1.1.5 **Use Case: Determination of the basic fare structure factors**

Use Case: FARE-001-NETWORK-005 (#5)	
Name	Determination of the basic fare structure factors
Source	Transmodel, FareXChange, BISON, TAP/TSI
Description	Determination of the basic fare structure factors. A fare structure factor represents the basic quantitative value of consumed PT service for which tariffs can be defined. A quantitative value can consist of space, time, qualitative factors or any combination. Define the fare structure factor in terms of:

	• **Space based.** The most common fare structure rules are space-based, more precisely distance-based. A space-based factor defines a value of distance, represented in a distance unit (e.g. KM), in stops, zones, stages, sections or combinations thereof. and / or • **Time based.** The time-based factors are described in a similar way to the space-based factors. A time-based factor describes values of time for which a tariff can be defined. A time-based factor is a value based on travel time, e.g. minute, hour. and / or • **Quality based factor.** Another way to represent PT consumption is by defining quality factors. Quality-based factors describe the quality categories that are experienced while travelling. For instance, the current level of congestion or occupancy (e.g. in %) may influence the tariff.
NeTEx contribution	NeTEx provides a means to exchange the factors between systems. The factors may be chosen though a planning system or identified from an existing system.
Main actors	Fare planning systems.
Main objects	PART1: FACILITY SET PART3: GEOGRAPHICAL STRUCTURE FACTOR, GEOGRAPHICAL INTERVAL, GEOGRAPHICAL UNIT, TIME STRUCTURE FACTOR, TIME INTERVAL, TIME UNIT, QUALITY STRUCTURE FACTOR, TIME DEMAND FACTOR. USAGE PARAMETERs: ROUND TRIP, ROUTING, FREQUENCY OF USE, INTERCHANGING, USAGE VALIDITY PERIOD, etc

5.5.1.2 Provide data for definition of fare pricing (part of fare structure)

The use cases are denoted as FARE-001-PRICING-X where X is the ordered number.

5.5.1.2.1 Use Case: Determination of price currency and unit

Use Case: FARE-001-PRICING-001 (#6)	
Name	Determination of price units and amounts
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	Prices must be given in a specific currency or units • Define the **currency** or currencies in which the prices are represented. E.g. EURO, GBP. • Define the **unit** the prices are represented in. E.g. 0.01, 1 journey The currency and unit can be defined at different levels of specificity: • For the entire tariff structure and implicitly for the tariff information components within.

	• Explicitly for the information components within the tariff structure, e.g. for a DISTANCE MATRIX or for individual DISTANCE MATRIX ELEMENTs. It is possible to have alternative prices in different currencies.
NeTEx contribution	NeTEx provides a means to exchange the actual currencies and units.
Main actors	Fare planning systems.
Main objects	PRICE UNIT, FARE PRICE. PRICE GROUP, FARE TABLE

5.5.1.2.2 Use Case: Setting rounding and calculation factors

Use Case: FARE-001-PRICING-002 (#7)	
Name	Setting rounding and calculation factors
Source	FareXChange, BISON, TAP/TSI
Description	Where fare calculations take place to derive a fare from base factors, or to derive a fare from another fare, the results may need to be rounded to the nearest viable currency amount, for example 50 cent intervals. Rounding factors can be set for whole fares Another datum of relevance is the actual start and end times of a fare day.
NeTEx contribution	NeTEx provides a means to exchange the actual tariff and different versions of it.
Main actors	Fare planning systems.
Main objects	ROUNDING, ROUNDING STEP, FARE DAY.

5.5.1.2.3 Use Case: Market segmentation and user eligibility criteria

Use Case: FARE-001-PRICING-002 (#8)	
Name	Market segmentation and user eligibility criteria.
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Particular products may be targeted at specific groups of users: for example, children, seniors, the disabled, disabled companions, veterans, holders of other products such as rail cards etc There may also be products for groups of users of a particular type. To define the eligible user profiles of a product, the following steps are required: • Determination of the criteria to define a user (e.g. age) and the required proof. • Determination of concessionary discounts to be given to these categories of users
NeTEx contribution	NeTEx provides a means to exchange the user profiles and related conditions, and the prices associated with them.
Main actors	Marketing planners, Fare planning systems
Main objects	USER PROFILE, GROUP TICKET, COMMERCIAL PROFILE, COMPANION PROFILE, TYPE OF CONCESSION

	USAGE PARAMETER PRICE, PRICING RULE, DISCOUNTING RULE, LIMITING RULE.

5.5.1.3 Provide data for determination of basic fare tariffs (part of fare structure)

5.5.1.3.1 Use Case: Define Flat (including zero) prices

Use Case: FARE-001-TARIFF-001 (#9)	
Name	Define Flat (including zero) fare prices
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Some systems, especially bus, use a simple system of flat fares. A flat fare is found in particular in systems for concessionary fares for elderly and disabled people (for example 50p for any journey in Greater Manchester in 2005/06). A special case is zero or "free" fares. In some areas such concessions extend to rail travel as well. Define the flat or zero fare and the conditions under which they are valid. Applicable conditions can be Network properties User profile categories Temporal categories Or any combination of the above. Note that concessionary travel is not "free" - typically it is priced and paid for by government. A related use case, outside of the scope of NeTEx, is to model the actual cost of this travel, requiring tin addition projected traffic volumes and operating costs and margins. However, the value to the user could be priced by using the normal prices for equivalent travel.
NeTEx contribution	Flat fares can be defined for different categories of user and other conditions.
Main actors	Fare planning systems.
Main objects	FARE STRUCTURE ELEMENTs, VALIDITY CONDITION, DAY TYPE. USER PROFILE, USAGE PARAMETER PRICE

5.5.1.3.2 Use Case: Determination of the basic fare structure tariff

Use Case: FARE-001-TARIFF-002 (#10)	
Name	Determination of the basic structure tariff
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI

Description	The base tariff is a set of Priceable elements for each of which a price is set without concessionary reductions. The tariffs are defined for categories or intervals of space (stops, zones or stages), travel time or combination. The basic tariff factors can be represented in several ways. • Define a **factor** unit price. Definition of the price per basic fare structure factor resulting in a linear price function. E.g. 1.05 EUR per Km, 0.79 GBP per minute, 1.50 EUR per zone. • Define a tier table that maps a consumed amount of factor units to the due tariff. E.g. e.g. three coupons for a two-zone journey. • Define an OD distance matrix Definition of a matrix that holds tariff distance and / or the actual tariff that is due while travelling from Origin to Destination. Origins and destinations can be defined as fare points, zones, stages or sections. The pricing may involve estimating traffic volumes along with running costs in order to arrive at a projected revenue and operating profit.
NeTEx contribution	NeTEx provides a means to exchange the actual basic prices and subsequent revisions.
Main actors	Fare planning systems.
Main objects	FARE STRUCTURE ELEMENT, GEOGRAPHICAL STRUCTURE FACTOR, GEOGRAPHICAL INTERVAL PRICE, TIME STRUCTURE FACTOR, TIME INTERVAL PRICE, etc DISTANCE MATRIX ELEMENT, DISTANCE MATRIX ELEMENT PRICE.

Examples of tariff tables:

Time factor units:

Hours travelled	tariff
0 to 0.5	2.0 EUR
0.5 to 1	3.0 EUR
1 to 1.5	3.75 EUR
Etc.	

Distance factor units:

Zones travelled	tariff
0 to 1	2.0 EUR
1 to 3	3.0 EUR
3 to 6	4.0 EUR
Etc.	

5.5.1.3.3 Use Case: Determination of tariff differentiation based on network properties

Use Case: FARE-001-TARIFF-003 (#11)	
Name	Determination of tariff differentiation based on network properties
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	Derivative tariffs can be based on network properties, e.g. single or return, mode of transport, express or stopping line, travel class. Derivative tariff prices are usually a simple multiple of the standard or base fare, but an arbitrary pricing rule may be indicated if necessary. • Define the network properties and accompanying multipliers that need to be used to differentiate tariffs. These properties will be represented by validity conditions. and • Define the representation of the derived tariffs. The following representation methods are possible: Multiplier(s) with accompanying conditions that restrict to the corresponding network property. Upon use, in case a condition applies, the selected multiplier is used to multiply the base tariff to obtain the derived tariff. For each determined base tariff, additional derivative tariffs can be added with accompanying conditions that restrict to the corresponding network property. Upon use, in case a condition applies, the corresponding tariff is selected.
NeTEx contribution	NeTEx provides a means to exchange the actual tariff and different versions of it.
Main actors	Fare planning systems.
Main objects	FARE PRICE (and specialisations), FARE STRUCTURE ELEMENTs, USAGE PARAMETERs (e.g. ROUND TRIP), CLASS OF USE PRICING RULE, DISCOUNTING RULE, LIMITING RULE.

5.5.1.3.4 Use Case: Addition of concessionary tariffs

Use Case: FARE-001-TARIFF-004 (#12)	
Name	Addition of concessionary tariffs
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	Concessionary tariffs can be derived from the base tariffs according to specific user profiles, e.g. reduction for the elderly or reduction for travellers with certain disabilities, or the armed forces. The derivative tariffs with the conditions that define their application can be contained by the tariff structure. Derivative tariffs usually are a simple multiple of the standard or base fare, but an arbitrary pricing rule may be indicated if necessary. • Define the user profile categories and accompanying multipliers that need to be used to differentiate tariffs. These properties will be represented by user profiles. • Define the representation of the derived tariff prices. The following representation methods are possible:

	Multiplier(s) with accompanying conditions that restrict to the corresponding profile characteristic. Upon use, in case a condition applies, the selected multiplier is used to multiply the base tariff to obtain the derived tariff. For each determined base tariff, additional derivative tariffs can be added with accompanying conditions that restrict to the corresponding profile characteristic. Upon use, in case a condition applies, the corresponding tariff is selected. Note: Derivative tariffs can be packaged either as separate profiles of a common product, or as separate products with separate sales offer packages. Which tariff derivative is actually applied during travelling is determined by the profile characteristic of the traveller that is present within the product.
NeTEx contribution	NeTEx provides a means to exchange the actual tariff.
Main actors	Fare planning systems
Main objects	FARE STRUCTURE ELEMENT, USER PROFILE, GROUP TICKET, COMMERCIAL PROFILE, COMPANION PROFILE, TYPE OF CONCESSION. USAGE PARAMETER PRICE, PRICING RULE, DISCOUNTING RULE, LIMITING RULE.

5.5.1.3.5 **Use Case: Addition of temporal derivative tariffs**

Use Case: FARE-001-TARIFF-005 (#13)	
Name	Addition of temporal derivative tariffs
Source	Transmodel, FareXChange, BISON, NeTEx TAP/TSI
Description	Derivative tariffs can be based on time of day and day type, e.g. Higher rates during rush hour on weekdays or a reduction during weekend or holidays. The derivative tariffs with the conditions that define their application can be contained by the tariff structure. The conditions are defined in the form of time band and day types. Derivative tariffs usually are a simple multiplier of the standard or base fare Define the temporal properties and accompanying multipliers that are used to differentiate tariffs. The temporal properties will be represented by validity conditions. and Define the representation of the derived fares. The following representation methods are possible: Multiplier(s) with accompanying validity conditions that restrict to the corresponding network property. This multiplier is used to multiply the base tariff to obtain the derived tariff. For each base tariff, additional derivative tariffs can be added with accompanying validity conditions that restrict to the corresponding network property.
NeTEx contribution	NeTEx provides a means to exchange the actual tariff and different versions of it.
Main actors	Fare planning systems.

Main objects	FARE INTERVAL PRICE, FARE STRUCTURE ELEMENT, FARE DEMAND FACTOR, FARE DEMAND FACTOR PRICE, DAY TYPE, TIMEBAND VALIDITY CONDITION

5.5.1.4 Provide data for determination of fare products

5.5.1.4.1 Use Case: Determination of the product access rights

Use Case: FARE-001-PRODUCT-001 (#14)	
Name	Determination of the product access rights
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	A traveller who wants to travel using PT transport is required to acquire a product that grants verifiable access to PT transport services. Access rights can be expressed as combinations of different types of rights: **Spatial access rights.** Access to stages, zones, stop areas, domains, routes (one-way, retour), etc. These access rights must be based on spatial or geographical entities as present in the PT network definition. May include restrictions on interchanging or the period of travel, etc. **Temporal access rights.** Access on certain operational days, day types, date and time periods. **Consumable access rights.** Access for a certain amount of services, e.g. 10 trips. To define the access rights of a product, the following steps are required: • Determination of the product access rights that are obtained on acquirement of the product as part of the definition of the product. • Determination of additional PT network properties that are required to express the access right with sufficient accuracy. Access rights including non-transport entities such as meals and add-ons, such as sleeping accommodation and parking
NeTEx contribution	NeTEx provides a means to exchange the actual product access rights and associated
Main actors	Fare planning systems, Passenger Information systems
Main objects	ACCESS RIGHT PARAMETER ASSIGNMENT, GENERIC PARAMETER ASSIGNMENT, ACCESS RIGHT IN PRODUCT. FARE PRODUCT, VALIDABLE ELEMENT, ACCESS RIGHT IN PRODUCT, USAGE Parameters. ROUTING, STEP LIMIT, ROUND TRIP, USAGE VALIDITY PERIOD, INTERCHANGING.

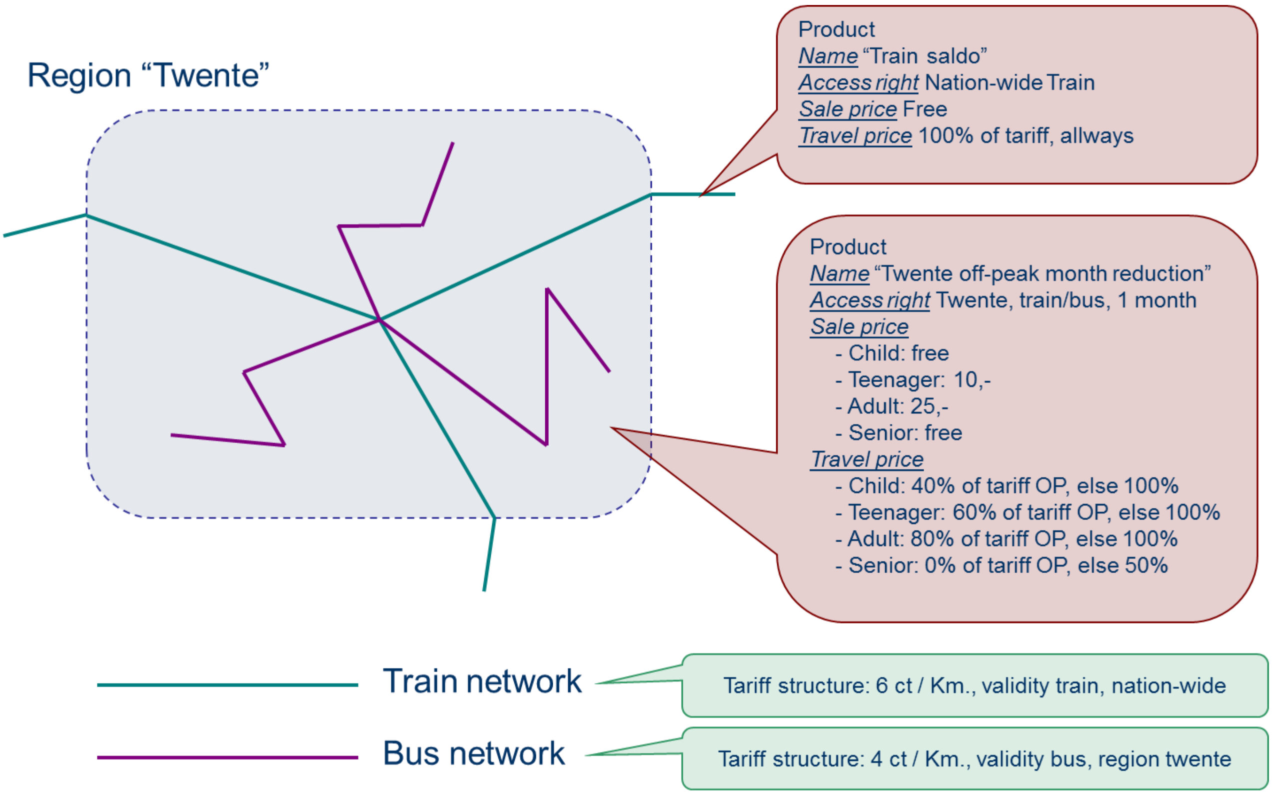


Figure 5 — Region “Twente” example (EXM)

5.5.1.4.2 **Use Case: Determination of product interoperability**

Use Case: FARE-001-PRODUCT-002 (#15)	
Name	Determination of product interoperability
Source	FareXChange, BISON, NeTEx, TAP/TSI
Description	Determination of product interoperability, e.g. validity under joint PT Operators or cross boundary validity and behaviour. This is a further refinement of the “Determination of the product access rights” use-case. Product interoperability is defined by the product access rights; the access rights may include other operators’ networks. Where pricing involves assembling journey legs from different operators, well defined boundary points are needed.
NeTEx contribution	NeTEx allows a global identifier scope to be enabled to allow the integration of elements from many different networks
Main actors	Fare Management Systems, Passenger Information Systems
Main objects	FAR PRODUCT, ENTITLEMENT, ENTITLEMENT PRODUCT, BORDER POINT, SERIES CONSTRAINT.

5.5.1.4.3 Use Case: Determination of product pre- and post- sale conditions

Use Case: FARE-001-PRODUCT-003 (#16)	
Name	Determination of product pre- and post- sale conditions
Source	FareXChange, BISON, TAP/TSI
Description	Conditions may attach to the retailing of product, for example: • When bookings must be made. • Conditions for exchanging and refund. • Minimum Stay. • Whether it may be exchanged for another product. • Whether it may or transferred to another user. • Whether there is a reservation fee, etc. • The amount of luggage allowed. • etc. etc These conditions need to be represented in a precise from that can be interpreted by a machine, over, as well as having text renderings suitable for passengers. Combination of PT Products
NeTEx contribution	Conditions may be specified and attached to products. Both machine readable representations and human readable text notices may be included.
Main actors	Fare Management Systems, Passenger Information Systems
Main objects	FARE PRODUCT, GENERIC PARAMETER ASSIGNMENT USAGE PARAMETER, e.g. REFUNDING, EXCHANGING, PURCHASE WINDOW, REPLACING, RESERVING, TRANSFERABILITY, LUGGAGE ALLOWANCE. USAGE PARAMETER PRICE, SUMMARY CONDITION, SALES NOTICE, NOTICE ASSIGNMENT

5.5.1.4.4 Use Case: Determination of distribution conditions

Use Case: FARE-001-PRODUCT-004 (#17)	
Name	Determination of distribution conditions
Source	FareXChange, BISON, NeTEx, TAP/TSI
Description	Particular products may only be available through specific retail channels or in specific countries. These conditions need to available in both machine and human readable from.
NeTEx contribution	NeTEx provides a means to exchange the actual product information and subsequent updates.
Main actors	Fare management systems, Passenger information systems, Retailing systems
Main objects	DISTRIBUTION CHANNEL, DISTRIBUTION ASSIGNMENT, SALES OFFER PACKAGE. FULFILMENT METHOD PRICE.

5.5.1.4.5 Use Case: Determination of materialized packages

Use Case: FARE-001-PRODUCT-005 (#18)	
Name	Determination of materialized packages
Source	FareXChange, BISON, NeTEx, TAP/TSI
Description	Products may be packaged on a number of different types of ticket, electronic and paper, and be sold through different channels with different methods of fulfilment. Packages may bundle together different products, for example a train and a metro ticket from different operators. Sometimes there will be an extra fee for a particular method of fulfilment.
NeTEx contribution	NeTEx provides a means to represent create sales offer packages comprising one or more product and with specific conditions as to distribution, fulfilment, etc.
Main actors	Fare management systems, Passenger information systems, Retailing systems
Main objects	SALES OFFER PACKAGE, SALES OFFER PACKAGE ELEMENT, DISTRIBUTION ASSIGNMENT, SALES OFFER PACKAGE FULFILMENT METHOD, FULFILMENT METHOD PRICE

5.5.1.4.6 Use Case: Determination of additional fees and charges for fare product

Use Case: FARE-001-PRODUCT-006 (#19)	
Name	Determination of additional fees and charges for fare product
Source	FareXChange, BISON, NeTEx, TAP/TSI
Description	Sometimes there will be an additional arbitrary fee e.g. for boarding, taxes, delivery or other reasons. Additional fees can be defined in the tariff structure. Additional fees can be defined within a travel product. The conditions that determine how and when the fees are charged are always part of the product definition.
NeTEx contribution	NeTEx provides a means to exchange the actual product information and subsequent updates.
Main actors	Fare planning systems, Passenger Information systems
Main objects	FARE STRUCTURE ELEMENT, USAGE PARAMETER, USAGE PARAMETER PRICE, USAGE PARAMETER, FULFILMENT METHOD PRICE

5.5.1.4.7 Use Case: Determination of required product deposits and credit limits

Use Case: FARE-001-PRODUCT-007 (#20)	
Name	Determination of product deposits and credit limits
Source	FareXChange, BISON, NeTEx
Description	When travelling with a pay-as-you-go or post specified product, sometimes a deposit is charged at the beginning of a trip that is returned when finishing the trip, in other cases a minimum balance is required. Define the deposit and or credit limit currency, unit and value.

NeTEx contribution	NeTEx provides a means to specify pay-as-you-go products the deposit and credit limits as part of the sales definition.
Main actors	Fare Management Systems, Passenger Information Systems
Main objects	CONDITION SUMMARY, CAPPED FARE PRODUCT, FARE PRODUCT PRICE, LIMITING RILE,

5.5.1.4.8 **Determination of the product price and concessionary parameters.**

Use Case: FARE-001-PRODUCT-008 (#21)	
Name	Determination of the product price and concessionary parameters.
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	Determination of the product price that is charged to the traveller. This charging takes place at specific moments according to specific methods. E.g. pay as you go, prepaid, post-paid etc. The price may be chosen from a precomputed table of all combinations, or be derived from a base price The product price is determined by the concessionary parameters that are acquired. Product prices depend on: User profile of the owner, Travel class, Time of travel (peak, of-peak) The nature of the product or sales offer package etc. A price can be defined as an amount (e.g. sale price of a month reduction pass is 30.00 euro) or as a calculation method (e.g. sale price for a one-way ticket is 60% of the base tariff as defined in the tariff structure, applicable to seniors). In case a product price (either for sale or usage) refers to the tariff structure, the calculation method can be defined several way's: A tariff structure may contain several tariffs with individual validity conditions for each factor. E.g. two tariffs between stop A and stop C, one for travel class 1 and one for travel class 2. The selection of one of the tariff options is determined by the product characteristics. The product definition may contain a reduction percentage that is applied to the tariff from the tariff structure. E.g. 60% of the base tariff for senior profile or 80% of the base tariff during off-peak hours for all profiles. For the latter example a calendar is also needed to define the period begin and end times. Any combination of the above two categories, e.g. a reduction that is applied to either the tariff for first or second class. To define the price of a product, the following steps are required: Determination of the concessionary parameter(s) that are relevant for the product. Determination of the product sale- and usage- prices or price calculation rules, depending on the concessionary parameters that are defined.

NeTEx contribution	NeTEx provides a means to exchange the actual product price information and different versions of it.
Main actors	Fare planning systems, Passenger Information systems
Main objects	TARIFF, FARE PRICEs, PRICING RULEs, PRICING PARAMETER SET USAGE PARAMETER, USAGE PARAMETER PRICE FARE PRODUCT, SALES OFFER PACKAGE, SALES OFFER PACKAGE ELEMENT

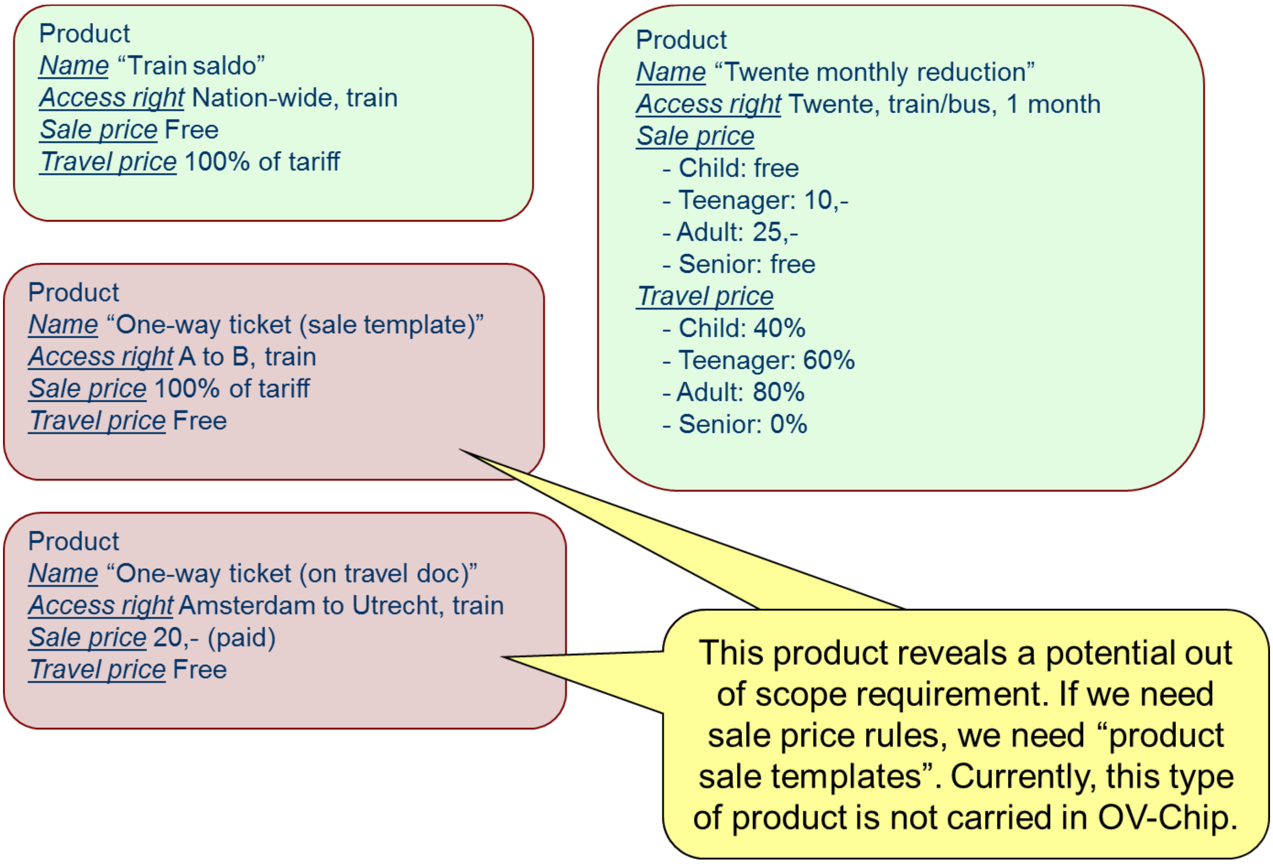


Figure 6 — Product pricing examples. (EXM)

Product
<u>Name</u> “Twente off-peak month reduction”
<u>Access right</u> Twente, train/bus, 1 month, first class
<u>Sale price</u>
- Child: free
- Teenager: 20,-
- Adult: 45,-
- Senior: free

<u>Travel price</u>
- Child: 40% of tariff OP, else 100%
- Teenager: 60% of tariff OP, else 100%
- Adult: 80% of tariff OP, else 100%
- Senior: 0% of tariff OP, else 50%

Calendar	
Daytype	Off-Peak
Sunday	00:00 – 23:59
Monday	00:00 – 07:00, 09:00 – 23:59
Tuesday	00:00 – 07:00, 09:00 – 23:59
Wednesday	00:00 – 07:00, 09:00 – 23:59
Thursday	00:00 – 07:00, 09:00 – 23:59
Friday	00:00 – 07:00, 09:00 – 23:59
Saturday	00:00 – 23:59

Tariff structure (partly)		
Distance tier	First class	Second class
0 -9 km	0,80	0,45
10 – 20 km	1,45	0,80
etc.		

Figure 7 — Product pricing example (EXM)

5.5.1.4.9 **Use Case: Periodic product re-price**

Use Case: FARE-001-PRODUCT-009 (#22)	
Name	Periodic product re-price
Source	FareXChange
Description	Existing fare structures are typically re-priced annually or at other intervals to account for inflation and cost changes. New prices need to be supplied for the same fare structure, typically applying a standard increase
NeTEx contribution	Prices can be exchanged separate from the underlying products.
Main actors	Fare planning systems, Passenger Information systems
Main objects	Part1: VALIDITY CONDITION Part3: FARE FRAME, PRICE GROUP, FARE TABLE, ELL ROUNDING, ROUNDING STEP

5.5.1.5 Provide data for distribution of information about fare products

5.5.1.5.1 **Use Case: Provide information about fare product sales channels**

Use Case: FARE-002-INFORMATION-001 (#23)	
Name	Provide information about fare product sales channels
Source	FareXChange, NeTEx, TAP/TSI

Description	To provide information about links (web sites) and places (retailers) where passengers may get fare information and where they may buy tickets / products
NeTEx contribution	NeTEx can describe (group of) distribution channels (including topographic places, i.e. country, distribution rights and payment methods) through which a product may or may not be distributed.
Main actors	Fare management systems, Journey planning systems
Main objects	DISTRIBUTION CHANNEL, FULFILMENT METHOD, DISTRIBUTION ASSIGNMENT, SALES OFFER PACKAGE, RETAIL CONSORTIUM

5.5.1.5.2 Use Case: Provide information on fare products

Use Case: FARE-002-INFORMATION-002 (#24)	
Name	Provide information on fare products
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	To provide information on available fare products and their rules and restrictions to passengers. To distribute general planned information on fares dedicated to be published (several types of displays including printed leaflets).
NeTEx contribution	For a sales offer package the assignment of a NOTICE to a SALES OFFER PACKAGE or a GROUP OF SALES OFFER PACKAGEs can be described.
Main actors	Passenger Information Systems
Main objects	FARE PRODUCT, SALES OFFER PACKAGE, SALES OFFER PACKAGE ELEMENT, GENERIC PARAMETER ASSIGNMENT, USAGE PARAMETER, DISTRIBUTION CHANNEL, NOTICE ASSIGNMENT, NOTICE, DISTRIBUTION ASSIGNMENT

5.5.1.5.3 Use Case: Provide fare information to on-line systems

Use Case: FARE-002-INFORMATION-003 (#25)	
Name	Provide fare information to on-line systems
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	To provide general planned information on fares to online passenger information services that provide fare information or enhance existing services with fare information. To provide base tariffs for a region, line or other applicable selection per profile Typically such systems will 1. Calculate trip 2. Identify available fare products on this trip 3. Get price parameter : provide price if it is scheduled/fixed, if not call a "remote" price calculator if available 4. Provide links/information on sales channels
NeTEx contribution	A request for a specific information on public transport, expressed during a PI TRANSACTION.
Main actors	Passenger Information Systems
Main objects	FARE FRAME with all OBJECTs, objects from PIQUERY package

5.5.1.5.4 Use Case: Optimize journey plan for price

Use Case: FARE-002-INFORMATION-004 (#26)	
Name	Optimize journey plan for price
Source	FareXChange, NeTEx, BISON
Description	To provide up to date fare parameters to identify conditions for lowest fare while planning a journey. Taking into account the fare products that are owned by the traveller and that that passenger information is conform to the actual charged prices during sale and travel
NeTEx contribution	NeTEx allows the detailed fares and travel conditions to be specified in machine readable form. These may be used by a journey planning engine to provide a price optimised query.
Main actors	Journey planning systems
Main objects	FARE QUERY, SALES OFFER PACKAGE, TRAVEL SPECIFCATION.

5.5.1.5.5 Use Case: Provide Information about loyalty programs

Use Case: FARE-002-INFORMATION-005 (#27)	
Name	Provide information about loyalty programs
Source	FareXChange, NeTEx
Description	To provide information about loyalty programs
NeTEx contribution	NeTEx can describe TRAVEL DOCUMENTs (ticket, card, etc.) allowing the right to travel or to consume joint-services. It is also possible to describe ENTITLEMENTs to access a service or to purchase a FARE PRODUCT issued by an organisation that may not be a PT operator (e.g. military card).
Main actors	3rd party systems
Main objects	SALE DISCOUNT RIGHT, ENTITLEMENT PRODUCT, TRAVEL DOCUMENT

5.5.1.5.6 Use Case: Provide zones for maps

Use Case: FARE-002-INFORMATION-006 (#28)	
Name	Provide zones for maps
Source	FareXChange, NeTEx,
Description	To show fare zones on topographical and schematic maps
NeTEx contribution	NeTEx allows relevant coordinate and presentation data to be associated with the Fare structure elements so that maps may be created to visualise fare zones and routings.
Main actors	Passenger Information systems, Publishing systems.
Main objects	Part1: TARIFF ZONE, SHCEDULED STOP POINT, LINE, LINE SECTION, NETWORK., SCHEMATIC MAP Part3: FARE ZONE

5.5.1.5.7 Use Case: Provide an overview of fare products / tariffs

Use Case: FARE-002-INFORMATION-007 (#29)	
Name	Provide an overview of fare products / tariffs
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	To provide an overview of available products and their prices and tariffs for a region, line or other applicable selection per profile, e.g. for comparison purposes
NeTEx contribution	NeTEx can provide a grouping of SALES OFFER PACKAGEs, their FARE PRODUCTs with ACCESS RIGHT PARAMETER ASSIGNMENT.
Main actors	Passenger Information systems, Publishing systems.
Main objects	SALES OFFER PACKAGE, FARE PRODUCT, CONDITION SUMMARY, NOTICE.

5.5.1.6 Provide data for organisation of fare products sales

5.5.1.6.1 Use Case: Provide up to date fare parameters for price calculation

Use Case: FARE-002-SALES-001 (#30)	
Name	Provide up to date fare parameters for price calculation
Source	
Description	To provide up to date fare parameters for price calculation during sale and usage for fare products To provide sale and usage prices, rules and restrictions on fare products for a specific passenger trip (taking into account a specific trip context, i.e. line, date, time, interchanges, trip duration, distance, etc.) and that passenger information is conform to the actual charged prices during sale and travel (exclusive yield managed fares).
NeTEx contribution	NeTEx provides structures for grouping of prices (specialization of PRICE GROUP) that may be associated with all or any of DISTANCE MATRIX ELEMENT, FARE STRUCTURE ELEMENT GEOGRAPHICAL INTERVAL, GROUP OF ACCESS RIGHT PARAMETERs, CLASS OF USE, OPERATOR, VEHICLE MODE, FARE PRODUCT.
Main actors	Fare management systems, Journey planning systems
Main objects	Part2 : VEHICLE JOURNEY Part3 FARE TABLE, PRICE GROUP, FARE PRICE, TRAVEL SPECIFICATION

5.5.1.6.2 Use Case: Product selection and purchase

Use Case: FARE-002-SALES-002 (#31)	
Name	Product selection and purchase
Source	Transmodel, FareXChange, NeTEx
Description	To describe a specific sales offer package and selected parameters as chosen by a user and as potentially sold to them as a transaction.

	The choice may be used to handoff between a passenger information system used to find a product and a reservation engine used to buy it. For electronic purchase without any visible materialization of a ticket the record may be used to provide an audit log that an app may use to give the user a record of the purchases they have made. There may be large numbers of such transactions exchanged independently of the fare information itself.
NeTEx contribution	A sales transaction with a set of parameters representing a specific product purchase can be described and exchanged.
Main actors	Retail systems, Passenger information system.
Main objects	SALES TRANSACTION FRAME with SALES TRANSACTION, TRAVEL SPEIFCATION, SPECIFIIC PARAMETER ASSIGNMENT, USAGE PARAMETER, SALES OFFER PACKAGE

5.5.1.6.3 Use Case: Provide Product specification for dynamic pricing

Use Case: FARE-002-SALES-003 (#32)	
Name	Provide Product specification for dynamic pricing
Source	Transmodel, FareXChange, NeTEx
Description	In a yield managed system, prices are determine dynamically at time of purchase by a pricing engine. Such systems typically require the user first to choose an product that meets their conditions as to type of user, time of travel etc, and then to request a price.
NeTEx contribution	NeTEx allows products and conditions to be specified precisely so that the correct inputs to a pricing engine can be provided and a suitable description of the conditions can be made.The rustling pried package may be described by a sales transaction.
Main actors	Retail systems, Passenger information system.
Main objects	FARE PRODUCT, FARE QUERY

5.5.1.6.4 Use Case: Provide information for recurring trips / travel

Use Case: FARE-002-SALES-004 (#33)	
Name	Provide information for recurring trips / travel
Source	Transmodel, FareXChange, NeTEx
Description	To provide the sale and usage price information for a recurring trip / travel and pricing alternatives using applicable travel products, taking to account that passenger information is conform to the actual charged prices during sale and travel.
NeTEx contribution	NeTEx can describe a REPEATED TRIP FARE QUERY which is a PASSENGER QUERY about the best fare products to use for repeated similar trips.
Main actors	Fare management systems
Main objects	REPEATED TRIP FARE QUERY, PASSENGER QUERY, FARE QUERY, SALES OFFER PACKAGE, TRAVEL SPECIFICATION.

5.5.1.7 Provide fare data for support of PT operations

5.5.1.7.1 Use Case: Information exchange between rail and local transport

Use Case: FARE-002-OPERATIONS-001 (#34)	
Name	Exchange information between rail and local transport
Source	Transmodel, FareXChange, NeTEx, TAP/TSI
Description	To exchange fare information between long distance (i.e. heavy rail) and local public transport
NeTEx contribution	NeTEx can describe a FARE FRAME which is a set of all fare data defined.
Main actors	Passenger information system
Main objects	FARE FRAME with FARE PRODUCTs, USAGE PARAMETERs, DISTANCE MATRIX ELEMENTs, FARE TABLEs

5.5.1.7.2 Plan fare information for service providers

Use Case: FARE-002-OPERATIONS-002 (#35)	
Name	Plan fare information for service providers
Source	Transmodel, FareXChange, BISON, NeTEx
Description	To plan general planned information on fares dedicated to be used by service providers, such as PT operators, as a basis for derivation of information flows that facilitate operational processes.
NeTEx contribution	FARE FRAME enables exchange of a set of pricing parameters and values that apply to an individual element in the frame.
Main actors	Fare management systems
Main objects	FARE FRAME with FARE PRODUCTs, FARE PRICEs, FARE TABLEs, ACCESS RIGHT in PRODUCTSs

5.5.1.7.3 Use Case: Provide fare information that applies to a PT Contract

Use Case: FARE-002-OPERATIONS-003 (#36)	
Name	Provide fare information that applies to a PT Contract
Source	Transmodel, FareXChange, BISON, NeTEx, TAP/TSI
Description	To provide base tariffs that apply a PT contract by a PTO To provide an overview of tariff units (distance, geo or time as applicable) and their tariffs for a PT contract To provide an overview of available and required products and their prices and tariffs for a PT contract
NeTEx contribution	FARE FRAME enables exchange of a set of pricing parameters (including TARIFFs) including set of parameters controlling pricing calculations.
Main actors	Fare management systems

Main objects	FARE FRAME with FARE PRODUCTs, FARE PRICEs, FARE TABLEs, FARE INTERVALS

5.5.1.7.4 **Use Case: Provide information about interoperable products**

Use Case: FARE-002-OPERATIONS-004 (#37)	
Name	Provide information about interoperable products
Source	NeTEx
Description	To provide information about inter-operable products properties and parameters. where the same product can be placed in multiple PT modes (i.e. bus, train) and multiple PTOs.
NeTEx contribution	NeTEx can provide that data thru FARE FRAME, which can group many PRE-ASSIGNED FARE PRODUCTs for different PT modes.
Main actors	Fare information systems of multiple operators
Main objects	FARE FRAMEs, FARE PRODUCTs, OPERATORS, FARE PRICE, DISTRIBUTION CHANNEL, TRAVE DOCUMENT, ACCESS RIGHT ASSIGNMENT, MODE, NETWORK USAGE PARAMETERS

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 Part1 for a detailed description of the Physical Model and XSD mapping rules.

7 Public Transport Fares – Conceptual and physical data model

7.1 Introduction

7.2 Conceptual Model overview

7.2.1 Functional Domains

Transmodel breaks down “fare collection”: into the following areas:

- **Fare policy** specification:
  - characterisation of different fare structures through spatial and/or temporal parameters (e.g. sections, zones, time periods, etc.),
  - specification of the access rights allowed on a network, i.e. access to services provided on a transport network (e.g. trip on the metro network, trip on the bus network, access to the 1st class waiting area,, etc.) within a fare structure and the ways of using them (e.g. trip on the metro network during a time period of 2 hours, without successive on-board validations, trip on an open bus network during 1h30 with mandatory on-board validations and with the obligation to show an entitlement to use this right),
  - specification of marketable access rights, called fare products (e.g. service consumption rights granted through a simple ticket), being possibly combinations of access rights determined by different fare structures (called also “chained fare products”),
  - description of sale principles applied to the fare products (e.g. specification of fare products sold as a package, and parameters describing the purchase rights, as for instance the

obligation to show an entitlement to purchase a certain fare product),

- **Sales management**:
  - management of the sales network (not covered by Transmodel V5.1),
  - sales operations (including fulfilment) (partly covered by Transmodel V5.1),
  - management of customers (partly covered by Transmodel V5.1),
  - collecting funds or accounting (not covered by Transmodel V5.1);

- **Pricing**:
  - pricing parameters specification (partly covered by Transmodel V5.1),
  - exact price calculation (not covered by Transmodel),

- **Consumption control**:
  - access right validation & control (covered by Transmodel V5.1),
  - fraud management (partly covered by Transmodel),
  - collection and aggregation of consumption data (not covered by Transmodel V5.1);
  - management of revenue sharing and clearing house activities (not covered in Transmodel V5.1);

- **Provision of information on fares**.

NOTE All the comments indicating " not covered by Transmodel V5.1" will lead to complements to Transmodel that will be integrated in Transmodel V6: NeTEx (including all its parts) can be said to be compliant with Transmodel V6 first part will be released at about the same time as NeTEx Part 3.

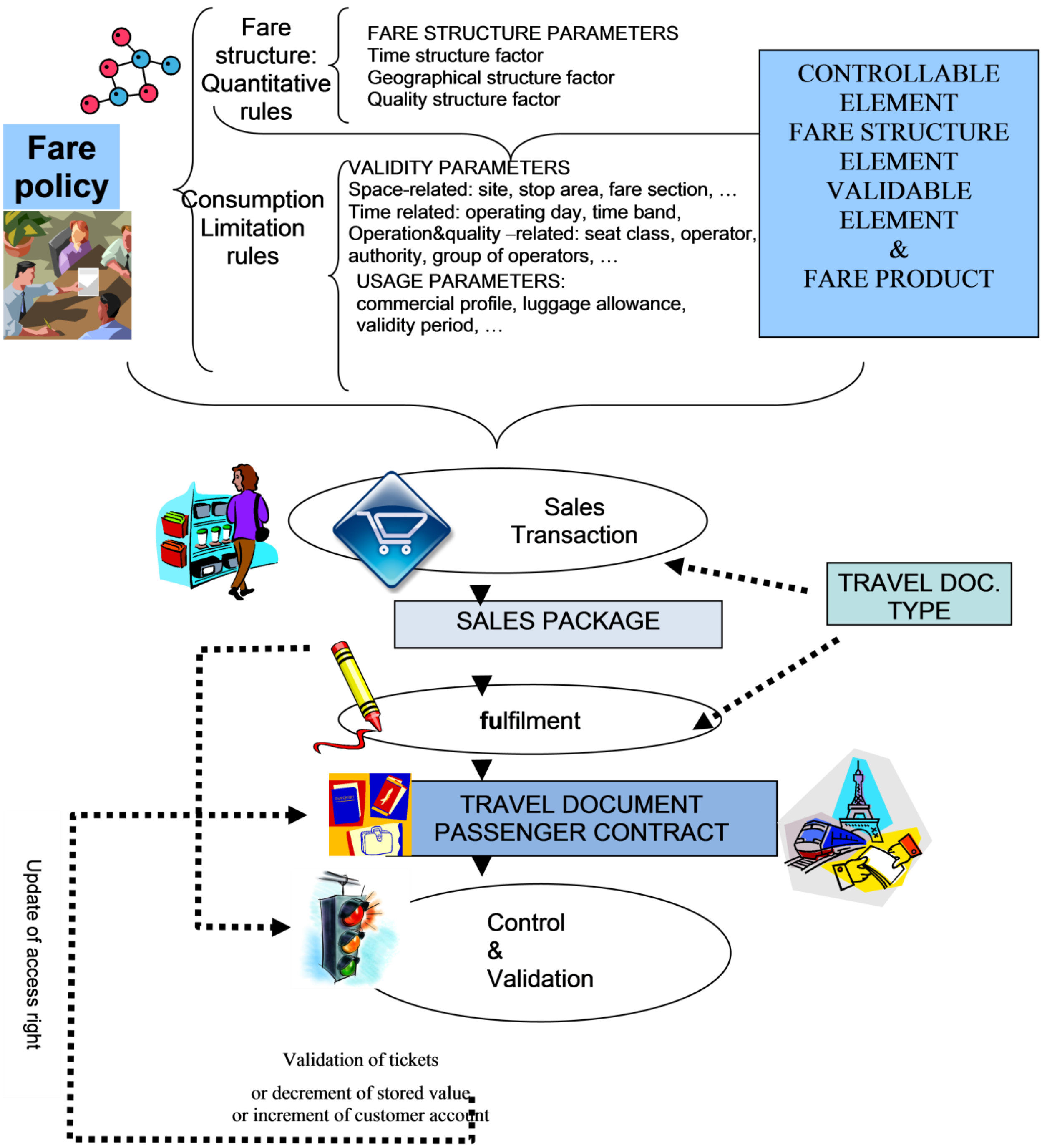


Figure 8 — Fare collection activities overview (SCHEMATIC)

7.2.2 Data Model Overview

The essential characteristic of the reference “fare collection” data model is that it is based on *access rights* (i.e. service consumption rights), rather than on prices.

An access right granted to a customer is a part of a service that a user is entitled to consume, and of which the service provider (or another organisation) is able to control the consumption. A large variety of prices may be attached to a particular access right, and it may be sold in a wide range of marketable combinations. It is possible therefore to describe a price or a marketable package by starting from the access right description, but the opposite is not true, or at least would result in a great complexity.

— Various access rights may be combined in order to form immaterial “fare products” (e.g. a “single ride” granted by a fare product called “simple ticket” or multiple trips during one month” granted by a fare product called “monthly pass”), which are marketable sets of access rights. One or several fare products may be associated to a “travel document” and materialised (e.g. a paper single ticket allowing only a “single ride” or an electronic card containing various fare products). Combinations of fare products and travel documents are sold to customers as “sales offer packages”. Each sold package is part of an individual “contract” with a particular customer.

— Controls are applied to access rights present on travel documents or in contracts, aimed at validating the consumption. The modelling of data related to control and validation of fare use is out of scope of NeTEx; however, the fact that fare products may represent “compound access rights” and that their elementary “components” have to be validated at a certain point or under some conditions, makes it necessary to be able to determine the elementary components of access rights (called in Transmodel CONTROLLABLE ELEMENTs, FARE STRUCTURE ELEMENTs, VALIDABLE ELEMENTs).

Pricing parameters are applied to access rights (either in a planning stage or, in the case of yield managed fares, in real time, according to specific principles), fare products and sales offer packages, in order to calculate the end price to be paid by the user.

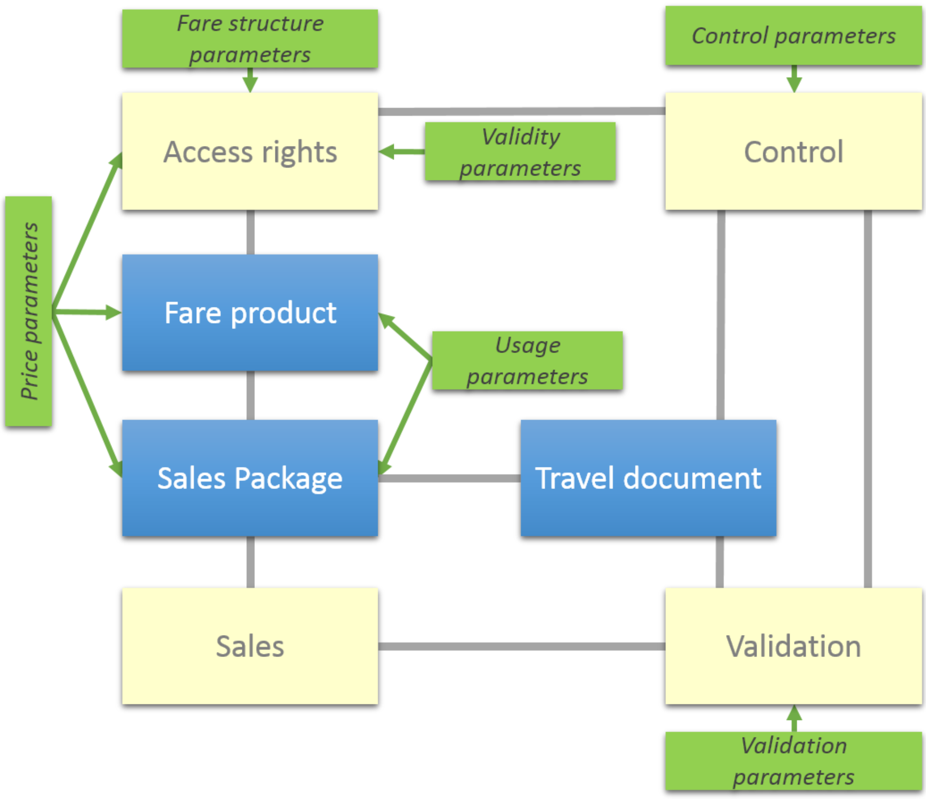


Figure 9 — Fare collection main processes and concepts (SCHEMATIC)

7.2.3 Main Concepts

7.2.3.1 Access Rights

The definition of the access rights to use a public transport service is linked to:

- a set of rules, determined mainly by space-, time or quality -related parameters - called in this document *fare structure parameters* - determining generic elements of the service offered for consumption, called also theoretical access rights (e.g. the possibility to carry out a ride on a service journey from one stop point to another on one single vehicle, the possibility to access a closed system such as a metro network, i.e. to access the metro system and carry out a trip on several vehicles etc.). Such elementary access rights may be combined in various marketable combinations;

- the definition of validity (limitation) rules, specifying the specific conditions to consume the generic or combined elements with use of *validity parameters* (e.g. particular tariff zone, particular line, specific operator, etc) or *usage parameters*, mainly linked to the characteristics of the practical use of the right, e.g. linked to the type of user (user profile, commercial profile, the maximum duration of a particular trip or set of trips, to the accompanying objects, such as luggage allowance, specific permissions, such as interruption of the use of the right, use of specific route, etc) ;

- the definition of means to materialise the access rights on travel documents (e.g. a throw-away ticket, an electronic travel card, etc.) and to control their consumption (e.g. using validators, turnstiles, manual controls, etc.).

7.2.3.2 Fare Structure and Tariff

The definition of fare structure elements is based on generic, mostly quantitative, rules that influence the access rights regulating the consumption of transport services, and thus the price a passenger has to pay for a specific consumption: limitation of the duration or the length of trips, the number of zones crossed, etc.

These rules describe the use of the transport system in terms of *space, time and service quality* (see section 7.5-Fare Structure). Therefore, space-based, temporal and quality parameters will be specified and attached to specific FARE STRUCTURE ELEMENTs by the fare structure.

The rules determining the access rights can be classified under two main categories:

- Global rules, determined by "fare structure parameters", used to determine the validity *of a range of generic access rights* providing the basis for calculating the price of their consumption. Such a set of rules is classically called "fare structure". A fare structure is mainly used to define generic pricing rules (for instance, the main types of public transport fare structure are known as "graduated fares" and "flat fares"). In a wider context, fare structure rules are also used to limit validity, even when the consumption is free of charge (e.g. limitation of the duration or the length of a trip, limitation of the number of zones crossed, etc.). Examples of Fare structure parameters are: GEOGRAPHICAL INTERVAL, GEOGRAPHICAL UNIT, TIME INTERVAL, etc (see section 7.5);

- Validity limitation rules which consist in assigning certain "limiting parameters" to specific access rights (see section 7.6-Access Rights Description-) For instance, a ride may be limited to a specific area, a trip limited by the latest possible start time, a pass valid only for students,, etc. Such limitations are expressed by two categories of parameters:

    - "validity parameters", which affect the physical characteristics of access rights (mainly in space or time); examples of validity limiting parameters are GROUP OF LINEs, DAY TYPE, etc.
    - "usage parameters", which affect the actual use of access rights, such as USER PROFILE, FREQUENCY OF USE, TRANSFERABILITY, etc.

A particular version of the fare structure fixes the values of the different parameters: this set of rules together with well-defined parameter values builds a TARIFF.

7.2.3.3 Fare Products

Finally, the access rights are advertised to the public.

A FARE PRODUCT is an immaterial marketable element made available to the public. It can be purchased and enables the owner to consume public transport or other services at specific conditions. It may consist of specified access rights (PRE-ASSIGNED FARE PRODUCT) or other products (discounts, amount of price unit, etc.).

In other words: *the set of access rights granted by a travel document represent a FARE PRODUCT*. It is a set of VALIDABLE ELEMENTs that are determined by fare structure parameters, validity parameters and usage parameters and may be considered as being combinations of FARE STRUCTURE ELEMENTs (and/or CONTROLLABLE ELEMENTs).

Each access right component level (CONTROLLABLE ELEMENTs, FARE STRUCTURE ELEMENTs VALIDABLE ELEMENTs) is characterised by the type of limitation rule that can be applied to access right components belonging to this level. The distinction between all three levels is only required for the most complex fare systems.

For NeTEx, the most relevant access right components are FARE STRUCTURE ELEMENTs and VALIDABLE ELEMENTs as they are the most relevant for the provision of information on fares. However, in some cases CONTROLLABLE ELEMENTs are also relevant, when the information on the controls is delivered.

To be comprehensive, the concept of a CONTROLLABLE ELEMENT is introduced in the reference data model. CONTROLLABLE ELEMENTs represent the most elementary components that are determined by the fare policy and are mainly dedicated to the “control process”.

7.2.3.4 Overview on Access Rights Levels, Parameter Types and Associated Concepts

7.2.3.4.1 Access Rights and Parameters

A fare structure is defined through a set of elements that represent access rights to the transport system. The access rights are determined through a range of rules that enable price calculation. The rules are basically of two types:

- global (mostly quantitative) rules, determined through fare structure parameters and
- particular (consumption) limitation rules, determined through validity and usage parameters.

Global, quantitative rules are, sometimes combined with validity limitation rules.

7.2.3.4.2 Basic Data Model Concepts

In terms of data model concepts, the description of access rights is organised in a hierarchy of three levels:

— CONTROLLABLE ELEMENTs,
— FARE STRUCTURE ELEMENTs
— VALIDABLE ELEMENTs.

A fare structure is defined through FARE STRUCTURE ELEMENTs that may be represented, for some needs and in some cases, as sequences of CONTROLLABLE ELEMENTs.

FARE STRUCTURE ELEMENTs may build sequences to be validated “in one go” (VALIDABLE ELEMENTs) i.e. as a set, without the possibility to use a single component as an isolated access right. In many cases a FARE STRUCTURE ELEMENT is already a VALIDABLE ELEMENT.

A FARE PRODUCT: is an immaterial marketable element (access right, discount right, etc), made available to the public.PREASSIGNED FARE PRODUCTs represent the access rights determined by a (or a sequence of) VALIDABLE ELEMENT(s).

FARE STRUCTURE ELEMENTs are typically determined through global rules, that may be combined with validity limitation rules.

To VALIDABLE ELEMENTs (in some cases to FARE STRUCTURE ELEMENTs as mentioned above) and to FARE PRODUCTs limitation rules apply, i.e. rules that either limit the validity by specific parameters: on one hand by

- validity parameters (e.g. particular DAY TYPEs, a particular LINE, a specific OPERATOR, etc) on the other hand
- by usage parameters, linked to the actual usage of the access rights (e.g. particular user profiles, luggage allowance, booking possibility, necessity to provide a particular entitlement, etc).

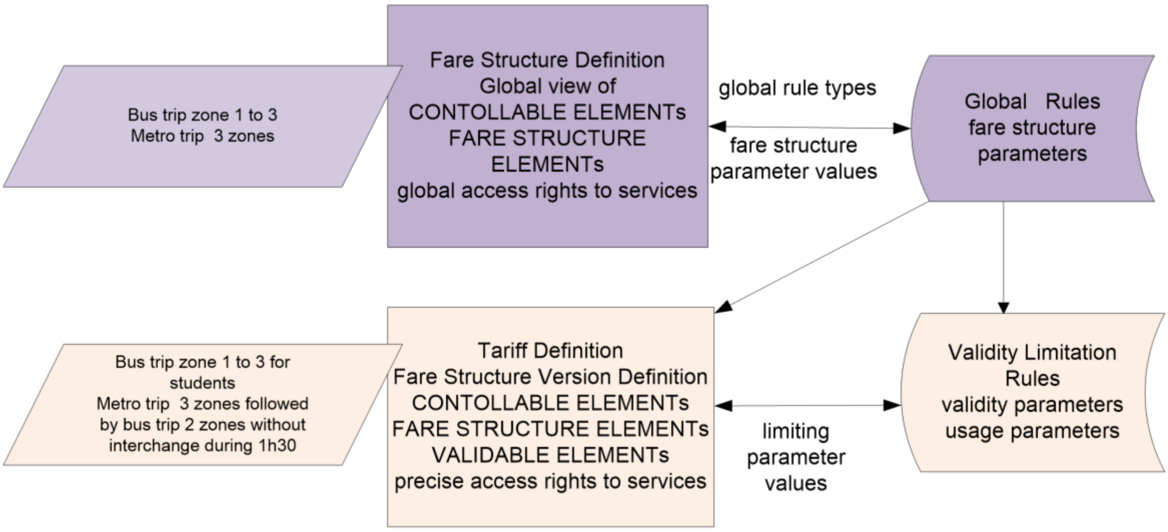


Figure 10 — Overview of the different parameters determining the components of access rights and examples (SCHEMATIC)

7.2.3.5 Travel Documents, Contracts and Sales Offer Packages

A FARE PRODUCT is immaterial, which means that the same FARE PRODUCT can be materialised on various TRAVEL DOCUMENTs according to a particular type (type of medium). For instance, a monthly pass may be incorporated on a specific ticket or stored on an electronic card.

Classical TRAVEL DOCUMENTs are anonymous, i.e. without any registration of the user.

If a registration takes place a CONTRACT is agreed between a customer and an organisation in charge of collecting fares for using services (authority, operator or another service provider), this consumption being ruled by the contract liabilities. Such an agreement is described by the CONTRACT entity.

A CONTRACT is immaterial, which means that is independent of the TRAVEL DOCUMENTs on which the whole CONTRACT or a subset of it may be loaded.

The FARE PRODUCTs are associated with TRAVEL DOCUMENTs in order to form packages suitable for selling. A SALES OFFER PACKAGE is defined as a package to be sold as a whole, consisting of one or several FARE PRODUCTs materialised thanks to one or several TRAVEL DOCUMENTs.

7.2.3.6 Sales Transactions

TRAVEL DOCUMENTs are usually allocated to customers on the occasion of a SALE TRANSACTION.

A SALE TRANSACTION is a log entry recording an elementary sale event. The following events shall be recorded as a SALE TRANSACTION:

- sale of one fixed SALES OFFER PACKAGE, which means the sale of one or several FARE PRODUCTs that are attached in a fixed way on one or several TRAVEL DOCUMENTs (e.g. classical tickets);
- sale of one reloadable SALES OFFER PACKAGE, which means that the sold FARE PRODUCTs are loaded on a TRAVEL DOCUMENT allowing such a loading (e.g. magnetic or electronic card);
- cancellation of loaded FARE PRODUCTs; this is meaningful in particular when a commercial offer allows to replace an already purchased SALES OFFER PACKAGE by a more attractive one.
- initial contracting procedure with a customer; this is meaningful in particular when such a procedure contains the CUSTOMER registration, before any product sale.

7.2.3.7 Prices

There is a large variety of methods to calculate the price to be paid. There is probably no generic solution to model all possible price generation algorithms. Therefore, the data model includes a set of price entities, which provide the data necessary to calculate the price in each of the cases. Specific algorithms are responsible for applying the local price calculation rules to this basic data.

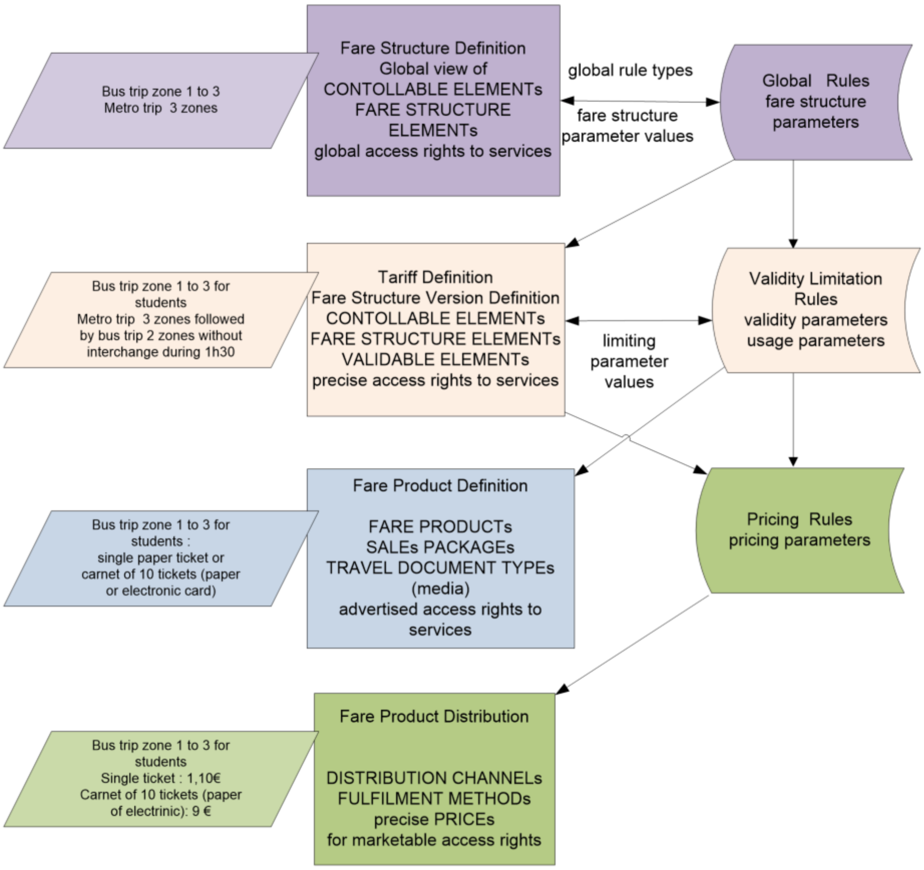


Figure 11 — Overview on Rules Types, Parameters and Concepts (SCHEMATIC)

7.3 Fare Model dependencies

NeTEx Part3 Fare model is modularised into a number of submodels defined as UML packages, these in turn depend on Part2 and Part1 packages.

- The FARE ZONE Models describe the network related fare constructs.
- The FARE STRUCTURE Models provide the various types of element used to represent fare structures
- The FARE PRODUCT Models describes the available FARE PRODUCTs.
- The USAGE PARAMETER Models describe the limiting conditions for the fare products.
- The FARE ACCESS RIGHT PARAMETER Models assign the access rights to specific products and limiting parameters.
- The SALE PACKAGE Model describes how the fare elements are combined as marketable components.
- The FARE PRICE Model and FARE TABLE Model present FARE PRICEs and PRICE GROUPs, and the FARE CALCULATION Model holds common pricing parameters.
- The TRAVEL DOCUMENT Model indicates the types of available travel document.
- The PARKING TARIFF Model records prices for PARKING.

- The FARE CONTRACT model describes identified CUSTOMERs and their contracts.
- The SALES TRANSACTION model records sales of SALES OFFER PACKAGEs as specific selections of fare elements.
- The FARE VALIDATION AND CONTROL Model describes basic validation and control Elements useful for grouping access rights and for interacting with downstream validation and control systems.
- The FARE FRAME Model describes the elements used to group fare data for exchange.

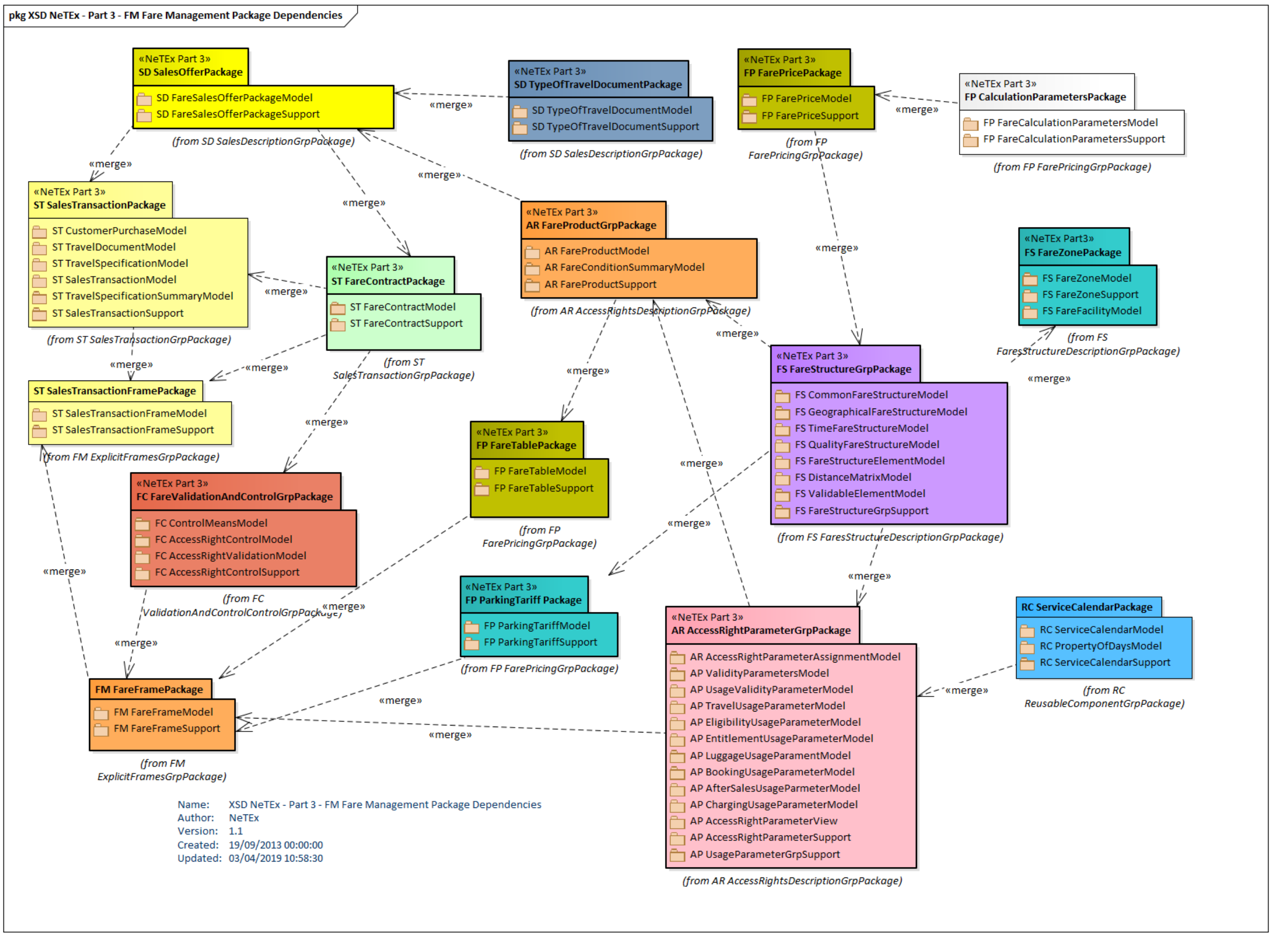


Figure 12 — Part 3 Fare Management Package Dependencies (UML)

The following diagram gives an overview of the dependencies between the models of NeTEx Part3.

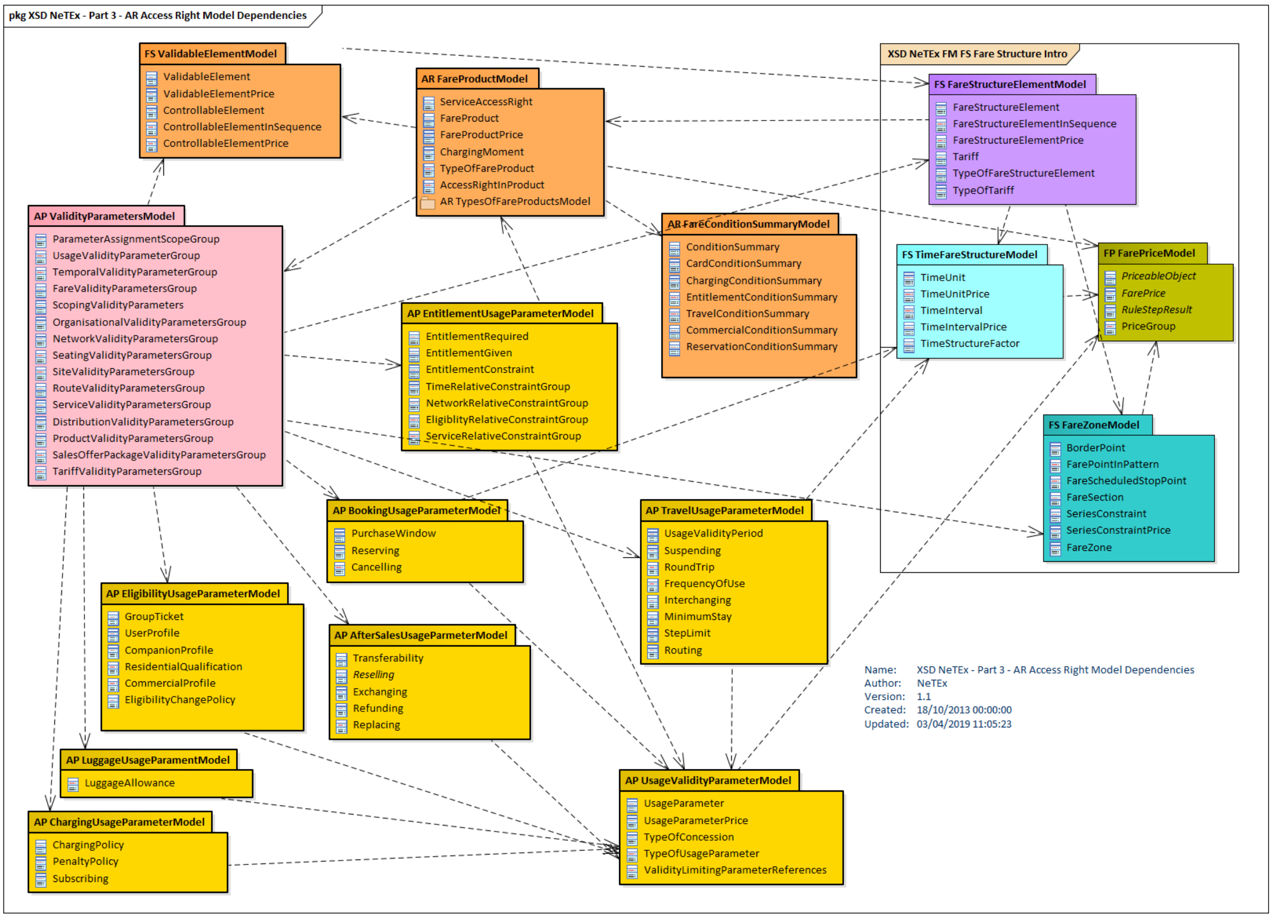


Figure 13 — Fare Model Dependencies (UML)

7.3.1 NeTEx Part3 Use of Version Frames

NeTEx uses VERSION FRAMEs (see Part1) as a mechanism for grouping related instances of data into a single package for exchange.Part1 and Part2 provide different types of concrete frames for different types of data. For example, stop related data may be exchanged with a SITE FRAME; timetable related data with a TIMETABLE FRAME. A COMPOSITE FRAME may be used to group different concrete frames with related data, for example a timetable and its stops (since elements on one type frame may reference elements in another).NeTEx Part3 provides two types of concrete VERSION FRAME to organise fare related data.

- A FARE FRAME is used to exchange fare definitions and prices (these may be exchanged together or separately). In effect it provides the reference data for sales transaction systems. Part1 & Part2 elements referenced by the fare data may be included using a COMPOSITE FRAME if required.

- A SALES TRANSACTION FRAME is used to exchange customer and sales data. Typically, there will be large volumes of such data generated by day to day use of the system that are aggregated and exchange frequently and independently of the FARE FRAME data elements.

7.3.2 Fare Frame

7.3.2.1 FARE FRAME – Conceptual MODEL

The elements of the FARE MODEL can be grouped with a FARE FRAME, which holds a coherent set of Fare related elements for data exchange. See VERSION FRAME in the NeTEx Framework section for general concepts relating to version frames.

FARE FRAMEs can be used to exchange all the various fare elements such as FARE STRUCTURE ELEMENTs, FARE PRODUCTs, SALES OFFER PACKAGEs, including prices. Using a COMPOSITE

FRAME, FARE FRAMEs can be assembled as a coherent, versioned set along with other types of NeTEx Data in other frames, for example SITE FRAMEs defining stops and TARIFF ZONEs, or SERVICE FRAMEs with VEHICLE JOURNEYs. The components of a FARE FRAME are described in detail in the following sections.

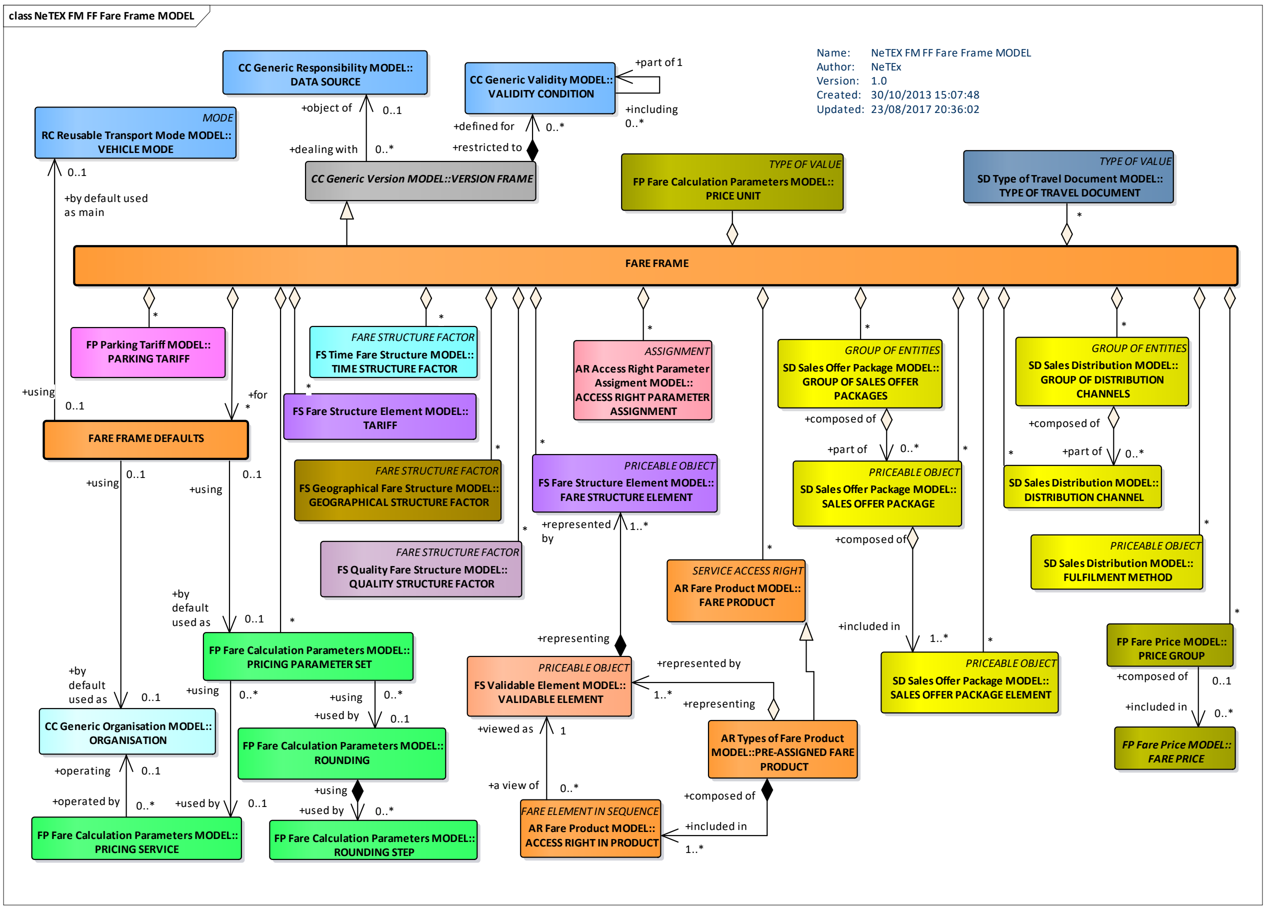


Figure 14 — Fare Frame Contents – Conceptual MODEL (UML)

7.3.2.2 Fare Frame – Physical Model

The following diagram shows an overview of the Physical model for a FARE FRAME.

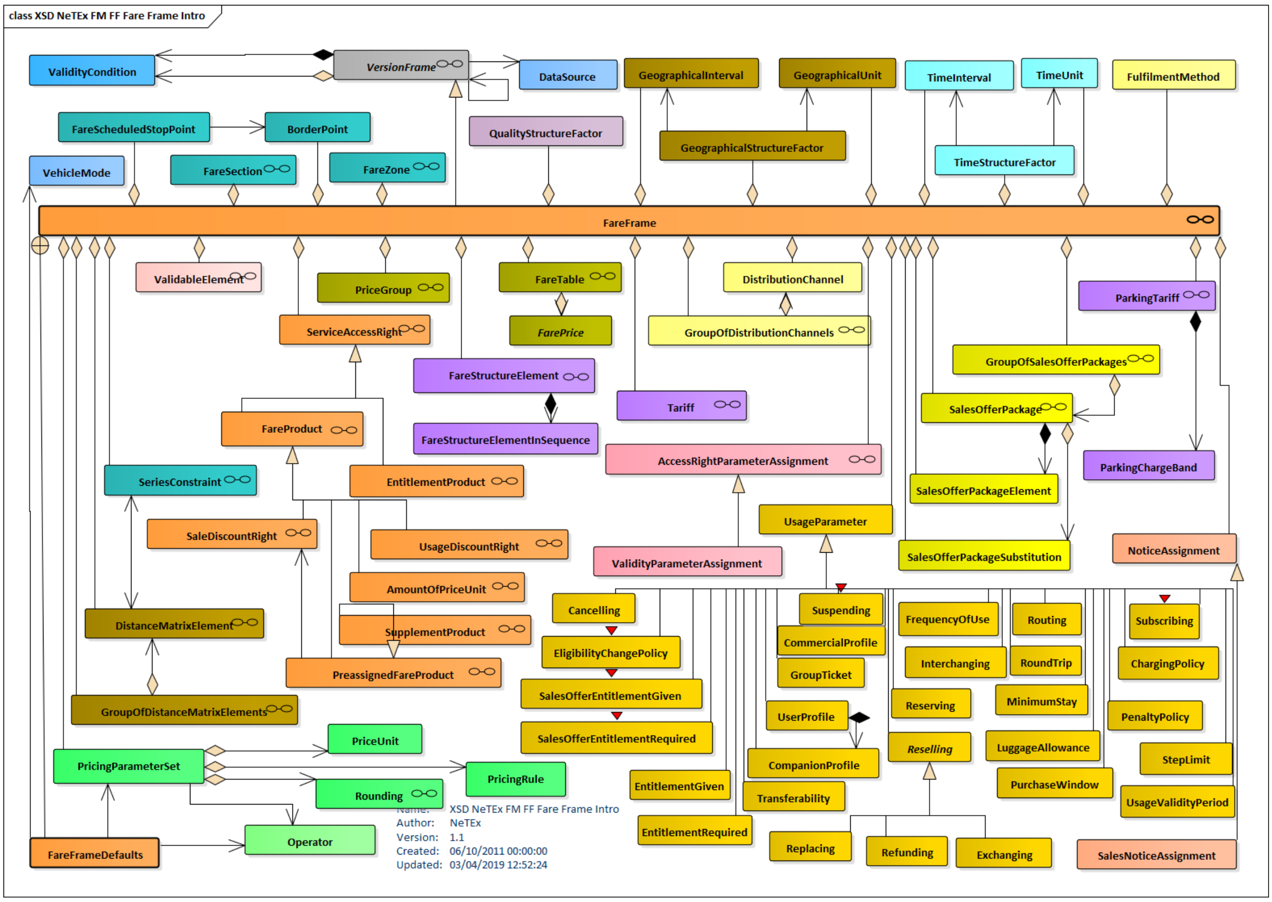


Figure 15 — Fare Frame Contents – Physical Model (UML)

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

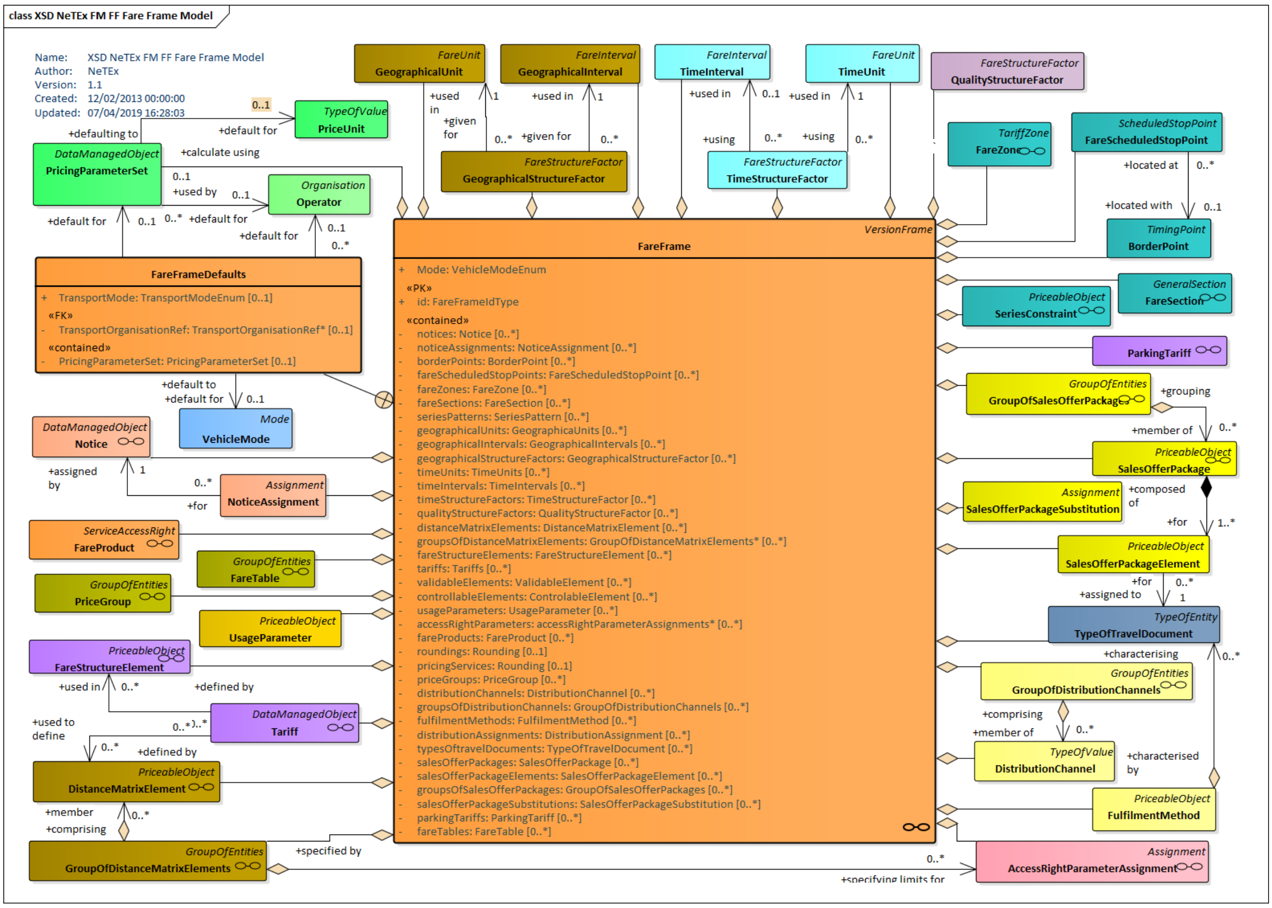


Figure 16 — Fare Frame – Physical Model Detail (UML)

7.3.2.3 Fare Frame — Attributes and XSD

FARE FRAMEs group together sets of fare data for exchange.

7.3.2.3.1 FareFrame – Model Element

A set of Fare data elements (FARE STRUCTURE ELEMENTs, FARE PRODUCTs, FARE PRICEs, etc.) to which the same VALIDITY CONDITIONs have been assigned.

Table 3 – *FareFrame* – Element

Classifi-cation	Name	Type	Cardi nality	Description
::>	::>	*VersionFrame*	::>	FARE FRAME inherits from VERSION FRAME.
«PK»	***id***	*FareFrameIdType*	1:1	Identifier of FARE FRAME.
«FK»	***ServiceCalendarRef***	*ServiceCalendarRef*	0:1	Reference to SERVICE CALENDAR for FARE FRAME.
XGRP	***FareDefaultsGroup***	***xmlGroup***	0:1	Default values for pricing calculations - see below.
XGRP	***NoticesInFrameGroup***	***xmlGroup***	0:1	NOTICE elements used in frame. See below.

XGRP	***FareRoutingInFrame-Group***	***xmlGroup***	0:1	FARE ROUTING elements used in frame. See below.
XGRP	***FareSeriesInFrameGroup***	***xmlGroup***	0:1	FARE SERIES CONSTRAINT elements used in frame. See below.
XGRP	***DistanceMatrixElementsInFrameGroup***	***xmlGroup***	0:1	DISTANCE MATRIX ELEMENT elements used in frame. See below.
XGRP	***FareStructure-ElementsInFrameGroup***	***xmlGroup***	0:1	FARE STRUCTURE elements used in frame. See below.
XGRP	***ValidableElements-InFrameGroup***	***xmlGroup***	0:1	VALIDABLE ELEMENT elements used in frame. See below.
XGRP	***FareProduct-ElementsInFrameGroup***	***xmlGroup***	0:1	FARE PRODUCT elements IN FRAME. See below.
XGRP	***FarePriceElements-InFrameGroup***	***xmlGroup***	0:1	FARE PRICE elements IN FRAME. See below.
XGRP	***SalesDistribution-InFrameGroup***	***xmlGroup***	0:1	SALES DISTRIBUTION elements IN FRAME. See below.
XGRP	***TravelDocuments-InFrameGroup***	***xmlGroup***	0:1	TRAVEL DOCUMENTs elements IN FRAME. See below.
XGRP	***SalesOfferPackage-InFrameGroup***	***xmlGroup***	0:1	SALES OFFER PACKAGE elements IN FRAME. See below.
XGRP	***ParkingTariffInFrame-Group***	***xmlGroup***	0:1	PARKING TARIFF elements IN FRAME. See below.

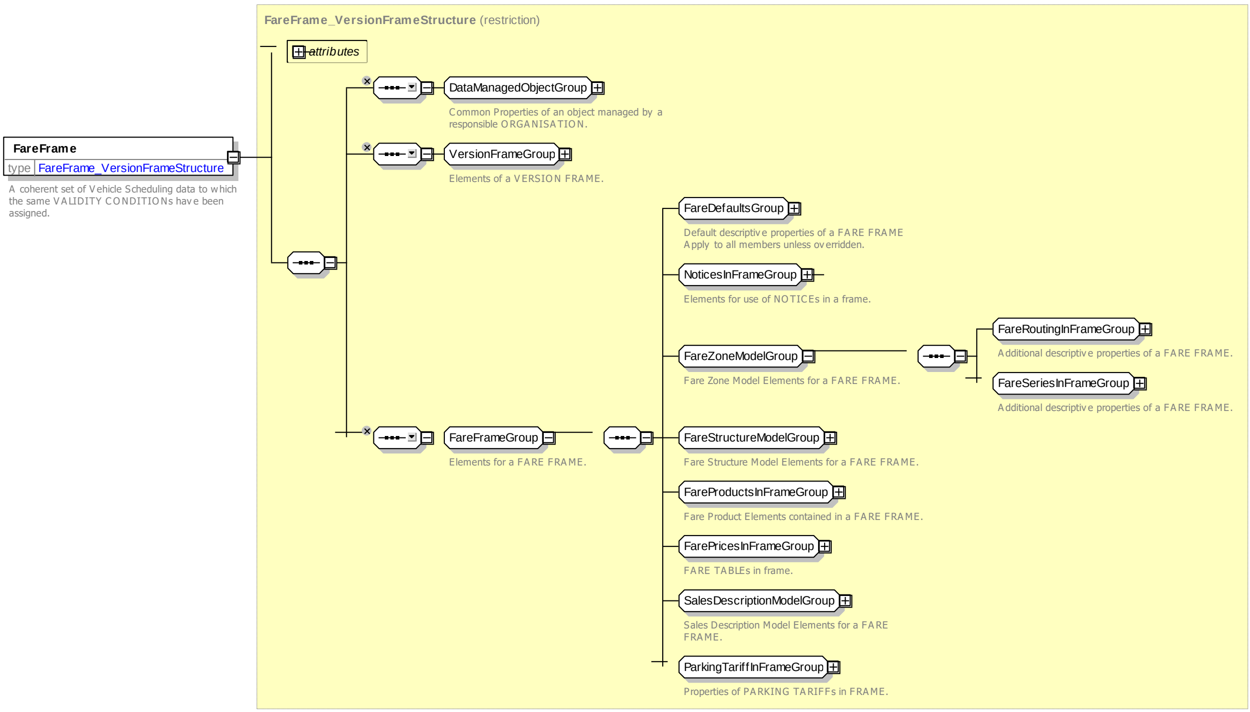


Figure 17 — *FareFrame* — XSD

7.3.2.3.2 ***FareDefaultsGroup — XML Group***

The set of default values for pricing etc., to be used for elements in a frame if not explicitly specified on individual elements.

Table 4 – *FareFrame / FareDefaults* – Element

Classifi-cation	Name	Type	Card in-ality	Description
«enum»	**Mode**	*VehicleModeEnum*	0:1	Default vehicle MODE for FARE FRAME. See Part1.
«FK»	***Transport-OrganisationRef***	*(TransportOrganisationRef) OperatorRef* | *AuthorityRef*	0:1	Default TRANSPORT ORGANISATION for FARE FRAME. See Part1.
«FK»	***Pricing-ParameterSet***	*PricingParameterSet*	0:1	PRICING PARAMETER SET associated with FARE FRAME. See later.

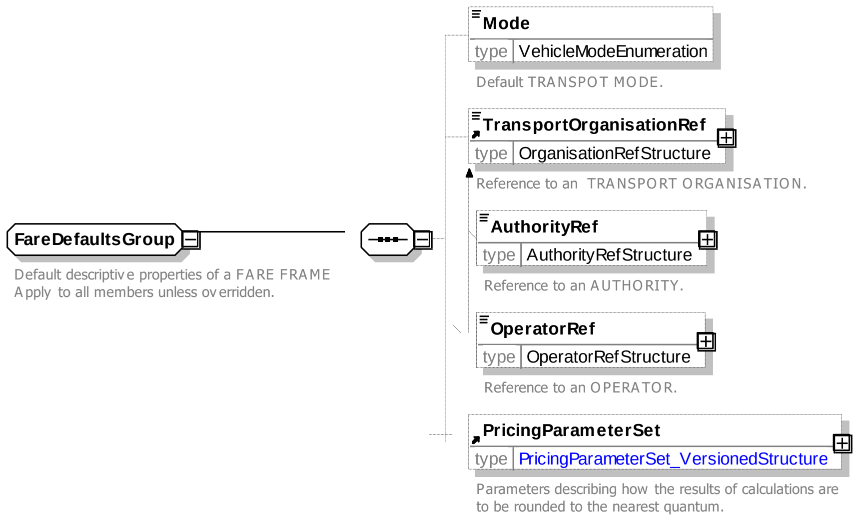


Figure 18 — *FareDefaultsGroup* — XSD

7.3.2.3.3 *NoticesInFrameGroup* — *XML Group*

The ***NoticesInFrameGroup*** holds the NOTICE related elements for the frame***.***

Table 5 – *NoticesInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***notices***	<u>*Notice*</u>	0:*	NOTICEs in the frame. See Part1 for a definition of the Notice Element.
«cntd»	***notice-Assignments***	<u>*NoticeAssignment*</u>	0:*	NOTICE ASSIGNMENTs in the frame. See NeTEx Part1 & Part2 for a definition of the NOTICE ASSIGNMENT element.

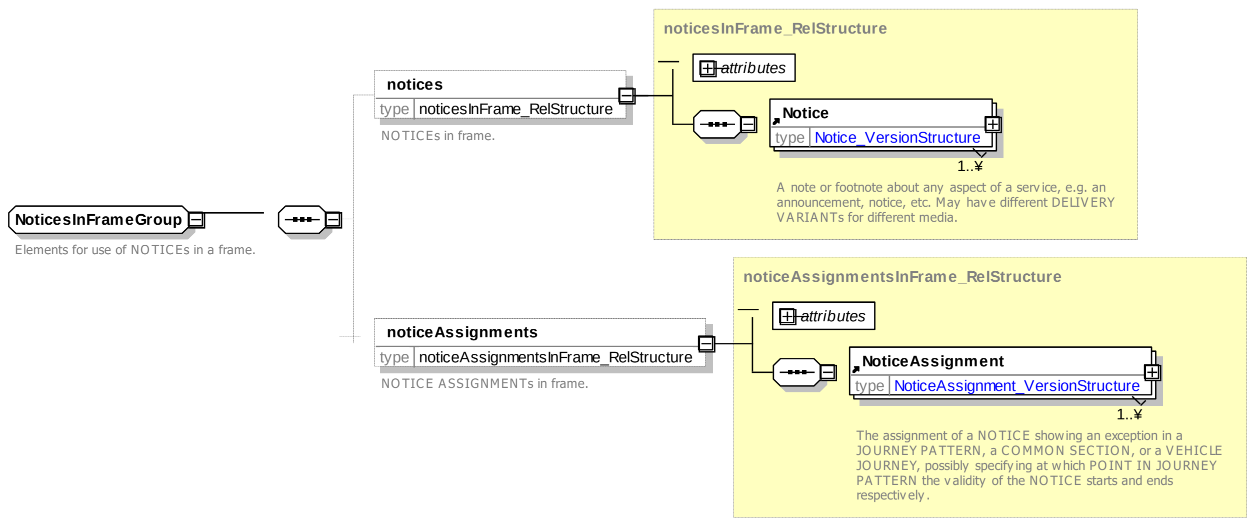


Figure 19 — NoticesInFrameGroup — XSD

7.3.2.3.4 Fare Zone — *Groups*

7.3.2.3.4.1 *FareRoutingInFrameGroup — XML Group*

The ***FareRoutingInFrameGroup*** holds the elements in the frame that relate the fare structures to the network.

Table 6 – *FareRoutingInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***borderPoints***	*BorderPoint*	0:*	BORDER POINTs in FARE FRAME.
«cntd»	***fareScheduled-StopPoints***	*FareScheduledStopPoint*	0:*	FARE SCHEDULED STOP POINTs in FARE FRAME.
«cntd»	***fareZone***	*FareZone*	0:*	FARE ZONEs in the FARE FRAME.
«cntd»	***fareSections***	*FareSection*	0:*	FARE SECTIONs in the FARE FRAME.

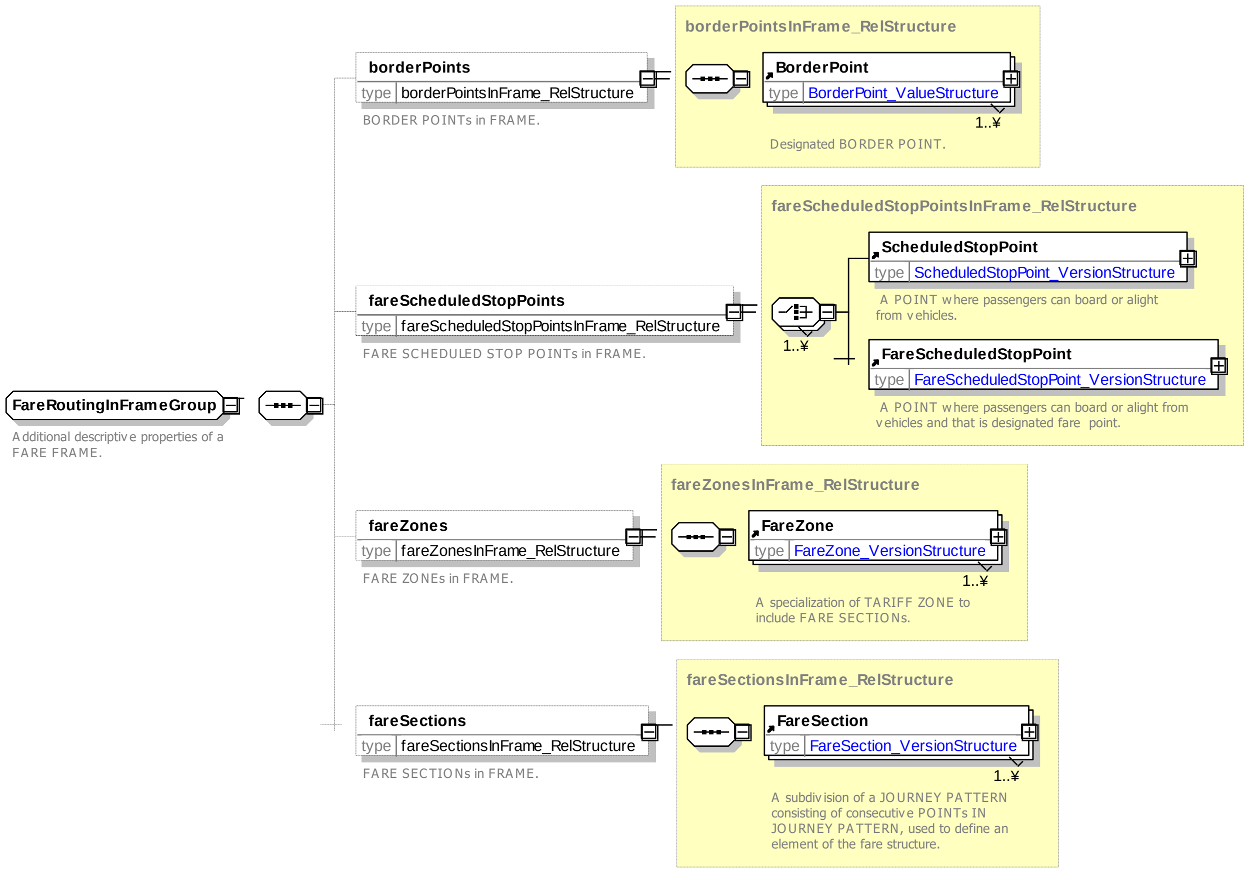


Figure 20 — *FareRoutingInFrameGroup* — XSD

7.3.2.3.4.2 *FareSeriesInFrameGroup — XML Group*

The ***FareSeriesInFrameGroup*** holds the SERIES CONSTRAINT elements for the frame. These limit the allowed routings.

Table 7 – *FareSeriesInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***series-Constraints***	*SeriesConstraint*	0:*	SERIES CONSTRAINTs in the FARE FRAME.

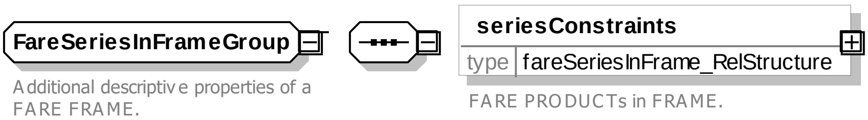


Figure 21 — *FareSeriesInFrameGroup* — XSD

7.3.2.3.5 Fare Structure — *Groups*

7.3.2.3.5.1 *FareStructureModelGroup — XML Group*

The ***FareStructureModelGroup*** holds the FARE STRUCTURE elements for the frame.

Table 8 – *FareStructureModelGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
[XGRP]	***Geographic-Factors-InFrameGroup***	*GeographicFactors-InFrameGroup*	0:*	Elements for GEOGRAPHIC FACTORs in FARE FRAME.
«cntd»	***TimeFactors-InFrameGroup***	*TimeFactors-InFrameGroup*	0:*	Elements for TIME FACTORs in FARE FRAME.
«cntd»	***FareStructure-Elements-InFrameGroup***	*FareStructureElements-InFrameGroup*	0:*	Elements for FARE STRUCTURE ELEMENTs in FARE FRAME.
«cntd»	***ValidableElements-InFrameGroup***	*ValidableElements-InFrameGroup*	0:*	Elements for VALIDABLE ELEMENTs in FARE FRAME.

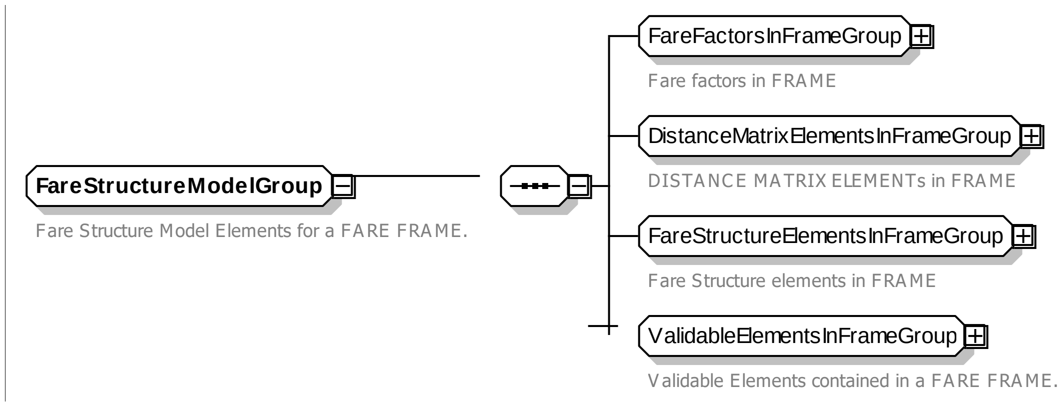


Figure 22 — *FareStructureModelGroup* — XSD

7.3.2.3.5.2 *FareFactorsInFrameGroup — XML Group*

The ***FareFactorsInFrameGroup*** holds the FARE FACTOR elements for the frame, including units, intervals and actual factors.

Table 9 – *FareFactorsInFrameGroup* – Group

Classification	Name	Type	Cardinality	Description
«cntd»	***geographical-Units***	*GeographicaUnits*	0:*	GEOGRAPHICAL UNITs in FARE FRAME.
«cntd»	***geographical-Intervals***	*GeographicalIntervals*	0:*	GEOGRAPHICAL INTERVALs in FARE FRAME.
«cntd»	***geographical-StructureFactors***	*Geographical-StructureFactor*	0:*	GEOGRAPHICAL STRUCTURE FACTORs in FARE FRAME.
«cntd»	***timeUnits***	*TimeUnits*	0:*	TIME UNITs in FARE FRAME.
«cntd»	***timeIntervals***	*TimeIntervals*	0:*	TIME INTERVALs in FARE FRAME.
«cntd»	***timeStructure-Factors***	*TimeStructureFactor*	0:*	TIME STRUCTURE FACTORs in FARE FRAME.
«cntd»	***qualityStructure-Factors***	*QualityStructureFactor*	0:*	QUALITY STRUCTURE FACTORs in FARE FRAME.

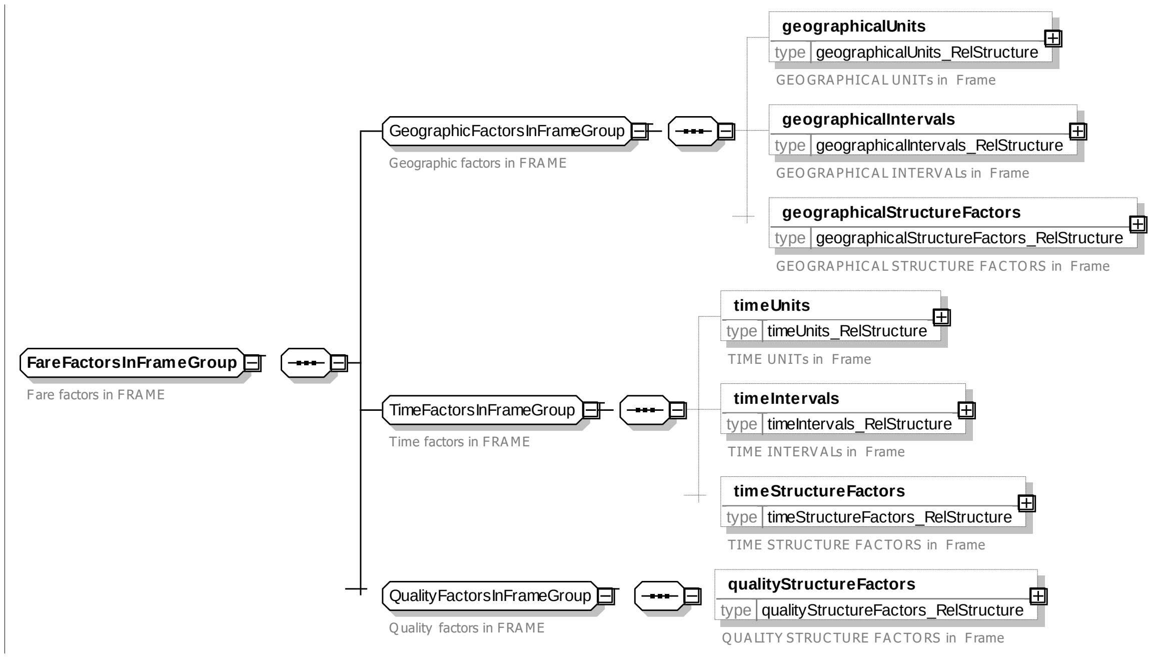


Figure 23 — *FareFactorsInFrameGroup*— XSD

7.3.2.3.5.3 *DistanceMatrixeElementsInFrameGroup — XML Group*

The ***DistanceMatrixeElementsInFrameGroup*** holds the DISTANCE MATRIX ELEMENT elements for the frame.

Table 10 – *DistanceMatrixeElementsInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***distanceMatrix-Elements***	*DistanceMatrixElement*	0:*	DISTANCE MATRIX ELEMENTs in the FARE FRAME.
«cntd»	***groupsOf-DistanceMatrix-Elements***	*GroupOfDistance-MatrixElements*	0:*	GROUPs OF DISTANCE MATRIX ELEMENTs in the FARE FRAME.

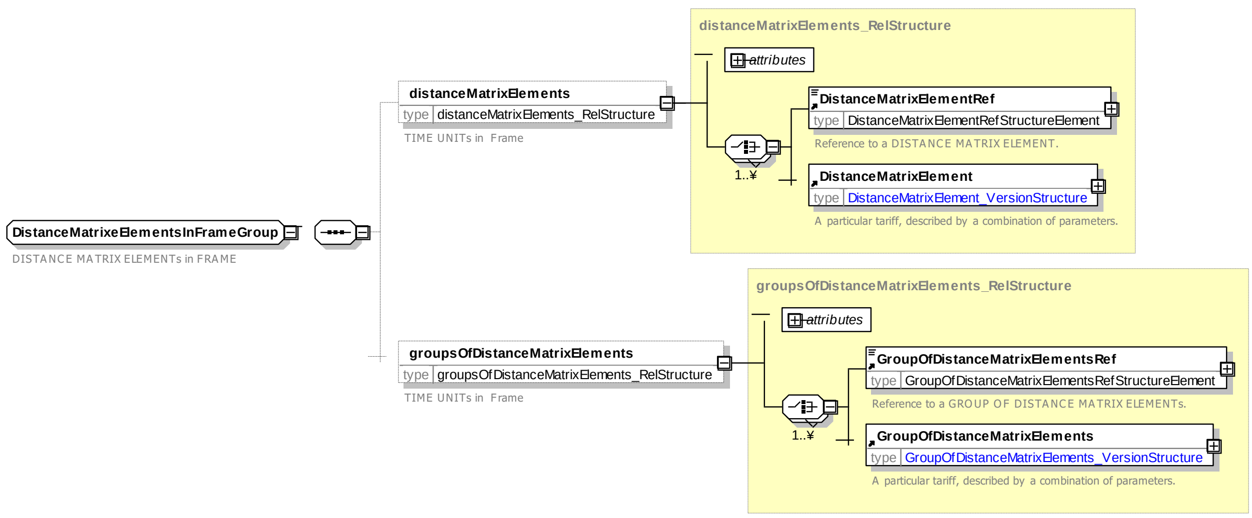


Figure 24 — *DistanceMatrixElementsInFrameGroup* — XSD

7.3.2.3.5.4 *FareStructureElementsInFrameGroup — XML Group*

The ***FareStructureElementsInFrameGroup*** holds the fare structure definition elements for the frame, including, FARE STRUCTURE ELEMENTs, and TARIFFs.

Table 11 – *FareStructureElementsInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***fareStructure-Elements***	*FareStructureElement*	0:*	FARE STRUCTURE ELEMENTs in FARE FRAME.
«cntd»	***tariffs***	*Tariff*	0:*	TARIFFs in the FARE FRAME.

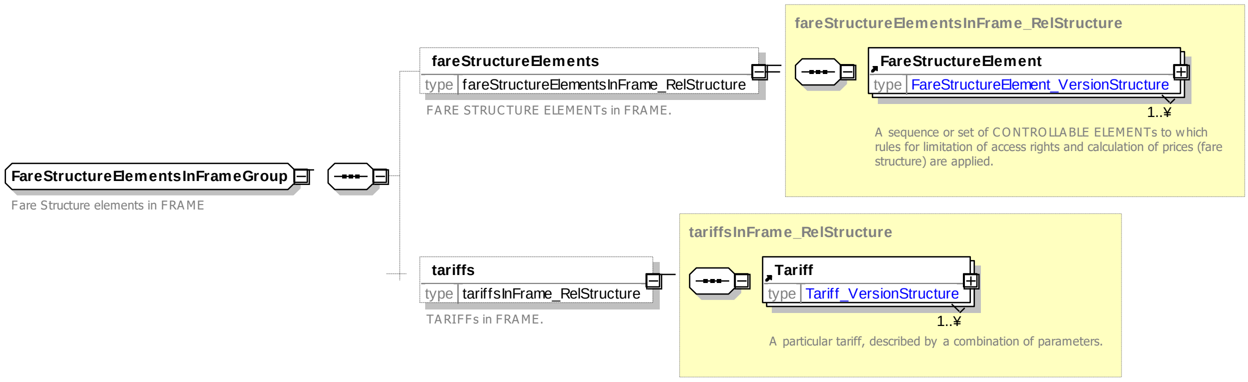


Figure 25 — *FareStructureElementsInFrameGroup* — XSD

7.3.2.3.5.5 *ValidableElementsInFrameGroup — XML Group*

The ***ValidableElementsInFrameGroup*** holds structural fare definition elements for the frame, including VALIDABLE ELEMENTs, CONTROLLABLE ELEMENTs.

Table 12 – *ValidableElementsInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***validable-Elements***	*ValidableElement*	0:*	VALIDABLE ELEMENTs in FARE FRAME.
«cntd»	***controllable-Elements***	*ControllableElement*	0:*	CONTROLLABLE ELEMENTs in FARE FRAME.

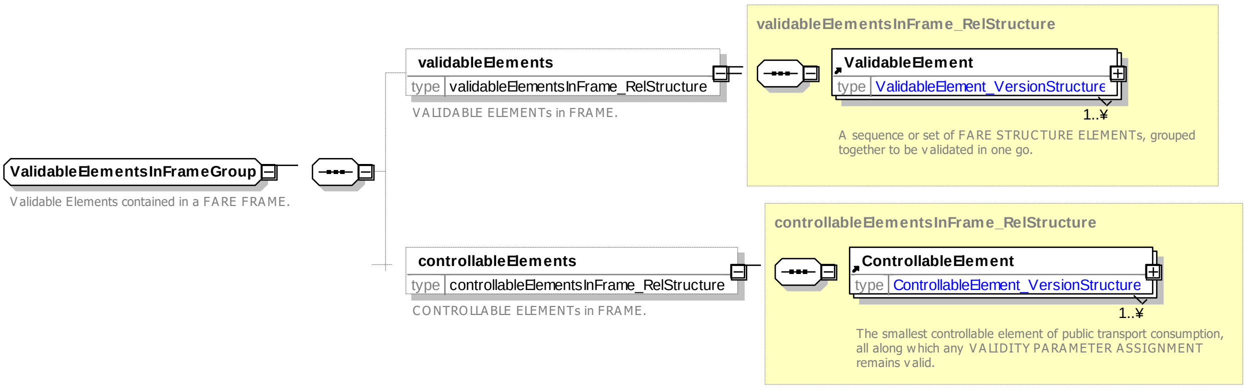


Figure 26 — *ValidableElementsInFrameGroup* — XSD

7.3.2.3.6 Fare Products — *Groups*

7.3.2.3.6.1 *FareProductsInFrameGroup — XML Group*

The ***FareProductsInFrameGroup*** holds the fare definition elements for the frame, including FARE USAGE PARAMETERs, ACCESS RIGHT PARAMETERs, FARE PRODUCTs, TARIFFs, and FARE PRICEs.

Table 13 – *FareProductsInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***usage-Parameters***	*UsageParameter*	0:*	USAGE PARAMETERs in the FARE FRAME.
«cntd»	***accessRight-Parameter-Assignments***	*AccessRightParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs in the FARE FRAME.
«cntd»	***fareProducts***	*FareProduct*	0:*	FARE PRODUCTs in the FARE FRAME.

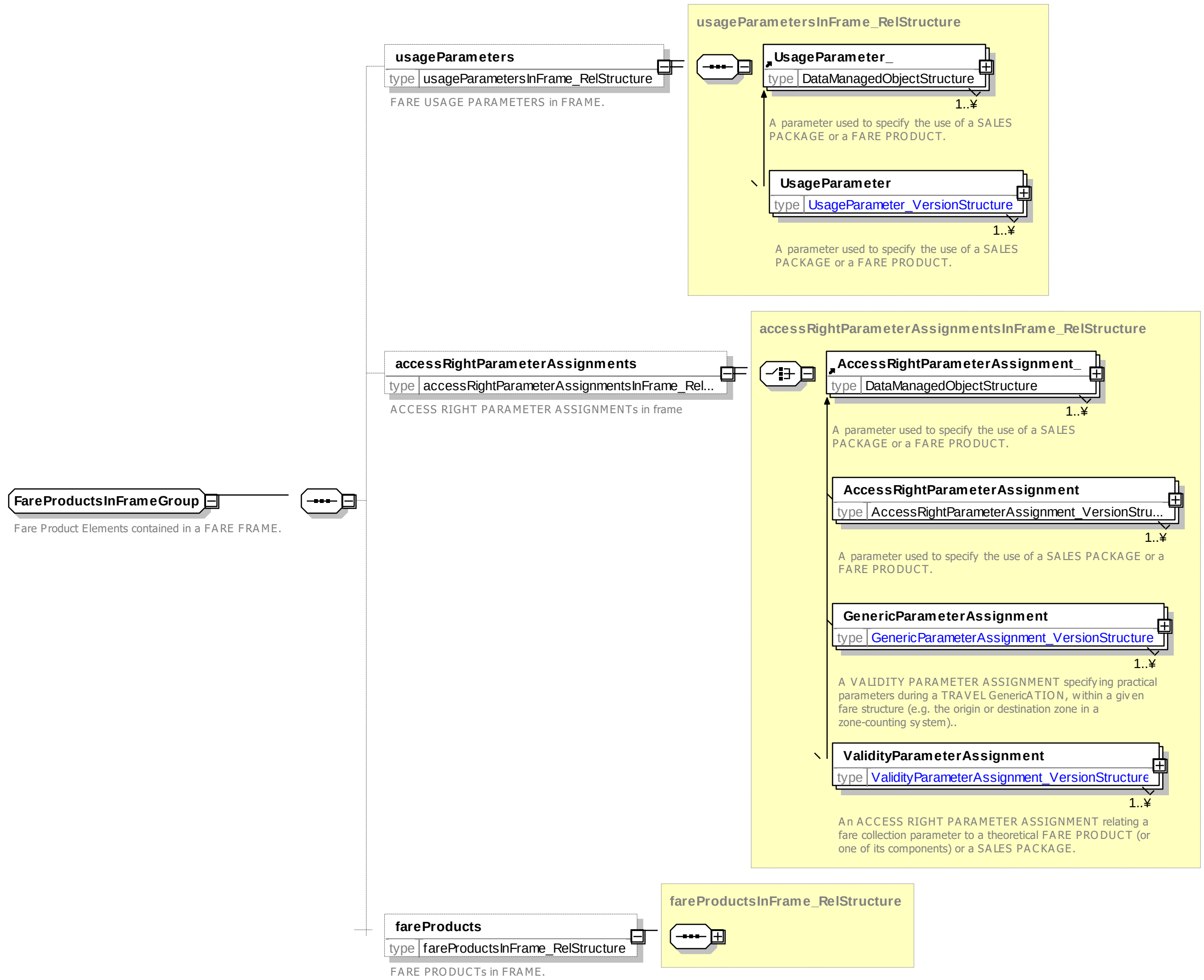


Figure 27 — *FareProductsInFrameGroup* — XSD

7.3.2.3.7 *FarePricesInFrameGroup — XML Group*

The ***FarePricesInFrameGroup*** holds the fare price elements for the frame, including PRICE GROUPs and FARE TABLEs.

Table 14 – *FarePricesInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***priceGroups***	*PriceGroup*	0:*	PRICE GROUPs in the FARE FRAME.
«cntd»	***fareTables***	*FareTable*	0:*	FARE TABLEs in the FARE FRAME.

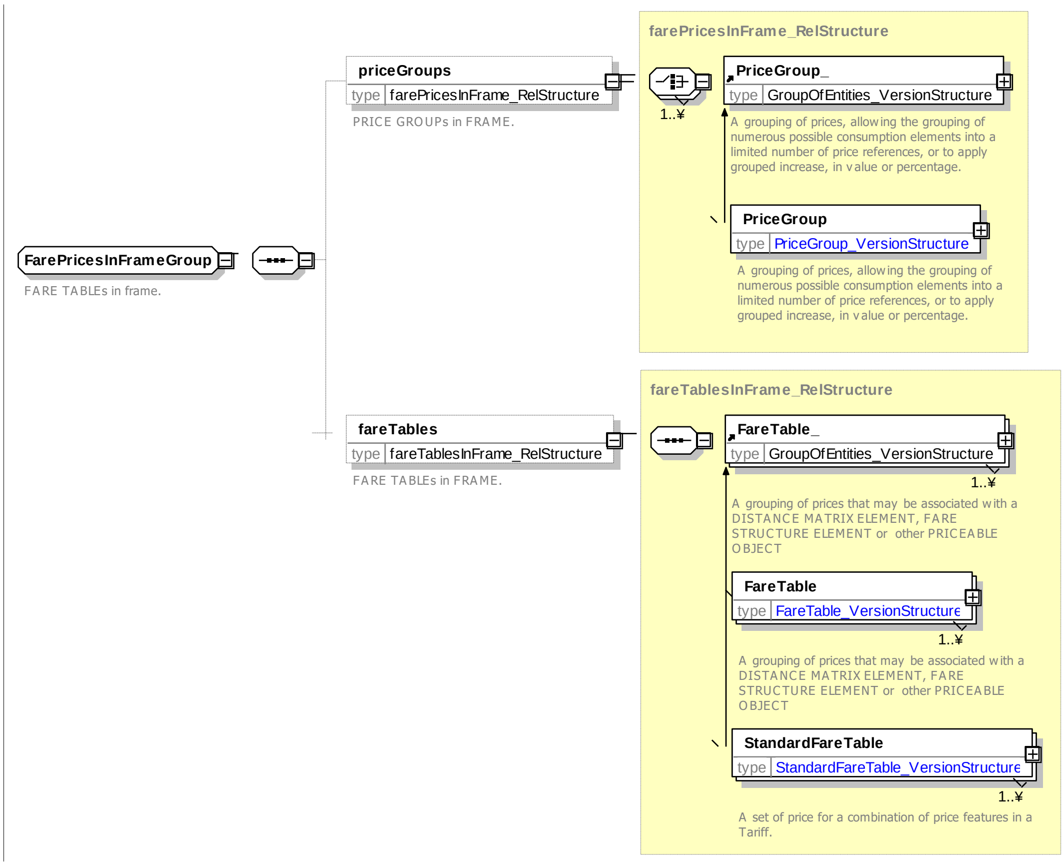


Figure 28 — *FarePricesInFrameGroup* — XSD

7.3.2.3.8 Sales Description — *Groups*

7.3.2.3.8.1 *SalesDistributionInFrameGroup — XML Group*

The ***SalesDistributionInFrameGroup*** holds the sales distribution elements for the frame including DISTRIBUTION CHANNELs, DISTRIBUTION ASSIGNMENTs and FULFILMENT METHODs.

Table 15 – *SalesDistributionInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***distribution-Channels***	*DistributionChannel*	0:*	DISTRIBUTION CHANNELs in FARE FRAME.
«cntd»	***groupsOf-Distribution-Channels***	*GroupOfDistribution-Channels*	0:*	GROUPs OF DISTRIBUTION CHANNELS in FARE FRAME.
«cntd»	***fulfilment-Methods***	*FulfilmentMethod*	0:*	FULFILMENT METHODs in FARE FRAME.

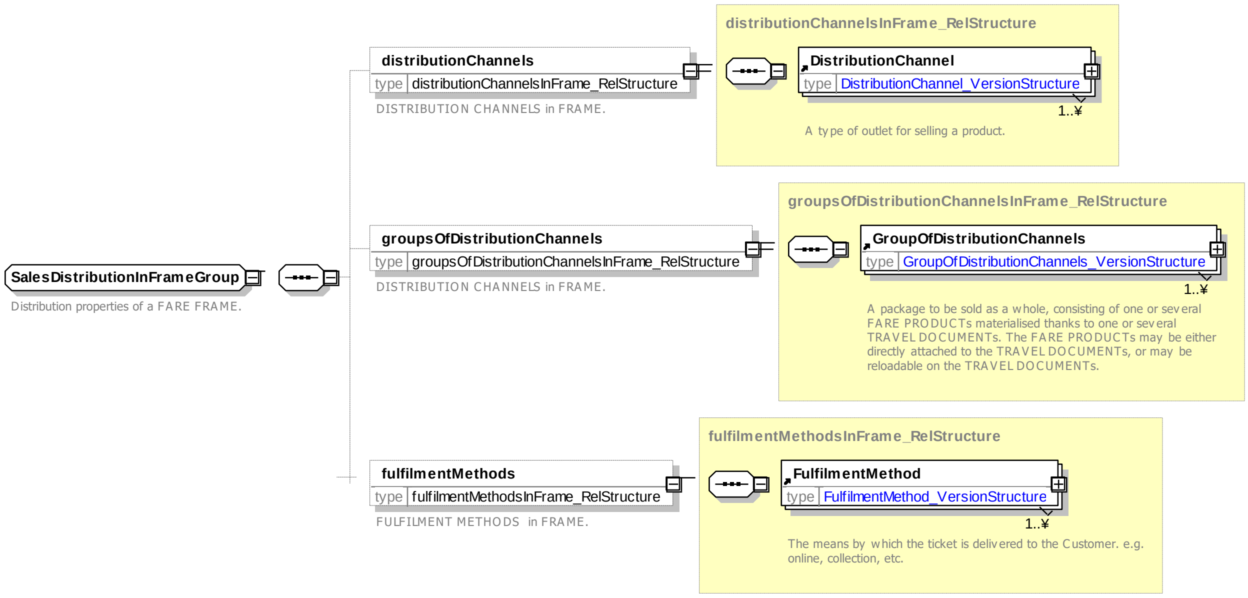


Figure 29 — ***SalesDistributionInFrameGroup*** — XSD

7.3.2.3.8.2 *TravelDocumentsInFrameGroup — XML Group*

The ***TravelDocumentsInFrameGroup*** holds the TRAVEL DOCUMENT elements for the frame.

Table 16 – *TravelDocumentsInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***travelDocuments***	*TravelDocument*	0:*	TRAVEL DOCUMENTs in the FARE FRAME.

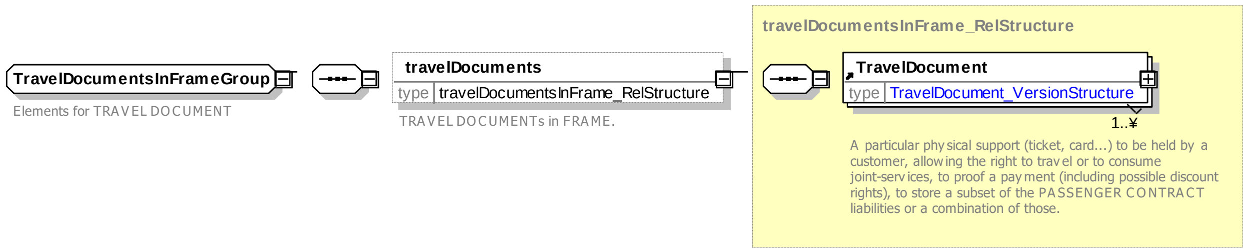


Figure 30 — ***TravelDocumentsInFrameGroup*** — XSD

7.3.2.3.8.3 *SalesOfferPackageInFrameGroup — XML Group*

The ***SalesOfferPackageInFrameGroup*** holds the SALES OFFER PACKAGE elements for the frame.

Table 17 – *SalesOfferPackageInFrameGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description

«cntd»	***salesOffer-Packages***	*SalesOfferPackage*	0:*	SALES OFFER PACKAGEs in FARE FRAME.
«cntd»	***salesOffer-Package-Elements***	*SalesOfferPackageElement*	0:*	SALES OFFER PACKAGE ELEMENTs in FARE FRAME.
«cntd»	***salesOffer-Package-Substitutions***	*SalesOfferPackage-Substitution*	0:*	SALES OFFER PACKAGE SUBSTITUTIONs in FARE FRAME.
«cntd»	***groupsOfSales-OfferPackages***	*GroupOfSalesOfferPackages*	0:*	GROUPs OF SALES OFFER PACKAGEs in FARE FRAME.
«cntd»	***distribution-Assignments***	*DistributionAssignment*	0:*	DISTRIBUTION ASSIGNMENTs in the FARE FRAME.

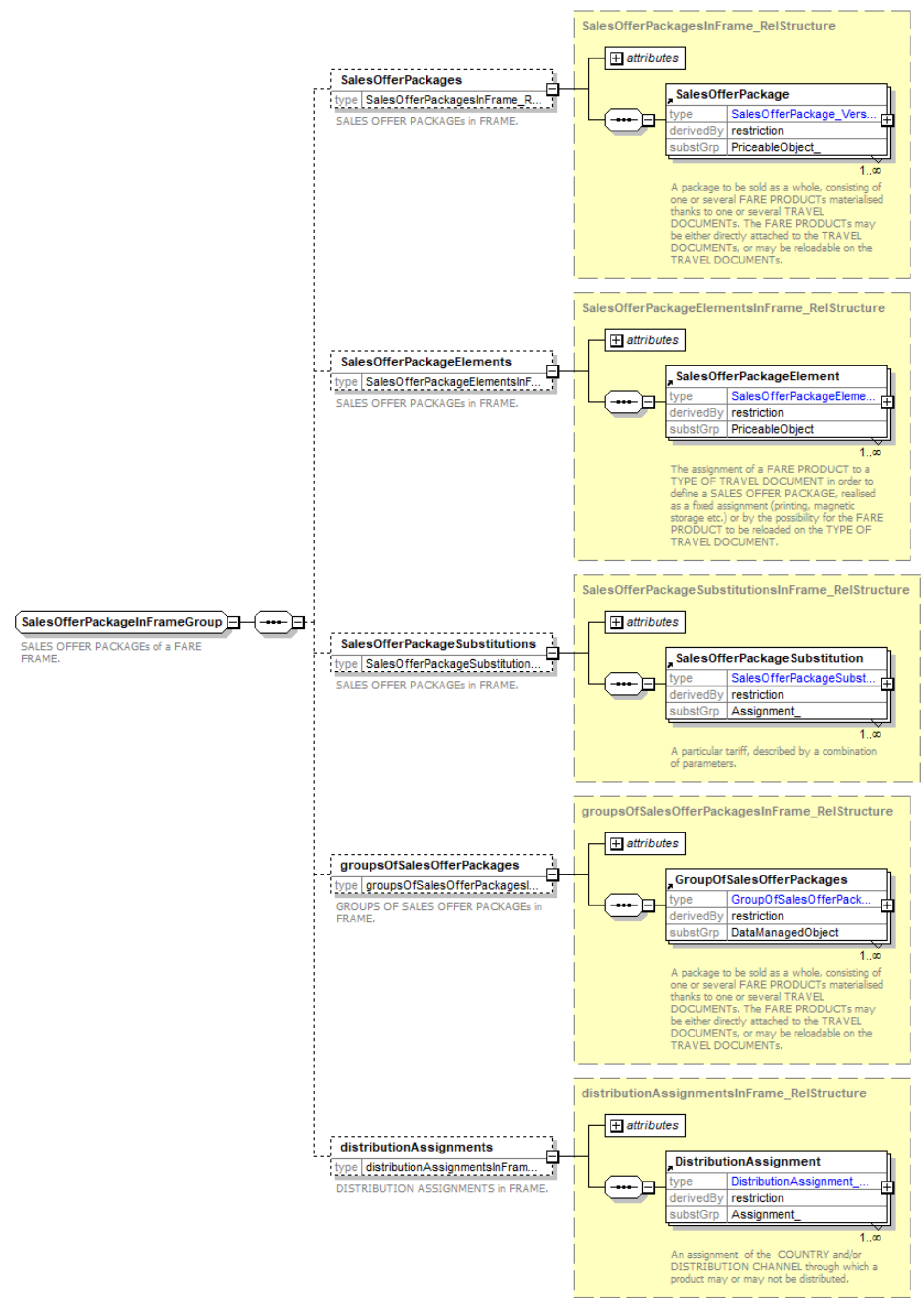


Figure 31 — *SalesOfferPackageInFrameGroup* — XSD

7.3.2.3.9 **ParkingTariff — *Groups***

7.3.2.3.9.1 ***ParkingTariffInFrameGroup— XML Group***

The ***ParkingTariffInFrameGroup*** holds the PARKING TARIFF elements for the frame.

Table 18 – *ParkingTariffInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description

«cntd»	***parkingTariffs***	*ParkingTariff*	0:*	PARKING TARIFFs in FARE FRAME.

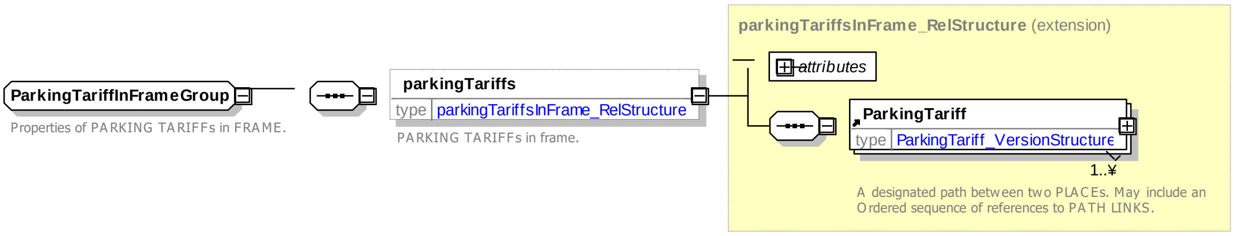


Figure 32 — *ParkingTariffInFrameGroup*— XSD

7.4 Reusable Fare Components

7.4.1 Fare Zone

7.4.1.1 FARE ZONE – Conceptual MODEL

NeTEx Part1 includes the concept of a TARIFF ZONE, which can be used to define the permanent fare zones of a system. A given SCHEDULED STOP POINT can belong to one or more TARIFF ZONEs. The NeTEx Part3 FARE ZONE MODEL presents additional concepts relating to the network that can be used additionally to underpin fare structures.

— FARE SCHEDULED STOP POINT extends a SCHEDULED STOP POINT with additional fare related attributes.

— A FARE ZONE is a specialization of TARIFF ZONE that may have FARE SECTIONs associated with it.

— FARE SECTIONs allow arbitrary sections of the network to be associated with a specific FARE ZONE.

— A BORDER POINT is used to distinguish certain points (often but not necessarily SCHEDULED STOP POINTs and/or TIMING POINTs) as having special significance for calculating international fares.

— A SERIES CONSTRAINT allows constraints on specific routings to be specified, for example that journeys may or must go via particular via points. They are mainly used for rail and may comprise one or more FARE POINTs in JOURNEY PATTERN.

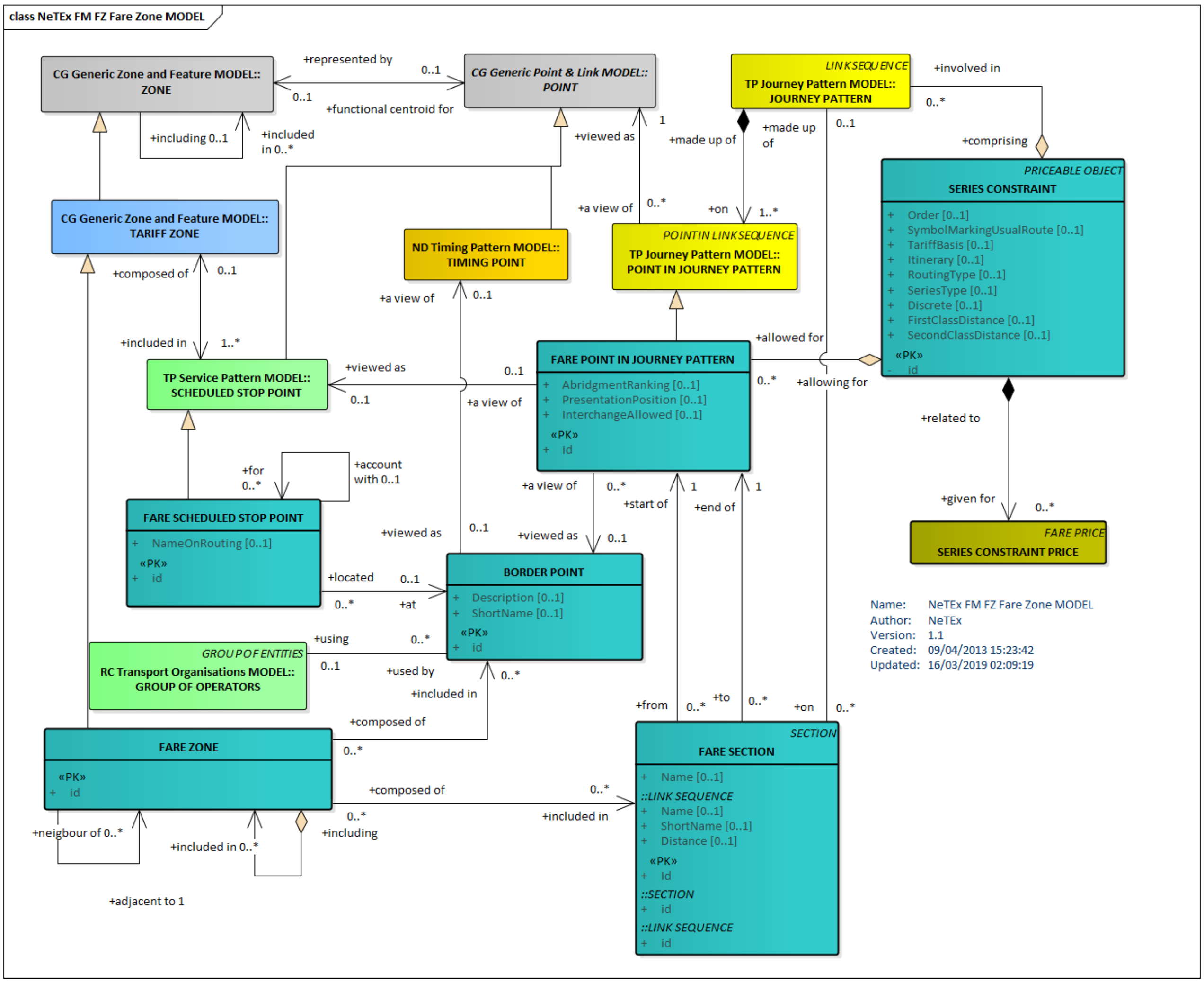


Figure 33 — Fare Zone – Conceptual MODEL (UML)

7.4.1.2 Fare Zone – Conceptual Examples

7.4.1.2.1 Example – Series

A SERIES CONSTRAINT allows a routing constraint to be specified concisely as a series of FARE POINTs in JOURNEY PATTERN – series are used in TAP TSI for many rail products. The TAP TSI B1 document defines by a notation for describing the constraints to staff and passengers as a "route description" or itinerary on a ticket.

For example: the following set of diagrams show four paths through a simple network from an origin station A to a destination station D, with two designated via stations B and C. Use of the network could be restricted to a specific path, as in (i) *[A * B * D]* (Figure 35) or (ii) *[A * C * D] (*Figure 35)*,* or it could allow any path within an envelope delineated by the intermediate stops, as in *[A * (B/C) * D].* In this case, the route description (B/C) does not mean that the passenger must take a train passing exactly by either B or C, it means that the passenger has a choice of all routes between a "tariff envelope" of a station on the left and a station on the right (route range), as shown in (i), (ii), (iii) *(*Figure 36)*,* and (iv) *(*Figure 37). Thus, a ticket [A * D] with SERIES CONSTRAINT (B/C) would allow travel along any of the following four routes:

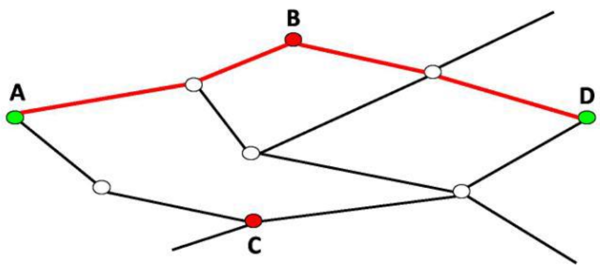


Figure 34 — Series Example (i) [A * B * D]

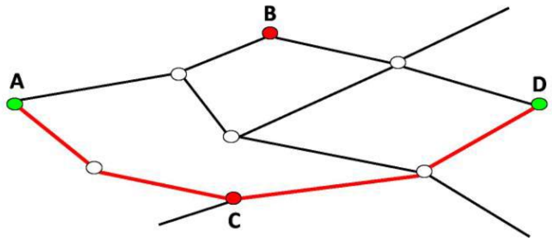


Figure 35 — Series Example (ii) [A * C * D]

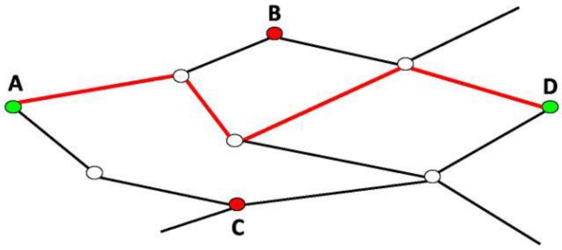


Figure 36 — Series Example (iii) [A * (B/C) * D]

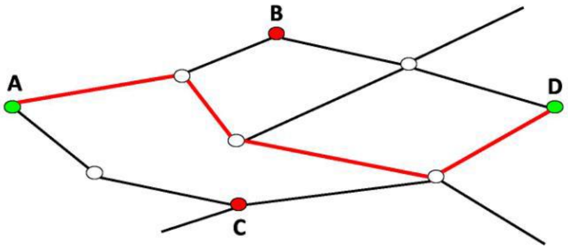


Figure 37 — Series Example (iv) [A * (B/C) * D]

7.4.1.2.1.1 Series — Notation for printed itineraries

A sequence can be indicated on a ticket or other media as a sequence of station names in an itinerary or route description.

Table 19 – Series Notation

Example	Notation		Note
A * B * C * D	Sequence	User must go via B and C	
A *(B/C) * D	Choice	User may travel within envelope of B/C	The order to show B or C is indicated by a Presentation Position

P** [* Y] * **X** [* Z *] **Q *Shown as* ***P** * **X** * **Q***	Abridgement	Intermediate stations may be omitted if there is insufficient space	The relative priority for omitting stations is indicated by an AbridgementRanking

The ***PresentationPosition*** attribute defines where a station appears in a printed route description.

— The value *requiredStation* ("1" in Figure 38) explicitly indicates that the station shall be served, i.e. that it is not on an optional route

— If the station is on an optional route to the left, it is given the value *optionalLeft* ("2" in Figure 38)

— If, however, it is to the right, it is given the value *optionalRight* (“3" in Figure 38).

— Where there is no station at all, *noStation* is entered ("0" in Figure 38).

	Station Code	Position
1st Station	16043	2
2nd Station	26002	3
3rd Station	23002	1
4th Station	27359	1
5th Station	00000	0

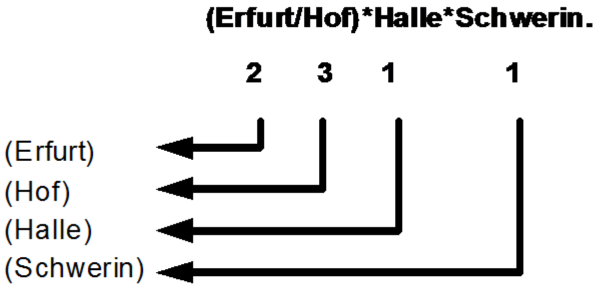


Figure 38 — Example: Use of Series

7.4.1.2.2 **Example – Fare Section**

A FARE SECTION allows designated sections of a network to be allocated to an arbitrary fare zone

For example, the following diagram shows a fare map of the London urban rail network, which designates certain sections as being subject to a TfL tariff and certain sections as being subject to a National Rail tariff. In this case. the allocation of sections to a specific network does not always correspond to their geographical location.

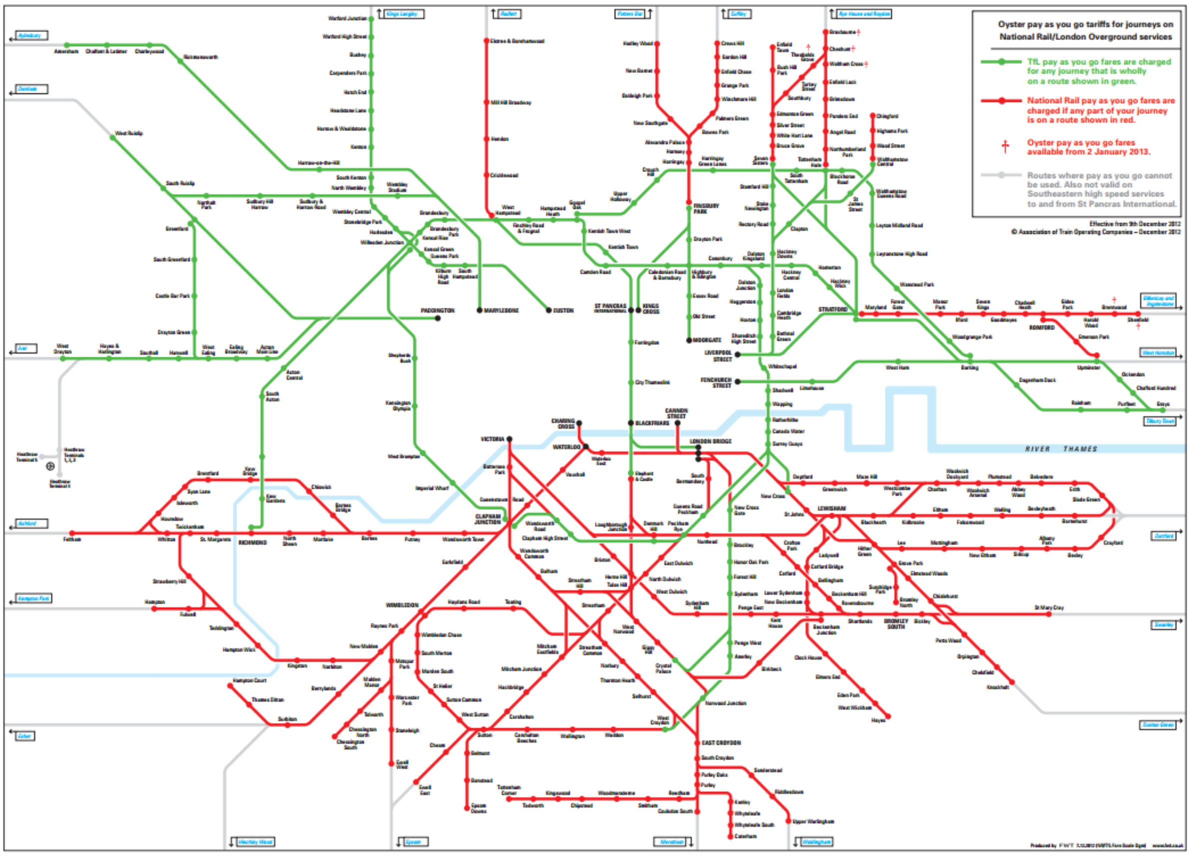


Figure 39 — Example: Transport for London rail fare sections

7.4.1.2.3 **Example – Border Point**

A BORDER POINT is a point on the network marking a boundary for fare calculation that is used by the rail operators both side of the border to determine international fares. A BORDER POINT may or may not be a SCHEDULED STOP POINT.

The following example (Figure 42) shows a BORDER POINT "0847" defined for "*Brest*" that corresponds to stations either side of the border

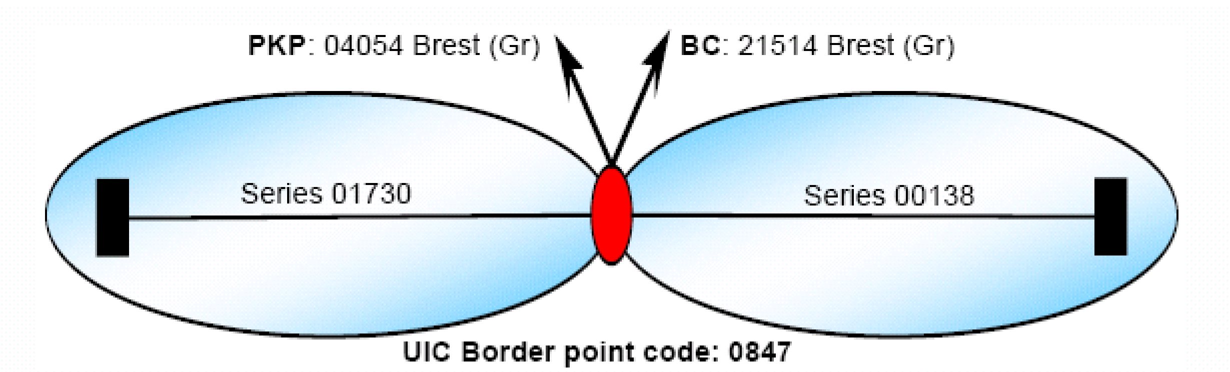


Figure 40 — Example: Border Point for Brest (from Tap/TSI B1)

7.4.1.3 Fare Zone – Physical model

The following diagram shows detailed attributes of the FARE ZONE model.

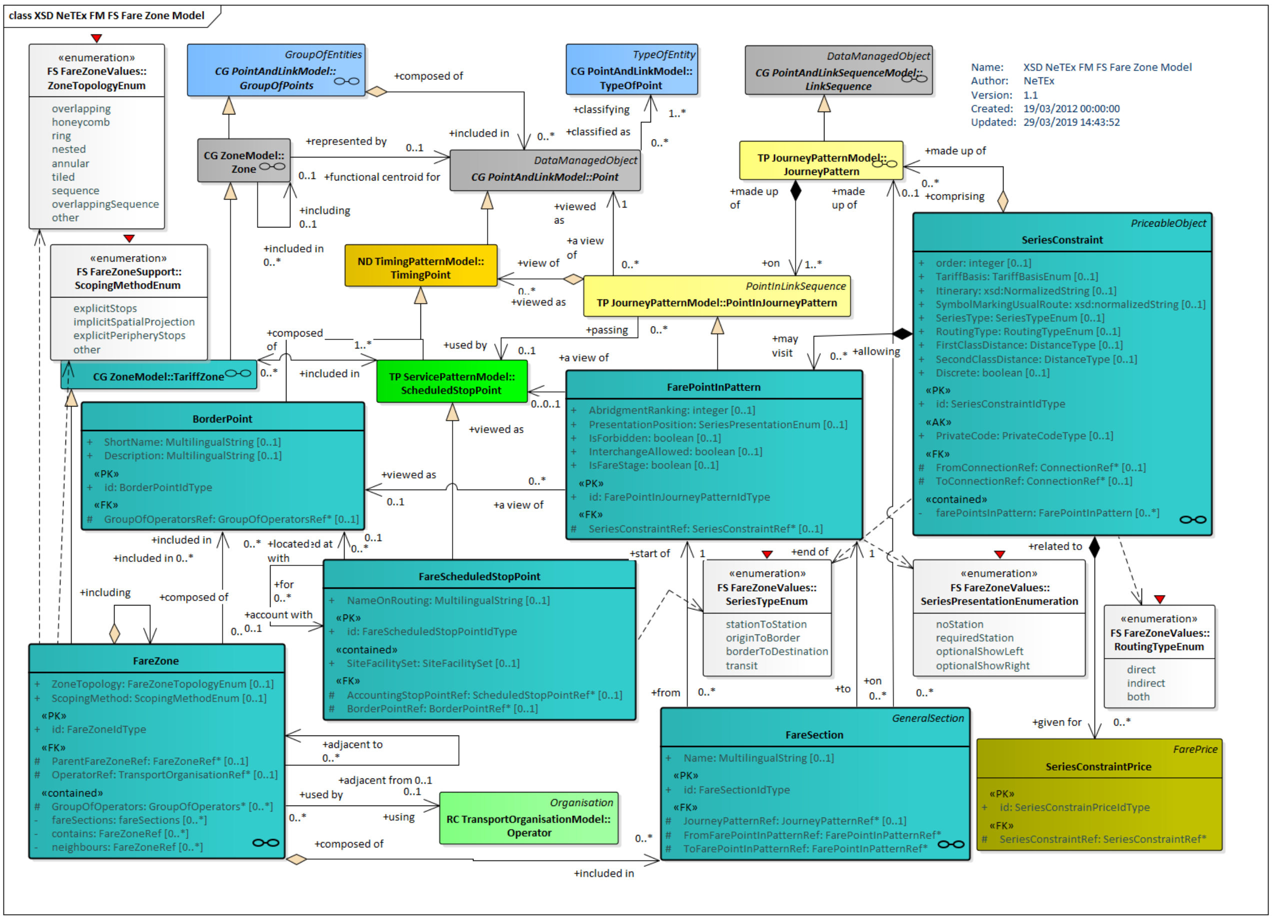


Figure 41 — Fare Zone – Physical Model (UML)

7.4.1.4 Fare Zone Model – Attributes and XSD

7.4.1.4.1 BorderPoint – Model Element

A Point on the Network marking a boundary for fare calculation. May or may not be a SCHEDULED STOP POINT.

Table 20 – *BorderPoint* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TimingPoint*	::>	BORDER POINT inherits from TIMING POINT. See NeTEx Part1.
«PK»	***id***	*BorderPointIdType*	1:1	Identifier of BORDER POINT.
	ShortName	*MultilingualString*	0:1	Short Name of BORDER POINT.

	Description	*MultilingualString*	0:1	Description of BORDER POINT.
«FK»	*GroupOf-OperatorsRef*	*GroupOfOperatorsRef*	0:1	OPERATORs related to BORDER POINT.

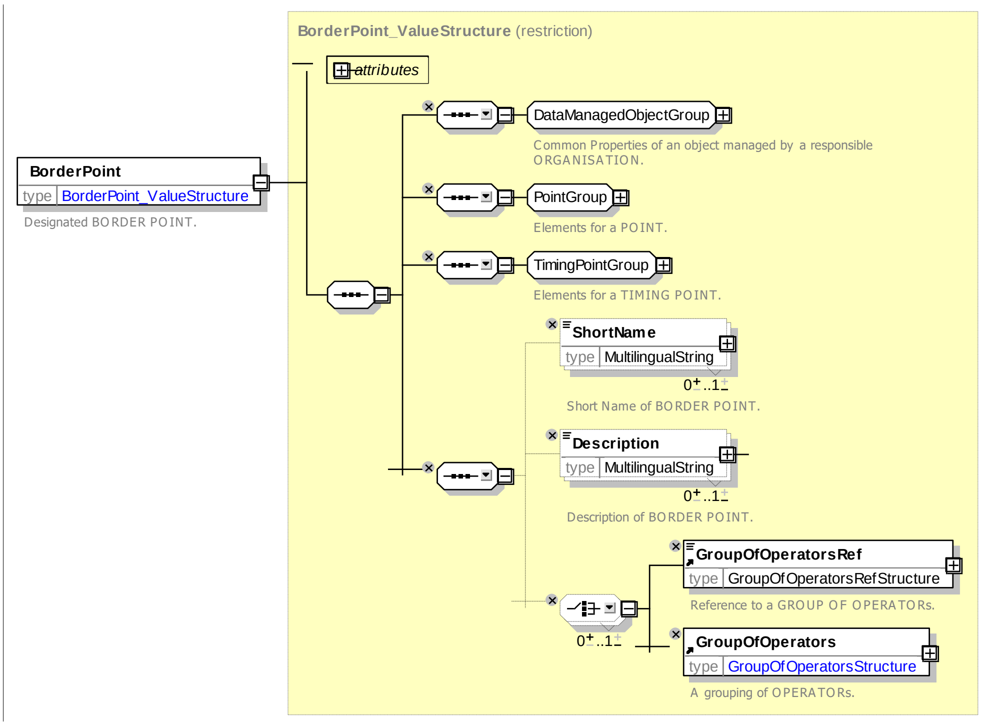


Figure 42 — *BorderPoint* — XSD

7.4.1.4.2 FareScheduledStopPoint – Model Element

A specialisation of SCHEDULED STOP POINT describing a stop with fare accounting and routing characteristics.

Table 21 – *FareScheduledStopPoint* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*ScheduledStopPoint*	::>	FARE SCHEDULED STOP POINT inherits from SCHEDULED STOP POINT. See NeTEx Part1.
«PK»	*id*	*FareStopPointIdType*	1:1	Identifier of FARE SCHEDULED STOP POINT.
	SiteFacilitySet	*SiteFacilitySetRef*	0:1	Set of Facilities available at the station.
	NameOnRouting	*MultilingualString*	0:1	Name to use to indicate station on routings and itineraries.
«FK»	*AccountingStop-PointRef*	*FareScheduled-StopPointRef*	0:1	Identifier of another station to use for accounting purposes for this station.

«FK»	***BorderPointRef***	*BorderPointRef*	0:1	BORDER POINT associated with FARE SCHEDULED STOP POINT.

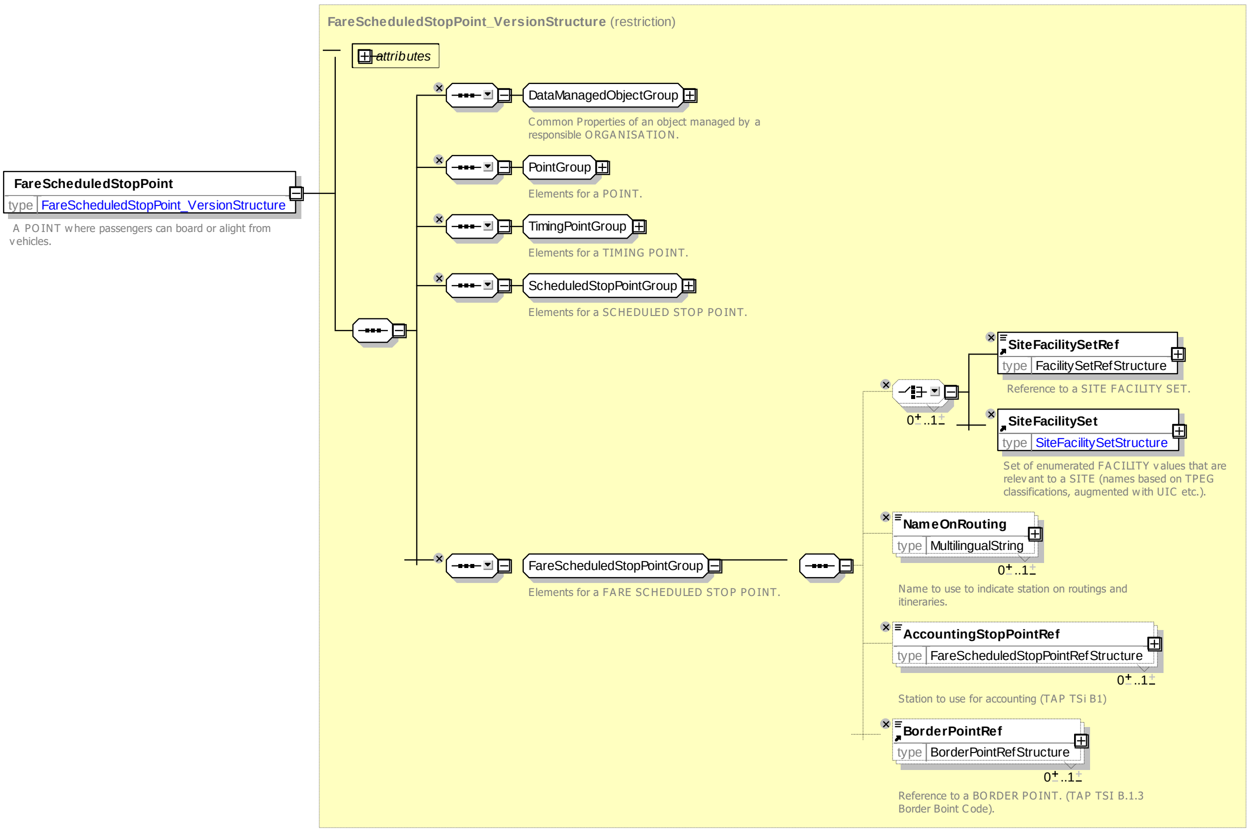


Figure 43 — *FareScheduledStopPoint* — XSD

7.4.1.4.3 FareZone – Model Element

A specialization of TARIFF ZONE to include designated FARE SECTIONs.

Table 22 – *FareZone* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*TariffZone*</u>	::>	FARE ZONE inherits from TARIFF ZONE. See NeTEx Part1.
«PK»	***id***	*FareZoneIdType*	1:1	Identifier of FARE ZONE.
«FK»	***ParentFareZoneRef***	*FareZoneRef*	0:1	Parent FARE ZONE of which this is part.
«enum»	***ZoneTopology***	*ZoneTopologyEnum*	0:1	Topology of FARE ZONE with regard to other zones. See allowed values below.
«enum»	***ScopingMethod***	*ScopingMethodEnum*	0:1	Indication of how member stops of a FARE ZONE are specified; See allowed values below. The default value is '*explicitStops'.*

«FK»	***Transport-OranisationRef***	*(TransportOrganisationRef) OperatorRef* | *AuthorityRef*	0:1	Reference to OPERATOR of FARE ZONE.
«cntd»	***GroupOf-Operators***	*GroupOfOperators*	0:*	Reference to GROUP OF OPERATORs (also AUTHORITies) for FARE ZONE.
«cntd»	***fareSections***	<u>*FareSection*</u>	0:*	FARE SECTIONs in FARE ZONE.
«cntd»	***neighbours***	*FareZoneRef*	0:*	Adjacent FARE ZONEs.

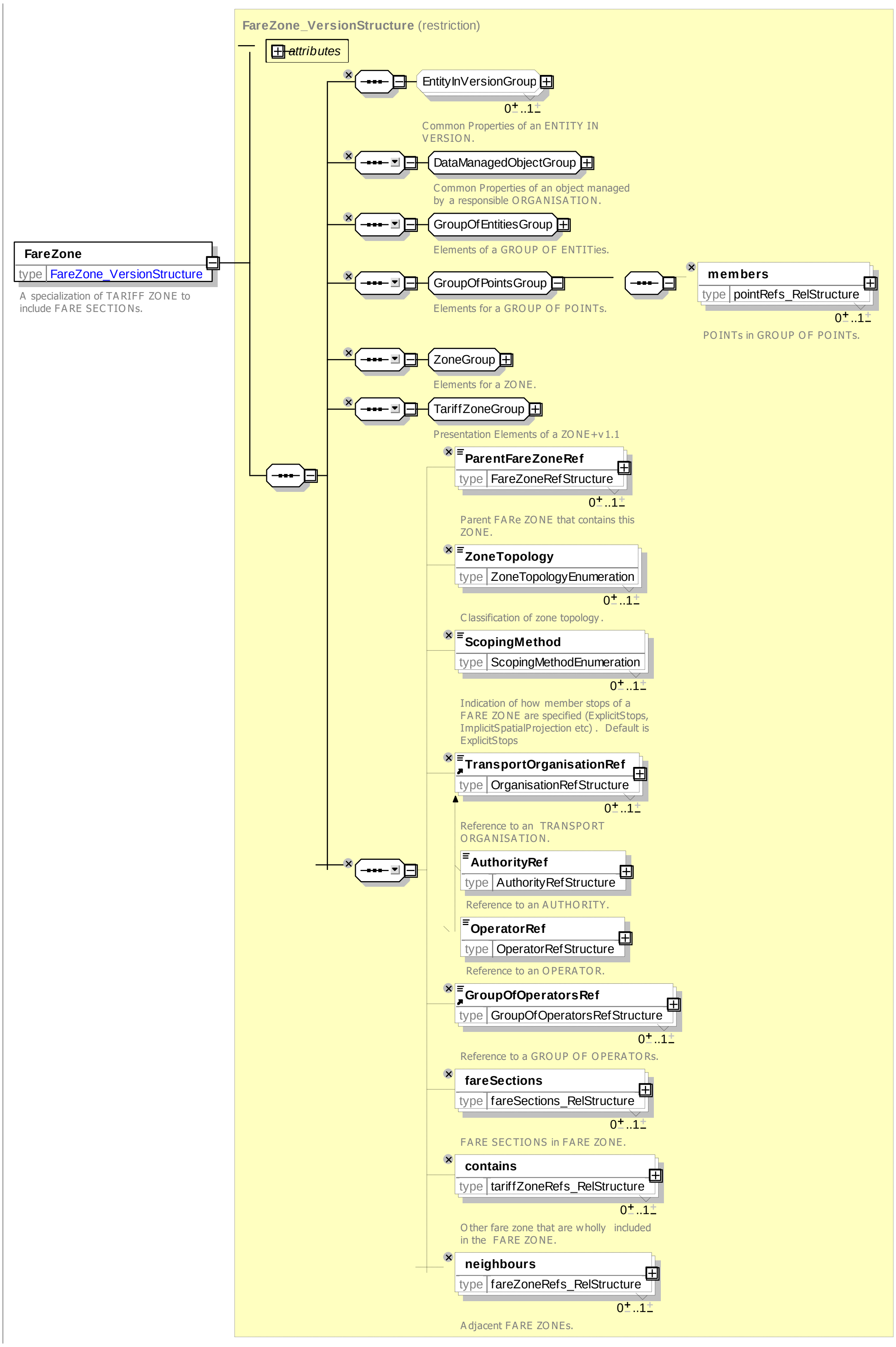


Figure 44 — *FareZone* — XSD

7.4.1.4.3.1 ZoneTopology – Allowed values

The following table shows the allowed values for ***ZoneTopology*** (*ZoneTopologyEnumeration*).

Table 23 – ZoneTopology – Allowed values

Value	Description
overlapping	Zones are of arbitrary shape and may overlap
honeycomb	Zones are arranged as a tiled honeycomb of regular polygons (e.g. Hexagons, squares etc. The zones do not overlap.
ring	Zones are arranged in rings. The nested inner zones are included in any containing outer zones. +v1.1
annular	Zones are arranged in tiled hollow rings. The area of any immediately nested zone is excluded from the containing outer zone. +v1.1
nested	Zones are nested, that is some zones are fully contained within other zones and are automatically included if the outer zone is selected. They may also overlap their neighbours.
tiled	Zones are arranged as adjacent tiles or arbitrary shapes that do not overlap.
sequence	Zones are arranged as adjacent tiles in sequence that touch at either or both ends. They do not overlap. +v1.1
overlappingSequence	Zones are arranged as adjacent tiles in sequence that touch at either or both ends. They may partially overlap such that some stops are in both zones. +v1.1
other	Zone has other or unspecified topology.

7.4.1.4.3.2 ScopingMethod – Allowed values

The following table shows the allowed values for ***ScopingMethod*** (*ScopingMethod Enumeration*).

Table 24 – ScopingMethod – Allowed values +v1.1

Value	Description
explicit	Stops are explicitly stated as members of the zone.
implicitSpatialProjection	Any stop that is spatially contained within the zone is assumed to be a member.
explicitPeripheryStops	The extent of the zone is indicated by a set of stops marking the border points on the periphery of the FARE ZONE. Any stop that is spatially contained within the indicated extent is assumed to be a member.
other	Other method.

7.4.1.4.4 **FareSection.**

A subdivision of a JOURNEY PATTERN consisting of consecutive POINTs IN JOURNEY PATTERN, used to define an element of the fare structure.

Table 25 – *FareSection* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	<u>*CommonSection*</u>	::>	FARE SECTION inherits from COMMON SECTION. See NeTEx Part1.
«PK»	***id***	*FareSectionIdType*	1:1	Identifier of FARE SECTION.
	Name	*MultilingualString*	0:1	Name of FARE SECTION.
«FK»	***JourneyPatternRef***	*JourneyPatternRef+*	0:1	Reference to a JOURNEY PATTERN that FARE SECTION follows.
«FK»	***FromFarePointRef***	*FarePointInPatternRef*	0:1	Reference to FARE POINT IN PATTERN at which FARE SECTION starts.
«FK»	***ToFarePointRef***	*FarePointInPatternRef*	0:1	Reference to FARE POINT IN PATTERN at which FARE SECTION ends.

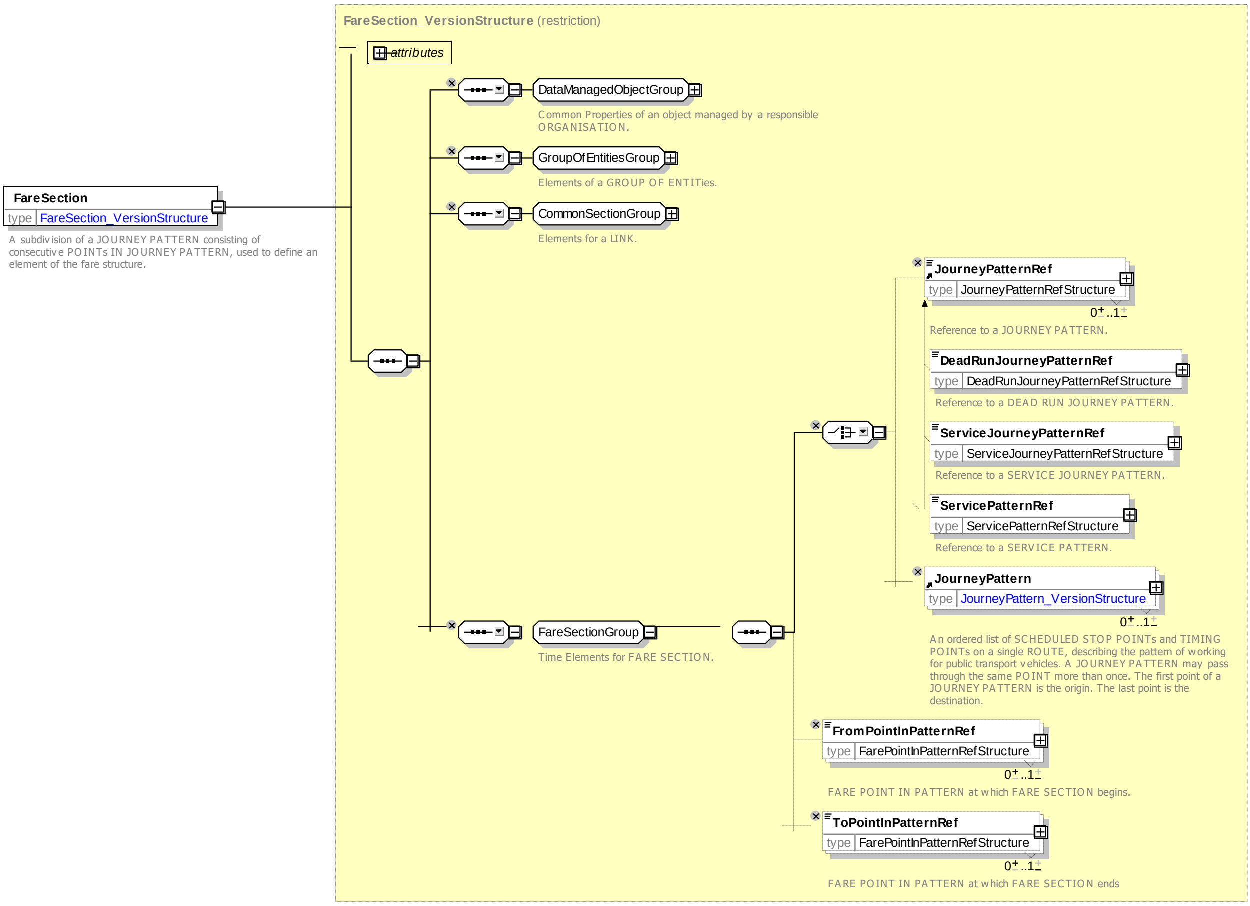


Figure 45 — *FareSection* — XSD

7.4.1.4.5 SeriesConstraint – Model Element

An extension of a DISTANCE MATRIX ELEMENT, a cell of an origin-destination matrix for TARIFF ZONEs or STOP POINTs, expressing a fare distance for the corresponding trip (as a value in km, number of fare units etc.) and possibly a constraint to allow travel only on specific routes.

Table 26 – *SeriesConstraint* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*PriceableObject*	::>	SERIES CONSTRAINT inherits from PRICEABLE OBJECT.
«PK»	***id***	*SeriesConstraintIdType*	1:1	Identifier of SERIES CONSTRAINT.
	order	*xsd:integer*	0:1	Relative priority when there are multiple SERIES CONSTRAINTs between the same points. (Equivalent to route number on TAP TSI Series Number).
	PrivateCode	*PrivateCodeType*	0:1	Private Code associated with element.
	Itinerary	*xsd:normalizedString*	0:1	Stylised text description of SERIES CONSTRAINT. See Tap TSI 5.1. and above.
	SymbolMarking-UsualRoute	*xsd:normalizedString*	0:1	Symbol to use to denote the usual route.
«enum»	***SeriesType***	*SeriesTypeEnum*	0:1	Classification of SERIES CONSTRAINT. The default value is '*stationToStation'*. See allowed values below.
«enum»	***RoutingType***	*RoutingTypeEnum*	0:1	Whether this is a direct i.e. no changes required point to point or indirect. See allowed values below.
«enum»	***FareBasis***	*FareBasisEnum*	0:1	Fare basis used to price Series. See allowed values below.
	Distance1stClass	*DistanceType*	0:1	Notional distance along SERIES CONSTRAINT for computation of First Class fares.
	Distance-2ndClass	*DistanceType*	0:1	Notional distance along SERIES CONSTRAINT for computation of Second Class fares.
	Discrete	*xsd:boolean*	0:1	Whether SERIES CONSTRAINT can only be used by itself, or whether it can be used in a chain of series.
«FK»	***FromConnection Ref***	*ConnectionRef*	0:1	Reference to CONNECTION associated with origin end of SERIES CONSTRAINT.
«FK»	***ToConnectionRef***	*ConnectionRef*	0:1	Reference to CONNECTION associated with destination end of SERIES CONSTRAINT.
«cntd»	***farePointsIn-Pattern***	*FarePointInPattern*	0:*	FARE POINTs IN PATTERN in SERIES CONSTRAINT.
«cntd»	***journeyPatterns***	*JourneyPatternRef+*	0:*	References to JOURNEY PATTERN or patterns. equivalent to the SERIES CONSTRAINT.
«cntd»	***prices***	*SeriesConstraintPrice*	0:*	SERIES CONSTRAINT PRICEs for the SERIES CONSTRAINT.
«cntd»	***replaces***	*SeriesConstraintRef*	0:*	Replaces the specified SERIES. (Needed for TAP TSI).

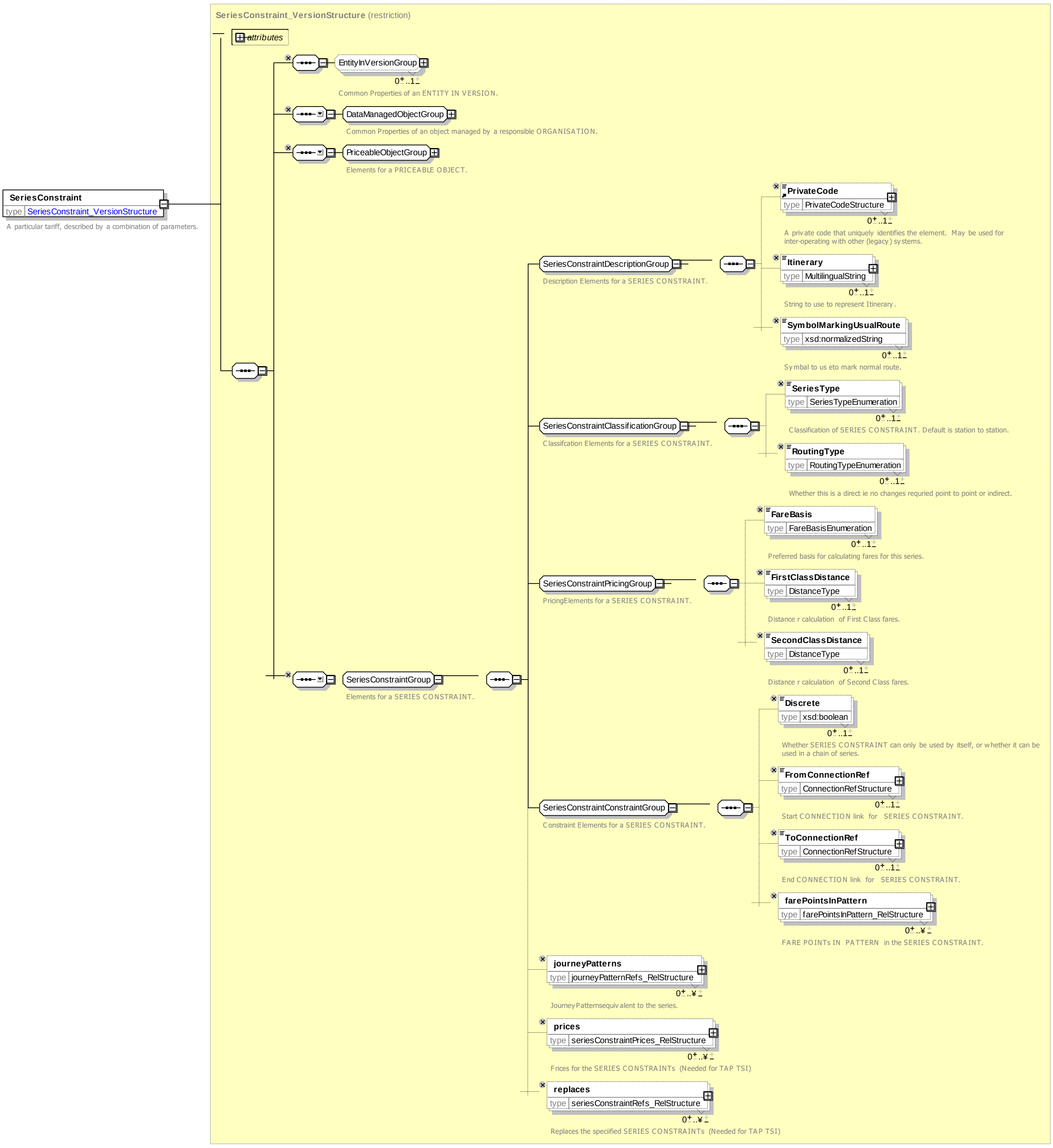


Figure 46 — *SeriesConstraint* — XSD

7.4.1.4.5.1 SeriesType – Allowed values

The following table shows the allowed values for ***SeriesType*** (*SeriesTypeEnumeration*).

Table 27 – *SeriesType* – Allowed values

Value	Description
stationToStation	Series runs from a station to a station.

originToBorder	Series runs from origin to border point.
borderToDestination	Series runs from border point to destination.
transit	Series is a transit series running between two border points.

7.4.1.4.5.2 RoutingType – Allowed values

The following table shows the allowed values for ***RoutingType*** (*RoutingTypeEnumeration*).

Table 28 – *RoutingType* – Allowed values

Value	Description
direct	No changes needed.
indirect	Changes needed.
both	Both direct and indirect routings.

7.4.1.4.5.3 FareBasis – Allowed values

The following table shows the allowed values for ***FareBasis*** (*FareBasisEnumeration*).

Table 29 – *FareBasis* – Allowed values

Value	Description
route	Distance based fare
distance	Route based fare.

7.4.1.4.6 FarePointInPattern – Model Element

A POINT IN PATTERN which represents the start or end of a FARE SECTION.

Table 30 – *FarePointInPattern* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PointInJourneyPattern*	::>	FARE POINT IN PATTERN inherits from POINT IN JOURNEY PATTERN. See NeTEx Part1.
«PK»	***id***	*FaresPointInPattern-IdType*	1:1	Identifier of a FARE POINT IN PATTERN.
	ScheduledStop-PointView	*ScheduledStopPointView*	0:1	Derived information about the SCHEDULED STOP POINT, such as its name – see NeTEx Part1.
	Abridgment-Ranking	*xsd:integer*	0:1	Relative ranking for omitting this FARE POINT IN PATTERN when presenting an abridged version of the series as an itinerary. 1=High, i.e. Omit first.
«enum»	***Presentation-Position***	*SeriesPresentationEnum*	0:1	Relative position for showing this FARE POINT IN PATTERN in an itinerary when there is a choice according to rail conventions. For example, *(A / B) * C* versus (*B/A) * C.* See allowed values below.
	IsForbidden	*xsd:boolean*	0:1	Whether use of fare point is forbidden- can be used to explicitly exclude certain routings. The default value is '*false*'.

	Interchange-Allowed	*xsd:boolean*	0:1	Whether interchange to another service is allowed at this STOP POINT.
	IsFareStage	*xsd:boolean*	0:1	Whether stop is considered to be a fare stage. +v1.1

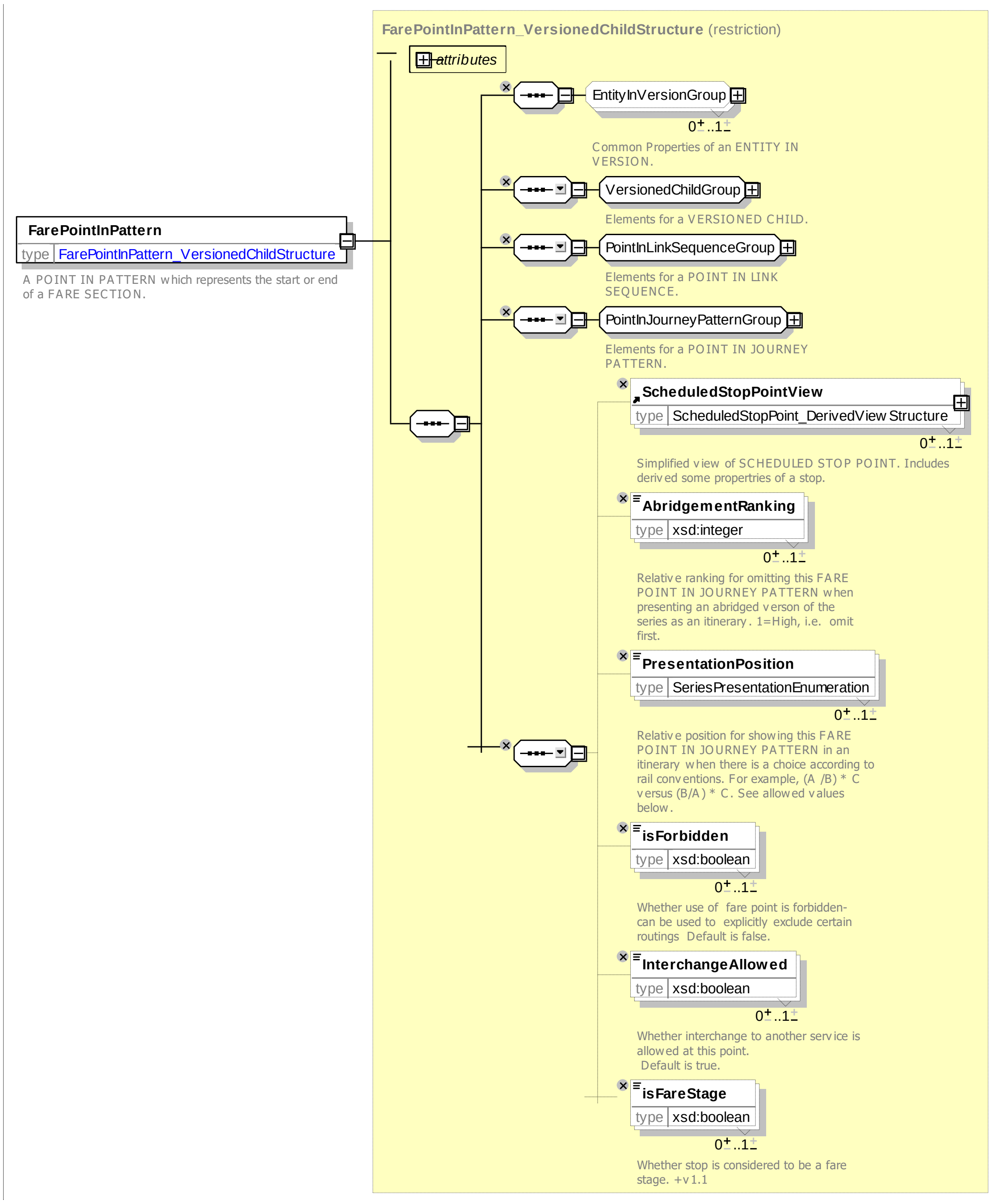


Figure 47 — *FarePointInPattern* — XSD

7.4.1.4.6.1 PresentationPosition – Allowed values

The following table shows the allowed values for ***PresentationPosition*** (*SeriesPresentationEnumeration*).

Table 31 – *SeriesPresentation* – Allowed values

Value	Description
noStation	In an itinerary, omit station.
requiredStation	In an itinerary, show as a required station, that is one that passenger must route by.
optionalLeft	In an itinerary, show as an optional station - first choice on left.
optionalRight	In an itinerary, show as an optional station - second choice on right.

7.4.1.4.7 SeriesConstraintPrice – Model Element

A set of all possible price features of a SERIES CONSTRAINT: default total price etc.

Table 32 – *SeriesConstraintPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	SERIES CONSTRAINT PRICE inherits from FARE PRICE.
«PK»	***id***	*SeriesConstraint-PriceIdType*	1:1	Identifier of SERIES CONSTRAINT PRICE.
		CHOICE	0:1	
«FK»	***SeriesConstraint Ref***	*SeriesConstraintRef*	0:1	SERIES CONSTRAINT for which this is the price.
«FK»	***GroupOfSeries-ConstraintsRef***	*GroupOf-SeriesConstraintsRef*	0:1	GROUP OF SERIES CONSTRAINTs for which this is the price.

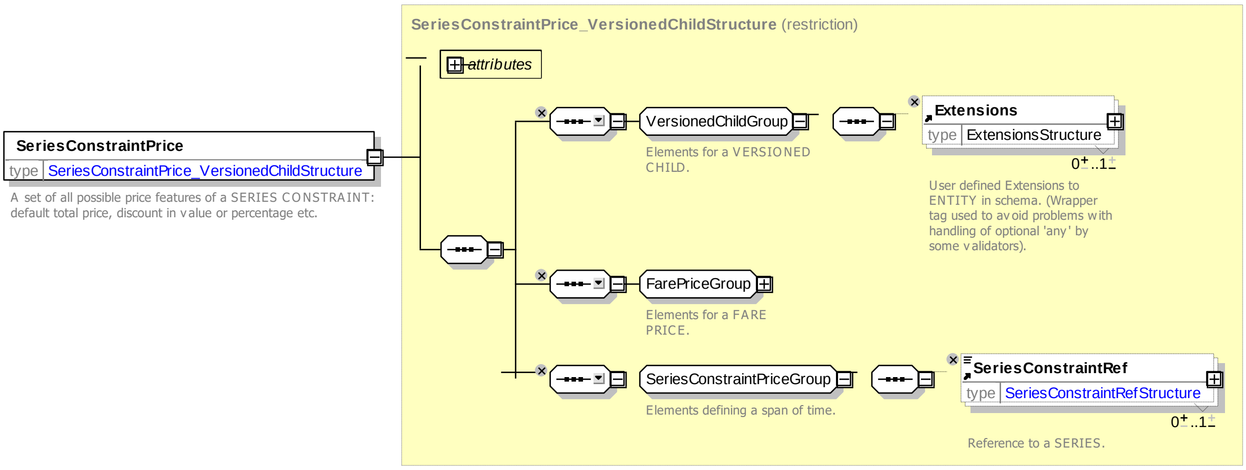


Figure 48 — *SeriesConstraintPrice* — XSD

7.4.1.5 Fare Zone – XML examples

7.4.1.5.1 Fare Zone: XML Example of Fare zones with Fare Sections

The following code fragment shows a FARE ZONE with some FARE SECTIONs defined for it

For EXAMPLE:

```
<fareZones>
    <FareZone id="nr:NationalRailOysterArea" version="any">
        <Name> Nation Rail Oyster Area</Name>
        <ParentFareZoneRef version="any" ref="nr:NetworkRailCardArea"/>
        <!-- <fareSections> -->
        <fareSections>
            <FareSectionRef
                ref="nr:Chingford_to_Liverpool_Street@Chingford@Walthamstow_Central">
            </FareSectionRef>
            <FareSectionRef
                ref="nr:Chingford_to_Liverpool_Street@Walthamstow_Central@Liverpool_Street">
            </FareSectionRef>
        </fareSections>
    <!--- ETC., ETC. -->
    </FareZone>
<fareZones>

<fareSections>
    <FareSection version="any"
        id="nr:Chingford_to_Liverpool_Street@Chingford@Walthamstow_Central">
        <Name>Chingford to Walthamstow Central</Name>
        <ServiceJourneyPatternRef version="any"  ref="nr:Chingford_to_Liverpool_Street"/>
        <FromPointInPatternRef version="any"
            ref="nr:PointInJourneyPattern:Chingford_to_Liverpool_Street@Chingford"/>
        <ToPointInPatternRef version="any"
            ref="nr:PointInJourneyPattern:Chingford_to_Liverpool_Street@Walthamstow_Central"/>
    </FareSection>
    <FareSection version="any"
        id="nr:Chingford_to_Liverpool_Street@Walthamstow_Central@Liverpool_Street">
        <Name> Walthamstow Central to Liverpool street</Name>
        <ServiceJourneyPatternRef version="any" ref="nr:Chingford_to_Liverpool_Street"/>
        <FromPointInPatternRef version="any"
            ref="nr:Chingford_to_Liverpool_Street@Walthamstow_Central"/>
        <ToPointInPatternRef version="any"
            ref="nr:Chingford_to_Liverpool_Street@Liverpool_Street"/>
    </FareSection>
```

7.4.1.5.2 Fare Zone: XML Example of Border Points

The following code fragment shows two BORDER POINT definitions.

For EXAMPLE:

```
<borderPoints>
    <BorderPoint id="tap:19" version="01" dataSourceRef="tap:uic">
        <Name>Kastrup(Gr)</Name>
        <ShortName>Kastrup(Gr)</ShortName>
        <Description>DSB; SJ</Description>
        <GroupOfOperators id="tap:19" version="01">
            <members>
                <OperatorRef ref="tap:DSB"/>
                <OperatorRef ref="tap:SJ"/>
        </members>
        </GroupOfOperators>
    </BorderPoint>
    <BorderPoint id="tap:20" version="01" dataSourceRef="tap:uic">
        <Name>Frederikshavn</Name>
        <ShortName>Frederikshavn</ShortName>
        <Description>DSB; SJ</Description>
        <GroupOfOperators id="tap:20" version="01">
            <members>
                <OperatorRef ref="tap:DSB"/>
```

```
            <OperatorRef ref="tap:SJ"/>
        </members>
    </GroupOfOperators>
</BorderPoint>
```

The following code fragment shows a SCHEDULED STOP POINT associated with the BORDER POINT '*19'* (Kastrup(Gr)).

For EXAMPLE:

```
<FareScheduledStopPoint id="tap:12340" version="01"
        dataSourceRef="tap:db">
    <Name lang="de">Kastrup</Name>
    <tariffZones>
        <TariffZoneRef version="01" ref="tap:12340"/>
    </tariffZones>
    <ShortName lang="de">Kastrup</ShortName>
    <StopType>railStation</StopType>
    <CountryRef ref="de"/>
    <BorderPointRef version="01" ref="tap:19"/>
</FareScheduledStopPoint>
```

7.4.1.5.3 Fare Zone: XML Example of SeriesConstraint with Fare Sections

The following code fragment shows a SERIES CONSTRAINT for the route *Nurnberg* **** (Erfurt/Hof) * Halle * Schwerin **** *Berlin* with three FARE POINTs IN PATTERN.

For EXAMPLE:

```
<DistanceMatrixElement id="tap:series555" version="01">
    <StartStopPointView>
        <FareScheduledStopPointRef ref="tap:999123"/>
        <Name lang="de">Nurnberg</Name>
        <ShortName lang="de">Nurnberg</ShortName>
    </StartStopPointView>
    <EndStopPointView>
        <FareScheduledStopPointRef ref="tap:999234"/>
        <Name lang="de">Berlin</Name>
        <ShortName lang="de">Berlin</ShortName>
    </EndStopPointView>
    <seriesConstraints>
        <SeriesConstraint id="tap:series555" version="01">
            <Itinerary>(Erfurt/Hof)*Halle*Schwerin</Itinerary>
            <SeriesType>stationToStation</SeriesType>
            <UseStandardFareCalculation>route</UseStandardFareCalculation>
            <farePointsInPattern>
                <FarePointInPattern id="tap:55501" version="01" order="1">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap:16043"/>
                        <Name>Erfurt</Name>
                    </ScheduledStopPointView>
                    <AbridgementRanking>2</AbridgementRanking>
                    <PresentationPosition>optionalLeft</PresentationPosition>
                </FarePointInPattern>
                <FarePointInPattern id="tap:55502" version="01" order="2">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap:26002"/>
                        <Name>Hof</Name>
                    </ScheduledStopPointView>
                    <AbridgementRanking>2</AbridgementRanking>
                    <PresentationPosition>optionalRight</PresentationPosition>
                </FarePointInPattern>
                <FarePointInPattern id="tap:55503" version="01" order="3">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap:23002"/>
                        <Name>Halle</Name>
                    </ScheduledStopPointView>
                    <PresentationPosition>required</PresentationPosition>
                </FarePointInPattern>
                <FarePointInPattern id="tap:55504" version="01" order="4">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap:27359"/>
```

```
                    <Name>Schwerin</Name>
                </ScheduledStopPointView>
                <AbridgementRanking>2</AbridgementRanking>
                <PresentationPosition>required</PresentationPosition>
            </FarePointInPattern>
        </farePointsInPattern>
    </SeriesConstraint>
</seriesConstraints>
</DistanceMatrixElement>
```

7.4.2 Fare Facility

7.4.2.1 Fare Facility – Conceptual MODEL

A FACILITY is a named amenity available to the public at a SITE or on a SERVICE. A facility has no further properties other than a name. An EQUIPMENT or LOCAL SERVICE is used to describe the further properties provided as part of particular facility.

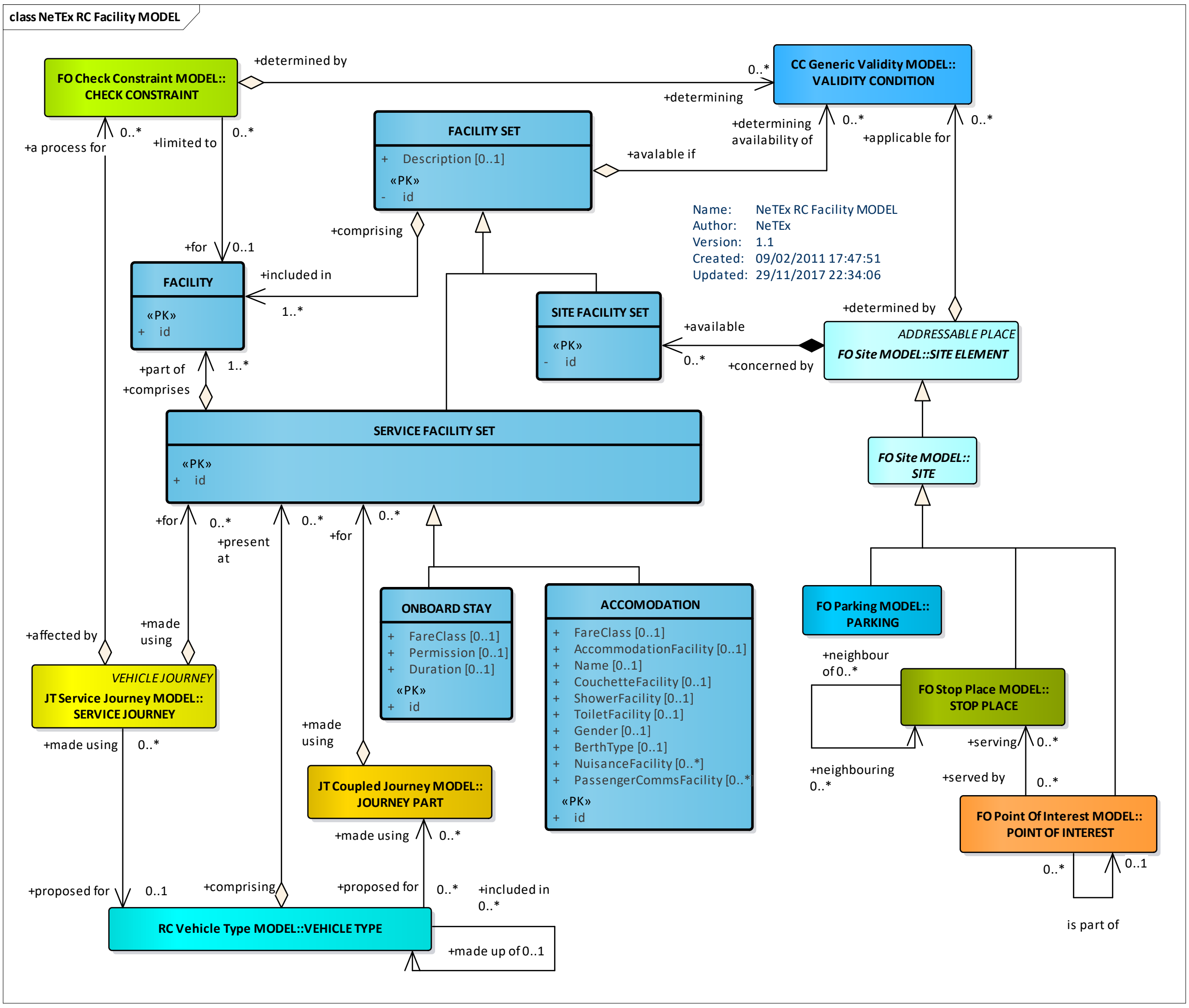


Figure 49 — Fare Facility– Conceptual MODEL (UML)

7.4.2.2 Fare Facility – Physical model

In NeTEx Part1 and Part2 various FACILITY elements are used to specify the allowed values of named properties of elements. The Fare Facility model describes additional NeTEx FACILITY values for NeTEx Part3. Facility elements are grouped in FACILITY SETs that can be associated with stops and other SITEs and also with VEHICLE JOURNEYs and other entities. The same mechanism can be used to associate facilities such as Couchette accommodate classes with specific fares using ACCESS RIGHT PARAMETER ASSIGNMENTs (See later).

The following diagram shows the Part3 Facility Model elements. For convenience of reference it also includes elements from the NeTEx Part1 model, in particular The ACCOMMODATION elements that can be associated with access rights.

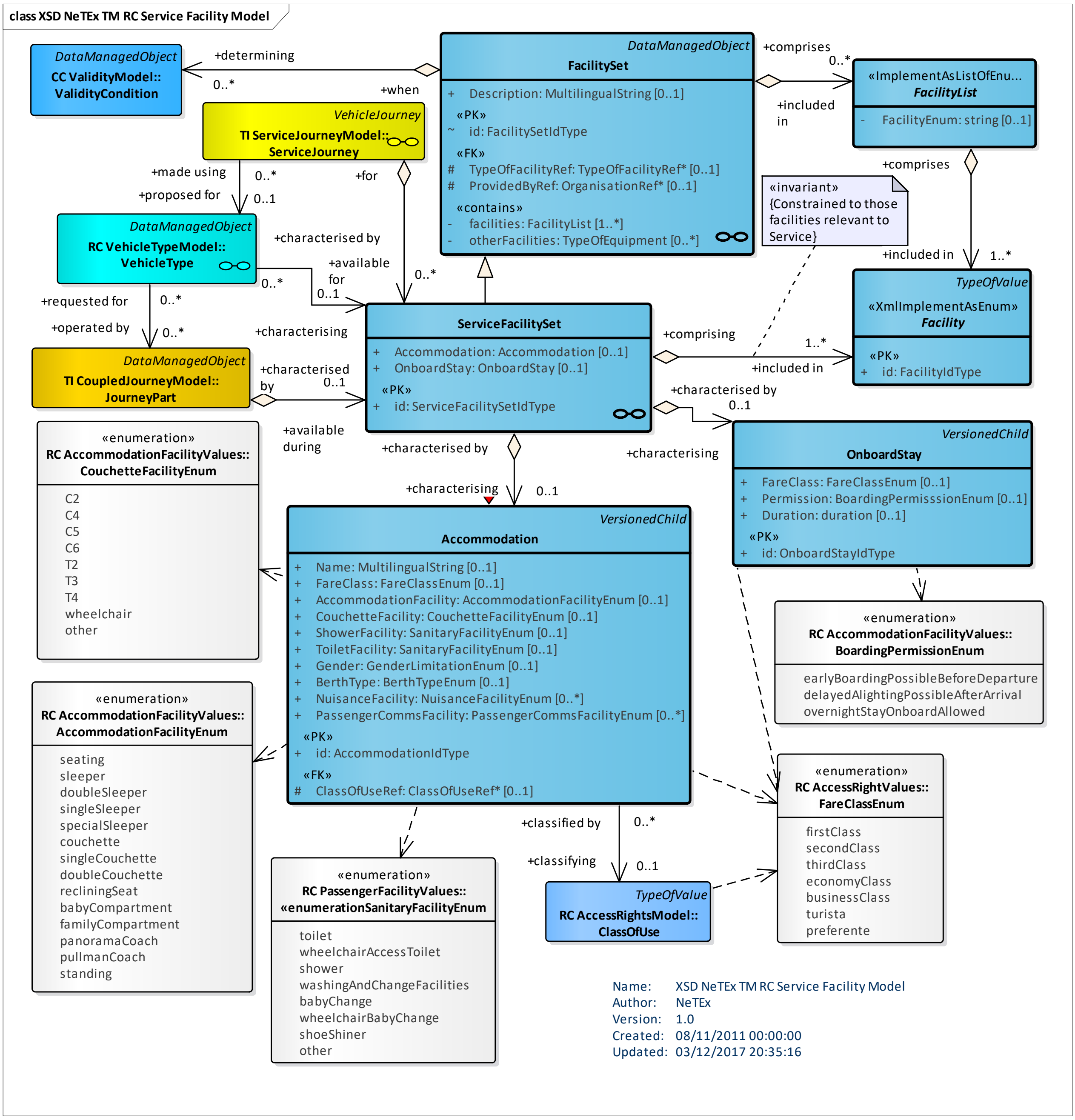


Figure 50 — Fare Facility– Physical Model (UML)

7.4.2.3 Fare Facility – Attributes and XSD

7.4.2.3.1.1 DistanceValidity – Allowed values

The following table shows the allowed values for ***DistanceValidity*** (*DistanceValidityEnumeration*).

Table 33 – *DistanceValidity* – Allowed values

Value	Description
individualPassengers	Distance valid for individual passengers.
baggage	Distance valid for Baggage.
groups	Distance valid for groups.

7.4.2.3.1.2 FareJourneySection – Allowed values

The following table shows the allowed values for ***FareJourneySection*** (*FareJourneySectionEnumeration*).

Table 34 – *FareJourneySection* – Allowed values

Value	Description
specialConditionsApply	Special conditions made available by Railway Undertaking.
passengerRightRegulationNotApplicable	Journey section for which the Passenger Rights Regulation is not applicable.
railOrFerry	Journey section by Rail or by Ferry.
railOrBus	Journey section by Rail or by Bus.
ferryOrBus	Journey section by Ferry or by Bus.

7.4.2.3.1.3 TripCoupon – Allowed values

The following table shows the allowed values for ***TripCoupon*** (*TripCouponEnumeration*).

Table 35 – *TripCoupon* – Allowed values

Value	Description
couponOnlyValidDuringOperatingHours	Coupon only valid during Operating hours.
couponAvailable	Series for which Pre-printed Coupons Exist.
oneWayCoupon	Coupon which cannot be used in the opposite direction.

7.4.3 Vehicle Seating

7.4.3.1 Vehicle Seating – Physical model

The vehicle seating model defines passenger seat elements that are referenced by TRAVEL DOCUMENTs to indicate which seat has been allocated. The model has not yet been defined in Transmodel but a skeleton implementation is provided in NeTEx so that appropriate references can be included on TRAVEL DOCUMENTs.

The following diagram shows the skeleton Vehicle Seating Model

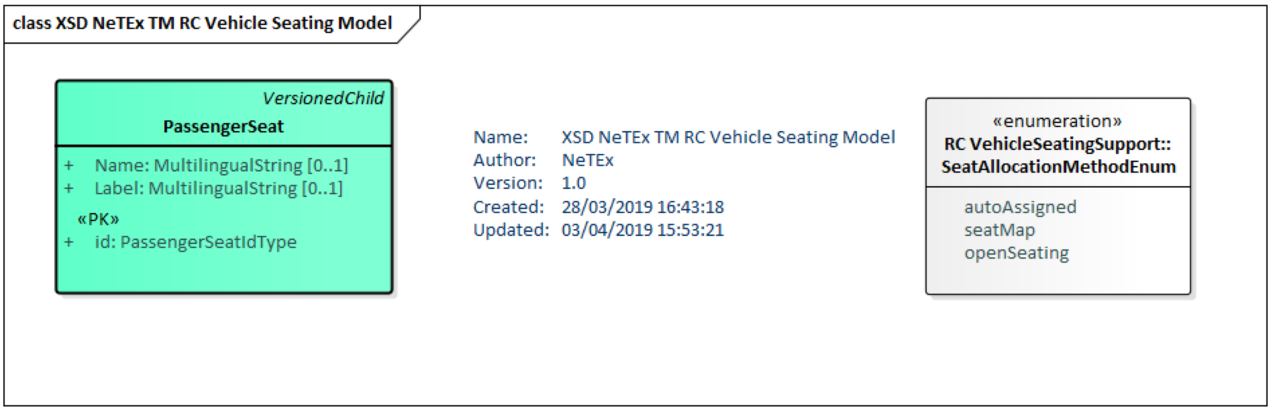


Figure 51 — Vehicle Seating – Physical Model (UML)

7.4.3.2 Fare Facility – Attributes and XSD

7.4.3.2.1.1 SeatAllocationMethod – Allowed values

The following table shows the allowed values for ***SeatAllocationMethod*** (*SeatAllocationMethodEnumeration)*.

Table 36 – *SeatAllocationMethod* – Allowed values

Value	Description
autoAssigned	Seats assigned automatically
seatMap	Seats can be selected from a seat map.
openSeating	No seat reservation

7.5 Fare Structure

A NeTEx fare structure is based on generic quantitative rules that describe the access rights regulating the consumption of transport services in terms of their spatial (GEOGRAPHICAL), temporal (TIME) and other (QUALITY) aspects. The FARE STRUCTURE model describes the structure and parameters attached to these rules.

7.5.1 Fare Structure – Model dependencies

The FARE STRUCTURE MODEL is made up of a number of submodels. These are described in turn later below.

— The COMMON STRUCTURE MODEL provides shared reusable framework elements for the fare structure.

— The GEOGRAPHICAL FARE STRUCTURE MODEL defines additional spatial aspects of the fare structure.

— The TIME FARE STRUCTURE MODEL defines additional temporal aspects of the fare structure.

— The QUALITY FARE STRUCTURE MODEL defines other qualitative aspects of the fare structure.

— The FARE STRUCTURE ELEMENT MODEL defines the core elements of the fare structure.

— The DISTANCE MATRIX ELEMENT MODEL shows the possible origin/destination elements for the case of an origin/destination fare structure.

— The VALIDABLE ELEMENT MODEL provides elements to group FARE STRUCTURE ELEMENTs for use in FARE PRODUCTs.

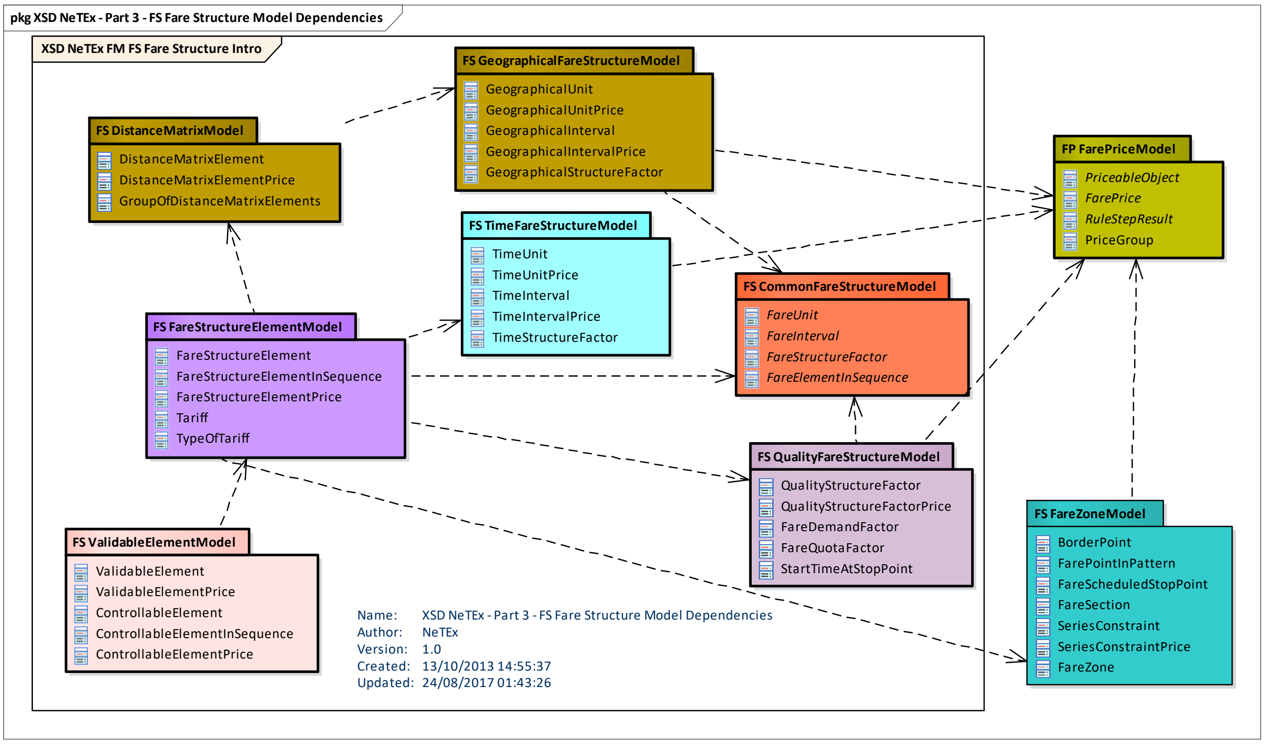


Figure 52 — Fare Structure Model Dependencies (UML)

7.5.2 Common Fare Structure

The COMMON FARE STRUCTURE model defines reusable abstract elements that provide certain common properties of the FARE STRUCTURE such as FARE STRUCTURE FACTOR, FARE INTERVAL and FARE UNIT. These are refined in specific submodels, for example:

— A TIME STRUCTURE FACTOR describes a temporal rule as a TIME INTERVAL of a specified GEOGRAPHICAL UNIT.

— A GEOGRAPHICAL STRUCTURE FACTOR describes a spatial rule as a GEOGRAPHICAL INTERVAL of a specified GEOGRAPHICAL UNIT.

— A QUALITY STRUCTURE FACTOR describes other types of rules.

7.5.2.1 Common Fare Structure: Fare Structure Factors – Conceptual MODEL

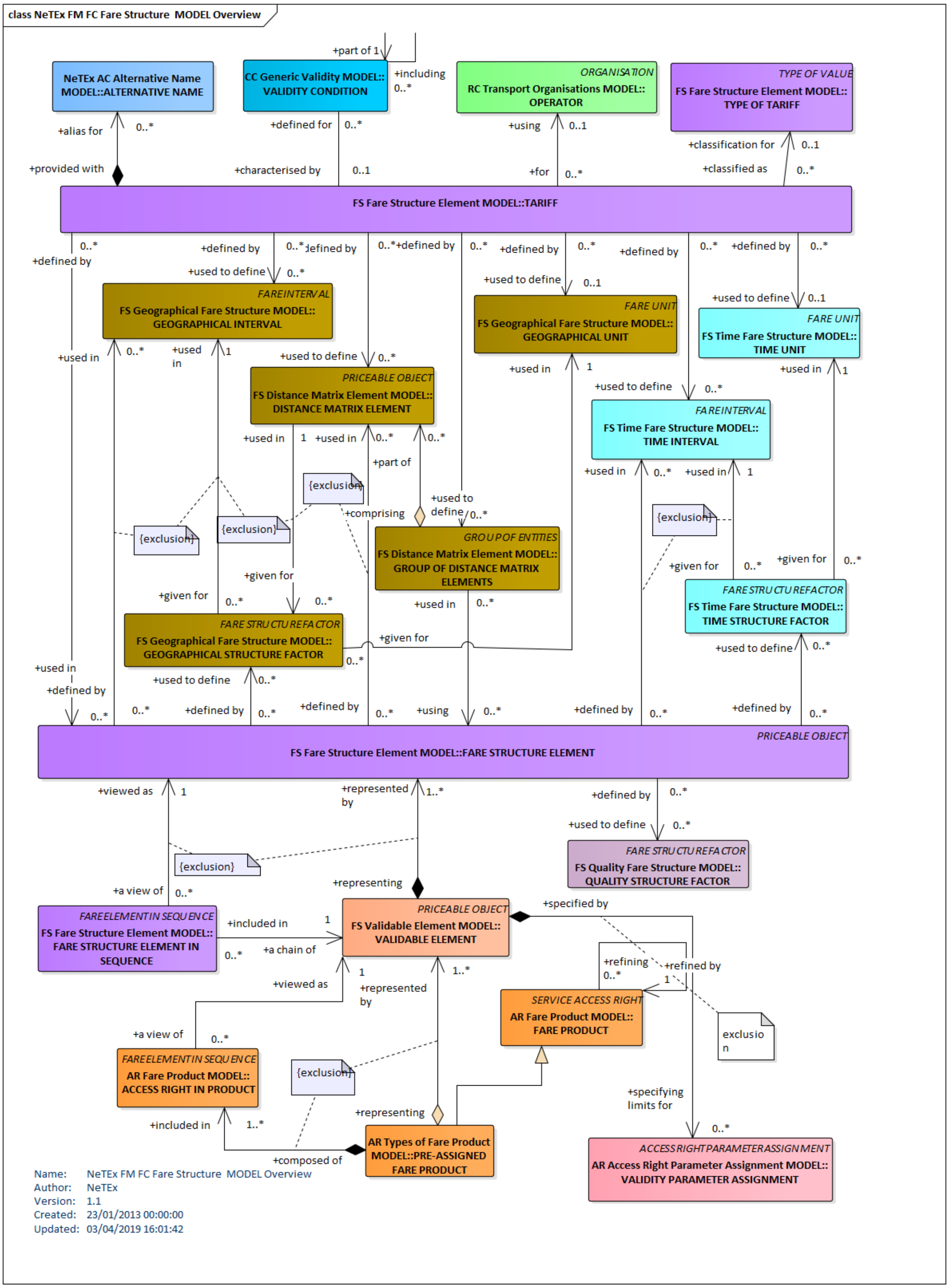


Figure 53 — Fare Structure Parameters – Conceptual MODEL (UML)

7.5.2.2 Common Fare Structure: Fare Structure Elements – Conceptual MODEL

Some fare structures involve the consumption of a sequence of elements in a specified order. The COMMON FARE STRUCTURE model also defines an abstract FARE ELEMENT IN SEQUENCE elements that is refined in other submodels to describe sequential aspects of the FARE STRUCTURE.

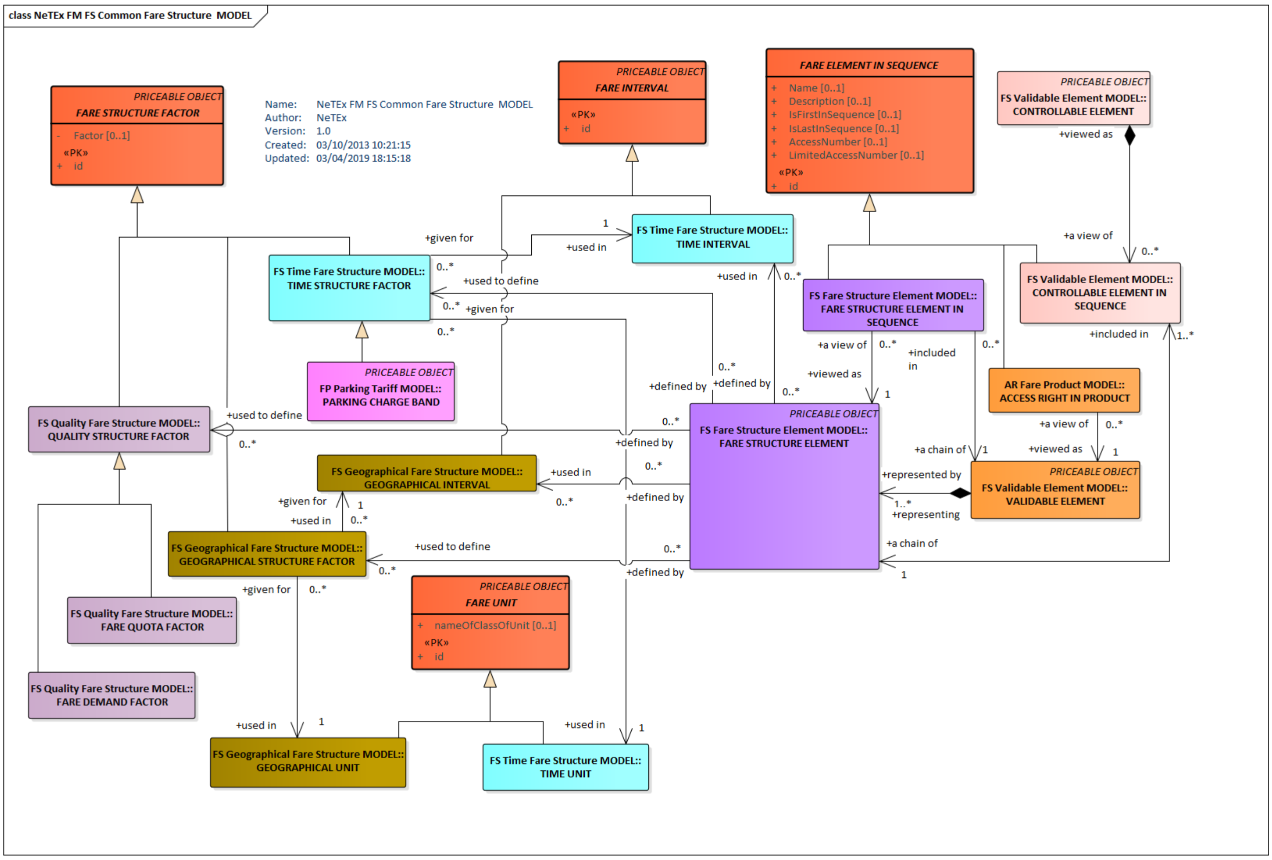


Figure 54 — Fare Elements in Sequence – Conceptual MODEL (UML)

7.5.2.3 Common Fare Structure – Physical model

The following figure shows the physical model for the common FARE STRUCTURE ELEMENTs.

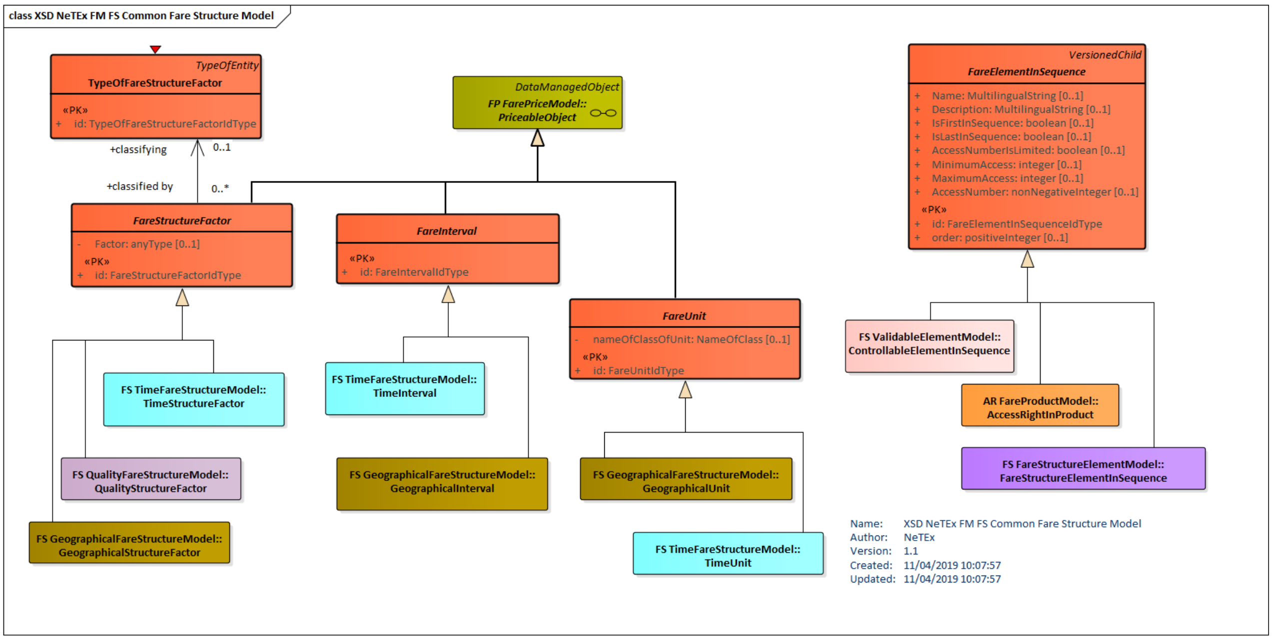


Figure 55 — Common Fare Structure Model – Physical Model (UML)

7.5.2.4 Common Fare Structure – Attributes and XSD

7.5.2.4.1 FareUnit

A unit associated with a FARE STRUCTURE FACTOR.

Table 37 – *FareUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	FARE UNIT inherits from PRICEABLE OBJECT.
«PK»	***id***	*FareUnitIdType*	1:1	Identifier of FARE UNIT.
	nameOfClass-OfUnit	*NameOfClass*	0:1	Type of Class used for zone; ***DistanceType***, etc e.g. ***TariffZone***. This is metadata to facilitate programming.

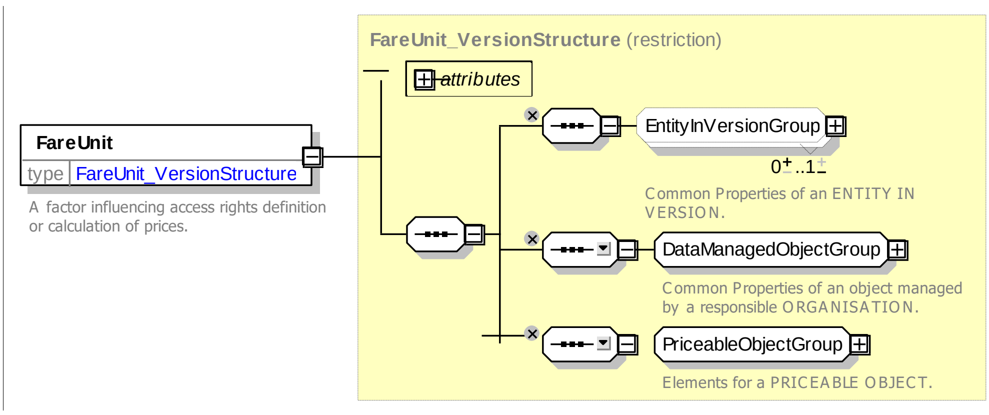


Figure 56 — *FareUnit* — XSD

7.5.2.4.2 FareInterval – Model Element

An interval-based aspect of the fare structure.

Table 38 – *FareInterval* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	FARE INTERVAL inherits from PRICEABLE OBJECT.
«PK»	***id***	*FareIntervalIdType*	1:1	Identifier of FARE INTERVAL.

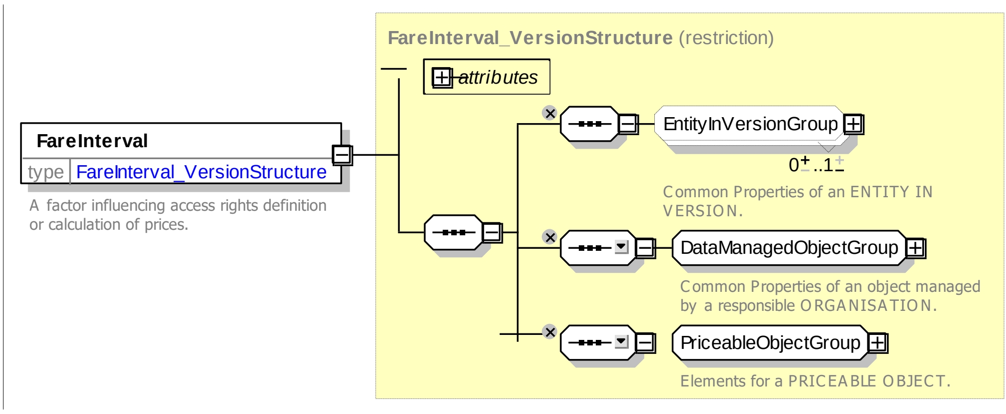


Figure 57 — *FareInterval* — XSD

7.5.2.4.3 FareStructureFactor – Model Element

A factor influencing access rights definition or calculation of prices (abstract framework element).

Table 39 – *FareStructureFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	FARE STRUCTURE FACTOR. inherits from PRICEABLE OBJECT.
«PK»	***id***	*FareStructureFactorIdType*	1:1	Identifier of FARE STRUCTURE FACTOR.
	PrivateCode	*PrivateCodeStructure*	0:1	External code associated with factor. +v1.1
	Factor	*xsd:anyType*	0:1	Arbitrary values associated with factor.
«PK»	***TypeOfFare-Structure-FactorRef***	*TypeOfFareStructure-FactorRef*	1:1	Reference to a classification of the FARE STRUCTURE FACTOR.

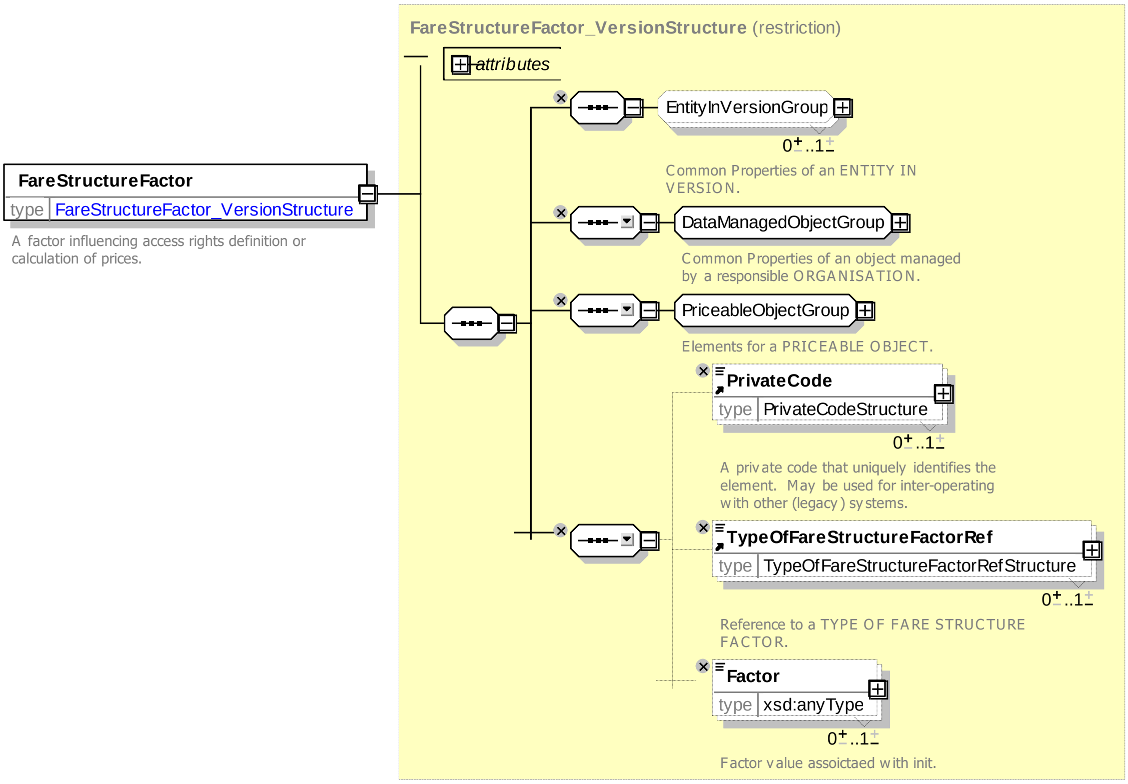


Figure 58 — *FareStructureFactor* — XSD

7.5.2.4.4 FareElementInSequence – Model Element

A FARE ELEMENT as a part of an ELEMENT, including its possible order in the sequence of FARE ELEMENTs (abstract framework element).

Table 40 – *FareElementInSequence* – – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	FARE ELEMENT IN SEQUENCE inherits from VERSIONED CHILD. See NeTEx Part1.
«PK»	***id***	*FareElementInSequence-IdType*	1:1	Identifier of FARE ELEMENT IN SEQUENCE.
«PK»	***order***	*xsd:positiveInteger*	0:1	Order of element within SEQUENCE.
	Name	*MultilingualString*	0:1	Name of FARE ELEMENT IN SEQUENCE.
	Description	*MultilingualString*	0:1	Description of FARE ELEMENT IN SEQUENCE.
	IsFirstInSequence	*xsd:boolean*	0:1	Whether element is the first in the sequence.
	IsLastInSequence	*xsd:boolean*	0:1	Whether element is the last in the sequence.
	Access-NumberIsLimited	*xsd:boolean*	0:1	Whether access number is limited. +v1.1
	MinimumAccess	*xsd:nonNegativeInteger*	0:1	Minimum number of accesses. +v1.1

	MaximumAccess	*xsd:nonNegativeInteger*	0:1	Maximum number of accesses. +v1.1
	AccessNumber	*xsd:nonNegativeInteger*	0:1	Access number in sequence.

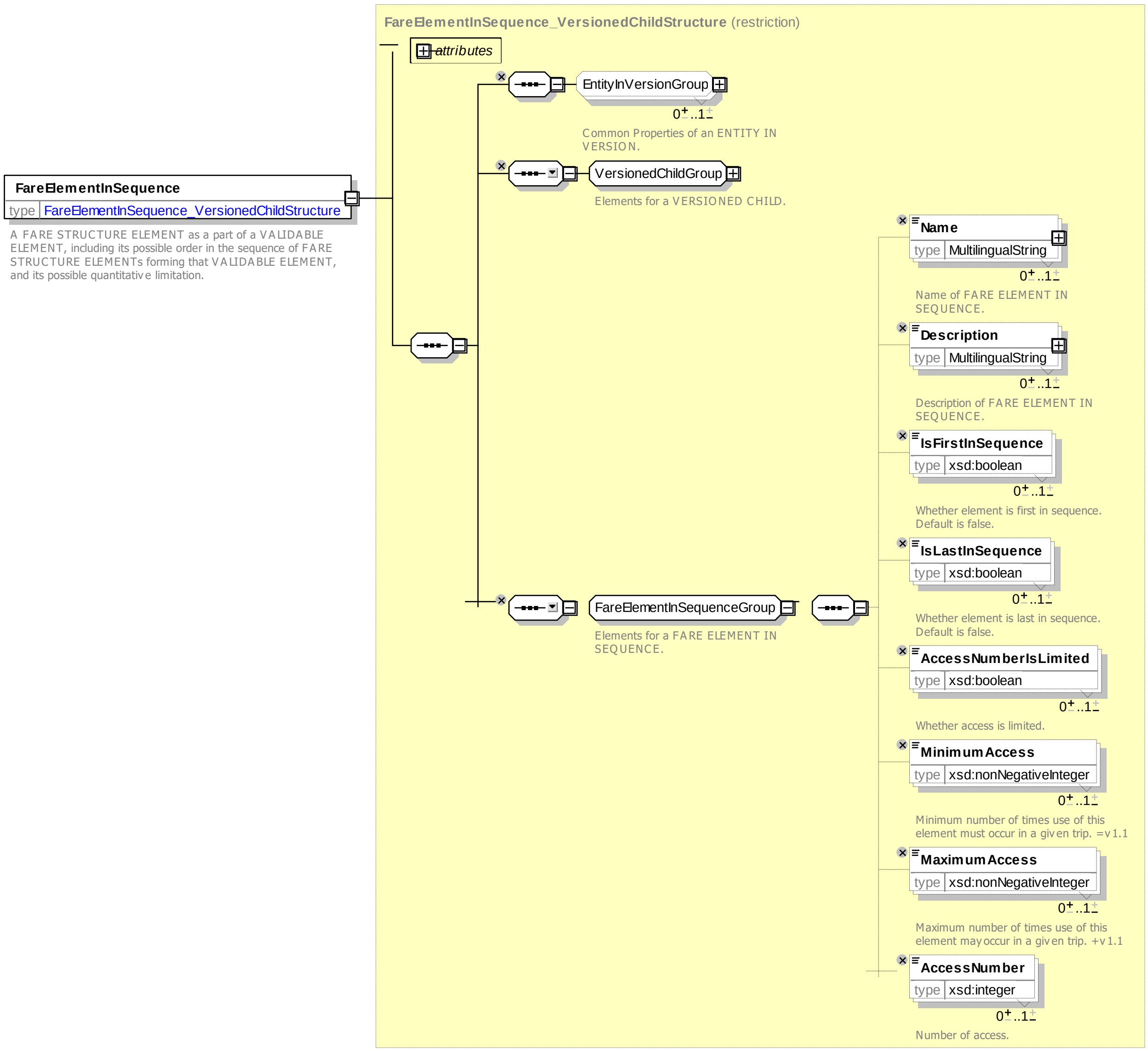


Figure 59 — *FareElementInSequence* — XSD

7.5.2.4.1 TypeOfFareStructureFactor– Model Element

A classification of a FARE STRUCTURE FACTOR.

Table 41 – 7.5.2.4.1 TypeOfFareStructureFactor– – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF FARE STRUCTURE FACTOR. Inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*TypeOfFareStructure-FactorIdType*	1:1	Identifier of TYPE OF FARE STRUCTURE FACTOR.

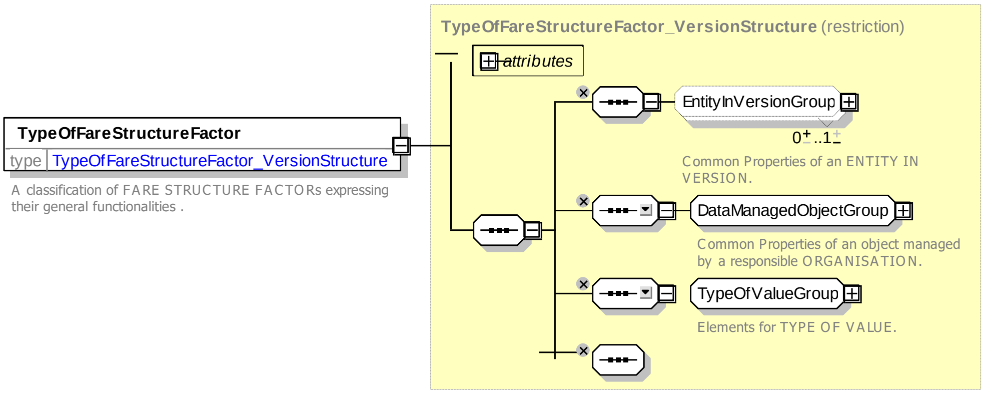


Figure 60 — *TypeOfFareStructureFactor* — XSD

7.5.3 Geographical Fare Structure

7.5.3.1 Geographical Fare Structure – Conceptual MODEL

The GEOGRAPHICAL FARE STRUCTURE model describes certain spatial aspects of the fare structure model and is made up of GEOGRAPHICAL STRUCTURE FACTORs.

7.5.3.1.1 Simple Space-based Factors

The most common fare structure rules are space-based, or more precisely, distance-based. The three main types are respectively progressive (based on intervals), graduated depending on a distance, and using zones. Some of these types may be combined together.

The entity GEOGRAPHICAL INTERVAL describes a classification of the FARE STRUCTURE ELEMENTs depending on their length, for instance:

— 1 zone (or fare section) crossed, 2 to 4 zones crossed, more than 4 zones crossed;

— ride length less than 5 km, between 5 and 15 km, more than 15 km;

— etc.

Each GEOGRAPHICAL INTERVAL will store the minimum and the maximum value describing the corresponding distance interval, on which a certain fare will be applied.

Graduated fare structures allow a calculation of fares depending on the distance covered during the trip. The distance is computed using a certain unit, the most classical being the distance in kilometres, the number of fare sections (or zones) or the number of stop points. Such a graduation unit is described by the entity GEOGRAPHICAL UNIT. The fare of a trip will be calculated by multiplying its length by a price parameter attached to the GEOGRAPHICAL UNIT.

Many networks will use TARIFF ZONEs. A TARIFF ZONE is a view of a ZONE, specifically defined for fare calculation. It is composed of SCHEDULED STOP POINTs. A TARIFF ZONE may have specific points on its borders, the TARIFF POINTs. Some such points activate an automatic detection of the boundaries: they are ACTIVATION POINTs.

A FARE SECTION is another type of fare structure parameter. It is a subdivision of a JOURNEY PATTERN, consisting of consecutive SCHEDULED STOP POINTs in that JOURNEY PATTERN.

Many graduated fare structures will use the number of TARIFF ZONEs or FARE SECTIONs as GEOGRAPHICAL UNIT. A projection of such TARIFF ZONEs or FARE SECTIONs on the used JOURNEY PATTERN will allow to derive the number of zones or sections crossed during a trip.

In many cases, the values used for applying the fare structure rules will be derived from the description of the actual element consumed. For instance, the length of a trip in km will be derived from the JOURNEY PATTERN description (using the length of the LINKs composing the JOURNEY PATTERN).

Some fare structure systems will use arbitrary fare distances between the origin and the destination of a FARE STRUCTURE ELEMENT. This is typically the case when a zone-matrix fare system is used. Some TARIFF ZONEs (usually a few) are defined and a specific fare distance parameter is defined for each possible origin/destination pair of TARIFF ZONEs. Such parameter values are likely to differ from an exact calculation based on the covered distance. These values are stored in the entity DISTANCE MATRIX ELEMENT.

In a similar way, the fare distance between SCHEDULED STOP POINTs may not be derived from the line description but stored with specific values. In such a case, a DISTANCE MATRIX ELEMENT will store the chosen value between each origin/destination pair of SCHEDULED STOP POINTs.

Therefore, a DISTANCE MATRIX ELEMENT will relate either two TARIFF ZONEs or two STOP POINTs.

7.5.3.1.2 Combined Space-based Factors

The simple spaced-based fare structures described above may be combined in more complex structures.

In most cases of fare structures using GEOGRAPHICAL INTERVALs, the fare will be flat within the range of each interval, which means that the fare is the same all along the interval. However, the fares may vary within each interval, depending on a graduation based on a GEOGRAPHICAL UNIT. Such a unit is not necessarily the same as the unit describing the interval. For instance, the fares may be graduated, the price per km differing according to the number of zones crossed (e.g. to allow lower prices for long trips).

Similarly, a graduated fare structure may be influenced by the type of trip, as regards the geography of the network. If the fare is based on the number of fare sections crossed, it may vary, for instance, depending on whether the trip is from a suburb to the city centre or between two suburbs. This structure will associate GEOGRAPHICAL INTERVALs (fare sections) and DISTANCE MATRIX ELEMENTs (using a set of TARIFF ZONEs, e.g. “centre” and “suburbs”).

The entity GEOGRAPHICAL STRUCTURE FACTOR allows to combine two simple structures in a complex factor. It is identified by a GEOGRAPHICAL UNIT, describing the used graduation unit, and by either a GEOGRAPHICAL INTERVAL or a DISTANCE MATRIX ELEMENT.

In real implementations of complex structures, GEOGRAPHICAL STRUCTURE FACTORs would probably be associated in sets related to one fare calculation rule, in order to allow an algorithm to choose the appropriate rule.

Users of simpler fare structures will implement the GEOGRAPHICAL STRUCTURE FACTOR entity by only using the parameters they require.

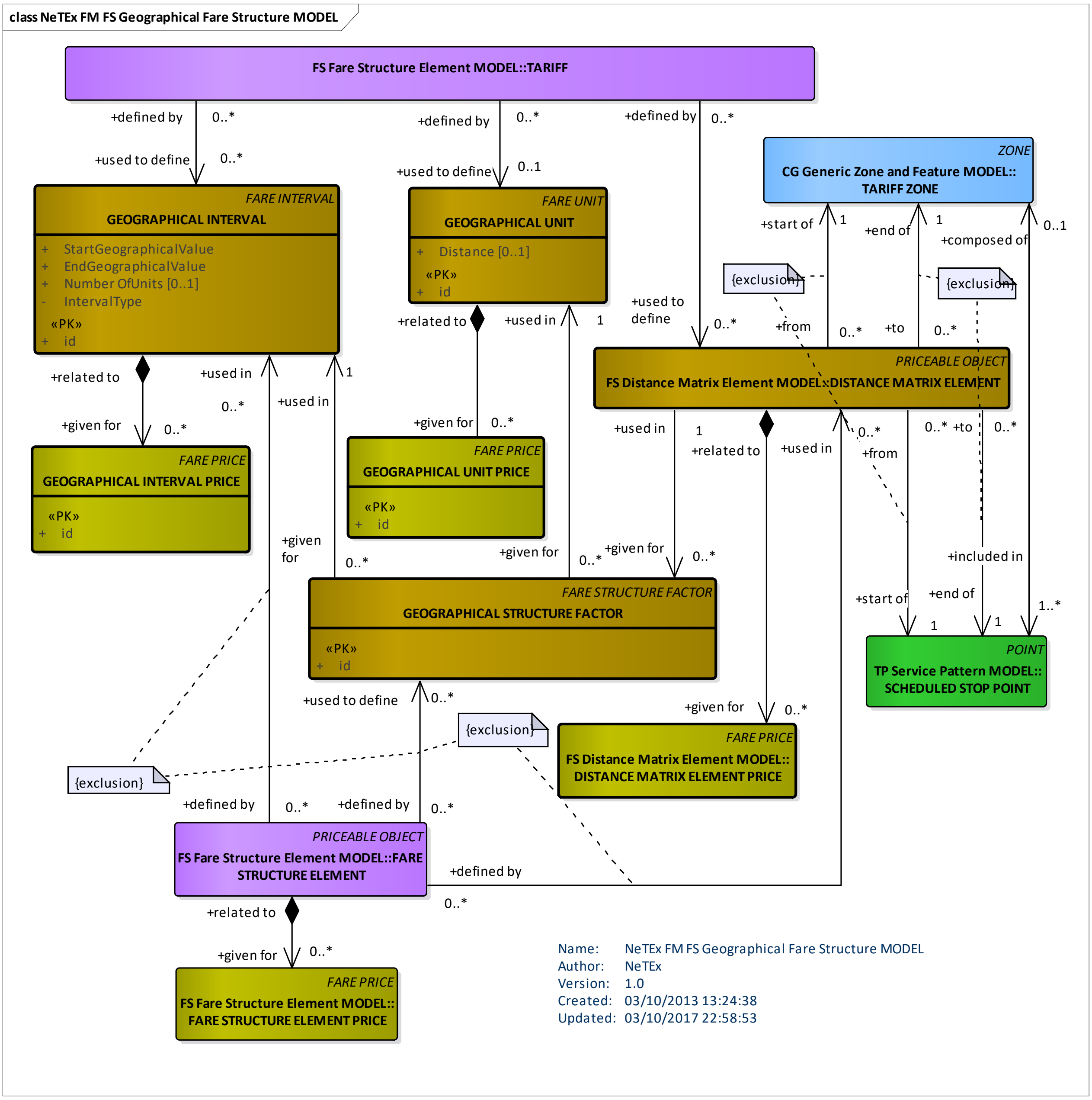


Figure 61 — Geographical Fare Structure – Conceptual MODEL (UML)

7.5.3.2 Geographical Fare Structure – Conceptual Examples

Geographical fare structures can be used with many different types of units. For example:

— Distance based fares – e.g. Kilometres.

— Fares based on the number of zones traversed.

— Fares based on the number of fare stages traversed.

7.5.3.3 Geographical Fare Structure – Physical model

The following figure shows the physical model for the GEOGRAPHICAL FARE STRUCTURE.

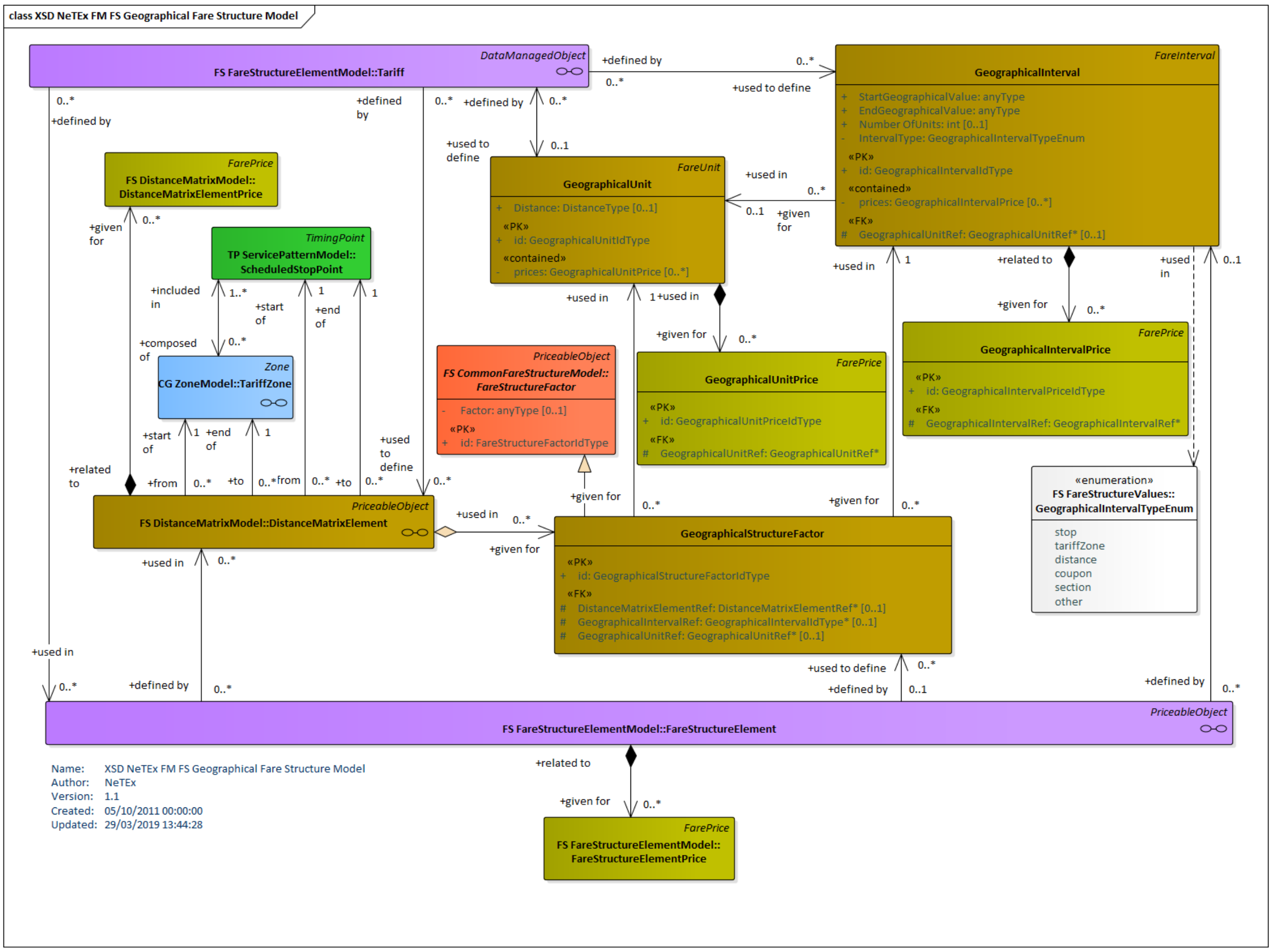


Figure 62 — Geographical Fare Structure – Physical Model (UML)

7.5.3.4 Geographical Fare Structure – Attributes and XSD

7.5.3.4.1 GeographicalUnit – Model Element

A unit for calculating geographical graduated fares.

Table 42 – *GeographicalUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	FareUnit	::>	GEOGRAPHICAL UNIT inherits from FARE UNIT.
«PK»	***id***	*GeographicalUnitIdType*	1:1	Identifier of GEOGRAPHICAL UNIT.
	Distance	*DistanceType*	0:1	If distance-based unit, length of unit.
«cntd»	***prices***	GeographicalUnitPrice	0:*	Prices associated with GEOGRAPHICAL UNIT

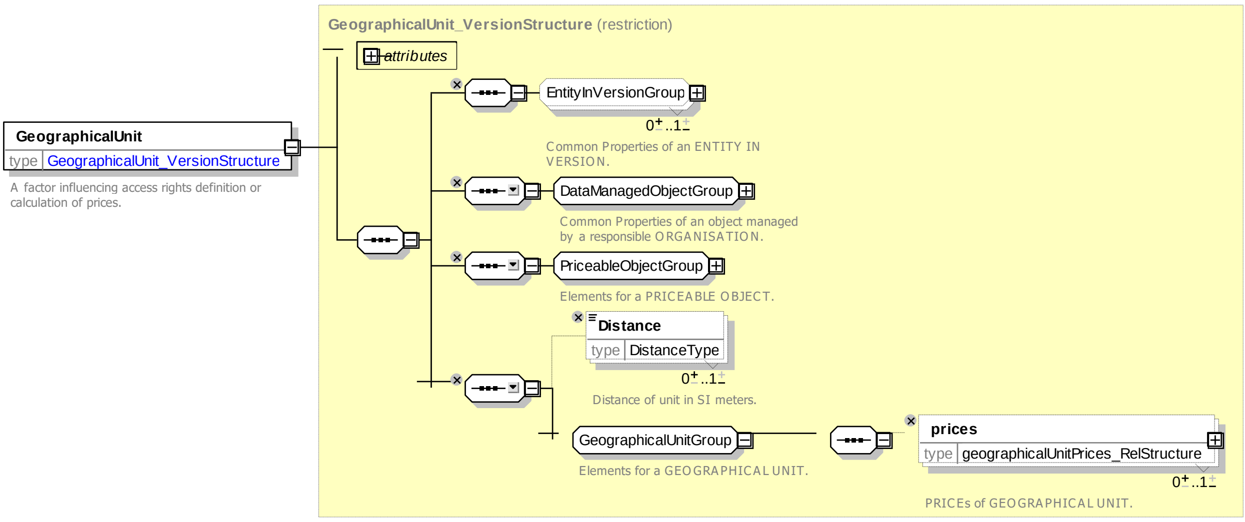


Figure 63 — *GeographicalUnit* — XSD

7.5.3.4.2 GeographicalUnitPrice – Model Element

A set of all possible price features of a GEOGRAPHICAL UNIT: default total price etc.

Table 43 – *GeographicalUnitPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	GEOGRAPHICAL UNIT PRICE inherits from FARE PRICE.
«PK»	***id***	*GeographicalUnitPriceIdType*	1:1	Identifier of GEOGRAPHICAL UNIT PRICE.
«FK»	***Geographical-UnitRef***	*GeographicalUnitRef*	0:1	Reference to GEOGRAPHICAL UNIT for which this is the price. If not given by context must be specified.

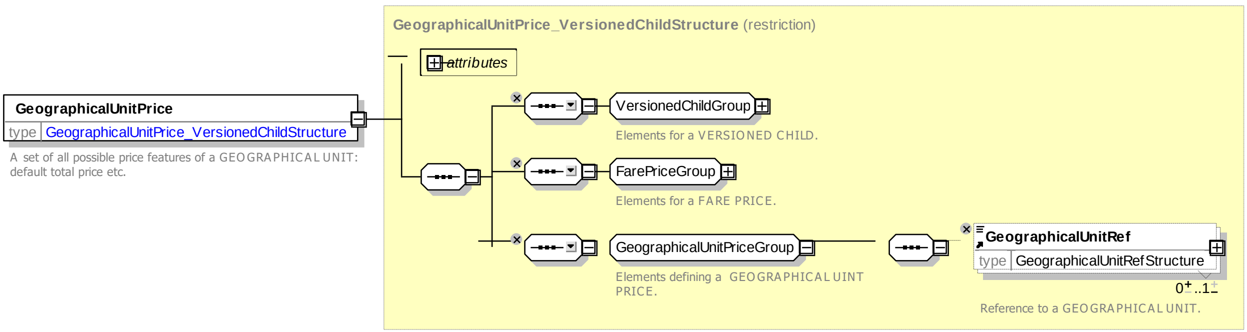


Figure 64 — *GeographicalUnitPrice* — XSD

7.5.3.4.3 GeographicalInterval – Model Element

A geographical interval specifying access rights for the FARE STRUCTURE ELEMENTs within the range of this interval: "2*0-5 km", "4-6 zones*", etc.

Table 44 – *GeographicalInterval* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	<u>*FareInterval*</u>	::>	GEOGRAPHICAL INTERVAL inherits from FARE INTERVAL.
«PK»	***id***	*GeographicalInterval-IdType*	1:1	Identifier of GEOGRAPHICAL INTERVAL.
	Start-Geographical-Value	*xsd:decimal*	0:1	Start value for GEOGRAPHICAL INTERVAL.
	End-Geographical-Value	*xsd:decimal*	0:1	End value for GEOGRAPHICAL INTERVAL.
	NumberOfUnits	*xsd:integer*	0:1	Number of units in GEOGRAPHICAL INTERVAL.
«enum»	***IntervalType***	*IntervalTypeEnum*	0:1	Classification of interval type. See allowed values below.
«FK»	***Geographical-UnitRef***	*GeographicalUnitRef*	0:1	GEOGRAPHICAL UNIT for interval.
«cntd»	***prices***	<u>*Geographical-IntervalPrice*</u>	0:*	Prices for the GEOGRAPHIC INTERVAL.

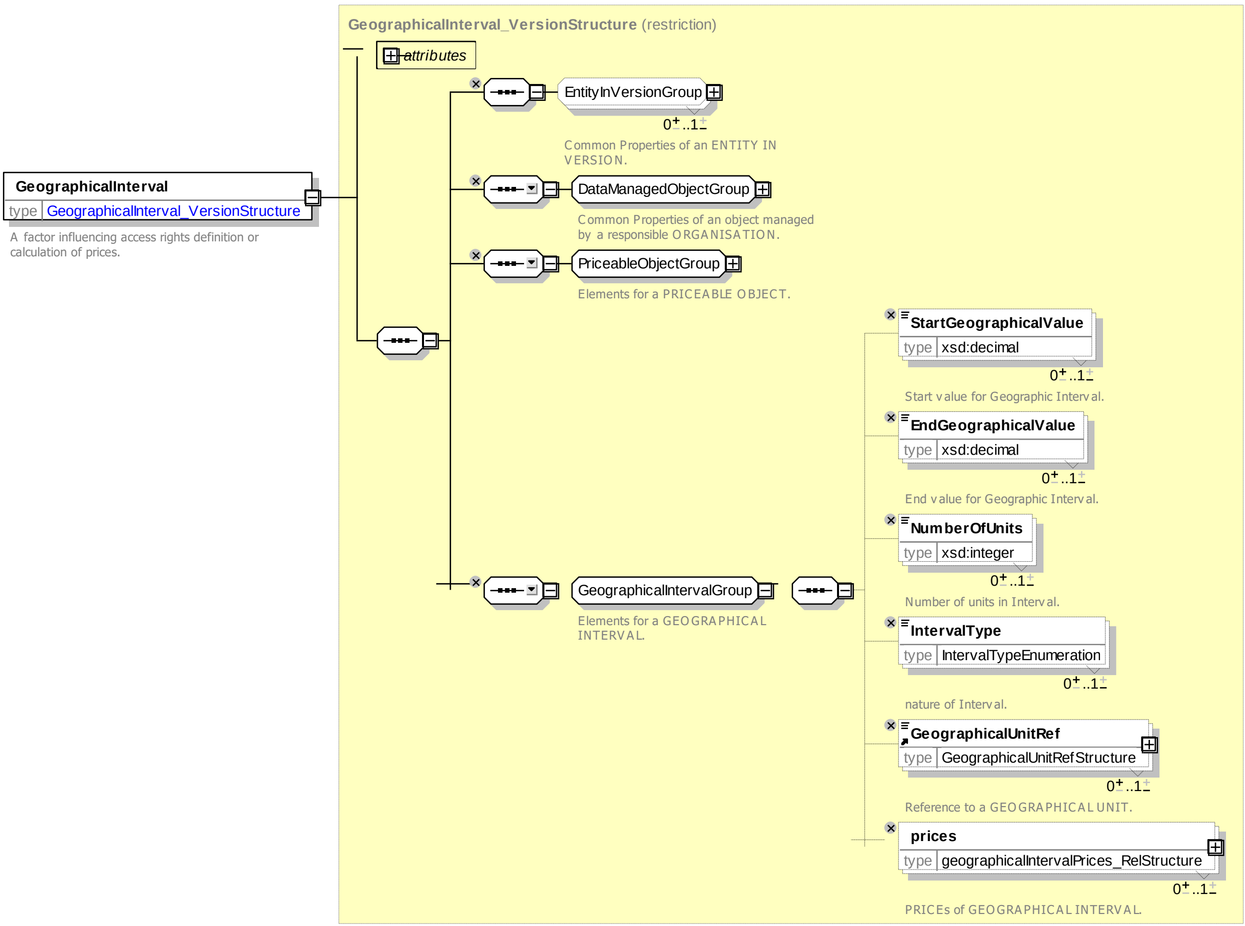


Figure 65 — *GeographicalInterval* — XSD

7.5.3.4.3.1 IntervalType – Allowed values

The following table shows the allowed values for ***IntervalType*** (*IntervalTypeEnumeration)*.

Table 45 – *IntervalType* – Allowed values

Value	Description
stop	Interval is a SCHEDULED STOP POINT.
tariffZone	Interval is a TARIFF ZONE
distance	Interval is a distance measurement.
section	Interval is a fare section. +v1.1
coupon	Interval is an arbitrary coupon unit.
other	Other

7.5.3.4.4 GeographicalIntervalPrice – Model Element

A set of all possible price features of a GEOGRAPHICAL INTERVAL: default total price etc.

Table 46 – *GeographicalIntervalPrice* – Element

Classifi-cation	Name	Type	Cardin ality	Description

::>	::>	*FarePrice*	::>	GEOGRAPHICAL INTERVAL PRICE inherits from FARE PRICE.
«PK»	***id***	*GeographicalIntervalPrice IdType*	1:1	Identifier of GEOGRAPHICAL INTERVAL PRICE.
«FK»	***Geographical-IntervalRef***	*GeographicalIntervalRef*	0:1	Reference to GEOGRAPHICAL INTERVAL for which this is the price. If not given by context must be specified.

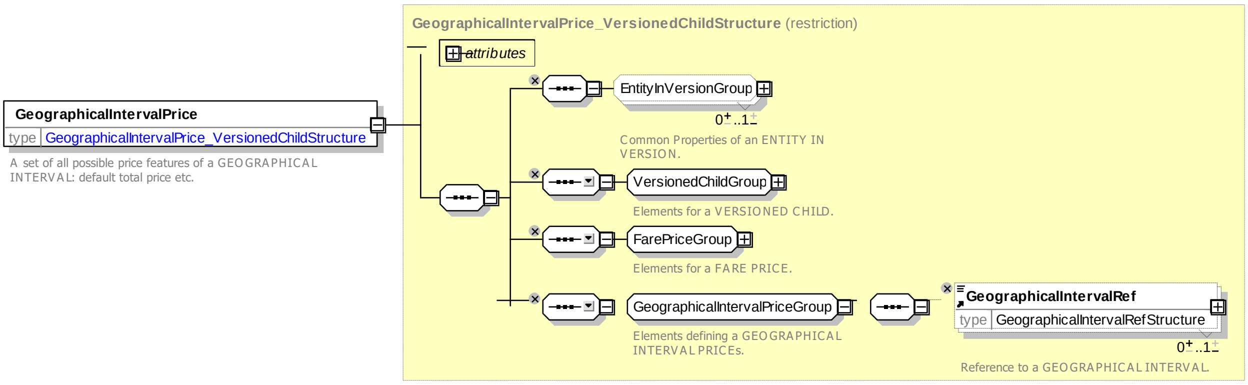


Figure 66 — *GeographicalIntervalPrice* — XSD

7.5.3.4.5 GeographicalStructureFactor.

The value of a GEOGRAPHICAL INTERVAL or a DISTANCE MATRIX ELEMENT expressed by a GEOGRAPHICAL UNIT.

Table 47 – *GeographicalStructureFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareStructureFactor*	::>	GEOGRAPHICAL STRUCTURE FACTOR inherits from FARE STRUCTURE FACTOR.
«PK»	***id***	*GeographicalStructure-FactorRef*	1:1	Identifier of GEOGRAPHICAL STRUCTURE FACTOR.
«FK»	***TariffRef***	*TariffRef*	0:1	Reference to a TARIFF.
«FK»	***DistanceMatrix-ElementRef***	*DistanceMatrix-ElementRef*	0:1	Reference to a DISTANCE MATRIX ELEMENT.
«FK»	***Geographical-IntervalRef***	*Geographical-IntervalIdType*	0:1	Reference to a GEOGRAPHICAL INTERVAL.
«FK»	***Geographical-UnitRef***	*GeographicalUnitRef*	0:1	Reference to GEOGRAPHICAL UNIT.
	NumberOfUnits	*NumberOfUnits*	0:1	Quantity of units.

	AmountFactor	*xsd:decimal*	0.1	Arbitrary amount factor associated with GEOGRAPHICAL STRUCTURE FACTOR.

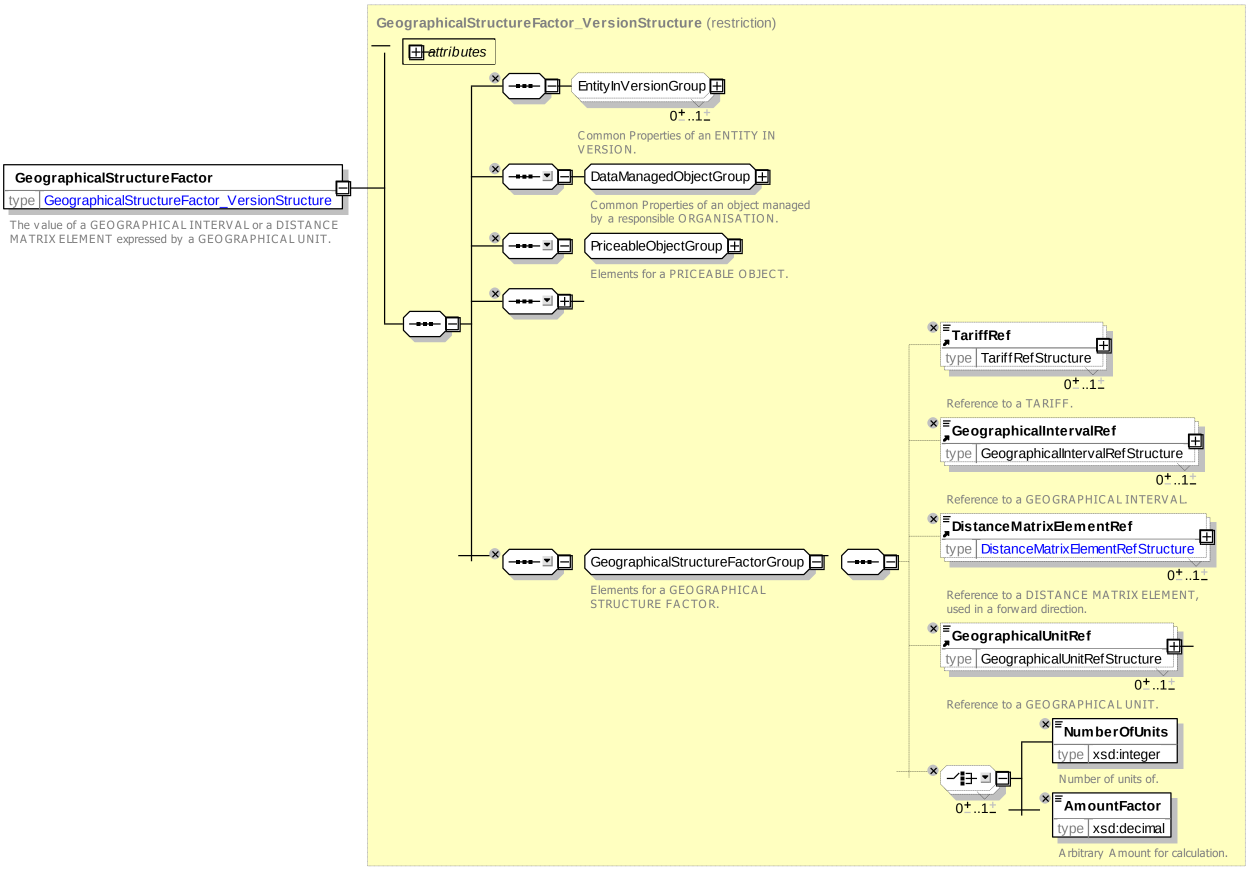


Figure 67 — *GeographicalStructureFactor* — XSD

7.5.3.5 Geographical Fare Structure – XML examples

7.5.3.5.1 Geographical Fare Structure: XML Example of stepped geographical intervals

The following code fragment shows the GEOGRAPHICAL STRUCTURE FACTOR definitions for a distance-based fare structure distance based that uses three bands for three different ranges of kilometres (*less than 50 kilometres; 50 to 150 kilometres; over 150 kilometres*)

For EXAMPLE:

```
<!-- === GEO FACTORs == -->
    <geographicalUnits>
        <GeographicalUnit id="tap:kilometre" version="any">
            <Name>Kilometre</Name>
        </GeographicalUnit>
    </geographicalUnits>

    <geographicalIntervals>
        <GeographicalInterval id="tap:up_to_50_ kilometer " version="any">
            <Name>Price per Kilometre up to 50 kilometers</Name>
            <StartGeographicalValue>1</StartGeographicalValue>
            <EndGeographicalValue>100</EndGeographicalValue>
            <prices>
                <GeographicalIntervalPrice id="tap:up_to_50_kilometer" version="any">
                    <Amount>1.30</Amount>
```

```
                </GeographicalIntervalPrice>
            </prices>
        </GeographicalInterval>

        <GeographicalInterval id="tap:50+100_Kilometre" version="any">
            <Name>Price per Kilometre 50 to 150 kilometers</Name>
            <StartGeographicalValue>51</StartGeographicalValue>
            <EndGeographicalValue>150</EndGeographicalValue>
            <prices>
                <GeographicalIntervalPrice id="tap:50+100_Kilometer" version="any">
                    <Amount>1.00</Amount>
                </GeographicalIntervalPrice>
            </prices>
        </GeographicalInterval>

        <GeographicalInterval id="tap: 150_kilometer_upwards " version="any">
            <Name>Price per Kilometre over 150 kilometers</Name>
            <StartGeographicalValue>151</StartGeographicalValue>
            <EndGeographicalValue>9999</EndGeographicalValue>
            <prices>
                <GeographicalIntervalPrice id="tap:150_kilometer_upwards" version="01">
                    <Amount>0.80</Amount>
                </GeographicalIntervalPrice>
            </prices>
        </GeographicalInterval>
    </geographicalIntervals>

    <geographicalStructureFactors>
        <GeographicalStructureFactor id="tap:rail_kilometer" version="01">
            <Name>Rail distance unit</Name>
            <GeographicalUnitRef ref="tap:kilometer"/>
        </GeographicalStructureFactor>
    </geographicalStructureFactors>
```

7.5.4 Time Fare Structure

7.5.4.1 Time Fare Structure – Conceptual MODEL

The TIME FARE STRUCTURE model describes the temporal aspects of the fare structure model.

The time-based fare structures are described in a similar way to the space-based structures. The entity TIME INTERVAL describes intervals of time (0-1 hour, 1-3 hours, etc.) during which a certain fare is applied to FARE STRUCTURE ELEMENTs. A graduated time-based structure will be defined using a TIME UNIT (e.g. days, hours or minutes).

Both types of structures may be combined into TIME STRUCTURE FACTORs. This allows for instance to specify a fare per hour spent, which varies depending on the range of days spent.

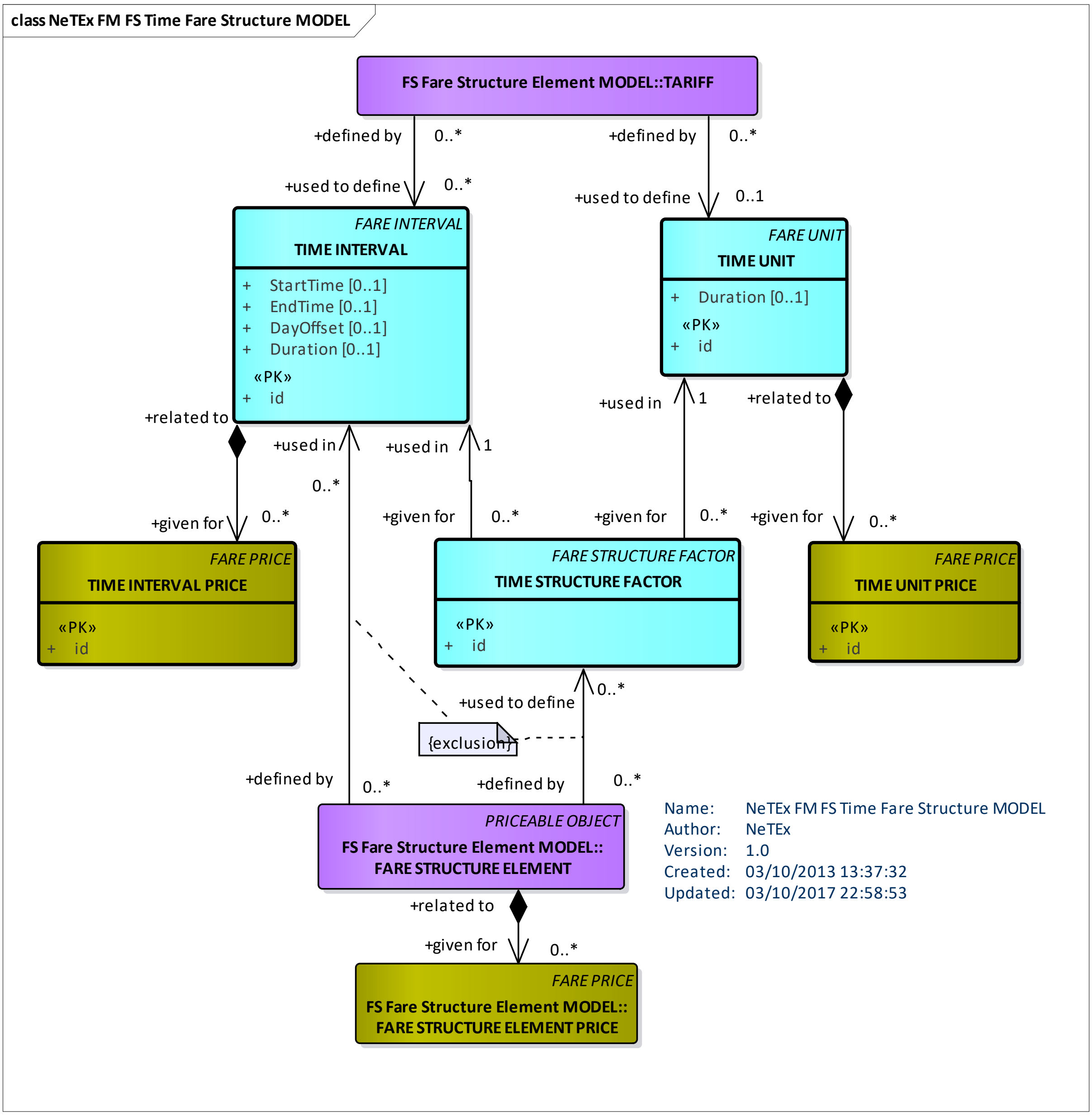


Figure 68 — Time Fare Structure – Conceptual MODEL (UML)

7.5.4.2 Time Fare Structure – Physical model

The following figure shows the physical model for the TIME FARE STRUCTURE.

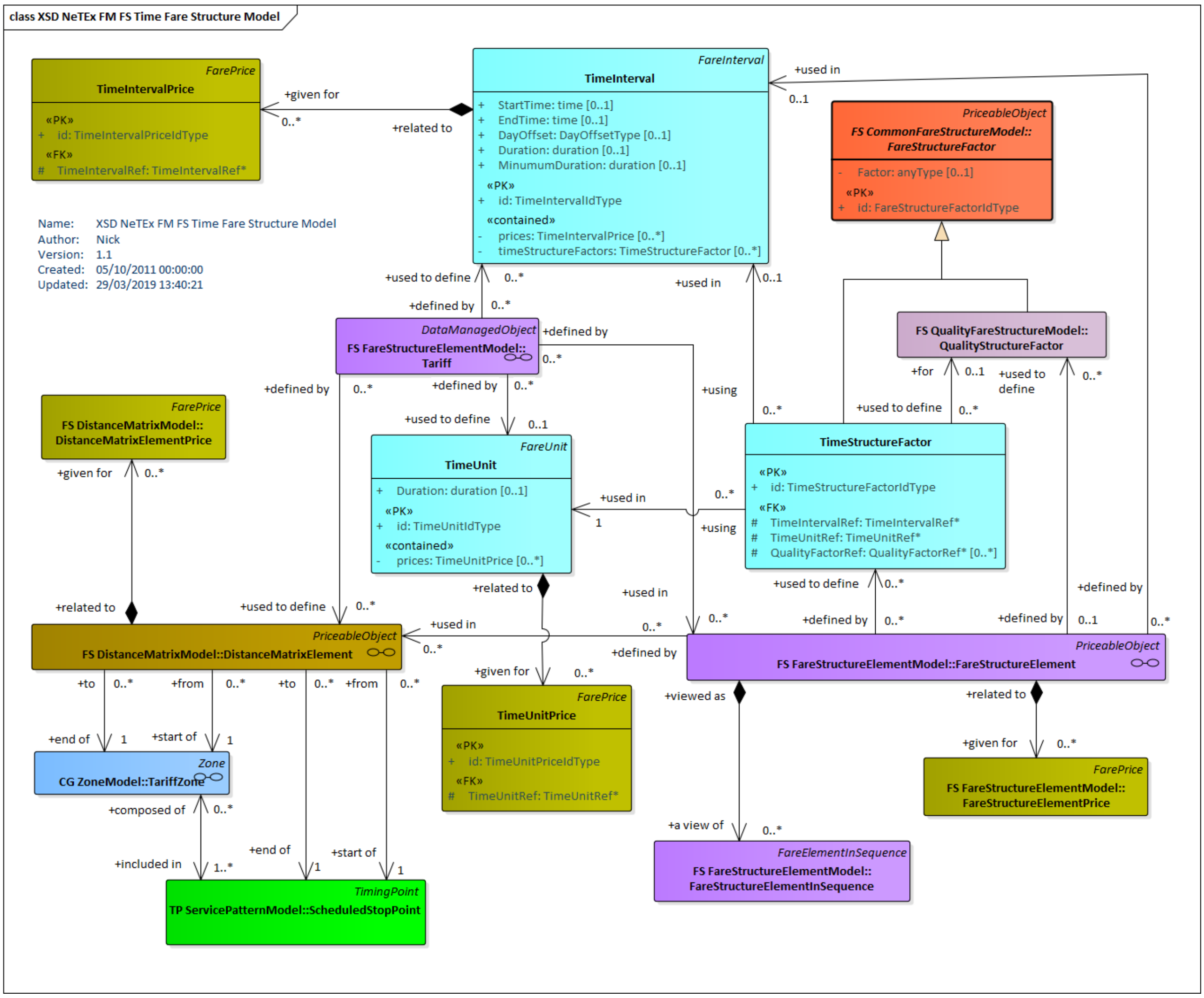


Figure 69 — Time Fare Structure – Physical Model (UML)

7.5.4.3 Time Fare Structure – Conceptual Examples

It is quite common to have a time-based fare structure, for example:

- — determined through the entity TIME INTERVAL that describes intervals of time (0-1 hour, 1-3 hours, etc.) during which a certain fare is applied to FARE STRUCTURE ELEMENTs.
- — a graduated time-based structure defined using a TIME UNIT (e.g. days, hours or minutes).

7.5.4.4 Time Fare Structure – Attributes and XSD

7.5.4.4.1 TimeUnit – Model Element

A unit for calculating time-based graduated fares.

Table 48 – *TimeUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareUnit*	::>	TIME UNIT inherits from FARE UNIT.
«PK»	***id***	*TimeUnitIdType*	1:1	Identifier of TIME UNIT.
	Type	*xsd:NCName*	0:1	Name of XML class associated with unit e.g. gday, gMonth. This is metadata.
	Duration	*xsd:duration*	0:1	Duration associated with unit, e.g. P1D, PT1S.
	prices	*TimeUnitPrice*	0:*	Prices associated with TIME UNIT.

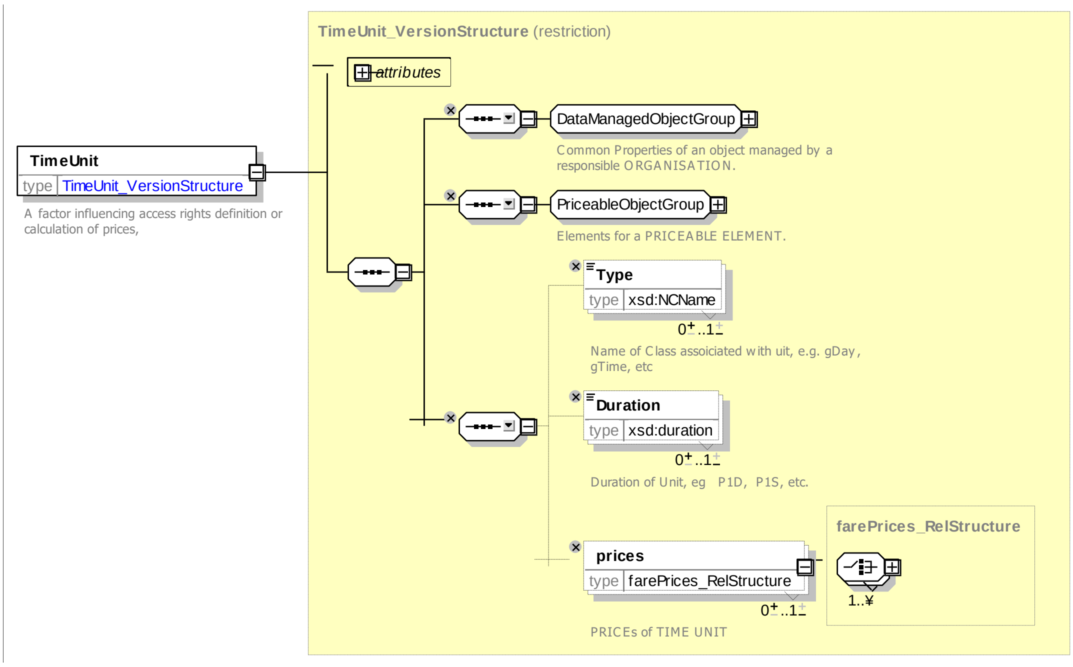


Figure 70 — *TimeUnit* — XSD

7.5.4.4.2 TimeUnitPrice – Model Element

A set of all possible price features of a TIME UNIT: default total price etc.

Table 49 – *TimeUnitPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	TIME UNIT PRICE inherits from FARE PRICE.
«PK»	***id***	*TimeUnitPrice IdType*	1:1	Identifier of TIME UNIT PRICE.
«FK»	***TimeUnitRef***	*TimeUnitRef*	0:1	Reference to TIME UNIT for which this is the price. If not given by context must be specified.

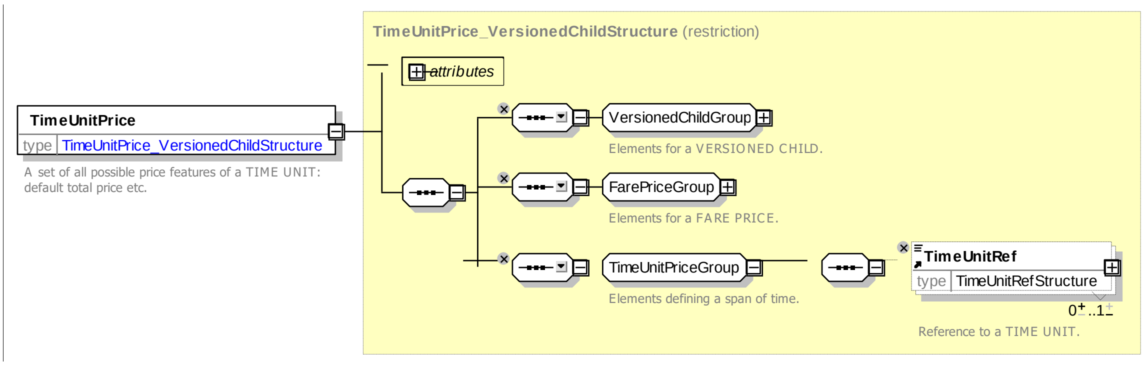


Figure 71 — *TimeUnitPrice* — XSD

7.5.4.4.3 TimeInterval – Model Element

A time-based interval specifying access rights for the FARE STRUCTURE ELEMENTs within the range of this interval: *"0-1 hours," "1-3 days"*, etc.

Table 50 – *TimeInterval* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareInterval*	::>	TIME INTERVAL inherits from FARE INTERVAL.
«PK»	***id***	*TimeIntervalIdType*	1:1	Identifier of TIME INTERVAL.
	StartTime	*xsd:time*	0:1	Start of TIME INTERVAL.
	EndTime	*xsd:time*	0:1	End of TIME INTERVAL.
	DayOffset	*DayOffsetType*	0:1	Day offset of end time from start time.
	Duration	*xsd:duration*	0:1	Interval expressed as duration.
	Minimum-Duration	*xsd:duration*	0:1	Minimum Duration for TIME INTERVAL. +v1.1
«cntd»	***prices***	*TimeIntervalPrice*	0:*	Prices for the TIME INTERVAL.
«cntd»	***timeStructure-Factors***	*TimeStructureFactor*	0:*	TIME STRUCTURE FACTORs using the TIME INTERVAL.

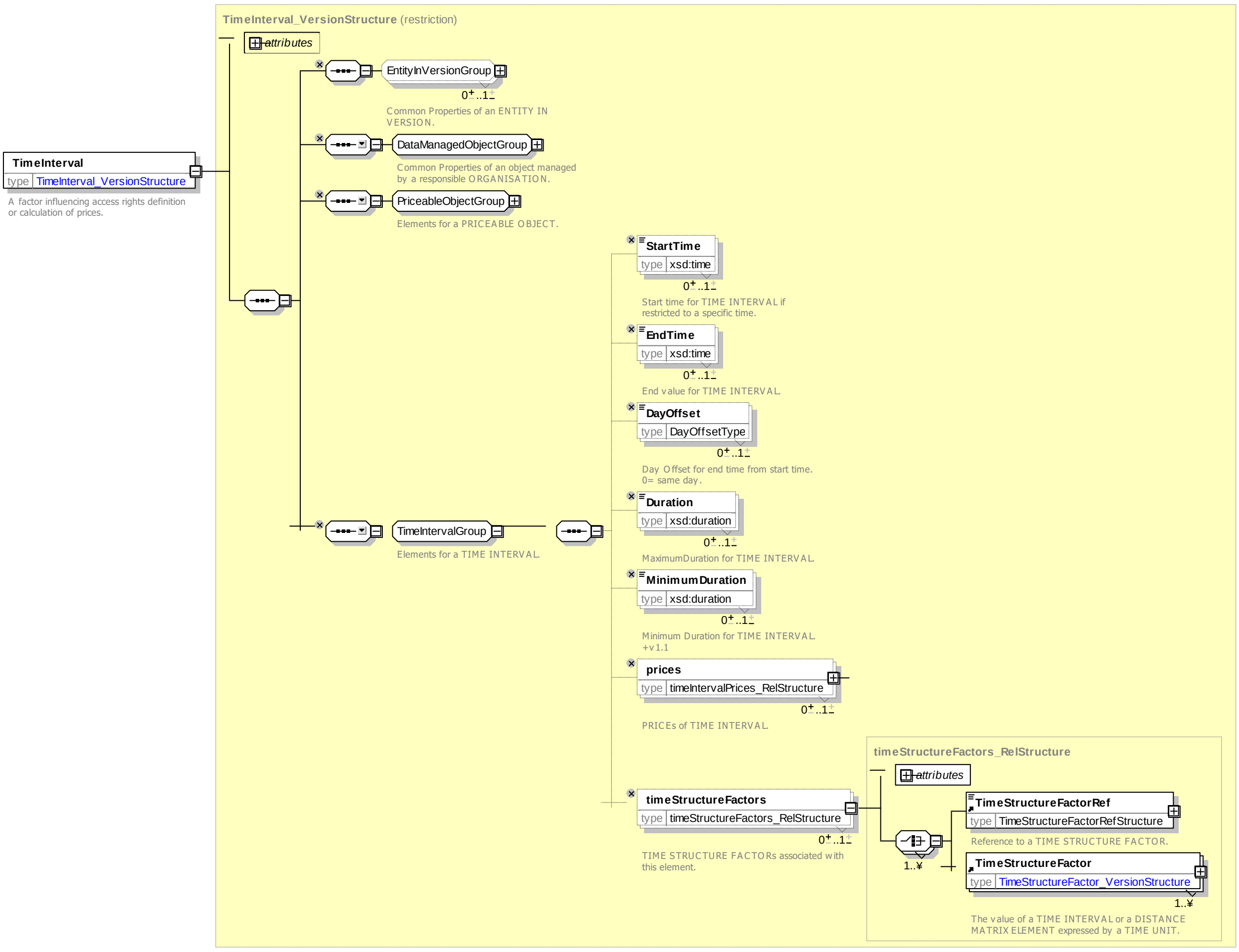


Figure 72 — *TimeInterval* — XSD

7.5.4.4.4 TimeIntervalPrice – Model Element

A set of all possible price features of a TIME INTERVAL, e.g. default total price etc.

Table 51 – *TimeIntervalPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	TIME INTERVAL PRICE inherits from FARE PRICE.
«PK»	***id***	*TimeIntervalPriceIdType*	1:1	Identifier of TIME INTERVAL PRICE.
«FK»	***TimeIntervalRef***	*TimeIntervalRef*	0:1	Reference to TIME INTERVAL for which this is the price. If not given by context must be specified.

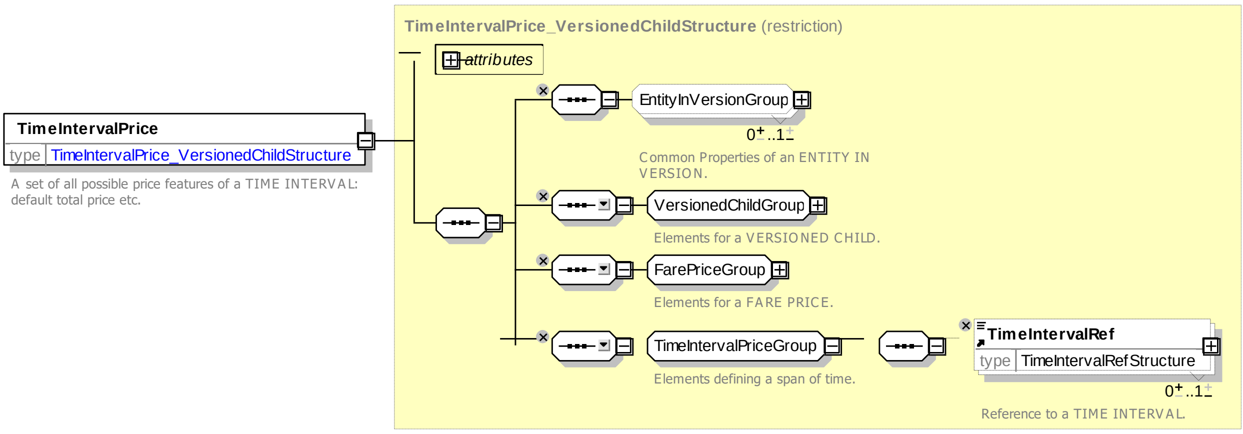


Figure 73 — *TimeIntervalPrice* — XSD

7.5.4.4.5 TimeStructureFactor – Model Element

The value of a TIME INTERVAL expressed by a TIME UNIT.

Table 52 – *TimeStructureFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareStructureFactor*	::>	TIME STRUCTURE FACTOR inherits from FARE STRUCTURE FACTOR.
«PK»	***id***	*TimeStructureFactor-IdType*	1:1	Identifier of TIME STRUCTURE FACTOR.
«FK	***TariffRef***	*TariffRef*	0:1	Reference to TARIFF associated with TIME STRUCTURE FACTOR.
«FK»	***TimeIntervalRef***	*TimeIntervalRef*	0:1	Reference to TIME INTERVAL associated with factor.
«FK»	***TimeUnitRef***	*TimeUnitRef*	0:1	Reference to TIME UNIT associated with factor.
«FK»	***QualityStructure-FactorRef***	*QualityStructure-FactorRef*	0:*	QUALITY FACTOR associated with the TIME STRUCTURE FACTOR.

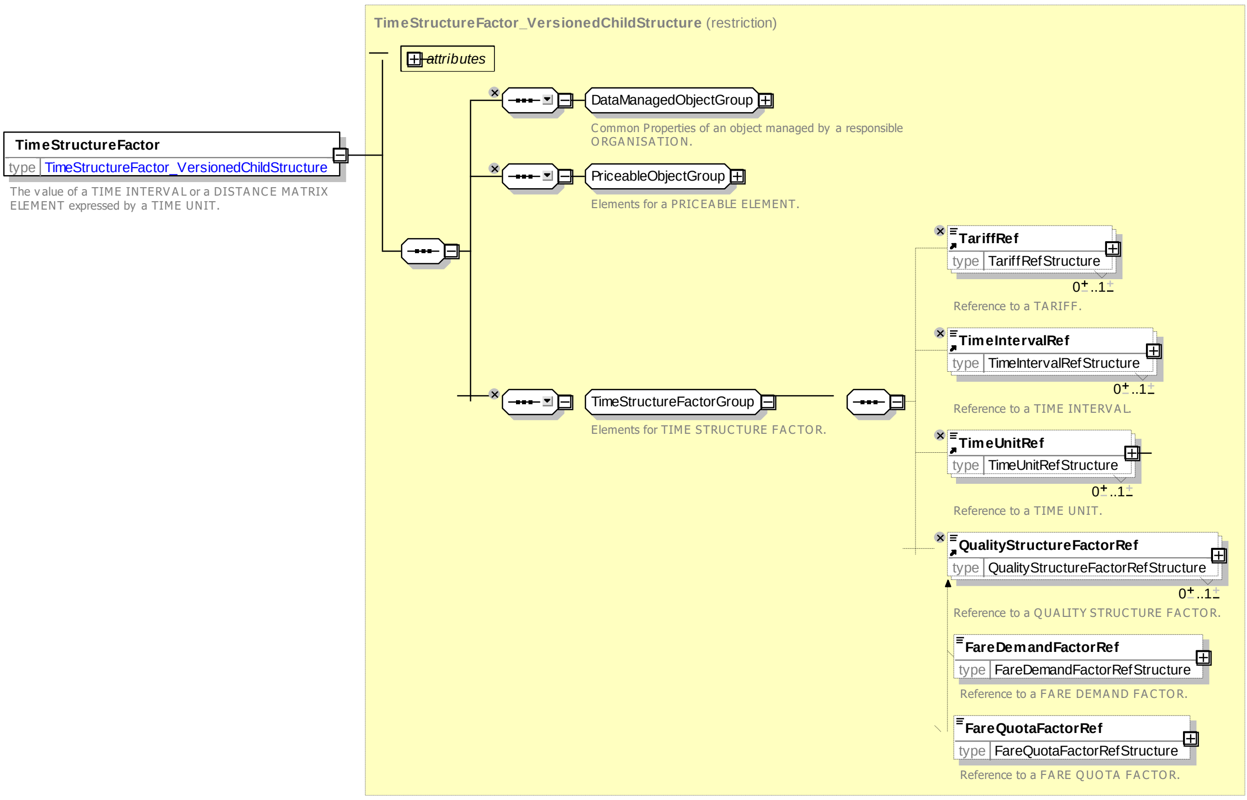


Figure 74 — *TimeStructureFactor* — XSD

7.5.4.5 Time Fare Structure – XML examples

7.5.4.5.1 Time Fare Structure: XML Example of Time Intervals

The following code fragment (based on TfL) shows a TARIFF with TIME INTERVAL and TIME STRUCTURE FACTOR definitions for the various intervals for which tickets are available; one hour, two hours or all day. In this last case, the all-day use ends at 02am the next day, (and not 24 hours after validation). A FARE STRUCTURE ELEMENT is then able to reference the available TIME STRUCTURE FACTORs.

For EXAMPLE:

```
<Tariff version="any" id="myfares:time_interval">
    <Name>Zonal Fare</Name>
    <TimeUnitRef version="any" ref="myfares:tu1"/>

    <timeIntervals>
        <TimeInterval version="any" id="myfares:ti_1h">
            <Name>One Hour</Name>
            <Duration>PT1H</Duration>
            <timeStructureFactors>
                <TimeStructureFactor version="any" id="myfares:tsf_1h">
                    <Name>One hour's use</Name>
                    <TimeUnitRef version="any" ref="myfares:hour "/>
                </TimeStructureFactor>
            </timeStructureFactors>
        </TimeInterval>
        <TimeInterval version="any" id="myfares:ti_2h">
            <Name>Two Hours</Name>
            <Duration>PT2H</Duration>
            <timeStructureFactors>
<               TimeStructureFactor version="any" id="myfares:tsf_2h">
                    <Name>Two hour's use</Name>
                    <TimeUnitRef version="any" ref="myfares:hour"/>
                </TimeStructureFactor>
            </timeStructureFactors>
```

```
        </TimeInterval>
        <TimeInterval version="any" id="myfares:ti_day">
            <Name>All Day Use on day of purchase. Fare day ends at 2 am</Name>
            <StartTime>06:00:00</StartTime>
            <EndTime>02:00:00</EndTime>
            <DayOffset>1</DayOffset>
            <timeStructureFactors>
                <TimeStructureFactor version="any" id="myfares:tsf_day">
                    <Name>All day use on day of purchase</Name>
                    <TimeUnitRef version="any" ref="myfares:day"/>
                </TimeStructureFactor>
            </timeStructureFactors>
        </TimeInterval>
    </timeIntervals>
Etc, etc.
    <fareStructureElements>
        <FareStructureElement version="any" id="myfares:durations">
            <timeStructureFactors>
                <TimeStructureFactorRef version="any" ref="myfares:tsf_1h"/>
                <TimeStructureFactorRef version="any" ref="myfares:tsf_2h"/>
                <TimeStructureFactorRef version="any" ref="myfares:tsf_day"/>
            </timeStructureFactors>
        </FareStructureElement>
    </fareStructureElements>
</Tariff>
```

7.5.5 Quality Fare Structure

7.5.5.1 Quality Fare Structure – Conceptual MODEL

QUALITY FARE STRUCTURE can be used to define arbitrary fare structure qualities**.**

For instance, the current level of congestion or occupancy (e.g. in %) may influence the fare or a limitation to the access rights. Some rail operators apply different fares if the reservation is made early or late (e.g. in number of days). Such a possibility is simply described by the entity QUALITY STRUCTURE FACTOR.

Two specialisations can be used for specific aspects: A FARE DEMAND FACTOR defines a ‘time band’ for travel, e.g. *peak* or *off-peak*, and a FARE QUOTA FACTOR defines a limited allocation of seats available at a particular price.

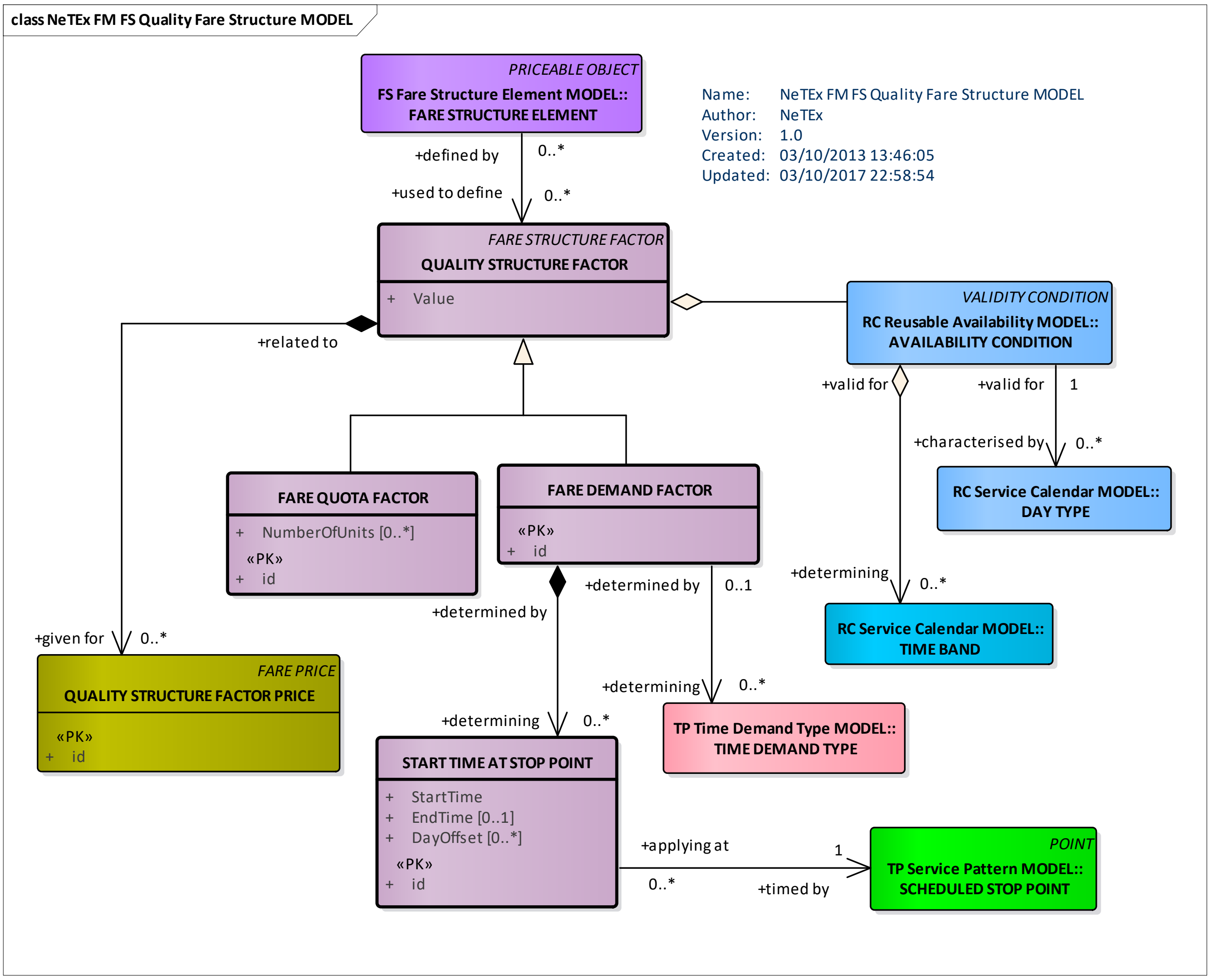


Figure 75 — Quality Fare Structure – Conceptual MODEL (UML)

7.5.5.2 Quality Fare Structure – Physical model

The following figure shows the physical model for the QUALITY FARE STRUCTURE.

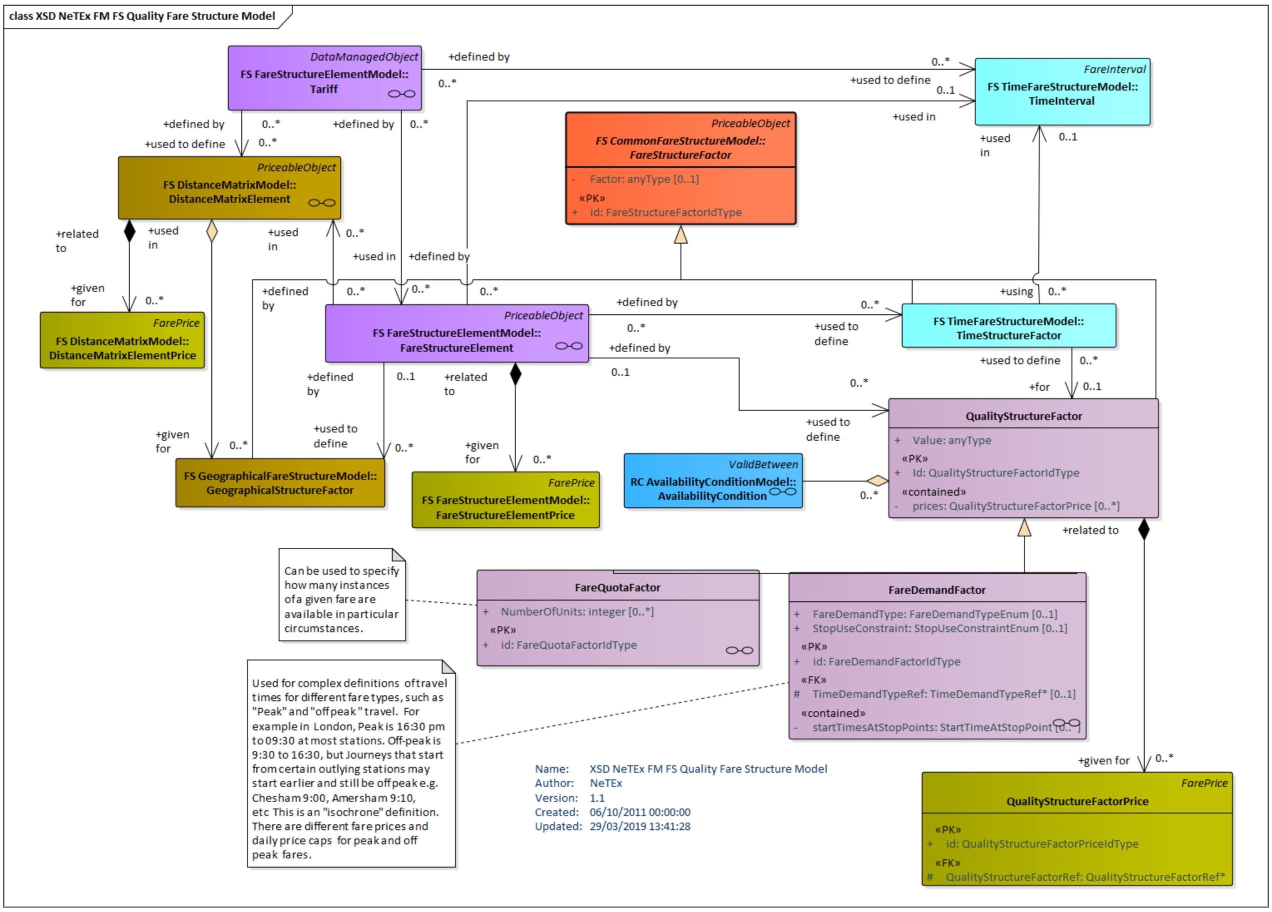


Figure 76 — Quality Fare Structure – Physical Model (UML)

7.5.5.3 Fare Demand Factor – Physical model

The following figure shows the physical model for the FARE DEMAND FACTOR. A FARE DEMAND FACTOR specifies a named period for travelling such as 'peak' or 'off peak'. The standard NeTEx VALIDITY CONDITION elements (See NeTEx Part1) can be used to specify the day types and timebands of the period. In large networks this furthermore may have a complex definition whereby the start and end are not uniform over the whole network but vary from stop to stop. This can be represented by the START TIME AT STOP element.

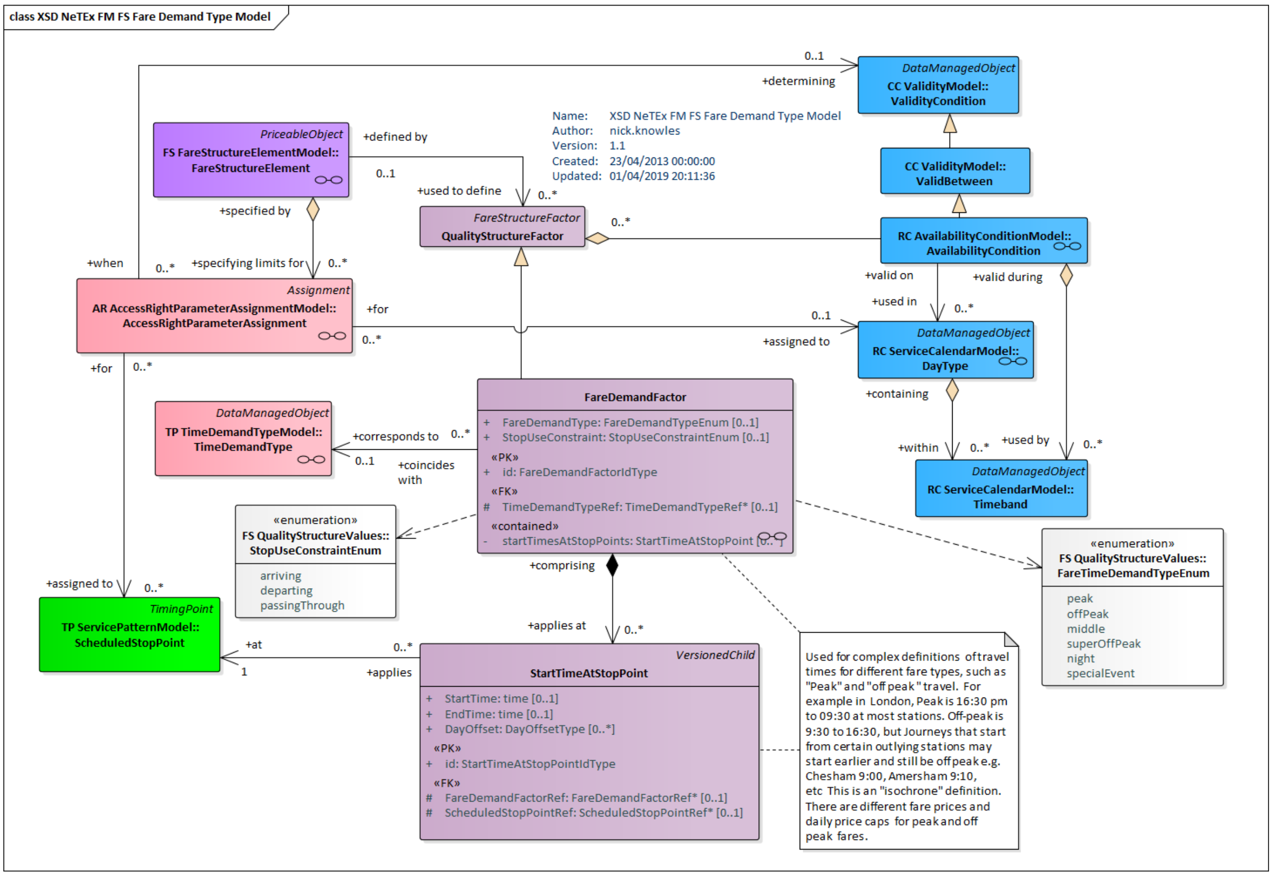


Figure 77 — Fare Demand Factor– Physical Model

7.5.5.4 Quality Fare Structure – Examples

It is quite common to have a time-based fare structure. Real-world examples include**.**

7.5.5.4.1 **Example – Fare Demand Factors**

The following example shows a definition of peak and off-peak times for the London underground, which can be represented as FARE DEMAND FACTORs. Note that for certain stations distant from the centre, the off-peak period starts at a different time.

Peak

- Monday-Friday 04:30-09:29

Off-peak

- Monday-Friday from 09:30-04:29 (the following day)
- Saturdays, Sundays and public holidays from 04:30-04:29 (the following day)

Special peak and off-peak times apply on journeys that start at stations north of Moor Park and from some Southern Region stations.

Figure 78 — *Fare Demand Factor* — TfL metro Example (EXM)

Stations north of Moor Park or Hatch End and Southern Region stations

If you travel from a station north of Moor Park or Hatch Endon a weekday after the times below, your Oyster single fare will count towards the off-peak cap instead of the peak cap.

North of Moor Park

Station	Touch in times
Chesham	After 09:00
Amersham	After 09:10
Chalfont & Latimer	After 09:15
Chorleywood	After 09:15
Rickmansworth	After 09:20

North of Hatch End

Station	Touch in times
Bushey	After 09:20
Carpenders Park	After 09:20
Watford High Street	After 09:10

Figure 79 — *Start Times At Stop Point* — TfL Metro Example (EXM)

7.5.5.5 Quality Fare Structure – Attributes and XSD

7.5.5.5.1 QualityStructureFactor – Model Element

A factor influencing access rights definition or calculation of prices, based on the quality: traffic congestion threshold, early/late reservation etc.

Table 53 – *QualityStructureFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareStructureFactor*	::>	QUALITY STRUCTURE FACTOR inherits from FARE STRUCTURE FACTOR.
«PK»	***id***	*QualityStructure-FactorIdType*	1:1	Identifier of QUALITY STRUCTURE FACTOR.
	Factor	*xsd:anyType*	0:1	General Factor amount.
	Value	*xsd:anyType*	0:1	Quantitative quality value.

	prices	*QualityStructureFactor-Price*	0:*	Price for QUALITY STRUCTURE FACTOR.

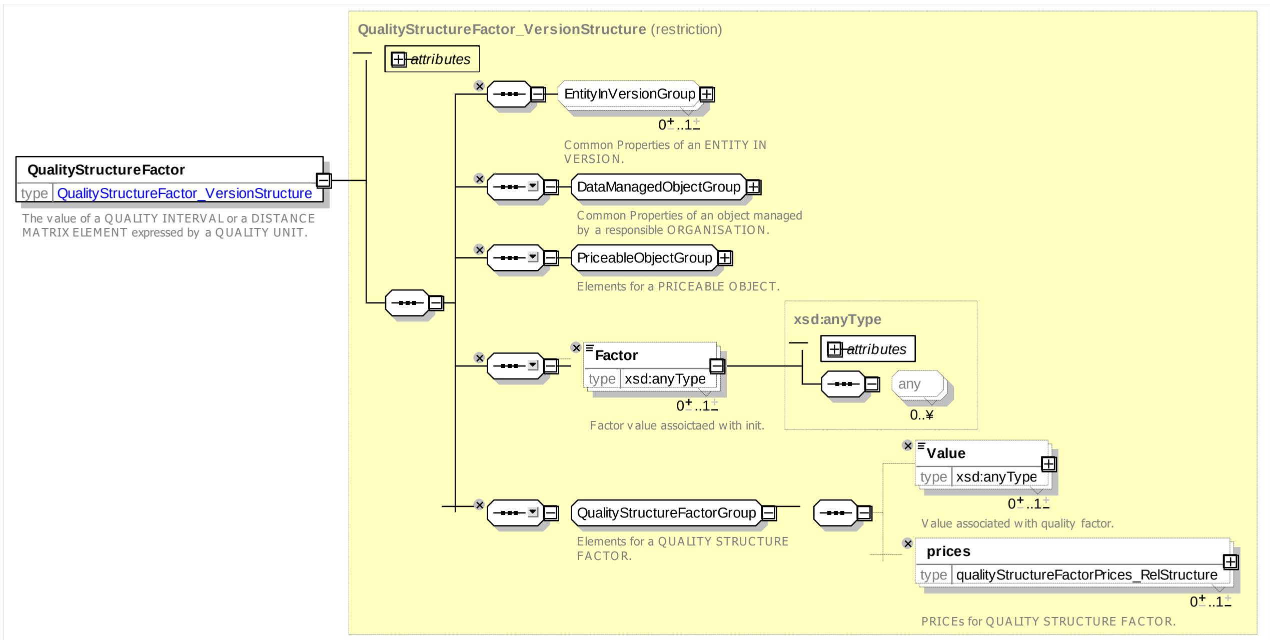


Figure 80 — *QualityStructureFactor* — XSD

7.5.5.5.2 QualityStructureFactorPrice – Model Element

A set of all possible price features of a QUALITY STRUCTURE FACTOR, e.g. default total price etc.

Table 54 – *QualityStructureFactorPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	QUALITY STRUCTURE FACTOR PRICE inherits from FARE PRICE.
«PK»	***id***	*QualityStructure-FactorPriceIdType*	1:1	Identifier of QUALITY STRUCTURE FACTOR PRICE.
«FK»	***QualityStructure FactorRef***	*QualityStructure-FactorRef*	0:1	Reference to QUALITY STRUCTURE FACTOR for which this is the price. If not given by context must be specified.

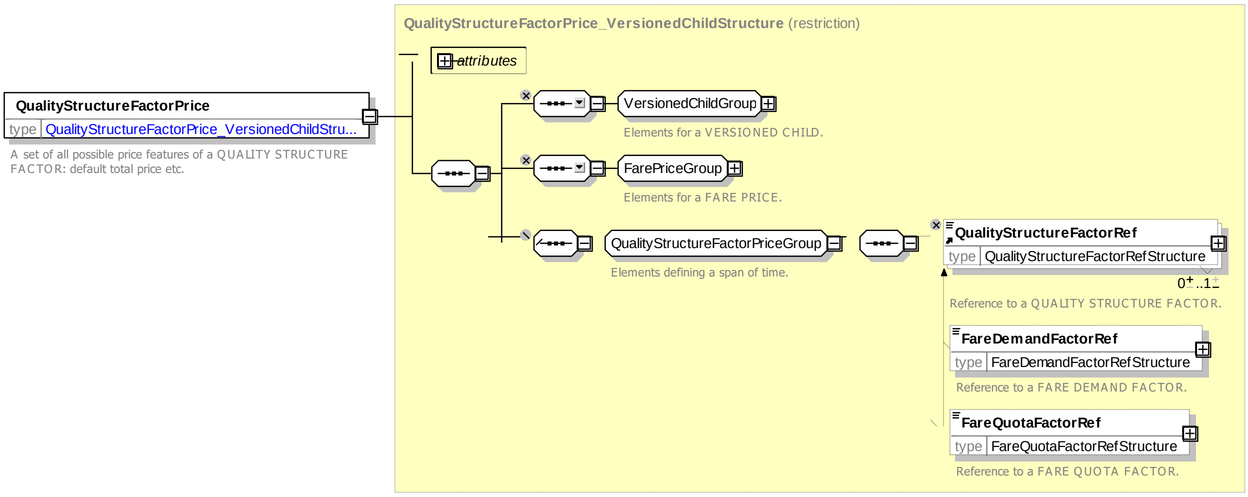


Figure 81 — *QualityStructureFactorPrice* — XSD

7.5.5.5.3 FareDemandFactor – Model Element

A named set of parameters defining a period of travel with a given price, for example; "*peak", "off-peak", "super off-peak",* etc.

Table 55 – *FareDemandFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*QualityStructureFactor*</u>	::>	FARE DEMAND FACTOR inherits from QUALITY STRUCTURE FACTOR.
«PK»	***id***	*FareDemandFactorIdType*	1:1	Identifier of a FARE DEMAND FACTOR.
«enum»	***FareDemand-Type***	*FareDemandTypeEnum*	0:1	TIME DEMAND TYPE corresponding to FARE DEMAND FACTOR. See allowed values below.
«FK»	***TimeDemand-TypeRef***	*TimeDemandTypeRef*	0:1	TIME DEMAND TYPE corresponding to FARE DEMAND FACTOR. See NeTEx Part2.
«enum»	***StopUse-Constraint***	*StopUseConstraintEnum*	0:1	Nature of constraint on uses of stop. See allowed values. +v1.1,
«cntd»	***startTimesAt-StopPoints***	<u>*StartTimeAtStopPoint*</u>	0:*	Start times at SCHEDULED STOP POINTs for FARE DEMAND TYPE.

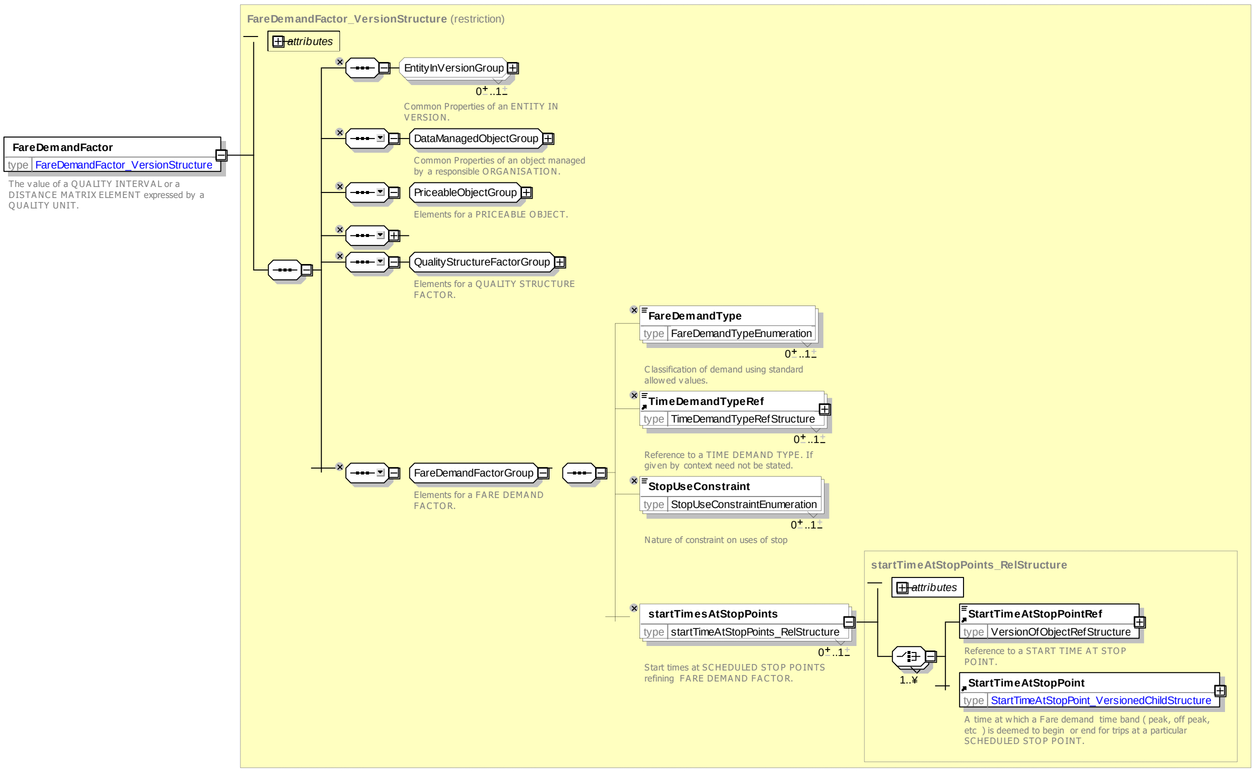


Figure 82 — *FareDemandFactor* — XSD

7.5.5.5.3.1 FareDemandType – Allowed values

The following table shows the allowed values for ***FareDemandType*** (*FareDemandTypeEnumeration*)

Table 56 – *FareDemandType* – Allowed values

Value	Description
peak	Peak travel time.
middle	Between peak and off-peak travel time.
offPeak	Off Peak travel time.
superOffPeak	Very low demand time. +v1.1
night	Night travel time.
specialEvent	Special event demand.

7.5.5.5.3.2 StopUseConstraint – Allowed values

The following table shows the allowed values for ***StopUseConstraint*** (*StopUseConstraintEnumeration*)

Table 57 – *StopUseConstraint* – Allowed values

Value	Description
arriving	Constraint is on arriving at stop.
departing	Constraint is on departing from stop.
passingThrough	Constraint is on passing through stop.
night	Night travel time.

7.5.5.5.4 **StartTimeAtStopPoint.**

A time at which a fare time band (time band peak, off peak) is deemed to begin for trips starting at a particular station.

Table 58 – *StartTimeAtStopPoint* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	START TIME AT STOP POINT inherits from VERSIONED CHILD. See NeTEx Part1.
«PK»	***id***	*StartTimeAtStop-PointIdType*	1:1	Identifier of START TIME AT STOP POINT
«FK»	***FareDemand-FactorRef***	*FareDemandFactorRef*	0:1	FARE DEMAND FACTOR for which start time applies.
«FK»	***ScheduledStop-PointRef***	*ScheduledStopPointRef*	1:1	SCHEDULED STOP POINT at which time band start applies.
	StartTime	*xsd:time*	0:1	Time at which time band starts at station.
	EndTime	*xsd:time*	0:1	Time at which time band ends at station.
	DayOffset	*DayOffsetType*	0:1	Day offset of end time from start time. Zero is same day.

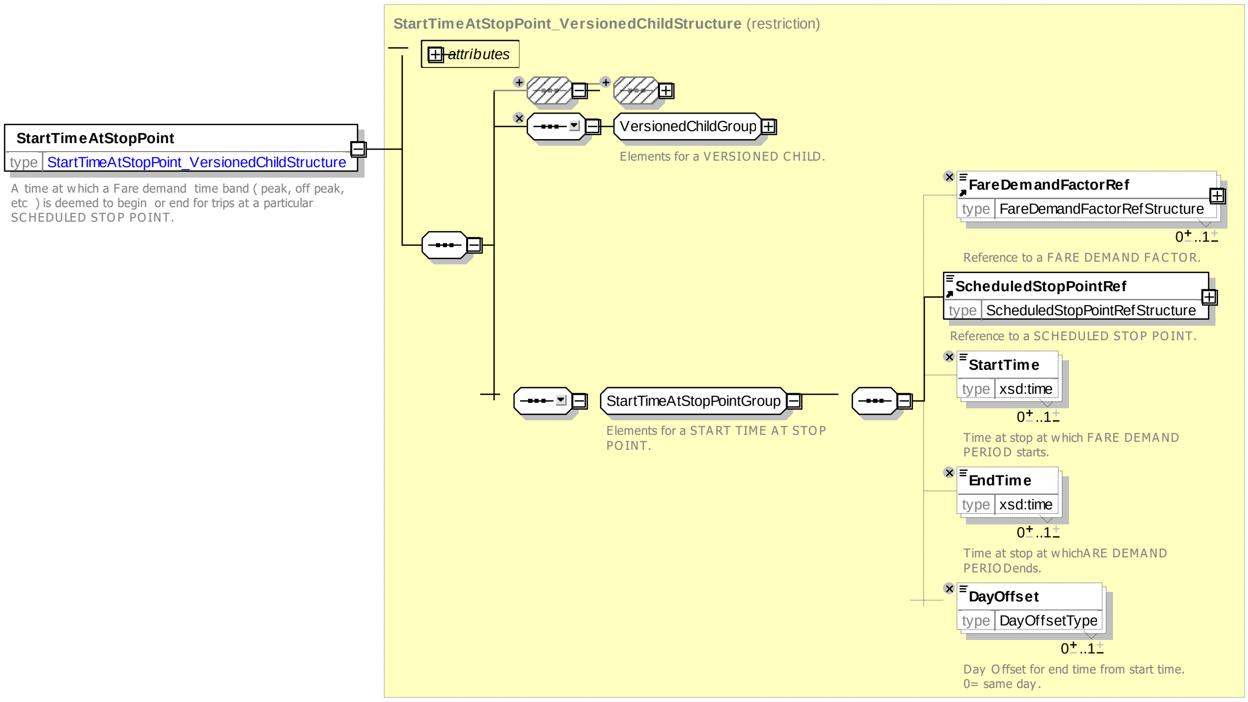


Figure 83 — *StartTimeAtStopPoint* — XSD

7.5.5.5.5 FareQuotaFactor – Model Element

A named set of parameters defining number of quota fares available. of a given denomination

Table 59 – *FareQuotaFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*QualityStructureFactor*	::>	FARE QUOTA FACTOR inherits from QUALITY STRUCTURE FACTOR.
«PK»	***id***	*FareQuotaFactorIdType*	1:1	Identifier of a FARE QUOTA FACTOR.
	NumberOfUnits	*xsd:integer*	0:*	Number of units available of product at a given price.

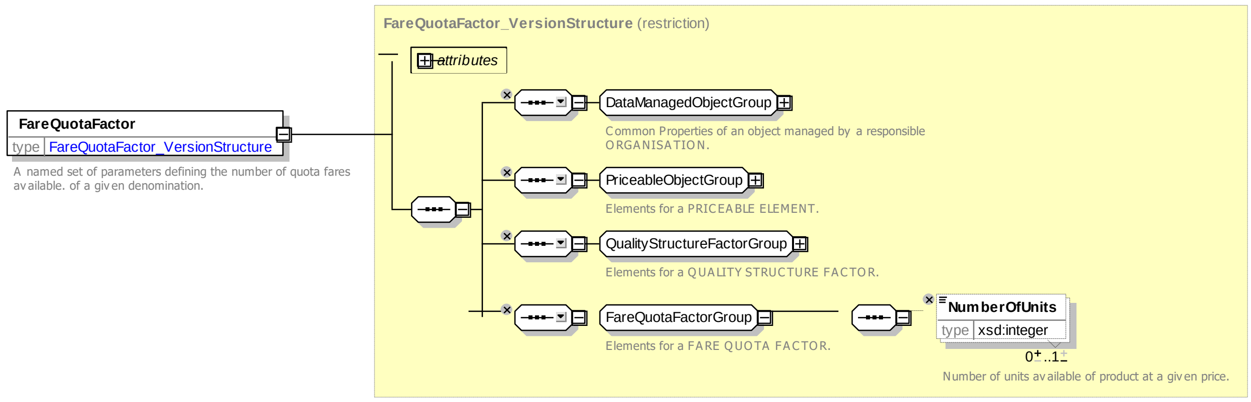


Figure 84 — *FareQuotaFactor* — XSD

7.5.5.6 Quality Fare Structure – XML examples

7.5.5.6.1 Fare Demand Factor: XML Example of Peak and Off-peak Fare Demand Factors

The following code fragment shows three FARE DEMAND FACTORs for any time, peak and off-peak travel. The off-peak travel defines separate start times for certain zones

For EXAMPLE:

```
    <qualityStructureFactors>
        <FareDemandFactor version="any" id="tfl:any_time">
            <Name>Anytimetravel </Name>
        </FareDemandFactor>
        <FareDemandFactor version="any" id="tfl:peak">
            <Name>Peak time travel </Name>
            <validityConditions>
                <AvailabilityConditionRef ref="tfl:Peak" version="any"/>
            </validityConditions>
        </FareDemandFactor>
        <FareDemandFactor version="any" id="tfl:offPeak">
            <Name>off peak time travel</Name>
            <Description>Has stop specific overrides
If you travel from a station north of Moor Park or Hatch End on a weekday after the times below,
your Oyster single fare will count towards the off-peak cap instead of the peak cap.

North of Moor Park
Station                       Touch in times
```

```
Chesham                    After 09:00
Amersham               After 09:10
Chalfont & Latimer         After 09:15
Chorleywood                After 09:15
Rickmansworth              After 09:20
                </Description>
            <validityConditions>
                <AvailabilityConditionRef ref="tfl:OffPeak" version="any"/>
            </validityConditions>
            <startTimesAtStopPoints>
                <StartTimeAtStopPoint version="any" id="tfl:Chesham">
                    <ScheduledStopPointRef ref="tfl:Chesham" version="any"/>
                    <StartTime>09:00:00</StartTime>
                </StartTimeAtStopPoint>
                <StartTimeAtStopPoint version="any" id="tfl:Amersham">
                    <ScheduledStopPointRef ref="tfl:Amersham" version="any"/>
                    <StartTime>09:10:00</StartTime>
                </StartTimeAtStopPoint>
                <StartTimeAtStopPoint version="any" id="tfl:Chalfont_and_Latimer">
                    <ScheduledStopPointRef ref="tfl:Chalfont_and_Latimer" version="any"/>
                    <StartTime>09:15:00</StartTime>
                </StartTimeAtStopPoint>
                <StartTimeAtStopPoint version="any" id="tfl:Chorleywood">
                    <ScheduledStopPointRef ref="tfl:Chorleywood" version="any"/>
                    <StartTime>09:15:00</StartTime>
                </StartTimeAtStopPoint>
                <StartTimeAtStopPoint version="any" id="tfl:Rickmansworth">
                    <ScheduledStopPointRef ref="tfl:Rickmansworth" version="any"/>
                    <StartTime>09:20:00</StartTime>
                </StartTimeAtStopPoint>
            </startTimesAtStopPoints>
        </FareDemandFactor>
    </qualityStructureFactors>
```

7.5.6 Fare Structure Element

7.5.6.1 Fare Structure Element – Conceptual MODEL

The FARE STRUCTURE ELEMENT MODEL describes the core elements – in particular, FARE STRUCTURE ELEMENT, FARE STRUCTURE ELEMENT IN SEQUENCE – of the fare structure. These can then be combined with other spatial, temporal and quality factors to specify the overall fare structure, as described later below.

A FARE STRUCTURE ELEMENT can be further related to VALIDABLE ELEMENTs and CONTROLLABLE ELEMENTs to describe the access rights to the element – see later below.

7.5.6.2 Fare Structure Elements

A fare structure consists of generic quantitative rules for the limitation of rights, allowing the calculation of prices (e.g. graduated fare structure, based on the number of fare zones crossed). These rules are applied to FARE STRUCTURE ELEMENTs.

A FARE STRUCTURE ELEMENT is defined as a sequence or a set of (one or several) CONTROLLABLE ELEMENTs, to which the same fare structure is applied.

This definition may be reformulated as follows: a FARE STRUCTURE ELEMENT is a sequence or a set of (one or several) of the most basic service elements, for all of which the values of the parameters characterizing it are constant.

For instance, if CONTROLLABLE ELEMENTs are rides on buses and if a zone-counting fare structure is applied, this structure will be applied either:

— if the fare system does not allow interchanges, to only one CONTROLLABLE ELEMENT, to which the FARE STRUCTURE ELEMENT will be identical;

— or, if interchanges are allowed, to a sequence of CONTROLLABLE ELEMENTs, this sequence building a FARE STRUCTURE ELEMENT.

If one of the fare structure rules changes during the consumption, another FARE STRUCTURE ELEMENT is defined.

In public transport, CONTROLLABLE ELEMENTs and FARE STRUCTURE ELEMENTs will often be merged into a single concept. However, the distinction is useful for instance in cases where free interchange is allowed, as described above. Typical examples of FARE STRUCTURE ELEMENTs are the following:

— a simple ride on a bus line. In such a case, the FARE STRUCTURE ELEMENT will be identical to the CONTROLLABLE ELEMENT;

— a trip on a PT network, with a price based on a zone-matrix system; as described above, such a FARE STRUCTURE ELEMENT will be composed of several CONTROLLABLE ELEMENTs if interchanges are allowed;

— a trip on a PT network, with a limited duration and flat fares;

— a ride on a train line, with a total price based on the number of fare sections crossed; such a FARE STRUCTURE ELEMENT will be composed of several CONTROLLABLE ELEMENTs for instance if the operator changes in between.

With respect to other services than public transport, a FARE STRUCTURE ELEMENT may be for instance an access to a car park, charged depending on the parking duration.

In other words: the concept FARE STRUCTURE ELEMENT is a service consumption right determined:

— by a set of quantitative (spatial, temporal) rules (i.e. determined by fare structure parameters, such as zones, sections, hours, etc.), called fare structure rules, allowing the price calculation;

— by a set of validity limiting parameters (validity parameters).

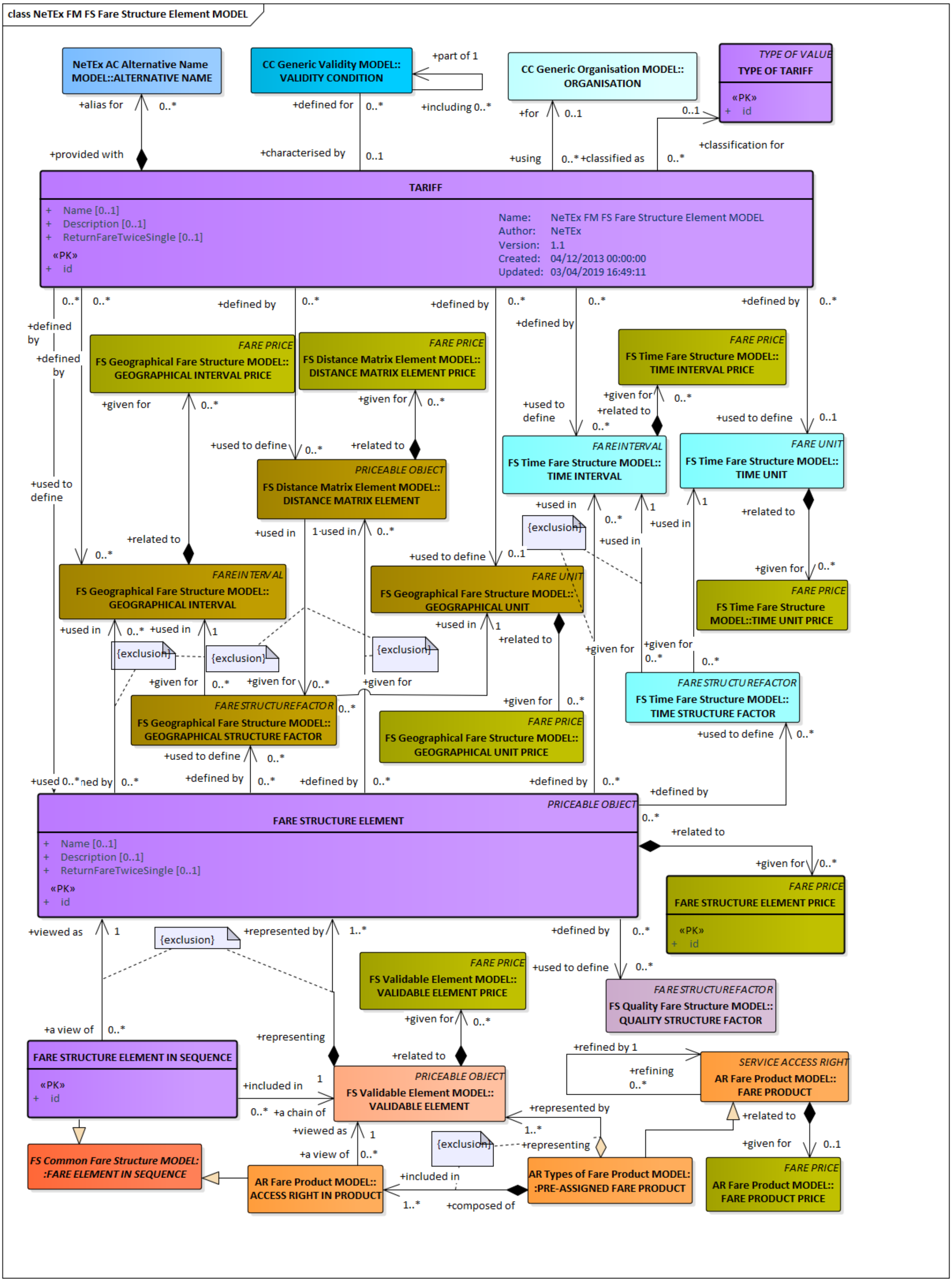


Figure 85 —Fare Structure Element– Conceptual MODEL (UML)

7.5.6.3 Tariff – Conceptual MODEL

The FARE STRUCTURE ELEMENT MODEL also describes TARIFFs – groupings of FARE STRUCTURE ELEMENTs subject to particular validity conditions.

In most cases, only one GEOGRAPHICAL (resp. TIME or QUALITY) STRUCTURE FACTOR is attached to each FARE STRUCTURE ELEMENT. In rare cases, different factors may apply to the same element, chosen by a rule depending on specific validity conditions. This is the case for instance when different fares are applied in summer than in other seasons. More simply, the fare structure may evolve and a version be replaced by another.

The entity TARIFF describes a VERSION of all parameters composing a particular fare structure. When applying fare structure rules, an algorithm will choose the parameters (e.g. a TIME INTERVAL) according to the valid TARIFF.

In real implementations, constraints should be set to ensure that, for a given TARIFF, only one factor of each type (e.g. GEOGRAPHICAL STRUCTURE FACTOR) is attached to any valid FARE STRUCTURE ELEMENT.

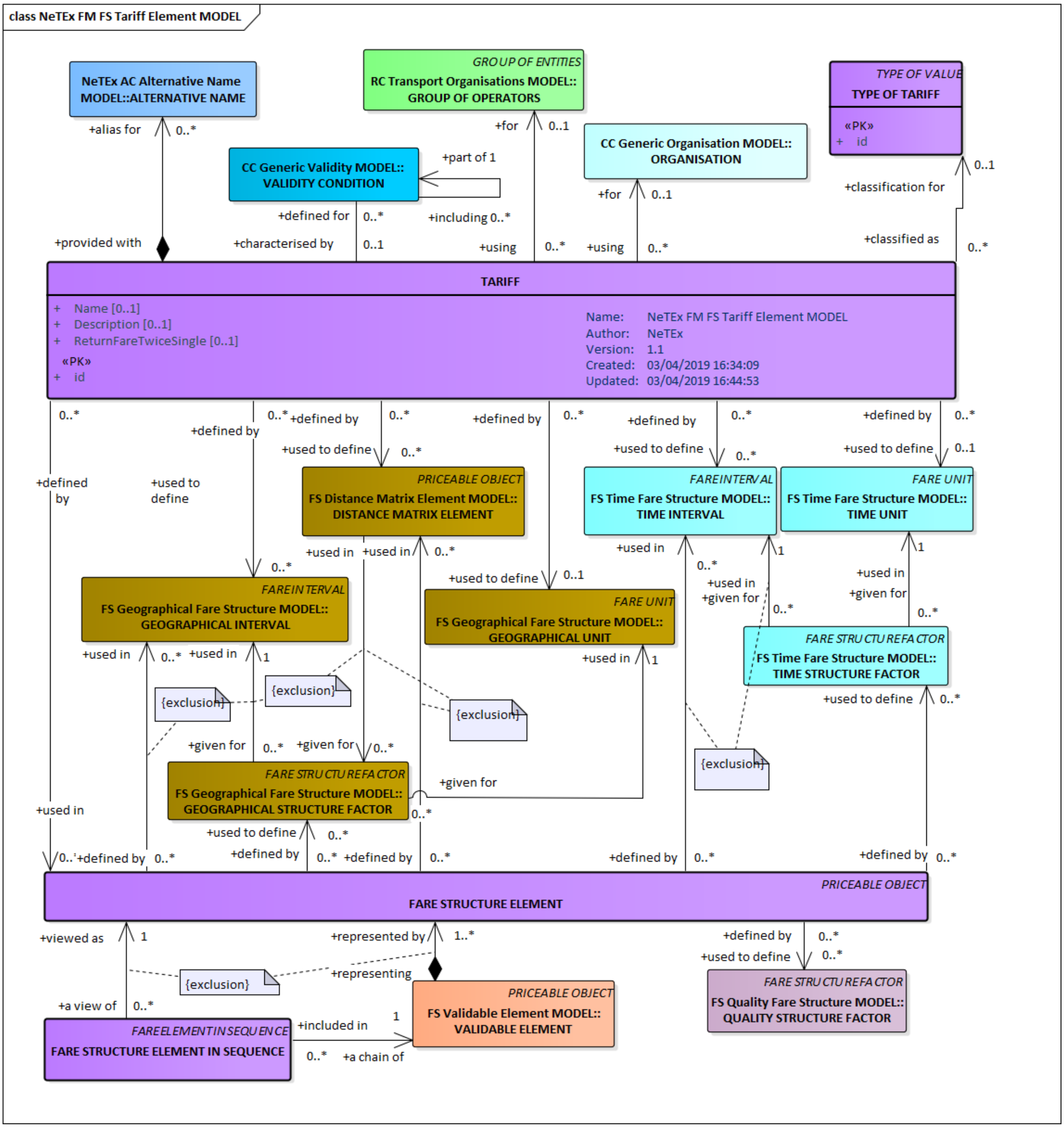


Figure 86 —Tariff Element– Conceptual MODEL (UML)

7.5.6.4 Fare Structure Element – Examples

The NeTEx fare structure elements can be used to represent many very different types of fare structure (graduated fares, flat fares etc). The following table introduces some of the more common types of fare structure found in public transport systems and shows the NeTEx elements that can be used to represent them.

Note that the FARE STRUCTURE MODEL elements describe only the fundamental basis of the fare structure; many other aspects (for example how rides can be combined, who can use a particular product, when they may travel etc) are described using access rights (see ACCESS RIGHT ASSIGNMENT PARAMETERs) and limitation parameters (see USAGE PARAMETERs below.). Similarly, consideration of which rights are purchased as a given product, such as single ticket, day pass or season ticket (FARE

PRODUCTs) and how the fare is packaged as a ticket (SALES OFFER PACKAGE, TRAVEL DOCUMENT) and distributed (DISTRIBUTION CHANNEL, FULFILMENT METHOD) are treated as separate concerns.

Note also that even if the fare structure is actually derived from some other underlying basis such as distance, stages or some other consideration, it is possible (by calculating all the individual fares) to present all fare structures as DISTANCE MATRIX ELEMENTs, that is, a matrix of point to point fares – see use cases for fare preparation.

Table 60 – Example – Fare Types and NeTEx Fare STRUCTURE ELEMENTs

Name	Description	Fare structure and limitation elements used	Real life examples
Network flat fare	Can make a single trip anywhere on the network regardless of the number of stops. (usually time limited).	VALIDABLE ELEMENT (Use ACCESS RIGHT ASSIGNMENT to assign to NETWORK or MODE. Use USAGE VALIDITY PERIOD to specify time limit. Use INTERCHANGING to limit number of transfers.)	TfL Bus fare, cash purchase. New York Metro ride.
Line flat fare	Can make a single ride anywhere on a single line of the network (usually time limited).	VALIDABLE ELEMENT (Use ACCESS RIGHT ASSIGNMENT to assign to LINE. Use USAGE VALIDITY PERIOD to specify time limit. Use INTERCHANGING to limit number of transfers.)	A lot of funicular or cable car
Point to point fare	Can travel between two designated stops.	VALIDABLE ELEMENT + DISTANCE MATRIX ELEMENT (Matrix element from SCHEDULED STOP POINT to SCHEDULED STOP POINT. Use SERIES CONSTRAINT to restrict to certain routings over network)	Standard kilometre based TAP/ TSI NRT rail fare
Point to point distance-based fare	Fare is charged between two stops according to "distance" (may be kilometres or number of arbitrary units)	VALIDABLE ELEMENT + DISTANCE MATRIX ELEMENT and/or + GEOGRAPHICAL FARE FACTOR + GEOGRAPHICAL INTERVAL (Matrix element from SCHEDULED STOP POINT to SCHEDULED STOP POINT)	Traditional rail fares.
Zonal fares	Fare is charged for use of a specified number of zones (but all zones are considered similar)	VALIDABLE ELEMENT + FARE STRUCTURE ELEMENT GEOGRAPHICAL FARE FACTOR + GEOGRAPHICAL INTERVAL. (Use zone for type of interval)	
Zone to Zone fare	Fare is charged for use of a zone and for each specific zone to zone combination.	VALIDABLE ELEMENT + DISTANCE MATRIX ELEMENT (Matrix element from TARIFF ZONE to TARIFF ZONE)	London underground zone fares.
Zone Sequence fare	Fare is charged for use of a zone and for use of a sequence of zones.	VALIDABLE ELEMENT + FARE STRUCTURE ELEMENT + FARE STRUCTURE ELEMENT IN SEQUENCE + FARE ZONE	Paris season tickets

Honeycomb zones		VALIDABLE ELEMENT + FARE STRUCTURE ELEMENT + FARE STRUCTURE ELEMENT IN SEQUENCE + FARE ZONE	Number of German cities
Stage Count Fare	Fare is charged for according to the number of fare stages passed.	VALIDABLE ELEMENT + DISTANCE MATRIX ELEMENT + GEOGRAPHICAL FARE FACTOR + GEOGRAPHICAL INTERVAL	Some long distance bus lines in Ile-de-France
Stage Fare -	Fare is charged for according to the specific sections reached	VALIDABLE ELEMENT + DISTANCE MATRIX ELEMENT + ZONE SECTION	
Trip Sequence fare	Fare is charged for use of a specified sequence of trip types.	VALIDABLE ELEMENT + FARE STRUCTURE ELEMENT + FARE STRUCTURE ELEMENT IN SEQUENCE	
[OTHER MORE COMPLEX ? TO DO			

7.5.6.4.1 Example – Zonal fare

The following map shows the London underground map, a classic example of a zone to zone fare system. The network is split into 9 zones

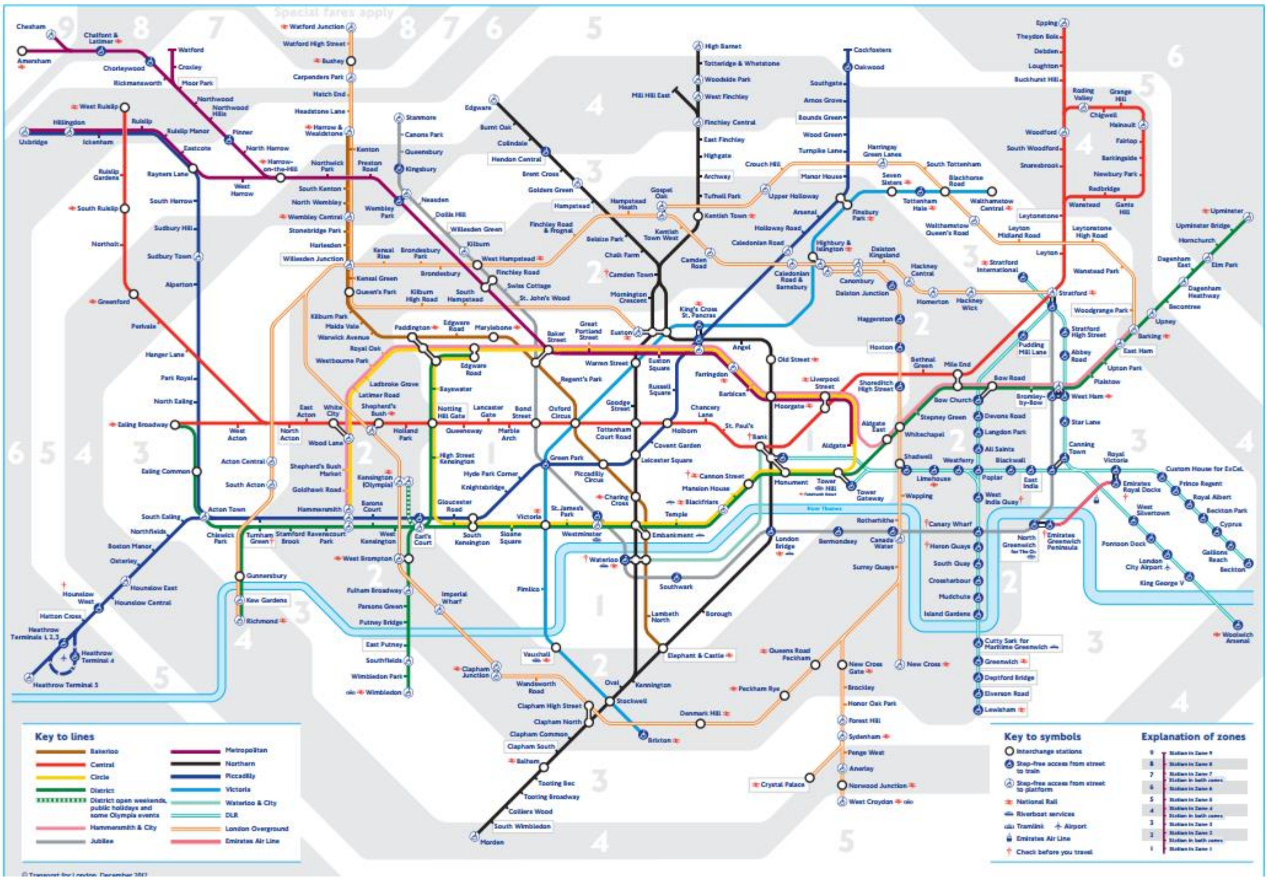


Figure 87 —Example Zone to Zone Fares – London Underground Map (EXM)

The following indicative price table shows that Metro and Rail fare sin London are charged on a zone to zone basis. Bus and Rail fares in contrast, are a flat fare.

Oyster pay as you go price guide

Tube, DLR and London Overground

Journeys	Peak	Off Peak
Zone 1 Only	£2.10	£2.10
Zones 1-2	£2.80	£2.10
Zones 1-4	£3.80	£2.70
Zones 1-6	£5.00	£3.00

Most National Rail Services

Journeys	Peak	Off Peak
Zone 1 Only	£2.20	£1.70
Zones 1-2	£2.40	£1.90
Zones 1-4	£3.60	£2.50
Zones 1-6	£5.70	£3.50

Bus and tram

Any Journey	£1.40

For more information visit
tfl.gov.uk/tickets

Figure 88 —Example Zone to Zone Fares – London Underground Prices (EXM)

7.5.6.5 Fare Structure Element – Physical model

The following figure shows the basic physical model for FARE STRUCTURE ELEMENTs and TARIFFs.

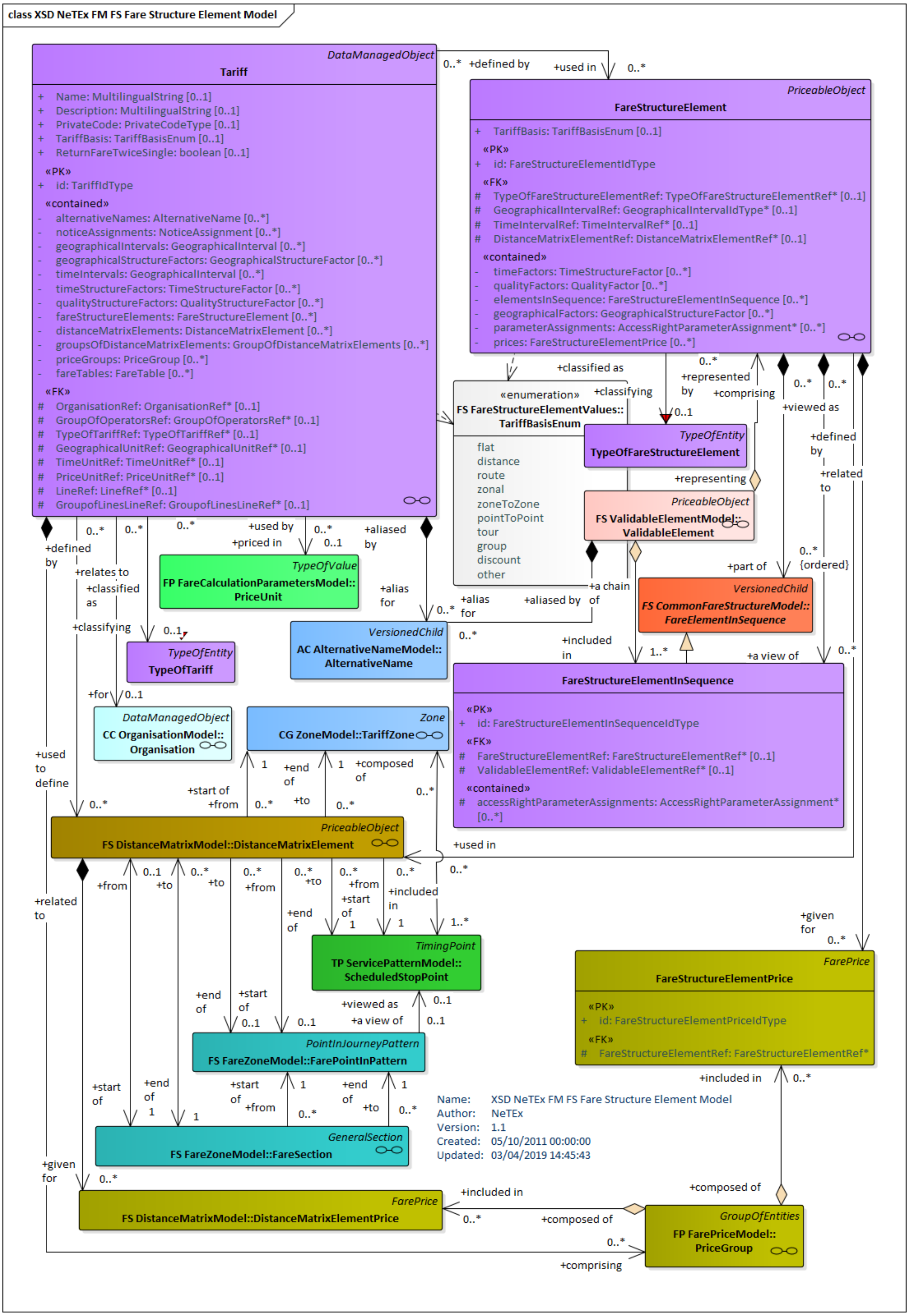


Figure 89 —Fare Structure Element– Physical Model (UML)

7.5.6.6 Fare Structure Element – Attributes and XSD

7.5.6.6.1.1 FareStructureElement – Model Element

A sequence or set of CONTROLLABLE ELEMENTs to which rules for limitation of access rights and calculation of prices (fare structure) are applied.

Table 61 – *FareStructureElement* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*PriceableObject*</u>	::>	FARE STRUCTURE ELEMENT inherits from PRICEABLE OBJECT.
«PK»	***id***	*FareStructureElement-IdType*	1:1	Identifier of FARE STRUCTURE ELEMENT.
«enum»	***TariffBasis***	*TariffBasisEnum*	0:1	TARIFF BASIS to be used for the element. See allowed values below.
XGRP	***FareStructure-ElementFactor-Group***	<u>***xmlGroup***</u>	1:1	FARE STRUCTURE FACTORs associated with the FARE STRUCTURE ELEMENT.
XGRP	***FareStructure-Component-Group***	<u>***xmlGroup***</u>	1:1	FARE STRUCTURE components associated with the FARE STRUCTURE ELEMENT.
«cntd»	***prices***	<u>*FareStructureElement-Price*</u>	0:*	Prices for the FARE STRUCTURE ELEMENT.

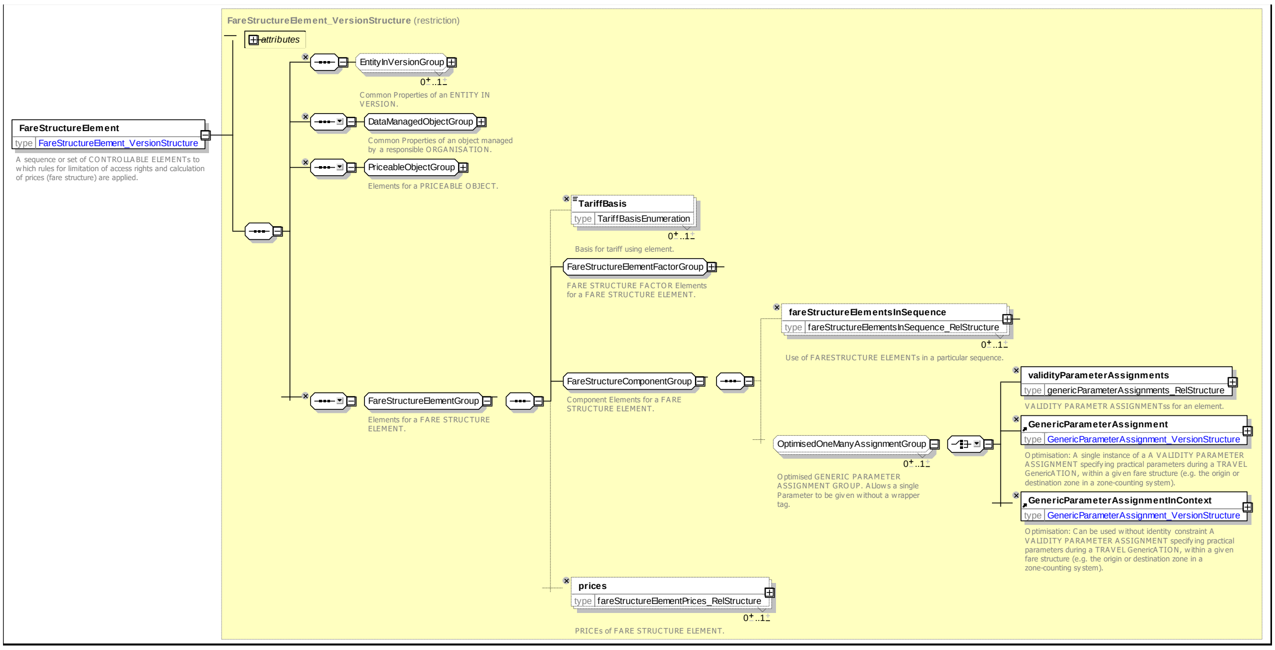


Figure 90 — *FareStructureElement* — XSD

7.5.6.6.1.2 FareStructureElementFactorGroup – Group

The ***FareStructureElementFactorGroup*** defines the various structure factors which may apply to a FARE STRUCTURE ELEMENT.

Table 62 – *FareStructureElementFactorGroup* – Group

Classifi-cation	Name		Type	Cardin ality	Description
			CHOICE		
«FK»	***a***	***Geographical-IntervalRef***	*GeographicalIntervalRef*	0:1	Reference to GEOGRAPHICAL INTERVAL associated with FARE STRUCTURE ELEMENT.
«cntd»	***b***	***geographical-Intervals***	*GeographicalInterval* | *GeographicalIntervalRef*	0:*	GEOGRAPHICAL INTERVALs associated with FARE STRUCTURE ELEMENT.
«cntd»	***c***	***geographical-Structure-Factors***	*GeographicalStructureFactor* | *GeographicalStructure-FactorRef*	0:*	GEOGRAPHICAL STRUCTURE FACTORs associated with the FARE STRUCTURE ELEMENT.
			Choice		
«FK»	***a***	***TimeInterval-Ref***	*TimeIntervalRef*	0:1	Reference to TIME INTERVAL associated with FARE STRUCTURE ELEMENT.
«cntd»	***b***	***timeIntervals***	*TimeInterval* | *TimeIntervalRef*	0:*	TIME STRUCTURE INTERVALs associated with the FARE STRUCTURE ELEMENT.
«cntd»	***c***	***timeStructure-Factors***	*TimeStructureFactor* | *TimeStructureFactorRef*	0:*	TIME STRUCTURE FACTORs associated with the FARE STRUCTURE ELEMENT.
			CHOICE		
«FK»	***a***	***Quality-Structure-FactorRef***	*QualityStructureFactor Ref*	0:1	Reference to QUALITY STRUCTURE FACTOR associated with FARE STRUCTURE ELEMENT.
«cntd»	***b***	***quality-Structure-Factors***	*QualityStructureFactor* | *QualityStructureFactor*	0:*	QUALITY STRUCTURE FACTORs associated with the FARE STRUCTURE ELEMENT.
			Choice		
«FK»	***a***	***Distance-Matrix-ElementRef***	*DistanceMatrixElementRef*	0:1	Reference to DISTANCE MATRIX ELEMENT associated with FARE STRUCTURE ELEMENT.
«FK»	***b***	***distanceMatrix Elements***	*DistanceMatrixElement* | *DistanceMatrixElementRef*	0:*	DISTANCE MATRIX ELEMENTs associated with FARE STRUCTURE ELEMENT.
«FK»	***c***	***GroupOf-Distance-Matrix-ElementsRef***	*GroupOfDistanceMatrix-ElementsRef*	0:1	Reference to GROUP OF DISTANCE MATRIX ELEMENTs associated with FARE STRUCTURE ELEMENT.

«cntd»	*d*	***GroupOfDistanceMatrix-Elements***	*GroupOfDistanceMatrix-Elements*	0:1	GROUP OF DISTANCE MATRIX ELEMENTs associated with FARE STRUCTURE ELEMENT.

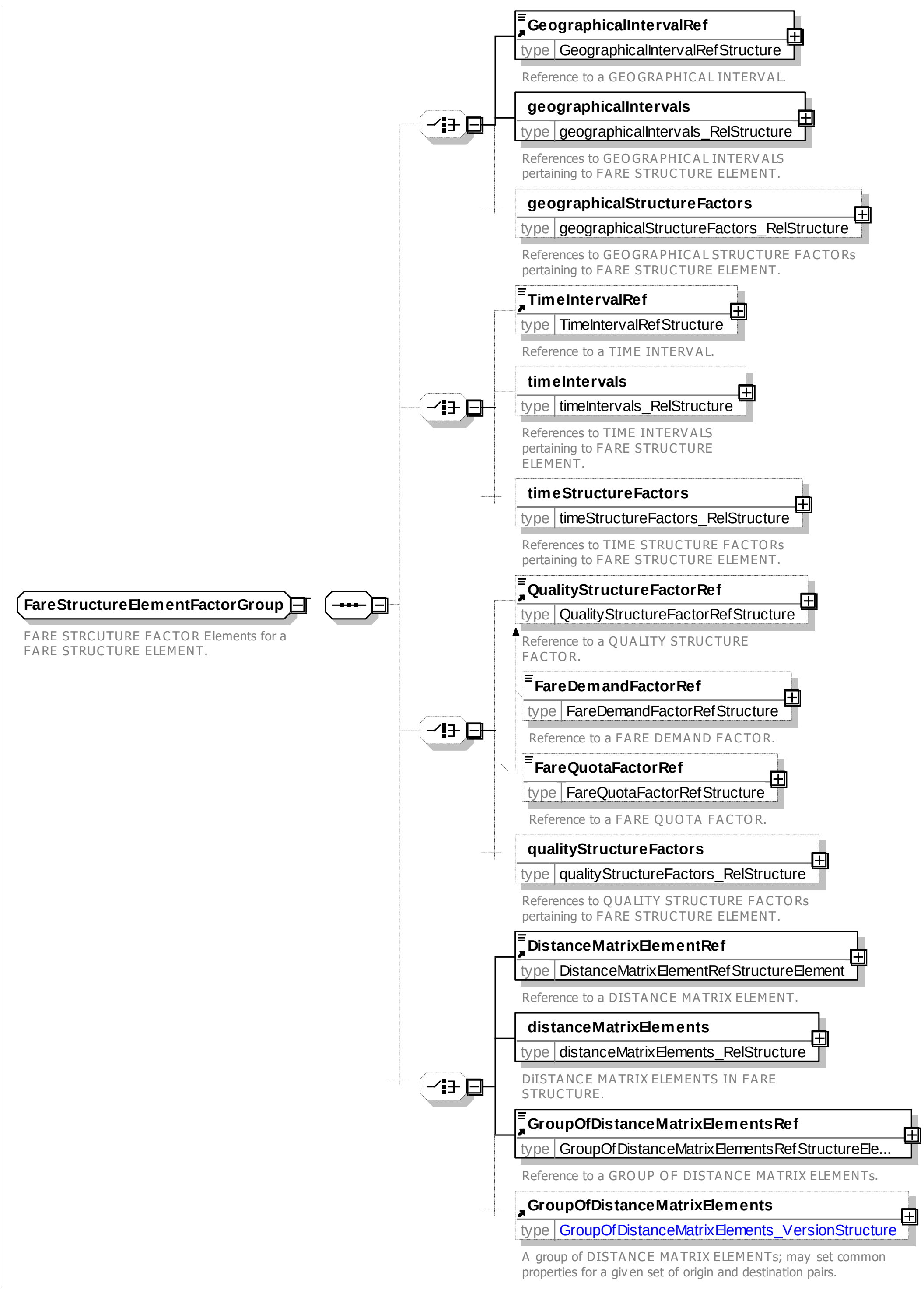


Figure 91 — *FareStructureElementFactorGroup* — XSD

7.5.6.6.1.3 FareStructureComponentGroup – Group

The ***FareStructureComponentGroup*** defines any component FARE STRUCTURE ELEMENTs IN SEQUENCE and ACCESS RIGHT PARAMETER ASSIGNMENTs that make up the FARE STRUCTURE ELEMENT.

Table 63 – *FareStructureComponentGroup* – Group

Classifi-cation	Name		Type	Cardinality	Description
«cntd»	***fareStructure-ElementsIn-Sequence***		*FareStructureElement-InSequence | Controllable-ElementInSequence*	0:*	Child FARE STRUCTURE ELEMENTS in SEQUENCE making up the FARE STRUCTURE ELEMENT.
			CHOICE		Either multiple parameters wrapped in a tag, or a single parameter (an optimisation).
«cntd»	***a***	***validity-Parameter-Assignments***	*AccessRightParameter-Assignment*	0:*	GENERIC PARAMETER ASSIGNMENTs associated with the FARE STRUCTURE ELEMENT.
«cntd»	***b***	***Generic-Parameter-Assignment***	*GenericParameter-Assignment*	0:1	A single GENERIC PARAMETER ASSIGNMENT associated with the FARE STRUCTURE ELEMENT.
«cntd»	***c***	***Generic-Parameter-AssignmentIn Context***	*GenericParameter-Assignment*	0:1	A single GENERIC PARAMETER ASSIGNMENT associated with the FARE STRUCTURE ELEMENT. No ID needs to be given – will be inferred from the assignment values. (OPTIMISATION).

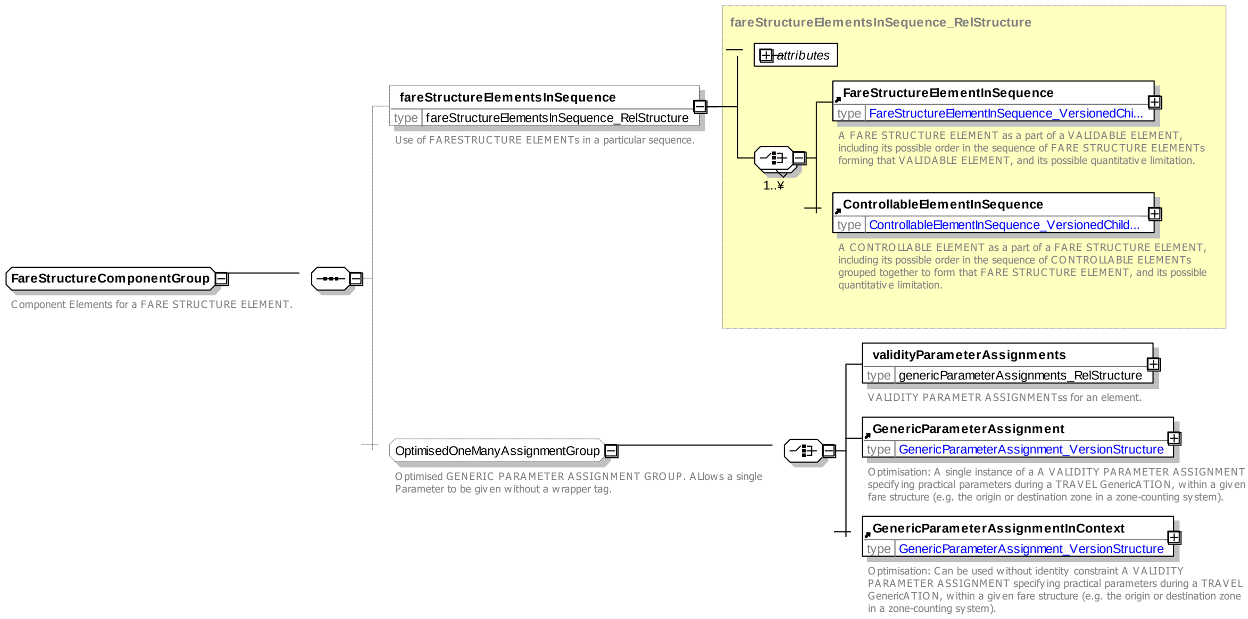


Figure 92 — *FareStructureComponentGroup* — XSD

7.5.6.6.2 FareStructureElementInSequence – Model Element

A FARE STRUCTURE ELEMENT as a part of a VALIDABLE ELEMENT, including its possible order in the sequence of FARE STRUCTURE ELEMENTs forming that VALIDABLE ELEMENT, and its possible quantitative limitation.

Table 64 – *FareStructureElementInSequence* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*FareElementInSequence*	::>	FARE STRUCTURE ELEMENT IN SEQUENCE inherits from FARE ELEMENT IN SEQUENCE.
«PK»	***id***	*FareStructureElement-InSequenceIdType*	1:1	Identifier of FARE STRUCTURE ELEMENT IN SEQUENCE.
«FK»	***FareStructure-ElementRef***	*FareStructureElementRef*	0:1	Reference to a FARE STRUCTURE ELEMENT.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	0:1	Reference to a VALIDABLE ELEMENT.
«cntd»	***validity-Parameter-Assignments***	*ValidityParameterAssignment*	0:*	VALIDITY PARAMETER ASSIGNMENTs associated with the ELEMENT IN SEQUENCE.

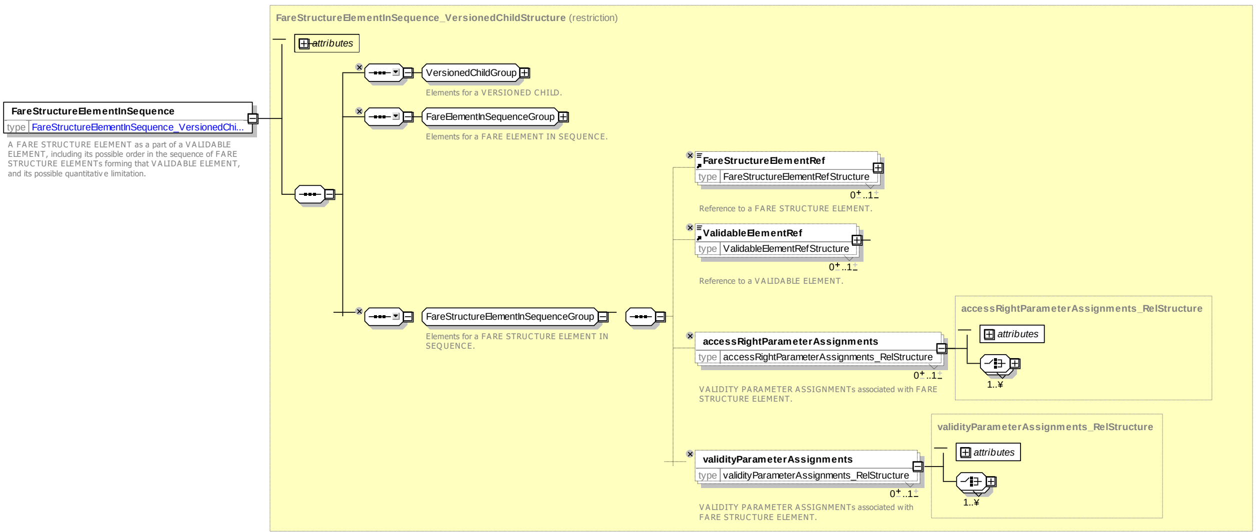


Figure 93 — *FareStructureElementInSequence* — XSD

7.5.6.6.3 FareStructureElementPrice – Model Element

A set of all possible price features of a FARE STRUCTURE ELEMENT: default total price, discount in value or percentage etc.

Table 65 – *FareStructureElementPrice* – Element

Classifi-cation	Name	Type	Cardin ality	Description

::>	::>	*FarePrice*	::>	FARE STRUCTURE ELEMENT PRICE inherits from FARE PRICE.
«PK»	***id***	*FareStructureElement-PriceIdType*	1:1	Identifier of FARE STRUCTURE ELEMENT PRICE.
«FK»	***FareStructure-ElementRef***	*FareStructureElementRef*	0:1	Reference to a FARE STRUCTURE ELEMENT for which this is the price. If not given by context, must be specified.

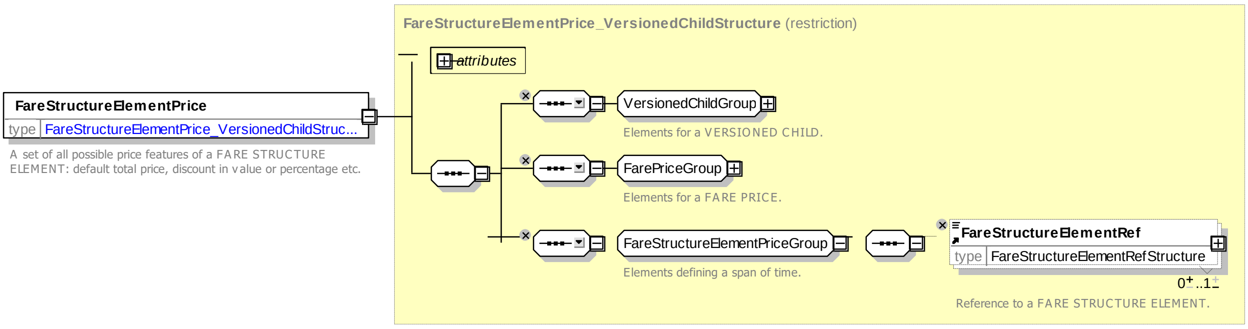


Figure 94 — *FareStructureElementPrice* — XSD

7.5.6.6.4 TypeOfFareStructureElement– Model Element

A classification of a FARE STRUCTURE ELEMENT.

Table 66 – 7.5.2.4.1 TypeOfFareStructureElement– – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*TypeOfEntity*</u>	::>	TYPE OF FARE STRUCTURE ELEMENT. Inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*TypeOfFareStructure-ElementIdType*	1:1	Identifier of TYPE OF FARE STRUCTURE ELEMENT.

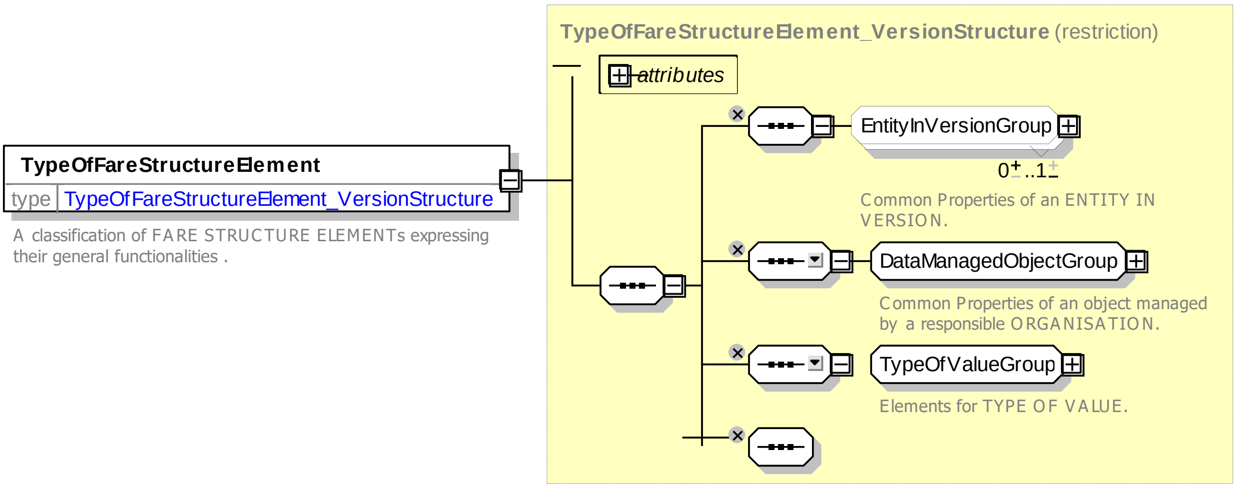


Figure 95 — *TypeOfFareStructureElement*— XSD

7.5.6.6.5 Tariff

A particular tariff, described by a combination of parameters.

Table 67 – *Tariff* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*DataManagedObject*	::>	TARIFF inherits from DATA MANAGED OBJECT. See NeTEx Part1.
«PK»	***id***	*TariffIdType*	1:1	Identifier of TARIFF.
XGRP	***TariffDescriptionGroup***	***xmlGroup***	0:1	Descriptive elements for TARIFF.
	PrivateCode	*PrivateCodeType*	0:1	Alternative identifier of an entity; can be used to associate with legacy systems.
XGRP	***TariffOrganisationGroup***	***xmlGroup***	0:1	Elements describing ORGANISATIONs operating TARIFF.
XGRP	***Tariff¬Applicability¬Group***	***xmlGroup***	0:1	Elements describing LINEs operating TARIFF.
XGRP	***TariffCalculationGroup***	***xmlGroup***	0:1	Elements describing calculation parameters for TARIFF.
XGRP	***TariffGeographicalGroup***	***xmlGroup***	0:1	Elements describing geographical distance components of TARIFF.
XGRP	***TariffTimeGroup***	***xmlGroup***	0:1	Elements describing time components of TARIFF.
XGRP	***TariffQualityGroup***	***xmlGroup***	0:1	Elements describing quality components of TARIFF.
XGRP	***TariffStructureGroup***	***xmlGroup***	0:1	Elements describing spatial components of TARIFF.
XGRP	***TariffPricesGroup***	***xmlGroup***	0:1	TARIFF PRICE elements for TARIFF.

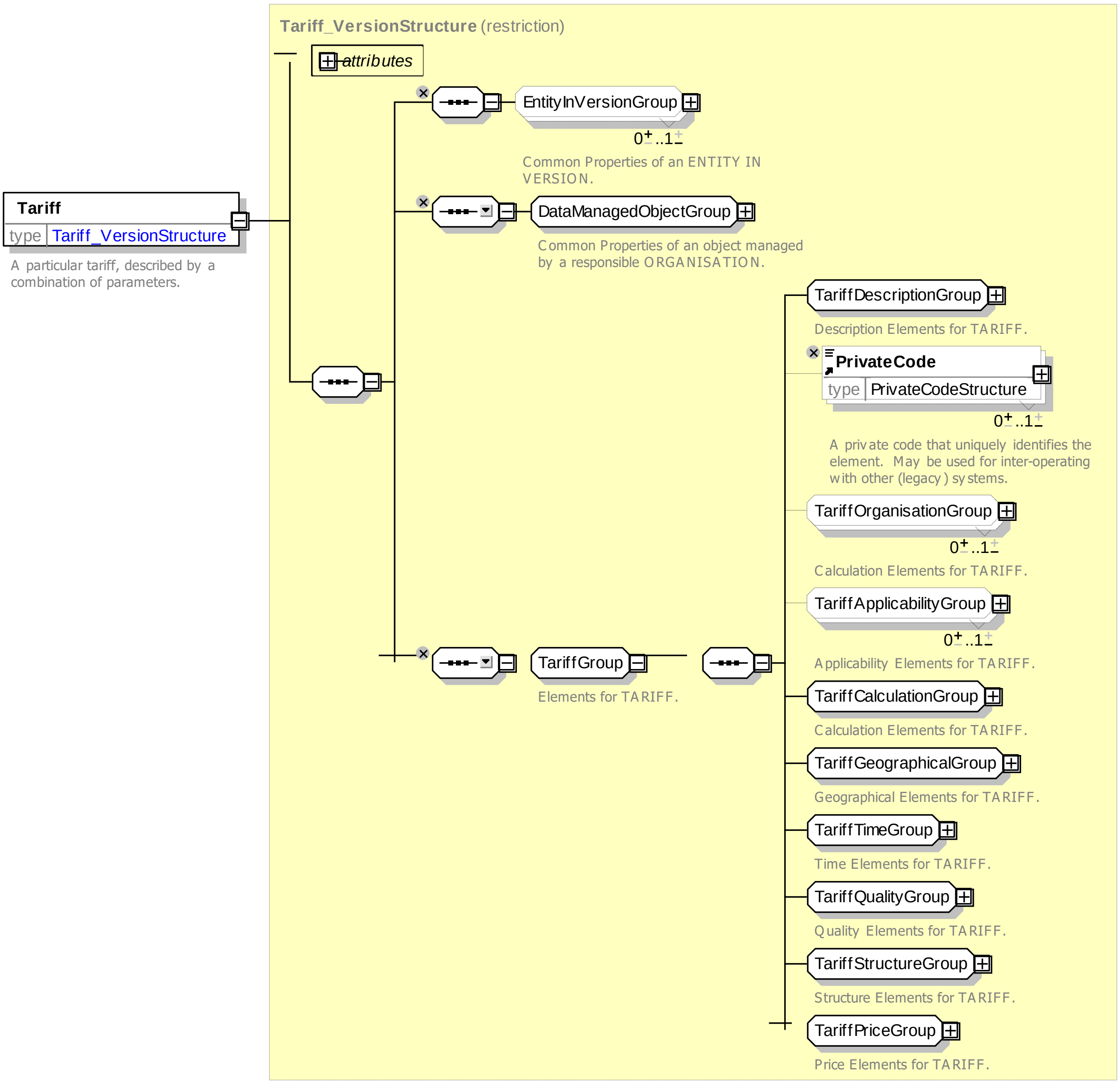


Figure 96 — *Tariff* — XSD

7.5.6.6.5.1 *TariffDescriptionGroup* – Group

The ***TariffDescriptionGroup*** describes the TARIFF.

Table 68 – *TariffDescriptionGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
	Name	*MultilingualString*	0:1	Name of TARIFF.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for TARIFF.
	Description	*MultilingualString*	0:1	Description of TARIFF.
«cntd»	***noticeAssignments***	*NoticeAssignment*	0:*	NOTICE ASSIGNMENTs for TARIFF.

«cntd»	***documentLinks***	*InfoLink*	0:*	Links for documents associated with TARIFF.

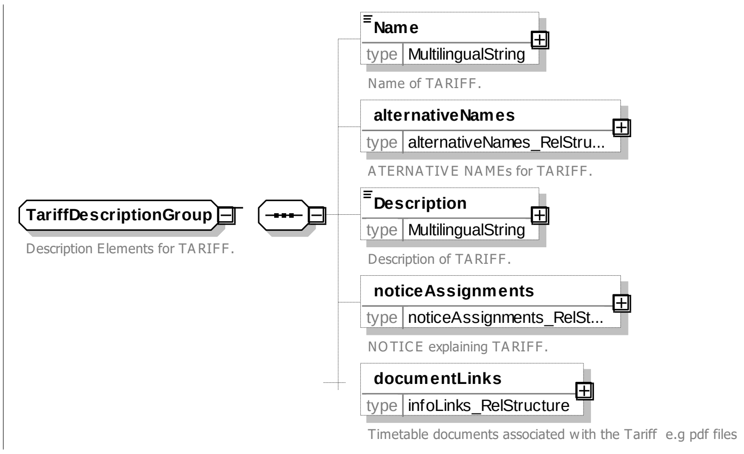


Figure 97 — *TariffDescriptionGroup* — XSD

7.5.6.6.5.2 TariffOrganisationGroup – Group

The ***TariffOrganisationGroup*** defines the ORGANISATIONs which provide the TARIFF.

Table 69 – *TariffOrganisationGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***OrganisationRef***	*(OrganisationRef)*	0:1	ORGANISATION to which TARIFF applies.
«FK»	***GroupOf-OrganisationsRef***	*GroupOf-OrganisationsRef*	0:1	GROUP OF ORGANISATIONs to which TARIFF applies.

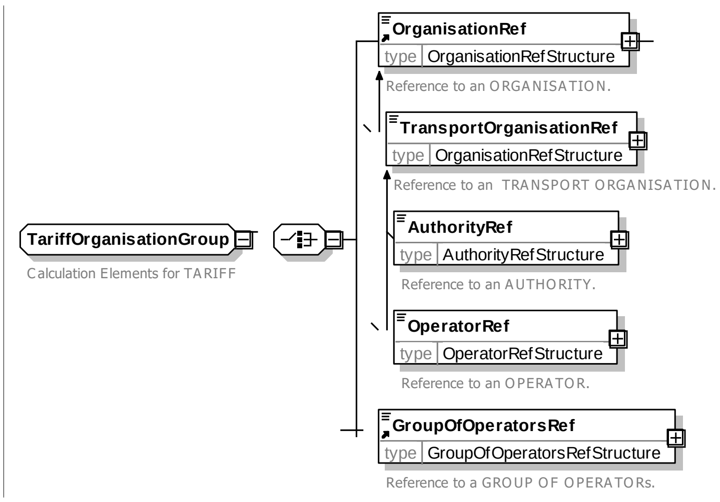


Figure 98 — *TariffOrganisationGroup* — XSD

7.5.6.6.5.3 TariffApplicabilityGroup – Group

The *TariffApplicabilityGroup* defines the LINEs which operate the TARIFF.

Table 70 – *TariffApplicabilityGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***LineRef***	*LineRef*	0:1	LINE to which TARIFF applies.
«FK»	***GroupOfLinesRef***	*GroupOfLinesRef*	0:1	GROUP OF LINEs to which TARIFF applies.

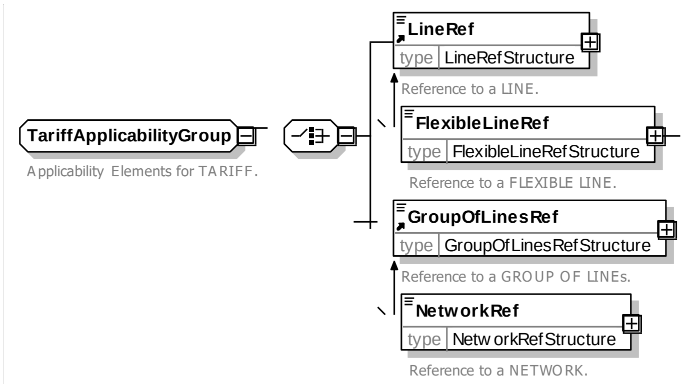


Figure 99 — *TariffApplicabilityGroup* — XSD

7.5.6.6.5.4 TariffCalculationGroup – Group

The ***TariffCalculationGroup*** defines parameters describing the basis for computing the TARIFF. May include classification with an arbitrary TYPE OF TARIFF.

Table 71 – *TariffCalculationGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***TypeOfTariffRef***	*TypeOfTariffRef*	0:1	Reference to a TYPE OF TARIFF.
«enum»	***TariffBasis***	*TariffBasisEnum*	0:1	Classification of how Tariff is priced. See allowed values below
	ReturnFare-TwiceSingle	*xsd:boolean*	0:1	Whether the return ticket is the double as the fare for a single ticket.

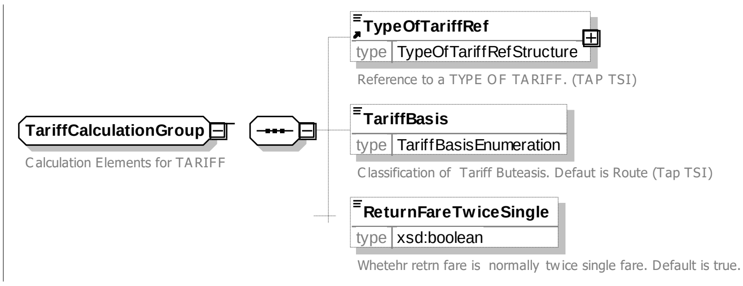


Figure 100 — *TariffCalculationGroup* — XSD

7.5.6.6.5.5 TariffGeographicalGroup – Group

The ***TariffGeographicalGroup*** defines the geographical fare structure elements underlying the TARIFF.

Table 72 – *TariffGeographicalGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***Geographical-UnitRef***	*GeographicalUnitRef*	0:1	Reference to GEOGRAPHICAL UNIT for TARIFF.
«cntd»	***geographical-Intervals***	*GeographicalInterval*	0:*	GEOGRAPHICAL INTERVALs associated with TARIFF.
«cntd»	***geographical-StructureFactors***	*Geographical-StructureFactor*	0:*	GEOGRAPHICAL STRUCTURE FACTORs associated with TARIFF.

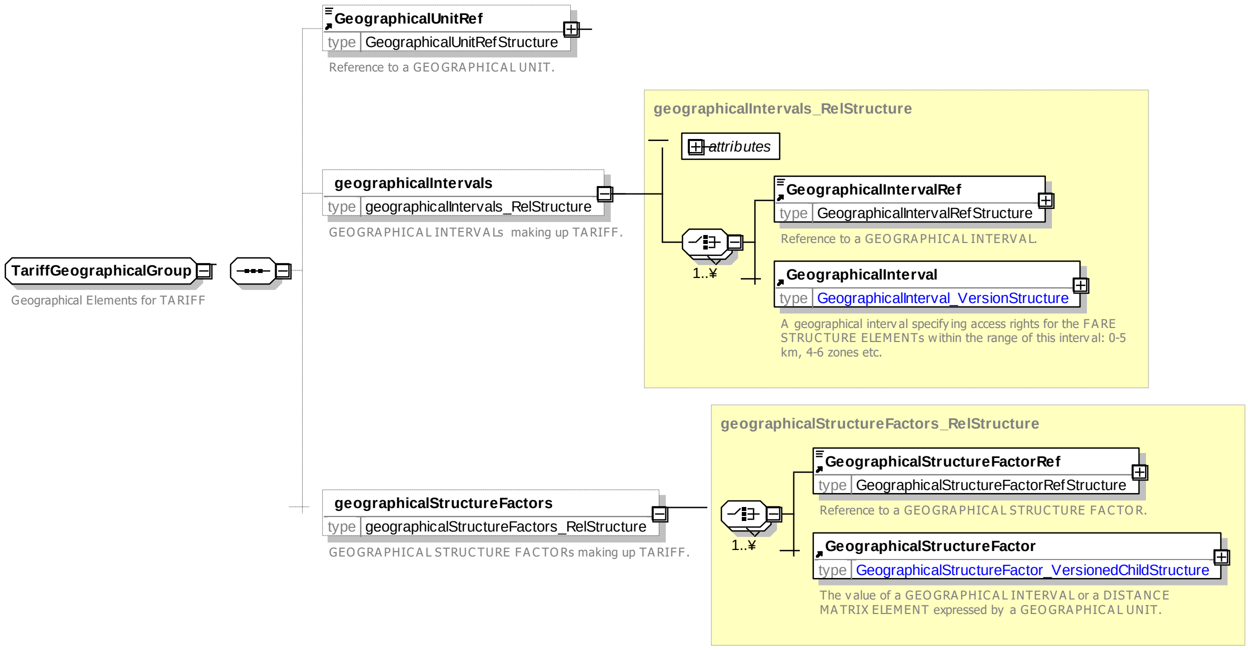


Figure 101 — *TariffGeographicalGroup* — XSD

7.5.6.6.5.6 TariffTimeGroup – Group

The ***TariffTimeGroup*** defines the time related fare structure elements underlying the TARIFF.

Table 73 – *TariffTimeGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***TimeUnitRef***	*TimeUnitRef*	0:1	Reference to TIME UNIT for TARIFF.
«cntd»	***timeIntervals***	*TimeInterval*	0:*	TIME INTERVALs associated with TARIFF.

«cntd»	***timeStructure-Factors***	*TimeStructureFactor*	0:*	TIME STRUCTURE FACTORs associated with TARIFF.

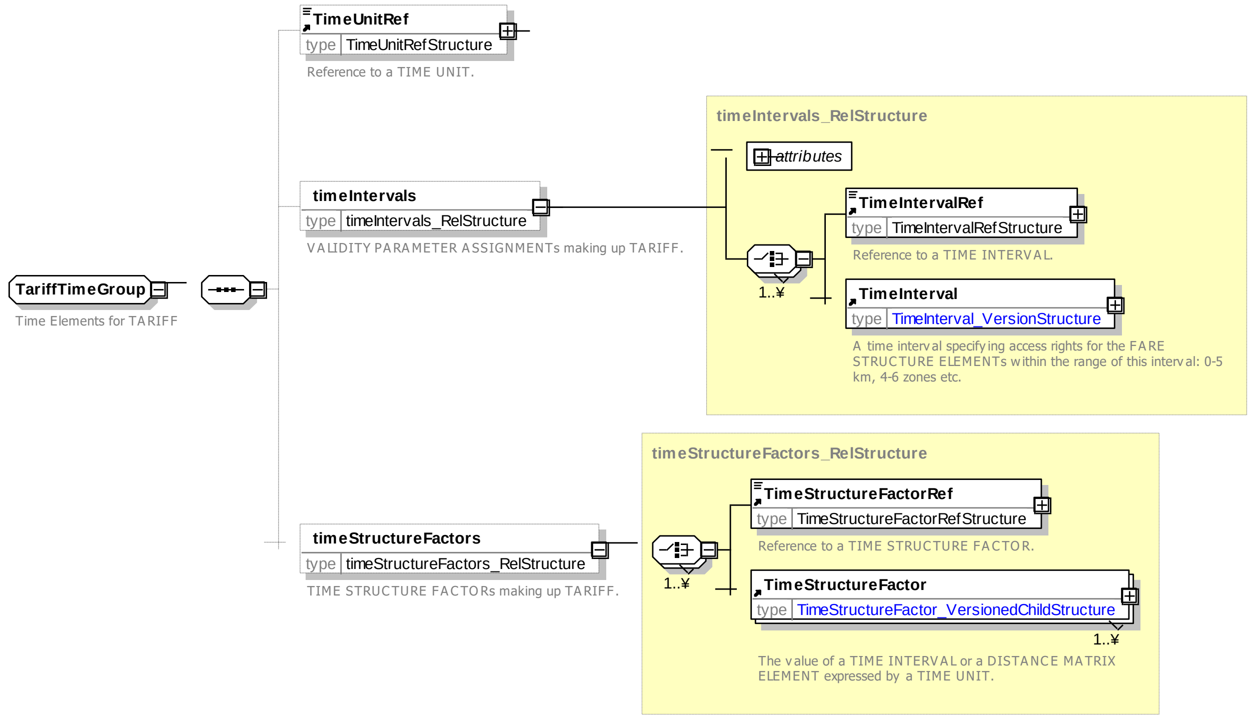


Figure 102 — *TariffTimeGroup* — XSD

7.5.6.6.5.7 TariffQualityGroup – Group

The ***TariffQualityGroup*** defines the quality related fare structure elements underlying the TARIFF.

Table 74 – *TariffQualityGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«cntd»	***qualityStructure-Factors***	*QualityStructureFactor*	0:*	QUALITY STRUCTURE FACTORs associated with TARIFF.

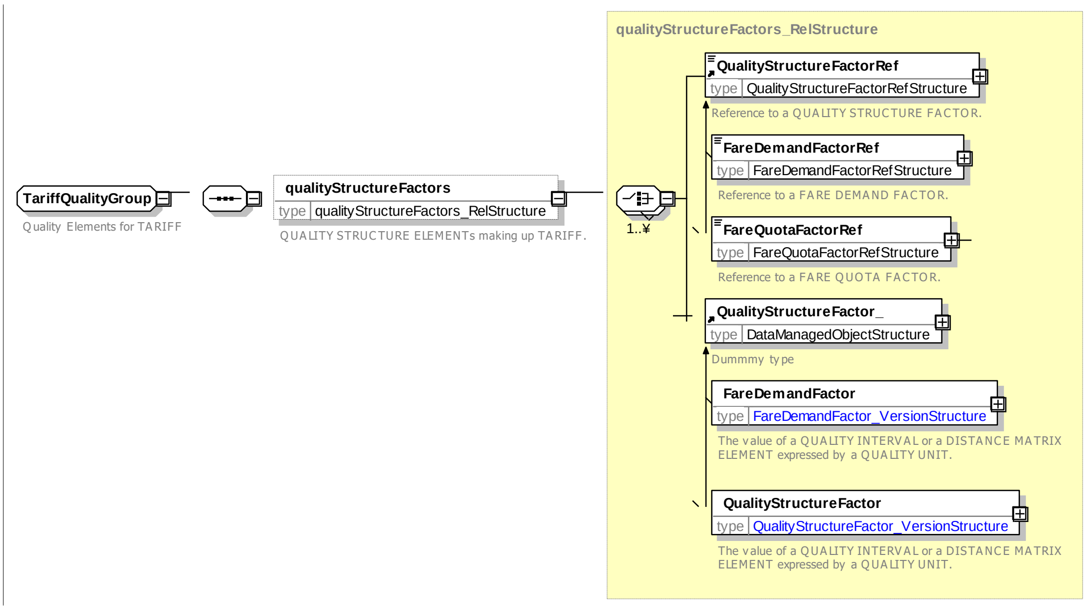


Figure 103 — *TariffQualityGroup* — XSD

7.5.6.6.5.8 TariffStructureGroup – Group

The ***TariffStructureGroup*** defines the fare structure elements comprising the TARIFF.

Table 75 – *TariffStructureGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«cntd»	***fareStructure-Elements***	*FareStructureElement*	0:*	FARE STRUCTURE ELEMENTs associated with TARIFF.
«cntd»	***distanceMatrix-Elements***	*DistanceMatrixElement*	0:*	DISTANCE MATRIX ELEMENTs associated with TARIFF.
«cntd»	***groupsOfDistance MatrixElements***	*GroupOfDistance-MatrixElements*	0:*	GROUPs OF DISTANCE MATRIX ELEMENTS associated with TARIFF.

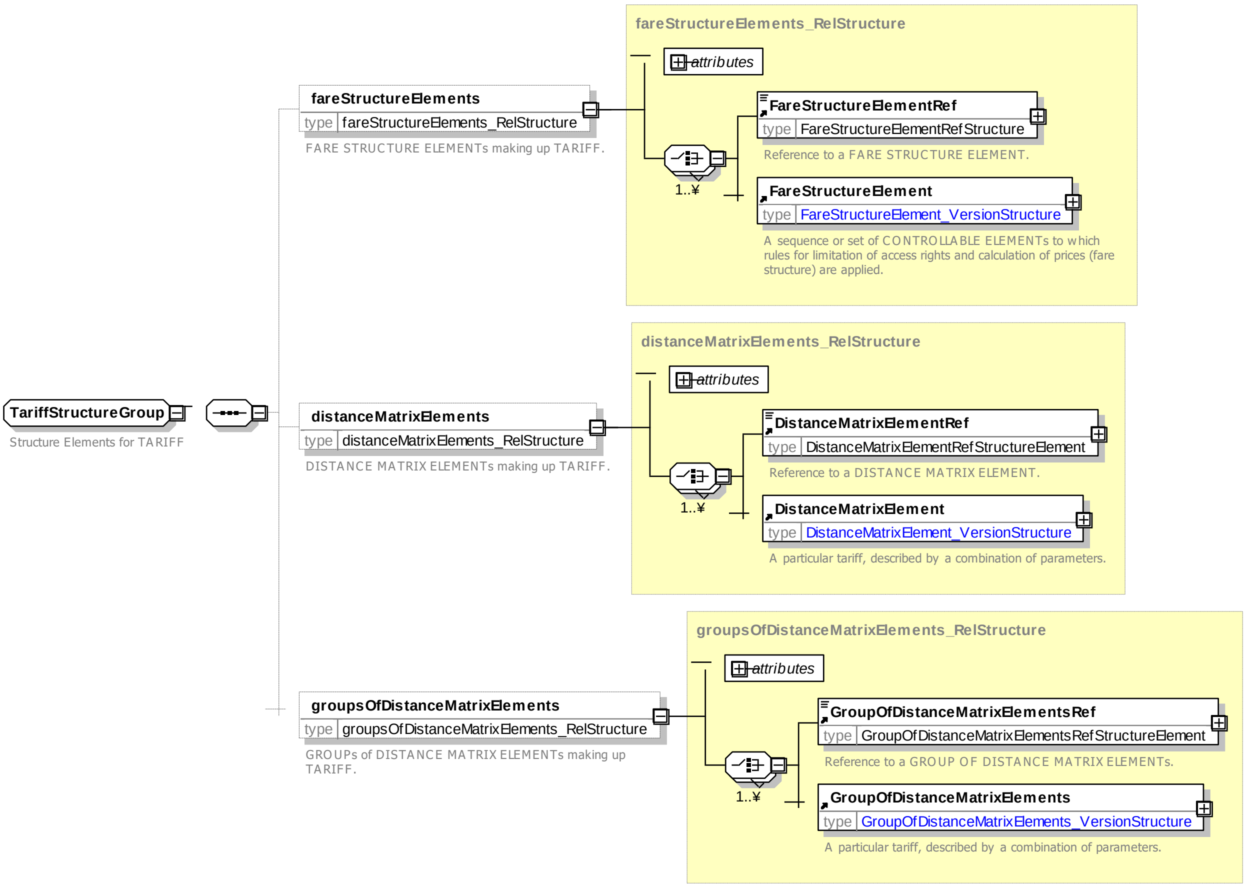


Figure 104 — *TariffStructureGroup* — XSD

7.5.6.6.5.9 TariffPricesGroup – Group

The ***TariffPricesGroup*** defines the pricing elements comprising the TARIFF.

Table 76 – *TariffPricesGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«cntd»	***priceGroups***	*PriceGroup*	0:*	PRICE GROUPs for the TARIFF.
«cntd»	***fareTables***	*FareTable*	0:*	FARE TABLEs for the TARIFF.

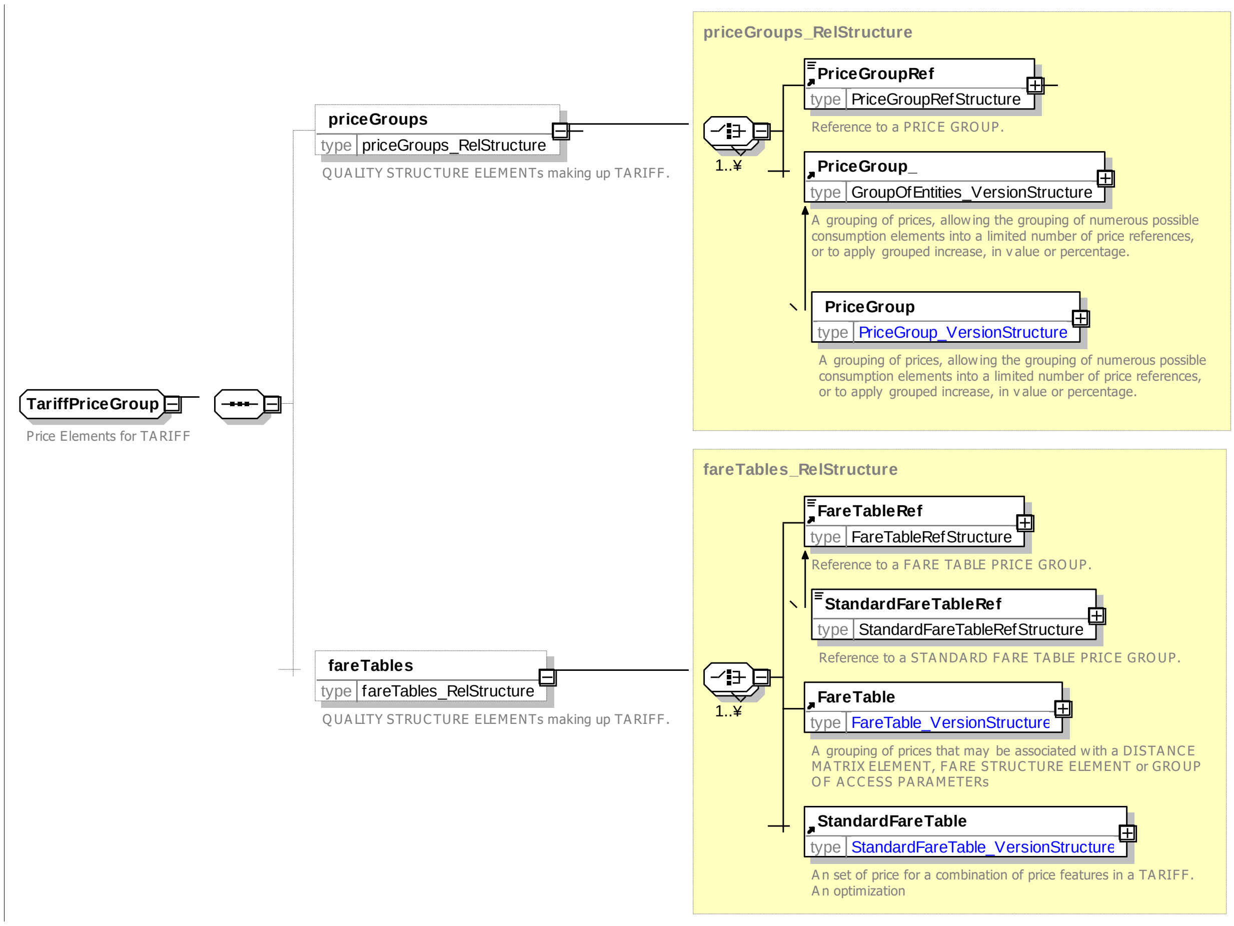


Figure 105 — *TariffPricesGroup* — XSD

7.5.6.6.6 TypeOfTariff – Model Element

A classification of TARIFFs to express the different classes of fares.

Table 77 – *TypeOfTariff* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF TARIFF inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfTariffIdType*	1:1	Identifier of TYPE OF TARIFF.

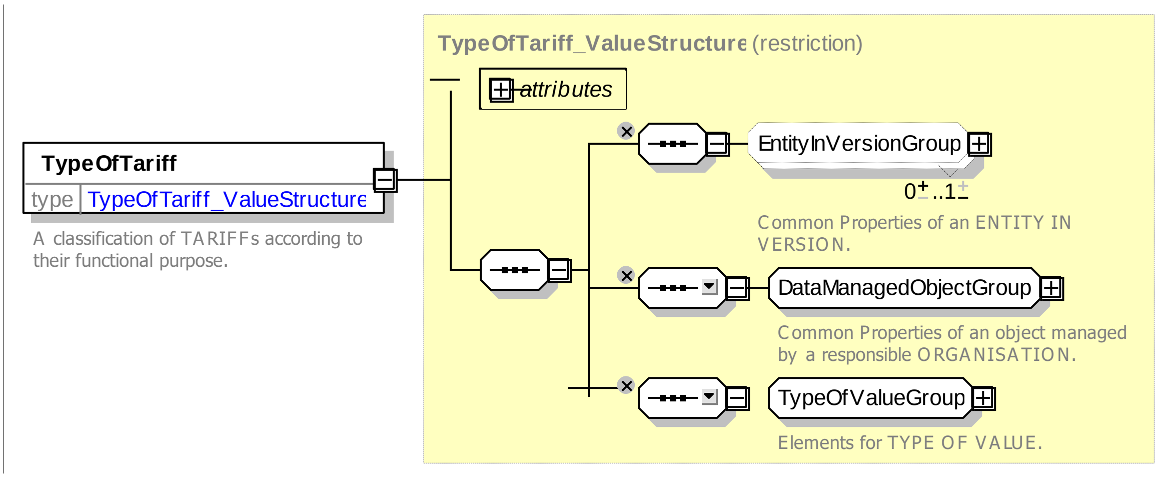


Figure 106 — *TypeOfTariff* — XSD

7.5.6.7 Fare Structure Element – XML examples

7.5.6.7.1 Fare Structure: XML Example of a Tariff with Fare Structure elements

The following code fragment (based on the TAP TSI NRT Tariff) shows a TARIFF for OPERATOR '*0106'.* Fares can be composed of a seat class (*first or second*), and a ROUND TRIP (*single or return*) combination, together with a choice of one out of three profiles (*adult / single, group ticket)*. Fares are available for set of routes represented by a GROUP OF DISTANCE MATRIX ELEMENTs, constrained to be one of a set of four SERIES CONSTRAINTs.

The example shows

a) A TARIFF Definition

b) A GROUP OF DISTANCE MATRIX ELEMENTs made up of four DISTANCE MATRIX ELEMENTs (each corresponding also to a SERIES CONSTRAINT).

c) A FARE STRUCTURE ELEMENT defining the allowed ROUND TRIP combinations as logically ORed alternatives, of which only one may be selected.

d) A FARE STRUCTURE ELEMENT defining the allowed seat classes as logically ORed alternatives, of which only one may be selected.

e) A FARE STRUCTURE ELEMENT defining the allowed USER PROFILEs as logically ORed alternatives, of which only one may be selected.

f) A FARE STRUCTURE ELEMENT defining the allowed SERIES CONSTRAINTs as logically ORed alternatives A FARE STRUCTURE ELEMENT defining the allowed USER PROFILEs as logically ORed alternatives, of which only one may be selected.

For EXAMPLE:

```
<Tariff id="tap:NrtProduct@Route@Basic01" version="01">
    <Name>Standard route based Fare table 1</Name>

    <validityConditions>
        <AvailabilityCondition id="tap:Tariff01" version="01">
            <FromDate>2011-01-01T00:00:00Z</FromDate>
            <ToDate>2014-01-01T00:00:00Z</ToDate>
```

```xml
        </AvailabilityCondition>
    </validityConditions>
    <OperatorRef ref="tap:0106" version="any"/>
    <TypeOfTariffRef ref="tap:B.1.1:01" version="any"/>
    <TariffBasis>route</TariffBasis>
    <ReturnFareTwiceSingle>true</ReturnFareTwiceSingle>
    <fareStructureElements>
        <FareStructureElementRef ref="tap:NrtProduct@round_trips" version="01"/>
        <FareStructureElementRef ref="tap:NrtProduct@fare_classes" version="01"/>
        <FareStructureElementRef ref="tap:NrtProduct@profiles" version="01"/>
        <FareStructureElementRef ref="tap:NrtProduct@series" version="01"/>
    </fareStructureElements>
    <groupsOfDistanceMatrixElements>
        <GroupOfDistanceMatrixElementsRef ref="tap:NrtProduct@Routes" version="01"/>
    </groupsOfDistanceMatrixElements>
</Tariff>


<groupsOfDistanceMatrixElements>
    <GroupOfDistanceMatrixElements id="tap:NrtProduct@Routes" version="01">
        <members>
            <DistanceMatrixElementRef ref="tap:series555" version="01"/>
            <DistanceMatrixElementRef ref="tap:series777" version="01"/>
            <DistanceMatrixElementRef ref="tap:series1234" version="01"/>
            <DistanceMatrixElementRef ref="tap:series1235" version="01"/>
        </members>
    </GroupOfDistanceMatrixElements>
</groupsOfDistanceMatrixElements>


<!-- === COMMON FARE STRUCTURE FACTORs === -->
<fareStructureElements>
    <FareStructureElement id="tap:NrtProduct@round_trips" version="01">
        <Name>Single or return tickets are available</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment id="tap:NrtProduct@roundTrip" version="01">
                <PreassignedFareProductRef ref="tap:NrtProduct" version="01"/>
                <LimitationGroupingType>OR</LimitationGroupingType >
                        <limitations>
                            <RoundTripRef ref="tap:single" version="any"/>
                            <RoundTripRef ref="tap:return" version="any"/>
                        </limitations>
                    </GenericParameterAssignment>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </FareStructureElement>

    <FareStructureElement id="tap:NrtProduct@fare_classes" version="01">
        <Name>First of second class tickets are available </Name>
        <validityParameterAssignments>
            <GenericParameterAssignment id="tap:NrtProduct@FareClass" version="01">
                <GroupingType>OR</GroupingType>
                <PreassignedFareProductRef ref="tap:NrtProduct" version="01"/>
                <includes>
                    <GenericParameterAssignment id="tap:NrtProduct@firstClass" version="01">

    <ValidityParameters><FareClass>firstClass</FareClass></ValidityParameters>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment id="tap:NrtProduct@secondClass" version="01">

    <ValidityParameters><FareClass>secondClass</FareClass></ValidityParameters>
                    </GenericParameterAssignment>
                </includes>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </FareStructureElement>

    <!-- PROFILEs -->
    <FareStructureElement id="tap:NrtProduct@profiles" version="01">
        <Name>Three types of ticket are available; adult and child and group</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment id="tap:NrtProduct@profiles" version="01">
                <GroupingType>OR</GroupingType>
                <includes>
                    <GenericParameterAssignment id="tap:NrtProduct@profiles@adult" version="01">
                        <limitations>
```

```
                        <UserProfileRef ref="tap:adult" version="any"/>
                    <limitations>
                </GenericParameterAssignment>
                <GenericParameterAssignment id="tap:NrtProduct@profiles@child" version="01">
                    <limitations>
                        <UserProfileRef ref="tap:child" version="any"/>
                    </limitations>
                </GenericParameterAssignment>
                <GenericParameterAssignment id="tap:NrtProduct@profiles@group" version="01">
                    <limitations>
                        <GroupTicketRef ref="tap:group_ticket" version="any"/>
                    </limitations>
                </GenericParameterAssignment>
            </includes>
        </GenericParameterAssignment>
    </validityParameterAssignments>
</FareStructureElement>

<!-- SERIES CONSTRAINTs -->
<FareStructureElement id="tap:NrtProduct@series" version="01">
    <Name>For Basic Tariffs </Name>
    <validityParameterAssignments>
        <GenericParameterAssignment id="tap:NrtProduct@series" version="01">
            <GroupingType>OR</GroupingType>
            <includes>
                <GenericParameterAssignment id="tap:NrtProduct@series@series555" version="01">
                    <ValidityParameters>
                        <SeriesConstraintRef ref="tap:series555" version="01"/>
                        <Directions>both</Directions>
                    </ValidityParameters>
                </GenericParameterAssignment>
                <GenericParameterAssignment id="tap:NrtProduct@series@series777" version="01">
                    <ValidityParameters>
                        <SeriesConstraintRef ref="tap:series777" version="01"/>
                        <Directions>both</Directions>
                    </ValidityParameters>
                </GenericParameterAssignment>
                <GenericParameterAssignment id="tap:NrtProduct@series@series1234"
version="01">
                    <ValidityParameters>
                        <SeriesConstraintRef ref="tap:series1234" version="01"/>
                        <Directions>both</Directions>
                    </ValidityParameters>
                </GenericParameterAssignment>
                <GenericParameterAssignment id="tap:NrtProduct@series@series1235"
version="01">
                    <ValidityParameters>
                        <SeriesConstraintRef ref="tap:series1235" version="01"/>
                        <Directions>both</Directions>
                    </ValidityParameters>
                </GenericParameterAssignment>
            </includes>
        </GenericParameterAssignment >
    </validityParameterAssignments>
</FareStructureElement>
</fareStructureElements>

…. etc, etc
```

7.5.6.7.2 Fare Structure: XML Example of a Tariff for a sequence of Fare Structure Elements

The following code fragment shows a TARIFF for a Fare Structure in which there are specific fares for particular sequences of zones, represented by a FARE STRUCTURE ELEMENT containing three FARE ELEMENTs IN SEQUENCE.

For EXAMPLE:

```
<Tariff version="any" id="mygtfsxm:DTA">
    <Name>GTFS Example 7 : Zonal fare structure</Name>
    <fareStructureElements>

        <!-- === sequences of use === -->
        <FareStructureElement version="any" id="mygtfsxm:F1">
```

```xml
            <Name>F1 = Sequence 1 2 3</Name>
            <fareStructureElementsInSequence>
                <FareStructureElementInSequence version="any" id="mygtfsxm:F1_01" order="1">
                    <IsFirstInSequence>true</IsFirstInSequence>
                    <FareStructureElementRef version="any" ref="mygtfsxm:1"/>
                </FareStructureElementInSequence>
                <FareStructureElementInSequence version="any" id="mygtfsxm:F1_02" order="2">
                    <FareStructureElementRef version="any" ref="mygtfsxm:2"/>
                </FareStructureElementInSequence>
                <FareStructureElementInSequence version="any" id="mygtfsxm:F1_03" order="3">
                    <IsLastInSequence>true</IsLastInSequence>
                    <FareStructureElementRef version="any" ref="mygtfsxm:3"/>
                </FareStructureElementInSequence>
            </fareStructureElementsInSequence>
            <prices>
                <FareStructureElementPrice version="any" id="mygtfsxm:F1">
                    <Amount>4.15</Amount>
                    <Currency>USD</Currency>
                </FareStructureElementPrice>
            </prices>
        </FareStructureElement>

…. etc, etc


<!-- === zones === -->
    <FareStructureElement version="any" id="mygtfsxm:1">
        <Name>Zone 1</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="mygtfsxm:1">
                <ValidityParameters>
                    <TariffZoneRef version="any" ref="mygtfsxm:1"/>
                </ValidityParameters>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </FareStructureElement>
    <FareStructureElement version="any" id="mygtfsxm:2">
        <Name>Zone 2</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="mygtfsxm:2">
                <ValidityParameters>
                    <TariffZoneRef version="any" ref="mygtfsxm:2"/>
                </ValidityParameters>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </FareStructureElement>
    <FareStructureElement version="any" id="mygtfsxm:3">
        <Name>Zone 3</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="mygtfsxm:3">
                <ValidityParameters>
                    <TariffZoneRef version="any" ref="mygtfsxm:3"/>
                </ValidityParameters>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </FareStructureElement>

…. etc, etc
```

7.5.7 Distance Matrix Element

7.5.7.1 DISTANCE MATRIX ELEMENT – Conceptual MODEL

The DISTANCE MATRIX MODEL allows point to point to fares to be described. Each DISTANCE MATRIX ELEMENT represents the fare between an origin and a destination pair; either two SCHEDULED STOP POINTs or two TARIFF ZONEs, or two FARE SECTIONs. A GROUP OF DISTANCE MATRIX ELEMENTs specifies a set of DISTANCE MATRIX ELEMENTs, allowing a common set of prices for between different origin-destination pairs if required.

There may be multiple SERIES CONSTRAINTs associated with a DISTANCE MATRIX ELEMENT, each representing a different routing constraint.

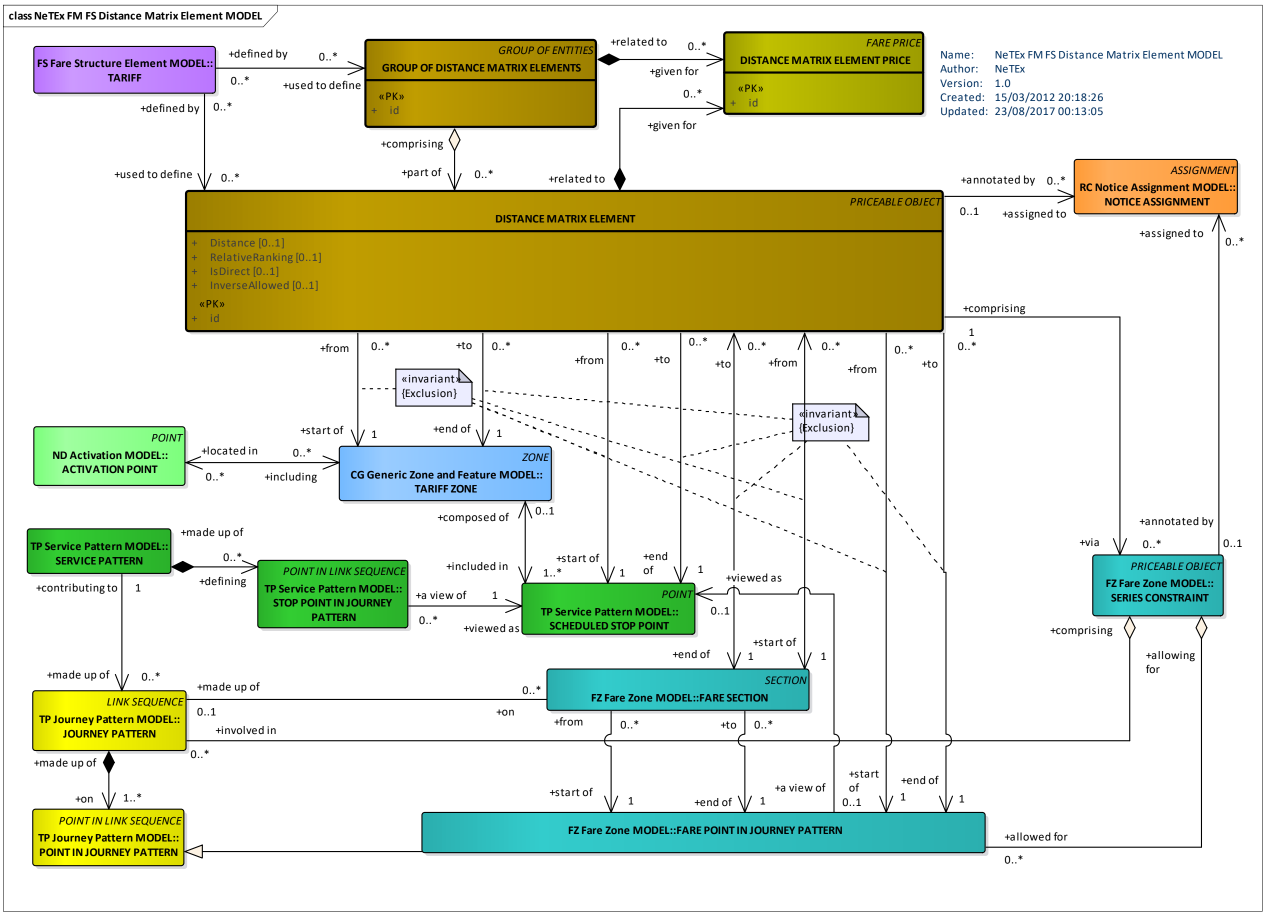


Figure 107 — Distance Matrix Element – Conceptual MODEL (UML)

7.5.7.2 Distance Matrix Element – Conceptual Examples

7.5.7.2.1 Example – Point to Point fare with absolute prices

The following partial table shows a classic point to point fare table. Each cell can be considered a DISTANCE MATRIX ELEMENT.

Outward (Absolute Fare Price)

Ask Av					
Bath Pl	£0.40				
Cam Sq	£0.50	£0.40			
Dee St	£0.75	£0.75	£0.50		
Ely Rd	£1.00	£1.00	£0.75	£0.40	
	Ask Av	Bath Pl	Cam Sq	Dee St	Ely Rd

Table 1-2 Example Triangular Fare Table with Absolute Prices

Figure 108 — Example: Distance Matrix Element – Triangular Fare table with absolute Prices (EXM)

7.5.7.2.2 Example – Point to Point fare with price groups

The following partial table shows a classic point to point fare table, but instead of absolute prices, price groups are used for the fares (P, Q, R, etc), so that many different fares may be link as single group.

Outward ('Virtual Fare Price')

Ask Av					
Bath Pl	P				
Cam Sq	Q	P			
Dee St	R	R	Q		
Ely Rd	S	S	R	P	
	Ask Av	Bath Pl	Cam Sq	Dee St	Ely Rd

Table 1-3 Example Triangular Fare Table with Fare bands

Figure 109 — Example: Distance Matrix Element – Triangular Fare table with price groups (EXM)

7.5.7.2.3 Example – Zone to Zone fare

The following partial table shows the zone-to-zone fares for travel by rail in the London area. Furthermore, there are different prices for different times of travel.

Adult National Rail only fares

Zone		Oyster pay as you go	
		Peak single	Off-peak single
Zone 1 only		£2.20	£1.70
Zones 1-2		£2.40	£1.90
Zones 1-3		£3.10	£2.20
Zones 1-4		£3.60	£2.50
Zones 1-5		£4.70	£2.90
Zones 1-6		£5.70	£3.50
Zones 1-7*		£5.70	£3.90
Zones 1-8*		£6.70	£3.90
Zones 1-9		£6.70	£3.90
Zone 2 only		£1.70	£1.50
Zones 2-3		£2.10	£1.70
Zones 2-4		£2.50	£1.90
Zones 2-5		£3.20	£2.20
Zones 2-6		£3.80	£2.40
Zones 2-7*		£3.90	£2.70

Figure 110 — Example: Distance Matrix Element – Zone to Zone fare on London (EXM)

7.5.7.3 Distance Matrix Element – Physical model

The following figure shows the physical model for DISTANCE MATRIX ELEMENTs.

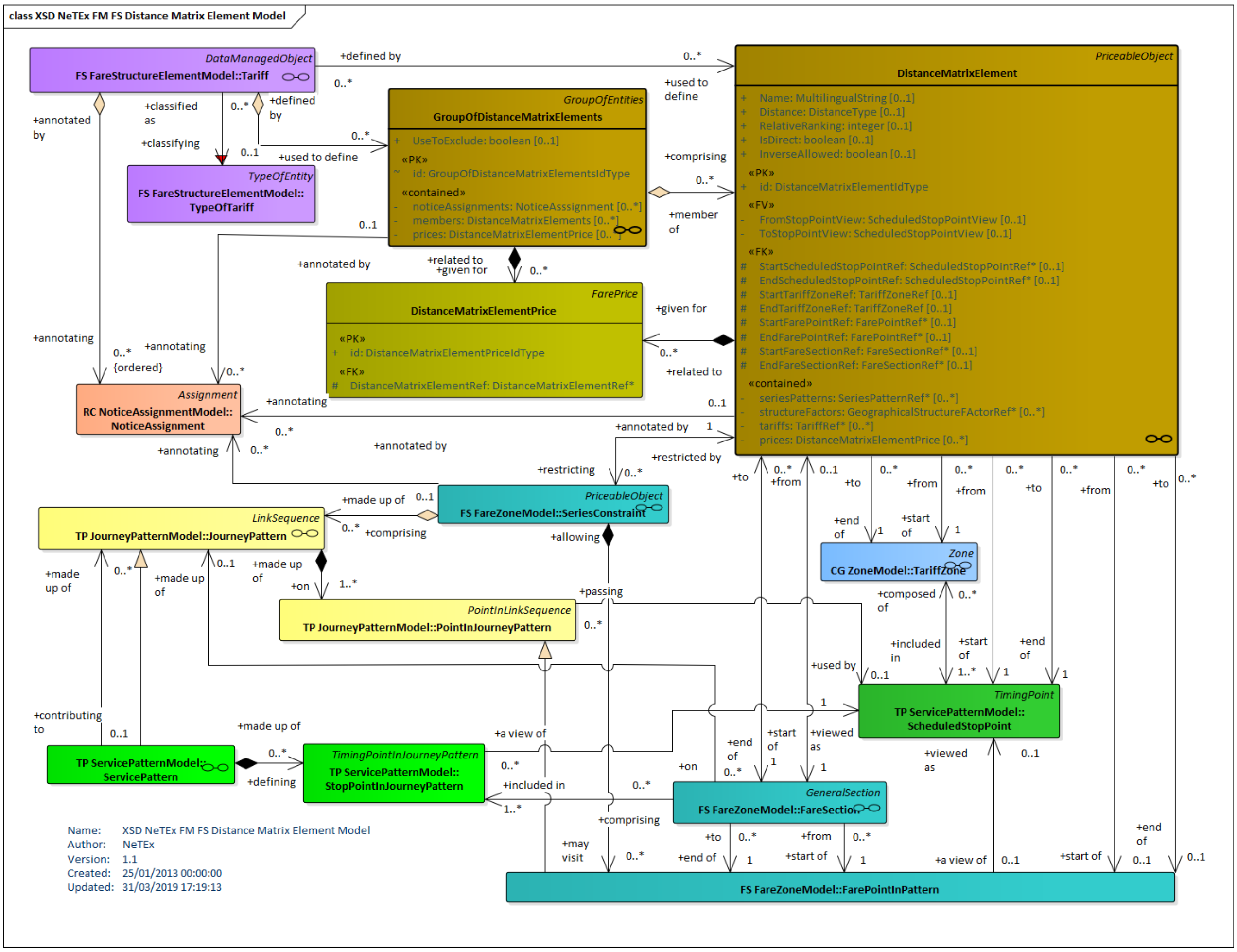


Figure 111 — Distance Matrix Element – Physical Model (UML)

7.5.7.4 Distance Matrix Element – Attributes and XSD

7.5.7.4.1 DistanceMatrixElement – Model Element

A cell of an origin-destination matrix for TARIFF ZONEs or STOP POINTs, expressing a fare distance for the corresponding trip: value in km, number of fare units etc.

Table 78 – *DistanceMatrixElement* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PriceableObject*	::>	DISTANCE MATRIX ELEMENT inherits from PRICEABLE OBJECT.
«PK»	***id***	*DistanceMatrix-ElementIdType*	1:1	Identifier of a DISTANCE MATRIX ELEMENT.
	Name	*MultilingualString*	0:1	Name of DISTANCE MATRIX ELEMENT.

XGRP	***DistanceMatrixElement-DetailsGroup***	***xmlGroup***	1:1	Detailed property elements for DISTANCE MATRIX ELEMENT.
XGRP	***DistanceMatrixElement-ODGroup***	***xmlGroup***	1:1	Origin and Destination elements for DISTANCE MATRIX ELEMENT.
XGRP	***DistanceMatrixElement-ComponentGroup***	***xmlGroup***	1:1	Component elements for DISTANCE MATRIX ELEMENT.
XGRP	***DistanceMatrixElement-PricingGroup***	***xmlGroup***	1:1	Pricing elements for DISTANCE MATRIX ELEMENT.

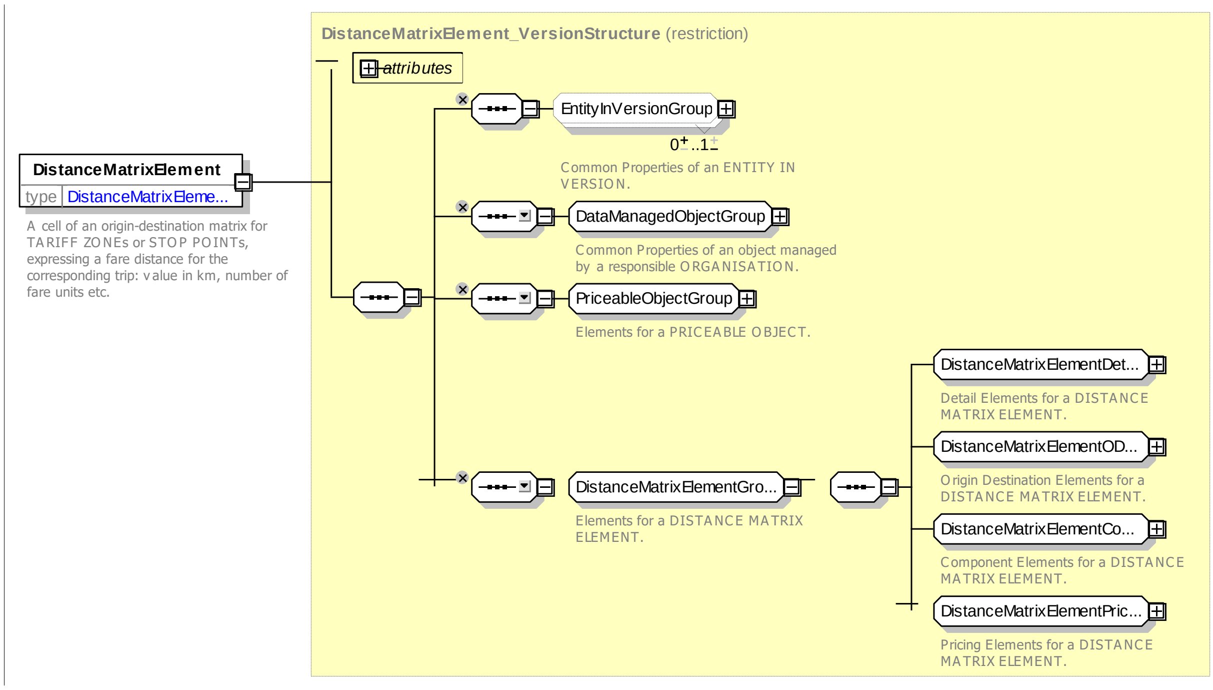


Figure 112 — *DistanceMatrixElement* — XSD

7.5.7.4.1.1 *DistanceMatrixElementDetailsGroup* – Group

The ***DistanceMatrixElementDetailsGroup*** defines basic properties of the DISTANCE MATRIX ELEMENT.

Table 79 – *DistanceMatrixElementDetailsGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Distance	*DistanceType*	0:1	Distance between origin and destination of a DISTANCE MATRIX ELEMENT.
	RelativeRanking	*xsd:integer*	0:1	Relative preference assigned to this element if there are multiple entries between two points.
	IsDirect	*xsd:boolean*	0:1	Whether journey is direct or requires changes.

	InverseAllowed	*xsd:boolean*	0:1	Whether an inverse element in the opposite direction with the same prices may be assumed – optimisation to reduce data volumes.

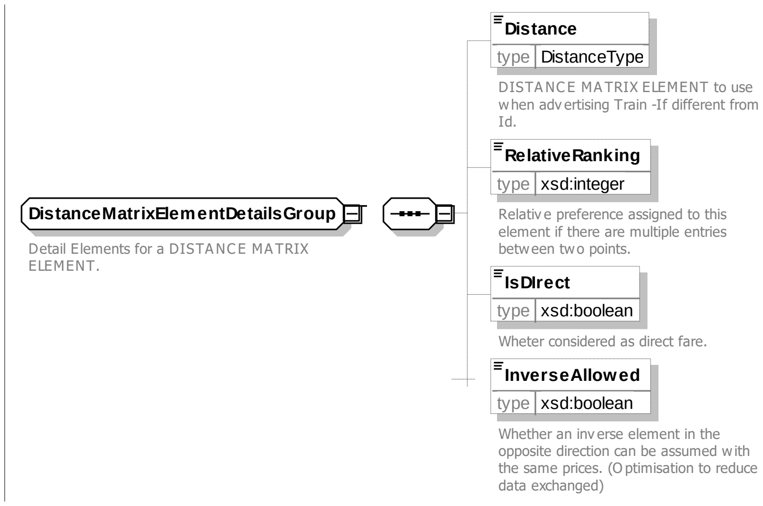


Figure 113 — *DistanceMatrixElementDetailsGroup* — XSD

7.5.7.4.1.2 *DistanceMatrixElementODGroup* – Group

The ***DistanceMatrixElementODGroup*** defines origin and destination elements of the DISTANCE MATRIX ELEMENT; these will be either SCHEDULED STOP POINTs, TARIFF ZONEs or FARE SECTIONs.

Table 80 – *DistanceMatrixElementODGroup* – Group

Classifi-cation	Name		Type	Cardinality	Description
			Choice	1:1	Origin of DISTANCE MATRIX ELEMENT
«FK»	***a***	***Start-StopPointRef***	*ScheduledStopPointRef*	0:1	Start SCHEDULED STOP POINT at which a DISTANCE MATRIX ELEMENT begins.
«FV»	***b***	***Start-StopPointView***	*ScheduledStopPointView*	0:1	Details of origin SCHEDULED STOP POINT.
«FK»	***c***	***Start-TariffZoneRef***	*TariffZoneRef*	0:1	Start TARIFF ZONE at which a DISTANCE MATRIX ELEMENT begins.
«FV»	***d***	***Start-TariffZone-View***	*TariffZoneView*	0:1	Details of origin TARIFF ZONE.
«FK»	***e***	***StartFare-SectionRef***	*FareSectionRef*	0:1	Start FARE SECTION at which a DISTANCE MATRIX ELEMENT begins.
«FK»	***f***	***StartFarePoint InPatternRef***	*FarePointInPatternRef*	0:1	Start FARE POINT IN PATTERN at which a DISTANCE MATRIX ELEMENT begins. (Handles case of repeated visits)
			Choice	1:1	Destination of DISTANCE MATRIX ELEMENT.

«FK»	***a***	***End-StopPointRef***	*ScheduledStopPointRef*	0:1	End SCHEDULED STOP POINT at which a DISTANCE MATRIX ELEMENT ends.
«FV»	***b***	***End-StopPointView***	*ScheduledStopPointView*	0:1	Details of destination SCHEDULED STOP POINT
«FK»	***c***	***EndTariff-ZoneRef***	*TariffZoneRef*	0:1	Final TARIFF ZONE at which a DISTANCE MATRIX ELEMENT ends.
«FV»	***d***	***EndTariff-ZoneView***	*TariffZoneView*	0:1	Details of origin TARIFF ZONE.
«FK»	***e***	***EndFare-SectionRef***	*FareSectionRef*	0:1	End FARE SECTION at which a DISTANCE MATRIX ELEMENT ends.
«FK»	***f***	***EndFarePoint-InPatternRef***	*FarePointInPatternRef*	0:1	End FARE POINT IN PATTERN at which a DISTANCE MATRIX ELEMENT ends. (Handles case of repeated visits).

Figure 114 — *DistanceMatrixElementODGroup* — XSD

7.5.7.4.1.3 *DistanceMatrixElementComponentGroup* – Group

The ***DistanceMatrixElementComponentGroup*** defines additional structural elements relating to the of the DISTANCE MATRIX ELEMENT; in particular SERIES CONSTRAINTs limiting the allowed routes and GEOGRAPHICAL STRUCTURE FACTORs, defining the geographical cost basis.

Table 81 – *DistanceMatrixElementComponentGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***series-Constraints***	*SeriesConstraintRef*	0:*	SERIES CONSTRAINTs associated with this DISTANCE MATRIX ELEMENT.
«cntd»	***structureFactors***	*GeographicalStructureFactorRef*	0:*	STRUCTURE FACTORs associated with this DISTANCE MATRIX ELEMENT.

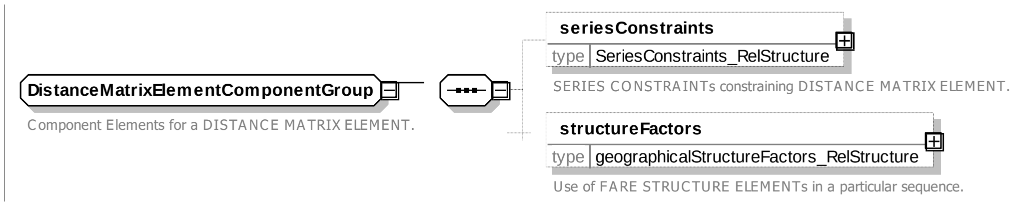


Figure 115 — *DistanceMatrixElementComponentGroup* — XSD

7.5.7.4.1.4 *DistanceMatrixElementPricingGroup* – Group

The ***DistanceMatrixElementPricingGroup*** defines pricing elements for a DISTANCE MATRIX ELEMENT.

Table 82 – *DistanceMatrixElementPricingGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***tariffs***	*TariffRef*	0:*	TARIFFs for the DISTANCE MATRIX ELEMENT.
FK	***FareTableRef***	*FareTableRef+*	0:1	Primary FARE TABLE for the DISTANCE MATRIX ELEMENT.
«cntd»	***fareTables***	*FareTableRef*	0:*	FARE TABLEs for the DISTANCE MATRIX ELEMENT.

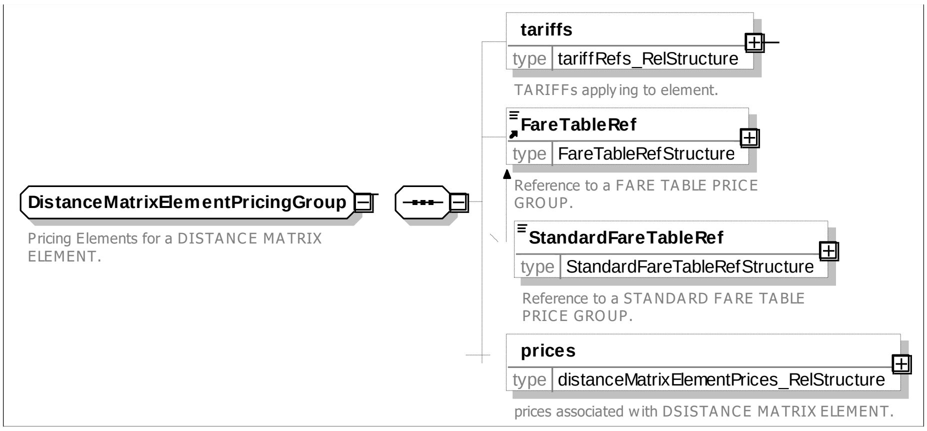


Figure 116 — *DistanceMatrixElementPricingGroup* — XSD

7.5.7.4.2 DistanceMatrixElementPrice – Model Element

A set of all possible price features of a DISTANCE MATRIX ELEMENT: default total price etc.

Table 83 – *DistanceMatrixElementPrice* – Element

Classifi-cation	Name		Type	Cardinality	Description
::>	::>		*FarePrice*	::>	DISTANCE MATRIX ELEMENT PRICE inherits from FARE PRICE.
«PK»	***id***		*DistanceMatrix-ElementPriceIdType*	1:1	Identifier of DISTANCE MATRIX ELEMENT PRICE.
			CHOICE	0:1	
«FK»	***a***	***Distance-Matrix-ElementRef***	*DistanceMatrixElementRef*	0:1	DISTANCE MATRIX ELEMENT for which this is the price.
«FK»	***b***	***GroupOf-Distance-Matrix-ElementsRef***	*GroupOf-DistanceMatrixElementsRef*	0:1	GROUP OF DISTANCE MATRIX ELEMENTs for which this is the price.

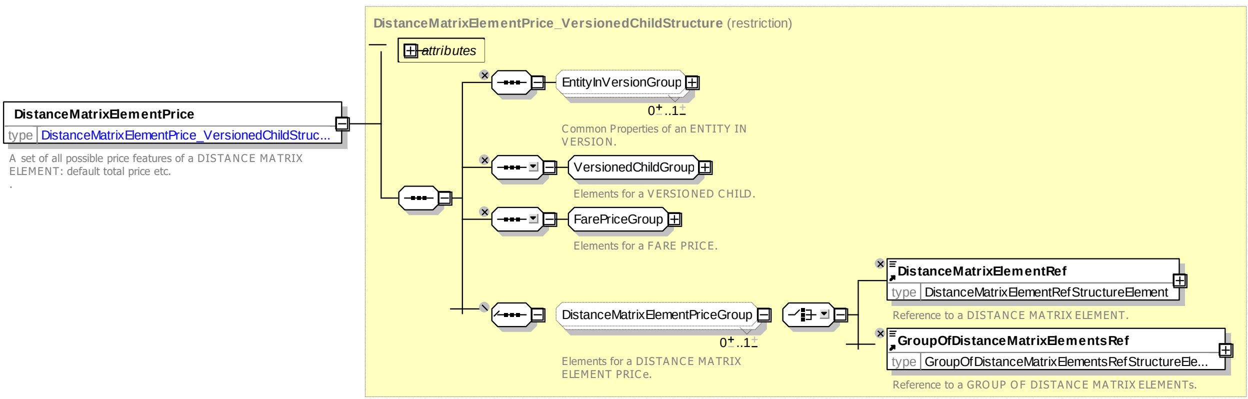


Figure 117 — *DistanceMatrixElementPrice* — XSD

7.5.7.4.3 GroupOfDistanceMatrixElements – Model Element

A grouping of DISTANCE MATRIX ELEMENTs. May be used to provide reusable Origin / Destination pairs.

Table 84 – *GroupOfDistanceMatrixElements* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*GroupOfEntities*	::>	GROUP of DISTANCE MATRIX ELEMENTs inherits from GROUP OF ENTITies.
«PK»	***id***	*GroupOfDistance-MatrixElementsIdType*	1:1	Identifier of GROUP of DISTANCE MATRIX ELEMENTS.
	UseToExclude	*xsd:boolean*	0:1	Whether contents of Group should be used to exclude (true) from a larger list. The default value is '*false* (i.e. "include")
«cntd»	***priceGroups***	*PriceGroup*	0:*	PRICE GROUPs for the GROUP OF DISTANCE MATRIX ELEMENTs.
«cntd»	***fareTables***	*FareTable*	0:*	FARE TABLEs for the GROUP OF DISTANCE MATRIX ELEMENTs.
	Distance	*DistanceType*	0:1	Distance between origins and destinations of a GROUP OF DISTANCE MATRIX ELEMENTs.
«cntd»	***structureFactors***	*GeographicalStructure-FactorRef*	0:*	References to GEOGRAPHICAL STRUCTURE FACTORs.
«cntd»	***notice-Assignments***	*NoticeAsssignment*	0:*	NOTICE ASSIGNMENTs for GROUP OF DISTANCE MATRIX ELEMENTs.
«cntd»	***members***	*DistanceMatrixElements*	0:*	References to members of the GROUP OF DISTANCE MATRIX ELEMENTs.
«cntd»	***prices***	*DistanceMatrixElementPrice*	0:*	Prices for the GROUP OF DISTANCE MATRIX ELEMENTs.

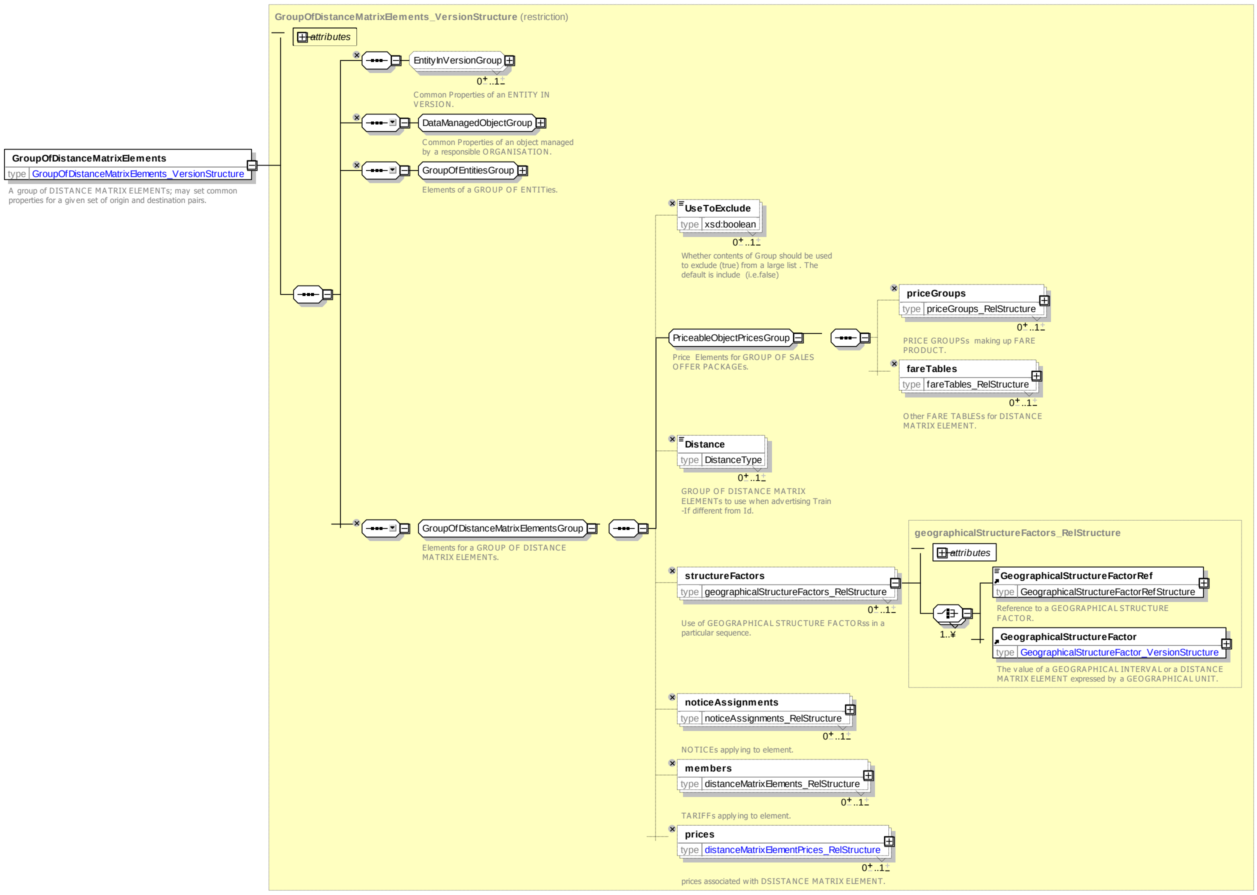


Figure 118 — *GroupOfDistanceMatrixElements* — XSD

7.5.7.5 Distance Matrix Element – XML examples

7.5.7.5.1 Distance Matrix Element: XML Example of Distance matrix elements

The following code fragment shows a GROUP OF DISTANCE MATRIX ELEMENTs with two DISTANCE MATRIX ELEMENTs providing point to point fares between Brussels and Cologne and Paris and Cologne.

For EXAMPLE:

```
<groupsOfDistanceMatrixElements>
    <GroupOfDistanceMatrixElements id="tap:Flex" version="01">
        <members>
            <DistanceMatrixElement id="tap:Flex:008814001+008015750" version="01">
                <Name>Brussels - Köln</Name>
                <StartStopPointRef ref="tap:008814001" version="01"/>
                <EndStopPointRef ref="tap:008015750" version="01"/>
                <prices>
                    <DistanceMatrixElementPrice id="tap:Flex:008814001+008015750" version="01">
                        <Amount>79</Amount>
                    </DistanceMatrixElementPrice>
                </prices>
            </DistanceMatrixElement>
            <DistanceMatrixElement id="tap:Flex:008727100+008015750" version="01">
                <Name>Paris - Köln</Name>
                <StartStopPointRef ref="tap:008727100" version="01"/>
                <EndStopPointRef ref="tap:008015750" version="01"/>
                <prices>
                    <DistanceMatrixElementPrice id="tap:Flex:008727100+008015750" version="01">
                        <Amount>169</Amount>
                    </DistanceMatrixElementPrice>
                </prices>
```

```
            </DistanceMatrixElement>
        </members>
    </GroupOfDistanceMatrixElements>
```

7.5.7.5.2 Distance Matrix Element: XML Example of Distance matrix elements

The following code fragment shows the same a GROUP OF DISTANCE MATRIX ELEMENTs using PRICE GROUPs rather than individual DISTANCE MATRIX ELEMENT PRICEs

For EXAMPLE:

```
<groupsOfDistanceMatrixElements>
    <GroupOfDistanceMatrixElements id="tap:Flex" version="01">
        <members>
            <DistanceMatrixElement id="tap:Flex:008814001+008015750" version="01">
                <Name>Brussels - Köln</Name>
                <StartStopPointRef ref="tap:008814001" version="01"/>
                <EndStopPointRef ref="tap:008015750" version="01"/>
                <priceGroups>
                    <PriceGroupRef ref="tap:Flex:008814001+008015750" version="01"/>
                </priceGroups>
            </DistanceMatrixElement>
            <DistanceMatrixElement id="tap:Flex:008727100+008015750" version="01">
                <Name>Paris - Köln</Name>
                <StartStopPointRef ref="tap:008727100" version="01"/>
                <EndStopPointRef ref="tap:008015750" version="01"/>
                <priceGroups>
                    <PriceGroupRef ref="tap:Flex:008814001+008015750" version="01"/>
                </priceGroups>
            </DistanceMatrixElement>
        </members>
    </GroupOfDistanceMatrixElements>
```

7.5.8 Validable & Controllable Elements

7.5.8.1 Validable & Controllable Elements – Conceptual MODEL

The control system of a Public Transport organisation is organised in order to regularly “validate” the consumption of access rights, i.e. that the passengers have the right ticket for the transport on which they are travelling. The validation process is aimed at specifying that an access right is valid, has been consumed and that this consumption was allowed. It uses the results of one or several consecutive controls.

Such a validated access right may include several components for which the fare structure is different. For instance, a fare product may include a discount for travellers using a car park and then public transport. If the fare structure of these two components is different (e.g. flat fares for public transport and price based on duration of stay for car parking), they will be described by two different FARE STRUCTURE ELEMENTs. The discount is granted only when the validation process recognises that both have been consumed in sequence.

Therefore, a VALIDABLE ELEMENT is defined as a sequence or a set of FARE STRUCTURE ELEMENTs, to be consumed as a whole (or validated in one go) i.e. it is not foreseen to use the different elements of the sequence separately in the sense that if one of the elements is consumed separately, then the whole access right is considered as consumed.

A FARE STRUCTURE ELEMENT, dedicated to be consumed as such, is identical to a VALIDABLE ELEMENT.

Typical examples of VALIDABLE ELEMENTs are the following:

— a simple FARE STRUCTURE ELEMENT to be validated in itself (e.g. a trip on a metro network). In such a case, the VALIDABLE ELEMENT will be identical to the FARE STRUCTURE ELEMENT;

- chained FARE STRUCTURE ELEMENTs of which the successive consumption allows a discount, as in the park and ride above example. Such a discount may be applied with a discounted joint ticket, or by a discount on the latest consumed element, or by a discount to both elements with a post-payment system;

- access rights (e.g. trips or rides) where the fare structure changes during consumption, for instance on a train link composed of two sections, each operated by an operator applying a different fare structure than the other.

In other words, the VALIDABLE ELEMENT provides a functional grouping (e.g. "*Metro trip", "rail trip" "rail return trip")* with which to relate fine grained access-right components to a FARE PRODUCT.

For example, a season pass FARE PRODUCT (such as a TfL Travel Card) might give the user the right to make a journey between a particular suburban station and the city centre (for which the VALIDABLE ELEMENT is a "rail trip" with designated start and end stations) and also the right to travel on the metro in the central zone (the VALIDABLE ELEMENT is a "metro trip" with a zonal restriction).

A VALIDABLE ELEMENT can be limited to a particular scope (e.g. MODE, OPERATOR, LINE etc) via an associated VALIDITY PARAMETER ASSIGNMENT.

It can be defined in terms of FARE STRUCTURE ELEMENT, FARE STRUCTURE ELEMENT IN SEQUENCE.

It may also indicate the consumption rights of a PREASSIGNED FARE PRODUCT or AMOUNT OF PRICE UNIT product and/or the allowed discount rights (USAGE DISCOUNT RIGHT)

In certain cases, it is useful to describe the rights in even further detail, in particular to relate it to the ticket checking process and the CONTROLLABLE ELEMENT allows this to be done.

7.5.8.1.1 **Controllable Elements**

The definition of a fare system always includes a basic level of access rights, for which the validity parameters controlled remain the same and are constantly valid. A CONTROLLABLE ELEMENT is defined as the smallest service element:

- of which the actual consumption can be controlled, by means of regular or occasional controls;

- throughout which any controlled parameter remains valid.

A CONTROLLABLE ELEMENT is the basic component of any access rights combination included in a fare product.

Three main types of CONTROLLABLE ELEMENTs will be found in public transport:

- rides on only one vehicle, for instance in buses, trams or other "open" systems. A ride from one SCHEDULED STOP POINT to another, during a VEHICLE JOURNEY, may represent such a CONTROLLABLE ELEMENT;

- trips, composed of sequences of rides, for instance in closed systems such as metro with entry/exit turnstiles. In such a case, interchanges are allowed within the same CONTROLLABLE ELEMENT and are not controlled;

- accesses to joint services (e.g. car park, fair, etc.), if any.

In complex situations, more detailed CONTROLLABLE ELEMENTs are defined. For instance, if a train line uses a track composed of two sections, each operated by a different operator, a single ride on this line will be composed of two CONTROLLABLE ELEMENTs, distinguished by the parameter OPERATOR.

Validity parameters may be attached to one CONTROLLABLE ELEMENT, either:

- at the start of the element, controlled by an entry control; for instance, the consumption should start at a specified SCHEDULED STOP POINT;
- at the end of the element, controlled by an exit control; for instance, the consumption should not end later than 4 p.m.;
- all along the element ("en route" parameter), possibly controlled by any entry, exit or en route control; for instance, the consumption should occur on line 18.

NOTE Control description is out of the scope of NeTEx.

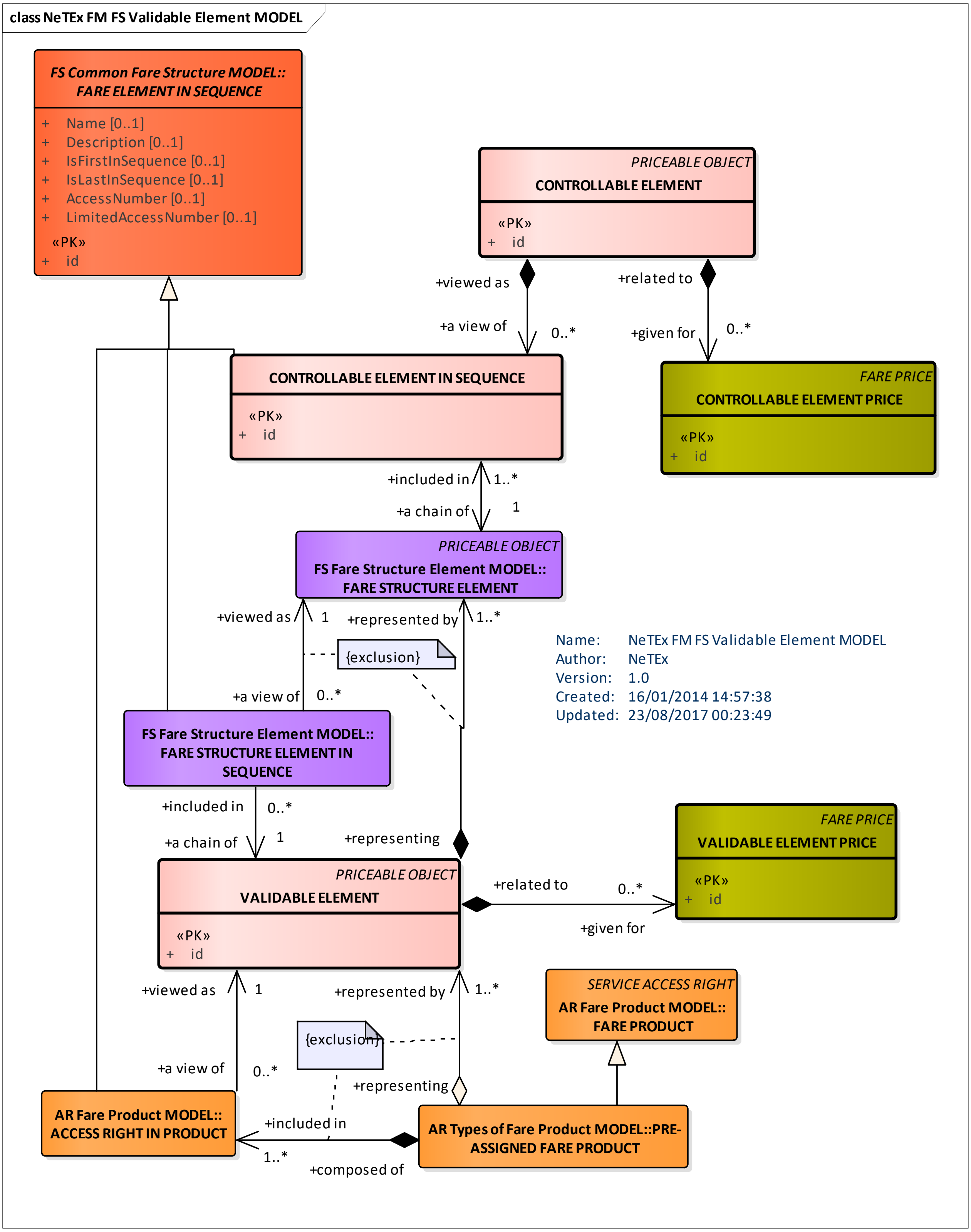


Figure 119 — Controllable & Validable Element Parameter Assignment – Conceptual MODEL (UML)

7.5.8.2 Validable & Controllable Elements – Physical model

The following figure shows the physical model for VALIDABLE & CONTROLLABLE ELEMENTs.

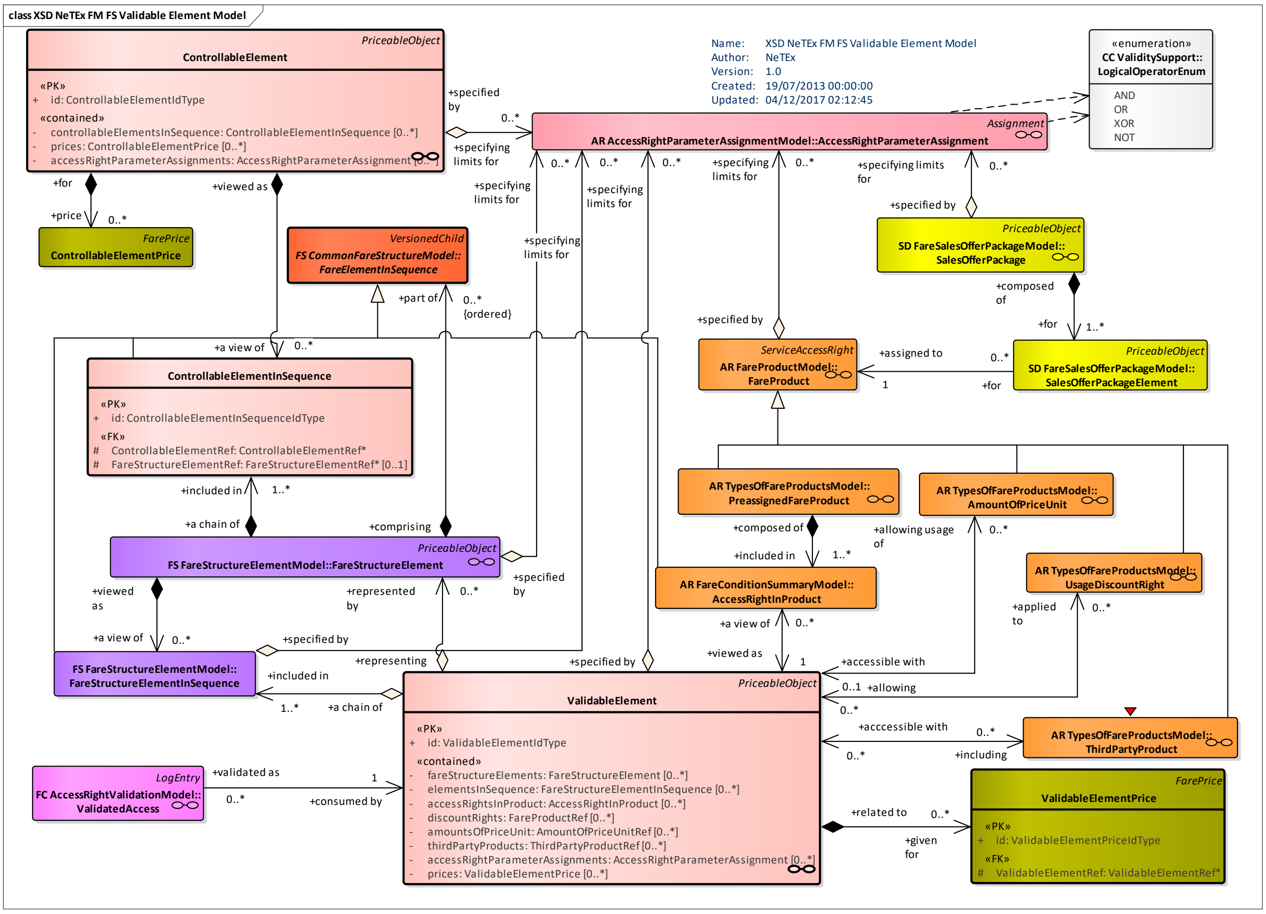


Figure 120 — Validable Element – Physical Model (UML)

7.5.8.3 Validable Element – Attributes and XSD

7.5.8.3.1 ValidableElement – Model Element

A sequence or set of FARE STRUCTURE ELEMENTs, grouped together to be validated in one go.

Table 85 – *ValidableElement* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*PriceableObject*	::>	VALIDABLE ELEMENT inherits from PRICEABLE OBJECT**.**
«PK»	***id***	*ValidableElementIdType*	1:1	Identifier of VALIDABLE ELEMENT.
XGRP	***ValidableElement-StructureGroup***	***xmlGroup***	1:1	Structure elements making up VALIDABLE ELEMENT.
XGRP	***ValidableElement-ProductGroup***	***xmlGroup***	1:1	Product elements making up VALIDABLE ELEMENT.

«cntd»	***prices***	*ValidableElementPrice*	0:*	VALIDABLE ELEMENT PRICEs for element.

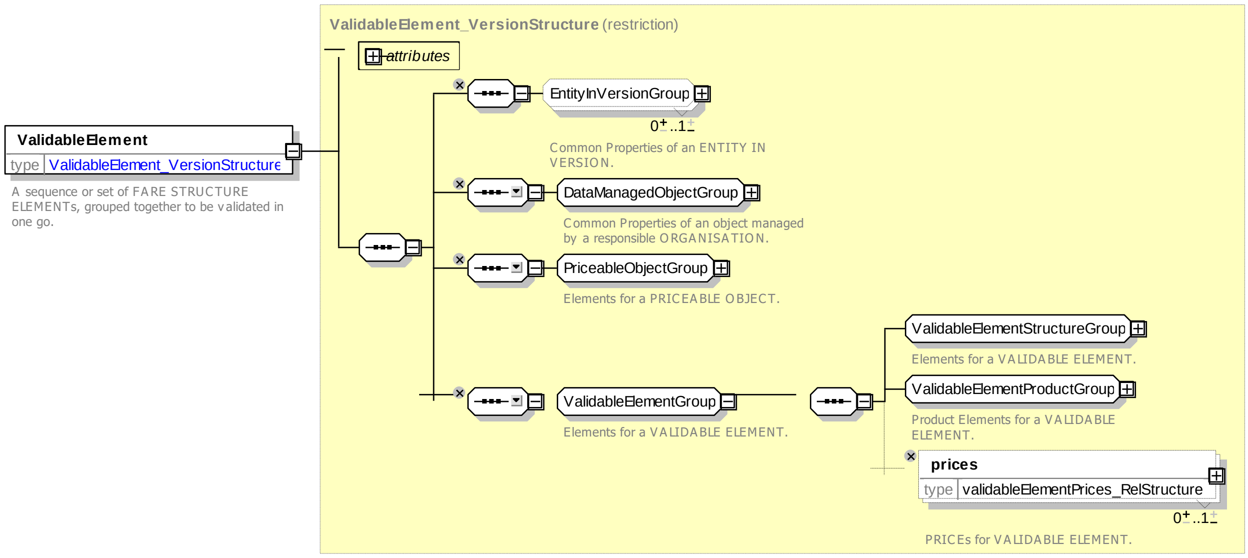


Figure 121 — *ValidableElement* — XSD

7.5.8.3.1.1 ValidableElementStructureGroup – Group

The ***ValidableElementStructureGroup*** defines fare structure elements comprising a VALIDABLE ELEMENT.

Table 86 – *ValidableElementStructureGroup* – Group

Classification	Name	Type	Cardinality	Description
«cntd»	***fareStructure-Elements***	*FareStructureElement*	0:*	FARE STRUCTURE ELEMENTs making up VALIDABLE ELEMENT.
«cntd»	***elements-InSequence***	*FareStructureElement-InSequence*	0:*	FARE STRUCTURE ELEMENTs IN SEQUENCE making up VALIDABLE ELEMENT.

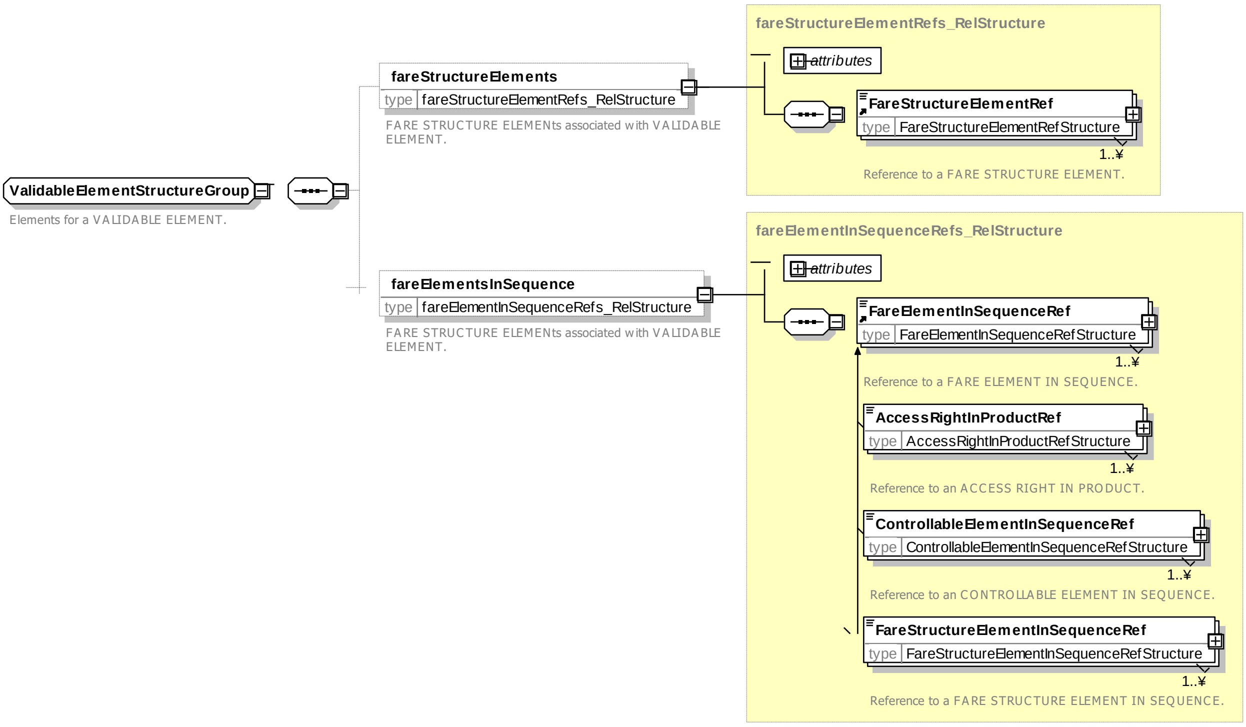


Figure 122 — *ValidableElementStructureGroup* — XSD

7.5.8.3.1.2 ValidableElementProductGroup – Group

The ***ValidableElementProductGroup*** defines FARE PRODUCT elements associated with a VALIDABLE ELEMENT.

Table 87 – *ValidableElementProductGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***accessRights-InProduct***	*AccessRightInProduct*	0:*	ACCESS RIGHT IN PRODUCT making up VALIDABLE ELEMENT.
«cntd»	***discountRights***	*FareProductRef+*	0:*	Discount rights in Product making up VALIDABLE ELEMENT.
«cntd»	***amountOfPrice-Units***	*AmountOfPriceUnitRef*	0:*	AMOUNTs OF PRICE UNIT making up VALIDABLE ELEMENT.
«cntd»	***thirdParty-Products***	*ThirdPartyProductRef*	0:*	THIRD PARTY PRODUCTs for VALIDABLE ELEMENT.
«cntd»	***validity-Parameter-Assignments***	*ValidityParameter-Assignment*	0:*	VALIDITY PARAMETER ASSIGNMENTs for VALIDABLE ELEMENT.

*

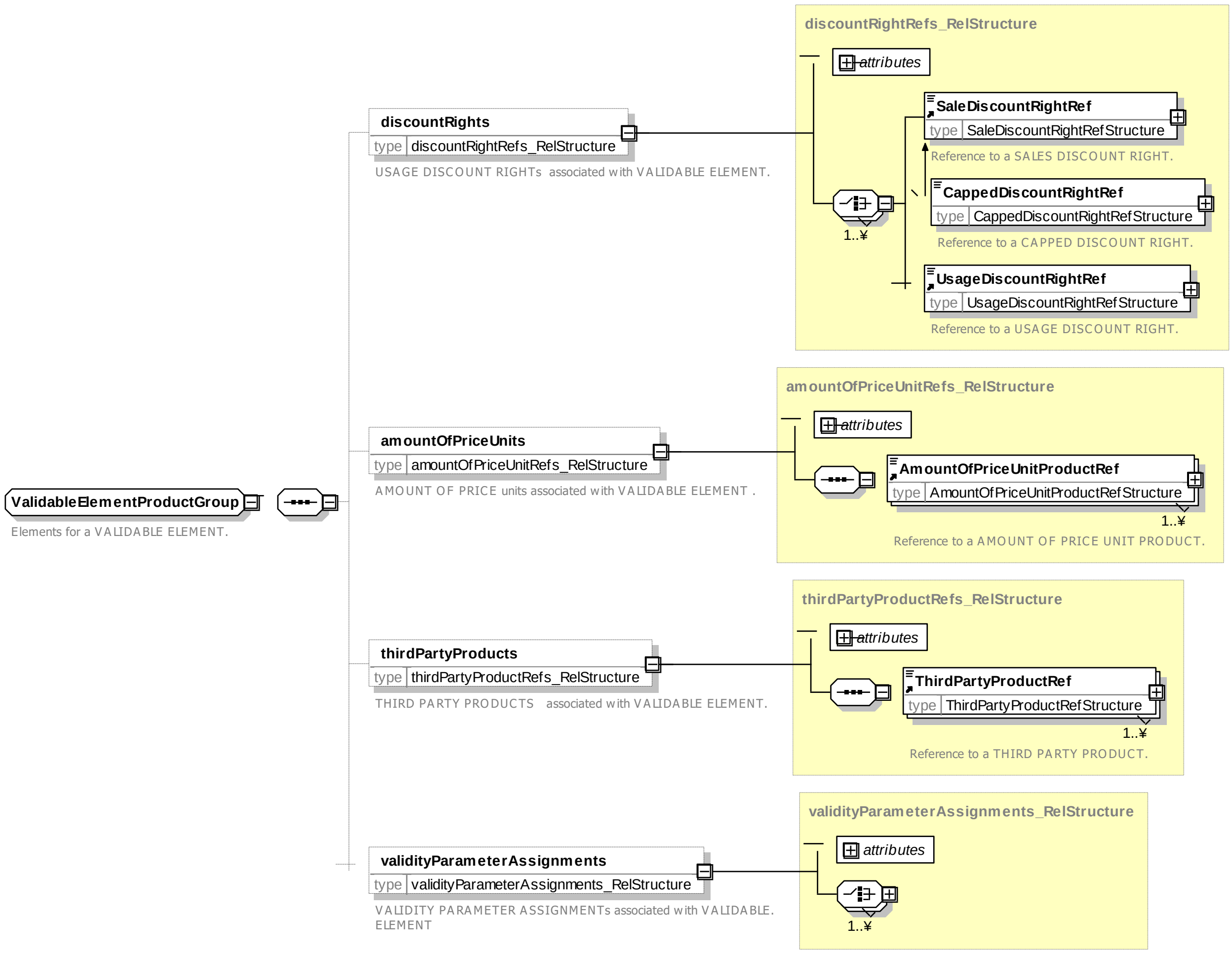


Figure 123 — *ValidableElementProductGroup* — XSD

7.5.8.3.2 **ValidableElementPrice – Model Element**

A set of all possible price features of a VALIDABLE ELEMENT: default total price, discount in value or percentage etc.

Table 88 – *ValidableElementPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	VALIDABLE ELEMENT PRICE inherits from FARE PRICEs
«PK»	***id***	*ValidableElement-PriceIdType*	1:1	Identifier of VALIDABLE ELEMENT PRICE.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	0:1	Reference to a VALIDABLE ELEMENT. If not given by context, must be specified.

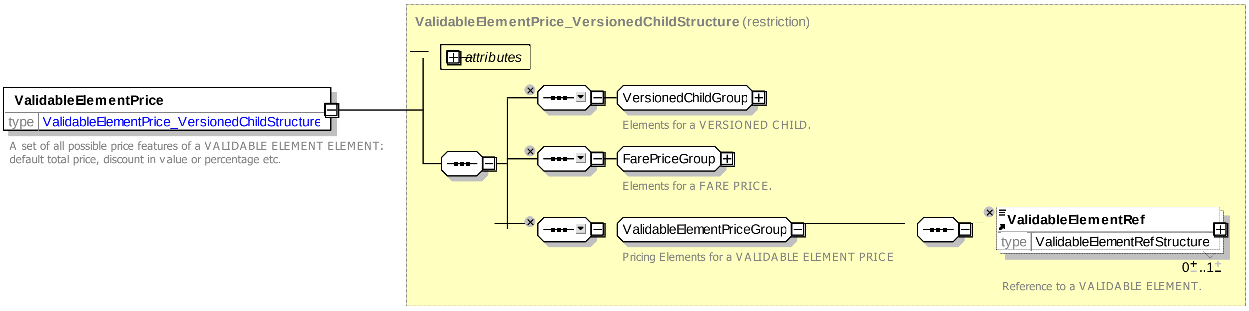


Figure 124 — *ValidableElementPrice* — XSD

7.5.8.3.3 ControllableElement – Model Element

The smallest controllable element of public transport consumption, all along which any VALIDITY PARAMETER ASSIGNMENT remains valid.

Table 89 – *ControllableElement* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PriceableObject*	::>	CONTROLLABLE ELEMENT inherits from PRICEABLE OBJECT.
«PK»	***id***	*ControllableElementIdType*	1:1	Identifier of CONTROLLABLE ELEMENT.
«cntd»	***accessRight-Parameter-Assignments***	*Access¬RightParameter-Assignment+*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs associated with CONTROLLABLE ELEMENT.
«cntd»	***controllable-Elements-InSequence***	*ControllableElement-InSequence*	0:*	CONTROLLABLE ELEMENTs IN SEQUENCE associated with CONTROLLABLE ELEMENT.
«cntd»	***prices***	*ControllableElementPrice*	0:*	CONTROLLABLE ELEMENT PRICEs for ELEMENT.

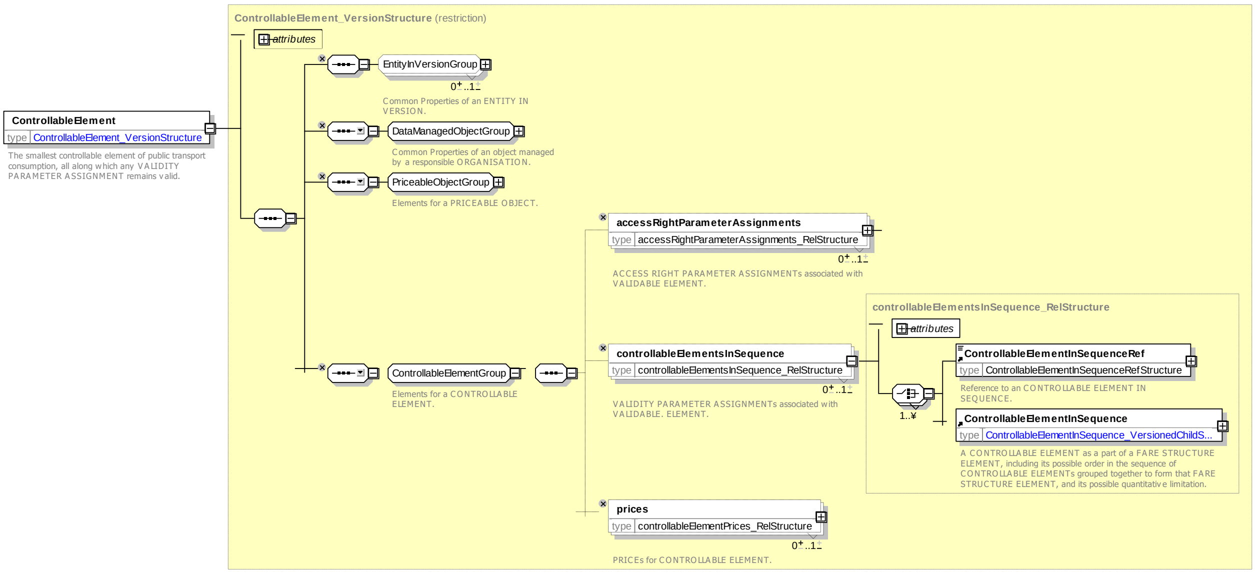


Figure 125 — *ControllableElement* — XSD

7.5.8.3.4 ControllableElementInSequence – Model Element

A CONTROLLABLE ELEMENT as a part of a FARE STRUCTURE ELEMENT, including its possible order in the sequence of CONTROLLABLE ELEMENTs grouped together to form that FARE STRUCTURE ELEMENT, and its possible quantitative limitation.

Table 90 – *ControllableElementInSequence* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*FareElementInSequence*	::>	CONTROLLABLE ELEMENT IN SEQUENCE inherits from FARE ELEMENT IN SEQUENCE.
«PK»	***id***	*ControllableElement-InSequenceIdType*	1:1	Identifier of CONTROLLABLE ELEMENT IN SEQUENCE.
«FK»	***Controllable-ElementRef***	*ControllableElementRef*	1:1	Reference to a CONTROLLABLE ELEMENT.
«FK»	***FareStructure-ElementRef***	*FareStructureElementRef*	0:1	Reference to a FARE STRUCTURE ELEMENT.
«cntd»	***accessRight-Parameter-Assignments***	*Access¬RightParameter-Assignment+*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs associated with CONTROLLABLE ELEMENT IN SEQUENCE.

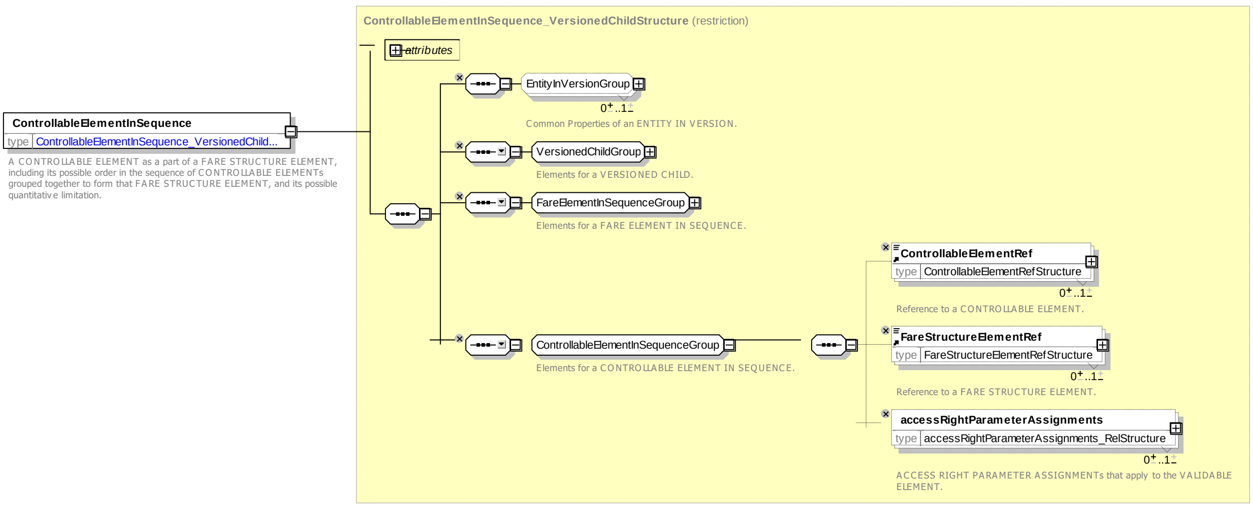


Figure 126 — *ControllableElementInSequence* — XSD

7.5.8.3.5 ControllableElementPrice – Model Element

A set of all possible price features of a CONTROLLABLE ELEMENT: default total price, discount in value or percentage etc.

Table 91 – *ControllableElementPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	CONTROLLABLE ELEMENT PRICE inherits from FARE PRICEs
«PK»	***id***	*ControllableElement-PriceIdType*	1:1	Identifier of CONTROLLABLE ELEMENT PRICE.
«FK»	***Controllable-ElementRef***	*ControllableElement-IdType*	0:1	Reference to a CONTROLLABLE ELEMENT. If not given by context, must be specified.

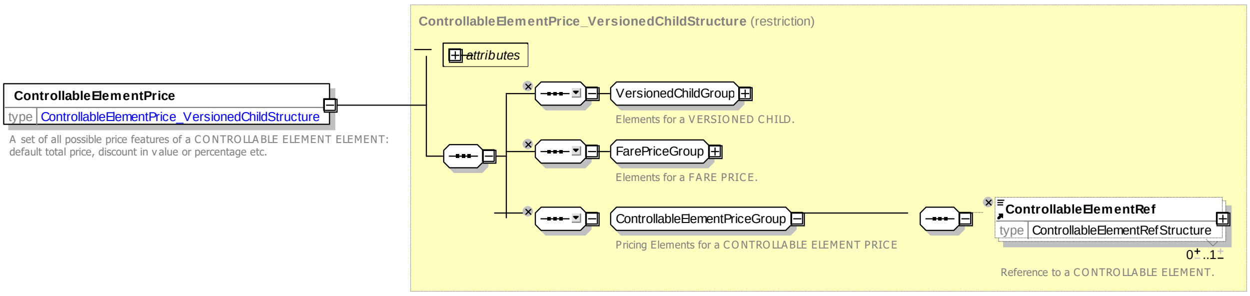


Figure 127 — *ControllableElementPrice* — XSD

7.5.8.4 Validable & Controllable Elements – XML examples

7.5.8.4.1 Validable & Controllable Elements: XML Example of a Tariff for a sequence of Fare Structure Elements

The following code fragment shows some basic VALIDABLE ELEMENTs for different modes.

For EXAMPLE:

```
<validableElements>
    <ValidableElement version="any" id="lul:metro_trip">
        <Name>A metro Trip</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="lul:metro_trip">
                <ValidityParameters><VehicleModes>metro</VehicleModes></ValidityParameters>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </ValidableElement>
    <ValidableElement version="any" id="lul::metro_part_trip">
        <Name>A metro Trip between validators - use Pink reader</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="lul:: metro_part_trip">
                <ValidityParameters><VehicleModes>metro</VehicleModes></ValidityParameters>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </ValidableElement>

    <ValidableElement version="any" id="nr:suburban_rail_trip">
        <Name>A rail Trip</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="nr: suburban_rail_trip">
                <ValidityParameters>
                    <VehicleModes>rail</VehicleModes>
                        <TransportSubmode>
                            <RailSubmode>local</RailSubmode>
```

```
                    </TransportSubmode>
                </ValidityParameters>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </ValidableElement>
```

7.6 Access Rights Description

The following diagram shows the models making up the Access Right Description submodel.

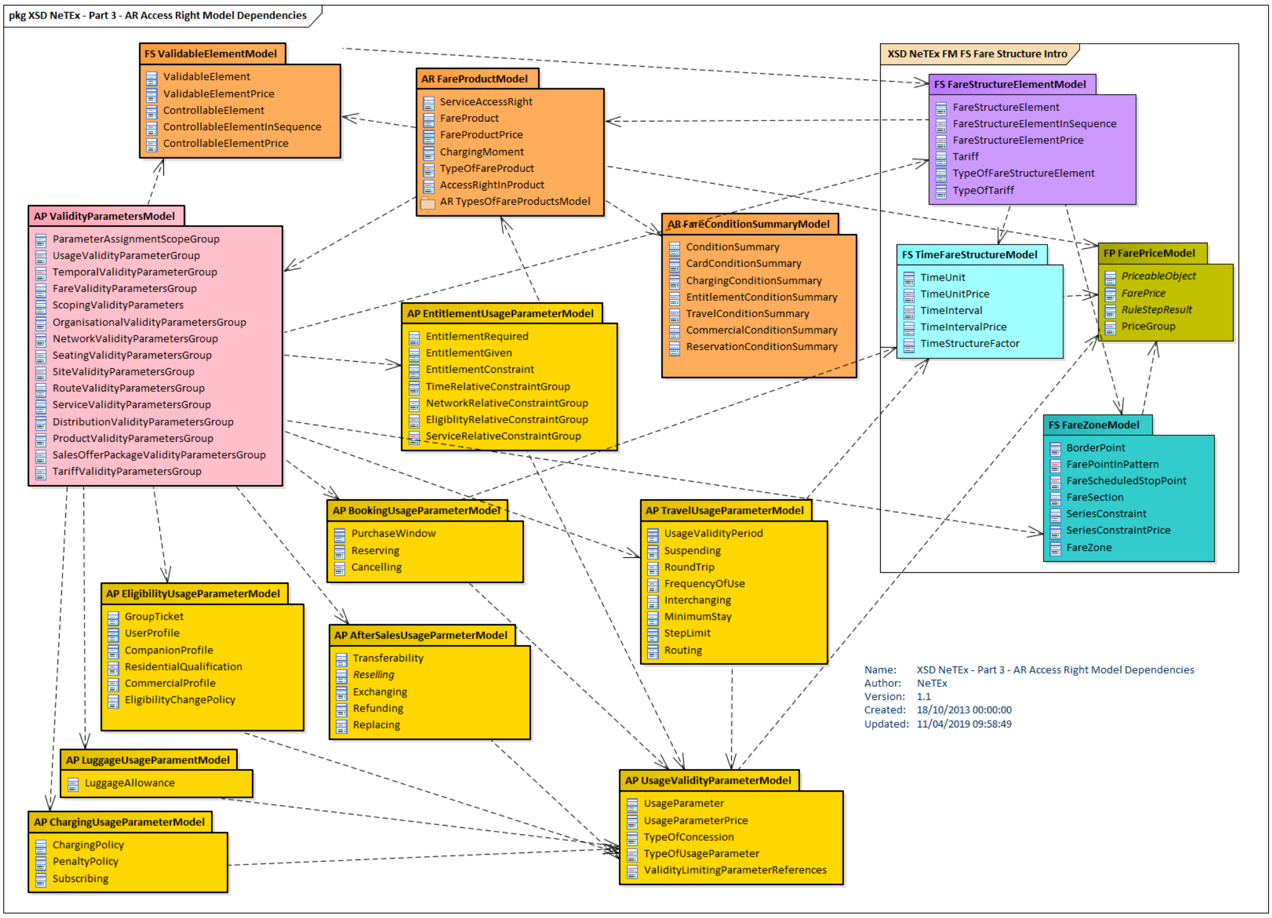


Figure 128 — Access Right Parameters – Model Dependencies

7.6.1 Access Right Parameters

7.6.1.1 Access Rights Parameters

7.6.1.1.1 ACCESS RIGHTS PARAMETERS – Conceptual MODEL

The ACCESS RIGHT PARAMETERs Model allows specific ACCESS RIGHTs to be associated with fare structure elements using a variety of validity parameters. The core element is the ACCESS RIGHT PARAMETER ASSIGNMENT which assigns a set of limiting parameters; this may be grouped with other ACCESS RIGHT PARAMETER ASSIGNMENTs and a logical operator to create complex combinations of conditions which can then be associated with a FARE STRUCTURE ELEMENT, DISTANCE MATRIX ELEMENT, GROUP OF DISTANCE MATRIX ELEMENTS, FARE PRODUCT, SALES OFFER PACKAGE, VALIDABLE ELEMENT, or CONTROLLABLE ELEMENT.

7.6.1.2 Validity Parameters and their Assignment

Apart from the quantitative parameters used in the fare structure such as time intervals and distance, other parameters may be used by a fare system in order to limit the validity of particular access rights. There may be more than one *theoretical* access right making up a fare structure as alternative possibilities – only a selection of which will be *consumed* in the specification of a particular trip. The definition of theoretical access rights or consumption controls, for instance, uses parameters referring to physical characteristics of the network

A number of parameters may be used by several functions of the fare system. For instance, a SCHEDULED STOP POINT may be used:

- to define a category of FARE STRUCTURE ELEMENTs corresponding to possible trips starting from the "central station" SCHEDULED STOP POINT;

- while using a distance-based fare structure, to specify the origin and destination of the intended trip when boarding the vehicle.

The processes that consist of assigning a fare parameter (e.g. a particular SCHEDULED STOP POINT) to either a theoretical or consumed access right are very similar. The assignment of such parameters to an element of the fare system is therefore described using a generic entity ACCESS RIGHT PARAMETER ASSIGNMENT, which may comprise one or more VALIDITY PARAMETER ASSIGNMENTS.

A VALIDITY PARAMETER ASSIGNMENT is used to specify a parameter limiting a theoretical access right (e.g. a TIME BAND limiting the validity of a possible trip). It has two specializations:

- A GENERIC PARAMETER ASSIGNMENT, which attaches a fixed parameter to a certain class of rights, denoting it as being theoretically allowed (possibly with multiple alternatives) within a given fare product and

- A SPECIFIC PARAMETER ASSIGNMENT, which assigns a limiting parameter to a particular right, within a certain fare structure, thus representing the choice of a specific set of parameters for consumption on an individual trip.

An ACCESS RIGHT PARAMETER ASSIGNMENT typically compares a parameter value to a characteristic of the related object. The attribute 'Assignment Type' allows for such a comparison. There are different types of possible comparisons, specified by the attribute assignment type, whose values are a comparison operator ('GT', 'EQ', 'LT', etc). They express that the compared characteristic, e.g.:

- 'EQ' is strictly equal to the parameter ("en route" parameter), e.g. the control mean is located at the STOP AREA "city centre", or the consumption should occur on LINE "27". This includes the case where a trip is specified as passing "via" a certain point.

- 'NE' is different than a certain value, e.g. in order to represent the rule 'the access right is valid on all bus network LINEs except for LINE 278 and LINE 66' or 'the access right to zone 4 is not valid between "2 a.m. – 4 a.m.'

- 'GE' is greater than or equal to the parameter ("start" parameter), e.g. the consumption has to end after "11.00 p.m."; the parameter must generally be monotonic for such a comparison to be meaningful. A further more complex semantic may be assumed for use with SCHEDULED STOP POINT; via points for the stop and subsequent stops within a vehicle journey, e.g. the consumption should start from the SCHEDULED STOP POINT "Central station" or from any further SCHEDULED STOP POINT in the JOURNEY PATTERN.

- 'LE' is equal or smaller than the parameter ("end" parameter), e.g. the consumption has to end before "11.00 p.m.". As above values must be monotonic.

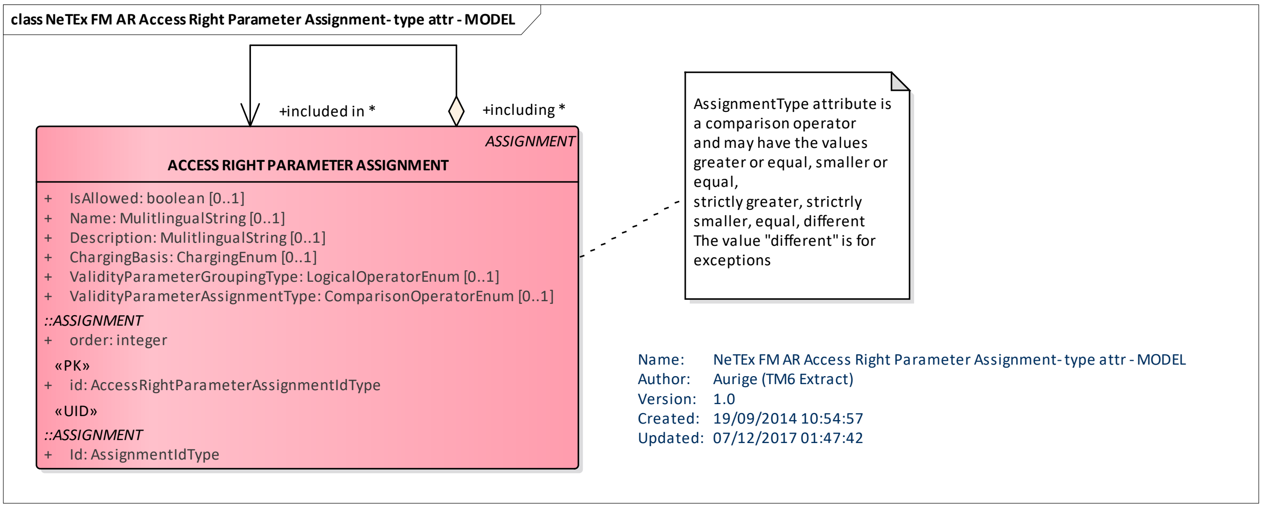


Figure 129 — ACCESS RIGHT ASSIGNMENTs Type (UML)

Access right limitation rules may be complex and involve several combinations of parameters and conditions. These rules may be expressed as logical propositions with logical operators (and, or, exclusive or). This means that different types of combinations of groups of parameters have to be taken into account and that the ACCESS RIGHT PARAMETER ASSIGNMENT is a multiple assignment. For that purpose, the attribute 'GroupingType' is defined, that has the values of a logical operator (*AND, OR, XOR*; if the value is none, the assignment is simple, only one type of parameter is considered).

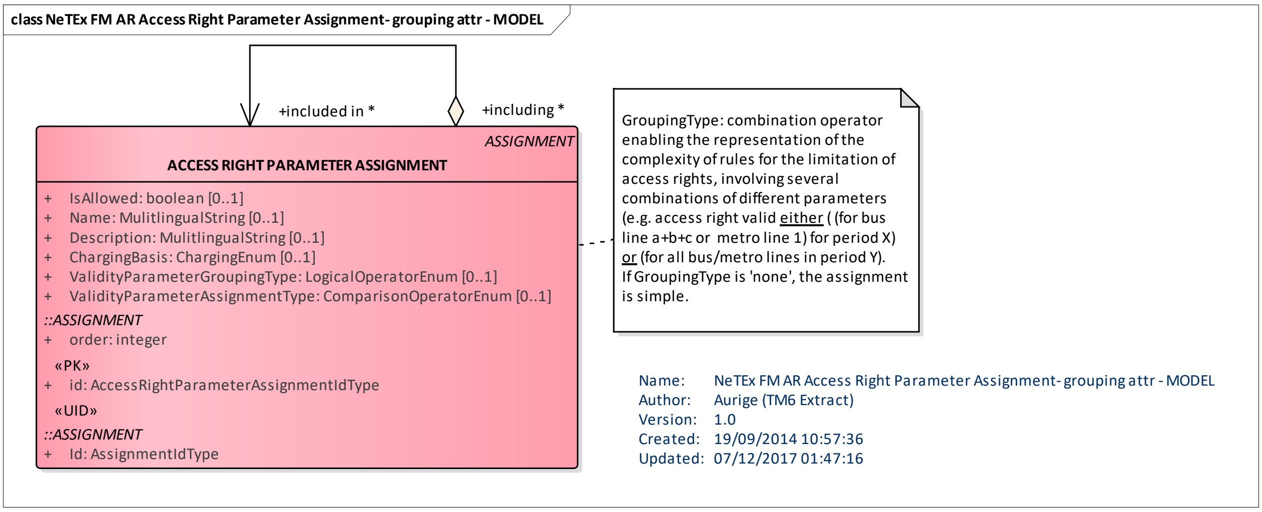


Figure 130 — ACCESS RIGHT ASSIGNMENT Grouping (UML)

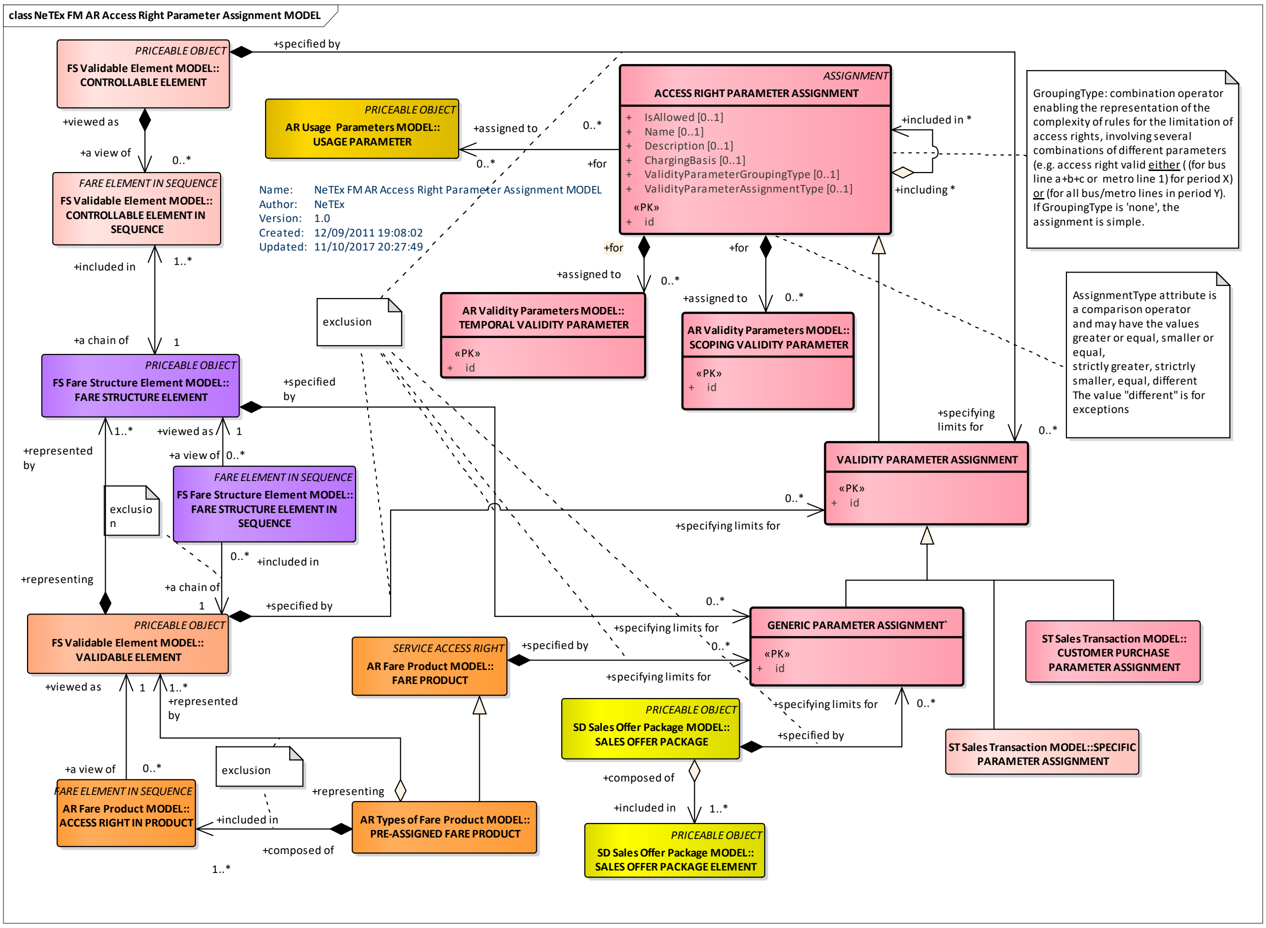


Figure 131 **—ACCESS RIGHT PARAMETER ASSIGNMENT – Conceptual MODEL**

The ACCESS RIGHT PARAMETER ASSIGNMENTs may involve the compound assignment of several assignments either ANDed or ORed together.

7.6.1.2.1 Generic Validity Parameters

The assignment of parameters to generic (theoretical) access rights is made through the GENERIC PARAMETER ASSIGNMENT entity, which is a sub-type of VALIDITY PARAMETER ASSIGNMENT. Such an assignment defines a limit attached to any practical instance of the concerned theoretical access right.

A VALIDITY PARAMETER ASSIGNMENT (therefore a GENERIC PARAMETER ASSIGNMENT) may assign a limiting or a usage parameter to either a CONTROLLABLE ELEMENT, a FARE STRUCTURE ELEMENT, a VALIDABLE ELEMENT, a FARE PRODUCT, or a SALES OFFER PACKAGE.

If the order of CONTROLLABLE ELEMENTs in a FARE STRUCTURE ELEMENT is not specified, which means that they may be consumed regardless the order, a limiting parameter may be attached to the first or the last element. This would use the corresponding attributes of the CONTROLLABLE ELEMENT IN SEQUENCE. For instance, if an access right allows several rides to reach a joint service, the consumption may be constrained to start during a specific TIME BAND. In such a case, a specific CONTROLLABLE ELEMENT will be created, describing the first possible ride, without any other specification than the assignment to the considered TIME BAND.

7.6.1.2.1.1 ACCESS RIGHTS PARAMETER ASSIGNMENT – Conceptual MODEL

The validity parameters are considered as being of two main types:

- TEMPORAL VALIDITY PARAMETERS, reflecting temporal limitations and
- SCOPING VALIDITY PARAMETERS, reflecting spatial and consumption limitations.

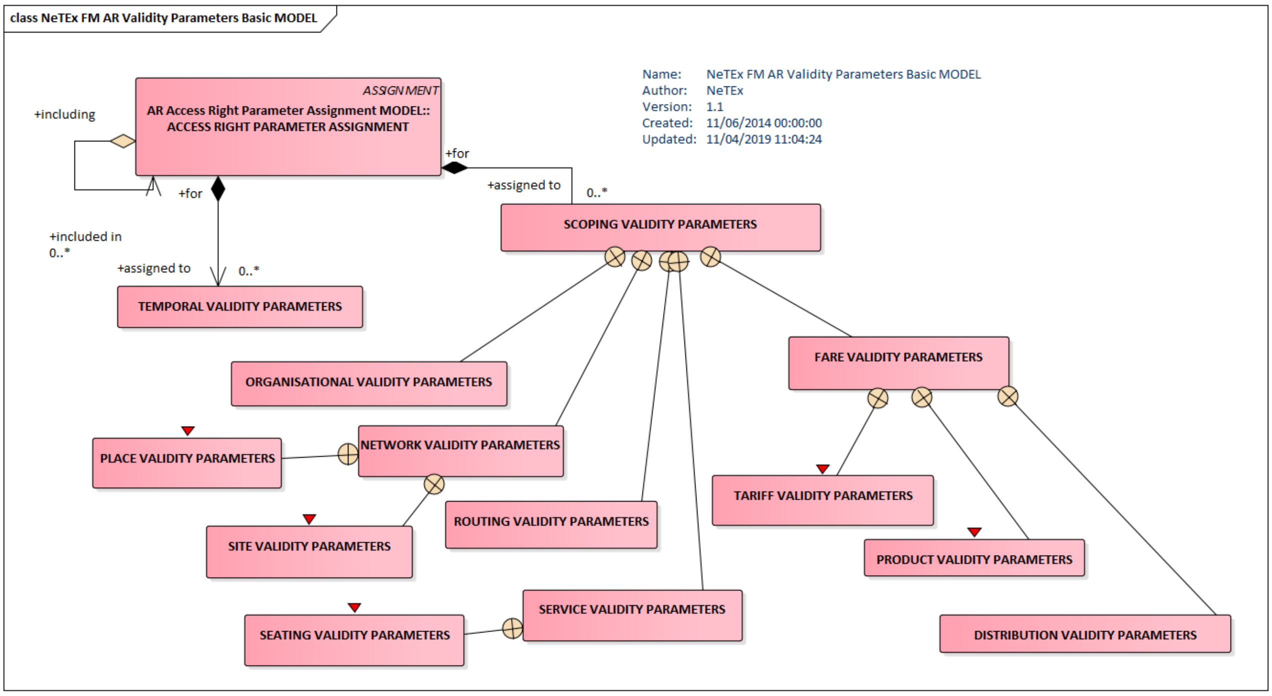


Figure 132 — Validity Parameters Basic – Conceptual MODEL (UML)

7.6.1.2.1.2 Validity Parameters – Conceptual MODEL

The SCOPING PARAMETERS in their turn may be further grouped as follows:

- ORGANISATIONAL VALIDITY PARAMETERS
- NETWORK VALIDITY PARAMETERS
  - SITE VALIDITY PARAMETERS
  - PLACE VALIDITY PARAMETERS
- ROUTING VALIDITY PARAMETERS
- SERVICE VALIDITY PARAMETERS
  - SEATING VALIDITY PARAMETERS
- FARE VALIDITY PARAMETERS.
  - TARIFF VALIDITY PARAMETERS.
  - PRODUCT VALIDITY PARAMETERS.
  - DISTRIBUTION VALIDITY PARAMETERS.

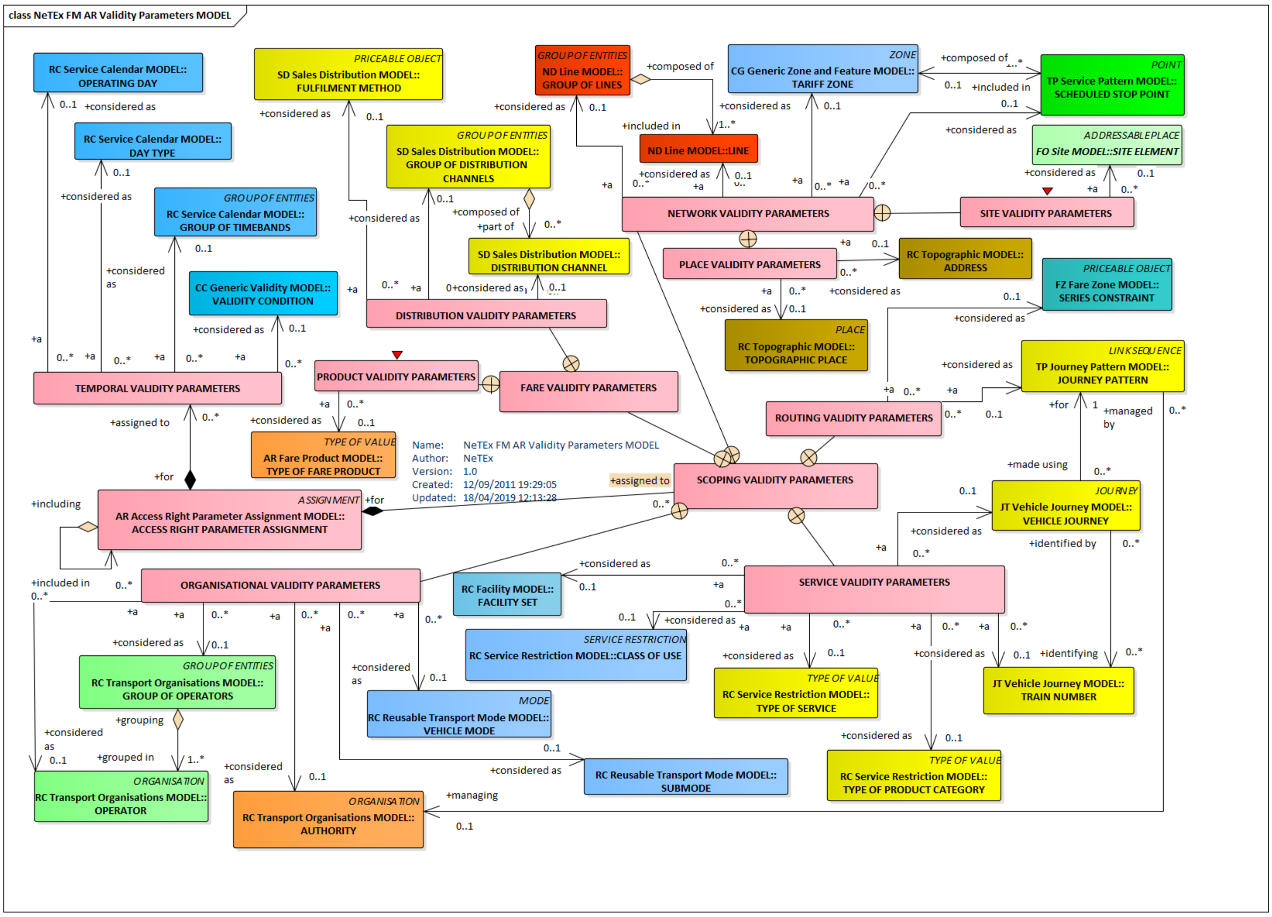


Figure 133 — Detailed Validity Parameters – Conceptual MODEL (UML)

7.6.1.2.2 Access Rights Parameters – Physical model

The following figure introduces the physical model for ACCESS RIGHT PARAMETER ASSIGNMENTs. An ACCESS RIGHT PARAMETER ASSIGNMENT can be used to associate one or more assignments with fare structure elements to specify the access rights to transport services, including; (a) which parts of the network and services may be used ("network validity parameters", "service validity parameters", "route validity parameters", "product validity parameters") (b) when they may be used ("temporal validity parameters") (c) The conditions and limitations on use (USAGE PARAMETERs).

The Fare Structure assignments may be made for an assignment "scope" of various combinations of FARE STRUCTURE ELEMENTs, FARE STRUCTURE ELEMENTs in SEQUENCE, DISTANCE MATRIX ELEMENTs, GROUP OF DISTANCE MATRIX ELEMENTs, VALIDABLE ELEMENTs, CONTROLLABLE ELEMENTs, FARE PRODUCTs and SALES OFFER PACKAGEs.

Each assignment may restrict the access rights to a particular temporal validity parameter (DAY TYPE, VALIDITY CONDITION, etc) and to validity parameters describing more access rights to the network or services (LINE, TARIFF ZONEs, SCHEDULED STOP POINTs, VEHICLE JOURNEYS, CLASS OF USE, TRAIN NUMBER, etc., etc.) using a comparison operator. (EQ, GT, LT etc., etc.).

Assignments may be combined using a logical operator (AND, OR, NOT, XOR) to create composite assignments.

There are several different types of ACCESS RIGHT ASSIGNMENT for use in specific circumstances.

— A GENERIC PARAMETER ASSIGNMENT is used in a FARE PRODUCT and SALES OFFER PACKAGE to indicate a theoretical set of possible limitation parameters that apply to a FARE STRUCTURE ELEMENT or PRODUCT

— A SPECIFIC PARAMETER ASSIGNMENT is used in a TRAVEL SPECIFICATION to indicate the specific choice of parameters consumed in an actual materialization of a fare product as a SALES TRANSACTION.

— DEVICE and CONTROL PARAMETER ASSIGNMENTs are used in fare collection and validation systems.

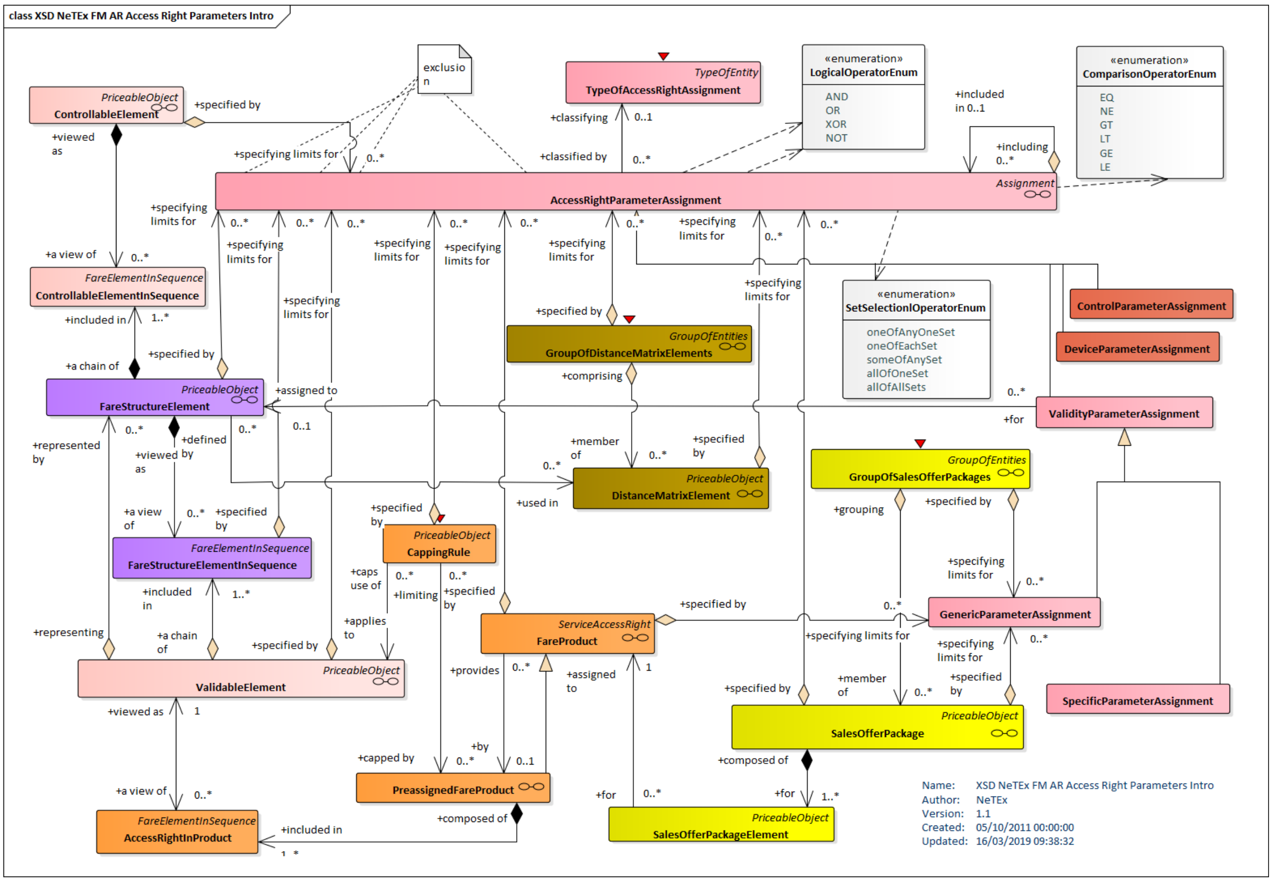


Figure 134 — Access Right Parameters – Physical Model (UML)

7.6.1.2.2.1 Access Rights Parameters: Detail – Physical model

The following figure summarises the basic physical model for ACCESS RIGHT PARAMETER ASSIGNMENTs including the scope elements. It omits the validity parameter details (see later below).

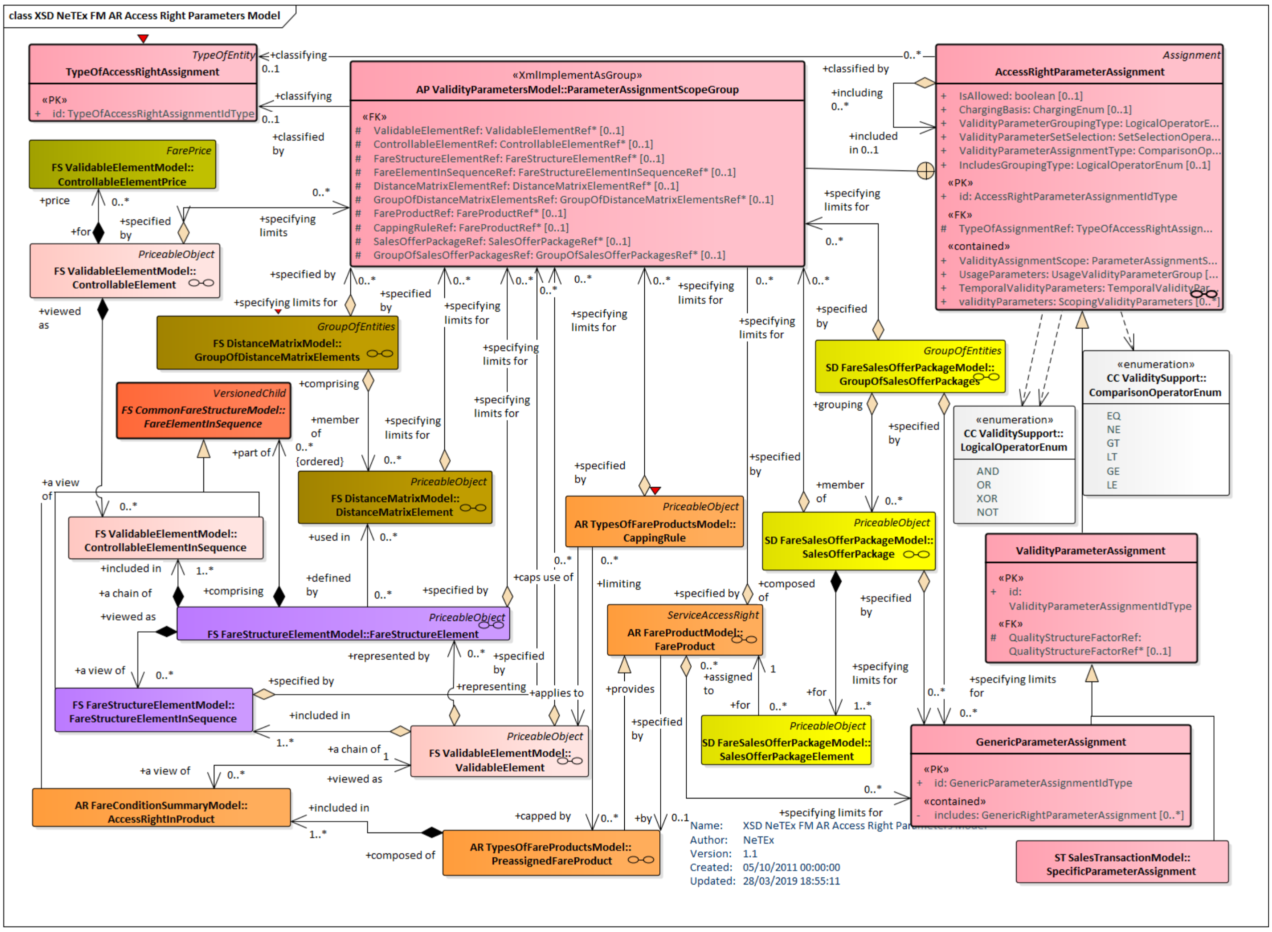


Figure 135 — Access Right Scope Parameters: Detail – Physical Model

7.6.1.2.2.2 Access Rights Parameters: Validity Parameters Summary – Physical model

The following figure summarises the validity parameters which may be assigned using an ACCESS RIGHT PARAMETER ASSIGNMENT. These include both temporal validity parameters (DAY TYPE, VALIDITY CONDITION etc) and other validity parameters that relate to the network, routing and services etc (CLASS OF USE, Network, OPERATOR, SCHEDULED STOP POINT, SERIES CONSTRAINT, etc., etc.).

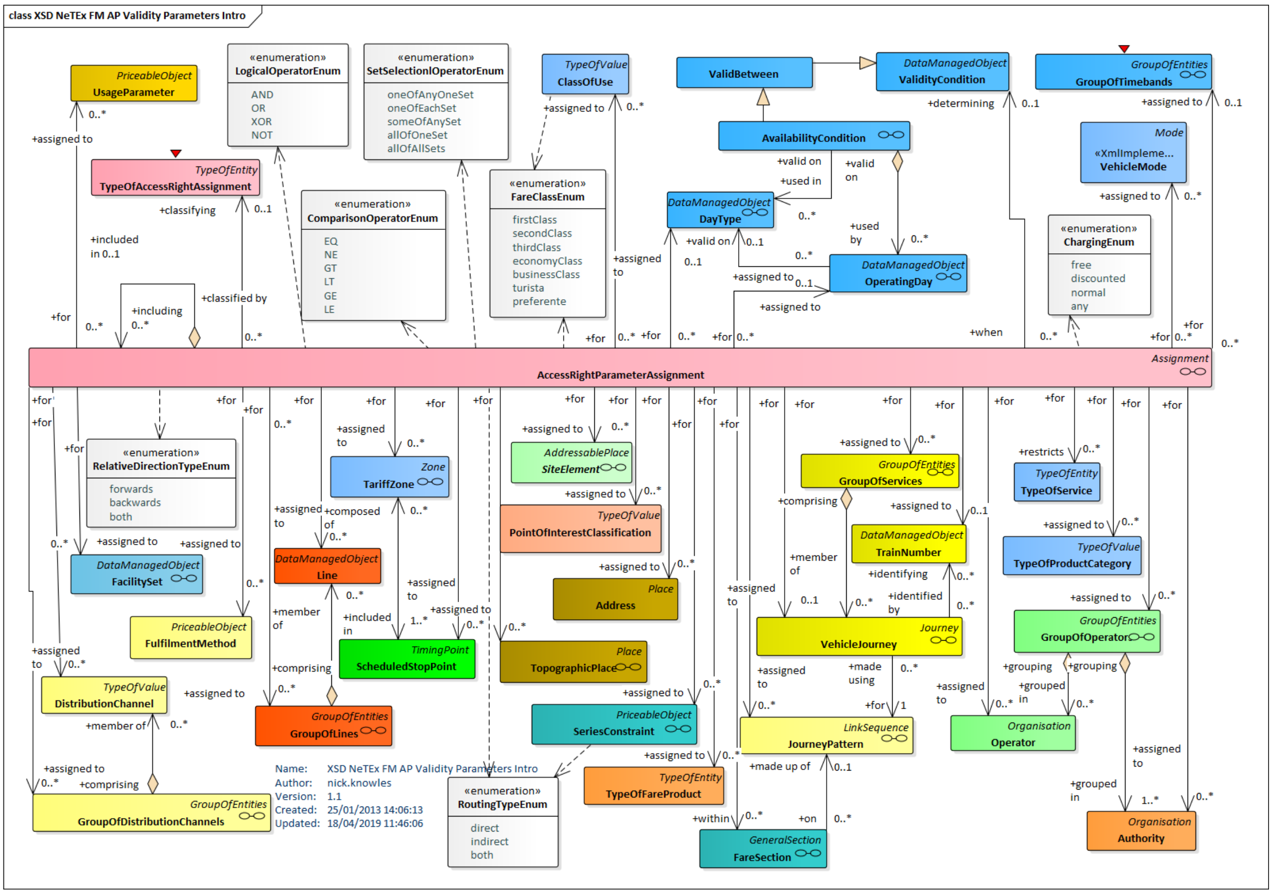


Figure 136 — Access Right Validity Parameters – Physical Model (UML)

7.6.1.2.2.3 Access Rights Parameters: Scoping Validity Parameters – Physical model

The following figure shows the main validity parameters of an ACCESS RIGHT PARAMETER ASSIGNMENT that may be used to restrict access rights for fare structure elements to specific elements of the network.

Organisation related parameters:

- Which OPERATORs or GROUPs of OPERATORs may be used.
- Which VEHICLE MODEs and submodes may be used.

Network related parameters:

- Which LINEs, GROUPs OF LINEs or NETWORKs, and VEHICLE MODEs may be used.
- Which TARIFF ZONEs, FARE SECTIONs, and which SCHEDULED STOP POINTs may be used. Also where a right is limited to part of a physical location this may be specified using a SITE ELEMENT.
- Which POINTs of INTEREST may be accessed.

Service related parameters:

- Which SERIES CONSTRAINTs on the routings must be followed.

- The specific TRAIN NUMBER, VEHICLE JOURNEY, JOURNEY PATTERN, TYPE OF PRODUCT CATEGORY (e.g. *ICE*, *Thalys* etc) of services which may be used.

- Which CLASS of USE and TYPE OF PRODUCT CATEGORY may be used.

SITE related parameters: see later below.

SEATING related parameters: see later below.

Fare related parameters: see later below

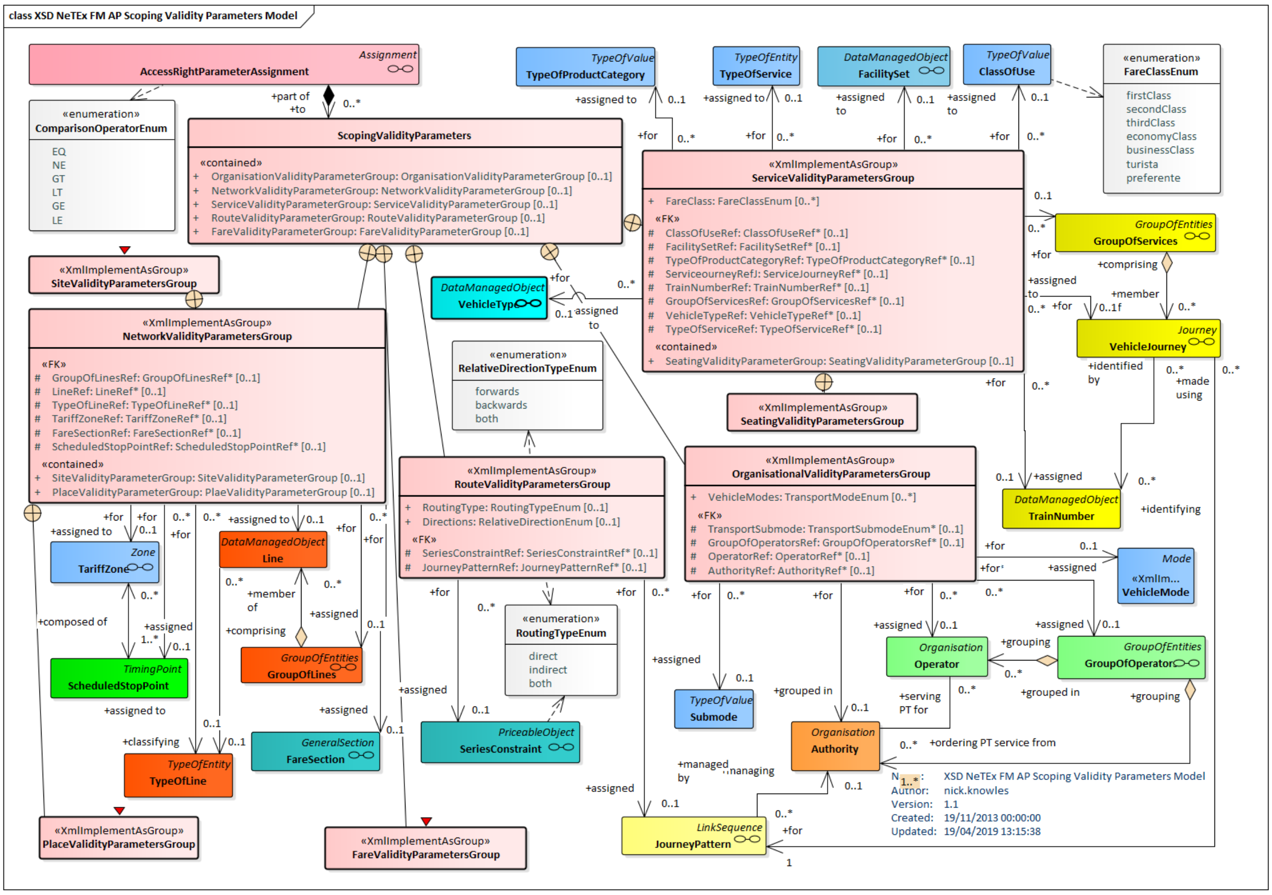


Figure 137 — Access Right Parameters: Scoping Validity Parameters – Physical Model (UML)

7.6.1.2.2.4 Access Rights Parameters: Temporal Validity Parameters – Physical model

The following figure shows the temporal validity parameters of an ACCESS RIGHT PARAMETER ASSIGNMENT which can be used to restrict when an assignment applies. These use general purpose elements described in detail in NeTEx PART1 including:

- The DAY TYPE or OPERATING DAY on which the assignment applies.
- The TIMEBANDs during which the assignment applies.
- The OPERATING PERIODs during which the assignment applies.
- The VALIDITY CONDITION or AVAILABILITY CONDITION restricting the assignment.

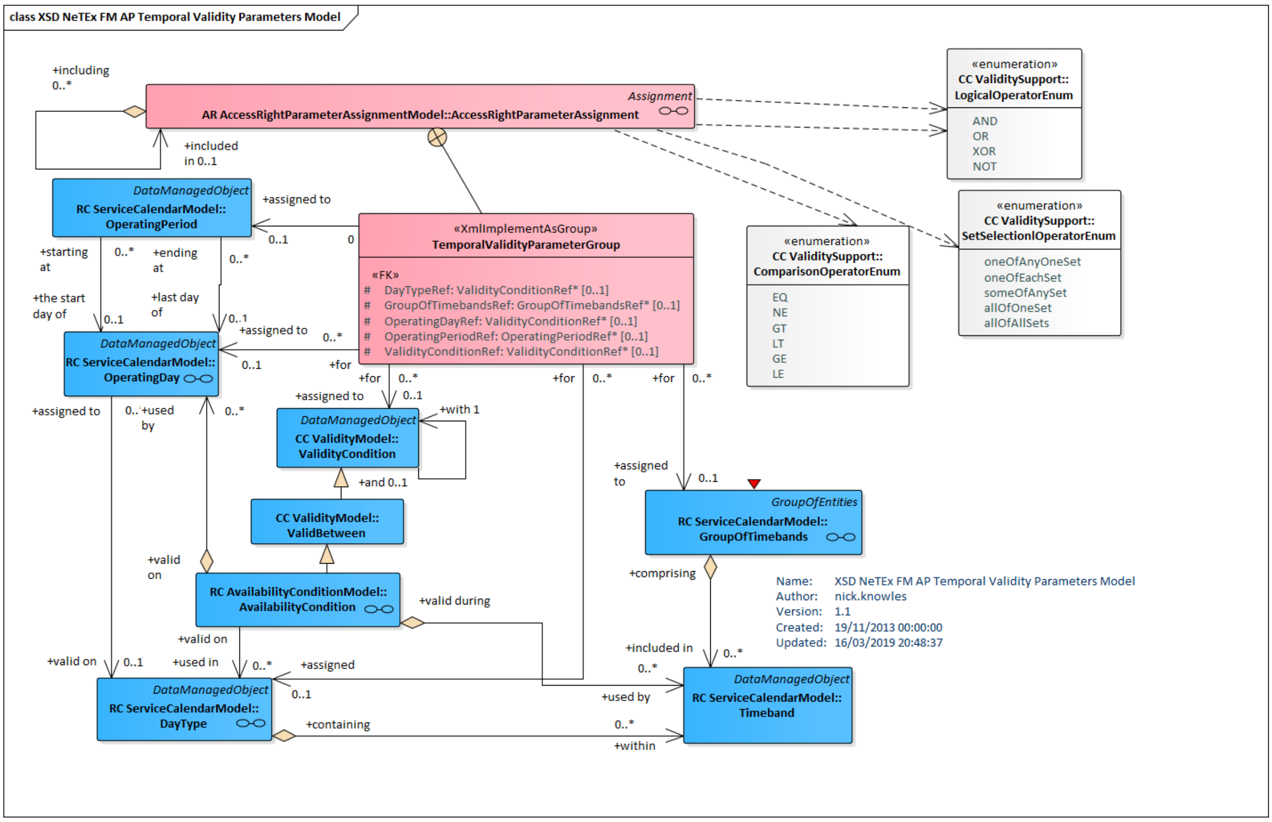


Figure 138 — Access Right Parameters: Temporal Validity Parameters – Physical Model (UML)

7.6.1.2.2.5 Access Rights Parameters: Site Scoping Validity Parameters – Physical model

The following figure shows the SITE validity parameters of an ACCESS RIGHT PARAMETER ASSIGNMENT that may be used to restrict access rights for fare structure elements to specific elements of the network.

- The STOP PLACE, PARKING or POINT OF INTEREST to which the assignment applies.
- The ADDRESS to which the assignment applies.
- The TOPOGRAPHIC PLACE to which the assignment applies.

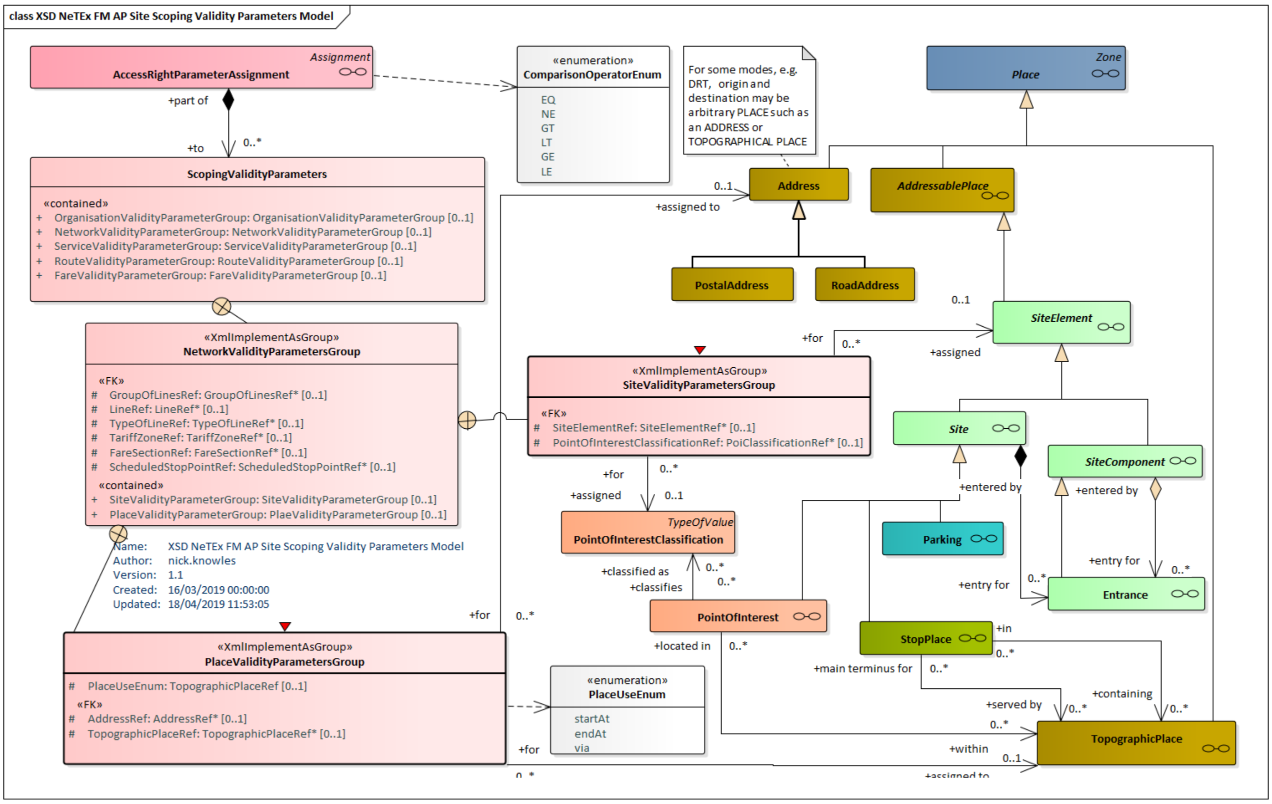


Figure 139 — Access Right Parameters: Site Scoping Validity Parameters – Physical Model (UML)

7.6.1.2.2.6 Access Rights Parameters: Seating Scoping Validity Parameters – Physical model

The following figure shows the PASSENGER SEATING validity parameters of an ACCESS RIGHT PARAMETER ASSIGNMENT that may be used to indicate access rights to a specific seat, including:

- The VEHICLE JOURNEY and TRAIN NUMBER to which the assignment applies.
- The FACILITY SET and ACCOMMODATION type (Seat, couchette, etc) to which the assignment applies.
- The PASSENGER SEAT to which the assignment applies.

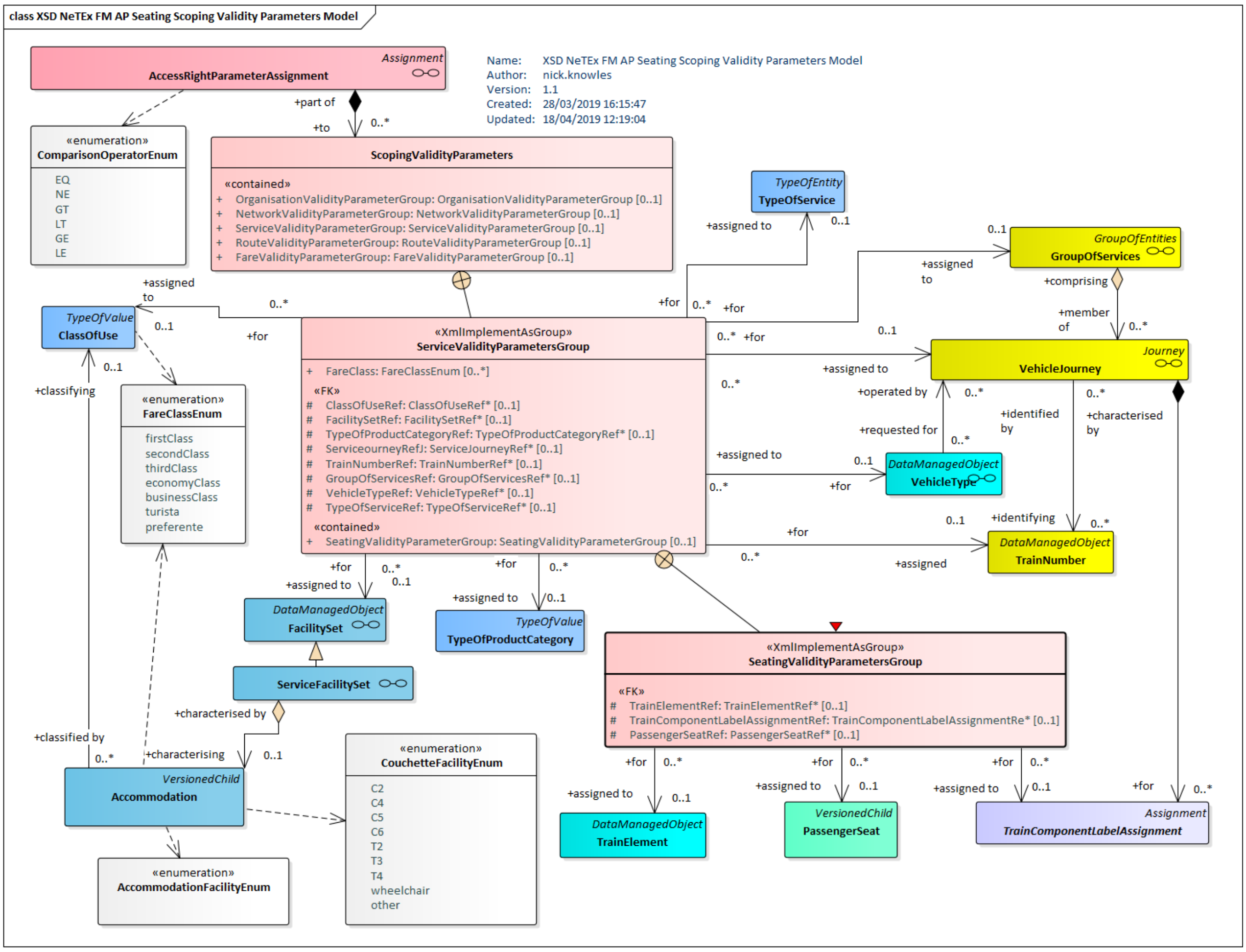


Figure 140 — Access Right Parameters: Seating Scoping Validity Parameters – Physical Model (UML)

7.6.1.2.2.7 Access Rights Parameters: Fare Scoping Validity Parameters – Physical model

The following figure shows the Fare validity parameters of an ACCESS RIGHT PARAMETER ASSIGNMENT that may be used to specify conditions attaching to specific elements of a fare product.

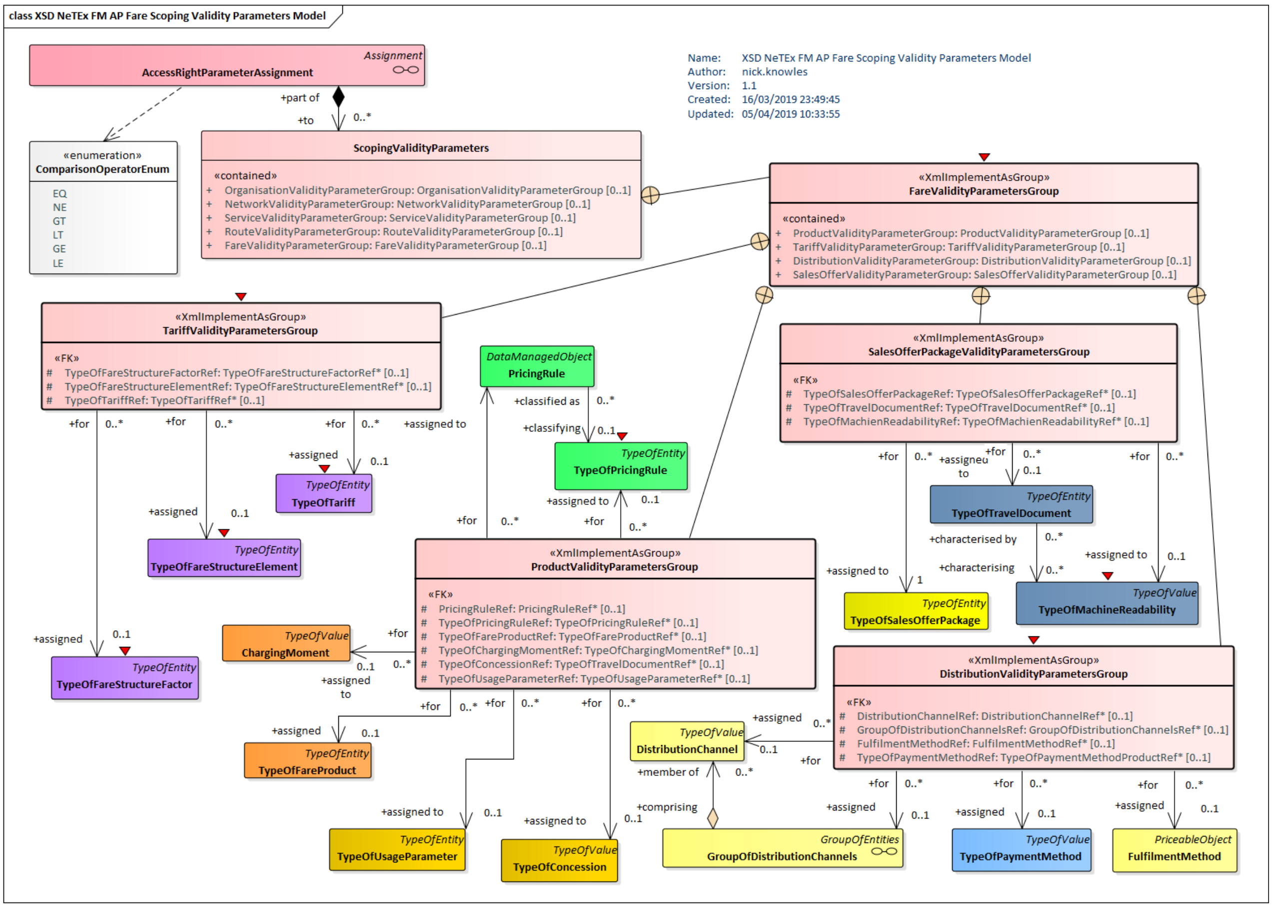


Figure 141 — Access Right Parameters: Fare Scoping Validity Parameters – Physical Model (UML)

7.6.1.2.2.8 Access Rights Parameters: Usage Validity Parameters – Physical model

The following figure shows the limitation i.e. USAGE PARAMETERs of an ACCESS RIGHT PARAMETER ASSIGNMENT which can be used to restrict. An operator can be used to specify how the values are combined.

For a detailed modelling of USAGE PARAMETERs see later below.0

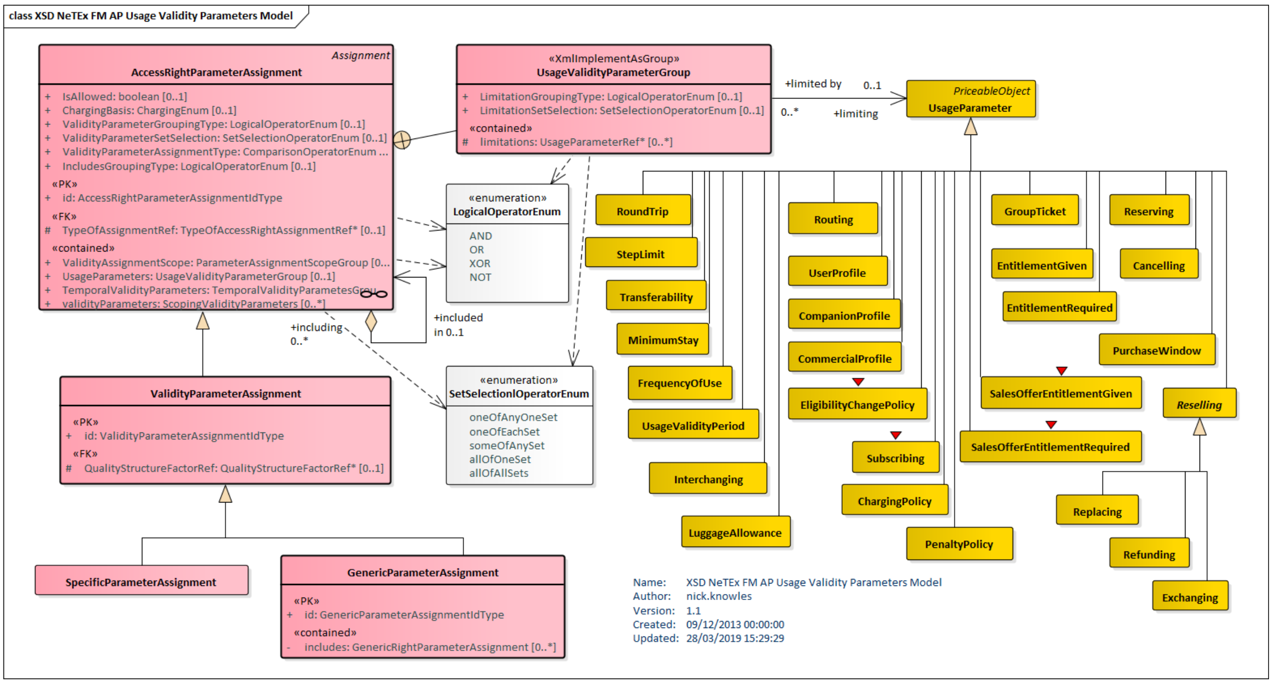


Figure 142 — Access Right Parameters: Usage Validity Parameters – Physical Model (UML)

7.6.1.2.3 Access Rights Parameters – Attributes and XSD

7.6.1.2.3.1 AccessRightParameterAssignment – Model Element

The assignment of a fare collection parameter (referring to geography, time, quality or usage) to an element of a fare system (access right, validated access, control mean, etc.).

Table 92 – *AccessRightParameterAssignment* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*Assignment*	::>	ACCESS RIGHT PARAMETER ASSIGNMENT inherits from ASSIGNMENT.
«PK»	***id***	*AccessRight-Parameter-AssignmentIdType*	1:1	Identifier of ACCESS RIGHT PARAMETER ASSIGNMENT.
XGRP	***AccessRightParameter-AssignmentProperties-Group***	***xmlGroup***	1:1	General properties of an ACCESS RIGHT PARAMETER ASSIGNMENT.

XGRP	***ParameterAssignment-ScopeGroup***	***xmlGroup***	1:1	The FARE STRUCTURE elements to which the ACCESS RIGHT PARAMETER ASSIGNMENT is made.
XGRP	***Usage-Validity-ParameterGroup***	***xmlGroup***	1:1	USAGE PARAMETERs by which ACCESS RIGHT PARAMETER ASSIGNMENT is limited.
XGRP	***AccessRightParameter-AssignmentValidity-ParameterGroup***	***xmlGroup***	1:1	Access right validity parameters by ACCESS RIGHT PARAMETER ASSIGNMENT is restricted.
XGRP	***AccessRight Parameter-Assignment-Includes-Group***	***xmlGroup***	1:1	Elements used to create composite ACCESS RIGHT PARAMETER ASSIGNMENTs.

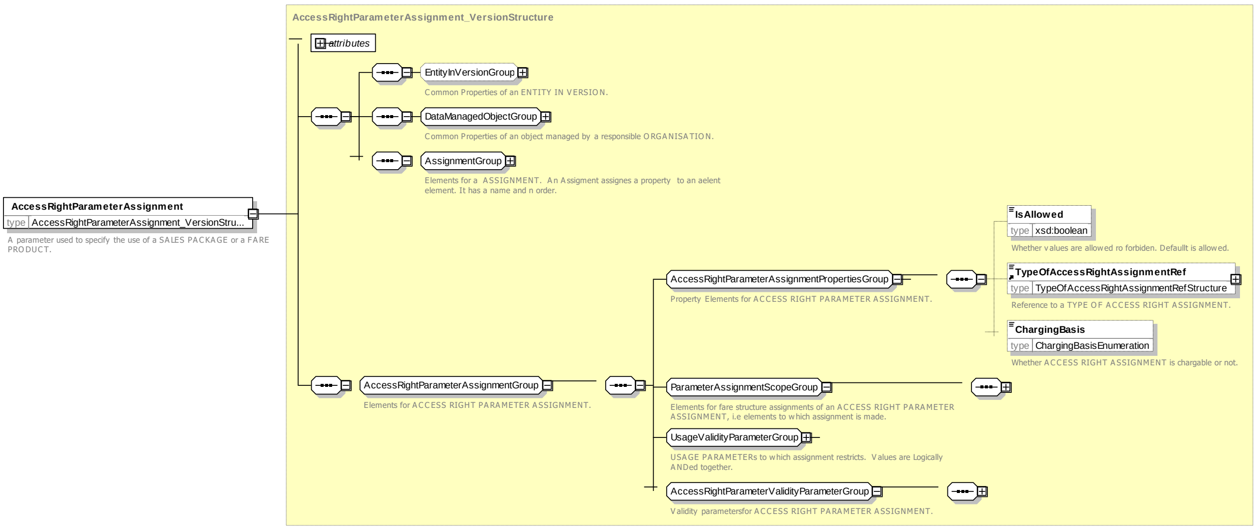


Figure 143 **— *AccessRightParameterAssignment* — XSD**

7.6.1.2.3.2 AccessRightParameterAssignmentPropertiesGroup – Group

The ***AccessRightParameterAssignmentPropertiesGroup*** define**s** basic properties an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 93 – *AccessRightParameterAssignmentPropertiesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	IsAllowed	*xsd:boolean*	0:1	Whether the specified assignments are allowed (true) or not (false).
«FK»	***TypeOf-AssignmentRef***	*TypeOAccessRight-AssignmentRef*	0:1	Classification of ACCESS RIGHT PARAMETER ASSIGNMENT.
«enum»	***ChargingBasis***	*ChargingBasisEnum*	0:1	Whether the specified assignment is for charged access, discounted access or free access. See allowed values below.

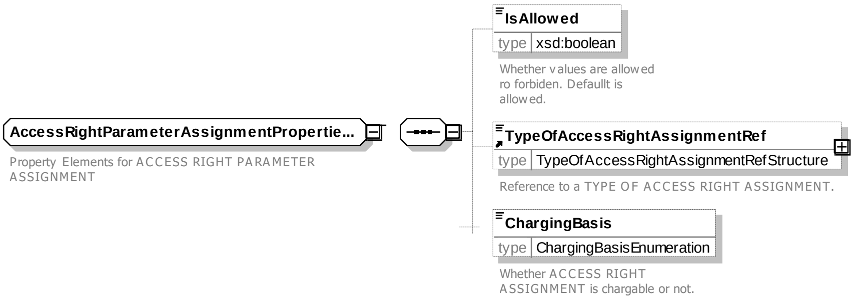


Figure 144 — *AccessRightParameterAssignmentPropertiesGroup* — XSD

7.6.1.2.3.2.1 ChargingBasis – Allowed values

The following table shows the allowed values for ***ChargingBasis*** (*ChargingBasisEnumeration*).

Table 94 – *ChargingBasis* – Allowed values

Value	Description
free	Use is Free.
discounted	Use is discounted.
normal	Use is charged normal fares
any	Use may be free, normal fare or discounted.

7.6.1.2.3.3 ParameterAssignmentScopeGroup – Group

The ***ParameterAssignmentScopeGroup*** species the fare structure elements to which an assignment is made by an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 95 – *ParameterAssignmentScopeGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***Validable-ElementRef***	*ValidableElementRef*	0:1	VALIDABLE ELEMENT to which assignment is made.
«FK»	***Controllable-ElementRef***	*ControllableElementRef*	0:1	CONTROLLABLE ELEMENT to which assignment is made.
«FK»	***FareProductRef***	*FareProductRef+*	0:1	FARE PRODUCT to which assignment is made.
«FK»	***TariffRef***	*TariffRef*	0:1	TARIFF to which assignment is made.
«FK»	***FareStructure-ElementRef***	*FareStructureElementRef*	0:1	FARE STRUCTURE ELEMENT to which assignment is made.
«FK»	***FareStructure-Element-InSequenceRef***	*FareStructure-Element-InSequenceRef*	0:1	FARE STRUCTURE ELEMENT IN SEQUENCE to which assignment is made.
«FK»	***Distance-MatrixElementRef***	*DistanceMatrixRef*	0:1	DISTANCE MATRIX ELEMENT to which ACCESS RIGHT PARAMETER is assigned.

«FK»	***Distance-MatrixInverseRef***	*DistanceMatrixRef*	0:1	DISTANCE MATRIX ELEMENT to which ACCESS RIGHT PARAMETER is assigned; reference is I inverse sense to that of element.
«FK»	***Distance-MatrixInverse-View***	*DistanceMatrixView*	0:1	VIEW of DISTANCE MATRIX ELEMENT to which ACCESS RIGHT PARAMETER is assigned, View includes details of origin and destination
«FK»	***SalesOffer-PackageRef***	*SalesOfferPackageRef*	0:1	SALES OFFER PACKAGE to which assignment is made.
«FK»	***GroupOf-DistanceMatrix-ElementsRef***	*GroupOfDistanceMatrix-ElementsRef*	0:1	GROUP OF DISTANCE MATRIX ELEMENTs to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***GroupOfSales-OfferPackages-Ref***	*GroupOfSalesOffer-PackagesRef*	0:1	GROUP OF SALES OFFER PACKAGEs to which assignment is made.

Figure 145 — *ParameterAssignmentScopeGroup* — XSD

7.6.1.2.3.4 UsageValidityParametersGroup – Group

The ***UsageValidityParametersGroup*** species the limitation conditions made by an ACCESS RIGHT PARAMETER ASSIGNMENT by reference one or more usage parameters.

Table 96 – *UsageValidityParameters*– Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***Limitations-GroupingType***	*BooleanOperatorEnum*	0:1	Logical operator for combining USAGE PARAMETERs elements. The default is '*AND*'. '*OR*' and '*XOR*' should only be used if parameters are all of the same type. See allowed values below.
«enum»	***Limitations-SetSelection-Type***	*SetOperatorEnum*	0:1	Where one or more parameter is a group containing multiple elements, (GROUP OF xxx), set operator for distinguishing between whole set and item interpretation of elements which are sets of elements. See allowed values below.
«FK»	***limitations***	*UsageParameterRef+*	0:*	References to USAGE PARAMETERs defining limitations made by ACCESS RIGHT PARAMETER ASSIGNMENT.

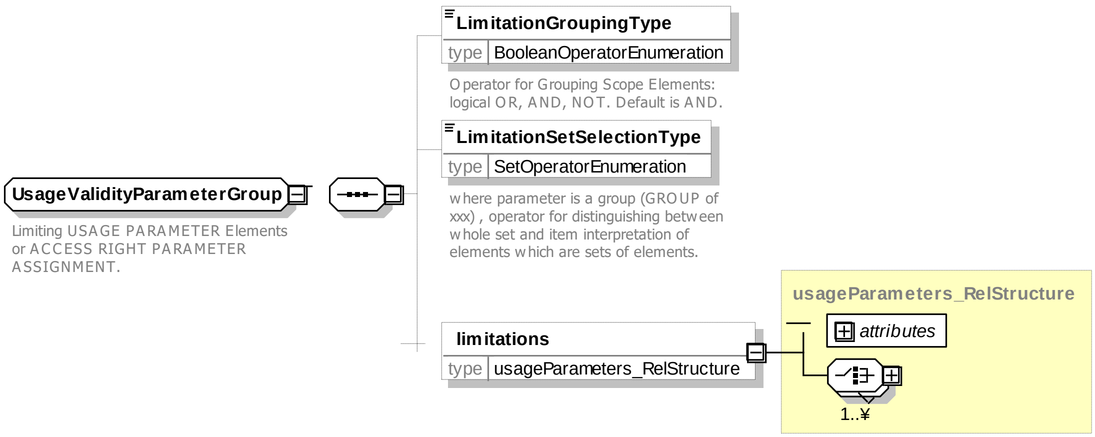


Figure 146 — *UsageValidityParameters*Group — XSD

7.6.1.2.3.5 AccessRightParameterValidityParameterGroup – Group

The ***AccessRightParameterValidityParameterGroup*** species the network and temporal access right restrictions made by an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 97 – *AccessRightParameter*ValidityParameter*Group* – Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***Validity-Parameter-AssignmentType***	*Comparison-OperatorEnum*	0:1	Comparison operator for matching validity parameter values. See allowed values below.
XGRP	***AccessRight-ScopeTemporal-Group***	***xmlGroup***	1:1	Time related properties assigned by ACCESS RIGHT PARAMETER ASSIGNMENT.

«enum»	***Validity-Parameter-GroupingType***	*BooleanOperatorEnum*	0:1	Logical operator for combining network validity parameters, e.g. '*AND*', '*OR*', '*XOR*'. See allowed values below.
«enum»	***Validity-SetSelection-Type***	*SetOperatorEnum*	0:1	Where one or more parameter is a group containing multiple elements, (GROUP OF xxx), set operator for distinguishing between whole set and item interpretation of elements which are sets of elements.
«cntd»	***temporalValidity-Parameters***	*TemporalValidity-ParametersGroup*	0:*	Temporal validity parameters assigned by ACCESS RIGHT PARAMETER ASSIGNMENT.
«cntd»	***validity-Parameters***	*LimitingValidity-ParametersGroup*	0:*	Validity parameters assigned by ACCESS RIGHT PARAMETER ASSIGNMENT.

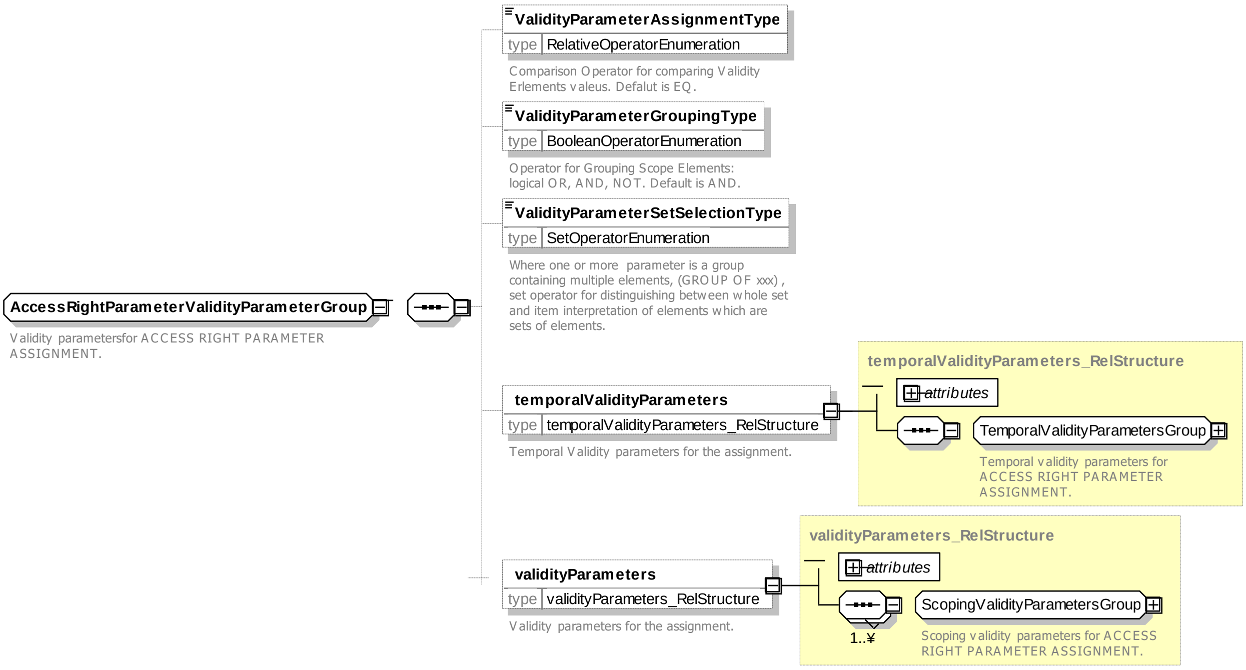


Figure 147 — *ValidityParameterAssignmentScopeGroup* — XSD

7.6.1.2.3.5.1 ValidityParameterAssignmentType – Allowed values

The following table shows the allowed values for ***ValdityParameterAssignmentType*** (*ComparisonOperatorEnumeration)*

Table 98 – *GroupingType* – Allowed values

Value	Description
EQ	Equal.
NE	Not equal.
GT	Greater than.
GE	Greater than or equal.
LT	Less than.
LE	Less than or equal.

7.6.1.2.3.5.2 ValidityParameterGroupingType – Allowed values

The following table shows the allowed values for ***ValidityParameterGroupingType*** (*BooleanOperator-Enumeration*)

Table 99 – ValidityParameterGroupingType – Allowed values

Value	Description
AND	Logical And
OR	Logical Or
XOR	Exclusive OR
NOT	Logical not

7.6.1.2.3.5.3 ValiditvParameterSetSelectionType – Allowed values

The following table shows the allowed values for ***ValidityParameterSetSelection*** (*BooleanOperator-Enumeration*)

Table 100 – ValidityParameterSetSelectionType – Allowed values

Value	Description
oneOfEachSet	One item from each specified referenced GROUP OF ENTITies of a given type must be selected/has been selected.
someOfAnySet	Multiple items from any referenced GROUP OF ENTITies of a given type may be selected/have been selected.
allOfOneSet	All items from one specified referenced GROUP OF ENTITies of a given type may be selected/have been selected.
allOfAllSets	All items from all referenced GROUPs OF ENTITies of a given type may be selected/have been selected.

7.6.1.2.3.6 TemporalValidityParametersGroup – Group

The ***TemporalValidityParametersGroup*** species the temporal access right restrictions made by an ACCESS RIGHT PARAMETER ASSIGNMENT. See NeTEx Part1 for further details on these elements.

Table 101 –*TemporalValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***DayTypeRef***	*ValidityConditionRef*	0:1	DAY TYPE to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***GroupOf-TimebandsRef***	*GroupOfTimebandsRef*	0:1	GROUP OF TIME BANDs to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***OperatingDayRef***	*OperatingDayRef*	0:1	OPERATING DAY to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***OperatingPeriod-Ref***	*OperatingPeriod-Ref*	0:1	OPERATING PERIOD to which ACCESS RIGHT PARAMETER is assigned. +v1.1
«FK»	***Validity-ConditionRef***	*ValidityConditionRef*	0:1	VALIDITY CONDITION to which ACCESS RIGHT PARAMETER is assigned.

Figure 148 — *TemporalValidityParametersGroup* — XSD

7.6.1.2.3.7 ScopingValidityParameters – Group

The Scoping Validity Parameters allows one or more validity parameters to be assigned. The many different possible parameters are organized into five groups. (Organisation, Network, Route, Service, Product) Multiple values are combined using the logical operator (AND, OR) specified by the ***ValidityParameterGroupingType***. AND is the default. For example LINE “22“ and SCHEDULED STOP POINT “4563” means that the assignment applies specifically to stop “4563” of LINE “22”.

Table 102 – ScopingValidityParameters – Element

Classifi-cation	Name	Type	Cardin ality	Description
XGRP	***OrganisationValidity-ParametersGroup***	***xmlGroup***	1:1	ORGANISATION related validity parameters for assignment.
XGRP	***NetworkValidity-ParametersGroup***	***xmlGroup***	1:1	NETWORK related validity parameters for assignment.
XGRP	***RouteValidity-ParametersGroup***	***xmlGroup***	1:1	ROUTE related validity parameters for assignment.
XGRP	***ServiceValidity-ParametersGroup***	***xmlGroup***	1:1	SERVICE related validity parameters for assignment.
XGRP	***ProductValidity-ParametersGroup***	***xmlGroup***	1:1	PRODUCT related validity parameters for assignment.

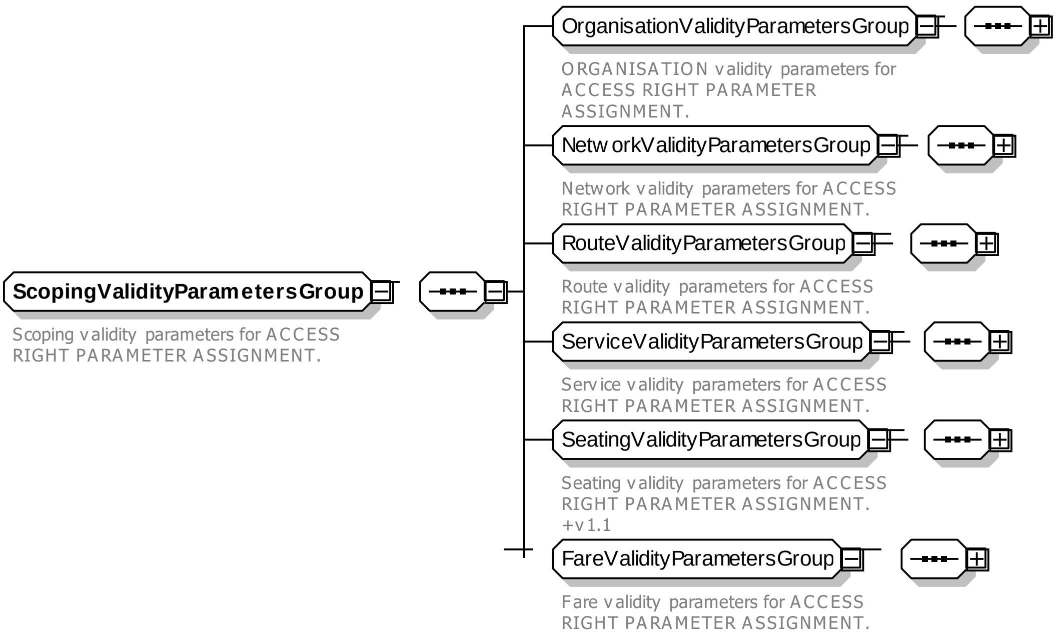


Figure 149 — *ScopingValidityParameterGroup* — XSD

7.6.1.2.3.7.1 OrganisationValidity-ParametersGroup – Group

The ***OrganisationValidityParametersGroup*** specifies validity parameters defining general access rights for MODE and ORGANISATION for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 103 – *OrganisationValidityParametersGroup* – Group

Classifi-cation	Name		Type	Cardin-ality	Description
«enum»	***VehicleModes***		*TransportModeEnum*	0:*	TRANSPORT MODEs to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1 for allowed values. See NeTEx Part1.
«enum»	***Transport-Submodel***		*TransportSubmodel-Enum*	0:1	TRANSPORT SUBMODE to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1 for allowed values. See NeTEx Part1.
«FK»	***GroupOf-OperatorsRef***		*GroupOfOperatorsRef*	0:1	GROUP OF OPERATORs to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
			CHOICE		
	a	***AllOperators-Ref***	*EmtyType*	0:1	ALL OPERATORS apply to ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	b	***OperatorRef***	*OperatorRef*	0:1	OPERATOR to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
			CHOICE		

	a	***AllAuthorities Ref***	*EmptyType*	0:1	ALL AUTHORITIies apply to ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	b	***AuthorityRef***	*AuthorityRef*	0:1	AUTHORITY to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.

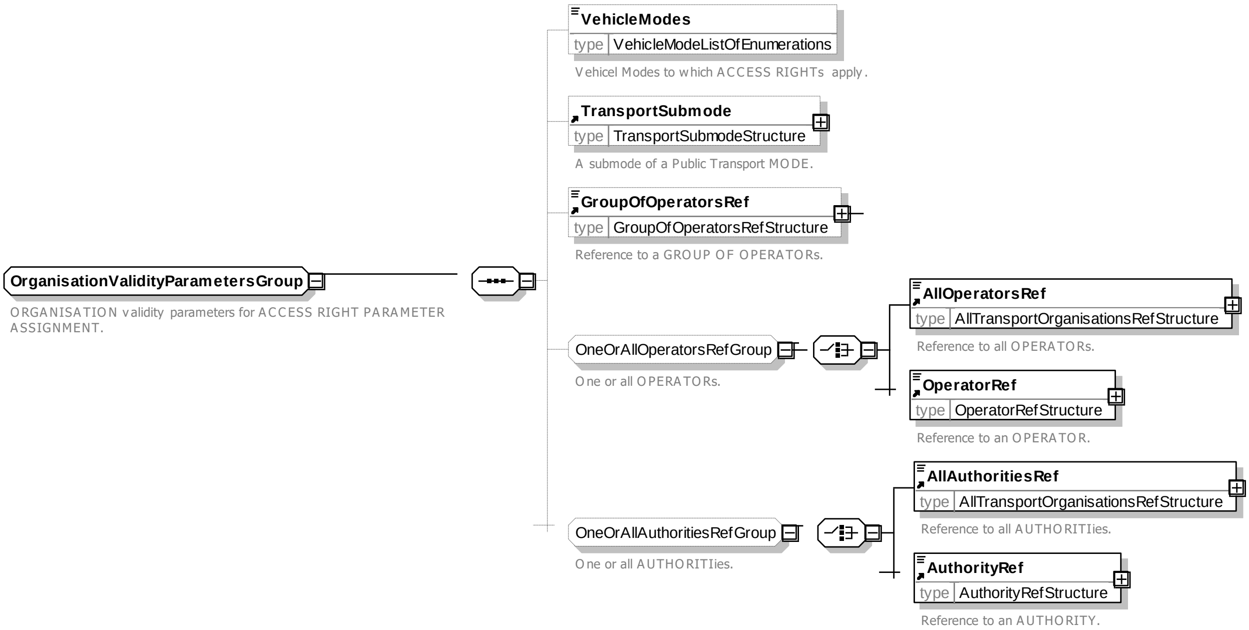


Figure 150 — *OrganisationValidityParametersGroup* — XSD

7.6.1.2.3.7.2 NetworkValidityParametersGroup – Group

The ***NetworkValidityParametersGroup*** specifies validity parameters defining access rights to network elements such as LINE, TARIFF ZONE and SCHEDULED STOP POINT for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 104 – *NetworkValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***GroupOfLinesRef***	*GroupOfLinesRef*	0:1	GROUP OF LINEs to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***LineRef***	*LineRef*	0:1	LINE to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***TypeOfLineRef***	*TypeOfLineRef*	0:1	TYPE OF LINE to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***TariffZoneRef***	*TariffZoneRef*	0:1	TARIFF ZONE to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***FareZoneRef***	*FareZoneRef*	0:1	FARE ZONE to which ACCESS RIGHT PARAMETER is assigned.

«FK»	***FareSectionRef***	*FareSectionRef*	0:1	FARE SECTION to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***Scheduled-StopPointRef***	*ScheduledStopPointRef*	0:1	SCHEDULED STOP POINT to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
XGRP	***PlaceValidity-ParameterGroup***	**xmlGroup**	1:1	PLACE validity parameters for assignment.
XGRP	***SiteValidity-ParameterGroup***	**xmlGroup**	1:1	SITE validity parameters for assignment.

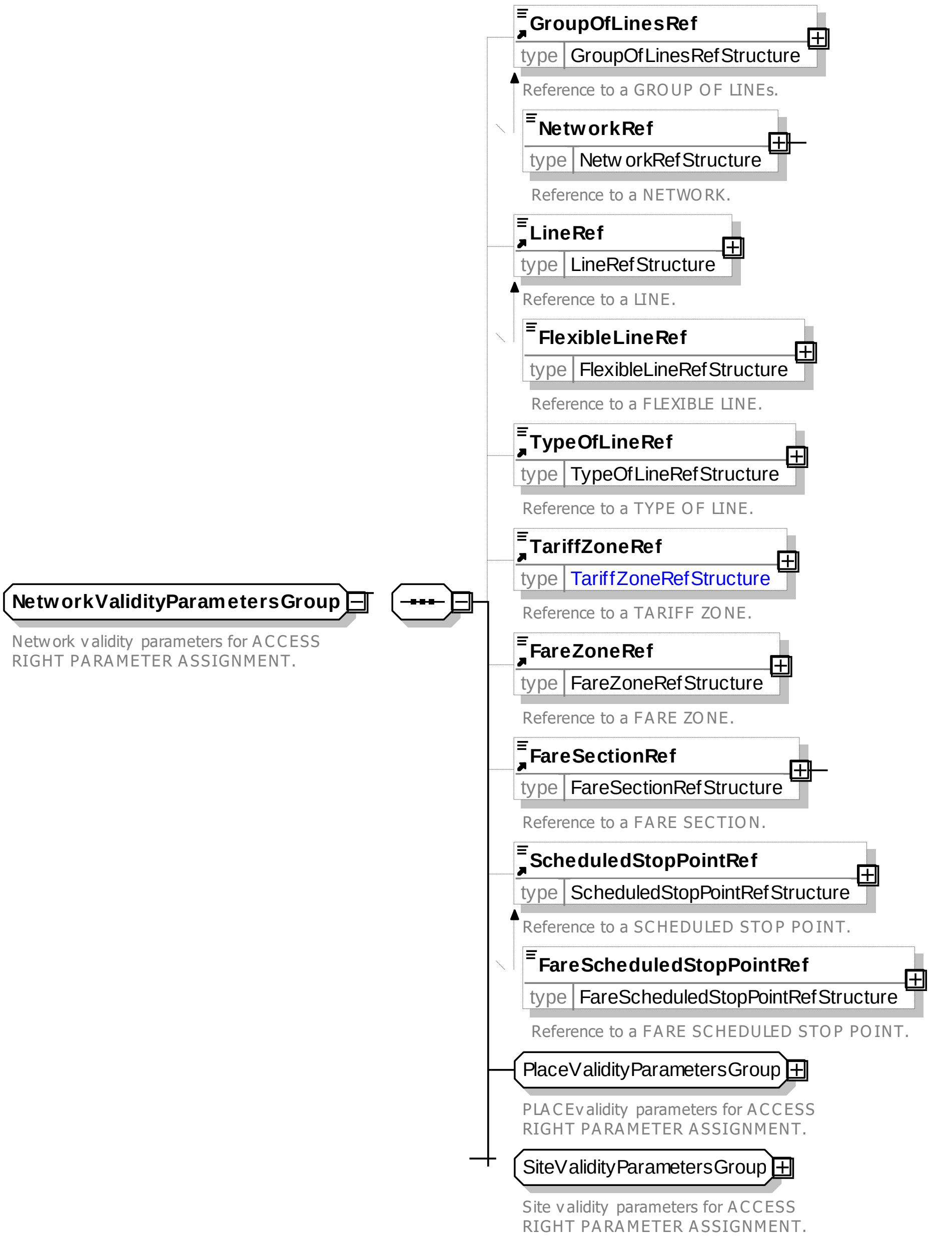


Figure 151 — *NetworkValidityParametersGroup* — XSD

7.6.1.2.3.7.3 PlaceValidityParametersGroup – Group

The ***PlaceValidityParametersGroup*** specifies validity parameters defining use of PLACE elements for an ACCESS RIGHT PARAMETER ASSIGNMENT. For flexible and private travel modes, the travel may be between an ADDRESS or a TOPOGRAPHIC PLACE.

Table 105 – *PlaceValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«enum»	***PlaceUse***	*PlaceUseENum*	0:1	Use of ADDRESS or TOPOGRAPHIC PLACE. See allowed values below.+v1.1
«FK»	***Topographic-PlaceRef***	*TopographicPlaceRef*	0:1	TOPOGRAPHIC PLACE to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1. +v1.1
«FK»	***AddressRef***	*PostalAddressRef* | *RoadAddressRef*	0:1	ADDRESS to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1. +v1.1

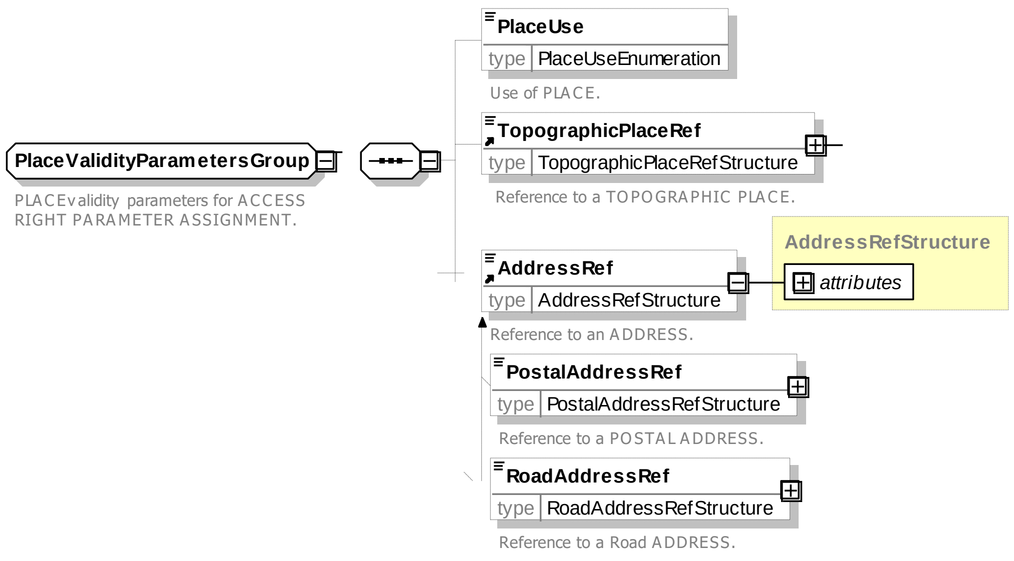


Figure 152 — *PlaceValidityParametersGroup* — XSD

7.6.1.2.3.7.4 PlaceUse – Allowed values

The following table shows the allowed values for ***PlaceUse*** (*PlaceUseEnumeration*).

Table 106 – *PlaceUse* – Allowed values

Value	Description
startAt	Place is origin.
endAt	Place is destination.
via	Go via place.
restrictTo	Restrict use to specified place.
other	Other use.

7.6.1.2.3.7.5 SiteValidityParametersGroup – Group

The ***SiteValidityParametersGroup*** specifies validity parameters defining access rights to SITE elements for an ACCESS RIGHT PARAMETER ASSIGNMENT. SITEs can be used for example to associate fare structure elements with POINTs OF INTEREST as for a travel product that also allows entry to museums and other tourist attractions, or to all SITEs of a given type of point interest using a POINT OF INTEREST CLASSIFICATION.

Table 107 – *SiteValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***SiteElementRef***	*SiteElementRef*	0:1	SITE ELEMENT to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***PointOfInterest-Classification-ElementRef***	*PointOfInterest-ClassificationElementRef*	0:1	POINT OF INTEREST CLASSIFICATION to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.

Figure 153 — *SiteValidityParametersGroup* — XSD

7.6.1.2.3.7.6 RouteValidityParametersGroup – Group

The ***RouteValidityParametersGroup*** specifies validity parameters defining access rights to particular routes (as in effect specified by DISTANCE MATRIX and SERIES CONSTRAINT elements) for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 108 – *RouteValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***RoutingType***	*RoutingTypeEnum*	1:1	Type of routing to which assignment applies. See allowed values earlier under FARE ZONE model.
«enum»	***Directions***	*RelativeDirectioeEnum*	0:1	Directions in which assignment applies. See NeTEx Part1.
«FK»	***SeriesConstraint Ref***	*SeriesConstraintRef*	0:1	SERIES CONSTRAINT to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***ServiceJourney-PatternRef***	*ServiceJourney-PatternRef*	0:1	SERVICE JOURNEY PATTERN to which ACCESS RIGHT PARAMETER is assigned.

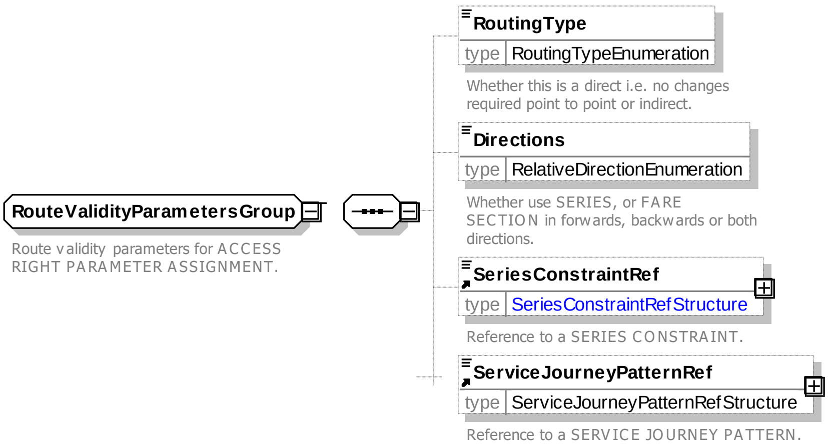


Figure 154 — *RouteValidityParametersGroup* — XSD

7.6.1.2.3.7.7 ServiceValidityParametersGroup – Group

The ***ServiceValidityParametersGroup*** specifies validity parameters defining access rights to particular services or types of service for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 109 – *ServiceValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	1:1	Reference to a CLASS OF USE (Seat Class).

«enum»	***FareClass***	*FareClassEnum*	0:1	FARE CLASS to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1 for allowed values.
«FK»	***FacilitySetRef***	*FacilitySetRef*	0:1	FACILITY SET provided or available for fare.
«FK»	***TypeOfProduct-CategoryRef***	*TypeOfProduct-CategoryRef*	0:1	Type of PRODUCT CATEGORY to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***Service-JourneyRef***	*ServiceJourneyRef*	0:1	VEHICLE JOURNEY to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***TrainNumberRef***	*TrainNumberRef*	0:1	TRAIN NUMBER to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***GroupOfServicesRef***	*GroupOfServicesRef*	0:1	GROUP OF SERVICEs to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***VehicleTypeRef***	*VehicleTypeRef*	0:1	VEHICLE TYPE to which assignment is made.
«FK»	***TypeOf-ServiceRef***	*TypeOfServiceRef*	0:1	TYPE OF SERVICE to which assignment is made, for example whether the assignment is a night train.

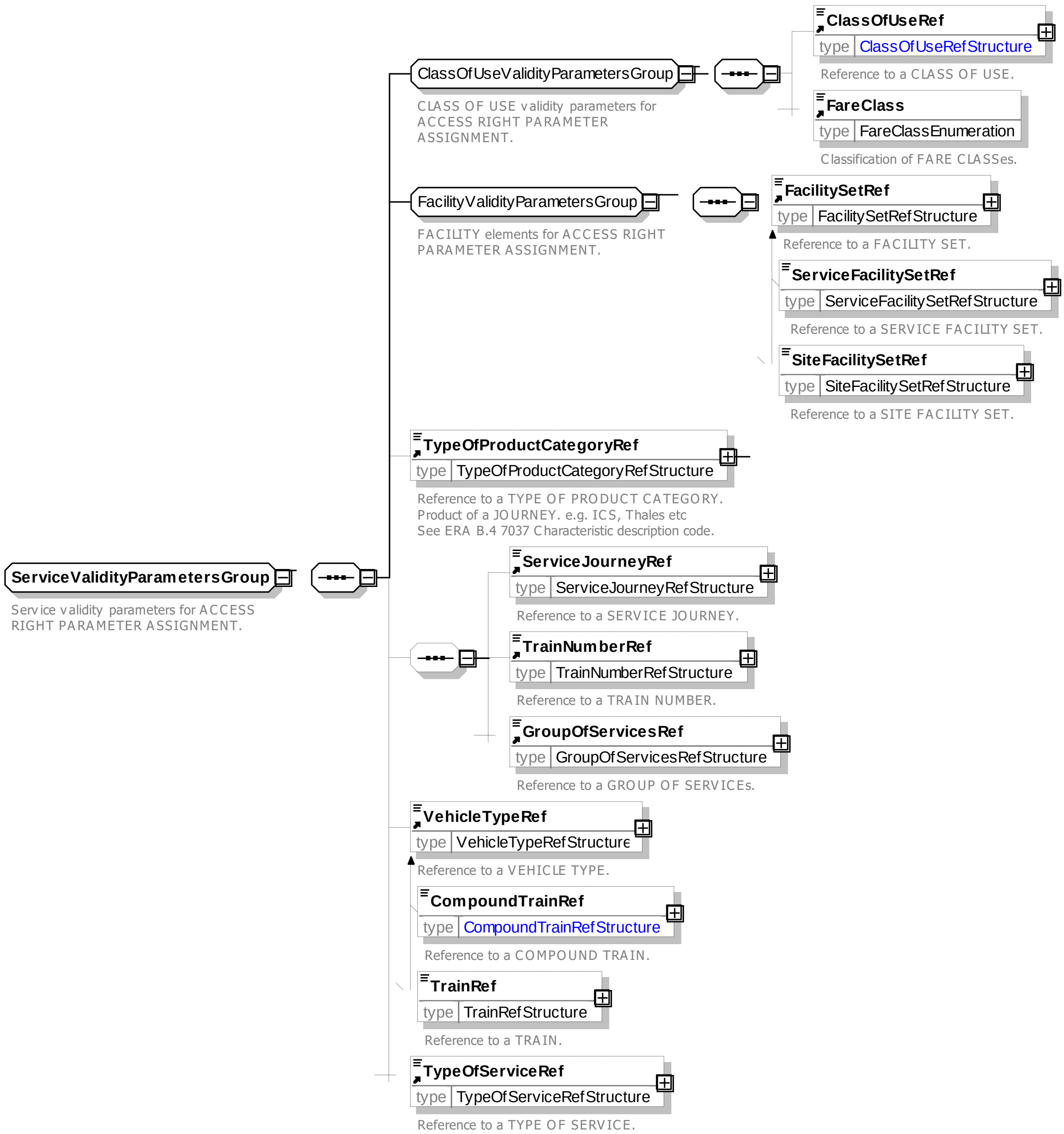


Figure 155 — *ServiceValidityParametersGroup* — XSD

7.6.1.2.3.7.8 *SeatingValidityParametersGroup* – Group

The ***SeatingValidityParametersGroup*** specifies conditions on seating for an ACCESS RIGHT PARAMETER ASSIGNMENT. For example, whether a purchase applies to a particular carriage or seat.

Table 110 – *SeatingValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***TrainElementRef***	*TrainElementRef*	0:1	Reference to a TRAIN ELEMENT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

«FK»	***TrainComponent Label-AssignmentRef***	*TrainComponentLabel-AssignmentlRef*	0:1	Reference to a TRAIN COMPONENT LABEL ASSIGNMENT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***PassengerSeat-Ref***	*PassengerSeatRef*	0:1	Reference to a PASSENGER SEAT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

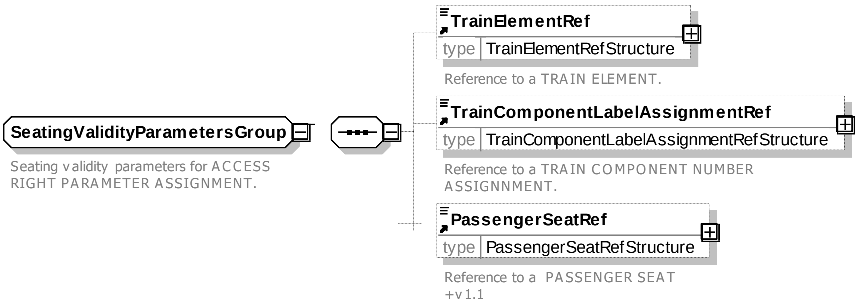


Figure 156 — *SeatingValidityParametersGroup* — XSD

7.6.1.2.3.7.9 *FareValidityParametersGroup* – Group

The *FareValidityParametersGroup* specifies conditions on fares for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 111 – *FareValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description
XGRP	***TariffValidity-ParameterGroup***	**xmlGroup**	01	TARIFF validity parameters for assignment.
XGRP	***ProductValidity-ParameterGroup***	**xmlGroup**	01	PRODUCT validity parameters for assignment.
XGRP	***SalesOfferValidity-ParameterGroup***	**xmlGroup**	01	SALES OFFER PACKAGE validity parameters for assignment.
XGRP	***DistributionValidity-ParameterGroup***	**xmlGroup**	01	DISTRIBUTION validity parameters for assignment.

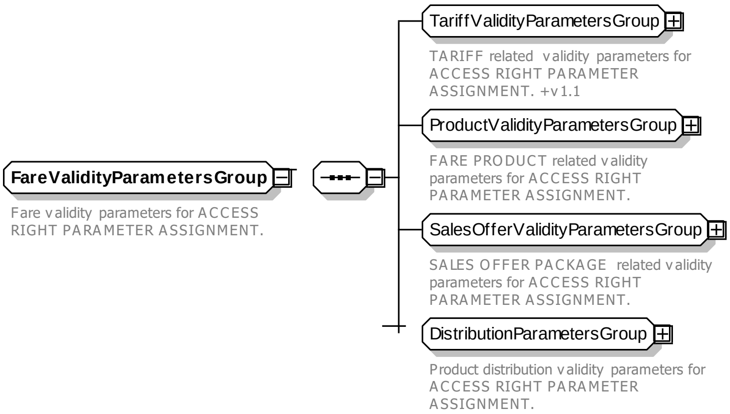


Figure 157 — *FareValidityParametersGroup* — XSD

7.6.1.2.3.7.10 *TariffValidityParametersGroup* – Group

The ***TariffValidityParametersGroup*** specifies conditions on based on tariff structure elements for an ACCESS RIGHT PARAMETER ASSIGNMENT. For example, where a condition applies to a specific FARE STRUCTURE ELEMENT.

Table 112 – *TariffValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***TypeOfFare-StructureFactorRef***	*TypeOfFare-StructureFactorRef*	0:1	Reference to a TYPE OF FARE STRUCTURE FACTOR to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOfFare-Structure-ElementRef***	*TypeOfFare-StructureElementRef*	0:1	Reference to a TYPE OF FARE STRUCTURE ELEMENT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOfTariffRef***	*TypeOfTariffRef*	0:1	Reference to a TYPE OF TARIFF to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

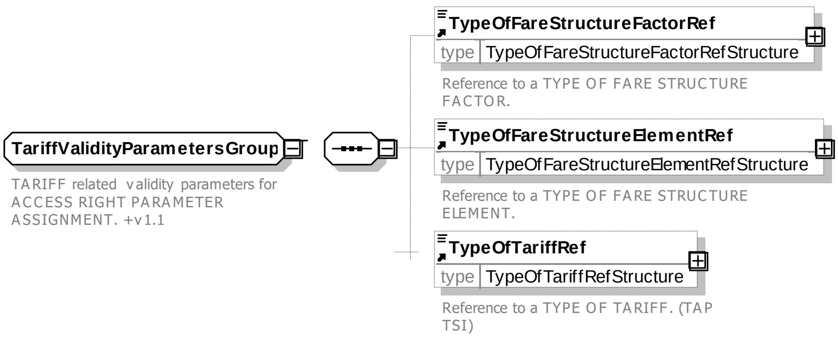


Figure 158 — *TariffValidityParametersGroup* — XSD

7.6.1.2.3.7.11 *ProductValidityParametersGroup* – Group

The ***ProductValidityParametersGroup*** specifies conditions on purchase or fulfilment for an ACCESS RIGHT PARAMETER ASSIGNMENT. For example, where a ticket may be purchased or collected, or whether a commercial condition such as refunding is restricted with a particular DISTRIBUTION CHANNEL.

Table 113 – *ProductValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description
«FK»	***PricingRuleRef***	*PricingRuleRef+*	0:1	Reference to a PRICING RULE to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOfPricing-RuleRef***	*TypeOfPricingRuleRef*	0:1	Reference to a TYPE OF PRICING RULE to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOfFare-ProductRef***	*TypeOfFareProductRef*	0:1	Reference to a TYPE OF FARE PRODUCT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOfUsage-ParameterRef***	*TypeOfUsageParameterRef*	0:1	Reference to a TYPE OF USAGE PARAMETER to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***ChargingMoment Ref***	*ChargingMomentRef*	0:1	Reference to a CHARGING MOMENT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOf-ConcessiontRef***	*TypeOfConcessionRef*	0:1	Reference to a TYPE OF CONCESSION to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

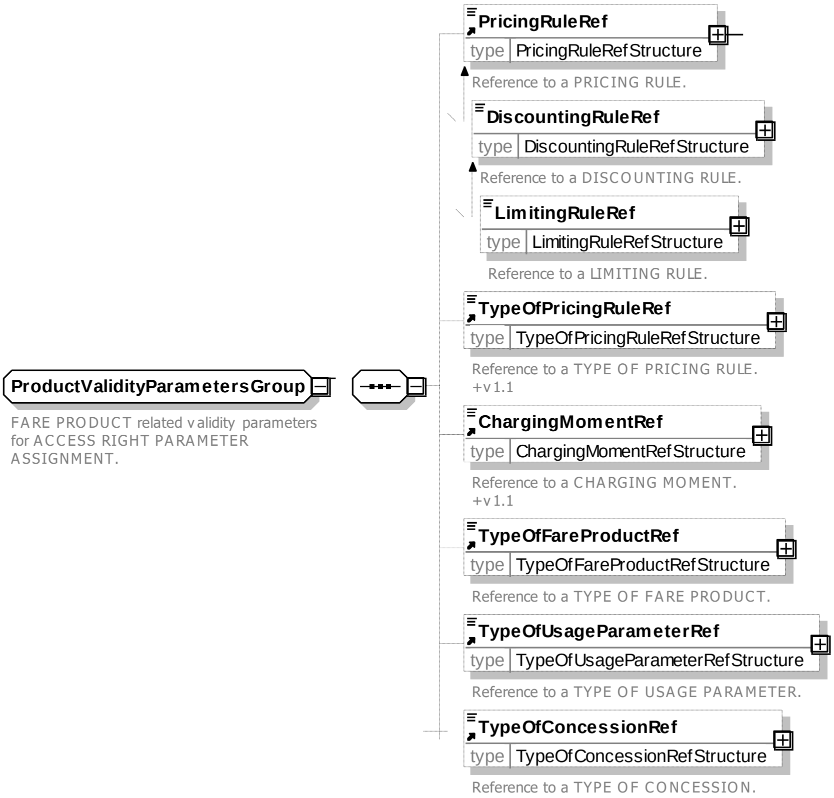


Figure 159 — *ProductValidityParametersGroup* — XSD

7.6.1.2.3.7.12 *SalesOfferValidityParametersGroup* – Group

The ***SalesOfferValidityParametersGroup*** specifies conditions on SALESS OFFER PACKAGE properties for an ACCESS RIGHT PARAMETER ASSIGNMENT. For example, where a ticket may be purchased or collected, or whether a commercial condition such as refunding is restricted with a particular DISTRIBUTION CHANNEL.

Table 114 – *SalesOfferValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description
«FK»	***TypeOfSales-OfferPackageRef***	*TypeOfSales-OfferPackageRef*	0:1	Reference to a TYPE OF SALES OFER to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOfTravel-DocumentRef***	*TypeOfTravelDocumentRef*	0:1	Reference to a TYPE OF TRAVEL DOCUMENT to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOMachine-ReadabilityRef***	*TypeOfMachine-ReadabilityRef*	0:1	Reference to a TYPE OF MACHINE READABILITY to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

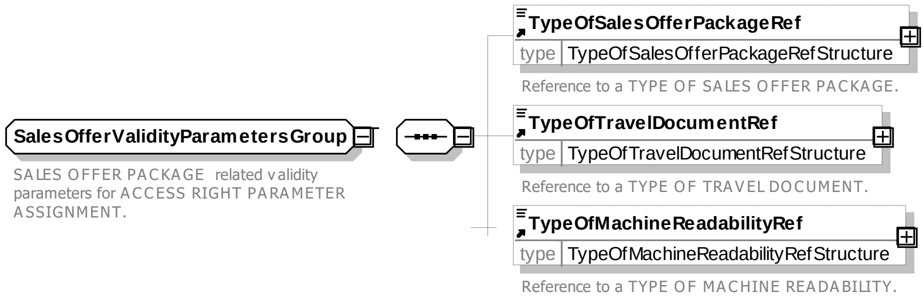


Figure 160 — *SalesOfferValidityParametersGroup* — XSD

7.6.1.2.3.7.13 *DistributionValidityParametersGroup* – Group

The ***DistributionValidityParametersGroup*** specifies conditions on purchase or fulfilment for an ACCESS RIGHT PARAMETER ASSIGNMENT relation to distribution. For example, where a ticket may be purchased or collected, or how it may be paid for.

Table 115 – *DistributionValidityParametersGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***Distribution-ChannelRef***	*FareStructureElement-Ref*	0:1	Reference to a DISTRIBUTION CHANNEL to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***GroupOf-Distribution-ChannelsRef***	*GroupOfDistribution-ChannelsRef*	0:1	Reference to a GROUP OF DISTRIBUTION CHANNELs to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***Fulfilment-MethodRef***	*FareStructureElement-Ref*	0:1	Reference to a FULFILMENT METHOD to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.
«FK»	***TypeOPayment-MethodRef***	*TypeOPayment-MethodRef*	0:1	Reference to a TYPE OF PAYMENT METHOD to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

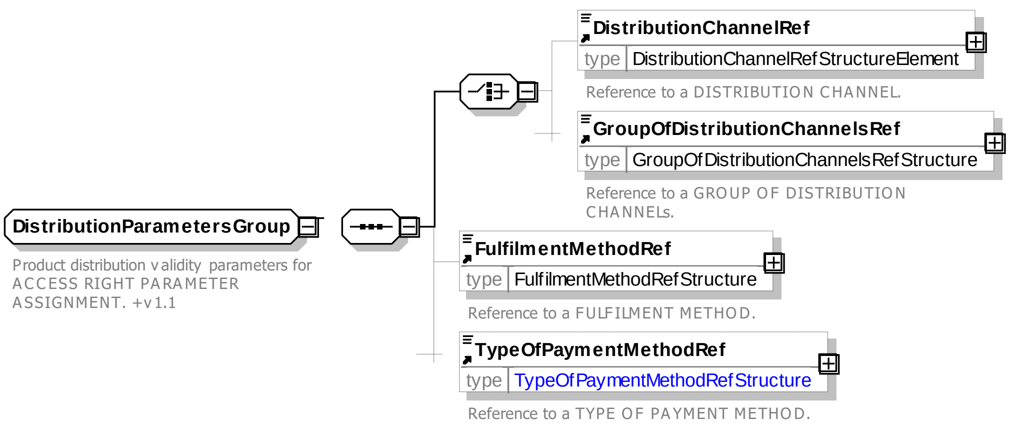


Figure 161 — *DistributionValidityParametersGroup* — XSD

7.6.1.2.3.8 AccessRightParameterAssignmentIncludesGroup – Group

The ***AccessRightParameterAssignmentIncludesGroup*** specifies rules for creating composite ACCESS RIGHT PARAMETER ASSIGNMENTs that combine simple assignments into complex conditions. A logical operator can be specified.

Table 116 – *AccessRightParameterAssignmentIncludesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***Includes-GroupingType***	*BooleanOperatorEnum*	0:1	Logical operator for combining included elements. The default is '*OR*'. See earlier for allowed values.
«cntd»	***includes***	*AccessRightParameter-Assignment+*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs making up a composite ACCESS RIGHT PARAMETER ASSIGNMENT.

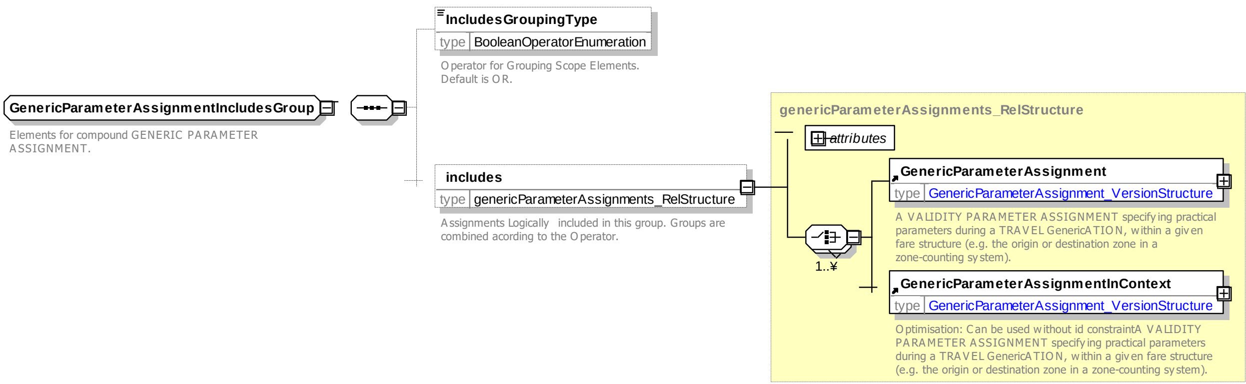


Figure 162 — *AccessRightParameterAssignmentIncludesGroup* — XSD

7.6.1.2.3.8.1 IncludesGroupingType – Allowed values

The following table shows the allowed values for the ***IncludesGroupingType*** (*BooleanOperator-Enumeration*)

Table 117 – *GroupingType* – Allowed values

Value	Description
AND	Logical And
OR	Logical Or
XOR	Exclusive OR
NOT	Logical not

7.6.1.2.3.9 ValidityParameterAssignment – Model Element

An ACCESS RIGHT PARAMETER ASSIGNMENT relating a fare collection parameter to a theoretical FARE PRODUCT (or one of its components) or a SALES OFFER PACKAGE.

Table 118 – *ValidityParameterAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*AccessRight-ParameterAssignment*	::>	VALIDITY PARAMETER ASSIGNMENT inherits from ACCESS RIGHT PARAMETER ASSIGNMENT.
«PK»	***id***	*ValidityParameter-AssignmentIdType*	1:1	Identifier of VALIDITY PARAMETER ASSIGNMENT.
«FK»	***QualityStructure-FactorRef***	*QualityStructure-FactorRef*	0:1	Reference to a QUALITY STRUCTURE FACTOR to which the ACCESS RIGHT PARAMETER ASSIGNMENT applies.

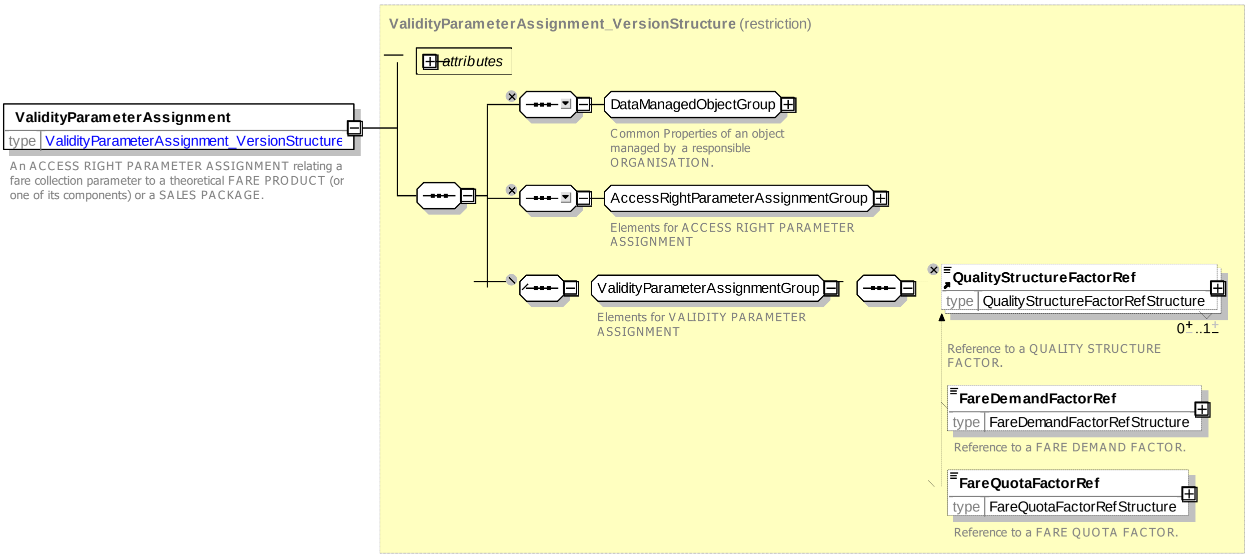


Figure 163 — *ValidityParameterAssignment* — XSD

7.6.1.2.3.10 GenericParameterAssignment – Model Element

A VALIDITY PARAMETER ASSIGNMENT specifying generic access rights for a class of products (e.g. a time band limit - 7 to 10 a.m. - for trips made with a student pass). May include alternatives from which a purchaser selects.

Table 119 – *GenericParameterAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*ValidityParameter-Assignment*	::>	GENERIC PARAMETER ASSIGNMENT inherits from VALIDITY PARAMETER ASSIGNMENTs
«PK»	***id***	*GenericParameter-AssignmentIdType*	1:1	Identifier of GENERIC PARAMETER ASSIGNMENT.

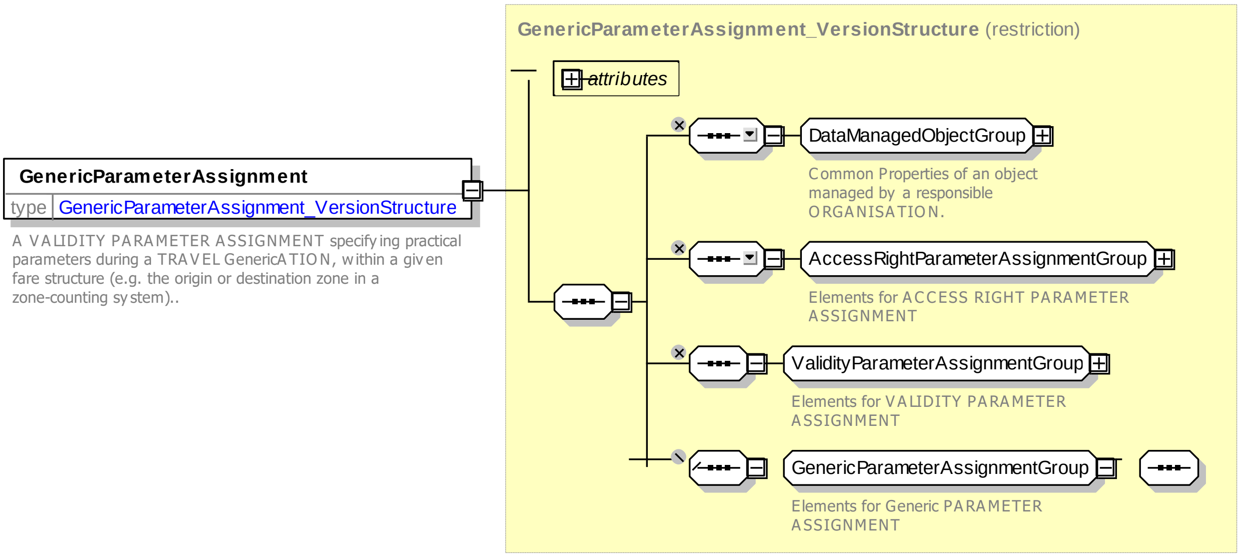


Figure 164 — *GenericParameterAssignment* — XSD

7.6.1.2.3.11 TypeOfAccessRightAssignment – Model Element

A classification of TARIFFs to express the different classes of fares.

Table 120 – *TypeOfAccessRightAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF ACCESS RIGHT ASSIGNMENT inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfAccessRight-AssignmentIdType*	1:1	Identifier of TYPE OF ACCESS RIGHT ASSIGNMENT.

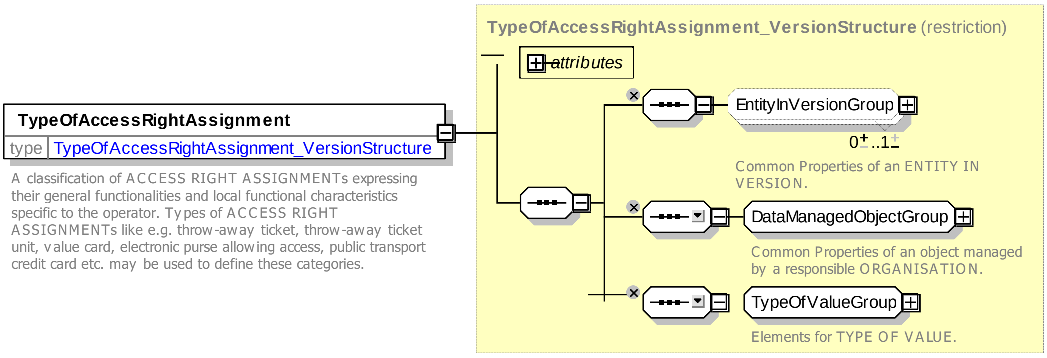


Figure 165 — *TypeOfAccessRightAssignment* — XSD

7.6.1.3 Usage Parameter

7.6.1.3.1 USAGE PARAMETERs – Conceptual MODEL

The validity of an access right (or of a marketable combination) may be limited by parameters related to the way of consuming them (user profile, frequency of use, transferability, etc.). They express in general additional rules than those expressed by the fare structure and validity parameters for CONTROLLABLE ELEMENTs or FARE STRUCTURE ELEMENTs. Such parameters are described by the generic entity USAGE PARAMETERs.

USAGE PARAMETERs specify various types of functional limitation on a fare element, for example, when it can be bought (PURCHASE WINDOW), who may buy it (USER PROFILE), whether it can be given to someone else (TRANSFERABILITY) etc., etc. The parameters fall into four main groups which are discussed in turn below:

- **Travel** USAGE PARAMETERs specify limitations on travel such as ROUND TRIP, ROUTING, FREQUENCY OF USE, INTERCHANGING, USAGE VALIDITY PERIOD, MINIMUM STAY.
  - ROUND TRIP expressing the properties relating to single or return trip use of an access right.
  - USAGE VALIDITY PERIOD describes a broad time limitation of access rights, especially passes. It may include a 'standard duration' of validity (1-day, 1 month…), time limitations ('start date' and 'end date', 'start time' and 'end time'), or a combination of both;
  - FREQUENCY OF USE describes the limitation of an access right, depending on frequency of use during a VALIDITY PERIOD. For instance, a product is offered at a special fare if it is used more than 50 times in a month;
  - INTERCHANGING expressing the limitations on making changes within a trip;
  - MINIMUM STAY, expressing the details of any minimum stay at the destination required to use the product;
  - STEP LIMIT, a geographical parameter limiting the access rights by counts of stops, sections or zones;
  - ROUTING, expressing the properties Limitations the limitations on routing of an access right.
  - SUSPENDING, describing the conditions that applyfor temporarilyy suspending an access right such as a season pass.
- **Eligibility** USAGE PARAMETERs specify limitations on who can use a product such as USER PROFILE, GROUP TICKET, COMPANION OR GROUP MEMBER, COMMERCIAL PROFILE, ENTITLEMENT GIVEN and ENTITLEMENT REQUIRED.
  - USER PROFILE, which describes the social profile of a customer. It is generally used to allow discounts based on age groups (e.g. under 18), gender, profession, social status (e.g. student, retired, unemployed), etc.;
  - COMMERCIAL PROFILE, which is used to describe customer categories depending on their commercial relations with the operator (frequent traveller, amount of purchase by a company, etc.). It is generally used to allow discounts;
  - GROUP TICKET describes the number and characteristics of persons possibly entitled to travel in addition to the holder of an access right;
  - COMPANION PROFILE, indicating the number and characteristics of persons entitled to travel in a group or as companions to another USER PROFILE;

  - RESIDENTIAL QUALIFICATION, categorising the users depending on their commercial relations with the operator (frequency of use, amount of purchase etc.), often used for allowing discounts;

  - ELIGIBILITY CHANGE POLICY, specifying the action to be taken if a users eligibility for a given profile changes.

- **Entitlement** USAGE PARAMETERs specify limitations on other product rights given or required by a product.

  - ENTITLEMENT REQUIRED, indicating whether an ENTITLEMENT PRODUCT is required to use access right;

  - ENTITLEMENT GIVEN, indicating whether a specific access right represents an ENTITLEMENT PRODUCT.

- **Luggage** USAGE PARAMETERs specify limitations on luggage such as LUGGAGE ALLOWANCE.

  - LUGGAGE ALLOWANCE describes the number and characteristics (weight or volume, bicycles, etc.) of luggage that the holder of an access right is entitled to carry;

- **Booking** USAGE PARAMETERs specify limitations on booking transactions such as PURCHASE WINDOW, TRANSFERABILITY, RESERVING, EXCHANGING, REFUNDING.

    - PURCHASE WINDOW, indicating the period in which the product must be purchased.

    - TRANSFERABILITY describes the right to transfer an access right to other persons than the original customer (number and characteristics of persons entitled to consume).

    - RESELLING, expressing the common resale conditions (i.e. for exchange or refund) attached to the product.

    - EXCHANGING indicating whether and how the access right may be exchanged for another access right.

    - REFUNDING indicating whether and how the purchased access right may be refunded.

    - REPLACING indicating whether and how the access right may be replaced (for example if a ticket is lost).

      - RESERVING indicating whether the access right requires reservation.

  - **Charging** USAGE PARAMETERs specify policies for additional aspects of charging such as CHARGING POLICY and PENALTY POLICY.

      - The CHARGING POLICY parameter specifies limitations on how a product may be charged. May be used to specify a minimum and maxim credit level.

      - The PENALTY POLICY parameter specifies rules relating to penalty fares that may be incurred.

      - The SUBSCRIBING POLICY parameter specifies rules relating to fares bought on subscription with regular payment intervals.

USAGE PARAMETERs may have one or more USAGE PARAMETER PRICE and be classified with a TYPE OF USER PARAMETER.

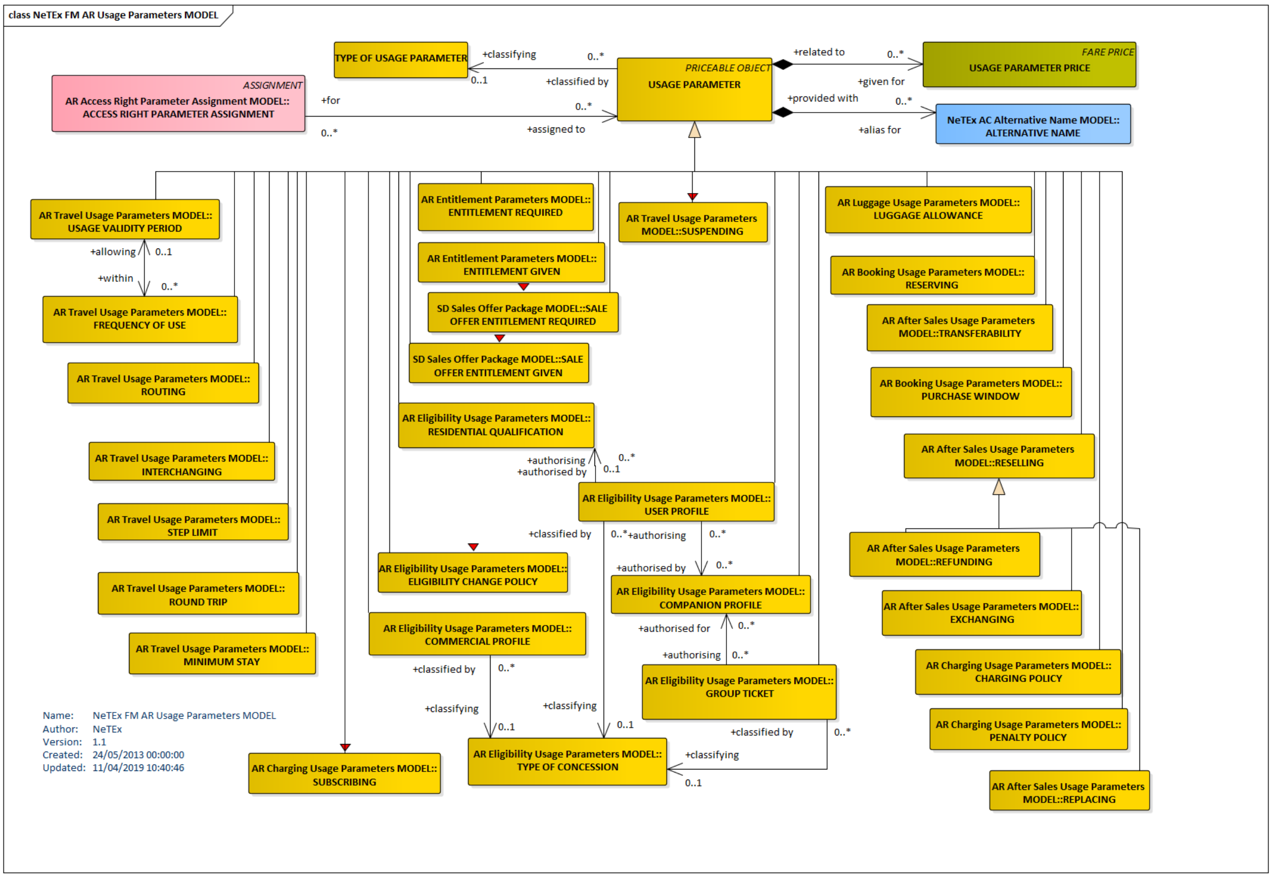


Figure 166 —Usage Parameters – Conceptual MODEL (UML)

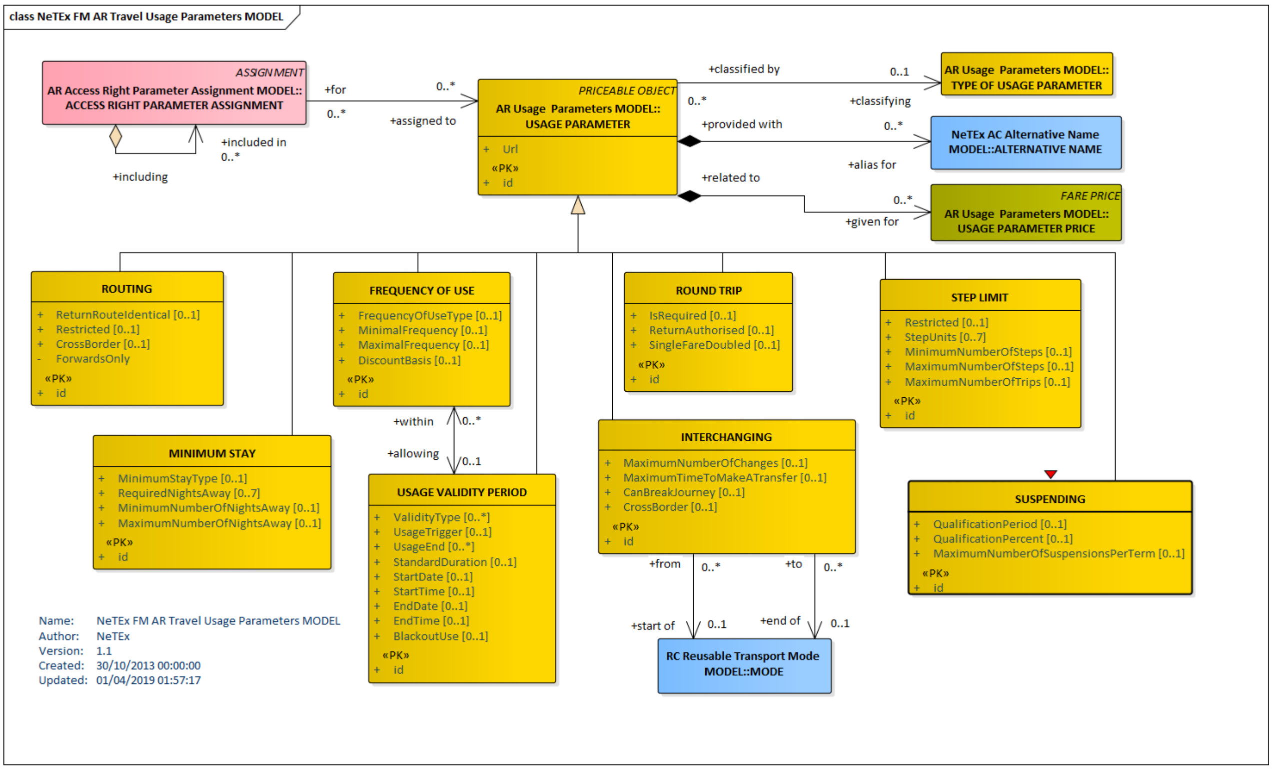


Figure 167 —Travel Usage Parameters – Conceptual MODEL (UML)

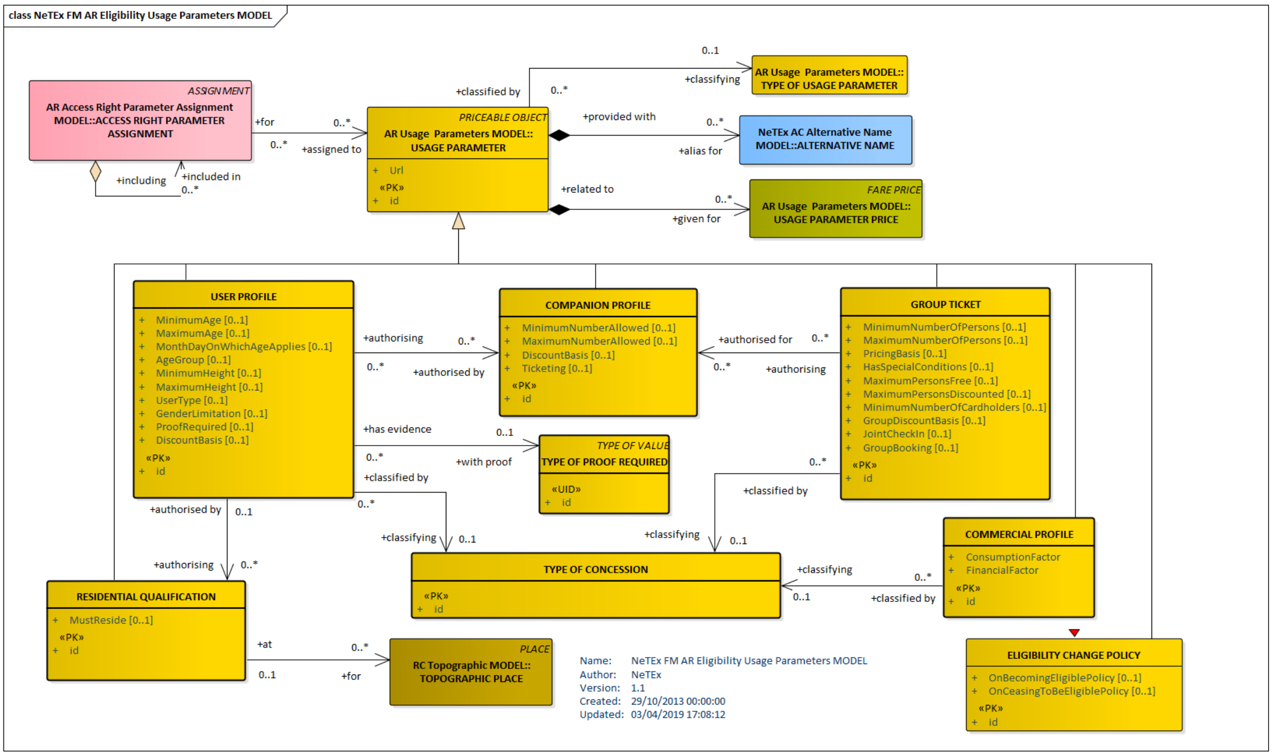


Figure 168 —Eligibility Usage Parameters – Conceptual MODEL (UML)

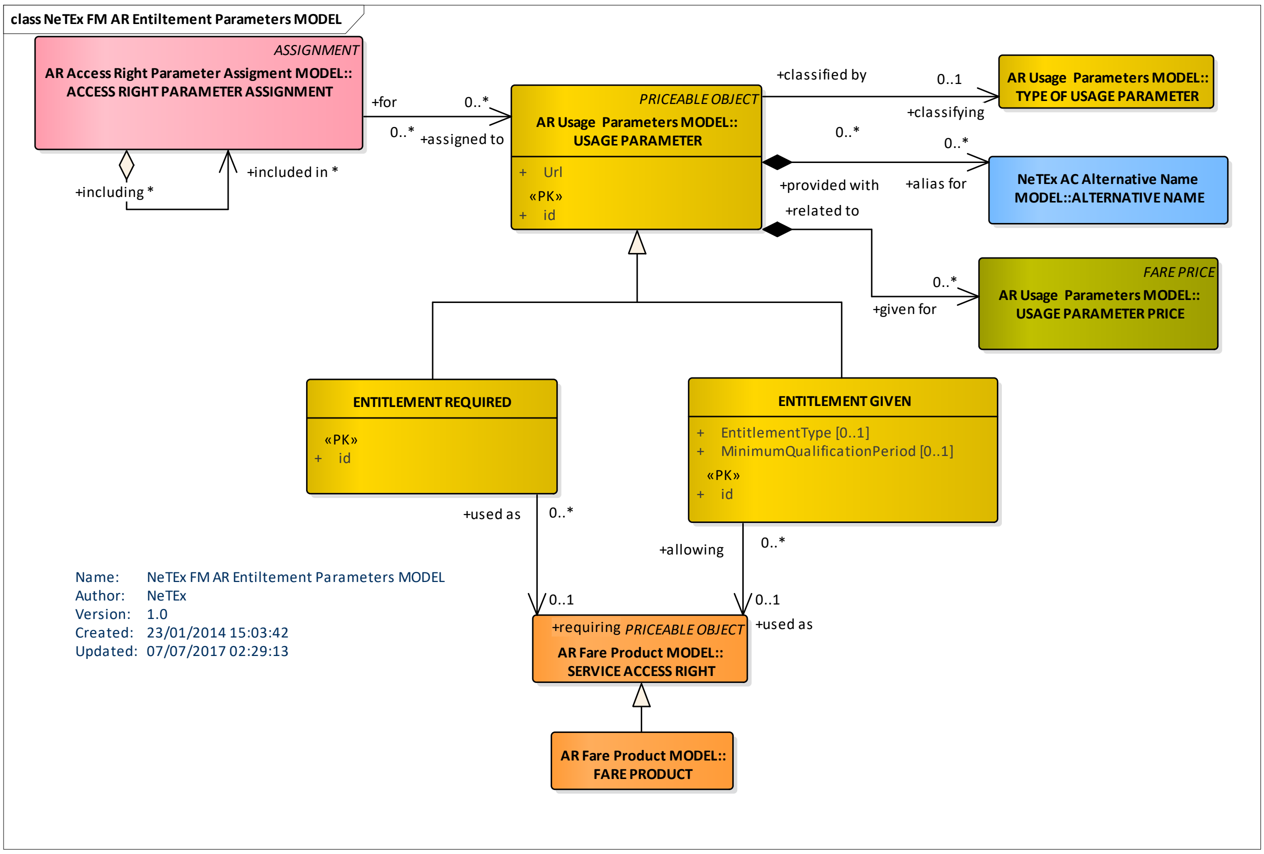


Figure 169 —Entitlement Usage Parameters – Conceptual MODEL (UML)

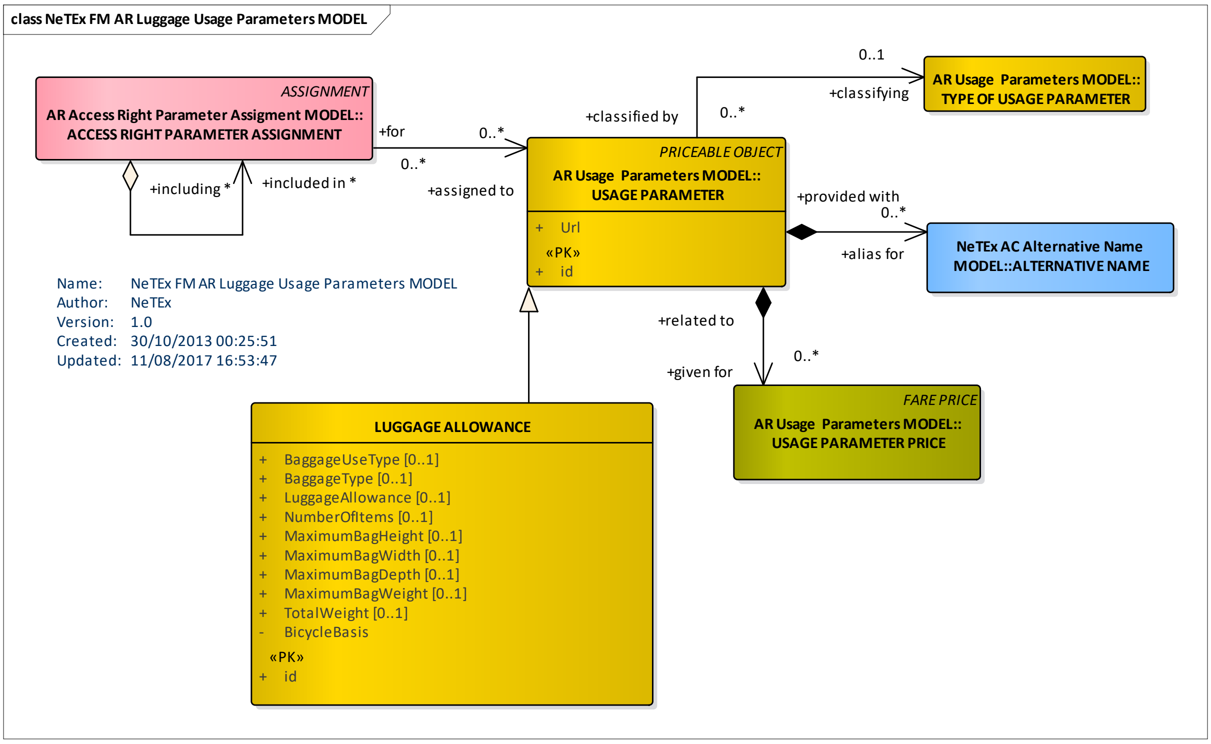


Figure 170 —Luggage Usage Parameters – Conceptual MODEL (UML)

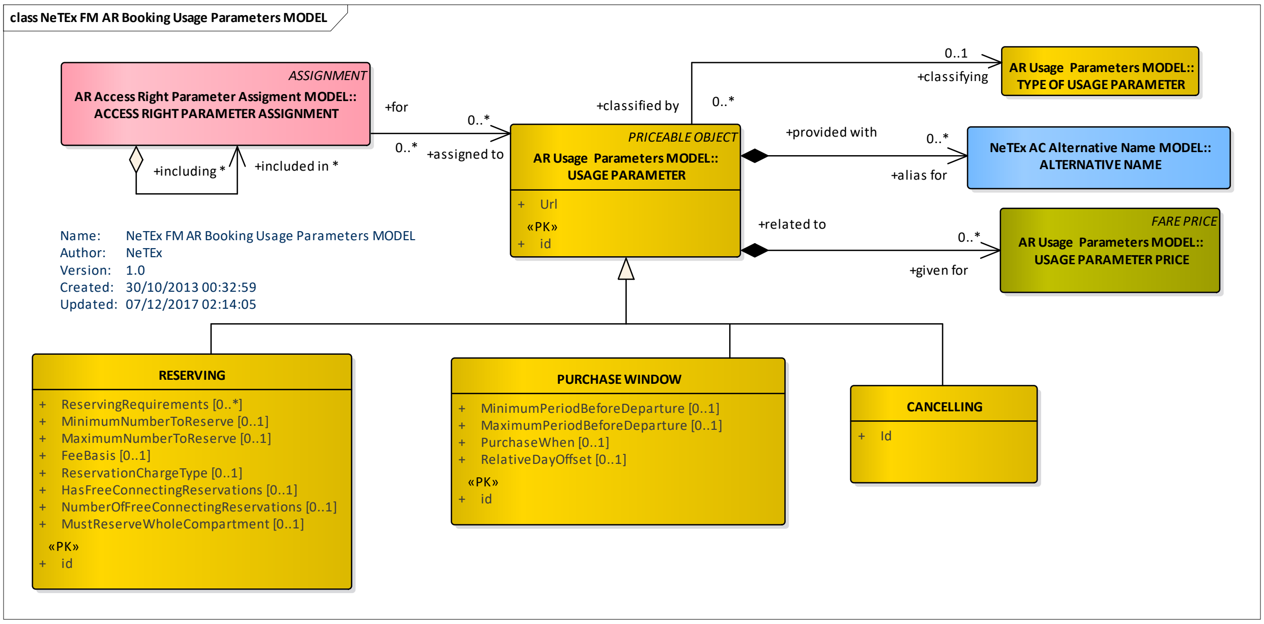


Figure 171 —Booking Usage Parameters – Conceptual MODEL (UML)

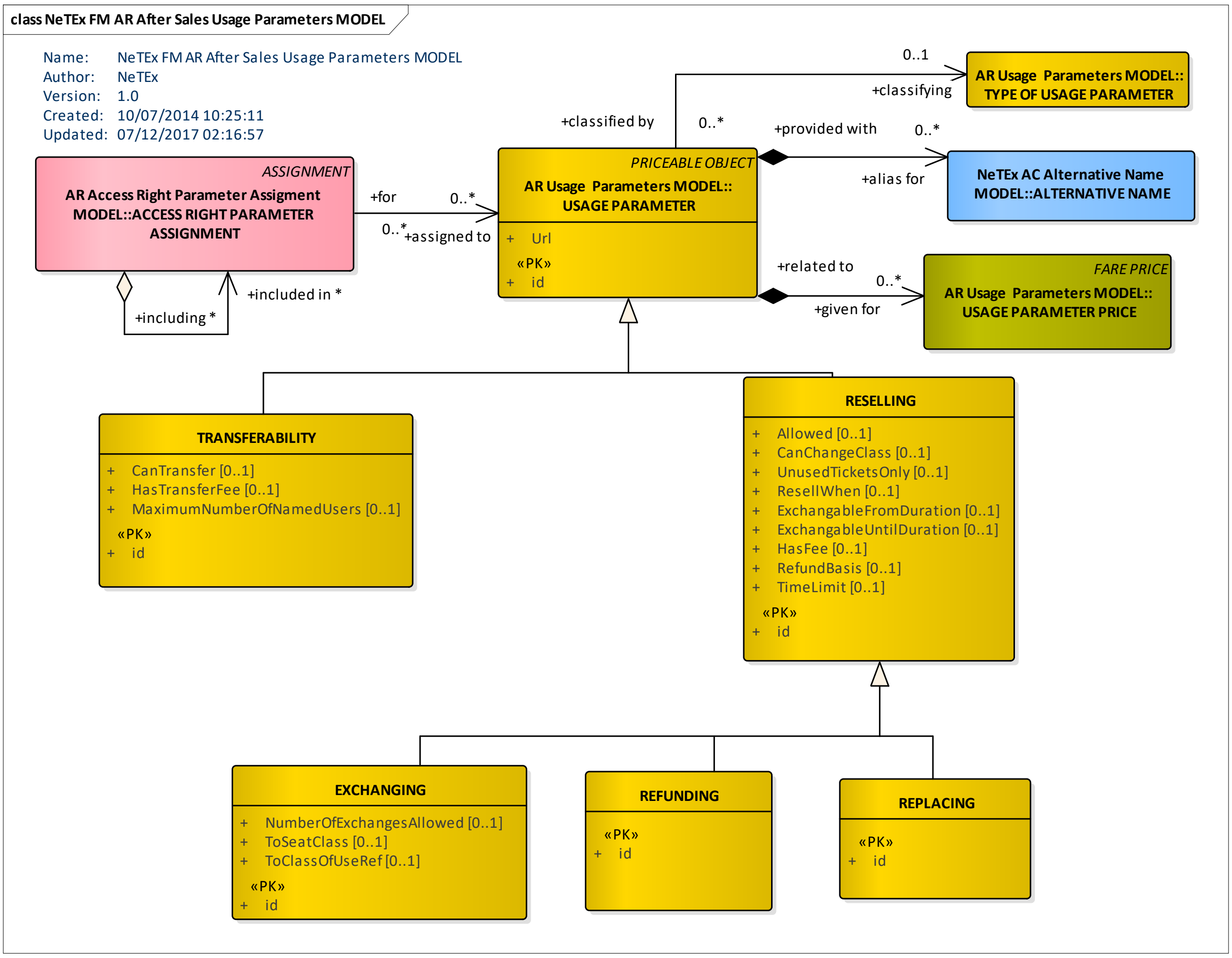


Figure 172 —After Sales Usage Parameters – Conceptual MODEL (UML)

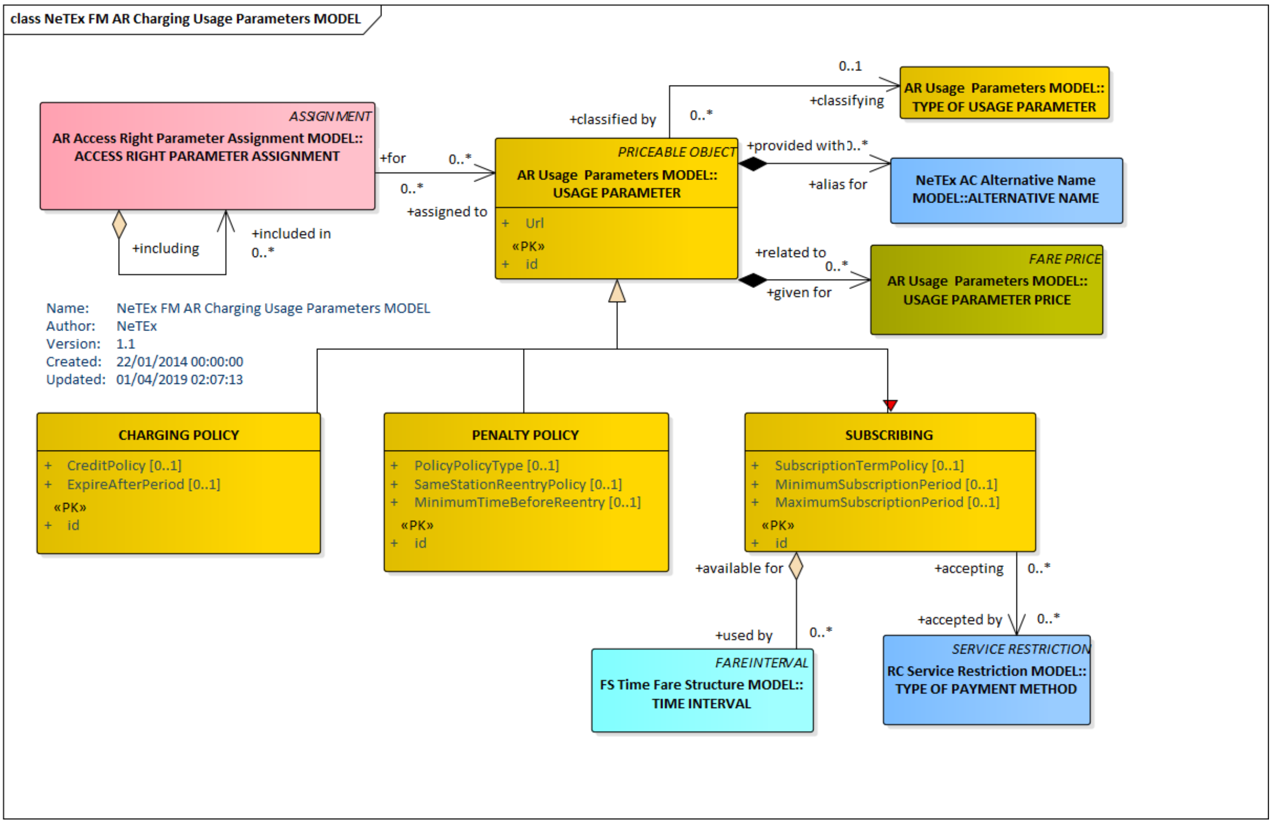


Figure 173 —Charging Usage Parameters – Conceptual MODEL (UML)

7.6.1.3.2 Usage Parameter – Conceptual Examples

7.6.1.3.2.1 Usage Parameters – Rail example

The following figure shows examples of some typical rail fare conditions for a Cheap Day return taken from the UK National Rail Enquiries website. NeTEx allows such conditions to be represented not only as text NOTICEs (associated with elements using NOTICE ASSIGNMENTs), but also as structured parameters that can be computed over in advanced fare information engines.

Ticket type

Ticket type	Anytime Day Single
Description	Anytime tickets allow you to travel at any time of the day. You may need to travel by a specific route or train company but the ticket will state this. You are allowed to break your journey.
Train Operator	London Midland, London Overground, Northern, London Underground, Hull Trains, Heathrow Express, Southeastern, Merseyrail, Southern, Grand Central Railway, ScotRail, c2c, Island Line, South Western Railway, Chiltern Railways, Great Northern, East Midlands Trains, CrossCountry, Virgin Trains East Coast, Great Western Railway, Gatwick Express, Arriva Trains Wales, Thameslink, Greater Anglia, TfL Rail, Heathrow Connect, TransPennine Express, Virgin West Coast
Train Operators and Connections	
Booking Deadlines	None.
Discounts	Child: Children (aged 5 to 15 inclusive) are offered a 50% discount. Up to two children aged under 5 can travel free with each fare paying passenger. Railcard: 16-25 Railcard, Two Together, Disabled Persons Railcard, HM Forces Railcard, Senior Railcard, Network Railcard and Annual Gold Card all offer 34% discount. Minimum fares / time restrictions may apply to tickets bought with a Railcard or other discount card. See terms and conditions of the appropriate Railcard or discount card for details. Railcard holders travelling on a discounted ticket must carry their Railcard when they travel. If a Railcard holder fails to produce their valid Railcard with their ticket, they will be required to pay a full priced ticket for their journey as if no Railcard and/or no ticket were held.
Refunds	Your ticket is refundable. If you decide not to use your ticket to make all or part of your intended journey then you can get a refund by returning your unused ticket to the ticket office or place of purchase (for tickets bought via websites, telesales or travel agents) within 28 days of the ticket expiry date. You may be required to pay an administration fee (up to a maximum of £10 per ticket). The refund amount will normally take into account any use you have made of the ticket and in some circumstances no refund will be paid.
Break of Journey	Outward: Break of journey is allowed. Return: Not applicable.
Availability	Tickets can be bought in advance or immediately before travel.
Validity	
Restrictions	The ticket that you have selected may require you to travel at specific times of the day, days of the week or on a specific route. Our Journey Planner will have already taken this into account with the selection that you have made and will only have shown tickets that are valid for the selected trains. Your ticket is associated with a 'Validity Code' applied by the train companies. If you would like to see full details of the trains on which your ticket is valid view the specific 'Validity Code' applicable to the journey plan that you have selected.

Figure 174 —Usage Parameters – Example of Rail conditions. (National Rail Enquiries) (EXM)

The example shown above could be handled with USAGE PARAMETERs and VALIDITY PARAMETER ASSIGNMENTs as follows:

- Train operators offering the fare can be indicated by OPERATOR or GROUP OF OPERATORs.
- Booking deadlines can be represented by the PURCHASE WINDOW usage parameter using the ***MinimumPeriodBeforeDeparture*** and ***MaximumPeriodBeforeDeparture*** attributes.
- Discount conditions for various classes of eligible user can be represented using the USER PROFILE usage parameter.
- “Changes to travel Plans” can be represented by the EXCHANGING usage parameter using the ***ExchangableTo*** attribute to indicate the type of allowed products to which can exchange.
- Reservation conditions are represented by the RESERVING parameter for example with the ***ReservingRequirements attribute***.
- “Break of Journey can be represented by the INTERCHANGING parameter using the ***CanBreakJourney*** attribute.
- Validity can be represented by USAGE VALIDITY PERIOD. Limitations on which trains, services, types of service can be specified using VALIDITY PARAMETER ASSIGNMENTs. Peak and off-peak periods can be defined using FARE DEMAND FACTORs.

Fare conditions can also be summarised in the CONDITION SUMMARY on a FARE PRODUCT or SALES OFFER PACKAGE.

FOR EXAMPLE.

```
<ConditionSummary>
    <HasTravelTimeRestrictions>true</HasTravelTimeRestrictions>
    <HasRouteRestrictions>true</HasRouteRestrictions>
    <CanBreakJourney>false</CanBreakJourney>
    <IsRefundable>true</IsRefundable>
    <IsExchangable>true</IsExchangable>
    <HasExchangeFee>true</HasExchangeFee>
    <HasDiscountedFares>true</HasDiscountedFares>
    <HasPurchaseConditions>true</HasPurchaseConditions>
    <RequiresReservation>false</RequiresReservation>
    <HasReservationFee>true</HasReservationFee>
</ConditionSummary>
```

7.6.1.3.2.2 Different Types of Parameters – Urban Transport

An example of different parameters is shown in the following rule ‘the access right valid for students, for trips on zone 1-4 during the time period of 1 hour on all bus network LINEs of the network operator N except for LINE 278 and LINE 66 and except for the time period 2a.m.-4 a.m.” may be expressed in terms of parameters as follows:
<u>Fare structure parameters:</u>
GEOGRAPHICAL INTERVAL: 1 – 4, GEOGRAPHICAL UNIT: zone
TIME INTERVAL: 0-1, TIME UNIT: hour
and
<u>Limiting parameters (validity parameters)</u>
OPERATOR= N and MODE = bus and (LINE different 278 or LINE different 66))
and
TIMEBAND different 2 a.m.- 4a.m.
and
<u>Limiting parameters (usage parameters)</u>
USER PROFILE: student

7.6.1.3.3 **Usage Parameter – Physical model**

The following figure introduces the common physical model for USAGE PARAMETERs. The various different physical

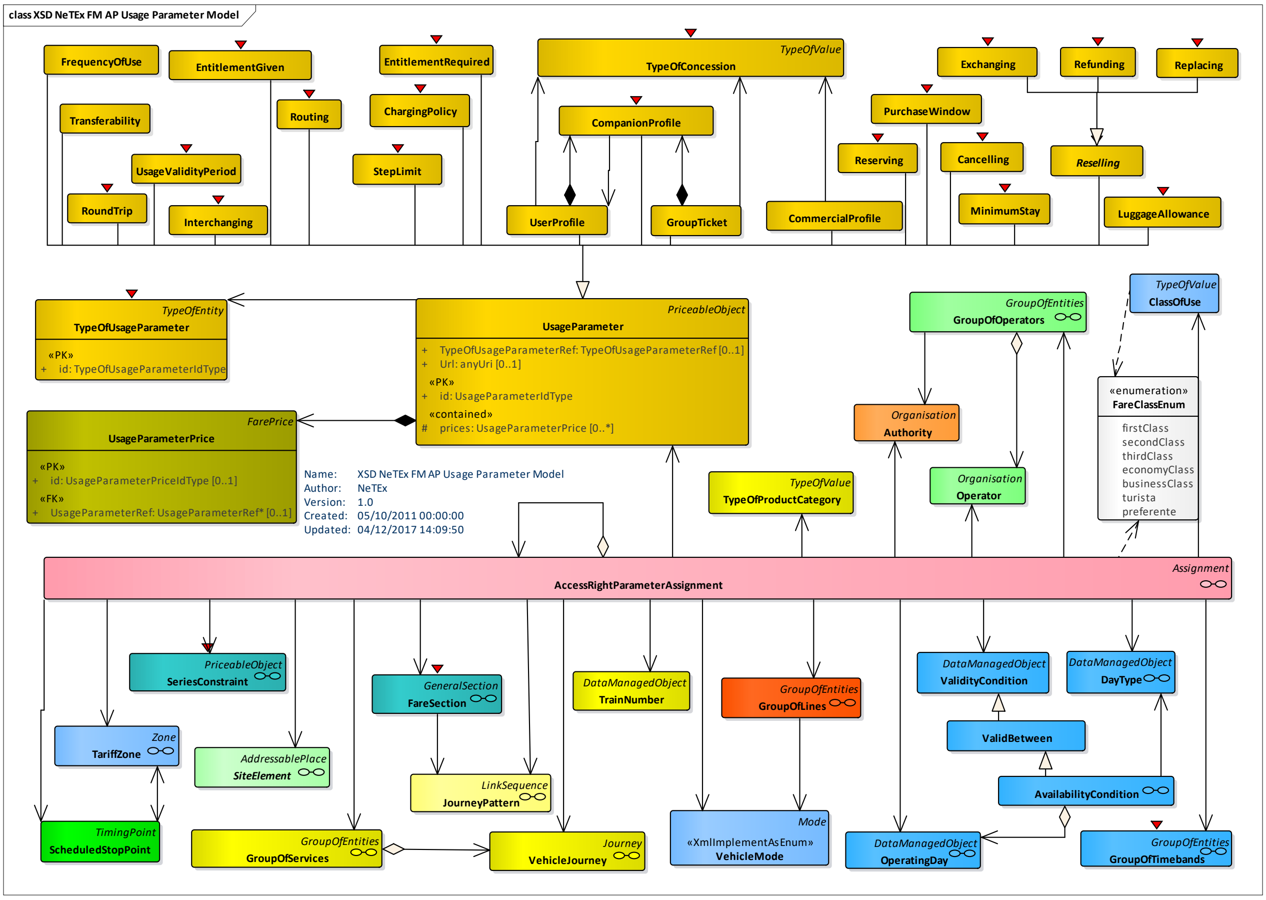


Figure 175 —Usage Parameters Summary – Physical Model (UML)

7.6.1.3.3.1 Usage Parameter: Travel – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing limits on travel.

— The ROUND TRIP parameter specifies whether single or return trips are available.

— The INTERCHANGING parameter specifies limitations on making interchanges between rides.

— The FREQUENCY OF USE parameter specifies how frequently trips may be made.

— The ROUTING parameter specifies some restrictions on which routes may be used (See also SERIES CONSTRAINT).

— The MINIMUM STAY parameter specifies if there is a restriction on how long the passenger must stay at the destination before returning.

— THE STEP LIMIT parameter specifies how many steps the trip may involve.

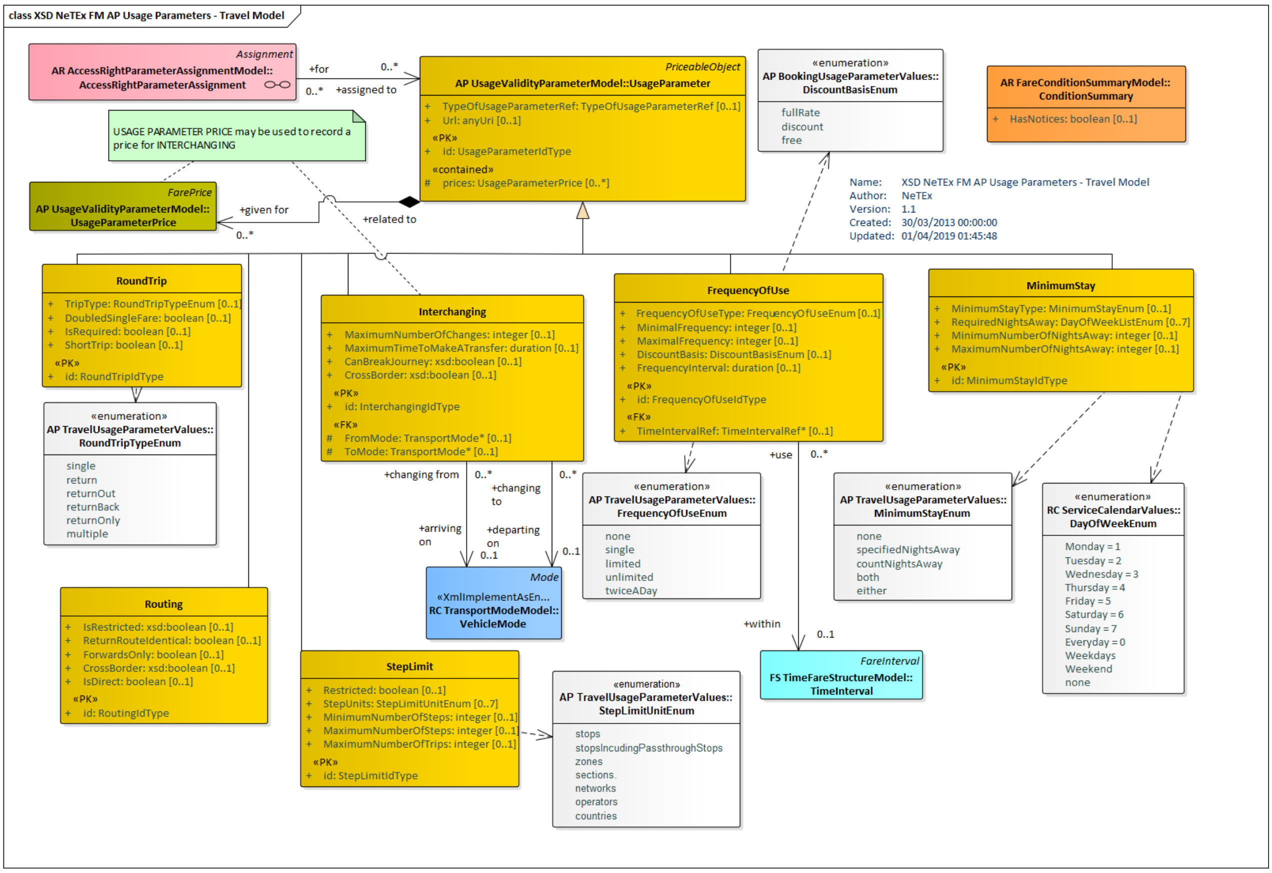


Figure 176 —Usage Parameters: Travel – Physical Model (UML)

7.6.1.3.3.2 Usage Parameter: Travel Validity – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing usage validity limits on travel.

- The USAGE VALIDITY PERIOD parameter specifies any limitations on how long the passenger may travel.
- The SUSPENDING parameter specifies conditions governing temporary suspension of a FARE PRODUCT, (i.e. a period pass or subscription).

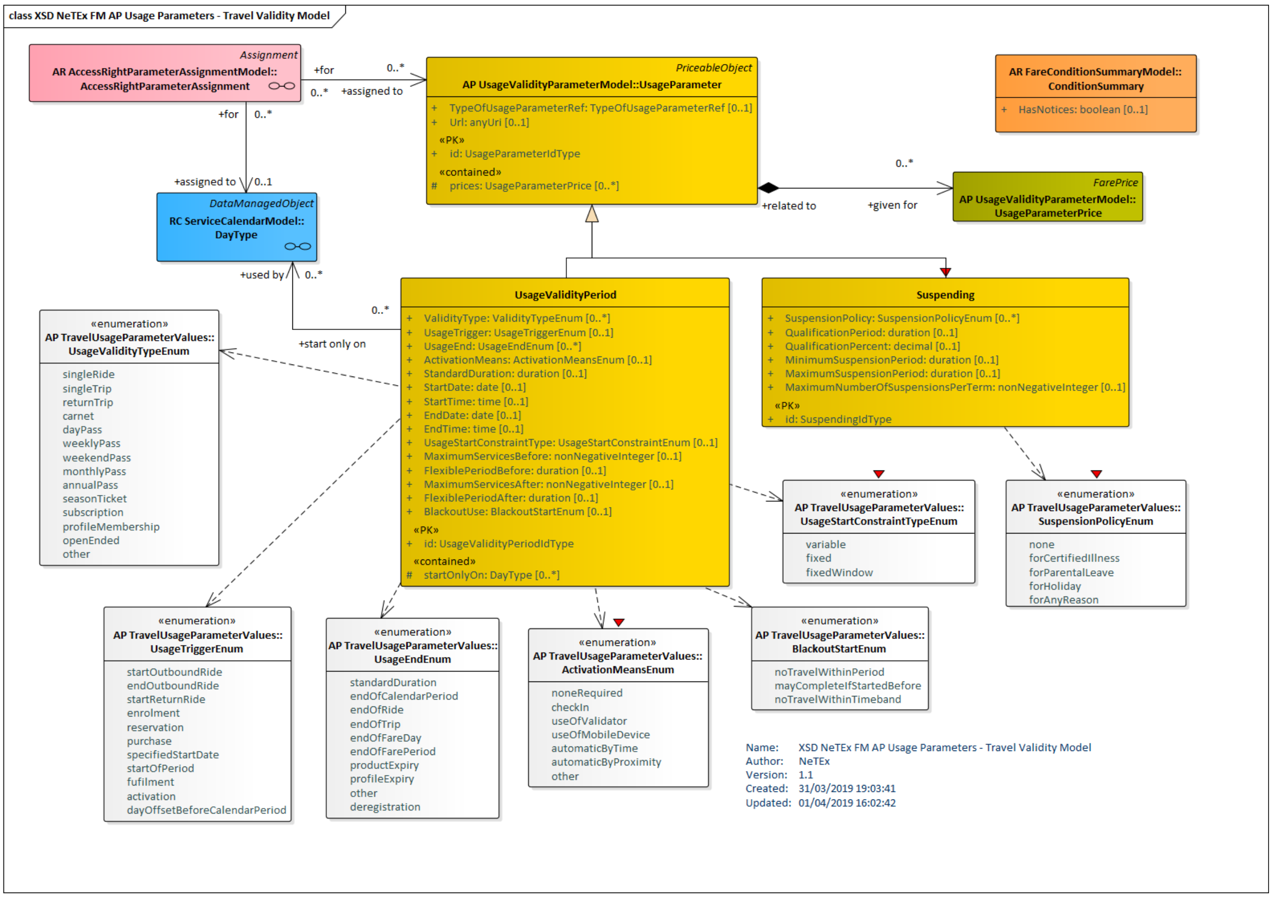


Figure 177 —Usage Parameters: Travel Validity – Physical Model (UML)

7.6.1.3.3.3 Usage Parameter: Eligibility – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing limits on eligibility for products.

— The USER PROFILE parameter specifies who may use a product. Residential restrictions may be specified by a RESIDENTIAL QUALIFICATION.

— The COMPANION PROFILE parameter specifies any limitations companions or members of a GROUP TICKET.

— The COMMERCIAL PROFILE parameter specifies properties relating to Frequent Traveller offers, such as for how many miles a trip counts.

— TYPE OF CONCESSION describes classes of users who may be eligible for GROUP TICKETs or USER PROFILEs.

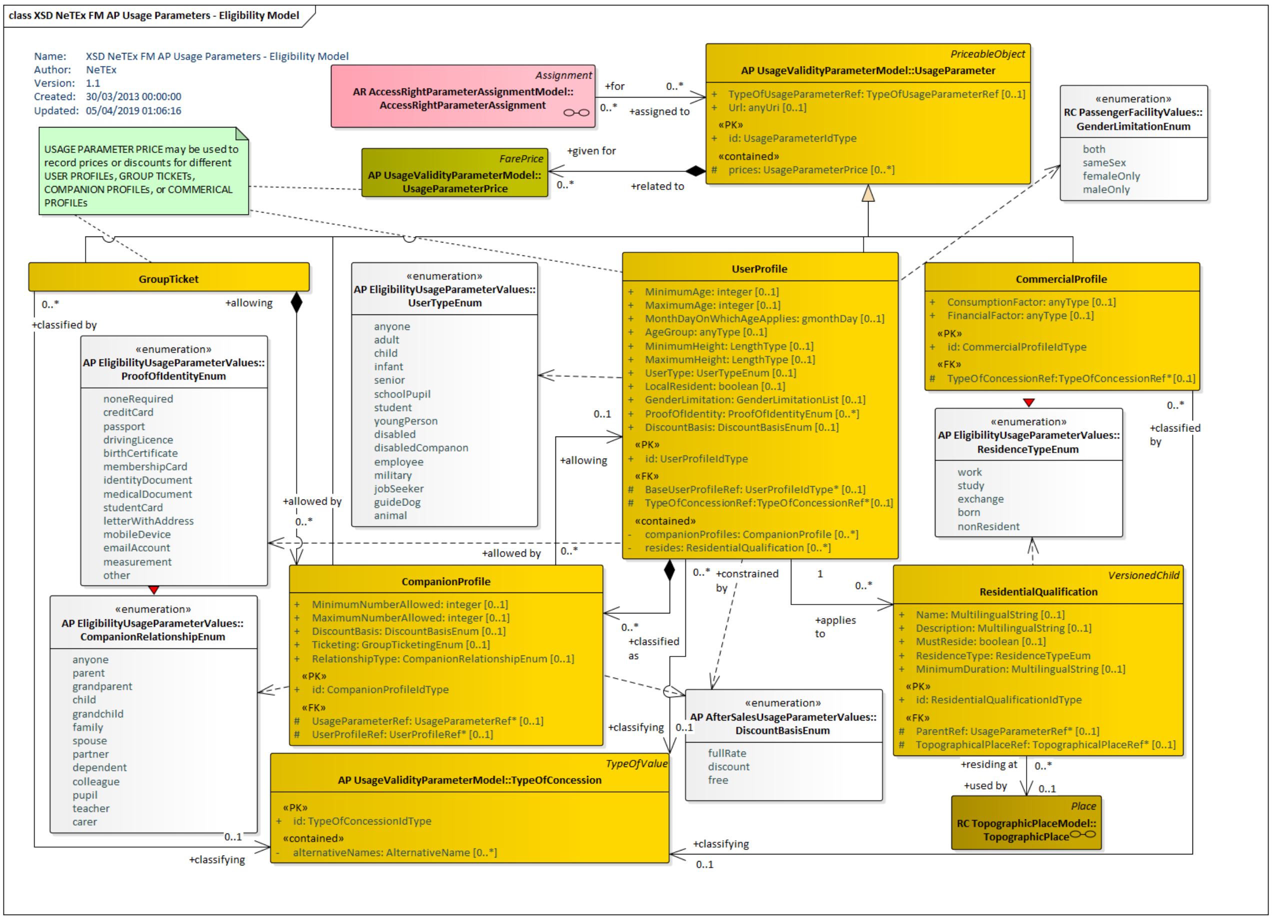


Figure 178 —Usage Parameters: Eligibility – Physical Model

7.6.1.3.3.4 Usage Parameter: Group Ticket Eligibility – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing limits on eligibility for group products.

— The GROUP TICKET parameter specifies properties of group tickets and who may use them.

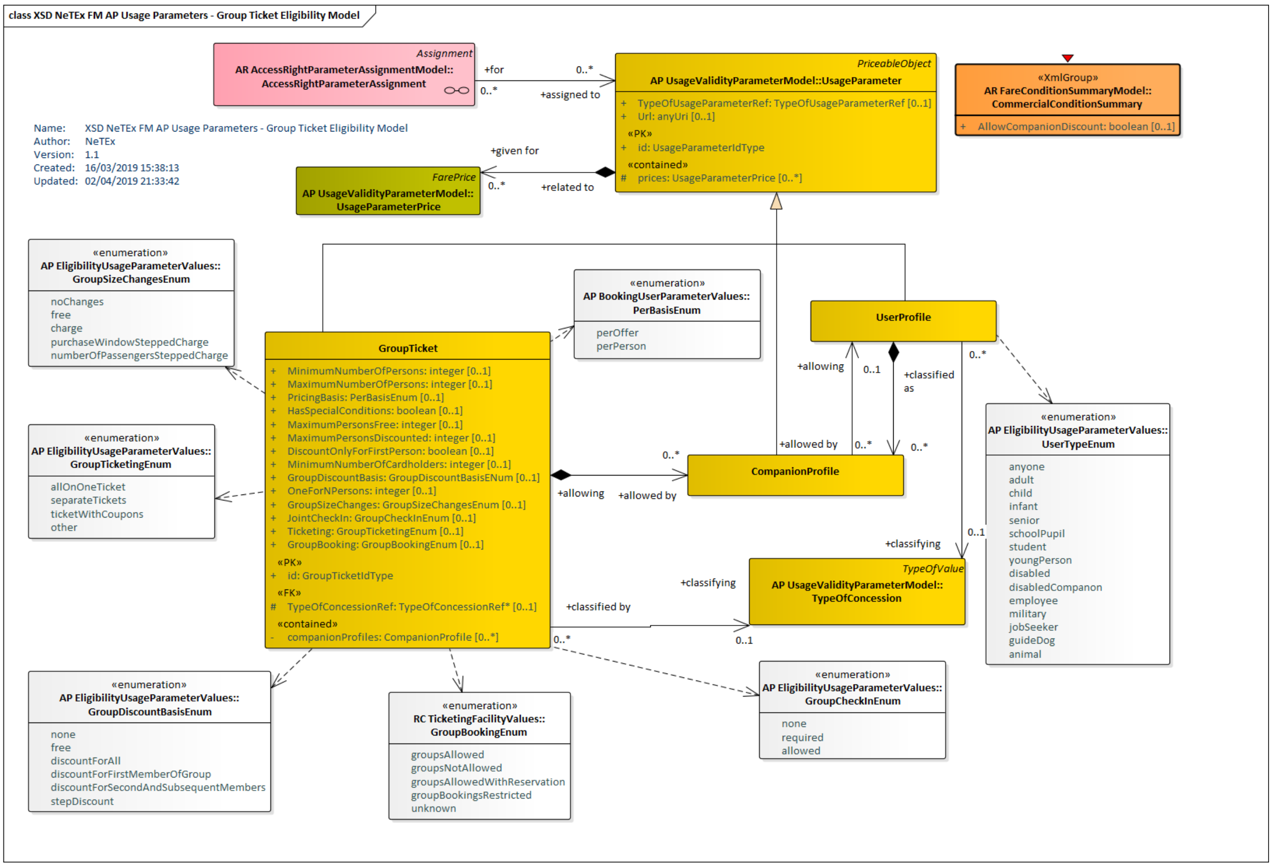


Figure 179 —Usage Parameters: Group Ticket Eligibility – Physical Model

7.6.1.3.3.5 Usage Parameter: Eligibility Change Policy – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing policy on changes eligibility for products.

— The ELIGIBILITY CHANGE POLICY parameter specifies what should happen if a user ceases or becomes eligible for a product.

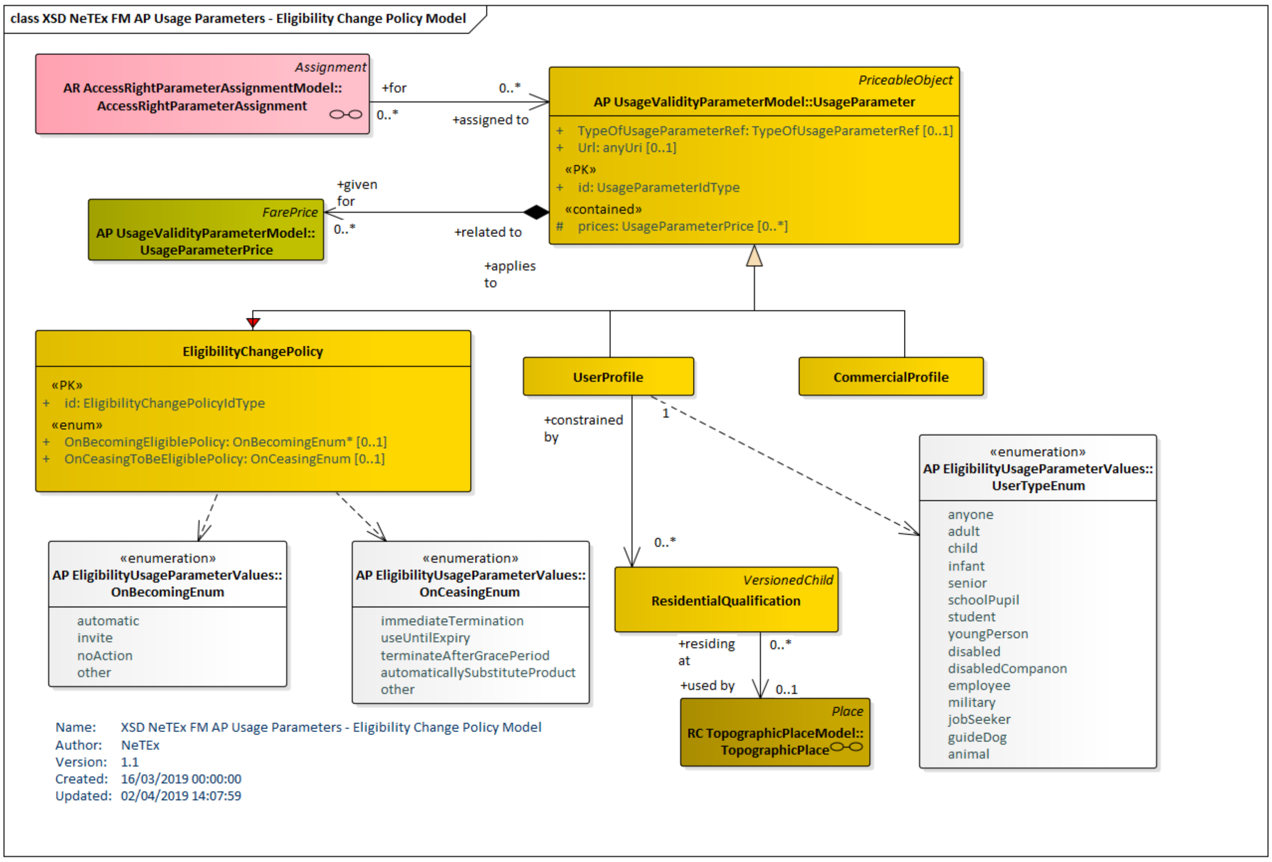


Figure 180 —Usage Parameters: Eligibility Change Policy – Physical Model

7.6.1.3.3.6 Usage Parameter: Entitlement – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing entitlements required or given by products.

- The ENTITLEMENT REQUIRED parameter specifies any prerequisite products for purchasing the product.
- The ENTITLEMENT GIVEN parameter specifies any entitlements given by the product.

Further conditions on the entitlement may be expressed using ENTITLEMENT CONSTRAINTs – see below

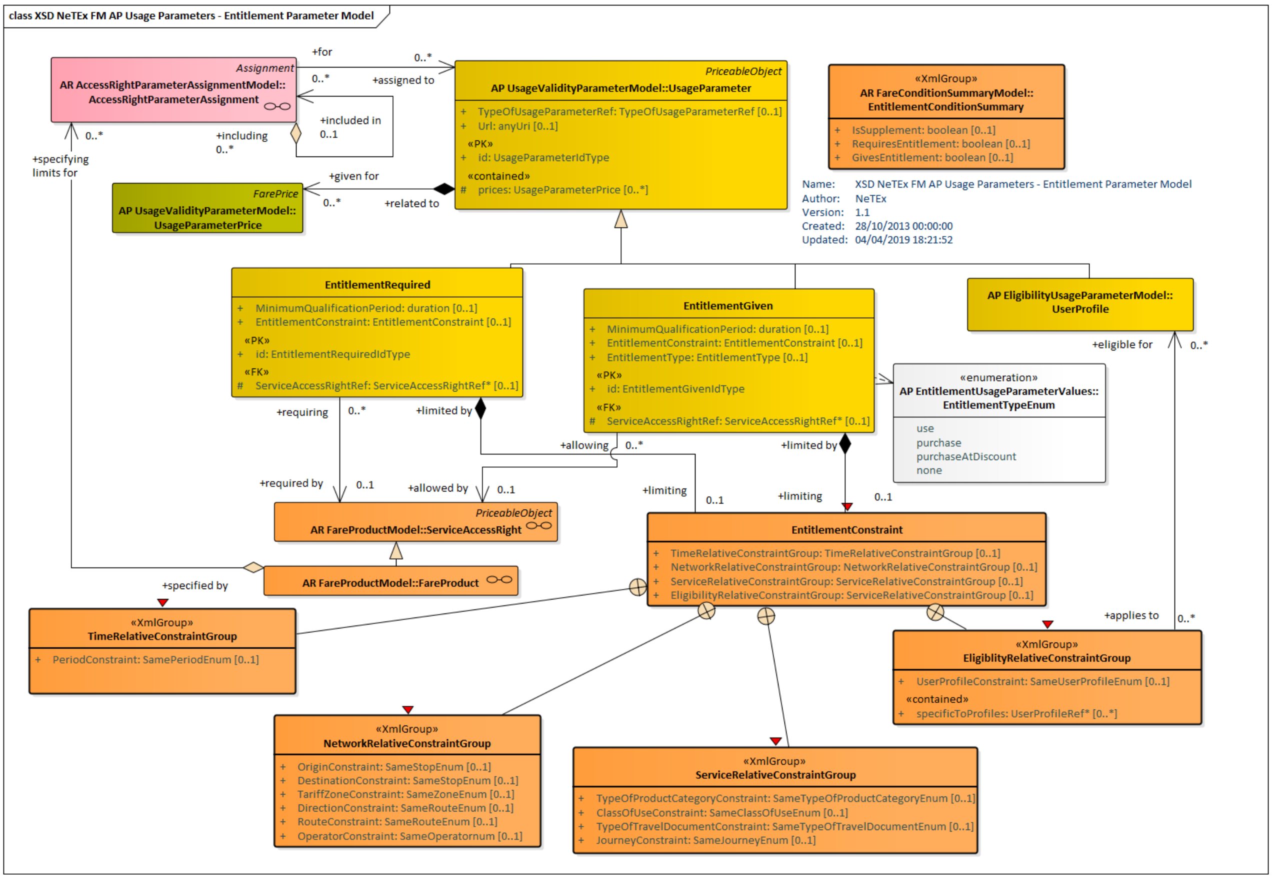


Figure 181 —Usage Parameters: Product Entitlement– Physical Model

7.6.1.3.3.7 Usage Parameter: Entitlement Constraints – Physical model

Where a product gives rights to another product, the dependent product may be constrained to particular properties of the prerequisite product. For example, a rail ticket might entitle the user to buy a local bus day pass at either end of the rail journey at a reduced rate; the allowed zones to purchased for the dependent product are restricted to the choices of origin and destination station made in the prerequisite product, and the day must be the same day of travel (or days of travel if it is a period return on different days).Such relative constraints can be expressed on an entitlement entity using an ENTITLEMENT CONSTRAINT. Other examples might include a requirement to use the same media (TYPE OF TRAVEL DOCUMENT), to be the same type of user (i.e. USER PROFILE), or to use the same OPERATOR.

The ENTITLEMENT CONSTRAINT attribute allows a number of different constraints to be specified.

— Temporal: restricting the use of the dependent product to similar time values as the prerequisite product, e.g. the same day, or the same period

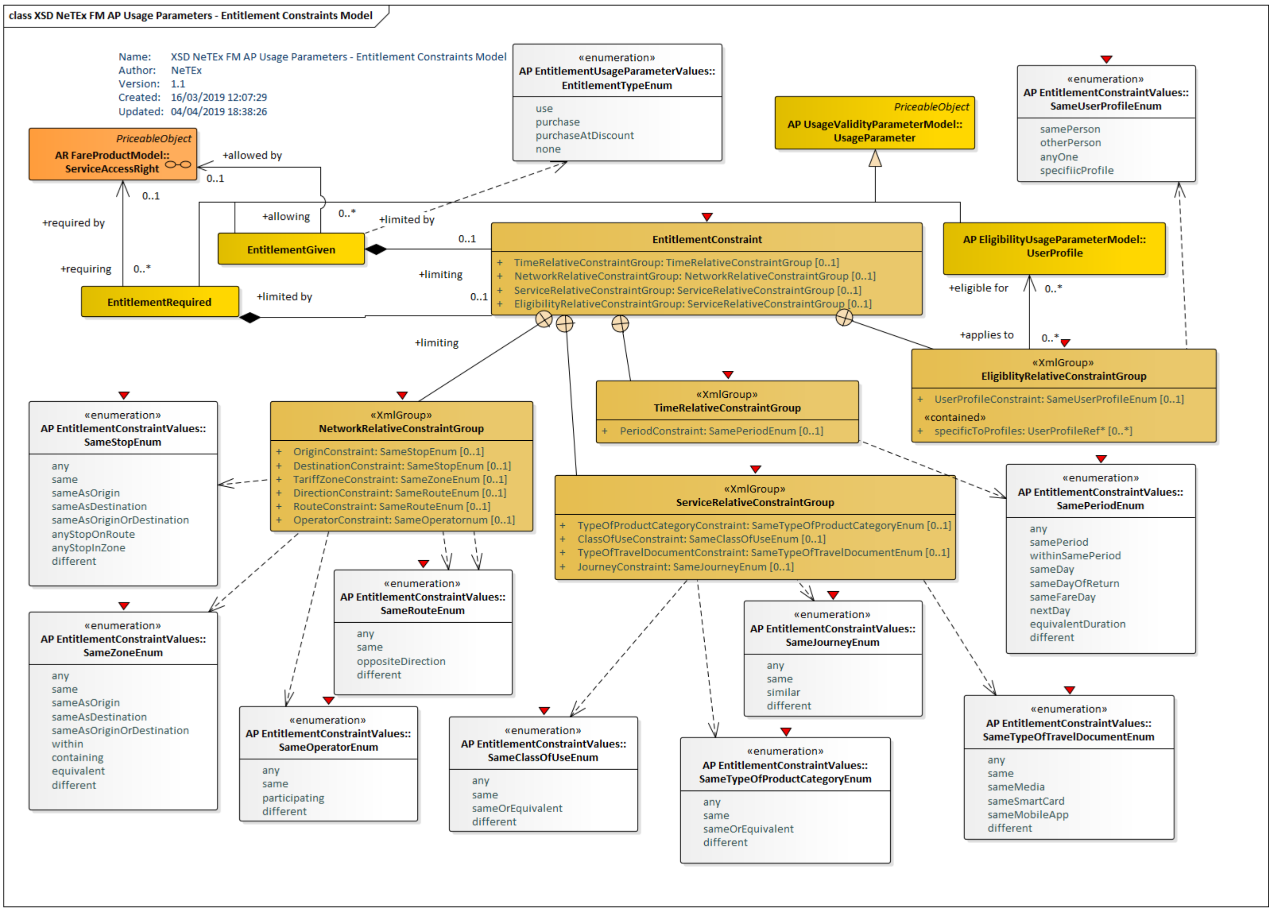
— Network: restricting the use of the dependent product to the same network as the prerequisite product, e.g. the same origin and destination, or stops on a route

— Service: restricting the use of the dependent product d to the similar service elements as the prerequisite product, for example the same class of use, or type of train (TYPE OF PRODUCT CATEGORY).

— Eligibility: restricting the use of the dependent product to the user profiles with a specifc relationship to the product holder, for example, the same person or family members.

Note that a SUPPLEMENT PRODUCT for a trip such as a seat reservation is normally assumed to be constrained to the same values as for the prerequisite product trip.

Figure 182 —Usage Parameters: Entitlement Relative Constraints– Physical Model



7.6.1.3.3.8 Usage Parameter: Luggage – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing limits on baggage.

— The LUGGAGE ALLOWANCE parameter specifies limitations on taking luggage.

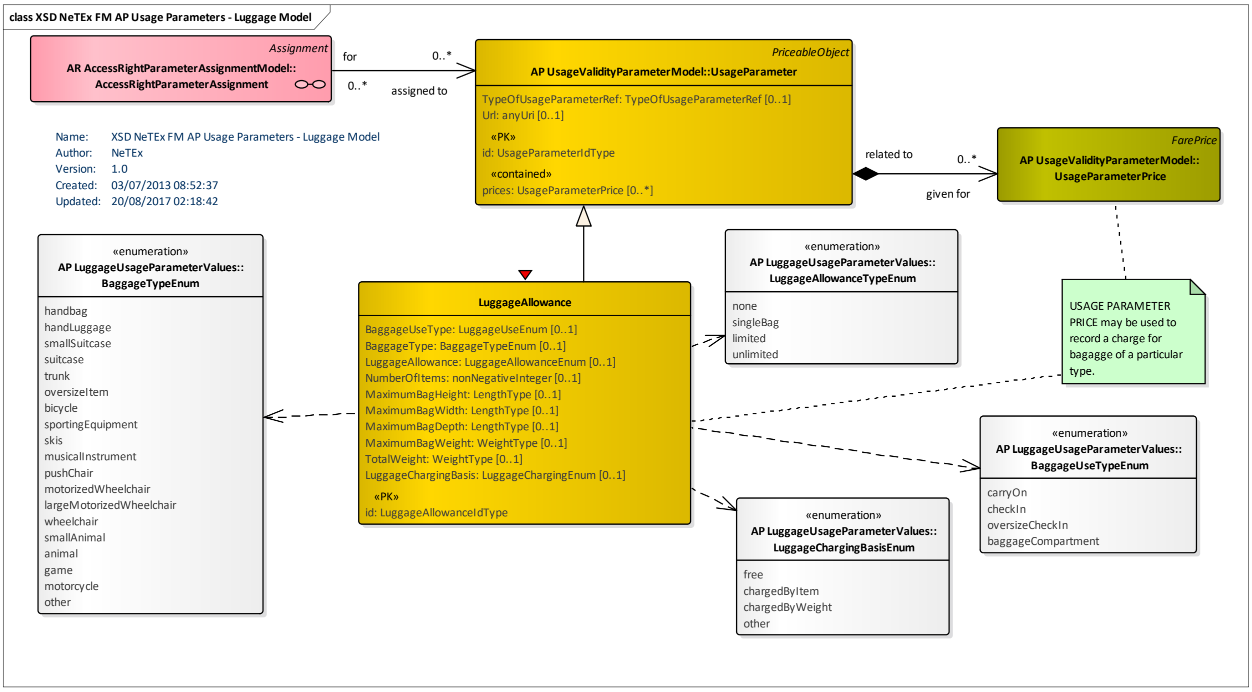


Figure 183 —Usage Parameters: Luggage – Physical Model

7.6.1.3.3.9 Usage Parameter: Booking – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing limits on products.

— The PURCHASE WINDOW parameter specifies limitations on when a product may be purchased.

— The RESERVING parameter specifies requirements for making reservations for a product, including information on BOOKING ARRANGEMENTs.

— The CANCELLING parameter specifies requirements for cancelling a booking.

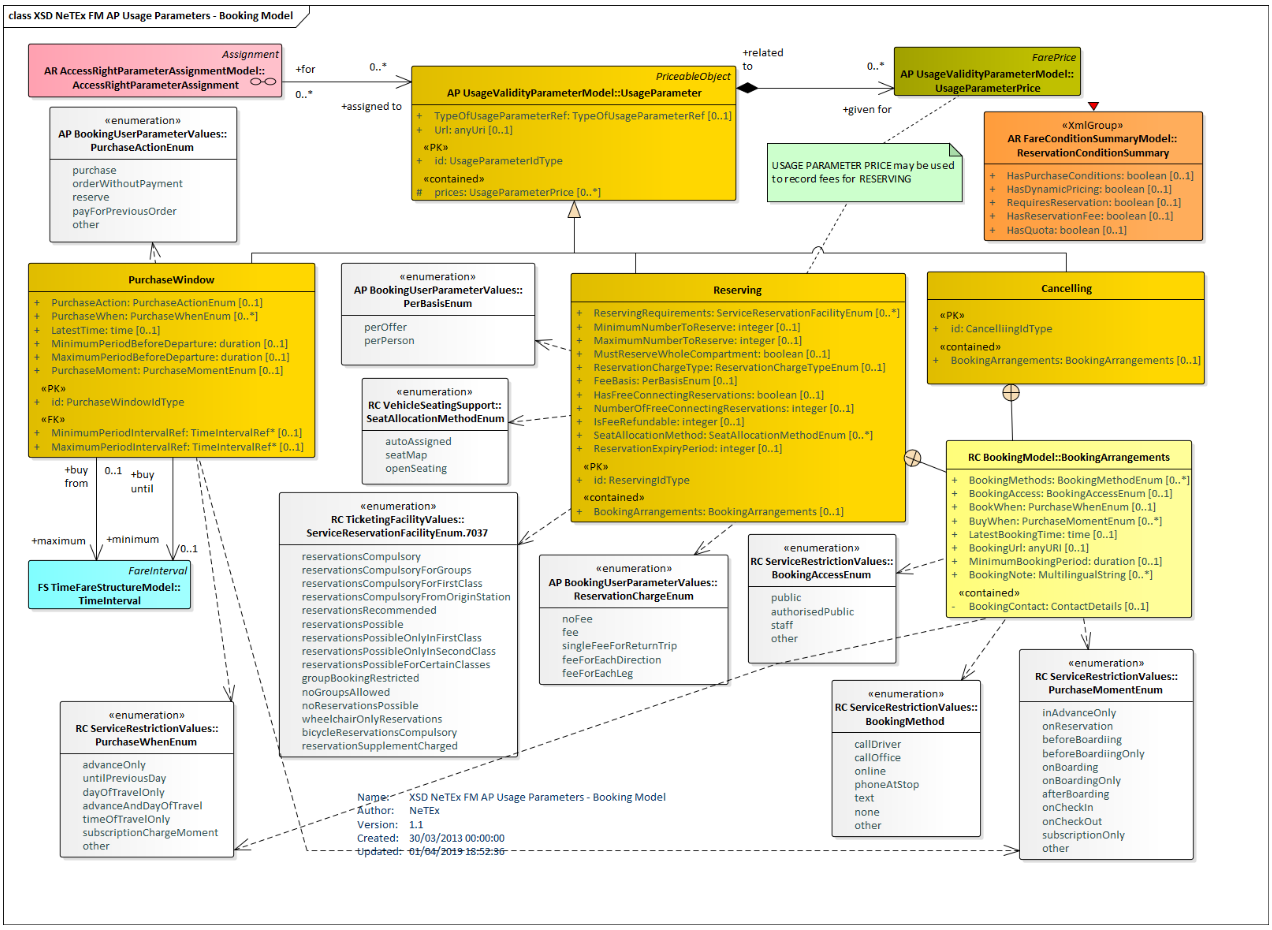


Figure 184 —Usage Parameters: Booking – Physical Model (UML)

7.6.1.3.3.10 Usage Parameter: After Sales – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing after sales conditions on products.

- The TRANSFERABILITY parameter specifies limitations on transferring a ticket to someone else.
- The REPLACING parameter specifies whether the product can be replaced if lost or stolen.
- The REFUNDING parameter specifies limitations on refunds for a product and other resale properties.
- The EXCHANGING parameter specifies limitations on exchanging tickets for other tickets.

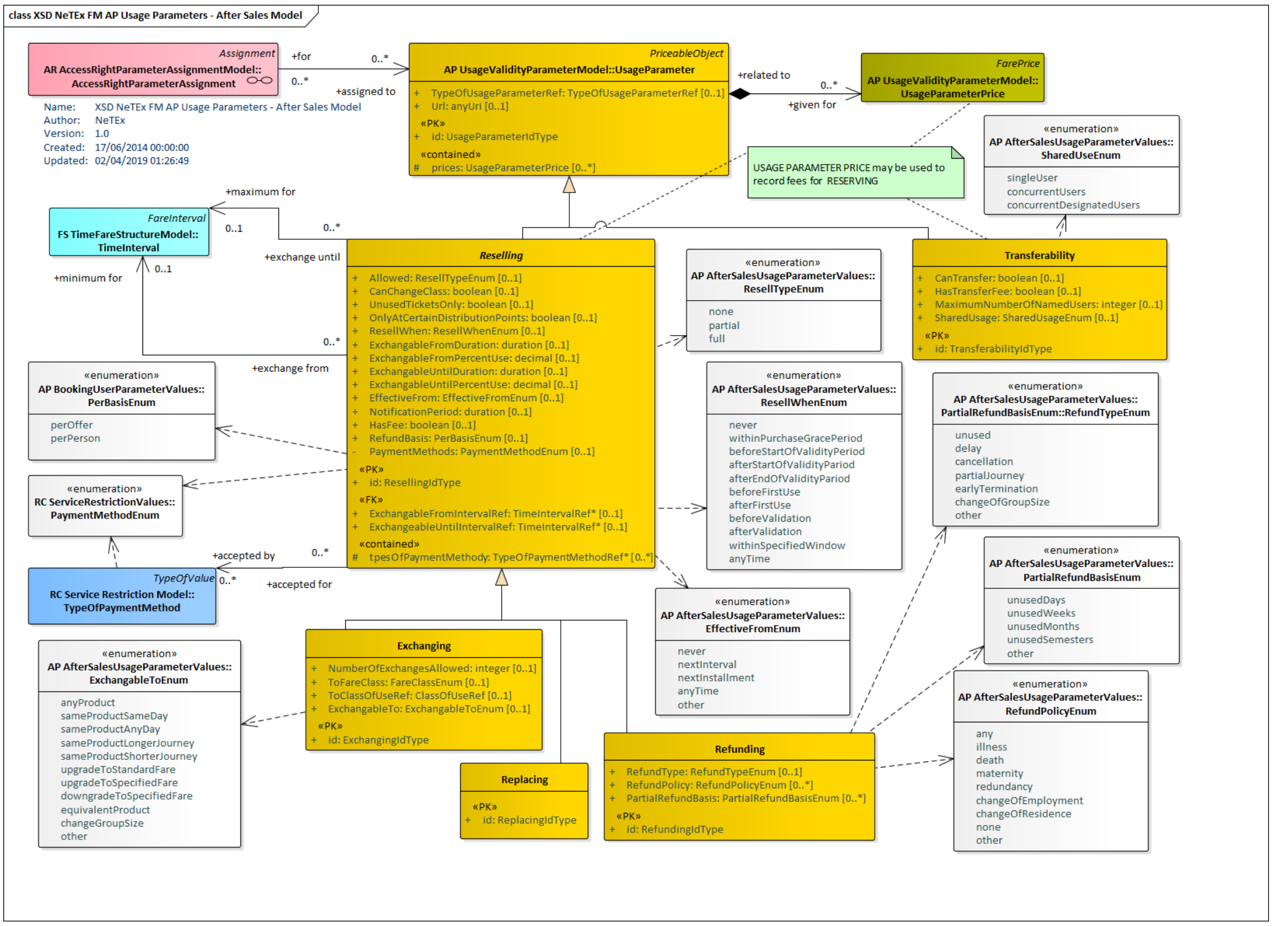


Figure 185 —Usage Parameters: Product After Sales – Physical Model (UML)

7.6.1.3.3.11 Usage Parameter: Charging – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing polices for charging for products.

— The CHARGING POLICY parameter specifies limitations on how a product may be charged. May be used to specify a minimum and maxim credit level.

— The PENALTY POLICY parameter specifies rules relating to penalty fares that may be incurred.

— The SUBSCRIBING parameter specifies rules relating to products bought on subscription, that is an agreement to make regular payments over a specified period in return for a discounted price.

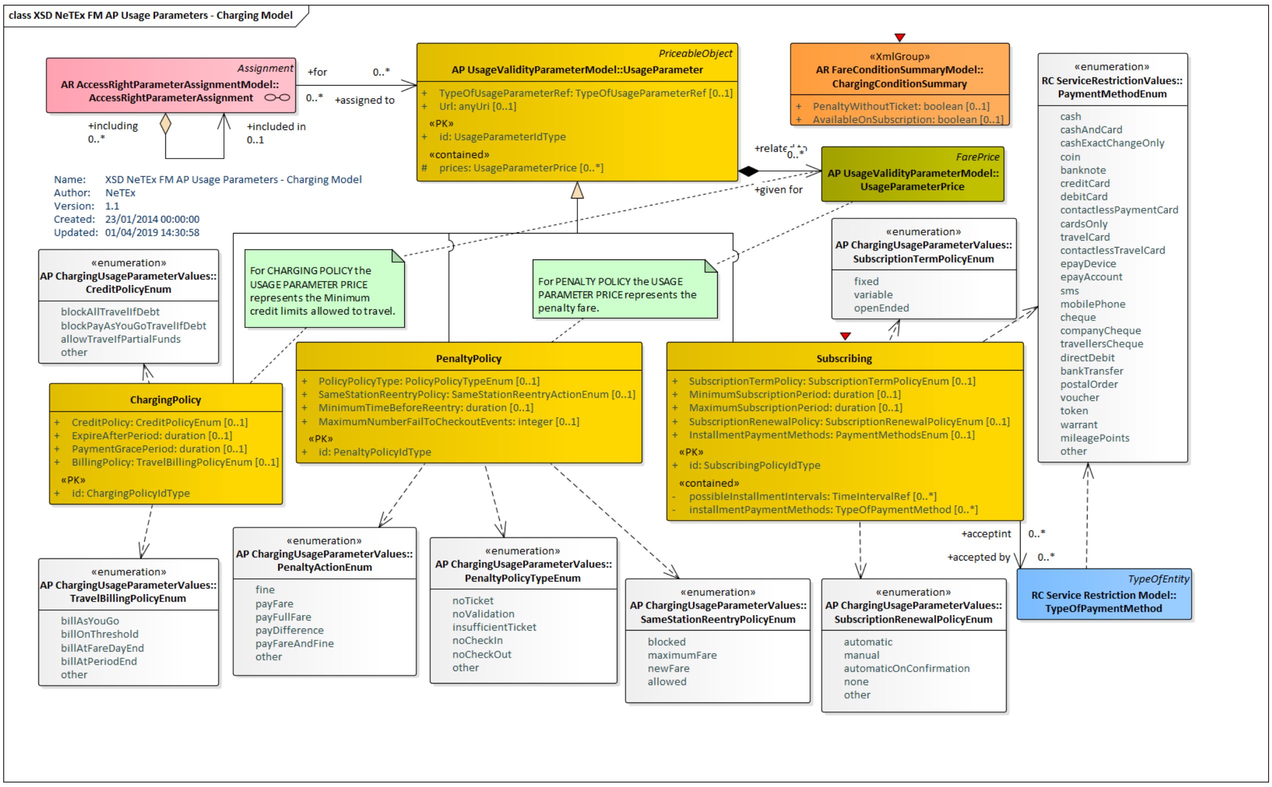


Figure 186 —Usage Parameters: ChargingPolicy – Physical Model (UML)

7.6.1.3.3.12 Usage Parameter Prices

Although all USAGE PARAMETERs have a PRICE the price has a different purpose in different cases.

Table 121 – *USAGE PARAMETERs* – Meaning of prices

Group	Usage Parameter	Comment on Price
Travel	ROUND TRIP	Pricing of single or return trip.
	USAGE VALIDITY PERIOD	Pricing for a trip of this length.
	FREQUENCY OF USE	Pricing for a product subject to this frequency of use constraint, if any
	INTERCHANGING	Price of making an interchange, if any
	MINIMUM STAY	Price for a minimum stay.
	STEP LIMIT	Pricing for a trip with the specified number of steps.
	ROUTING	Pricing for a route with these restrictions.
Eligibility	USER PROFILE	Price for a user of this type, if any
	COMMERCIAL PROFILE	Pricing for the commercial offer.
	GROUP TICKET	Pricing for a group ticket of this type.

	COMPANION PROFILE	Pricing for taking a companion along, if any.
	RESIDENTIAL QUALIFICATION	Pricing for someone who meets this residential criterion, if any.
Entitlement	ENTITLEMENT REQUIRED	Pricing for use for required product, if any
	ENTITLEMENT GIVEN,	Discount that gives to other product, if any
Luggage	LUGGAGE ALLOWANCE	Price of luggage carriage of the specified type, if any
Booking	PURCHASE WINDOW	Price of a ticket bought within the specified window, if any
	TRANSFERABILITY	Price of transferability, if any
	EXCHANGING	Fee for exchanging ticket, if any
	REFUNDING	Fee for refunding ticket, if any.
	REPLACING	Fee for replacing a lost ticket, if any.
	RESERVING	Fee for reserving a ticket, if any.
Charging	CHARGING POLICY	Limits of credit associated with product.
	PENALTY POLICY	Fees for use deemed to incur penalty.
	SUBSCRIBING	Fee for setting up a subscription.
	SUSPENDING	Fee for suspending a perod pass or subscription.

7.6.1.3.4 **Usage Parameter: General – Attributes and XSD**

7.6.1.3.4.1 UsageParameter – Model Element

A parameter used to specify the use of a SALES OFFER PACKAGE or a FARE PRODUCT.

Table 122 – *UsageParameter* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	USAGE PARAMETER inherits from PRICEABLE OBJECT.
«PK»	***id***	*UsageParameterIdType*	1:1	Identifier of USAGE PARAMETER.
	Url	*xsd:anyUri*	0:1	Url associated with parameter.
«FK»	***TypeOf-UsageParameter-Ref***	*TypeOf-UsageParameterRef*	0:1	Type of USAGE PARAMETER.
«cntd»	***fareTables***	*FareTable*	0:*	FARE TABLEs for the USAGE PARAMETER.

«cntd»	***prices***	*UsageParameterPrice*	0:*	USAGE PARAMETER PRICEs for the USAGE PARAMETER.

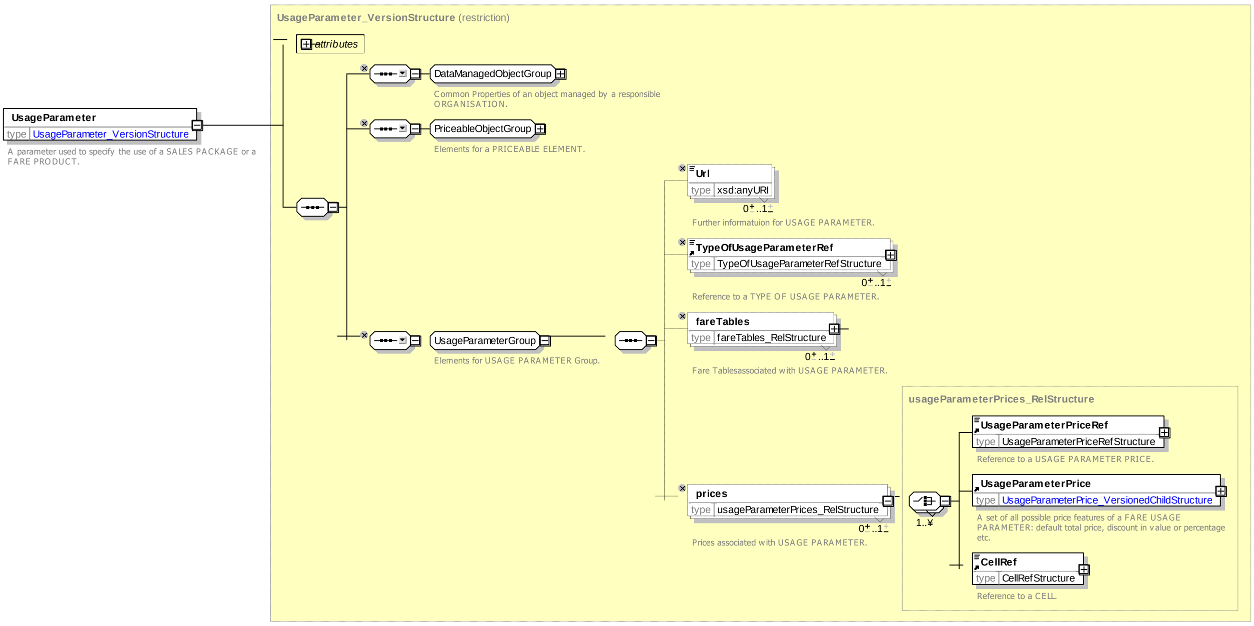


Figure 187 — *UsageParameter* — XSD

7.6.1.3.4.2 UsageParameterPrice – Model Element

A set of all possible price features of a USAGE PARAMETER: discount in value or percentage etc.

Table 123 – *UsageParameterPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	USAGE PARAMETER PRICE inherits from FARE PRICEs
«PK»	***id***	*UsageParameterPrice-IdType*	0:1	Identifier of USAGE PARAMETER PRICE.
«FK»	***UsageParameter Ref***	*UsageParameterRef+*	0:1	USAGE PARAMETER for which this is the PRICE.

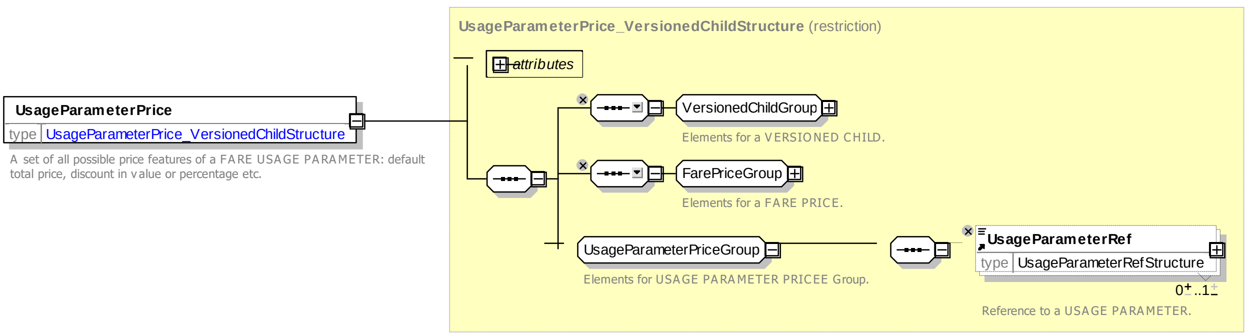


Figure 188 — *UsageParameterPrice* — XSD

7.6.1.3.4.3 TypeOfUsageParameter – Model Element

A classification of USER PROFILE by type of person eligible to use it

Table 124 – *TypeOfUsageParameter* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF USAGE PARAMETER inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*TypeOfUsageParameter-IdType*	1:1	Identifier of TYPE OF USAGE PARAMETER.

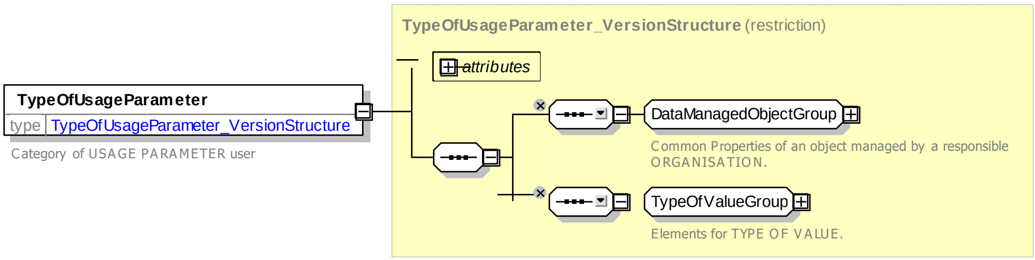


Figure 189 — *TypeOfUsageParameter* — XSD

7.6.1.3.5 Usage Parameter: Travel – Attributes and XSD

7.6.1.3.5.1 RoundTrip – Model Element

Properties relating to single or return trip use of an access right.

Table 125 – *RoundTrip* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	ROUND TRIP inherits from USAGE PARAMETER.
«PK»	***id***	*RoundTripIdType*	1:1	Identifier of ROUND TRIP.
	TripType	*xsd:boolean*	0:1	Whether return trip is allowed.
	Double-SingleFare	*xsd:boolean*	0:1	Whether fare for return trip is single fare doubled.
	ShortTrip	*xsd:boolean*	0:1	Whether trip is classified as a short trip for fares.
	IsRequired	*xsd:boolean*	0:1	Whether return trip is required.

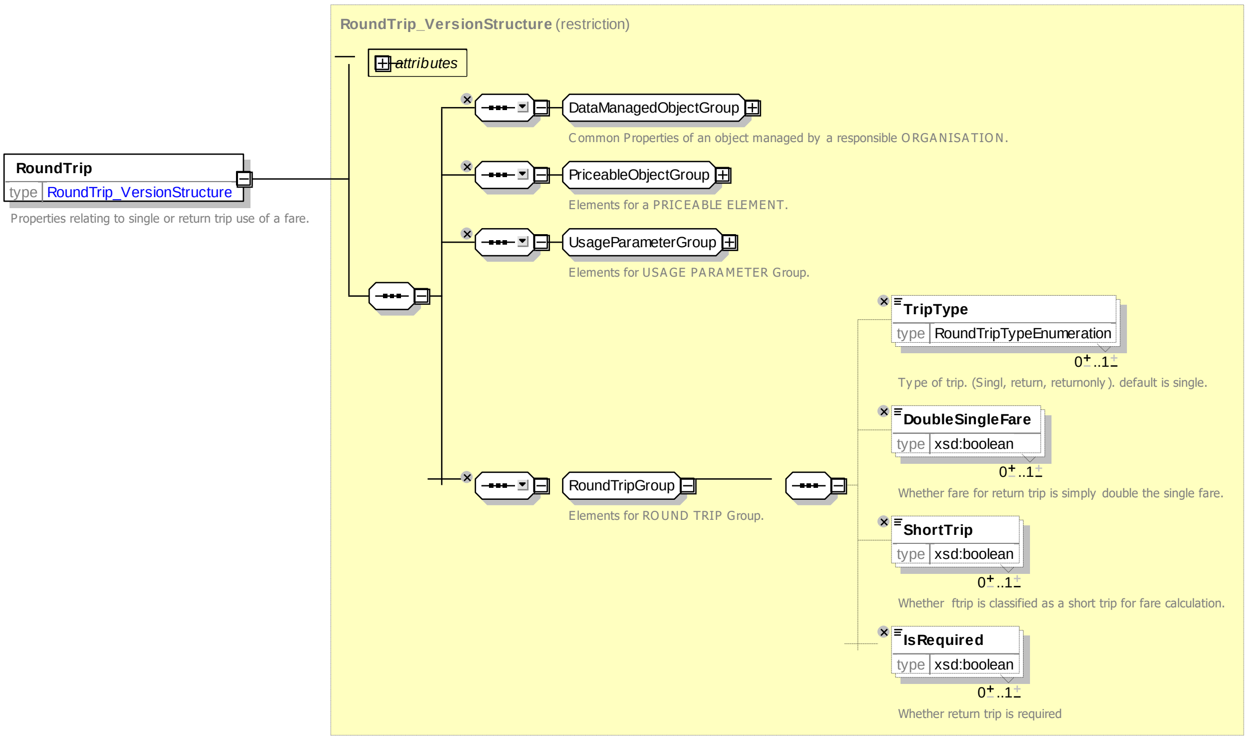


Figure 190 — *RoundTrip* — XSD

7.6.1.3.5.1.1 *RoundTripType* – Allowed values

The following table shows the allowed values for ***RoundTripType*** *(RoundTripTypeEnumeration)*.

Table 126 – *RoundTripType* – Allowed values

Value	Description
single	Single trip.
return	Outbound and return trip.
returnOut	Outbound leg of return journey. +v1.1
returnBack	Return leg of return journey. +v1.1
returnOnly	Return trip only
multiple	Multiple trip.

7.6.1.3.5.2 Routing – Model Element

Limitations on routing of an access right.

Table 127 – *Routing* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*UsageParameter*	::>	ROUTING inherits from USAGE PARAMETER.
«PK»	***id***	*RoutingIdType*	1:1	Identifier of ROUTING.
	Return-RouteIdentical	*xsd:boolean*	0:1	Whether return route must be same as outbound route.

	ForwardsOnly	*xsd:boolean*	0:1	Whether passenger may only take routes that proceed in a single direction. (They may not use product to achieve a return trip for the cost of a single trip).
	IsRestricted	*xsd:boolean*	0:1	Whether only allowed on certain routes or series.
	CrossBorder	*xsd:boolean*	0:1	Whether the routing is across a border.

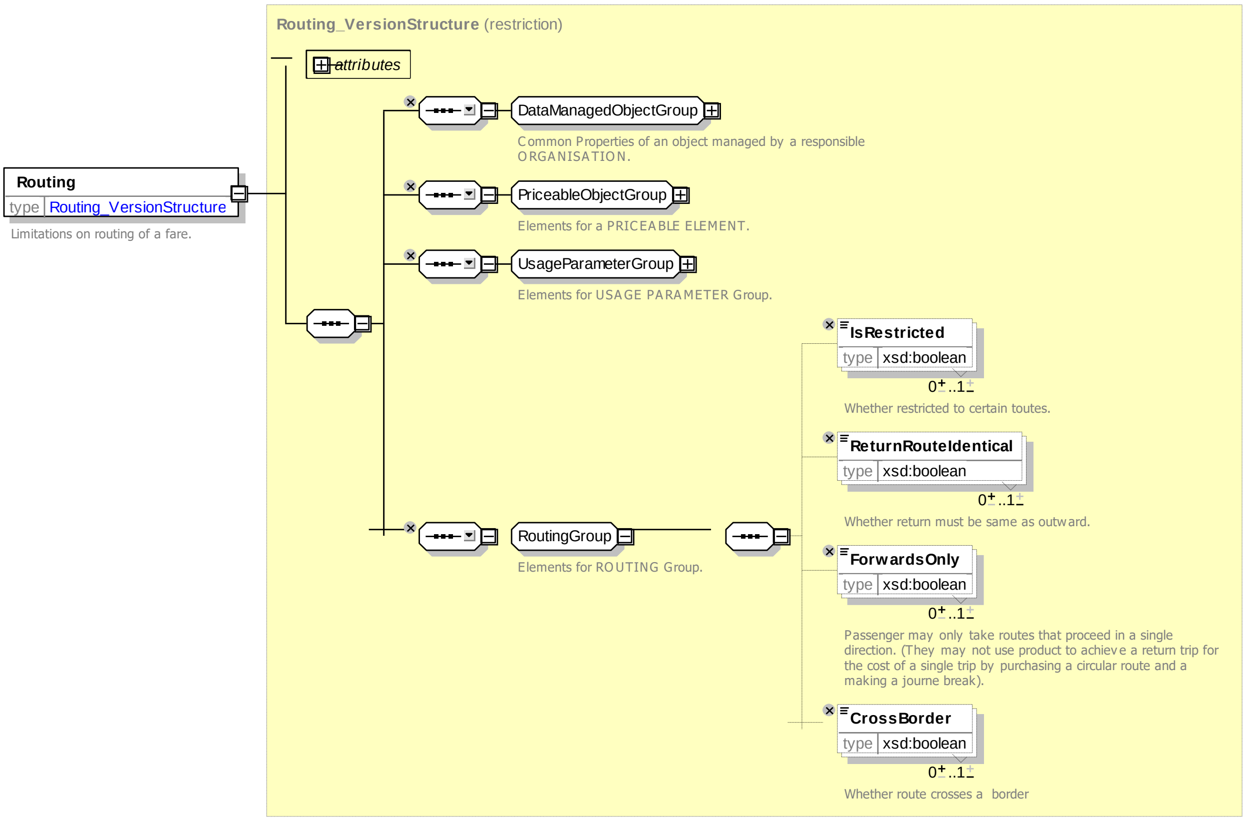


Figure 191 — *Routing* — XSD

7.6.1.3.5.3 FrequencyOfUse – Model Element

The limits of usage frequency for a FARE PRODUCT (or one of its components) or a SALES OFFER PACKAGE during a specific VALIDITY PERIOD. There may be different tariffs depending on how often the right is consumed during the period.

Table 128 – *FrequencyOfUse* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	FREQUENCY OF USE inherits from USAGE PARAMETER.
«PK»	***id***	*FrequencyOfUseIdType*	1:1	Identifier of FREQUENCY OF USE.

«enum»	***Frequency-OfUseType***	*FrequencyOfUseEnum*	0:1	Type of Frequency of Use. See allowed values below.
	Minimal-Frequency	*xsd:integer*	0:1	Minimum number of times can be used.
	Maximal-Frequency	*xsd:integer*	0:1	Maximum number of times can be used.
	Frequency-Interval	*xsd:duration*	0:1	Interval within which frequency is measured. If absent forever.
«FK»	***TimeIntervalRef***	*TimeIntervalRef*	0:1	Interval within which frequency is measured. - as reference to arbitrary time interval.
«enum»	***DiscountBasis***	*DiscountBasisEnum*	0:1	Nature of discount for number of journeys. See allowed values below.

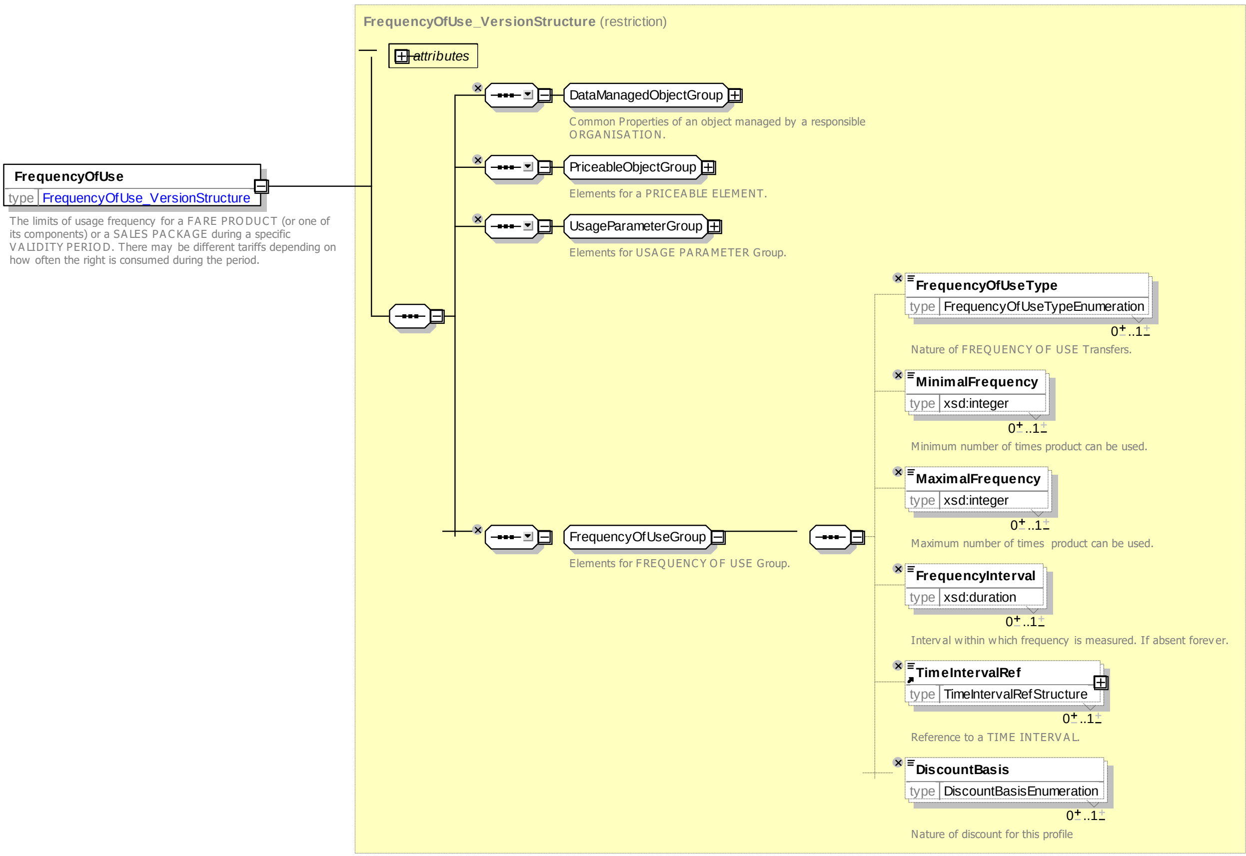


Figure 192 — *FrequencyOfUse* — XSD

7.6.1.3.5.3.1 FrequencyOfUseType – Allowed values

The following table shows the allowed values for ***FrequencyOfUseType*** (*FrequencyOfUseEnumeration*).

Table 129 – *FrequencyOfUseType* – Allowed values

Value	Description
none	No use changes allowed.
single	Single use allowed.

limited	Limited use allowed.	*unlimited*	Unlimited use allowed.	*twiceADay*	Can be used twice a day.

7.6.1.3.5.3.2 DiscountBasis – Allowed values

The following table shows the allowed values for ***DiscountBasis*** (*DiscountBasisEnumeration)*.

Table 130 – *DiscountBasis* – Allowed values

Value	Description
none	No companion allowed.
free	Companion allowed for free
discount	Companion allowed at discount

7.6.1.3.5.4 Interchanging – Model Element

Limitations on making changes within a trip.

Table 131 – *Interchanging* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	<u>*UsageParameter*</u>	::>	INTERCHANGING inherits from USAGE PARAMETER.
«PK»	***id***	*InterchangingIdType*	1:1	Identifier of INTERCHANGING.
	CanInterchange	*xsd:boolean*	0:1	Whether an interchange can be made.
«enum»	***FromMode***	*VehicleModeEnum*	0:1	TRANSPORT MODE from which user is interchanging. See NeTEx Part1 for allowed values.
«enum»	***ToMode***	*VehicleModeEnum*	0:1	TRANSPORT MODE to which user is interchanging. See NeTEx Part1 for allowed values.
	Maximum-NumberOf-Changes	*xsd:integer*	0:1	Maximum number of transfers that can be made on a trip.
	MaximumTime-ToMakeA-Transfer	*xsd:duration*	0:1	Maximum time allowed to make a transfer.
	CanBreak-Journey	*xsd:boolean*	0:1	Whether the journey can be broken at an interchange point.
	CrossBorder	*xsd:boolean*	0:1	Whether the interchange is across a border.
«enum»	***RegisterBreakOf Journey***	*RegisterBreakOfJourney-Enum*	0:*	Whether the Journey can be interrupted, i.e. leave stop point and return. See allowed values below. +v1.1

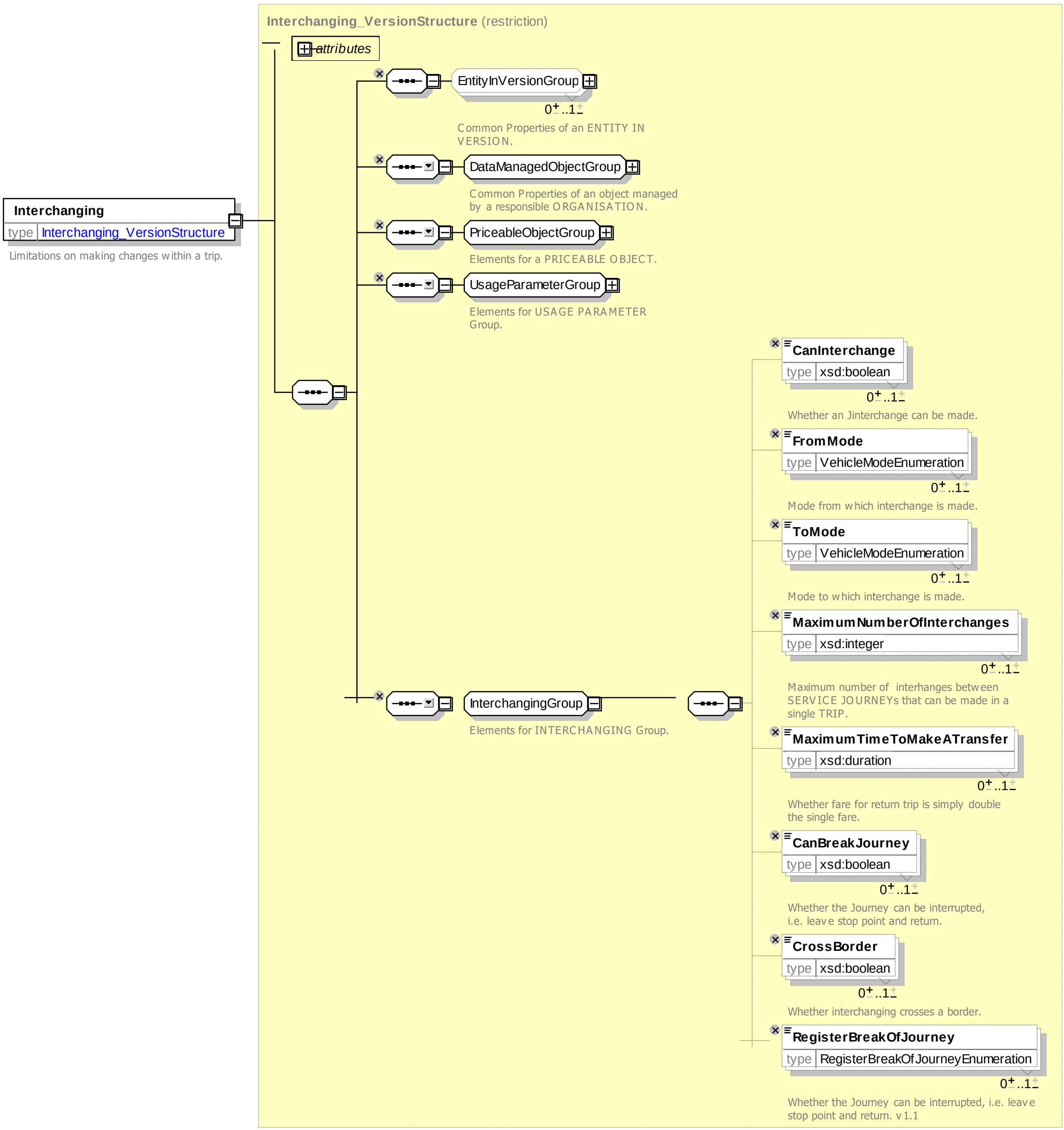


Figure 193 — *Interchanging* — XSD

7.6.1.3.5.4.1 RegisterBreakOfJourney – Allowed values

The following table shows the allowed values for ***RegisterBreakOfJourney*** (*RegisterBreakOf-JourneyEnumeration*).

Table 132 – RegisterBreakOfJourney – Allowed values

Value	Description
none	No action needed.

markByStaff	Journey break must be marked by operator staff.
markByValidator	Journey break must be marked by validator.
markByMobileApp	Journey break must be marked using mobile application.
other	Journey break must be marked by other means.

7.6.1.3.5.5 MinimumStay – Model Element

Details of any minimum stay at the destination required to use the product.

Table 133 – *MinimumStay* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*UsageParameter*	::>	MINIMUM STAY inherits from USAGE PARAMETER.
«PK»	***id***	*MinimumStayIdType*	1:1	Identifier of MINIMUM STAY parameter.
	Minimum-StayType	*MinimumStay*	0:1	Nature of Minimum stay requirements. See allowed values below.
«enum»	***RequiredNights-Away***	*DayOfWeekEnum*	0:7	Specific nights which must be spent away. See NeTEx Part1.
	Minimum-NumberOfNights Away	*xsd:integer*	0:1	Minimum number of nights that must be spent away.
	Maximum-NumberOfNights Away	*xsd:integer*	0:1	Minimum number of nights that can be spent away.

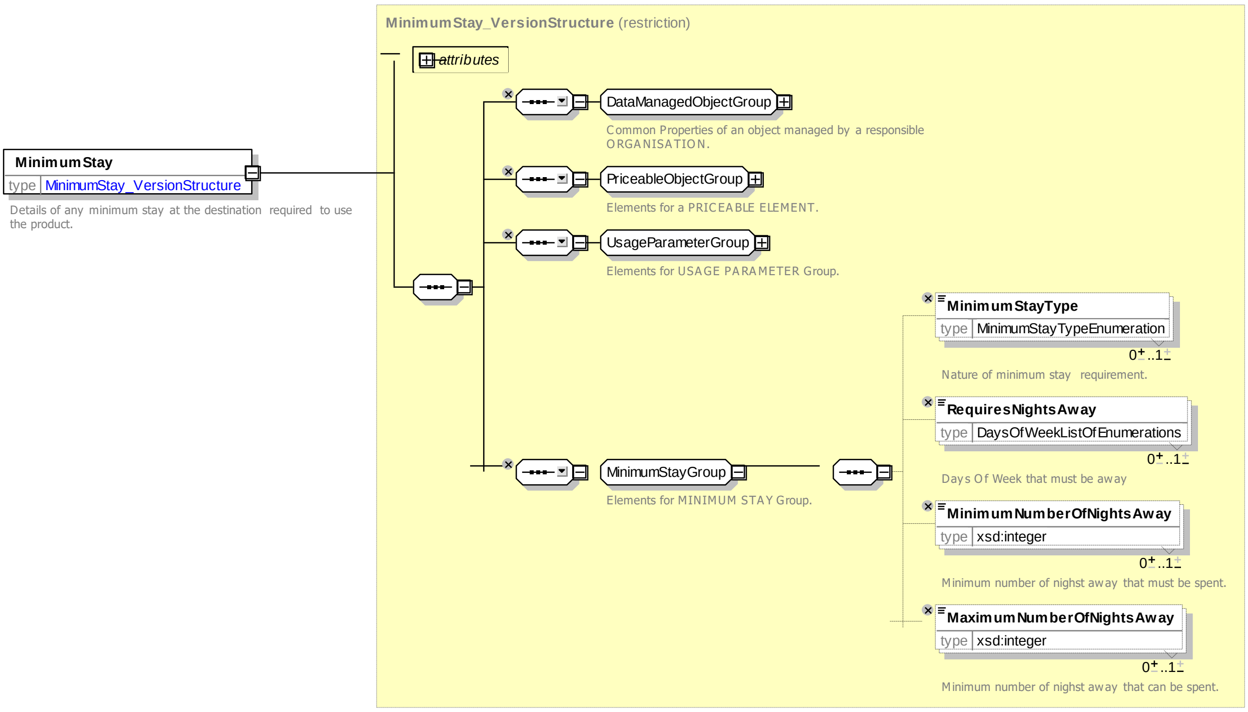


Figure 194 — *MinimumStay* — XSD

7.6.1.3.5.5.1 MinimumStay – Allowed values

The following table shows the allowed values for ***MinimumStay*** (*MinimumStayEnumeration*).

Table 134 – MinimumStay – Allowed values

Value	Description
none	Starts on purchase.
specifiedNightsAway	Must spend specified nights away.
countNightsAway	Must spend the specified number of nights away.
either	Must spend either the specified number of nights away or the specified nights
both	Must spend the specified number of nights away which must be from the specified nights.

7.6.1.3.5.6 StepLimit – Model Element

Geographical parameter limiting the access rights by counts of stops, sections or zones.

Table 135 – *StepLimit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	STEP LIMIT inherits from USAGE PARAMETER.
«PK»	***id***	*StepLimitIdType*	1:1	Identifier of STEP LIMIT parameter.

	Restricted	*xsd:boolean*	0:1	Whether restricted to a number of stops.
«enum»	***StepUnits***	*StepUnitEnum*	0:`	Units in which steps are counted. See allowed values below.
	Minimum-NumberOfSteps	*xsd:integer*	0:1	Minimum number of steps allowed.
	Maximum-NumberOfSteps	*xsd:integer*	0:1	Maximum number of steps allowed.
	Maximum-NumberOfTrips	*xsd:integer*	0:1	Maximum number of trips allowed.

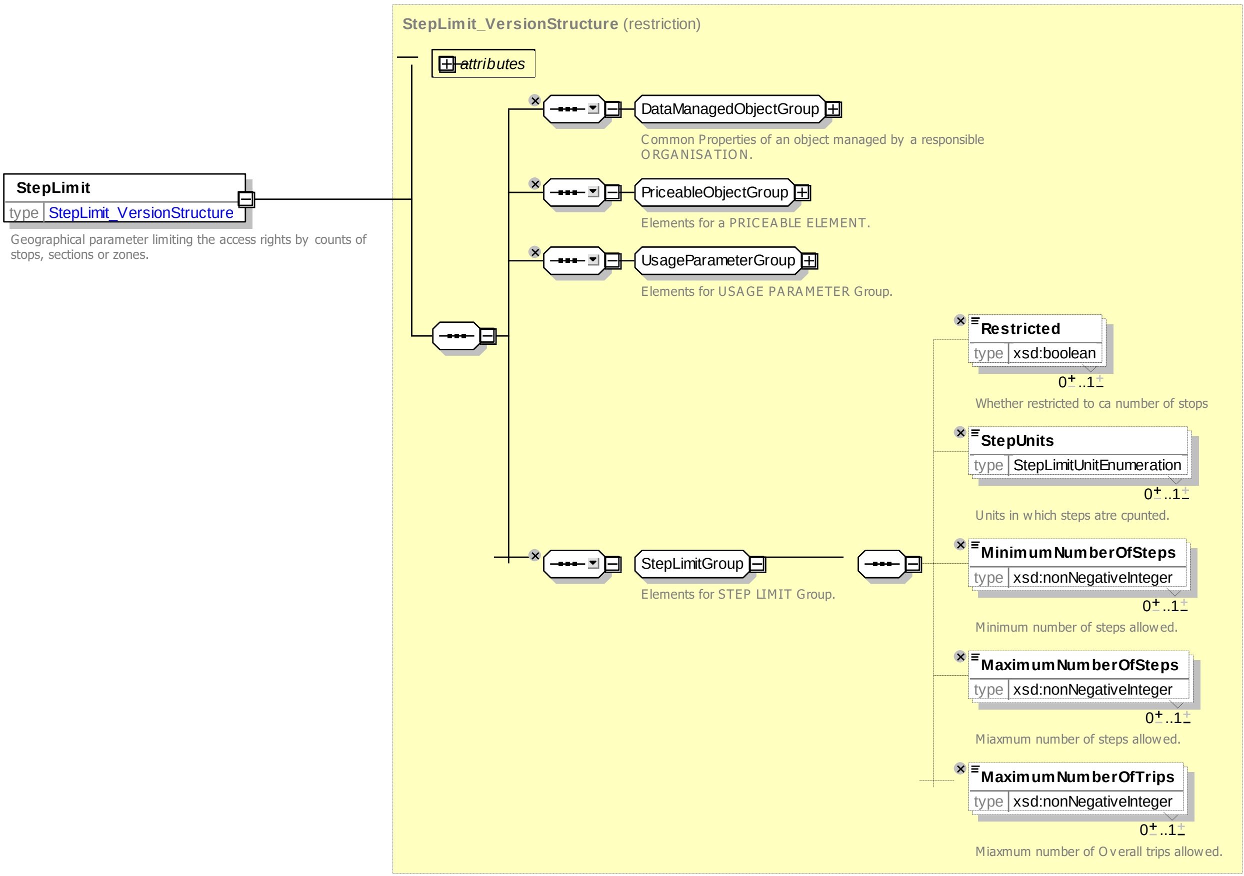


Figure 195 — *StepLimit* — XSD

7.6.1.3.5.6.1 StepLimit – Allowed values

The following table shows the allowed values for ***StepLimit*** (*StepLimitEnumeration*).

Table 136 – StepLimit – Allowed values

Value	Description
stops	Step limit applies to number of stops at which user enters or changes.
stopsIncludingPassThroughStops	Step limit applies to number of stops including stops passed though.

sections	Step limit applies to number of sections passed though.
zones	Step ep limit applies to number of zones passed though.
networks	Step limit applies to number of networks passed though. +v1.1
operators	Step limit applies to number of operators used. +v1.1
countries	Step limit applies to number of countries passed though. +v1.1

7.6.1.3.5.7 UsageValidityPeriod – Model Element

A time limitation for validity of a FARE PRODUCT or a SALES OFFER PACKAGE. It may be composed of a standard duration (e.g. 3 days, 1 month) and/or fixed start/end dates and times.

Table 137 – *UsageValidityPeriod* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	USAGE VALIDITY PERIOD inherits from USAGE PARAMETER.
«PK»	***id***	*UsageValidityPeriodIdType*	1:1	Identifier of USAGE VALIDITY PERIOD.
«enum»	***ValidityType***	*ValidityTypeEnum*	0:*	Type of USAGE VALIDITY PERIOD. See allowed values below.
«enum»	***UsageTrigger***	*UsageTriggerEnum*	0:1	Trigger event that starts validity period. See allowed values below.
«enum»	***UsageEnd***	*UsageEndEnum*	0:1	Classification of when the end of the Usage validity period occurs. May be a specified period (Standard Duration) or an event, e.g. end of trip. See allowed values below.
	Standard-Duration	*xsd:duration*	0:1	Duration of VALIDITY PERIOD after departure. or validation
«enum»	***ActivationMeans***	*ActivationMeansEnum*	0:1	Means of activatiing start of period. See allowed values below. +v1.1
	StartDate	*xsd:date*	0:1	Start date for VALIDITY PERIOD.
	StartTime	*xsd:time*	0:1	Start time for VALIDITY PERIOD.
	EndDate	*xsd:date*	0:1	End date for VALIDITY PERIOD.
	EndTime	*xsd:time*	0:1	End time for VALIDITY PERIOD.
XGRP	***UsageValidity-PeriodStart-ConstraintGroup***	***xmlGroup***	0:1	Elements controlling the allowed start of period. See below.
«enum»	***BlackoutUse***	*BlackoutStartEnum*	0:1	When start of travel restriction applies. See allowed values below.

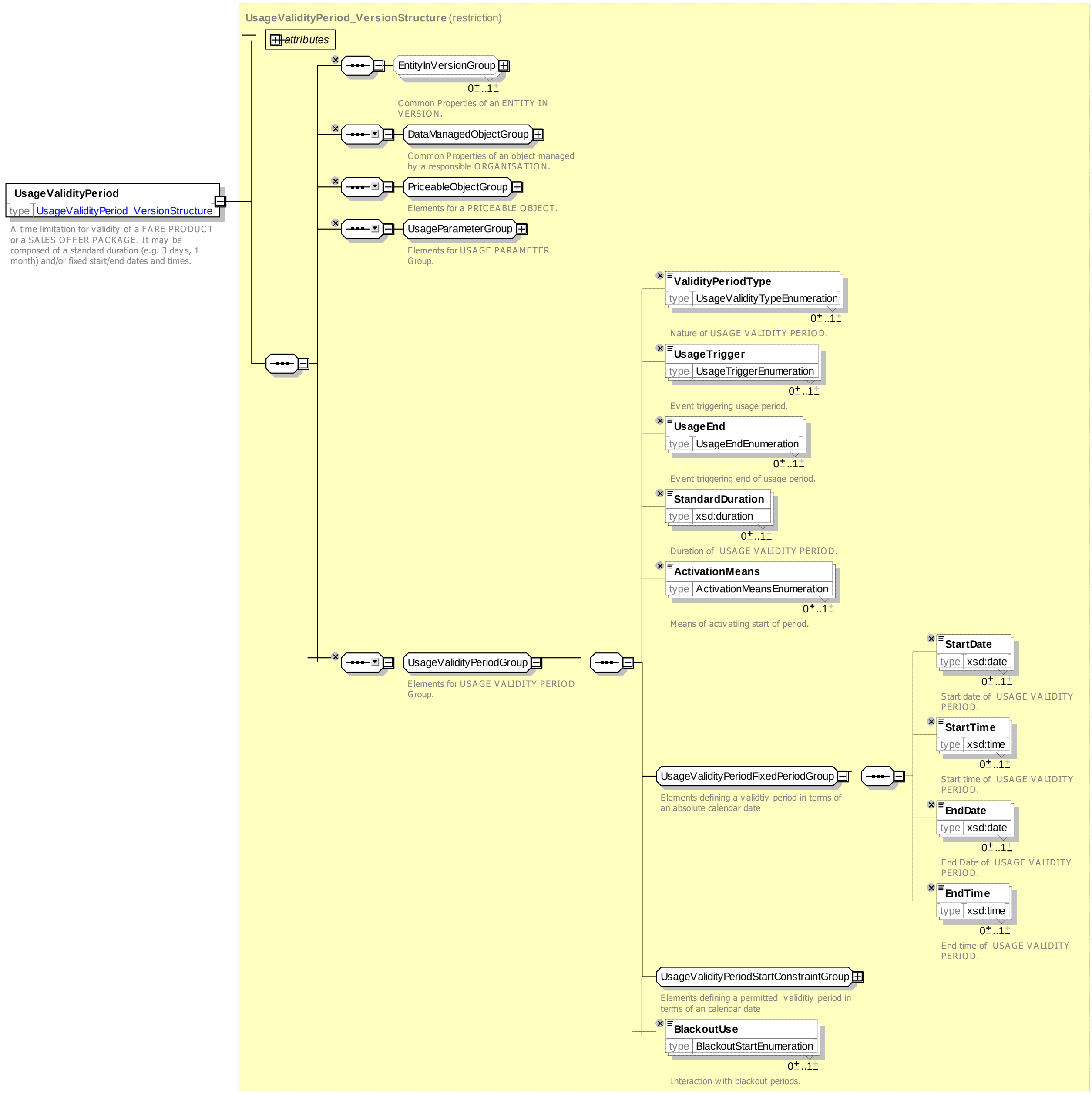


Figure 196 — *UsageValidityPeriod* — XSD

7.6.1.3.5.7.1 UsageValidityType – Allowed values

The following table shows the allowed values for ***UsageValidityType*** (*UsageValidityTypeEnumeration).*

Table 138 – *UsageValidityType* – Allowed values

Value	Description
singleRide	A single ride.
singleTrip	A single trip.
returnTrip	A return trip.
carnet	A number of individual trips.
dayPass	Ticket valid for a day.
weeklyPass	Valid for one week.
weekendPass	Valid for one weekend.

monthlyPass	Valid for one month.
annualPass	Valid for one year.
seasonTicket	Ticket valid for specified period of several days, weeks or months.
profileMembership	Ticket valid while member of a COMMERCIAL PROFILE or USER PROFILE.
subscription	Product valid for specified period of subscription. +V1.1
openEnded	Ticket valid until otherwise notified.
other	Other Validity period

7.6.1.3.5.7.2 UsageTrigger – Allowed values

The following table shows the allowed values for ***UsageTrigger*** (*UsageTriggerEnumeration)*.

Table 139 – *UsageTrigger* – Allowed values

Value	Description
startOfPeriod	Start of period. Beginning time for period in which first use occurs (for Capping products)
startOutboundRide	Start of outbound trip.
endOutboundRide	End of outbound trip.
startReturnRide	Start of return trip
enrolment	Validity period starts when user registers (e.g. creates a customer account).
reservation	Validity period starts when user makes a reservation.
purchase	Starts on purchase.
activation	Validity period starts when user activates a product.
specifiedStartDate	Start date specified at purchase - may be different from purchase date.
fulfilment	Starts on collection.
dayOffsetBeforeCalendarPeriod	Becomes valid a given number days before start of calendar period where number of days is specified by a MONTH VALIDITY OFFSET.

7.6.1.3.5.7.3 UsageEnd – Allowed values

The following table shows the allowed values for ***UsageEnd*** (*UsageEndEnumeration*).

Table 140 – *UsageEnd* – Allowed values

Value	Description
standardDuration	Period Ticket valid for specified after validation. May be in terms of trip or a specified period.
endOfCalendarPeriod	Ticket valid to end of calendar period.

endOfRide	Ticket valid to end of ride.
endOfTrip	Ticket valid to end of trip - may be several rides.
endOfFareDay	Ticket valid to end of fare day.
endOfFarePeriod	Ticket valid to end of fare period.
productExpiry	Product valid to end of product - for a travel card withe expiry date.
deregistration	Product valid until deregistration.
profileExpiry	Ticket valid while member of a profile. Stops when ends.
other	Other Validity period.

7.6.1.3.5.7.4 ActivationMeans – Allowed values

The following table shows the allowed values for ***ActivationMeans*** (*ActivationMeansEnumeration*).

Table 141 – *ActivationMeans* – Allowed values

Value	Description
noneRequired	No activation required.
checkIn	Activation occurs automatically on check-in at barrier, etc.
useOfValidator	Activation occurs on use of validator.
useOfMobileDevice	Activation is made by using mobile device.
automaticByTime	Activation occurs automatically at specified time of travel.
automaticByProximity	Activation occurs automatically by proximity to stop and/or vehicle.
other	Other means of activation.

7.6.1.3.5.7.5 BlackoutStart – Allowed values

The following table shows the allowed values for ***BlackoutStart*** condition (*BlackoutStartEnumeration*).

Table 142 – *BlackoutStart* – Allowed values

Value	Description
mayTravelAnytime	May travel at any time.
noTravelWithinPeriod	No travel permitted within exclusion period.
mayCompleteIfStartedBefore	Outward and return journeys may be completed if started before exclusion period.
noTravelWithinTimeband	No travel permitted within exclusion time band of a day.

7.6.1.3.5.8 UsageValidityPeriodStartConstraintGroup – Group

Elements relating to the start of the Validity Period.

Table 143 – *UsageValidityPeriodStartConstraintGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description
«enum»	***UsageStart-ConstraintType***	*UsageStartConstraint-Enum*	0:1	Whether start type of trip or pass is variable or fixed. See allowed values below. +v1.1
«cntd»	***startOnlyOn***	*DayType*	0:*	Prices for the USAGE PARAMETER.
	Maximum-ServicesBefore	*xsd:nonNegativeInteger*	0:1	If ***UsageStartConstraintType*** is "fixedWindow", maximum number of services before the booked train that may also be used. +v1.1
	FlexiblePeriod-Before	*xsd:duration*	0:1	If ***UsageStartConstraintType*** is "fixedWindow", maximum period before the booked train during which other trains may also be caught. +v1.1
	Maximum-ServicesAfter	*xsd:nonNegativeInteger*	0:1	If ***UsageStartConstraintType*** is "fixedWindow", maximum number of services after the booked train that may also be used. +v1.1
	FlexiblePeriodAfter	*xsd:duration*	0:1	If ***UsageStartConstraintType*** is "fixedWindow", maximum period after the booked train during which other trains may also be caught. +v1.1

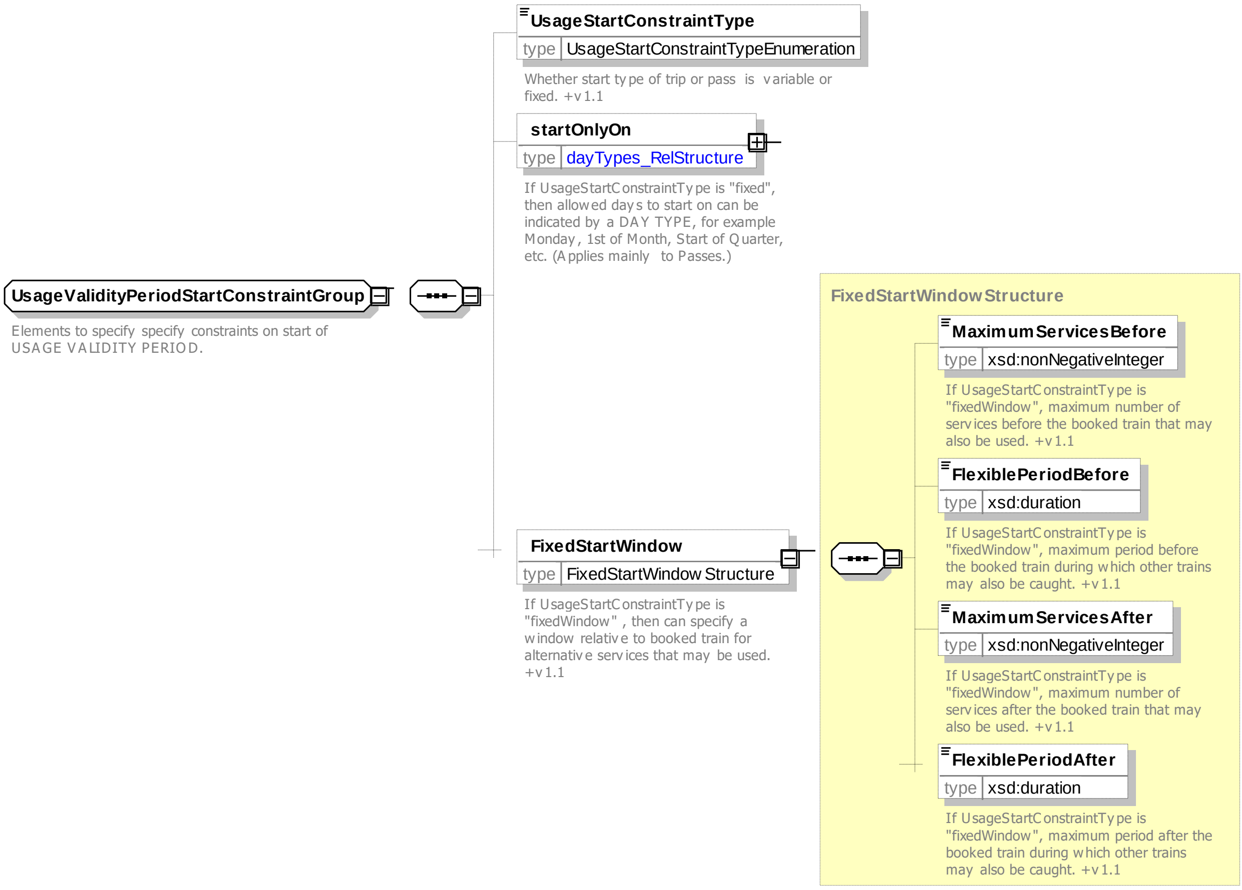


Figure 197 — *UsageValidityPeriodStartConstraintGroup* — XSD

7.6.1.3.5.8.1 UsageStartConstraintType – Allowed values

The following table shows the allowed values for ***UsageStartConstraintType*** condition (*UsageStartConstraintTypeEnumeration*).

Table 144 – *UsageStartConstraintType* – Allowed values

Value	Description
variable	Validity start date can be chosen by user.
fixed	Validity start date is constrained. For a pass to certain days of week, month or year. For a trip to a specific train.
fixedWindow	Validity start date for a trip is constrained relative to start of booked service, e.g. may catch previous train as well.
noTravelWithinTimeband	No travel permitted within exclusion time band of a day.

7.6.1.3.5.9 Suspending – Model Element

Conditions governing temporary suspension of a FARE PRODUCT, (i.e. period pass or subscription).

Table 145 – *Suspending* – Model Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	<u>*UsageParameter*</u>	::>	SUSPENDING inherits from USAGE PARAMETER.
«PK»	***id***	*SuspendingIdType*	1:1	Identifier of USAGE VALIDITY PERIOD.
«enum»	***SuspensionPolicy***	*SuspensionPolicyEnum*	0:*	Allowed policies for suspending term of product.
	QualificationPeriod	*duration*	0:1	Minimum duration that must have occurred before a suspension is allowed.
	QualificationPercent	*decimal*	0:1	Minimum proportion of term that must have occurred before a suspension is allowed.
	MinimumSuspension-Period	*duration*	0:1	Minimum duration allowed for a suspension.
	Maximum-SuspensionPeriod	*duration*	0:1	Maximum duration allowed for a suspension.
	MaximumNumberOf-SuspensionsPerTerm	*nonNegativeInteger*	0:1	Maximum duration allowed for a suspension. with the term of the fare product or subscription.

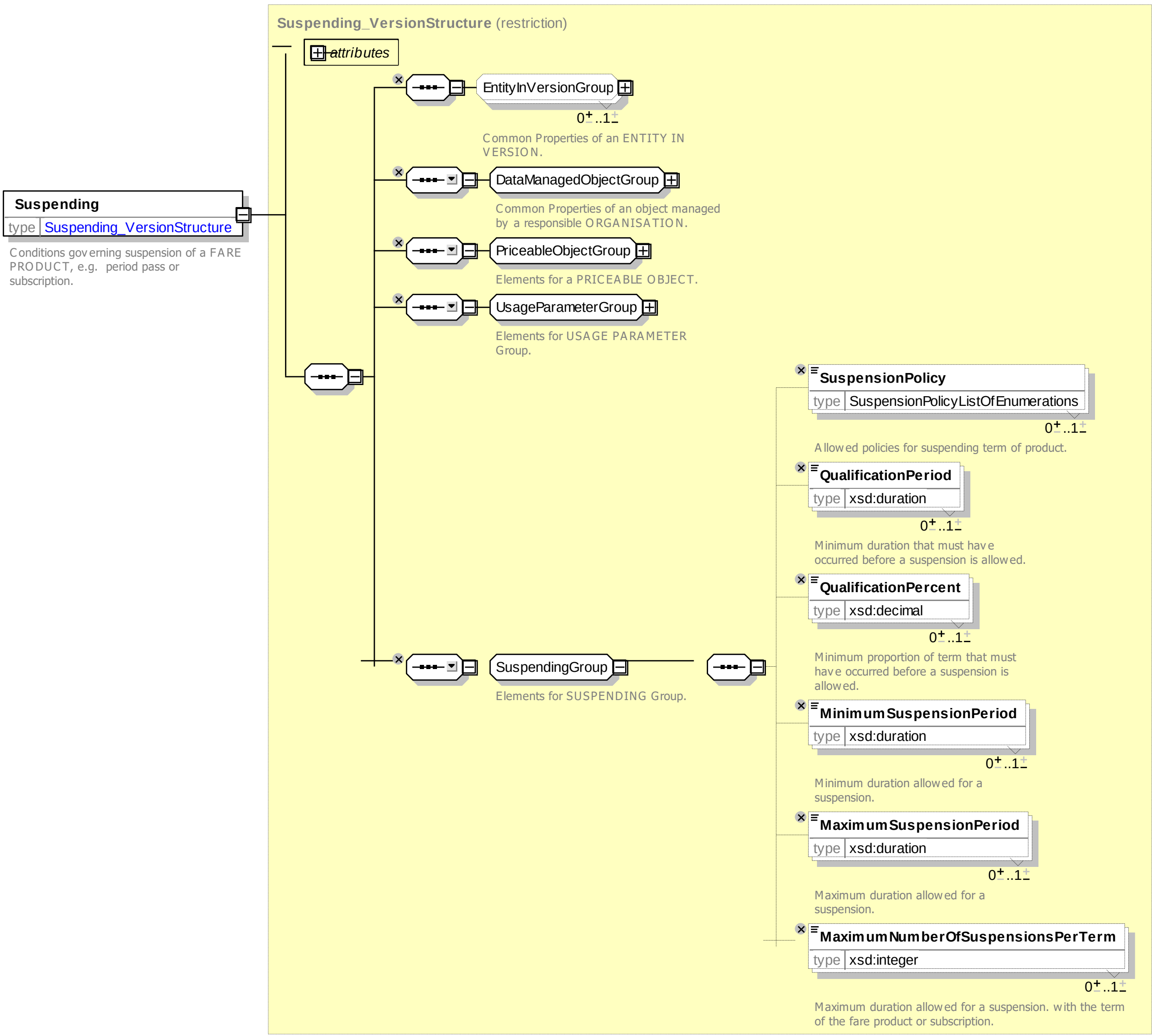


Figure 198 — *Suspending* — XSD

7.6.1.3.5.9.1 SuspensionPolicy – Allowed values

The following table shows the allowed values for ***SuspensionPolicy*** (*SuspensionPolicyEnumeration).*

Table 146 – *SuspensionPolicy* – Allowed values

Value	Description
none	Suspension not allowed.
forCertifiedIllness	Suspension allowed for illness.
forParentalLeave	Suspension allowed for parental leave.
forHoliday	Suspension allowed for holiday.
forAnyReason	Suspension allowed for any reason
weeklyPass	Valid for one week.

weekendPass	Valid for one weekend.
monthlyPass	Valid for one month.
seasonTicket	Ticket valid for specified period of several days, weeks or months.
profileMembership	Ticket valid while member of a COMMERCIAL PROFILE or USER PROFILE.
openEnded	Ticket valid until otherwise notified.
other	Other Validity period

7.6.1.3.6 Usage Parameter: Eligibility – Attributes and XSD

7.6.1.3.6.1 GroupTicket – Model Element

The number and characteristics of persons entitled to travel in addition to the holder of an access right.

Table 147 – *GroupTicket* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	<u>*UsageParameter*</u>	::>	GROUP TICKET inherits from USAGE PARAMETER.
«PK»	***id***	*GroupTicketIdType*	1:1	Identifier of GROUP TICKET.
«FK»	***TypeOf-ConcessionRef***	*TypeOfConcessionRef*		Type of concession to which this group applies.
XGRP	***GroupTicket-SizeGroup***	<u>***xmlGroup***</u>	0:1	Elements relating to size of group.
XGRP	***GroupTicket-CalculationGroup***	<u>***xmlGroup***</u>	0:1	Elements relating to calculation of group discount.
«enum»	***JointCheckIn***	*GroupCheckInEnum*	0:1	Whether joint check in is required. See allowed values below.
«enum»	***Ticketing***	*GroupTicketingEnum*	0:1	Nature of tickets issued for group. See allowed values +v1.1
«enum»	***GroupBooking-Facility***	*GroupBookingEnum*	0:1	Type of Group Booking allowed. See NeTEx Part1.

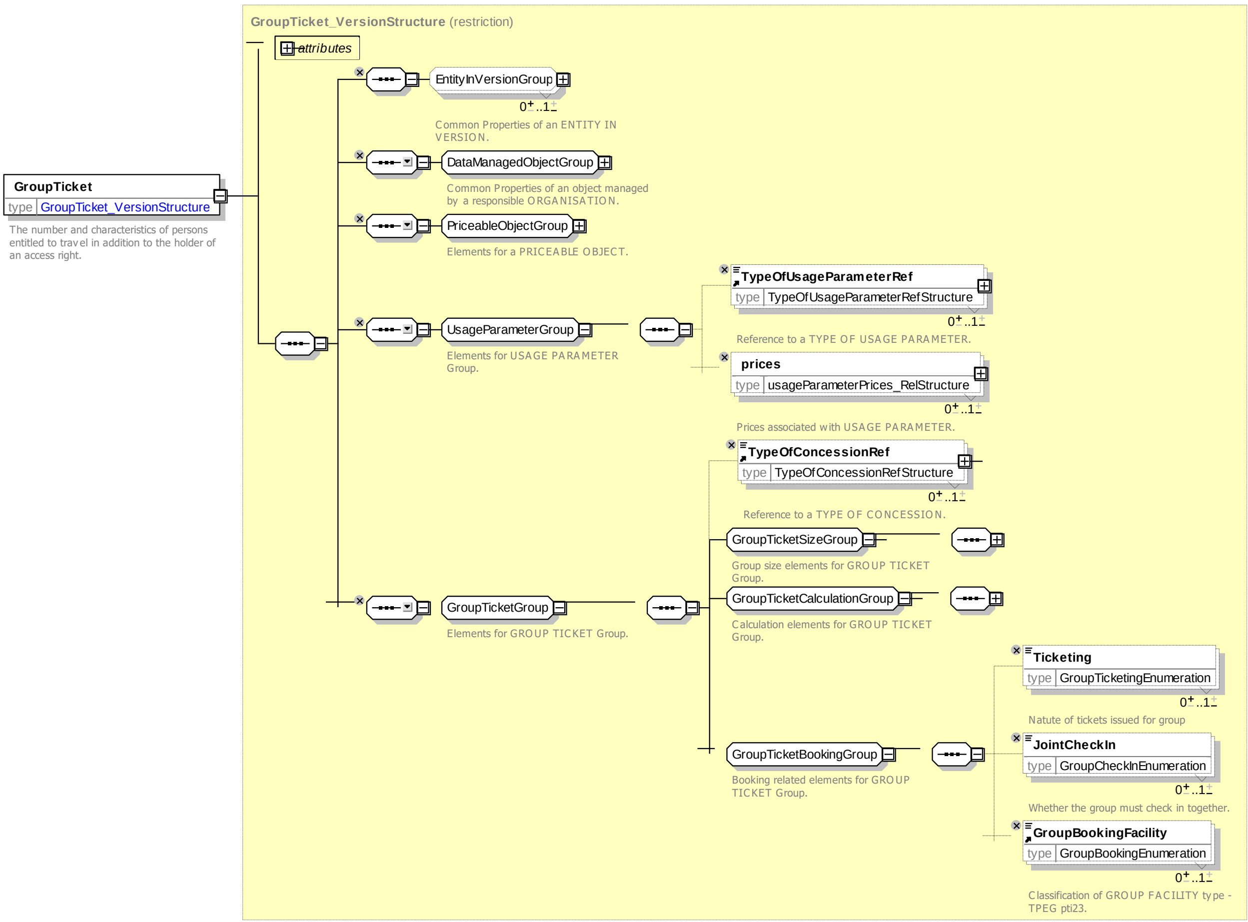


Figure 199 — *GroupTicket* — XSD

7.6.1.3.6.1.1 GroupCheckIn – Allowed values

The following table shows the allowed values for ***GroupCheckIn*** (*GroupCheckInEnumeration*).

Table 148 – *GroupCheckIn* – Allowed values

Value	Description
none	No group check in.
required	Passengers must check in together.
allowed	Passengers may check in together.

7.6.1.3.6.1.2 GroupTicketing – Allowed values

The following table shows the allowed values for the ***Ticketing*** of groups. (*GroupTicketingEnumeration*).

Table 149 – *7.6.1.3.6.3.3 GroupTicketing* – Allowed values

Value	Description
allOnOneTicket	A single ticket is issued for the whole group.
separateTickets	Separate tickets are issued for each member of the group.
ticketWithCoupons	There is a main ticket with coupons for each member.

other	Other ticketing.

7.6.1.3.6.2 GroupTicketSizeGroup – Group

The ***GroupTicketSizeGroup*** specifies the number and characteristics of persons entitled to travel in addition to the holder of an access right.

Table 150 – *GroupTicketSizeGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Minimum-NumberOf-Persons	*NumberOfPersons*	0:1	Minimum number of persons overall allowed on GROUP TICKET.
	Maximum-NumberOf-Persons	*NumberOfPersons*	0:1	Maximum number of persons overall allowed on GROUP TICKET.
	Minimum-NumberOf-CardHolders	*NumberOfPersons*	0:1	Minimum number of card holders required to qualify for this GROUP TICKET.
«cntd»	***companion-Profiles***	*CompanionProfile*	0:*	COMPANION OR GROUP allowed in each USER PROFILE category.

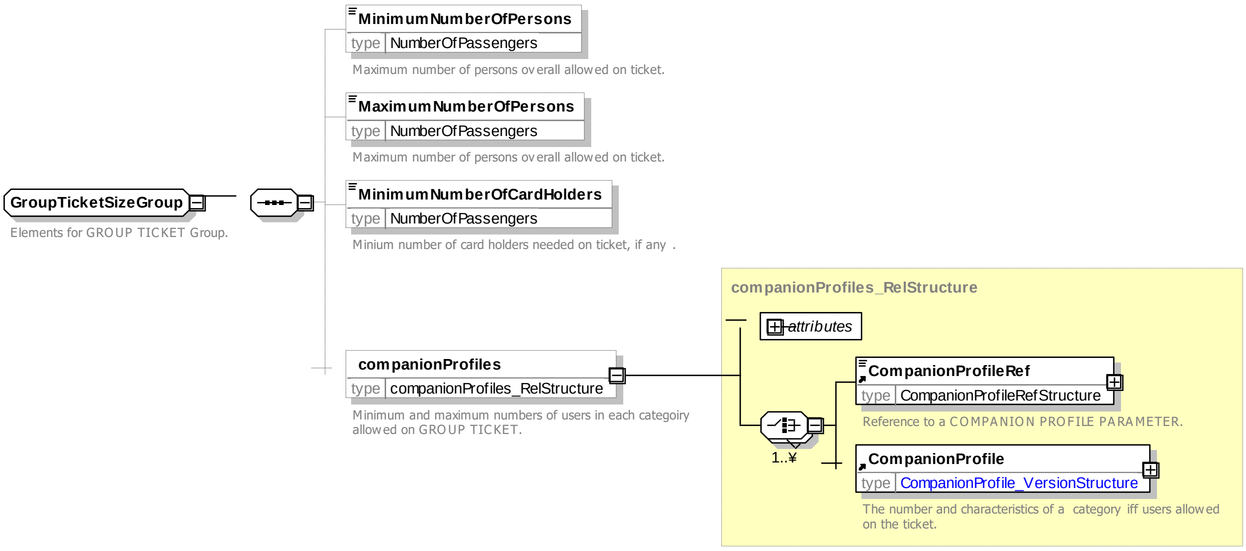


Figure 200 — *GroupTicketSizeGroup* — XSD

7.6.1.3.6.2.1 GroupSizeChanges – Allowed values

The following table shows the allowed values for group size changes ***GroupSizeChanges***. (*GroupSizeChangeEnumeration*).

Table 151 – *7.6.1.3.6.3.3 GroupSizeChanges* – Allowed values

Value	Description
noCharge	Group size cannot be changed
free	No charge to change group size.
charge	Charge for changing group size.
discountForFirstMemberOfGroup	Discount for first member of group only.
purchaseWindowSteppedCharge	Group size can be changed, charges are according to a sliding scale according to the length of time before travel (as specified by several EXCHANGING parameters)
numberOfPassengersSteppedCharge	Group size can be changed, charges are according to a sliding scale according to the number of passengers changed.

7.6.1.3.6.3 *GroupTicketCalculationGroup* – Group

The *GroupTicketCalculationGroup* specifies parameters affecting the calculation of the group discount.

Table 152 – *GroupTicketCalculationGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«enum»	***PricingBasis***	*PerBasisEnum*	0:1	Basis on which pricing is done - per whole group or per member. See allowed values below.
	Maximum-PersonsFree	*NumberOfPassengers*	0:1	Number of persons allowed free on ticket.
	Maximum-Persons-Discounted	*NumberOfPassengers*	0:1	Maximum number of persons for which a group discount is allowed.
	DiscountOnly-ForFirstPerson	*xsd:boolean*	0:1	Whether there is only a discount for the first person in the group.
	MinimumNumber OfCardHolders	*NumberOfPassengers*	0:1	Minimum number of persons in the group who must hold a qualifying railcard for the discount to be granted.
	OneForNPersons	*NumberOfPassengers*	0:1	Whether discount is on a one-for-n basis. Intermediate numbers are rounded down.
	GroupSize-Changes	*GroupSizeChangesEnum*	0:1	Possibilities for changing the number of people in the group. See allowed values below.

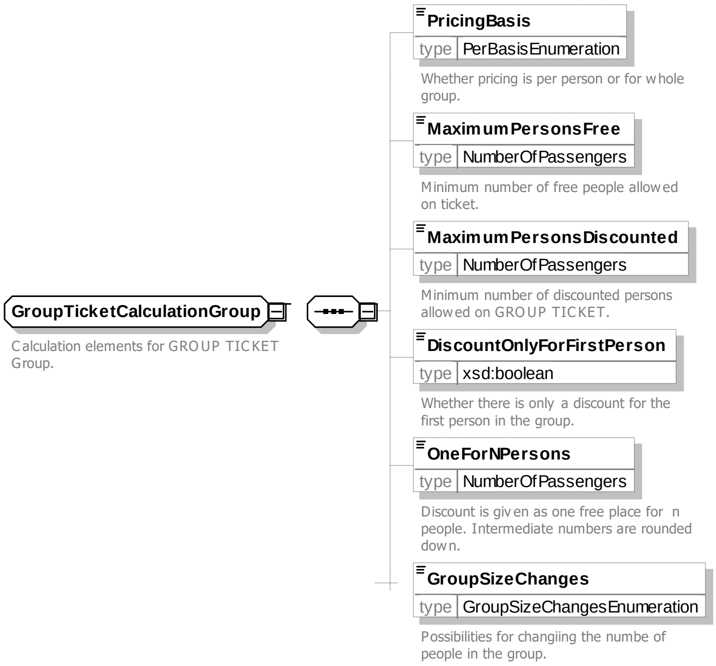


Figure 201 — *GroupTicketCalculationGroup* — XSD

7.6.1.3.6.3.1 GroupDiscountBasis – Allowed values

The following table shows the allowed values for group discount ***GroupDiscountBasis***. (*GroupDiscount-BasisEnumeration*).

Table 153 – *GroupDiscountBasis* – Allowed values

Value	Description
none	No companion allowed.
free	All members free.
discountForAll	Discount for all members of group.
discountForFirstMemberOfGroup	Discount for first member of group only.
discountForSecondAndSubsequentMembersOfGroup	Discount for second and subsequent member of group.
stepDiscount	Discount depends on number of people in group.

7.6.1.3.6.4 UserProfile – Model Element

The social profile of a passenger, based on age group, education, profession, social status, sex etc., often used for allowing discounts: 18-40 years old, graduates, drivers, unemployed, women etc.

Table 154 – *UserProfile* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	USER PROFILE inherits from USAGE PARAMETER.

«PK»	***id***	*UserProfileIdType*	1:1	Identifier of USER PROFILE.
«FK»	***BaseUserProfile-Ref***	*UserProfileIdType*	0:1	Base USER PROFILE which this profile refines.
«FK»	***TypeOf-ConcessionRef***	*TypeOfConcessionRef*	0:1	Classification by type of concession.
«enum»	***UserType***	*UserTypeEnum*	0:1	Classification of user type.
XGRP	***UserProfile-Qualification-Group***	***xmlGroup***	0:1	Elements describing eligibility conditions for user.
«enum»	***GenderLimitation***	*GenderLimitationList*	0:1	Gender required by USER PROFILE. Relevant for single sex accommodation products.
«enum»	***ProofRequired***	*ProofOfIdentityEnum*	0:*	Proof required for type of user. See allowed values below.
«enum»	***DiscountBasis***	*DiscountBasisEnum*	0:1	Nature of discount for this type of user. See earlier for allowed values.
«cntd»	***companion-Profiles***	*CompanionProfile*	0:*	COMPANION PROFILEs describing users who may travel with user.

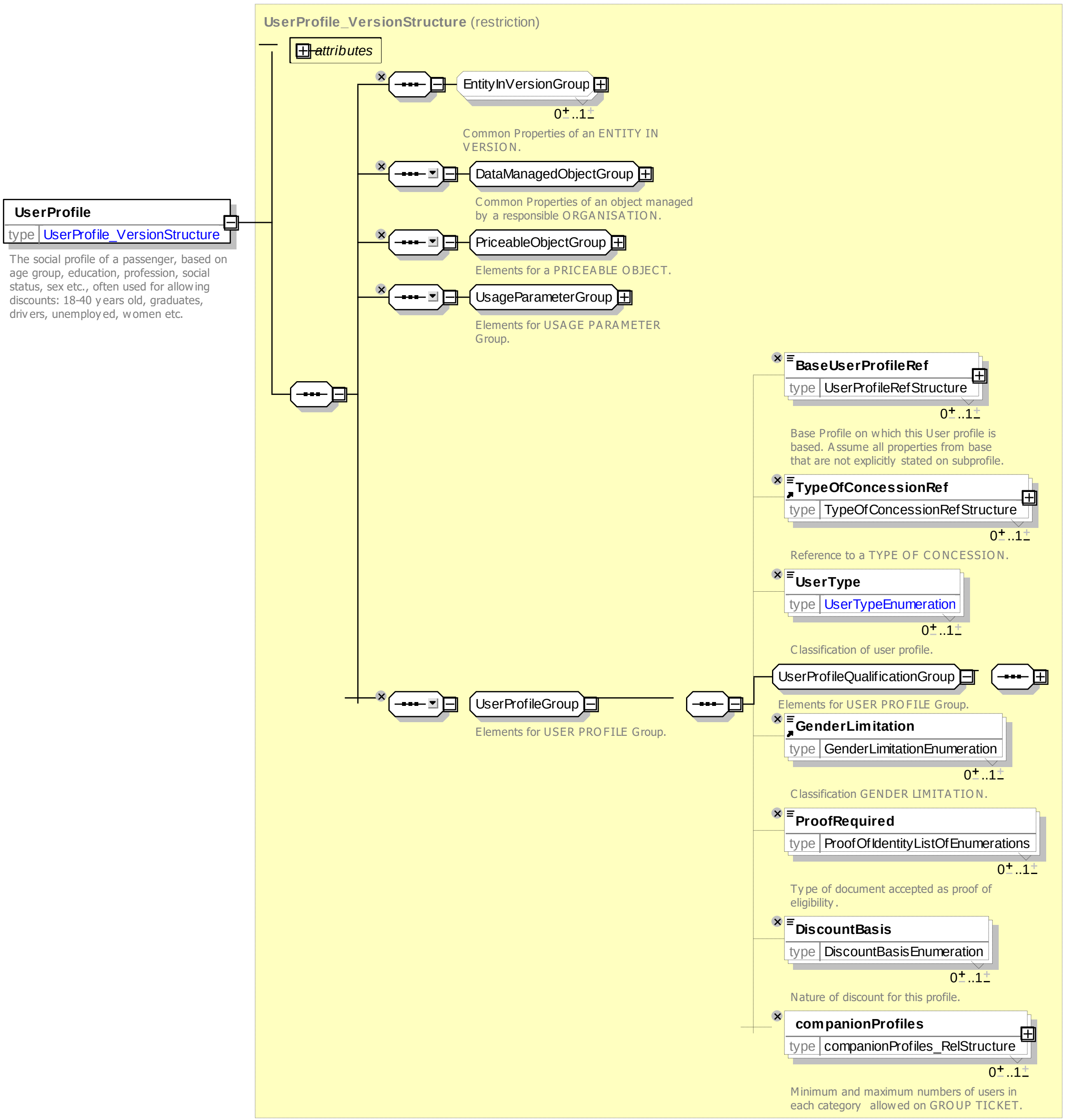


Figure 202 — *UserProfile* — XSD

7.6.1.3.6.4.1 UserType – Allowed values

The following table shows the allowed values for ***UserType*** (*UserTypeEnumeration*).

Table 155 – *UserType* – Allowed values

Value	Description
anyone	*User is any type of person.*
adult	*User is a human.*
child	*User is a child.*
infant	*User is an infant.*
senior	*User is a senior.*
schoolPupil	*User is school pupil. +v1.1*
student	*User is a student.*

youngPerson	*User is a young person. +v1.1*
disabled	*User is a guide dog.*
disabledCompanion	*User is a guide dog. +v1.1*
employee	*User is an employee of a company.*
military	*User is a member of the armed forces. +v1.1*
jobSeeker	*User is unemployed. +v1.1*
guideDog	*User is a guide dog. +v1.1*
animal	*User is an animal.*

7.6.1.3.6.4.2 *UserProfileQualificationGroup* – Group

The ***UserProfileQualificationGroup*** specifies attributes describing the eligibility of a user to belong to a USER PROFILE.

Table 156 – *UserProfileQualificationGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	MinimumAge	*xsd:integer*	0:1	Minimum age for membership of USER PROFILE.
	MaximumAge	*xsd:integer*	0:1	Maximum age for membership of USER PROFILE.
	MonthDayOn-WhichAge-Applies	*xsd:gmonthDay*	0:1	Day / Month on which age applies. if any.
	MinimumHeight	*LengthType*	0:1	Minimum height for membership of USER PROFILE. For example, to restrict access for health and safety reasons.
	MaximumHeight	*LengthType*	0:1	Maximum weight for membership of USER PROFILE. This may be relevant for example for judging large dogs, or a limit on children.
	LocalResident	*xsd:boolean*	0:1	Whether user must be local resident. The default value is '*true'*.
«cntd»	***resides***	*ResidentialQualification*	0:*	RESIDENTIAL QUALIFICATIONs for USER PROFILE – if more than one, these will be logically ORed together.

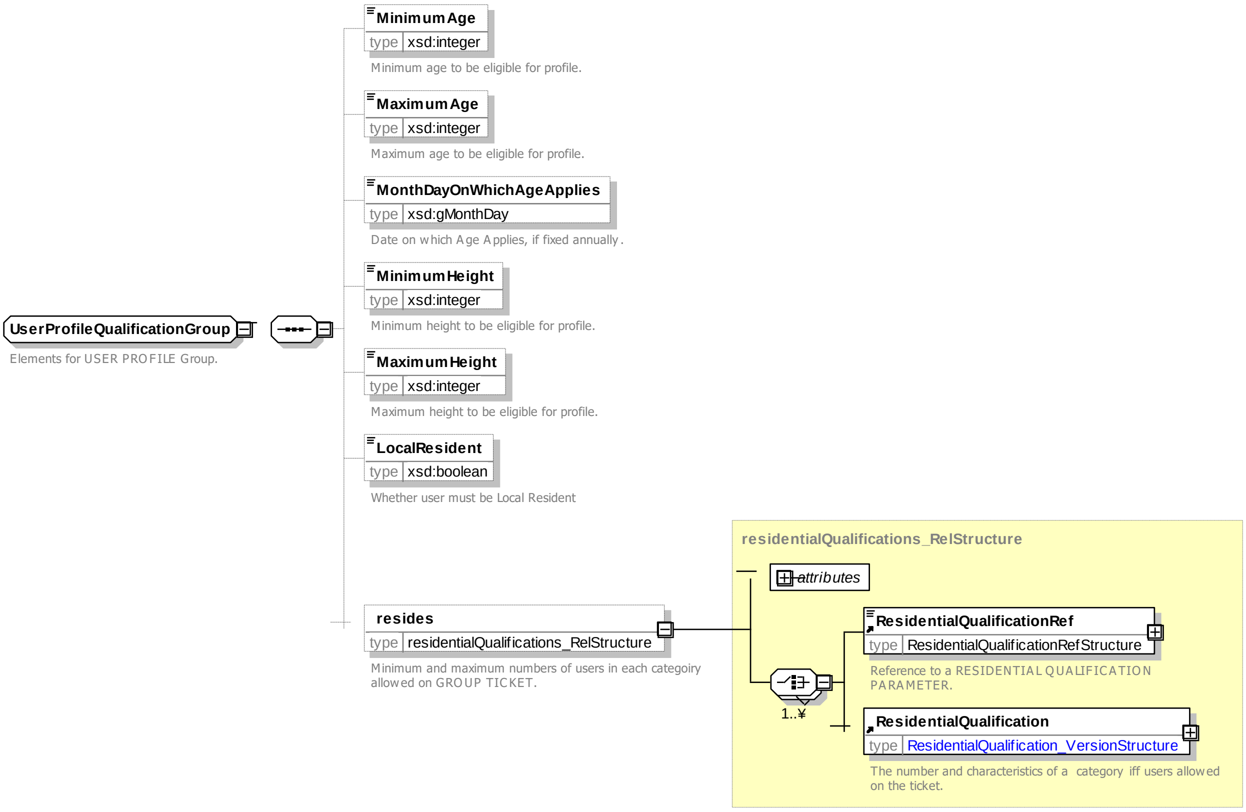


Figure 203 — *UserProfileQualificationGroup*— XSD

7.6.1.3.6.4.3 ProofRequired – Allowed values

The following table shows the allowed values for ***ProofRequired*** (*ProofOfIdentityEnumeration*).

Table 157 – *ProofRequired* – Allowed values

Value	Description
noneRequired	No proof required.
passport	Proof is to show a passport.
drivingLicence	Proof is to show a driving licence.
birthCertificate	Proof is to show a birth certificate. +v1.1
membershipCard	Proof is to show an Identify document. such as a passport or driving licence.
studentCard	Proof is to show a student card.
identityDocument	Proof is to show an Identify document.
creditCard	Proof is to show a credit card.
medicalDocument	Proof is to show a medical document or letter from a medical authority.
letterWIthAddress	Proof is to show a letter or bill from an organisation to the applicant’s address.
measurement	Height or other physical measurement.

emailAccount	Proof is to respond from a valid email account. +v1.1
mobileDevice	Proof is to respond from a mobile device associated with an account. +v1.1
other	Other proof.

7.6.1.3.6.5 ResidentialQualification – Model Element

The RESIDENTIAL QUALIFICATION element describes a requirement to live in a certain area.

Table 158 – *ResidentialQualification* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	RESIDENTIAL QUALIFICATION inherits from VERSIONED CHILD. See NeTEx Part1.
«PK»	***id***	*ResidentialQualification-IdType*	1:1	Identifier of RESIDENTIAL QUALIFICATION.
	Name	*MultilingualString*	0:1	Name of RESIDENTIAL QUALIFICATION.
	Description	*MultilingualString*	0:1	Description of RESIDENTIAL QUALIFICATION.
«FK»	***ParentRef***	*UsageParameterRef+*	0:1	Parent USER PROFILE for whom this specifies a RESIDENTIAL QUALIFICATION.
	MustReside	*xsd:boolean*	0:1	Whether the user must or must not reside in specified TOPOGRAPHIC PLACE.
«FK»	***Topographical-PlaceRef***	*TopographicalPlaceRef*	0:1	TOPOGRAPHIC PLACE for which residency rule applies. See NeTEx Part1.
«enum»	***ResidenceType***	*ResidenceTypeEnum*	0:1	Classification of type of residence required, e.g. live, work, study.
	Minimum-Duration	*xsd:duration*	0:1	Minimum period of residency needed to qualify.

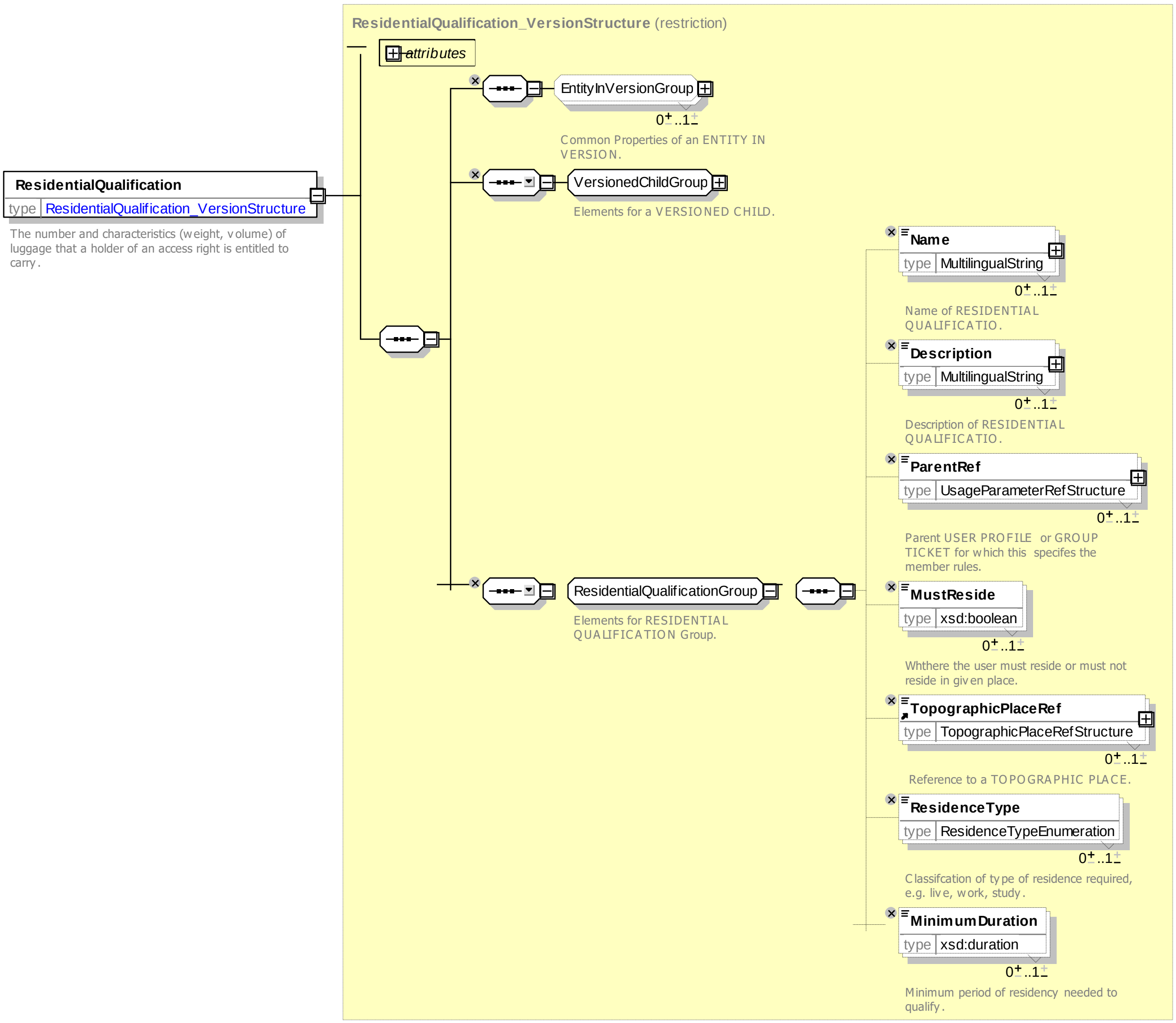


Figure 204 — *ResidentialQualification* — XSD

7.6.1.3.6.5.1 ResidenceType – Allowed values

The following table shows the allowed values for ***ResidenceType*** (*ResidenceTypeEnumeration*).

Table 159 – *ResidenceType* – Allowed values

Value	Description
live	Must live in specified area.
work	Must work in specified area.
study	Must be in full time education in specified area.
exchange	Must be on qualifying exchange programme in specified area.
born	Must have been born in the specified area.
nonResident	Must not reside in the specified area.

7.6.1.3.6.6 CompanionProfile – Model Element

The COMPANION PROFILE specifies the number and characteristics of persons entitled to travel in addition to the holder of an access right, for example children, wheelchair carer, etc.

Table 160n – *CompanionProfile* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*UsageParameter*	::>	COMPANION PROFILE inherits from USAGE PARAMETER.
«PK»	***id***	*GroupTicketUserIdType*	1:1	Identifier of COMPANION PROFILE.
	ParentRef	*UsageParameterRef+*	0:1	Parent USER PROFILE for whom this specifies an allowed companion type.
«FK»	***UserProfileRef***	*UserProfileRef*	0:1	Reference USER PROFILE defining a category of people eligible to be a companion.
«enum»	***Companion-Relationship***	*CompanionRelationship-Enum*	0:1	Required relationship of companion to eligible user. See allowed values below. +v1.1.
	MinimumNumberOf-Persons	*xsd:integer*	0:1	Minimum number of persons overall allowed of this type.
	MaximumNumberOf-Persons	*xsd:integer*	0:1	Maximum number of persons overall allowed of this type.
«enum»	***DiscountBasis***	*DiscountBasisEnum*	0:1	Nature of discount for this type of user. See allowed values earlier.

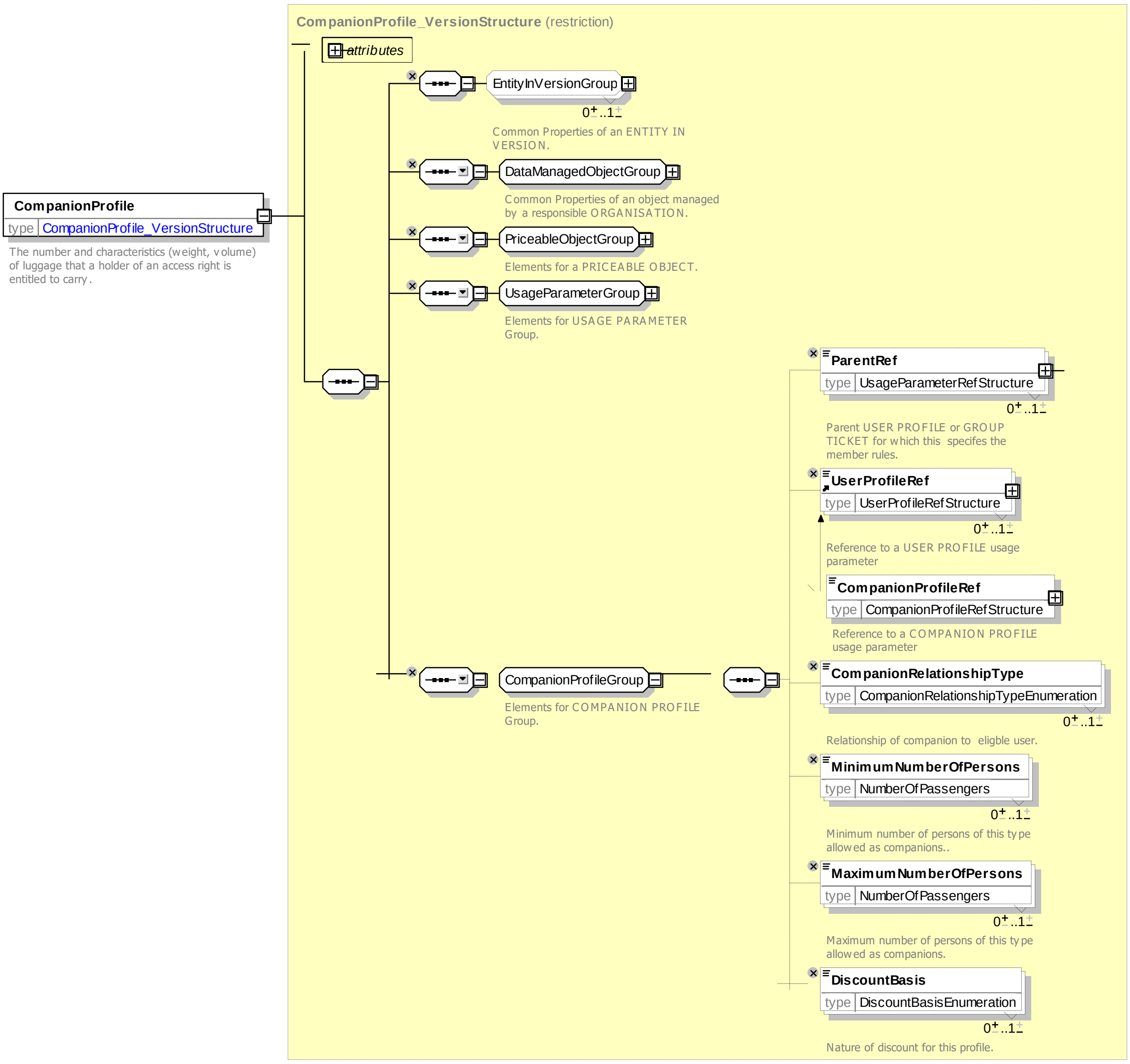


Figure 205 — *CompanionProfile* — XSD

7.6.1.3.6.6.1 CompanionRelationship – Allowed values

The following table shows the allowed values for ***CompanionRelationship*** (*CompanionRelationshipEnumeration*).

Table 161 – *CompanionRelationship* – Allowed values

Value	Description
anyone	Anyone
parent	Parent
grandparent	Grandparent
child	Childe
grandchild	Grandchild
family	Family
spouse	Spouse
partner	Partner
dependent	Dependent
colleague	Colleague
pupil	Pupil
teacher	Teacher
carer	Carer

7.6.1.3.6.7 CommercialProfile – Model Element

A category of users depending on their commercial relations with the operator (frequency of use, amount of purchase etc.), often used for allowing discounts.

Table 162 – *CommercialProfile* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	COMMERCIAL PROFILE inherits from USAGE PARAMETER.
«PK»	***id***	*CommercialProfileIdType*	1:1	Identifier of COMMERCIAL PROFILE.
«FK»	***TypeOf-ConcessionRef***	*TypeOfConcessionRef*	0:1	Reference to a TYPE OF CONCESSION.
	Consumption-Amount	*xsd:anyType*	0:1	Consumption amount associated with COMMERCIAL PROFILE.
	Consumption-Units	*xsd:anyType*	0:1	Units for Consumption amount associated with COMMERCIAL PROFILE.
	GeneralGroupOf EntitiesRef	*GeneralGroupOf-EntitiesRef*	0:1	GROUP OF ORGANISATIONs or other entities associated with the COMMERCIAL PROFILE.

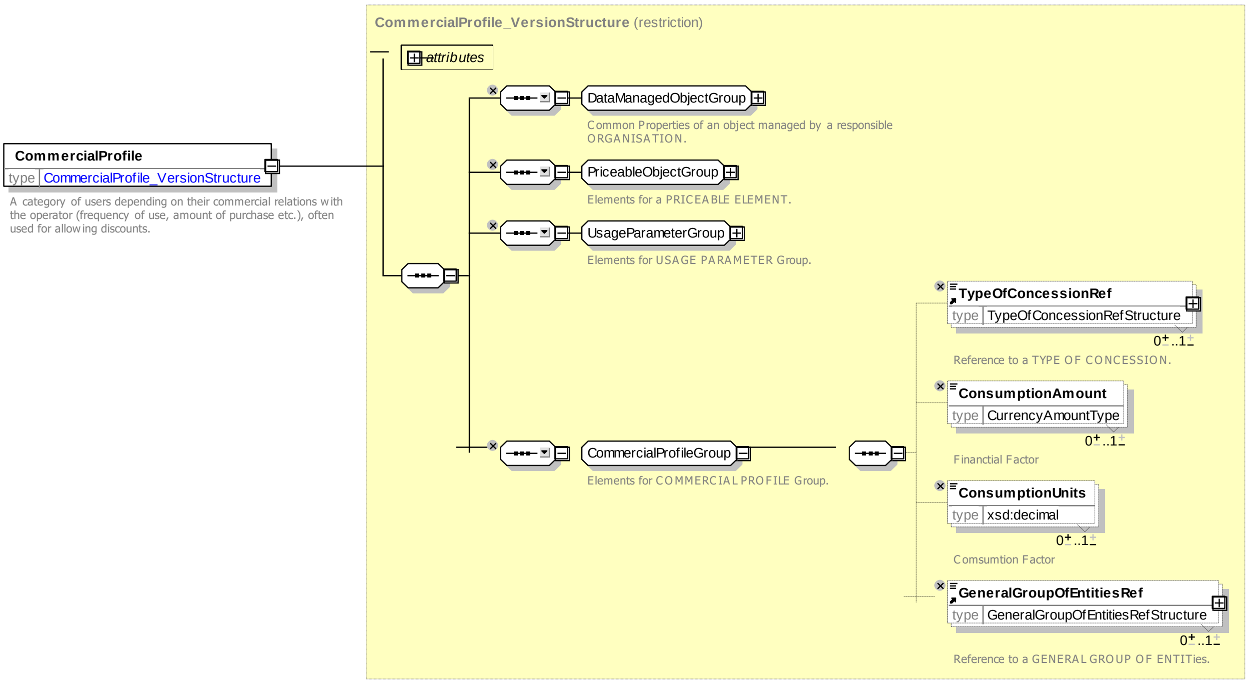


Figure 206 — *CommercialProfile* — XSD

7.6.1.3.6.8 TypeOfConcession – Model Element

A classification of USER PROFILE by type of person eligible to use it

Table 163 – *TypeOfConcession* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF CONCESSION inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*TypeOfConcession-IdType*	1:1	Identifier of TYPE OF CONCESSION.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for VALUE.

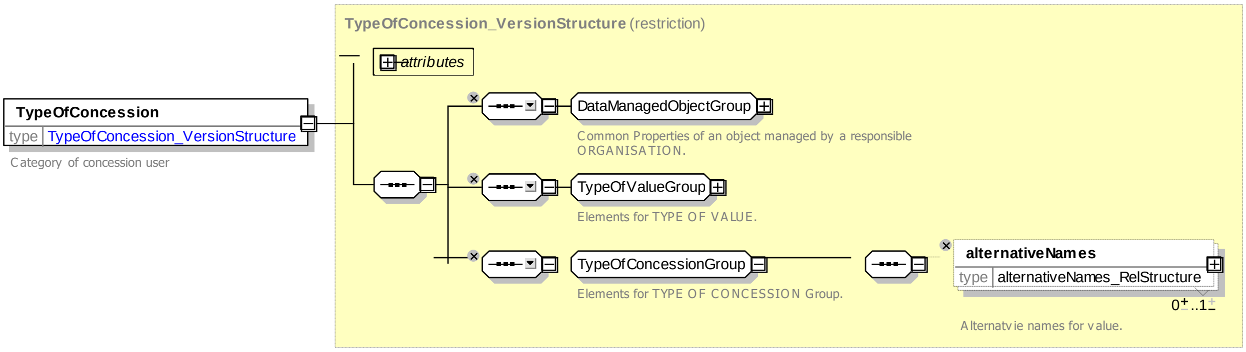


Figure 207 — *TypeOfConcession* — XSD

7.6.1.3.6.9 EligibilityChangePolicy – Model Element

A classification of USER PROFILE by type of person eligible to use it

Table 164 – *EligibilityChangePolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*UsageParameter*	::>	ELIGIBILITY CHANGE POLICY inherits from USAGE PARAMETER.
«PK»	***id***	*EligibilityChangePolicy-IdType*	1:1	Identifier of ELIGIBILITY CHANGE POLICY.
«enum»	***OnBecoming-EligiblePolicy***	*OnBecomingEnumeration*	0:1	Policy to apply on product holder becoming eligible. See allowed values.
«enum»	***OnCeasingToBe-EligiblePolicy***	*OnCeasingEnum*	0:1	Policy to apply on product holder ceasing to be eligible. See allowed values below.

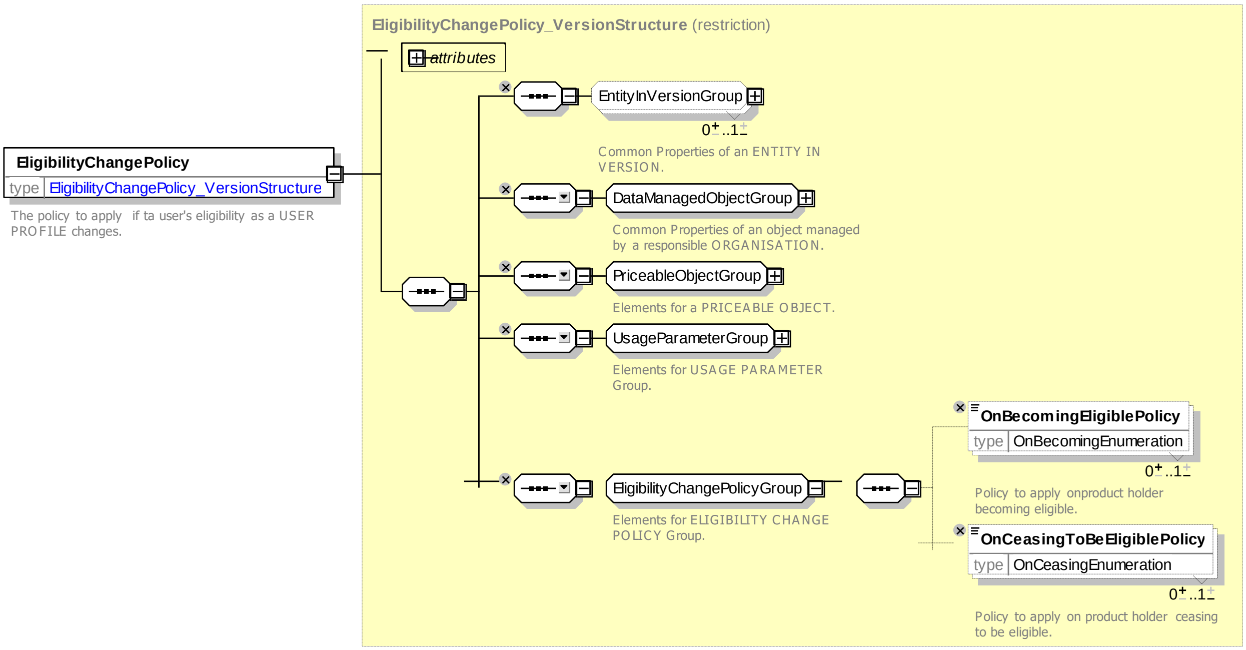


Figure 208 — *7.6.1.3.6.8 EligibilityChangePolicy* — XSD

7.6.1.3.6.9.1 OnBecoming – Allowed values

The following table shows the allowed values for ***OnBecomingEligible*** (*OnBecomingEnumeration).*

Table 165 – *OnBecoming*– Allowed values

Value	Description
automatic	Automatically enrol or upgrade the user. To reflect status change.
invite	Invite the user to change products.
noAction	Take no action.
other	Other action.

7.6.1.3.6.9.2 OnCeasing – Allowed values

The following table shows the allowed values for ***OnCeasingEligible*** (*OnCeasingEnumeration).*

Table 166 – *OnCeasing*– Allowed values

Value	Description
onCeasingEligibility	Allowed values for Ceasing to be eligible.
immediateTermination	If user ceases to be eligible, automatically terminate validity of an eligibility dependent product.
useUntilExpiry	If user ceases to be eligible, they may go on using the product until it expires.
terminateAfterGracePeriod	If user ceases to be eligible, termination take place after the end of a grace period

automaticallySubstituteProduct	If user ceases to be eligible, automatically substitute them with an appropriate replacement product.
noAction	If user ceases to be eligible, take no action.
other	Other action.

7.6.1.3.7 Usage Parameter: Entitlement – Attributes and XSD

7.6.1.3.7.1 EntitlementRequired – Model Element

Receiving of entitlement from another FARE PRODUCT.

Table 167 – *EntitlementRequired* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	ENTITLEMENT REQUIRED inherits from USAGE PARAMETER.
«PK»	***id***	*EntitlementRequired-IdType*	1:1	Identifier of ENTITLEMENT REQUIRED.
«FK»	***ServiceAccess-RightRef***	*ServiceAccessRightRef*	0:1	Entitlement comes from the referenced FARE PRODUCT.
	Minimum-Qualification-Period	*xsd:duration*	0:1	Minimum period that required product must be held in order to be eligible.
	Entitlement-Constraint	*EntitlementConstraint*	0:1	Constraints on related product or offer. +v1.1

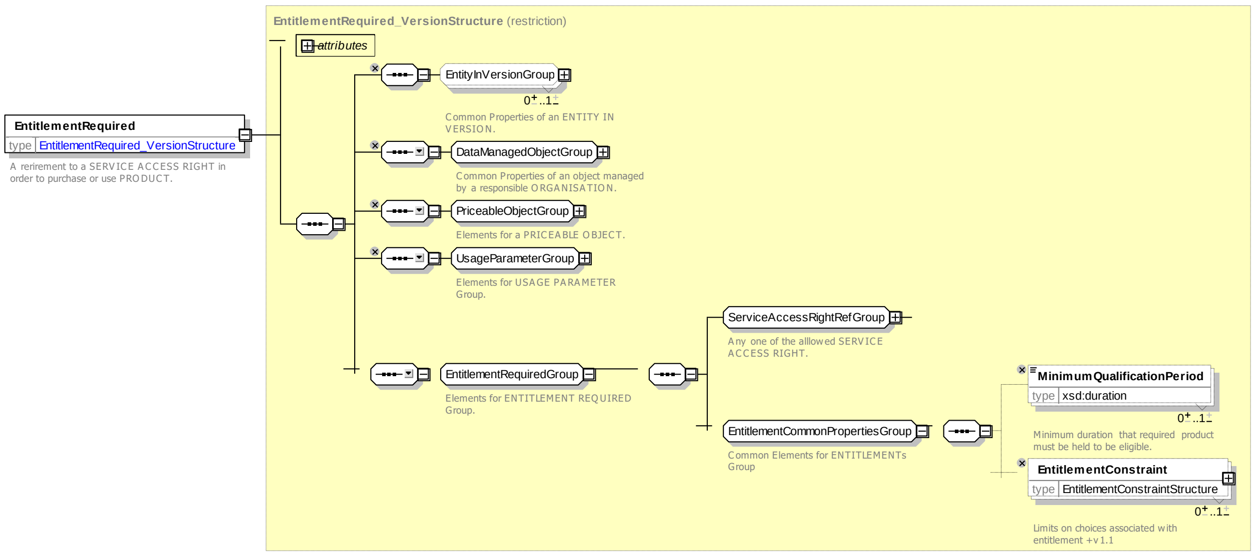


Figure 209 — *EntitlementRequired* — XSD

7.6.1.3.7.2 EntitlementGiven – Model Element

Granting of entitlement to another FARE PRODUCT.

Table 168 – *EntitlementGiven* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*UsageParameter*</u>	::>	ENTITLEMENT GIVEN inherits from USAGE PARAMETER.
«PK»	***id***	*EntitlementGivenIdType*	1:1	Identifier of ENTITLEMENT GIVEN.
«FK»	***ServiceAccess-RightRef***	*ServiceAccessRightRef*	0:1	Entitlement comes from the referenced FARE PRODUCT.
	Minimum-Qualification-Period	*xsd:duration*	0:1	Minimum period that product must be held for entitlement to be granted.
	Entitlement-Constraint	<u>*EntitlementConstraint*</u>	0:1	Constraints on related product or offer. +v1.1
«enum»	***EntitlementType***	*EntitlementTypeEnum*	0:1	Type of entitlement. See allowed values below.

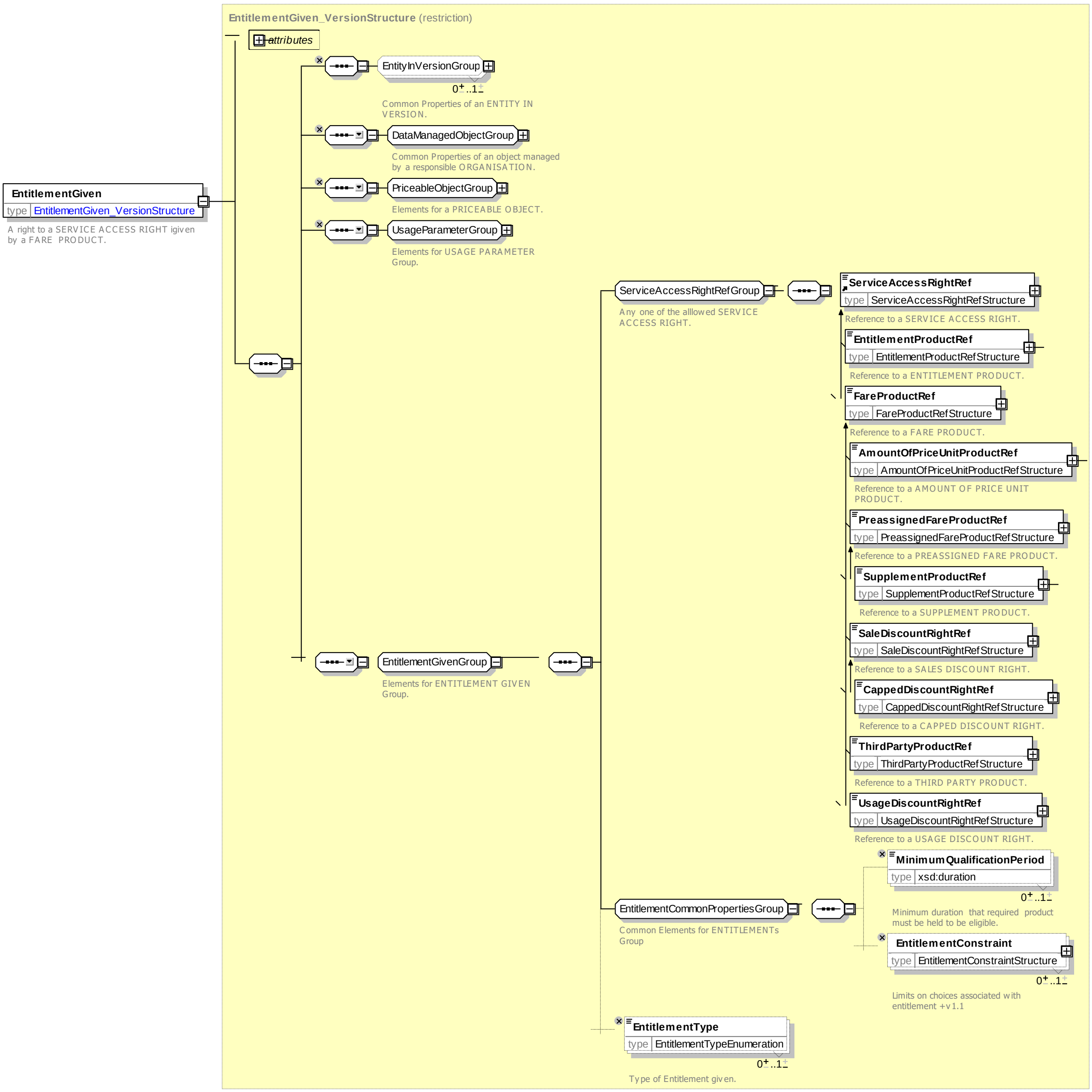


Figure 210 — *EntitlementGiven* — XSD

7.6.1.3.7.2.1 EntitlementType – Allowed values

The following table shows the allowed values for ***EntitlementType*** (*EntitlementTypeEnumeration*)

Table 169 – *EntitlementType* – Allowed values

Value	Description
use	Entitlement is to use product.
purchase	Entitlement is to purchase product.
none	No entitlement.

7.6.1.3.7.3 EntitlementConstraint – Model Element

Constraints on choices for an dependent entitled product relative to the required choices for the prerequisite entitling product.

Table 170 – *7.6.1.3.7.3* *EntitlementConstraint* – Element

Classifi-cation	Name	Type	Cardin ality	Description
«enum»	***PeriodConstraint***	*SamePeriodEnum*	0:1	Constraint on validity period of associated product, e.g. Same time, same day, same period. See allowed values below.
«enum»	***OriginConstraint***	*SameStopEnum*	0:1	Constraint on origin SCHEDULED STOP POINT of associated product. E.g. same stop.
«enum»	***Destination-Constraint***	*SameStopEnum*	0:1	Constraint on destination SCHEDULED STOP POINT of product. See allowed values.
«enum»	***TariffZone-Constraint***	*SameZoneEnum*	0:1	Constraint on TARIFF ZONE of associated product. See allowed values.
«enum»	***Direction-Constraint***	*SameRouteEnum*	0:1	Constraint on DIRECTION of use of associated product. See allowed values.
«enum»	***RouteConstraint***	*SameRouteEnum*	0:1	Constraint on ROUTE of associated product. See allowed values.
«enum»	***Operator-Constraint***	*SameOperatorEnum*	0:1	Constraint on OPERATOR of associated product. E.g. same operator. See allowed values.
«enum»	***TypeOfProduct-Category-Constraint***	*SameTypeOfProduct-CategoryEnum*	0:1	Constraint on TYPE OF PRODUCT CATEGORY of associated product. See allowed values.
«enum»	***ClassOfUse-Constraint***	*SameClassOfUseEnum*	0:1	Constraint SERVICE JOURNEY of associated product. See allowed values.
«enum»	***TypeOfTravel-Document-Constraint***	*SameTypeOfTravelDocument Enum*	0:1	Constraint on TYPE OF TRAVEL DOCUMENT of associated product. See allowed values.
«enum»	***Journey-Constraint***	*SameJourneyEnum*	0:1	Constraint on SERVICE JOURNEY of associated product. See allowed values.
«enum»	***UserConstraint***	*SameUserEnum*	0:1	Constraint on CUSTOMER and USER PROFILEs of associated product. See allowed values.
«cntd»	***specificProfiles***	*UserProfileRef*	0:*	Specific user profiles to which USER PROFILEs to which entitlement applies.

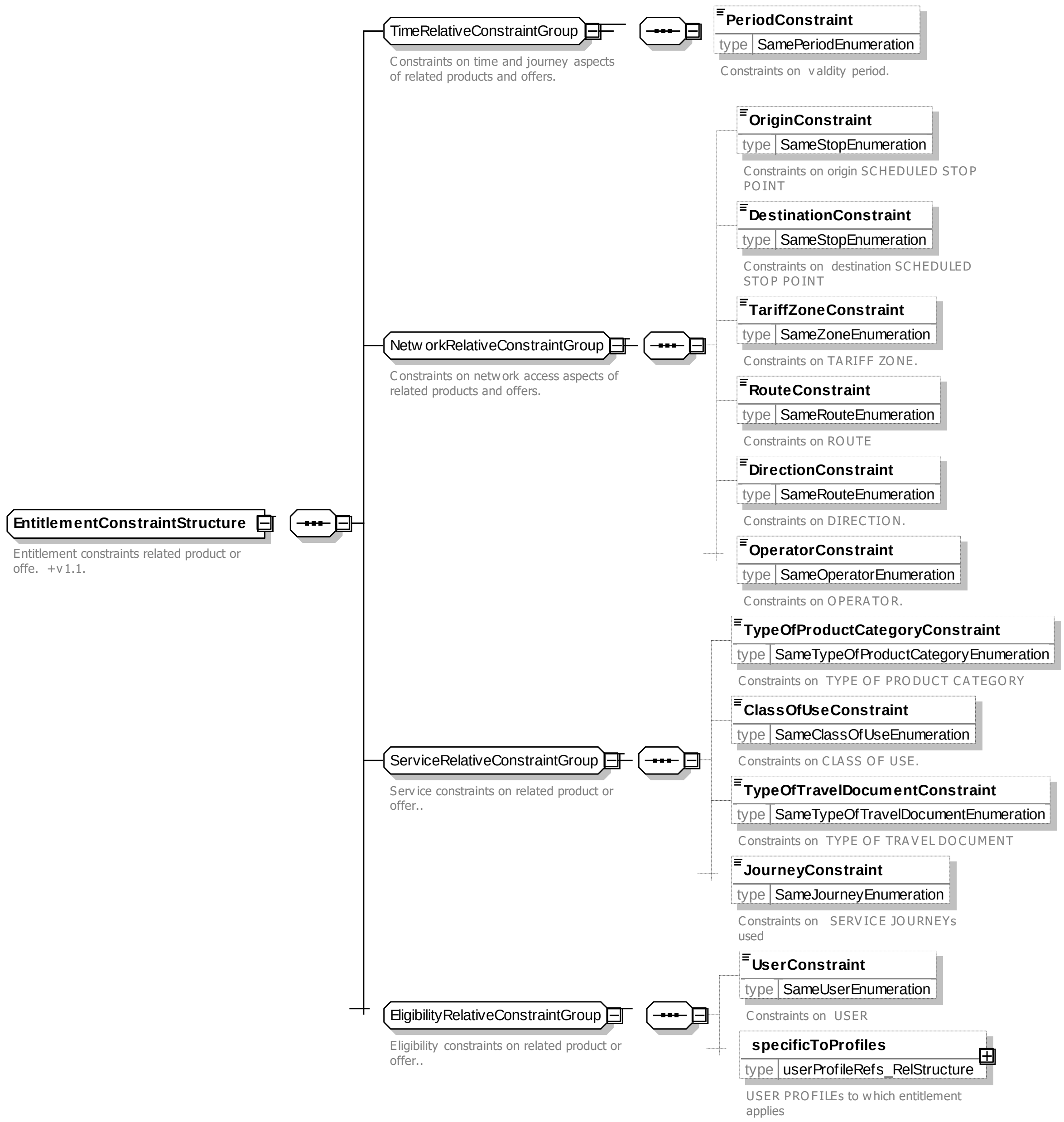


Figure 211 — *EntitlementConstraint* — XSD

7.6.1.3.7.3.1 PeriodConstraint – Allowed values

The following table shows the allowed values for ***PeriodConstraint*** (*SamePeriodEnumeration*).

Table 171 – *SamePeriod* – Allowed values

Value	Description
any	No period constraint.
samePeriod	Must be for same period as related product.
withinSamePeriod	Must be within period of related product.
sameDay	Must be for same day as related product.

sameDayOfReturn	Must be for same day of return of related product.
sameFareDay	Must be for same FARE DAY as related product.
nextDay	Must be for next FARE DAY as that of related product.
equivalentDuration	Must be equivalent to period of related product.
different	Must be different from period of related product.

7.6.1.3.7.3.2 StopConstraint – Allowed values

The following table shows the allowed values for ***StopConstraint*** (*SameStopEnumeration*).

Table 172 – *SameStop* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.
sameAsOrigin	Must be same as origin of related product.
sameAsDestination	Must be same as destination of related product.
sameAsOriginOrDestination	Must be for same origin or destination of related product.
anyStopOnRoute	Must be within one of the stops on the route of the related product.
anyStopInZone	Must be within zone of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.3 ZoneConstraint – Allowed values

The following table shows the allowed values for ***ZoneConstraint*** (*SameZoneEnumeration*).

Table 173 – *SameZone* – Allowed values

Value	Description
any	No constraint.
same	Must be same as related product.
sameAsOrigin	Must be same as origin of related product.
sameAsDestination	Must be same as destination of related product.
sameAsOriginOrDestination	Must be for same origin or destination of related product.

within	Must be within zone of related product.
containing	Must contain zone of related product.
equivalent	Must be equivalent to access right ZONE of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.4 RouteConstraint – Allowed values

The following table shows the allowed values for ***RouteConstraint*** (*SameRouteEnumeration*).

Table 174 – *SameRoute* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.
oppositeDirection	Must be opposite to that of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.5 OperatorConstraint – Allowed values

The following table shows the allowed values for ***OperatorConstraint*** (*SameOperatorEnumeration)*.

Table 175 – *SameOperator* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.
participating	Must be participant in interoperating agreement with operator of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.6 TypeOfProductCategoryConstraint – Allowed values

The following table shows the allowed values for ***TypeOfProductCategoryConstraint*** (*SameTypeOfProductCategoryEnumeration*).

Table 176 – *SameTypeOfProductCategory* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.

sameOrEquivalent	Must be equivalent to that of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.7 ClassOfUseConstraint – Allowed values

The following table shows the allowed values for ***ClassOfUseConstraint*** (*SameClassOfUseEnumeration*).

Table 177 – *SameClassOfUse* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.
sameOrEquivalent	Must be equivalent to that of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.8 TypeOfTravelDocumentConstraint – Allowed values

The following table shows the allowed values for ***TypeOfTravelDocumentConstraint*** (*SameTypeOfTravelDocumentEnumeration)*.

Table 178 – *SameTypeOfTravelDocument* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.
sameMedia	Must use same media as that of related product.
sameSmartCard	Must be on same smart card ad that of related product.
sameMobileApp	Must use same mobile app as that of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.9 JourneyConstraint – Allowed values

The following table shows the allowed values for ***JourneyConstraint*** (*SameJourneyEnumeration*).

Table 179 – *SameJourney* – Allowed values

Value	Description
any	No constraint.
same	Must be same as that of related product.
similar	Must be similar to that of related product.
different	Must be different from that of related product.

7.6.1.3.7.3.10 UserConstraint – Allowed values

The following table shows the allowed values for ***UserConstraint*** (*SameUserEnumeration*).

Table 180 – *SameUser* – Allowed values

Value	Description
anyone	No constraint on eligibility.
samePerson	Must be same person as that of associated product.
differentPerson	Must be diferent person as that of associated product.
specific	Must be belomg to a given USER PROFILE.

7.6.1.3.8 Usage Parameter: Luggage – Attributes and XSD

7.6.1.3.8.1 LuggageAllowance – Model Element

The number and characteristics (weight, volume) of luggage that a holder of an access right is entitled to carry.

Table 181 – *LuggageAllowance* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*UsageParameter*</u>	::>	LUGGAGE ALLOWANCE inherits from USAGE PARAMETER.
«PK»	***id***	*LuggageAllowance-IdType*	1:1	Identifier of LUGGAGE ALLOWANCE.
«enum»	***BaggageUseType***	*BaggageUseEnum*	0:1	Use of baggage covered by the allowance. See allowed values below.
«enum»	***BaggageType***	*LuggageUseEnum*	0:1	Type of baggage covered by the allowance. See allowed values below.
«enum»	***Luggage-AllowanceType***	*LuggageAllowanceEnum*	0:1	Classification of allowance type. See allowed values below.
	Maximum-NumberOfItems	*xsd:nonNegativeInteger*	0:1	Number of bags allowed.
	MaximumBagHeight	*LengthType*	0:1	Maximum bag height.
	MaximumBagWidth	*LengthType*	0:1	Maximum bag width.
	MaximumBagDepth	*LengthType*	0:1	Maximum bag depth.
	MaximumBagWeight	*WeightType*	0:1	Maximum bag weight.
	TotalWeight	*WeightType*	0:1	Total Weight limit of LUGGAGE ALLOWANCE.

«enum»	***Luggage-ChargingBasis***	*LuggageCharging-BasisEnum*	0:1	Basis on which luggage is charged. See allowed values below.

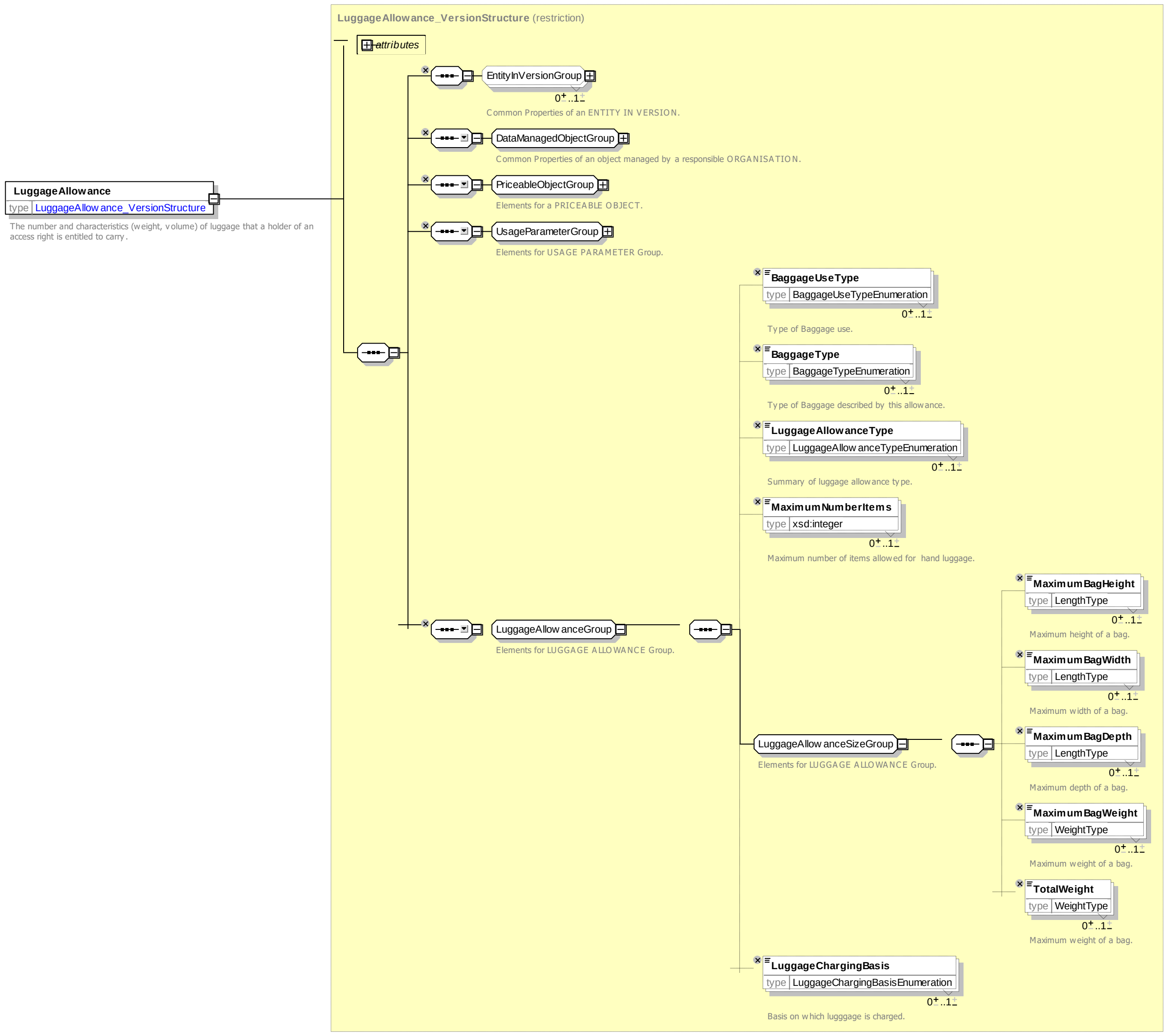


Figure 212 — *LuggageAllowance* — XSD

7.6.1.3.8.1.1 BaggageType – Allowed values

The following table shows the allowed values for ***BaggageType*** (*BaggageTypeEnumeration).*

Table 182 – *BaggageType* – Allowed values

Value	Description
handbag	Hand bag.
handLuggage	Hand luggage.
smallSuitcase	Small suitcase.
suitcase	Suitcase.
trunk	Trunk.
oversizeItem	Oversized item.
bicycle	Bicycles.

sportingEquipment	Sporting equipment.
skis	Skis.
musicalInstrument	Musical Instruments.
pushChair	Push chair.
motorizedWheelchair	Motorized Wheelchair.
largeMotorizedWheelchair	Large on street Motorized Wheelchair.
wheelchair	Wheelchair (non-motorized).
smallAnimal	Small animal.
animal	Animal.
game	Dead Game animals.
motorcycle	Motor cycle.
other	Other baggage item.

7.6.1.3.8.1.2 LuggageAllowance – Allowed values

The following table shows the allowed values for ***LuggageAllowance*** (*LuggageAllowanceEnumeration).*

Table 183 – *LuggageAllowance* – Allowed values

Value	Description
none	Luggage is to carry on.
unlimited	Unlimited baggage allowance.
single	Single bag allowed.
limited	Baggage limited by restriction.

7.6.1.3.8.1.3 LuggageUse – Allowed values

The following table shows the allowed values for ***LuggageUse*** (*LuggageUseTypeEnumeration*)

Table 184 – *LuggageUseType* **– Allowed values**

Value	Description
carryOn	Luggage is to carry on.
checkIn	Luggage is to check in.
oversizeCheckIn	Oversize bag check in.

7.6.1.3.8.1.4 LuggageChargingBasis – Allowed values

The following table shows the allowed values for **LuggageChargingBasis** (*LuggageCharging-BasisEnumeration).*

Table 185 – *LuggageChargingBasis* – Allowed values

Value	Description
free	Luggage is free.
chargedByItem	Luggage is charged by item (subject to item size and weight restrictions).
chargedByWeight	Luggage is charged by weight.
other	Luggage is charge don a different basis

7.6.1.3.9 Usage Parameter: Booking – Attributes and XSD

7.6.1.3.9.1 PurchaseWindow – Model Element

Period in which the product must be purchased.

Table 186 – *PurchaseWindow* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*UsageParameter*	::>	PURCHASE WINDOW inherits from USAGE PARAMETER.
«PK»	***id***	*PurchaseWindowIdType*	1:1	Identifier of PURCHASE WINDOW.
«enum»	***PurchaseAction***	*PurchaseActionEnum*	0:1	Action governed by Purchase Window. The default value is '*purchase*'. See allowed values below. +v1.1
«enum»	***PurchaseWhen***	*PurchaseWhenEnum*	0:1	When purchase may be made. See Part1 for allowed values.
	LatestTime	*xsd:duration*	0:1	Latest time on specified last day when ticket can be purchased.
	MinimumPeriod-BeforeDeparture	*xsd:duration*	0:1	Minimum duration before departure that ticket may be purchased.
«FK»	***MinimumPeriod-IntervalRef***	*TimeIntervalRef*	0:1	Minimum period before departure that purchase must be made - as arbitrary interval.
	MaximumPeriod-BeforeDeparture	*xsd:duration*	0:1	Maximum duration before departure that ticket may be purchased.
«FK»	***MaximumPeriod-IntervalRef***	*TimeIntervalRef*	0:1	Maximum period before departure that purchase must be made - as arbitrary interval.
«enum»	***PurchaseMoment***	*PurchaseMomentEnum*	0:1	Permitted moments of purchase. See Part1 for allowed values. +v1.1

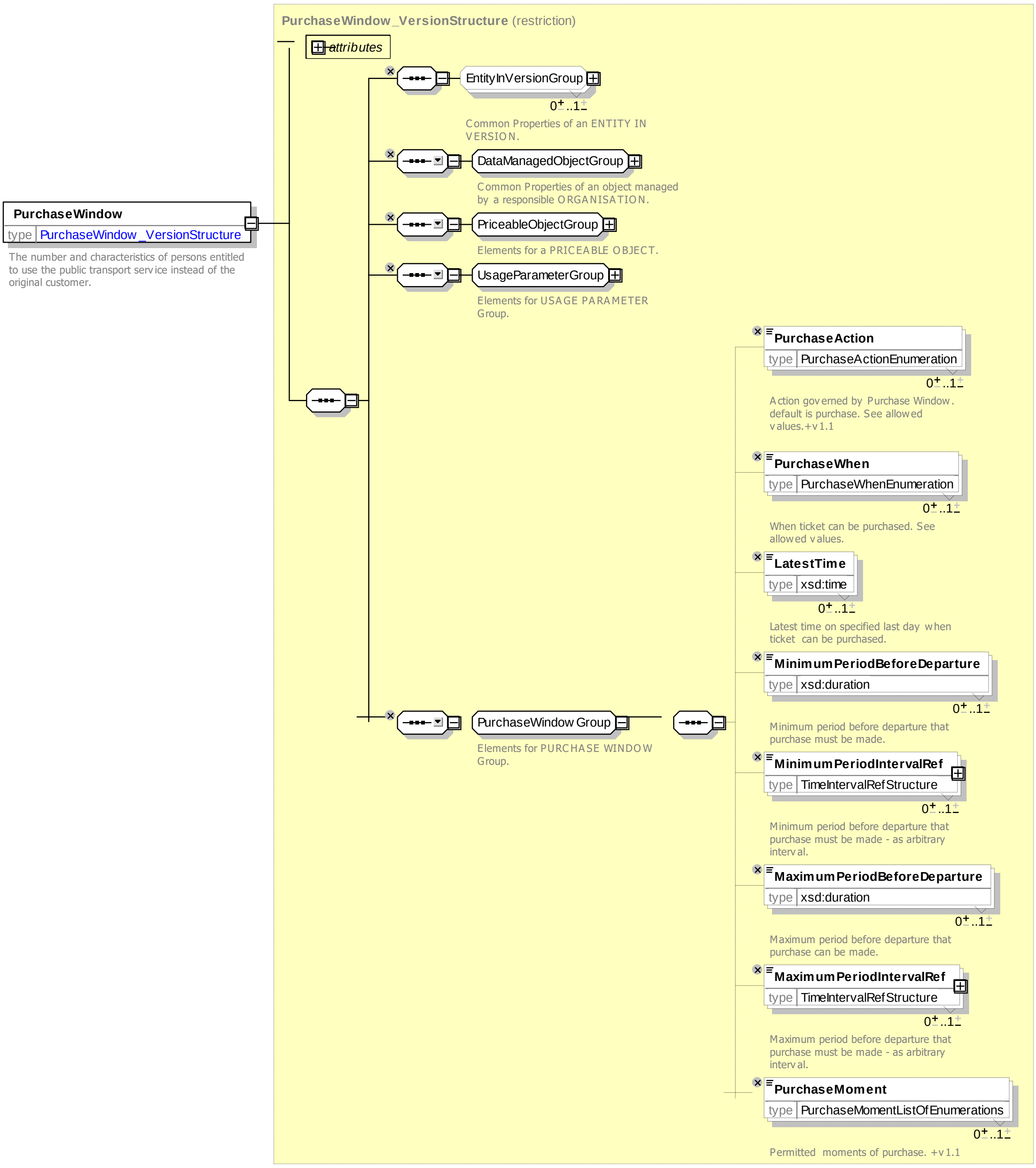


Figure 213 — *PurchaseWindow* — XSD

7.6.1.3.9.1.1 PurchaseAction – Allowed values

The following table shows the allowed values for ***PurchaseAction*** (*PurchaseActionEnumeration*).

Table 187 – *PurchaseAction* – Allowed values

Value	Description
purchase	Purchase
orderWithoutPayment	Order without payment
reserve	Reserve

payForPreviousOrder	Pay for pervious order	*payInstallment*	Pay subscription installment.
subscribe	Set up subscription.	*other*	Other

7.6.1.3.9.1.2 PurchaseWhen – Allowed values

The following table shows the allowed values for ***PurchaseWhen*** (*PurchaseWhenEnumeration*).

Table 188 – *PurchaseWhen* – Allowed values

Value	Description
advanceOnly	Purchase may only be made in advance.
untilPreviousDay	Purchase may only be made in advance up until the day previous to travel.
dayOfTravelOnly	Purchase may only be made on day of travel.
advanceAndDayOfTravel	Purchase may be made in advance or on day of travel.
timeOfTravelOnly	Purchase may only be made at time of travel.
subscriptionChargeMoment	Purchase may made at one of the agreed charging moments for a subscription. advance. +v1.1
other	Other limitation on who may make a booking.

7.6.1.3.9.2 Reserving – Model Element

Indicating whether the access right requires reservation and any limitations on making and changing reservations.

Table 189 – *Reserving* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*UsageParameter*	::>	RESERVING inherits from USAGE PARAMETER.
«PK»	***id***	*ReservingIdType*	1:1	Identifier of RESERVING.
«enum»	***Reserving-Requirements***	*ServiceReservation-FacilityEnum*	0:*	Nature of reservations required. See NeTEx Part1 for allowed values.
	Minimum-NumberToReserve	*NumberOfPassengers*	0:1	Minimum number of persons allowed on a reservation.
	MaximumNumber-ToReserve	*NumberOfPassengers*	0:1	Minimum number of persons allowed on a reservation.
	MustReserve-Whole-Compartment.	*xsd:boolean*	0:1	Whether a whole compartment must be reserved.

«enum»	***Reservation-ChargeType***	*Reservation-ChargeTypeEnum*	0:1	Nature of reservation fee. See allowed values below.
«enum»	***FeeBasis***	*PerBasisEnum*	0:1	Basis on which refund is made. See allowed values below.
	HasFree-Connecting-Reservations	*xsd:boolean*	0:1	Whether connecting reservations are all free or not.
	NumberOfFree-Connecting-Reservations	*xsd:integer*	0:1	Number of free connecting reservations allowed.
	IsFeeRefundable	*xsd:boolean*	0:1	Whether reservation fees is refundable. +v1.1
«cntd»	***Booking-Arrangements***	*BookingArrangements*	0:1	Booking arrangements. See Part1 Service Restrictions Model.
«enum»	***SeatAllocation-Method***	*SeatAllocationMethod-Enum*	0:1	Method for allocating seat. See allowed values.
	Reservation-ExpiryPeriod	*xsd:duration*	0:1	Period after which reservation without payment will expire if not paid for. +v1.1

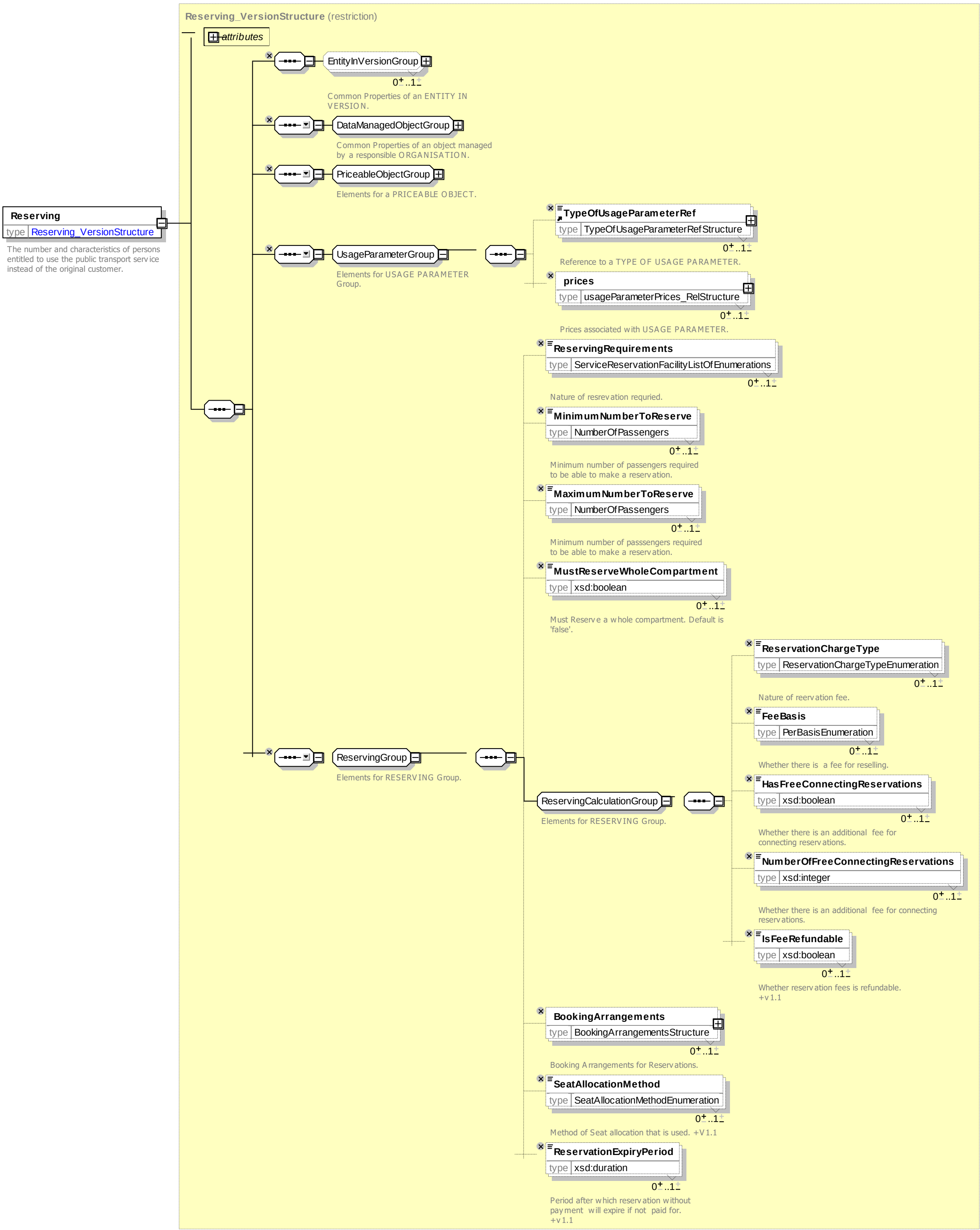


Figure 214 — *Reserving* — XSD

7.6.1.3.9.2.1 ReservationCharge – Allowed values

The following table shows the allowed values for ***ReservationCharge*** (*ReservationChargeEnumeration*).

Table 190 – ReservationCharge – Allowed values

Value	Description
noFee	No reservation fee.
fee	Reservation fee.
singleFeeForReturnTrip	Single reservation fee for return trip.
feeForEachDirection	Separate reservation fee is for each direction of travel.
feeForEachLeg	Separate reservation fee is for each leg.

7.6.1.3.9.2.2 FeeBasis – Allowed values

The following table shows the allowed values for ***FeeBasis*** (*PerBasisEnumeration*).

Table 191 – PerBasis – Allowed values

Value	Description
perOffer	Refund is per offer.
perPerson	Refund is per person.

7.6.1.3.9.2.3 SeatAllocationMethod – Allowed values

The following table shows the allowed values for ***SeatAllocationMethod*** (*SeatAllocationMethod-Enumeration*).

Table 192 – *SeatAllocationMethod* – Allowed values

Value	Description
autoAssignment	A seat will be assigned automatically by an algorithm.
seatMap	The passenger may choose a specific seat from the available seats.
openSeating	It is not possible to reserve a specific seat

7.6.1.3.9.3 Cancelling – Model Element

Requirements for cancelling a booking.

Table 193 – *Cancelling* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*UsageParameter*</u>	::>	CANCELLING inherits from USAGE PARAMETER.
«PK»	***id***	*CancellingIdType*	1:1	Identifier of CANCELLING element.
	Booking-Arrangements	<u>*BookingArrangements*</u>	0:1	Arrangements for cancelling a booking. See Part1 Service restrictions Model

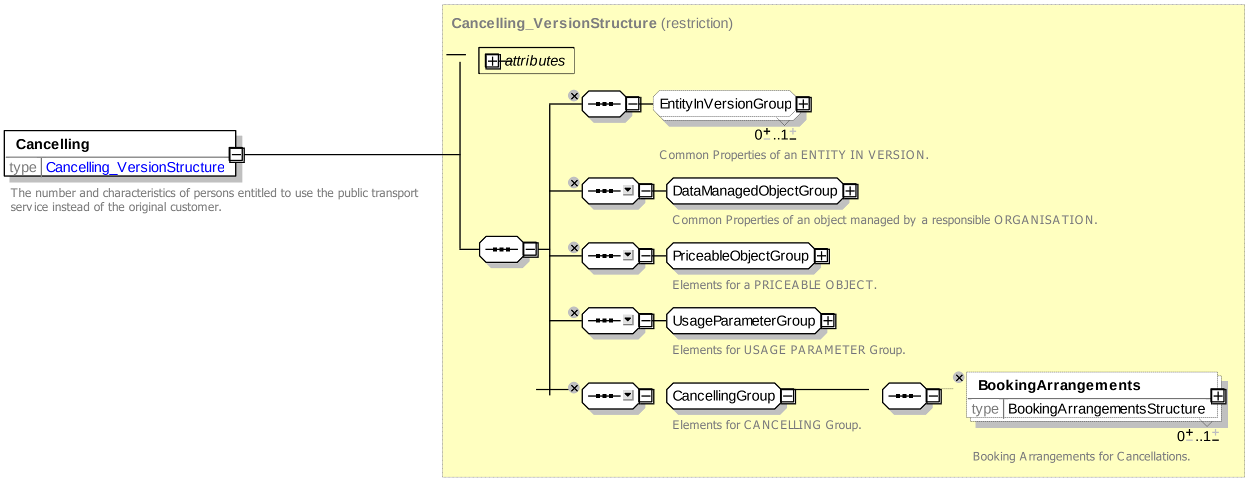


Figure 215 — Cancelling — XSD

7.6.1.3.9.4 BookingArrangements – *Group*

Information about booking to make a cancellation or other change. See also Part1 for details.

Table 194 – *BookingArrangements* Group– Group

Classification	Name	Type	Cardinality	Description
	BookingContact	*Contact*	0:1	Contact for Booking.
«enum»	***BookingMethods***	*BookingMethodEnum*	0:*	Booking method for FLEXIBLE LINE.
«enum»	***BookingAccess***	*BookingAccessEnum*	0:1	Who can make a booking. See Part1.
«enum»	***BookWhen***	*PurchaseWhenEnum*	0:1	When Booking can be made. See Part1
«enum»	***BuyWhen***	*PurchaseMomentEnum*	0:*	When purchase can be made. See Part1.
	LatestBooking-Time	*xsd:time*	0:1	Latest time in day that booking can be made.
	MinimumBookingPeriod	*xsd:duration*	0:1	Minimum interval in advance of departure day or time that service may be ordered.
	BookingUrl	*xsd:anyURI*	0:1	URL for booking.
	BookingNote	*MultilingualString*	0:1	Note about booking the FLEXIBLE LINE.

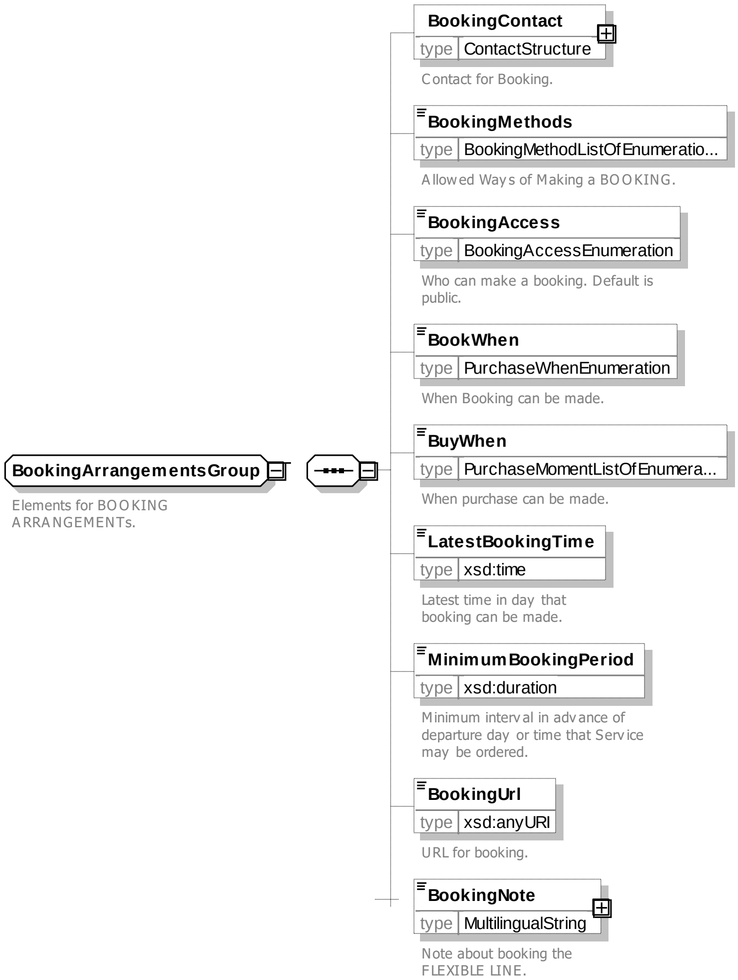


Figure 216 — *BookingArrangementsGroup* — XSD

7.6.1.3.1 Usage Parameter: After Sales – Attributes and XSD

7.6.1.3.1.1 Transferability – Model Element

The number and characteristics of persons entitled to use the public transport service instead of the original customer.

Table 195 – *Transferability* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	TRANSFERABILITY inherits from USAGE PARAMETER.
«PK»	***id***	*TransferabilityIdType*	1:1	Identifier of TRANSFERABILITY.
	CanTransfer	*xsd:boolean*	0:1	Whether ticket can be transferred to someone else.
	Maximum-NumberOf-Named-Transferees	*NumberOfPassengers*	0:1	Where a product can be used by a limited number of named users, maximum number of users allowed.

	HasTransferFee	*xsd:boolean*	0:1	Whether there is a charge for making a transfer.
«enum»	***SharedUsage***	*SharedUsageEnum*	0:1	Indicates the nature of the permitted sharing, if any, of products that can be shared, e.g. Trips from a multi-trip carnet. See allowed values +v1.1

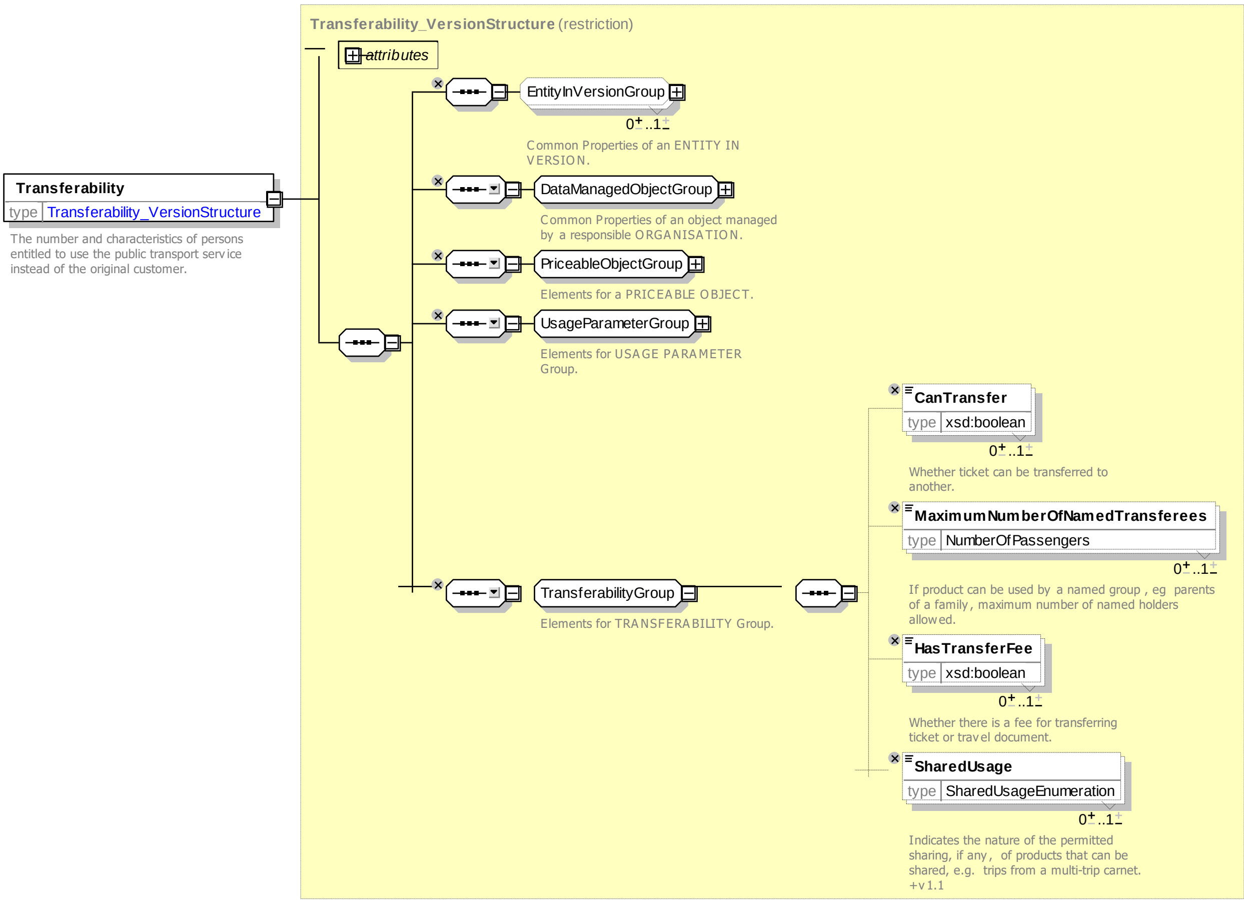


Figure 217 — *Transferability* — XSD

7.6.1.3.1.1.1 SharedUsage – Allowed values

The following table shows the allowed values for ***SharedUsage*** *(SharedUsageEnumeration)*.

Table 196 – *SharedUsage* – Allowed values

Value	Description
singleUser	Product can only be used by only one person at a time.
concurrent Users	Product can be used by several persons at a time., e.g. carnet.
concurrentDesignatedUsers	Product can be shared at the same time but only with designated types of companions, e.g. children.

7.6.1.3.1.2 Reselling – Model Element

Common resale conditions (i.e. for exchange or refund) attaching to the product.

Table 197 – *Reselling* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	RESELLING inherits from USAGE PARAMETER.
«PK»	***id***	*ResellingIdType*	1:1	Identifier of RESELLING.
«enum»	***Allowed***	*ResellTypeEnum*	0:1	Whether exchange or refund is allowed. See allowed values below.
	CanChangeClass	*xsd:boolean*	0:1	Whether user can change class.
	Unused-TicketsOnly	*xsd:boolean*	0:1	Whether it is possible to exchange partially used tickets.
	OnlyAtCertain-Distribution-Points	*xsd:boolean*	0:1	Whether distribution is restricted to certain points.
XGRP	***Reselling-PeriodGroup***	***xmlGroup***	0:1	When Period may take place.
	HasFee	*xsd:boolean*	0:1	Whether these is a fee for a refund or exchange.
«enum»	***RefundBasis***	*PerBasisEnum*	0:1	Basis on which refund is made. See allowed values below.
«enum»	***PaymentMethods***	*PaymentMethodEnum*	0:*	PAYMENT METHODs that may be used for transaction. See Part1 RC service Restriction model. +v1.1
«ctd»	***TypeOfPayment Method***	*TypeOfPaymentMethod*	0:*	PAYMENT METHODs that may be used for transaction. +v1.1

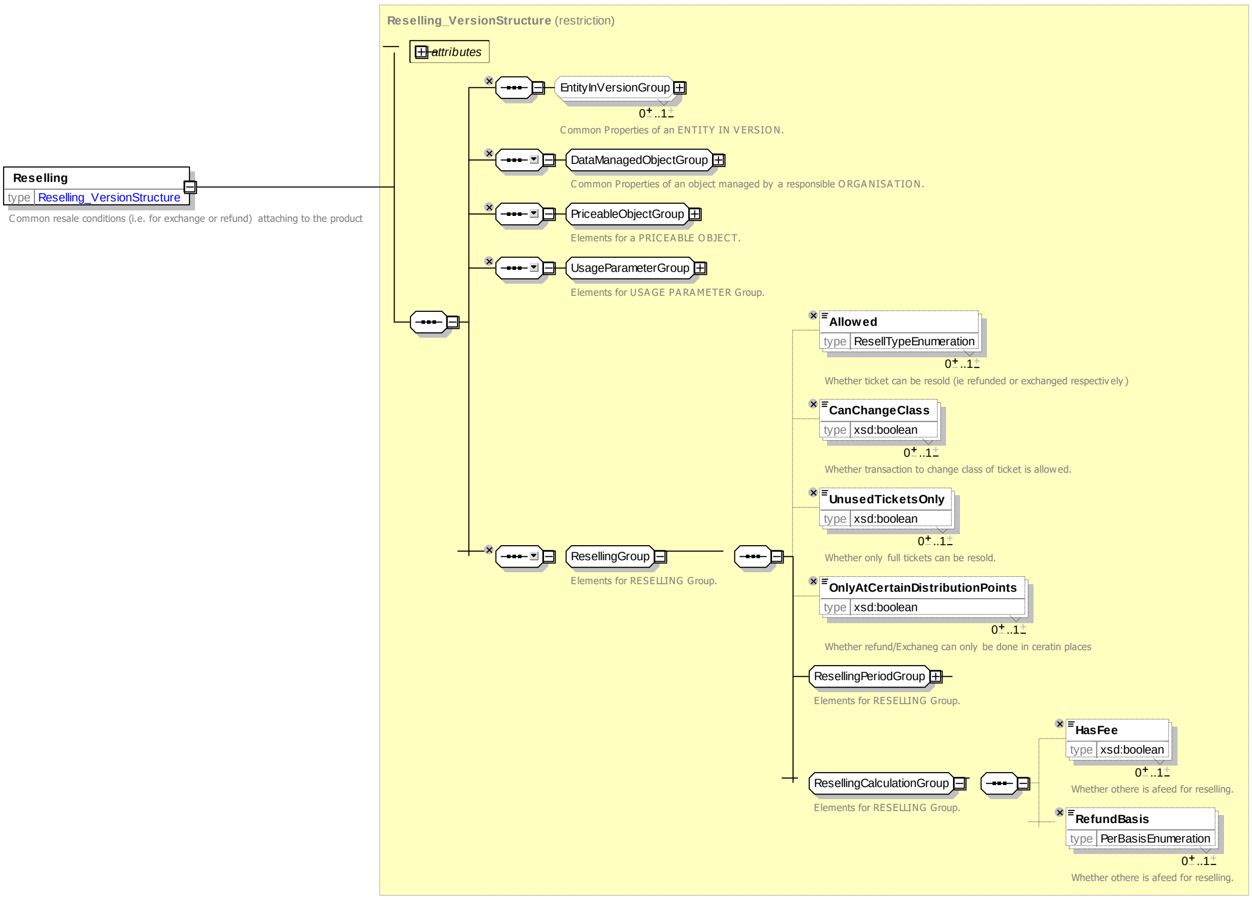


Figure 218 — *Reselling* — XSD

7.6.1.3.1.2.1 ResellType – Allowed values

The following table shows the allowed values for ***ResellType*** (*ResellTypeEnumeration*).

Table 198 – ResellType – Allowed values

Value	Description
none	Ticket can never be exchanged or refunded.
partial	Partial refund or exchange allowed.
full	Full refund allowed.

7.6.1.3.1.3 ResellingPeriod – Group

The ResellingPeriod group species when a refund or exchange may take place.

Classifi-cation	Name		Type	Cardinality	Description
«enum»	***ResellWhen***		*ResellWhenEnum*	0:1	Event marking when the is exchangeable status of the ticket changes. See allowed values below.
			CHOICE		From when refund/exchange can be made
	a	***Exchangeable-FromAnyTime***	*EmptyType*	0:1	Can be exchanged or refunded from any point after purchase.

	b	***Exchangable-FromDuration***	*xsd:duration*	0:1	Duration to start of period before (negative) or after (positive) the trigger point, (i.e. either Start Of Validity or First Use) after which ticket may be exchanged or refunded.
	c	***Exchangable-FromPercentUse***	*xsd:decimal*	0:1	Can be exchanged once a certain percentage of duration or use has been achieved. +v1.1
«FK»	**Exchangeable-FromIntervalRef**		*TimeIntervalRef*	0:1	TimeInterval determining period from which exchange can be made relative to trigger point.
			CHOICE		Until when refund/exchange can be made
	a	***Exchangeable-UntilAnyTime***	*EmptyType*	0:1	Can be exchanged or refunded up until any point after purchase.
		Exchangable-UntilDuration	*xsd:duration*	0:1	Duration to end of period before (negative) or after (positive) the trigger point (i.e. either Start Of Validity or First Use) after which ticket may be exchanged or refunded.
		Exchangable-UntilPercentUse	*xsd:decimal*	0:1	Can be exchanged until a certain percentage of duration or use has been achieved. +v1.1
«FK»	**Exchangeable-UntilIntervalRef**		*TimeIntervalRef*	0:1	TimeInterval determining period up until which exchange can be made relative to trigger point.
«enum»	**EffectiveFrom**		*EffectiveFromEnum*		Constraint on when change can be made see allowed values. +v1.1
	NotificationPeriod		*xsd:duration*	0:1	Notice period needed before action is effective. +v1.1

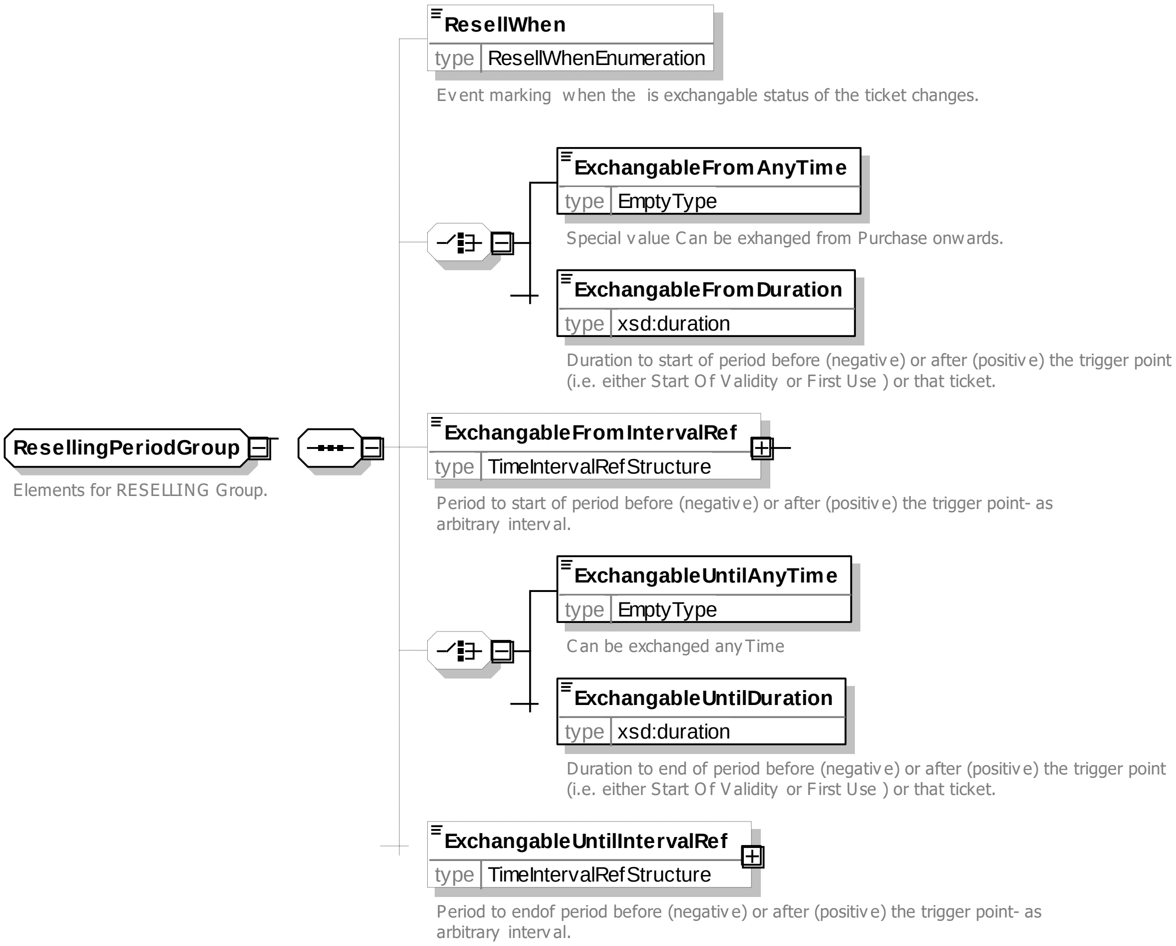


Figure 219 — ResellingPeriodGroup — XSD

7.6.1.3.1.3.1 ResellWhen – Allowed values

The following table shows the allowed values for ***ResellWhen*** (*ResellWhenEnumeration*).

Table 199 – *ResellWhen* – Allowed values

Value	Description
never	No transaction allowed, i.e. Ticket can never be exchanged or refunded.
withinPurchaseGracePeriod	Transaction allowed within purchase cool off period.
beforeStartOfValidityPeriod	Transaction allowed before start of Validity period of ticket.
afterStartOfValidityPariod	Transaction allowed after start of Validity period of ticket.
afterEndOfValidityPariod	Transaction allowed after end of Validity period of ticket.
beforeFirstUse	Transaction allowed before ticket first used.
afterFirstUse	Transaction still allowed after ticket has been partially used.
beforeValidation	Transaction allowed before ticket first validated.

afterValidation	Transaction allowed after ticket first validated.
withinSpecifiedWindow	Transaction allowed within specified window.
anyTime	Transaction allowed at any time.
other	Other condition.

7.6.1.3.1.3.2 EffectiveFrom – Allowed values

The following table shows the allowed values for ***EffectiveFrom*** *(EffectiveFromEnumeration)*.

Table 200 – *EffectiveFrom* – Allowed values

Value	Description
never	Cannot be made at any time.
nextInterval	Can take place at end of next product interval.
nextInstallment	Can take place at end of next subscription instalment.
anyTime	Can be made at any time.
other	Other condition

7.6.1.3.1.4 Exchanging – Model Element

Whether and how access rights may be exchanged for other access rights.

Table 201 – *Exchanging* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Reselling*	::>	EXCHANGING inherits from RESELLING.
«PK»	***id***	*ExchangingIdType*	1:1	Identifier of EXCHANGING.
	NumberOf-Exchanges-Allowed	*xsd:integer*	0:1	Number of times a ticket may be exchanged.
«enum»	***ToFareClass***	*FareClassEnum*	0:1	Fare class to which can be exchanged. See NeTEx Part1. (From class would be expression as the Seat class on an ACCESS RIGHT PARAMETER ASSIGNMENT.)
«FK»	***ToClass-OfUseRef***	*ClassOfUseRef*	0:1	CLASS OF USE class to which can be exchanged.
«enum»	***ExchangableTo***	*ExchangableToEnum*	0:1	Type of exchange allowed. The default is '*anyProduct*', i.e. to any other fare. See allowed values below.

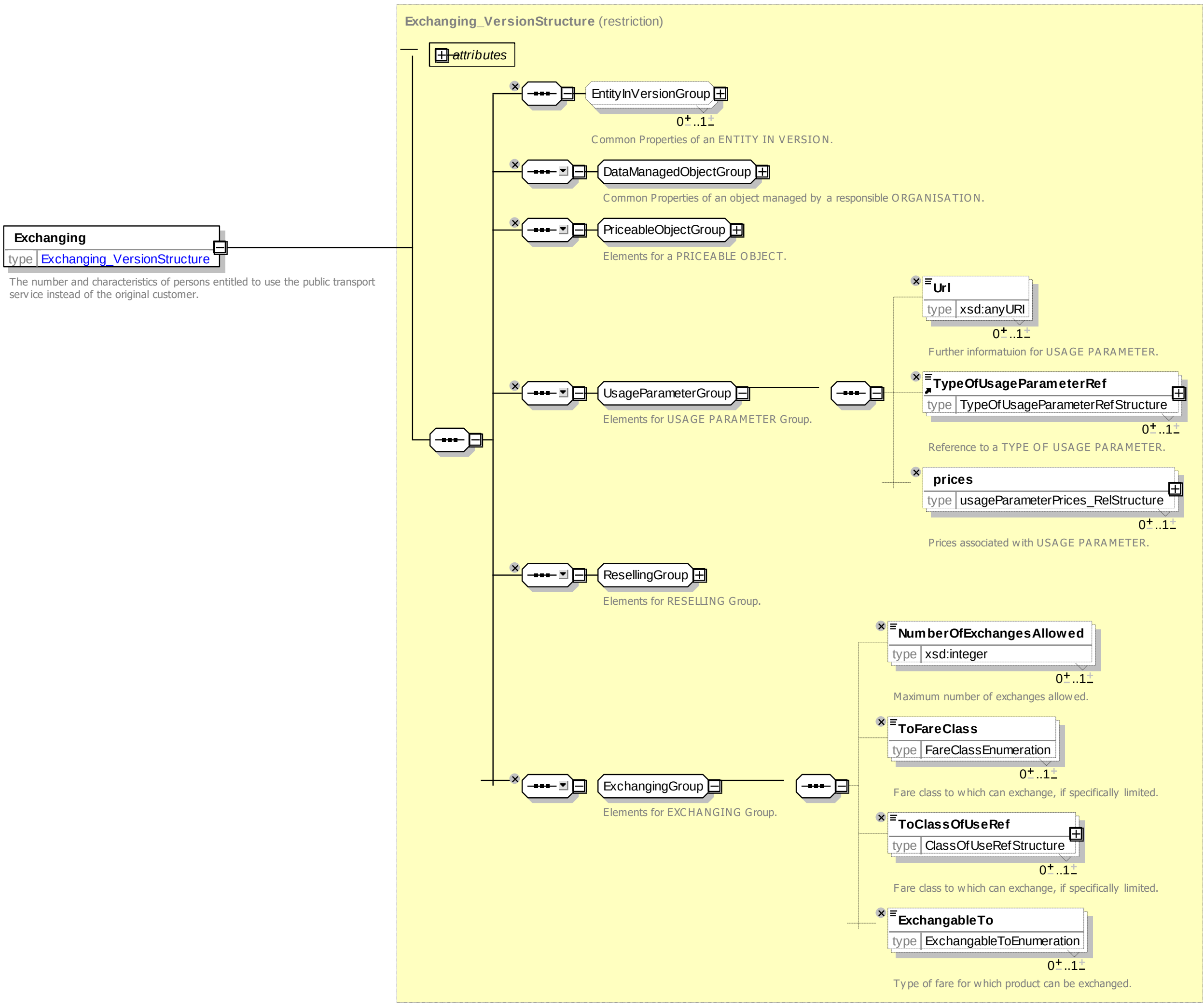


Figure 220 — *Exchanging* — XSD

7.6.1.3.1.4.1 ExchangableTo – Allowed values

The following table shows the allowed values for ***ExchangableTo*** *(ExchangableToEnumeration)*.

Table 202 – *ExchangableTo* – Allowed values

Value	Description
anyProduct	Can exchange to any other fare.
sameProductSameDay	Can exchange to fares of the same type for travel on the same date.
sameProductLongerJourney	Can exchange to fares of the same type for longer journey. +v1.1
sameProductShorterJourney	Can exchange to fares of the same type for shorter journey. +v1.1
sameProductAnyDay	Can exchange to fares of the same type for travel on the any date.
upgradeToStandardFare	Can exchange as upgrade to full standard fare.
upgradeToSpecifiedFare	Can exchange as an upgrade to a specified fare.

downgradeToSpecifiedFare	Can exchange as a downgrade to a specified fare.
equivalentProduct	Can exchange for an equivalent product.
changeGroupSize	Can change group size
other	Other condition.

7.6.1.3.1.5 Refunding – Model Element

Whether and how the product may be refunded.

Table 203 – *Refunding* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Reselling*	::>	REFUNDING inherits from RESELLING.
«PK»	***id***	*RefundingIdType*	1:1	Identifier of REFUNDING.
«enum»	***RefundType***	*RefundTypeEnum*	0:1	Classification of REFUNDING. See allowed values below.
«enum»	***RefundPolicy***	*RefundPolicyEnum*	0:*	Reasons for giving refunds. See allowed values. +v1.1
«enum»	***PartialRefund-Basis***	*PartialRefundBasisEnum*	0:*	Basis on which partial refunds of period passes etc are calculated. See allowed values. +v1.1
«enum»	***PaymentMethod***	*PaymentMethodEnum*	0:*	DEPRECATED – Use ***PaymentMethods*** on RESELLING higher in hierarchy

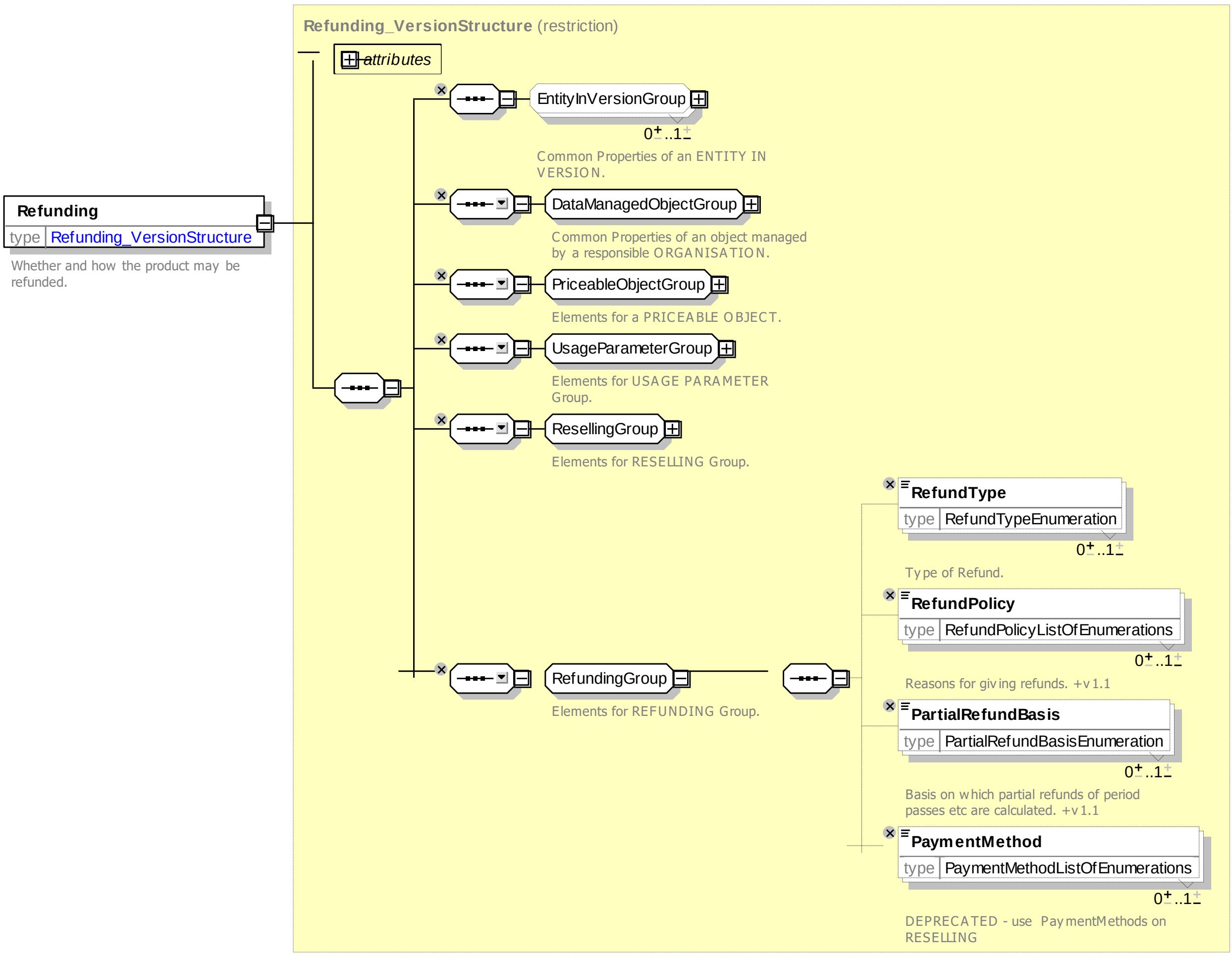


Figure 221 — *Refunding* — XSD

7.6.1.3.1.5.1 RefundType – Allowed values

The following table shows the allowed values for ***RefundType*** *(RefundTypeEnumeration)*.

Table 204 – *RefundType* – Allowed values

Value	Description
unused	Refund is because the product was unused.
delay	Refund is because the passenger’s trip was delayed.
cancellation	Refund is because the passenger‘s journey was cancelled.
partialJourney	Refund is because the product was only party unused.
earlyTermination	Refund is because the product was terminated early. +v1.1
changeOfGroupSize	Refund is because the group size was changed. +v1.1
other	Refund is because of some other reason.

7.6.1.3.1.5.2 RefundPolicy – Allowed values

The following table shows the allowed values for ***RefundPolicy*** *(RefundPolicyEnumeration)*.

Table 205 – *RefundPolicy* – Allowed values

Value	Description
any	Any reason
illness	Death
death	Refund of unused ticket.
maternity	Maternity
redundancy	Redundancy
changeOfEmployment	Change of Employment.
changeOfResidence	Change of Residence.
none	No refunds given.
other	Other reason.

7.6.1.3.1.5.3 PartialRefundBasis – Allowed values

The following table shows the allowed values for ***PartialRefundBasis*** *(PartialRefundBasisEnumeration)*.

Table 206 – *PartialRefundBasis* – Allowed values

Value	Description
unusedDays	Refund is for unused days.
unusedWeeks	Refund is for unused whole weeks.
unusedMonths	Refund is for unused whole months.
unusedSemesters	Refund is for unused whole academic terms.
other	Other refund.

7.6.1.3.1.6 Replacing – Model Element

Whether and how access rights may be replaced if lost or stolen.

Table 207 – *Replacing* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Reselling*	::>	REPLACING inherits from RESELLING.
«PK»	***id***	*ReplacingIdType*	1:1	Identifier of REPLACING.

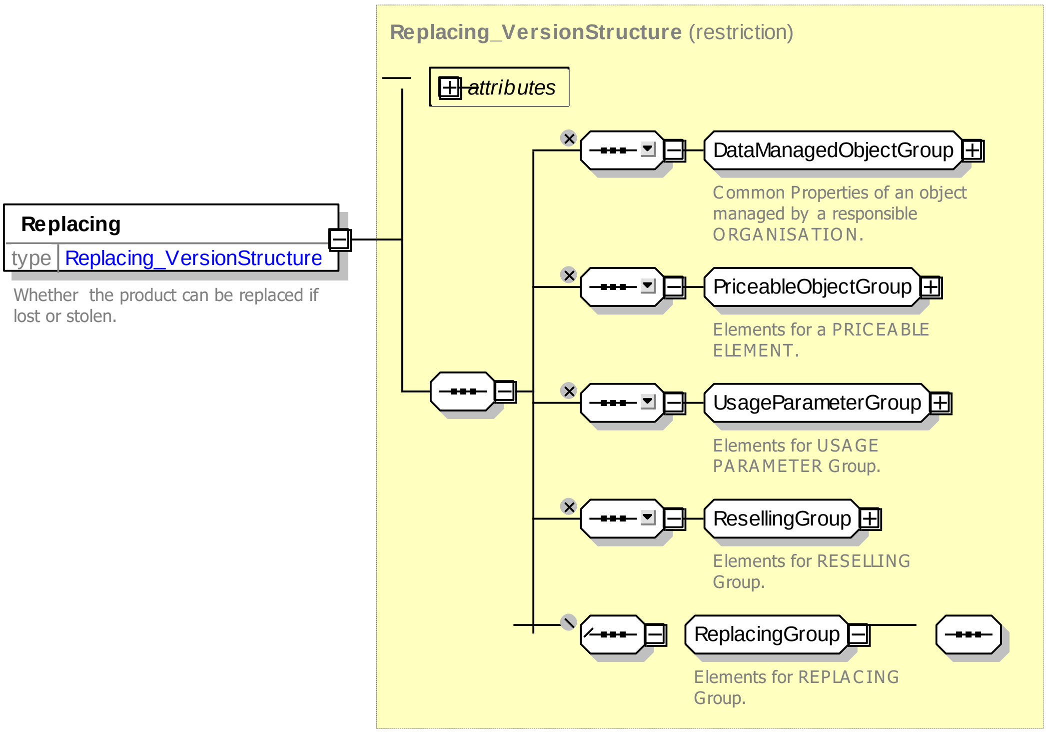


Figure 222 — *Replacing* — XSD

7.6.1.3.2 **Usage Parameter: Charging – Attributes and XSD**

7.6.1.3.2.1 ChargingPolicy – Model Element

Policy regarding different aspects of charging such as credit limits.

Table 208 – *ChargingPolicy* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	CHARGING POLICY inherits from USAGE PARAMETER.
«PK»	***id***	*ChargingPolicyIdType*	1:1	Identifier of CHARGING POLICY.
«enum»	***CreditPolicy***	*CreditPolicyEnumeration*	0:1	Policy for traveling on credit – See allowed values below.
"	***ExpireAfter-Period***	*xsd:duration*	0:1	Any expiry period on collecting a rebate or adjustment.
	PaymentGrace-Period	*xsd:duration*	0:1	Period after purchase by which time payment must be settled. +v1.1
«enum»	***BillingPolicy***	*TravelBillingPolicy-Enumeration*	0:1	Policy for billing frequency – See Allowed values below. +v1.1

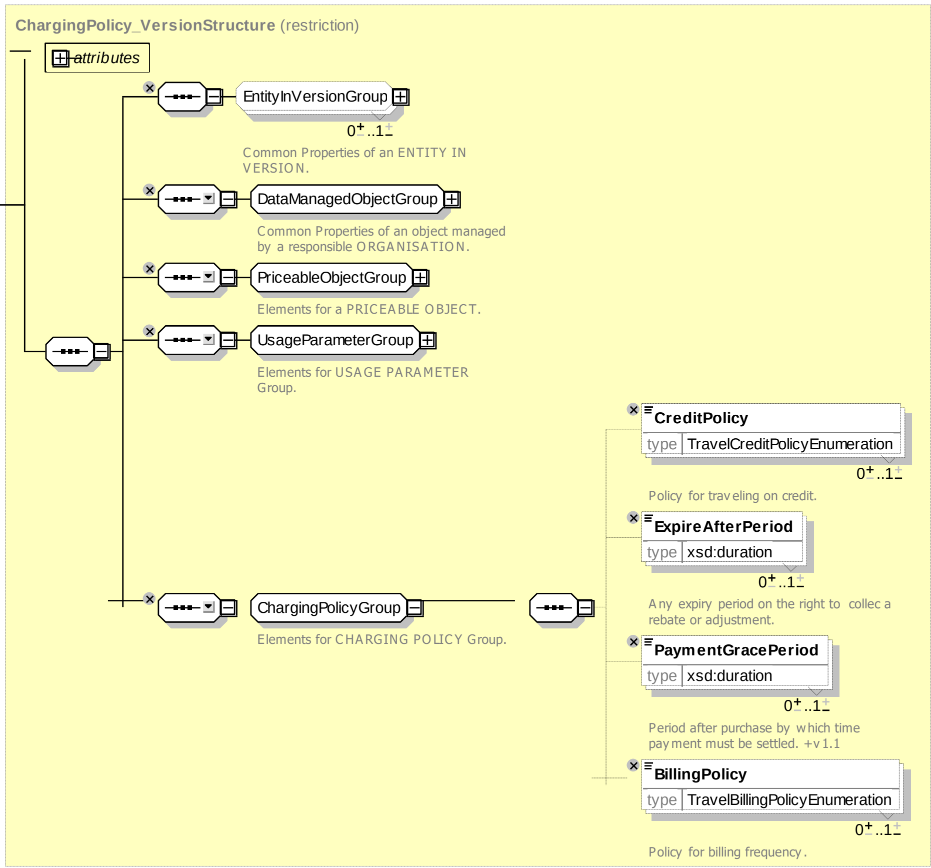


Figure 223 — *ChargingPolicy* — XSD

7.6.1.3.2.1.1 TravelCreditPolicy – Allowed values

The following table shows the allowed values for ***TravelCreditPolicy*** (*TravelCreditPolicyEnumeration)*.

Table 209 – *TravelCreditPolicy* – Allowed values

Value	Description
allowTravel	Can travel even if credit is negative.
blockPayAsYouGoTravel	Block all pay as you go travel but allow prepaid travel.
blockAllTravel	Block all travel, even using other products.
other	Other policy.

7.6.1.3.2.1.2 TravelBillingPolicy – Allowed values

The following table shows the allowed values for ***TravelBillingPolicy*** (*TravelBillingPolicyEnumeration*).

Table 210 – *TravelBillingPolicy* – Allowed values

Value	Description
billAsYouGo	Bill for use immediately on incurring travel.
billOnThreshold	Only raise bill when threshold is reached.
billAtFareDayEnd	Bill at end of every fare day.
billAtPeriodEnd	Bill at end of a specified period.

7.6.1.3.2.2 PenaltyPolicy – Model Element

Policy regarding different aspects of penalty charges, for example repeated entry at the same station, no ticket etc.

Table 211 – PenaltyPolicy – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	PENALTY POLICY inherits from USAGE PARAMETER.
«PK»	***id***	*PenaltyPolicyIdType*	1:1	Identifier of PENALTY POLICY.
«enum»	***PenaltyPolicyType***	*PenaltyPolicyEnum*	0:1	Classification of Penalty Policy. See below.
«enum»	***SameStation-EntryPolicy***	*SameStation-EntryPolicyEnum*	0:1	Policy for allowing re-entry at the same station within a certain time. See below.
	MinimumTime-BeforeRentry	*xsd:duration*	0:1	Minimum time before can re-enter at the same station before incurring penalty.
	Maximum-NumberOfFail-ToCheckOutEvents	*xsd:duration*	0:1	Limit on the number of fail-to-checkout events allowed before suspension. +v1.1

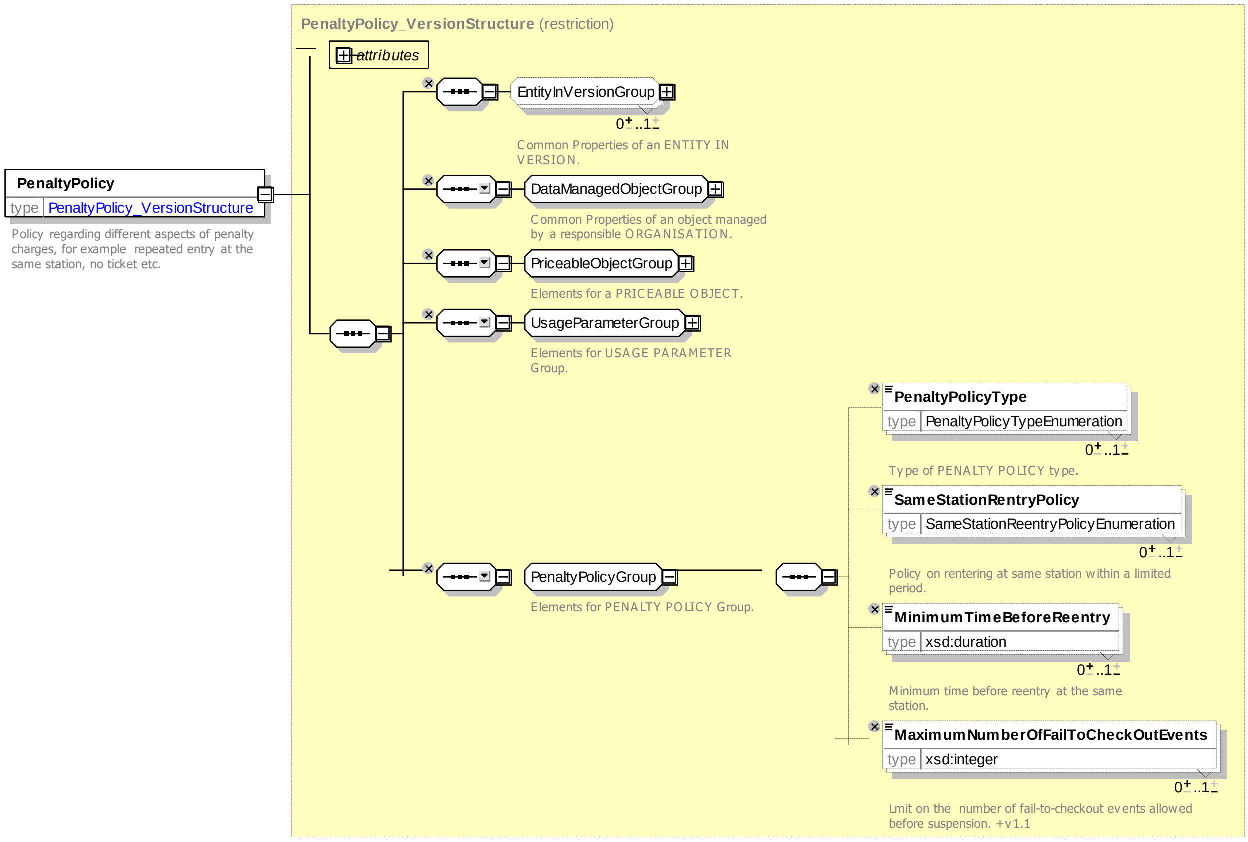


Figure 224 — PenaltyPolicy — XSD

7.6.1.3.2.2.1 PenaltyPolicyType – Allowed values

The following table shows the allowed values for PenaltyPolicyType (PenaltyPolicyTypeEnumeration).

Table 212 – PenaltyPolicyType – Allowed values

Value	Description
noTicket	Penalty is for having no ticket.
noCheckIn	Penalty is incurred if failed to check in
noCheckOut	Penalty is incurred if checked in and failed to check out.
noValidation	Penalty is incurred if have valid ticket but failed to validate it.
other	Other type of penalty.

7.6.1.3.2.2.2 SameStationEntryPolicy – Allowed values

The following table shows the allowed values for ***SameStationEntryPolicy*** (*SameStationEntryPolicyEnumeration)*.

Table 213 – *SameStationEntryPolicy* – Allowed values

Value	**Description**
blocked	Re-entry not allowed.
newFare	Re-entry allowed and new fare charged.
maximumFare	Charge maximum fare to complete previous journey and start new journey.
allowed	Can re-enter without penalty and resume journey.

7.6.1.3.2.3 Subscribing – Model Element

Parameters governing subscription to a product allowing payment at regular intervals.

Classifi-cation	**Name**	**Type**	**Cardin ality**	**Description**
::>	::>	*UsageParameter*	::>	SUBSCRIBING inherits from USAGE PARAMETER.
«PK»	***id***	*SubscribingIdType*	1:1	Identifier of SUBSCRIBING.
«enum»	***Subscription-TermType***	*SubscriptionTermTypeEnum*	0:1	Types of subscription term allowed. See allowed values below.
	Minimum-Subscription-Period	*duration*	0:1	Minimum duration allowed for a subscription.

	Maximum-Subscription-Period	*duration*	0:1	Maximum duration allowed for a subscription.
«enum»	***Subscription-RenewalPolicy***	*SubscriptionRenewalPolicy-Enum*	0:1	Policy on renewing subscription. See allowed values below.
«cntd»	***possibleInstallmentIntervals***	*TimeIntervalRef*	0:*	Allowed billing Intervals for payment in installment.r
«enum»	***Installment-PaymentMethods***	*PaymentMethodsEnum*	0:1	Allowed means of payment of installations as standard value. See allowed values.
«cntd»	***installment-PaymentMethods***	*TypeOfPaymentMethod*	0:*	Allowed means of payment of installations as TYPE OF PAYMENT METHOD.

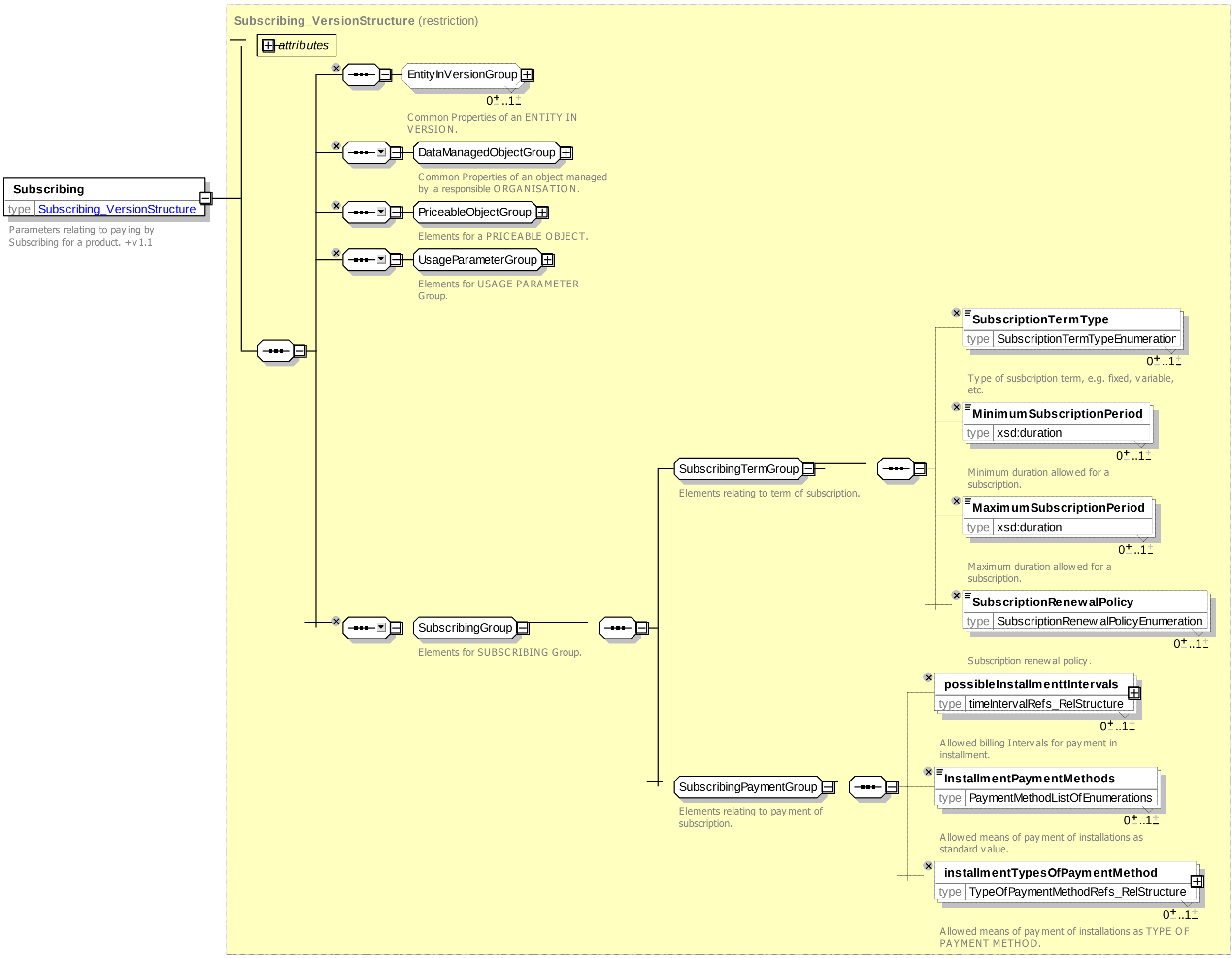


Figure 225 — Subscribing — XSD

7.6.1.3.2.3.1 SubscriptionTermType– Allowed values

The following table shows the allowed values for ***SubscriptionTermType*** (*SubscriptionTermTypeEnumeration*).

Table 214 – *SubscriptionTermType* – Allowed values

Value	Description
fixed	Subscription must be for a fixed term.
variable	Subscription can be for an arbitrary term.
openEnded	Subscription can be open-ended.

7.6.1.3.2.3.2 SubscriptionRenewalPolicy– Allowed values

The following table shows the allowed values for ***SubscriptionRenewalPolicy*** (*SubscriptionRenewal-PolicyEnumeration*).

Table 215 – *SubscriptionRenewalPolicy*– Allowed values

Value	Description
automatic	Renew automatically at end of term.
manual	Renew on request.
automaticOnConfirmation	Confirm and renew automatically at end of subscription term
none	No renewal allowed.
other	Other.

7.6.1.3.3 **Usage Parameter: Travel – XML examples**

7.6.1.3.3.1 Usage Parameter: Travel: XML Example of Round Trip parameter

The following code fragment shows two ROUND TRIP usage parameters for *single* and *return* journeys.

For EXAMPLE:

```
<RoundTrip version="any" id="tfl:single">
    <TripType>single</TripType>
</RoundTrip>

<RoundTrip version="any" id="tfl:return">
    <TripType>return</TripType>
</RoundTrip>
```

7.6.1.3.3.2 Usage Parameter: Travel: XML Example of Routing parameter

The following code fragment shows two ROUTING usage parameters one restricting routes to the same as the outbound, one restricting to cross-border routes

For EXAMPLE:

```
<Routing version="any" id="tap:return_same_Way">
    <IsRestricted>true</IsRestricted>
    <ReturnRouteIdentical>true</ReturnRouteIdentical>
</Routing>

<Routing version="any" id="tap:international">
    <Name>International</Name>
    <CrossBorder>true</CrossBorder>
</Routing>
```

7.6.1.3.3.3 Usage Parameter: Travel: XML Example of Frequency of Use parameter

The following code fragment shows two FREQUENCY OF USE usage parameters one for a single ride, one for unlimited use as on a season pass.

For EXAMPLE:

```
<FrequencyOfUse version="any" id="tfl:one_trip">
    <Name>Single use </Name>
    <FrequencyOfUseType>single</FrequencyOfUseType>
    <MinimalFrequency>1</MinimalFrequency>
    <MaximalFrequency>1</MaximalFrequency>
</FrequencyOfUse>

<FrequencyOfUse version="any" id="tfl:unlimited_use">
    <Name>Unlimited use pass</Name>
    <FrequencyOfUseType>unlimited</FrequencyOfUseType>
</FrequencyOfUse>
```

7.6.1.3.3.4 Usage Parameter: Travel: XML Example of Interchanging parameter

The following code fragment shows two INTERCHANGING usage parameters one prohibiting any change, the other allowing interchanges but not journey breaks (i.e. leaving the station for an extended period).

For EXAMPLE:

```
<Interchanging version="any" id="tfl:NoInterchange_CannotBreakJourney">
    <CanInterchange>false</CanInterchange>
    <CanBreakJourney>false</CanBreakJourney>
</Interchanging>

<Interchanging version="any" id="tfl:CanInterchange_CannotBreakJourney">
    <CanInterchange>true</CanInterchange>
    <CanBreakJourney>false</CanBreakJourney>
</Interchanging>
```

7.6.1.3.3.5 Usage Parameter: Travel: XML Example of Usage Validity Period parameter

The following code fragment shows various USAGE VALIDITY PERIOD usage parameters, for simple rides or until eh end of the fare day.

For EXAMPLE:

```
<UsageValidityPeriod version="any" id="tfl:end_of_ride">
    <Name>Valid only until end of a single ride complete</Name>
    <UsageEnd>endOfRide</UsageEnd>
</UsageValidityPeriod>

<UsageValidityPeriod version="any" id="tfl: end_of_return">
    <Name>Valid only until end of a return ride complete</Name>
    <UsageTrigger>startOutboundRide</UsageTrigger>
    <UsageEnd>endOfTrip</UsageEnd>
</UsageValidityPeriod>

<UsageValidityPeriod version="any" id="tfl: end_of_fare_day">
    <Name>Valid until ride complete</Name>
    <UsageEnd>endOfFareDay</UsageEnd>
</UsageValidityPeriod>
```

The following code fragment shows various USAGE VALIDITY PERIOD usage parameters that depend on the life time of the product purchased, for time limited rides.

For EXAMPLE:

```
<UsageValidityPeriod version="any" id="tfl:TravelCard@1D_Off_peak">
    <Name>Travel Card valid 1 day off peak </Name>
```

```xml
        <UsageTrigger>specifiedStartDate</UsageTrigger>
        <UsageEnd>productExpiry</UsageEnd>
        <StandardDuration>P1D</StandardDuration>
        <BlackoutUse>noTravelWithinPeriod</BlackoutUse>
    </UsageValidityPeriod>

    <UsageValidityPeriod version="any" id="tfl: TravelCard@1D_any_time">
        <Name>Travel Card valid 1 day </Name>
        <UsageTrigger>specifiedStartDate</UsageTrigger>
        <UsageEnd>productExpiry</UsageEnd>
        <StandardDuration>P1D</StandardDuration>
        <BlackoutUse>mayTravelAnytime</BlackoutUse>
    </UsageValidityPeriod>

    <UsageValidityPeriod version="any" id="tfl: TravelCard@@6M_any_time">
        <Name>Travel Card cards valid 6 month </Name>
        <UsageTrigger>specifiedStartDate</UsageTrigger>
        <UsageEnd>productExpiry</UsageEnd>
        <StandardDuration>P6M</StandardDuration>
        <BlackoutUse>mayTravelAnytime</BlackoutUse>
    </UsageValidityPeriod>

    <UsageValidityPeriod version="any" id="tfl: TravelCard@18Plus_1Y">
        <Name>Travel Card cards valid for course length, max 3 years </Name>
        <Description>If you're on a course that is longer than one year, you can get a card valid
for the length of your course or up to three years, whichever is sooner.</Description>
        <UsageTrigger>specifiedStartDate</UsageTrigger>
        <UsageEnd>profileExpiry</UsageEnd>
        <StandardDuration>P1Y</StandardDuration>
    </UsageValidityPeriod>

    <UsageValidityPeriod version="any" id="tfl:TravelCardValidityPeriod@18Plus3Y">
        <Name>Travel Card cards valid for course length, max 3 years </Name>
        <Description>If you're on a course that is longer than one year, you can get a card valid
for the length of your course or up to three years, whichever is sooner.</Description>
        <UsageTrigger>specifiedStartDate</UsageTrigger>
        <UsageEnd> productExpiry </UsageEnd>
        <StandardDuration>P3Y</StandardDuration>
    </UsageValidityPeriod>
```

7.6.1.3.3.6 Usage Parameter: Travel: XML Example of Minimum Stay parameter

The following code fragment shows various MINIMUM STAY usage parameters, for any three nights or a weekend.

For EXAMPLE:

```xml
    <MinimumStay id="tap:3nights_away" version="any">
        <Name>Minimum stay for three nights</Name>
        <MinimumStayType>countNightsAway</MinimumStayType>
        <MinimumNumberOfNightsAway>3</MinimumNumberOfNightsAway>
    </MinimumStay>

    <MinimumStay id="tap:weekend" version="any">
        <Name>Minimum stay for weekend</Name>
        <MinimumStayType>specified_nights_away</MinimumStayType>
        <RequiresNightsAway>Saturday</RequiresNightsAway>
        <MinimumNumberOfNightsAway>1</MinimumNumberOfNightsAway>
        <MaximumNumberOfNightsAway>7</MaximumNumberOfNightsAway>
    </MinimumStay>

    <MinimumStay id="tap:weekend" version="any">
        <Name>Minimum stay for at least one day out of weekend </Name>
        <MinimumStayType>atLeastNSpecifiedNights</MinimumStayType>
        <RequiresNightsAway>Friday Saturday Sunday</RequiresNightsAway>
        <MinimumNumberOfNightsAway>1</MinimumNumberOfNightsAway>
    </MinimumStay>
```

7.6.1.3.4 Usage Parameter: Product – XML examples

7.6.1.3.4.1 Usage Parameter: Product: XML Example of User Profile parameter

The following code fragment shows various USER PROFILE usage parameters, for an adult, infant, child etc.

For EXAMPLE:

```
    <UserProfile version="any" id="tfl:adult">
        <Name>Adult </Name>
        <BaseUserProfileRef version="any" ref="tfl:anyone"/>
        <TypeOfConcessionRef version="any" ref="tfl:none"/>
        <MinimumAge>17</MinimumAge>
        <DiscountBasis>none</DiscountBasis>
    </UserProfile>

    <UserProfile version="any" id="tfl:disabled_person">
        <Name>Disabled Person Fare</Name>
        <BaseUserProfileRef version="any" ref="tfl:concession"/>
        <TypeOfConcessionRef version="any" ref="tfl:disabled_person"/>
        <DiscountBasis>discount</DiscountBasis>
    </UserProfile>

    <UserProfile version="any" id="tfl:infant">
        <Name>Child Fare</Name>
        <Description> Under-5s http://www.tfl.gov.uk/tickets/14414.aspx
Under-5s travel free if with someone who has a valid ticket, is using Oyster to pay as they go,
has a Freedom Pass, 60+ London Oyster photocard or a Veterans Oyster photocard.</Description>
        <BaseUserProfileRef version="any" ref="tfl:child"/>
        <TypeOfConcessionRef version="any" ref="tfl:infant"/>
        <MinimumAge>0</MinimumAge>
        <MaximumAge>4</MaximumAge>
        <DiscountBasis>free</DiscountBasis>
    </UserProfile>

    <UserProfile version="any" id="tfl:child11To15">
        <Name>11 to 15 year old</Name>
        <Description> http://www.tfl.gov.uk/tickets/14416.aspx

You must have an 11-15 Oyster photocard to get:
Child-rate fares on the Tube, DLR and London Overground
7 Day, Monthly and longer period Travel cards at child rate

Holders of the following can buy a Zones 1-9 Off-Peak Day Travel card after 10:00 for just £2 each
for up to four children travelling with them:
Gold Card, Network RailCard, Family and Friends RailCard, HM Forces RailCard
        </Description>
        <priceGroups>
            <FareTable version="any" id="tfl:child_11_to_15+11_to_15_Oyster_card_holder>
                <members>
                    <FareProductPrice version="any" id="tfl:child11To15@TravelCard_on_Oyster">
                        <DiscountAsPercentage>50.00</DiscountAsPercentage>
                        <PreassignedFareProductRef version="any" ref="tfl:PayAsYouGo_fare"/>
                    </FareProductPrice>
                    <FareProductPrice version="any" id="tfl:child_11_to_15@Oyster">
                        <Name> 50 % discount for Oyster travel</Name>
                        <DiscountAsPercentage>50.00</DiscountAsPercentage>
                        <PreassignedFareProductRef version="any" ref="tfl:PayAsYouGo_fare"/>
                    </FareProductPrice>
                </members>
                <EntitlementRequiredRef version="any" ref="tfl:Oyster_PayAsYouGo_right_holder"/>
            </FareTable>
            <FareTable version="any" id="tfl:child11To15@studentRailCard">
                <members>
                    <FareProductPrice version="any" id="tfl:child_11_to_15@student_RailCard">
                        <Name>34 % discount for RailCard travel</Name>
                        <DiscountAsPercentage>34.00</DiscountAsPercentage>
                        <PreassignedFareProductRef version="any" ref="tfl:Prepaid_fare"/>
                    </FareProductPrice>
                </members>
                <UserProfileRef version="any" ref="tfl:rail_card_holder/>
            </FareTable>
```

```xml
        </priceGroups>
        <BaseUserProfileRef version="any" ref="tfl:child"/>
        <TypeOfConcessionRef version="any" ref="tfl:child"/>
        <MinimumAge>11</MinimumAge>
        <MaximumAge>15</MaximumAge>
    </UserProfile>

    <UserProfile version="any" id="tfl:senior">
        <Name>Eligible for Oyster Freedom Pass Fare</Name>
        <BaseUserProfileRef version="any" ref="tfl:concession"/>
        <TypeOfConcessionRef version="any" ref="tfl:senior"/>
        <MinimumAge>60</MinimumAge>
    </UserProfile>

    <UserProfile version="any" id="tfl:senior_roesident">
        <Name>Eligible for Oyster Freedom Pass Fare</Name>
        <BaseUserProfileRef version="any" ref="tfl:senior"/>
        <TypeOfConcessionRef version="any" ref="tfl:senior"/>
        <MinimumAge>60</MinimumAge>
        <MonthDayOnWhichAgeApplies>--04-06</MonthDayOnWhichAgeApplies>
        <LocalResident>true</LocalResident>
    </UserProfile>
```

7.6.1.3.4.2 Usage Parameter: Product: XML Example of Group Ticket parameter

The following code fragment shows two different GROUP TICKET usage parameters, indicating the right to purchase other products.

For EXAMPLE:

```xml
    <GroupTicket version="any" id="tfl:groupDayTicket">
        <Name>Travel anytime</Name>
        <Description> Groups http://www.tfl.gov.uk/tickets/14416.aspx
        For groups of 10 or more fare-paying passengers you can get Group Day Tickets. Provides
unlimited travel at any time and on any day within the zones paid for on Tube, DLR, London
Overground and some National Rail services*. You can only get these as paper tickets, not on an
Oyster card.

Group Day tickets are available from Tube or London Overground station ticket offices and Tube
station touchscreen machines. You can also get Group Day Tickets from some National Rail stations
that Tube and London Overground services run through.

        </Description>
            <MinimumNumberOfPersons>10</MinimumNumberOfPersons>
    </GroupTicket>

    <GroupTicket version="any" id="tfl:family_group">
        <Name>up to 4 children allowed with adult Oyster holders</Name>
        <MinimumNumberOfCardHolders>1</MinimumNumberOfCardHolders>
        <members>
            <CompanionProfile version="any" id="tfl:family_group@adult">
                <UserProfileRef version="any" ref="tfl:adult"/>
                <MinimumNumberOfPersons>1</MinimumNumberOfPersons>
                <MaximumNumberOfPersons>2</MaximumNumberOfPersons>
            </CompanionProfile>
            <CompanionProfile version="any" id="tfl:family_group@children">
                <UserProfileRef version="any" ref="tfl:child"/>
                <MinimumNumberOfPersons>1</MinimumNumberOfPersons>
                <MaximumNumberOfPersons>4</MaximumNumberOfPersons>
            </CompanionProfile>
        </members>
        <MaximumPersonsFree>4</MaximumPersonsFree>
    </GroupTicket>
```

7.6.1.3.4.3 Usage Parameter: Product: XML Example of Entitlement Given parameter

The following code fragment shows various ENTITLEMENT GIVEN usage parameters, indicating the right to purchase other products.

For EXAMPLE:

```xml
    <EntitlementGiven version="any" id="tfl:TravelCard_discount">
```

```
        <Name>GIVING ENTITLEMENT use discount from a TravelCard</Name>
        <SaleDiscountRightRef version="any" ref="tfl:TravelCard_sale_discount"/>
        <EntitlementType>purchase</EntitlementType>
    </EntitlementGiven>

    <EntitlementGiven version="any" id="tfl:Oyster_PayAsYouGo_right">
        <Name>GIVING ENTITLEMENT right to buy Oyster discounted fares</Name>
        <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
        <EntitlementType>purchase</EntitlementType>
    </EntitlementGiven>

    <EntitlementGiven version="any" id="tfl:discounted_prepaid_fare">
        <Name>GIVING ENTITLEMENT a pay as you go fare</Name>
        <PreassignedFareProductRef version="any" ref="tfl:Prepaid_fare"/>
        <EntitlementType>purchaseAtDiscount</EntitlementType>
    </EntitlementGiven>

    <EntitlementGiven version="any" id="tfl:annual_pass_GoldCard">
        <Name>GIVING ENTITLEMENT Gold card because has annual pass </Name>
        <SaleDiscountRightRef version="any" ref="tfl:GoldCard"/>
        <EntitlementType>use</EntitlementType>
        <MinimumQualificationPeriod>P1Y</MinimumQualificationPeriod>
    </EntitlementGiven>
```

7.6.1.3.4.4 Usage Parameter: Product: XML Example of Entitlement Required parameter

The following code fragment shows various ENTITLEMENT REQUIRED indicating a requirement to hold other products.

For EXAMPLE:

```
    <EntitlementRequired version="any" id="tfl: Oyster_PayAsYouGo_right_holder">
        <Name>Entitlement from having Right to use Oyster pay as You go</Name>
        <CappedDiscountRightRef version="any" ref="tfl: Oyster_PayAsYouGo_right"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="tfl:travel_card_holder">
        <Name>Entitlement from having a TravelCard</Name>
        <PreassignedFareProductRef version="any" ref="tfl:TravelCard"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="tfl:Gold_card_holder>
        <Name>Entitlement from having a GoldCard</Name>
        <SaleDiscountRightRef version="any" ref="tfl:GoldCard"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="nr:rail_card_holder>
        <Name>Entitlement from having a RailCard</Name>
        <SaleDiscountRightRef version="any" ref="nr:RailCard"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="tfl:FreedomPass_hlder">
        <Name>Entitlement from having a Freedom Pass</Name>
        <SaleDiscountRightRef version="any" ref="tfl:FreedomPass"/>
    </EntitlementRequired>
```

7.6.1.3.4.5 Usage Parameter: Product: XML Example of Commercial Profile parameter

The following code fragment shows various COMMERCIAL PROFILE usage parameters, for an adult, infant, child etc.

7.6.1.3.5 Usage Parameter: Luggage Allowance – XML examples

7.6.1.3.5.1 Usage Parameter: Luggage Allowance: XML Example of Luggage allowance parameter

The following code fragment shows two LUGGAGE ALLOWANCE usage parameters for *carry-on* and *checked-in* baggage.

For EXAMPLE:

```
<LuggageAllowance version="any" id="tap:carry_on">
    <Name>Carry-on Baggage</Name>
    <BaggageUseType>checkIn</BaggageUseType>
    <BaggageType>handLuggage</BaggageType>
    <LuggageAllowanceType>singleBag</LuggageAllowanceType>
    <MaximumNumberItems>1</MaximumNumberItems>
    <MaximumBagHeight>1</MaximumBagHeight>
    <MaximumBagWidth>0.5</MaximumBagWidth>
    <MaximumBagDepth>0.3</MaximumBagDepth>
    <TotalWeight>10</TotalWeight>
</LuggageAllowance>

<LuggageAllowance version="any" id="tap:checkin">
    <Name>Check in</Name>
    <BaggageUseType>checkIn</BaggageUseType>
    <BaggageType>suitcase</BaggageType>
    <LuggageAllowanceType>limited</LuggageAllowanceType>
    <MaximumNumberItems>3</MaximumNumberItems>
    <MaximumBagHeight>2</MaximumBagHeight>
    <MaximumBagWidth>1</MaximumBagWidth>
    <MaximumBagDepth>1</MaximumBagDepth>
    <TotalWeight>30</TotalWeight>
</LuggageAllowance>
```

7.6.1.3.6 Usage Parameter: Booking – XML examples

7.6.1.3.6.1 Usage Parameter: Booking: XML Example of Purchase Window parameter

The following code fragment shows PURCHASE WINDOW usage parameters for several different minimum and maximum periods. Intervals can be specified as simple durations or be related to a TIME INTERVAL.

For EXAMPLE:

```
<PurchaseWindow version="any" id="tfl:time_of_travel">
    <Name>Purchase from Ticket Machine or Validator on board at time of Travel</Name>
    <PurchaseWhen>timeOfTravelOnly</PurchaseWhen>
    <MaximumPeriodBeforeDeparture>P0D</MaximumPeriodBeforeDeparture>
</PurchaseWindow>

<PurchaseWindow version="any" id="tfl:same_day">
    <Name>Purchase from Office</Name>
    <Description>At ticket machines7 Day Travel cards and Group Day tickets must be bought on
the start date</Description>
    <PurchaseWhen>dayOfTravelOnly</PurchaseWhen>
    <MaximumPeriodBeforeDeparture>P0D</MaximumPeriodBeforeDeparture>
</PurchaseWindow>

<PurchaseWindow version="any" id="tfl:up_to_4_days_in_advance">
    <Name>Purchase from Ticket Machine</Name>
    <Description>At ticket offices Day Travel cards and Group Day tickets can be bought up to
seven days in advance of the start date</Description>
    <MaximumPeriodBeforeDeparture>P4D</MaximumPeriodBeforeDeparture>
</PurchaseWindow>

<PurchaseWindow version="any" id="tfl:18Plusup_to_4_weeks_in_advance">
    <Name>Apply for 18 Plus</Name>
    <Description>Once you've fully enrolled with your education establishment, you can apply
up to four weeks before your course start date, or on your 18th birthday. Applications received
sooner than then this will be rejected.</Description>
    <MaximumPeriodItervalRef>P28D</MaximumPeriodBeforeDeparture>
</PurchaseWindow>
```

```
    <PurchaseWindow version="any" id="nre::at_least_two_months">
        <Name>Purchase supersavers</Name>
        <Description>Must be bought at least two months ahead and up to six months
ahead</Description>
        <MinimumPeriodIntervalRef version="any" ref="nre:3Months"/>
        <MaximumPeriodIntervalRef version="any" ref="nre:6Months"/>
    </PurchaseWindow>
```

7.6.1.3.6.2 Usage Parameter: Booking: XML Example of Transferability parameter

The following code fragment shows a TRANSFERABILITY usage parameter specify that transfers are allowed but required payment of a fee.

For EXAMPLE:

```
    <Transferability version="any" id="tfl:can_transfer">
        <CanTransfer>true</CanTransfer>
        <HasTransferFee>true</HasTransferFee>
    </Transferability>
```

7.6.1.3.6.3 Usage Parameter: Booking: XML Example of Refund parameter

The following code fragment shows a REFUND usage parameter allowing a 50% refund before the start of ticket validity until two hours before travel. The refunds are only available at certain ticket offices (which could be specified by a DISTRIBUTION ASSIGNMENT parameter for the SALES OFFER PACKAGE) and there is a NOTICE ASSIGNMENT to an associated NOTICE element (not shown) with refund conditions.

For EXAMPLE:

```
    <Refunding version="any" id="thx:remboursement_2H_50_percent">
        <Description lang="fr">PremsOui jusqu_à H (2) du voyage aller</Description>
        <noticeAssignments>
            <NoticeAssignment id="thx:remboursement_2H_50percent@ExchangeRenfe@01" version="01">
                <NoticeRef ref="thx:Notice:Product@ExchangeRenfe@01" version="01"/>
                <Mark>(1)</Mark>
                <Advertised>true</Advertised>
            </NoticeAssignment>
        </noticeAssignments>
        <prices>
            <UsageParameterPrice version="any" id="thx:exchange_2H_50percent">
                <DiscountAsPercentage>50</DiscountAsPercentage>
            </UsageParameterPrice>
        </prices>
        <Allowed>partial</Allowed>
        <OnlyAtCertainDistributionPoints>true</OnlyAtCertainDistributionPoints>
        <ResellWhen>beforeStartOfValidity</ResellWhen>
        <ExchangeableUntilDuration>PT2H</ExchangeableUntilDuration>
    </Refunding>
```

7.6.1.3.6.4 Usage Parameter: Booking: XML Example of Reserving parameter

The following code fragment shows various RESERVING usage parameters.

For EXAMPLE:

```
    <Reserving version="any" id="thx:must_reserve">
        <Name lang="fr">La réservation rest e obligatoire</Name>
        <ReservingRequirements>reservationsCompulsory</ReservingRequirements>
    </Reserving>

    <Reserving version="any" id="tap:group_required">
        <Name>Group booking reservation requirements</Name>
        <ReservingRequirements>reservationsPossible</ReservingRequirements>
        <MinimumNumberToReserve>1</MinimumNumberToReserve>
        <MaximumNumberToReserve>5</MaximumNumberToReserve>
        <ReservationChargeType>singleFeeForReturnTrip</ReservationChargeType>
        <FeeBasis>perPerson</FeeBasis>
        <HasFreeConnectingReservations>true</HasFreeConnectingReservations>
```

```
        <NumberOfFreeConnectingReservations>0</NumberOfFreeConnectingReservations>
        <BookingArrangements>
            <BookingMethods>other</BookingMethods>
            <BookingUrl>www.bahn.de</BookingUrl>
        </BookingArrangements>
    </Reserving>

    <Reserving version="any" id="thx:must_reserve_whole_compartment">
        <ReservingRequirements>reservationsCompulsory</ReservingRequirements>
        <MinimumNumberToReserve>4</MinimumNumberToReserve>
        <MaximumNumberToReserve>4</MaximumNumberToReserve>
        <MustReserveWholeCompartment>true</MustReserveWholeCompartment>
    </Reserving>
```

7.6.1.3.6.5 Usage Parameter: Charging: XML Example of ChargingPolicy parameter

The following code fragment shows CHARGING POLICY usage parameters to specify that a passenger may no travel on a card if.

For EXAMPLE:

```
<!-- CHARGING POLICY -->
    <ChargingPolicy version="any" id="tfl:PAYG">
        <Name>Allowed credit limits and Policy if exceeded </Name>
        <prices>
            <UsageParameterPrice version="any" id="tfl:give_credit">
                <Name>Maximum price allowed</Name>
                <LimitingRuleRef version="any" ref="tfl:max-8.50"/>
            </UsageParameterPrice>
        </prices>
        <CreditPolicy>blockAllTravel</CreditPolicy>
    </ChargingPolicy>
```

The following code fragment shows a CHARGING POLICY used to specify that a rebate of 20% is to be provided (It could be used along with a CAPPING RULE and a USAGE VALIDITY PERIOD.

For EXAMPLE:

```
    <ChargingPolicy version="any" id="tfl:Cableway5+">
        <Name>Restrospective discount if more than five trips - collect </Name>
        <UsageParameterTypeRef version="any" ref="tfl:rebate"/>
<prices>
            <UsageParameterPrice version="any" id="tfl:UsageParameterPrice@ChargingPolicy@Refund">
                <Name>Maximum price allowed</Name>
                <DiscountingRuleRef version="any" ref="tfl:pct20%"/>
            </UsageParameterPrice>
        </prices>

        <ExpireAfterPeriod>P42D</ExpireAfterPeriod>
    </ChargingPolicy>
```

7.6.1.3.6.6 Usage Parameter: Charging: XML Example of PenaltyPolicy parameter

The following code fragment shows PENALTY POLICY usage parameters to specify that if a passenger fails to check out at the end of a Pay As Of You Go journey they must pay the maximum fare of £8.50.

For EXAMPLE:

```
    <!-- PENALTY POLICY -->
    <PenaltyPolicy version="any" id="tfl:PAYG">
        <Name>Pay as you go capping period is one day </Name>
        <prices>
            <UsageParameterPrice version="any" id="tfl:no_check_out">
                <Name>Maximum price to charge for a journey if dont check out</Name>
                <LimitingRuleRef version="any" ref="tfl:max-8.50"/>
            </UsageParameterPrice>
        </prices>
        <SameStationRentryPolicy>maximumFare</SameStationRentryPolicy>
        <MinimumTimeBeforeReentry>PT60M</MinimumTimeBeforeReentry>
    </PenaltyPolicy>
```

7.6.2 Fare Product

7.6.2.1 FARE PRODUCT – Conceptual MODEL

The FARE PRODUCT MODEL describes the fare products available, that is a named set of features (access rights, discount rights etc), specific to a CHARGING MOMENT.

A FARE PRODUCT is an immaterial marketable element made available to the public. It can be purchased and enables the owner to consume public transport or other services at specific conditions. It may consist of specified access rights (PRE-ASSIGNED FARE PRODUCT) or other products (discounts, amount of price unit, etc.).

A FARE PRODUCT is immaterial, that is, it is independent of any physical representation but can be materialised on various TRAVEL DOCUMENTs. For instance, a “monthly pass” FARE PRODUCT may be variously incorporated on a specific paper ticket or stored on an electronic card.

A FARE PRODUCT is specific to a particular CHARGING MOMENT, which is a combination of:

- payment method (pre-payment or post-payment);
- account location (account stored on a TRAVEL DOCUMENT or in a central account).

The fact that FARE PRODUCTs are distinguished according to the CHARGING MOMENT shows the intrinsic characteristic of a FARE PRODUCT; they are access rights as advertised and presented to the public. The same access rights when presented to the public (i.e. when they become FARE PRODUCTs) may differ, for instance, the “access right to the metro network” may be advertised as two products: one as prepaid (materialised as a simple ticket), another as post-paid (materialised on an electronic card).

The classical examples of CHARGING MOMENT are the following:

- pre-payment with cancellation (throw-away tickets);
- pre-payment with debit on a TRAVEL DOCUMENT (value card);
- pre-payment without registration of the consumption (unlimited pass);
- post-payment (electronic card with central account and monthly debiting);
- free of charge.

These main categories may be subdivided according to the operator specific requirements.

The same FARE PRODUCT can be used in one or more SALES OFFER PACKAGEs (see later) to described a marketable product that the user can actually buy materialised onto a TYPE OF TRAVEL DOCUMENT, for example a metro trip might be available as both a paper ticket and as a smartcard transaction.

The CHARGING MOMENT – i.e. the point at which the passenger pays for the product is normally fundamental to the choice of products. For example, prepaid, post-paid, etc.

A given FARE PRODUCT (and subsequent SALES OFFER PACKAGE) may comprise a number of different values for each feature of the fare structure. For example, a FARE PRODUCT for a set of point-to-point journeys (each represented by a DISTANCE MATRIX ELEMENT) might include parameters for *first class, second class, single* and *return* use (i.e. combinations of ROUND TRIP and CLASS OF USE usage parameters); each allowed for different USER PROFILEs such as *adult, child, senior* and *student – and* every separate combination having a separate price. Thus, there is not normally a separate FARE PRODUCT for each combination of features that a user may buy and it is possible to represent a large set of offerings by a single FARE PRODUCT – as in the case of TAP TSI NRT (standard unreserved) fares.

The user's actual purchase will be described by a TRAVEL SPECIFICATION (see later below) which indicates which specific features have been selected, for example *an adult single second class ticket between Lille and Valenciennes*.

The FARE PRODUCT is itself an abstract concept – there are a number of concrete specializations.

The most classical FARE PRODUCTs are combinations of specified access rights (single ticket, commuter week ticket, monthly pass, etc.). Such a PRE-ASSIGNED FARE PRODUCT is defined as a FARE PRODUCT consisting of one or several VALIDABLE ELEMENTs.

Typical examples of PRE-ASSIGNED FARE PRODUCTs are the following:

— any VALIDABLE ELEMENT that is directly marketable, e.g. access right granted by a single ticket, access right granted by a park and ride discount ticket, etc. In such a case, the PRE-ASSIGNED FARE PRODUCT is identical to the VALIDABLE ELEMENT;

— a week card allowing one or two specified trips for each day of the week, each trip being defined as a VALIDABLE ELEMENT that should be consumed in one go during a specified time band of the considered day;

— a monthly pass allowing the unlimited consumption of several specified trips, each being defined as a VALIDABLE ELEMENT.

The four main types of FARE PRODUCTs are the following:

— PRE-ASSIGNED FARE PRODUCT is a marketable combination of specified VALIDABLE ELEMENTs. It is the most common FARE PRODUCT in public transport (materialised e.g. as single ticket, monthly pass etc.);

— AMOUNT OF PRICE UNIT is a FARE PRODUCT expressed by a specified number of PRICE UNITs (currency unit, token, etc.). It is not pre-assigned, which means that it gives the right to consume any VALIDABLE ELEMENT from a specified list. The main types of AMOUNT OF PRICE UNIT are value cards or electronic purses, which are debited for each transaction. In some cases, single tickets should be considered as AMOUNT OF PRICE unit, when it is required to punch a variable number of tickets according to the length of the intended trip;

— SALE DISCOUNT RIGHT is a FARE PRODUCT allowing its holder to benefit from discounts when purchasing specific SALES OFFER PACKAGEs. Train companies for instance usually propose such discounts (e.g. 30 % discount card);

— USAGE DISCOUNT RIGHT is a FARE PRODUCT allowing its holder to benefit from discounts when consuming specified VALIDABLE ELEMENTs. For instance, such a product grants to its holder a discount when consuming park and ride sequences, whereas parking or PT rides consumed alone are charged at the normal fare. This kind of discount is particularly meaningful with post-payment methods.

Two further types of FARE PRODUCTs also exist:

— CAPPED DISCOUNT RIGHT a refinement of a SALE DISCOUNT RIGHT used for advanced electronic pay as you go fares, where once a certain amount of consumption has been achieved within a certain interval, a cap (as specified by one or more CAPPING RULES) is applied, for example limiting the daily use to no more than the cost of a day pass

— SUPPLEMENT PRODUCT: An ancillary product, such as a seat class upgrade or a meal, that can only be purchased in addition to another product.

In addition, two other types of non-travel “product”, can be declared and referenced

— an ENTITLEMENT PRODUCT: may also be used to represent non-transport related qualifications such as disability cards, military cards or pensioner passes that are pre-requisites for the purchase or consumption of travel products.

— a THIRD PARTY PRODUCT: A FARE PRODUCT that is marketed together with a Public Transport FARE PRODUCT. It is a product not fully described by the system.

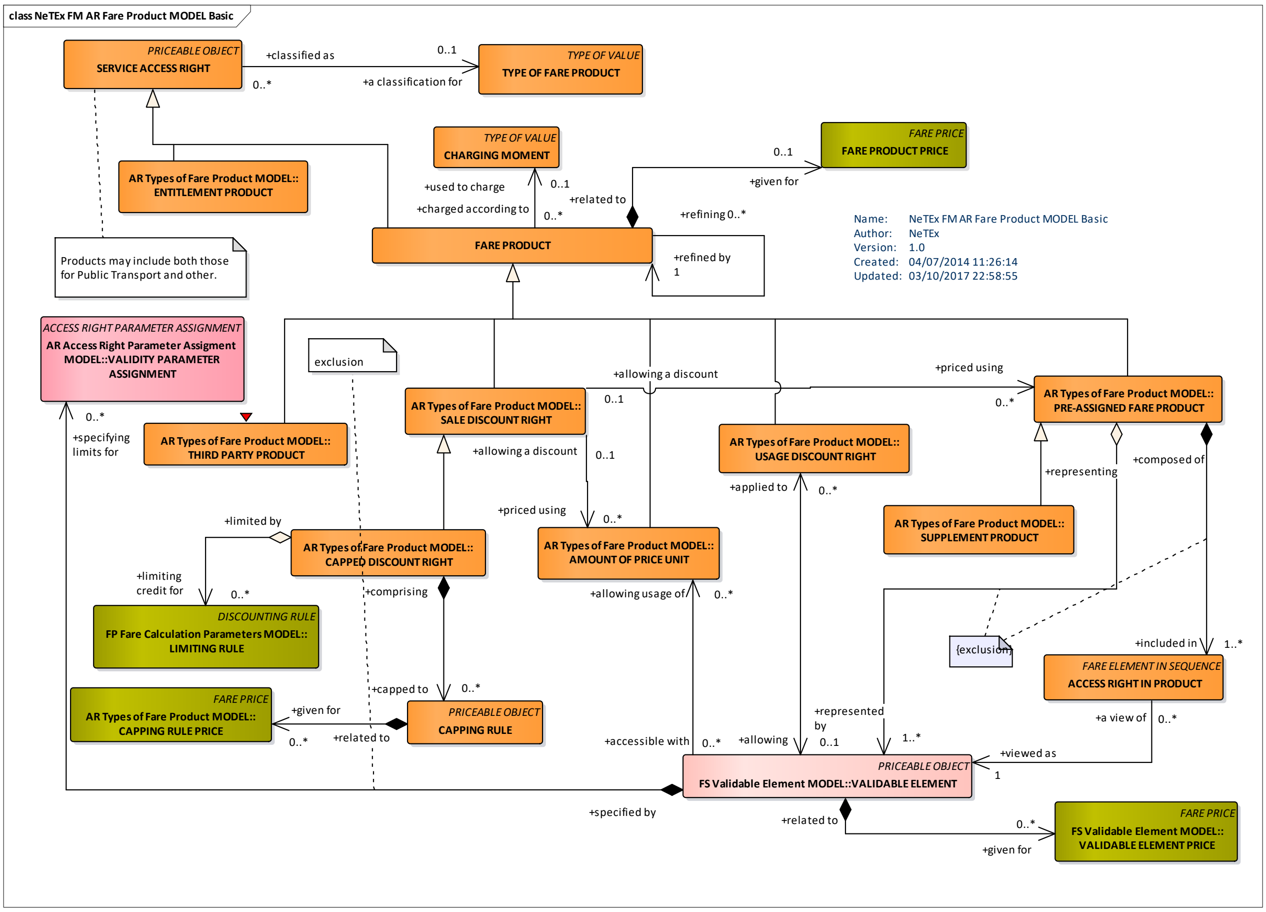


Figure 226 — Fare Product – Basic Conceptual MODEL (UML)

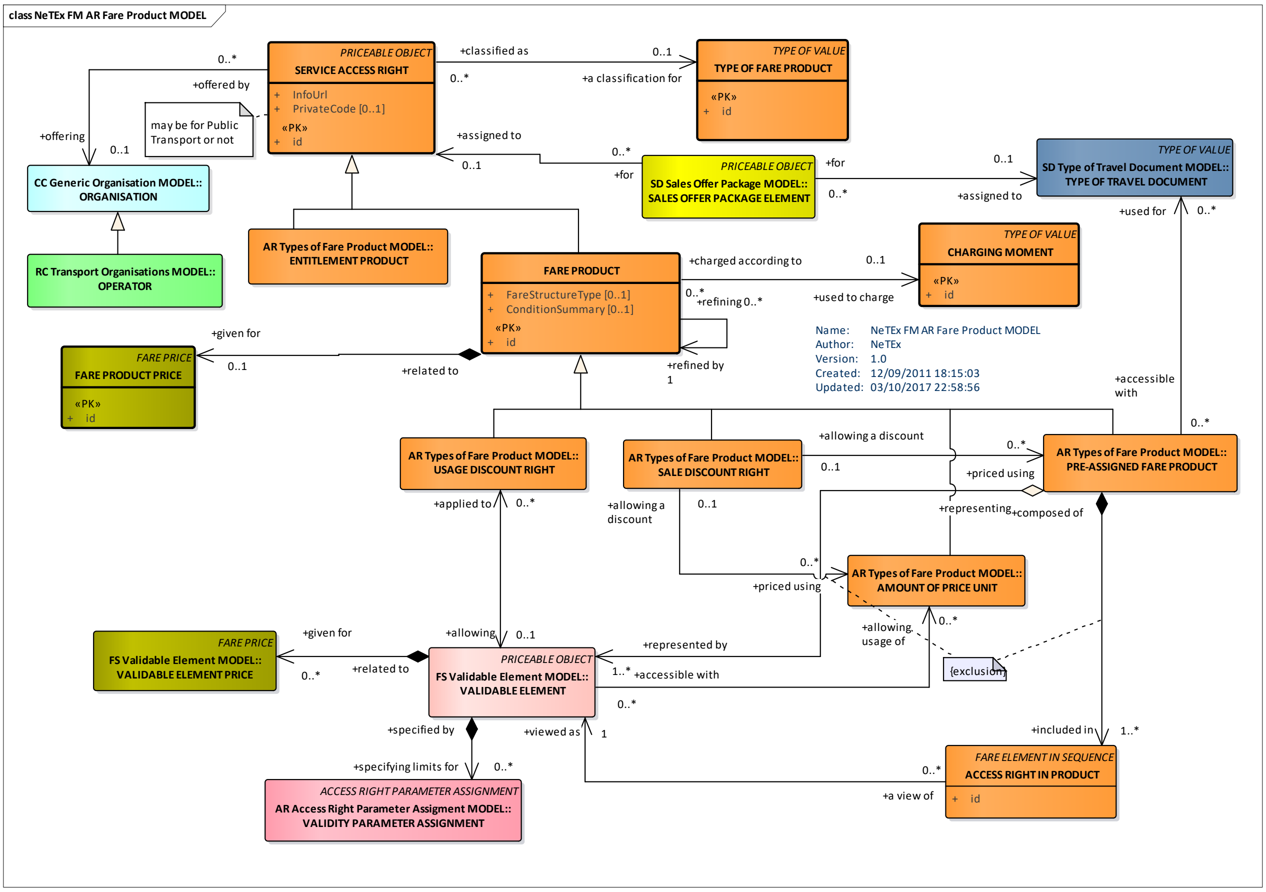


Figure 227 — Fare Product – Conceptual MODEL (UML)

7.6.2.1.1 Fare Product Example – Rail products with condition

The following figure shows a set of rail products with complex purchase and usage conditions and varying commercial conditions as to refund and exchange.

Gamme de prix C

ELIPSOS Trainhôtel

Juin 2012

Cette fiche annule et remplace la Gamme de prix C 2011/2012

Nom du produit	Classe de service	Type de passager	Conditions	Échange (1)	Remboursement	
					Avant H	Après H
Adulte	S, T, L, N, R, A	PT00AD	Sans condition	Oui jusqu'à H (2)	90%	0%
Enfant	S, T, L, N, R, A	PT0012	Enfants de 4 à moins de 12 ans le jour du voyage			
Loisir	S, T, L, N, R, A	ND00AD	Aller/Retour obligatoire	Oui jusqu'à H (2) du voyage aller	90%	0%
Prem's	R	PP05AD	Achat jusqu'à J-14. Places limitées.	Non	0%	0%
Mini à deux	T, N	PP05AD	Achat jusqu'à J-14. Places limitées. 2 personnes voyageant ensemble		50%	0%
Duo	T, N, P, A	PE07AD	2 personnes voyageant ensemble (3). Places limitées. Compartiment « Famille » sauf en classe A	Oui jusqu'à H (2)	90%	0%
Espace Plus (4)	R	PE08AD	3 ou 4 personnes voyageant ensemble. Compartiment « Famille »			
Jeune (5)	S, T, L, N, R, A	PT0026	De 12 à moins de 26 ans le jour du voyage			
Senior*	S, T, L, N, R, A	PT0060	Avoir au moins 60 ans le jour du voyage			
Congrès – Salons	S, T, L, N, R, A	CN20AD	Sur présentation d'un justificatif à dates spécifiques (mail de confirmation ou billet d'entrée)	Oui jusqu'à H (2) du voyage aller	90%	0%
Pass (6) (7)	S, T, L, N, R, A	EP00AD	Titulaire d'un Pass couvrant au moins un des pays du parcours	Oui jusqu'à H (2)	90%	0%
Groupe Adulte (TS)	R, N, T	GR00AD	Minimum 12 adultes. Demande effectuée par le client au 0810 879 479 (Agence Commerciale Voyageurs)	Jusqu'à J-30 avec 20% de retenue De J-29 à J-8 avec 30% de retenue A partir de J-7 avec 100% de retenue	80% jusqu'à J-30 70% de J-29 à J-7 0% à partir de J-7	0%
Groupe Adulte (AR)		GR02AD				
Guide d'handicapé ou d'aveugle*	S, T, L, N, R, A	GG99AD	L'handicapé ou l'aveugle doivent être titulaire d'une carte d'invalidité. Pour un handicapé en fauteuil roulant, celui-ci doit être pliable.	Oui jusqu'à H (2)	90%	0%
FIP Loisir (7)	S, T, L, N, R, A	EM01AD	Carte FIP	Oui jusqu'à H (2)	90 %	0%
FIP Service (7)	N,R,S,L,T	CT02AD	Sur présentation de la Carte FIP		90%	0%
Enfant partageant un lit (8)	B	PE0112	Cabine entièrement réservée. Compartiment « Famille »	Non	90%	90%
Animaux domestiques (9)	B	CH50CH	Cabine entièrement réservée (10). Compartiment « famille »			

(1) L'échange des billets émis par "RENFE" se fait uniquement en Gare de Paris Austerlitz, Paris Montparnasse, Paris Gare du Nord, Paris St Lazare (NEV Amsterdam), Paris Gare de Lyon, Poitiers, Blois, Les Aubrais - Orléans et Limoges.

(2) L'échange ou le remboursement s'effectue jusqu'à H (Heure de départ du train).

(3) En classe P, les enfants de moins de 4 ans et de 4 à 12 ans ne peuvent pas partager un lit. Néanmoins, une des deux personnes partageant la cabine peut être un enfant occupant un lit. Il est alors considéré comme la 2e personne pour bénéficier de ce tarif.

(4) Sur Mosaïque +, indiquez toujours 4 personnes même si la cabine est occupée par 3 voyageurs.

(5) Tarif proposé également aux titulaires de la carte « ISIC » (International Student Identity Card) sans condition d'âge, ainsi qu'aux porteurs de la carte « Carnet Jove » âgés de 14 à moins de 30 ans.

(6) Eurail Global Pass, Eurail Select Pass (incluant au moins un des pays concernés par le trajet du train), Eurail France - Italy Pass,Eurail France-Switzerland Pass, Eurail France-Spain Pass, Eurail Italy-Spain Pass, Eurail Spain Pass, France Rail Pass, Eurail Portugal - Spain Pass, Eurail Benelux - France Pass, Eurail France-Germany Pass, InterRail Global Pass, InterRail Spain Pass,InterRail France Pass.
Uniquement sur les OD du Trainhôtel Pau Casals (Barcelone - Zurich) : Swiss Pass, Eurail Austria - Switzerland Pass, Eurail Germany - Switzerland Pass, InterRail Swiss Pass et uniquement sur les OD du Trainhôtel Salvador Dali (Barcelone - Milan) : InterRail Italy Pass, Eurail Italy Pass, Eurail Greece - Italy Pass.

(7) Le client a accès à toutes les classes de service indépendamment de son type de Pass ou sa carte FIP (1re/2e).
Pour le FIP Service : Carte FIP + ordre de mission à présenter au point de vente habilité.

(8) Pour un enfant de 4 à moins de 12 ans, émettre un billet ouvert en 2e classe en précisant le point frontière (Port Bou frontière sur les relations Paris/Barcelone, Barcelone/Milan ou Barcelone/Zurich et Hendaye frontière sur la relation Paris/Madrid). L'enfant de moins de 4 ans voyage gratuitement et partage le lit de la personne qui l'accompagne à condition que la cabine soit entièrement réservée.

(9) Les animaux domestiques de petite taille doivent être transportés dans un contenant (taille maximale 60x35x35). Émettre un billet ouvert en 2e classe en précisant le point frontière*. Les chiens "guide d'aveugle ou d'assistance" sont gratuits (pas d'émission de billet).

(10) Condition non exigée pour les chiens "guide d'aveugle ou d'assistance". En cas d'opposition des autres voyageurs à la présence du chien, le personnel essaiera de trouver, dans la mesure du possible, une solution de reclassement.

* Pour effectuer une vente en cabine double Grande Classe au tarif Senior associé au tarif Guide Handicapé, le dossier doit être constitué sur **Mosaïque classique** (Pavé Accueil > Informations).

Figure 228 **— Fare Table – Example Rail Products and Fare Conditions**

7.6.2.2 Fare Product – Physical model

The following figure shows the common physical model for FARE PRODUCTs.

All FARE PRODUCTs are specialisations of the SERVICE ACCESS RIGHT element. FARE PRODUCTs have a CHARGING MOMENT that indicates the moment the transaction is performed – this may be before, at the start or after travel. The conditions of purchase can be summarised using a CONDITION SUMMARY. An ENTITLEMENT PRODUCT represents products that are not for actual travel but which are required in order to purchase or use a FARE PRODUCT.

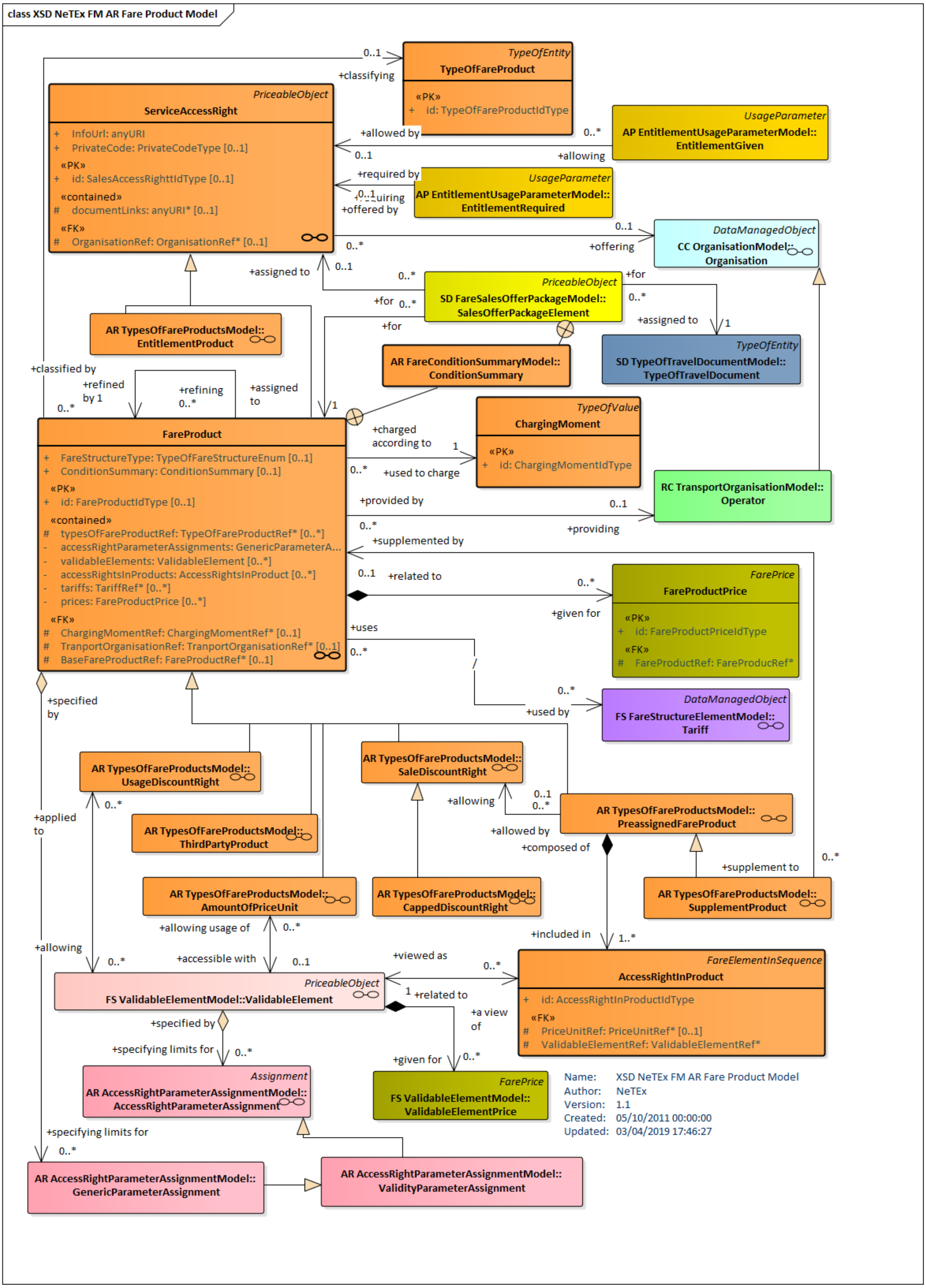


Figure 229 — Fare Product – Physical Model (UML)

7.6.2.2.1 Types of Fare Product – Physical model

FARE PRODUCT is abstract - the following figure shows the physical model for the different concrete types of FARE PRODUCT that are available.

- PREASSIGNED FARE PRODUCT – a classical fare charged before the trip and giving the right to access public transport.
- AMOUNT OF PRICE UNIT – a fare product made up of a number of units that can be consumed such as a carnet or strip of validable coupons.
- SALES DISCOUNT RIGHT – A right to buy further products at a discount.
- USAGE DISCOUNT RIGHT: A right to consume products at a discount.
- CAPPED DISCOUNT RIGHT a refinement of a SALE DISCOUNT RIGHT used for advanced electronic pay as you go fares- see below.
- SUPPLEMENT PRODUCT: An ancillary product, such as a seat class upgrade.

In addition, two other type of non-travel “product”, can be declared and referenced.

- an ENTITLEMENT PRODUCT: may be used to represent non-transport related qualifications such as disability cards, military cards or pensioner passes.
- a THIRD PARTY PRODUCT: A product from another organisation not fully described by the systems.

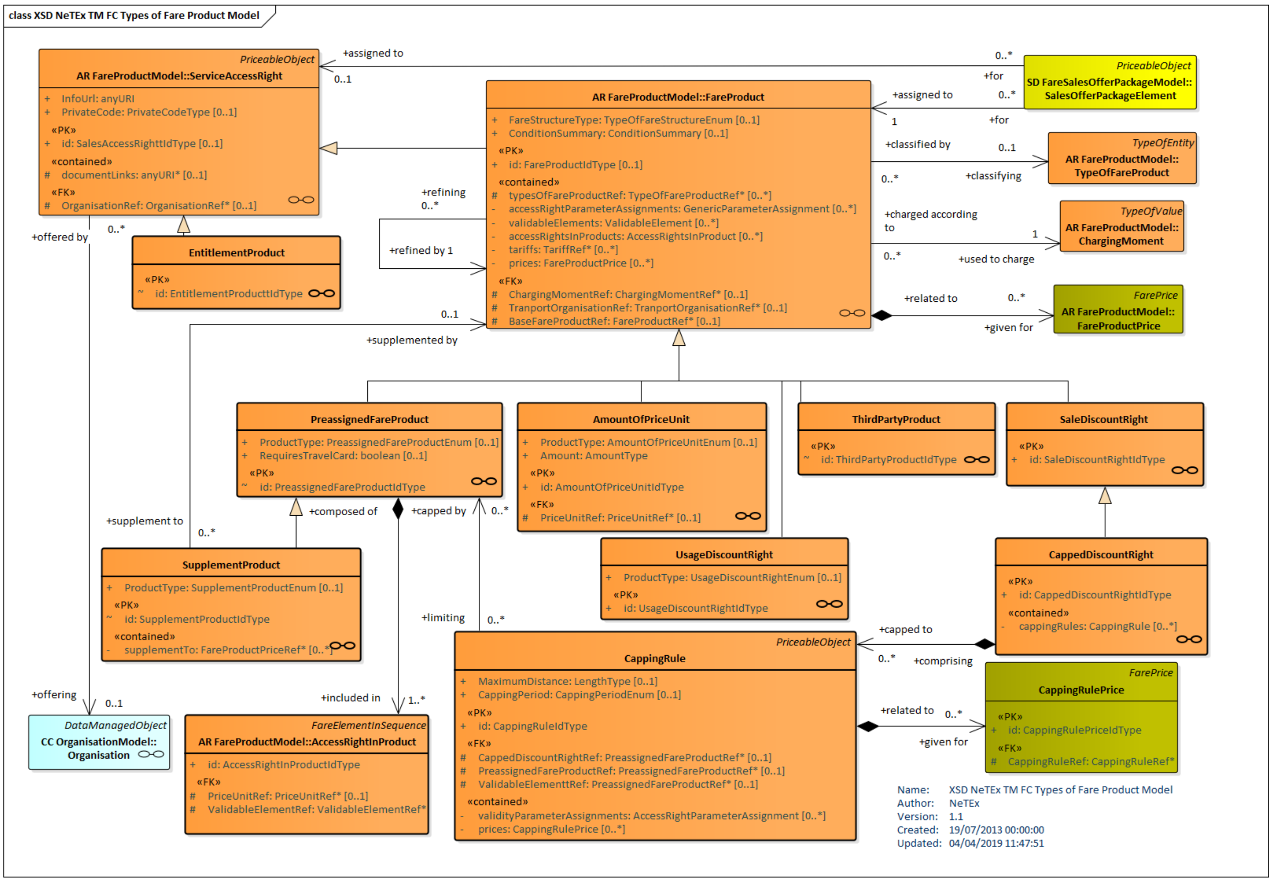


Figure 230 — Types of Fare Product – Physical Model (UML)

7.6.2.2.2 Capped Discount Product – Physical model

The following figure shows the physical model for the CAPPED DISCOUNT PRODUCT, which can be used for advanced electronic pay as you go fares, such that once a certain amount of consumption has been achieved within a certain interval, a cap (as specified by one or more CAPPING RULES) is applied, for example limiting the daily use to no more than the cost of a day pass.

With the same product There may be different caps for different VALIDABLE ELEMENTs, e.g. metro Trip, bus trip, river trip etc, each specified by a different CAPPING RULE.

In some cases, the start of the capping period is fixed (e.g. Monday, 1st of the moth etc) in other cases it is variable. This can be specified using attributes on a USAGE VALIDITY PERIOD parameter.

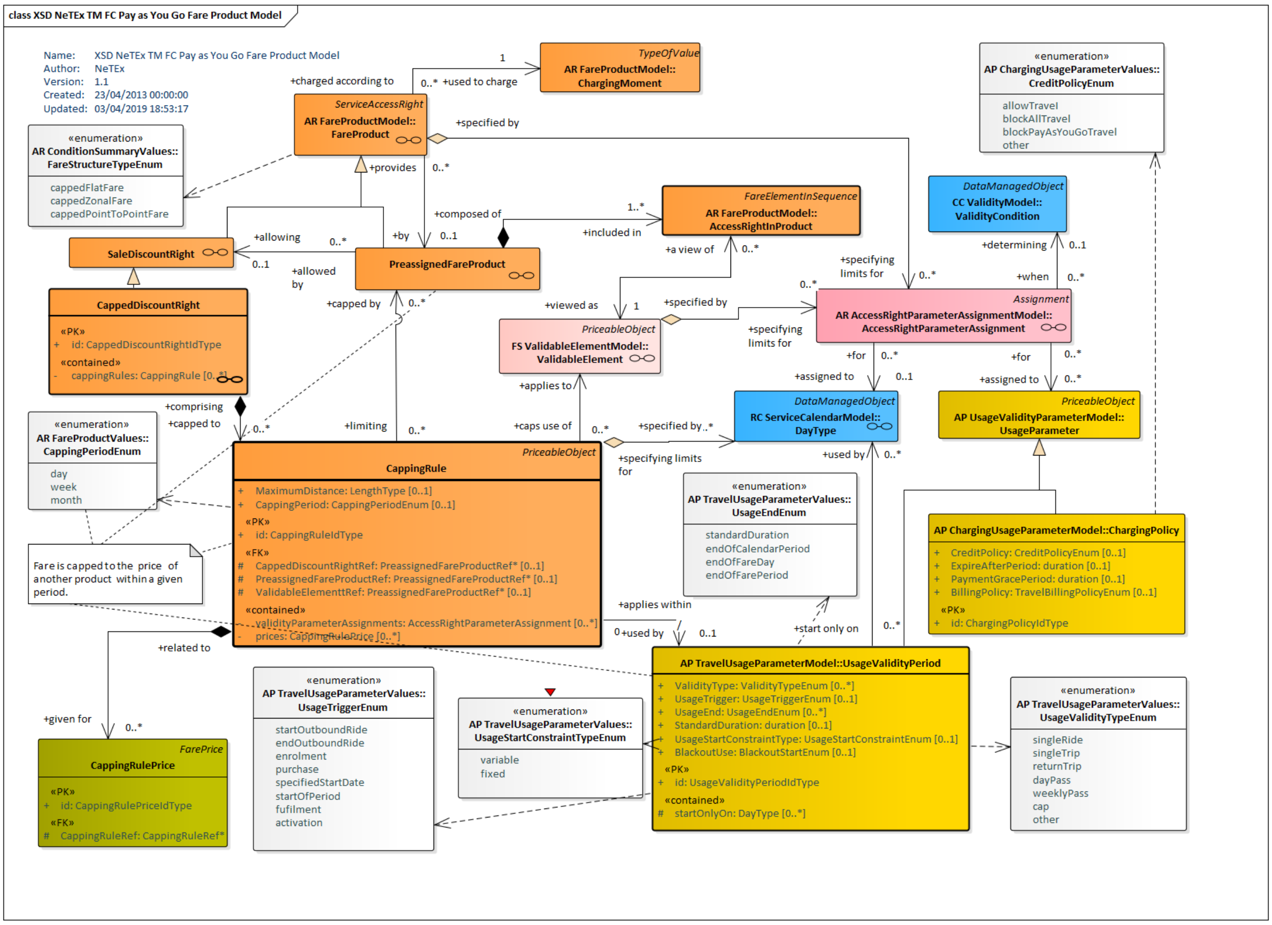


Figure 231 — Pay as You Go Fare Product – Physical Model (UML)

7.6.2.2.3 Fare Condition Summary – Physical model

A FARE CONDITION SUMMARY can be used to provide a high-level description of a product for product comparison and passenger information purposes. The summary typically indicates merely the existence of a condition - the actual conditions themselves are described more exactly by USAGE PARAMETERs, ACCESS RIGHT ASSIGNMENTs and other elements. The summary can include information about:

— Requirements concerning cards related to the product.

— Commercial conditions for refund, exchange, etc.

— Conditions limiting travel times, routes, etc.

— Conditions concerning entitlements.

— Conditions affecting reservation.

— The following figure shows the physical model for the FARE CONDITION SUMMARY.

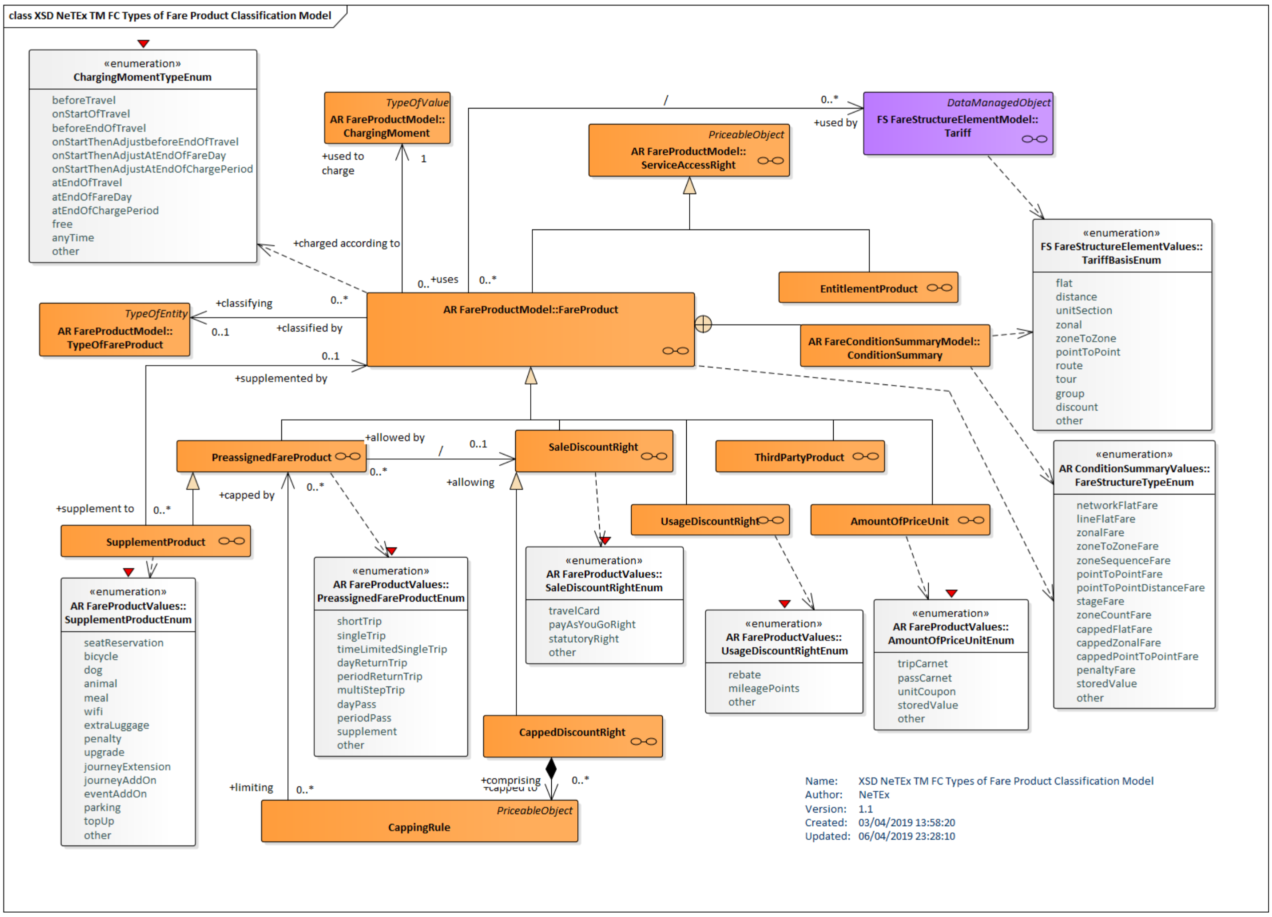


Figure 232 — Fare Condition Summary– Physical Model (UML)

7.6.2.2.4 Fare Product Classification – Physical model

In practice, only certain combinations of tariff and product type are found in common use, with some types being more common on specific modes. NeTEx allows arbitrary user defined classifications to be made using a TYPE OF FARE PRODUCT ELEMENT. It also provides a number of enumerated values for the most common product types.

The following figure shows the enumerated values for each specialisation of FARE PRODUCT.

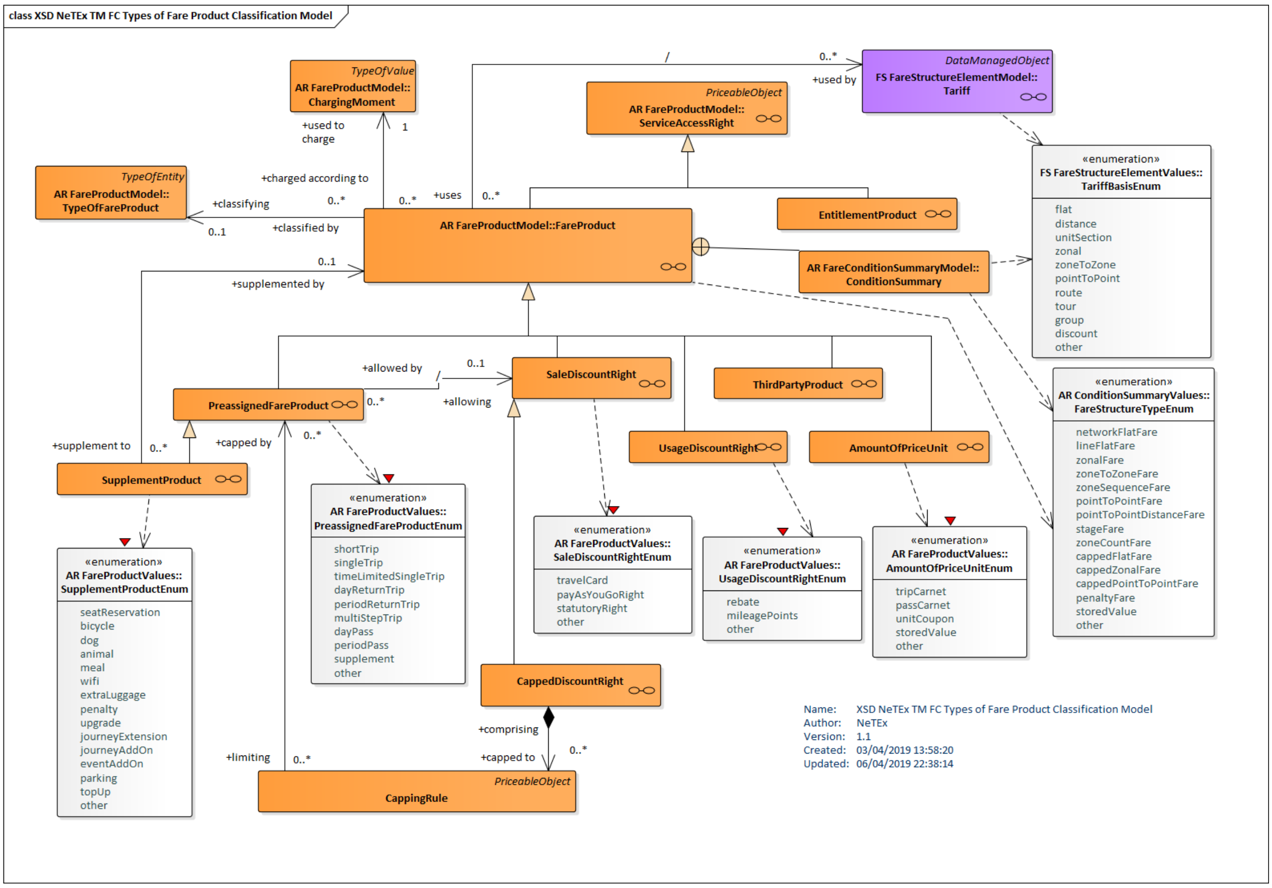


Figure 233 — Fare Product Classification– Physical Model (UML)

7.6.2.3 Fare Product – Attributes and XSD

7.6.2.3.1 ServiceAccessRight – Model Element

An immaterial marketable element (access rights, discount rights etc).

Table 216 – *ServiceAccessRight* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*PriceableObject*</u>	::>	SERVICE ACCESS RIGHT inherits from PRICEABLE OBJECT.
«PK»	***id***	*ServiceAccessRightIdType*	1:1	Identifier of SERVICE ACCESS RIGHT.
«AK»	***PrivateCode***	*PrivateCodeType*	0:1	Alternative identifier of an entity; can be used to associate with legacy systems.
	InfoUrl	*xsd:anyURI*	0:1	Link for product information.
«cntd»	***documentLinks***	<u>*InfoLink*</u>	0:*	InfoLinks for external links. For PFDs, etc =V1.1

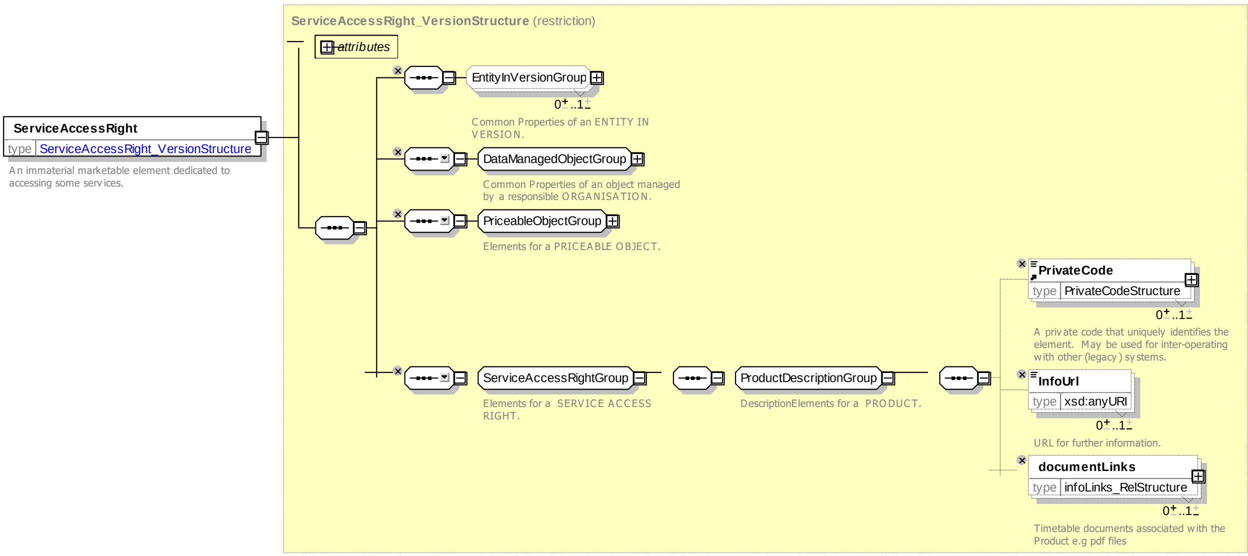


Figure 234 — *ServiceAccessRight* — XSD

7.6.2.3.2 FareProduct – Model Element

An immaterial marketable element (access rights, discount rights etc), specific to a CHARGING MOMENT.

Table 217 – *FareProduct* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*PriceableObject*	::>	FARE PRODUCT inherits from SERVICE ACCESS RIGHT.
«PK»	***id***	*FareProductIdType*	0:1	Identifier of FARE PRODUCT.
«FK»	***Charging-MomentRef***	*ChargingMomentRef*	0:1	Reference to a CHARGING MOMENT for product +v1.1.
«enum»	***Charging-MomentType***	*Charging-MomentTypeEnum*	0:1	Enumeration of standardised Charging moment values. See allowed values. _v1.1.
«FK»	***typesOfFare-ProductRef***	*TypeOfFareProductRef*	0:*	Classifications of FARE PRODUCT. (made *:* in v1.1)
«FK»	***Transport-OrganisationRef***	*(TransportOrganisationRef) OperatorRef | AuthrityRef*	0:1	OPERATOR or AUTHORITY in charge of the FARE PRODUCT.
«cntd»	***ConditionSummary***	*ConditionSummary*	0:1	Summary description of conditions on FARE PRODUCT.
XGRP	***FareProduct-RelationGroup***	***xmlGroup***	0:1	Elements relating to association of FARE PRODUCT.
XGRP	***FareProduct-ValidityGroup***	***xmlGroup***	0:1	Elements relating to validity of FARE PRODUCT.

XGRP	***FareProduct-PricingGroup***	***xmlGroup***	0:1	Elements relating to pricing of FARE PRODUCT.

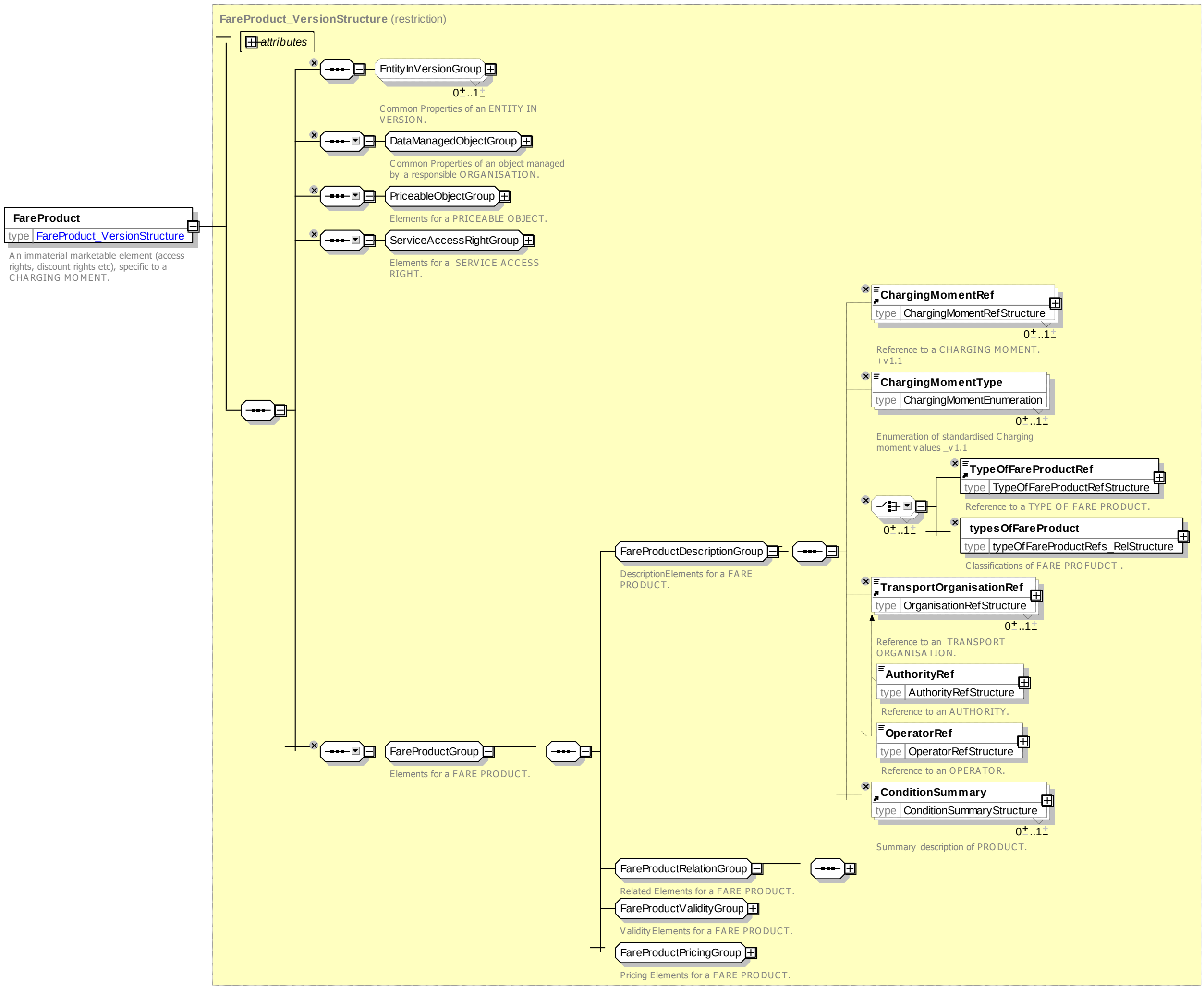


Figure 235 — *FareProduct* — XSD

7.6.2.3.2.1 ChargingMomentType – Allowed values

The following table shows the allowed values ***ChargingMomentType*** (*ChargingMoment Enumeration*).

Table 218 – ChargingMomentType – Allowed values

Value	Description
beforeTravel	Pay before travel.
onStartOfTravel	Pay on start of travel.
beforeEndOfTravel	Before end Of travel.
onStartThenAdjustAtEndOfTravel	Down pay on start then adjust at end of travel.
onStartThenAdjustAtEndOfFareDay	Down pay on start then adjust at end of fare day.

onStartThenAdjustAtEndOfChargePeriod	Down pay on start then adjust at end of charge Period.
atEndOfTravel	At end of travel.
atEndOfFareDay	At end of fare day.
atEndOfChargePeriod	At end of charge Period.
free	Free.
anyTime	At any time.
other	Other.

7.6.2.3.2.2 FareProductRelationGroup – Group

The ***FareProductRelationGroup*** specifies any base product of which this is a refinement. The refined product is assumed to have any of the properties of the base product which are not specifically overridden on the dependent product.

Table 219 – *FareProductRelationGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***BaseFare-ProductRef***	*FareProductRef+*	0:1	Another FARE PRODUCT which this product extends. Will assume all properties of base product unless specifically overridden.

Figure 236 — *FareProductRelationGroup* — XSD

7.6.2.3.2.3 *FareProductValidityGroup* – Group

The ***FareProductValidityGroup*** specifies attributes describing the validity of a FARE PRODUCT.

Table 220 – *FareProductValidityGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***validity-Parameter-Assignments***	*AccessRightParameter-Assignment*	0:*	VALIDITY PARAMETER ASSIGNMENTs relating to FARE PRODUCT.
«cntd»	***validable-Elements***	*ValidableElement*	0:*	VALIDABLE ELEMENTs for FAR PRODUCT.
«cntd»	***accessRights-InProduct***	*AccessRightInProduct*	0:*	ACCESS RIGHTs in PRODUCT for FAR PRODUCT.

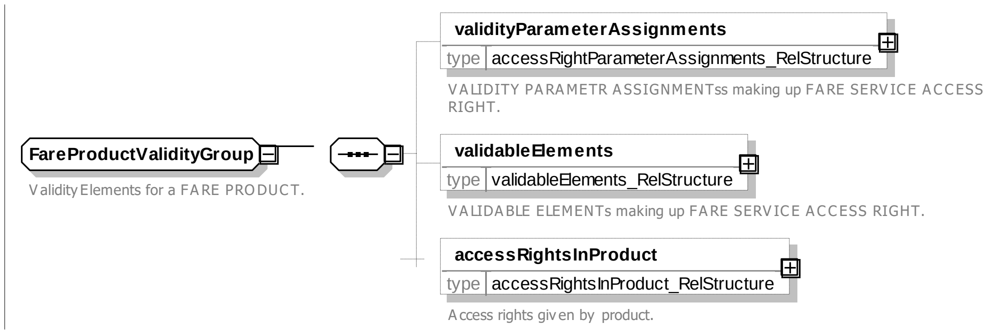


Figure 237 — *FareProductValidityGroup* — XSD

7.6.2.3.2.4 FareProductPricingGroup – Group

The ***FareProductPricingGroup*** specifies pricing properties of a FARE PRODUCT.

Table 221 – *FareProductPricingGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***tariffs***	*TariffRef*	0:*	TARIFFs used by FARE PRODUCT.
«cntd»	***prices***	*FareProductPrice*	0:*	FARE PRODUCT PRICEs in PRICE GROUP.

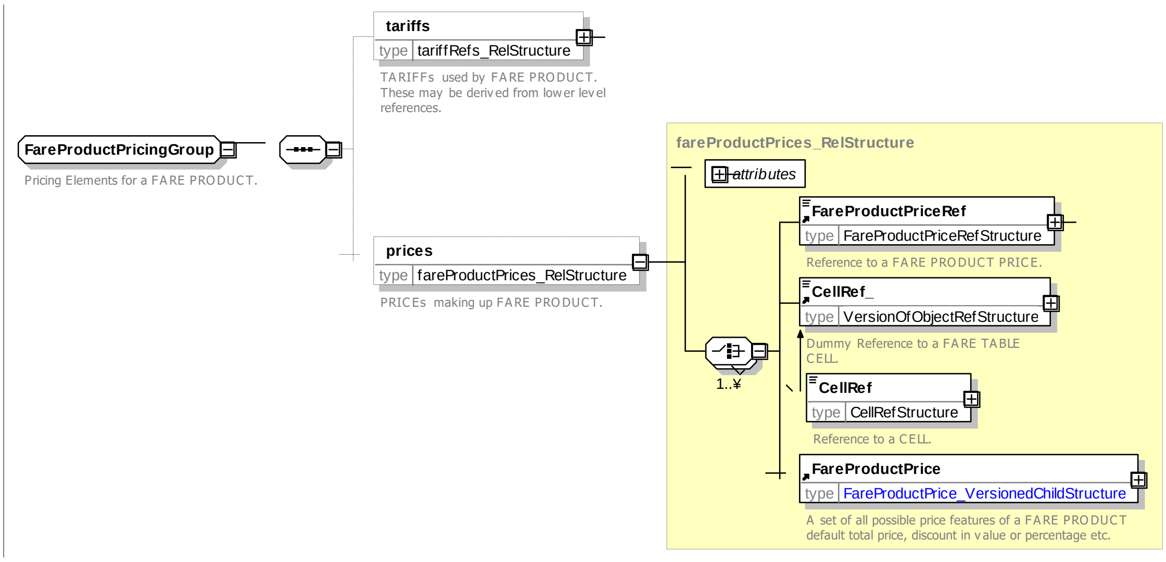


Figure 238 — *FareProductPricingGroup* — XSD

7.6.2.3.3 FareProductPrice – Model Element

A set of all possible price features of a FARE PRODUCT: default total price, discount in value or percentage etc.

Table 222 – *FareProductPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	FARE PRODUCT PRICE inherits from FARE PRICE
«PK»	***id***	*FareProductPriceIdType*	1:1	Identifier of FARE PRODUCT PRICE.
«FK»	***FareProductRef***	*FareProductRef+*	0:1	FARE PRODUCT for which this is the price.

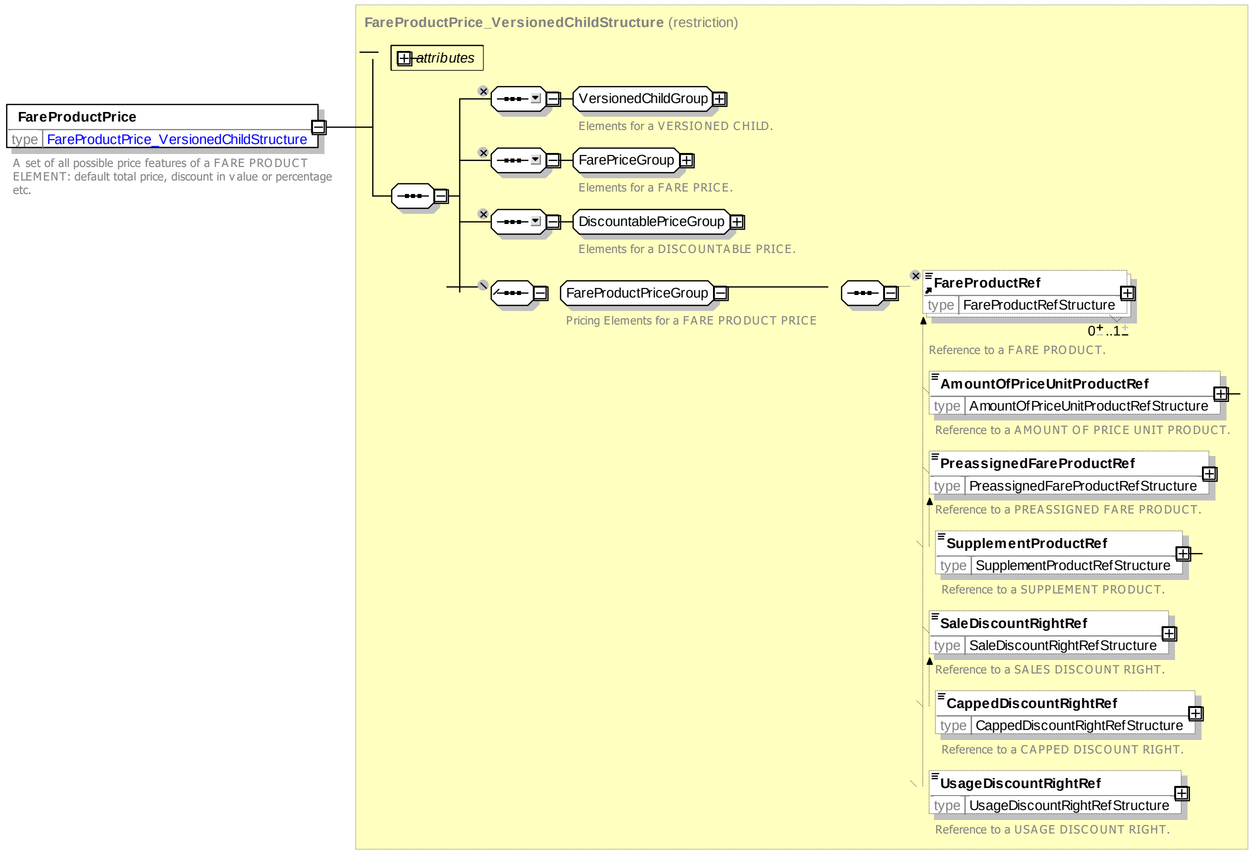


Figure 239 **— *FareProductPrice* — XSD**

7.6.2.3.4 AccessRightInProduct – Model Element

A VALIDABLE ELEMENT as a part of a PRE-ASSIGNED FARE PRODUCT, including its possible order in the set of all VALIDABLE ELEMENTs grouped together to define the access right assigned to that PRE-ASSIGNED FARE PRODUCT.

Table 223 – *AccessRightInProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareElementInSequence*	::>	ACCESS RIGHT IN PRODUCT inherits from FARE ELEMENT IN SEQUENCE.
«PK»	***id***	*AccessRightInProduct-IdType*	1:1	Identifier of ACCESS RIGHT IN PRODUCT.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	0:1	Reference to a VALIDABLE ELEMENT for which access rights are specified.
«FK»	***Preassigned-FareProducRef***	*PreassignedFareProducRef*	0:1	Reference to a PRE ASSIGNED FARE PRODUCT for which access rights are specified.

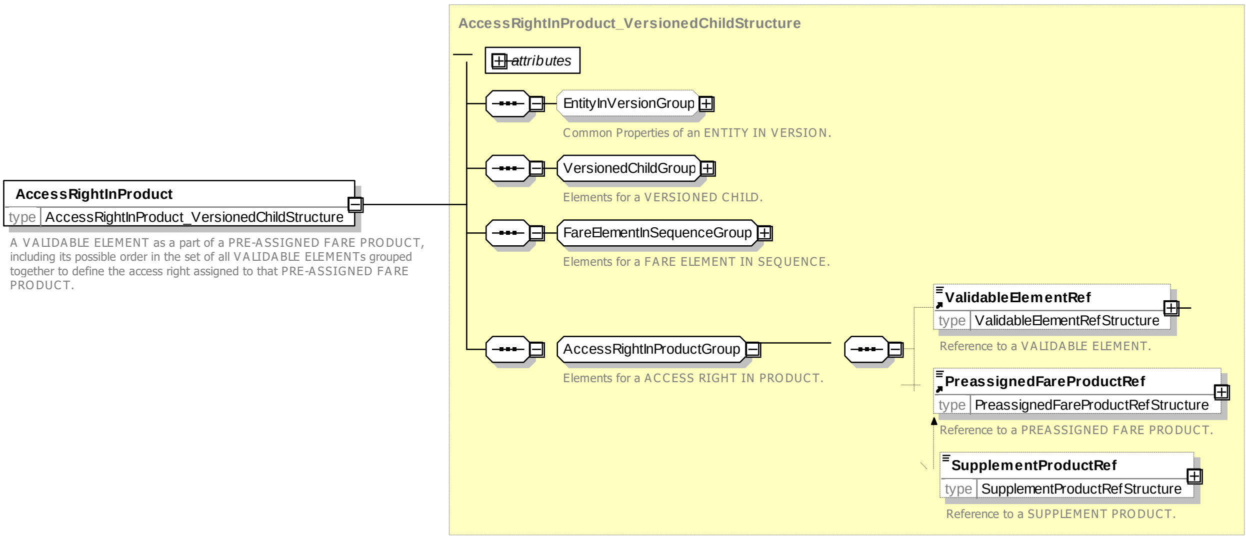


Figure 240 — *AccessRightInProduct* — XSD

7.6.2.3.5 ConditionSummary – Model Element

A summary of the properties of a FARE PRODUCT or PACKAGE that can be used to generate passenger information.

Table 224 – *ConditionSummary* – Element

Classifi-cation	Name	Type	Cardin ality	Description
«enum»	***FareStructureType***	*FareStructureTypeEnum*	1:1	Classification of fare type.
«enum»	***TariffBasis***	*TariffBasisEnum*	0:1	Basis for Tariff.
	HasNotices	*xsd:boolean*	0:1	Whether there are notices associated with the product.
XGRP	***Condition-SummaryCardGroup***	***xmlGroup***	1:1	Elements relating to cards on CONDITION SUMMARY.
XGRP	***ConditionSummary-EntitlementGroup***	***xmlGroup***	1:1	Elements relating to entitlement conditions on CONDITION SUMMARY.
XGRP	***ConditionSummary-TravelGroup***	***xmlGroup***	1:1	Elements relating to travel conditions on CONDITION SUMMARY.
XGRP	***ConditionSummary-CommercialGroup***	***xmlGroup***	1:1	Elements relating to commercial conditions on CONDITION SUMMARY.
XGRP	***ConditionSummary-ReservationGroup***	***xmlGroup***	1:1	Elements relating to reservation conditions on CONDITION SUMMARY.
XGRP	***ConditionSummary-ChargingGroup***	***xmlGroup***	1:1	Elements relating to charging conditions on CONDITION SUMMARY.

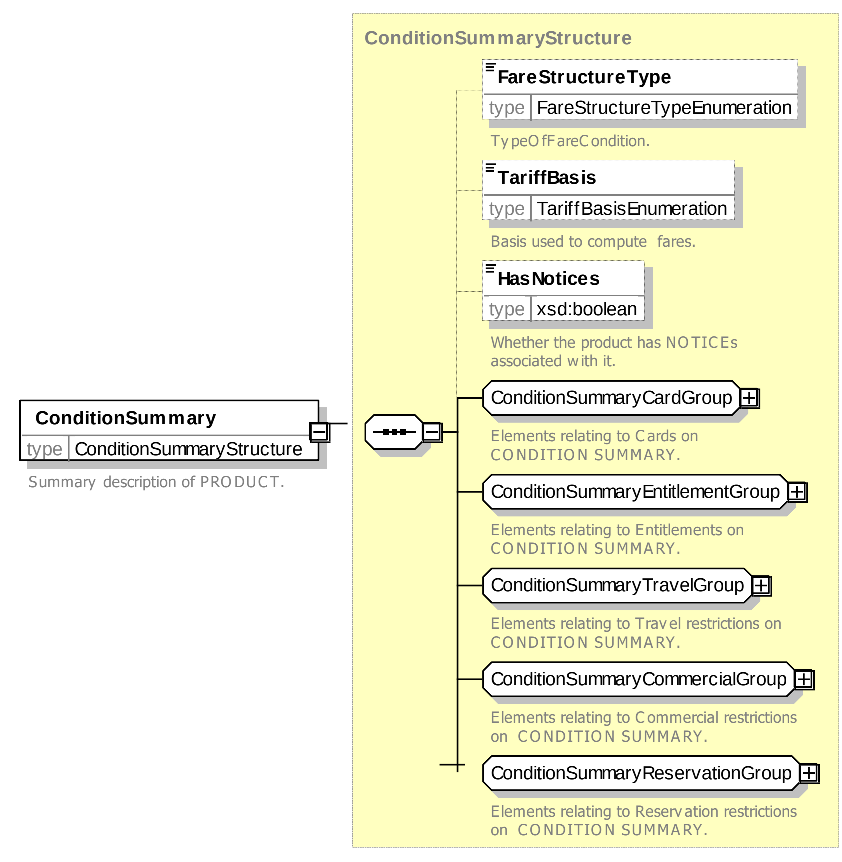


Figure 241 — *ConditionSummary* — XSD

7.6.2.3.5.1 FareStructureType – Allowed values

The following table shows the allowed values for Fare Structure Type. (*FareStuctureEnumeration*)

Table 225 – FareStructureType – Allowed values

Value	Description
networkFlatFare	Flat fare applying to whole network.
lineFlatFare	Flat fare applying to group of lines or line.
zonalFare	Zonal flat fare.
zoneToZoneFare	Zone to zone fare.
zoneSequenceFare	Zones in sequence fare.
pointToPointFare	Point to point fare.
stageFare	Fare stage fare.
cappedZonalFare	Capped zonal fare.
cappedFlatFare	Capped flat fare.
store	
other	Other fare.

7.6.2.3.5.2 TariffBasis – Allowed values

The following table shows the allowed values ***TariffBasis*** (*TariffBasisEnumeration*).

Table 226 – TariffBasis – Allowed values

Value	Description
flat	Tariff is flat, i.e. not based on spatial or period elements. +v1.1

distance	Tariff is based on distance.
route	Tariff is based on the route taken.
zoneToZone	Tariff is based on specified zone to zone transitions. +v1.1
pointToPoint	Tariff is based on specified point to point transitions. +v1.1
tour	Tariff is based on a specific tour.
group	Tariff is based on size of group.
discount	Tariff is for discount rates. +v1.1
other	Another Tariff Basis. +v1.1

7.6.2.3.5.3 ConditionSummaryCardGroup – Group

The ***ConditionSummaryCommercialGroup*** summarises the conditions on travel cards associated with a FARE PRODUCT and /or SALES OFFER PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs.

Table 227 – *ConditionSummaryCardGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	ProvidesCard	*xsd:boolean*	0:1	Whether a card is provided with the product.
	GoesOnCard	*xsd:boolean*	0:1	Whether the product goes on a card.
	IsPersonal	*xsd:boolean*	0:1	Whether the product is a sold anonymously or to an identified person.
	RequiresPhoto	*xsd:boolean*	0:1	Whether use of the product requires a photo to be provided.
	MustCarry	*xsd:boolean*	0:1	Whether use of the card must be carried in order to use product.
	RequiresAccount	*xsd:boolean*	0:1	Whether the product requires the user to register for an account for billing. +v1.1

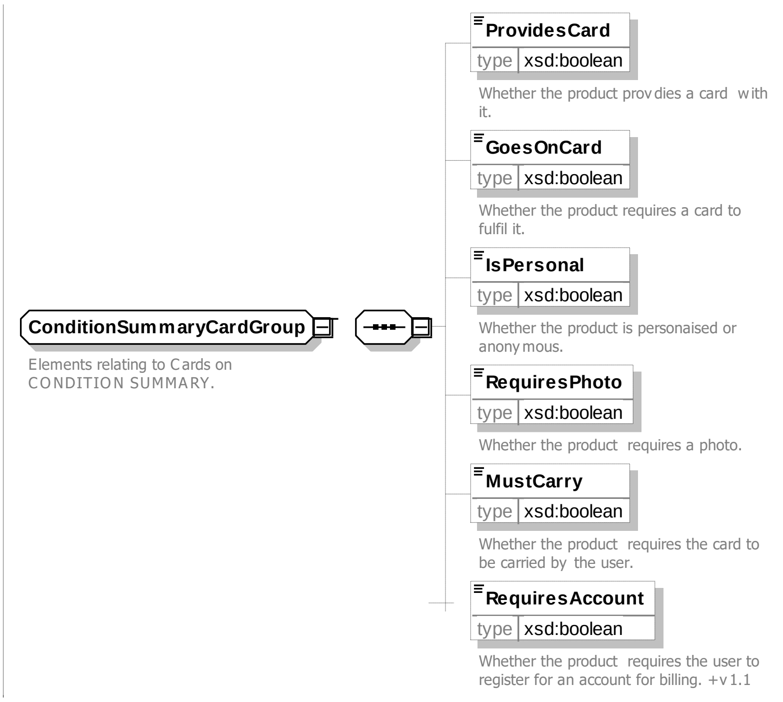


Figure 242 — *ConditionSummaryCardGroup* — XSD

7.6.2.3.5.4 ConditionSummaryEntitlementGroup – Group

The ***ConditionSummaryCommercialGroup*** summarises the entitlement conditions associated with a FARE PRODUCT and /or SALES OFFER PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs.

Table 228 – *ConditionSummaryEntitlementGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	IsSupplement	*xsd:boolean*	0:1	Whether the package is a supplement to another product
	Requires-Entitlement	*xsd:boolean*	0:1	Whether the product requires entitlement to other products.
	GivesEntitlement	*xsd:boolean*	0:1	Whether the product grants entitlements to other products.

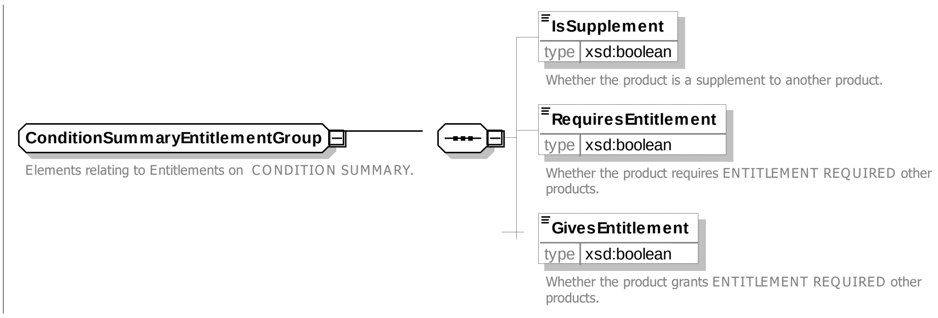


Figure 243 — *ConditionSummaryEntitlementGroup* — XSD

7.6.2.3.5.5 ConditionSummaryTravelGroup – Group

The ***ConditionSummaryTravelGroup*** summarises whether there are restrictions on travel associated with a FARE PRODUCT and /or SALES OFFER PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs and ACCESS RIGHT ASSIGNMENTs.

Table 229 – *ConditionSummaryTravelGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***HasOperator-Restrictions***	*Operator RestrictionEnum*	0:1	Limitations as to which OPERATOR's services may be used. See allowed values below.
	HasTravelTime-Restrictions	*xsd:boolean*	0:1	Whether limitations apply as to when travel may take place.
	HasRoute-Restrictions	*xsd:boolean*	0:1	Whether limitations apply as to the route that may be used.
«enum»	***HasTrain-Restrictions***	*TrainRestrictionEnum*	0:1	Limitations as to which trains may be used. See allowed values below.
	HasZone-Restrictions	*xsd:boolean*	0:1	Whether limitations apply as to the area in which travel may take place.
	CanBreak-Journey	*xsd:boolean*	0:1	Whether the user is allowed to break journey, i.e. leave transport network, at an intermediate point.
	ReturnTripsOnly	*xsd:boolean*	0:1	Whether must buy a return trip.

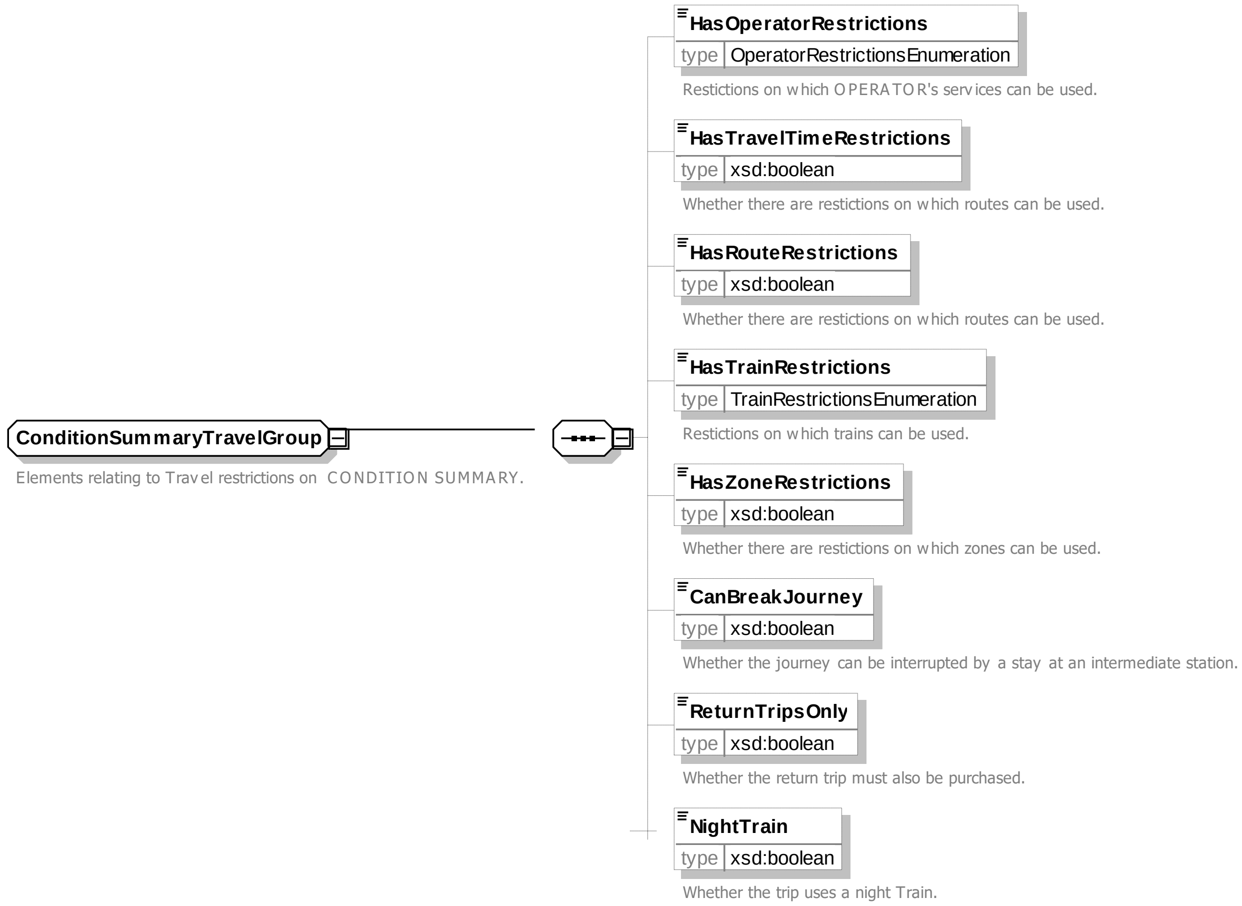


Figure 244 — *ConditionSummaryTravelGroup* — XSD

7.6.2.3.5.5.1 OperatorRestrictions – Allowed values

The following table shows the allowed values for ***OperatorRestrictions*** (*OperatorRestrictionsEnumeration*).

Table 230 – *OperatorRestrictions* – Allowed values

Value	Description
anyTrain	Can travel on services of any OPERATOR.
restricted	Restricted to certain OPERATORs' services.
specifiedOperatorOnly	Can travel only on a specific OPERATOR's services.

7.6.2.3.5.5.2 TrainRestrictions – Allowed values

The following table shows the allowed values for ***TrainRestrictions*** (*TrainRestrictionsEnumeration*).

Table 231 – *TrainRestrictions* – Allowed values

Value	Description
anyTrain	Can travel on any train.
restricted	Restricted to certain types of train.

specifiedTrainOnly	Can travel only on a specific train at a specified time and date.
specifiedTrainsOnly	Can travel only on certain trains and time.
specifiedTrainAndConnections	Can travel only on a specific train and on appropriate onward connections.

7.6.2.3.5.6 ConditionSummaryCommercialGroup – Group

The ***ConditionSummaryCommercialGroup*** summarises the commercial conditions associated with a FARE PRODUCT and /or SALES OFFER PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs, FARE PRICEs and ACCESS RIGHT ASSIGNMENTs.

Table 232 – *ConditionSummaryCommercialGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	CanChangeClass	*xsd:boolean*	0:1	Whether user can change class
	IsRefundable	*xsd:boolean*	0:1	Whether the ticket is refundable
	IsExchangable	*xsd:boolean*	0:1	Whether the ticket is exchangeable
	HasExchangeFee	*xsd:boolean*	0:1	Whether there is a fee for exchanges.
	HasDiscounted-Fares	*xsd:boolean*	0:1	Whether discounted Fares are allowed.
	AllowAdditional-Discounts	*xsd:boolean*	0:1	Whether more than one discount may be applied, e.g. Child + Companion.
	Allow-Companion-Discount	*xsd:boolean*	0:1	Whether there is a companion discount.
	HasMinimum-Price	*xsd:boolean*	0:1	Whether there is a minimum price when combining elements.
	Requires-PositiveBalance	*xsd:boolean*	0:1	Whether the product requires a positive stored balance to be used.

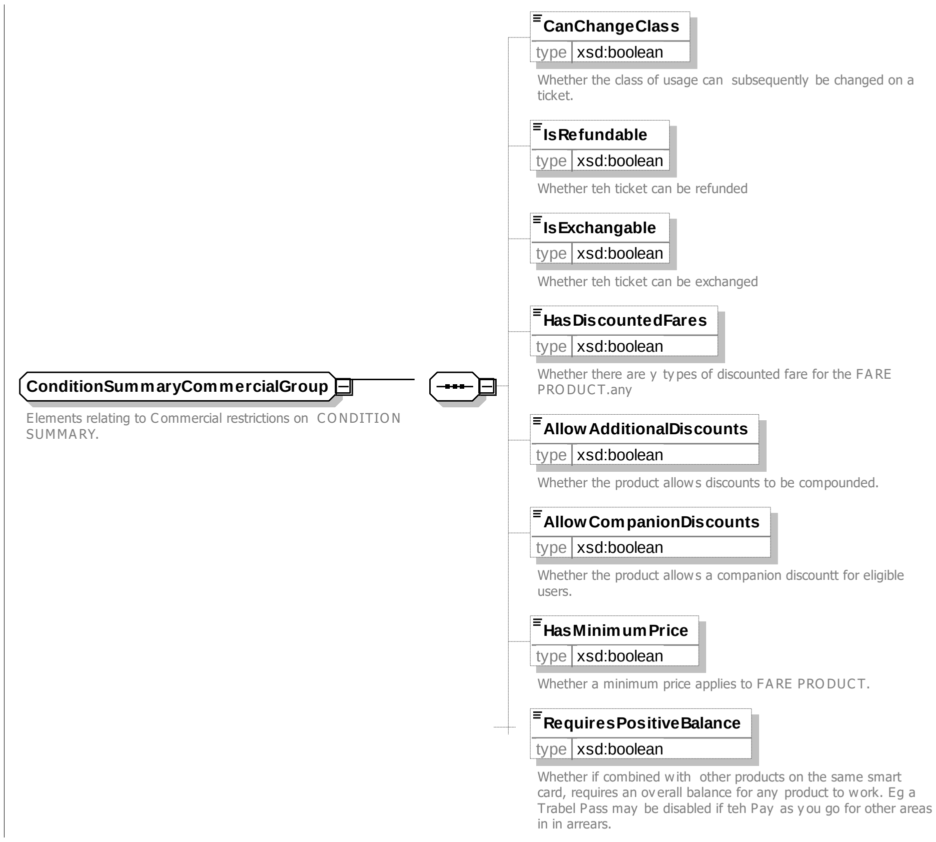


Figure 245 — *ConditionSummaryCommercialGroup* — XSD

7.6.2.3.5.7 ConditionSummaryChargingGroup – Group

The ***ConditionSummaryChargingGroup*** summarises the options relating to charging associated with a FARE PRODUCT and /or SALES OFFER PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs, FARE PRICEs and ACCESS RIGHT ASSIGNMENTs.

Table 233 – *ConditionSummaryReservationGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	PenaltyWIthout-Ticket	*xsd:boolean*	0:1	Whether there is a penalty for travelling without a ticket, i.e. tickets cannot be bought on-board. +v1.1
	AvailableOn-Subscription	*xsd:boolean*	0:1	Whether the product is available on subscription. +v1.1

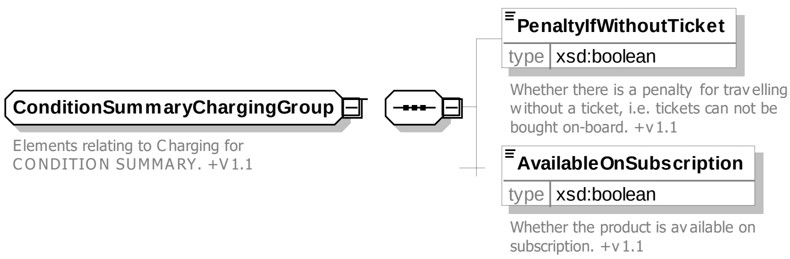


Figure 246 — *ConditionSummaryChargingGroup* — XSD

7.6.2.3.5.8 ConditionSummaryReservationGroup – Group

The ***ConditionSummaryReservationGroup*** summarises the restrictions on booking associated with a FARE PRODUCT and /or SALES OFFER PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs, FARE PRICEs and ACCESS RIGHT ASSIGNMENTs.

Table 234 – *ConditionSummaryReservationGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	HasPurchase-Conditions	*xsd:boolean*	0:1	Whether purchase conditions apply to the sale of the product, e.g. when must be bought or who may purchase.
	HasDynamic-Pricing	*xsd:boolean*	0:1	Whether product has dynamic pricing.
	Requires-Reservation	*xsd:boolean*	0:1	Whether a Reservation is required.
	HasReservation-Fee	*xsd:boolean*	0:1	Whether there is a fee for Reservations.
	HasQuota	*xsd:boolean*	0:1	Whether limited quota for the offer or it can be sold in unlimited numbers.

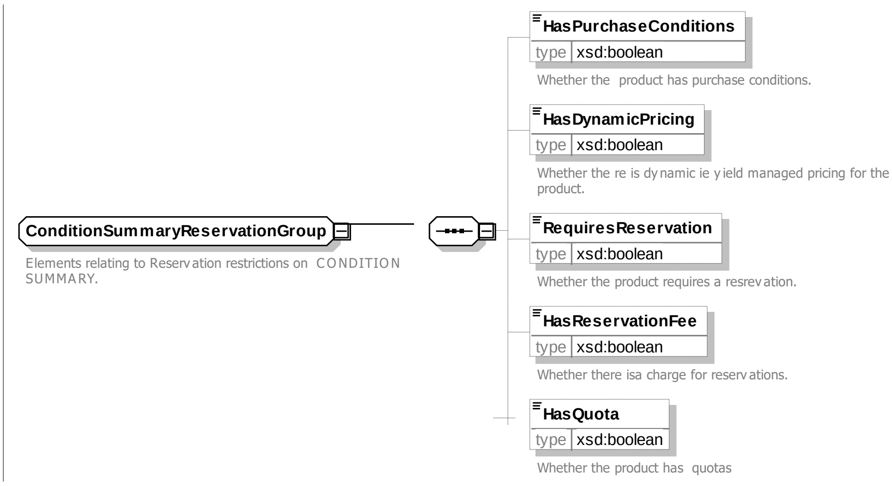


Figure 247 — *ConditionSummaryReservationGroup* — XSD

7.6.2.3.5.8.1 FareUseType – Allowed values

The following table shows the allowed values for ***FareUseType*** (FareUseTypeEnumeration).

Table 235 – *FareUseType* – Allowed values

Value	Description
single	Fare is a single, i.e. one-way trip.
return	Fare is a return, i.e. two-way trip.
multiple	Fare is a multiple, i.e. for repeated trips.

7.6.2.3.5.8.2 TariffBasis – Allowed values

Allowed values for ***TariffBasis*** *(TariffBasisEnumeration*).

Table 236 – TariffBasis – Allowed values

Value	Description
distance	Tariff is based on distance.
route	Tariff is based on the route taken.
group	Tariff is based on size of group.

7.6.2.3.6 TypeOfFareProduct – Model Element

A classification of FARE PRODUCTs.

Table 237 – *TypeOfFareProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF FARE PRODUCT inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfFareProduct-IdType*	1:1	Identifier of TYPE OF FARE PRODUCT.

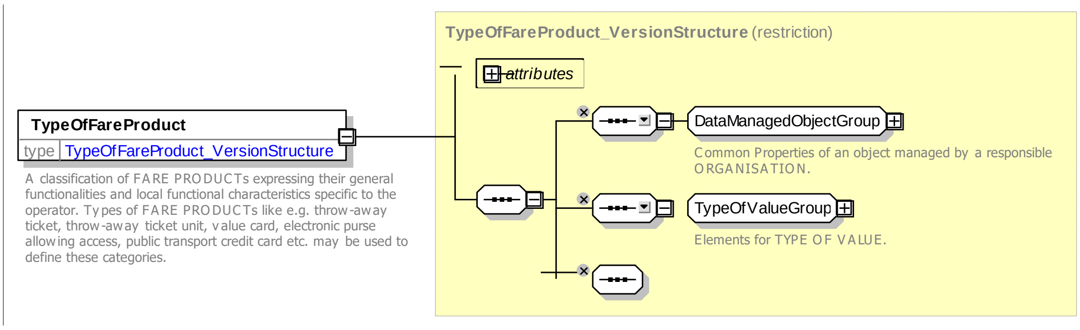


Figure 248 — *TypeOfFareProduct* — XSD

7.6.2.3.7 ChargingMoment – Model Element

A classification of FARE PRODUCTs according to the CHARGING MOMENT and the account location: pre-payment with cancellation (throw-away), pre-payment with debit on a value card, pre-payment without consumption registration (pass), post-payment etc.

Table 238 – *ChargingMoment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF CHARGING MOMENT inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*ChargingMomentIdType*	1:1	Identifier of TYPE OF CHARGING MOMENT.

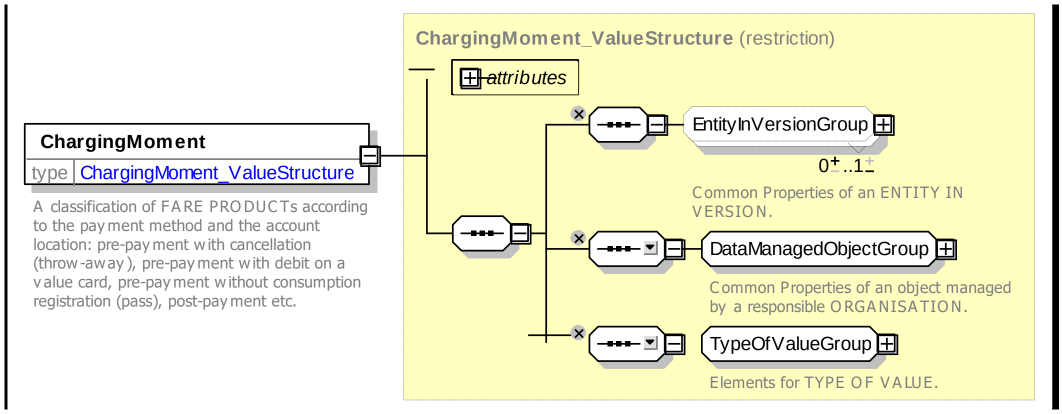


Figure 249 — *ChargingMoment* — XSD

7.6.2.4 Types of Fare Product – Attributes and XSD

7.6.2.4.1 PreassignedFareProduct – Model Element

A FARE PRODUCT consisting of one or several VALIDABLE ELEMENTs, specific to a CHARGING MOMENT.

Table 239 – *PreassignedFareProduct* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*FareProduct*	::>	PREASSIGNED FARE PRODUCT inherits from FARE PRODUCT.
«PK»	***id***	*Preassigned-FareProductIdType*	1:1	Identifier of PREASSIGNED FARE PRODUCT.
«enum»	***ProductType***	*PreassignedFareProduct-Enum*	1:1	Classification of PREASSIGNED FARE PRODUCT. See allowed values. +v1.1

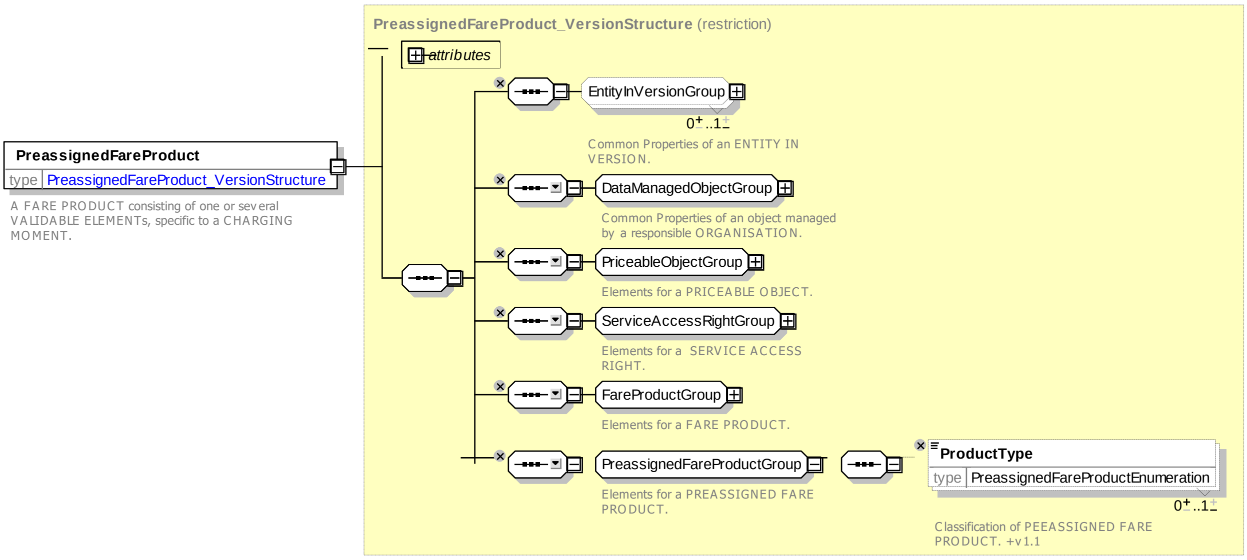


Figure 250 — *PreassignedFareProduct* — XSD

7.6.2.4.1.1 PreassignedFareProduct / ProductType – Allowed values

The following table shows the allowed values for ***PreassignedFareProduct / ProductType (****Preassigned-FareProductEnumeration*).

Table 240 – PreassignedFareProduct / ProductType – Allowed values

Value	Description
singleTrip	Single trip.
shortTrip	Short single trip (“short hop”).
timeLimitedSingleTrip	Single trip, time limited.
dayReturnTrip	Day return trip.
periodReturnTrip	Period return trip.
multistepTrip	Multistep trip.
dayPass	Day pass.
periodPass	Period pass (season ticket).
supplement	Supplement to another product.
other	Other.

7.6.2.4.2 AmountOfPriceUnit – Model Element

A FARE PRODUCT consisting of a stored value of PRICE UNITs: an amount of money on an electronic purse, amount of units on a value card etc.

Table 241 – *AmountOfPriceUnit* – Element

Classifi-cation	Name	Type	Card in-ality	Description
::>	::>	<u>*FareProduct*</u>	::>	AMOUNT OF PRICE UNIT inherits from FARE PRODUCT.
«PK»	***id***	*AmountOfPriceUnitIdType*	1:1	Identifier of AMOUNT OF PRICE UNIT.
«enum»	***ProductType***	*AmountOfPriceUnitEnum*	1:1	Classification of AMOUNT OF PRICE UNIT. See allowed values. +v1.1

«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a PRICE UNIT.
	Amount	*xsd:decimal*	0:1	Number of units.

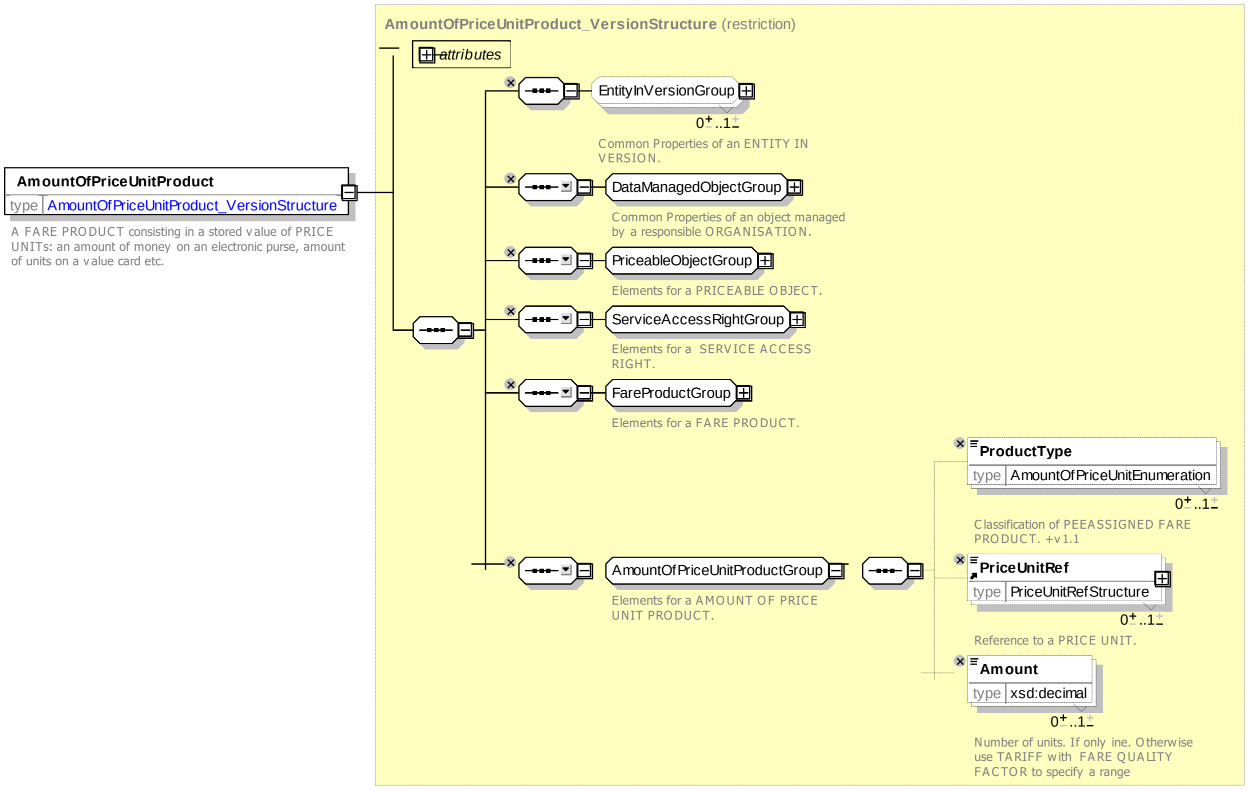


Figure 251 — *AmountOfPriceUnit* — XSD

7.6.2.4.2.1 AmountOfPrice / Product Type – Allowed values

The following table shows the allowed values for ***AmountOfPrice / ProductType*** (*AmountOfPriceUnitEnumeration*).

Table 242 – 7.6.2.4.2 AmountOfPrice / ProductType– Allowed values

Value	Description
tripCarnet	Carnet of trips.
passCarnet	Carnet of passes.
unitCoupon	Unit coupons.
storedValue	Holds stored value (currency or other unit).
other	Other

7.6.2.4.3 UsageDiscountRight – Model Element

A FARE PRODUCT allowing a customer to benefit from discounts when consuming VALIDABLE ELEMENTs.

Table 243 – *UsageDiscountRight* – Element

Classifi-cation	Name	Type	Card in-ality	Description
::>	::>	*FareProduct*	::>	USAGE DISCOUNT RIGHT inherits from FARE PRODUCT.
«PK»	***id***	*UsageDiscounRightIdType*	1:1	Identifier of USAGE DISCOUNT RIGHT.
«enum»	***ProductType***	*UsageDiscountRightEnum*	1:1	Classification of USAGE DISCOUNT RIGHT. See allowed values. +v1.1

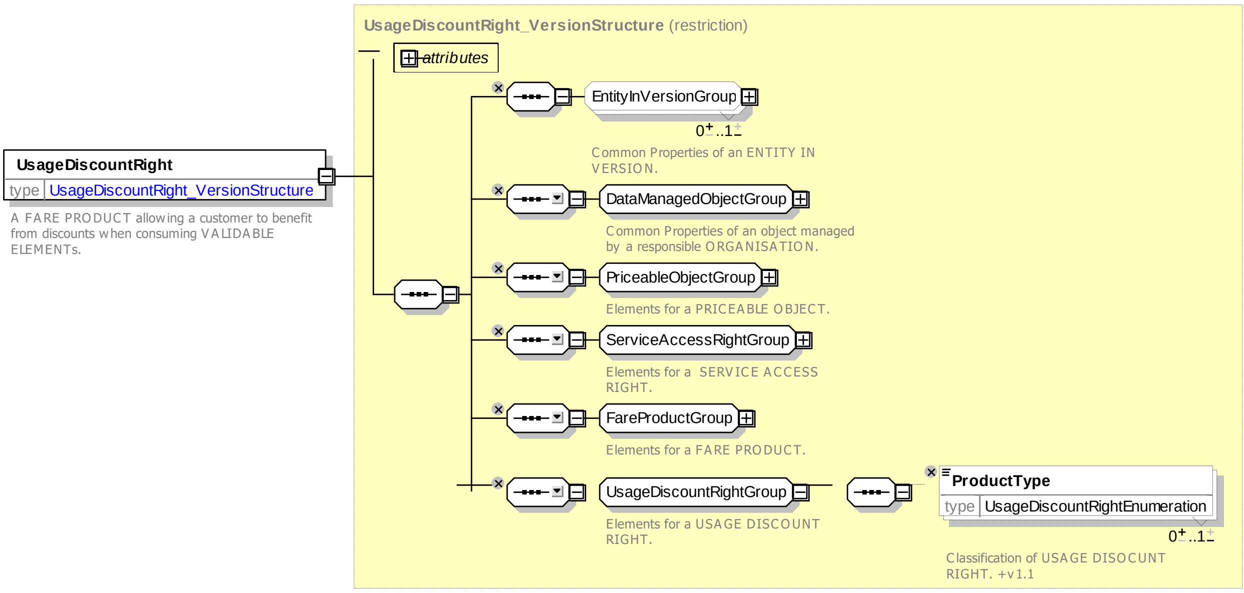


Figure 252 — *UsageDiscountRight* — XSD

7.6.2.4.3.1 UsageDiscountRight / Product Type – Allowed values

The following table shows the allowed values for ***UsageDiscountRight / ProductType*** (*UsageDiscountRightUnitEnumeration*).

Table 244 – 7.6.2.4.2 AmountOfPrice/ ProductType– Allowed values

Value	Description
mileagePoints	Discount given as mileage points
usageRebate	Discount given as rebate.
other	Other

7.6.2.4.4 SaleDiscountRight – Model Element

A FARE PRODUCT allowing a customer to benefit from discounts when purchasing SALES OFFER PACKAGEs.

Table 245 – *SaleDiscountRight* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*FareProduct*	::>	SALE DISCOUNT RIGHT inherits from FARE PRODUCT.
«PK»	***id***	*SaleDiscountRightIdType*	1:1	Identifier of SALE DISCOUNT RIGHT.
«enum»	***ProductType***	*SaleDiscountRightEnum*	1:1	Classification of SALE DISCOUNT RIGHT. See allowed values. +v1.1

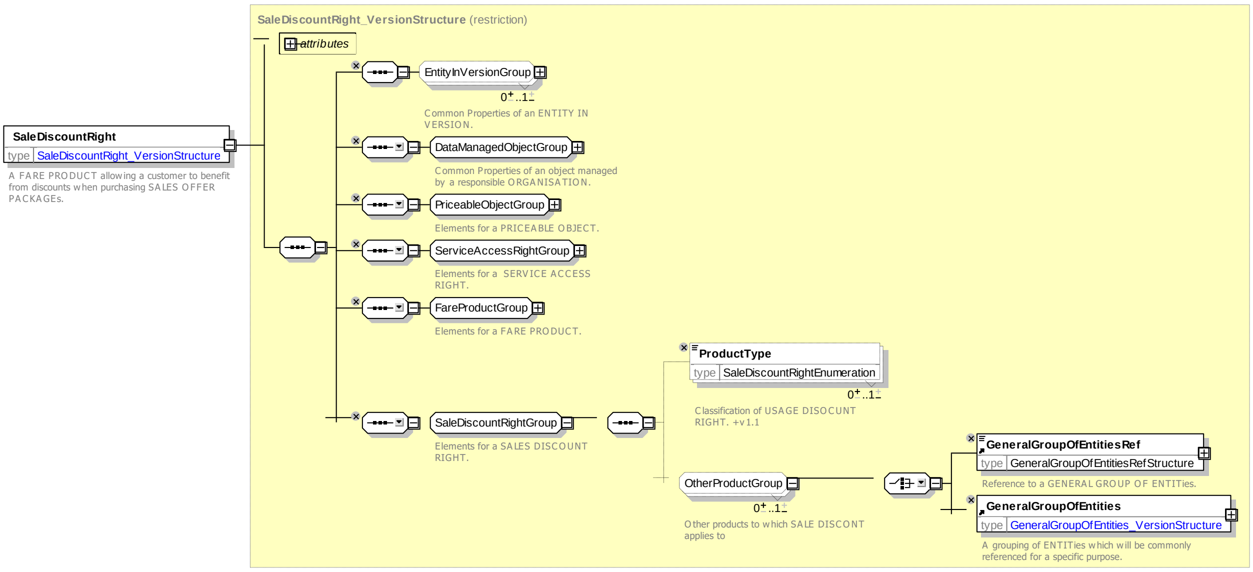


Figure 253 — *SaleDiscountRight* — XSD

7.6.2.4.4.1 UsageDiscountRight / Product Type – Allowed values

The following table shows the allowed values for ***UsageDiscountRight / ProductType*** (*UsageDiscountRighttEnumeration*).

Table 246 – 7.6.2.4.2 AmountOfPrice/ ProductType– Allowed values

Value	Description
travelCard	Discount is a Travel card that can be used to obtain a discount when buying certain other prododucts.
payAsYouGoRight	Discount is an automatic discount that is applied when paying with a Smartcard, device or Account based product.
other	Other

7.6.2.4.5 **CappedDiscountRight – Model Element**

A specialisation of SALE DISCOUNT RIGHT where the discount is expressed as a capping limit for a given time interval. For example, the London Oyster card fare, which charges for each journey at a reduced price until travel equivalent to a day pass has been consumed.

Table 247 – *CappedDiscountRight* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*SaleDiscountRight*	::>	CAPPED DISCOUNT RIGHT inherits from SALE DISCOUNT RIGHT***.***
«PK»	***id***	*CappedDiscountRight-IdType*	1:1	Identifier of CAPPED DISCOUNT RIGHT.
«cntd»	***cappingRules***	*CappingRule*	0:*	A set of parameters set a price cap on a product.

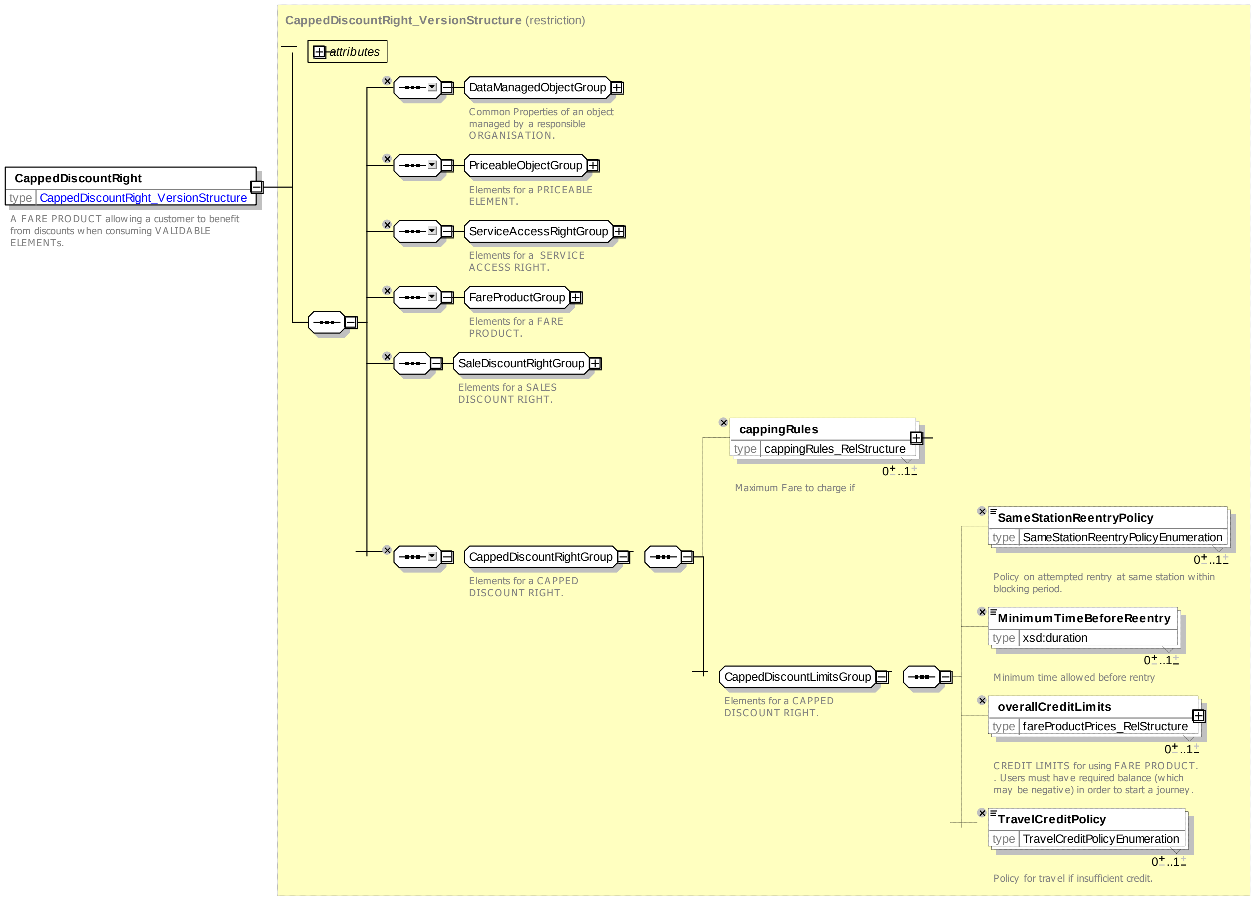


Figure 254 — *CappedDiscountRight* — XSD

7.6.2.4.6 **CappingRule – Model Element**

A capping limit for a given time interval, where the capping is expressed by another product. For example, the London Oyster card fare, which charges for each journey at a reduced price until travel equivalent to a day pass for the mode of travel has been consumed. A CAPPING RULE is a PRICEABLE OBJECT and may have USAGE PARAMETERS such as a USAGE VALIDITY PERIOD to specify how long the capping period is and a CHARGING POLICY to specify rules about travelling under credit.

Table 248 – *CappingRule* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	CAPPING RULE inherits from PRICEABLE OBJECT.
«PK»	***id***	*CappingRuleIdType*	1:1	Identifier of CAPPING RULE.
«cntd»	***Maximum-Distance***	*LengthType*	0:*	Capping distance if distance-based cap.
«enum»	***CappingPeriod***	*CappingPeriodEnum*	0:1	Period over which capping applies, e.g. daily. See allowed values below. A quantitative value can be set with a USAGE VALIDITY PERIOD, along with a more detailed definition of the start and end times.
«FK»	***CappedDiscount-RightRef***	*CappedDiscount-RightRef*	0:1	CAPPED DISCOUNT RIGHT for which this rule applies.
«FK»	***PreassignedFare ProductRef***	*PreassignedFare-ProductRef*	0:1	PREASSIGNED FARE PRODUCT whose prices set cap the for this product.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	0:1	VALIDABLE ELEMENT for which capping applies.
«cntd»	***validityParamete Assignments***	*ValidityParameterAssign ment+*	0:*	VALIDITY PARAMETER ASSIGNMENTS for this rule.
«cntd»	***prices***	*CappIngRulePrice*	0:*	Capping FARE PRICEs for this rule.

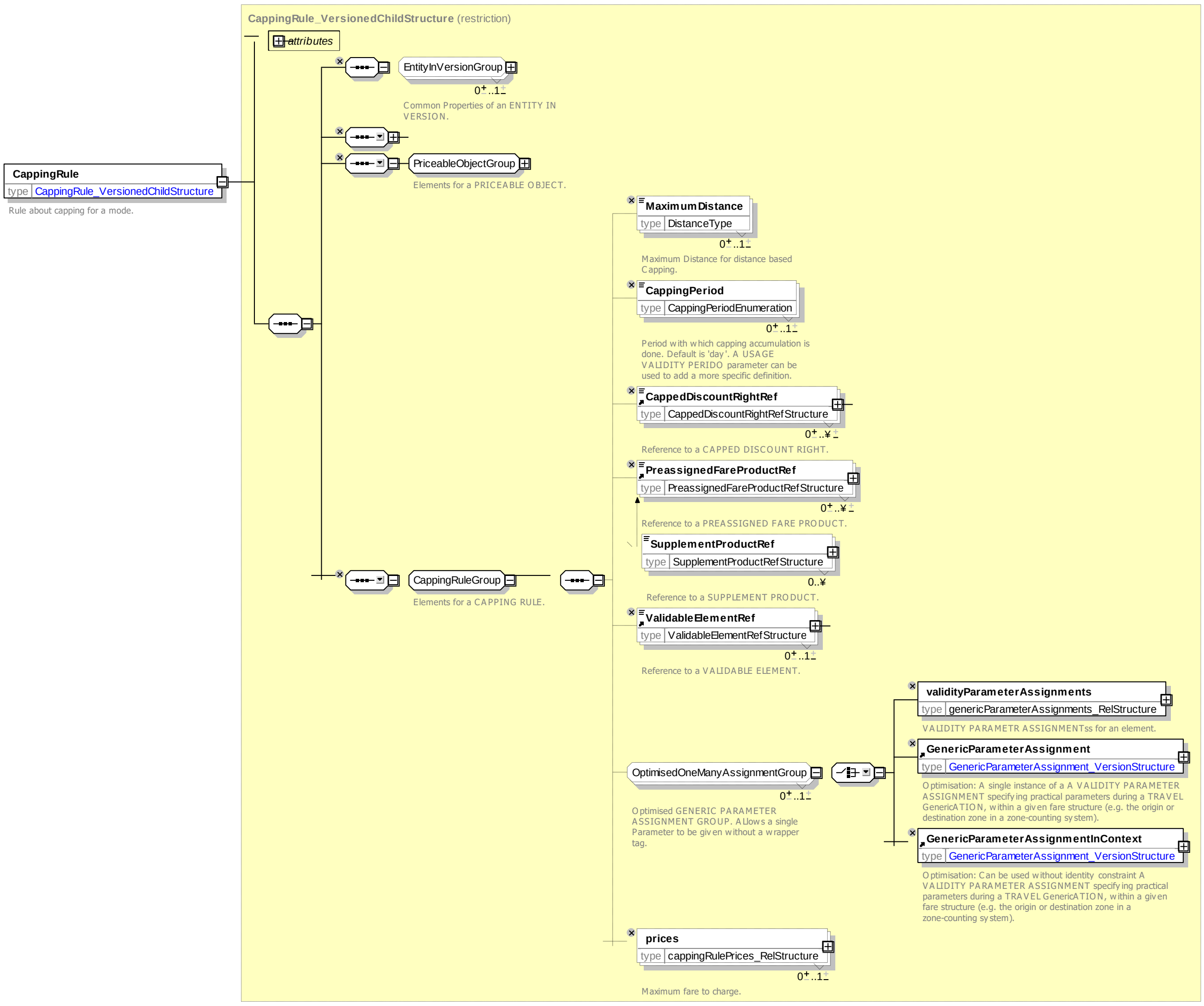


Figure 255 — *CappingRule* — XSD

7.6.2.4.6.1 CappingPeriod – Allowed values

The following table shows the allowed values for **CappingPeriod** (*CappingPeriodEnumeration*).

Table 249 – CappingPeriod – Allowed values

Value	Description
day	Capping period is current fare day.
week	Capping measurement period is fare week.
month	Capping period is fare month.

7.6.2.4.7 CappingRulePrice – Model Element

A set of all possible price features of a CAPPING RULE: default total price, discount in value or percentage etc.

Table 250 – *CappingRulePrice* – Element

Classifi-cation	Name	Type	Cardinality	Description

::>	::>	*FarePrice*	::>	CAPPING RULE PRICE inherits from FARE PRICE.
«PK»	***id***	*CappingRulePriceIdType*	1:1	Identifier of CAPPING RULE PRICE.
«FK»	***CappingRuleRef***	*CappingRuleRef*	0:1	CAPPING RULE for which this is the price. If not given by context, must be specified.

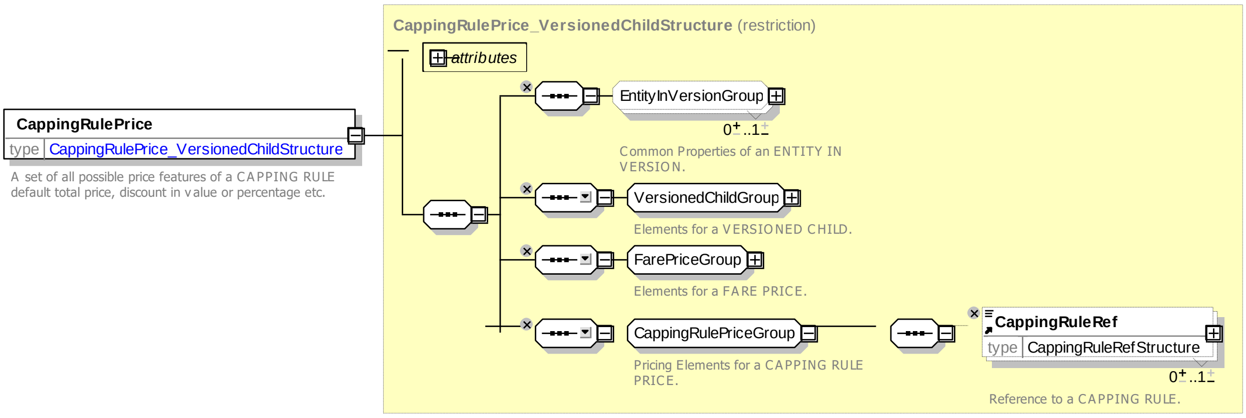


Figure 256 — *CappingRulePrice* — XSD

7.6.2.4.8 SupplementProduct – Model Element

An additional FARE PRODUCT that may be used to describe additional purchases entitled by another product.

A SUPPLEMENT PRODUCT is usually constrained by some or all of the parameters of the supplemented product, e.g. same service, same route, etc.

Table 251 – *SupplementProduct* – Element

Classifi-cation	Name		Type	Cardin ality	Description
::>	::>		*PreassignedFareProduct*	::>	SUPPLEMENT PRODUCT inherits from PREASSIGNED FARE PRODUCT.
«PK»	***id***		*SupplementProductIdType*	1:1	Identifier of SUPPLEMENT PRODUCT.
«enum»	***Supplement-ProductType***		*SupplementProduct-TypeEnum*	0:1	Classification of SUPPLEMENT PRODUCT. See allowed values. +v1.1
	Choice				
«FK»	*a*	***SupplementTo FareProduct-Ref***	*FareProductRef+*	0:1	Reference to base PRE ASSIGNED FARE PRODUCT OFFER for which this is a supplement.
«cntd»	*b*	***SupplementTo***	*FareProductRef+*	0:*	Reference to base PRE ASSIGNED FARE PRODUCT OFFER for which this is a supplement.

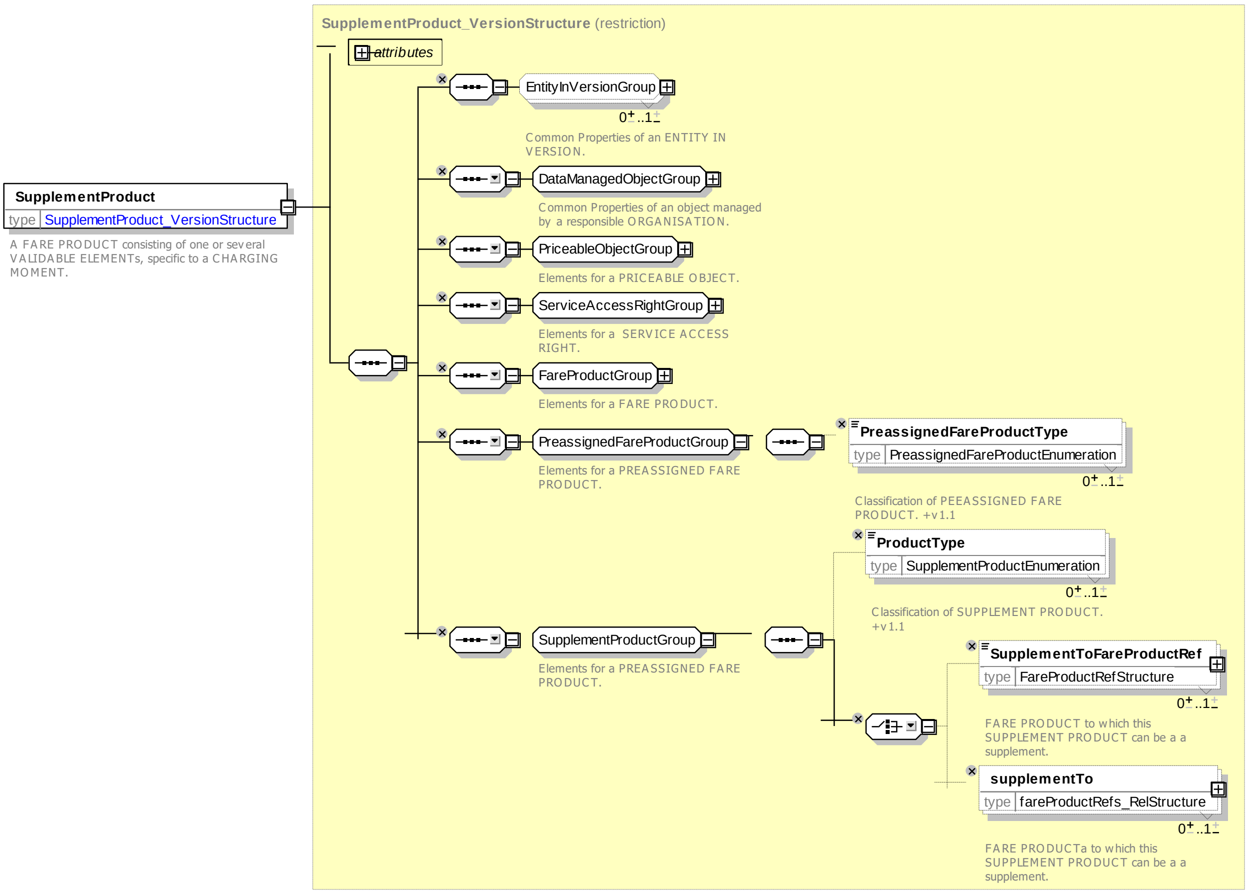


Figure 257 — *SupplementProduct* — XSD

7.6.2.4.8.1 SupplementProductType – Allowed values

The following table shows the allowed values for ***SupplementProductType*** ***(****SupplementProductEnumeration)*.

Table 252 – *SupplementProductType* – Allowed values

Value	Description
bicycle	Right to take a bicycle.
dog	Right to take a dog.
animal	Right to take an animal.
meal	Right to consume a meal.
wifi	Right to use wifi.
extraLuggage	Right to take extra luggage.
upgrade	Upgarde, e.g. to 1st Class, or a different train type.
journeyExtension	Extension to journey.
journeyAddOn	Additional travel added to journey, e.g. taxi, urban travel, etc.
eventAddOn	Add-on event bundled with a journey.
topUp	Top up to stored value
parking	Parking bundled with journey.
other	Other right.

7.6.2.4.9 **EntitlementProduct – Model Element**

An additional ACCESS RIGHT that gives entitlement to buy or use other products and may be a perquisite.

Table 253 – *EntitlementProduct* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*ServiceAccessRight*	::>	ENTITLEMENT PRODUCT inherits from SERVICE ACCESS RIGHT
«PK»	***id***	*EntitlementProductIdType*	1:1	Identifier of ENTITLEMENT PRODUCT.
«FK»	***General-OrganisationRef***	*GeneralOrganisationRef*	0:1	Reference to ORGANISATION offering product.
«cntd»	***prices***	*(FarePrice)*	0:*	FARE PRICEs for this product.

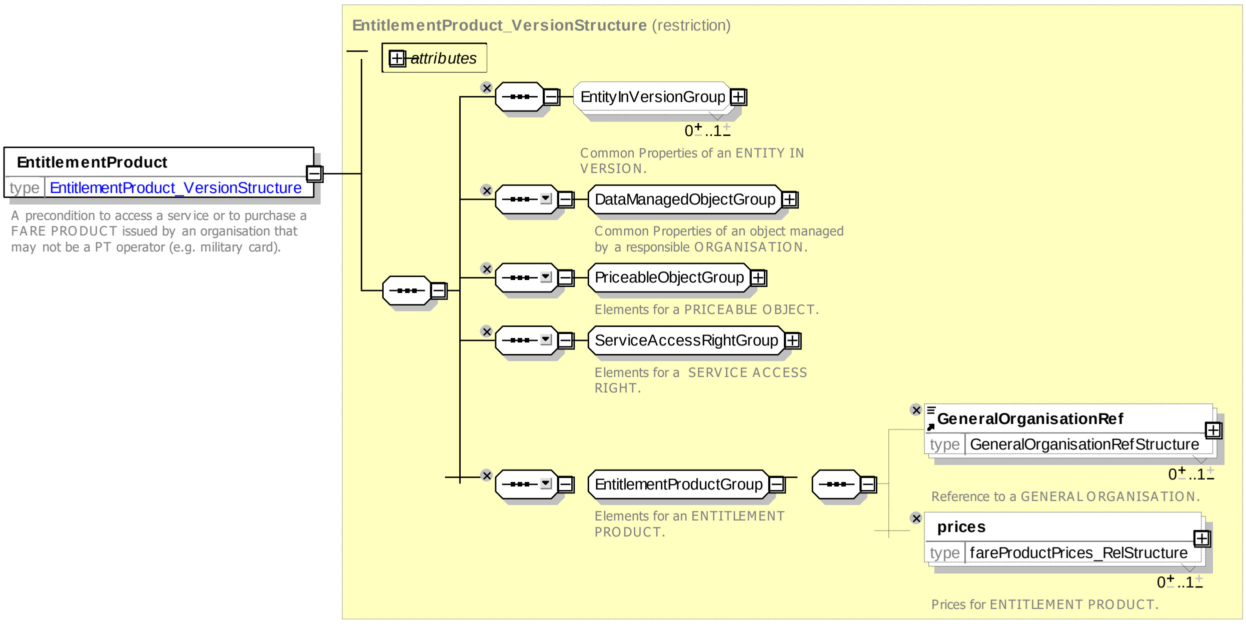


Figure 258 — *EntitlementProduct* — XSD

7.6.2.4.10 **ThirdPartyProduct – Model Element**

A FARE PRODUCT that is marketed together with a Public Transport Fare Product.

Table 254 – *ThirdPartyProduct* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*FareProduct*	::>	THIRD PARTY PRODUCT inherits from FARE PRODUCT.
«PK»	***id***	*ThirdPartyProductIdType*	1:1	Identifier of THIRD PARTY PRODUCT.

			CHOICE		
«cntd»	***a***	***GeneralGroupOf Entities***	*GeneralGroupOfEntities*	0:1	GENERAL GROUP OF ENTITIES associated with Third PARTY product.
«cntd»	***a***	***GeneralGroupOf EntitiesRef***	*GeneralGroup-OfEntitiesRef*	0:1	RRefrence to GENERAL GROUP OF ENTITIES associated with Third PARTY product.

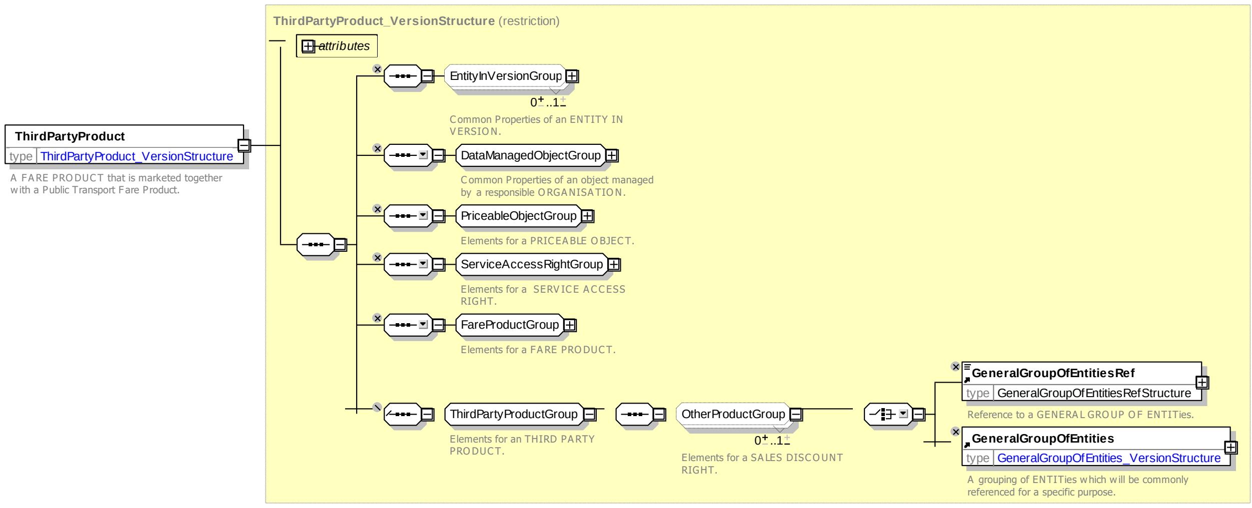


Figure 259 — *ThirdPartyProduct* — XSD

7.6.2.4.11 FareProduct: XML Example of Preassigned Fare product – Prepaid ticket

The following code fragment shows two PREASSIGNED FARE PRODUCTs for trips on the TfL area, one for a cash prepaid trip, one for a pay as you go trip that is charged electronically by adjustments at the entry and exit gates. There are different restrictions on the VALIDABLE ELEMENTs for each different mode.

For EXAMPLE:

```
    <PreassignedFareProduct version="any" id="tfl:Prepaid_fare">
        <Name>Cash Ride Paid for before ride - </Name>
        <ChargingMomentRef version="any" ref="tfl:prepaid"/>

        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl:Prepaid_fare@singleRide">
                <Name>A single ride</Name>
                <limitations>
                    <FrequencyOfUseRef version="any" ref="tfl:one_trip"/>
                    <RoundTripRef version="any" ref="tfl:single"/>
                    <UsageValidityPeriodRef version="any"
ref="tfl:UsageValidityPeriod:endOfRide"/>
                </limitations>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any" id="tfl:Prepaid_fare@returnRide">
                <Name>A single ride</Name>
                <limitations>
                    <FrequencyOfUseRef version="any" ref="tfl:one_trip"/>
                    <RoundTripRef version="any" ref="tfl:return"/>
                    <UsageValidityPeriodRef version="any"
ref="tfl:UsageValidityPeriod:endOfRide"/>
                </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>

        <accessRightsInProduct>
```

```
            <AccessRightInProduct id="tfl:Prepaid_fare@metroTrip" version="any">
                <Name>Metro trip </Name>
                <Description>from entry barrier to exit barrier </Description>
                <ValidableElementRef ref="lul:metro_trip" version="any"/>
                <
            </AccessRightInProduct>

            <AccessRightInProduct id="tfl:Prepaid_fare@railTrip" version="any">
                <Name>Metro trip </Name>
                <ValidableElementRef ref="nr:railTrip" version="any"/>

            <AccessRightInProduct id="tfl:Prepaid_fare@busOrTramTrip" version="any">
                <Name>Bus or tram trip trip </Name>
                <ValidableElementRef version="any" ref="lbsl::busOrTramTrip"/>
            </AccessRightInProduct>
            <AccessRightInProduct id="tfl:Prepaid_fare@riverTrip" version="any">
                <Name>River trip </Name>
                <ValidableElementRef ref="lrs:riverTrip" version="any"/>
            </AccessRightInProduct>
            <AccessRightInProduct id="tfl:Prepaid_fare@river_hopOnOff" version="any">
                <Name>River trip </Name>
                <ValidableElementRef ref="lrs:river_hopOnOff" version="any"/>
            </AccessRightInProduct>
            <AccessRightInProduct id="tfl:Prepaid_fare@cablewayTrip" version="any">
                <Name>River trip </Name>
                <ValidableElementRef ref="ea::cablewayTrip" version="any"/>
            </AccessRightInProduct>
        </accessRightsInProduct>
    </PreassignedFareProduct>


    <PreassignedFareProduct version="any" id="tfl:PayAsYouGo_fare">
        <Name>Cash Ride Paid for before ride - </Name>
        <ChargingMomentRef version="any" ref="tfl:payAsYouGo"/>

        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl:PayAsYouGo_fare@Ride">
                <Name>Ticket Allows only one trip</Name>
                <limitations>
                    <FrequencyOfUseRef version="any" ref="tfl:one_trip"/>
                    <UsageValidityPeriodRef version="any" ref="tfl:endOfRide"/>
                </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>

        <accessRightsInProduct>
            <AccessRightInProductRef ref="tfl:Prepaid_fare@metroTrip" version="any"/>
            <AccessRightInProductRef ref="tfl:Prepaid_fare@railTrip" version="any"/>
            <AccessRightInProductRef ref="tfl:Prepaid_fare@busOrTramTrip" version="any"/>
            <AccessRightInProductRef ref="tfl:Prepaid_fare@riverTrip" version="any"/>
            <AccessRightInProductRef ref="tfl:Prepaid_fare@river_hopOnOff" version="any"/>
            <AccessRightInProductRef ref="tfl:Prepaid_fare@cablewayTrip" version="any"/>
        </accessRightsInProduct>
    </PreassignedFareProduct>
```

7.6.2.4.12 FareProduct: XML Example of Preassigned Fare product – Travel Pass

The following code fragment shows a PREASSIGNED FARE PRODUCT for a pass to travel trips on the TfL network., The card is available in 1 day 7 day and monthly verisions not transferable.

For EXAMPLE:

```
    <PreassignedFareProduct version="any" id="tfl:TravelCard">
        <Name>Travel card Not On Oyster</Name>
        <Description>Travelcards let you travel as often as you like on bus, Tube, tram, DLR,
London Overground and National Rail services within the London travel zones. They are valid for
either one or 7 days and are issued as paper tickets from the Visitor Shop.</Description>
        <ChargingMomentRef version="any" ref="tfl:prepaid"/>
        <ConditionSummary>
            <ProvidesCard>true</ProvidesCard>
            <GivesEntitlement>true</GivesEntitlement>
        </ConditionSummary>
```

```xml
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl:TravelCard@give@TCDisco">
                <EntitlementGivenRef version="any" ref="tfl:TravelCard_sale_discount"/>
                <EntitlementGivenRef version="any" ref="tfl:Annual_pass_GoldCard"/>
            </GenericParameterAssignment>
            <GenericParameterAssignment version="any" id="tfl:TravelCard@periods">
                <Name>Travel Periods </Name>
                <LimitationGroupingType>OR</ LimitationGroupingType >
                <limitations>
                    <UsageValidityPeriodRef version="any"
                        ref="tfl:TravelCardValidityPeriod@1DAnyTime"/>
                    <UsageValidityPeriodRef version="any"
                        ref="tfl:TravelCardValidityPeriod@1DOffPeak"/>
                    <UsageValidityPeriodRef version="any"
                        ref="tfl:TravelCardValidityPeriod@1WAnyTime"/>
                </limitations>
            </GenericRightParameterAssignment>


            <GenericParameterAssignment version="any" id="tfl:TravelCard@Transferability">
                <Name>No Transfer by anyone.</Name>
                <limitations>
                    <TransferabilityRef ref="tfl:NoTransfer" version="any"/>
                    <UserProfileRef version="any" ref="tfl:anyone"/>
                </limitations>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any" id="tfl:TravelCard@UserProfile@anyone">
                <Name>Anyone can use at full fare.</Name>
                <ChargingBasis>normalFare</ChargingBasis>
                <limitations>
                    <UserProfileRef version="any" ref="tfl:anyone"/>
                </limitations>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any"
                        id="tfl:TravelCard@UserProfile@concession">
                <Name>Concession card holder</Name>
                <ChargingBasis>discounted</ChargingBasis>
                <limitations>
                    <UserProfileRef version="any" ref="tfl:concession"/>
                </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </PreassignedFareProduct>
```

7.6.2.4.13 FareProduct: XML Example of Amount of Price Unit product

The following code fragment shows a AMOUNT OF PRICE UNIT product used to defined an a top up for an electronic card – the effect is to add some cash to the card that are not specifc to any particular fare. The size of the transactiosn is limited to between 5 and 50 GBP.

For EXAMPLE:

```xml
<!--- ====OYSTER TOP UP ==== -->
    <AmountOfPriceUnitProduct version="any" id="tfl:Oyster_top_up">
        <Name>Cash Top up Payment before ride - </Name>
        <ChargingMomentRef version="any" ref="tfl:prepaid"/>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <GoesOnCard>true</GoesOnCard>
            <RequiresEntitlement>true</RequiresEntitlement>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl:Oyster_top_up@Req@PAYG">
                <limitations>
                    <EntitlementRequiredRef version="any" ref="tfl:PayAsYouGo_right_holder"/>
                </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>
        <prices>
            <FareProductPrice version="any" id="tfl:Oyster_top_up">
```

```xml
                <Name>Top up must be between the following </Name>
                <MinimumPrice>5</MinimumPrice>
                <MaximumPrice>50</MaximumPrice>
                <RoundingRef ref="tfl:Rounding:TopUpIn5PoundSteps" version="any"/>
            </FareProductPrice>
        </prices>
    </AmountOfPriceUnitProduct>
```

7.6.2.4.14 FareProduct: XML Example of Sales Discount product

The following code fragment shows a SALE DISCOUNT RIGHT that a TfL Travel CARD Holder has that grants discounted Payas you go fares on metro rail and bus, and discounted fares on river and cableway.

For EXAMPLE:

```xml
    <SaleDiscountRight version="any" id="tfl:TCSDiIsco">
        <Name>Discount Rights associated with a Travel card</Name>
        <Description> Travel card allows some things to be bought at a discount</Description>
        <ChargingMomentRef version="any" ref="tfl:prepaid"/>
        <ConditionSummary>
            <ProvidesCard>true</ProvidesCard>
        </ConditionSummary>

        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl:TCSDiIsco@Req@TravelCard">
                <limitations>
                    <EntitlementRequiredRef version="any" ref="tfl:Req:travel_card_holder"/>
                </limitations>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any"
                        id="tfl:TCSDiIsco@Give@Discounted_PayAsYouGo_fare">
                <limitations>
                    <EntitlementGivenRef version="any" ref="tfl:PayAsYouGo_fare"/>
                    <EntitlementGivenRef version="any" ref="tfl:Discounted_prepaid_fare"/>
                </limitations>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any" id="tfl:TCSDiIsco@modes">
                <Name>Travel card allows use of bus metro and overgrund - 1 Day Products
only</Name>
                <GroupingType>OR</GroupingType>
                <includes>
                    <GenericParameterAssignment version="any"
                        id="tfl:TCSDiIsco@VehicleMode@river">
                        <Description>Scheduled Riverboat services at 1/3 off - (Show your
Travelcard at the time of travel)</Description>
                        <ChargingBasis>discounted</ChargingBasis>
                        <ValidityParameters>
                            <VehicleModes>water</VehicleModes>
                            <GroupOfLinesRef version="any" ref="lrs:RiverBus"/>
                        </ValidityParameters>
                    </GenericParameterAssignment>

                    <GenericParameterAssignment version="any"
                        id="tfl:TCSDiIsco@VehicleMode@cableway">
                        <ChargingBasis>discounted</ChargingBasis>
                        <ValidityParameters>
                            <VehicleModes>cableway</VehicleModes>
                        </ValidityParameters>
                    </GenericParameterAssignment>
                </includes>
            </GenericParameterAssignment >

            <GenericParameterAssignment version="any" id="tfl:TCSDiIsco@transferability">
                <Name>Card can not be used by another adult</Name>
                <limitations>
                    <TransferabilityRef ref="tfl:NoTransfer" version="any"/>
                </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>
        <priceGroups>
            <FareTable version="any" id="tfl:TCSDiIsco@metro_tram_bus">
```

```
                <members>
                    <FareProductPrice version="any" id="tfl:TCSDiIsco@metro_tram_bus">
                        <Name>Travelcards you can get a third off single tickets if you have a
Travelcard, whether it's a paper one or on your Oyster card.</Name>
                        <DiscountAsPercentage>34.00</DiscountAsPercentage>
                    </FareProductPrice>
                </members>
                <PreassignedFareProductRef version="any" ref="tfl:Prepaid_fare"/>
            </FareTable>
        </priceGroups>

    </SaleDiscountRight>
```

7.6.2.4.15 FareProduct: XML Example of Capped Discount Right product

The following code fragment shows a CAPPED DISCOUNT RIGHT that described the TfL OYSTER Pay as you go product which limits the purchase coust of multiple fares on the same day to the price of a day pass on the equivalent zones and demand periods (ie peak and offp eak)

There are different CAPPING RULEs for different modes (e.g. cableway is excluded)

For EXAMPLE:

```
<!-- OYSTER PAY AS YOU GO CARD -->
    <CappedDiscountRight version="any" id="tfl::OysterPAYGRight">
        <Name>Oyster Pay as You go discount capped</Name>
        <Description> right to purchase with fare capped at day pass rate </Description>
        <ChargingMomentRef version="any" ref="tfl:PAYG"/>
        <ConditionSummary>
            <ProvidesCard>true</ProvidesCard>
        </ConditionSummary>

        < validityParameterAssignments >
            <GenericParameterAssignment version="any" id="tfl:OysterPAYGRight@Give@PAYGFare">
                <limitations>
                    <EntitlementGivenRef version="any" ref="tfl:PAYGFare"/>
                </limitations>
            </GenericParameterAssignment>

            < GenericParameterAssignment version="any" id="tfl:Oyster@interchanging">
                <Description>Splitting your journey http://www.tfl.gov.uk/tickets/14872.aspx You
can't split your journey when using Oyster pay as you go. You can with National Rail
tickets.</Description>
                <GroupingType>AND</GroupingType>

                <includes>
                    <GenericParameterAssignment version="any" id="tfl:Oyster@all_modes">
                        <Name>Can go on all modes </Name>
                        <ValidityParameters><VehicleModes>metro tram bus
cableway</VehicleModes></ValidityParameters>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment version="any" id="tfl:Oyster@rail@Overground">
                        <Name>Can go on overground Rail journeys</Name>
                        <ValidityParameters>
                            <VehicleModes>rail</VehicleModes>
                            <FareZoneRef version="any" ref="tfl:Overground"/>
                        </ValidityParameters>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment version="any"
id="tfl:Oyster@FareZone@rail@NROysterArea">
                        <Name>Can go on Rail journeys within Rail Oyster area</Name>
                        <ValidityParameters>
                            <VehicleModes>rail</VehicleModes>
                            <FareZoneRef version="any" ref="nr:NationalRail_Oyster_area"/>
                        </ValidityParameters>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment version="any" id="tfl:Oyster@river">
                        <Name>34 % discount on river </Name>
                        <Description>Travelcards You can get a third off single tickets if you
have a Travelcard, whether it's a paper one or on your Oyster card.</Description>
                        <ChargingBasis>discounted</ChargingBasis>
                        <ValidityParameters><VehicleModes>water</VehicleModes>
                        </ValidityParameters>
                    </GenericParameterAssignment>
```

```
                        <GenericParameterAssignment version="any" id="tfl:Oyster@can_take_children">
                            <limitations>
                                <GroupTicketRef version="any" ref="tfl:family_group"/>
                            </limitations>
                        </GenericParameterAssignment>
                    </includes>
                </GenericParameterAssignment >

                <GenericParameterAssignment version="any" id="tfl:Oyster@Oyster@refunding">
                    <Name>You will need to pay £5 deposit when you get a new Oyster card. This is
refundable if you return the card.</Name>
                    <RefundingRef ref="tfl:Refunding:Refundable" version="any"/>
                </GenericParameterAssignment>

                <GenericParameterAssignment version="any" id="tfl:Oyster@transferability">
                    <Name>Transferability</Name>
                    <GroupingType>OR</GroupingType>
                    <includes>
                        <GenericParameterAssignment version="any"
id="tfl:Oyster@Transferability@adult">
                            <Name>Adult over 18 or under</Name>
                            <Description>You can give your Oyster card to someone else
http://www.tfl.gov.uk/tickets/19798.aspx
    If your Oyster only has pay as you go credit at adult rate on it you can let someone else use
it, even if the card is registered in your name.</Description>
                            <ChargingBasis>normalFare</ChargingBasis>
                            <limitations>
                                <TransferabilityRef ref="tfl:can_transfer" version="any"/>
                                <UserProfileRef version="any" ref="tfl:adult"/>
                            </limitations>
                        </GenericParameterAssignment>
                        <GenericParameterAssignment version="any"
id="tfl:Oyster@Transferability@concessions">
                            <Name>Disabled card holder</Name>
                            <ChargingBasis>free</ChargingBasis>
                            <limitations>
                                <TransferabilityRef ref="tfl:NoTransfer" version="any"/>
                                <UserProfileRef version="any" ref="tfl:concession"/>
                            </limitations>
                        </GenericParameterAssignment>
                    </includes>
                </ GenericParameterAssignment >
            </validityParameterAssignments >

            <prices>
                <FareProductPrice version="any" id="tfl:Oyster@costOfCard">
    <Description>You will need to pay £5 deposit when you get a new Oyster card. This is
refundable if you return the card.</Description>
                            <Amount>5.00</Amount>
                </FareProductPrice>

                </ prices >


                <cappingRules>
                    <CappingRule id="tfl:Oyster@rail" version="any">
                        <Name>Capping rule for Rail Travel</Name>
                        <CappingPeriod>day</CappingPeriod>
                        <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
                        <!-- =CAPPING PRODUCT -->
                        <PreassignedFareProductRef version="any" ref="tfl:TravelCard_on_Oyster"/>
                        <ValidableElementRef version="any" ref="nr:railTrip"/>
                        <validityParameterAssignments>
                            <GenericParameterAssignment version="any" id="tfl:
@CappingRule@PAYG@rail">
                                <limitations>
                                    <ChargingPolicyRef version="any" ref="tfl:PAYG"/>
                                    <PenaltyPolicyRef version="any" ref="tfl:PAYG"/>
                                    <UsageValidityPeriodRef version="any" ref="tfl:PAYG@1D"/>
                                    <UserProfileRef version="any" ref="tfl:adult"/>
                                </limitations>
                            </GenericParameterAssignment>
                        </validityParameterAssignments>
                        <fareTables>
                            <FareTable id="tfl:Oyster@rail" version="any">
                                <Name> Maximum fare price</Name>
```

```
                        <cells>
                            <Cell version="any" id="tfl:Oyster@max_oprice@rail@adult">
                                <CellPrice>
                                    <Name>Maximum daily price to charge</Name>
                                    <LimitingRuleRef version="any" ref="tfl:max-8.50"/>
                                </CellPrice>
                                <ValidableElementRef version="any" ref="nr:mrailTrip"/>
                                <UserProfileRef version="any" ref="tfl:adult"/>
                                <UsageValidityPeriodRef version="any" ref="tfl:PAYG@1D"/>
                            </Cell>
                        </cells>
                    </FareTable>
                </fareTables>
            </CappingRule>
            <CappingRule id="tfl:Oyster@metro" version="any">
                <Name>Capping rule for Metro Travel</Name>
                <CappingPeriod>day</CappingPeriod>
                <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
                <PreassignedFareProductRef version="any" ref="tfl:TravelCard_on_Oyster"/>
                <ValidableElementRef version="any" ref="lul:metro_trip"/>
                <validityParameterAssignments>
                    <GenericParameterAssignment version="any" id="tfl:
@CappingRule@PAYG@metro">
                        <limitations>
                            <ChargingPolicyRef version="any" ref="tfl:PAYG"/>
                            <PenaltyPolicyRef version="any" ref="tfl:PAYG"/>
                            <UsageValidityPeriodRef version="any" ref="tfl:PAYG@1D"/>
                            <UserProfileRef version="any" ref="tfl:adult"/>
                        </limitations>
                    </GenericParameterAssignment>
                </validityParameterAssignments>
                <fareTables>
                    <FareTable id="tfl:Oyster@metro" version="any">
                        <cells>
                            <Cell version="any" id="tfl:Oyster@MaxPrice@metro@adult">
                                <CellPrice>
                                    <Name>Maximum daily price to charge</Name>
                                    <LimitingRuleRef version="any" ref="tfl:max-8.50"/>
                                </CellPrice>
                                <ValidableElementRef version="any" ref="lul:metro_trip"/>
                                <UserProfileRef version="any" ref="tfl:adult"/>
                                <UsageValidityPeriodRef version="any" ref="tfl:PAYG@1D"/>
                            </Cell>
                        </cells>
                    </FareTable>
                </fareTables>
            </CappingRule>
            <CappingRule id="tfl:Oyster@bus" version="any">
                <Name>Capping rule for Bus Travel</Name>
                <CappingPeriod>day</CappingPeriod>
                <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
                <PreassignedFareProductRef version="any" ref="lbsl:TravelCard_on_Oyster@Bus"/>
                <ValidableElementRef version="any" ref="lbsl::busOrTramTrip"/>
                <validityParameterAssignments>
                    <GenericParameterAssignment version="any" id="tfl: @PAYG@bus">
                        <limitations>
                            <ChargingPolicyRef version="any" ref="tfl:PAYG"/>
                            <UsageValidityPeriodRef version="any" ref="tfl:PAYG@1D"/>
                            <UserProfileRef version="any" ref="tfl:adult"/>
                        </limitations>
                    </GenericParameterAssignment>
                </validityParameterAssignments>
                <fareTables>
                    <FareTable id="tfl:Oyster@bus" version="any">
                        <cells>
                            <Cell version="any" id="tfl:Oyster@max_price@Bus_Tram@adult">
                                <CellPrice>
                                    <Name>Maximum daily price to charge</Name>
                                    <LimitingRuleRef version="any" ref="tfl:max-4.40"/>
                                </CellPrice>
                                <ValidableElementRef version="any"
ref="lbsl::busOrTramTrip"/>
                                <UserProfileRef version="any" ref="tfl:adult"/>
                                <UsageValidityPeriodRef version="any" ref="tfl:PAYG@1D"/>
                            </Cell>
                        </cells>
                    </FareTable>
```

```
                <\/fareTables>
            </CappingRule>
            <CappingRule id="tfl:Oyster@river" version="any">
                <Name>Capping rule for River Travel</Name>
                <Description>River services http://www.tfl.gov.uk/tickets/19798.aspx
You can use Oyster pay as you go to pay for journeys on KPMG Thames Clipper river services which
is cheaper than the cash fare. This does not count towards the daily price cap.
 </Description>
                <CappingPeriod>none</CappingPeriod>
                <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
                <ValidableElementRef version="any" ref="lrs:riverTrip"/>
            </CappingRule>
```

7.6.2.4.16 FareProduct: XML Example of Supplement product

The following code fragment shows a SUPPLEMENT PRODUCT that sets up permission to trigger an automatic top up of an electronic card (it doesn't itself add value to the card) The top up can be between can be between 20 or 40 GBP and can be purchased up to 8 days before collection.

For EXAMPLE:

```
    <SupplementProduct version="any" id="tfl:Oyster_Auto_top_up">
        <Name>Auto Cash Top up Payment before ride - </Name>
        <Description>Auto top-up makes sure you never run out of pay as you go credit by
automatically topping up your Oyster card with money from your credit or debit card, whenever your
pay as you go balance falls below £10.

        How to set up Auto top-up

    Create or log in to your Oyster online account
    Select 'add/renew/top-up ticket' and then 'pay as you go with Auto top-up'
    Add some pay as you go credit (minimum £10)
    Select a Auto top-up amount (£20 or £40)
    Choose a station or tram stop from the dropdown list for where you want to activate Auto top-
up
    You need to activate within eight days
        </Description>
        <ChargingMomentRef version="any" ref="tfl:prepaid"/>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <GoesOnCard>true</GoesOnCard>
            <IsPersonal>true</IsPersonal>
            <RequiresEntitlement>true</RequiresEntitlement>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <validityParameterAssignments>

            <GenericParameterAssignment version="any" id="tfl:Oyster_Auto_top_up@Req@PAYG">
                <Name>Top up a card </Name>
                <Description>You can't set up an Oyster online account if you have an Oyster
photocard, Freedom Pass or Visitor Oyster card. </Description>
            <EntitlementRequiredRef version="any" ref="tfl: PayAsYouGo_right_holder"/>
            </GenericParameterAssignment>
            <GenericParameterAssignment version="any" id="tfl:Oyster_Auto_top_up@purchase_window">
                <Description>You need to activate within eight days</Description>
                <limitations>
                    <PurchaseWindowRef version="any" ref="tfl:collect_within_8_days"/>
                </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>
        <prices>
            <FareProductPrice version="any" id="tfl:Oyster_Auto_top_up">
                <Name>Auto Top up must be between the following </Name>
                <LimitingRule " id="tfl:Oyster_Auto_top_up">
                    <MinimumPrice>20</MinimumPrice>
                    <MaximumPrice>40</MaximumPrice>
                </LimitingRule>
                <RoundingRef ref="tfl:Rounding:TopUp_in_20_pound_steps" version="any"/>
            </FareProductPrice>
        </prices>
    </SupplementProduct>
```

7.7 Pricing

7.7.1 Fare Calculation Parameters

7.7.1.1 Fare Calculation Parameters – Conceptual MODEL

There is a large variety of methods to calculate the price to be paid:

- access rights with fixed prices;
- access rights with graduated fares, to be specified when purchasing or booking;
- debiting process occurring at each validation, with fixed or graduated fares;
- combination of access rights of which the price is either calculated by addition of the elementary prices, with a possible percentage discount, or by another specific method;
- application of other discounts (e.g. according to the consumer profile);
- application of sale parameters which may influence the price (e.g. means of payment).

All these different methods can be applied to a public transport network. Often a combination of these methods will be used. There is probably no generic solution to model all possible price generation algorithms. Therefore, the data model includes a set of price entities, which provide the data necessary to calculate the price in each of the cases. Specific algorithms are responsible for applying the local price calculation rules to this basic data.

The price entities may bear some limitations as regards the amount of the price, expressed by the following attributes;

- Maximum Price (as value, as percentage, as multiple)
- Minimum Price (as value, as percentage, as multiple)
- Price

The price itself is computed applying PRICING RULEs, one of them being the DISCOUNTING RULE determined

- discount (or supplement) in percentage or
- discount (or supplement) in value.

depending on the USAGE PARAMETERs.

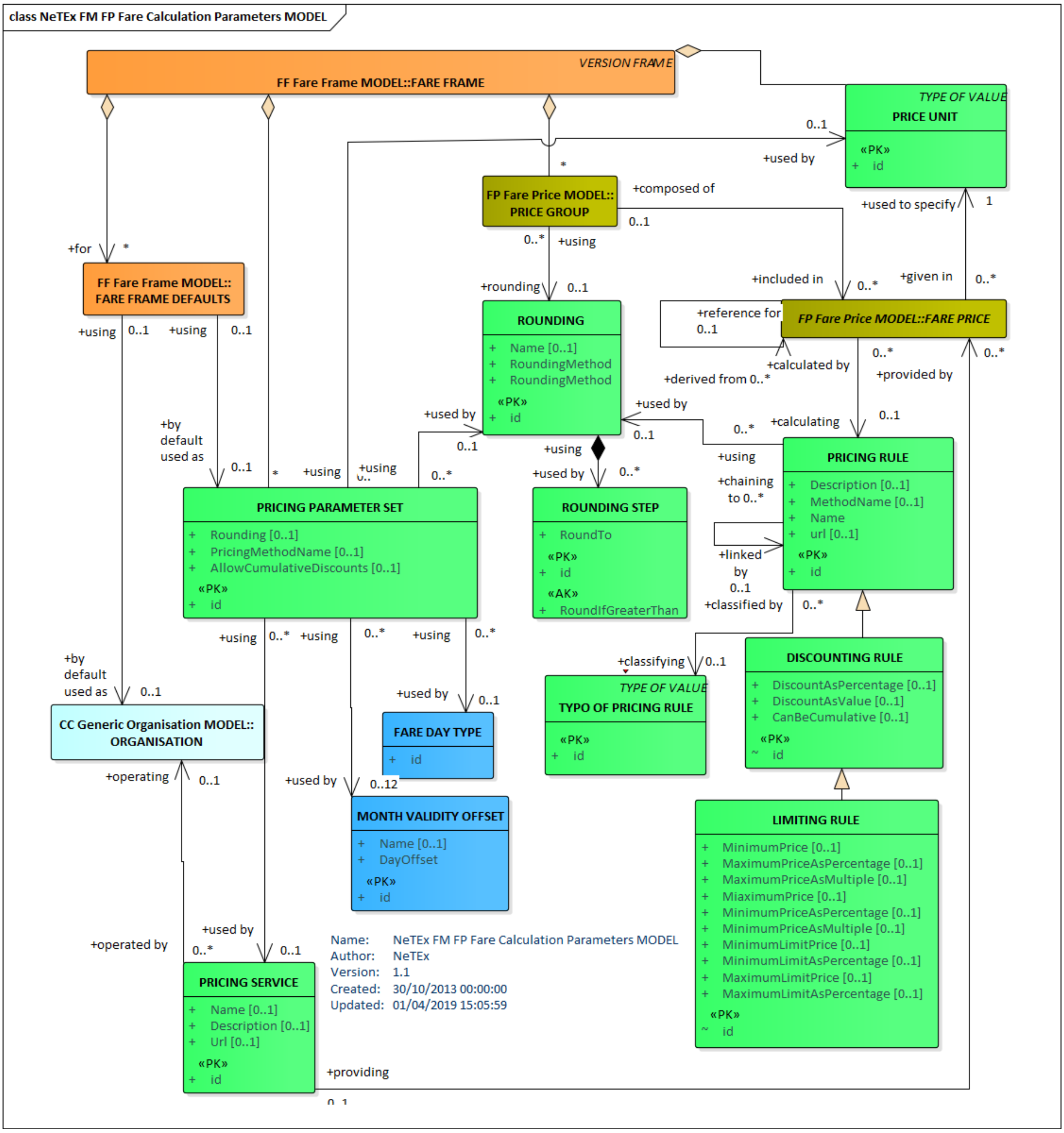


Figure 260 — Fare Calculation Parameters – Conceptual MODEL (UML)

7.7.1.2 Fare Calculation Parameters – Physical model

The following figure shows the physical model for PRICE CALCULATION PARAMETERs.

The PRICING PARAMETERs provides a container to hold various common factors that are used in pricing calculations.

- ROUNDING defines rules for rounding the results of calculations. This may be by interval or to a prescribed ROUNDING STEP.

- FARE DAY TYPE defines the day end for fare purposes.
- PRICING SERVICE indicates any dynamic service to use to fetch prices.

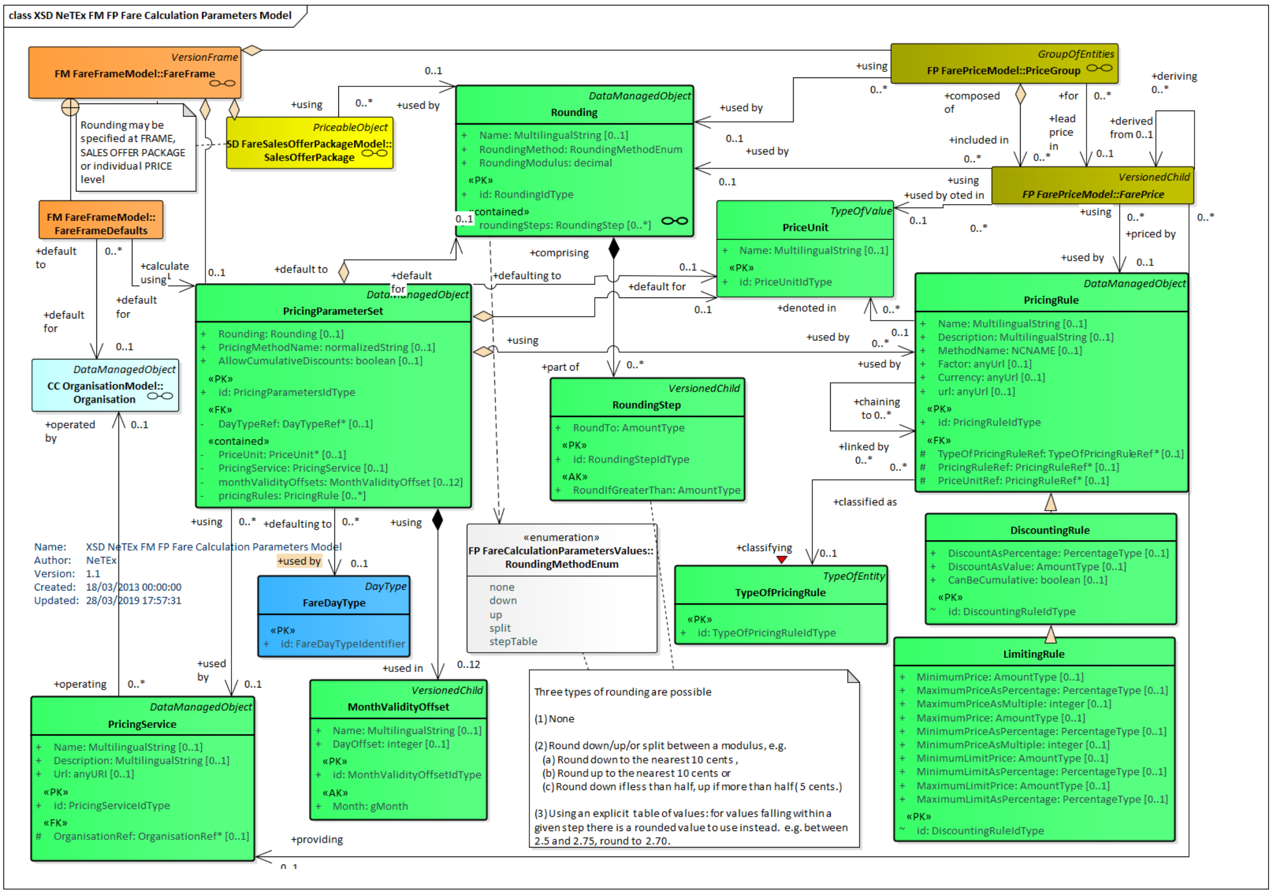


Figure 261 — Fare Calculation Parameters – Physical Model (UML)

7.7.1.3 Fare Calculation Parameters – Attributes and XSD

7.7.1.3.1 PricingParameterSet – Model Element

A set of reusable Pricing Parameters directing the rounding of values that are the result of calculations.

Table 255 – *PricingParameterSet* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*DataManagedObject*</u>	::>	PRICING PARAMETER SET inherits from DATA MANAGED OBJECT. See NeTEx Part1.
	id	*PricingParameterSet-IdType*	1:1	Identifier of PRICING PARAMETER SET.
	Name	*MultilingualString*	0:1	Name of PRICING PARAMETER SET.
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a default PRICE UNIT.
«cntd»	***priceUnits***	<u>*PriceUnit*</u>	0:*	Available PRICE UNITs.

«cntd»	***pricingRules***	*PricingRule*	0:1	PRICING RULEs available to use in pricing.
	AllowCumulative Discounts	*xsd:boolean*	0:1	Whether cumulative discounts are allowed.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Reference to a default ROUNDING.
«cntd»	***roundings***	*Rounding*	0:1	ROUNDINGs available to use in pricing.
«FK»	***DayTypeRef***	*DayTypeRef*	0:1	Default FARE DAY.
«cntd»	***monthValidity-Offsets***	*MonthValidityOffset*	0:12	Day offsets for each month in year to use to decide activation date of certain products.
«cntd»	***pricingServices***	*PricingService*	0:*	PRICING SERVICEs available to use.

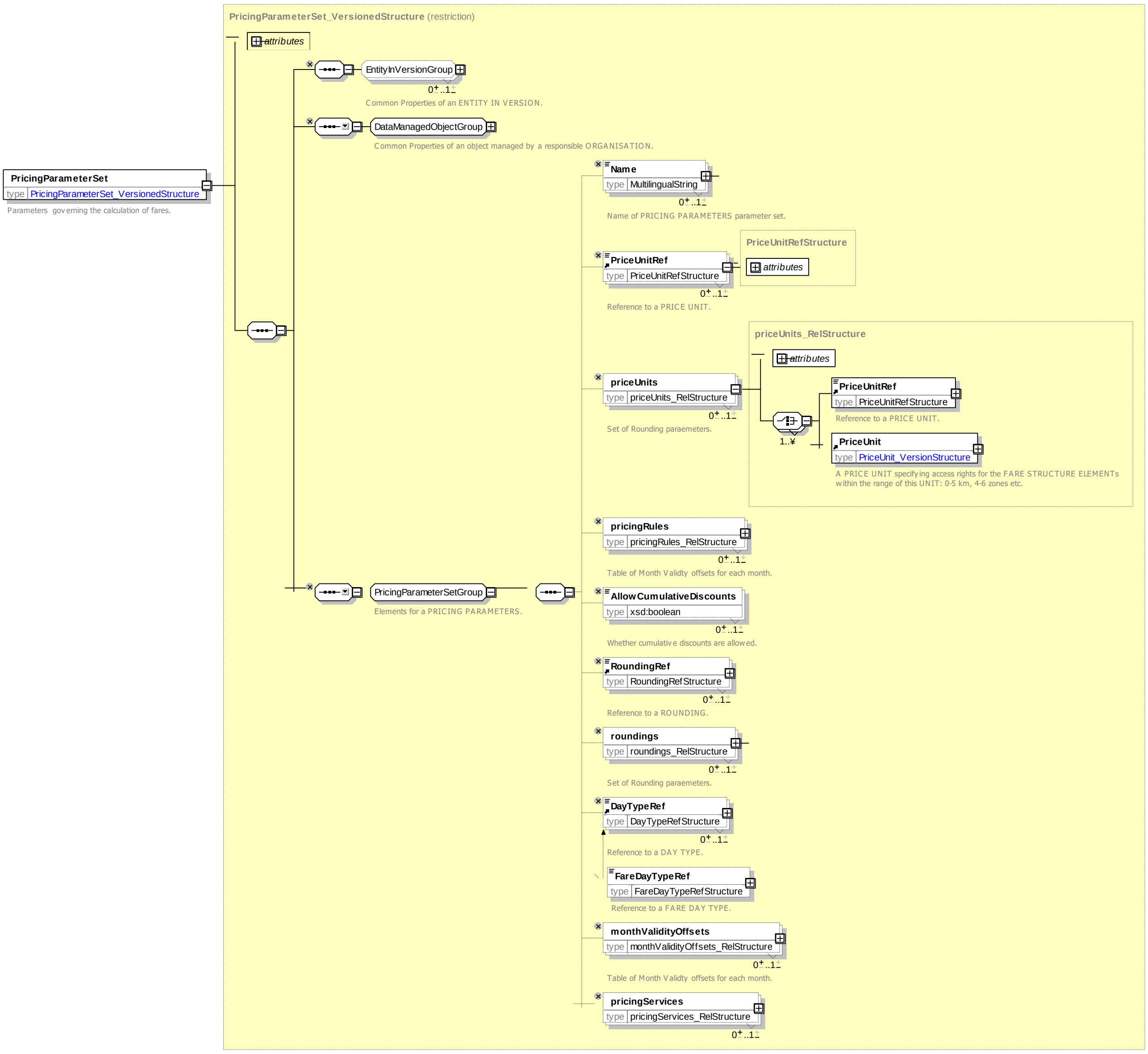


Figure 262 — *PricingParameterSet* — XSD

7.7.1.3.1.1 *MonthValidityOffset — Element*

Days before (negative) or after (positive) the start of the month that a product with a calendar period driven activation becomes valid.

Table 256 – *MonthValidityOffset* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	MONTH VALIDITY OFFSET inherits from DATA MANAGED OBJECT.
	Month	*month*	1:1	Month number
	Name	*MultilingualString*	0:1	Name of MONTH VALIDITY OFFSET.
	DayOffset	*xsd:integer*	1:1	Number days relative to start of month.

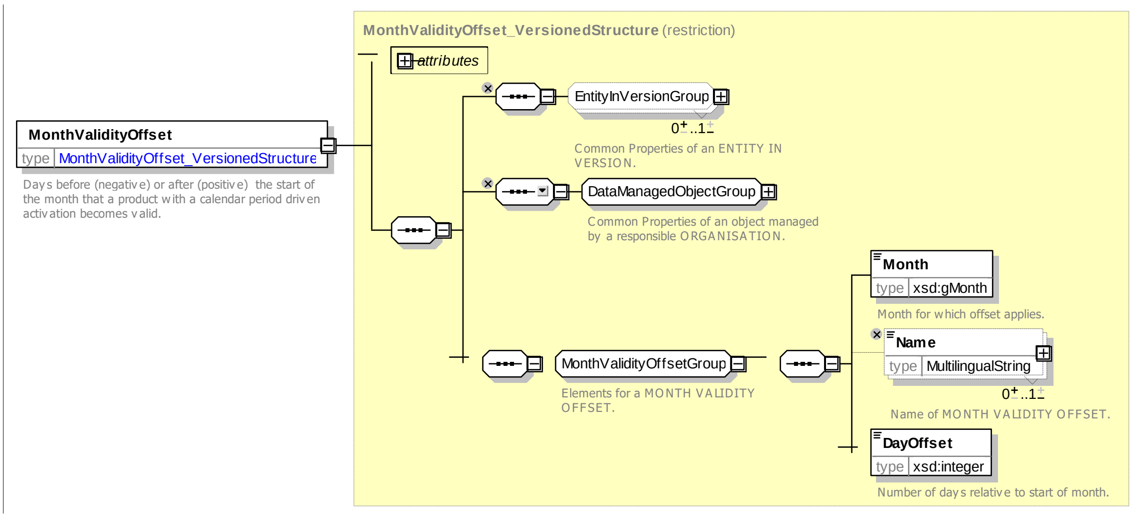


Figure 263 — *MonthValidityOffset* — XSD

7.7.1.3.2 **PricingRule – Model Element**

A named rule for compute one price from another price.

Table 257 – *PricingRule* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*DataManagedObject*	::>	PRICING RULE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PricingRuleIdType*	1:1	Identifier of PRICING RULE.
	Name	*MultilingualString*	0:1	Name of PRICING RULE.
	Description	*MultilingualString*	0:1	Description of PRICING RULE.

	MethodName	*xsd:NCNAME*	0:1	Calculation method associated with PRICING RULE.
«FK»	***TypeOfPricing-RuleRef***	*TypeOfPricing-RuleRef*	0:1	Classification of PRICING RULE. +v1.1
«FK»	***PricingRuleRef***	*PricingRuleRef+*	0:1	PRICING RULE to chain to from this one.
	Factor	*xsd:decimal*	0:1	Numeric factor associated with PRICING RULE.
	Currency	*CurrencyType*	0:1	Currency associated with PRICING RULE. +v1.1
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	PRICE UNIT for PRICING RULE. +v1.1
	url	*xsd:anyURI*	0:1	URL associated with PRICING RULE.

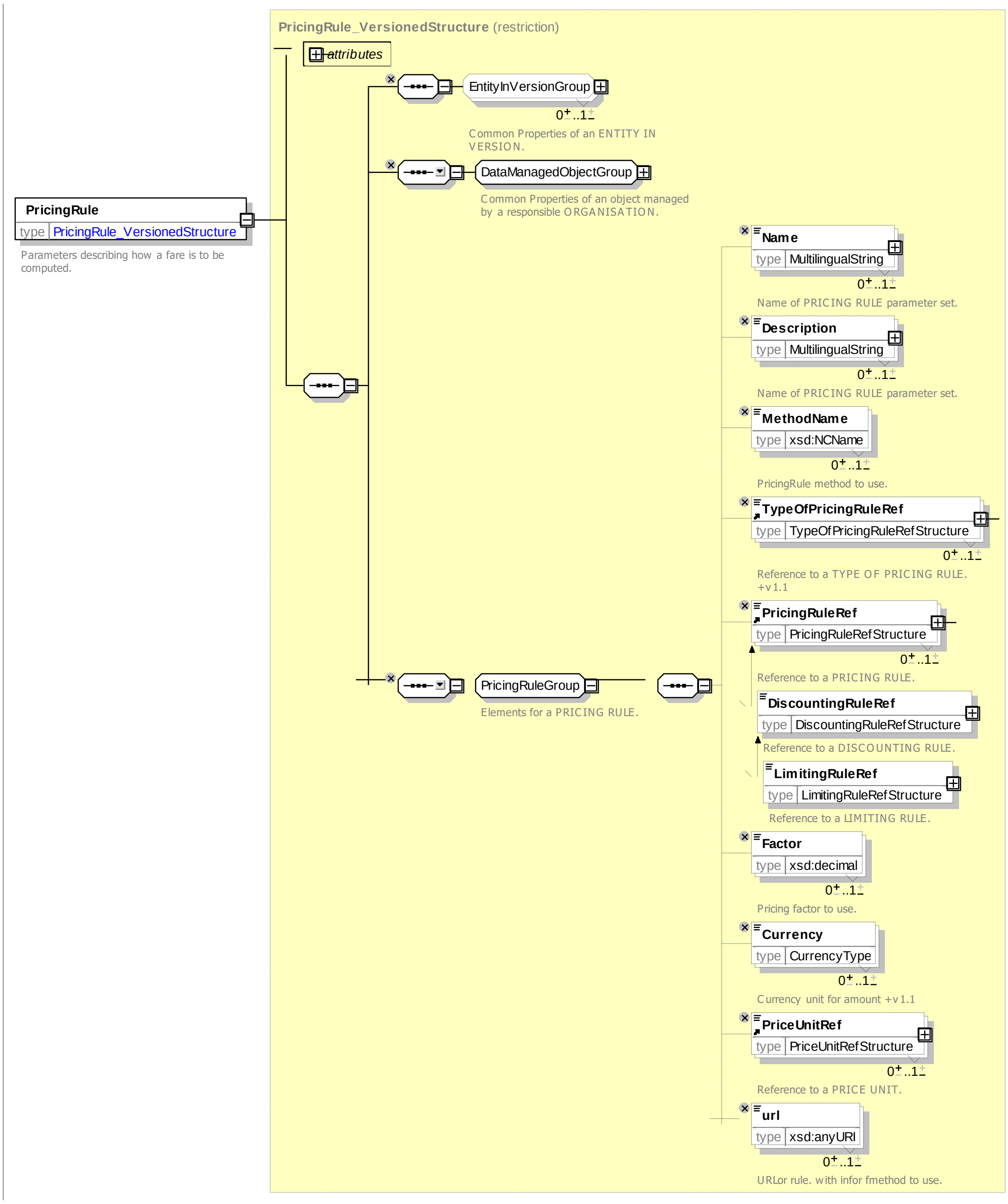


Figure 264 — *PricingRule* — XSD

7.7.1.3.3 DiscountingRule – Model Element

Parameters of a rule for computing a discounted price from another price.

Table 258 – *DiscountingRule* – Element

Classifi-cation	Name	Type	Cardinality	Description

::>	::>	*PricingRule*	::>	DISCOUNTING RULE inherits from PRICING RULE.
«PK»	***id***	*DiscountingRuleIdType*	1:1	Identifier of DISCOUNTING RULE.
	Discount-AsPercentage	*PercentageType*	0:1	Discount of PRICE as a percentage.
	DiscountAsValue	*AmountType*	0:1	Discount of PRICE as a value.
	CanBe-Cumulative	*xsd:boolean*	0:1	Whether discount can be used cumulatively in combination with other discounts.

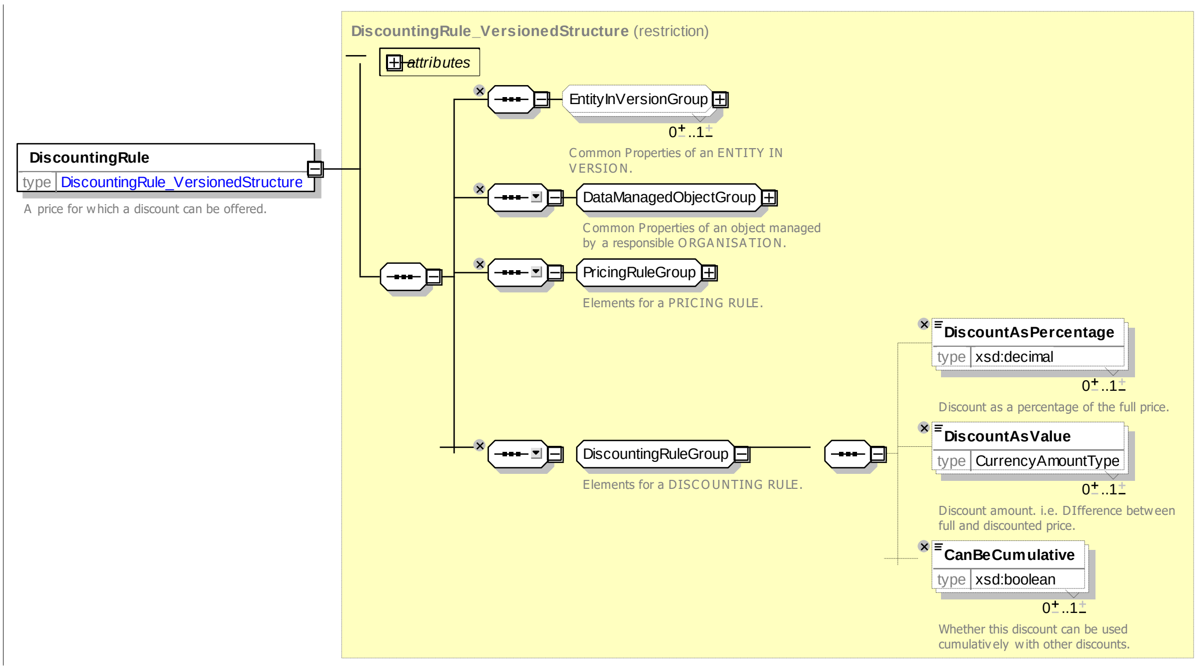


Figure 265 — *DiscountingRule* — XSD

7.7.1.3.4 LimitingRule – Model Element

Parameters of a rule for computing a price from another price subject to minima or maxima.

Table 259 – *LimitingRule* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*DiscountingRule*	::>	LIMITING RULE from DISCOUNTING RULE.
«PK»	***id***	*LimitingRuleIdType*	1:1	Identifier of LIMITING RULE.
XGRP	***LimitingRule-CappingGroup***	***xmlGroup***	1:1	Capping elements for LIMITING RULE.
XGRP	***LimitingRuleGroup***	***xmlGroup***	1:1	Limit elements for LIMITING RULE.

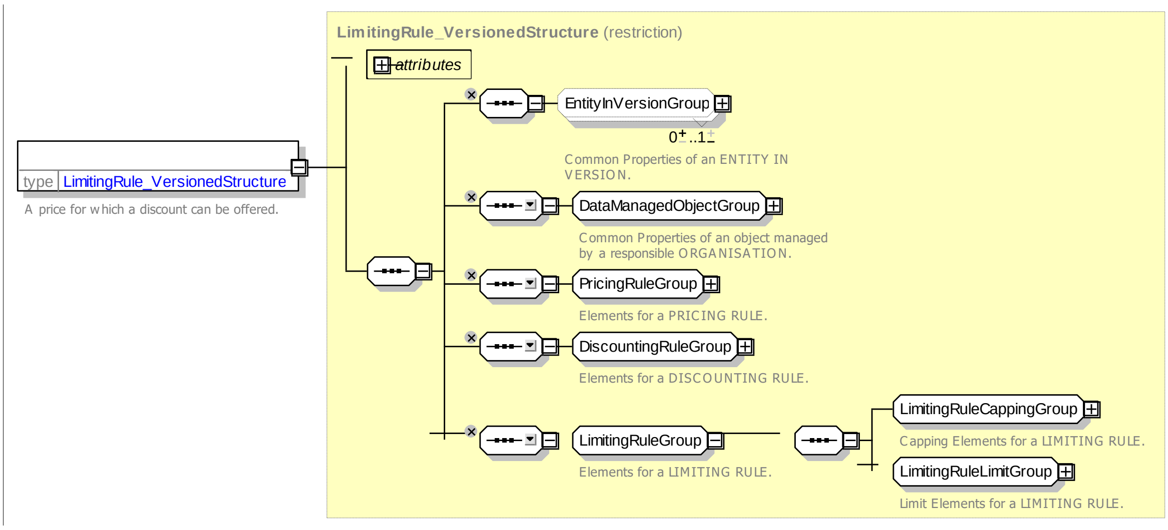


Figure 266 — *LimitingRule* — XSD

7.7.1.3.4.1 LimitingRuleCappingGroup – Group

The ***LimitingRuleCappingGroup*** specifies minimum and maximum prices for a LIMITING RULE. A calculated price below or above a limit must be adjusted to this value.

Table 260 – *LimitingRuleCappingGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	MinimumPrice	*AmountType*	0:1	Minimum amount at which to cap discounted fare.
	MinimumPrice-AsPercentage	*PercentageType*	0:1	Minimum PRICE expressed as a percentage of the total price.
	MinimumPrice-AsMultiple	*PercentageType*	0:1	Minimum PRICE expressed as a multiple of a unit fare.
	MaximumPrice	*AmountType*	0:1	Maximum amount at which to cap discounted fare.
	MaximumPrice-AsPercentage	*PercentageType*	0:1	Maximum PRICE expressed as a percentage of the total price.
	MaximumPrice-AsMultiple	*PercentageType*	0:1	Maximum PRICE expressed as a multiple of a unit fare.

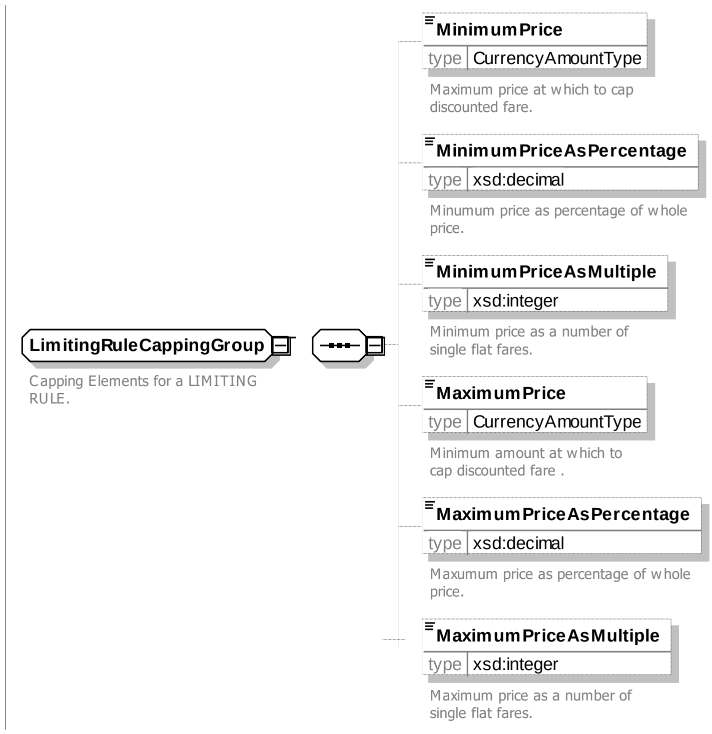


Figure 267 — *LimitingRuleCappingGroup* — XSD

7.7.1.3.4.2 LimitingRuleLimitGroup – Group

The ***LimitingRuleLimitGroup*** specifies pricing limits for a LIMITING RULE. A limit price set an additional boundary on allowed prices for providing a product. Found for example in Tap TSI B.3

Table 261 – *LimitingRuleLimitGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Minimum-LimitPrice	*AmountType*	0:1	Limiting amount below which resulting fare may not be sold.
	MinimumLimit-AsPercentage	*PercentageType*	0:1	Minimum limit expressed as a percentage of the total price.
	MaximumLimit-Price	*AmountType*	0:1	Limiting amount above which resulting fare may not be sold.
	MaximumLimit-AsPercentage	*PercentageType*	0:1	Maximum limit expressed as a percentage of the total price.

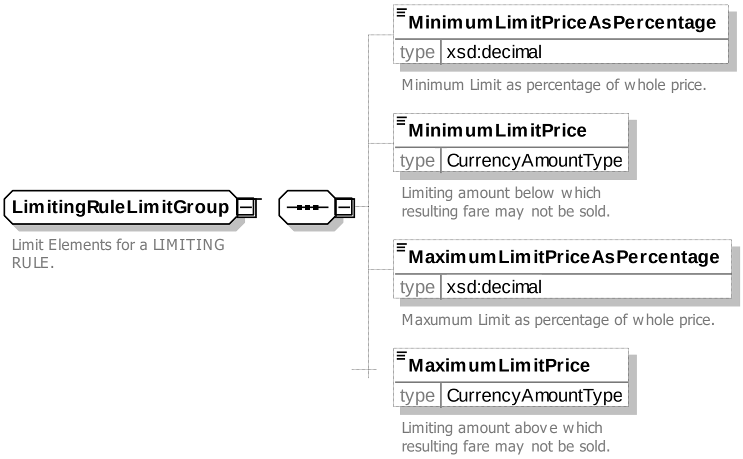


Figure 268 — *LimitingRuleLimitGroup* — XSD

7.7.1.3.5 Rounding – Model Element

Parameters directing the rounding of values that are the result of calculations.

Table 262 – *Rounding* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	ROUNDING inherits from DATA MANAGED OBJECT. See NeTEx Part1.
	id	*RoundingIdType*	1:1	Identifier of ROUNDING.
	Name	*MultilingualString*	0:1	Name of ROUNDING.
«enum»	***RoundingMethod***	*RoundingMethodEnum*	0:1	Method to use to *round: down, up, split, none.* See allowed values below.
	Rounding-Modulus	*decimal*	0:1	Amount by which rounding is to be quantised, i.e. results should be quantised to nearest whole multiple of this value, for example, 0.10, 0.20, 0.30 cents, or 1.00 Euro, 1.6 Euro, etc.
«cntd»	***roundingSteps***	*RoundingStep*	0:*	Table of explicit ROUNDING STEPs.

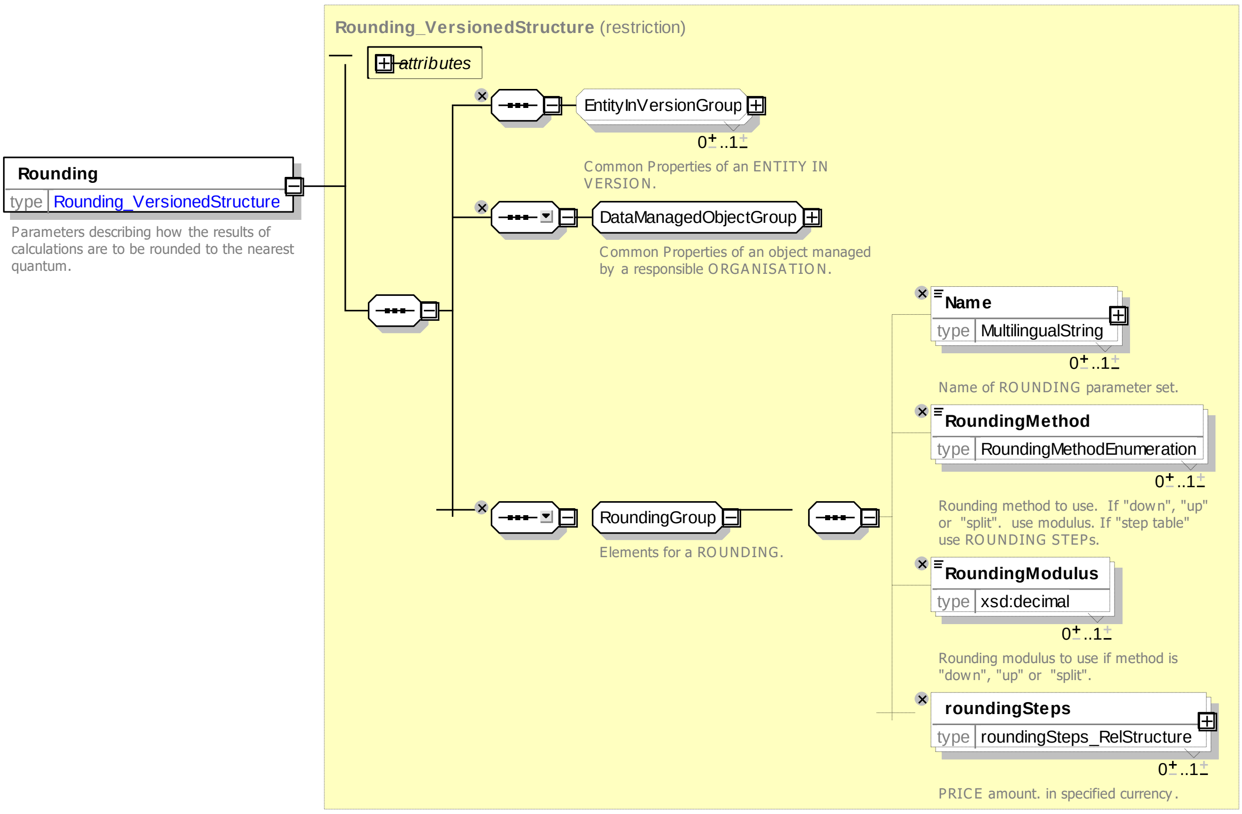


Figure 269 — *Rounding* — XSD

7.7.1.3.5.1 RoundingStep – Model Element

A rounding step to use to round a range of values. If step table rounding is used, any value larger than the step key and smaller that the next step key should be rounded to the 'Round To' value.

Table 263 – *RoundingStep* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	ROUNDING STEP inherits from VERSIONED CHILD. See NeTEx Part1.
	id	*RoundingStepIdType*	1:1	Identifier of ROUNDING STEP.
«FK»	***RoundingRef***	*RoundingRef*	0:1	ROUNDING element to which ROUNDING STEP belongs. If not given by context must be specified.
«PK»	***RoundIf-GreaterThan***	*AmountType*	1:1	Start value for range; round if result value greater than range key and less than range key of next step table value.
	RoundTo	*AmountType*	1:1	Value to which to round.

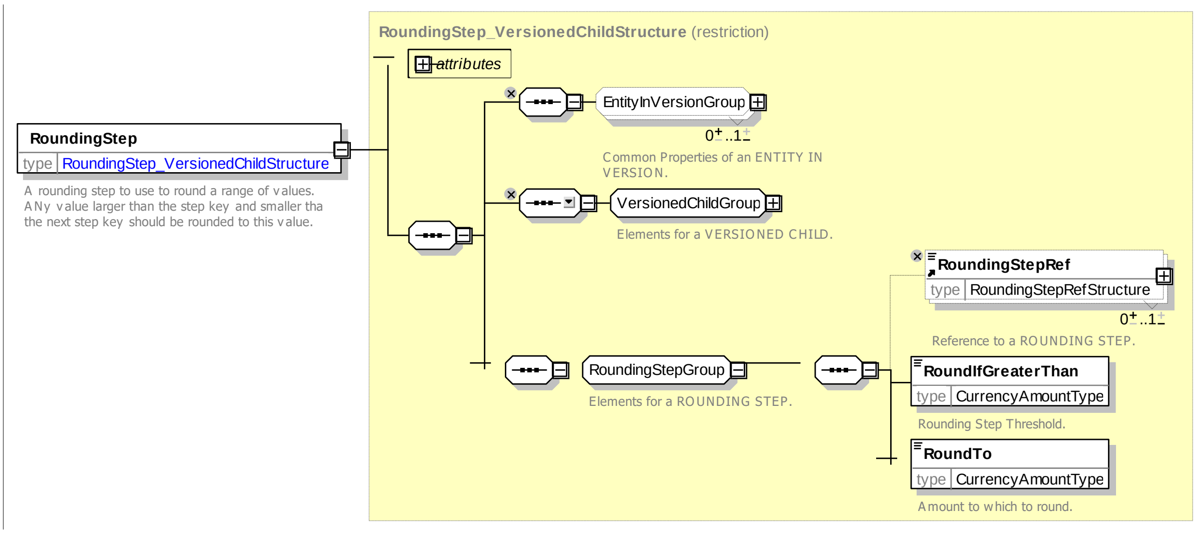


Figure 270 — *RoundingStep* — XSD

7.7.1.3.6 FareDayType – Model Element

A type of day used in the fare collection domain, characterised by one or more properties which affect the definition of access rights and prices in the fare system.

Table 264 – *FareDayType* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DayType*	::>	FARE DAY TYPE inherits from DAY TYPE.
	id	*FareDayTypeIdType*	1:1	Identifier of FARE DAY TYPE.

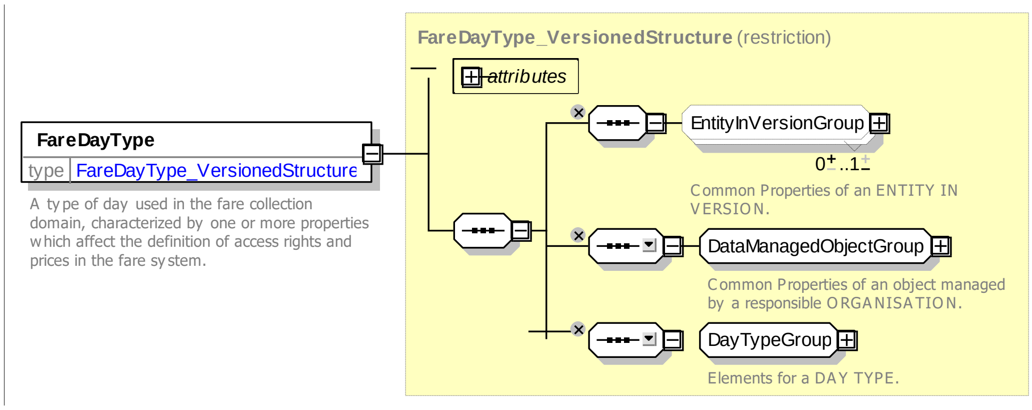


Figure 271 — *FareDayType* — XSD

7.7.1.3.7 PricingService – Model Element

A web service used to provide prices dynamically at time of booking or purchase.

Table 265 – *PricingService* – Element

Classifi-cation	Name	Type	Cardinality	Description

::>	::>	*DataManagedObject*	::>	PRICING SERVICE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PricingServiceIdType*	1:1	Identifier of PRICING SERVICE.
	Name	*MultilingualString*	0:1	Name of PRICING SERVICE.
	Description	*MultilingualString*	0:1	Description of PRICING SERVICE.
	Url	*xsd:anyURI*	0:1	Url at which service is available.
«FK»	***OrganisationRef***	*(OrganisationRef)*	0:1	ORGANISATION that provides service. has contact details etc.

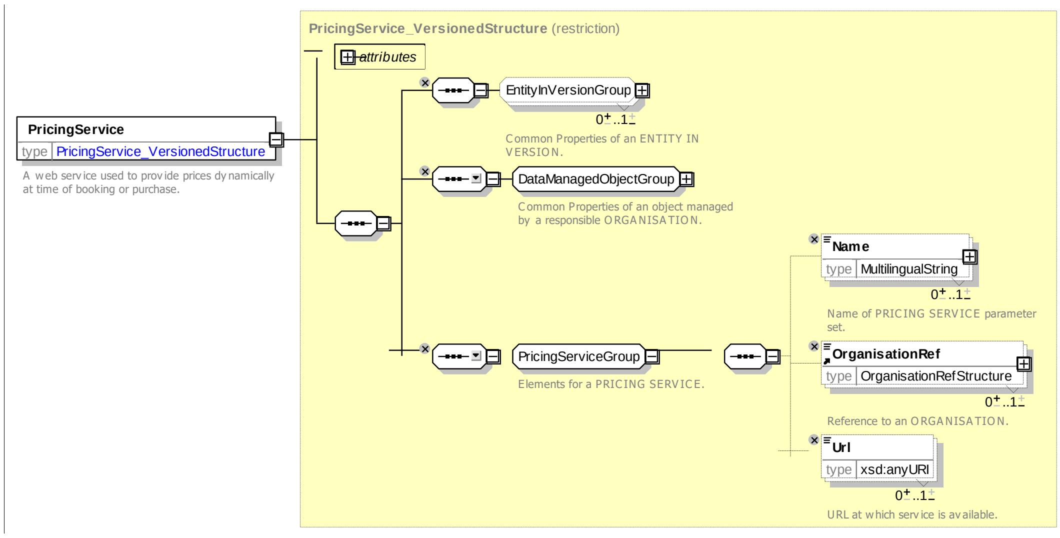


Figure 272 — *PricingService* — XSD

7.7.1.3.7.1 RoundingMethod – Allowed values

The following table shows the allowed values for ***RoundingMethod (****RoundingMethodEnumeration****)***.

Table 266 – RoundingMethod – Allowed values

Value	Description
none	No rounding.
down	Round down using rounding modulus.
up	Round up using rounding modulus.
split	Round down if below 0.5 up if above 0.5.
stepTable	Use a rounding table.

7.7.1.4 Fare Calculation Parameters – XML examples

7.7.1.4.1 Fare Calculation Parameters: XML Example of PricingRules

The following code fragment shows a PRICING PARAMETER SET with two discounting rules and a ROUNDING to round to the dearest 0.50 units.

For EXAMPLE:

```xml
<!--- ==== Pricing Parameters======= -->
    <PricingParameterSet version="any" id="cdla::tap">
        <pricingRules>
            <DiscountingRule version="any" id="cdla::£0-Off">
                <DiscountAsValue>0</DiscountAsValue>
            </DiscountingRule>
            <DiscountingRule version="any" id="cdla::25%">
                <DiscountAsPercentage>25</ DiscountAsPercentage >
            </DiscountingRule>

        </pricingRules>
            <Rounding id="tap:Rounding:split" version="any">
                <RoundingMethod>split</RoundingMethod>
                <RoundingModulus>0.50</RoundingModulus>
            </Rounding>
    </PricingParameterSet>
```

7.7.1.4.2 Fare Calculation Parameters: XML Example of PricingRules

The following code fragment shows a PRICING PARAMETER SET with some DISCOUNTING and LIMITING RULEs and some ROUNDING rules:

```xml
<PricingParameterSet version="any" id="tfl:tfl">
    <pricingRules>
        <PricingRule version="any" id="tfl:2x">
            <Name>Return twice single</Name>
            <Factor>2</Factor>
        </PricingRule>
        <DiscountingRule version="any" id="tfl:100%">
            <DiscountAsPercentage>100</DiscountAsPercentage>
        </DiscountingRule>
        <DiscountingRule version="any" id="tfl:50%">
            <DiscountAsPercentage>50</DiscountAsPercentage>
        </DiscountingRule>
        <DiscountingRule version="any" id="tfl:34%">
            <DiscountAsPercentage>34</DiscountAsPercentage>
        </DiscountingRule>
        <LimitingRule version="any" id="tfl:5-50">
            <MinimumPrice>5</MinimumPrice>
            <MaximumPrice>50</MaximumPrice>
        </LimitingRule>
        <LimitingRule version="any" id="tfl:20-40">
            <MinimumPrice>20</MinimumPrice>
            <MaximumPrice>40</MaximumPrice>
        </LimitingRule>
        <LimitingRule version="any" id="tfl:max-4.40">
            <MaximumLimitPrice>4.40</MaximumLimitPrice>
        </LimitingRule>
        <LimitingRule version="any" id="tfl:max-7.00">
            <MaximumLimitPrice>7.00</MaximumLimitPrice>
        </LimitingRule>
        <LimitingRule version="any" id="tfl:max-7.70">
            <MaximumLimitPrice>7.70</MaximumLimitPrice>
        </LimitingRule>
    </pricingRules>

        <roundings>
            <Rounding id="tfl:TopUpIn5PoundSteps" version="any">
                <Name>Oyster Top ups must be in in £5 Multiples</Name>
                <RoundingMethod>up</RoundingMethod>
                <RoundingModulus>5</RoundingModulus>
            </Rounding>
            <Rounding id="tfl:TopUpIn20PoundSteps" version="any">
                <Name>Oyster AUTO Top ups must be in in £20 Multiples</Name>
                <RoundingMethod>up</RoundingMethod>
                <RoundingModulus>20</RoundingModulus>
            </Rounding>
        </roundings>
        <FareDayTypeRef version="any" ref="tfl:DayFrom0430"/>
    </PricingParameterSet>
```

7.7.2 Fare Price

7.7.2.1 Fare Price – Conceptual MODEL

The FARE PRICE Model allows fares to be defined for fare structure elements.

An element which can have a price is a specialization of PRICEABLE OBJECT.

There are different types of FARE PRICE for each PRICEABLE OBJECT, for example DISTANCE MATRIX ELEMENT PRICE, FARE PRODUCT PRICE, etc., etc.

FARE PRICEs can be in any PRICE UNIT (currency or otherwise) and can have a start date.

FARE PRICEs may be an absolute amount (e.g. 23.00 Euros) or be derived as using a PRICING RULE from another price. The FARE PRICE may indicate the price and rule from which it is derived as well as the resulting amount.

- A DISCOUNTING RULE specifies parameters relating to discounting; discounts may be specified as either a percentage (e.g. 10%) or an absolute amount (e.g. 5 Euros).

- A LIMITING RULE may be used to set may be set on the results, for example to set a minimum and maximum price.

More than one rule may be applied successively to derive a price; PRICING RULES may be chained together. Where a price is derived from another price, the intermediate derivation steps may be recorded using a RULE STEP RESULT. For example, a concessionary price might be calculated from a full fare by applying a discount, then deducting tax.

A price entity is defined for each PRICEABLE OBJECT,

- CONTROLLABLE ELEMENT PRICE;
- FARE STRUCTURE ELEMENT PRICE;
- GEOGRAPHICAL INTERVAL PRICE;
- DISTANCE MATRIX PRICE;
- TIME INTERVAL PRICE;
- VALIDABLE ELEMENT PRICE;
- USAGE PARAMETER PRICE;
- FARE PRODUCT PRICE;
- SALES OFFER PACKAGE PRICE,
- FULFILMENT METHOD PRICE,
- Etc (see figure below)

It may be necessary to group price entities into PRICE GROUPs, in order:

- to group all possible access rights or products into a few categories, each of them having a price reference (products of price ‘A’, ‘B’, etc.); this provides a reminder for the users;
- to group prices into categories to which the same increase, in value or percentage, may be applied.

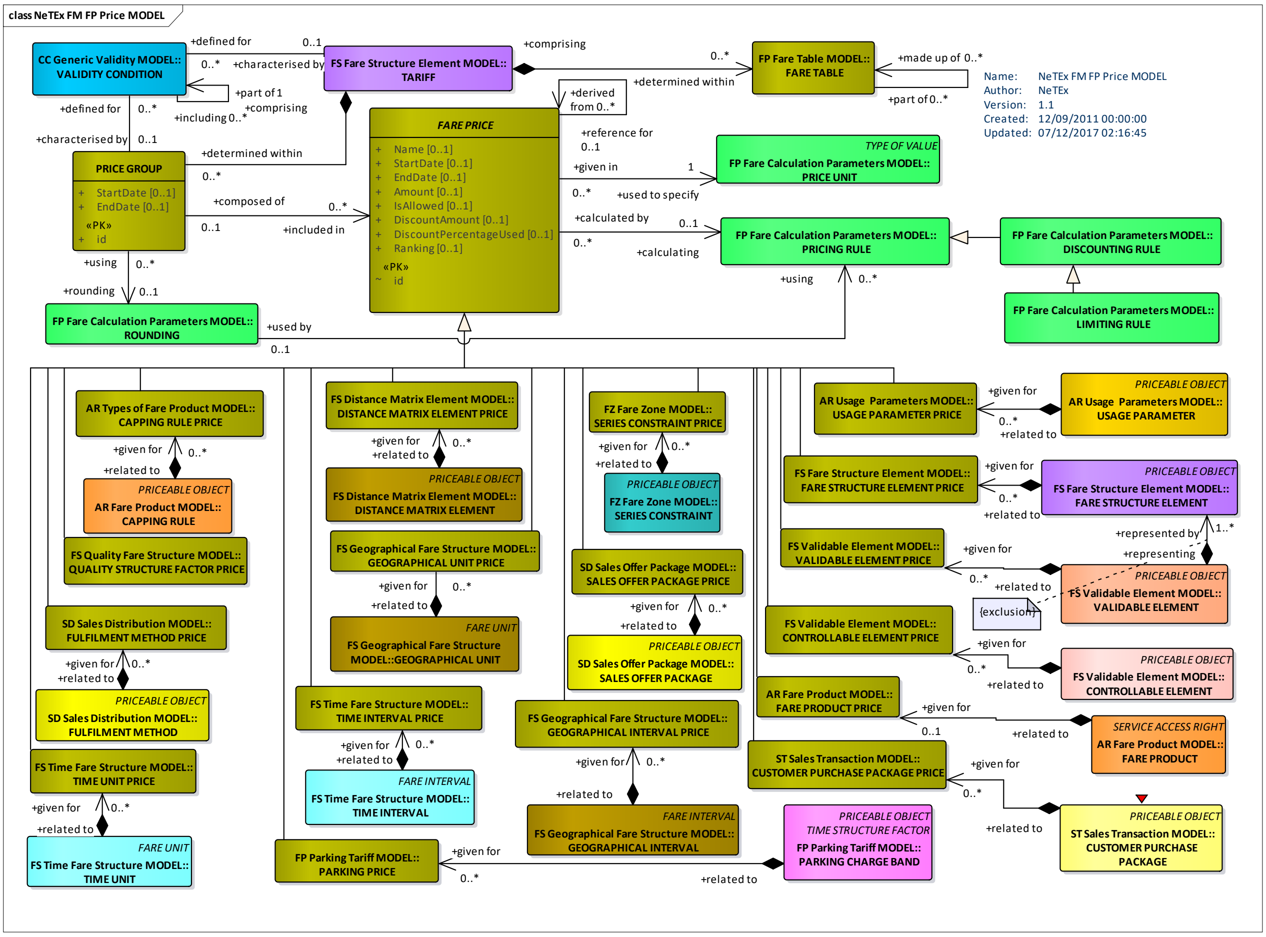


Figure 273 —Priceable Object – Conceptual MODEL (UML)

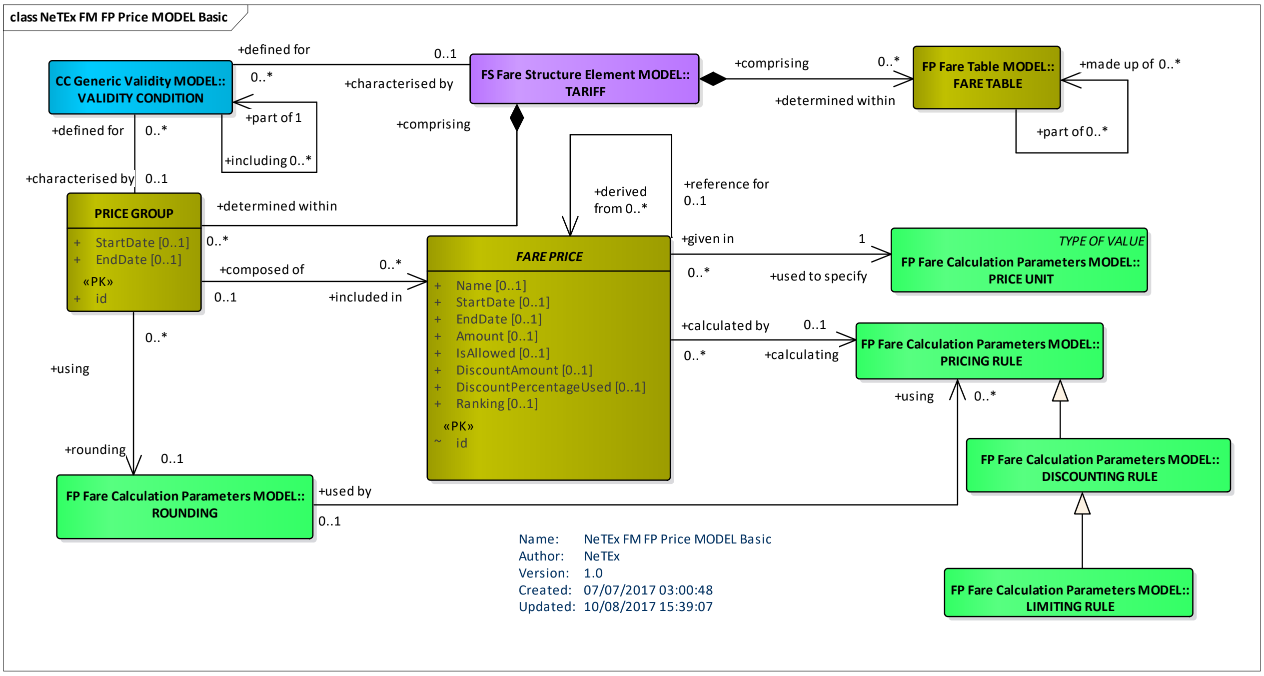


Figure 274 —Price – Conceptual MODEL (UML)

7.7.2.2 Fare Price – Physical model

The following figure shows the overall physical model for FARE PRICEs.

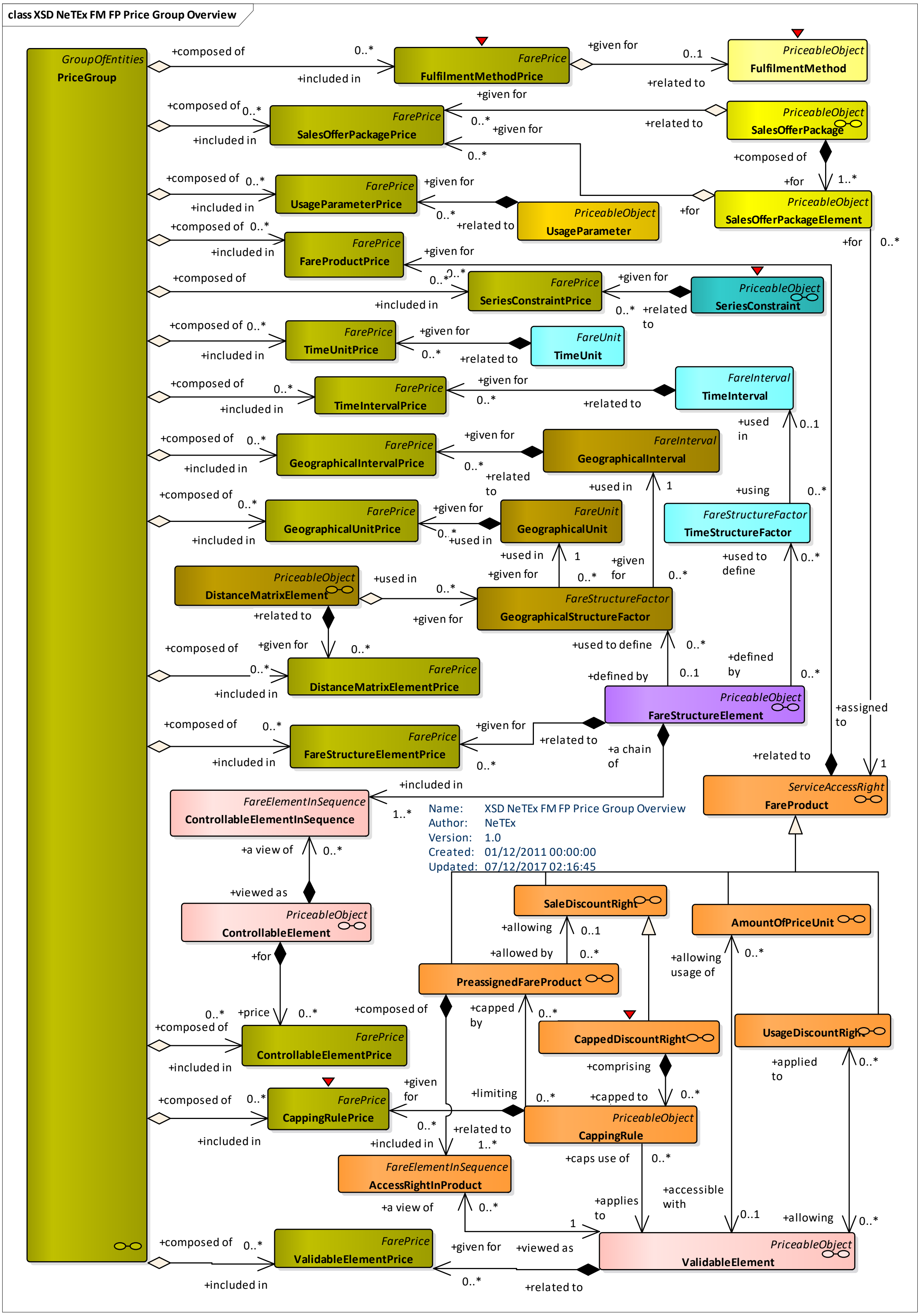


Figure 275 —Price Group Overview– Physical Model (UML)

7.7.2.3 Fare Price Details – Physical model

The following figure shows the details of the physical model for FARE PRICEs.

Basic price details are specified on FARE PRICE.

In normal practice certain prices will be explicitly stated, for example TIME INTERVAL PRICE, GEOGRAPHICAL INTERVAL PRICE, DISTANCE MATRIX ELEMENT PRICE, etc, and others will be derived from these basic prices using a PRICING RULE

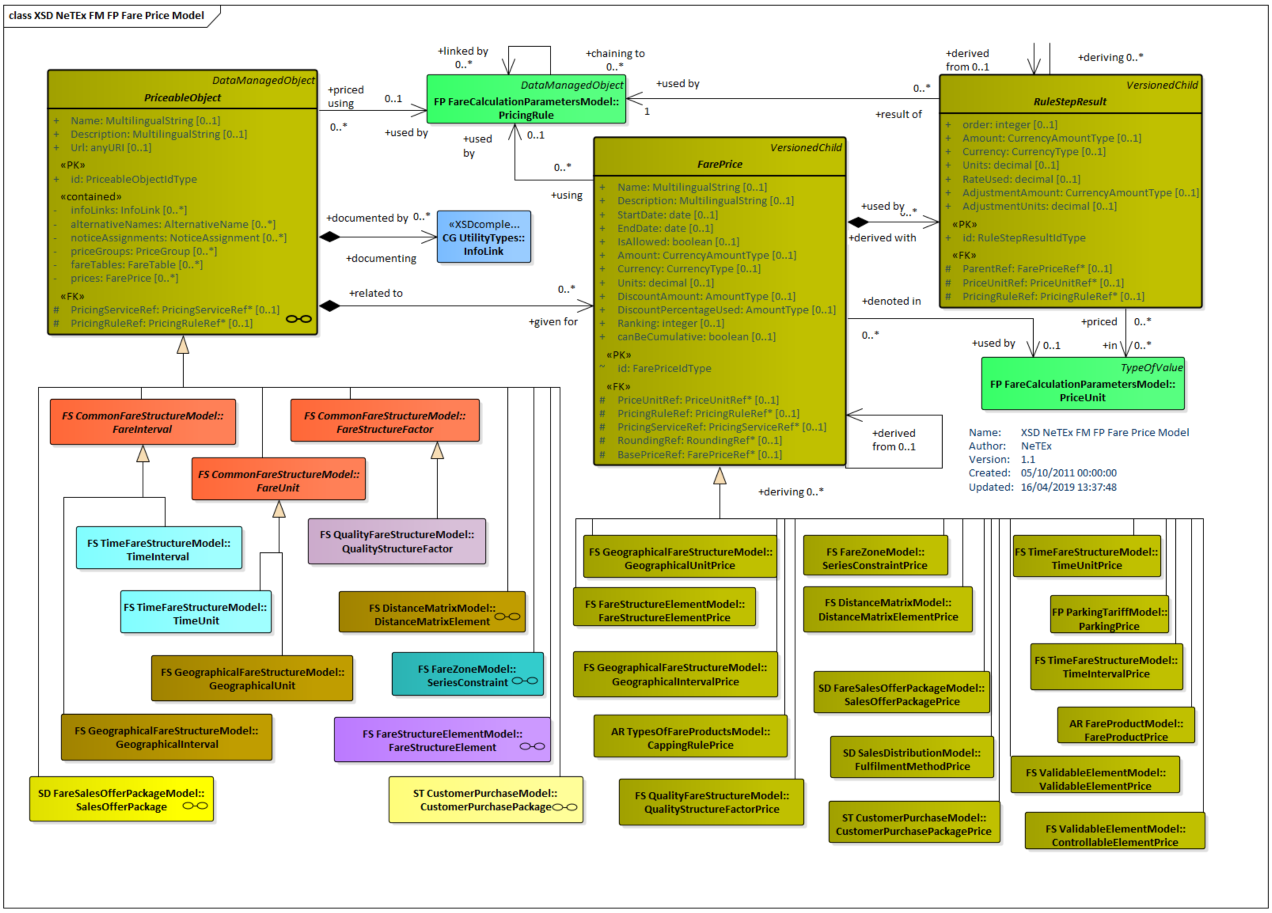


Figure 276 —Price Group – Physical Model (UML)

7.7.2.4 Fare Price – Attributes and XSD

7.7.2.4.1 PriceUnit – Model Element

A unit to express prices: amount of currency, abstract fare unit, ticket unit or token etc.

Table 267 – *PriceUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*TypeOfValue*</u>	::>	PRICE UNIT inherits from TYPE OF VALUE.
«PK»	***id***	*PriceUnitIdType*	1:1	Identifier of PRICE UNIT.

	Precision	*xsd:integer*	0:1	Precision of PRICE UNIT.

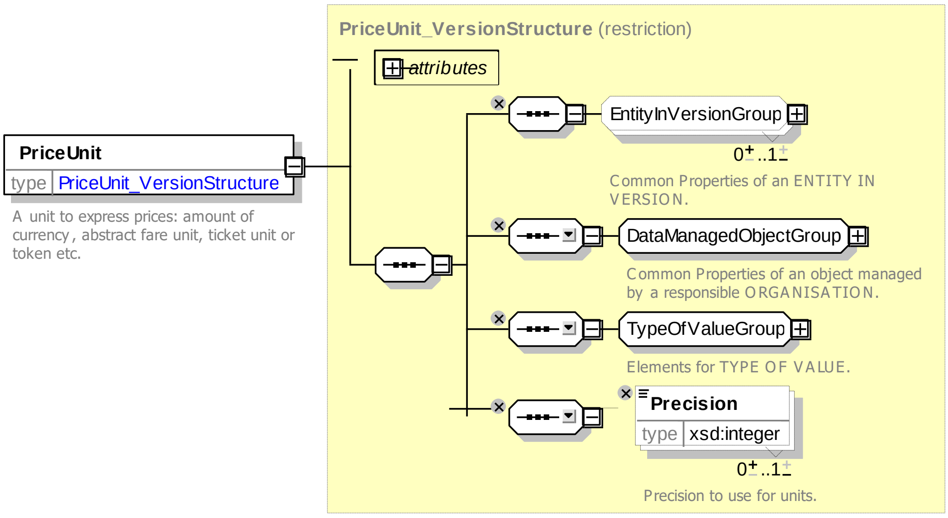


Figure 277 — *PriceUnit* — XSD

7.7.2.4.2 FarePrice – Model Element

A set of all possible price features for a Fare element.

Table 268 – *FarePrice* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*VersionedChild*	::>	FARE PRICE inherits from VERSIONED CHILD
«PK»	***id***	*FarePriceIdType*	1:1	Identifier of FARE PRICE.
	Name	*MultilingualString*	0:1	Name of PRICE.
	Description	*MultilingualString*	0:1	Description of PRICE. +v1.1
	PrivateCode	*PrivateCode*	0:1	External identifier of PRICE. +v1.1
	StartDate	*xsd:date*	0:1	Start date for PRICE validity.
	EndDate	*xsd:date*	0:1	End date for PRICE validity.
	Amount	*AmountType*	0:1	Price in a specified currency.
	Currency	*CurrencyType*	0:1	Currency ISO 4217 code (This in an optimization to allow PRICE UNITs to be omitted).
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a PRICE UNIT; may be a currency.
	Units	*xsd:decimal*	0:1	Amount in designated unit.
«cntd»	***ruleStepResults***	*RuleStepResult*	0:*	RULE STEP RESULTS describing derivation of price.

	IsAllowed	*xsd:boolean*	0:1	Whether the FARE PRICE is allowed. The default is '*true*'.
«FK»	***PricingServiceRef***	*PricingServiceRef*	0:1	Reference to a PRICE SERVICE which can provide / provided price.
XGRP	***FarePriceCalculation-Group***	***xmlGroup***	0:1	Elements governing the calculation of prices.
	Ranking	*xsd:integer*	0:1	Relative ranking of price relative to other prices.

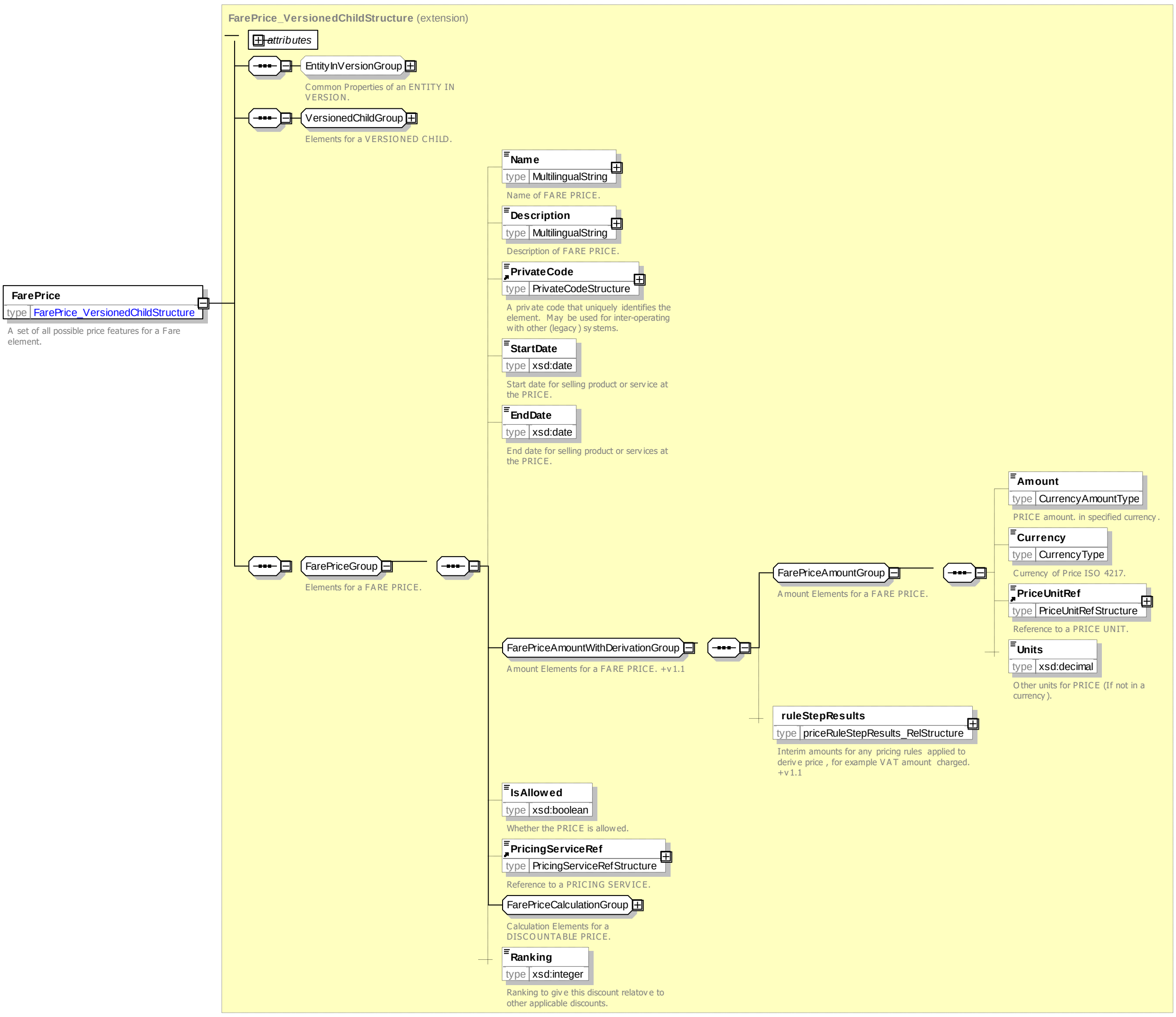


Figure 278 — *FarePrice* — XSD

7.7.2.4.2.1 FarePriceCalculationGroup – Group

The ***FarePriceCalculationGroup*** specifies attributes affecting the calculation of a price from another price.

Table 269 – *FarePriceCalculationGroup* – Group

Classifi-cation	Name		Type	Cardinality	Description
«FK»	***FarePriceRef***		*FarePriceRef+*	0:1	Reference to a FARE PRICE from which this fare price is derived using a PRICING RULE.
			choice		
«FK»	***a***	***PricingRule-Ref***	*PricingRuleRef+*	0:1	Reference to a PRICING RULE used to derive price.
«cntd»	b	***PricingRule***	*PricingRule*	0:1	PRICING RULE used to derive price.
	CanBe-Cumulative		*xsd:boolean*	0:1	Whether discount can be used cumulatively in combination with other discounts.
«FK»	***RoundingRef***		*RoundingRef*	0:1	Rounding to use on calculation.

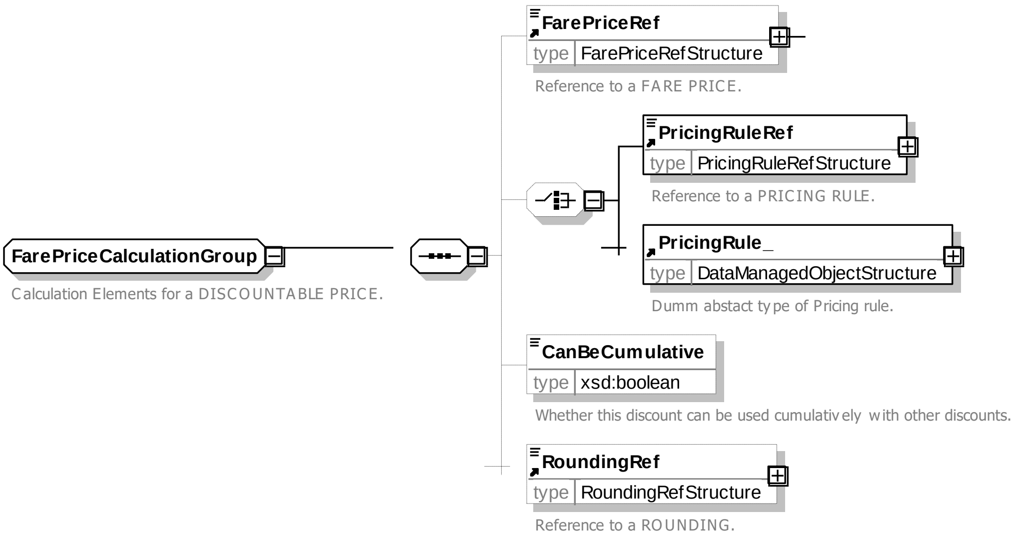


Figure 279 — *FarePriceCalculationGroup* — XSD

7.7.2.4.2.2 RuleStepResult – Element

The ***RuleStepResult*** holds one or more intermediate steps relating to the calculation of a price from another price., for example the tax or a discount that has been applied.

Table 270 – *RuleStepResult* – Element

Classifi-cation	Name	Type	Cardinality	Description
«PK»	***id***	*RuleStepResultIdType*	1:1	Identifier of PRICE RULE STEP RESULT.
«atr»	***order***	*xsd:integer*	1:1	Order in which step was done.

«FK»	***FarePriceRef***	*FarePriceRef+*	0:1	Reference to a FARE PRICE from which this fare price is derived using a PRICING RULE
	Amount	*AmountType*	0:1	Price in a specified currency.
	Currency	*CurrencyType*	0:1	Currency ISO 4217 code (This in an optimization to allow PRICE UNITs to be omitted).
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a PRICE UNIT.
	Units	*xsd:decimal*	0:1	Amount in designated unit.
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a PRICE UNIT; may be a currency.
	RateUsed	*xsd:decimal*	*0:1*	Discount rate used.
	Adjustment-Amount	*CurrencyType*	0:1	Step calculation amount, in same currency as STEP RESULT Amount. (i.e. difference between Base Amount and result Amount. PRICE) +v1.1
	AdjustmentUnits	*xsd:decimal*	*0:1*	Step calculation Units, in same PRICE UNIT as STEP RESULT Amount.(i.e. difference between base Units and Result Units. +v1.1
«FK»	***PricingRuleRef***	*PricingRuleRef+*	0:1	Reference to a PRICING RULE used in calculation step to derive price.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Reference to a ROUNDING method used in calculation step to derive price.
«FK»	***RoundingStep-Ref***	*RoundingStepRef*	0:1	Reference to a ROUNDING STEP used in calculation step to derive price.
	Narrative	*MultilingualString*	0:1	Textual explanation of calculation

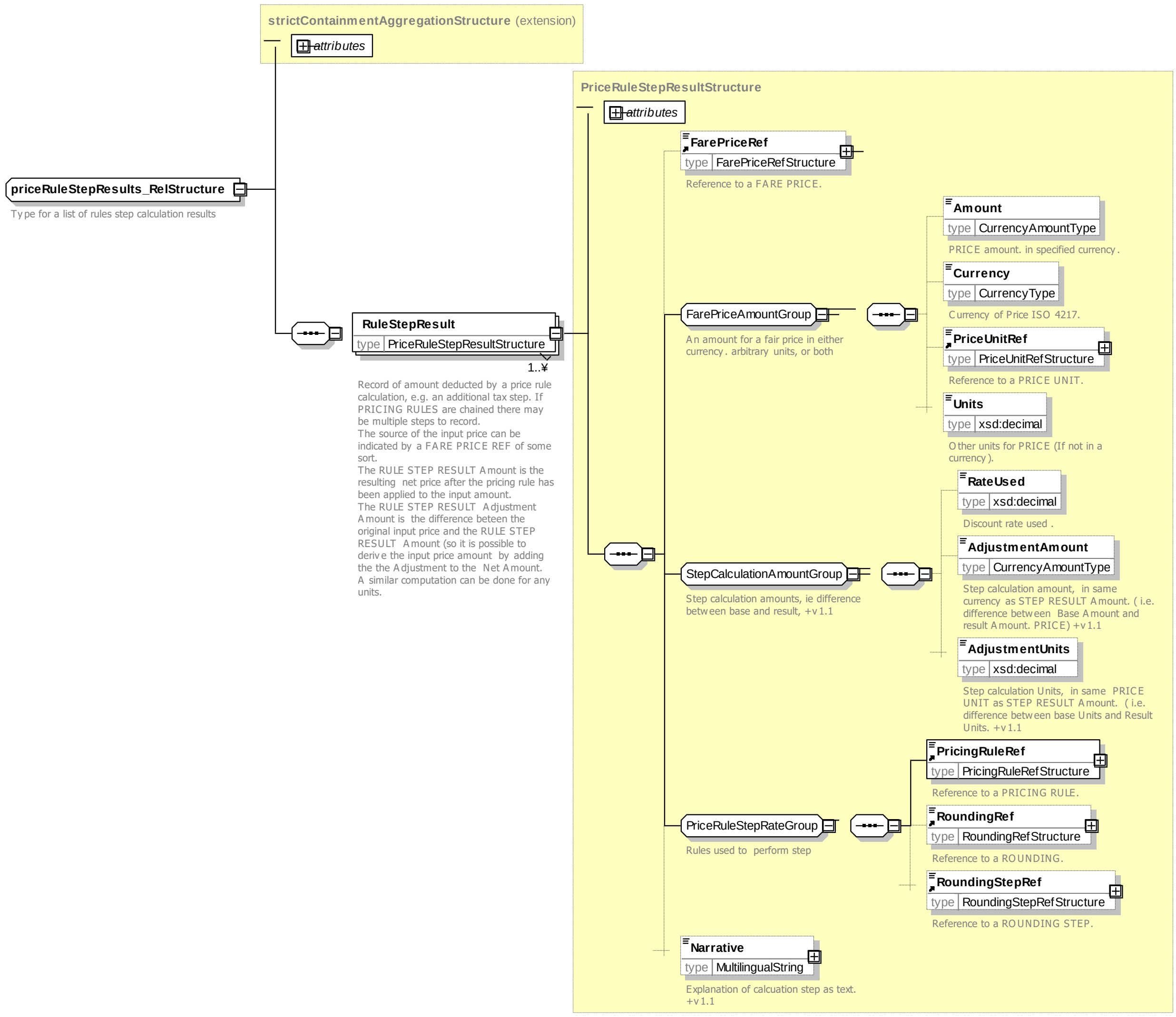


Figure 280 — RuleStepResult — XSD

7.7.2.4.3 PriceGroup – Model Element

A grouping of prices, allowing the grouping of numerous possible consumption elements into a limited number of price references, or to apply grouped increases, in value or percentage.

Table 271 – *PriceGroup* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*GroupOfEntities*</u>	::>	PRICE GROUP inherits from GROUP OF ENTITIes.
«PK»	***id***	*PriceGroupIdType*	1:1	Identifier of PRICE GROUP.
	StartDate	*xsd:date*	0:1	Start date for PRICE GROUP.
	EndDate	*xsd:date*	0:1	End date for PRICE GROUP.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Rounding to use on calculation.

«FK»	***members***	*FarePrice* | *FarePriceRef+*	0:*	PRICEs in PRICE GROUP. Each price can represents a priceable combination that shares the common price.
«FK»	***FarePriceRef***	*FarePriceRef+*	0:1	Reference to a FARE PRICE to be used as the price for the PRICE GROUP.

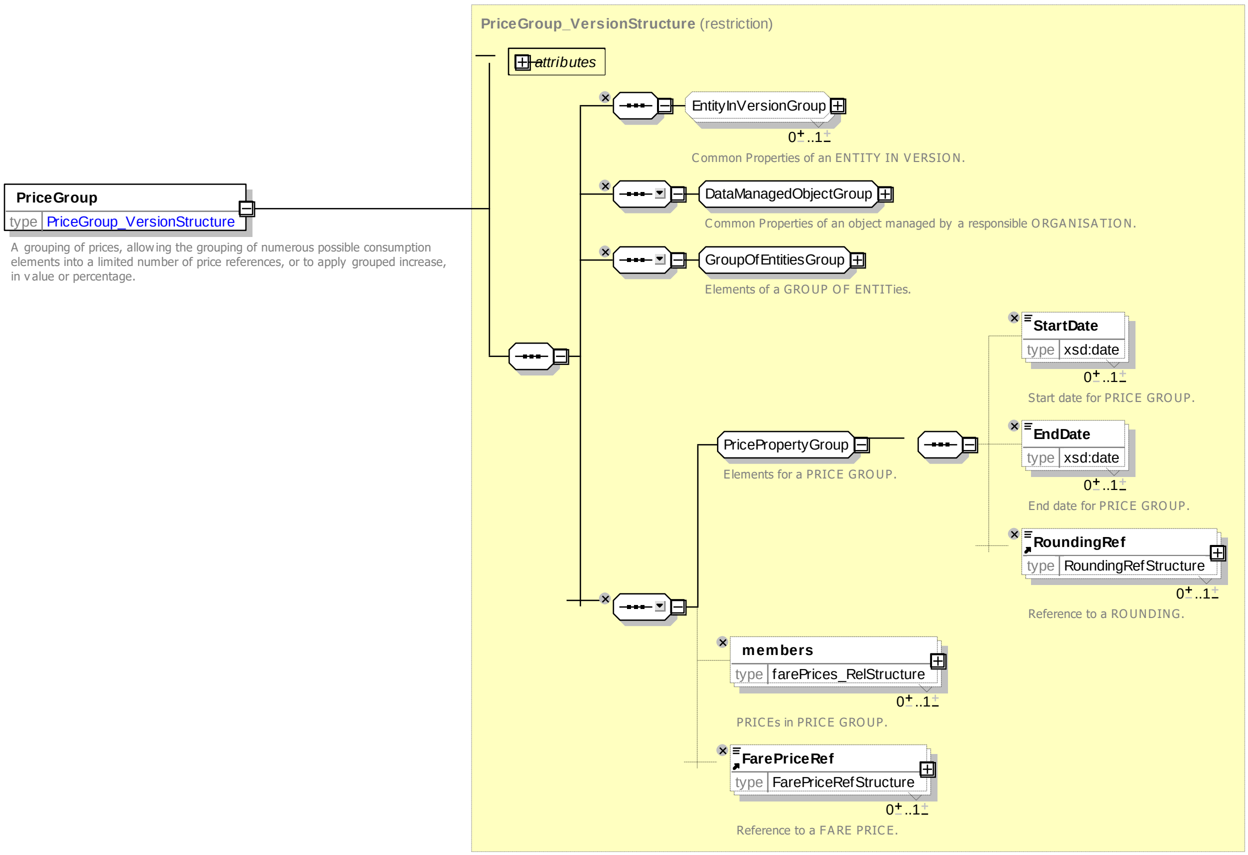


Figure 281 — *PriceGroup* — XSD

7.7.2.4.4 PriceableObject – Model Element

An element which may have a FARE PRICE.

Table 272 – *PriceableObject* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	PRICEABLE OBJECT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PriceableObjectIdType*	1:1	Identifier of PRICEABLE OBJECT.
	Name	*MultilingualString*	0:1	Name of PRICEABLE OBJECT.
	Description	*MultilingualString*	0:1	Description of PRICEABLE OBJECT.

	Url	*xsd:AnyURI*	0:1	URL to web page with information about PRICEABLE OBJECT. +v1.1
«cntd»	***infoLinks***	*InfoLink*	0:*	Additional hyperlinks for +v1.1
«cntd»	***alternativeNames***	*AlternativeName*	0:*	ALTERNATIVE Name for element.
«cntd»	***notice-Assignments***	*NoticeAssignment*	0:*	NOTICE ASSIGNMENTs associated with the element.
«FK»	***PricingService-Ref***	*PricingServiceRef*	0:1	PRICING SERVICE to use to fetch prices dynamically.
«FK»	***PricingRuleRef***	*PricingRuleRef+*	0:1	Default PRICING RULE to use to derive prices from this element.
«cntd»	***priceGroups***	*PriceGroup*	0:*	PRICE GROUPs associated with PRICEABLE OBJECT.
«cntd»	***fareTables***	*FareTable*	0:*	FARE TABLEs associated with PRICEABLE OBJECT.

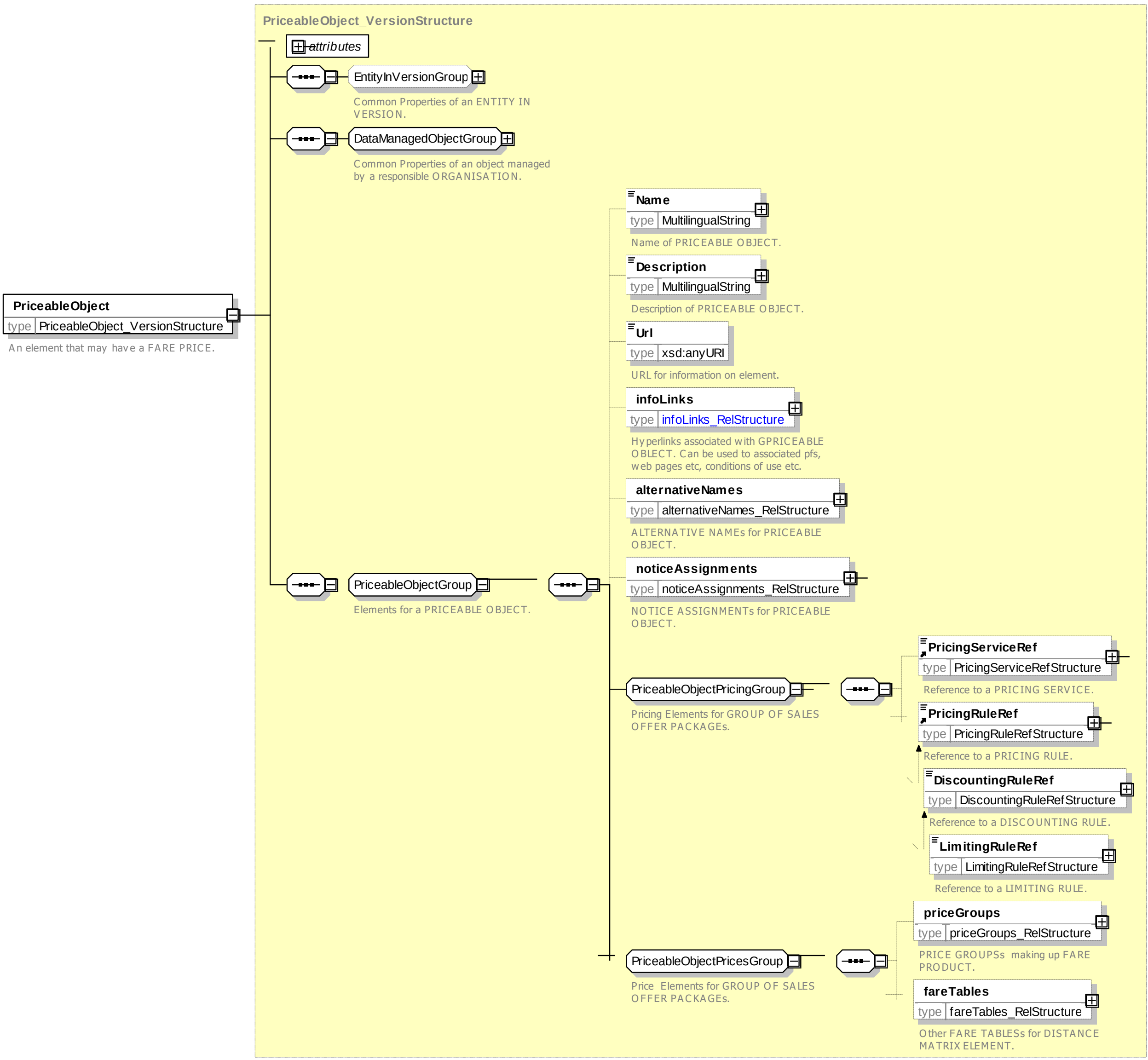


Figure 282 — *PriceableObject* — XSD

7.7.2.5 Fare Price – XML examples

7.7.2.5.1 FarePrice: XML Example of Price

The following code fragment shows *a* simple FARE STRUCTURE ELEMENT PRICE.

For EXAMPLE:

```
<FareStructureElementPrice version="any"
    id="nr:@WatfordJunction@Day@Adult@WatfordJunction@Zone_1-9">
            Name>Zone 1-9 to Watford Junction*</Name>
            <Amount>15.70</Amount>
            <FareStructureElementRef version="any" ref="tfl:Zones_1-9-Watford_Junction"/>
</FareStructureElementPrice>
```

7.7.2.5.2 FarePrice: XML Example of Price

The following code fragment shows a FARE PRODUCT PRICE for a product bought with a student rail card derived from a USAGE PARAMETER PRICE for a student that has a specific DISCOUNTING RULE of 25% for the holders of it.

For EXAMPLE:

```
<FareProductPrice version="any" id="nr:trip@student_RailCard">
        <Amount>15.00</Amount>
        <UsageParameterPriceRef version="any" ref="nr: student_RailCard@25pct"/>
        <SaleDiscountRightRef version="any" ref="nr:trip@StudentRailCard"/>
</FareProductPrice>

<UsageParameterPrice version="any" id="nr:student_RailCard@25pct ">
        <Amount>20.00</Amount>
        <DiscountingRuleRef version="any" ref=" tfl:concession@25pct "/>
        <UsageParameterRef version="any" ref="tfl:concession"/>

</ UsageParameterPrice >

<DiscountingRule version="any" id="nr: concession@25pct ">
        <DiscountAsPercentage>20.00</Amount>
<DiscountingRule >
```

7.7.3 Fare Table

7.7.3.1 Fare Table – Conceptual MODEL

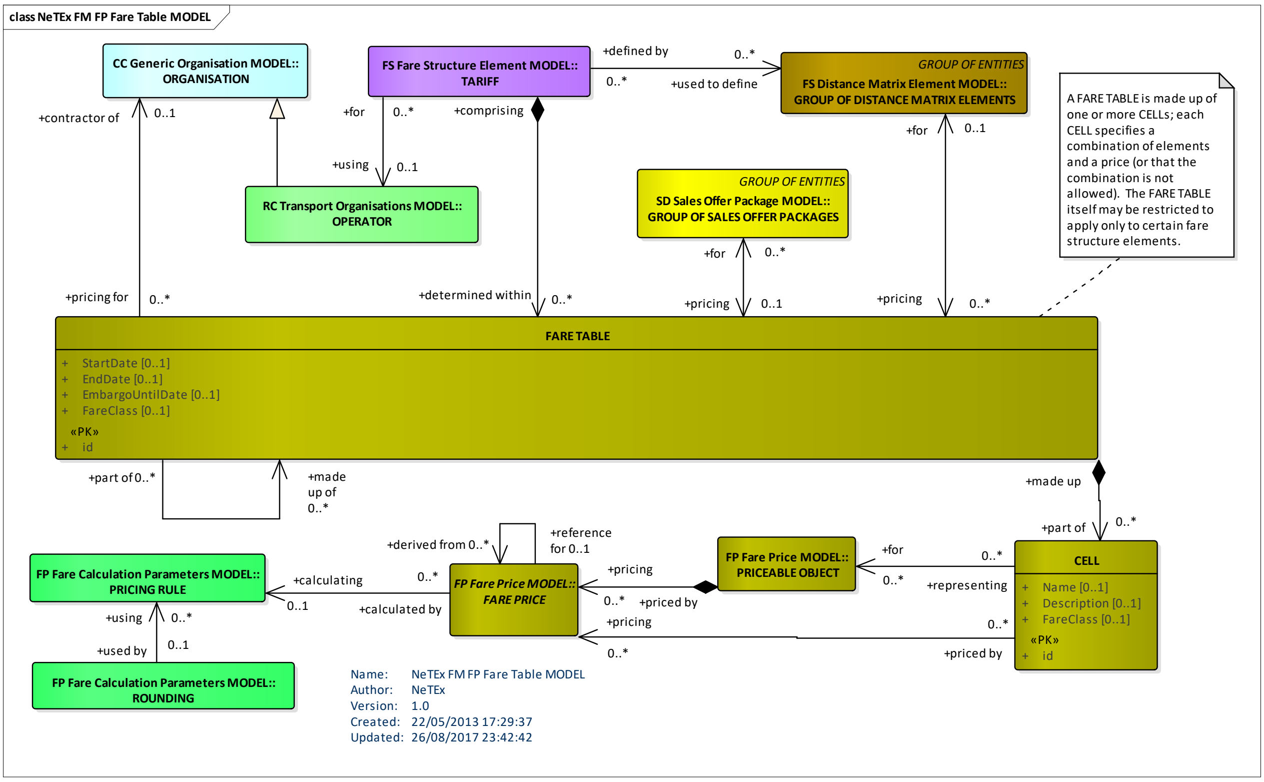


Figure 283 — Fare Table – Conceptual MODEL (UML)

7.7.3.2 Fare Table – Conceptual Examples

7.7.3.2.1 Fare Table – Conceptual Examples: Rail Fares

The following figures shows part of a multi-dimensional table for rail fares. Each row shows prices for different FARE PRODUCT / USER PROFILE (Adult, child, etc) combination. There are different fares for different accommodation and classes of use (these can be represented by FARE STRUCTURE ELEMENTs) for two different DISTANCE MATRIX elements.

			Trainhôtel PAU CASALS (3) – Au départ de Barcelone França – Gérone, Figureras – Perpignan à destination de Genève – Lausanne – Fribourg – Bern et Zurich													
Nom du produit	Trajet	Code	Single Grande Classe		Double Grande Classe (5)		Single Affaires		Double Affaires (5)		T4 Touriste				Sièges Super inclinables	
Classe de service			S		T		L		N		R		P		A	
Adulte	TS	PT00AD	491	421	339	291	409	345	284	243	196	163	-	-	158	140
Enfant	TS	PT0012	344	295	237	204	286	242	199	171	137	115	-	-	111	98
Loisir	AR	ND00AD	688	590	474	408	572	484	398	342	274	230	-	-	222	196
Prem's	TS	PP05AD	-	-	-	-	-	-	-	-	93	74	-	-	-	-
Mini à deux (prix pour 2)	TS	PP05AD	-	-	378	346	-	-	278	238	-	-	-	-	-	-
Duo (prix pour 2)	TS	PE07AD	-	-	440	380	-	-	370	316	-	-	322	272	206	182
Espace Plus (prix pour 4)	TS	PE08AD	-	-	-	-	-	-	-	-	508	424	-	-	-	-
Jeune	TS	PT0026	344	295	237	204	286	242	199	171	137	115	-	-	111	98
Senior	TS	PT0060	344	295	237	204	286	242	199	171	137	115	-	-	111	98
Congrès – Salons	AR	CN20AD	590	506	406	350	490	414	340	292	236	196	-	-	190	168
Pass	TS	EP00AD	222	222	173	173	179	179	119	119	77	77	-	-	52	52
Groupe Adulte	TS	GR00AD	-	-	210	177	-	-	176	148	117	98	-	-	-	-
	AR	GR02AD	-	-	-	338	-	-	-	284	-	172	-	-	-	-
Guide handicapé	TS	GG99AD	222	222	173	173	179	179	119	119	77	77	-	-	52	52
FIP Loisir	TS	EM01AD	222	222	173	173	179	179	119	119	77	77	-	-	52	52
FIP service (6)	TS	CT02AD	222	222	173	173	30	30	20	20	10	10	-	-	-	-
Enfant partageant un lit	TS	PE0112	60	60	60	60	60	60	60	60	60	60	-	-	-	-
Animaux domestiques	TS	CH50CH	60	60	60	60	60	60	60	60	60	60	60	60	60	60

			Trainhôtel SALVADOR DALI (4) – Au départ de Barcelone França – Gérone – Figueras – Perpignan à destination de Bardonecchia – Torino – Novara et Milano													
Nom du produit	Trajet	Code	Single Grande Classe		Double Grande Classe (5)		Single Affaires		Double Affaires (5)		T4 Touriste				Sièges Super inclinable	
Classe de service			S		T		L		N		R		P		A	
Adulte	TS	PT00AD	501	423	338	290	400	341	280	242	194	163	-	-	156	139
Enfant	TS	PT0012	351	297	237	203	280	239	196	170	136	115	-	-	109	98
Loisir	AR	ND00AD	702	594	474	406	560	478	392	340	272	230	-	-	218	196
Prem's	TS	PP05AD	-	-	-	-	-	-	-	-	93	74	-	-	-	-
Mini à deux (prix pour 2)	TS	PP05AD	-	-	378	346	-	-	278	238	-	-	-	-	-	-
Duo (prix pour 2)	TS	PE07AD	-	-	440	378	-	-	364	316	-	-	318	278	202	182
Espace Plus (prix pour 4)	TS	PE08AD	-	-	-	-	-	-	-	-	504	424	-	-	-	-
Jeune	TS	PT0026	351	297	237	203	280	239	196	170	136	115	-	-	109	98
Senior	TS	PT0060	351	297	237	203	280	239	196	170	136	115	-	-	109	98
Congrès – Salons	AR	CN20AD	602	508	406	348	480	410	336	292	232	196	-	-	188	168
Pass	TS	EP00AD	222	222	173	173	179	179	119	119	77	77	-	-	52	52
Groupe Adulte	TS	GR00AD	-	-	206	179	-	-	174	147	118	105	-	-	-	-
	AR	GR02AD	-	-	402	336	-	-	338	278	228	190	-	-	-	-
Guide handicapé	TS	GG99AD	222	222	173	173	179	179	119	119	77	77	-	-	52	52
FIP Loisir	TS	EM01AD	222	222	173	173	179	179	119	119	77	77	-	-	52	52
FIP service (6)	TS	CT02AD	222	222	173	173	30	30	20	20	10	10	-	-	-	-
Enfant partageant un lit	TS	PE0112	60	60	60	60	60	60	60	60	60	60	-	-	-	-
Animaux domestiques	TS	CH50CH	60	60	60	60	60	60	60	60	60	60	60	60	60	60

1) Trainhôtel n°**409/407** (quotidien)
2) Trainhôtel n°**477/475** (quotidien)
3) Trainhôtel n°**11273/273** (circulation le mardi, jeudi et dimanche dans le sens Sud → Nord (Barcelone Milan et Zurich)
4) Trainhôtel n°**11274/274** (circulation le lundi, mercredi et vendredi dans le sens Nord → Sud (Milan et Zurich Barcelone)
5) La cabine Double Grande Classe et la cabine Double Affaires doivent être utilisées uniquement par deux personnes qui se connaissent (réserver en compartiment « Famille »).
6) Carte FIP + ordre de mission à présenter au poste de vente habilité.

IMPORTANT !
Lors d'une vente, pour toute difficulté liée à HERMES, veuillez contacter l'Assistance Gares au 30 40 50 ou 01 56 79 61 50.

P = Cabine Classe Touriste occupée uniquement par 2 personnes qui se connaissent. Sur Mosaïque+ dans « saisie type de lit », sélectionner T2 et dans « compartiment », sélectionner « Famille ».
S et T = En Classe Single et Double Grande Classe, le dîner, les boissons et le petit-déjeuner sont inclus.
L et N = En Classe Single et Double « Affaires », le petit déjeuner est inclus.
R = En cabine Classe Touriste, compartiment exclusivement dédié aux personnes du même sexe (Homme ou Dame) et devient mixte lorsque 2, 3 ou 4 personnes voyagent ensemble ; la cabine est entièrement réservée et devient dans ce cas compartiment « Famille ».

Le Trainhôtel est une offre internationale et il est interdit de descendre du train en dehors de sa gare de destination.

La réservation est ouverte à J-3 mois et jusqu'au départ du Trainhôtel. La réservation reste obligatoire.

Figure 284 — Fare Table – Example Rail Fare Table (EXM)

7.7.3.2.2 Fare Table – Conceptual Examples: Urban Fares

The following figures shows part of a complex multi-dimensional table for London PT fares. Each tab shows a table for a different USER PROFILE (Adult, student, etc). There are different fares for different products (Cash, Pay-as-you-go) and different times of travel (peak, off-peak). The fares are point to point fares and the fully resolved fare (i.e. including discounts) has been statically computed for every fare combination.

Accessibility | Help & Contact | Sitemap

Transport for London

Search: [ ] Search

Home | Live travel news | Getting around | Tickets | Road users | Corporate | Business & partners

Tickets > Fares

Tube, DLR and London Overground | National Rail | Single fare finder | Bus and tram | River | Emirates Air Line

Save money with Oyster pay as you go

- Store credit and use it journey by journey
- Never run out of credit with Auto top-up
- Cheaper than cash for single fares

Travelcards

- Travel as much as you like, as often as you like
- Add Travelcards to your Oyster card

Adult | 18+ student | 16-18 | 11-15 | 5 -10 | Jobcentre Plus | Bus & Tram | Railcard | Groups

Adult

You need an Oyster card to:

- Use pay as you go and daily price capping
- Buy 7 Day, Monthly and longer period Travelcards

You do not need an Oyster card to pay cash single fares or to buy Day Travelcards.

Zone	Cash	Oyster pay as you go				Travelcards				
		Peak single	Off-peak single	Peak price cap	Off-peak price cap	Day Anytime	Day Off-peak	7 Day	Monthly	Annual
Zone 1 only	£4.50	£2.10	£2.10	£8.40	£7.00	£8.80	£7.30	£30.40	£116.80	£1,216
Zones 1-2	£4.50	£2.80	£2.10	£8.40	£7.00	£8.80	£7.30	£30.40	£116.80	£1,216
Euston - Zone 2*	£4.50	£2.20	£2.10	£8.40	£7.00	£8.80	£7.30	£30.40	£116.80	£1,216
Zones 1-3	£4.50	£3.20	£2.70	£10.60	£7.70	£11.00	£8.00	£35.60	£136.80	£1,424
Euston - Zone 3*	£4.50	£3.00	£2.70	£10.60	£7.70	£11.00	£8.00	£35.60	£136.80	£1,424
Zones 1-4	£5.50	£3.80	£2.70	£10.60	£7.70	£11.00	£8.00	£43.60	£167.50	£1,744
Euston -	£5.50	£3.40	£2.70	£10.60	£7.70	£11.00	£8.00	£43.60	£167.50	£1,744

Figure 285 — Fare Table – Example TfL Fare Table (EXM)

7.7.3.3 Fare Table – Physical model

The following figure shows the physical model for FARE TABLEs.

A FARE TABLE allows the representation of groups of prices for combinations of fare elements. In effect it defines a multi-dimensional matrix of CELLs, each of which may state a FARE PRICE (as either a reference or directly) for a combination of one or more fare elements. For example, one might have USER PROFILE + DISTANCE MATRIX ELEMENT + CLASS OF USE references on each cell in order to define adult and child fares for first and second class.

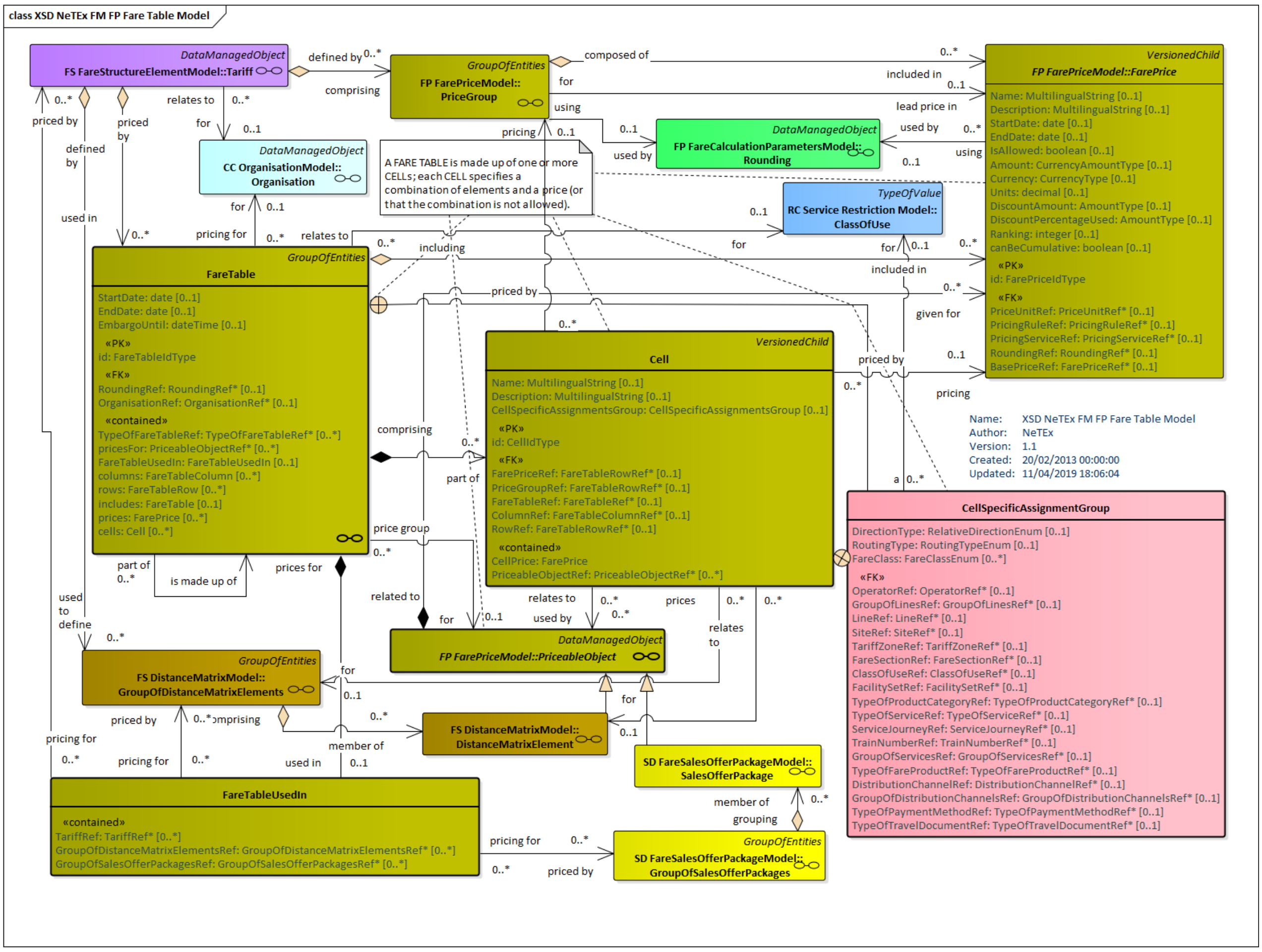


Figure 286 — Fare Table– Physical Model (UML)

7.7.3.3.1 Fare Table Overview – Physical model

The following figure summarises the key fare structure and pricing elements that can be associated with FARE TABLES and FARE TABLE CELLs. Each cell represents the price for a combination of all elements from the FARE TABLE and any elemenst specifc to itself.

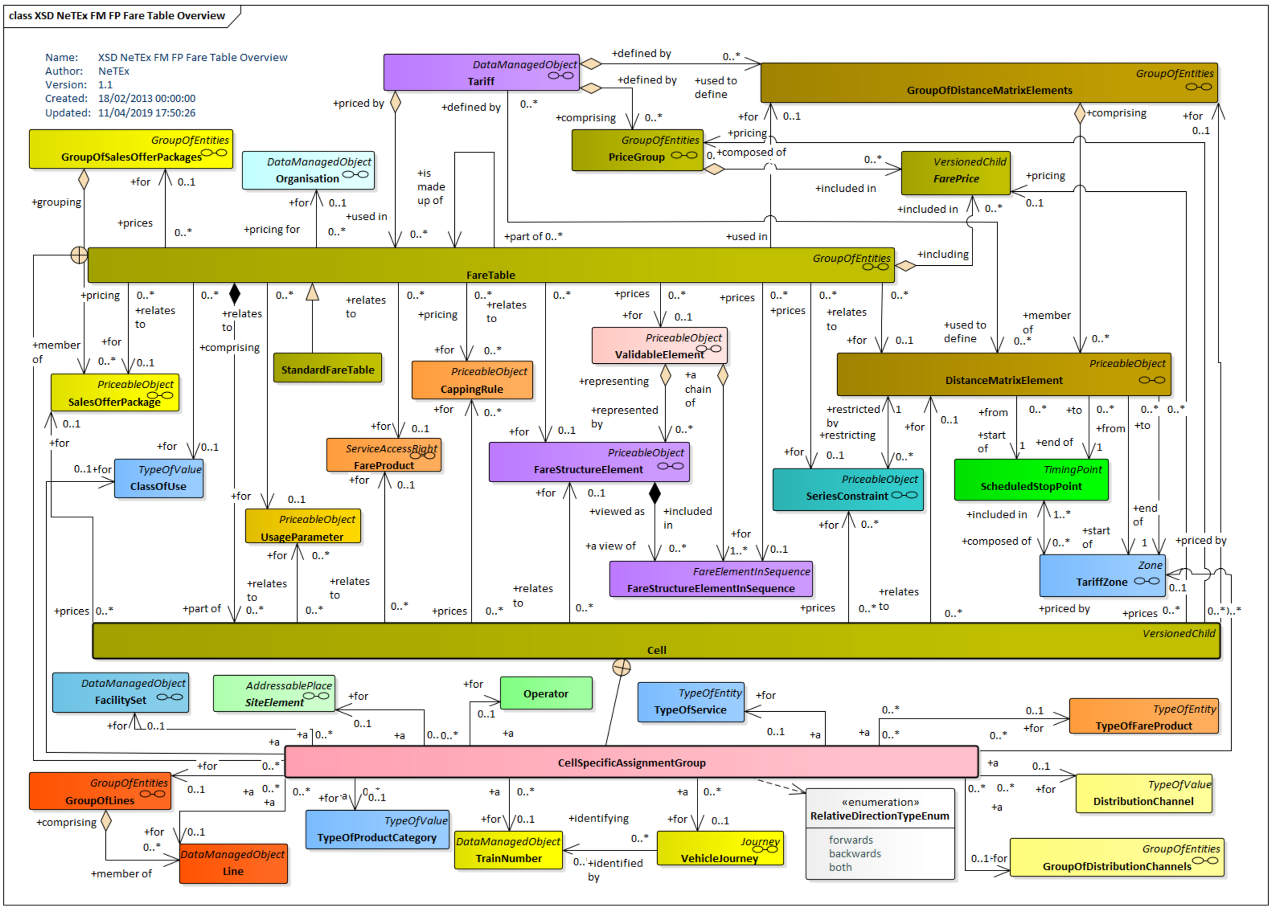


Figure 287 — Fare Table Overview – Physical Model (UML)

7.7.3.3.2 Fare Table Row & Column Headings – Physical model

A FARE TABLE can have row and column headings. These can reference an ENTITY for which they describe the price. The following figure shows further details of the physical model for FARE TABLEs.

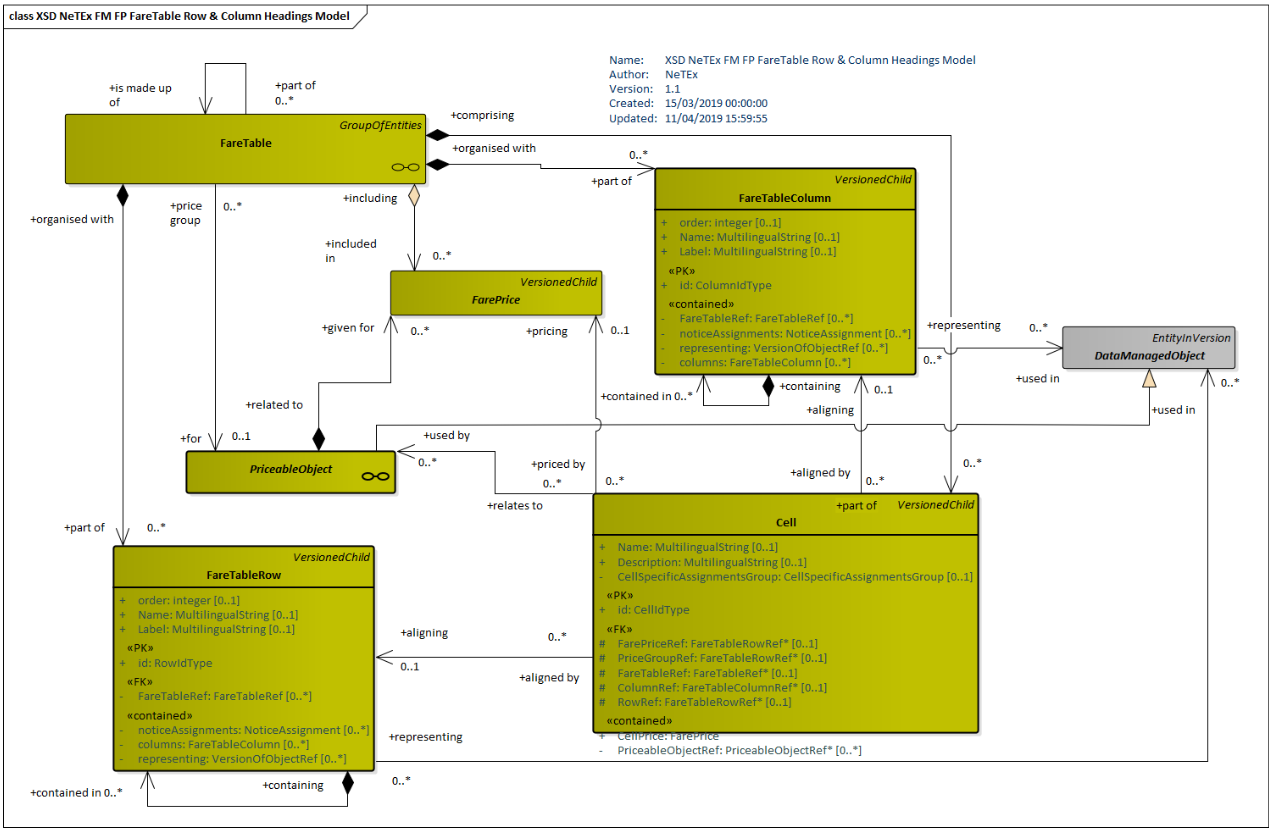


Figure 288 — Fare Table Row & Column Headings – Physical Model (UML)

7.7.3.3.3 Cell Price Introduction – Physical model

The following figure introduces the physical model for FARE TABLE CELLs.

Each cell may reference one or more PRICEABLE OBJECTs as well as other pricint elements (see CELL PRICE ASSIGNMENT below).

Cells may also be assigned to ROWs and COLUMNs for presentation purposes.

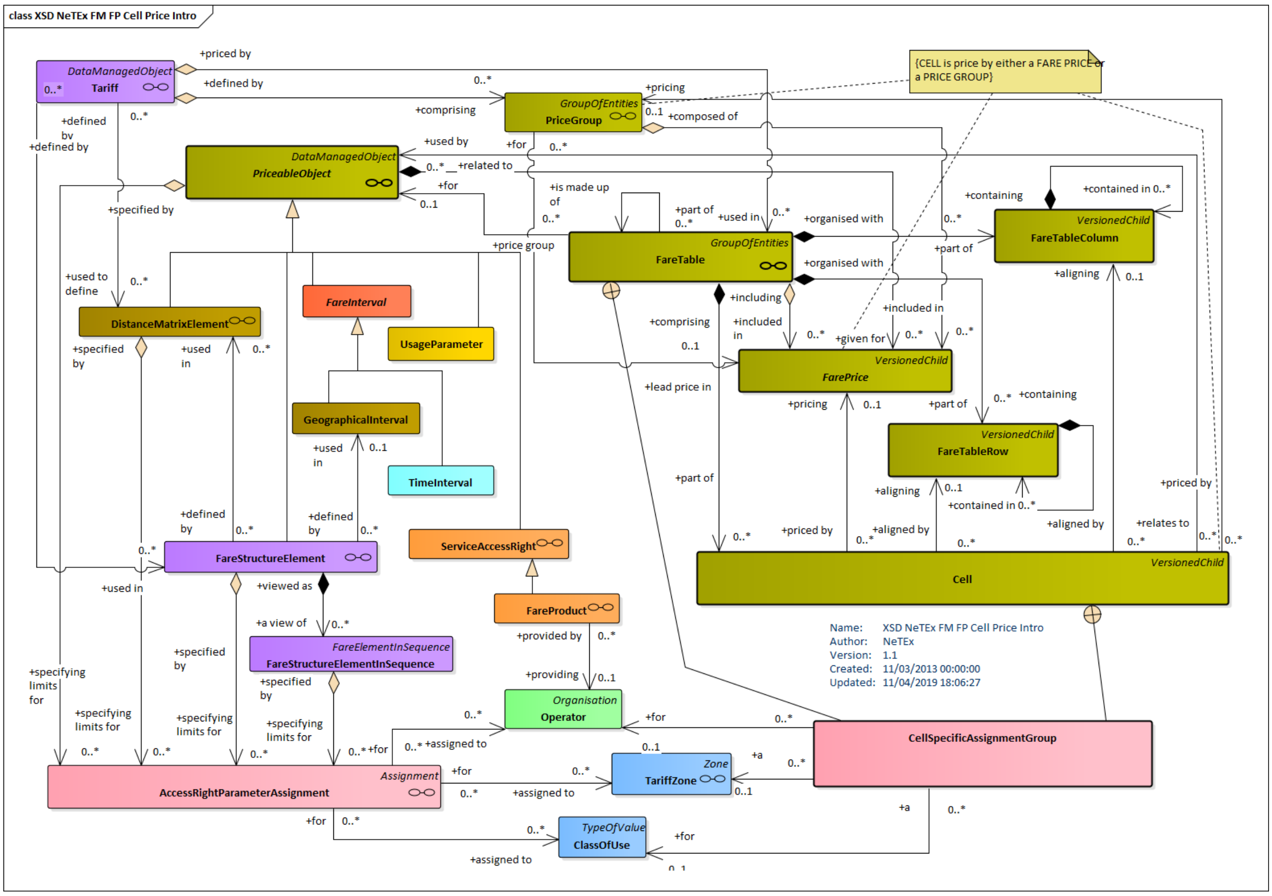


Figure 289 — Cell Price Introduction – Physical Model (UML)

7.7.3.3.4 Cell Price Assignment– Physical model

Each CELL represents the combination of on or more PRICEABLE OBJECT such as a FARE STRUCTURE ELEMENT, FARE PRODUCT, USAGE PARAMETER, DISTANCE MATRIX ELEMENT, SERIES CONSTRAINT, TIME INTERVAL, etc.

In a product defenition, some of these elements may themselves have been defined as lists of choices of further pricing factors, such as TARIFF ZONE, CLASS OF USE, PRODUCT CATEGORY, etc.; there may be a separate price for each possible combination of these respective factors. The CELL allows each combination of element to be specified directly along with an amount in cuuermncy or other PRICE UNIT.

The CELL can be considered as a “flattened” optimisation of a VALIDITY PARAMETER ASSIGNMENT that states the selection of parameters that apply to an individual price.

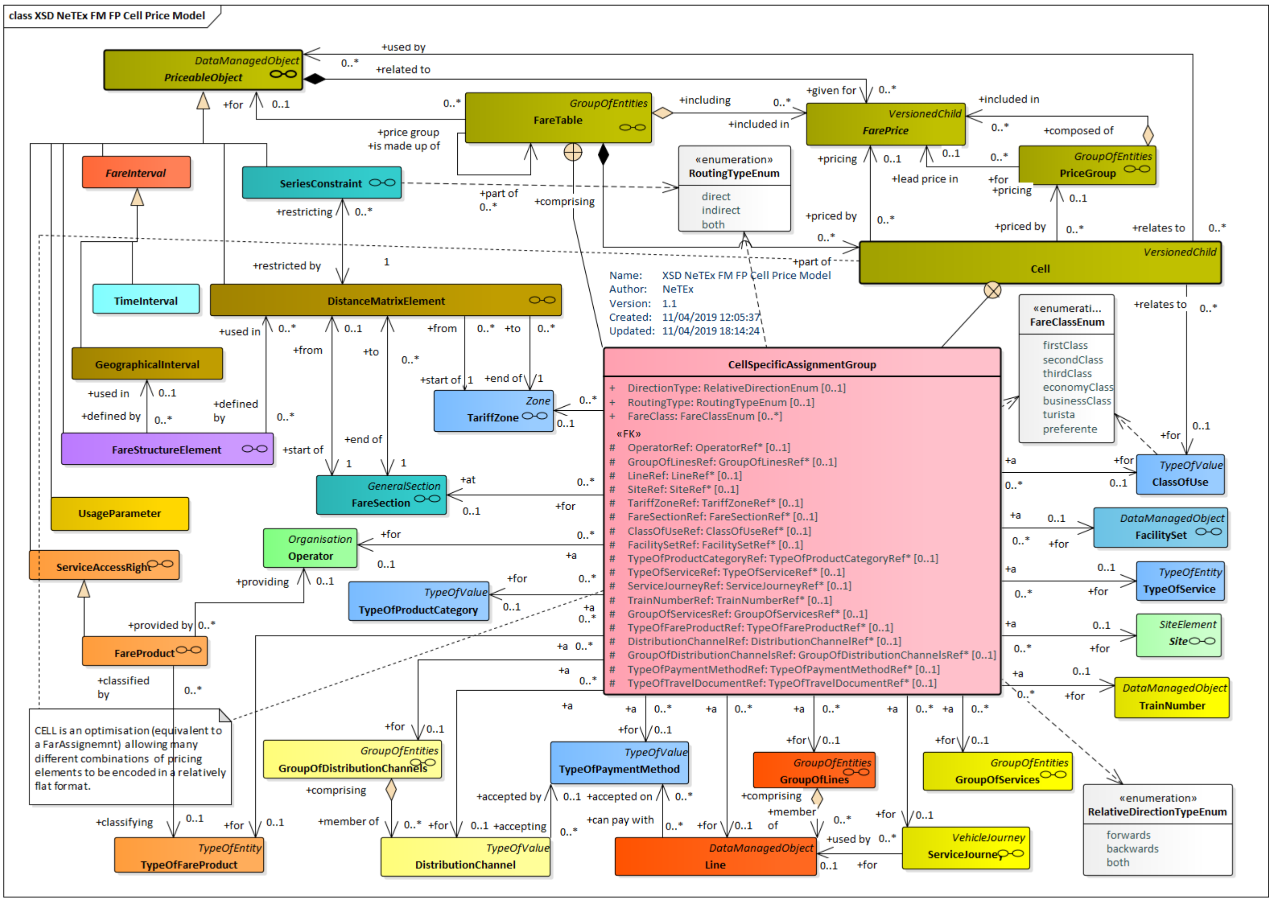


Figure 290 — Cell Price Assignment Model – Physical Model (UML)

7.7.3.3.5 **Standard Fare Table – Physical model**

The following figure shows the physical model for STANDARD FARE TABLEs, an optimisation of FARE TABLE that allows exchange a predefined set of fare types in an efficient format. Each STANDARD FARE TABLE can hold four prices - for a first and second class single and return.

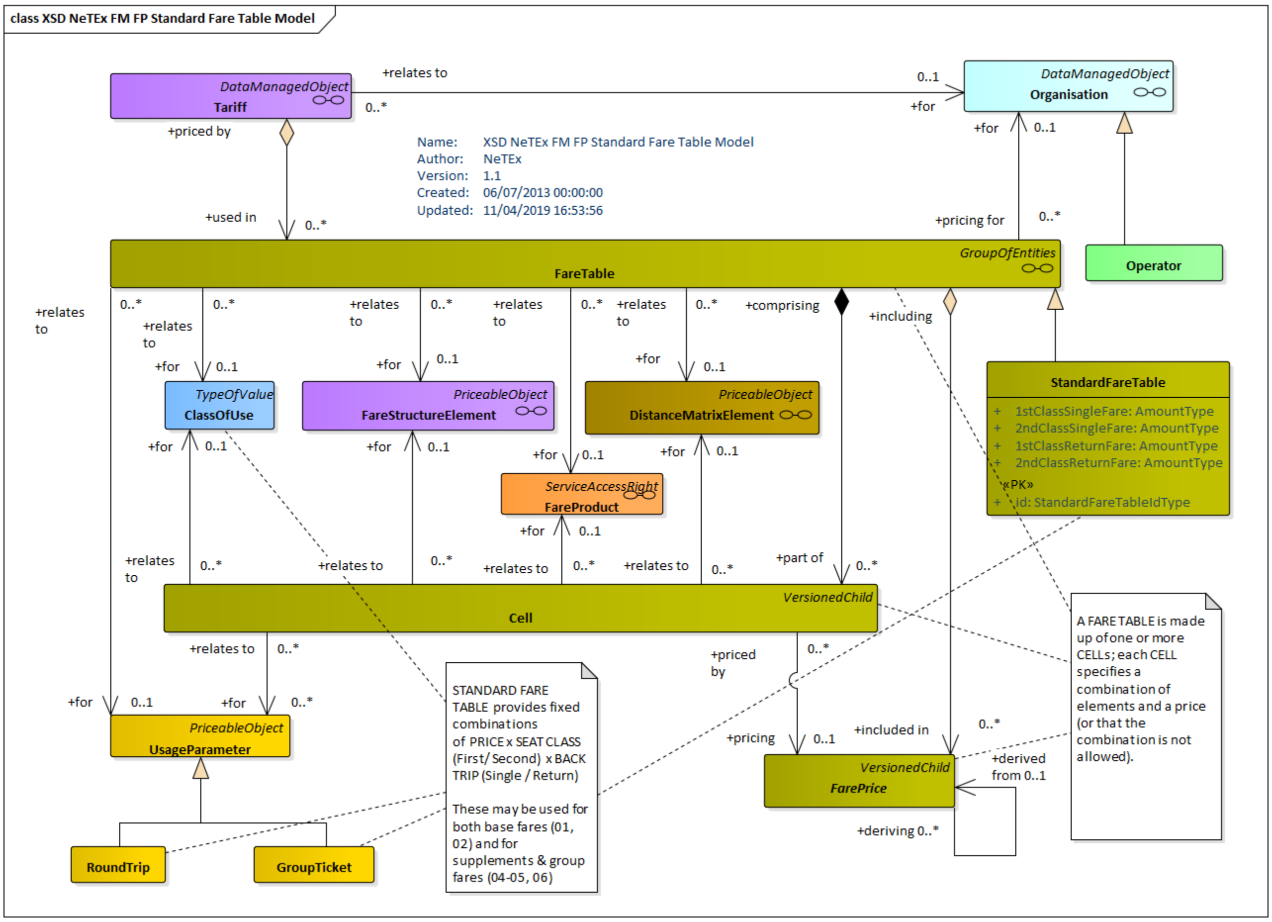


Figure 291 — Standard Fare Table – Physical Model (UML)

7.7.3.4 Fare Table – Attributes and XSD

7.7.3.4.1 **FareTable – Model Element**

A grouping of prices that may be associated with various combinations fare elements such as the DISTANCE MATRIX ELEMENT, FARE STRUCTURE ELEMENT, GEOGRAPHICAL INTERVAL, TIME INTERVAL, USAGE PARAMETER, etc.

Table 273 – *FareTable* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceGroup*	::>	FARE TABLE inherits from PRICE GROUP.
«PK»	***id***	*FareTableIdType*	1:1	Identifier of FARE TABLE.
	StartDate	*xsd:date*	0:1	Start date for PRICE validity.
	EndDate	*xsd:date*	0:1	End date for PRICE validity.

«FK»	***RoundingRef***	*RoundingRef*	0:1	Rounding to use on calculation.
XGRP	***FareTableReferences-Group***	***xmlGroup***	0:*	Fare structure elements which may be given a price and so associated with this CELL.
XGRP	***FareTableCommon-AssignmentsGroup***	***xmlGroup***	0:1	Aspects of a fare structure element for a CELL.
XGRP	***FareTable-HeadingsGroup***	***xmlGroup***	0:1	Row and column headings for the table. CELLs may reference these. See below.
	EmbargoUntil	*xsd:dateTime*	0:1	Prices must not be released until this date.
«cntd»	***prices***	*FarePrice*	0:*	An optimization – CELL declared as a simple FARE PRICE without additional associations.
«cntd»	***cells***	*Cell*	0:*	A tuple within a FARE TABLE that associates one or more fare entities with a price.
«cntd»	***noticeAssignments***	*NoticeAssignment*	0:*	NOTICEs that apply to whole FARE TABLE

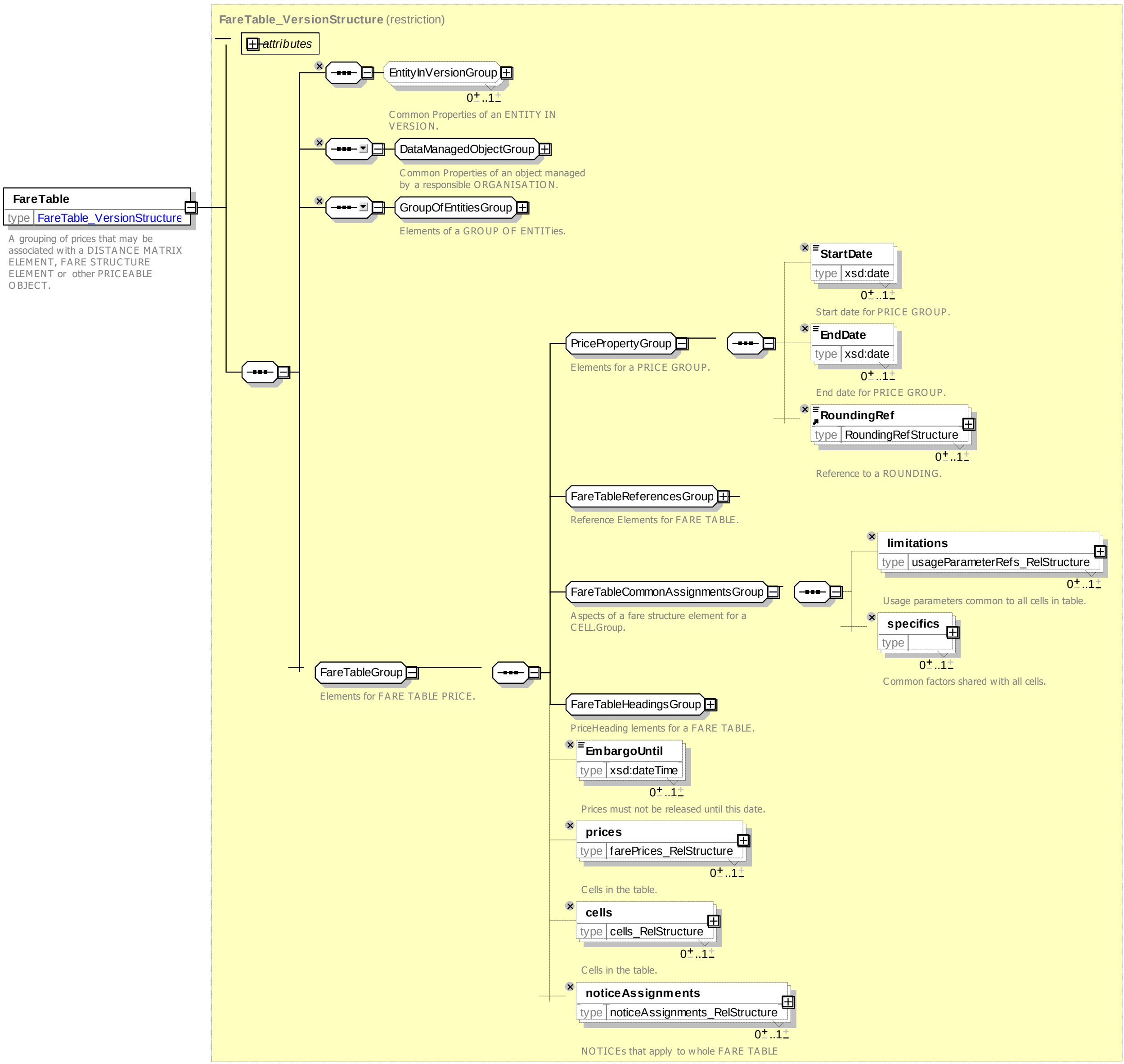


Figure 292 — *FareTable* — XSD

7.7.3.4.1.1 FareTableReferencesGroup – Group

The ***FareTableReferencesGroup*** specifies the associated fare structure elements for which prices are provided by the FARE TABLE.

Table 274 – *FareTableReferencesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***TypeOfFareTableRef***	*TypeOfFareTableRef*	0:1	Classification of FARE TABLE apply.
«cntd»	***pricesFor***	*PriceableObjectRef+*	0:*	PRICEABLE OBJECT elements which may be given a price and so associated with this CELL.
«cntd»	***usedIn***	*Choice*	0:1	A fare element associated with the FARE TABLE.

«FK»	*a*	***TariffRef***	*TariffRef*	1:*	TARIFF to which PRICEs of FARE TABLE apply.
«FK»	*b*	***GroupOf-Distance-Matrix-ElementsRef***	*GroupOfDistanceMatrix-ElementsRef*	1:*	GROUP OF DISTANCE MATRIX ELEMENTs associated with a FARE TABLE.
«FK»	*c*	***GroupOf-SalesOffer-PackagesRef***	*GroupOf-SalesOfferPackagesRef*	1:*	GROUP OF SALES OFFER PACKAGEs associated with a FARE TABLE.
«FK»	***OrganisationRef***		*(OrganisationRef)*	0:1	OPERATOR or AUTHORITY to which FARE PRICEs apply.

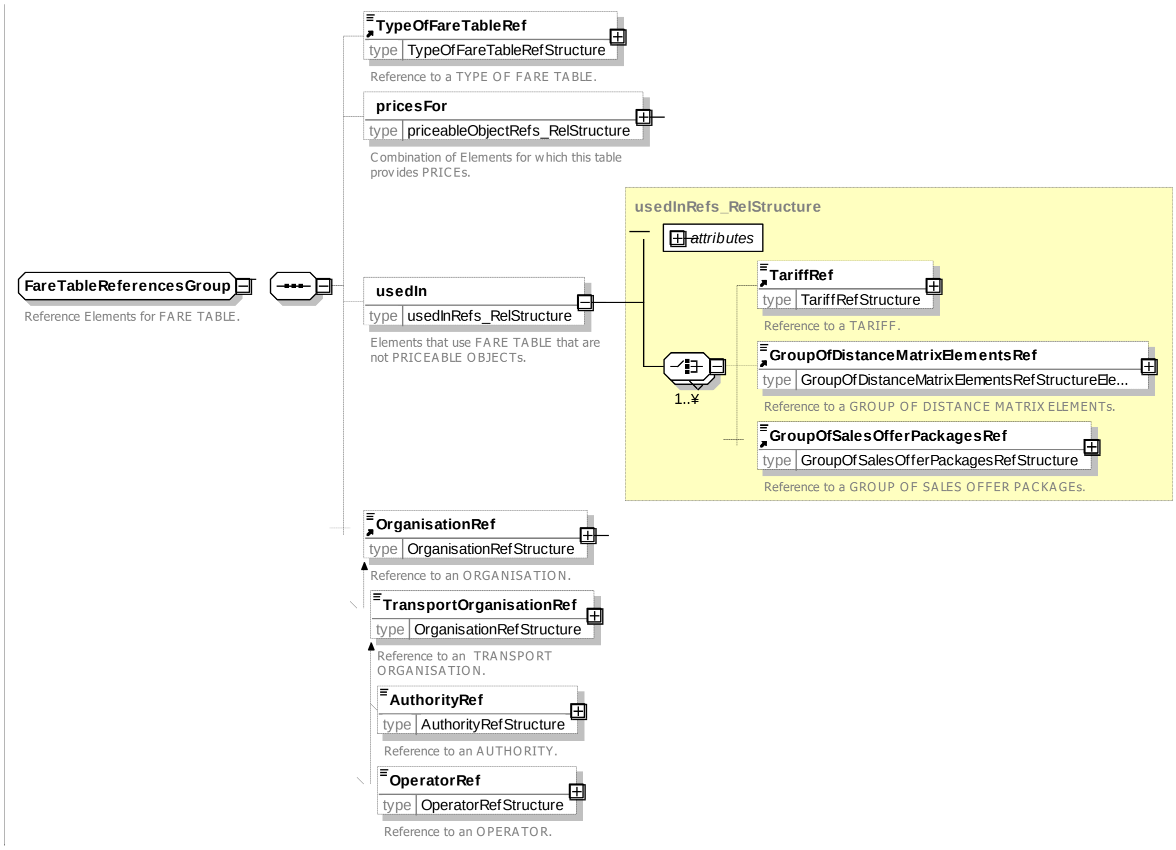


Figure 293 — *FareTableReferencesGroup* — XSD

7.7.3.4.1.2 FareTableCommonAssignmentsGroup – Group

The ***FareTableCommonAssignmentsGroup*** specifies the associated fare structure elements for which prices are provided by the FARE TABLE.

Table 275 – *FareTableCommonAssignmentsGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***limitations***	*UsageParameterRef+*	0:*	USAGE PARAMETER or PARAMETERs to which the CELL PRICE applies.

XGRP	***CellSpecific-NetworkGroup***	***xmlGroup***	0:1	Combination of network related elements for which FARE TABLE or CELL provides a price.
XGRP	***CellSpecific-RoutingGroup***	***xmlGroup***	0:1	Combination of routing related elements for which FARE TABLE or CELL provides a price.
XGRP	***CellSpecific-ServiceGroup***	***xmlGroup***	0:1	Combination of service-related elements for which FARE TABLE or CELL provides a price.
XGRP	***CellSpecific-Distribution-Group***	***xmlGroup***	0:1	Combination of distribution related elements for which FARE TABLE or CELL provides a price.

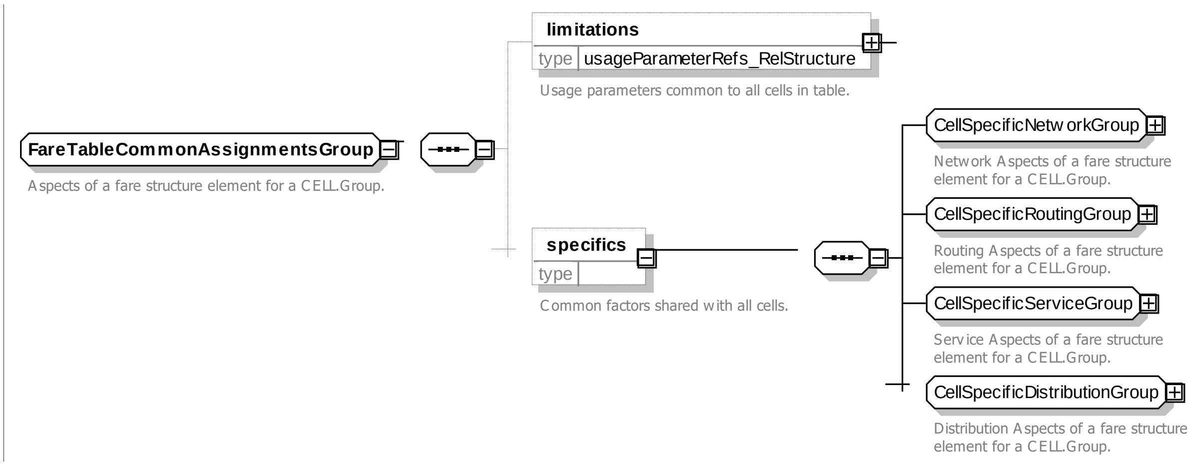


Figure 294 — *FareTableCommonAssignmentsGroup* — XSD

7.7.3.4.1.3 FareTableHeadingsGroup – Group

The ***FareTableHeadingsGroup*** specifies any row and column headings which may be used to present the FARE TABLE.

Table 276 – *FareTableHeadingsGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***columns***	*FareTable-ColumnHeading*	0:*	Column headings to use when presenting table.
«cntd»	***rows***	*FareTableRowHeading*	0:*	Row headings to use when presenting table.
«cntd»	***includes***	*FareTable*	0:*	FARE TABLEs nested within this table. Can be recursive. +v1.1

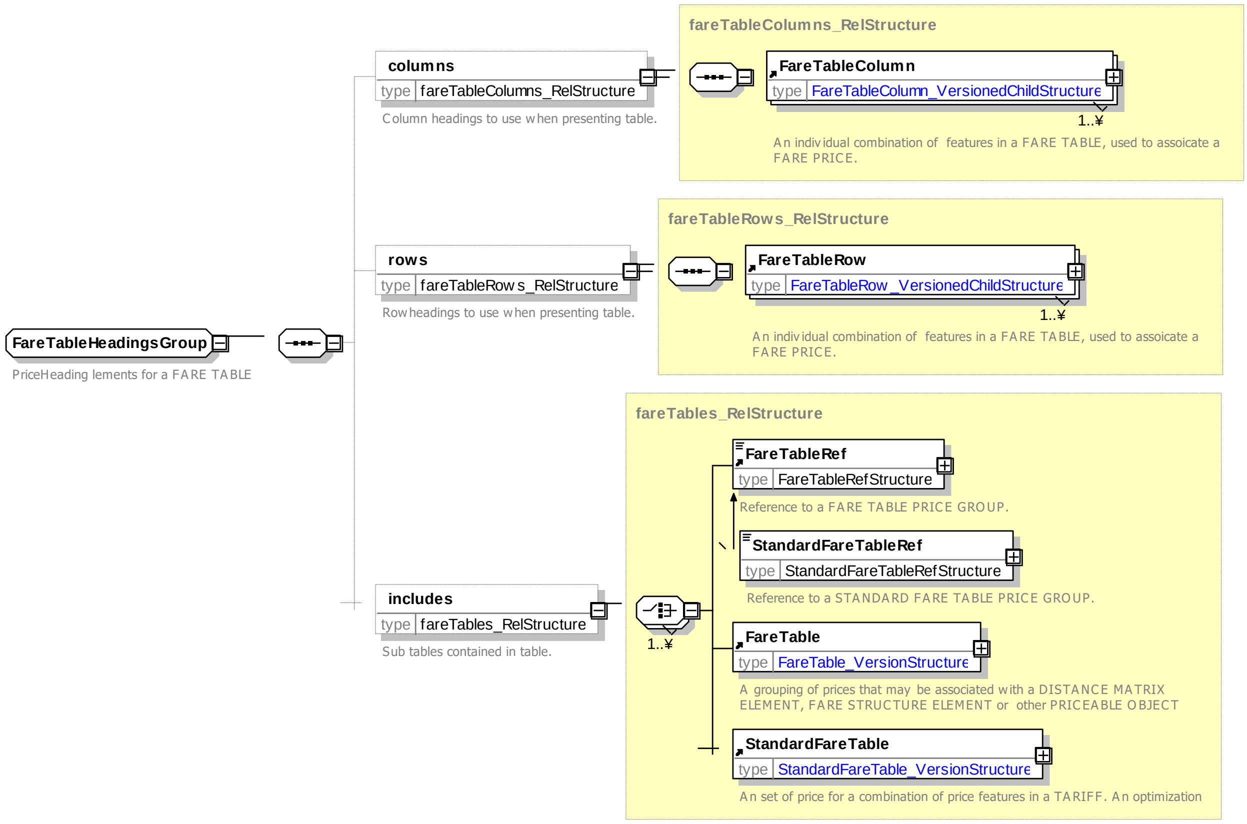


Figure 295 — *FareTableHeadingsGroup* — XSD

7.7.3.4.1 FareTableColumn – Model Element

A Column heading for a FARE TABLE.

Table 277 – *Column* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	FARE TABLE COLUMN inherits from VERSIONED CHILD
«PK»	***id***	*FareTableColumnIdType*	1:1	Identifier of FARE TABLE COLUMN.
	Name	*MultilingualString*	0:1	Name of FARE TABLE COLUMN.
	Description	*MultilingualString*	0:1	Description of FARE TABLE COLUMN.
«FK»	***FareTableRef***	*FareTableRef*	1:1	Reference to a FARE TABLE containing COLUMN
«cntd»	**notice-Assignments**	*NoticeAssignments*	0:*	NOTICEs that apply to whole FARE TABLE COLUMN.
«cntd»	***representing***	*(VersionOfObjectRef)*	0:*	ENTITIES that column represents. +v1.1
«cntd»	***columns***	*FareTable-ColumnHeading*	0:*	Nested FARE TABLE COLUMN headings to use when presenting table. Recursive. +v1.1

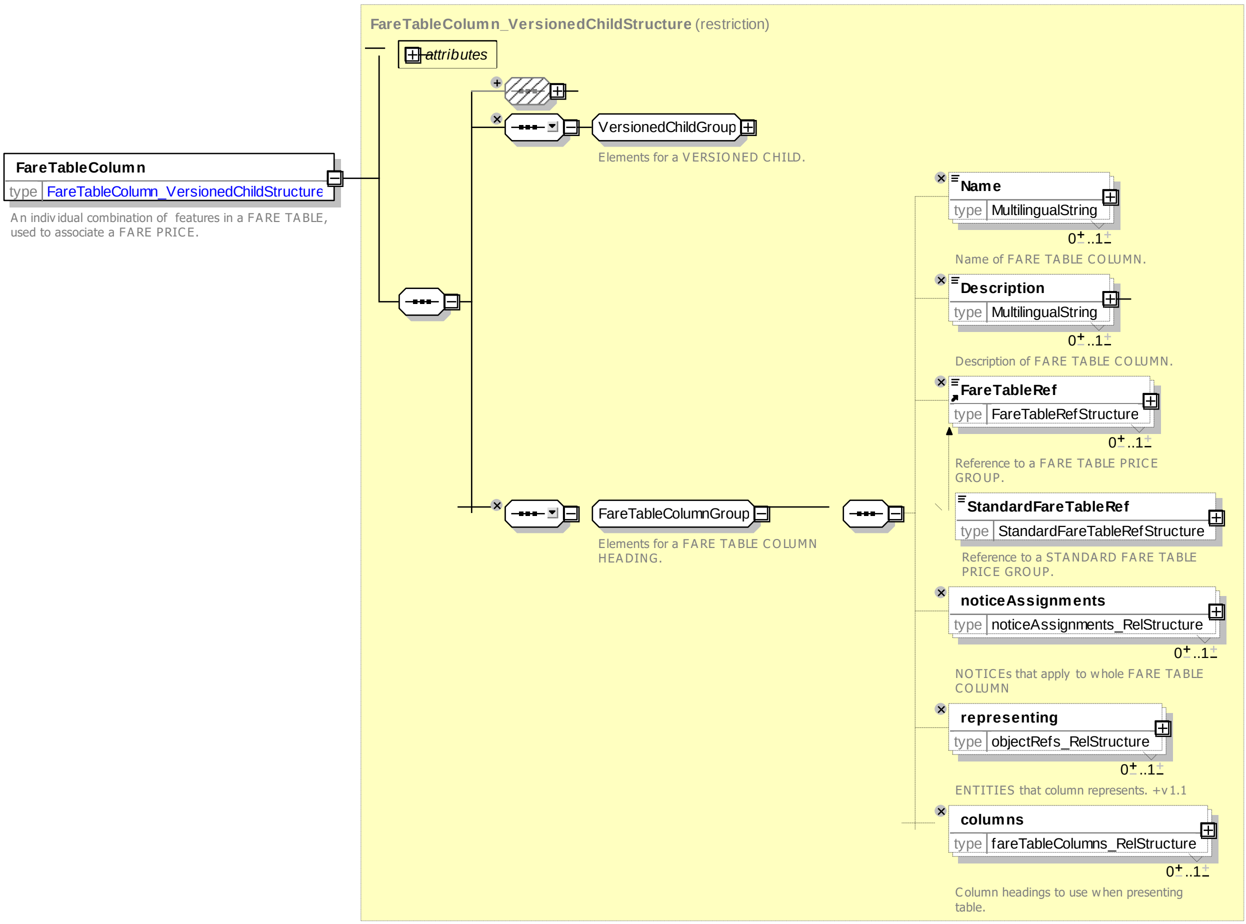


Figure 296 — *FareTableColumn* — XSD

7.7.3.4.1 FareTableRow – Model Element

A Row heading for a FARE TABLE.

Table 278 – *Column* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	FARE TABLE ROW inherits from VERSIONED CHILD.
«PK»	***id***	*FareTableRowIdType*	1:1	Identifier of FARE TABLE ROW.
	Name	*MultilingualString*	0:1	Name of FARE TABLE ROW.
	Description	*MultilingualString*	0:1	Description of FARE TABLE ROW.
«FK»	***FareTableRef***	*FareTableRef*	1:1	Reference to a FARE TABLE containing FARE TABLE ROW.
«cntd»	**notice-Assignments**	*NoticeAssignments*	0:*	NOTICEs that apply to whole FARE TABLE ROW.

«cntd»	***representing***	*(VersionOfObjectRef)*	0:*	ENTITIES that FARE TABLE ROW represents. +v1.1.
«cntd»	***rows***	*FareTableRowHeading*	0:*	Nested FARE TABLE ROW headings to use when presenting table. Recursive.

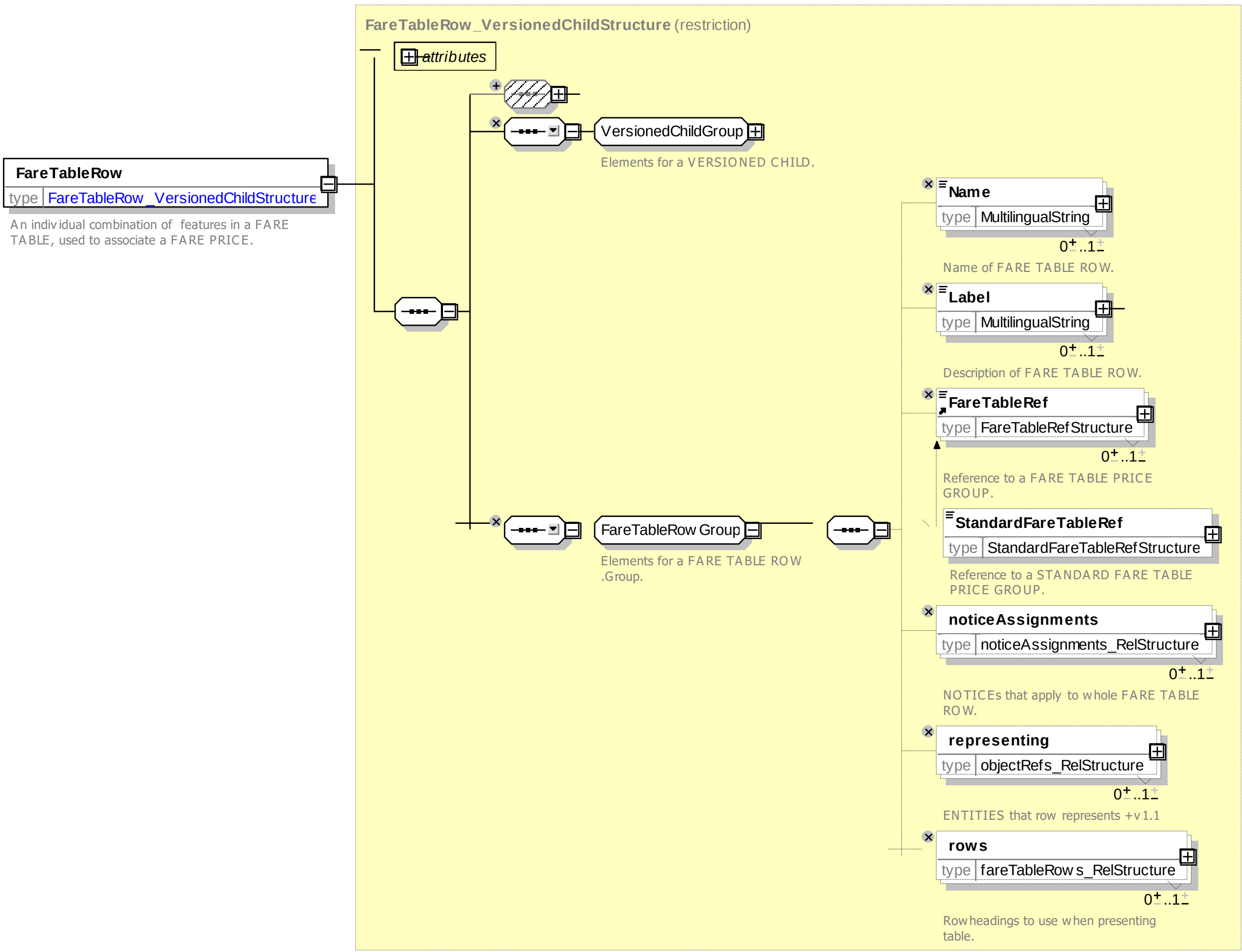


Figure 297 — *FareTableRow*— XSD

7.7.3.4.2 Cell – Model Element

A unique individual combination of features within a FARE TABLE, used to associate a FARE PRICE with a fare element.

Table 279 – *Cell* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	CELL inherits from VERSIONED CHILD
«PK»	***id***	*CellIdType*	1:1	Identifier of CELL.
	Name	*MultilingualString*	0:1	Name of CELL.

	Description		*MultilingualString*	0:1	Description of CELL.
	price		*Choice*	1:1	One of the following three
«cntd»	***a***	***CellPrice***	*FarePriceStructure*	1:1	Fare Price held inline in CELL. See FARE PRICE for description.
«FK»	***b***	***FarePriceRef***	*FarePriceRef+*	1:1	Reference to another FARE PRICE providing price for CELL
«FK»	***c***	***Price-GroupRef***	*PriceGroupRef*	1:1	Reference to a FARE GROUP providing price for CELL via its lead price.
XGRP	***CellReferences-Group***		***xmlGroup***	0:1	Fare Structure which may be given a price and so be associated with this CELL.
XGRP	***CellHeadings-Group***		***xmlGroup***	0:1	Table headings associated with this CELL.

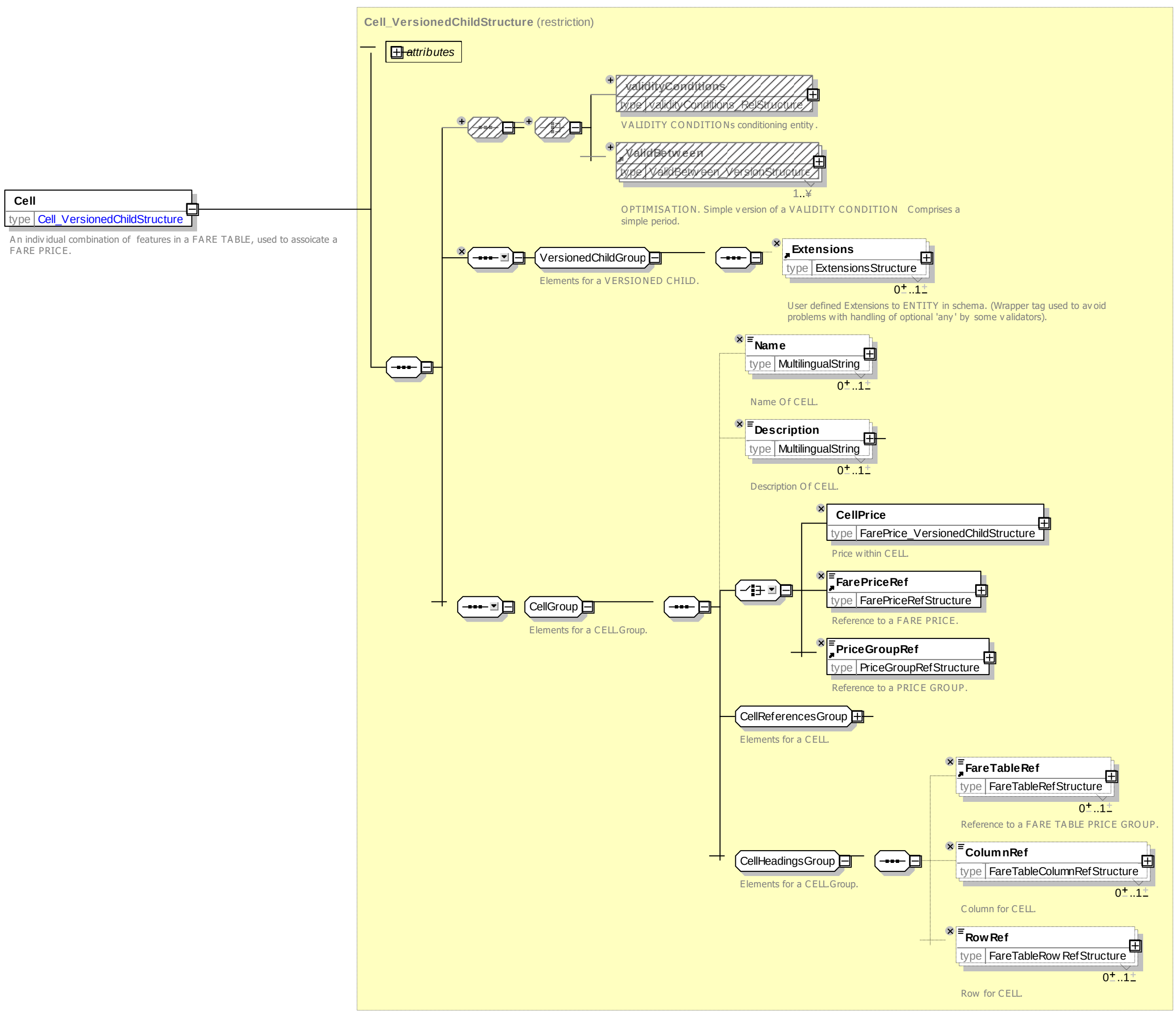


Figure 298 — *Cell* — XSD

7.7.3.4.2.1 CellReferencesGroup – Group

References to the elements for which the CELL provides a price. May be a combination of multiple elements.

Table 280 – *CellReferencesGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description
«cntd»	***PriceableObjectRef***	*PriceableObjectRef+*	0:*	Fare structure elements which may be given a price and so associated with this CELL.
«FK»	***GroupOf-DistanceMatrix-ElementsRef***	*GroupOf-DistanceMatrix-ElementsRef*	0:1	Reference to a GROUP OF DISTANCE MATRIX ELEMENTS) associated with an individual CELL or FARE TABLE.
XGRP	***CellSpecific-RoutingGroup***	***xmlGroup***	0:1	Routing elements for which CELL provides a price.
XGRP	***CellSpecific-NetworkGroup***	***xmlGroup***	0:1	Network elements for which CELL provides a price.
XGRP	***CellSpecific-ServiceGroup***	***xmlGroup***	0:1	Service elements for which CELL provides a price.
XGRP	***CellSpecific-DistributionGroup***	***xmlGroup***	0:1	Distribution elements for which CELL provides a price.

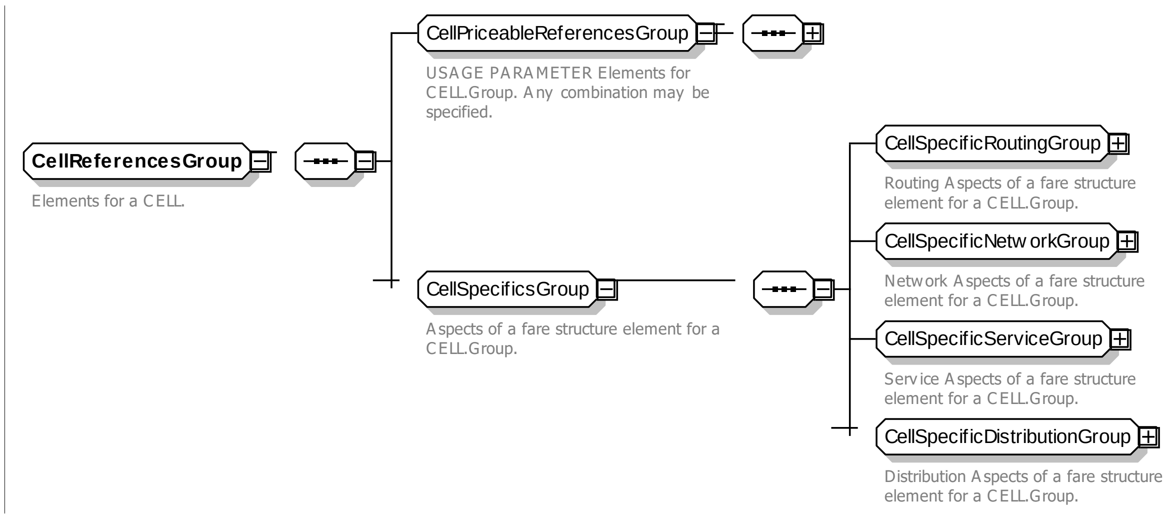


Figure 299 — *CellReferencesGroup* — XSD

7.7.3.4.2.2 CellSpecificNetworkGroup – Group

References to the network elements for which the CELL provides a price. May be a combination of multiple elements.

Table 281 – *CellReferencesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description

«FK»	***GroupOfLines-Ref***	*GroupOfLinesRef*	0:1	A GROUP OF LINEs for which the CELL provides a price.
«FK»	***LineRef***	*LineRef*	0:1	A LINE for which the CELL provides a price.
«FK»	***SiteRef***	*SiteRef*	0:1	A SITE for which the CELL provides a price.
«FK»	***TariffZoneRef***	*TariffZoneRef*	0:1	A TARIFF ZONE for which the CELL provides a price.
«FK»	***FareSectionRef***	*FareSectionRef*	0:1	A FARE SECTION for which CELL provides a price.

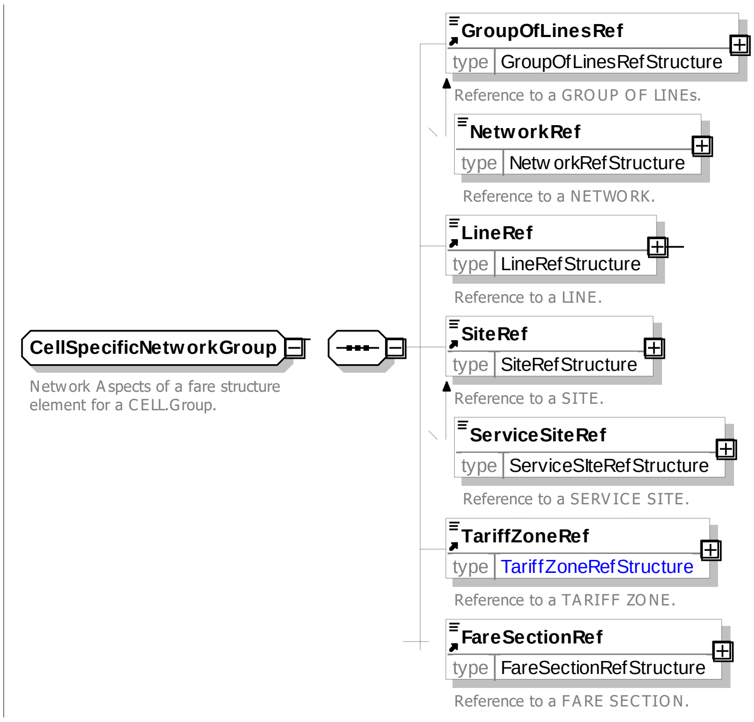


Figure 300 — CellSpecificNetworkGroup — XSD

7.7.3.4.2.3 CellSpecificRoutingGroup – Group

References to the routing related elements for which the CELL provides a price.

Table 282 – CellSpecificRoutingGroup – Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***DirectionType***	*RelativeDirectionEnum*	0:1	For fares for DISTANCE MATRIX ELEMENTs, DIRECTION in which price applies. See Part1 for allowed values.
«enum»	***RoutingType***	*RoutingTypeEnum*	0:1	Whether fare is for direct (i.e. no changes required point to point fare) or indirect routing. See allowed values below.

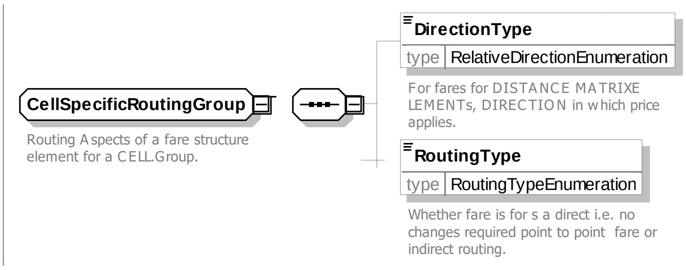


Figure 301 — CellSpecificRoutingGroup — XSD

7.7.3.4.2.4 CellSpecificServiceGroup – Group

References to the Service elements for which the CELL provides a price. May be a combination of multiple elements.

Table 283 – CellSpecificServiceGroup – Group

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***FareClass***	*FareClassEnum*	0:1	A FARE CLASS for which the CELL provides a price. See Part1.
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	A CLASS OF USE (Seat Class) for which the CELL provides a price.
«FK»	***FacilitySetRef***	*FacilitySetRef*	0:1	A FACILITY SET for which the CELL provides a price.
«FK»	***TypeOfProduct-CategoryRef***	*TypeOfProduct-CategoryRef*	0:1	A TYPE OF PRODUCT CATEGORY for which the CELL provides a price.

«FK»	***TypeOfService-Ref***	*TypeOfServiceRef*	0:1	A TYPE OF SERVICE for which the CELL provides a price.
«FK»	***ServiceJourney-Ref***	*ServiceJourneyRef*	0:1	A SERVICE JOURNEY for which the CELL provides a price.
«FK»	***TrainNumberRef***	*TrainNumberRef*	0:1	A TRAIN NUMBER for which the CELL provides a price.
«FK»	***GroupOfServices Ref***	*GroupOfServicesRef*	0:1	A GROUP OF SERVICEs for which the CELL provides a price.

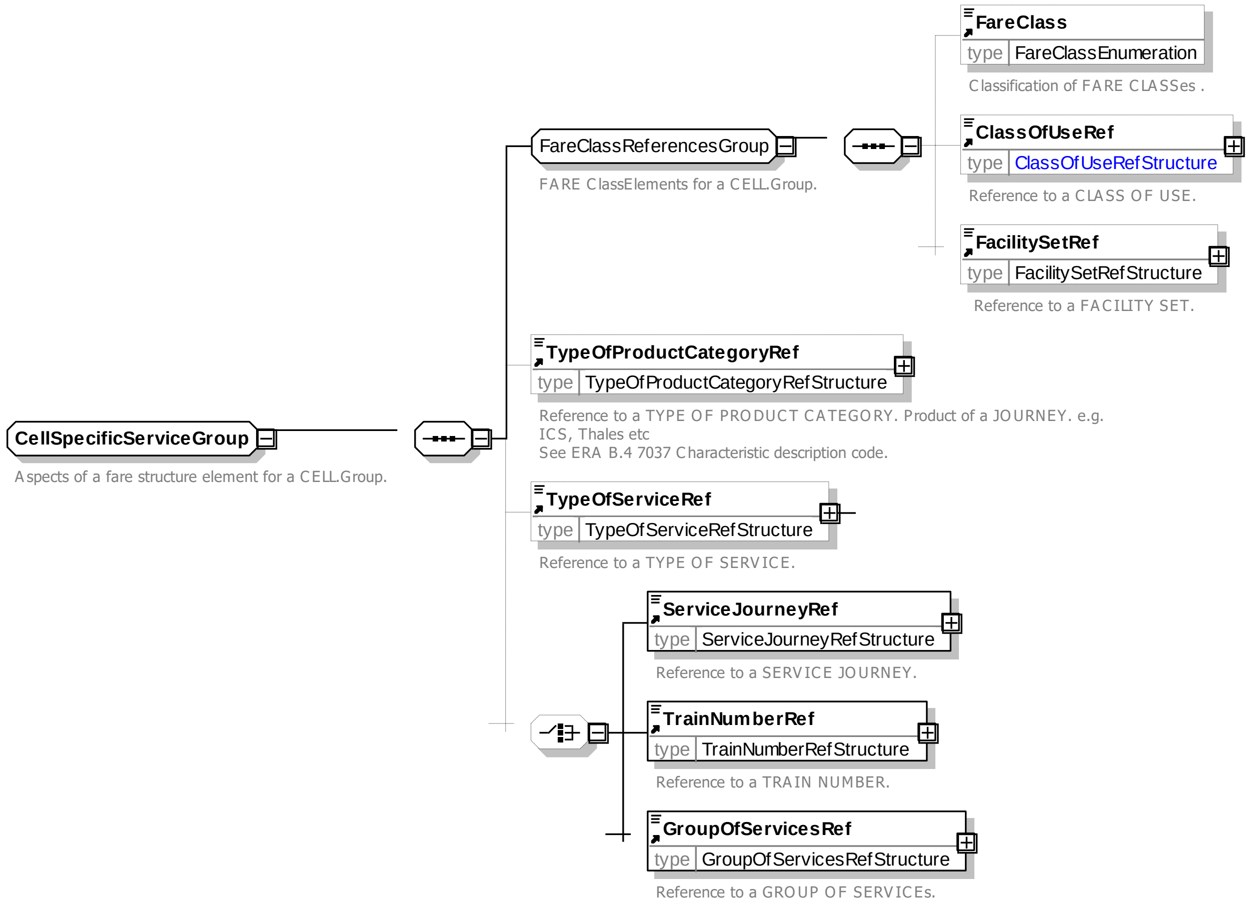


Figure 302 — CellSpecificServiceGroup — XSD

7.7.3.4.2.5 CellSpecificDistributionGroup – Group

References to the elements for which the CELL provides a price. May be a combination of multiple elements.

Table 284 – *CellReferencesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***TypeOfFare-ProductRef***	*TypeOfFareProductRef*	0:1	TYPE OF FARE PRODUCT for which the CELL provides a price.

«FK»	***Distribution-ChannelRef***	*DistributionChannelRef*	0:1	DISTRIBUTION CHANNEL for which the CELL provides a price.
«FK»	***GroupOf-Distribution-ChannelsRef***	*GroupOfDistribution-ChannelsRef*	0:1	GROUP OF DISTRIBUTION CHANNELs for which the CELL provides a price.
«enum»	***PaymentMethods***	*PaymentMethodEnum*	0:1	PaymentMethod standard value for which the CELL provides a price. See Part1.
«FK»	***TypeOfPayment-MethodRef***	*TypeOfPayment-MethodRef*	0:1	TYPE OF PAYMENT METHOD for which the CELL provides a price.
«FK»	***TypeOfTravel-DocumentRef***	*TypeOfTravelDocument-Ref*	0:1	TYPE OF TRAVEL DOCUMENT for which the CELL provides a price.

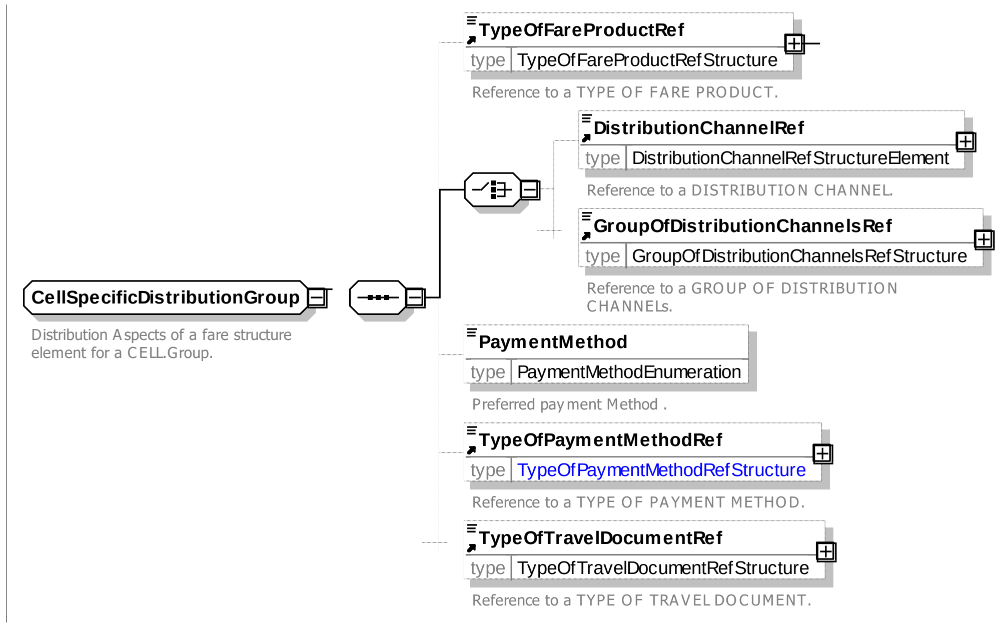


Figure 303 —

***CellSpecificDistributionGroup* — XSD**

7.7.3.4.2.6 CellHeadingsGroup – Group

References to the row and column of the FARE TABLE associated with CELL.

Table 285 – *CellHeadingsGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***ColumnRef***	*ColumnRef*	0:1	Reference to a column in the FARE TABLE to which this CELL should be assigned.
«FK»	***RowRef***	*RowRef*	0:1	Reference to a row in the FARE TABLE to which this CELL should be assigned.

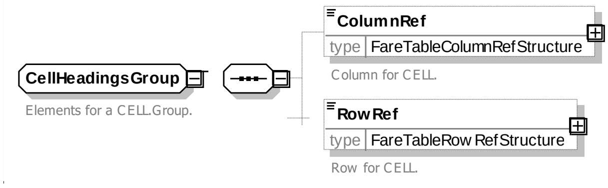


Figure 304 — *CellHeadingsGroup* — XSD

7.7.3.4.3 **StandardFareTable – Model Element**

A predefined grouping of four prices (First /Second Class x Single / Return) that may be used as an optimised FARE TABLE.

Table 286 – *StandardFareTable* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareTable*	::>	STANDARD FARE TABLE inherits from FARE TABLE.
«PK»	***id***	*StandardFare-TableIdType*	1:1	Identifier of STANDARD FARE TABLE.
	1stClass-SingleFare	*AmountType*	0:1	Price of a First-Class single fare.
	2ndClass-SingleFare	*AmountType*	1:1	Price of Second-Class single fare.
	1stClass-ReturnFare	*AmountType*	0:1	Price of a First Class return fare.
	2ndClass-ReturnFare	*AmountType*	0:1	Price of a Second Class return fare.

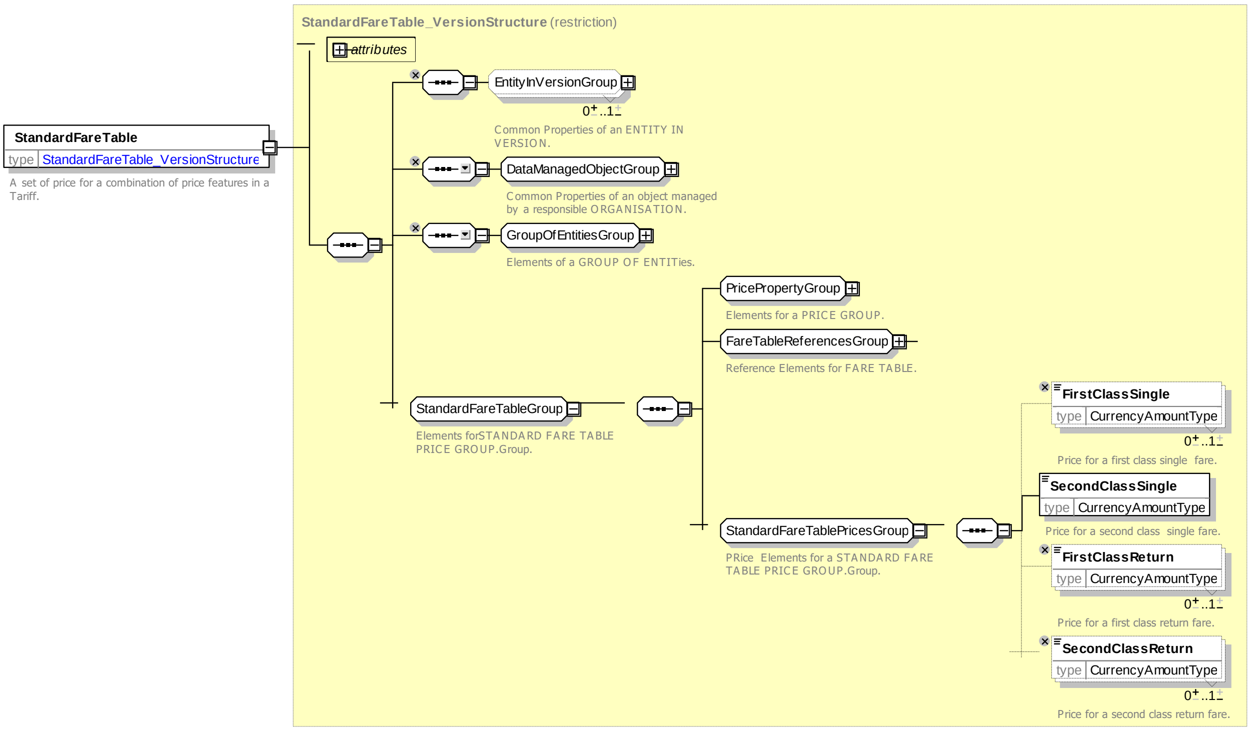


Figure 305 — *StandardFareTable* — XSD

7.7.3.4.1 TypeOfFareTable – Model Element

A classification of a FARE TABLE.

Table 287 – *TypeOfFareTable* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF FARE TABLE inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfFareTableIdType*	1:1	Identifier of TYPE OF FARE TABLE.

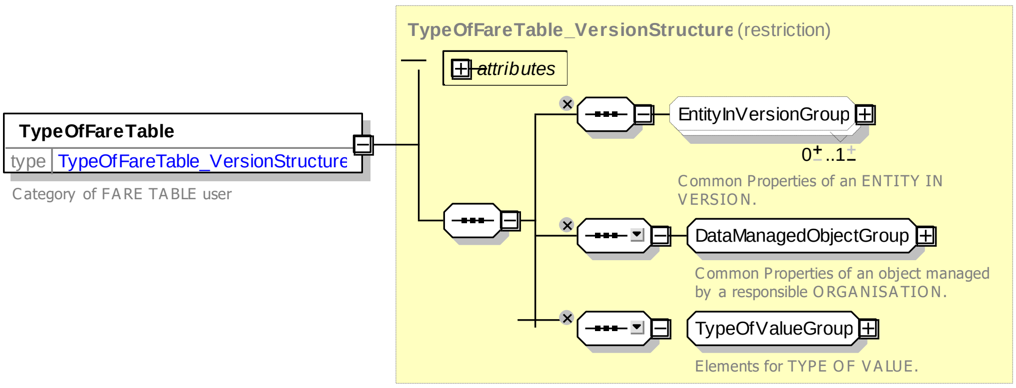


Figure 306 — *TypeOfFareTable* — XSD

7.7.3.5 Fare Table – XML examples

7.7.3.5.1 Fare Table: XML Example of Cell prices

The following partial table shows FARE TABLE with a set of CELLs with prices for various combinations of USER PROFILE (set at the table level) FARE STRUCTURE ELEMENT, FARE STRUCTURE FACTOR, and FARE PRODUCT (set at the CELL level).

```xml
<FareTable version="any" id="lbsl:Bus_Tram@18Plus">
        <Name> Bus Fare Prices - 18+Student </Name>
        <UserProfileRef version="any" ref="tfl:18Plus"/>
        <cells>
            <Cell version="any" id="lbsl:Bus_Tram@Cash@18Plus">
                <Price>
                    <Name>Single fare </Name>
                    <Amount>2.40</Amount>
                </Price>
                <FareStructureElementRef version="any" ref="lbsl:bus_or_tram_trip"/>
                <PreassignedFareProductRef version="any" ref="tfl:Prepaid_fare"/>
            </Cell>
            <Cell version="any" id="lbsl::Bus_Tram@Oyster@Day@18Plus">
                <Price>
                    <Name>Single fare Oyster with capping</Name>
                    <Amount>1.40</Amount>
                    <MaximumLimitPrice>4.40</MaximumLimitPrice>
                </Price>
                <TimeStructureFactorRef version="any" ref="tfl:1d"/>
                <FareStructureElementRef version="any" ref="tfl:1DayPass"/>
                <PreassignedFareProductRef version="any" ref="tfl:PayAsYouGo_fare"/>
            </Cell>
            <Cell version="any" id="lbsl:Bus_Tram@Oyster@Week@18Plus">
                <Price>
                    <Name>weekly TravelCard_on_Oyster Bus pass</Name>
                    <Amount>13.70</Amount>
                </Price>
                <TimeStructureFactorRef version="any" ref="tfl:1w"/>
                <FareStructureElementRef version="any" ref="tfl:1WeekPass"/>
                <PreassignedFareProductRef version="any" ref="tfl:TravelCard_on_Oyster"/>
            </Cell>
            <Cell version="any" id="lbsl:Bus_Tram@Oyster@Monthly@18Plus">
                <Price>
                    <Name>Monthly TravelCard_on_Oyster Bus pass</Name>
                    <Amount>52.70</Amount>
                </Price>
                <TimeStructureFactorRef version="any" ref="tfl:1m"/>
                <FareStructureElementRef version="any" ref="tfl:1MonthPass"/>
                <PreassignedFareProductRef version="any" ref="tfl:TravelCard_on_Oyster"/>
            </Cell>
        </cells>
    </FareTable>
```

7.7.3.5.2 Fare Table: XML Example of Cells with Referenced prices

The following partial table shows a set of cells with prices for various combinations of QUALITY STRUCTURE FACTOR, SALE DISCOUNT RIGHT (both set at the tabek level) and VALIDABLE ELEMENT (Set at the CELL level). Some standard concesstions PRICEsare referenced.

```xml
<fareTables>
    <FareTable version="any" id="tfl:FareTable:FreedomPass@Londoner">
        <Name>Freedom Pass prices for Londoners</Name>
        <QualityStructureFactorRef version="any" ref="tfl:Resident"/>
        <SaleDiscountRightRef version="any" ref="tfl:FreedomPass"/>
        <cells>
            <Cell version="any" id="tfl:FreedomPass@Londoner@bus" order="1">
                <UsageParameterPriceRef version="any" ref="tfl:concession@free"/>
                <ValidableElementRef version="any" ref="lbsl:ValidableElement:busOrTramTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Londoner@metro" order="2">
                <UsageParameterPriceRef version="any" ref="tfl:concession@free"/>
                <ValidableElementRef version="any" ref="lul:ValidableElement:metroTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Londoner@cableway" order="3">
```

```
                <UsageParameterPriceRef version="any" ref="tfl:concession@34pct"/>
                <ValidableElementRef version="any" ref="ea:ValidableElement:cablewayTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Londoner@water" order="4">
                <UsageParameterPriceRef version="any" ref="tfl:concession@34pct"/>
                <ValidableElementRef version="any" ref="lrs:ValidableElement:riverTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Londoner@overground" order="5">
                <UsageParameterPriceRef version="any" ref="tfl:concession@free"/>
                <ValidableElementRef version="any" ref="nr:ValidableElement:suburbanRailTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Londoner+national_rail" order="6">
                <UsageParameterPriceRef version="any" ref="tfl:concession@34pct"/>
                <ValidableElementRef version="any" ref="nr:ValidableElement:railTrip"/>
            </Cell>
        </cells>
    </FareTable>
    <FareTable version="any" id="tfl:FareTable:FreedomPass@Non_Londoner">
        <Name>Freedom Pass prices for Non Londoners</Name>
        <QualityStructureFactorRef version="any" ref="tfl:Non_Resident"/>
        <SaleDiscountRightRef version="any" ref="tfl:FreedomPass"/>
        <cells>
            <Cell version="any" id="tfl:FreedomPass@Non_Londoner@bus" order="1">
                <Name>Bus: Travel free at any time on buses in London showing this symbol </Name>
                <UsageParameterPriceRef version="any" ref="tfl:concession@free"/>
                <ValidableElementRef version="any" ref="lbsl:ValidableElement:busOrTramTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Non_Londoner@cableway" order="2">
                <UsageParameterPriceRef version="any" ref="tfl:concession@34pct"/>
                <ValidableElementRef version="any" ref="ea:ValidableElement:cablewayTrip"/>
            </Cell>
            <Cell version="any" id="tfl:FreedomPass@Non+Londoner@water" order="3">
                <UsageParameterPriceRef version="any" ref="tfl:concession@34pct"/>
                <ValidableElementRef version="any" ref="lrs:ValidableElement:riverTrip"/>
            </Cell>
        </cells>
    </FareTable>
```

7.7.3.5.3 Fare Table: XML Example of Simple prices

The following partial table shows a simple fare prices used as sells for combinations of USER PROFILE, ROUND TRIP, FARE DEMAND FACTOR and PREASSIGNED FARE (all set at the tabek level) with VALIDABLE ELEMENT (Set at the CELL level).

```
<FareTable version="any" id="nr:FareTable:OysterRailOnly@Single@offPeak@Adult@WatfordJunction">
    <Name> Fare Prices - Oyster Off-Peak Single Rail Only - Watford Junction to Zones</Name>

    <UserProfileRef version="any" ref="tfl:adult"/>
    <RoundTripRef version="any" ref="tfl:single"/>
    <FareDemandFactorRef version="any" ref="tfl:offPeak"/>
    <PreassignedFareProductRef version="any" ref="tfl:PayAsYouGo_fare"/>
    <prices>
        <DistanceMatrixElementPrice version="any"
id="nr:OysterRailOnly@Single@offPeak@Adult@Watford_Junction@Zone_1">
            <Name>Zone 1 to Watford Junction*</Name>
            <Amount>4.50</Amount>
            <DistanceMatrixElementRef version="any" ref="nr:Watford_Junction@Zone_1"/>
        </DistanceMatrixElementPrice>

        <DistanceMatrixElementPrice version="any"
id="nr:OysterRailOnly@Single@offPeak@Adult@Watford_Junction@Zone_2">
            <Name>Zone 2 to Watford Junction*</Name>
            <Amount>2.70</Amount>
            <DistanceMatrixElementRef version="any" ref="nr:Watford_Junction@Zone_2"/>
        </DistanceMatrixElementPrice>

        <DistanceMatrixElementPrice version="any"
id="nr:OysterRailOnly@Single@offPeak@Adult@Watford_Junction@Zone_3">
            <Name>Zone 3 to Watford Junction*</Name>
            <Amount>2.20</Amount>
            <DistanceMatrixElementRef version="any" ref="nr:Watford_Junction@Zone_3"/>
        </DistanceMatrixElementPrice>
    </prices>
```

7.8 Sales Description

7.8.1 Fare Sales Distribution

7.8.1.1 Fare Sales Distribution – Conceptual MODEL

Passenger information may need to include information on where particular products may be purchased. The FARE DISTRIBUTION MODEL specifies rules for where and how products may be purchased, for example over the counter, on-line, from self-service ticket machines, etc. SALES OFFER PACKAGEs can be restricted to specific DISTRIBUTION CHANNELs or GROUPs OF DISTRIBUTION CHANNELs using a DISTRIBUTION ASSIGNMENT (see SALES OFFER PACKAGE Model).

The concerns of DISTRIBUTION CHANNEL – how a product may be purchased - and FULFILMENT METHOD – how a purchase is subsequently delivered – are separated as they may be distinct. For example, a product bought on-line might be fulfilled either by mail, self-printing, collection from a machine, or by automatic adding to an on-line account. Where distribution, or the execution of certain functions such as refunds is limited to certain points of sale this can also be indicated.

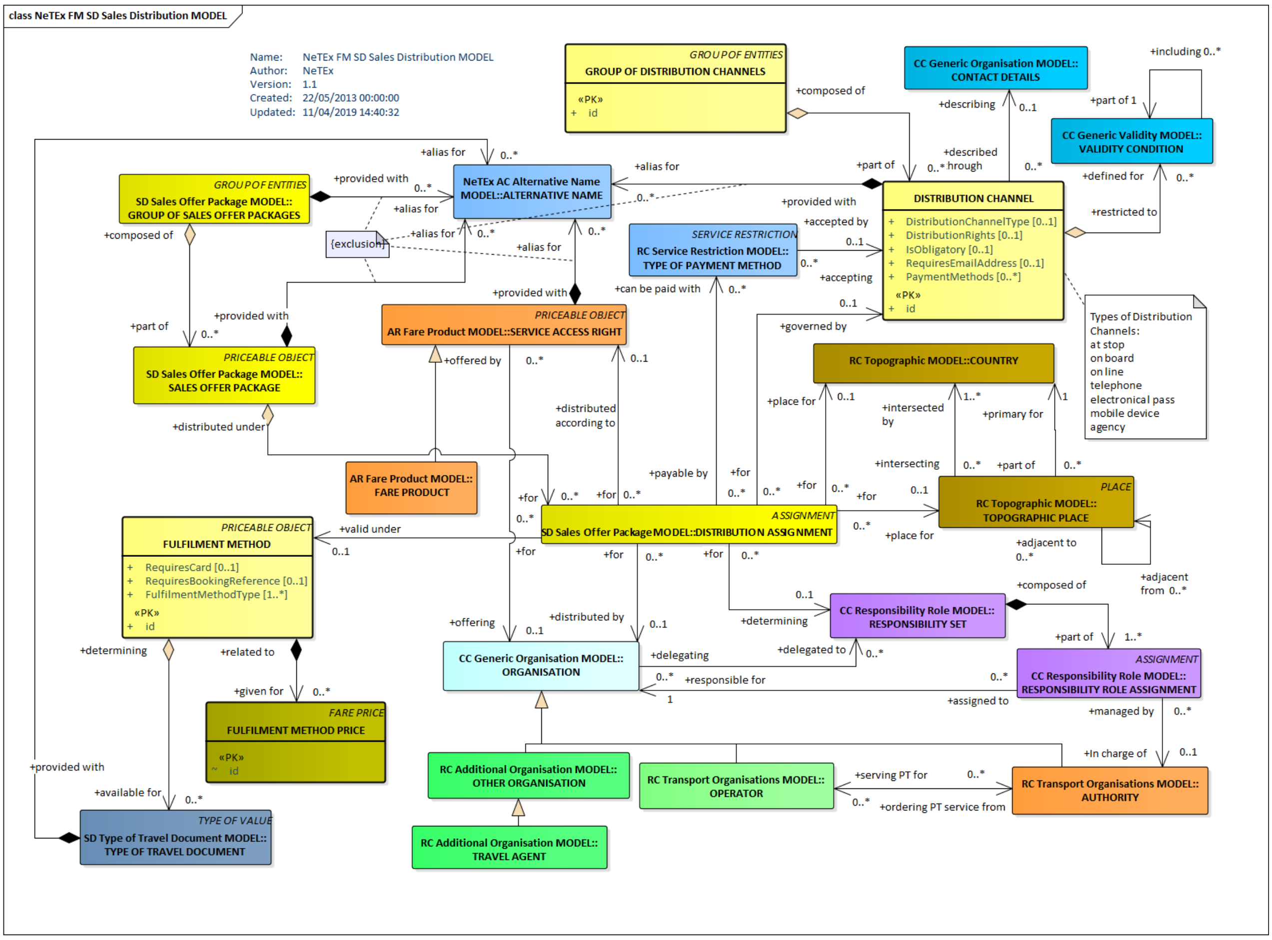


Figure 307 — Sales Distribution – Conceptual MODEL (UML)

7.8.1.2 Fare Sales Distribution – Physical model

The following figure shows the physical model for SALES DISTRIBUTION.

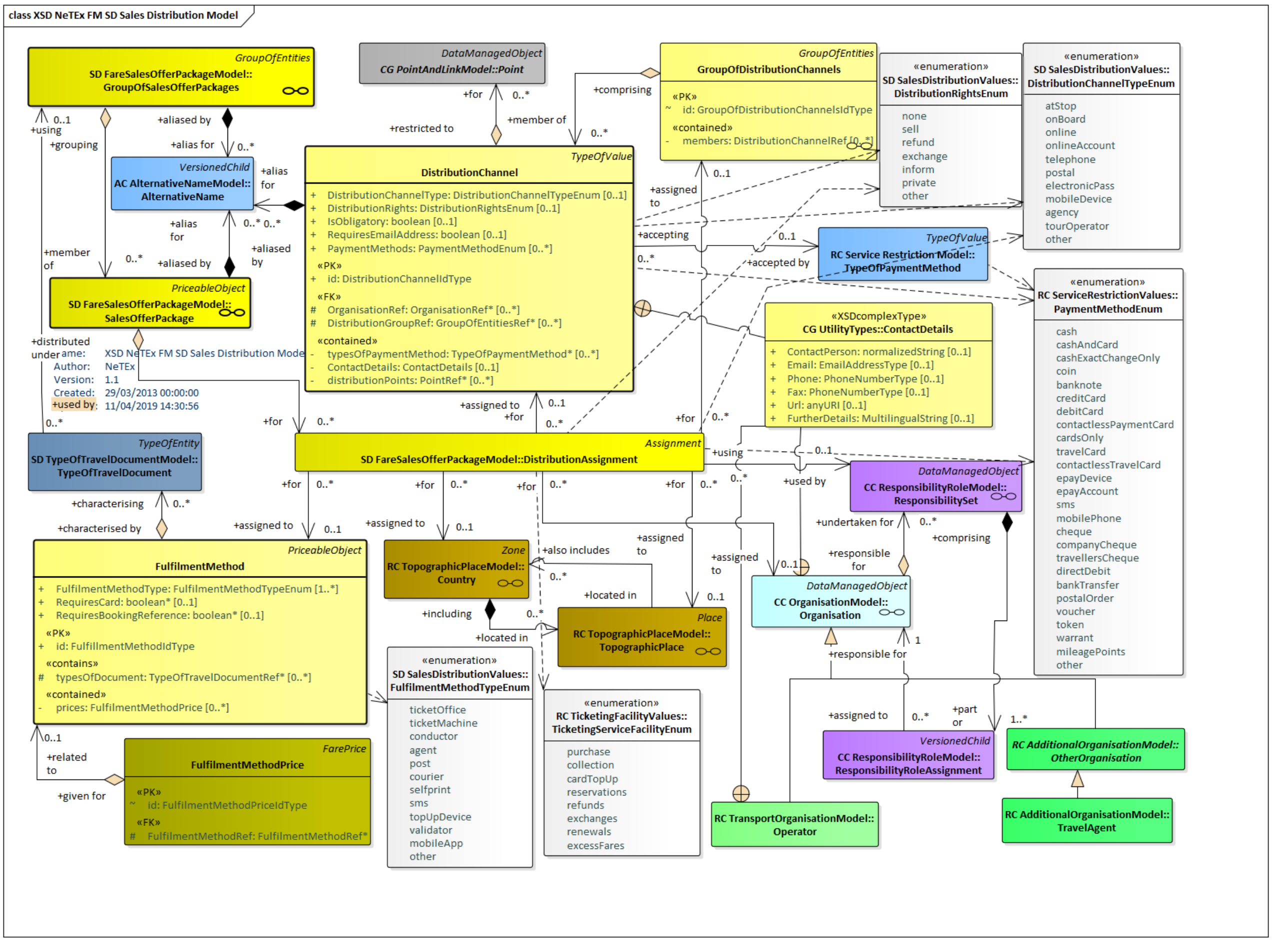


Figure 308 — Sales Distribution – Physical Model (UML)

7.8.1.3 Fare Sales Distribution Classifications – Physical model

The following figure shows the standard values available to classify SLES OFFER and SALES DISTRIBUTION elements.

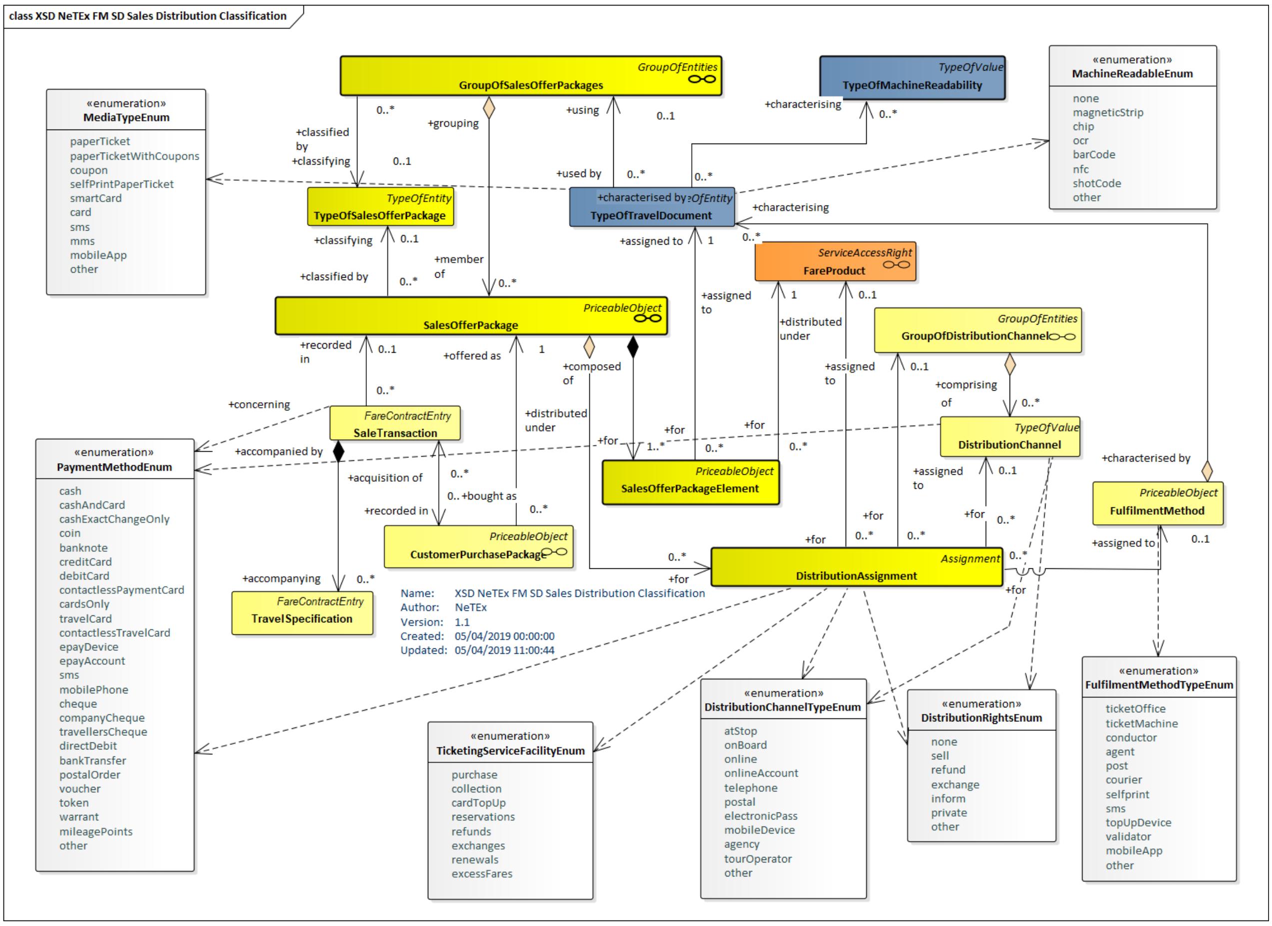


Figure 309 — Sales Distribution Classification – Physical Model (UML)

7.8.1.4 Fare Sales Distribution – Attributes and XSD

7.8.1.4.1 DistributionChannel – Model Element

A type of outlet for selling a product.

Table 288 – *DistributionChannel* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	DISTRIBUTION CHANNEL inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*DistributionChannelIdType*	1:1	Identifier of a DISTRIBUTION CHANNEL.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for DISTRIBUTION CHANNEL.
«enum»	***Distribution-ChannelType***	*DistributionChannel-TypeEnum*	0:1	Type of DISTRIBUTION CHANNEL. See below for allowed values.
	IsObligatory	*xsd:boolean*	0:1	Whether the option to use the channel is obligatory, that is, must be allowed.
	RequiresEmail-Address	*xsd:boolean*	0:1	Whether to use the channel requires an email address.

«FK»	***OrganisationRef***	*(OrganisationRef)*	0:1	ORGANISATION associated with channel.
«enum»	***PaymentMethods***	*PaymentMethodEnum*	0:*	Payment methods supported on this distribution. See NeTEx Part1 for allowed values.
«cntd»	***typesOfPayment Method***	*TypeOfPaymentMethodRef*	0:*	PAYMENT METHOD – open type. +v 1.1
«enum»	***Distribution-Rights***	*DistributionRightsEnum*	0:1	Default distribution rights for the DISTRIBUTION CHANNEL.
«cntd»	***distribution-Points***	*PointRef+*	0:*	Points to which distribution is restricted, if any. For example, that a ticket can only be bought at a specific station.
«FK»	***Distribution-GroupRef***	*GroupOfEntitiesRef*	0:1	GROUP OF ENTITies, e.g. places, organisations or other entities (E.g. on-board specific journeys or services places) to which distribution is restricted, if any. For example, that a ticket can only be bought at a specific station.

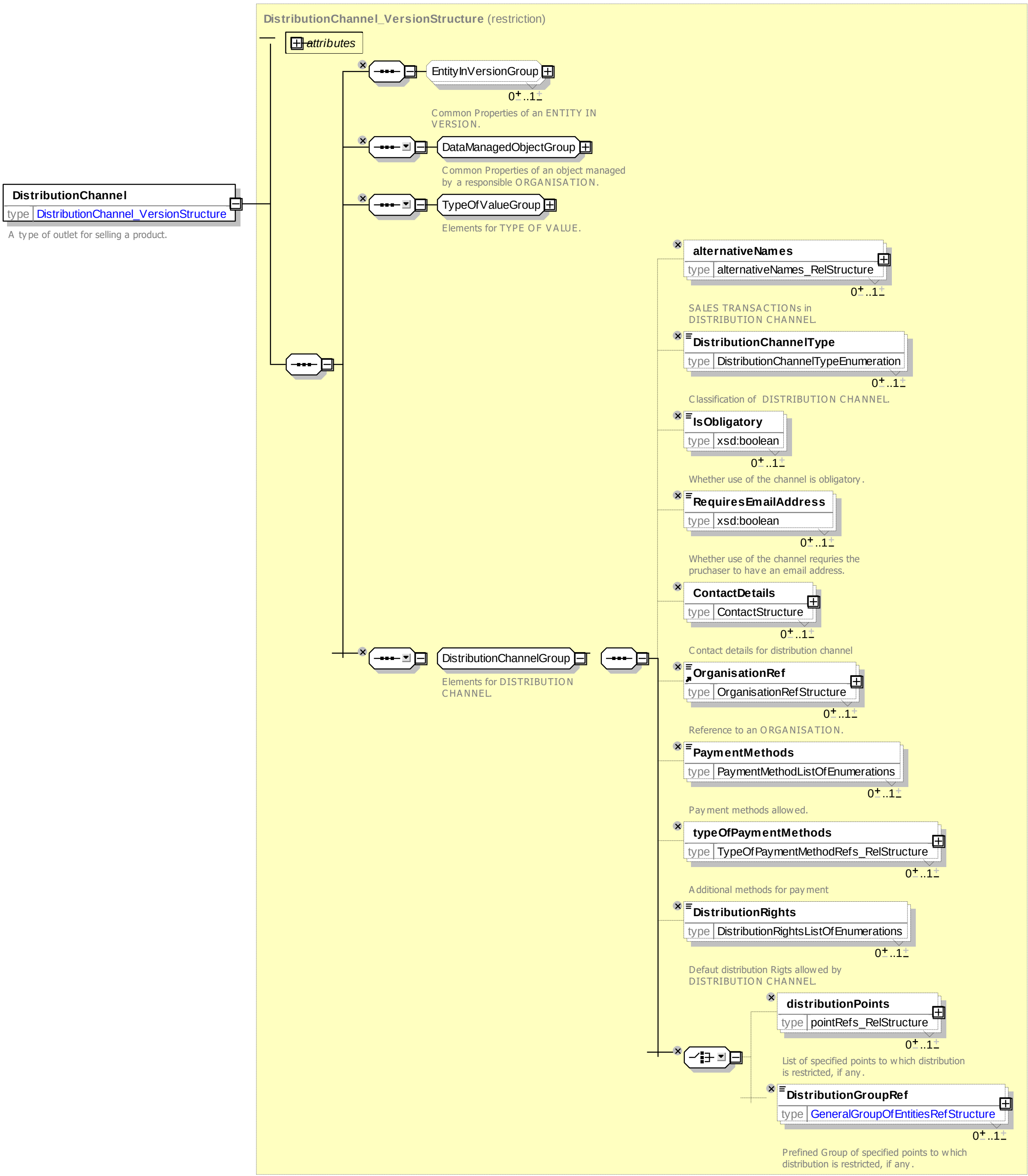


Figure 310 — *DistributionChannel* — XSD

7.8.1.4.1.1 DistributionChannelType – Allowed values

The following table shows the allowed values for ***DistributionChannelType***. (*DistributionChannelTypeEnumeration).*

Table 289 – *DistributionChannelType* – Allowed values

Value	Description
atStop	At stop sales.
onBoard	On-board sales.

online	On-line sales.
onlineAccount	On-line sales using online account, e.g. for subscription. +v1.1
telephone	Telephone sales.
electronicPass	Electronic sale.
mobileDevice	Mobile device.
agency	Third party agency.
tourOperator	Tour operator.
other	Another channel.

7.8.1.4.1.2 DistributionRights – Allowed values

The following table shows the allowed values for ***DistributionRights*** (*DistributionRightsEnumeration*)

Table 290 – *DistributionRights* – Allowed values

Value	Description
none	No right to distribute product.
sell	Distributor may sell product.
refund	Distributor may refund product.
exchange	Distributor may exchange product.
inform	Distributor may inform public of product.
private	Distributor may use information about product for own purposes but not publicize it.
other	Other channel.

7.8.1.4.2 GroupOfDistributionChannels – Model Element

A grouping of DISTRIBUTION CHANNELs.

Table 291 – *GroupOfDistributionChannels* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*GroupOfEntities*</u>	::>	GROUP of DISTRIBUTION CHANNELs inherits from GROUP OF ENTITIES. See NeTEx Part1.
«PK»	***id***	*GroupOfDistribution-ChannelsIdType*	1:1	Identifier of GROUP of DISTRIBUTION CHANNELs.
«cntd»	***members***	*DistributionChannelRef*	0:*	References to DISTRIBUTION CHANNELs that are members of group.

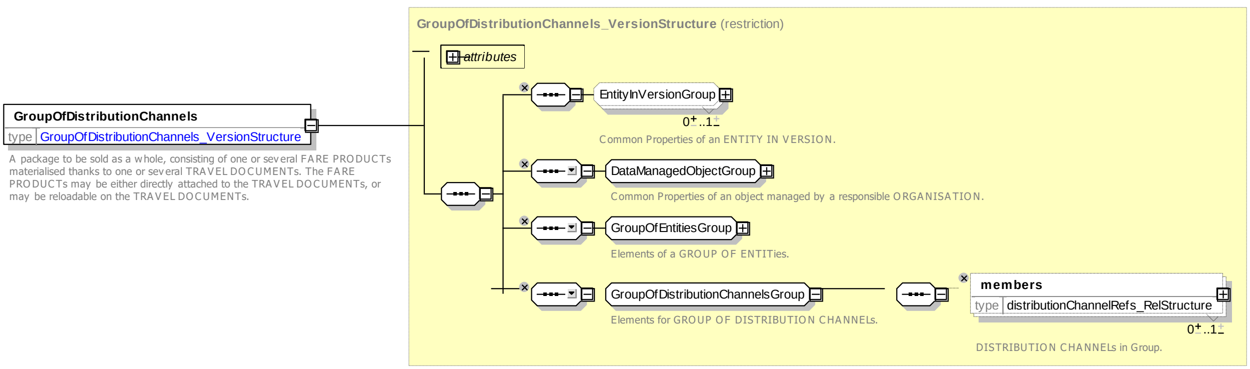


Figure 311 — *GroupOfDistributionChannels* — XSD

7.8.1.4.3 FulfilmentMethod – Model Element

The means by which the ticket is delivered to the Customer. e.g. online, collection, etc.

Table 292 – *FulfilmentMethod* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PriceableObject*	::>	FULFILMENT METHOD inherits from PRICEABLE OBJECT.
«PK»	***id***	*FulfillmentMethodIdType*	1:1	Identifier of FULFILMENT METHOD.
«enum»	***Fulfilment-MethodType***	*FulfilmentMethod-TypeEnum*	0:1	Type of FULFILMENT METHOD. See allowed values below.
	RequiresCard	*xsd:boolean*	0:1	Whether collecting ticket requires credit card used to purchase.
	Requires-Booking-Reference	*xsd:boolean*	0:1	Whether collecting ticket requires booking reference.
«cntd»	***typesOf-Document***	*TypeOfTravel-DocumentRef*	0:*	Reference to TYPEs OF TRAVEL DOCUMENT allowed by method.
«cntd»	***prices***	*FulfilmentMethodPrice*	0:*	FULFILMENT METHOD PRICEs associated with the FULFILMENT METHOD.

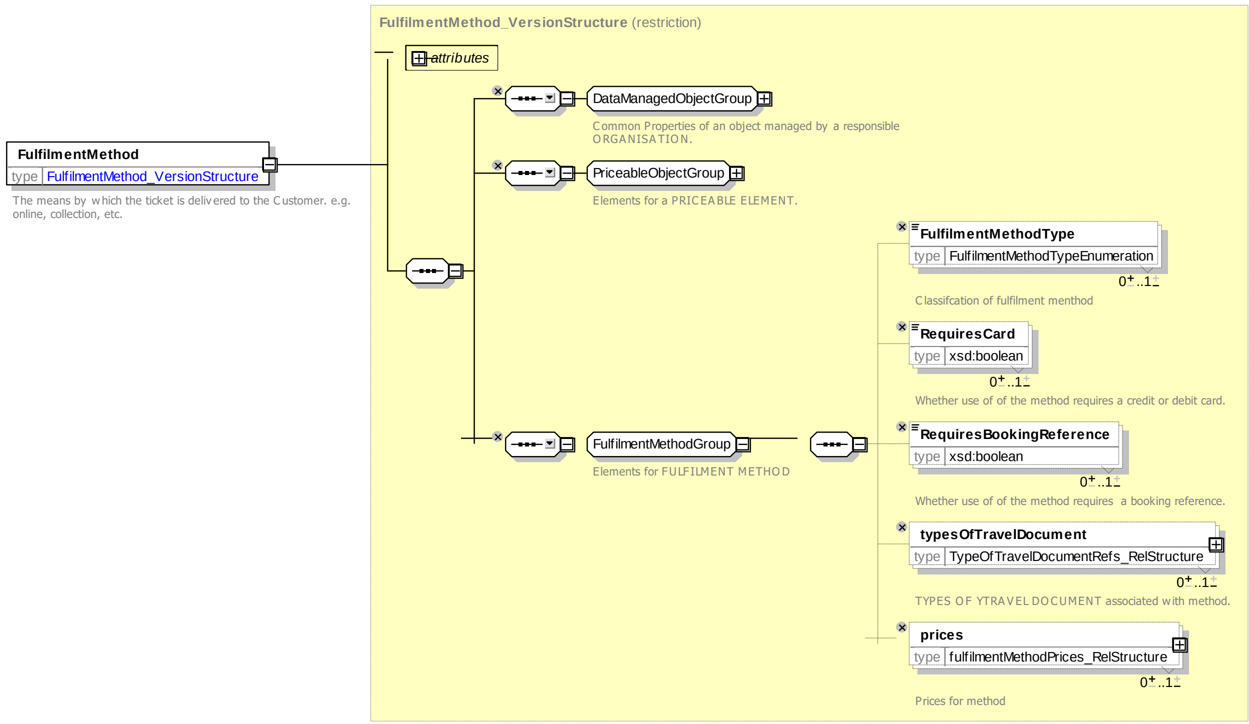


Figure 312 — *FulfilmentMethod* — XSD

7.8.1.4.3.1 FulfilmentMethodType – Allowed values

The following table shows the allowed values for ***FulfilmentMethodType*** (*FulfilmentMethodType-Enumeration*).

Table 293 – *FulfilmentMethodType* – Allowed values

Value	Description
ticketOffice	Fulfilment is at a ticket office in a station.
ticketMachine	Fulfilment is using a self-service machine.
conductor	Fulfilment by conductor on board.
agent	Fulfilment is by a travel agent.
post	Fulfilment is by post.
courier	Fulfilment is by courier +v1.1.
selfprint	Ticket is printed by customer.
sms	Fulfilment is by SMS.
topUpDevice	Fulfilment is by automatic top up of a travel card using a device e.g. in station machine, or ATM machine.
validator	Fulfilment is by automatic charging by a validator.
mobileApp	Fulfilment is by a mobile application.
other	Fulfilment is by some other method.

7.8.1.4.4 FulfilmentMethodPrice – Model Element

A set of all possible price features of a FULFILMENT METHOD, default total price etc.

Table 294 – *FulfilmentMethodPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*FarePrice*</u>	::>	FULFILMENT METHOD PRICE inherits from FARE PRICE
«PK»	***id***	*FulfilmentMethod-PriceIdType*	1:1	Identifier of FULFILMENT METHOD PRICE.
«FK»	***Fulfilment-MethodRef***	*FulfilmentMethodRef*	0:1	Reference to a FULFILMENT METHOD for which this is the price. If not given by containing context must be specified.

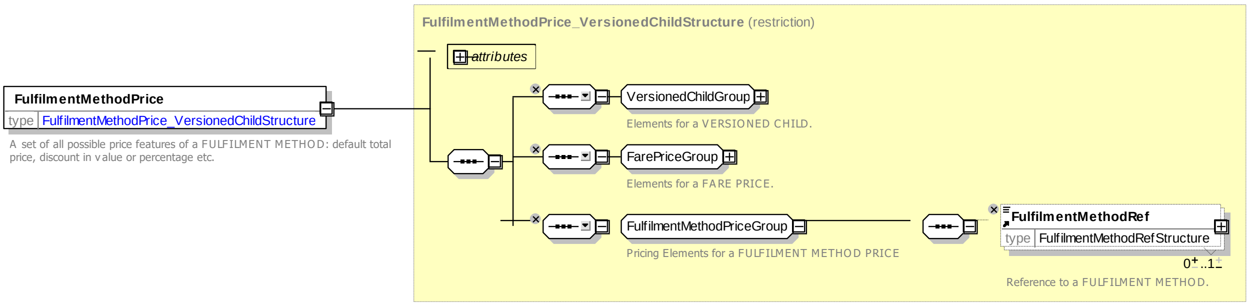


Figure 313 — *FulfilmentMethodPrice* — XSD

7.8.1.4.5 **Fare Sales Distribution – XML examples Fare Sales Distribution: XML Example of fulfilment methods**

The following code fragment indicates which of certain TYPEs OF TRAVEL DOCUMENT can be distributed using a number of different FULFILMENT METHOD.

For EXAMPLE:

```
<fulfilmentMethods>

        <FulfilmentMethod version="any" id="tfl:Self_Service_Machine">
            <Name>Product is delivered by machine</Name>
            <FulfilmentMethodType>ticketMachine</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
            </typesOfTravelDocument>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl:Over_Counter">
            <Name>Product is delivered over counter</Name>
            <FulfilmentMethodType>ticketOffice</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
            </typesOfTravelDocument>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl:Validator">
            <Name>Product is delivered electronically by touch to validator at barrier</Name>
            <FulfilmentMethodType>validator</FulfilmentMethodType>
            <RequiresCard>true</RequiresCard>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:nfcPaymentCard"/>
            </typesOfTravelDocument>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl:Electronic_Collection">
            <Name>Product previously purchased is delivered electronically by device at designated
collection point, e.g. touch to validator at barrier</Name>
            <FulfilmentMethodType>validator</FulfilmentMethodType>
            <RequiresCard>true</RequiresCard>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
            </typesOfTravelDocument>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl:Normal_post">
            <Name>Product is delivered by normal post</Name>
            <FulfilmentMethodType>post</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:membershipCard"/>
            </typesOfTravelDocument>
            <prices>
                <FulfilmentMethodPrice version="any" id="tfl:Normal_post">
            <Amount>0.00</Amount>
                </FulfilmentMethodPrice>
            </prices>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl:Registered_post">
            <Name>Product is delivered by registered post</Name>
            <FulfilmentMethodType>post</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl:membershipCard"/>
```

```
            </typesOfTravelDocument>
            <prices>
                <FulfilmentMethodPrice version="any" id="tfl:Registered_post">
            <Amount>5.65</Amount>
                </FulfilmentMethodPrice>
            </prices>
        </FulfilmentMethod>
</fulfilmentMethods>
```

7.8.1.4.6 Fare Sales Distribution: XML Example of fulfilment methods

The following code fragment shows a DISTRIBUTION ASSIGNMENT to limit the sale of Visitor oyster cards to online purchase from TfL Visitor centres with postal distribution.

For EXAMPLE:

```
    <SalesOfferPackage version="any" id="tfl:SalesOfferPackage:VisitorOysterCard">
            <Name>6Visitor Oyster Product is carried on Oyster card </Name>
            <Description>Visitor Oyster cards are plastic smartcards you can use instead of paper
tickets. Put pay as you go credit on it which you use when you travel. It is the cheapest way to
pay for single journeys on bus, Tube, tram, DLR, London Overground and most National Rail services
in London.

Visitor Oyster cards are valid across all travel zones in London and automatically calculate the
best value fare for all the journeys you make in a single day if you have touched in and out.
Visitor Oyster card are only available for adults. However, an adult with a Visitor Oyster card
can take up to 4 children under the age of 11 years to travel for free on Tube, DLR and London
Overground services.
            <distributionAssignments>
                <DistributionAssignment version="any" id="tfl:VisitorOysterCard" order="1">
                    <Name>Can o </Name>
                    <DistributionChannelRef version="any" ref="tfl:Visitor_centres"/>
                    <DistributionChannelType>online</DistributionChannelType>
                    <AllowedInChannel>true</AllowedInChannel>
                    <RestrictedToChannel>true</RestrictedToChannel>
                    <TicketingServiceFacilityList>purchase</TicketingServiceFacilityList>
                    <PaymentMethods>cardsOnly</PaymentMethods>
                    <RequiresRegistration>true</RequiresRegistration>
                    <FulfilmentMethodRef version="any" ref="tfl:NormalPost"/>
                </DistributionAssignment>
            </distributionAssignments>
            <SalesOfferPackageRef version="any" ref="tfl:OnOysterCard"/>
        </SalesOfferPackage>
```

7.8.2 Type of Travel Document

7.8.2.1 TYPE OF TRAVEL DOCUMENT – Conceptual MODEL

The TYPE OF TRAVEL DOCUMENT MODEL indicates the available materialisations of products as tickets on media.

The TRAVEL DOCUMENT entity describes an individual physical support satisfying a TYPE OF TRAVEL DOCUMENT, which may be loaded with various contents: sold FARE PRODUCTs, results of VALIDATION ENTRies, CUSTOMER identification, etc.

TRAVEL DOCUMENTs are usually allocated to customers on the occasion of a SALE TRANSACTION.

In most cases, TRAVEL DOCUMENTs are individually managed in an operator database, if they belong belonging to identified customers (reloadable value card, discount right document, etc.). This is of course mandatory for post payment methods. In general cases, the sale or control processes only register the category of the TRAVEL DOCUMENT.

TRAVEL DOCUMENTs are classified by a TYPE OF TRAVEL DOCUMENT, which expresses:

— their general characteristics (type of medium, types of compatible fare products, etc.);

— their local functional characteristics, specific to the operator or the authority (specific fare products stored on this type, type of retailer, etc.).

The classical general TYPEs OF TRAVEL DOCUMENTs include the following:

— single-use throw-away ticket, giving the right to consume only one VALIDABLE ELEMENT (e.g. one trip);

— throw-away ticket unit, for which the access right is granted by using a certain number of throw-away units (generally by punching them together in a validator);

— value card, debited by a certain amount for each consumption of VALIDABLE ELEMENTs;

— reloadable electronic purse, allowing access to the PT network; debited by each purchase.

— PT credit card, with post-payment on a central account;

— document attesting the right to benefit from a discount;

— etc.

The fare applied to an access right may be expressed not directly in currency but by an abstract unit. The entity PRICE UNIT lists all possible units used to express prices: amount of currency, token, abstract fare unit to be debited from value card, etc.

A relationship between TYPE OF TRAVEL DOCUMENT and PRICE UNIT expresses that the TRAVEL DOCUMENT may materialise a specific PRICE UNIT (throw-away ticket unit).

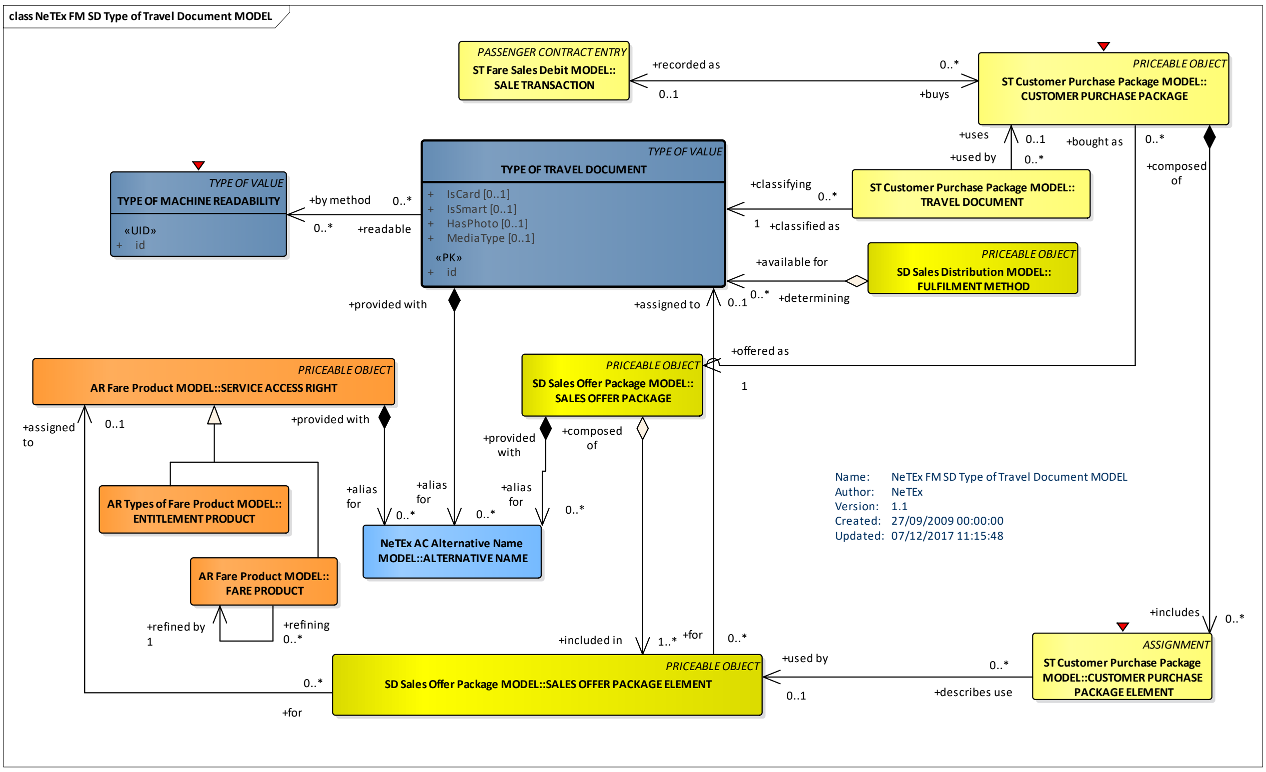


Figure 314 — Type of Travel Document – Conceptual MODEL (UML)

7.8.2.2 Type of Travel Document – Physical model

The following figure shows the physical model for TYPE OF TRAVEL DOCUMENTs.

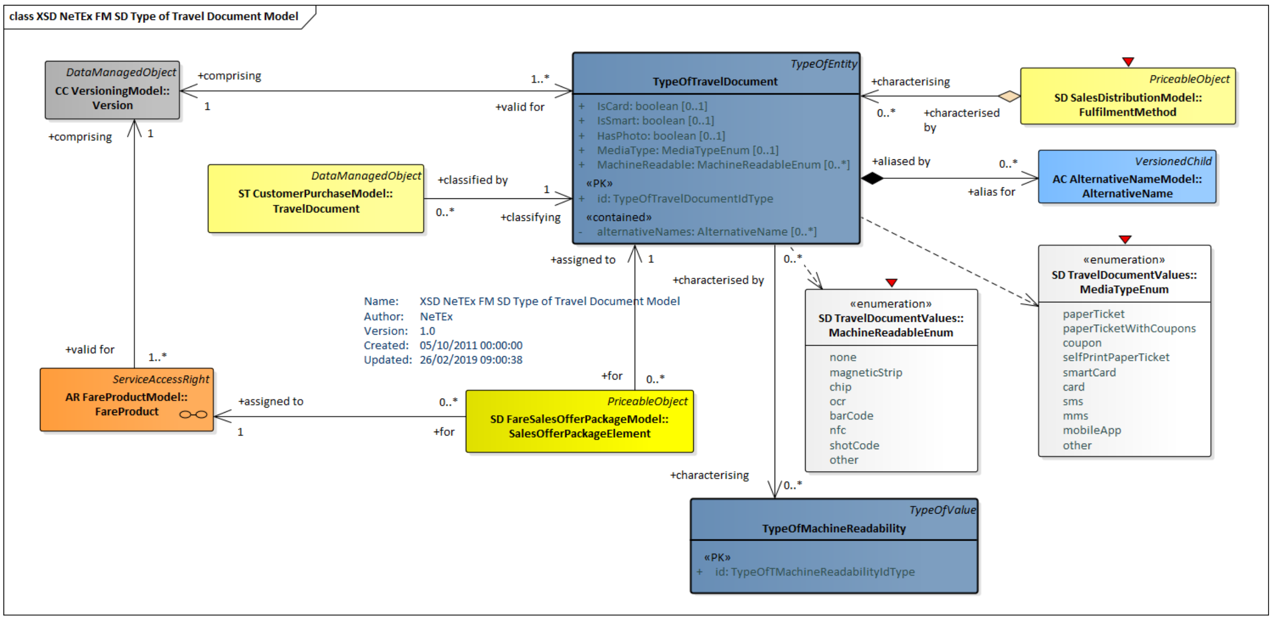


Figure 315 — Type of Travel Document – Physical Model (UML)

7.8.2.3 Fare Travel Document – Attributes and XSD

7.8.2.3.1 TypeOfTravelDocument – Model Element

A classification of TRAVEL DOCUMENTs expressing their general function and local functional characteristics specific to the operator. Types of TRAVEL DOCUMENTs like e.g. throw-away ticket, throw-away ticket unit, value card, electronic purse allowing access, public transport credit card, etc. may be used to define these categories.

Table 295 – *TypeOfTravelDocument* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*TypeOfValue*</u>	::>	TYPE OF TRAVEL DOCUMENT inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*TypeOfTravel-DocumentIdType*	1:1	Identifier of TYPE OF TRAVEL DOCUMENT.
	IsCard	*xsd:boolean*	0:1	Whether the TRAVEL DOCUMENT is materialised as a card.
	IsSmartCard	*xsd:boolean*	0:1	Whether the TRAVEL DOCUMENT is materialised on a smart card or mobile device.
	HasPhoto	*xsd:boolean*	0:1	Whether the TRAVEL DOCUMENT has a photo.

«enum»	***MediaType***	*MediaTypeEnum*	0:1	Classification of the TRAVEL DOCUMENT by Media type. See allowed values below.
«enum»	***Machine-Readable***	*MachineReadableEnum*	0:1	Classification of the TRAVEL DOCUMENT by Machine Readable mechanism standard values. See allowed values below.
«cntd»	***typesOfMachine Readabilities***	*TypeOfMachine-ReadabilityRef*	0:*	Classification of the TRAVEL DOCUMENT by TYPE OF MACHINE READABILITY.
«cntd»	***alternativeNames***	*AlternativeName*	0:*	ALTERNATIVE NAMEs for element.

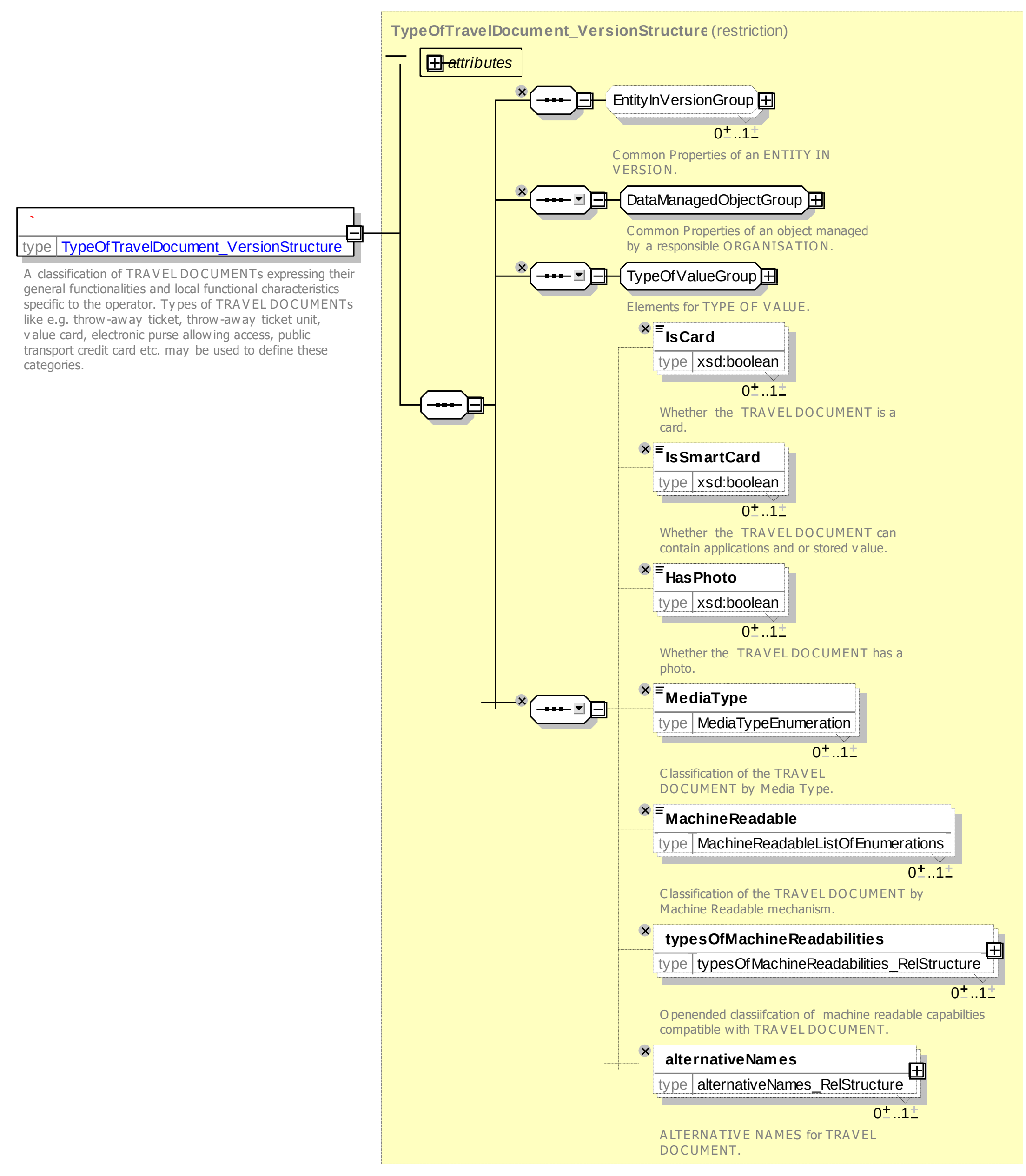


Figure 316 — *TypeOfTravelDocument* — XSD

7.8.2.3.1.1 MachineReadable – Allowed values

The following table shows the allowed values for ***MachineReadable*** *(MachineReadableEnumeration).*

Table 296 – *MachineReadable* – Allowed values

Value	Description
none	Not Machine readable.
magneticStrip	Readable by magnetic strip reader.
chip	Readable by chip reader
ocr	Readable by Optical Character Recognition
barCode	Readable by bar code reader.
shotCode	Shot code.
nfc	Readable by Near Field Communication.
other	Other channel.

7.8.2.3.1.2 MediaType – Allowed values

The following table shows the allowed values for ***MediaType*** (*MediaTypeEnumeration*).

Table 297 – *MediaType* – Allowed values

Value	Description
paperTicket	Travel document is paper ticket.
paperTicketWithCoupons	Travel document is a paper ticket with coupons.
coupon	Travel document is a coupon.
mobileApp	Travel document is a mobile app.
selfPrintPaperTicket	Travel document is self-print paper ticket.
smartCard	Travel document is on a smart card.
card	Travel document is travel Card.
sms	Travel document is an SMS message.
mms	Travel document is a multimedia SMS message.
other	Travel document is some other media.

7.8.3 Sales Offer Package

7.8.3.1 SALES OFFER PACKAGE – Conceptual MODEL

The FARE SALES OFFER PACKAGE MODEL describes the products marketed to the user and available to purchase.

7.8.3.1.1 Sales Offer Packages

The FARE PRODUCTs are associated with TRAVEL DOCUMENTs in order to form packages suitable for selling. A SALES OFFER PACKAGE is defined as a package to be sold as a whole, consisting of one or several FARE PRODUCTs materialised thanks to one or several TRAVEL DOCUMENTs.

The FARE PRODUCTs may be either directly attached to the TRAVEL DOCUMENTs (printing, magnetic storage, etc.), or may be reloadable on TRAVEL DOCUMENTs (such as electronic purses or passes).

In most cases, a SALES OFFER PACKAGE will only consist of one FARE PRODUCT on one TRAVEL DOCUMENT, but more complex combinations are possible. For instance, a USAGE DISCOUNT RIGHT with its own TRAVEL DOCUMENT may be packed with an AMOUNT OF PRICE UNIT on an electronic purse.

Such combinations allow an offer of temporary (e.g. during a promotion week) or permanent packages to be made.

SALES OFFER PACKAGEs are described by SALES OFFER PACKAGE ELEMENTs, each of which associates a specific FARE PRODUCT with a specific TYPE OF TRAVEL DOCUMENT, and specifies the number of TRAVEL DOCUMENTs within the package element.

SALES OFFER PACKAGEs must comprise only SALES OFFER PACKAGE ELEMENTs of which the TYPE OF TRAVEL DOCUMENT is compatible with the CHARGING MOMENT of the corresponding FARE PRODUCT. For instance, if the CHARGING MOMENT of a FARE PRODUCT is 'post-payment on central account', it can only be associated with TRAVEL DOCUMENTs allowing this payment possibility (e.g. electronic card).

A SALES OFFER PACKAGE may sometimes be subject to a parameter limitation. For instance, a SALES OFFER PACKAGE may be restricted to be sold only in a certain STOP AREA. Such a limitation is specified by a GENERIC PARAMETER ASSIGNMENT.

A SALES OFFER PACKAGE may be made up of one or more SALES OFFER PACKAGE ELEMENTs, each specifying a component of the PACKAGE that is purchased as a whole. A GROUP OF SALES OFFER PACKAGEs allows common features to be reused on many specific packages.

The user's actual purchase will be described by a TRAVEL SPECIFICATION (see later below) which indicates which specific features of the SALES OFFER PACKAGE have been selected, for example the SALE PACKAGE might include a *first class return, first class single, second class single*, etc, of which only one of which will be selected in a TRAVEL SPECIFICATION. Thus, a given SALES OFFER PACKAGE may comprise a number of different values for each feature of the fare structure – there is not necessarily a separate SALES OFFER PACKAGE for each combination of features that a user may buy.

A DISTRIBUTION ASSIGNMENT can be used to specify restrictions on the distribution and fulfilment of the package. A SALES OFFER PACKAGE SUBSTITUTION allows a preference to be indicated for choosing alternative packages if a given package is not available.

There may be NOTICEs associated with the SALES OFFER PACKAGE, assigned using a SALES NOTICE ASSIGNMENT.

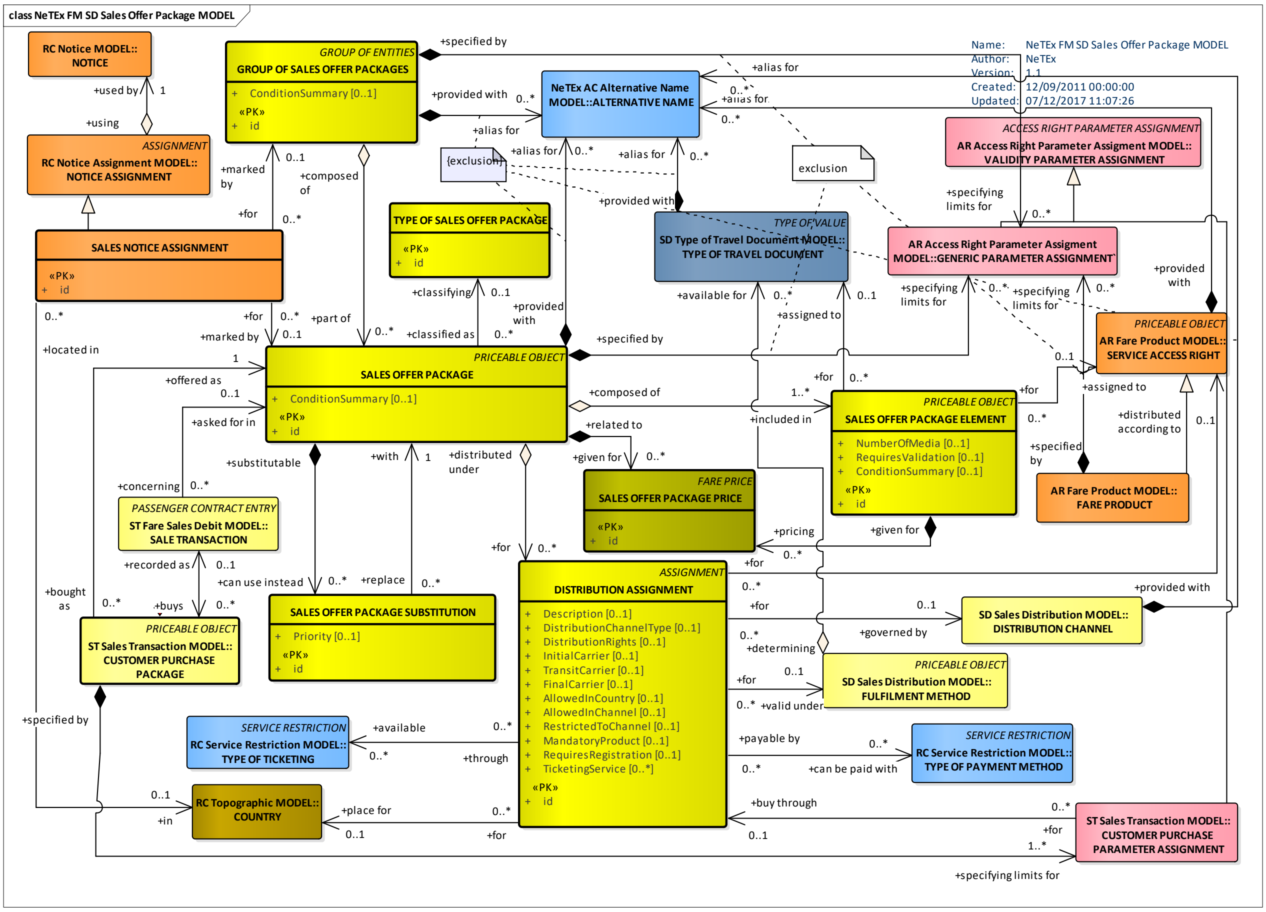


Figure 317 — Sales Offer Package– Conceptual MODEL (UML)

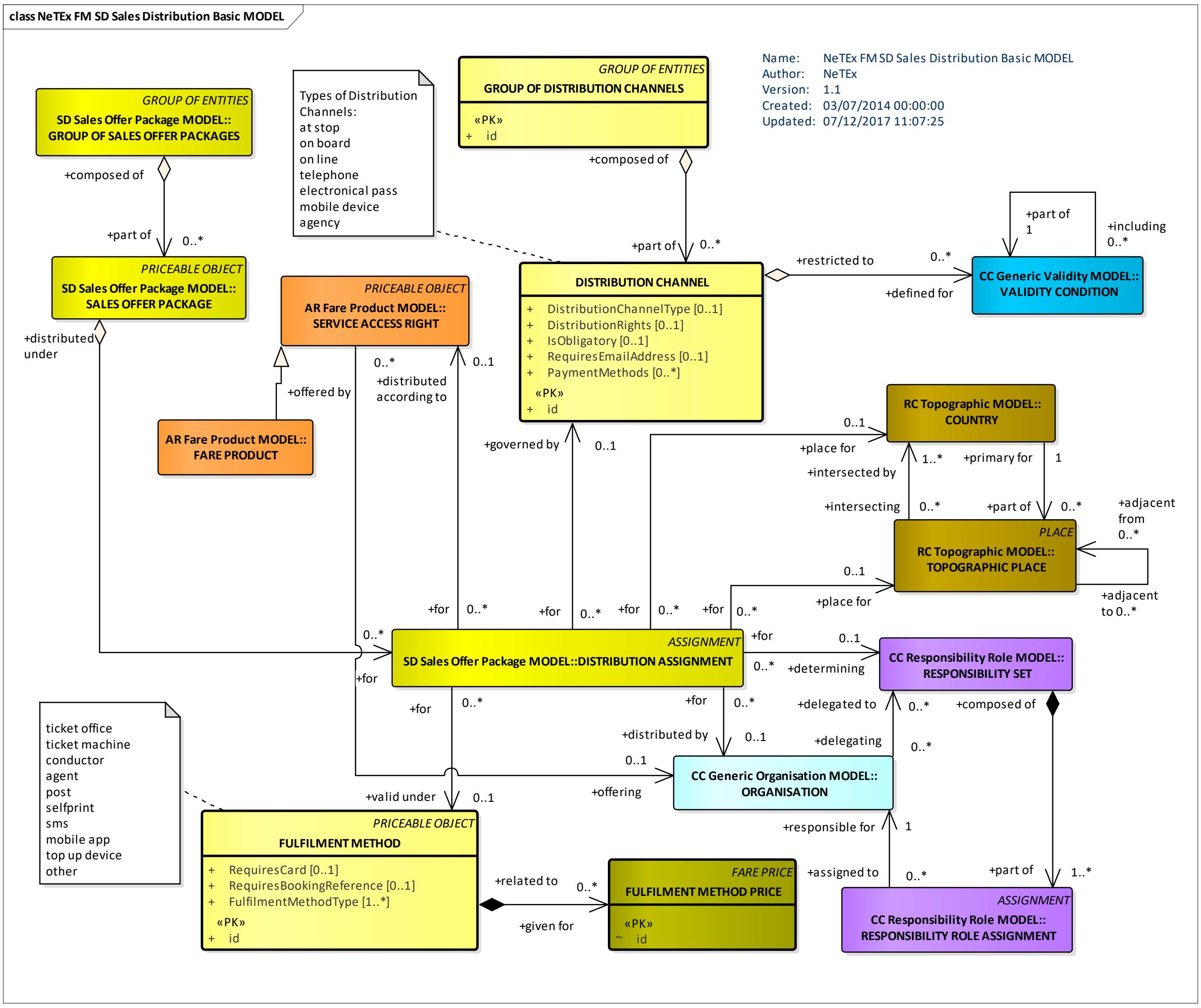


Figure 318 — Sales Distribution – Conceptual MODEL (UML)

7.8.3.2 Fare Sales Offer Package – Physical model

The following figure shows the physical model for SALES OFFER PACKAGEs.

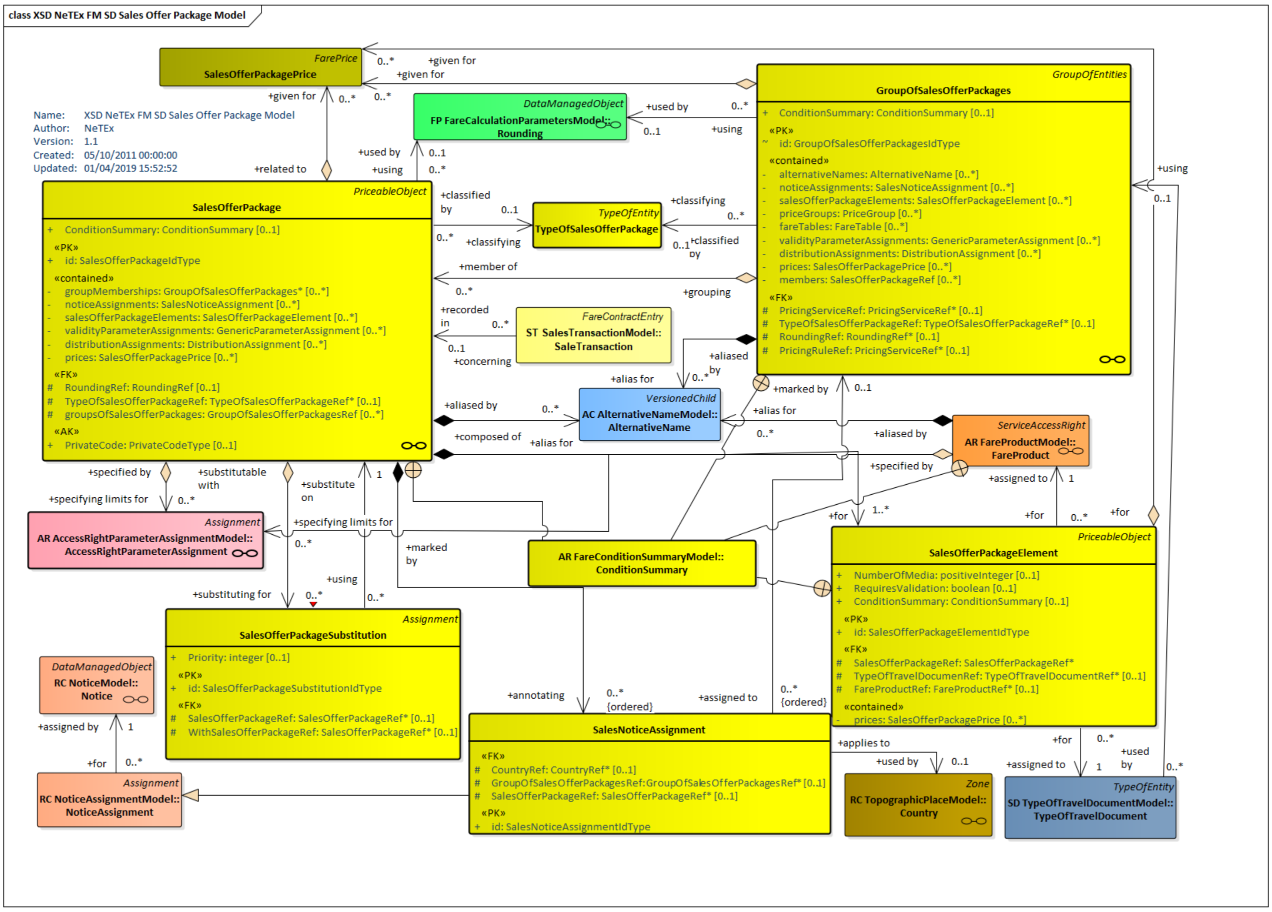


Figure 319 — Sales Offer Package – Physical Model (UML)

7.8.3.2.1 **Group of Fare Sales Offer Package – Physical model**

The following figure shows the physical model for a GROUP of SALES OFFER PACKAGEs.

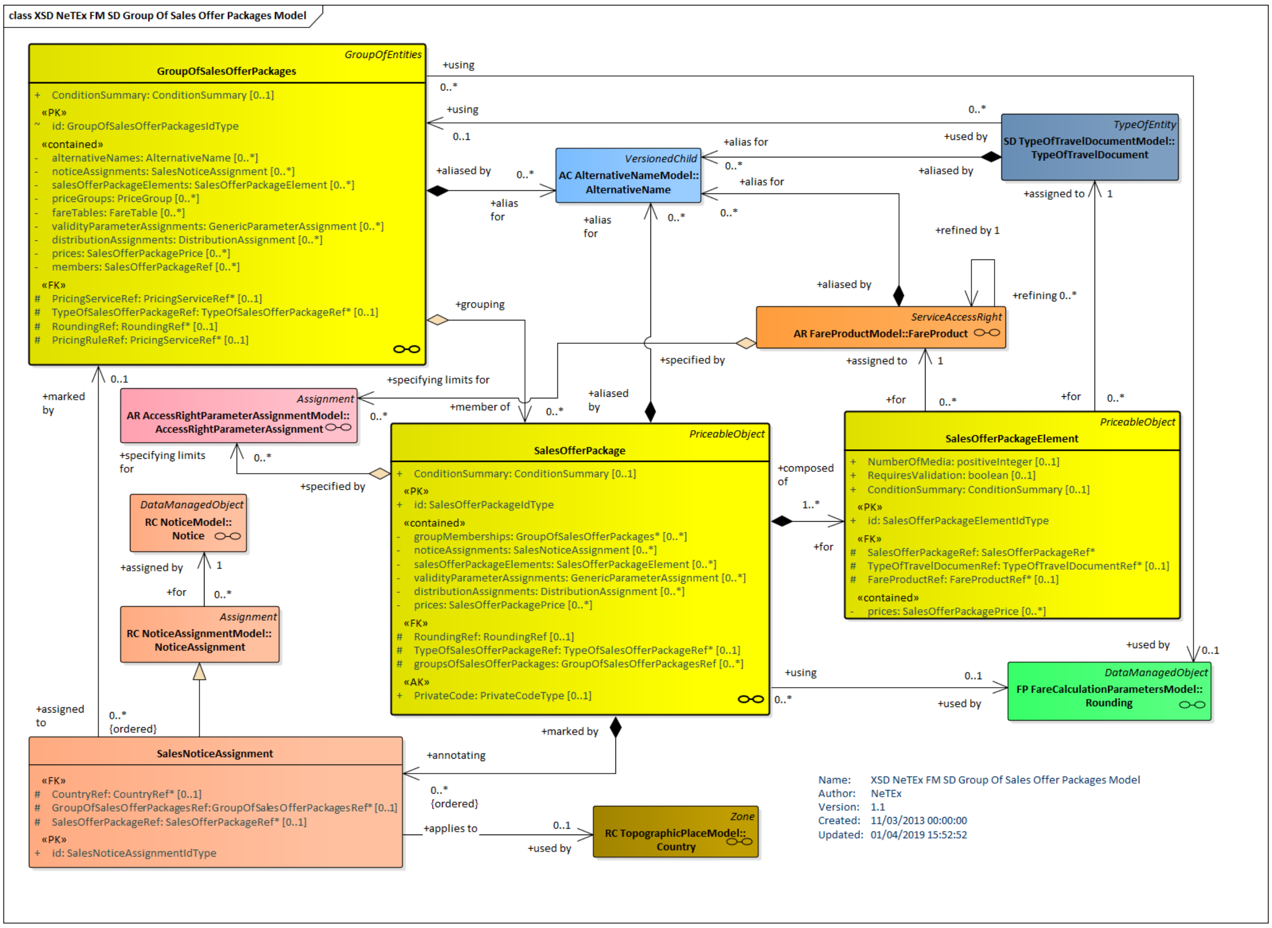


Figure 320 — Group Of Sales Offer Packages – Physical Model (UML)

7.8.3.2.1 Sales Offer Package Entitlement – Physical model

A SALES OFFER PACKAGE may give an entitlement or require an entitlement to another SALES OFFER PACKAGE. This is distinct from the dependency between FARE PRODUCTs, so separate SALES OFFER ENTITLEMENT REQUIRED and SALES OFFER ENTITLEMENT GIVEN elements are used.

The following figure shows the physical model for a SALES OFFER PACKAGE ENTITLEMENTs.

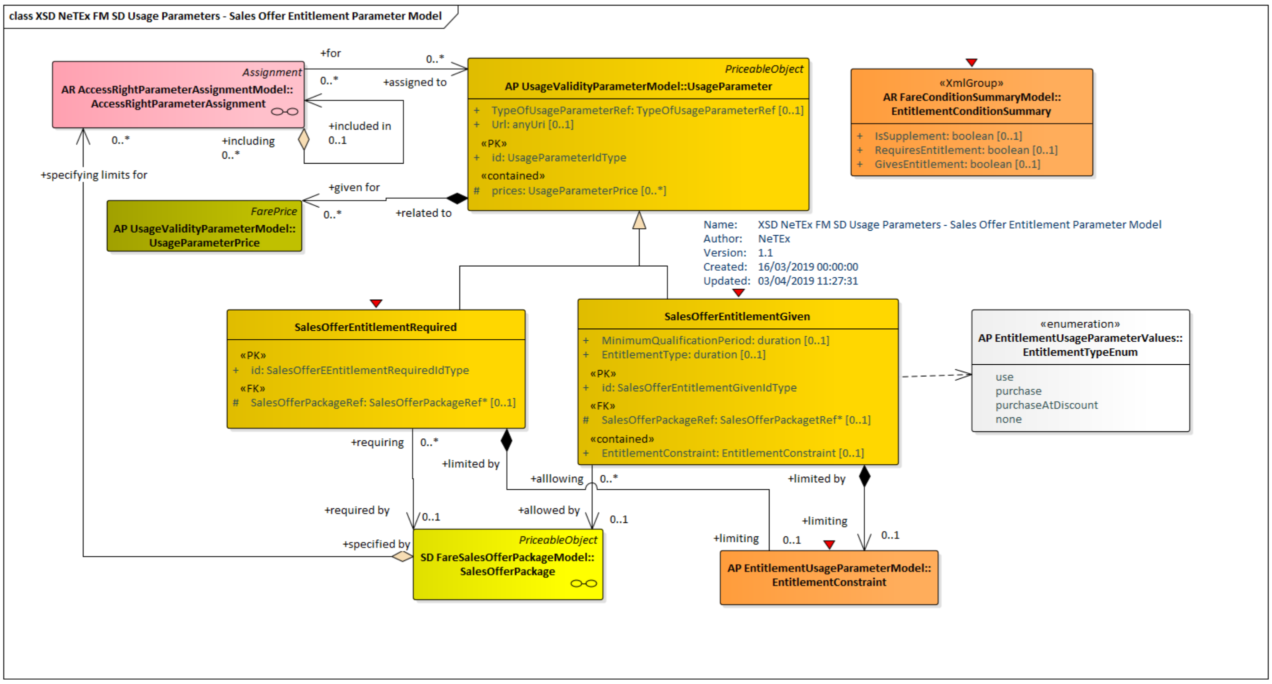


Figure 321 - Sales Offer Package Entitlement – Physical Model (UML)

7.8.3.2.2 Distribution Assignment – Physical model

The following figure shows the physical model for the DISTRIBUTION ASSIGNMENT of SALES OFFER PACKAGEs.

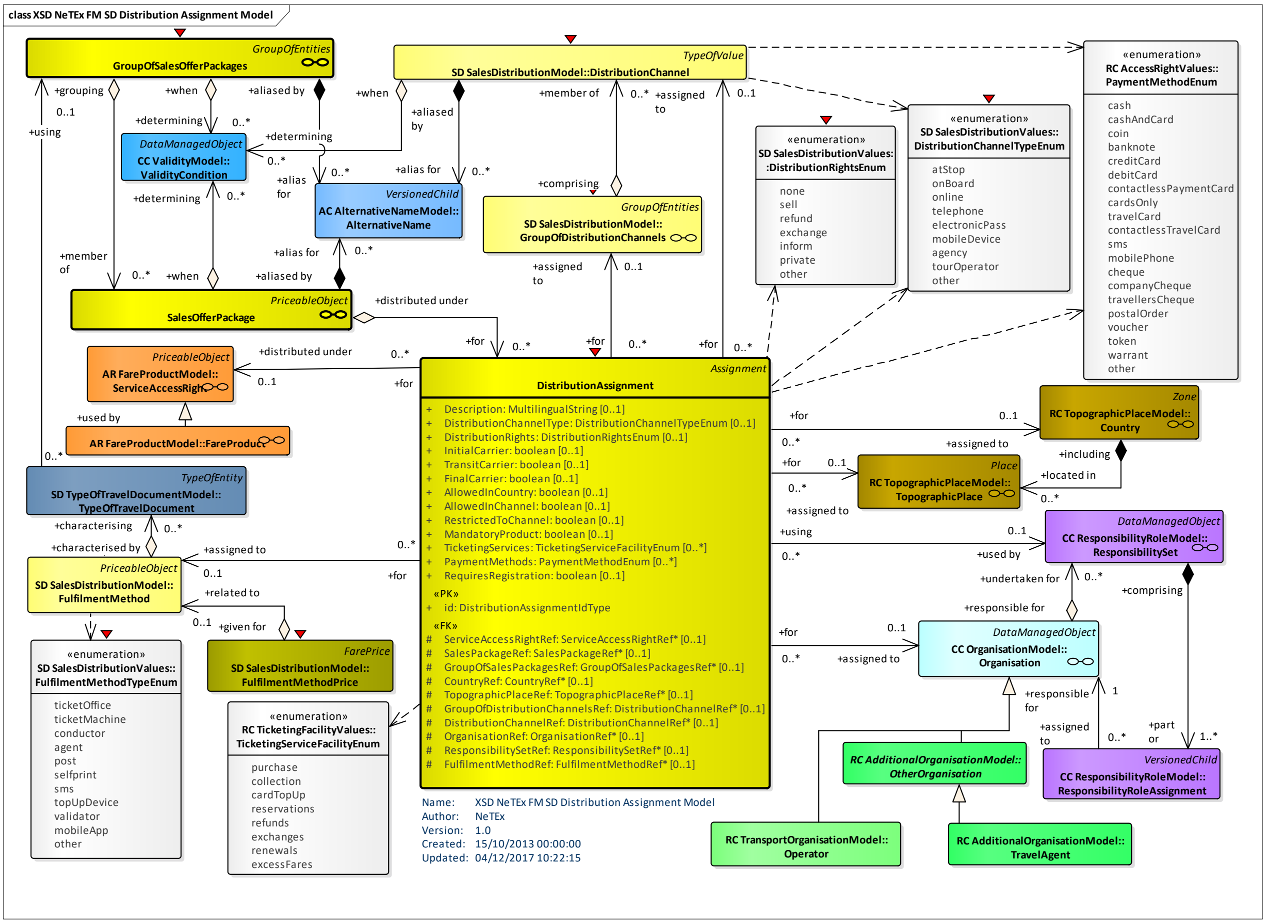


Figure 322 — Distribution Assignment – Physical Model

7.8.3.3 Fare Sales Offer Package – Attributes and XSD

7.8.3.3.1 SalesOfferPackage – Model Element

A package to be sold as a whole, consisting of one or several FARE PRODUCTs materialised thanks to one or several TRAVEL DOCUMENTs. The FARE PRODUCTs may be either directly attached to the TRAVEL DOCUMENTs, or may be reloadable on the TRAVEL DOCUMENTs.

Table 298 – *SalesOfferPackage* – Model Element

Classifi-cation	Name	Type	Card in-ality	Description
::>	::>	*PriceableObject*	::>	SALES OFFER PACKAGE inherits from PRICEABLE OBJECT.
«PK»	***id***	*SalesOfferPackageIdType*	1:1	Identifier of a SALES OFFER PACKAGE.
«AK»	***PrivateCode***	*PrivateCodeType*	0:1	Alternative identifier of an entity. can be used to associate with legacy systems.
XGRP	***SalesOffer-Package-CommonGroup***	***xmlGroup***	0:1	Common properties of SALES OFFER PACKAGE and GROUP OF SALES OFFER PACKAGEs.

«cntd»	***groupsOfSales-OfferPackages***	*GroupOfSalesOffer-PackagesRef*	0:*	GROUPs OF SALES OFFER PACKAGEs with which this SALES OFFER PACKAGE shares common properties.
«cntd»	***salesOffer-Package-Substitutions***	*SalesOfferPackageSubstitution*	0:*	Allowed SALES OFFER PACKAGE SUBSTITUTIONs for the SALES OFFER PACKAGE.

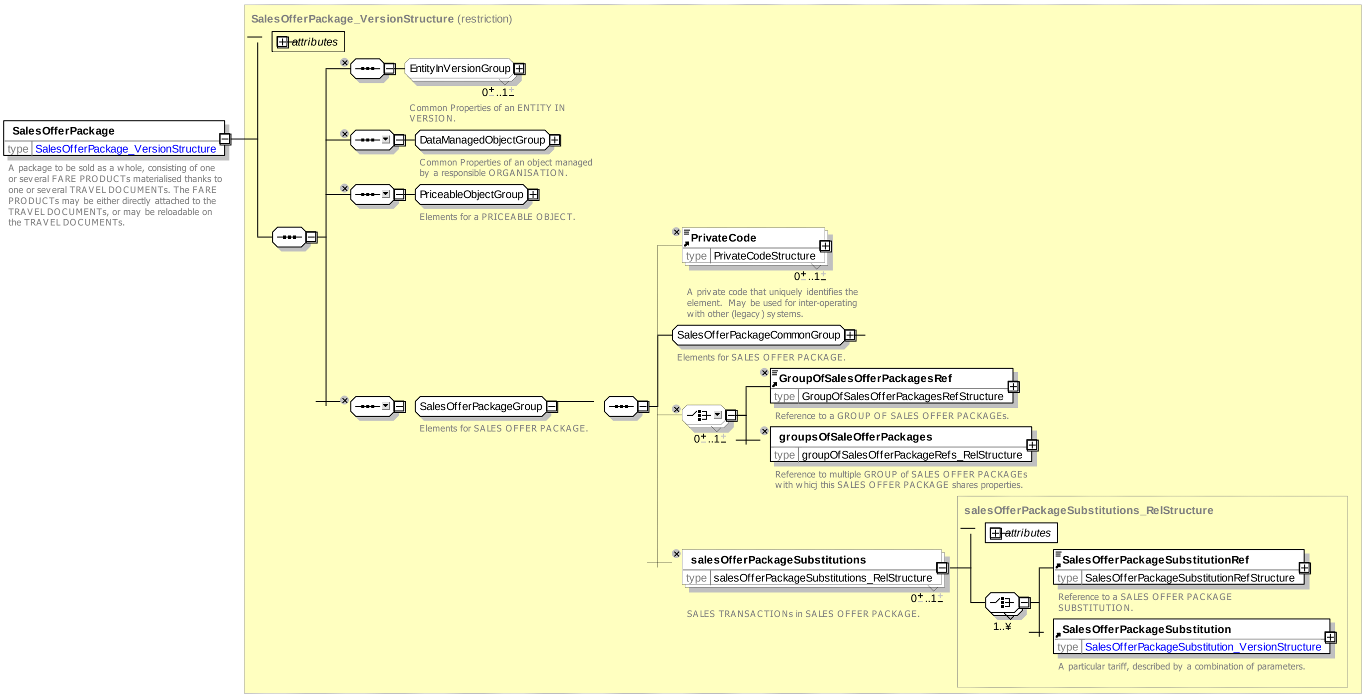


Figure 323 — *SalesOfferPackage* — XSD

7.8.3.3.1.1 SalesOfferPackageCommonGroup – Group

The ***SalesOfferPackageCommonGroup*** specifies the common properties of SALES OFFER PACKAGE and GROUPs OF SALES OFFER PACKAGEs.

Table 299 – *SalesOfferPackageCommonGroup* – Model Element– Element

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***TypeOfSales-OfferPackageRef***	*TypeOfSalesOffer-PackageRef*	0:1	Type of SALES OFFER PACKAGE.
«cntd»	***Condition-Summary***	<u>*ConditionSummary*</u>	0:1	Summary description of conditions of a SALES OFFER PACKAGE that can be used to provide passenger information.
«cntd»	***validity-Parameter-Assignments***	*GenericAccess-RightParameter-Assignment*	0:*	GENERIC PARAMETER ASSIGNMENTs (i.e. ACCESS RIGHT PARAMETER ASSIGNMENTs) associated with the SALES OFFER PACKAGE.

«cntd»	***distribution-Assignments***	*DistributionAssignment*	0:*	DISTRIBUTION ASSIGNMENTs for the SALES OFFER PACKAGE.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Rounding to use on calculation
«cntd»	***prices***	*SalesOfferPackagePrice*	0:*	SALES OFFER PACKAGE PRICEs associated with the FARE
«cntd»	***salesOffer-Package-Elements***	*SalesOfferPackage-Element*	0:*	SALES OFFER PACKAGE ELEMENTs associated with the SALES OFFER PACKAGE.

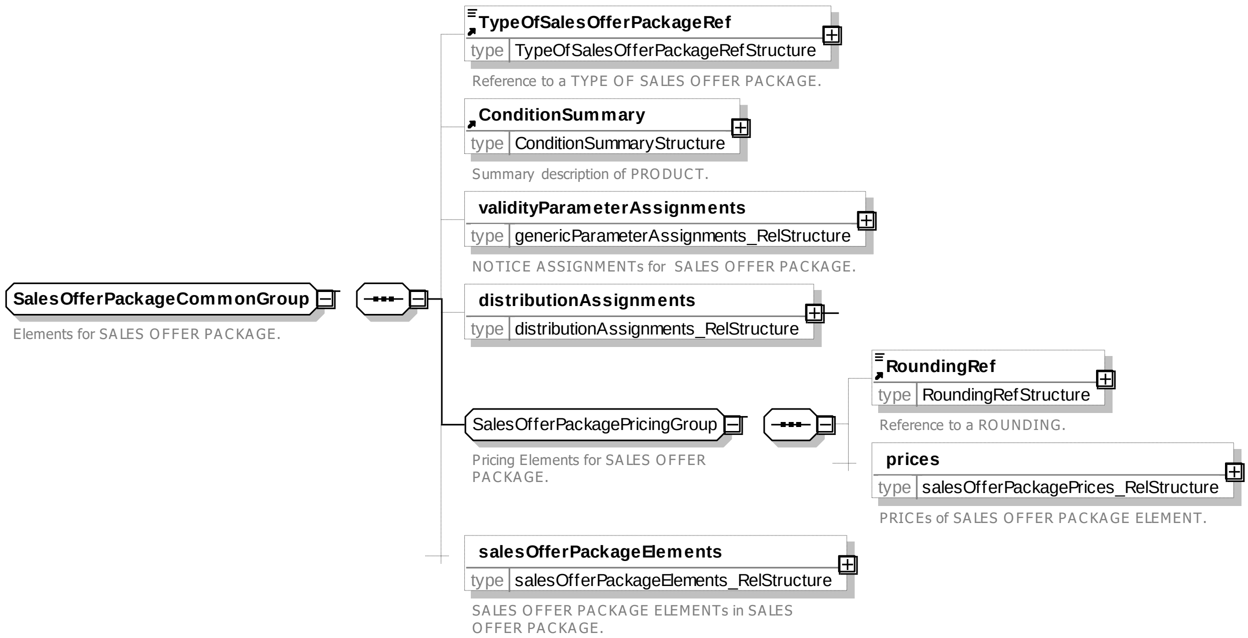


Figure 324 — *SalesOfferPackageCommonGroup* — XSD

7.8.3.3.2 SalesOfferPackageElement – Model Element

The assignment of a FARE PRODUCT to a TYPE OF TRAVEL DOCUMENT in order to define a SALES OFFER PACKAGE, realised as a fixed assignment (printing, magnetic storage etc.) or by the possibility for the FARE PRODUCT to be reloaded on the TYPE OF TRAVEL DOCUMENT.

Table 300 – *SalesOfferPackageElement* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	SALES OFFER PACKAGE ELEMENT inherits from PRICEABLE OBJECT.
«PK»	***id***	*SalesOfferPackage-ElementIdType*	1:1	Identifier of SALES OFFER PACKAGE ELEMENT.
	Requires-Validation	*xsd:boolean*	0:1	Whether element requires validation before it can be used.

«cntd»	***Condition-Summary***	*ConditionSummary*	0:1	Summary description of SALES OFFER PACKAGE properties.
«FK»	***SalesOffer-PackageRef***	*SalesOfferPackageRef*	1:1	Reference to a SALES OFFER PACKAGE of which this is part. If not given by containing context must be specified.
«FK»	***TypeOfTravel-DocumenRef***	*TypeOfTravel-DocumentRef*	0:1	Reference to a TYPE OF TRAVEL DOCUMENT.
«FK»	***FareProductRef***	*FareProductRef+*	0:1	FARE PRODUCT associated with this SALES OFFER PACKAGE.
«cntd»	***validity-Parameter-Assignments***	*GenericParameter-Assignment*	0:*	GENERIC PARAMETER ASSIGNMENTs associated with the SALES OFFER PACKAGE ELEMENT.
«cntd»	***prices***	*SalesOfferPackagePrice*	0:*	SALES OFFER PACKAGE PRICEs associated with the SALES OFFER PACKAGE ELEMENT.

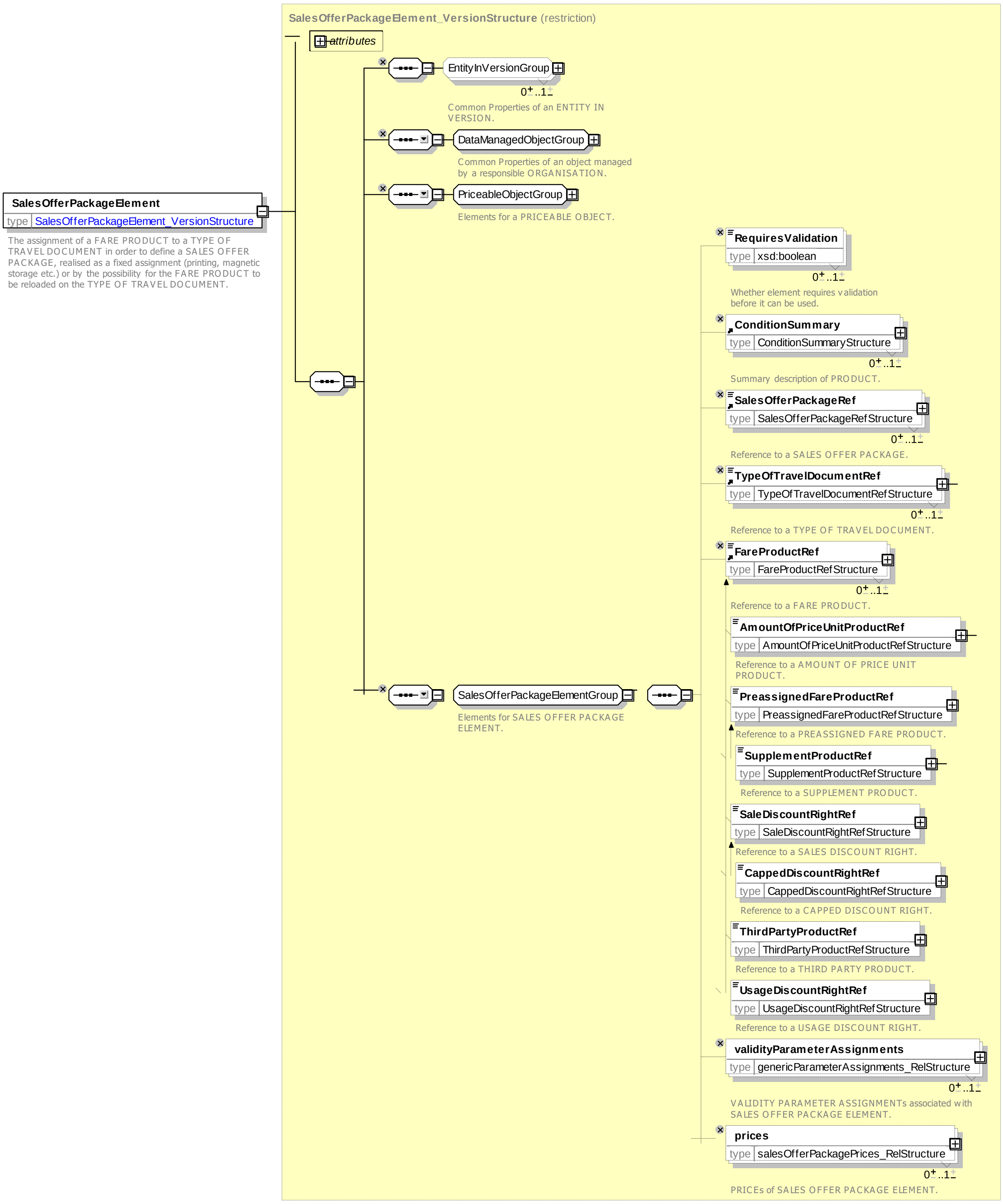


Figure 325 — *SalesOfferPackageElement* — XSD

7.8.3.3.3 SalesOfferPackagePrice – Model Element

A set of all possible price features of a SALES OFFER PACKAGE: default total price etc.

Table 301 – *SalesOfferPackagePrice* – Element

Classifi-cation	Name		Type	Cardinality	Description
::>	::>		*FarePrice*	::>	SALES OFFER PACKAGE PRICE inherits from FARE PRICE.
«PK»	***id***		*SalesOfferPackage-PriceIdType*	1:1	Identifier of SALES OFFER PACKAGE PRICE.
			CHOICE	1:1	Reference to element for which this is the price.
«FK»	***a***	***SalesOffer-PackageRef***	*SalesOfferPackageRfg*	1:1	Reference to a SALES OFFER PACKAGE. If not given by containing context must be specified.
«FK»	***b***	***SalesOffer-Package-ElementRef***	*SalesOfferPackage-ElementtRefe*	1:1	Reference to a SALES OFFER PACKAGE ELEMENT. If not given by containing context must be specified.

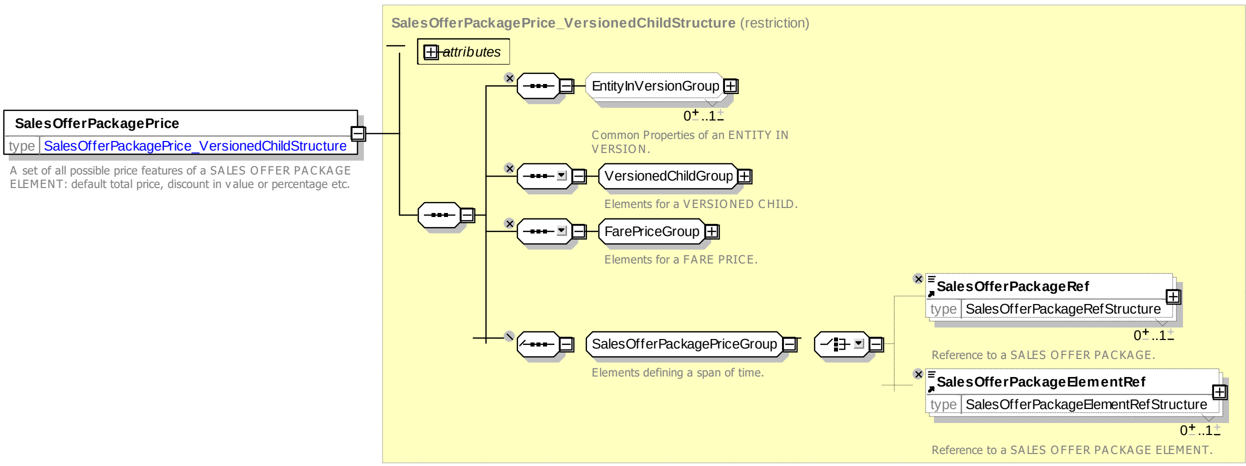


Figure 326 — *SalesOfferPackagePrice* — XSD

7.8.3.3.1 Usage Parameter: Sales Offer Package Entitlements – Attributes and XSD

7.8.3.3.1.1 SalesOfferPackageEntitlementRequired – Model Element

Receiving of entitlement from another SALES OFFER PACKAGE.

Table 302 – *SalesOfferPackageEntitlementRequired* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*UsageParameter*	::>	SALES OFFER PACKAGE ENTITLEMENT REQUIRED inherits from USAGE PARAMETER.
«PK»	***id***	*EntitlementRequiredIdType*	1:1	Identifier of SALES OFFER PACKAGE ENTITLEMENT REQUIRED.

«FK»	***ServiceAccess-RightRef***	*ServiceAccessRightRef*	0:1	Entitlement comes from the referenced SALES OFFER PACKAGE.
	Minimum-Qualification-Period	*xsd:duration*	0:1	Minimum period that required product must be held in order to be eligible.
«cntd»	***Entitlement-Constraint***	<u>*EntitlementConstraint*</u>	0:1	Constraints on related product or offer. +v1.1

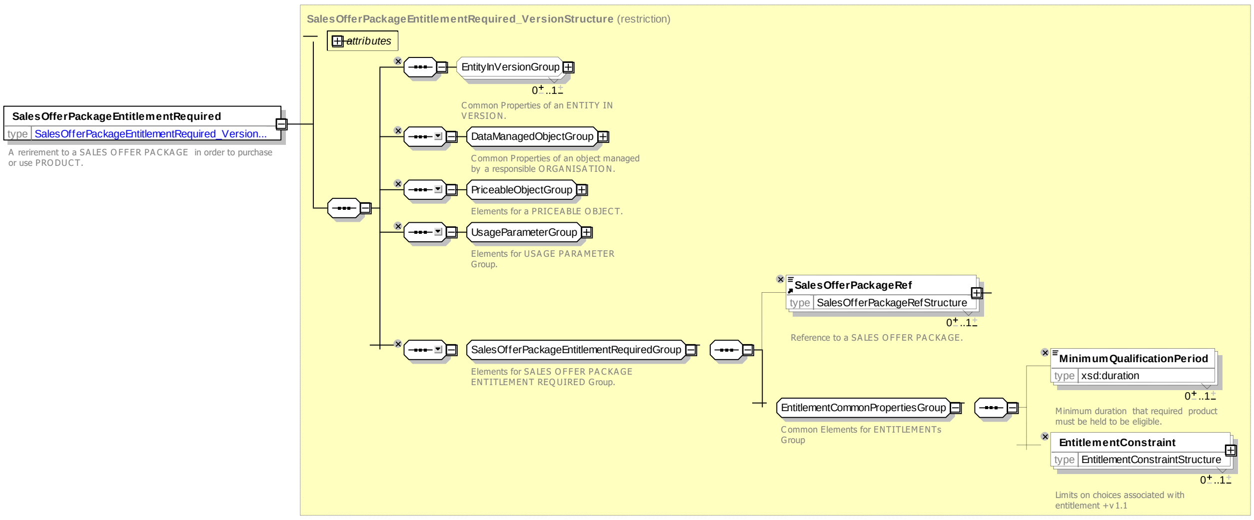


Figure 327 — *SalesOfferPackageEntitlementRequired* — XSD

7.8.3.3.1.2 SalesOfferPackageEntitlementGiven – Model Element

Granting of entitlement to another SALES OFFER PACKAGE.

Table 303 – *SalesOfferPackageEntitlementGiven* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	<u>*UsageParameter*</u>	::>	SALES OFFER PACKAGE ENTITLEMENT GIVEN inherits from USAGE PARAMETER.
«PK»	***id***	*EntitlementGivenIdType*	1:1	Identifier of SALES OFFER PACKAGE ENTITLEMENT GIVEN.
«FK»	***ServiceAccess-RightRef***	*ServiceAccessRightRef*	0:1	Entitlement comes from the referenced SALES OFFER PACKAGE.
	Minimum-Qualification-Period	*xsd:duration*	0:1	Minimum period that product must be held for entitlement to be granted.
«cntd»	***Entitlement-Constraint***	<u>*EntitlementConstraint*</u>	0:1	Constraints on related product or offer. See ENTITLEMENT GIVEN earlier. +v1.1
«enum»	***EntitlementType***	*EntitlementTypeEnum*	0:1	Type of entitlement. See allowed values for ENTITLEMENT GIVEN.

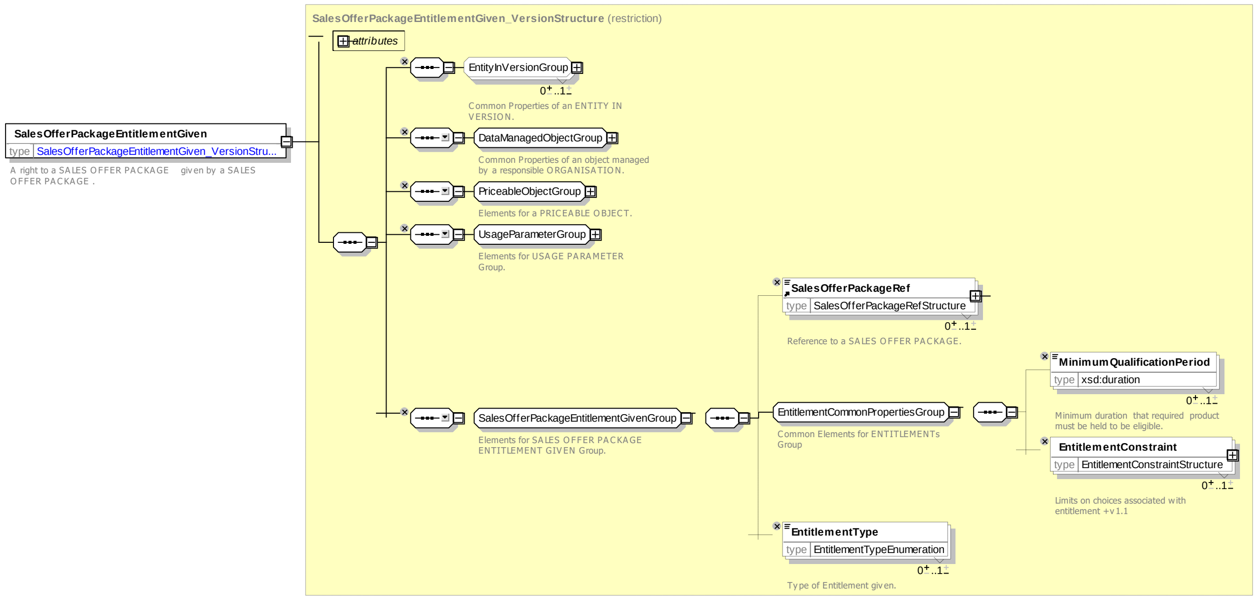


Figure 328 — *SalesOfferPackageEntitlementGiven* — XSD

7.8.3.3.2 GroupOfSalesOfferPackages – Model Element

A grouping of SALES OFFER PACKAGEs.

Table 304 – *GroupOfSalesOfferPackages* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*GroupOfEntities*</u>	::>	GROUP of SALES OFFER PACKAGEs inherits from GROUP OF ENTITIES. See NeTEx Part1.
«PK»	***id***	*GroupOfSalesOffer-PackagesIdType*	1:1	Identifier of GROUP of SALES OFFER PACKAGEs.
«cntd»	***alternativeNames***	<u>*AlternativeName*</u>	0:*	ALTERNATIVE NAMEs for GROUP of SALES OFFER PACKAGEs.
«cntd»	***notice-Assignments***	<u>*NoticeAssignment*</u>	0:*	NOTICE ASSIGNMENTs for GROUP of SALES OFFER PACKAGEs.
«FK»	***PricingService-Ref***	*PricingServiceRef*	0:1	PRICING SERVICE to use to fetch prices dynamically.
«FK»	***PricingRuleRef***	*PricingRuleRef+*	0:1	Default PRICING RULE to use to derive prices from this element.
«cntd»	***priceGroups***	<u>*PriceGroup*</u>	0:*	PRICE GROUPs associated with this element.
«cntd»	***fareTables***	<u>*FareTable*</u>	0:*	FARE TABLEs associated with this element.
XGRP	***SalesOffer-Package-CommonGroup***	<u>***xmlGroup***</u>	0:1	Common properties of SALES OFFER PACKAGE and GROUP OF SALES OFFER PACKAGES.

«cntd»	***members***	*SalesOfferPackageRef*	0:*	References to members of GROUP of SALES OFFER PACKAGEs. See above.

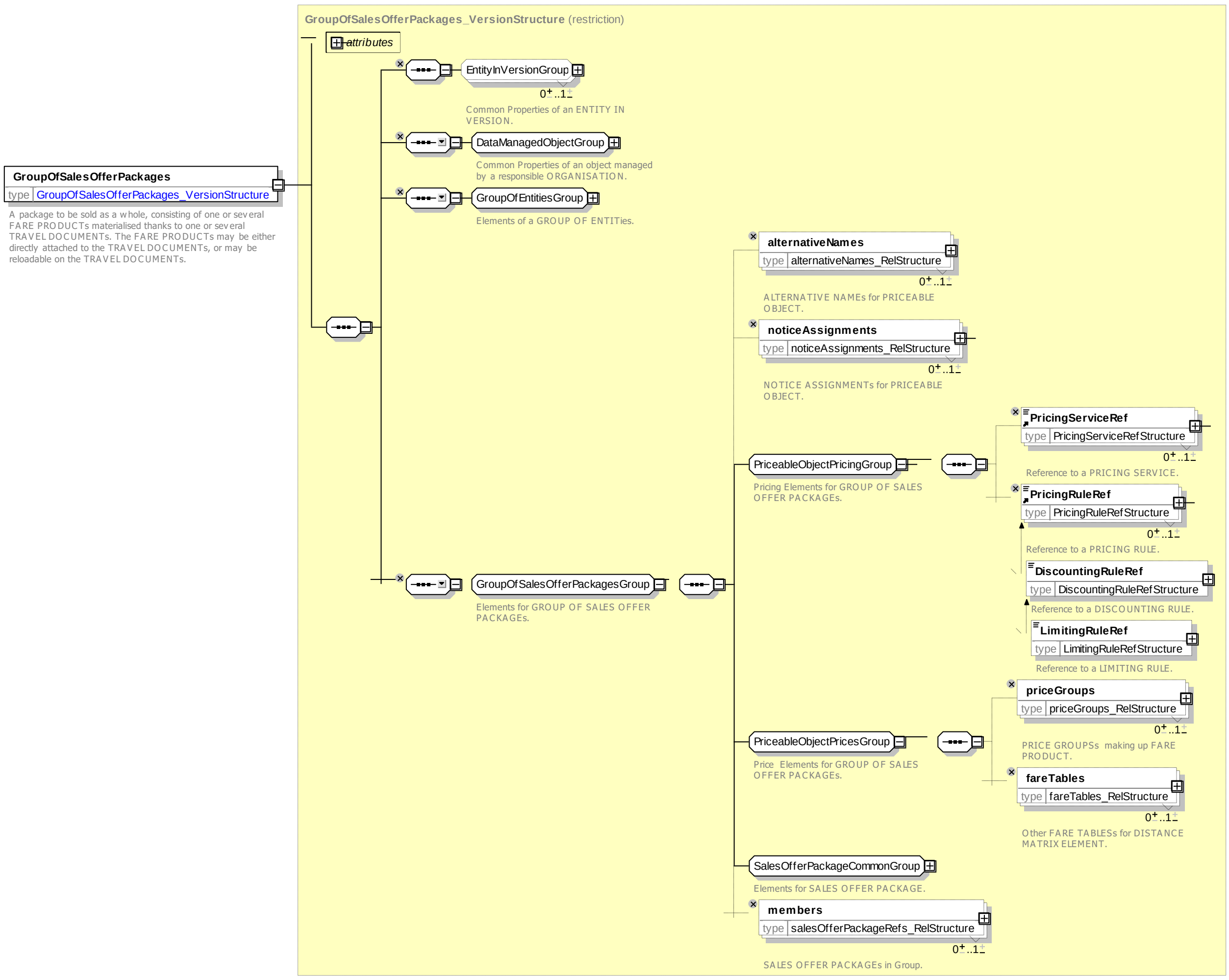


Figure 329 — *GroupOfSalesOfferPackages* — XSD

7.8.3.3.3 DistributionAssignment – Model Element

An assignment of the COUNTRY and/or DISTRIBUTION CHANNEL through which a product may or may not be distributed.

Table 305 – *DistributionAssignment* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*Assignment*	::>	DISTRIBUTION ASSIGNMENT inherits from ASSIGNMENT.
«PK»	***id***	*DistributionAssignmentIdType*	1:1	Identifier of a DISTRIBUTION ASSIGNMENT.
«FK»	***ServiceAccess-RightRef***	*ServiceAccessRightRef*	0:1	SERVICE ACCESS RIGHT (FARE PRODUCT) for which this specifies the DISTRIBUTION ASSIGNMENT.

«FK»	***SalesOffer-PackageRef***	*SalesOfferPackageRef*	0:1	SALES OFFER PACKAGE for which this specifies the DISTRIBUTION ASSIGNMENT.
«FK»	***GroupOfSales-OfferPackages-Ref***	*GroupOfSalesOffer-PackagesRef*	0:1	GROUP OF SALES OFFER PACKAGEs for which this specifies the DISTRIBUTION ASSIGNMENT.
	Distribution-Rights	***xmlGroup***	0:1	Distribution rights associated with this DISTRIBUTION ASSIGNMENT. See below.
XGRP	***Distribution-ThroughGroup***	***xmlGroup***	0:1	Elements governing CHANNELS by which distribution may be made: See ***DistributionThroughGroup.***
XGRP	***Distribution-ByGroup***	***xmlGroup***	0:1	Elements for who may distribute. See ***DistributionByGroup.***
XGRP	***Distribution-DetailsGroup***	***xmlGroup***	0:1	Elements for details about the distribution. See ***DistributionDetailsGroup.***
	notice-Assignments	*NoticeAssignment*	0:*	NOTICE ASSIGNMENTs associated with this DISTRIBUTION ASSIGNMENT.

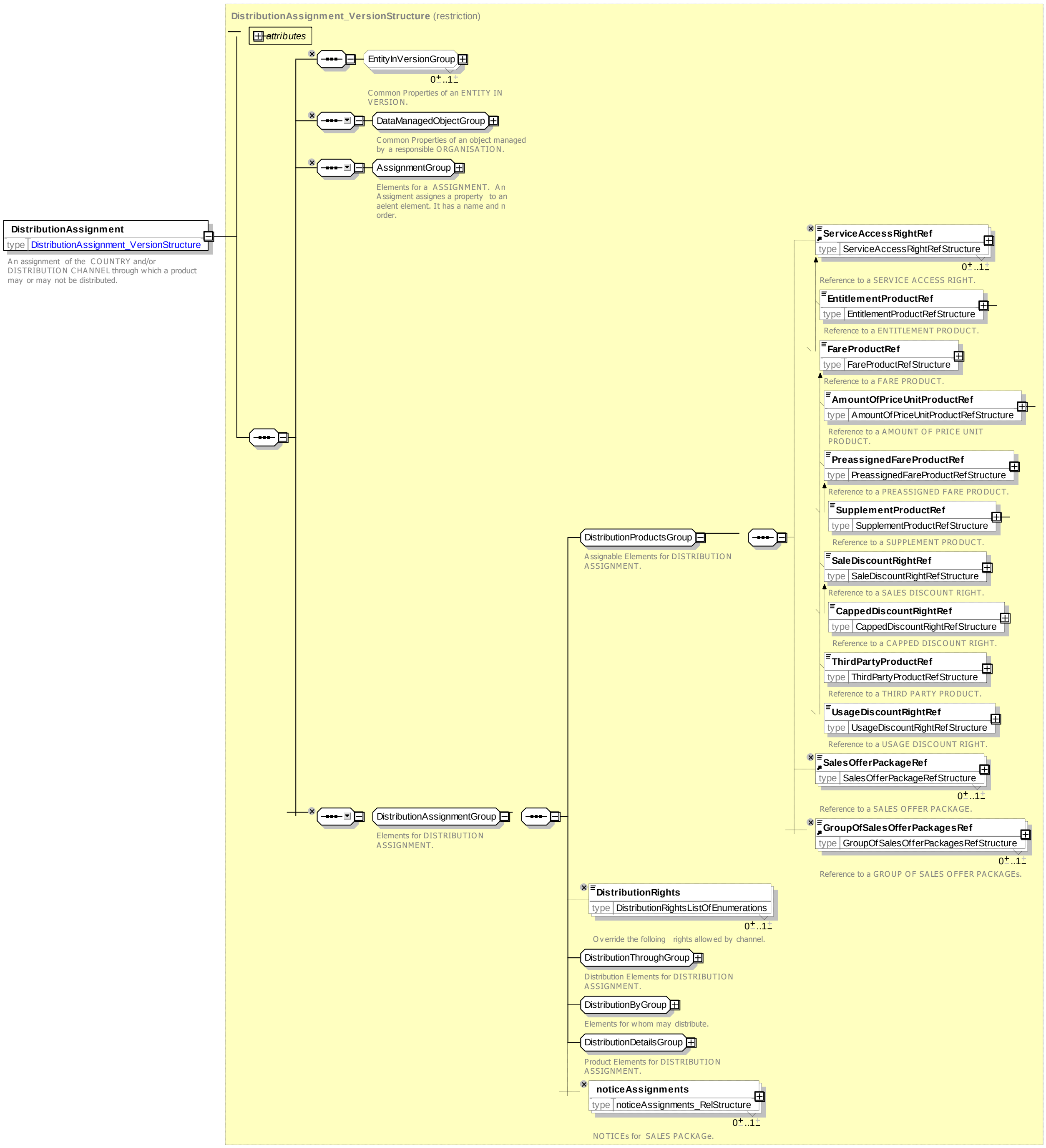


Figure 330 — ***DistributionAssignment* — XSD**

7.8.3.3.3.1 DistributionThroughGroup– Group

The ***DistributionThroughGroup*** group specifies the elements relating to the channels through which distribution of products can be made.

Table 306 – DistributionThroughGroup– Model Element– Element

Classifi-cation	**Name**	**Type**	**Cardinality**	**Description**
		CHOICE		Country in which distribution may take place.

	a	***AllCountries Ref***	*AllCountriesRef*	0:1	Distribution may be made in all countries.
«FK»	***b***	***CountryRef***	*CountryRef*	0:1	COUNTRY in which distribution can be made as described by this DISTRIBUTION ASSIGNMENT.
	AllowedIn-Country		*xsd:boolean*	0:1	Whether distribution is allowed or forbidden in the specified country.
«FK»	***Topographic-PlaceRef***		*TopographicPlaceRef*	0:1	TOPOGRAPHIC PLACE for which this specifies the DISTRIBUTION ASSIGNMENT.
			CHOICE		Channel by which distribution can be made.
	a	***All-Distribution-ChannelsRef***	*AllDistribution-ChannelsRef*	0:1	Distribution may be made through all channels.
«FK»	***b***	***Distribution-ChannelRef***	*DistributionChannelRef*	0:1	DISTRIBUTION CHANNEL for which this specifies the DISTRIBUTION ASSIGNMENT.
«FK»	***c***	***GroupOf-Distribution-ChannelsRef***	*GroupOfDistribution-ChannelsRef*	0:1	GROUP OF DISTRIBUTION CHANNELs for which this specifies the DISTRIBUTION ASSIGNMENT.
«enum»	***Distribution-ChannelType***		*DistributionChannel-TypeEnum*	0:1	Type of DISTRIBUTION CHANNEL. See earlier.
	AllowedIn-Channel		*xsd:boolean*	0:1	Whether distribution is allowed or forbidden by the specified DISTRIBUTION CHANNEL.
	RestrictedTo-Channel		*xsd:boolean*	0:1	Whether distribution is restricted to only the specified DISTRIBUTION CHANNELs.
	Mandatory-Product		*xsd:boolean*	0:1	Whether product is mandatory.

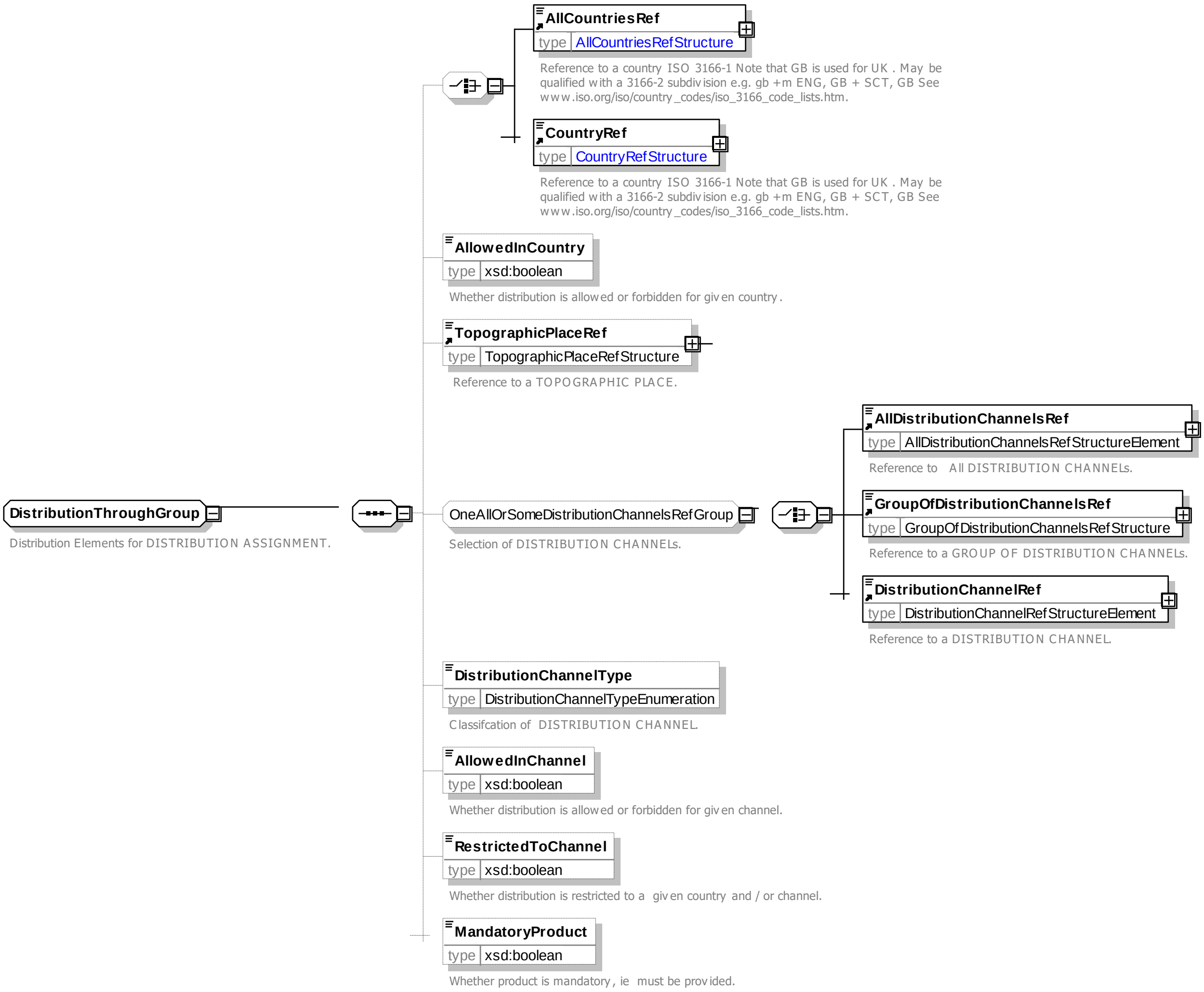


Figure 331 — *DistributionThroughGroup* — XSD

7.8.3.3.3.2 DistributionByGroup– Group

The ***DistributionByGroup*** group specifies the elements relating to the ORGANISATIONs who may undertake distribution of products and the responsibilities they may undertake. See NeTEx Part1 for further details on roles and RESPONSIBILITies.

Table 307 – *DistributionByGroup*– Model Element– Element

Classifi-cation	Name	Type	Cardinality	Description
	InitialCarrier	*xsd:boolean*	0:1	Distribution by carrier of first leg of trip.
	TransitCarrier	*xsd:boolean*	0:1	Distribution by carrier of middle of trip.
	FinalCarrier	*xsd:boolean*	0:1	Distribution by carrier of final leg of trip.
		Choice		Organisation who may distribute.

«FK»	***a***	***All-Organisations Ref***	*AllOrganisationsRef*	0:1	All ORGANISATIONs may distribute.
«FK»	***b***	***Organisation Ref***	*(OrganisationRef)*	0:1	ORGANISATION for which this specifies the DISTRIBUTION ASSIGNMENT.
«FK»	***Responsibility-SetRef***		*ResponsibilitySetRef*	0:1	RESPONSIBILITY SET describing the DISTRIBUTION ASSIGNMENT.

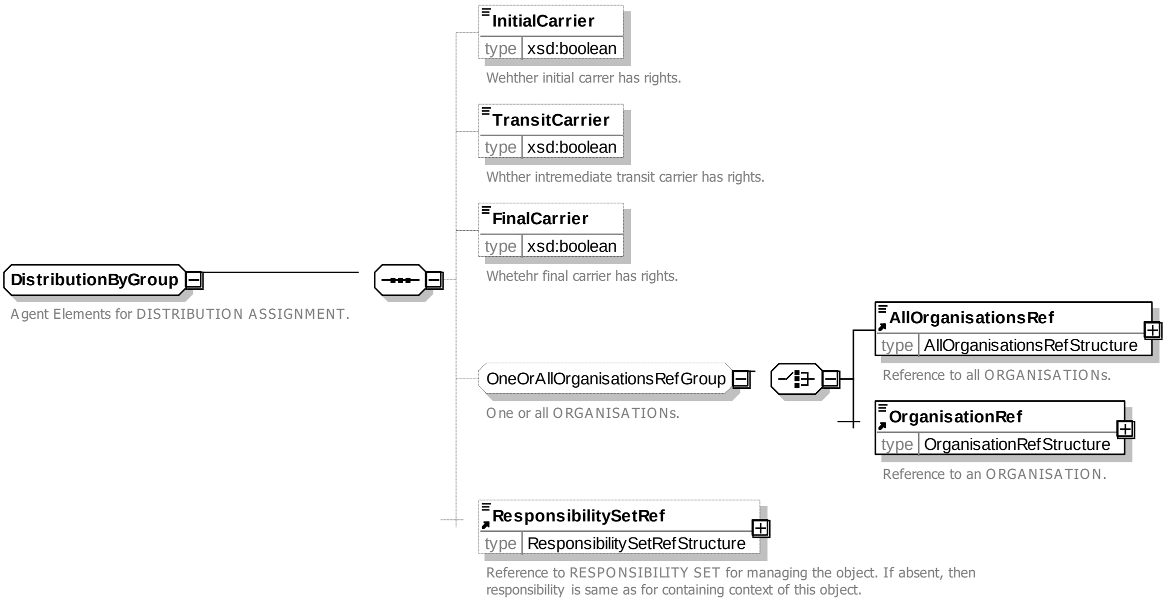


Figure 332 — *DistributionByGroup* — XSD

7.8.3.3.3.3 DistributionDetailsGroup– Group

The ***DistributionDetailsGroup*** group specifies the properties that can be set by a DISTRIBUTION ASSIGNMENT.

Table 308 – *DistributionDetailsGroup*– Model Element– Element

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***Ticketing-ServiceFacility***	*TicketingService-FacilityEnum*	0:*	List of TICKETING SERVICE FACILITies, e.g. purchase, collection. top up. See Part1
«enum»	***PaymentMethods***	*PaymentMethodEnum*	0:*	Payment method supported on this distribution. See Part1
	Requires-Registration	*xsd:boolean*	0:1	Whether distribution requires the customer to register a personal identity either online or otherwise.
«FK»	***Fulfilment-MethodRef***	*FulfilmentMethodRef*	0:1	FULFILMENT METHOD to be used with this DISTRIBUTION CONDITION.

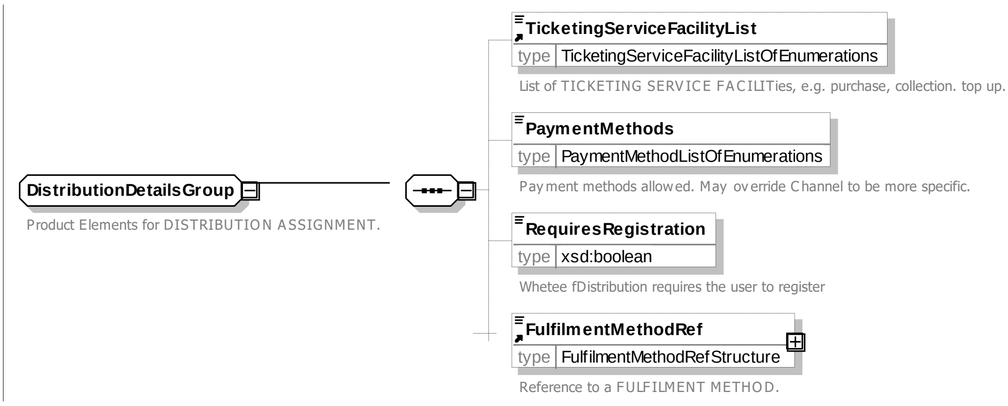


Figure 333 — *DistributionDetailsGroup*— XSD

7.8.3.3.4 SalesNoticeAssignment –

The assignment of a NOTICE to a SALES OFFER PACKAGE or a GROUP OF SALES OFFER PACKAGEs.

Table 309 – *SalesNoticeAssignment* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*NoticeAssignment*	::>	SALES NOTICE ASSIGNMENT inherits from NOTICE ASSIGNMENT
«FK»	***CountryRef***	*CountryRef*	0:1	Reference to a COUNTRY to which NOTICE applies.
«PK»	***id***	*SalesNotice-AssignmentIdType*	1:1	Identifier of a SALES NOTICE ASSIGNMENT.
«FK»	***GroupOfSales-OfferPackages-Ref***	*GroupOfSalesOffer-PackagesRef*	0:1	Reference to a GROUP OF SALES OFFER PACKAGEs to which assignment is made.
«FK»	***SalesOffer-PackageRef***	*SalesOfferPackageRef*	0:1	Reference to a SALES OFFER PACKAGE of which this is part.

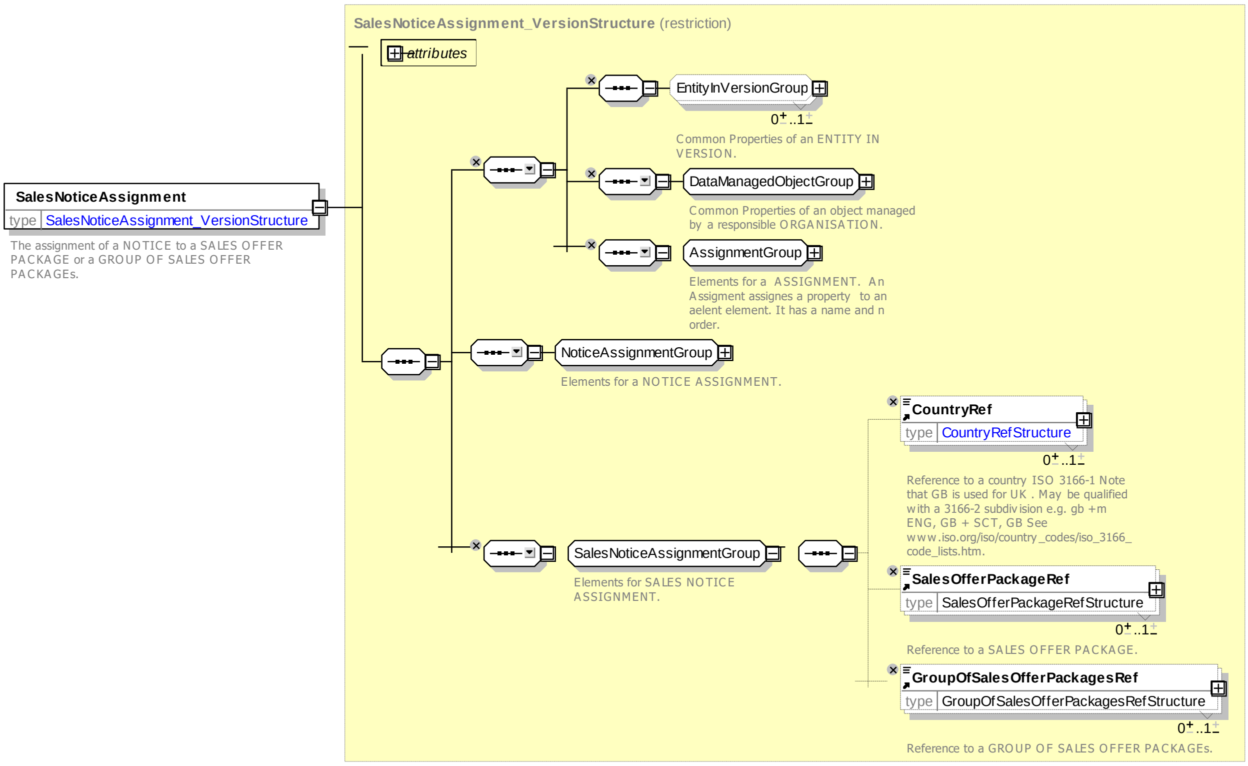


Figure 334 — *SalesNoticeAssignment* — XSD

7.8.3.3.5 SalesOfferPackageSubstitution – Model Element

Information on the preferred substitution of packages with another package if quota restricted product is no longer available. The relative priority is specified using the ***order*** attribute inherited from ASSIGNMENT.

Table 310 – *SalesOfferPackageSubstitution* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Assignment*	::>	SALES OFFER PACKAGE SUBSTITUTION inherits from ASSIGNMENT.
«PK»	***id***	*SalesOfferPackage-SubstitutionIdType*	1:1	Identifier of SALES OFFER PACKAGE SUBSTITUTION.
«FK»	***SalesOffer-PackageRef***	*SalesOfferPackageRef*	0:1	First SALES OFFER PACKAGE in combination. If not given by containing context must be specified.
«FK»	***WithSalesOffer-PackageRef***	*SalesOfferPackageRef*	1:1	Second SALES OFFER PACKAGE in combination.

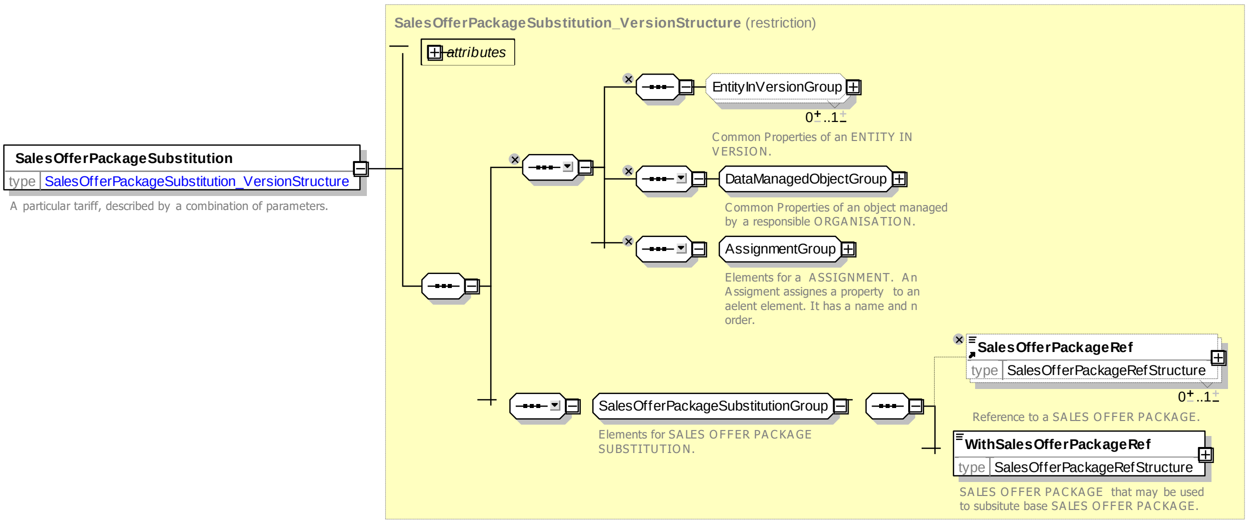


Figure 335 — *SalesOfferPackageSubstitution* — XSD

7.8.3.3.6 **TypeOfSalesOfferPackage – Model Element**

A classification of a SALES OFFER PACKAGE.

Table 311 – *TypeOfSalesOfferPackage* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF SALES OFFER PACKAGE inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfSalesOfferPackageIdType*	1:1	Identifier of TYPE OF SALES OFFER PACKAGE.

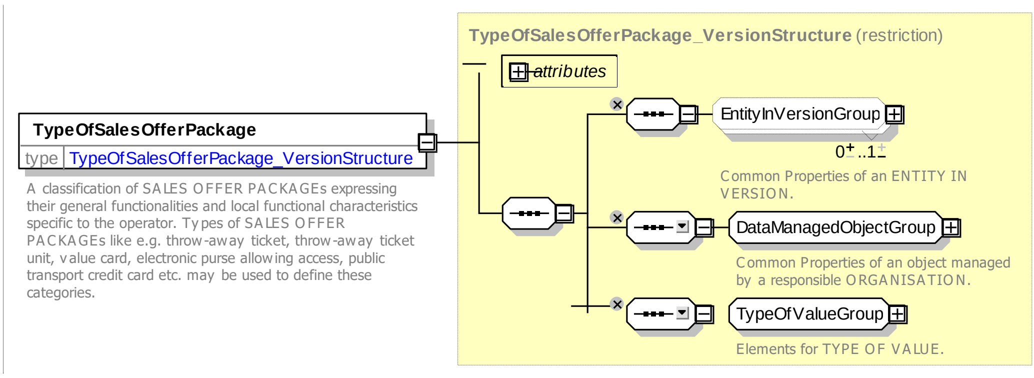


Figure 336 — *TypeOfSalesOfferPackage* — XSD

7.8.3.4 Fare Sales Offer Package – XML examples

7.8.3.4.1 **Sales Offer Package: XML Example of tickets**

The following code fragment defines separate SALES OFFER PACKAGEs for paper ticket, group ticket (also on paper) and an electronic ticket.

For EXAMPLE:

```xml
<salesOfferPackages>
    <SalesOfferPackage version="any" id="tfl:Paper_ticket">
        <Name>Single Mag or paper ticket is issued</Name>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesOfferPackageElements>
            <SalesOfferPackageElement version="any" id="tfl:Paper_ticket@PrepaidFare@single">
                <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
                <PreassignedFareProductRef version="any" ref="tfl:Prepaid_fare"/>
            </SalesOfferPackageElement>

        </salesOfferPackageElements>
    </SalesOfferPackage>

    <SalesOfferPackage version="any" id="tfl:Group_day_ticket">
        <Name>Group day ticket, paper</Name>
        <Description>For groups of 10 or more fare-paying passengers you can get Group Day
Tickets. these allow unlimited travel at any time and on any day within the zones paid for.
You can only get these as paper tickets, not on an Oyster card, and they must be purchased upon
arrival in London.</Description>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesOfferPackageElements>
            <SalesOfferPackageElement version="any" id="tfl:Group_day_ticket@FareProduct">
                <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
                <PreassignedFareProductRef version="any" ref="tfl:GroupDayTicket"/>
            </SalesOfferPackageElement>
        </salesOfferPackageElements>
    </SalesOfferPackage>

    <!-- Other Ticket -->
    <SalesOfferPackage version="any" id="tfl:PayAsYouGo_fare">
        <Name>On Card ticket purchase</Name>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <GoesOnCard>true</GoesOnCard>
            <IsRefundable>false</IsRefundable>
        </ConditionSummary>
        <salesOfferPackageElements>
            <SalesOfferPackageElement version="any" id="tfl:PayAsYouGo_fare@single">
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <PreassignedFareProductRef version="any" ref="tfl:PayAsYouGo_fare"/>
            </SalesOfferPackageElement>
        </salesOfferPackageElements>
    </SalesOfferPackage>
```

7.8.3.4.2 Sales Offer Package: XML Example of card purchase

The following example (Based on the TfL Oyster card) shows a SALES OFFER PACKAGE for an electronic card that has separate SALES OFFER PACKAGE ELEMENTs for two different discount rights – the right to buy Pay as you go fares at a discount and the right to buy Travel card products at a discount.

For EXAMPLE:

```xml
    <SalesOfferPackage version="any" id="tfl:OnOysterCard">
        <Name>Product is carried on Oyster card </Name>
        <ConditionSummary>
            <ProvidesCard>true</ProvidesCard>
            <IsPersonal>false</IsPersonal>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesOfferPackageElements>
            <SalesOfferPackageElement version="any" id="tfl:OnOysterCard@Oyster_PayAsYouGo_right">
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
            </SalesOfferPackageElement>
            <SalesOfferPackageElement version="any" id="tfl:OnOysterCard@TCSDiIsco">
                <Name>Also can be used to get discount </Name>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
```

```
            <SaleDiscountRightRef version="any" ref="tfl:TCSDiIsco"/>
        </SalesOfferPackageElement>
    </salesOfferPackageElements>
</SalesOfferPackage>
```

7.8.3.4.3 Sales Offer Package: XML Example of card Top up products

SALES OFFER PACKAGEs may be used for marketable transactions other than simple ticket purchase. The following code fragment defines separate SALES OFFER PACKAGEs to (a) top up an oyster card by an arbitrary amount; and (b) to register for automatic top up

For EXAMPLE:

```
        <SalesOfferPackage version="any" id="tfl:Oyster_top_up">
            <Name>Card Top up Put value on card</Name>
            <ConditionSummary>
                <ProvidesCard>false</ProvidesCard>
                <GoesOnCard>true</GoesOnCard>
                <IsRefundable>true</IsRefundable>
            </ConditionSummary>
            <salesOfferPackageElements>
                <SalesOfferPackageElement version="any"
id="tfl:Oyster_top_up@PrepaidFare@Oyster_top_up">
                    <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                    <AmountOfPriceUnitProductRef version="any" ref="tfl:Oyster_top_up"/>
                </SalesOfferPackageElement>
            </salesOfferPackageElements>
        </SalesOfferPackage>

        <SalesOfferPackage version="any" id="tfl:Oyster_Auto_top_up">
            <Name>Automatically top up when credit threshold drops to specified amount</Name>
            <ConditionSummary>
                <ProvidesCard>false</ProvidesCard>
                <GoesOnCard>true</GoesOnCard>
                <IsRefundable>true</IsRefundable>
            </ConditionSummary>
            <salesOfferPackageElements>
                <SalesOfferPackageElement version="any"
                    id="tfl:Oyster_Auto_top_up@PrepaidFare@Oyster_top_up">
                    <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
                    <SupplementProductRef version="any" ref="tfl:Oyster_Auto_top_up"/>
                </SalesOfferPackageElement>
            </salesOfferPackageElements>
        </SalesOfferPackage>
```

7.8.3.4.4 Sales Offer Package: XML Example of Group of Sale Packages to share properties

There may be many similar SALES OFFER PACKAGEs sharing a common attributes. The following code fragment shows separate SALES OFFER PACKAGEs for various types of personalised Oyster Photo cards, all of which share common base properties defined by a GROUP OF SALES OFFER PACKAGEs.

For EXAMPLE:

```
<groupsOfSalesOfferPackages>
    <GroupOfSalesOfferPackages version="any" id="tfl:on_OysterCard_group">
        <Name>Current Products carried on Oyster card </Name>
        <ConditionSummary>
            <ProvidesCard>true</ProvidesCard>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesOfferPackageElements>
            <SalesOfferPackageElement version="any"
id="tfl:onOysterPhotoCardGroup@OysterPayAsYouGo">
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartPhotoCard"/>
                <CappedDiscountRightRef version="any" ref="tfl:Oyster_PayAsYouGo_right"/>
            </SalesOfferPackageElement>
            <SalesOfferPackageElement version="any"
                    id="tfl:onOysterPhotoCardGroup@TravelCard_on_Oyster">
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartPhotoCard"/>
                <PreassignedFareProductRef version="any" ref="tfl:TravelCard_on_Oyster"/>
```

```
            </SalesOfferPackageElement>
            <SalesOfferPackageElement version="any"
                    id="tfl:onOysterPhotoCardGroup@TCSDiIsco">
                <Name>Can also can be used to get discount </Name>
                <TypeOfTravelDocumentRef version="any" ref="tfl:smartPhotoCard"/>
                <SaleDiscountRightRef version="any" ref="tfl:TCSDiIsco"/>
            </SalesOfferPackageElement>
        </salesOfferPackageElements>
            <members>
                <SalesOfferPackageRef version="any" ref="tfl:60PlusLondonOysterPhotoCard"/>
                <SalesOfferPackageRef version="any" ref="tfl:11-15Zip_OysterPhotoCard"/>
                <SalesOfferPackageRef version="any" ref="tfl:16PlusZip_OysterPhotoCard"/>
                <SalesOfferPackageRef version="any" ref="tfl:18Plus_OysterPhotoCard"/>
                <SalesOfferPackageRef version="any" ref="tfl:Veterans_OysterPhotoCard"/>
                <SalesOfferPackageRef version="any" ref="tfl:Apprentice_OysterPhotocard"/>
            </members>
    </GroupOfSalesOfferPackages>
</groupsOfSalesOfferPackages>

<salesOfferPackages>
    <SalesOfferPackage version="any" id="tfl:60PlusLondonOysterPhotoCard">
        <Name>60 + London Oyster Product is carried on Oyster card </Name>

        <validityParameterAssignments>
            <GenericParameterAssignment
                <id="tfl: 60PlusLondonOysterPhotoCard@UserProfile">
                    <limitations>
                        <UserProfileRef version="any" ref="tfl: senior"/>
                    </limitations>
            </GenericParameterAssignment>
        </validityParameterAssignments>
        <GroupOfSalesOfferPackagesRef version="any" ref="tfl:on_OysterPhotoCard_group"/>
    </SalesOfferPackage>

    <SalesOfferPackage version="any" id="tfl:11-15Zip_OysterPhotoCard">
            <Name>11-15 ZIP Oyster Product is carried
            <validityParameterAssignments>
                <GenericParameterAssignment  id="tfl:11-15Zip_OysterPhotoCard@UserProfile">
                    <limitations>
                        <UserProfileRef version="any" ref="tfl:child11-15"/>
                    </limitations>
                </GenericParameterAssignment>
            </validityParameterAssignments>
            <GroupOfSalesOfferPackagesRef version="any" ref="tfl:on_OysterPhotoCard_group"/>
        </SalesOfferPackage>
Etc., etc
</salesOfferPackages>
```

8 Sales Transactions

NeTEx separates the data generated by use of a system to sell fares (SALES TRANSACTIONs) from the fare structure use to locate products and prices to be referenced in the products. Typically, SALES TRANSACTIONs, will be generated each day in relatively large volume and also are grouped and exchanged in a SALES TRANSACTION FRAME. SALES TRANSACTION may contain references to fare structure elements and prices, but the description of these (Which change less frequently) is exchanged separately in a FARE FRAME as described above. Both types of frame may be grouped together in a single COMPOSITE FRAME if both sets of data are needed.

8.1 Sales Transaction – Model dependencies

NeTEx Part3 Sales Transaction model is modularised into a number of submodels defined as UML packages, these in turn depend on Part3 and Part1 packages and:

— The FARE CONTRACT model describes identified CUSTOMERs and their FARE CONTRACTs.

— The RETAIL model identifies RETAIL CONSORTIUMs, ORGANISATIONs who sell products, and RETAIL DEVICEs used to sell products.

— The SALES TRANSACTION model records sales of SALES OFFER PACKAGEs. TRAVEL SPECIFICATIONs describe each specific selection of theoretical fare elements for an individual SALES TRANSACTION.

— The SALES TRANSACTION FRAME model describes the elements used to group data for exchange.

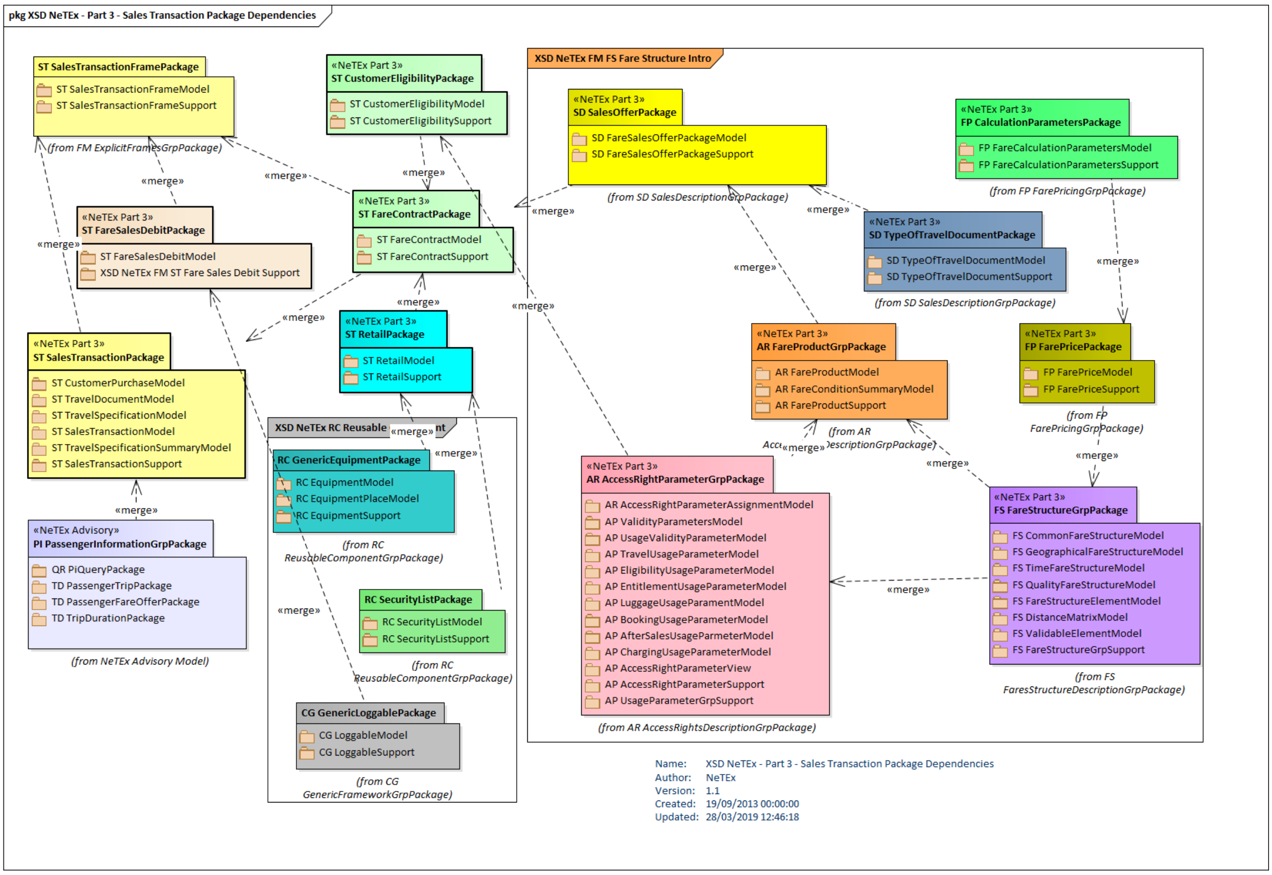


Figure 337 — Sales Transaction Package Dependencies (UML)

The following diagram shows the dependencies between the SALES TRANSACTION MODELs.

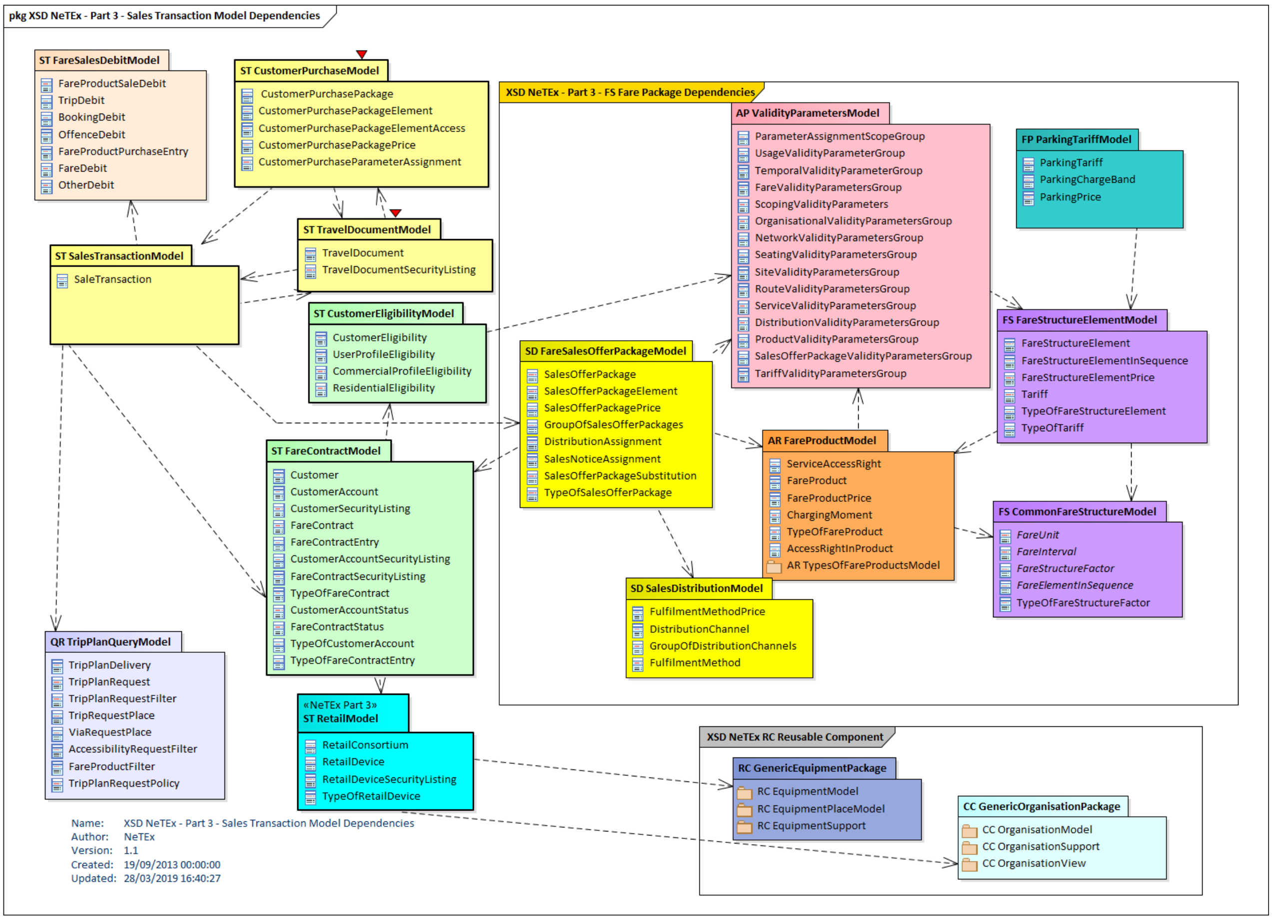


Figure 338 — Sales Transaction Model Dependencies (UML)

8.1.1 Sales Transaction Frame

8.1.1.1 SALES TRANSACTION FRAME – Conceptual MODEL

The elements of the SALES TRANSACTION MODEL and the FARE CONTRACT MODEL can be grouped with a SALES TRANSACTION FRAME, which holds a coherent set of fare sales related elements for the data exchange of fare transactions. It can be used to exchange descriptions of customers and their purchases. See VERSION FRAME in the NeTEx Framework section for general concepts relating to version frames.

Typically, large volumes of transaction data will be generated by day to day use of the system by customers as they buy tickets. The SALES TRANSACTION FRAME allows such data to be exchanged separately from the fare structures, products and prices (all of which will remain constant for extended periods of time), while still referencing the relevant static entities so that an importing system can correctly interpret the data.

SALES TRANSACTION FRAMEs can be assembled as a coherent, versioned set along with other types of NeTEx Data in other frames, for example a separate FARE FRAMEs defining fare structure data that is referenced by the sales transactions. The components of a SALES TRANSACTION FRAME are described in the following sections.

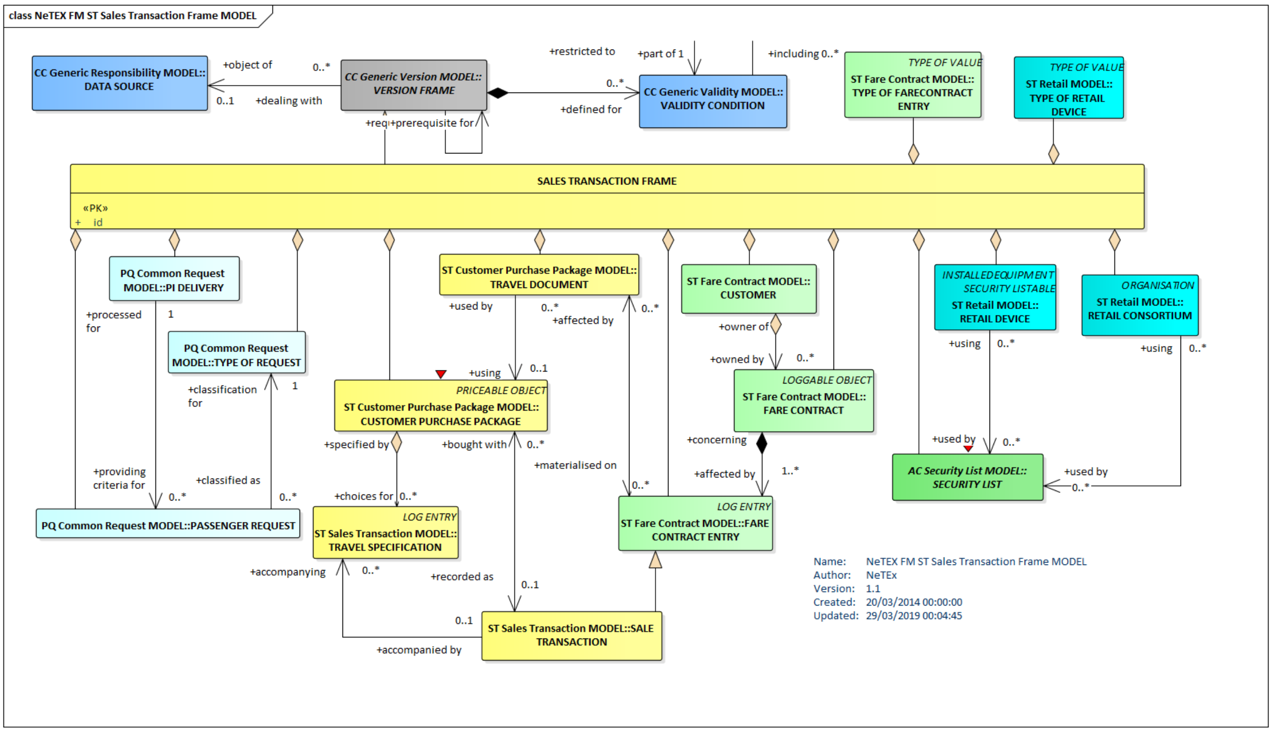


Figure 339 — Sales Transaction Frame – Conceptual MODEL (UML)

8.1.1.2 Sales Transaction Frame – Physical Model

The following diagram shows the Physical model for a SALES TRANSACTION FRAME.

A SALES TRANSACTION FRAME groups together sets of CUSTOMER and SALES TRANSACTION data for exchange between systems.

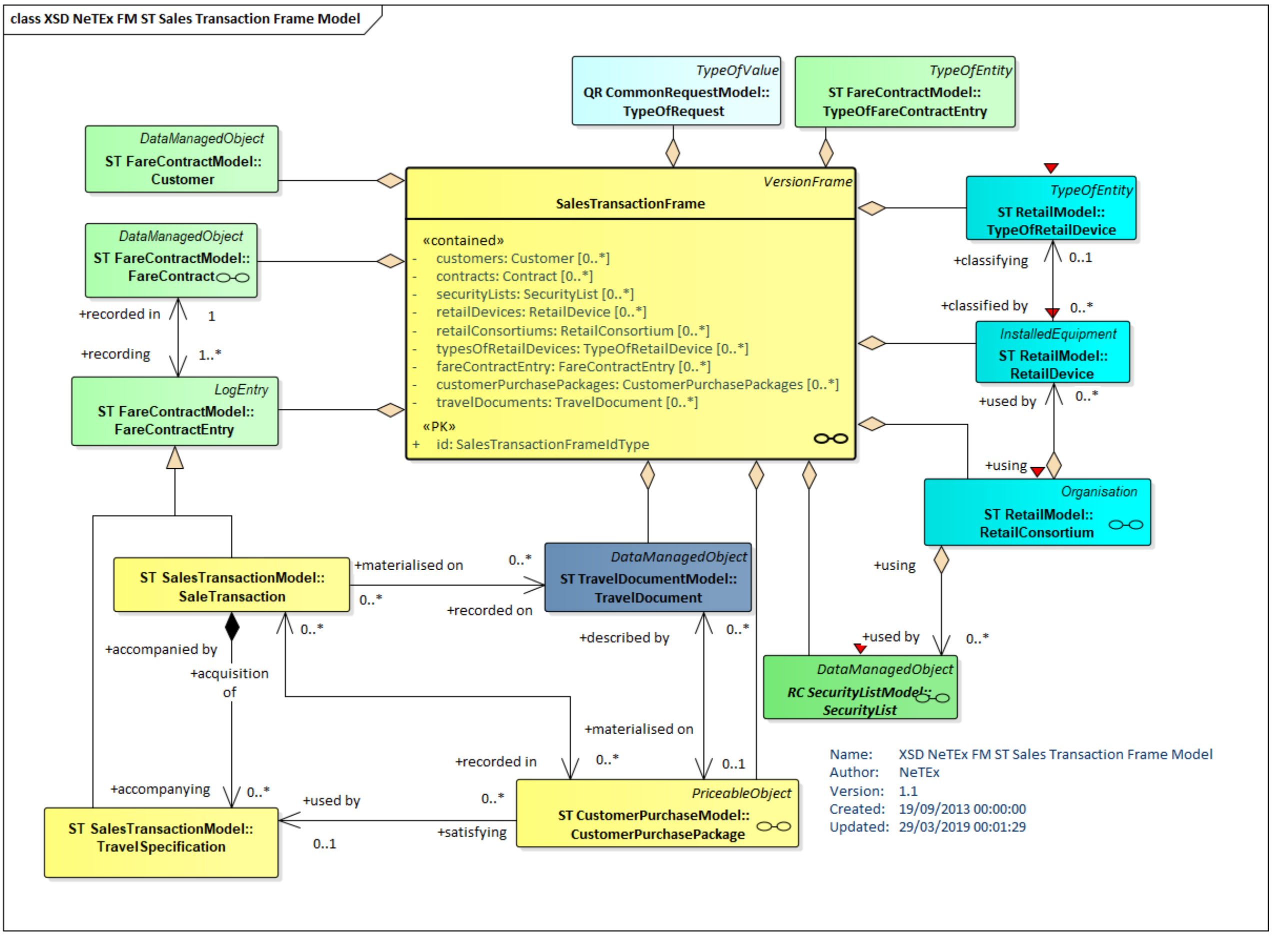


Figure 340 — Sales Transaction Frame – Physical Model Detail (UML)

8.1.1.3 Sales Transaction Frame – Attributes and XSD

8.1.1.3.1 SalesTransactionFrame – Model Element

A set of SALES TRANSACTION data elements (CUSTOMERs, SALES TRANSACTIONs, etc.) to which the same VALIDITY CONDITIONs have been assigned.

Table 312 – *SalesTransactionFrame* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	<u>*VersionFrame*</u>	::>	SALES TRANSACTION FRAME inherits from VERSION FRAME.
«PK»	***id***	*SalesTransactionFrame IdType*	1:1	Identifier of SALES TRANSACTION FRAME.
«cntd»	***retail-Consortiums***	<u>*RetailConsortium*</u>	0:*	RETAIL CONSORTIUMs in SALES TRANSACTION FRAME.
«cntd»	***retailDevices***	<u>*RetailDevice*</u>	0:*	RETAIL DEVICEs in SALES TRANSACTION FRAME.

«cntd»	***customers***	*Customer*	0:*	CUSTOMERs in SALES TRANSACTION FRAME.
«cntd»	***customer-Accounts***	*CustomerAccount*	0:*	CUSTOMER ACCOUNTs in SALES TRANSACTION FRAME. +v1.1
«cntd»	***fareContracts***	*FareContract*	0:*	FARE CONTRACTs in SALES TRANSACTION FRAME.
«cntd»	***blacklists***	*BlackList*	0:*	BLACKLISTs in SALES TRANSACTION FRAME.
«cntd»	***whitelists***	*WhiteList*	0:*	WHITELISTs in SALES TRANSACTION FRAME. +v1.1
«cntd»	***travel-Specifications***	*TravelSpecification*	0:*	TRAVEL SPECIFICATIONs in SALES TRANSACTION FRAME.
«cntd»	***typesOfTravel-Documents***	*TypeOfTravelDocument*	0:*	TYPEs OF TRAVEL DOCUMENTs in SALES TRANSACTION FRAME.
«cntd»	***travelDocuments***	*TravelDocument*	0:*	TRAVEL DOCUMENT in SALES TRANSACTION FRAME.
«cntd»	***customer-Purchase-Packages***	*CustomerPurchase-Package*	0:*	CUSTOMER PURCHASE PACKAGEs in SALES TRANSACTION FRAME.

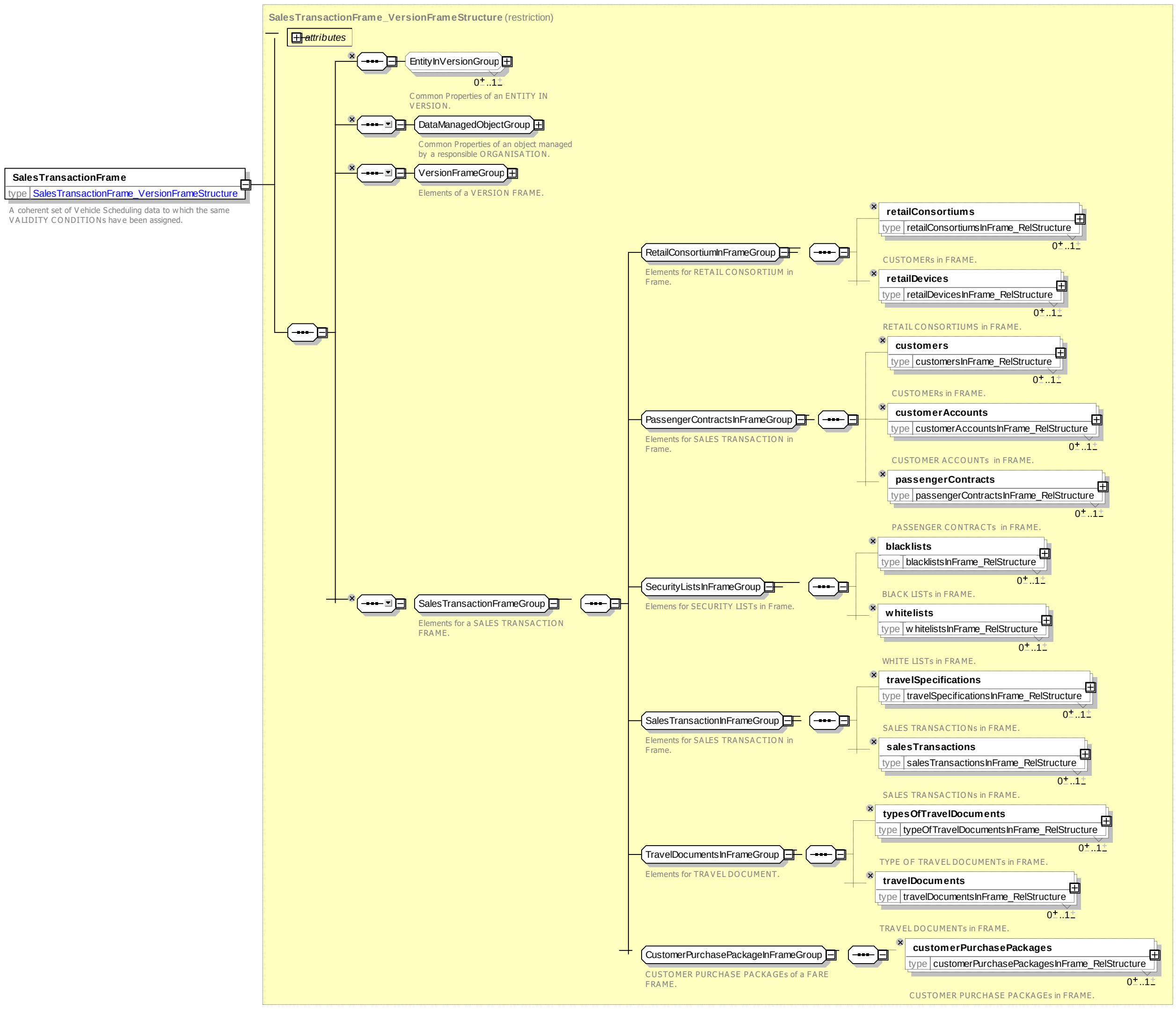


Figure 341 — *SalesTransactionFrame* — XSD

8.1.2 Fare Contract

8.1.2.1 FARE CONTRACT – Conceptual MODEL

The FARE CONTRACT MODEL describes CUSTOMERs for public transport. The purchase of a ticket implies a contract between the customer and the operator. The CONTRACT ENTRies that record events that take place such as booking or collecting a ticket, obtaining a refund, etc, are all subject to this contract.

NeTEx provides a minimal representation of this FARE CONTRACT which can be used to organise and track ticket purchases. A TRAVEL SPECIFICATION is a specialisation of CONTRACT ENTRY used to record a ticket PURCHASE.

CONTRACTs and CUSTOMERs may be placed on a BLACKLIST to identify them to systems for control purposes.

8.1.2.1.1 **Fare Contracts**

Classical TRAVEL DOCUMENTs are anonymous, i.e. without any registration of the user (payer and/or consumer). Modern techniques allow an individual registration, enabling additional commercial possibilities (e.g. special offers in accordance with the customer profile, post-payment, etc.). This can be represented by an explicit “contract” entity between the user and the vendor (Note that formally an implicit legal contract exists regardless of any such explicit representation). A contract is agreed between a customer and an organisation in charge of collecting fares for using services (authority, operator or another service provider), this consumption being ruled by the contract liabilities. Such an agreement is described by the FARE CONTRACT entity. Various types of FARE CONTRACTs may exist: single classical fare product, discount contract, identified card allowing the purchase of monthly passes, etc.

A FARE CONTRACT is immaterial – no contract document exists normally - and is in principle materialised by a particular TRAVEL DOCUMENT. However, this relationship is not one-to-one:

- only a subset of a FARE CONTRACT may be stored on a TRAVEL DOCUMENT;
- a CUSTOMER PURCHASE PACKAGE describes the detailed parameters underlying a specific purchase.
- FARE CONTRACT may be not (yet) stored on a TRAVEL DOCUMENT (e.g. in case of booking via Internet);
- a FARE CONTRACT may be subject to be stored on several TRAVEL DOCUMENTs (e.g. a card identifying the customer associated to monthly coupons);
- the TRAVEL DOCUMENT associated with a FARE CONTRACT may be replaced (in case of loss, failure, etc.);
- the same TRAVEL DOCUMENT may, in some cases, contain several FARE CONTRACTs.
- This variability necessitates making both concepts independent.

8.1.2.1.2 Customer Accounts

A customer may be registered for one or more CUSTOMER ACCOUNTS; an account holds a balance and payment preference. FARE CONTRACTS and other SALE TRANSACTIONs may be allocated to a specific CUSTOMER ACCOUNT.

8.1.2.1.3 Transport Customers

Any FARE CONTRACT concerns an individual customer, whose identity is only registered in some cases (in particular, if the equipment allows this). A classical disposable single ticket does not bear the holder’s identity and, even with modern fare collection techniques, the customer may remain anonymous.

Identified customers are described by the CUSTOMER entity. A CUSTOMER is registered with his/her identity and the relevant characteristics, among which are a USER PROFILE and possibly a COMMERCIAL PROFILE. A registered CUSTOMER may sometimes be an organisation, in case of a FARE PRODUCT possibly used by several persons (TRANSFERABILITY parameter).

The CUSTOMER entity describes the person or organisation allowed to consume the services, and not the person or organisation having paid for this usage. The latter is often related to a FARE CONTRACT account. Such concepts, dealing with payment procedures, are not covered by the reference model.

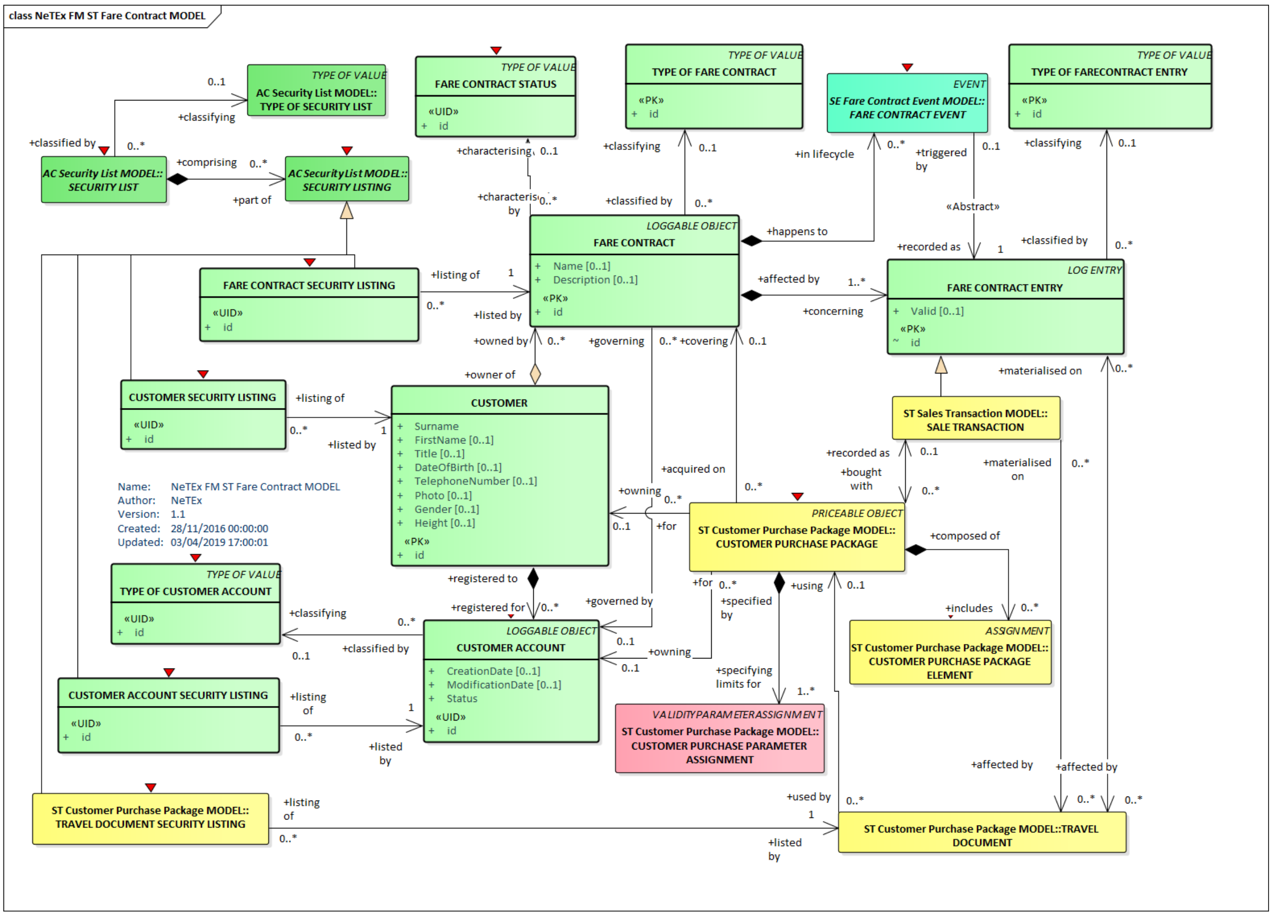


Figure 342 — Fare Contract – Conceptual MODEL (UML)

8.1.2.2 Fare Contract – Physical model

The following figure shows the physical model for FARE CONTRACT s.

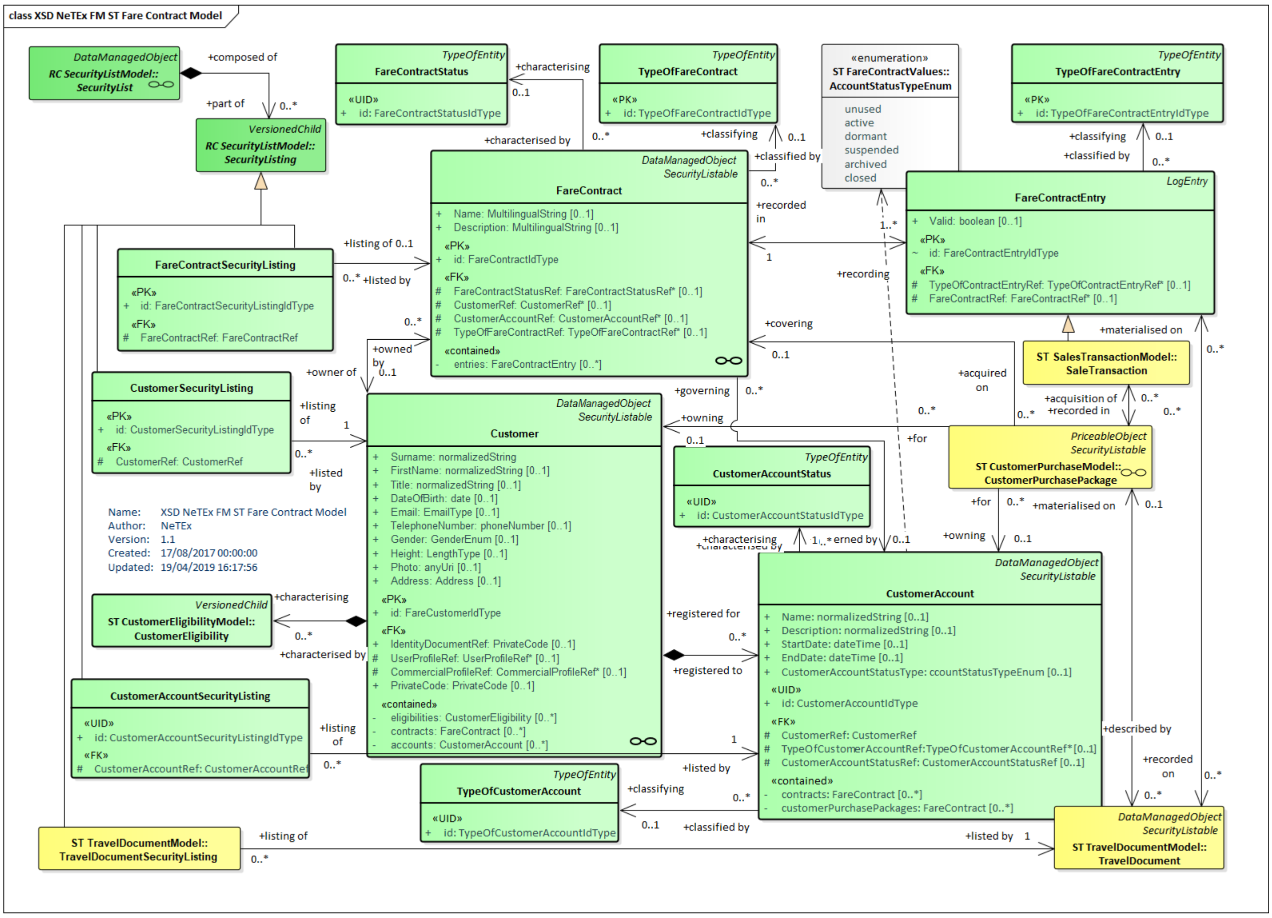


Figure 343 — Fare Contract – Physical Model (UML)

8.1.2.3 Fare Contract – Attributes and XSD

8.1.2.3.1 Customer – Model Element

An identified person or organisation involved in a fare process. There may be a FARE CONTRACT between the CUSTOMER and the OPERATOR or the AUTHORITY ruling the consumption of services.

Table 313 – *Customer* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	CUSTOMER inherits from DATA MANAGED OBJECT.
«PK»	***id***	*FareCustomerIdType*	1:1	Identifier of a CUSTOMER.
	Surname	*xsd:normalizedString*	0:1	Surname of a FARE CUSTOMER
	FirstName	*xsd:normalizedString*	0:1	First name of a CUSTOMER.
	Title	*xsd:normalizedString*	0:1	Title of address of a CUSTOMER.
	DateOfBirth	*xsd:date*	0:1	Date of Birth of a CUSTOMER.
«enum»	***Gender***	*GenderEnum*	0:1	Gender of CUSTOMER. See Part1 facilities.

	Height	*LengthType*	0:1	Height of CUSTOMER.
	Photo	*xsd:anyUri*	0:1	Photo of CUSTOMER.
	email	*EmailAddressType*	0:1	Email address of a CUSTOMER. +v1.1
	Phone	*TelephoneContact*	0:1	Telephone number of a CUSTOMER.
	PostalAddress	*Address*	0:1	Postal address of a CUSTOMER.
«FK»	***Identity-DocumentRef***	*PrivateCode*	0:1	A document that identifies the CUSTOMER.
«cntd»	***customer-Eligibilities***	*CustomerEligibility*	0:*	CUSTOMER ELIGIBILITies for the CUSTOMER.
«cntd»	***customer-Accounts***	*CustomerAccount*	0:*	CUSTOMER ACCOUNTs for the CUSTOMER.
«cntd»	***fareContracts***	*FareContract*	0:*	FARE CONTRACTs for the CUSTOMER.

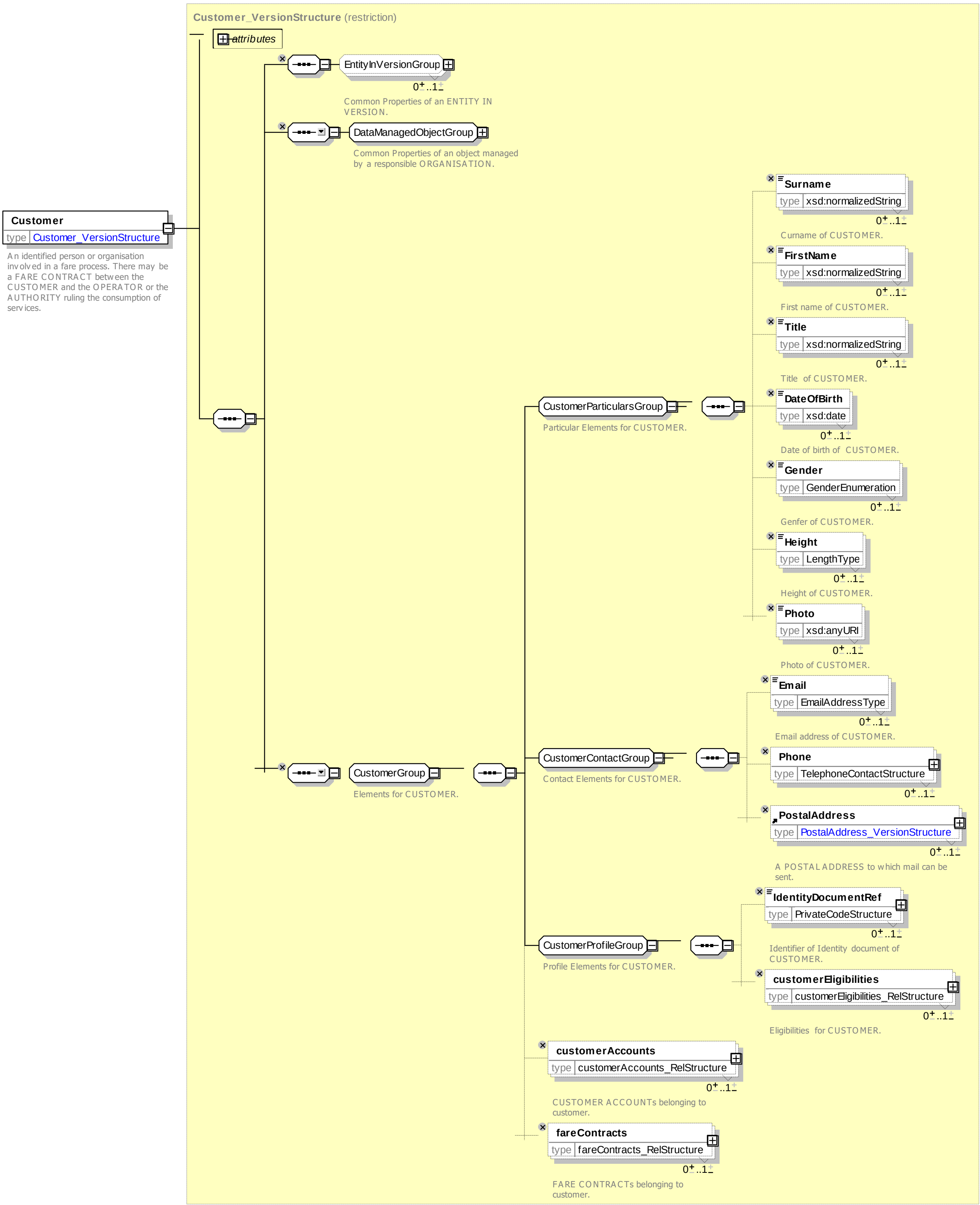


Figure 344 — *Customer* — XSD

8.1.2.4 CustomerSecurityListing – Model Element

A listing of a CUSTOMER on a SECURITY LIST.

Table 314 – CustomerSecurityListing – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*SecurityListing*	::>	CUSTOMER SECURITY LISTING inherits from SECURITY LISTING.
«PK»	***id***	*CustomerSecurityListing-IdType*	1:1	Identifier of CUSTOMER SECURITY LISTING.
«FK»	***CustomerRef***	*CustomerRef*	0:1	Reference to a CUSTOMER who is on list.

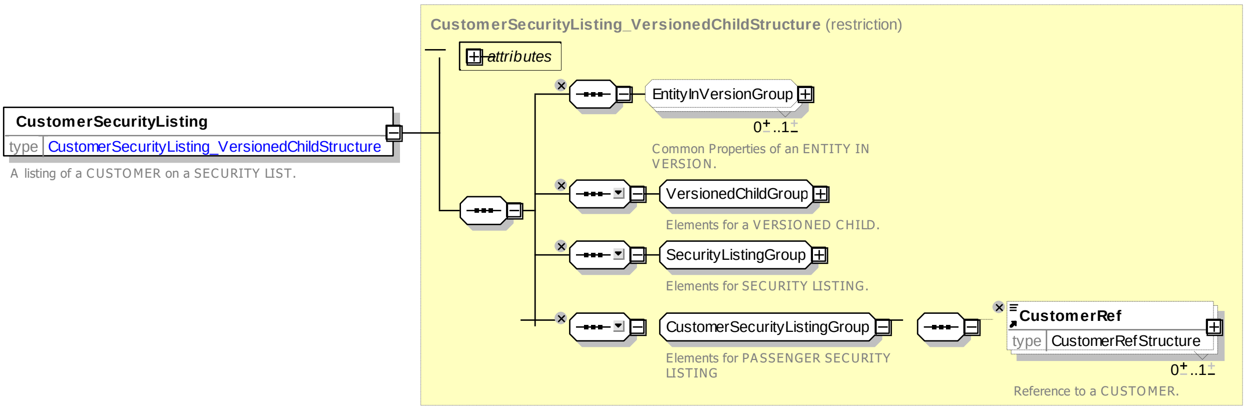


Figure 345 — CustomerSecurityListing — XSD

8.1.2.4.1 CustomerAccount – Model Element

An identified person or organisation involved in a fare process. There may be a FARE CONTRACT between the CUSTOMERACCOUNT and the OPERATOR or the AUTHORITY ruling the consumption of services.

Table 315 – *CustomerAccount* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*DataManagedObject*	::>	CUSTOMER ACCOUNT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*CustomerAccountIdType*	1:1	Identifier of a CUSTOMERACCOUNT.
	Name	*MultilingualString*	0:1	Name of CUSTOMER ACCOUNT.
	Description	*MultilingualString*	0:1	Description of CUSTOMER ACCOUNT.
	StartDate	*dateTime*	0:1	Creation date of CUSTOMER ACCOUNT.
	EndDate	*dateTime*	0:1	Close date of CUSTOMER ACCOUNT.
«FK»	***CustomerRef***	*CustomerRef*	1:1	CUSTOMER that holds account

«FK»	***TypeOfCustomer AccountRef***	*TypeOfCustomerAccount Ref*	0:1	Type of CUSTOMER ACCOUNT.
«FK»	***Customer-Account-StatusRef***	*CustomerAccount-StatusRef*	0:1	Status of a CUSTOMER ACCOUNT.
«enum»	***Customer-Account-StatusType***	*Customer-Account-StatusTypEnum.*	0:1	Status Type of CUSTOMER ACCOUNT. See allowed values below. +v1.1
«FK»	***Commercial-ProfileRef***	*CommercialProfileRef*	0:1	Reference to a COMMERCIAL PROFILE to which the CUSTOMER ACCOUNT corresponds.
«cntd»	***fareContracts***	*FareContract*	0:*	FARE CONTRACTs for the CUSTOMER ACCOUNT.
«cntd»	***customer-Purchase-Packages***	*Customer-Purchase-Package*	0:*	CUSTOMER PURCHASE PACKAGEs for the CUSTOMERACCOUNT.

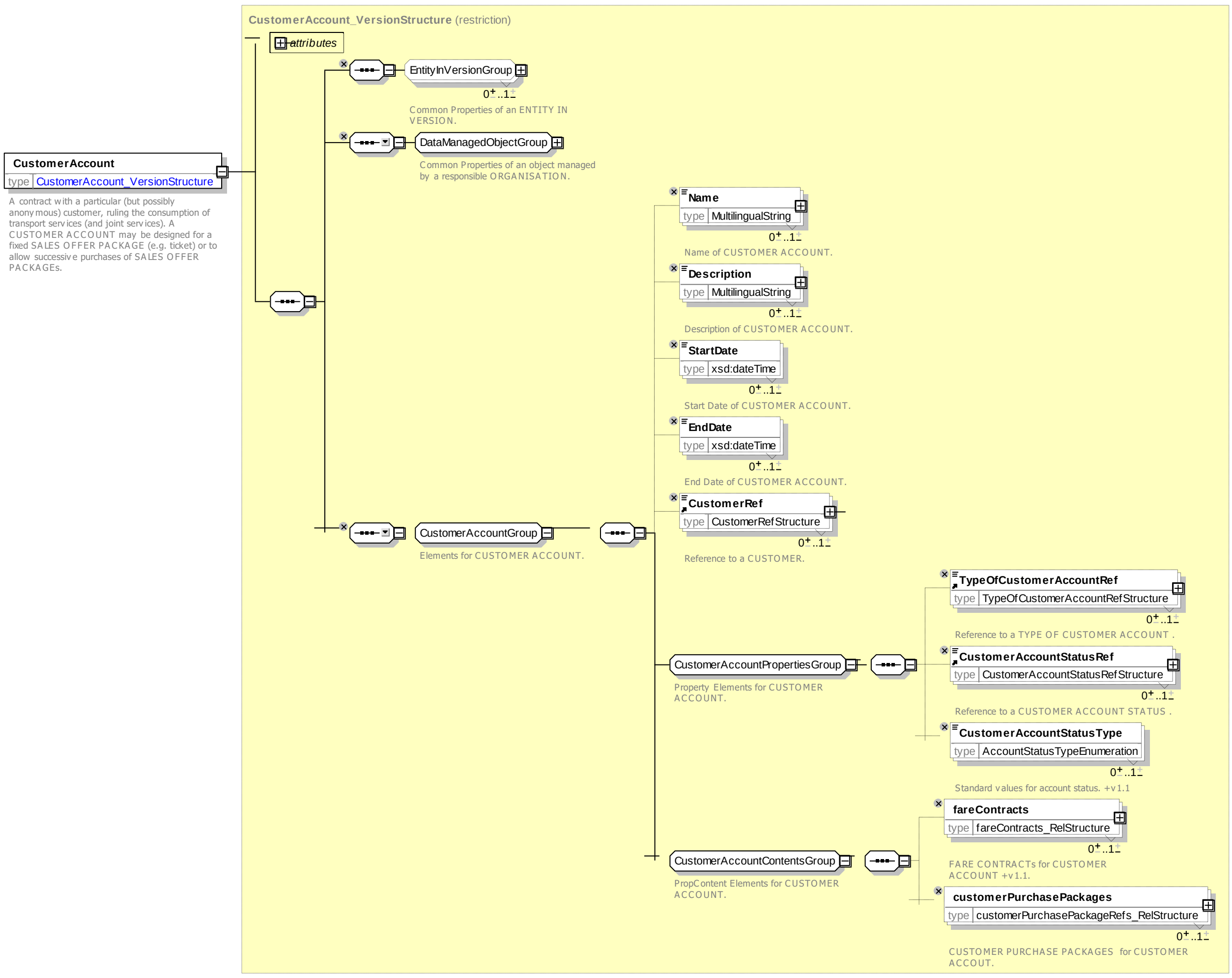


Figure 346 — *CustomerAccount* — XSD

8.1.2.4.2 CustomerAccountSecurityListing – Model Element

A listing of a CUSTOMER ACCOUNT on a SECURITY LIST.

Table 316 – CustomerAccountSecurityListing – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*SecurityListing*	::>	CUSTOMER ACCOUNT SECURITY LISTING inherits from SECURITY LISTING.
«PK»	***id***	*CustomerAccount-SecurityListingIdType*	1:1	Identifier of CUSTOMER ACCOUNT SECURITY LISTING.
«FK»	***Customer-AccountRef***	*CustomerAccountRef*	0:1	Reference to a CUSTOMER ACCOUNT.

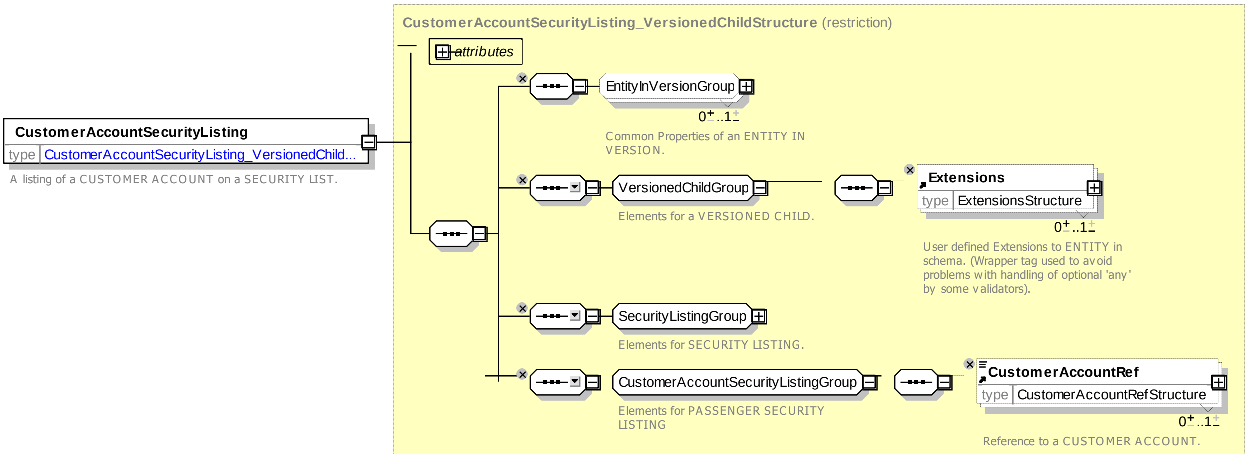


Figure 347 — CustomerAccountSecurityListing — XSD

8.1.2.4.3 CustomerAccountStatus– Model Element

The current status of a CUSTOMER ACCOUNT.

Table 317 – CustomerAccountStatus– Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	CUSTOMER ACCOUNT STATUS LISTING inherits TYPE OF VALUE.
«PK»	***id***	*CustomerAccountStatus-IdType*	1:1	Identifier of CUSTOMER ACCOUNT STATUS.

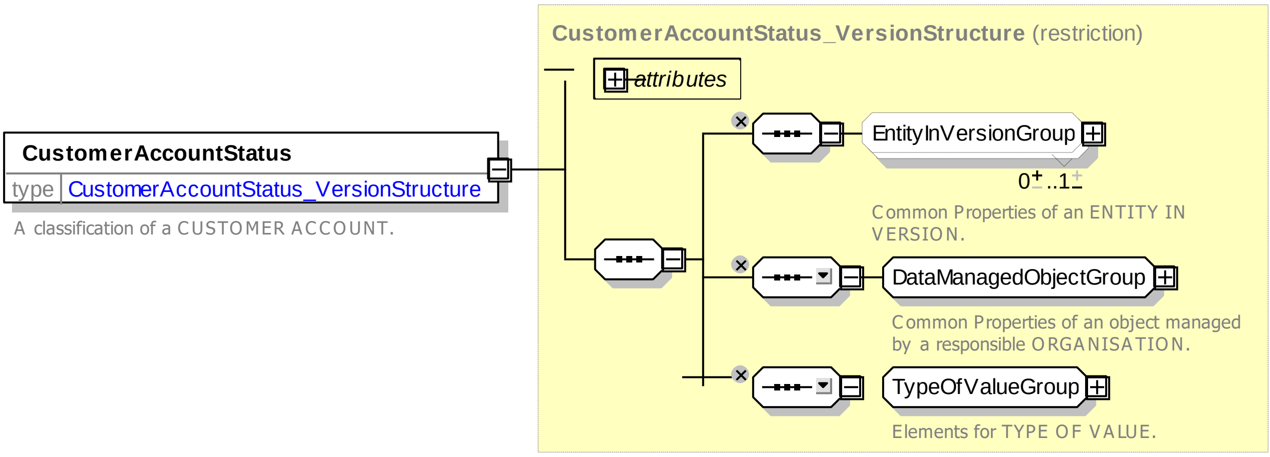


Figure 348 — CustomerAccountStatus— XSD

8.1.2.4.4 FareContract – Model Element

A contract with a particular (but possibly anonymous) customer, ruling the consumption of transport services (and joint services). A FARE CONTRACT may be designed for a fixed SALES OFFER PACKAGE (e.g. ticket) or to allow successive purchases of SALES OFFER PACKAGEs.

Table 318 – *FareContract* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*DataManagedObject*	::>	FARE CONTRACT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*FareContractIdType*	1:1	Identifier of a FARE CONTRACT.
	Name	*MultilingualString*	0:1	Name of FARE CONTRACT.
	Description	*MultilingualString*	0:1	Description of FARE CONTRACT.
	StartDate	*xsd:dateTime*	0:1	Start date of Contract
	EndDate	*xsd:dateTime*	0:1	End date of Contract
	Status	*xsd:normalizedString*	0:1	Status of FARE CONTRACT.
«FK»	***CustomerRef***	*CustomerRef*	0:1	Reference to the CUSTOMER for the FARE CONTRACT.
«FK»	***Customer-AccountRef***	*CustomerAccountRef*	0:1	Reference to the CUSTOMER ACCOUNT associated with the FARE CONTRACT. +V1.1
«FK»	***TypeOf-FareContractRef***	*TypeOfFareContractRef*	0:1	Type of FARE CONTRACT.
«cntd»	***fareContract-Entries***	*FareContractEntry*	0:*	FARE CONTRACT ENTRies making up the FARE CONTRACT.

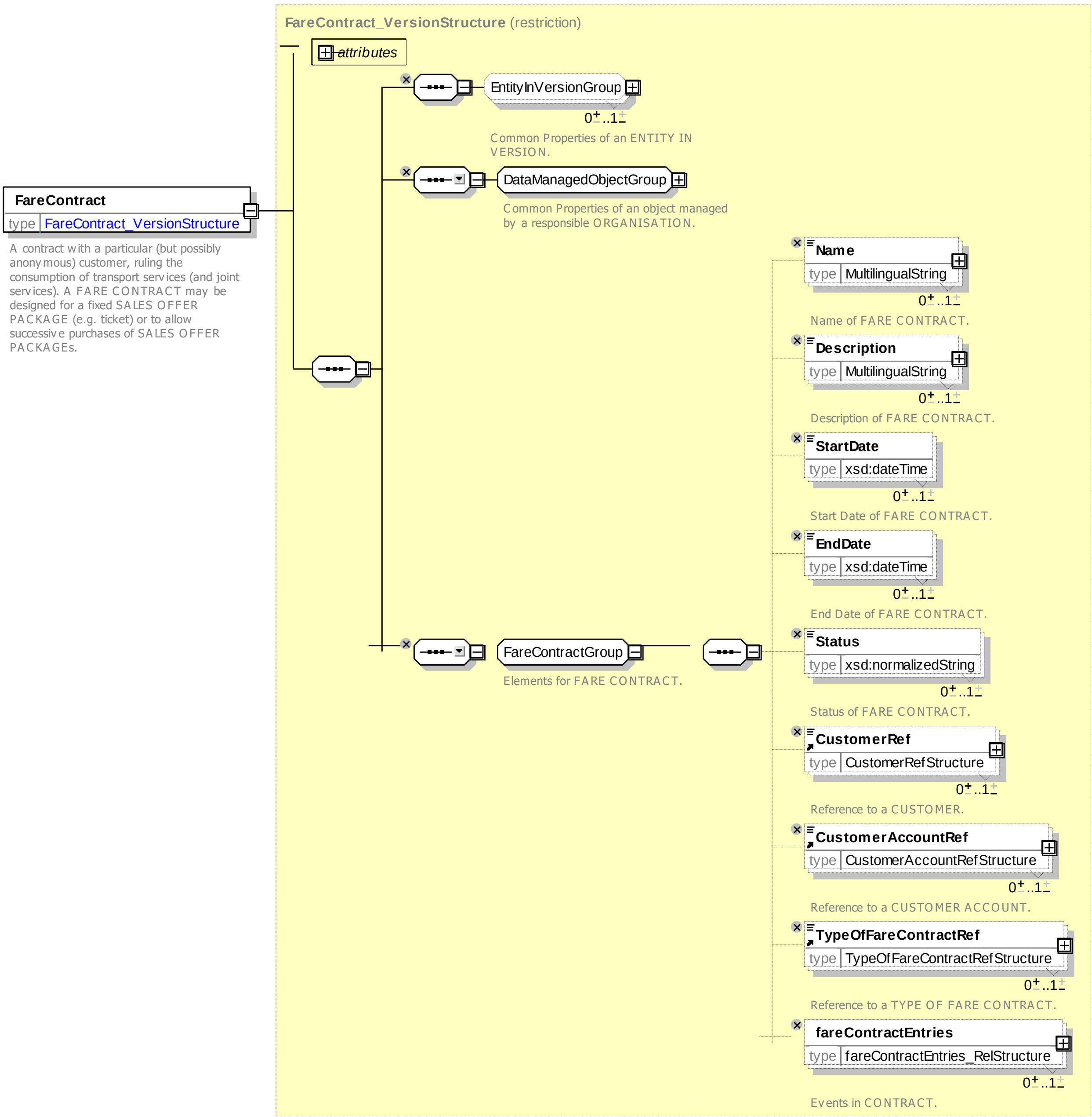


Figure 349 — *FareContract* — XSD

8.1.2.4.5 FareContractSecurityListing – Model Element

A listing of a FARE CONTRACT on a SECURITY LIST.

Table 319 – FareContractSecurityListing – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*SecurityListing*	::>	FARE CONTRACT SECURITY LISTING inherits from SECURITY LISTING.

«PK»	***id***	*FareContract-SecurityListingIdType*	1:1	Identifier of FARE CONTRACT SECURITY LISTING.
«FK»	***FareContractRef***	*FareContractRef*	0:1	Reference to a FARE CONTRACT.

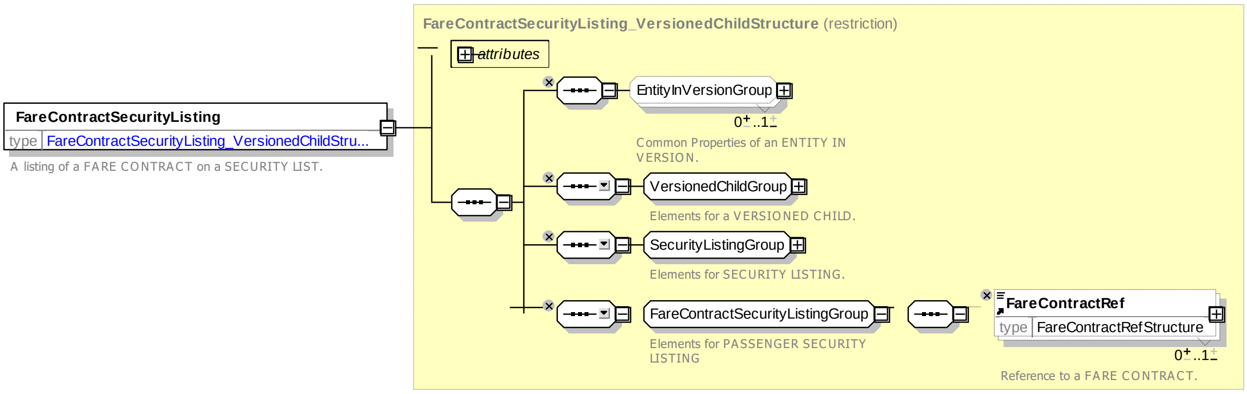


Figure 350 — FareContractSecurityListing — XSD

8.1.2.4.6 FareContractEntry– Model Element

A FARE CONTRACT ENTRY records an event occurring in the life of a -FARE CONTRACT: initial contracting, sales, validation entries, etc. A subset of a FARE CONTRACT ENTRY is often materialised on a TRAVEL DOCUMENT.

Transmodel 6.0 identifies numerous other types of log entry; these have not yet been implemented in NeTEx.

Table 320 – *FareContractEntry* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*LogEntry*	::>	FARE CONTRACT ENTRY inherits from LOG ENTRY. See NeTEx Part1.
«PK»	***id***	*FareContract-EntryIdType*	1:1	Identifier of FARE CONTRACT ENTRY.
	IsValid	*xsd:boolean*	0:1	Whether the FARE CONTRACT ENTRY is valid or not.
«FK»	***TypeOfFareContract-EntryRef***	*TypeOfFareContract-EntryRef*	0:1	Type of FARE CONTRACT ENTRY.
«FK»	***FareContractRef***	*FareContractRef*	0:1	Reference to a FARE CONTRACT to which the FARE CONTRACT ENTRY applies.

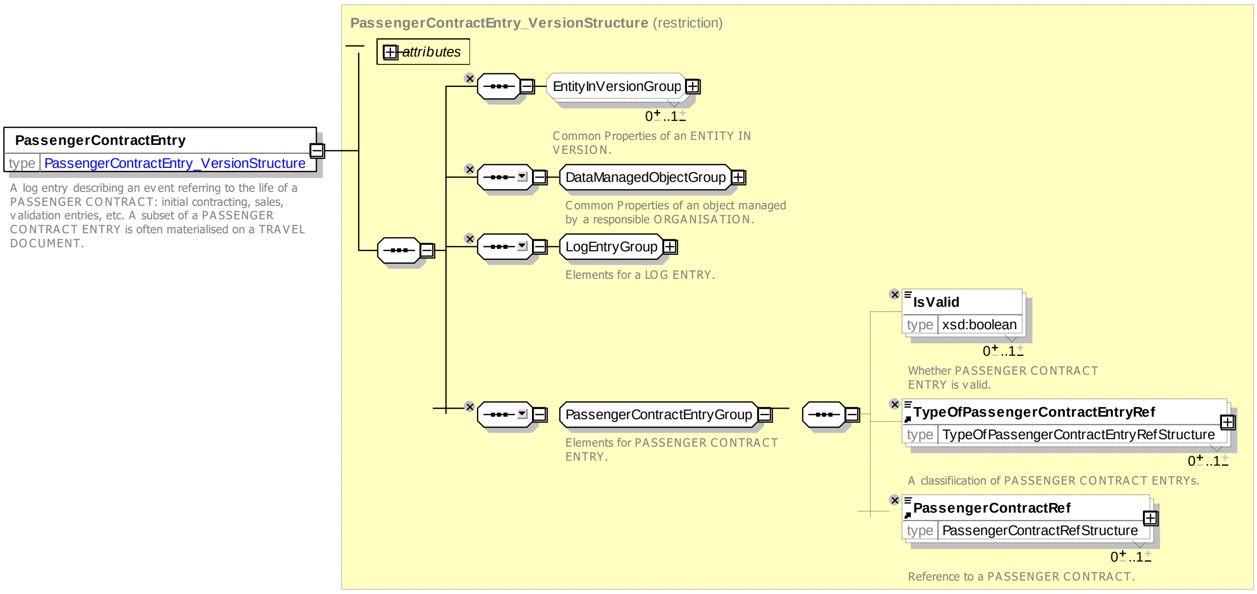


Figure 351 — *FareContractEntry*— XSD

8.1.2.4.7 TypeOfCustomerAccount – Model Element

A classification of CUSTOMER ACCOUNT.

Table 321 – *TypeOfCustomerAccount* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF CUSTOMER ACCOUNT inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfCustomerAccountIdType*	1:1	Identifier of TYPE OF CUSTOMER ACCOUNT.

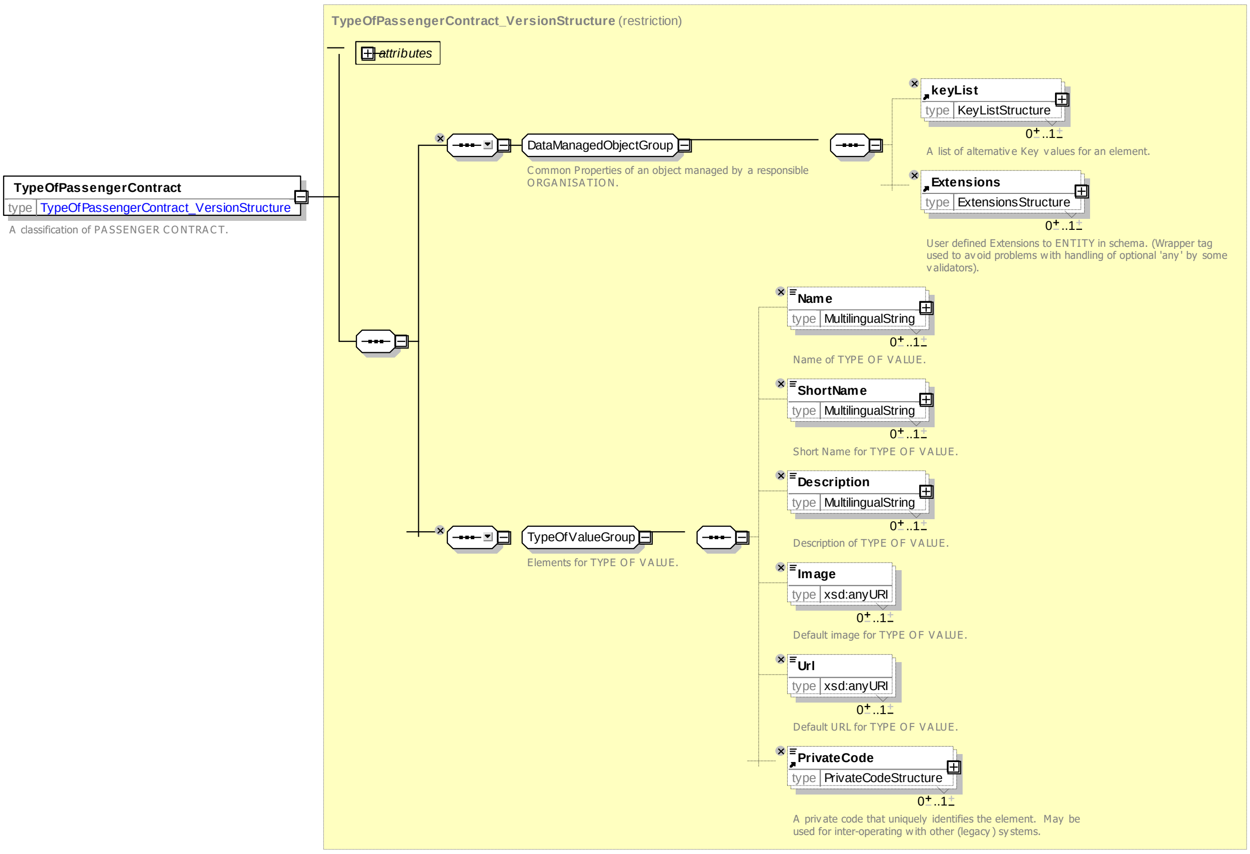


Figure 352 — *TypeOfCustomerAccount* — XSD

8.1.2.4.8 TypeOfFareContract – Model Element

A classification of FARE CONTRACT.

Table 322 – *TypeOfFareContract* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF FARE CONTRACT inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfFareContract-IdType*	1:1	Identifier of TYPE OF FARE CONTRACT.

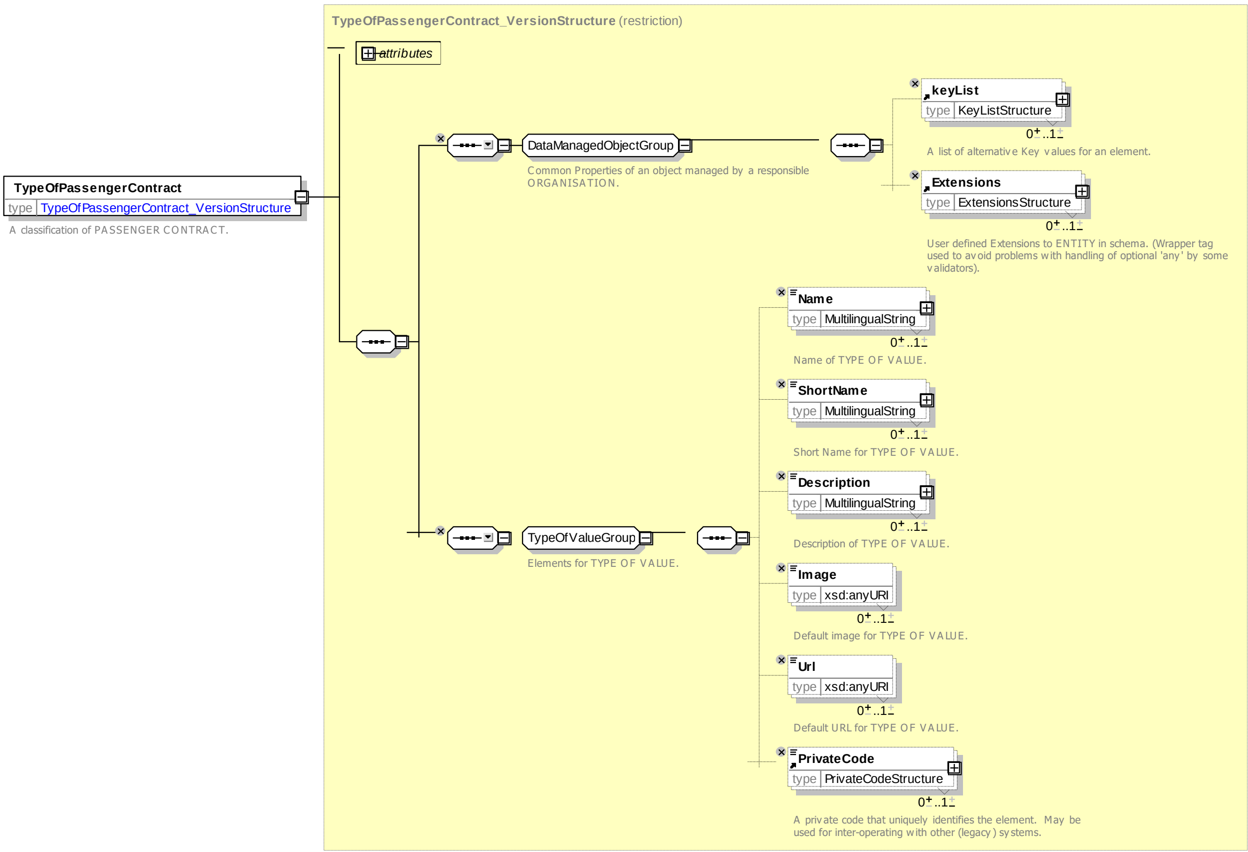


Figure 353 — *TypeOfFareContract* — XSD

8.1.2.4.9 TypeOfFareContractEntry – Model Element

A classification of FARE CONTRACT ENTRies.

Table 323 – *TypeOfFareContractEntry* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF FARE CONTRACT ENTRY inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfFareContractEntry-IdType*	1:1	Identifier of TYPE OF FARE CONTRACT ENTRY.

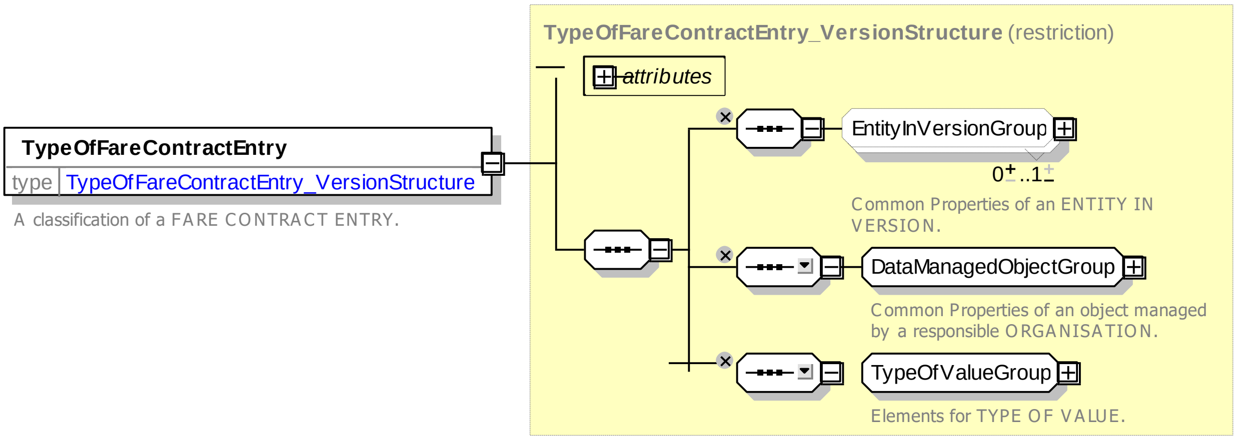


Figure 354 — *TypeOfFareContractEntry*— XSD

8.1.2.5 Fare Contract – XML Examples.

8.1.2.5.1 Fare Contract: XML Example of registered customer

The following code fragment shows a registered CUSTOMER.

For EXAMPLE:

```
<customers>
    <Customer version="any" id="xpl:122222">
        <Surname>Johnson</Surname>
        <FirstName>Boris</FirstName>
        <Title>Mr</Title> <Gender>male</Gender>
        <Height>180</Height>
        <UserProfileRef version="any" ref="tfl:adult"/>
        <fareContracts>
            <FareContract version="any" id="xpl:Oyster-12222">
                <Name> Registered Oyster card Oyster-122222 with Transactions </Name>
                <CustomerRef version="any" ref="xpl:122222"/>
                <fareContractEntries>
                    <SalesTransaction version="any" etc., etc.
….etc etc
                </fareContractEntries>
            </FareContract>
        </fareContracts>
    </Customer>
</customers>
```

8.1.2.5.2 Fare Contract: XML Example of Blacklist

The following code fragment shows a BLACKLIST with a two FARE CONTRACTs on it.

For EXAMPLE:

```
<blacklists>
    <Blacklist version="any" id="tfl:Stolen_cards">
    <Name>Stolen card list</Name>
    <fareContracts>
        <FareContractRef version="any" ref="xpl:Oyster-12222"/>
        <FareContractRef version="any" ref="xpl:Oyster-75632"/>
    </fareContracts>
    </Blacklist>
</blacklists>
```

8.1.3 Customer Eligibility

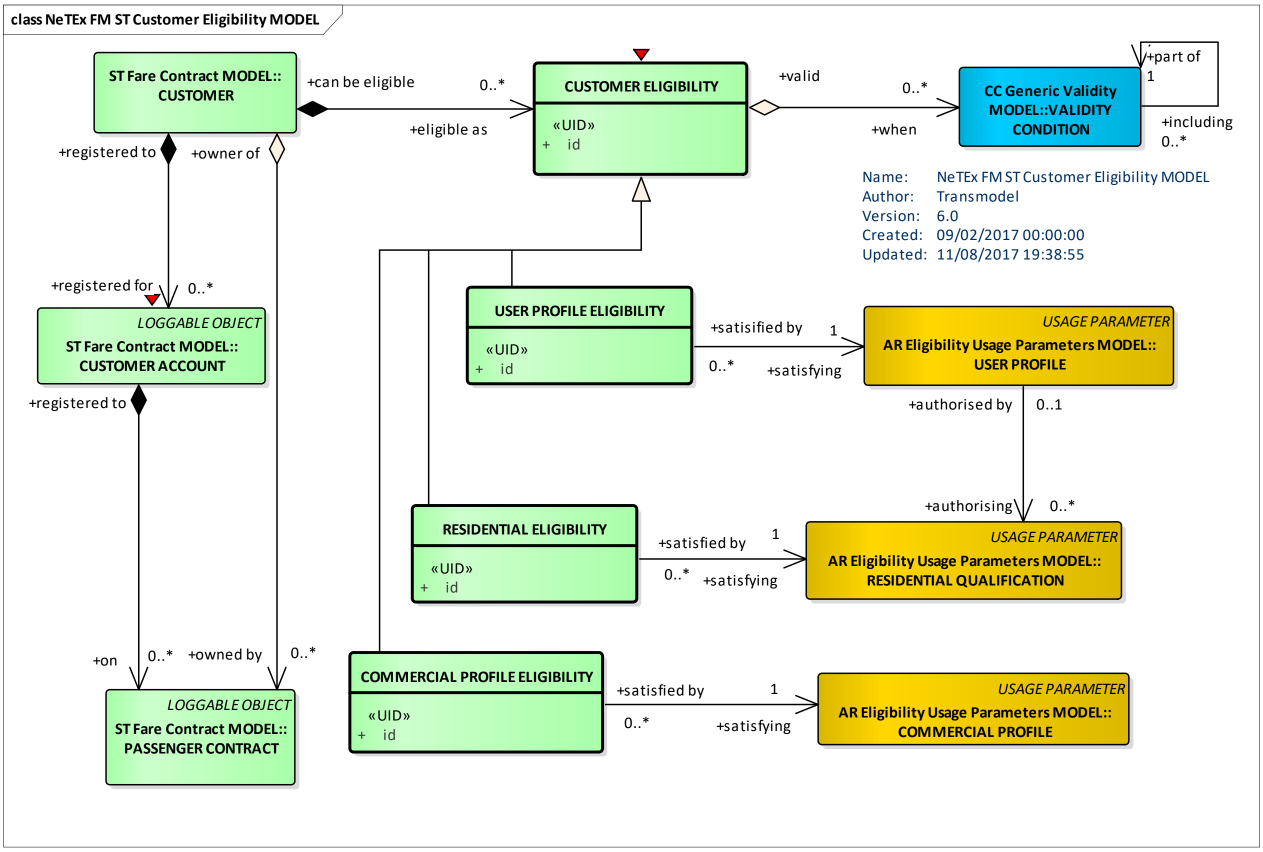
8.1.3.1 CUSTOMER ELIGIBILITY – Conceptual MODEL

The CUSTOMER ELIGIBILITY MODEL describes a CUSTOMER eligibility for different types of FARE PRODUCT. A CUSTOMER ELIGIBILITY is specialised to describe specific associations.

— USER PROFILE ELIGIBILITY indicates whether a customer satisfies a given USER PROFILE, e.g. adult, mobility impaired etc.;

— COMMERCIAL PROFILE ELIGIBILITY indicates whether a customer satisfies a given COMMERCIAL PROFILE, e.g. rail way staff, frequent traveller, etc.

— RESIDENTIAL ELIGIBILITY indicates whether a customer satisfies a residential qualification, for example to live in a city to get a free senior pass.



—

Figure 355 — Customer Eligibility – Conceptual MODEL (UML)

8.1.3.2 Customer Eligibility – Physical model

The following figure shows the physical model for FARE CONTRACTs.

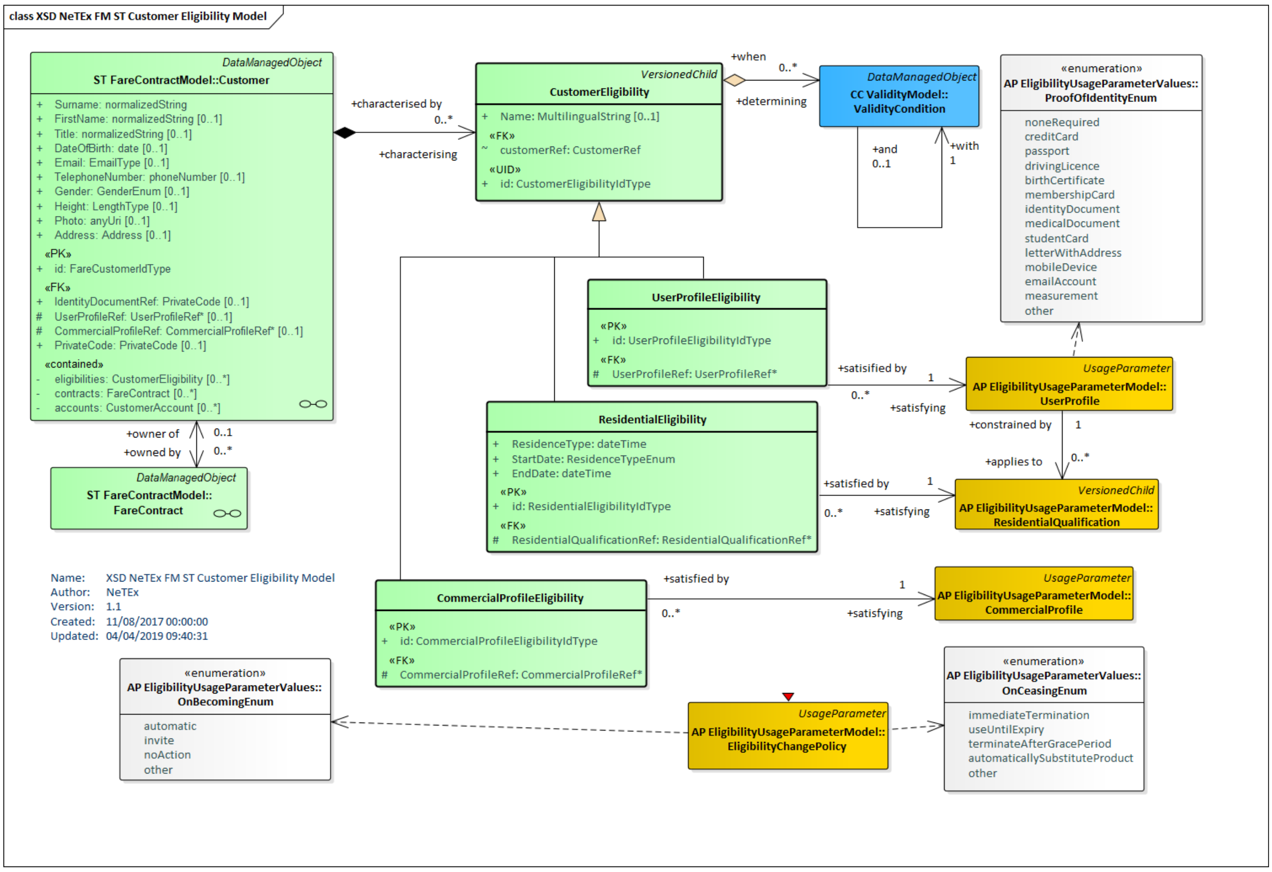


Figure 356 — Customer Eligibility – Physical Model (UML)

8.1.3.3 Customer Eligibility – Attributes and XSD

8.1.3.3.1 CustomerEligibility – Model Element

An identified person or organisation involved in a fare process. There may be a FARE CONTRACT between the CUSTOMER and the OPERATOR or the AUTHORITY ruling the consumption of services.

Table 324 – *CustomerEligibility* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*DataManagedObject*</u>	::>	CUSTOMER inherits from DATA MANAGED OBJECT.
«PK»	***id***	*CustomerEligibilityIdType*	1:1	Identifier of a CUSTOMER ELIGIBILITY.
	Name	*MultilingualString*	0:1	Name of CUSTOMER ELIGIBILITY.
«FK»	***CustomerRef***	*CustomerRef*	1:1	Reference to a CUSTOMER to whom eligibility applies.

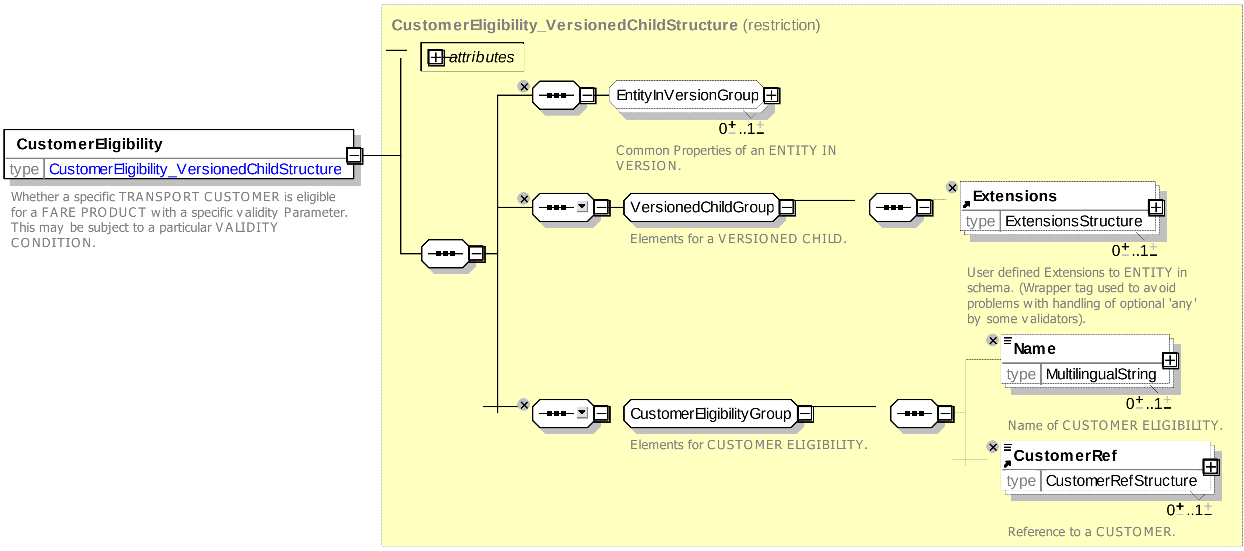


Figure 357 — *CustomerEligibility* — XSD

8.1.3.3.2 UserProfileEligibility – Model Element

Whether a specific TRANSPORT CUSTOMER is eligible for a FARE PRODUCT with a specific USER PROFILE as a validity parameter.

Table 325 – *UserProfileEligibility* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*CustomerEligibility*	::>	USER PROFILE ELIGIBILITY inherits from CUSTOMER ELIGIBILITY.
«PK»	***id***	*CustomerEligibilityIdType*	1:1	Identifier of a CUSTOMER ELIGIBILITY.
«FK»	***UserProfileRef***	*UserProfileRef*	1:1	Reference to a USER PROFILE to whom eligibility applies.

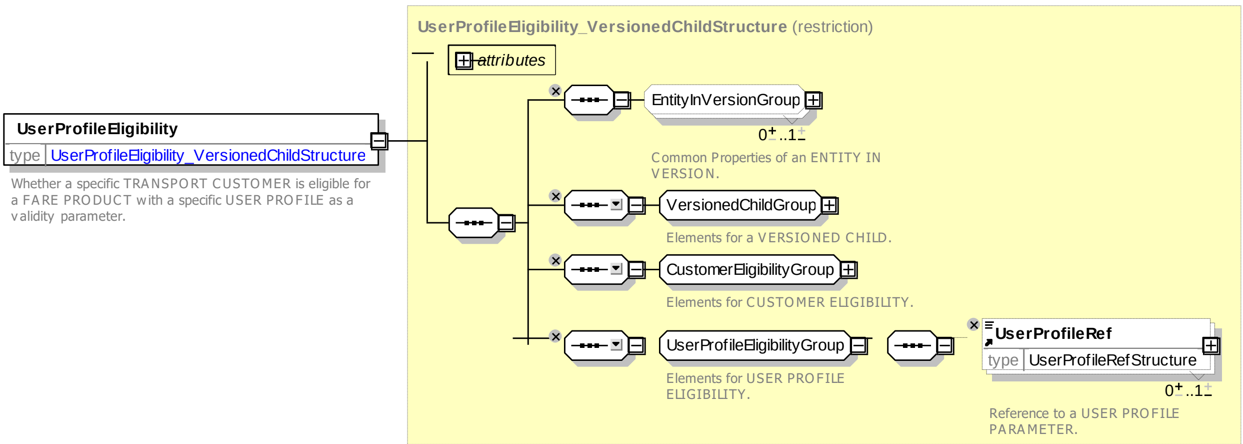


Figure 358 — *UserProfileEligibility* — XSD

8.1.3.3.3 CommercialProfileEligibility – Model Element

Whether a specific TRANSPORT CUSTOMER is eligible for a FARE PRODUCT with a specific COMMERCIAL PROFILE as a validity parameter.

Table 326 – *CommercialProfileEligibility* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*CustomerEligibility*	::>	COMMERCIAL PROFILE ELIGIBILITY inherits from CUSTOMER ELIGIBILITY.
«PK»	***id***	*CustomerEligibilityIdType*	1:1	Identifier of a CUSTOMER ELIGIBILITY.
«FK»	*CommercialProfile Ref*	*CommercialProfileRef*	1:1	Reference to a COMMERCIAL PROFILE to whom eligibility applies.

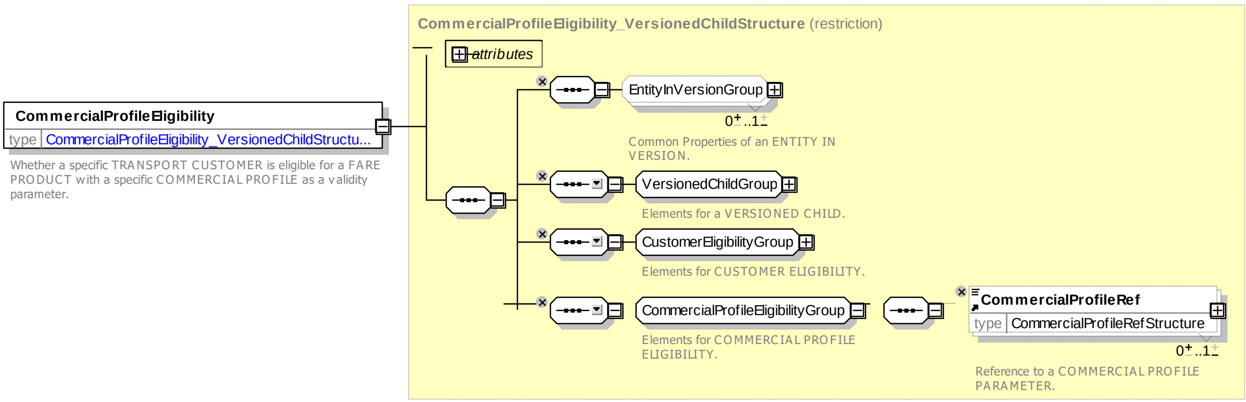


Figure 359 — *CommercialProfileEligibility* — XSD

8.1.3.3.4 ResidentialQualificationEligibility – Model Element

Whether a specific TRANSPORT CUSTOMER is eligible for a FARE PRODUCT with a specific RESIDENTIAL QUALIFICATION as a validity parameter.

Table 327 – *ResidentialQualificationEligibility* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*CustomerEligibility*	::>	RESIDENTIAL QUALIFICATION ELIGIBILITY inherits from CUSTOMER ELIGIBILITY.
«PK»	***id***	*CustomerEligibilityIdType*	1:1	Identifier of a CUSTOMER ELIGIBILITY.
«FK»	***Residential-QualificationRef***	*ResidentialQualification-Ref*	1:1	Reference to a RESIDENTIAL QUALIFICATION to whom eligibility applies.
«enum»	***ResidenceType***	*ResidenceTypeEnum*	1.0	Type of Residency. See allowed values for RESIDENTIAL QUALIFCATION.+v1.1
	StartDate	*xsd:date*	1.0	Date residence started. +v1.1

	EndDate	*xsd:date*	1.0	Date residence end. +v1.1

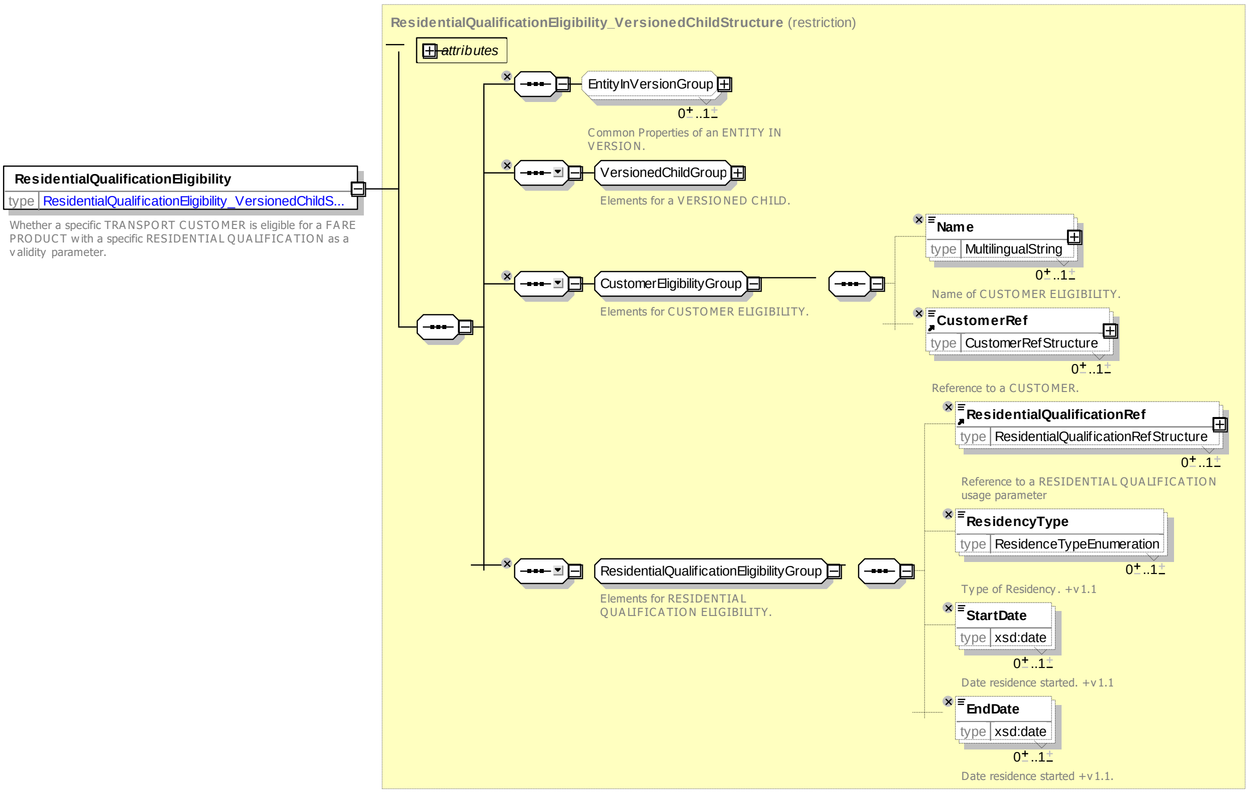


Figure 360 — *ResidentialQualificationEligibility* — XSD

8.1.4 Retail

8.1.4.1 Retail – Conceptual MODEL

The Retail MODEL describes the retail organisations who may sell products. It also allows information about current and blacklisted RETAIL DEVICEs used to sell products to be captured as part of the SALES TRANSACTION.

A RETAIL CONSORTIUM is a legally incorporated ORGANISATION with two or more members. It registers both RETAIL DEVICEs and BLACKLISTs of watched RETAIL DEVICEs, CUSTOMERs and CUSTOMER CONTRACTs.

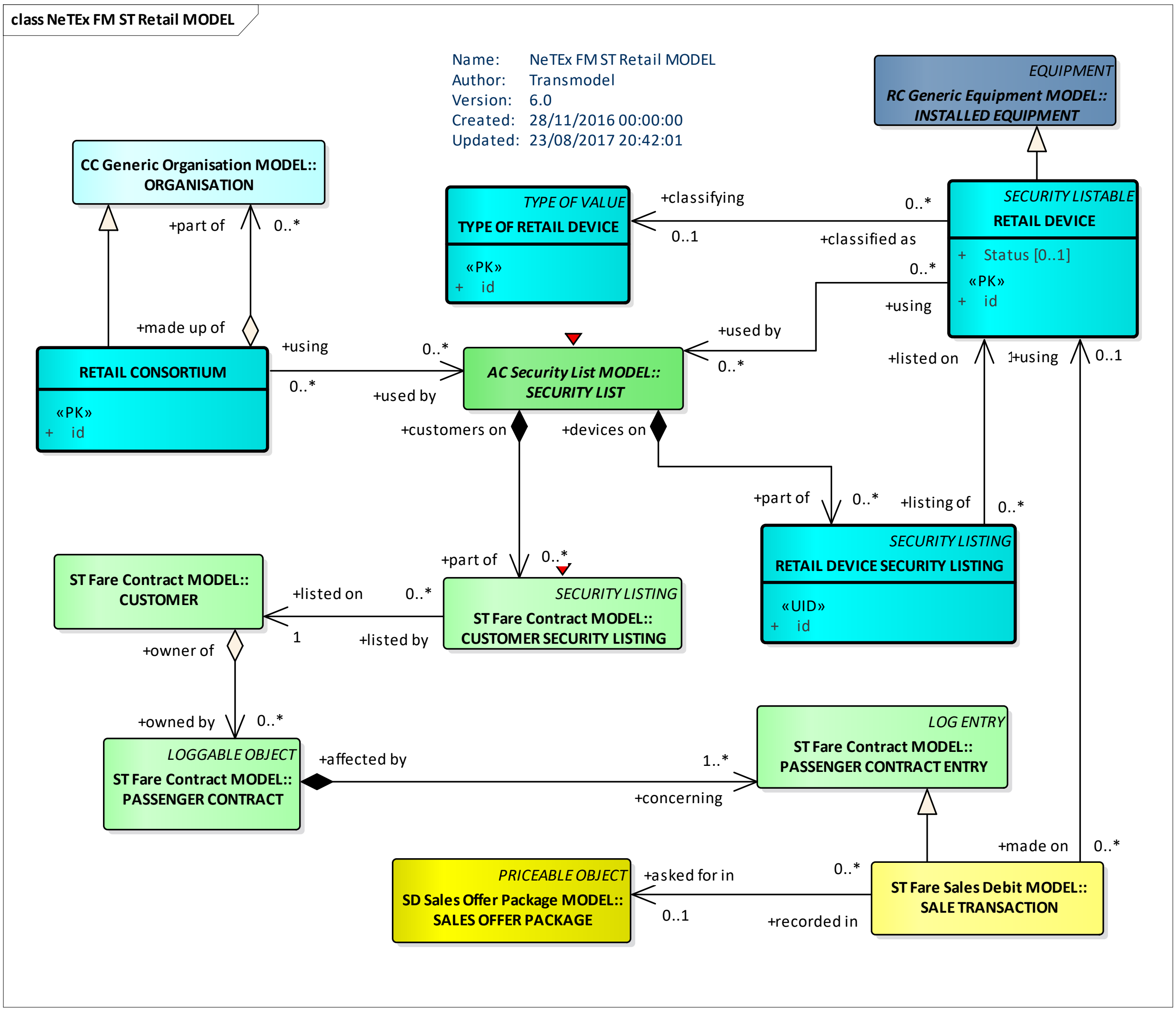


Figure 361 — Retail – Conceptual MODEL (UML)

8.1.4.2 Retail – Physical model

The following figure shows the physical model for RETAIL.

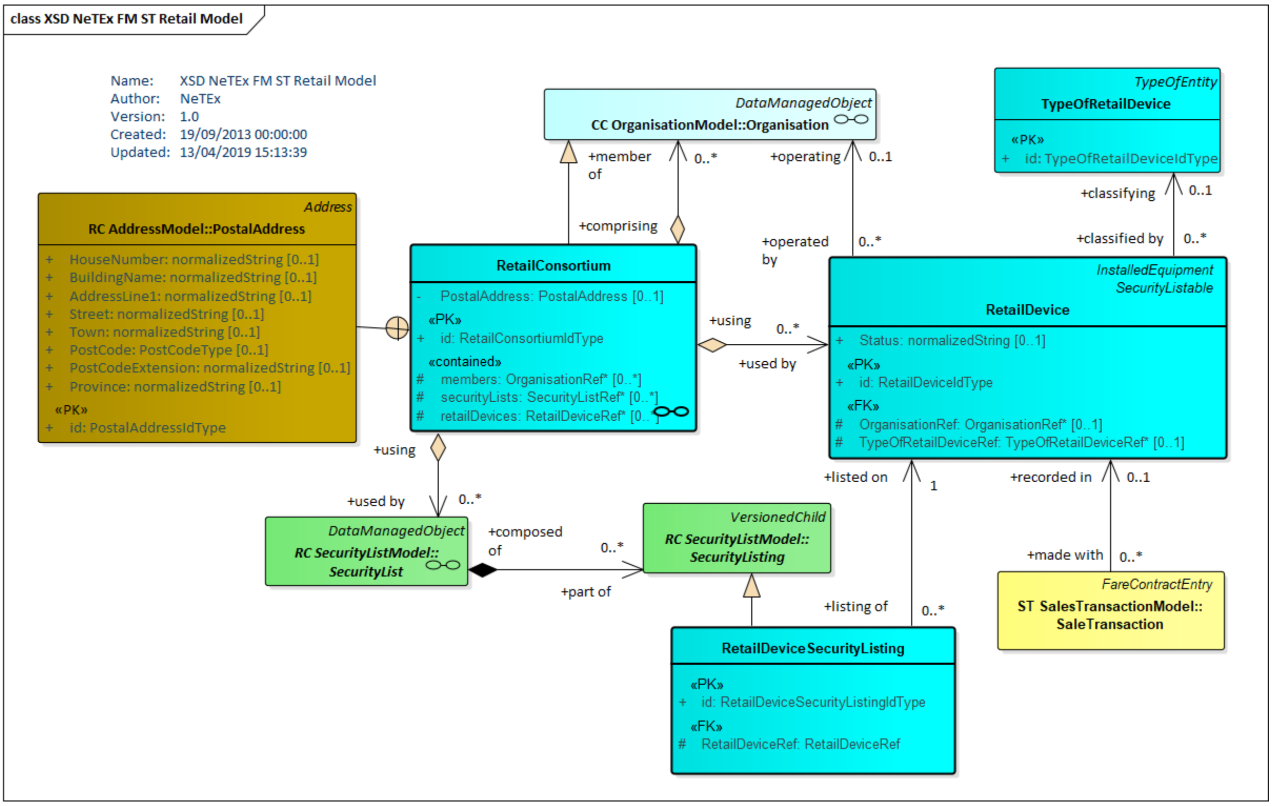


Figure 362 — Retail– Physical Model (UML)

8.1.4.3 Retail – Attributes and XSD

8.1.4.3.1 Retail Consortium – Model Element

A group of ORGANISATIONs, formally incorporated as a RETAIL ORGANISATION, who are retailers of fare products and who share common security processes.

Table 328 – *RetailConsortium* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	<u>*Organisation*</u>	::>	RETAIL CONSORTIUM inherits from ORGANISATION. See NeTEx Part1.
«PK»	***id***	*RetailConsortiumIdType*	1:1	Identifier of a RETAIL CONSORTIUM.
	PostalAddress	<u>*PostalAddress*</u>	0:1	Postal ADDRESS of RETAIL CONSORTIUM.
«cntd»	***members***	*(OrganisationRef)*	0:*	Members of the RETAIL CONSORTIUM.
«cntd»	***blacklists***	*BlackListRef*	0:*	Blacklists shared by the RETAIL CONSORTIUM.
«cntd»	***retailDevices***	*RetailDeviceRef*	0:*	Retail devices registered with the RETAIL CONSORTIUM.

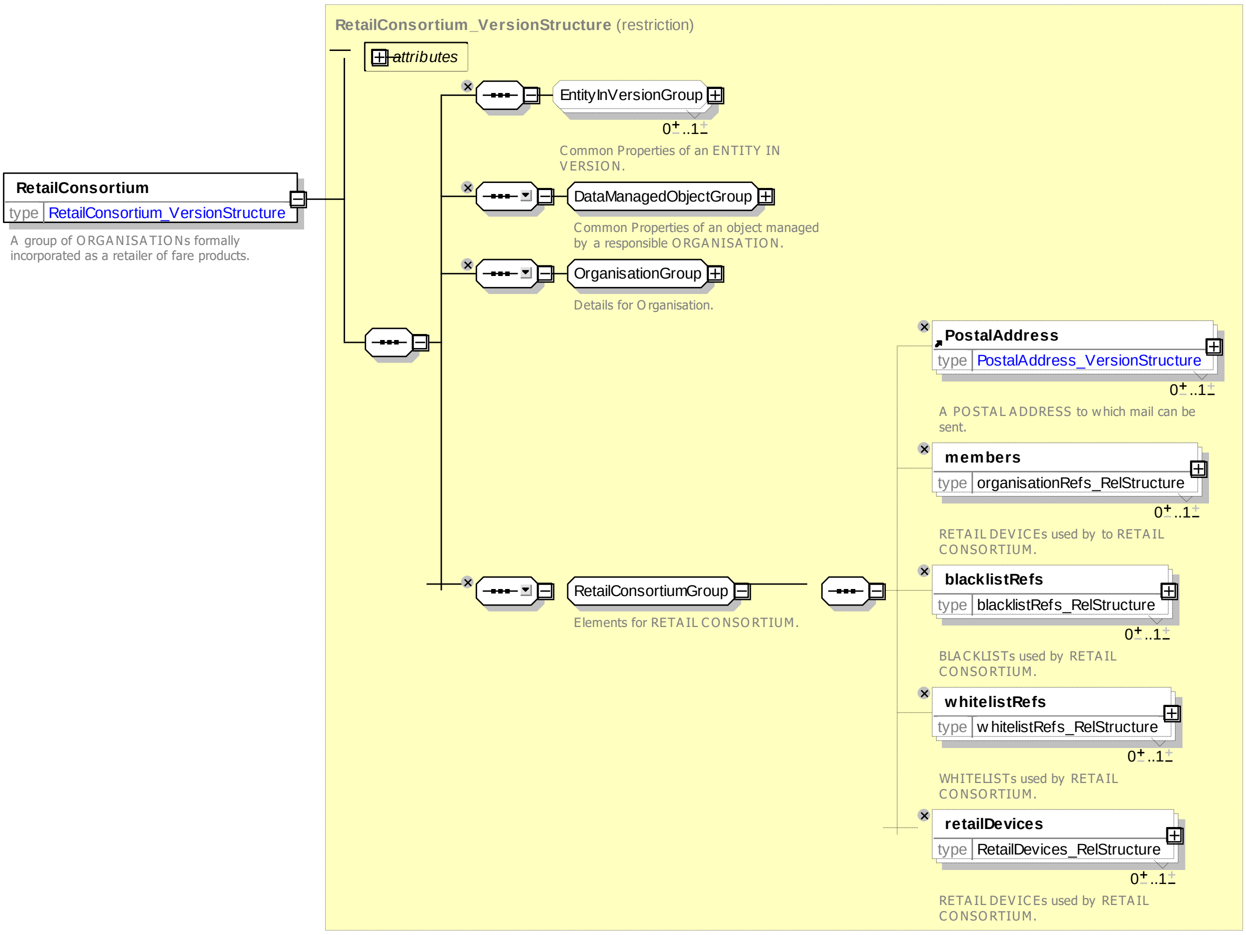


Figure 363 — *RetailConsortium* — XSD

8.1.4.3.2 Retail Device – Model Element

A retail device used to sell fare products. Can be used to record fulfilment.

Table 329 – *RetailDevice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*InstalledEquipment*	::>	***RetailDevice*** inherits from ***InstalledEquipment***
«PK»	***id***	*RetailDeviceIdType*	1:1	Identifier of a RETAIL DEVICE.
	Status	xsd:*normalizedString*	0:1	Status of RETAIL DEVICE.
«FK»	***OrganisationRef***	*(OrganisationRef)*	0:1	Reference to the ORGANISATION operating the equipment.
«FK»	***TypeOfRetail-DeviceRef***	*TypeOfRetailDeviceRef*	0:1	Type of RETAIL DEVICE.

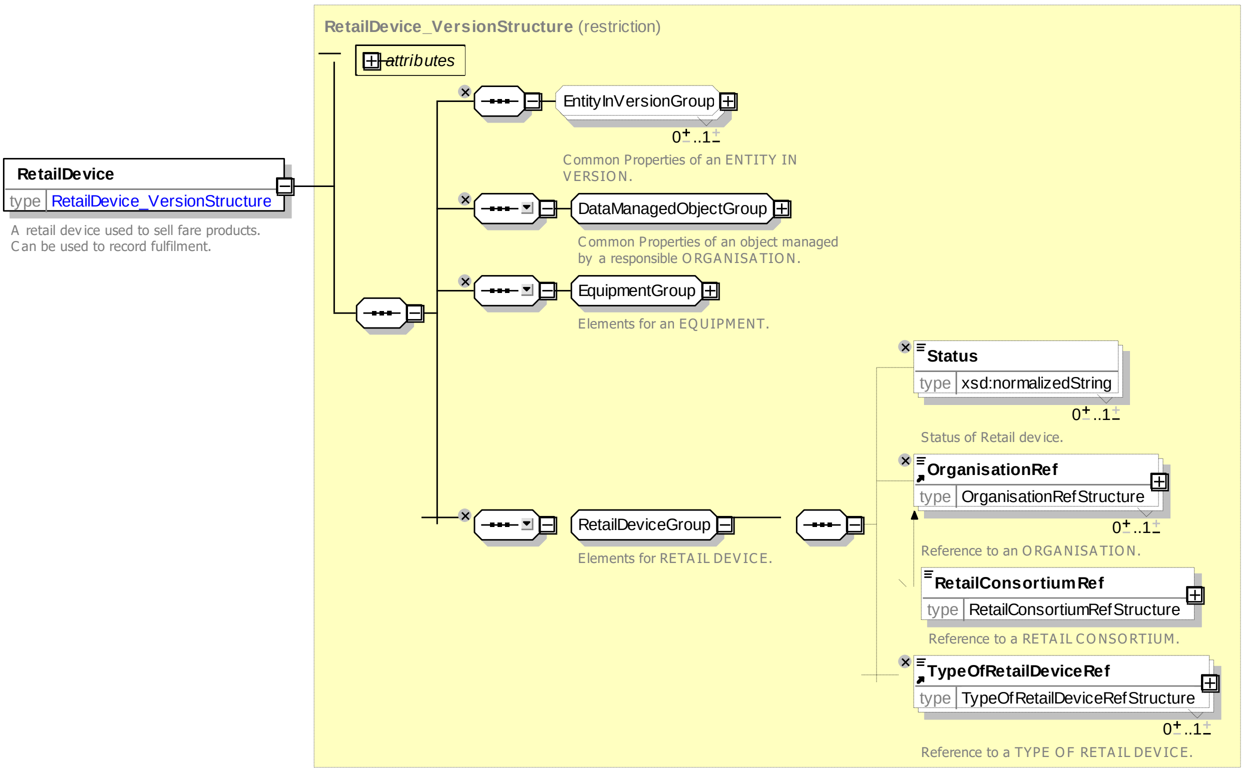


Figure 364 — *RetailDevice* — XSD

8.1.4.3.3 RetailDeviceSecurityListing – Model Element

A listing of a RETAIL DEVICE on a SECURITY LIST.

Table 330 – RetailDeviceSecurityListing – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*SecurityListing*	::>	RETAIL DEVICE SECURITY LISTING inherits from SECURITY LISTING.
«PK»	***id***	*RetailDeviceSecurity-ListingIdType*	1:1	Identifier of RETAIL DEVICE SECURITY LISTING.
«FK»	***RetailDeviceRef***	*RetailDeviceRef*	0:1	Reference to a RETAIL DEVICE.

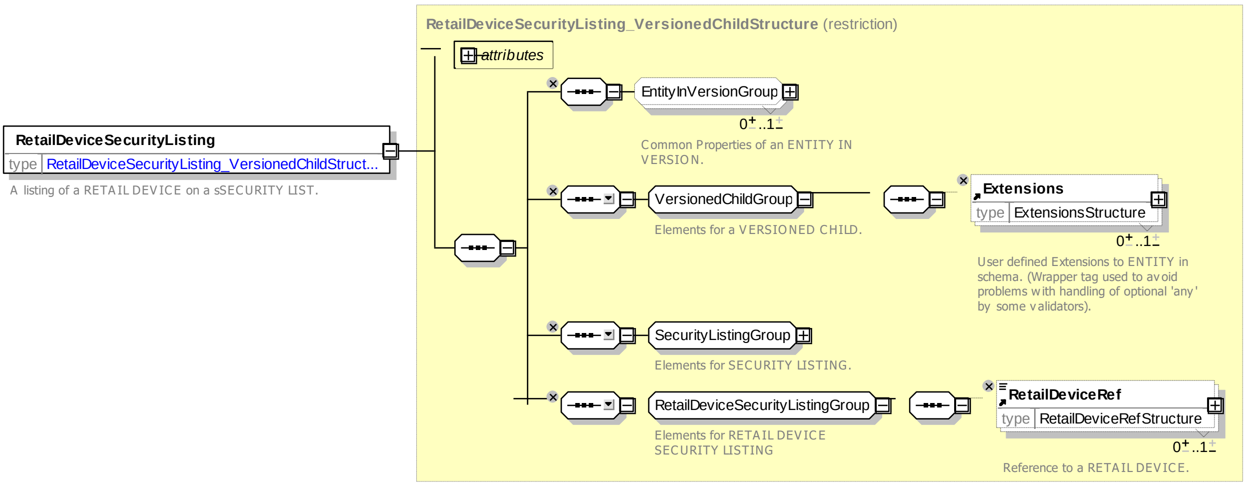


Figure 365 — RetailDeviceSecurityListing — XSD

8.1.4.3.4 **Type of Retail Device –**

A classification of RETAIL DEVICEs.

Table 331 – *TypeOfRetailDevice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	***TypeOfRetailDevice*** inherits from ***TypeOfValue***
«PK»	***id***	*TypeOfRetailDevice-IdType*	1:1	Identifier of TYPE OF RETAIL DEVICE.

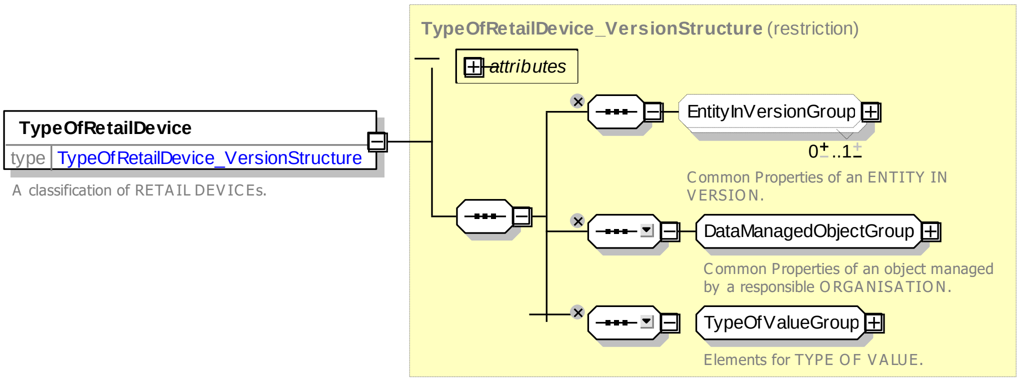


Figure 366 — *TypeOfRetailDevice* — XSD

8.1.4.4 Retail Consortium – XML examples

8.1.4.4.1 **Retail Consortium – XML Example of Consortium**

The following code fragment shows a RETAIL CONSORTIUM with two registered devices.

For EXAMPLE:

```
<retailConsortiums>
    <RetailConsortium version="any" id="tfl:Oyster">
        <Name>Oyster plc</Name>
        <ShortName>Oyster</ShortName>
        <members>
            <OperatorRef version="any" ref="tfl:lbsl"/>
            <OperatorRef version="any" ref="tfl:lul"/>
            <OperatorRef version="any" ref="tfl:dlr"/>
            <OperatorRef version="any" ref="tfl:Emirates"/>
            <OperatorRef version="any" ref="nr:nr"/>
Etc., etc.
        </members>
        <retailDevices>
            <RetailDevice version="any" id="tfl:4421">
                <Name>Ticket machine 2 at KingsX</Name>
            </RetailDevice>
            <RetailDevice version="any" id="tfl:4422">
                <Name>Ticket machine3 at KingsX</Name>
            </RetailDevice>
Etc., etc.
        </retailDevices>
    </RetailConsortium>
</retailConsortiums>
```

8.1.5 Sales Transaction

8.1.5.1 SALES TRANSACTION – Conceptual MODEL

The SALES TRANSACTION model records product sales either to identified CUSTOMERs as entries against a FARE CONTRACT, or as anonymous transactions.

Each SALES TRANSACTION describes the purchase of a SALES OFFER PACKAGE. The actual access rights purchased and any limitations on them are described by a TRAVEL SPECIFICATION. Thus, for example a FARE PRODUCT might be available for different stops and for different classes of user. The TRAVEL SPECIFICATION would record which specific stops (e.g. as a reference to a DISTANCE MATRIX ELEMENT) and what is the actual CLASS OF USE, and identify the type of user, for example as a USER PROFILE reference.

A SALES TRANSACTION may result in the creation of a CUSTOMER PURCHASE PACKAGE, a structured representation of the specific purchase, with components that may be controlled and validated.

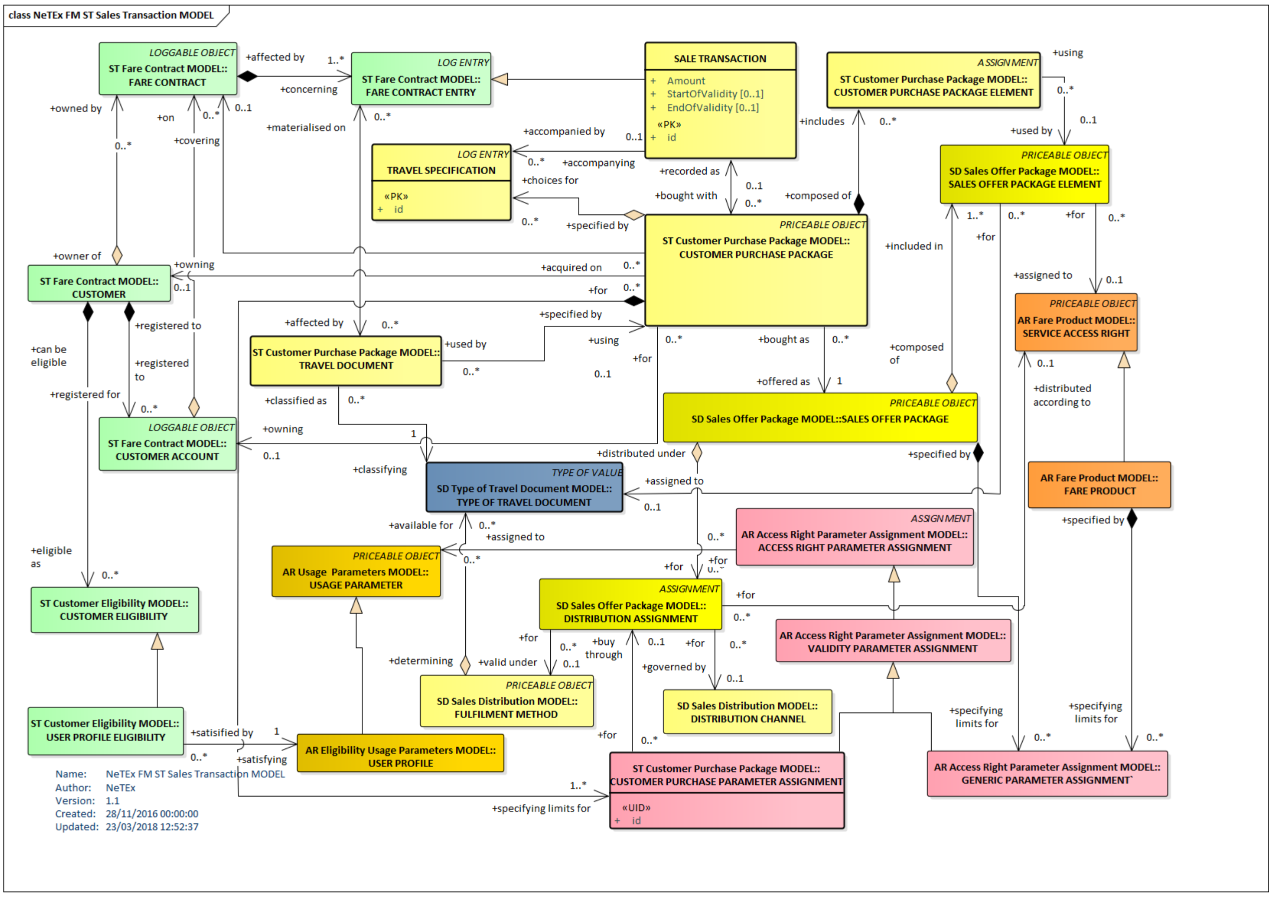


Figure 367 — Sales Transaction – Conceptual MODEL (UML)

8.1.5.2 Sales Transaction – Physical model

The following figure shows the physical model for SALES TRANSACTIONs.

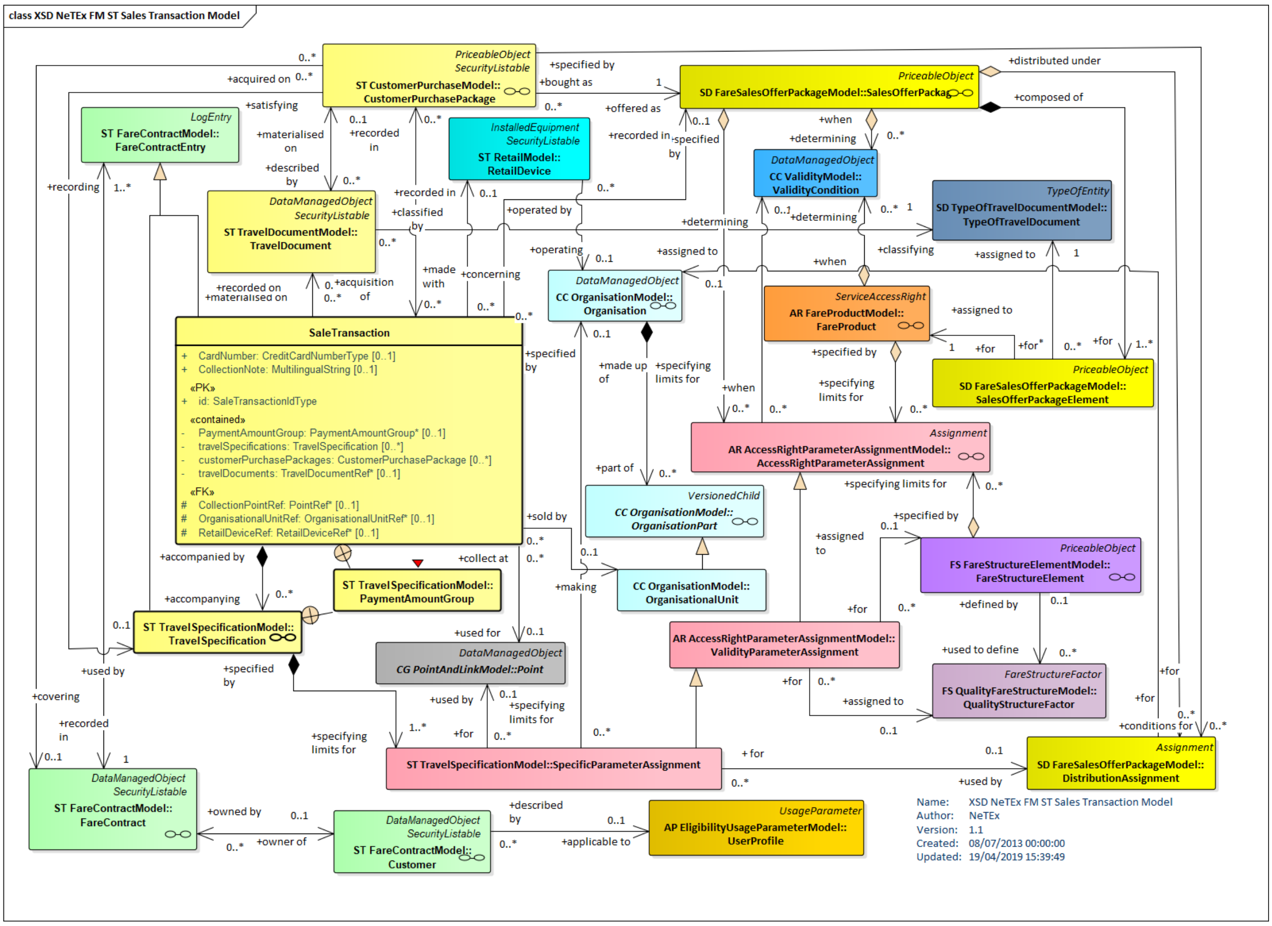


Figure 368 — Sales Transaction – Physical Model (UML)

8.1.5.3 Sales Transaction Payment Amount – Physical model

Both a SALES TRANSACTION and a TRAVEL SPECIFICATION can hold a price as a currency amount, and, or in arbitrary pricing units. in addition, a derivation of the price from a base price can also be given. The derivation is made up of one or more RULE STEP RESULTs, each showing the result of applying a PRICING RULE to an input price and using any pricing parameters such as ROUNDING and ROUNDING STEP.

The PAYMENT METHOD can also be indicated.

The following figure shows the physical model for a payment amount.

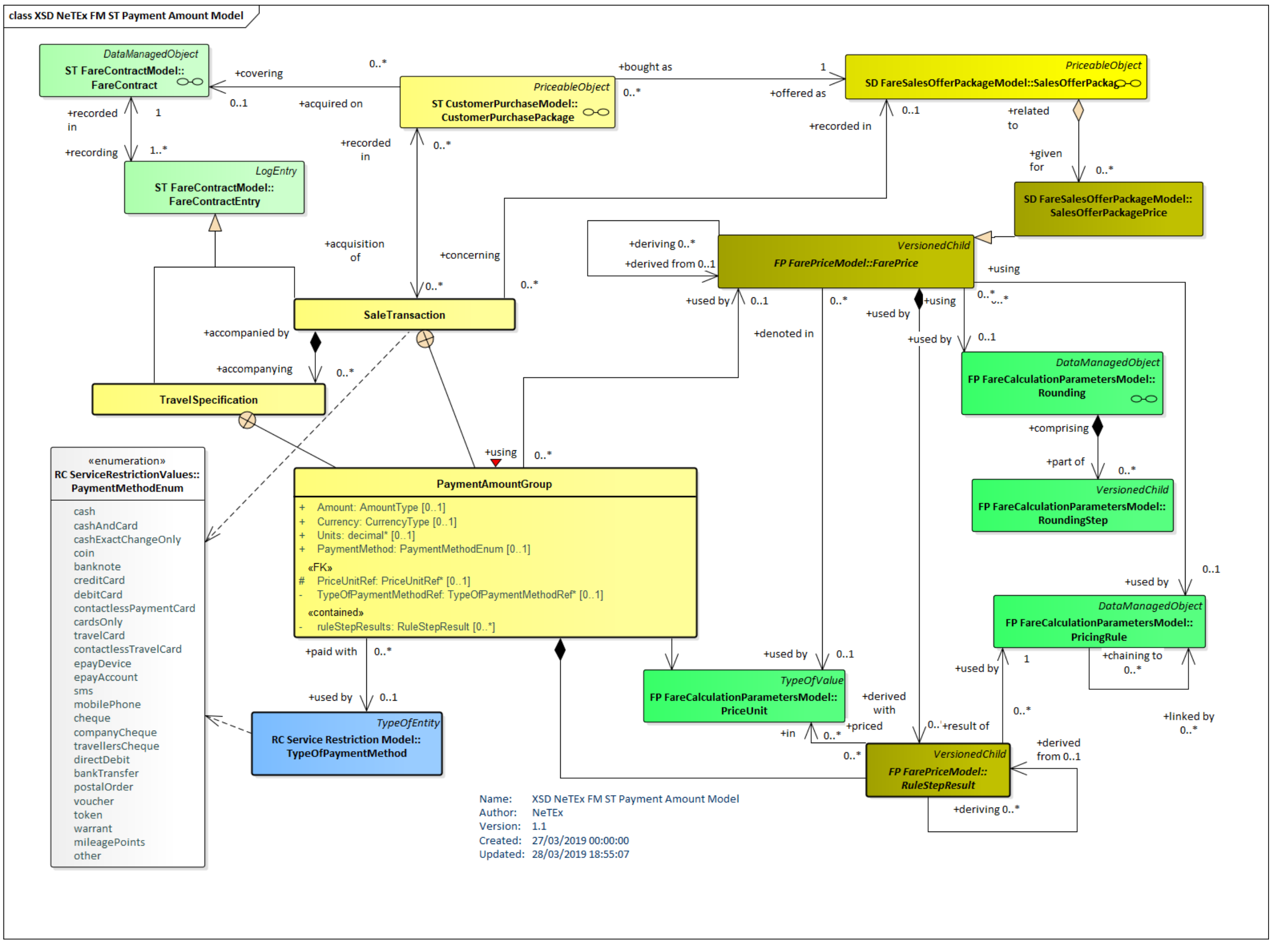


Figure 369 — Sales Transaction Payment Amount– Physical Model (UML)

8.1.5.4 Sales Transaction – Attributes and XSD

8.1.5.4.1 SalesTransaction – Model Element

A SALE OF a FIXED PACKAGE or a SALE OF a RELOADABLE PACKAGE.

Table 332 – *SalesTransaction* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*FareContractEntry*	::>	SALES TRANSACTION inherits from FARE CONTRACT ENTRY.
«PK»	***id***	*SaleTransactionIdType*	1:1	Identifier of SALES TRANSACTION.
XGRP	***PaymentAmount-Group***	**xmlGroup**	0:1	Payment amount elements for SALES TRANSACTION.
	CardNumber	*CreditCardNumberType*	0:1	Card number of card used to pay for transaction.
«cntd»	***travel-Specifications***	*TravelSpecification*	0:*	TRAVEL SPECIFICATIONs for SALES TRANSACTION.

«cntd»	***customer-Purchase-Packages***	*CustomerPurchasePackage Ref*	0:*	CUSTOMER PURCHASE PACKAGEs bought by SALES TRANSACTION.
«cntd»	***travelDocuments***	*TravelDocumentRef*	0:*	Reference to a TRAVEL DOCUMENT, i.e. ticket.
«FK»	***Collection-PointRef***	*PointRef*	0:1	Point at which TRAVEL DOCUMENT will be collected.
	CollectionNote	*MultilingualString*	0:1	Information on where TRAVEL DOCUMENT is to be collected.
«FK»	***Organisational-UnitRef***	*OrganisationalUnitRef*	0:1	ORGANISATION UNIT making sale.
«FK»	***RetailDeviceRef***	*RetailDeviceRef*	0:1	RETAIL DEVICE used to make SALES TRANSACTION.

SalesTransaction

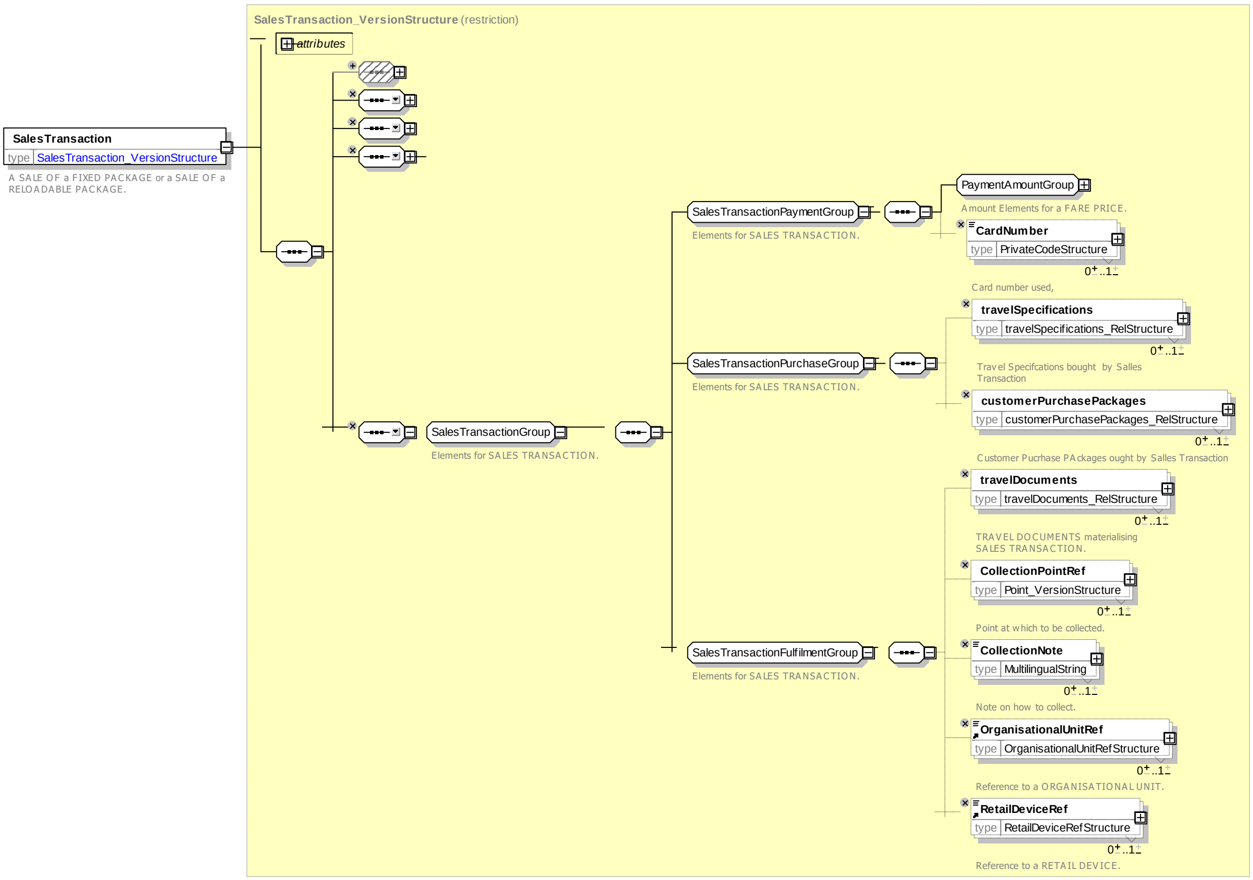


Figure 370 — *SaleTransaction* — XSD

8.1.5.4.1.1 PaymentAmountGroup – Group

Payment amount elements for a SALES TRANSACTION.

Table 333 – *SalesTransactionPaymentGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
	Amount	*AmountType*	0:1	Amount of SALES TRANSACTION.
	Currency	*CurrencyType*	0:1	Currency of SALES TRANSACTION.
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	PRICE UNIT of SALES TRANSACTION.
	Units	*xsd:integer*	0:1	Number of units.
«cntd»	***ruleStepResults***	*RuleStepResult*	0:*	RULE STEP RESULTS FOR calculation steps used to compute amount.
«enum»	***PaymentMethod***	*PaymentMethodEnum*	0:1	Payment Method used to pay for transaction. See Part1.
«FK»	***TypeOfPayment-Method***	*TypeOfPaymentMethod*	0:1	TYPE OF PAYMENT used to pay for transaction.

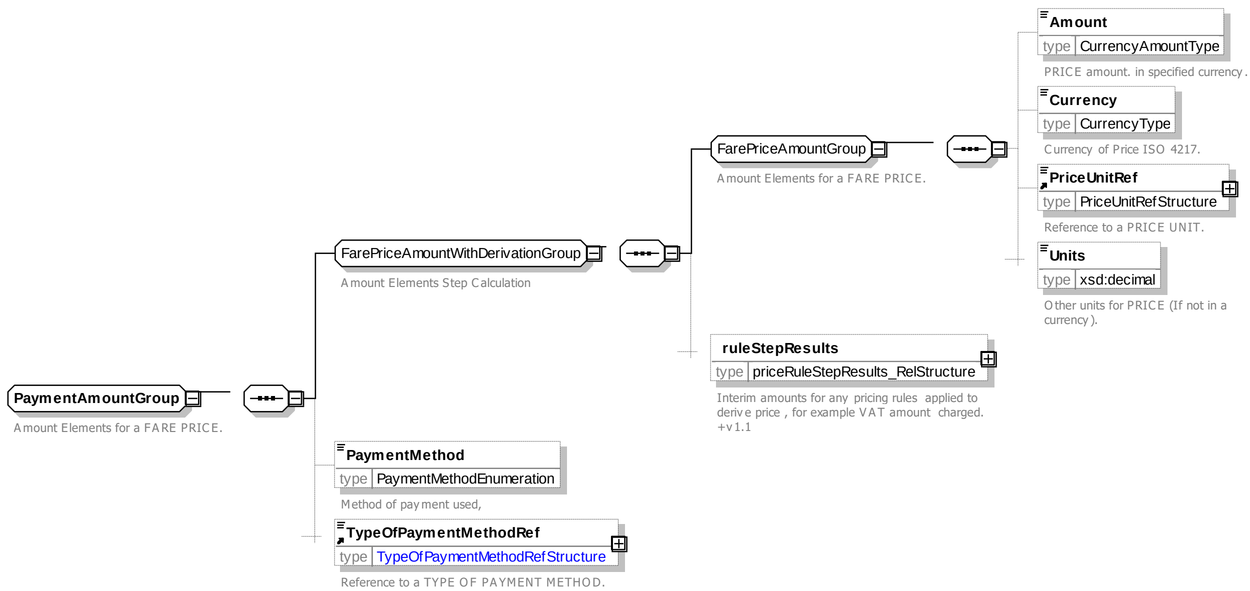


Figure 371 —*PaymentAmountGroup* — XSD

8.1.5.4.2 SpecificParameterAssignment – Model Element

A VALIDITY PARAMETER ASSIGNMENT specifying practical parameters during a TRAVEL SPECIFICATION, within a given fare structure (e.g. the origin or destination zone in a zone-counting system).

Table 334 – *SpecificParameterAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*ValidityParameter-Assignment*	::>	SPECIFIC PARAMETER ASSIGNMENT inherits from VALIDITY PARAMETER ASSIGNMENT.

«PK»	***id***	*SpecificParameter-AssignmentIdType*	1:1	Identifier of SPECIFIC PARAMETER ASSIGNMENT.
	AccessNumber	*xsd:positiveInteger*	0:1	Access number for this assignment.
«enum»	***Includes-GroupingType***	*BooleanOperatorEnum AND | OR |NOT XOR*	0:1	Operator for Grouping scope elements. See ACCESS RIGHT ASSIGNMENT for allowed values.
«cntd»	***includes***	*SpecificParameter-AssignmentRef | SpecificParameter-Assignment*	0:*	SPECIFIC PARAMETER ASSIGNMENTs logically included in this group. Groups are combined according to the Operator.
«FK»	***Distribution-AssignmentRef***	*Distribution-AssignmentRef*	0:1	DISTRIBUTION ASSIGNMENT parameters used in SPECIFIC PARAMETER ASSIGNMENT.
«FK»	***Retailing-OrganisationRef Ref***	*RetailingOrganisationRef*	0:1	RETAILING ORGANISATION used in SPECIFIC PARAMETER ASSIGNMENT.
«FK»	***Collection-PointRef***	*(PointRef)*	0:1	COLLECTION POINT used in SPECIFIC PARAMETER ASSIGNMENT.

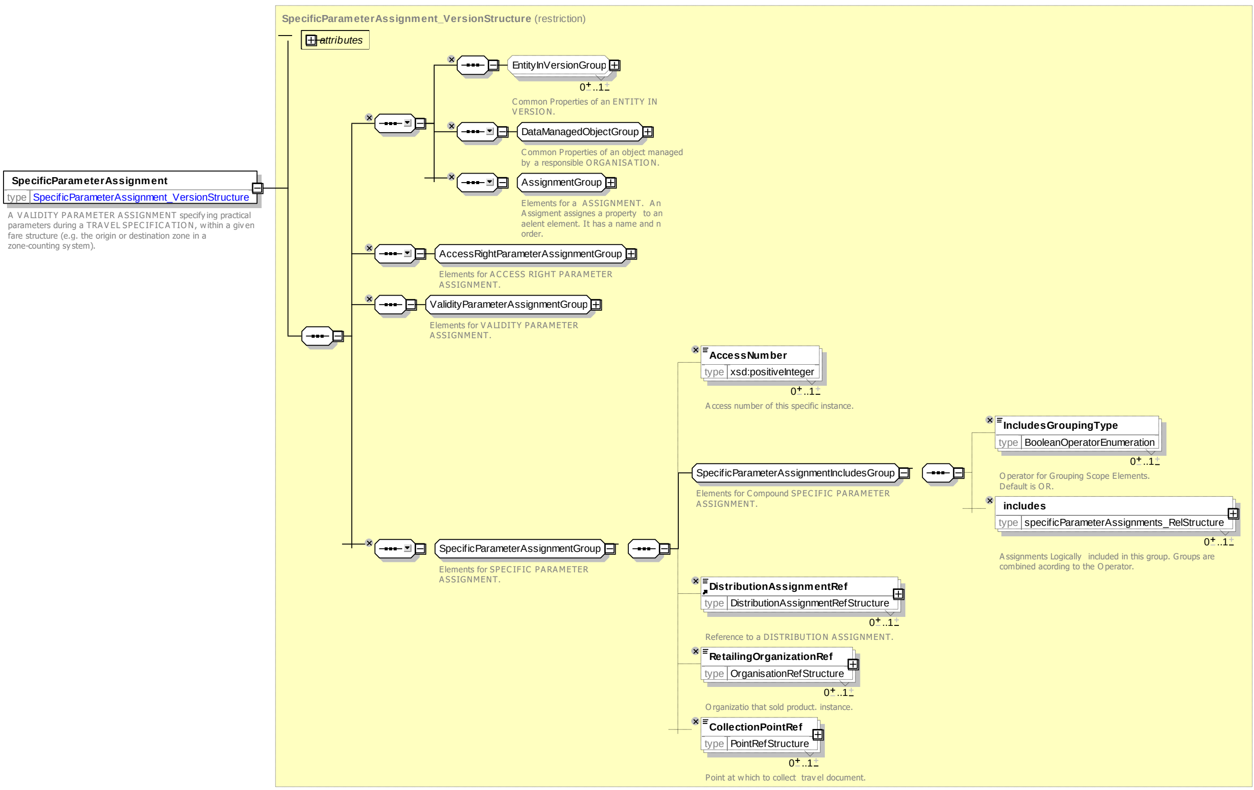


Figure 372 — *SpecificParameterAssignment* — XSD

8.1.5.5 Sales Transaction – XML examples

8.1.5.5.1 Sales Transaction: XML Example of Simple paper ticket sale

The following code fragment records the purchase of a ticket to travel in Zone 1 only on the metro as an anonymous SALE TRANSACTION & TRAVEL SPECIFICATION, grouped using a PASSENGER CONTRACT.

For EXAMPLE:

```
<FareContract version="any" id="xpl:Anon001">
    <Name> Zone 1 Paper ticket for Cash </Name>
    <fareContractEntries>
        <SalesTransaction version="any" id="xpl:Anon001@trans001@purchaseTicket@Ticket">
            <Name>Buy Single Cash ticket for zone 1 </Name>
            <Description> £ 4.50 cash </Description>
            <Date>2013-07-08T01:07:00</Date>
            <Amount>4.50</Amount>
        </SalesTransaction>

        <TravelSpecification version="any" id="xpl:Anon001@trans001@purchaseTicket@Ticket">
            <Name>Single fare zone 1 </Name>
            <Date>2013-07-08T01:07:00</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:purchase"/>
            <Amount>4.50</Amount>
            <StartOfValidity>2013-07-08T01:07:00</StartOfValidity>
            <SalesOfferPackageRef version="any" ref="tfl:Ticket"/>
            <SalesTransactionRef version="any" ref="xpl:Anon001@trans001@purchaseTicket@Ticket"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl:Anon001@trans001@01">
                    <ValidableElementRef version="any" ref="lul:metroTrip"/>
                </SpecificParameterAssignment>

                <SpecificParameterAssignment version="any" id="xpl:Anon001@trans001@01">
                    <Name>Fare zones 1 only</Name>
                    <Description of purchase?</Description>
                    <FareStructureElementRef version="any" ref="tfl:Zone_1_only"/>
                    <FareDemandFactorRef version="any" ref="tfl:anyTime"/>
                    <limitations>
                        <FrequencyOfUseRef version="any" ref="tfl:one_trip"/>
                        <UserProfileRef version="any" ref="tfl:adult"/>
                    </limitations>
                    <ValidityParameters>
                        <ScheduledStopPointRef ref="tfl:Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any" ref="tfl:SelfServiceMachine"/>
                    </ValidityParameters>
                </SpecificParameterAssignment>
                <SpecificParameterAssignment version="any" id="xpl:Anon001@trans001@02">
                    <Name>Metro trip can be made up of multiple steps</Name>
                    <FareStructureElementRef version="any" ref="tfl:metroTrip"/>
                    <limitations>
                        <RoundTripRef version="any" ref="tfl:single"/>
                    </limitations>
```

```
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>
    </fareContractEntries>
</FareContract>
```

8.1.5.5.2 **Sales Transaction: XML Example of Group ticket sale**

The following code fragment shows an anonymous sale of a GROUP TICKET to travel in Zones 1-6 for one adult and 10 children on the metro.

For EXAMPLE:

```
<FareContract version="any" id="xpl:Anon002">
    <Name> Day Group Paper Ticket </Name>
    <fareContractEntrys>
        <SalesTransaction version="any" id="xpl:Anon002@trans001@purchaseTicket@GroupDayTicket">
            <Name>Buy Group Day ticket for zone 1-6 4 Adults 10 Children </Name>
            <Description> 4 @£ 4.50 each + 10@ £ 1.70 each </Description>
            <Date>2013-07-08T01:07:15</Date>
            <Amount>28.70</Amount>
        </SalesTransaction>

        <TravelSpecification version="any"
                id="xpl:Anon002@trans001@purchaseTicket@GroupDayTicket@Adult">
            <Name>Adults on Day ticket </Name>
            <Date>2013-07-08T01:09:15</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:purchase"/>
            <Amount>4.50</Amount>
            <Units>4</Units>
            <SalesOfferPackageRef version="any" ref="tfl:GroupDayTicket"/>
            <SalesTransactionRef version="any"
                ref="xpl:Anon002@trans001@purchaseTicket@GroupDayTicket"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any"
                    id="xpl:Anon002@trans001@purchaseTicket@GroupDayTicket@01">
                    <Name>Fare zones 1-6</Name>
                    <limitations>
                        <GroupTicketRef version="any" ref="tfl:groupDayTicket"/>
                        <UserProfileRef version="any" ref="tfl:adult"/>
                    </limitations>
                    <ValidityParameters>
                        <ScheduledStopPointRef ref="tfl:Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any"
                        ref="tfl:TubeStationTicketOffices"/>
                    </ValidityParameters>
                    <FareStructureElementRef version="any" ref="tfl:Zones_1-6"/>
                    <FareDemandFactorRef version="any" ref="tfl:any_time"/>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

        <TravelSpecification version="any"
                id="xpl:Anon002@trans001@purchaseTicket@GroupDayTicket@Child">
            <Name>Children on Day ticket </Name>
            <Date>2013-07-08T01:07:15</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:purchase"/>
            <Amount>1.70</Amount>
            <Units>10</Units>
            <SalesOfferPackageRef version="any" ref="tfl:GroupDayTicket"/>
            <SalesTransactionRef version="any"
                ref="xpl:Anon002@trans001@purchaseTicket@GroupDayTicket"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl:Anon002@trans001@ 01">
                    <Name>Fare zones 1-6</Name>
                    <limitations>
                        <GroupTicketRef version="any" ref="tfl:groupDayTicket"/>
                        <UserProfileRef version="any" ref="tfl:child"/>
                    </limitations>
                    <ValidityParameters>
                        <ScheduledStopPointRef ref="tfl:Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any"
                            ref="tfl:TubeStationTicketOffices"/>
```

```
                    </ValidityParameters>
                    <FareStructureElementRef version="any" ref="tfl:Zones_1-6"/>
                    <FareDemandFactorRef version="any" ref="tfl:any_time"/>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>
    </fareContractEntriess>
</FareContract>
```

8.1.5.5.3 Sales Transaction: XML Example of Card Transactions

The following code fragment shows a series of FARE CONTRACT ENTRies on an anonymous Oyster card (i.e. FARE CONTRACT) including (a) Card purchase (b) a seven day pass for zone 1 (c) Adding a PAY as YOU go credit for use in other zones and making a trip within the allowed zone.

For EXAMPLE:

```
<!-- Example Use of an Oyster Card -->
<FareContract version="any" id="xpl:Oyster-12345">
    <Name> Anonymous Oyster card Oyster-12345 with Transactions </Name>
    <fareContractEntries>
        <SalesTransaction version="any" id="xpl:Oyster-12345@trans001@purchase_new_card">
            <Name>Buy an Travel Card Oyster Card with 7 days for zones 1 and 2</Name>
            <Description> Card 5.00 deposit + Pass @ £30.40 </Description>
            <Date>2013-07-08T01:07:00</Date>
            <Amount>35.40</Amount>
        </SalesTransaction>

        <TravelSpecification version="any"
                id="xpl:Oyster-12345@trans001@purchase_new_cardAdultOysterCard">
            <Name>Adult Oyster Card</Name>
            <Date>2013-07-08T01:07:00</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:purchase"/>
            <Amount>5.00</Amount>
            <StartOfValidity>2013-07-08T01:07:00</StartOfValidity>
            <SalesOfferPackageRef version="any" ref="tfl:AdultOysterCard"/>
            <SalesTransactionRef version="any" ref="xpl:Oyster-12345@trans001@purchase_new_card"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl:Oyster-12345@trans001@01">
                    <limitations>
                        <UserProfileRef version="any" ref="tfl:adult"/>
                    </limitations>
                    <ValidityParameters>
                        <DistributionChannelRef version="any"
                            ref="tfl:Tube_Station_ticket_offices"/>
                    </ValidityParameters>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

        <TravelSpecification version="any" id="xpl:Oyster-
12345@trans001@purchase_new_card@TravelCard_on_Oyster">
            <Name>TravelCard_on_Oyster 7 Day Travel card for zones 1 and 2 </Name>
            <Date>2013-07-08T01:07:00</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:purchase"/>
            <Amount>30.40</Amount>
            <SalesOfferPackageRef version="any" ref="tfl:TravelCard_on_Oyster"/>
            <SalesTransactionRef version="any" ref="xpl:Oyster-12345@trans001@purchase_new_card"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any"
                id="xpl:Oyster-12345@trans001@Zones_1-2">
                    <FareStructureElementRef version="any" ref="tfl:Zones_1-2"/>
                    <limitations>
                        <UserProfileRef version="any" ref="tfl:adult"/>
                    </limitations>
                </SpecificParameterAssignment>
                <SpecificParameterAssignment version="any" id="xpl:Oyster-12345@trans001@@1Week">
                    <Name> 1 week pass </Name>
                    <FareStructureElementRef version="any" ref="tfl:1_Week_Pass"/>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

        <!--    Prepay £10 O N an Travel Card Oyster Card with -->
```

```xml
        <SalesTransaction version="any" id="xpl:Oyster-12345@trans002@Oyster_top_up">
            <Name>Top up Oyster card with £10 </Name>
            <Description> £10 </Description>
            <Date>2013-07-08T01:07:10</Date>
            <Amount>10.00</Amount>
        </SalesTransaction>
        <TravelSpecification version="any" id="xpl:Oyster-
12345@trans002@Oyster_top_up@Oyster_top_up">
            <Name>Add 10 Pay as you go</Name>
            <Date>2013-07-08T01:07:10</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:purchase"/>
            <Amount>10</Amount>
            <SalesOfferPackageRef version="any" ref="tfl:Oyster_top_up"/>
            <SalesTransactionRef version="any" ref="xpl:Oyster-12345@trans002@Oyster_top_up"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl:Oyster-12345@trans002@01">
                    <Name>Top up ay at Kings Cross </Name>
                    <ValidityParameters>
                        <DistributionChannelRef version="any" ref="tfl:Self_Service_machine"/>
                    </ValidityParameters>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

        <!--    Start a peak time trip inside travel zone -->
        <SalesTransaction version="any" id="xpl:Oyster-12345@trans003@touchIn">
            <Name>Oyster Touch in in a zone inside travel card </Name>
            <Description> Raises liability for max far £.8.40 ( £2.40) </Description>
            <Date>2013-07-08T01:09:00</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:yellowTouchIn"/>
            <Amount>8.40</Amount>
        </SalesTransaction>
        <TravelSpecification version="any" id="xpl:Oyster-12345@trans003@touchIn@touchIn">
            <Name>Add Max fare </Name>
            <Date>2013-07-08T01:08:50</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:yellow_touch_in"/>
            <Amount>8.40</Amount>
            <SalesOfferPackageRef version="any" ref="tfl:PayAsYouGo_fare"/>
            <SalesTransactionRef version="any" ref="xpl:Oyster-12345@trans003@touchIn"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl:Oyster-12345@trans003@01">
                    <Name>Touch in at Kings Cross </Name>
                    <FareStructureElementRef version="any" ref="tfl:Zone_1_only"/>
                    <FareDemandFactorRef version="any" ref="tfl:peak"/>
                    <limitations>
                        <UserProfileRef version="any" ref="tfl:adult"/>
                    </limitations>
                    <ValidityParameters>
                        <ScheduledStopPointRef ref="tfl:Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any" ref="tfl:Validator"/>
                    </ValidityParameters>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>
        <!--    End a peak time trip inside travel zone covered by card -->
        <SalesTransaction version="any" id="xpl:Oyster-12345@trans004@purchase">
            <Name>Oyster Touch out in a zone inside travel card </Name>
            <Description>Raises liability for max incremental fare zones 2-8 £.8.40 ( £2.40)
</Description>
            <Date>2013-07-08T01:09:10</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:yellow_touch_In"/>
            <Amount>-8.40</Amount>
        </SalesTransaction>

        <TravelSpecification version="any"
                 id="xpl:Oyster-12345@trans004@purchase@yellowTouchOut">
            <Name>Touch out to complete Fare within zone </Name>
            <Date>2013-07-08T01:09:10</Date>
            <TypeOfFareContractEntryRef version="any" ref="tfl:yellow_touch_Out"/>
            <Amount>0.00</Amount>
            <SalesOfferPackageRef version="any" ref="tfl:PayAsYouGo_fare"/>
            <SalesTransactionRef version="any" ref="xpl:Oyster-12345@trans004@purchase"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl:Oyster-12345@trans004@01">
                    <Name>Touch out at Oxford Circus. </Name>
                    <Description>Counts as Peak because journey started before 9.00</Description>
```

```xml
                        <FareStructureElementRef version="any" ref="tfl:Zone_1_only"/>
                    <limitations>
                        <UserProfileRef version="any" ref="tfl:adult"/>
                    </limitations>
                    <FareDemandFactorRef version="any" ref="tfl:peak"/>
                </SpecificParameterAssignment>
                <SpecificParameterAssignment version="any" id="xpl:Oyster-12345@trans004@02
                    <FareStructureElementRef version="any" ref="tfl:metro_trip"/>
                    <Name>Touch out at Oxford Circus. </Name>
                    <limitations>
                        <RoundTripRef version="any" ref="tfl:single"/>
                    </limitations>
                    <ValidityParameters>
                        <ScheduledStopPointRef ref="tfl:Oxford_Circus" version="any"/>
                    </ValidityParameters>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

    </fareContractEntries>
</FareContract>
</salesContracts>
```

8.1.6 Travel Specification

8.1.6.1 Pre-purchase Specification

OPERATORs or authorities using a graduated (e.g. distance- or zone-based) fare structure will ask the customer to specify (e.g. on the occasion of purchasing) details of the intended consumption (e.g. a travel origin and destination).

The parameters to be used by a customer when purchasing (or using) a generic access right are described by the TRAVEL SPECIFICATION entity. A TRAVEL SPECIFICATION is a specialisation of FARE CONTRACT ENTRY, which implies the following properties:

— a TRAVEL SPECIFICATION is associated with an individual FARE CONTRACT;
— a TRAVEL SPECIFICATION is timed by the 'date' and 'time' attributes of a FARE CONTRACT ENTRY;
— a TRAVEL SPECIFICATION may concern an identified CUSTOMER, through the relationship between FARE CONTRACT ENTRY and CUSTOMER;
— the parameters assigned by a TRAVEL SPECIFICATION are, in most cases, used on an individual TRAVEL DOCUMENT, through the relationship between FARE CONTRACT ENTRY and TRAVEL DOCUMENT.

TRAVEL SPECIFICATION is itself specialised into two subtypes; a REQUESTED TRAVEL SPECIFICATION used to describe the customer's requested choices and an OFFERED TRAVEL SPECIFICATION, representing the set of options actually offered for purchase. often these will be the same, but sometimes they may differ; for example the user may be offered a different class of use, a different route or even different origin and destination stops.

A TRAVEL SPECIFICATION shall include at least one parameter assignment. Such an assignment is described by a SPECIFIC PARAMETER ASSIGNMENT, which is a specialisation of VALIDITY PARAMETER ASSIGNMENT which is, in turn, a specialisation of ACCESS RIGHT PARAMETER ASSIGNMENT.

8.1.6.2 Purchase Specification

A TRAVEL SPECIFICATION often accompanies a SALE TRANSACTION, occurring at the same time or soon after.

— A persistent representation of the customer's purchase can be made with a CUSTOMER PURCHASE PACKAGE; this retains a representation of the specific choices as TRAVEL SPECIFICATIONs with SPECIFIC PARAMETER AS

A SALE TRANSACTION records the purchase by a customer of a given SALES OFFER PACKAGE, composed of many possible generic access rights (e.g. *Week or month passes for a multi zone system*). The actual choice of specific parameters (e.g. *a week pass for zones 1 and 2*) is recorded via SPECIFIC PARAMETER ASSIGNMENTs in a CUSTOMER PURCHASE PACKAGE associated with the concerned FARE CONTRACT (i.e. the contract owning the concerned package), referencing the appropriate generic access right (e.g. the FARE STRUCTURE ELEMENT used for applying the zone counting rule).

The assignments described by the subtypes of VALIDITY PARAMETER ASSIGNMENT are therefore of three different types:

- A GENERIC PARAMETER ASSIGNMENT assigns a parameter in a fixed way to a certain generic access right. For instance, if it attaches a FARE DAY TYPE limiting the use of a FARE STRUCTURE ELEMENT, any FARE PRODUCT sold comprising this element shall comply with this limitation; GENERIC PARAMETER ASSIGNMENTS are used to define the available options of a FARE PRODUCT and SALES OFFER PACKAGE.
- A SPECIFIC PARAMETER ASSIGNMENT assigns a parameter to a CUSTOMER PURCHASE PACKAGE offered to the customer, representing the selection of a SALES OFFER PACKAGE eith a particular combination of parameters selected from the generic access rights included in the package (e.g. the customer intends to consume a generic FARE STRUCTURE ELEMENT starting from the specified origin zone).
- A CUSTOMER PURCHASE PARAMETER ASSIGNMENT assigns the a parameter to a CUSTOMER PURCHASE PACKAGE purchased by a customer, based on one of the generic access rights included in the package (e.g. the customer has bought the right to consume a generic FARE STRUCTURE ELEMENT starting from the specified origin zone).

A TRAVEL SPECIFICATION may be used to record the parameters used to request a product using a FARE QUERY. A suggested model for trip and fare queries is available as an informative extension to the NeTEx Part3 UML model, but is not described in the standard document.

8.1.6.3 Travel Specification – Physical model

The following figure shows the physical model for a TRAVEL SPECIFICATION.

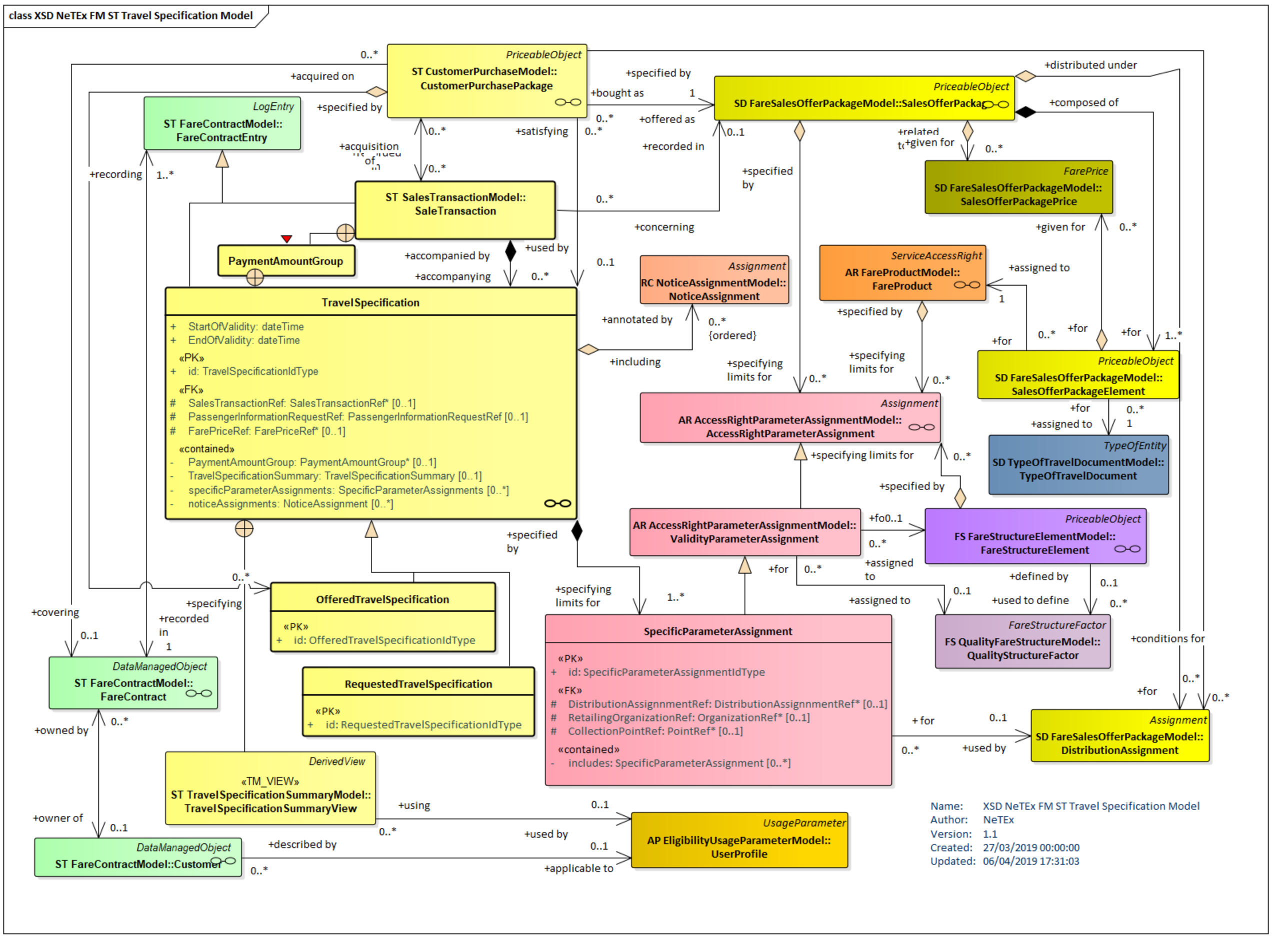


Figure 373 — Travel Specification – Physical Model (UML)

8.1.6.4 Travel Specification Summary – Physical model

The ***TravelSpecificationSummaryView*** is a view element that provides a further summary of the key parameters of an intended consumption (e.g. origin, destination and time) or purchase. The detailed parameters can be derived from the detailed parameter assignments of the TRAVEL SPECIFICATION.

The following figure shows the physical model for a ***TravelSpecificationSummaryView***.

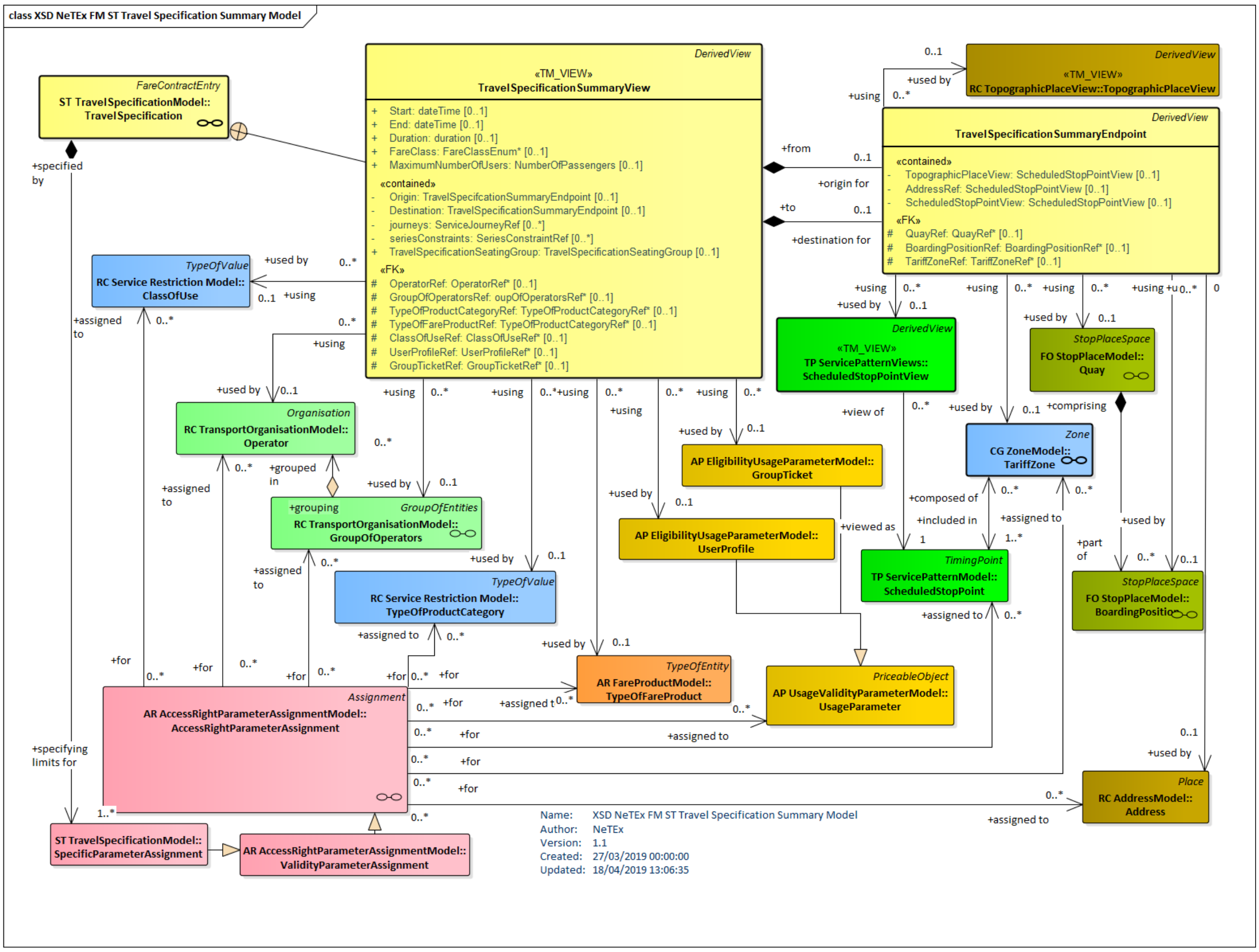


Figure 374 — Travel Specification Summary – Physical Model (UML)

8.1.6.5 Travel Specification Seating Summary – Physical model

The ***TravelSpecificationSummaryView*** includes elements the key parameters for seating (e.g. carriage, seat number, couchette type, etc.) in an intended consumption or purchase. The detailed parameters can be derived from the detailed parameter assignments of the TRAVEL SPECIFICATION.

The following figure shows the physical model for a ***TravelSpecificationSummaryView seating elements.***

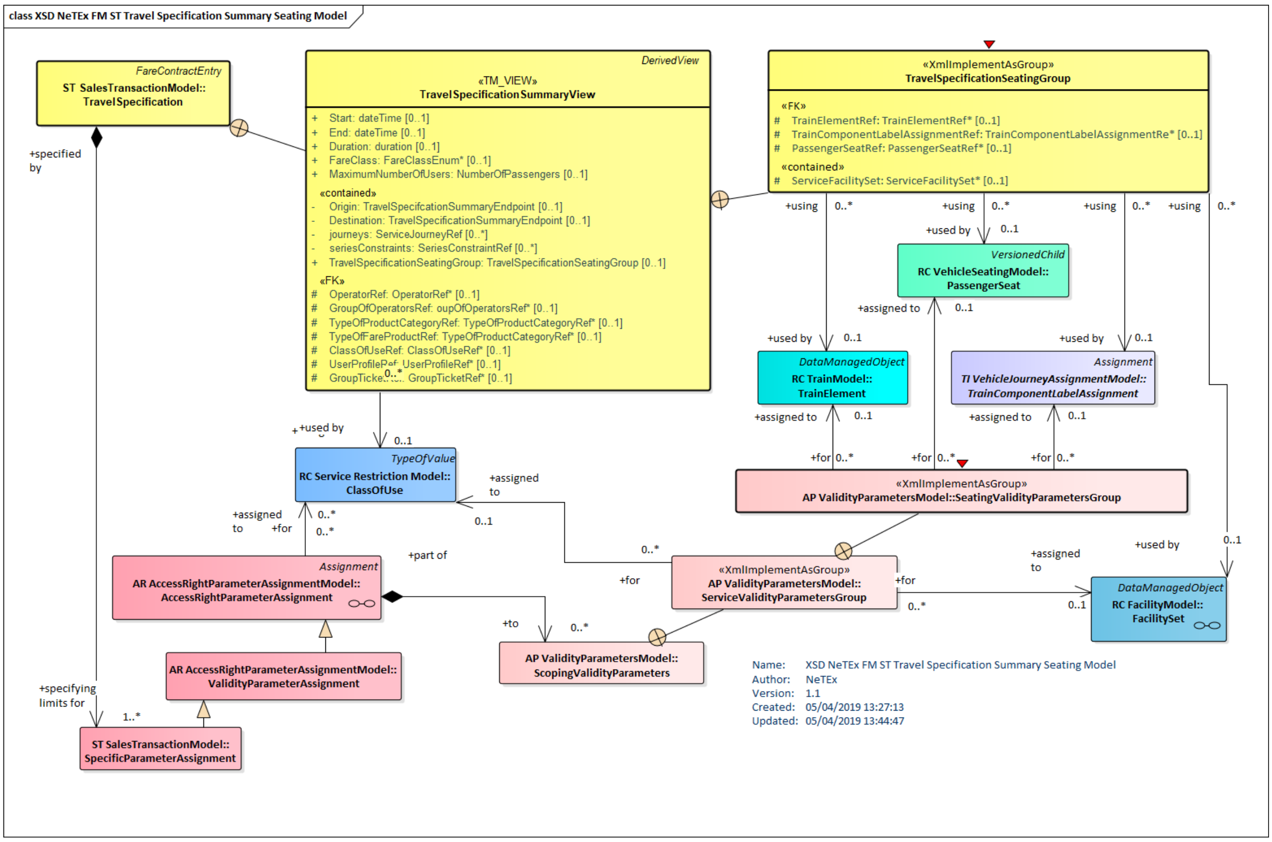


Figure 375 — TravelSpecificationSummarySeatingGroup – Physical Model (UML)

8.1.6.6 Travel Specification – Attributes and XSD

8.1.6.6.1 TravelSpecification – Model Element

The recording of the specification by a customer of parameters giving details of an intended consumption (e.g. origin and destination of a travel).

Table 335 – *TravelSpecification* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareContractEntry*	::>	TRAVEL SPECIFICATION inherits from FARE CONTRACT ENTRY
«PK»	***id***	*TravelSpecificationIdType*	1:1	Identifier of TRAVEL SPECIFICATION.
«FK»	***Sales-TransactionRef***	*SalesTransactionRef*	0:1	SALES TRANSACTION for TRAVEL SPECIFICATION.
«cntd»	***PasengerInform-ationRequestRef***	*PassengerInformation-RequestRef*	1:1	Reference to PASSENGER INFORMATION REQUEST that triggered specification.
XGRP	***Travel-Specification-PriceGroup***	**xmlGroup**	0:1	Price elements for TRAVEL SPECIFICATION.
	StartOfValidity	*xsd:dateTime*	0:1	Start of validity of travel.

	EndOfValidity	*xsd:dateTime*	0:1	End of validity of travel.
«cntd»	***Travel-Specification-Summary***	*TravelSpecificationSummary*	0:1	Summary of key parameters of TRAVEL SPECIFICATION. +V1.1
«cntd»	***specific-Parameter-Assignments***	*SpecificParameter-Assignment*	0:*	SPECIFIC PARAMETER ASSIGNMENTs relating to TRAVEL SPECIFICATION.
«cntd»	***notice-Assignments***	*NoticeAssignment*	0:*	NOTICE ASSIGNMENTs relating to TRAVEL SPECIFICATION.

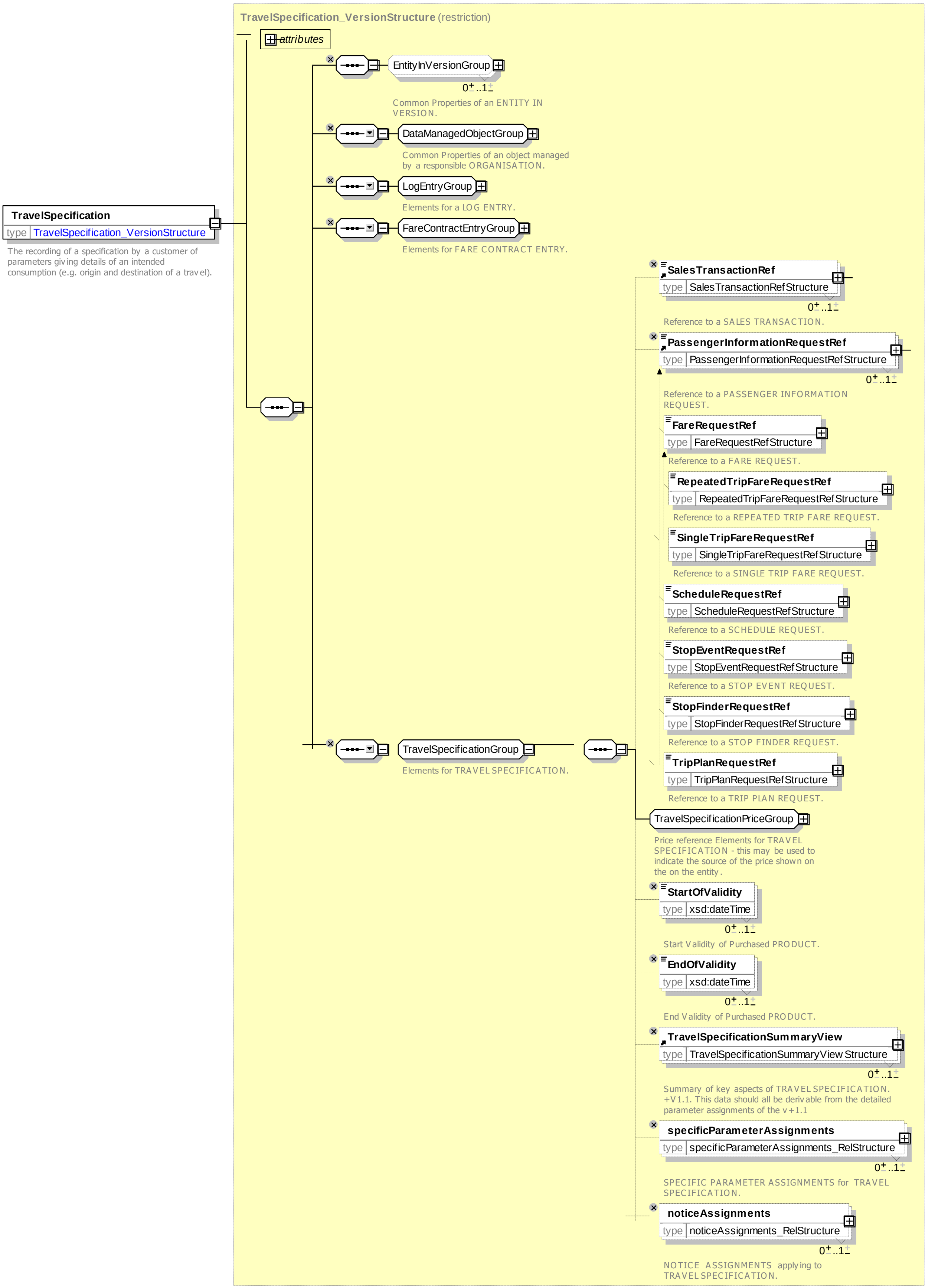


Figure 376 — *TravelSpecification* — XSD

8.1.6.6.1.1 TravelSpecificationPriceGroup – Group

The ***TravelSpecificationPriceGroup*** holds elements describing the FARE PRICE used and a record of the calculation.

Table 336 – *TravelSpecificationPriceGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***FarePriceRef***	*FarePriceRef+*	0:1	Reference to Price used for TRAVEL SPECIFICATION.
	PaymentAmount-Group	*PaymentAmountGroup*	0:1	Payment amount including possible derivation steps.

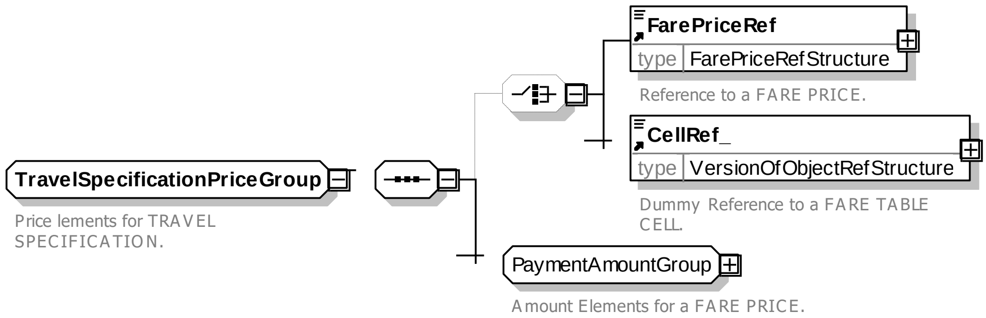


Figure 377 — *TravelSpecification*PriceGroup — XSD

8.1.6.6.1.2 PaymentAmountGroup – Group

A description of a payment, amount possibly including the derivation steps.

Table 337 – *PaymentAmountGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Amount	*AmountType*	0:1	Amount of Currency.
	Currency	*CurrencyType*	0:1	Currency for amount.
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	PRICE UNIT for AMOUNT
	Units	*xsd:integer*	0:1	Number of units
«cntd»	***ruleStepResults***	*RuleStepResult*	0:*	RULE STEP RESULTS FOR calculation steps used to compute amount.
«enum»	***PaymentMethod***	*PaymentMethodEnum*	0:1	Payment Method value for payment of amount.
«FK»	***TypeOfPayment MethodRef***	*TypeOfPaymentMethodRef*	0:1	TYPE OF PAYMENT METHOD for payment.

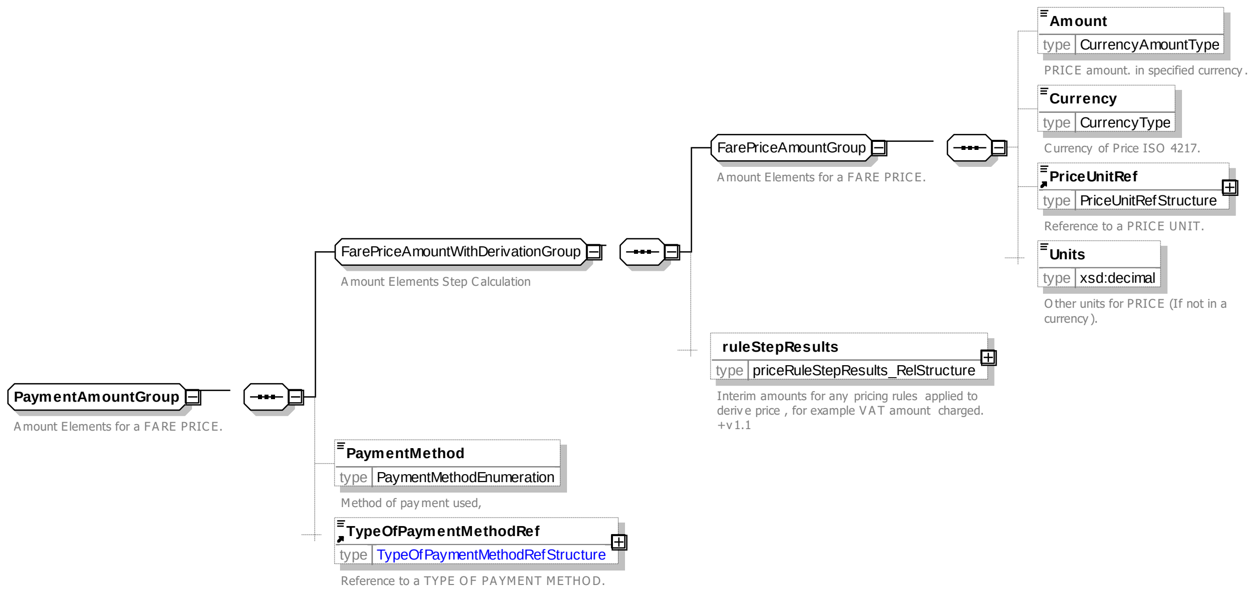


Figure 378 — *PaymentAmount*Group — XSD

8.1.6.6.2 RequestedTravelSpecification – Model Element

A set of parameters giving details of an intended consumption of access rights requested by a TRANSPORT CUSTOMER (e.g. origin and destination of a travel, class of travel, etc.).

Table 338 – *RequestedTravelSpecification* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*TravelSpecificationEntry*	::>	REQUESTED TRAVEL SPECIFICATION inherits from TRAVEL SPECIFICATION.
«PK»	***id***	*RequestedTravelSpecification IdType*	1:1	Identifier of REQUESTED TRAVEL SPECIFICATION.

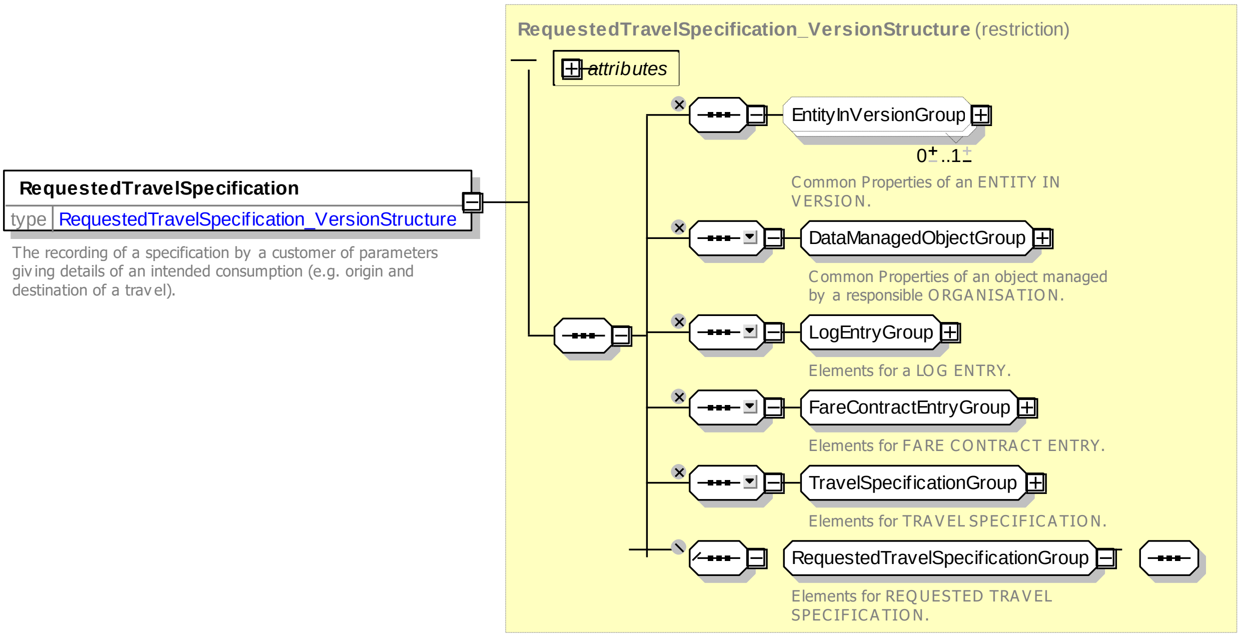


Figure 379 — Requested*TravelSpecification* — XSD

8.1.6.6.3 OfferedTravelSpecification – Model Element

A set of parameters giving details of an intended consumption of access rights offered by a TRANSPORT CUSTOMER (e.g. origin and destination of a travel, class of travel, etc.).

Table 339 – Offered*TravelSpecification* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*TravelSpecificationEntry*	::>	OFFERED TRAVEL SPECIFICATION inherits from TRAVEL SPECIFICATION.
«PK»	***id***	*OfferedTravelSpecification-IdType*	1:1	Identifier of OFFERED TRAVEL SPECIFICATION.

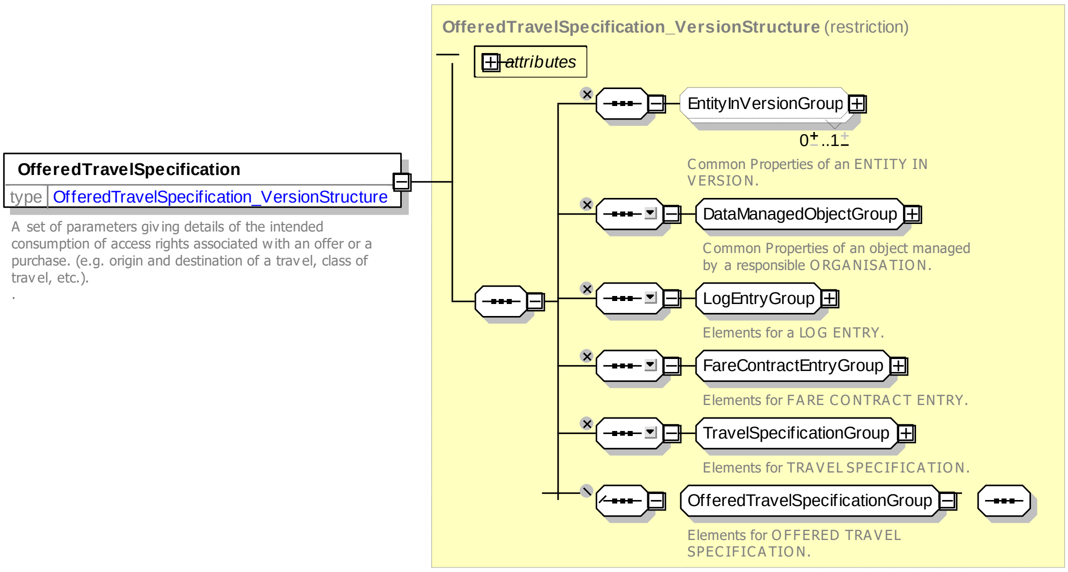


Figure 380 — Offered*TravelSpecification* — XSD

8.1.6.6.1 *TravelSpecificationSummaryView* – Model Element

A summary of the key parameters of an intended consumption (e.g. origin, destination and time of travel).

Table 340 – *8.2.7.5.1 TravelSpecificationSummaryView* – Element

Classifi-cation	Name	Type	Cardinality	Description
«cntd»	***Origin***	*TravelSpecification-SummaryEndpoint*	0:1	Origin of travel.
«cntd»	***Destination***	*TravelSpecification-SummaryEndpoint*	0:1	Destination of travel.
XGRP	***TravelSpecification-SummaryJourneyGroup***	***xmlGroup***	0:1	Elements summarising travel access rights for TRAVEL SPECIFICATION.
XGRP	***TravelSpecification-SummaryFareGroup***	***xmlGroup***	0:1	Elements summarising fare product attributes for TRAVEL SPECIFICATION.
XGRP	***TravelSpecification-SummarySeatingGroup***	***xmlGroup***	0:1	Elements summarising seating access rights for TRAVEL SPECIFICATION.

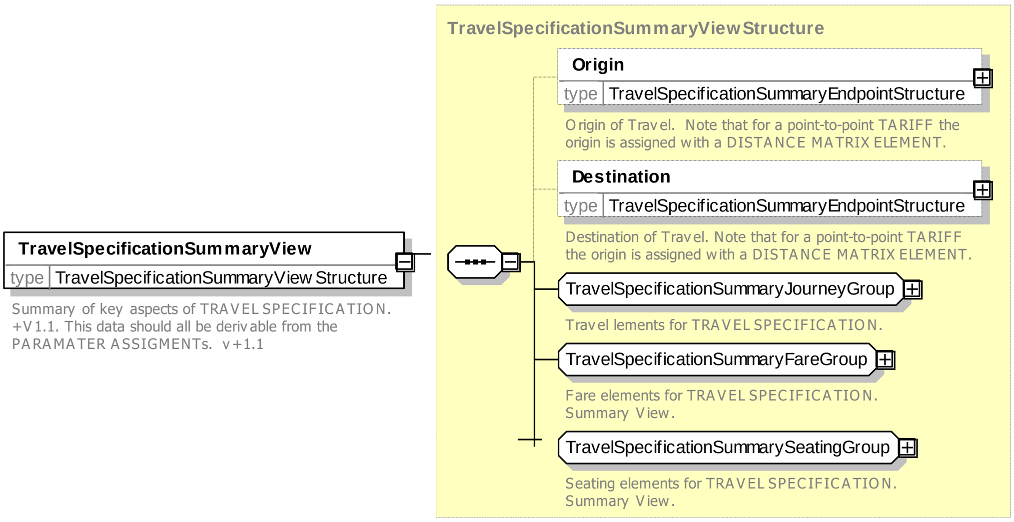


Figure 381 — *8.2.7.5.1 TravelSpecificationSummaryView* — XSD

8.1.6.6.1.1 TravelSpecificationSummaryEndpoint – Model Element

A summary of the key parameters of an intended consumption (e.g. origin, destination and time of travel).

Table 341 – *8.2.7.5.1 TravelSpecificationSummaryEndpoint* – Element

Classifi-cation	Name	Type	Cardin ality	Description
«FV»	***Topographic-PlaceView***	*Topographic-PlaceView*	0:1	View of TOPOGRAPHIC PLACE for TRAVEL SPECIFICATION, with some derived values.
«FK»	***AddressRef***	*AddressRef*	0:1	ADDRESS for origin or destination of TRAVEL SPECIFICATION.
«FV»	***ScheduledStop-PointView***	*ScheduledStopPointView*	0:1	View of SCHEDULED STOP POINT for origin or destination of TRAVEL SPECIFICATION.
«FK»	***QuayRef***	*QuayRef*	0:1	QUAY for origin or destination of TRAVEL SPECIFICATION.
«FK»	***BoardingPosition Ref***	*BoardingPositionRef*	0:1	BOARDING POSITION for origin or destination of TRAVEL SPECIFICATION.
«FK»	***TariffZoneRef***	*TariffZoneRef*	0:1	TARIFF ZONE for origin or destination of TRAVEL SPECIFICATION.

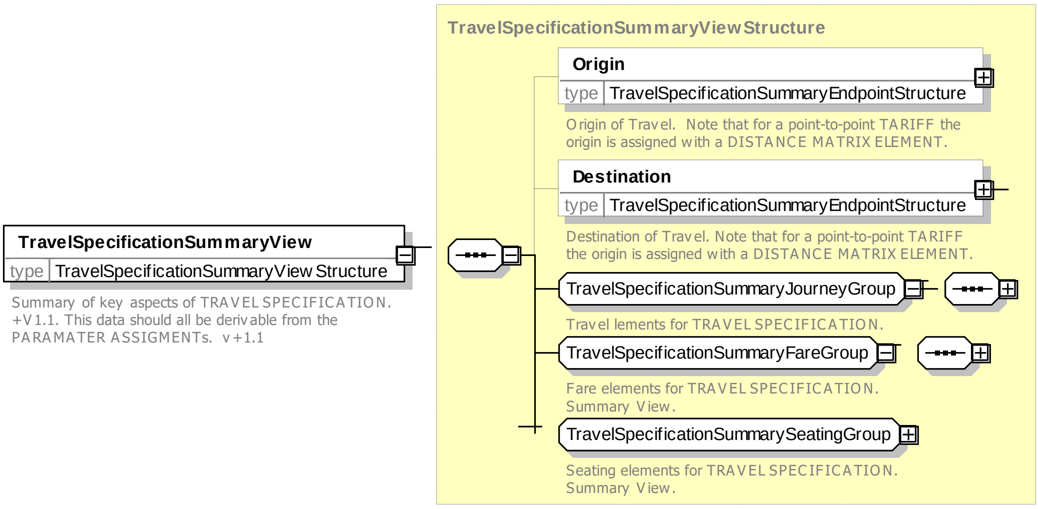


Figure 382 — *TravelSpecificationSummaryEndPoint* — XSD

8.1.6.6.1.2 TravelSpecificationSummaryJourneyGroup – Group

A summary of the key journey parameters of a TRAVEL SPECIFICATION.

Table 342 – *TravelSpecificationSummaryJourneyGroup* – Group

Classifi-cation	Name	Type	Cardin ality	Description
	Start	*dateTime*	0:1	Start time of travel.
	End	*dateTime*	0:1	End time of travel.
	Duration	*duration*	0:1	Duration of travel.
«cntd»	***journeys***	*ServiceJourneyRef*	0:*	Elements summarising travel.
«cntd»	***series-Constraints***	*SeriesConstraintRef*	0:*	SERIES CONSTRAINTs i.e. routing for travel.
«FK»	***OperatorRef***	*OperatorRef*	0:1	Operator for trip.
«FK»	***GroupOf-OperatorsRef***	*GroupOfOperatorsRef*	0:1	GROUP OF OPERATORs for trip.

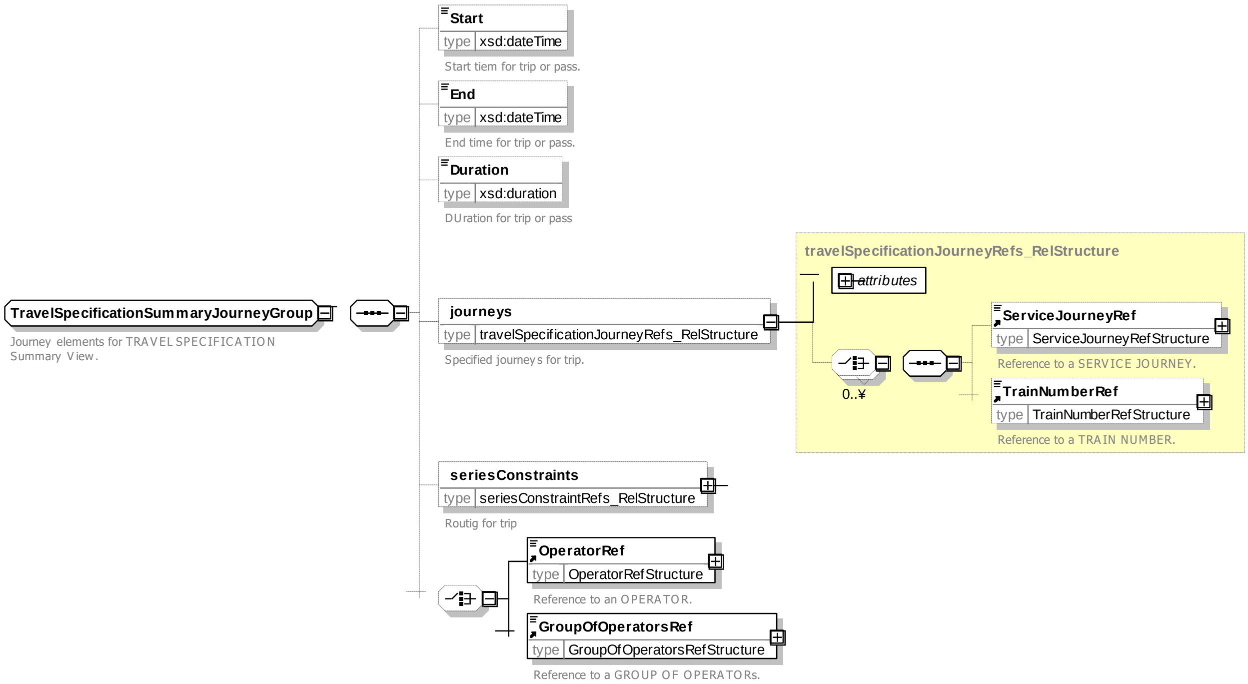


Figure 383 — *TravelSpecificationSummaryJourneyGroup* — XSD

8.1.6.6.1.3 TravelSpecificationSummaryFareGroup – Group

A summary of the key fare parameters of a TRAVEL SPECIFICATION.

Table 343 – *8.2.7.5.1 8.2.7.5.2 TravelSpecificationSummaryFareGroup* – Group

Classifi-cation	Name	Type	Card in-ality	Description
«FK»	***TypeOfProduct-CategoryRef***	*TypeOfProductCategoryRef*	0:1	TYPE OF PRODUCT CATEGORY allowed for trip.
«FK»	***TypeOfFare-ProductRef***	*TypeOfFareProductRef*	0:1	TYPE OF FARE PRODUCT allowed for trip.
«enum»	***FareClass***	*FareClassEnum*	0:1	CLASS OF USE for TRAVEL SPECIFICATION. See Part1 for allowed values.
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	CLASS OF USE for TRAVEL SPECIFICATION.
«FK»	***UserProfileRef***	*UserProfileRef*	0:1	USER PROFILE required for travel.
«FK»	***GroupTicketRef***	*GroupTicketRef*	0:1	GROUP TICKET, if any, for TRAVEL SPECIFICATION.
	Maximum-NumberOfUsers	*NumberOfPassengers*	0:1	If GROUP TICKET, maximum number of users making the trip.

Figure 384 — *8.2.7.5.1 TravelSpecificationSummaryFareGroup* — XSD

8.1.6.6.1.4 TravelSpecificationSummary Seating Group – Group

A summary of the key seating parameters of a TRAVEL SPECIFICATION.

Table 344 – *8.2.7.5.1 8.2.7.5.2 TravelSpecificationSummarySeatingGroup* – Group

Classifi-cation	Name	Type	Card in-ality	Description
«FK»	***TrainElementRef***	*TrainElementRef*	0:1	Reference to a TRAIN ELEMENT.
«FK»	***TrainComponent Label-AssignmentRef***	*TrainComponentLabel-AssignmentlRef*	0:1	Reference to a TRAIN COMPONENT LABEL ASSIGNMENT.
«FK»	***PassengerSeat-Ref***	*PassengerSeatRef*	0:1	Reference to a PASSENGER SEAT.
«cntd»	***ServiceFacility-Set***	*ServiceFacilitySet*	0:1	SERVICE FACILITY SEAT with available facilities.

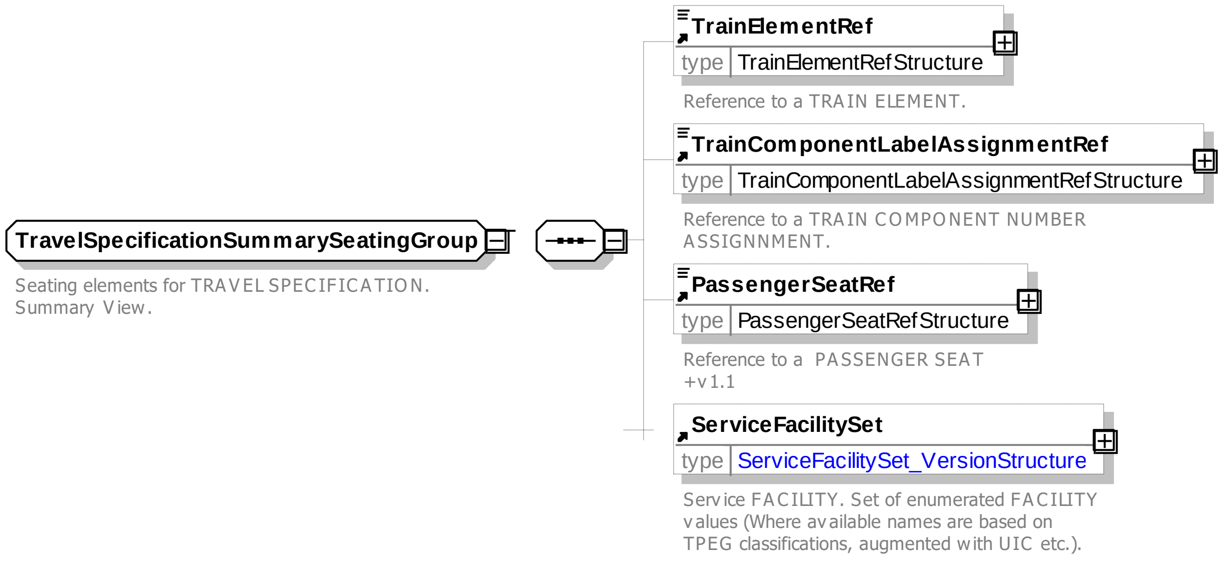


Figure 385 — *TravelSpecificationSummarySeatingGroup* — XSD

8.1.7 Customer Purchase Package

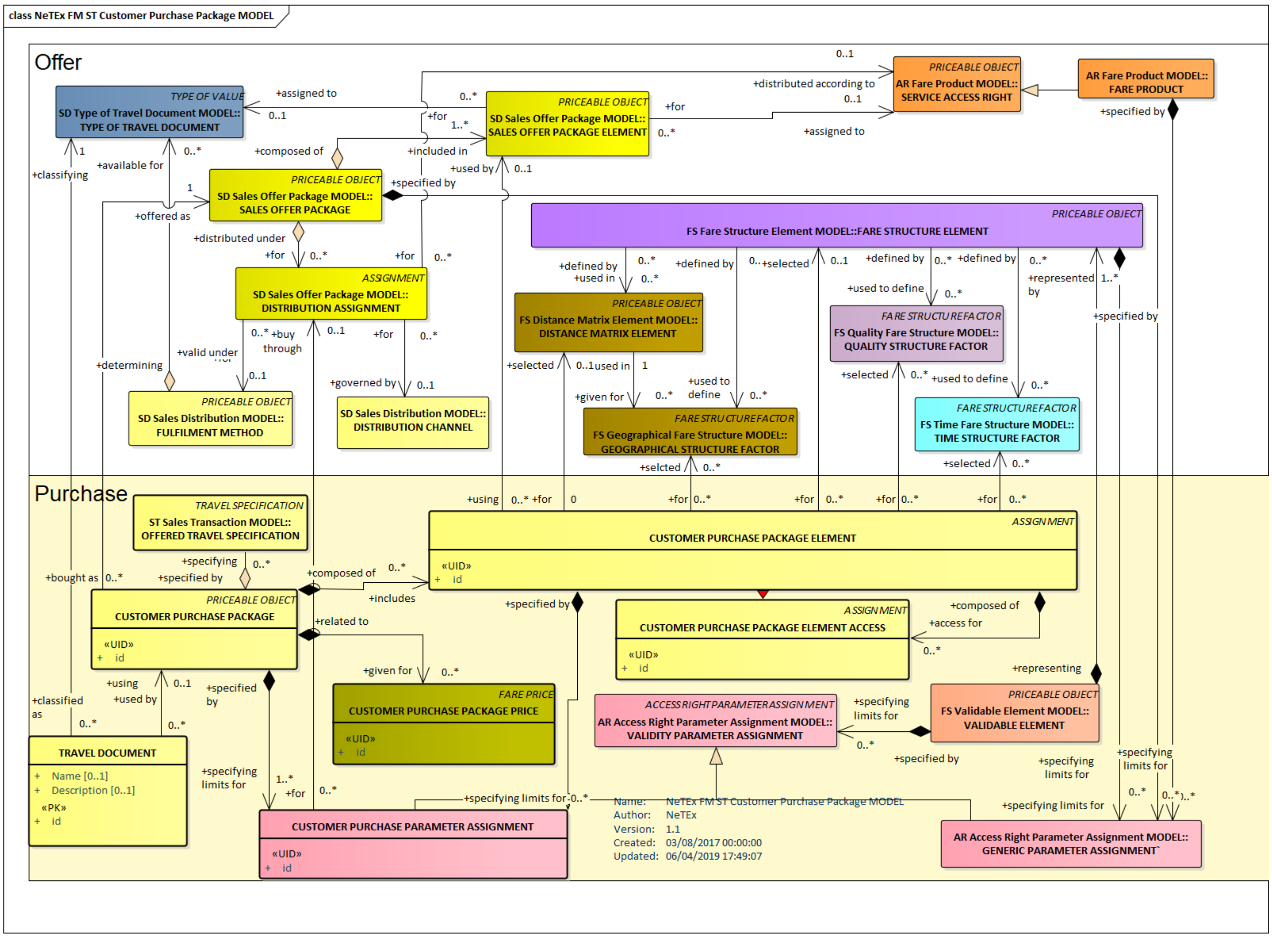
8.1.7.1 CUSTOMER PURCHASE PACKAGE – Conceptual MODEL

The CUSTOMER PURCHASE PACKAGE model describes an individual product sale.

Each CUSTOMER PURCHASE PACKAGE describes the purchase of a SALES OFFER PACKAGE. The actual access rights intended to purchase and any limitations on them are described by a TRAVEL SPECIFICATION. Thus, for example a FARE PRODUCT might be available for different stops and for different classes of user. The CUSTOMER PURCHASE PACKAGE would record which specific stops (e.g. as a reference to a DISTANCE MATRIX ELEMENT) and what is the actual CLASS OF USE, and identify the type of user, for example as a USER PROFILE reference.

— a CUSTOMER PURCHASE PACKAGE may concern an identified CUSTOMER, through the relationship between FARE CONTRACT ENTRY and CUSTOMER;
— the parameters assigned by a TRAVEL SPECIFICATION are, in most cases, stored on an individual TRAVEL DOCUMENT, through the relationship between FARE CONTRACT ENTRY and TRAVEL DOCUMENT.

A CUSTOMER PURCHASE PACKAGE often accompanies a SALE TRANSACTION, occurring at the same time or soon after.



class NeTEx FM ST Customer Purchase Package MODEL
Offer
TYPE OF VALUE
SD Type of Travel Document MODEL::
TYPE OF TRAVEL DOCUMENT
PRICEABLE OBJECT
SD Sales Offer Package MODEL::
SALES OFFER PACKAGE ELEMENT
PRICEABLE OBJECT
AR Fare Product MODEL::
SERVICE ACCESS RIGHT
AR Fare Product MODEL::
FARE PRODUCT
PRICEABLE OBJECT
SD Sales Offer Package MODEL::
SALES OFFER PACKAGE
PRICEABLE OBJECT
FS Fare Structure Element MODEL::FARE STRUCTURE ELEMENT
ASSIGNMENT
SD Sales Offer Package MODEL::
DISTRIBUTION ASSIGNMENT
PRICEABLE OBJECT
FS Distance Matrix Element MODEL::
DISTANCE MATRIX ELEMENT
FARE STRUCTURE FACTOR
FS Quality Fare Structure MODEL::
QUALITY STRUCTURE FACTOR
PRICEABLE OBJECT
SD Sales Distribution MODEL::
FULFILMENT METHOD
SD Sales Distribution MODEL::
DISTRIBUTION CHANNEL
FARE STRUCTURE FACTOR
FS Geographical Fare Structure MODEL::
GEOGRAPHICAL STRUCTURE FACTOR
FARE STRUCTURE FACTOR
FS Time Fare Structure MODEL::
TIME STRUCTURE FACTOR
Purchase
TRAVEL SPECIFICATION
ST Sales Transaction MODEL::
OFFERED TRAVEL SPECIFICATION
ASSIGNMENT
CUSTOMER PURCHASE PACKAGE ELEMENT
«UID»
+ id
PRICEABLE OBJECT
CUSTOMER PURCHASE PACKAGE
«UID»
+ id
ASSIGNMENT
CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS
«UID»
+ id
FARE PRICE
CUSTOMER PURCHASE PACKAGE PRICE
«UID»
+ id
ACCESS RIGHT PARAMETER ASSIGNMENT
AR Access Right Parameter Assignment MODEL::
VALIDITY PARAMETER ASSIGNMENT
PRICEABLE OBJECT
FS Validable Element MODEL::
VALIDABLE ELEMENT
TRAVEL DOCUMENT
+ Name [0..1]
+ Description [0..1]
«PK»
+ id
CUSTOMER PURCHASE PARAMETER ASSIGNMENT
«UID»
+ id
AR Access Right Parameter Assignment MODEL::
GENERIC PARAMETER ASSIGNMENT
Name: NeTEx FM ST Customer Purchase Package MODEL
Author: NeTEx
Version: 1.1
Created: 03/08/2017 00:00:00
Updated: 06/04/2019 17:49:07

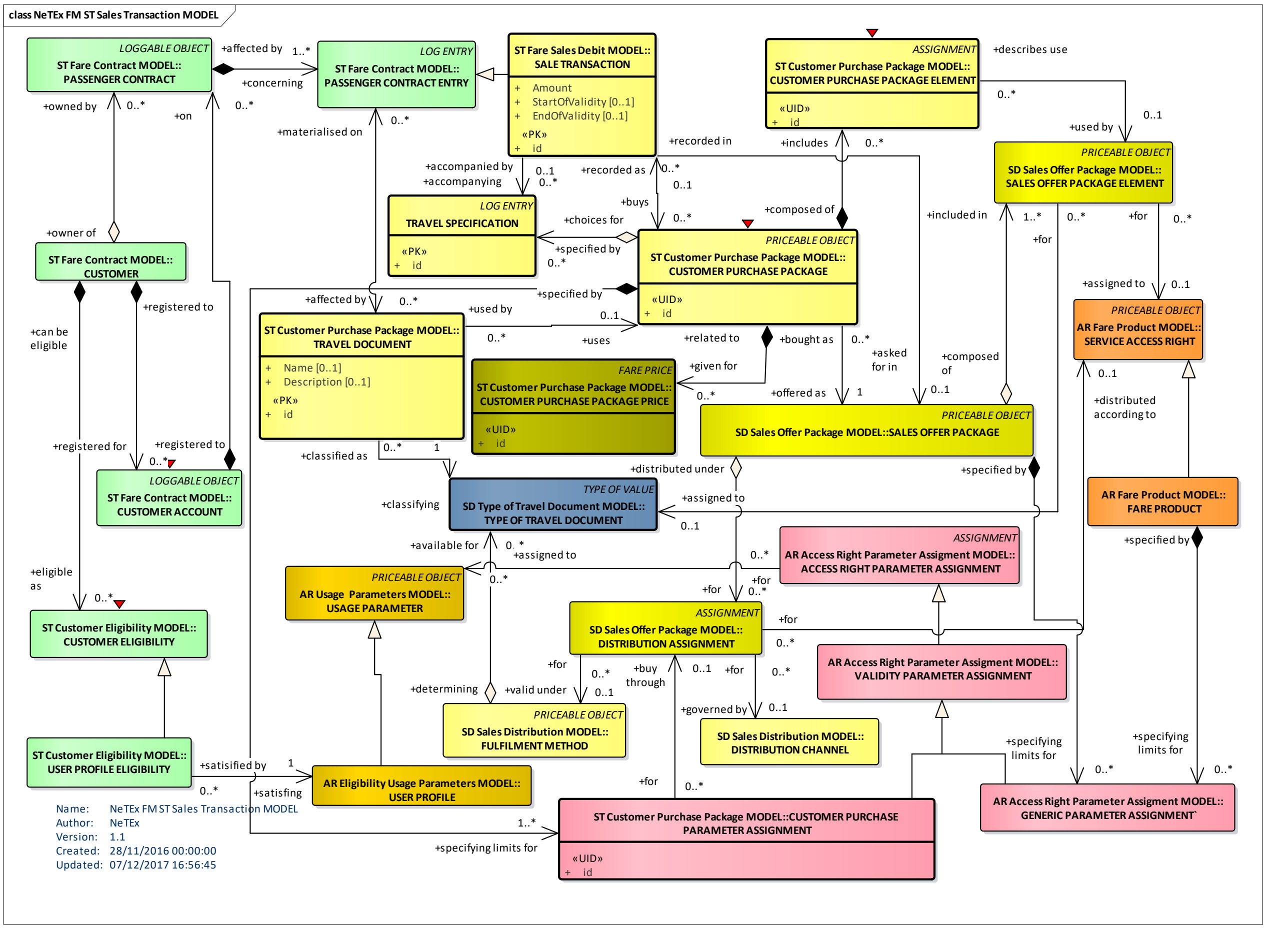


Figure 386 — Customer Purchase Package – Conceptual MODEL (UML)

8.1.7.2 Customer Purchase Package – Physical model

The following figure shows the physical model for a CUSTOMER PURCHASE PACKAGE.

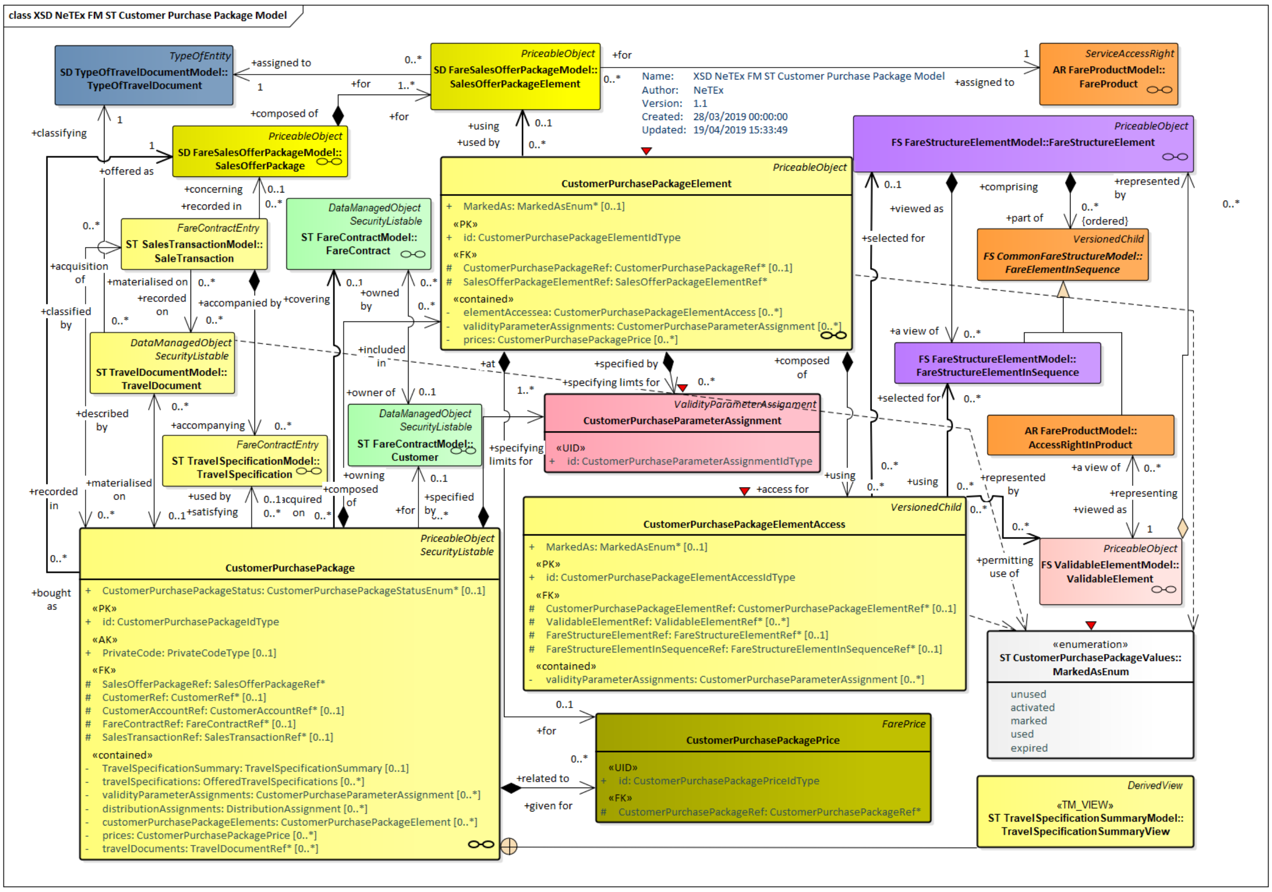


Figure 387 — Customer Purchase Package – Physical Model (UML)

8.1.7.3 Travel Document – Physical model

The following figure shows the physical model for a TRAVEL DOCUMENT.

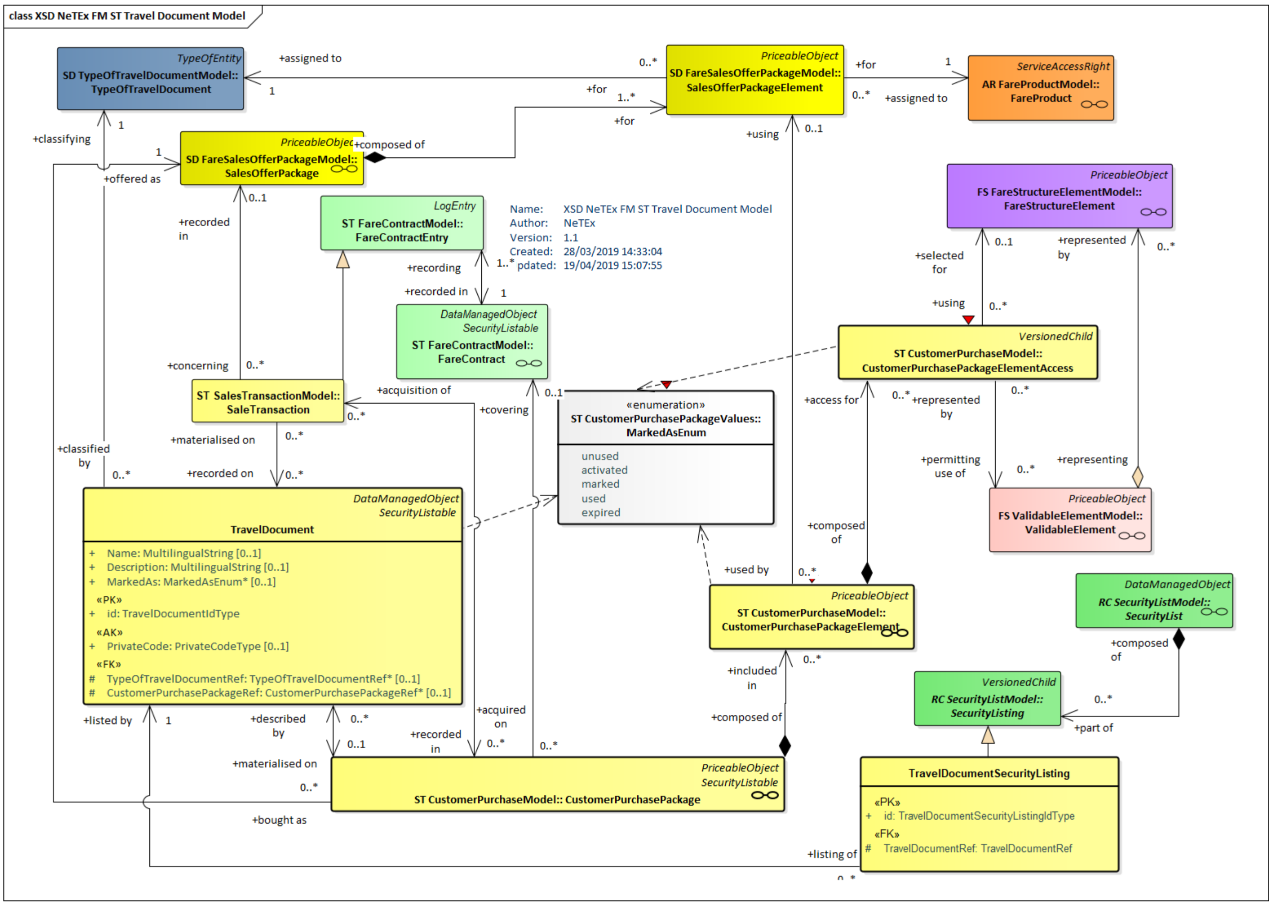


Figure 388 — TravelDocument – Physical Model (UML)

8.1.7.4 Customer Purchase Package – Attributes and XSD

8.1.7.4.1 CustomerPurchasePackage – Model Element

A purchase of a SALES OFFER PACKAGE by a CUSTOMER, giving access rights to one or several FARE PRODUCTs materialised as one or several TRAVEL DOCUMENTS.

Table 345 – CustomerPurchasePackage – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*PriceableObject*	::>	CUSTOMER PURCHASE PACKAGE inherits from PRICEABLE OBJECT.
«PK»	***id***	*CustomerPurchasePackage-IdType*	1:1	Identifier of a CUSTOMER PURCHASE PACKAGE.
«AK»	***PrivateCode***	*PrivateCodeType*	0:1	Alternative identifier of CUSTOMER PURCHASE PACKAGE.
«FK»	***SalesOffer-PackageRef***	*SalesOfferPackageRef*	1:1	Reference to a SALES OFFER PACKAGE for which this is a purchase.

«FK»	***CustomerRef***	*CustomerRef*	0:1	Reference to a CUSTOMER for which this is a purchase.
«FK»	***Customer-AccountRef***	*CustomerAccountRef*	**0:1**	Reference to a CUSTOMER ACCOUNT for which this is a purchase.
«FK»	***FareContractRef***	*FareContractRef*	0:1	Reference to a FARE CONTRACT for which this is a purchase.
«enum»	***Customer-Purchase-PackageStatus***	*CustomerPurchasePackage-StatusEnum*	0:1	Status value for CUSTOMER PURCHASE PACKAGE. See allowed values below. +v1.1
XGRP	***Customer-Purchase-Package-SelectionGroup***	***xmlGroup***	0:1	FARE PRODUCT and SALES OFFER PACKAGE Selection elements for CUSTOMER PURCHASE PACKAGE
«FK»	***SalesTransaction Ref***	*SalesTransactionRef*	0:1	Reference to a SALES TRANSACTION recording initial purchase.
«cntd»	***sales-Transactions***	*SalesTransactionRef*	0:*	Reference to additional SALES TRANSACTION recording any additional purchases.
«cntd»	***prices***	*CustomerPurchasePackage-Price*	0:*	CUSTOMER PURCHASE PACKAGE PRICEs associated with the CUSTOMER PURCHASE PACKAGE.
«cntd»	***travelDocuments***	*TravelDocumentRef*	0:*	TRAVEL DOCUMENTS s associated with the CUSTOMER PURCHASE PACKAGE.

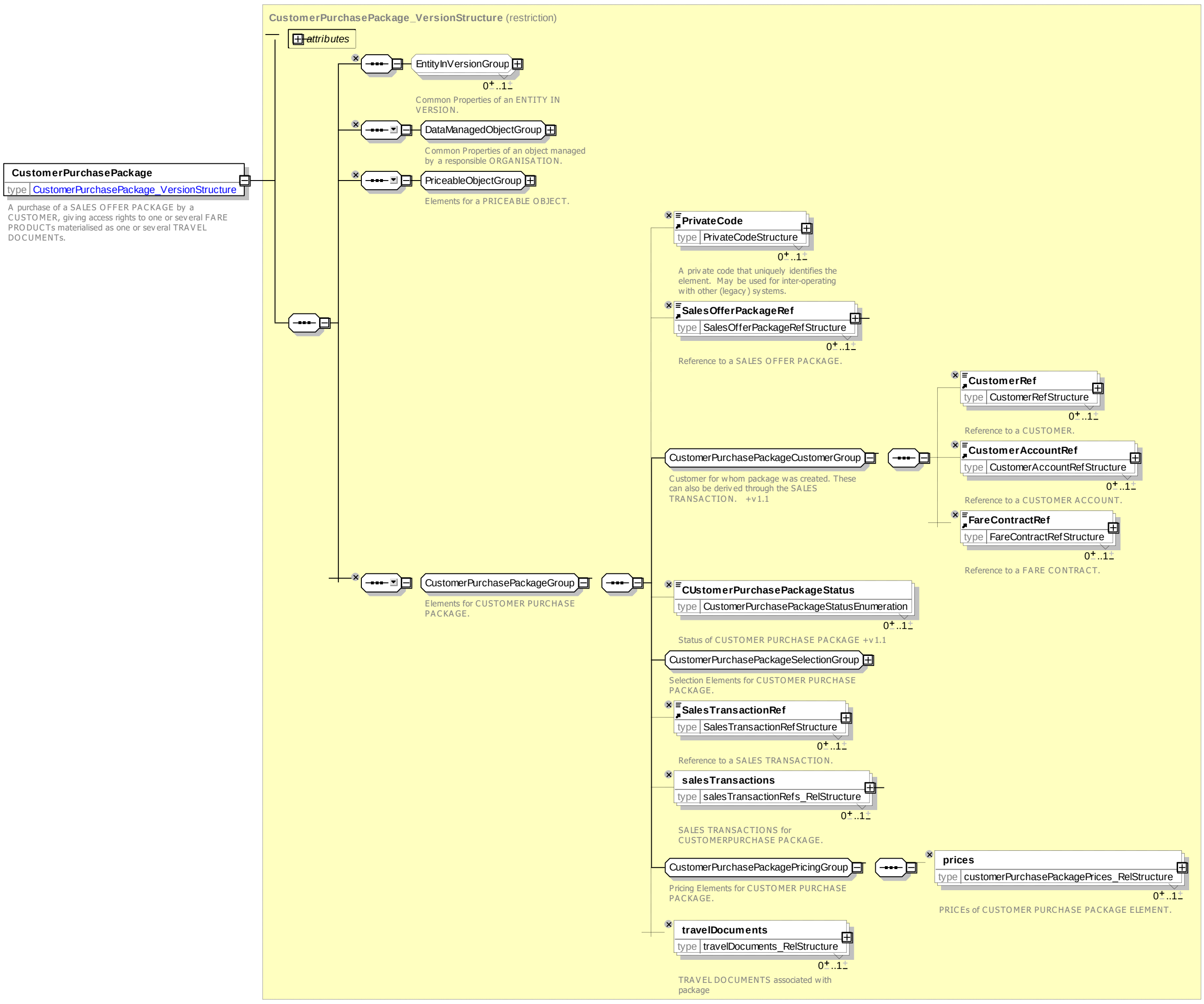


Figure 389 — CustomerPurchasePackage — XSD

8.1.7.4.1.1 CustomerPurchasePackageSelectionGroup – Group

Elements of a CUSTOMER PURCHASE PACKAGE relating to the selection of parameters.

Table 346 – 8.2.7.5.1 CustomerPurchasePackageSelectionGroup – Group

Classifi-cation	Name	Type	Cardin-ality	Description
«cntd»	***Travel-Specification-Summary***	*TravelSpecification-Summary*	0:1	Summary of key parameters of CUSTOMER PURCHASE PACKAGE.
«cntd»	***travel-Specifications***	*TravelSpecification*	0:*	TRAVEL SPECIFICATIONs associated with the CUSTOMER PURCHASE PACKAGE.

«cntd»	***validity-Parameter-Assignments***	*CustomerPurchase-ParameterAssignment*	0:*	CUSTOMER PURCHASE PACKAGE PARAMETER ASSIGNMENTs associated with the CUSTOMER PURCHASE PACKAGE.
«cntd»	***distribution-Assignments***	*DistributionAssignment*	0:*	DISTRIBUTION ASSIGNMENTs associated with the CUSTOMER PURCHASE PACKAGE.
«cntd»	***customer-Purchase-Package-Elements***	*CustomerPurchasePackage Element*	0:*	CUSTOMER PURCHASE PACKAGE ELEMENTs associated with the CUSTOMER PURCHASE PACKAGE.

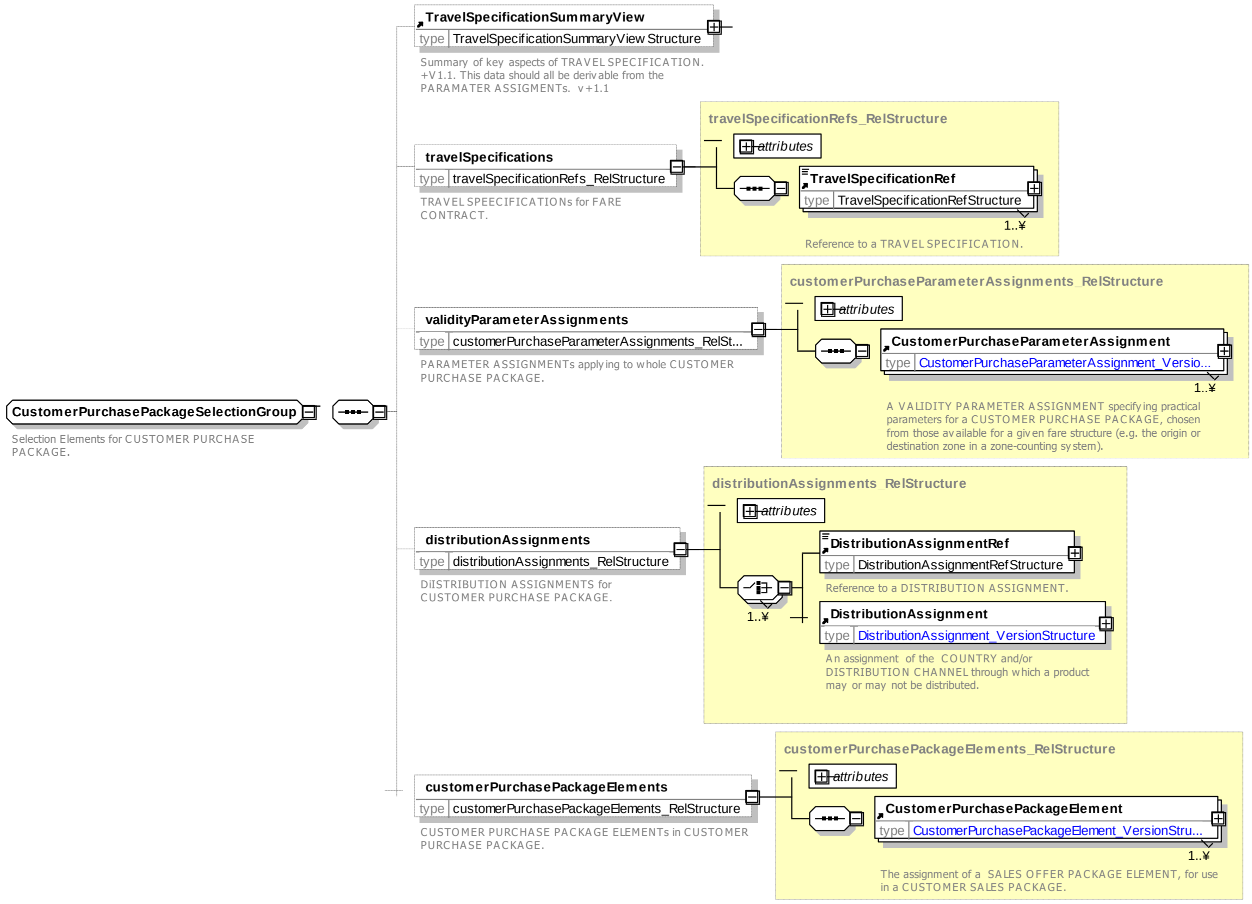


Figure 390 — CustomerPurchasePackageSelectionGroup — XSD

CustomerPurchasePackageStatus – Allowed values

The following table shows the allowed values for ***CustomerPurchasePackageStatus*** (*CustomerPurchase-PackageStatusEnumeration*).

Table 347 – *CustomerPurchasePackageStatus*– Allowed values

Value	Description
reserved	Reserved but not paid for.
ordered	Purchased but not paid for.
paidFor	Paid for.

unused	Not yet used.
activated	Activated for use.
partiallyUsed	Partially used.

used	Fully used.
archived	Archived.
other	Other status

8.1.7.4.1 CustomerPurchasePackageElement – Model Element

The assignment of a SALES OFFER PACKAGE ELEMENT, for use in a CUSTOMER SALES PACKAGE.

Table 348 – CustomerPurchasePackageElement – Element

Classifi-cation	Name	Type	Card in-ality	Description
::>	::>	*PriceableObject*	::>	***CustomerPurchasePackageElement*** inherits from ***PriceableObject***
«PK»	***id***	*CustomerPurchase-PackageElementIdType*	1:1	Identifier of a CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***Customer-Purchase-PackageRef***	*CustomerPurchasePackageRef*	0:1	Reference to a CUSTOMER PURCHASE PACKAGE.
«FK»	***SalesOffer-PackageElement Ref***	*SalesOfferPackageElementRef*	1:1	Reference to a SALES OFFER PACKAGE ELEMENT for which this is a purchase.
«enum»	***MarkedAs***	*MarkedAsEnum*	0:1	Status of the CUSTOMER PURCHASE PACKAGE ELEMENT. See allowed values below.
«cntd»	***element-Accesses***	*CustomerPurchasePackage-ElementAccess*	0:*	CUSTOMER PURCHASE PACKAGE ELEMENT ACCESSes making up the CUSTOMER PURCHASE PACKAGE ELEMENT.
«cntd»	***validity-Parameter-Assignments***	*CustomerPurchaseParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.
«cntd»	***prices***	*CustomerPurchasePackage-Price*	0:*	CUSTOMER PURCHASE PACKAGE PRICEs associated with the CUSTOMER PURCHASE PACKAGE ELEMENT.

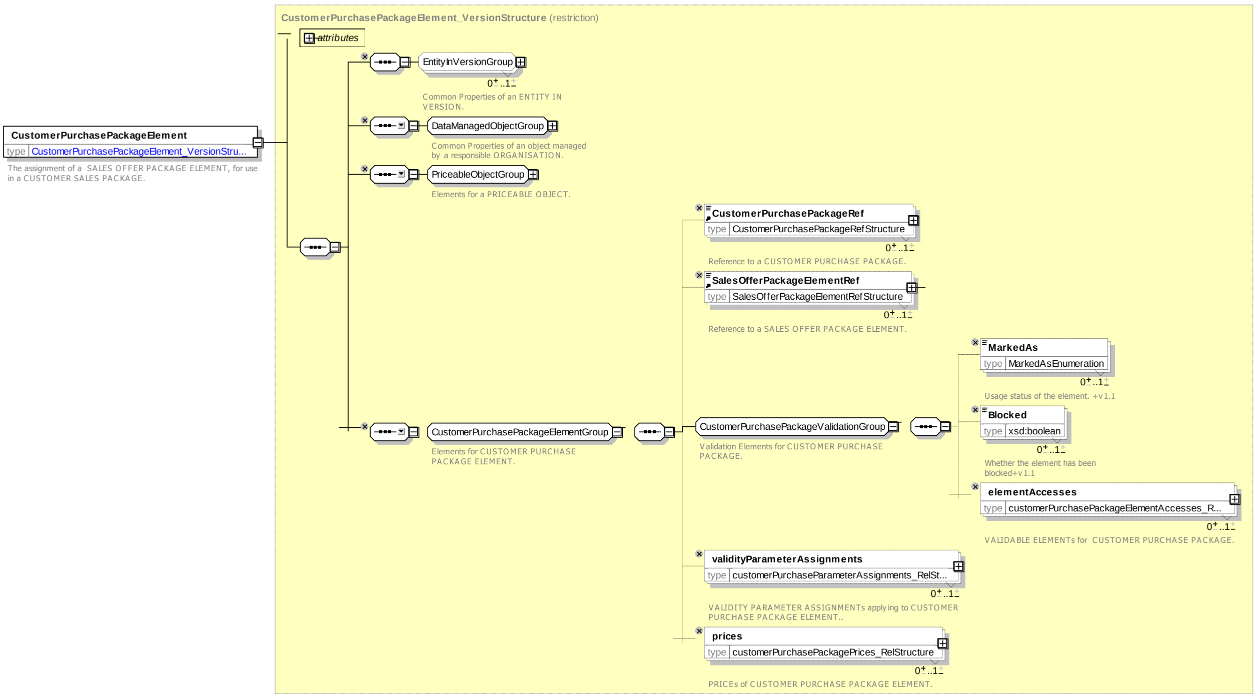


Figure 391 — CustomerPurchasePackageElement — XSD

8.1.7.4.1.1 CustomerPurchasePackageElementSelectionGroup– Group

Parameters selected for a SALES OFFER PACKAGE ELEMENT.

Table 349 – *CustomerPurchasePackageElementSelectionGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***DistanceMatrix-ElementRef***	*DistanceMatrixElementRef*	0:1	Reference to a DISTANCE MATRIX ELEMENT selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***Geographical-StructureFactor-Ref***	*GeographicalStructure-FactorRef*	0:1	Reference to a GEOGRAPHICAL STRUCTURE selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***Geographical-IntervalRef***	*GeographicalIntervalRef*	0:1	Reference to a GEOGRAPHICAL INTERVAL selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***TimeStructure-FactorRef***	*TimeStructureFactorRef*	0:1	Reference to a TIME STRUCTURE FACTOR selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***TimeIntervalRef***	*TimeIntervalRef*	0:1	Reference to a TIME INTERVAL selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***QualityStructure FactorRef***	*QualityStructure-FactorRef*	0:1	Reference to QUALITY STRUCTURE FACTOR selected for the CUSTOMER PURCHASE PACKAGE ELEMENT.

«cntd»	***validable-Elements***	*ValidableElementRef*	0:*	VALIDABLE ELEMENTs associated with the CUSTOMER PURCHASE PACKAGE ELEMENT.

Figure 392 — CustomerPurchasePackageElementSelectionGroup— XSD

8.1.7.4.1.2 MarkedAs – Allowed values

The following table shows the allowed values for ***MarkedAs*** (*MarkedAsEnumeration*).

Table 350 – *MarkedAs* – Allowed values

Value	Description
unused	Unused.
activated	Activated.
marked	Marked as partially used or used.
used	Used.
expired	Expired.

8.1.7.4.1 **CustomerPurchasePackageElementAccess – Model Element**

Access to a VALIDABLE ELEMENT by a specific CUSTOMER PURCHASE PACKAGE through use of CUSTOMER PURCHASE PACKAGE.

This is needed for validation of complex SALES OFFER PACKAGEs containing tariffs structures that have FARE STRUCTURE ELEMENTs IN SEQUENCE, in such a case a given SALES PACKAGE ELEMENT may have multiple VALIDABLE ELEMENTs associated with it, each of which can be separately validated and marked.

Table 351 – CustomerPurchasePackageElementAccess – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*VersionedChild*	::>	CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS inherits from VERSIONED CHILD
«PK»	***id***	*CustomerPurchasePackage-ElementAccessIdType*	1:1	Identifier of a CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS.
«FK»	***Customer-Purchase-Package-ElementRef***	*CustomerPurchasePackage-ElementRef*	0:1	Reference to a parent CUSTOMER PURCHASE PACKAGE ELEMENT.
«FK»	***ValidableElement Ref***	*ValidableElementRef*	0:1	VALIDABLE ELEMENT associated with the CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS.
«FK»	***FareStructure-ElementRef***	*FareStructureElementRef*	0:1	FARE STRUCTURE ELEMENT associated with the CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS.
«FK»	***FareStructure-Element-InSequenceRef***	*FareStructureElement-InSequenceRef*	0:1	FARE STRUCTURE ELEMENT IN SEQUENCE associated with the CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS.
«enum»	***MarkedAs***	*MarkedAsEnum*	0:1	Usage/marking status of CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS. See allowed values above.
«cntd»	***validity-Parameter-Assignments***	*CustomerPurchase-ParameterAssignment*	0:*	CUSTOMER PURCHASE PARAMETER ASSIGNMENTs selected for the CUSTOMER PURCHASE PACKAGE ELEMENT ACCESS.

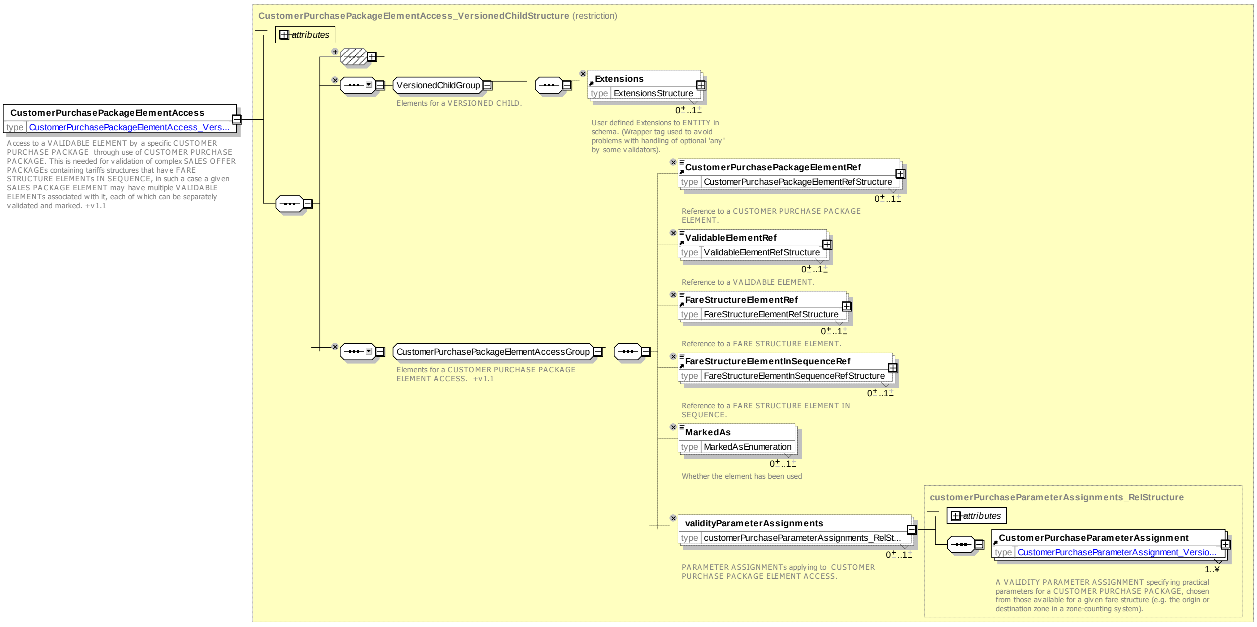


Figure 393 — CustomerPurchasePackageElementAccess — XSD

8.1.7.4.2 CustomerPurchasePackageParameterAssignment – Model Element

A VALIDITY PARAMETER ASSIGNMENT specifying practical parameters chosen for a CUSTOMER PURCHASE PACKAGE within a given fare structure (e.g. the origin or destination zone in a zone-counting system).

Table 352 – *CustomerPurchasePackageParameterAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*ValidityParameter-Assignment*	::>	CUSTOMER PURCHASE PACKAGE PARAMETER ASSIGNMENT inherits from VALIDITY PARAMETER ASSIGNMENT.
«PK»	***id***	*CustomerPurchase-PackageParameter-AssignmentIdType*	1:1	Identifier of CUSTOMER PURCHASE PACKAGE PARAMETER ASSIGNMENT.

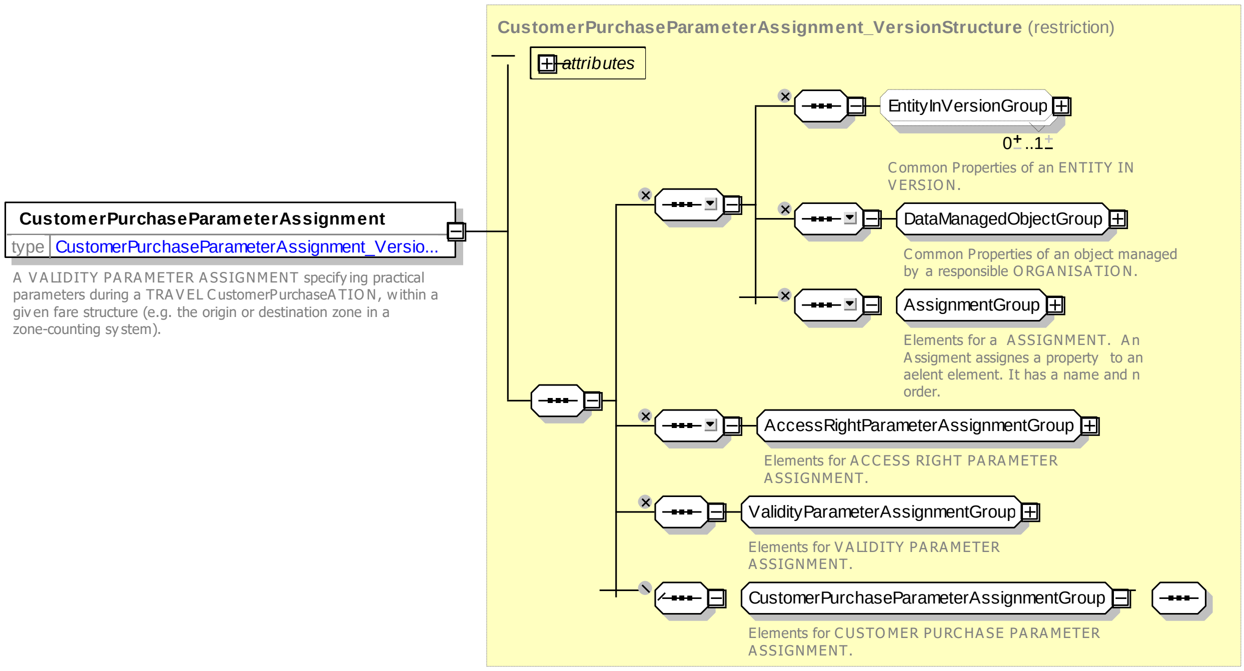


Figure 394 — *CustomerPurchasePackageParameterAssignment* — XSD

8.1.7.4.1 CustomerPurchasePackagePrice – Model Element

A specialisation of FARE PRICE that defines the price of a CUSTOMER PURCHASE PACKAGE.

Table 353 – CustomerPurchasePackageElementPrice – Element

Classification	Name		Type	Cardinality	Description
::>	::>		*FarePrice*	::>	CUSTOMER PURCHASE PACKAGE PRICE inherits from FARE PRICE
«PK»	***id***		*CustomerPurchasePackagePriceIdType*	1:1	Identifier of CUSTOMER PURCHASE PACKAGE PRICE.
	CHOICE				
«FK»	a	***CustomerPurchasePackageRef***	*CustomerPurchasePackagRef*	1:0	Reference to a CUSTOMER PURCHASE PACKAGE.
«FK»	***b***	***CustomerPurchasePackageElementRef***	*CustomerPurchasePackagElementRef*	1:0	Reference to a CUSTOMER PURCHASE PACKAGE ELEMENT.

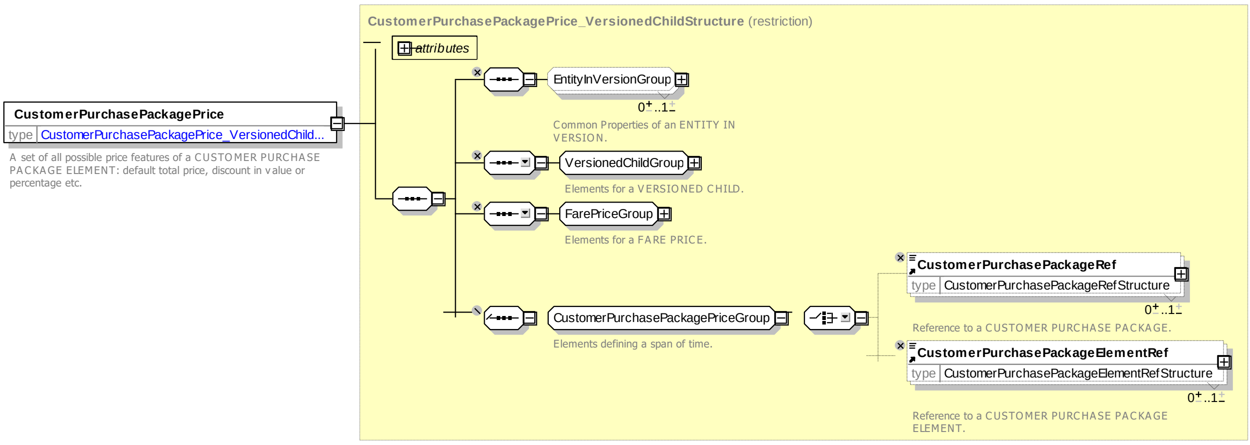


Figure 395 — CustomerPurchasePackagetPrice — XSD

8.1.7.5 TravelDocument – Model Element

A particular physical support (ticket, card, etc.) to be held by a customer, allowing the right to travel or to consume joint-services, to proof a payment (including possible discount rights), to store a subset of the CONTRACT liabilities or a combination of those.

The

Table 354 – 8.2.7.5.1 TravelDocument – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	TRAVEL DOCUMENT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*TravelDocumentIdType*	1:1	Identifier of TRAVEL DOCUMENT.
	Name	*MultilingualString*	0:1	Name of TRAVEL DOCUMENT.
	Description	*MultilingualString*	0:1	Description of TRAVEL DOCUMENT.
	PrivateCode	*PrivateCodeStructure*	0:1	External identifier of TRAVEL DOCUMENT.
«FK»	***TypeOfTravelDocumentRef***	*TypeOfTravelDocumentRef*	0:1	Reference to a TYPE OF TRAVEL DOCUMENT.
«FK»	***Customer-Purchase-PackageRef***	*CustomerPurchase-PackageRef*	0:1	Reference to a CUSTOMER PURCHASE PACKAGE.
«enum»	***MarkedAs***	*MarkedAsEnum*	0:1	Usage Status of the TRAVEL DOCUMENT. See allowed values above. +v1.1

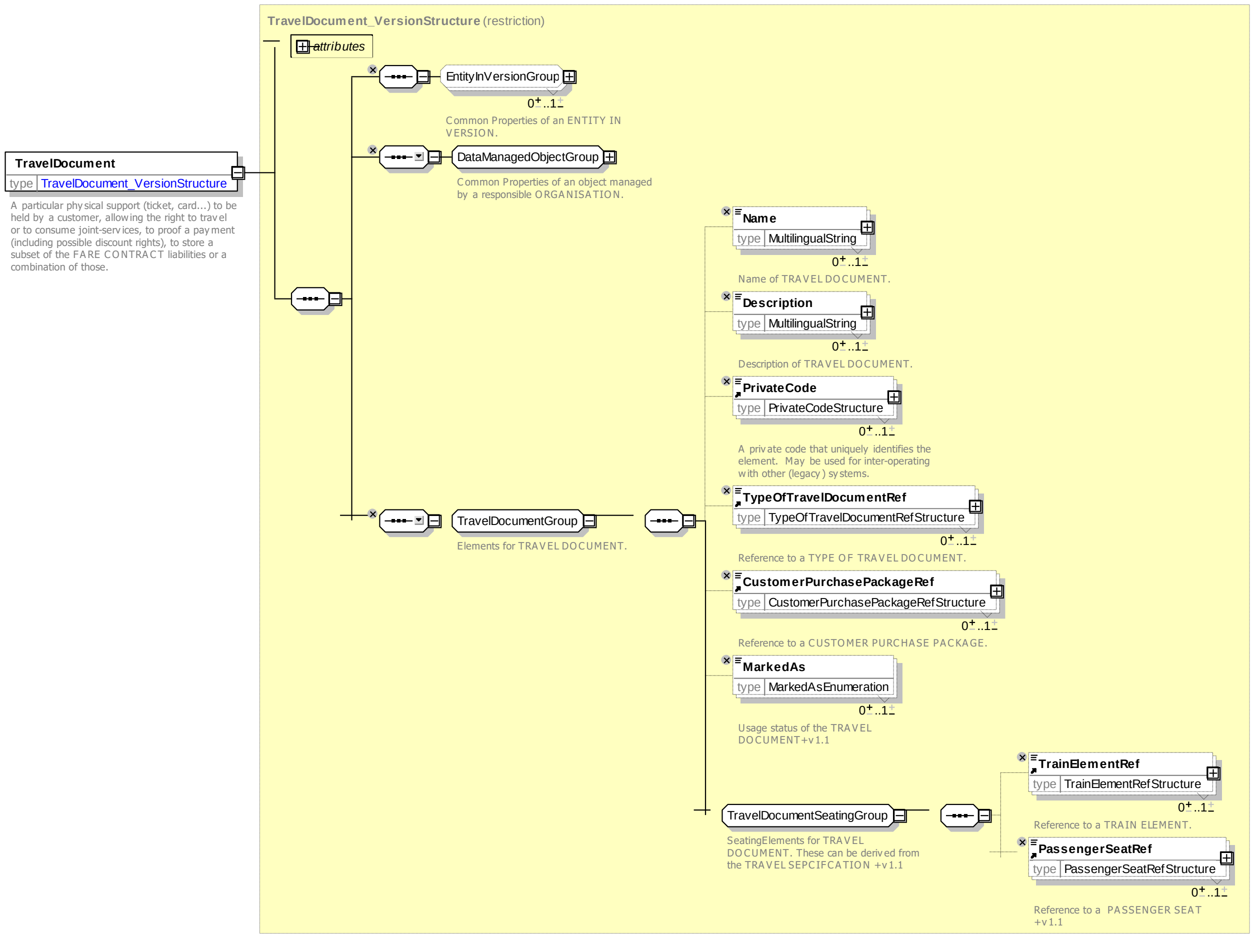


Figure 396 — 8.2.7.7 TravelDocument — XSD

8.1.7.6 TravelDocumentSecurityListing – Model Element

The presence of a TRAVEL DOCUMENT on a SECURITY LIST.

Table 355 – 8.2.7.5.1 8.2.7.7 TravelDocumentSecurityListing – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*SecurityListing*	::>	TRAVEL DOCUMENT SECURITY LISTING inherits from SECURITY LISTING.
«PK»	***id***	*TravelDocumentSecurityListingIdType*	1:1	Identifier of a TRAVEL DOCUMENT SECURITY LISTING.
«FK»	***TravelDocumentRef***	*TravelDocumentRef*	1:1	TRAVEL DOCUMENT that is on SECURITY LIST.

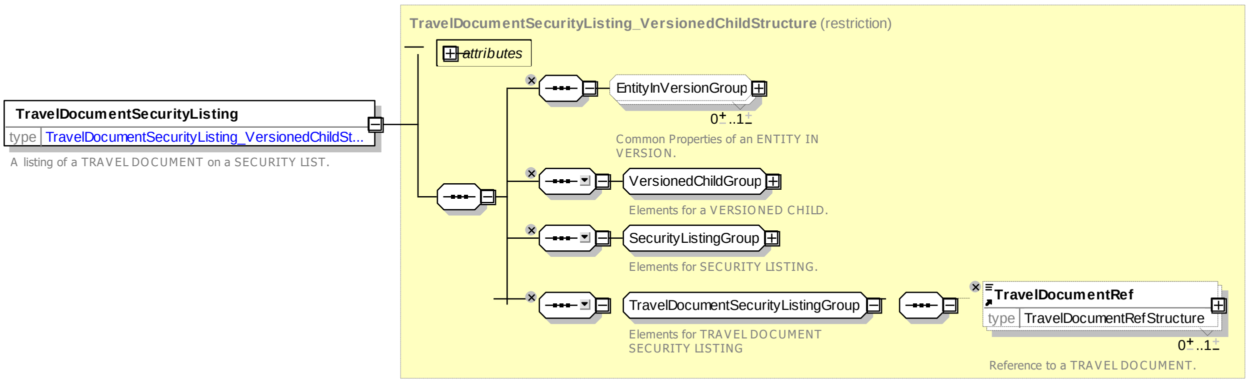


Figure 397 — 8.2.7.7 TravelDocumentSecurityListing — XSD

8.1.7.1 Travel Document – XML examples

8.1.7.1.1 Types of Travel Document – XML Example of document type definitions

The following code fragment defines a number of TYPEs OF TRAVEL DOCUMENT.

For EXAMPLE:

```
<ValueSet version="any" id="cdla:Types_of_TravelDocument" classOfValues="TypeOfTravelDocument">
    <Name>Types of Travel Document </Name>
    <values>
        <TypeOfTravelDocument version="any" id="tfl:paper_ticket">
            <Name>Ticket on paper</Name>
            <MediaType>paperTicket</MediaType>
            <MachineReadable>magneticStrip</MachineReadable>
        </TypeOfTravelDocument>
        <TypeOfTravelDocument version="any" id="tfl:membershipCard">
            <Name>Membership Travel Card to show eligibility </Name>
            <IsCard>true</IsCard>
            <IsSmartCard>false</IsSmartCard>
            <MediaType>card</MediaType>
            <MachineReadable>none</MachineReadable>
        </TypeOfTravelDocument>
        <TypeOfTravelDocument version="any" id="tfl:smartCard">
            <Name>Smart Card</Name>
            <IsCard>true</IsCard>
            <IsSmartCard>true</IsSmartCard>
            <HasPhoto>false</HasPhoto>
            <MediaType>smartCard</MediaType>
            <MachineReadable>nfc</MachineReadable>
        </TypeOfTravelDocument>
        <TypeOfTravelDocument version="any" id="tfl:smartPhotoCard">
            <Name>Smart Card</Name>
            <IsCard>true</IsCard>
            <IsSmartCard>true</IsSmartCard>
            <HasPhoto>true</HasPhoto>
            <MediaType>smartCard</MediaType>
            <MachineReadable>nfc</MachineReadable>
        </TypeOfTravelDocument>
        <TypeOfTravelDocument version="any" id="tfl:nfcPaymentCard">
            <Name>Smart Card</Name>
            <IsCard>true</IsCard>
            <IsSmartCard>true</IsSmartCard>
            <MediaType>smartCard</MediaType>
            <MachineReadable>nfc</MachineReadable>
        </TypeOfTravelDocument>
    </values>
</ValueSet>
```

8.1.7.1.2 Travel Document – XML Example of Document definitions

The following code fragment shows two TRAVEL DOCUMENTs, one for a paper ticket, one for a smart card.

For EXAMPLE:

```xml
<travelDocuments>
    <TravelDocument version="any" id="tfl-t:Ticket67567">
        <Name>Ticket </Name>
        <TypeOfTravelDocumentRef version="any" ref="tfl:paper_ticket"/>
    </TravelDocument>
    <Name>Oyster </Name>

    <TravelDocument version="any" id="tfl:OysterPhotoCard987564">
        <Name>Oyster </Name>
        <TypeOfTravelDocumentRef version="any" ref="tfl:smartPhotoCard"/>
    </TravelDocument>
</travelDocuments>
```

8.1.7.1.3 Travel Document – XML Example of Sales Offer Package

The following code fragment shows a reference to a TYPE OF TRAVEL DOCUMENT by a SALES OFFER PACKAGE.

For EXAMPLE:

```xml
<SalesOfferPackage version="any" id="tfl:TravelCard_on_Oyster">
    <Name>TravelCard on Oyster</Name>
    <Description>Loaded onto card?</Description>
    <ConditionSummary>
        <ProvidesCard>false</ProvidesCard>
        <GoesOnCard>true</GoesOnCard>
        <IsRefundable>true</IsRefundable>
    </ConditionSummary>
    <salesOfferPackageElements>
        <SalesOfferPackageElement version="any" id="tfl:TravelCard_on_Oyster@TravelCard">
            <TypeOfTravelDocumentRef version="any" ref="tfl:smartCard"/>
            <PreassignedFareProductRef version="any" ref="tfl:TravelCard_on_Oyster"/>
        </SalesOfferPackageElement>
    </salesOfferPackageElements>
</SalesOfferPackage>
```

Annex A
(normative)

Extensions to NeTEx Part1 & 2

Extensions to the Part1 & Part2 Model listed in the NeTEx Part3 version Specification have now been consolidated into the NeTEx Part1 & Part 2 documents for NeTEx 1.1.

Annex B
(informative)

ERA – TAP TSI annexes B1, B2 and B3 mapping

The NeTEx Part3 model is intended to support a full mapping of the TAP TSI models for rail fare date.

- B1 (NRT Fares) – non reserved standard fares.
- B2 (IRT fares) - reserved standard fares.
- B3 Special fares

The Mappings to B1, B2 and B3 are described as separate documents. A brief summary is provided here.

B.1 Summary of mapping of B1 (NRT) fares

	Tap	Name	NeTEx
B.1	**TCVG**	Station list (Gare)	FARE SCHEDULED STOP POINT
B.1	**TCVC**	Carrier	OPERATOR
B.1	**TCVS**	Series	FARE STRUCTURE ELEMENT + GENERIC PARAMETER ASSIGNMENT + DISTANCE MATRIX ELEMENT + SERIES CONSTRAINT
B.1	**TCVM**	SeriesInfo	NOTICE + DELIVERY VARIANT
B.1	**TCVT**	Product Table (Trains)	FARE PRODUCT
B.1	**TCVO**	Product Offer	SUPPLEMENT PRODUCT
B.1	**TCVP**	Fare Table	TARIFF + GENERIC PARAMETER ASSIGNMENT
B.1	**TCVP-H**	Distance Based	STANDARD FARE TABLE
B.1	**TCVP-I**	Route Based	STANDARD FARE TABLE
B.1	**TCVP-J**	Set Based	STANDARD FARE TABLE + USAGE PARAMETER
Code List	**B.1.1.**	Type Of Fare	TYPE OF TARIFF
Code List	**B.1.3.**	Border Point	BORDER POINT

B.2 Summary of mapping of B2 (IRT) fares

Tap	Name	NeTEx
B2	**Tariffs**	SALES OFFER PACKAGE + FARE PRODUCT + (ACCESS RIGHT PARAMETER ASSIGNMENT → (GROUP TICKET, USER PROFILE, MINIMUM STAY, PURCHASE WINDOW, EXCHANGING) + AVAILABILITY CONDITION + ALTERNATIVE NAMES
B2	**Range**	GROUP OF SALES OFFER PACKAGES
B2	**Cards Memo**	NOTICE + DELIVERY VARIANT
B2	**Exclusion**	ACCESS RIGHT PARAMETER ASSIGNMENT + AVAILABILITY CONDITION
B2	**Sales Conditions**	DISTRIBUTION ASSIGNMENT
B2	**After Sales**	EXCHANGING
B2	**Price**	DISTANCE MATRIX ELEMENT + DISTANCE MATRIX PRICE (and CELL + PRICE)
B2	**Zone**	TARIFF ZONE
B2	**Grouped OD**	GROUP OF DISTANCE MATRIX ELEMENTs
B2	**Name Cards Memo**	SALES NOTICE ASSIGNMENT
B2	**Distribution**	DISTRIBUTION CHANNEL
B2	**Combinations**	SALES OFFER PACKAGE SUBSTITUTION
B5.1	**Train Category (B2.3 /B.4.7009)**	TYPE OF PRODUCT CATEGORY
B5.1	**Passenger Type (B.2.4 / B.4.5261)**	TYPE OF CONCESSION
B5.1	**Tariff Code (B.2.2 / B.5.42)**	TYPE OF TARIFF

B.3 Summary of mapping of B3 (Special) fares

Tap	Tap	Name	NeTEx
B3	**OFAT**	Offer Authorisation	

BF	**OFOF**	Offer	SALES OFFER PACKAGE + RESERVING + VALIDITY CONDITION + ALTERNATIVE NAME + FARE PRODUCT/SUPPLEMENT PRODUCT
B3	**OFCO**	Conditions of offer	SALES OFFER PACKAGE SUMMARY + [GENERIC PARAMETER ASSIGNMENT → USER PROFILE + GROUP TICKET + TRANSFERABILITY + USAGE VALIDITY PERIOD + EXCHANGING + TRANSFERABLE + MINIMUM STAY + PURCHASE WINDOW] + USAGE PARAMETER PRICE + DISCOUNTING RULE + ROUNDING + AVAILABILITY CONDITION
B3	**OFFC**	Fare table per class	FARE TABLE + TARIFF + FARE PRICE + LIMITING RULE
B3	**OFTP**	Type of passenger	USER PROFIL E + ALTERNATIVE NAME + FARE PRODUCT
B3	**OFPA**	Passenger	USER PROFILE + COMPANION PROFILE+ FARE PRICE + ROUNDING + DISCOUNTING RULE + FARE PRODUCT
B3	**OFNP**	Number of passengers	GROUP TICKET + USER PROFILE + GENERIC PARAMETER ASSIGNMENT + FARE PRODUCT?
B3	**OFRE**	Type of discount	FARE PRODUCT + ALTERNATIVE NAMEs
B3	**OFAR**	Additional discount	USER PROFILE + CELL + FARE PRICE + ROUNDING + DISCOUNTING RULE
B3	**OFFP**	Companion	USER PROFILE + CELL + FARE PRICE + ROUNDING + FARE PRODUCT
B3 (B1)	**OFSE**	Series	GENERIC PARAMETER ASSIGNMENT + SERIES CONSTRAINT
B3	**OFTR**	Trains	GENERIC PARAMETER ASSIGNMENT + AVAILABILITY CONIDITION + TRAIN NUMBER
B3	**OFID**	Blackout periods	GENERIC PARAMETER ASSIGNMENT + AVAILABILTY CONDITION
B3	**OFGB**	After sales	GENERIC PARAMETER ASSIGNMENT + EXCHANGING + USAGE PARAMETER PRICE + DISCOUNTING RULE + LIMITING RULE
B3	**OFME**	Memo	NOTICE + DELIVERY VARIANT
B3	**OFFS**	Fare and supplement	GENERIC PARAMETER ASSIGNMENT + SERVICE JOURNEY + FACILITY SET
B3	**OFRT**	Reservations (reservation table)	RESERVING + CELL + FARE PRICE + (GENERIC PARAMETER ASSIGNMENT --> CLASS OF USE)
B5.1		Facility codes	FACILITY SET + FACILITY
B5.1	**B.2.3**	Train Category (B2.3 /B.4.7009)	TYPE OF PRODUCT CATEGORY
(B1)	**TCVP**	Fare table explanations (Prix)	NOTICE + NOTICE ASSIGNMENT
(B1)		H-Distance-based fare tables	STANDARD FARE TABLE
(B1)		I- Route-based fare tables	STANDARD FARE TABLE
(B1)		Set fare tables	STANDARD FARE TABLE
B5.1	**B.2.4**	Passenger Type (B.2.4 / B.4.5261)	TYPE OF CONCESSION
B5.1	**B.2.2**	Tariff Code (B.2.2 / B.5.42)	TYPE OF TARIFF

Annex C
(informative)

NeTEx Passenger Information Query model

The PI QUERY Model is provided as an informative appendix to indicate how the NeTEx data elements can related to APIs and web services that deliver transport data to the end user. The appendix does not seek to define a definitive set of services, nor to set out an exchange format, but merely to give guidance as to which NeTEx elements are relevant for typical passenger information queries, and to identify useful query criteria. The essential PI QUERY model (or a relevant subset) may be implemented in a wide variety of concrete services using different service technologies(http, CORBA, etc) and renderings (XML, JSON etc) and syntaxes.

C.1 PiRequest

C.1.1 PI Request dependencies

NeTEx PI REQUEST model is modularised into a number of submodels defined as UML packages, these in turn depend on Part3, Part2 and Part1 packages.

— The REQUEST Package describe PI Queries.

— The PASSENGER TRIP PACKAGE describes the Passenger Trips returned by.

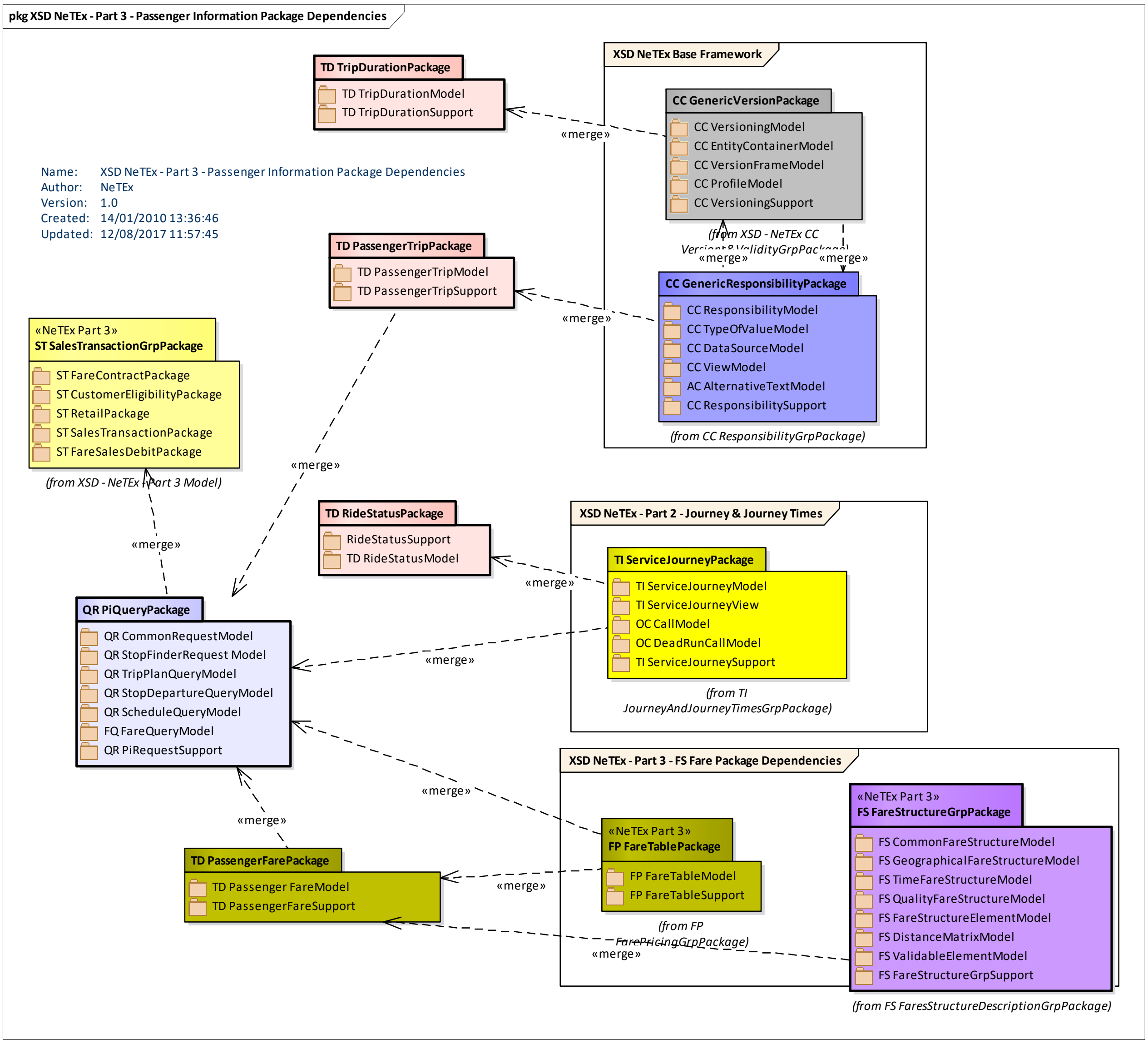


Figure 398 — PiRequest Package Dependencies

The following diagram gives an overview of the dependencies between the PI REQUEST models and NeTEx Part3.

— The COMMON REQUEST Model describe common query constructs used by all the different Request types.

— The STOP FINDER REQUEST Model indicates the elements relevant for a query to find public transport stops.

— The TRIP PLANNER REQUEST Model indicates the elements relevant for a query to find public transport journeys between given locations.

— The STOP DEPARTURE REQUEST Model indicates the elements relevant for a query to find departures at a stop.

— The SCHEDULE REQUEST Model indicates the elements relevant for a query to find public transport timetables.

— The FARE REQUEST Model indicates the elements relevant for queries to find the fares for public transport journeys between given locations.

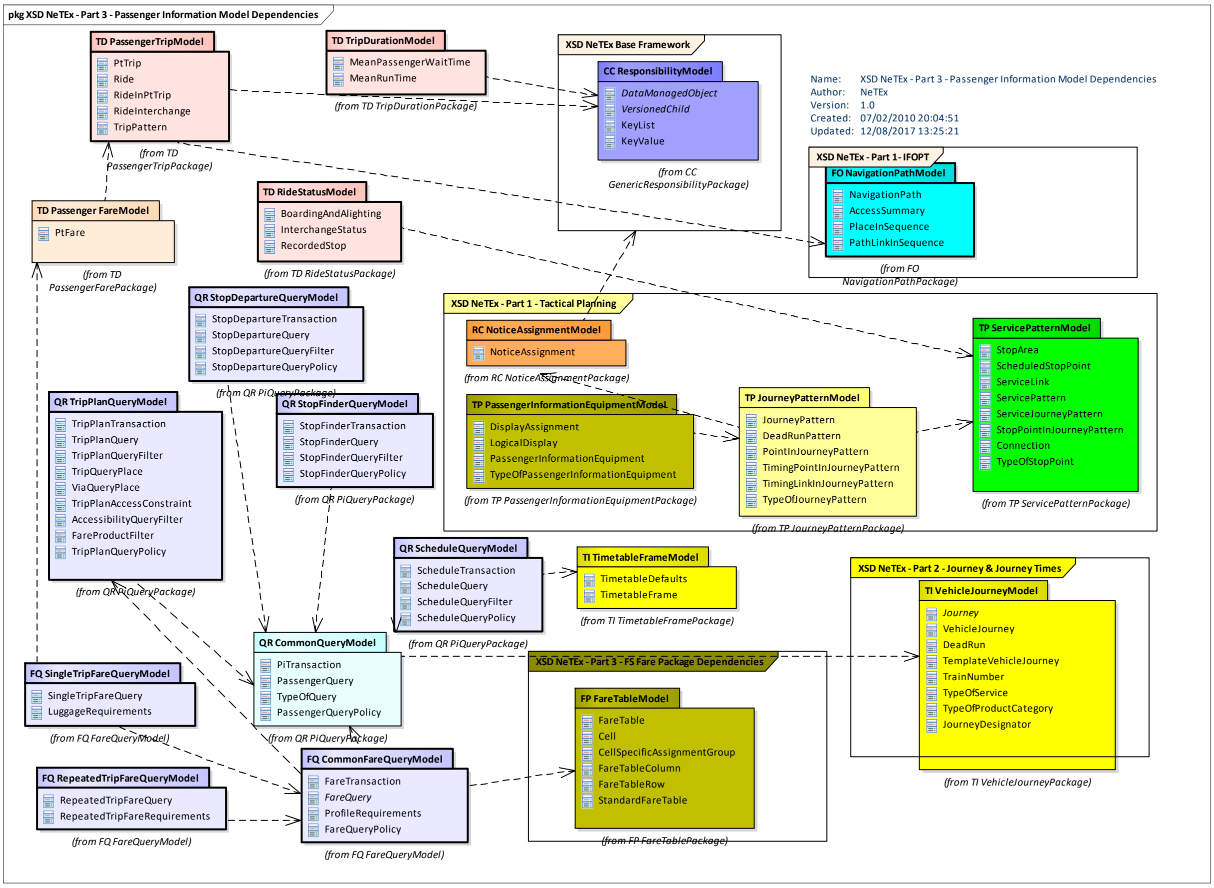


Figure 399 — PiRequest Model Dependencies

C.1.2 Pi Request

This model is useful to specifiy information services that access passenger information through a service API, for example journey planners, fare queries, etc,

C.1.2.1 Common PI Request

Each PT DELIVERY (corresponding to an http request or remote procedure call) may contain one or more PASSENGER REQUESTs. There are a number of different types of concrete Delivery (STOP FINDER DELIVERY, TRIP PLAN DELIVERY, etc.), each with its own corresponding query type (STOP FINDER REQUEST, TRIP PLAN, etc.) and a specific result type.

C.1.2.1.1 PiRequest – Conceptual MODEL- Overview

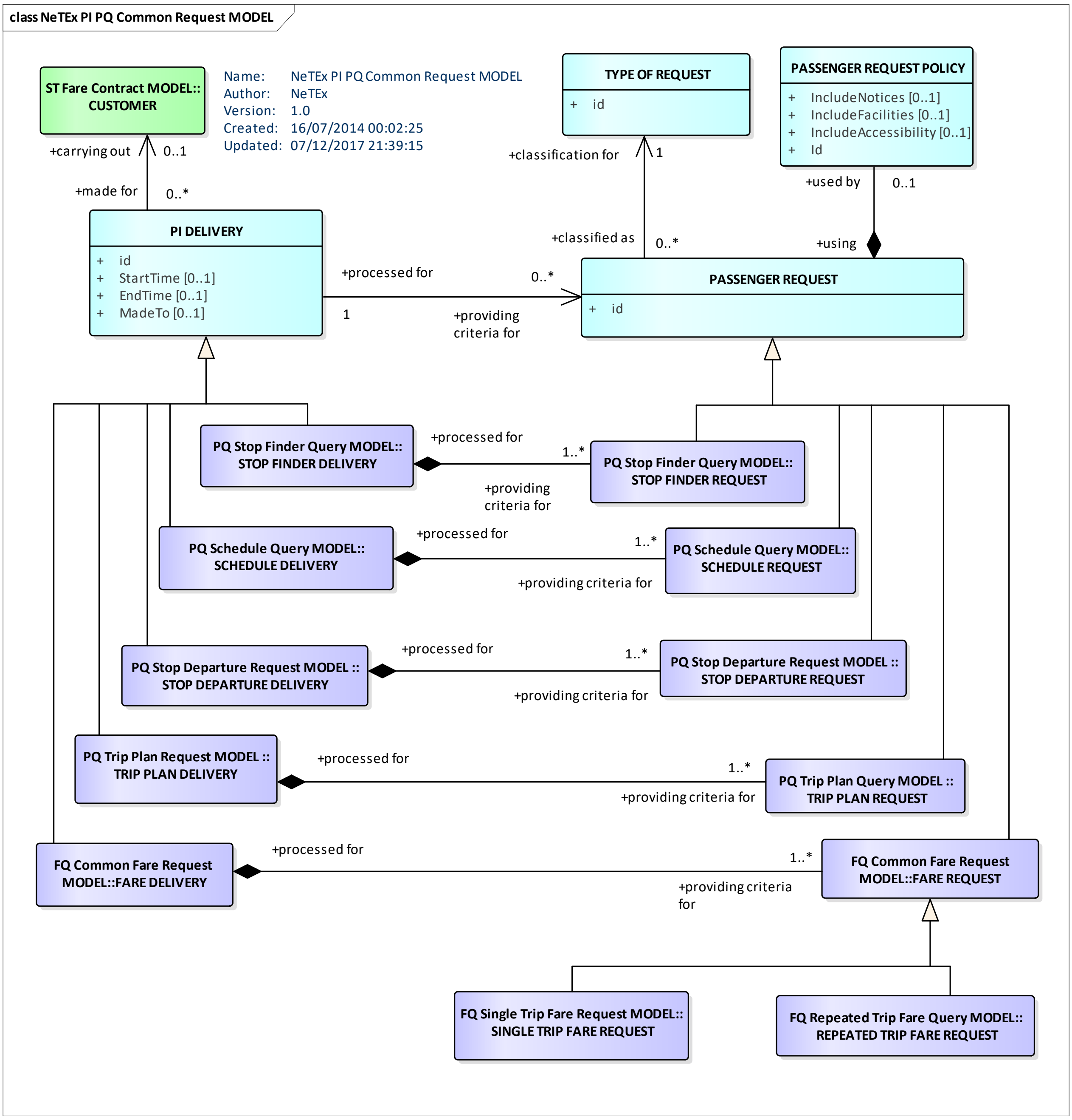


Figure 400 — PiRequest – Conceptual Model

C.1.2.1.2 PiRequest – Physical model – Overview

The following figure introduces the physical model for PI REQUESTs.

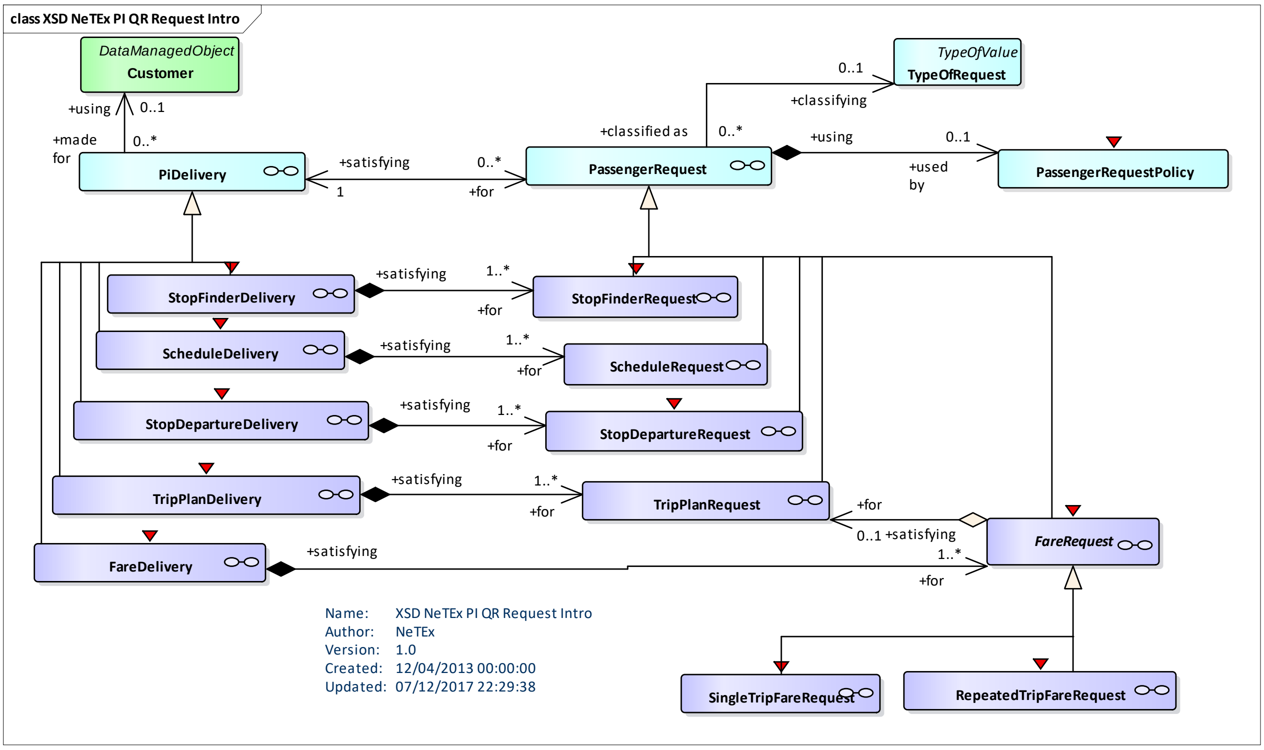


Figure 401 — PiRequest – Physical Model: Overview

C.1.2.1.3 PiRequest – Physical model – Details

The following figure shows the physical model for PI REQUESTs.

The PT DELIVERY provides a container for the elements needed to make a query or queries, and logs attributes useful for audit such as start time and end time. The REQUEST specifies the actual search criteria.

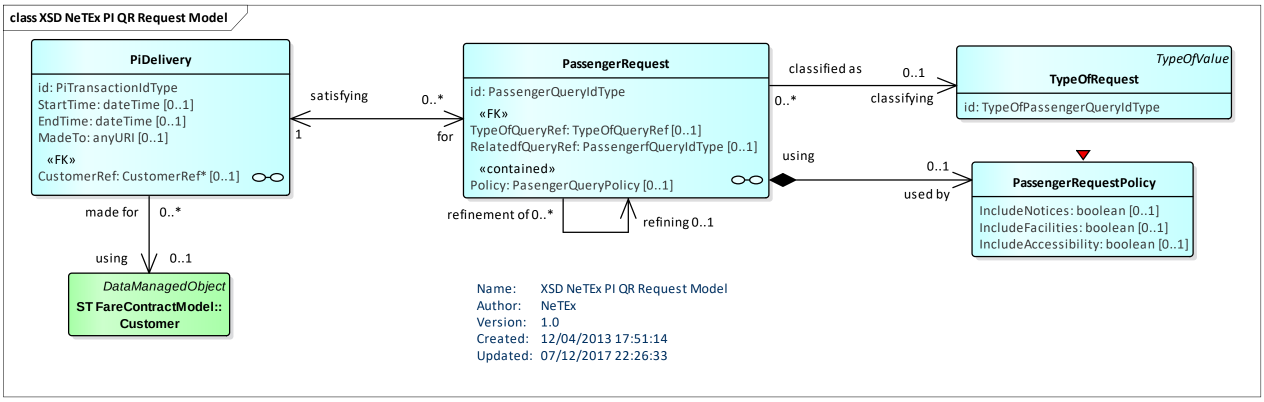


Figure 402 — PiRequest – Physical Model: Details

C.1.2.1.4 PiRequest – Attributes and XSD

C.1.2.1.4.1 PiDelivery – Model Element

A connection of a passenger to the operator information system, directly or via an employee, including one or several queries.

Table 356 – *PiDelivery* – Element

Classifi-cation	Name	Type	Cardinality	Description
	id	*PiDeliveryIdType*	1:1	Identifier of PI DELIVERY.
	StartTime	*xsd:dateTime*	0:1	Start time for Delivery.
	EndTime	*xsd:dateTime*	0:1	Time when Delivery was completed.
	MadeTo	*xsd:anyURI*	0:1	Uri to which Delivery was made.
	UserProfileRef	*UserProfileRef*	0:1	Reference to USER PROFILE of user making query.
	CustomerRef	*CustomerRef*	0:1	Reference to CUSTOMER making query.

C.1.2.1.4.2 PassengerRequest – Model Element

A request for a specific information on public transport, expressed during a PI DELIVERY. It is specialized by different types of specific query such as TRIP PLAN REQUEST, FARE REQUEST etc.

Table 357 – *PassengerRequest* – Element

Classifi-cation	Name	Type	Cardin ality	Description
	id	*PassengerRequestIdType*	1:1	Identifier of a PASSENGER REQUEST.
«FK»	***TypeOfRequest-Ref***	*TypeOfRequestRef*	0:1	Type of PASSENGER REQUEST.
«FK»	***RelatedRequest-Ref***	*PassengerRequestIdType*	0:1	If refinement to previous query, identifier of related query.

C.1.2.1.4.3 PassengerRequestPolicy – Model Element

Optimisation criteria to be used to when computing and decorating the query results.

Table 358 – *PassengerRequestPolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
	IncludeNotices	*xsd:boolean*	0:1	Whether results should include NOTICEs associated with result elements.

	IncludeFacilities	*xsd:boolean*	0:1	Whether results should include information about facilities of result elements.
	IncludeAccessibility	*xsd:boolean*	0:1	Whether results should include ACCESSIBILITY ASSESSMENTs associated with result elements.

C.1.2.1.4.4 TypeOfRequest – Model Element

A classification of PASSENGER QUERies.

Table 359 – *PassengerRequestType* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF PASSENGER REQUEST inherits from TYPE OF VALUE.
	id	*TypeOfPassenger-RequestIdType*	1:1	Identifier of a TYPE OF PASSENGER REQUEST.

C.1.2.2 Stop Finder Request

A STOP FINDER REQUEST returns a list of SCHEDULED STOP POINTs of a given MODE that correspond to a given point or zone location. Location may also be specified using a stop name or a post code.

Each STOP FINDER DELIVERY (corresponding to an http request or remote procedure call) may contain one or more STOP FINDER REQUESTs.)

C.1.2.2.1 Stop Finder Request – Conceptual MODEL

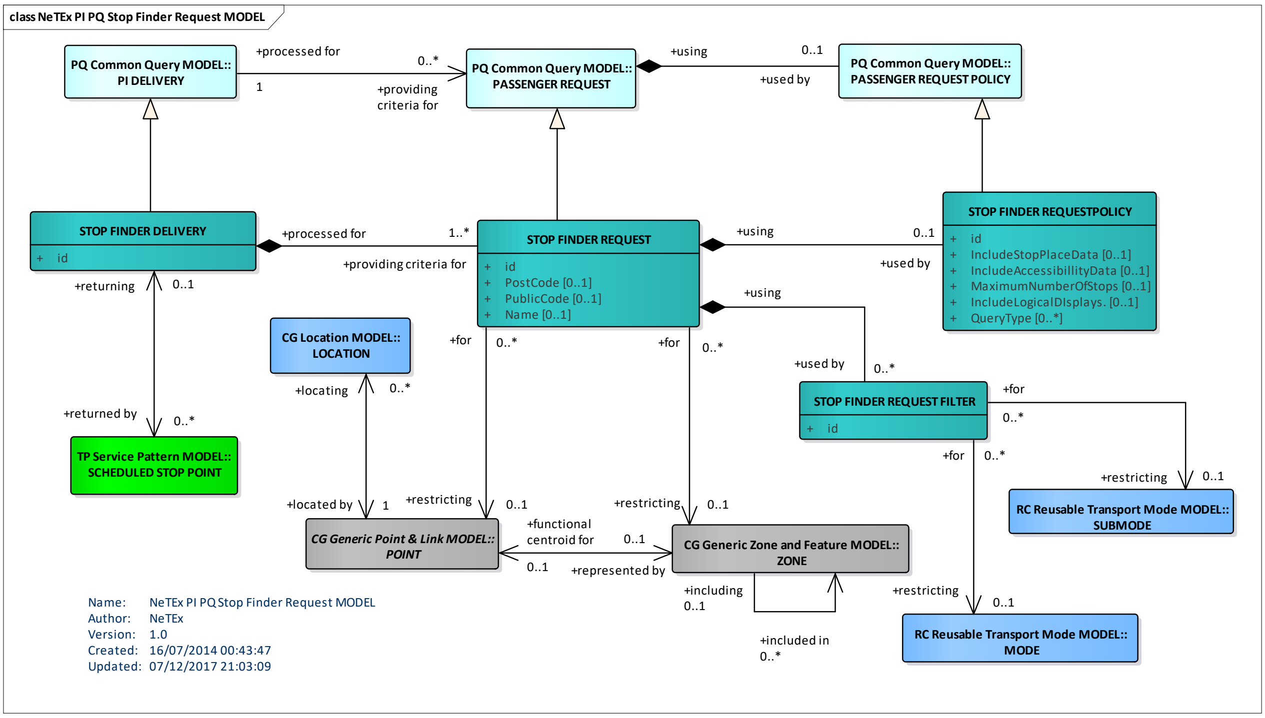


Figure 403 — Stop Finder Request– Conceptual Model

C.1.2.2.2 Stop Finder Request – Physical model – Overview

The following figure introduces the physical model for a STOP FINDER REQUEST.

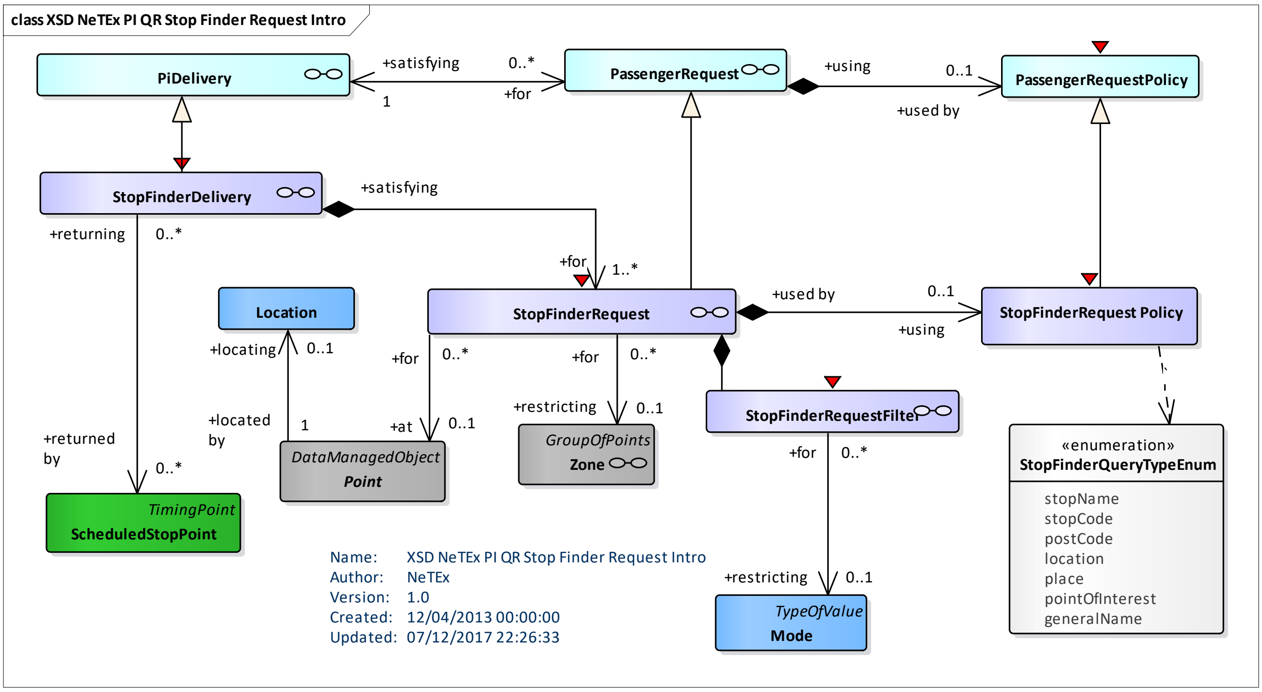


Figure 404 — Stop Finder Request– Physical Model: Overview

C.1.2.2.3 Stop Finder Request – Physical model – Details

The following figure shows the physical model for STOP FINDER REQUEST.

A ZONE may be a STOP, POINT of INTEREST, PARKING or a TOPOGRAPHIC PLACE (See NeTEx part1).

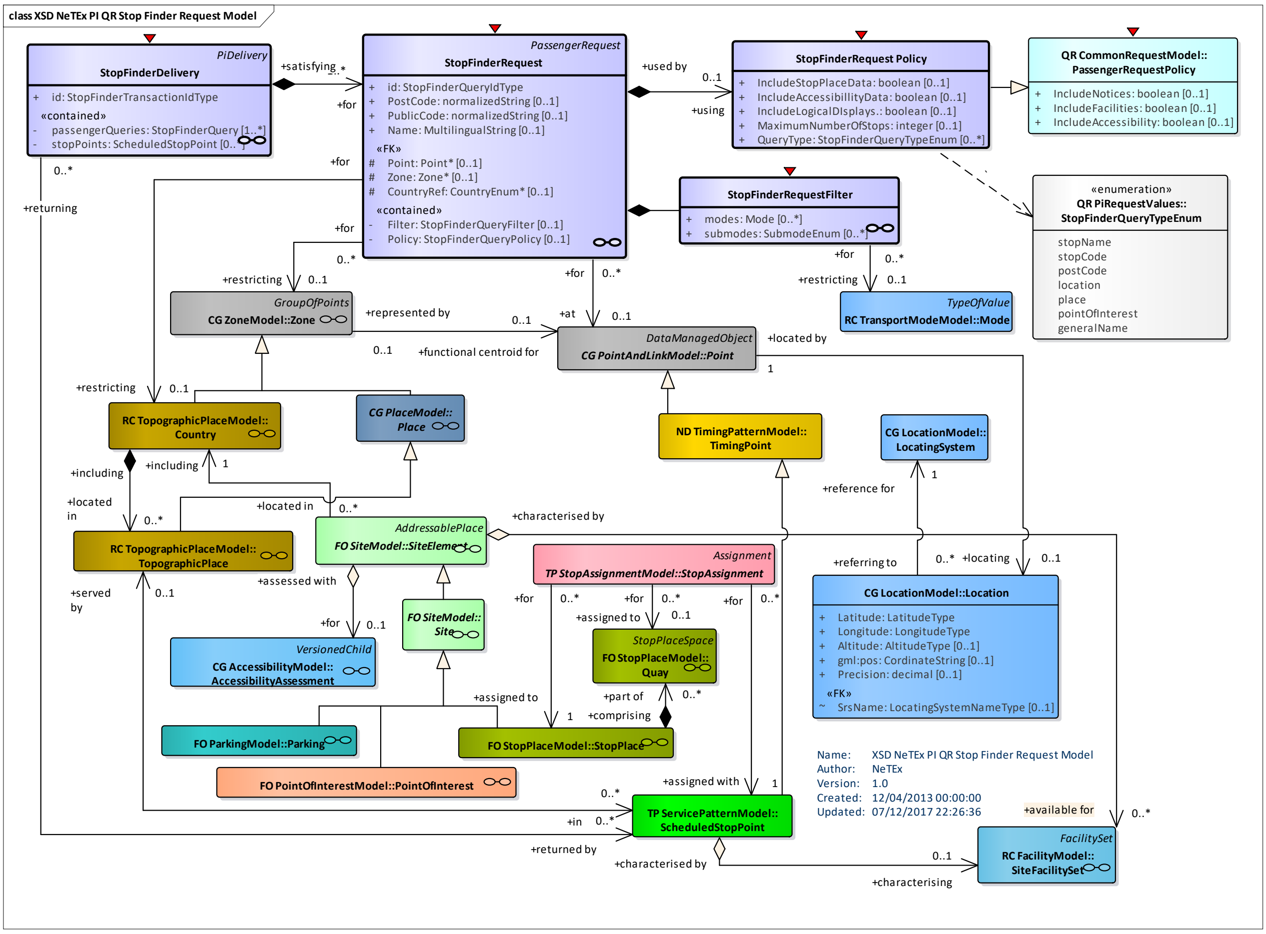


Figure 405 — Stop Finder Request – Physical Model: Details

C.1.2.2.4 Stop Finder– Attributes and XSD

C.1.2.2.4.1 StopFinderDelivery – Model Element

A specialization of PI DELIVERY to make one or more STOP FINDER QUERies.

Table 360 – *StopFinderDelivery*– Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PiDelivery*	::>	STOP FINDER DELIVERY inherits from PI DELIVERY.
	id	*StopFinderDelivery-IdType*	1:1	Identifier of a STOP FINDER DELIVERY.

«cntd»	***passenger-Queries***	*StopFinderRequest*	1:*	STOP FINDER QUERies used in the Delivery.
«cntd»	***stopPoints***	*ScheduledStopPoint*	0:*	SCHEDULED STOP POINTs returned in response to query.

C.1.2.2.4.2 StopFinderRequest – Model Element

A PASSENGER REQUEST to find a stop. Stops matching all of the given search criteria will be returned.

Table 361 – *StopFinderRequest*– Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PiRequest*	::>	STOP FINDER REQUEST inherits from PI REQUEST.
	id	*StopFinderRequestIdType*	1:1	Identifier of a STOP FINDER REQUEST.
«FK»	***Point***	*(Point)*	0:1	POINT for which stops are to be found.
«FK»	***Zone***	*(Zone)*	0:1	ZONE in which to find stops. A ZONE may be as TOPOGRAPHIC PLACE, STOP PLACE, SITE, PARKING, etc.
	PostCode	*xsd:normalizedString*	0:1	Post code to use to find stops.
	PublicCode	*xsd:normalizedString*	0:1	Stop code to use to find stop.
	Name	*MultilingualString*	0:1	Name to use to find stop.
«FK»	***CountryRef***	*CountryEnum*	0:1	Reference to COUNTRY of a TOPOGRAPHIC PLACE. See NeTEx Part1.
«enum»	***RequestType***	*StopFinderRequest-TypeEnum*	0:*	Type of STOP FINDER REQUEST. See allowed values below.
«cntd»	***Filter***	*StopFinderRequestFilter*	0:1	Additional filter parameters.
«cntd»	***Policy***	*StopFinderRequestPolicy*	0:1	Criteria for selecting and decorating stop data.

C.1.2.2.4.3 StopFinderRequestFilter – Model Element

Filter parameters used to limit the results of the query.

Table 362 – *StopFinderRequestFilter* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
	modes	*Mode*	0:*	MODEs of transport to include in query. See Part1
«enum»	***submodes***	*SubmodeEnum*	0:*	Submodes of transport to include in query. See Part1.

C.1.2.2.4.4 StopFinderRequestPolicy – Model Element

Optimisation criteria to be used to when computing and decorating the query results.

Table 363 – *StopFinderRequestPolicy* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerRequest-Policy*	::>	STOP FINDER REQUEST POLICY inherits from PASSENGER REQUEST POLICY.
	Include-StopPlaceData	*xsd:boolean*	0:1	Whether results should include information about STOP PLACEs, QUAYs and BOARDING POSITIONs.
	Include-AccessibilityData	*xsd:boolean*	0:1	Whether results should include accessibility data or stop.
	Include-LogicalDisplays.	*xsd:boolean*	0:1	Whether results should include information about LOGICAL DISPLAYs associated with stop. start and end.
	Maximum-NumberOfStops	*xsd:integer*	0:1	Maximum number of stops to include in results.
«enum»	***RequestType***	*StopFinderRequest-TypeEnum*	0:*	Type of Stop Finder Request.

StopFinderRequestType – Allowed values

The following table shows the allowed values for ***RequestType***. (*StopFinderRequestTypeEnumeration).*

Table 364 – *RequestType* – Allowed values

Value	Description
stopName	Request by stop name.
stopCode	Request by stop code.
postCode	Request by post code.
location	Request by coordinates.
place	Request by place name.
pointOfInterest	Request by point of interest name.
generalName	Request by name; might be stop or place.

C.1.2.3 Trip Plan Request

A TRIP PLAN REQUEST returns a list of PT TRIPs that satisfy the given search criteria. Each PT TRIP may involve multiple trip legs.

Each TRIP PLAN DELIVERY (corresponding to an http request or remote procedure call) may contain one or more TRIP PLAN REQUESTs.

C.1.2.3.1 Trip Plan Request – Conceptual MODEL

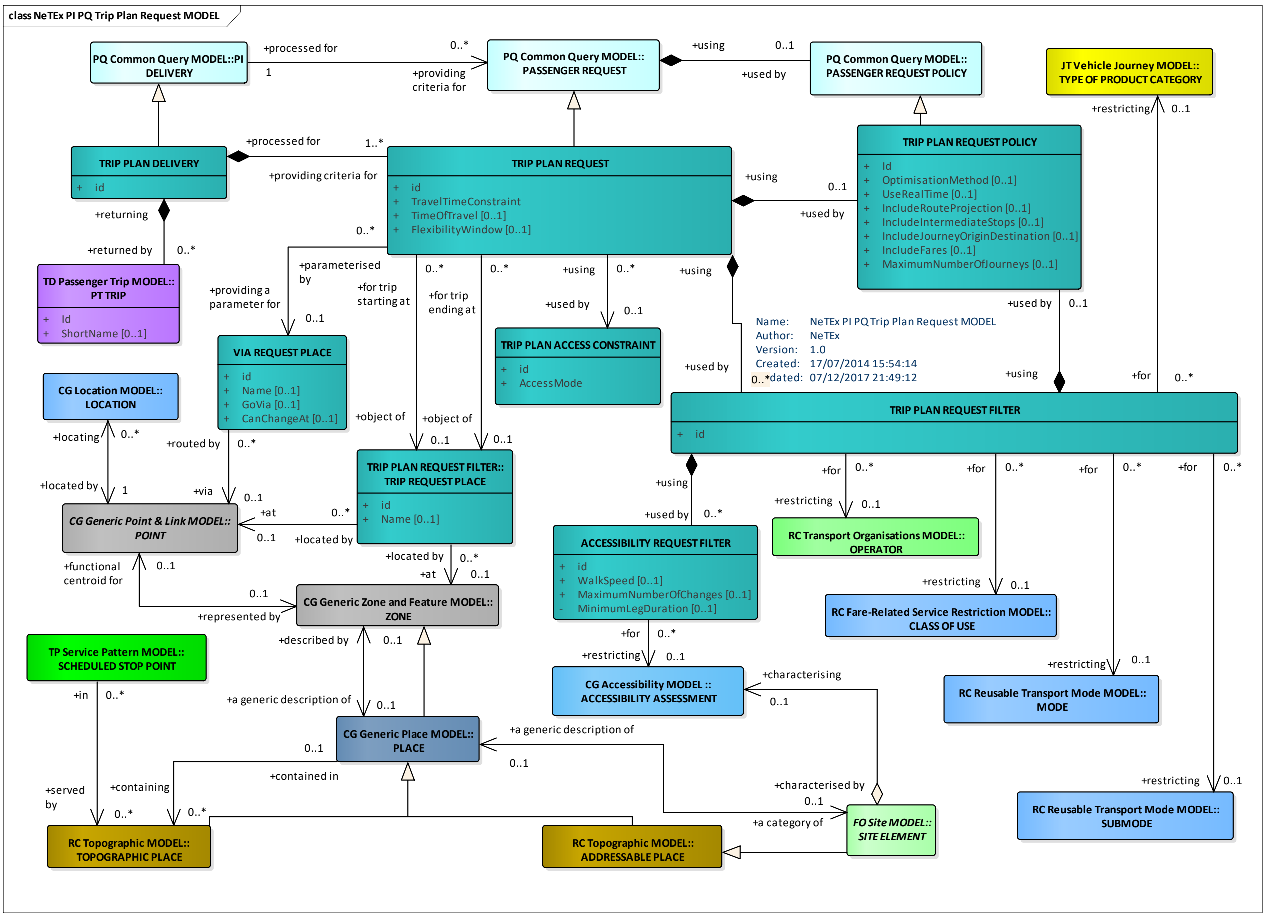


Figure 406 — Trip Plan Request– Conceptual Model

The fundamental criteria (Origin and destination TRIP REQUEST PLACE, time of travel etc) must be supplied. Other additional criteria can also be specified.

- VIA REQUEST PLACE can be used to constrain the journey to specific via points.

- ACCESS CONSTRAINT can be used to limit the access modes and time for the access leg to reach the Public Transport.

- TRIP PLAN REQUEST FILTER can be used to set selection criteria for the journey plan such as MODEs, OPERATORs, PRODUCT CATEGORies.

- FARE PRODUCT FILTER can be used to set filters to select only journey for which specific types of product are available.

- ACCESSIBILITY REQUEST FILTER can be used to set additional requirements for mobility.

- TRIP PLAN REQUEST POLICY species parameters controlling the way the journey plan is computed.

C.1.2.3.2 Trip Plan Request – Physical model – Overview

The following figure introduces the physical model for a TRIP PLAN REQUEST.

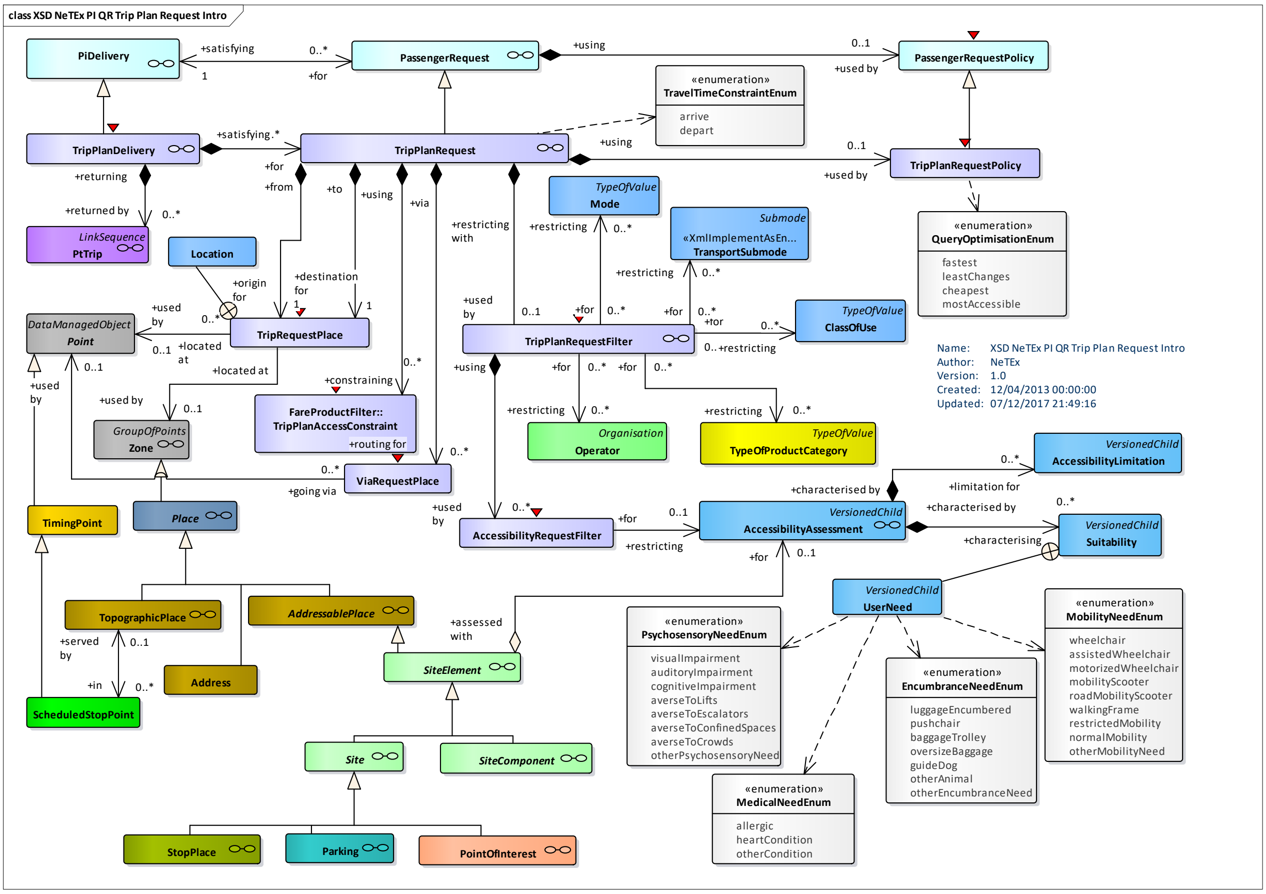


Figure 407 — Trip Plan Request– Physical Model: Overview

C.1.2.3.3 Trip Plan Request – Physical model – Details

The following figure shows the physical model for TRIP PLAN REQUEST.

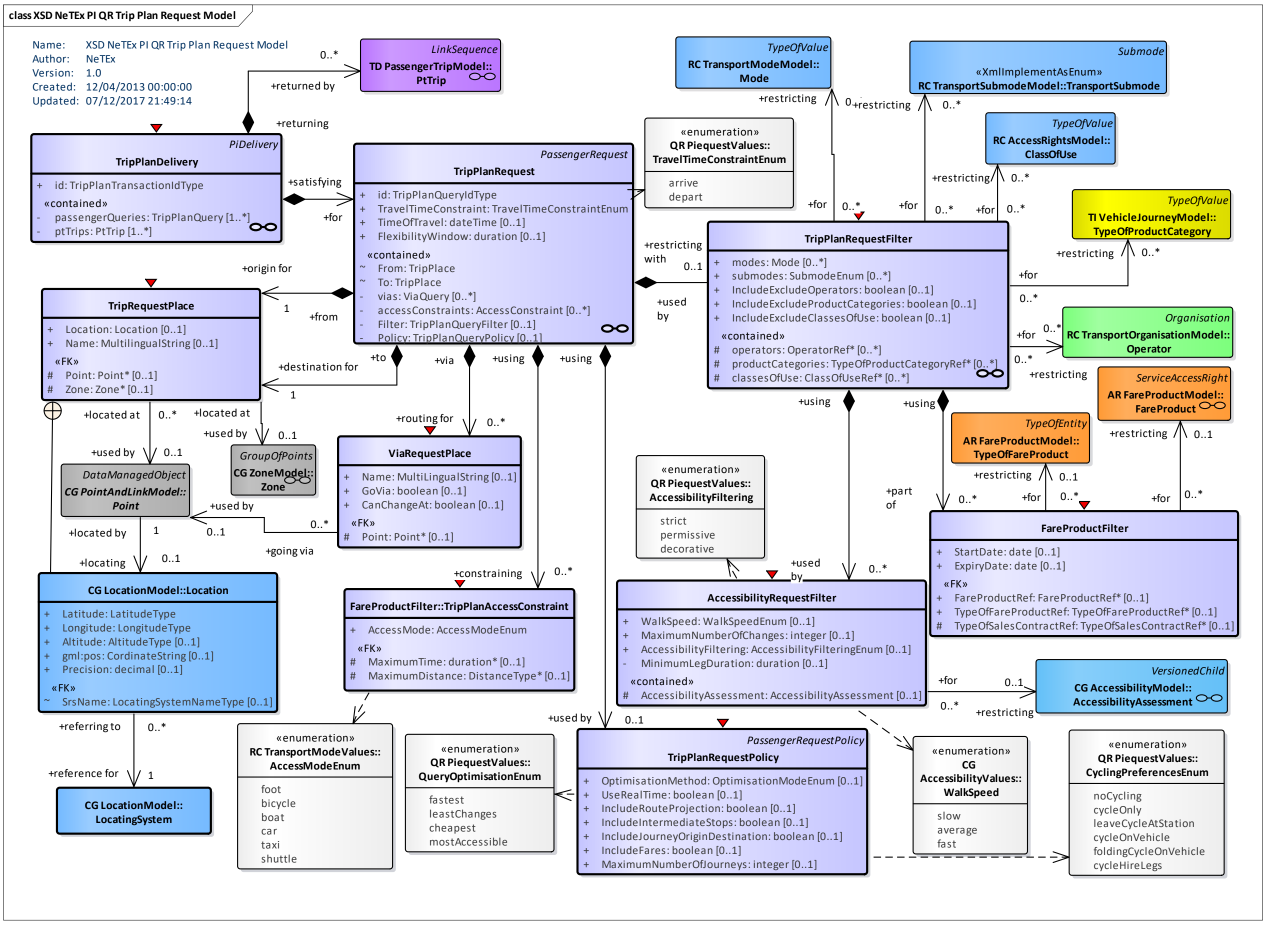


Figure 408 — Trip Plan Request – Physical Model: Details

C.1.2.3.4 Trip Plan Request – Attributes and XSD

C.1.2.3.4.1 TripPlanDelivery – Model Element

A specialization of PI DELIVERY used to make one or more TRIP PLAN QUERies.

Table 365 – *TripPlanDelivery* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	<u>*PiDelivery*</u>	::>	TRIP PLAN DELIVERY inherits from PI DELIVERY
	id	*TripPlanDeliveryIdType*	1:1	Identifier of a TRIP PLAN DELIVERY.
«cntd»	***passenger-Queries***	<u>*TripPlanRequest*</u>	1:*	TRIP PLAN queries used in Delivery.
«cntd»	***ptTrips***	<u>*PtTrip*</u>	1:*	Passenger TRIPs returned in response to query.

C.1.2.3.4.2 TripPlanRequest – Model Element

A PASSENGER REQUEST concerning an optimal trip proposal, according to a specified OPTIMISATION POLICY.

Table 366 – *TripPlanRequest* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PiRequest*	::>	TRIP PLAN REQUEST inherits from PI REQUEST.
	id	*TripPlanRequestIdType*	1:1	Identifier of a TRIP PLAN REQUEST.
«cntd»	***From***	*TripPlace*	1:1	Place at which desired TRIP starts.
«cntd»	***To***	*TripPlace*	1:1	Place at which desired TRIP ends.
«enum»	***TravelTime-Constraint***	*TravelTime-ConstraintEnum*	1:1	Constraint on how to use travel time: to arrival or to departure. See allowed values below.
	TimeOfTravel	*xsd:dateTime*	0:1	Time of travel to use in conjunction with constraint.
	Flexibility-Window	*xsd:duration*	0:1	Flexibility to travel before or after specified time. e.g. '3 days'.
«cntd»	***vias***	*ViaRequest*	0:*	Constrain routing to go via the specified points.
«cntd»	***access-Constraints***	*AccessConstraint*	0:*	To reach a stop using a given access mode (walk, cycle etc) how far/long to constrain suggested journeys.
«cntd»	***Filter***	*TripPlanRequestFilter*	0:1	Additional filter parameters. See ***TripPlan-RequestFilter*** below.
«cntd»	***Policy***	*TripPlanRequestPolicy*	0:1	Additional processing parameters. See below.

TravelTimeConstraint– Allowed values

The following table shows the allowed values for ***TravelTimeConstraint***. (*TravelTime-ConstraintEnumeration).*

Table 367 – *TravelTimeConstraint* – Allowed values

Value	Description
arrive	Plan trip to arrive at specified time.
depart	Plan trip to depart at specified time.

C.1.2.3.5 TripRequestPlace – Model Element

Description of a place to use for origin or destination of trip. If multiple values supplied, e.g. both a ZONE and a name, values are logically ANDed together

Table 368 – *TripRequestPlace* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	***Point***	*(Point)*	0:1	POINT at which desired TRIP starts or ends.
	Location	*Location*	0:1	Coordinates of POINT at which desired TRIP starts or ends.
«FK»	***Zone***	*Zone*	0:1	ZONE at which desired TRIP starts or ends. ZONEs may include TOPOGRAPHICAL PLACEs – see NeTEx PART1.
	Name	*MultilingualString*	0:1	Name of location – may be a PLACE or STOP POINT.
	PublicCode	*PublicCodeType*	0:1	Public code used to identify location.

C.1.2.3.5.1 TripPlanAccessConstraint – Model Element

Parameters limiting the time and nature of the access leg used to reach the PT stop. For example, to be able to specify, '*Walk 5 minutes, cycle 20 minutes' drive 30 minutes'.*

Table 369 – *TripPlanAccessConstraint* – Element

Classifi-cation	Name	Type	Cardinality	Description
«enum»	***AccessMode***	*AccessModeEnum*	1:1	MODE of access for which ACCESS CONSTRAINT applies. See NeTEx Part1.
«FK»	***MaximumTime***	*xsd:duration*	0:1	Maximum duration tolerated for this access mode.
«FK»	***Maximum-Distance***	*DistanceType*	0:1	Maximum distance desired for this access mode.

C.1.2.3.5.2 ViaRequestPlace – Model Element

A routing location used to constrain the journeys returned. Only VEHICLE JOURNEYs whose JOURNEY PATTERNs do or do not pass through the specified points will be returned.

Table 370 – *ViaRequestPlace* – Element

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***PointRef***	*PointRef*	0:1	POINT by which desired TRIP goes/does not go via. See NeTEx Part1.
	Name	*MultilingualString*	0:1	Name of VIA.

	GoVia	*xsd:boolean*	0:1	Whether VIA point is to be taken or avoided ("not via")
	CanChangeAt	*xsd:boolean*	0:1	Whether VIA point can be used to interchange or is forbidden for interchange.

C.1.2.3.5.3 TripPlanRequestFilter – Model Element

Filter parameters used to limit the MODEs, PRODUCT CATEGORies and CLASSes OF USE of the journeys returned.

Table 371 – *TripPlanRequestFilter* – Element

Classifi-cation	Name	Type	Cardin ality	Description
	modes	*Mode*	0:*	MODEs of transport to include in query.
	submodes	*Submode*	0:*	SUBMODEs of transport to include in query.
«cntd»	***operators***	*OperatorRef*	0:*	OPERATOR by which to filter results. See NeTEx Part1.
	IncludeExclude-Operators	*xsd:boolean*	0:1	Whether specified OPERATORs are to be included or excluded.
«cntd»	***productCategories***	*TypeOfProduct-CategoryRef*	0:*	Type of PRODUCT CATEGORY on which to filter results.See NeTEx Part2.
	IncludeExclude-ProductCategories	*xsd:boolean*	0:1	Whether specified PRODUCT CATEGORies are to be included or excluded in results.
«cntd»	***classesOfUse***	*ClassOfUseRef*	0:*	CLASS OF USE on which to filter results. Only journeys which match this class of service will be returned.
	IncludeExclude-ClassesOfUse	*xsd:boolean*	0:1	Whether specified CLASSes OF USE are to be included or excluded in results.

C.1.2.3.5.4 AccessibilityRequestFilter – Model Element

Criteria used to limit the ACCESSIBILITY properties of the journeys returned.

Table 372 – *AccessibilityRequestFilter* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
«cntd»	***Accessibility-Assessment***	*Accessibility-Assessment*	0:1	Default ACCESSIBILITY ASSESSMENT for JOURNEYs in the TIMETABLE, specifying. See NeTEx Part1.
«enum»	***WalkSpeed***	*WalkSpeedEnum*	0:1	Walk speed to use. See allowed values.

	MaximumNumberOf-Changes	*xsd:integer*	0:1	Maximum number of changes allowed.
«enum»	***AccessibilityFiltering***	*Accessibility-FilteringEnum*	0:1	Whether filtering is strict - must meet all criteria or permissive - if no accessible journeys found, others will be shown. See below.
	MinimumLeg-Duration	*xsd:duration*	0:1	Use walk speed for journeys longer than this.

AccessibilityFiltering – Allowed values

The following table shows the allowed values for ***AccessibilityFiltering***. (*AccessibilityFilteringEnumeration).*

Table 373 – *AccessibilityFiltering* – Allowed values

Value	Description
strict	Filtering is strict. Only journeys that satisfy criteria will be included.
permissive	Filtering is permissive. If journeys that satisfy criteria cannot be found, others will be included.
decorative	All journeys will be included, along with accessibility attributes, whether they are accessible or not.

C.1.2.3.5.5 FareProductFilter – Model Element

The FARE PRODUCTs to which to restrict results of a query.

Table 374 – *FareProductFilter* – Element

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***FareProductRef***	*FareProductRef+*	0:1	FARE PRODUCT held by user, for example rail card.
«FK»	***TypeOfFare-ProductRef***	*TypeOfFareProductRef*	0:1	Type of FARE PRODUCT held by user.
	StartDate	*xsd:date*	0:1	Start date of the product.
	ExpiryDate	*xsd:date*	0:1	Expiry date of the product.
«FK»	***TypeOfSales-ContractRef***	*TypeOfSalesContractRef*	0:1	Type of SALES CONTRACT held by CUSTOMER.

C.1.2.3.5.6 TripPlanRequestPolicy – Model Element

Optimisation criteria to be used to when computing and decorating trip plans.

Table 375 – *TripPlanRequestPolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
«enum»	***Optimisation-Method***	*Optimisation-MethodEnum*	1:1	How to optimize the query. See allowed values below.
	UseRealTime	*xsd:boolean*	0:1	Whether plan should use real-time data if available and if relevant.
	IncludeRoute-Projection	*xsd:boolean*	0:1	Whether results should include a plot of the trip.
	Include-Intermediate-Stops	*xsd:boolean*	0:1	Whether results should include information about intermediate CALLs that the journey makes between the trip start and end stop points.
	IncludeJourney-OriginDestination	*xsd:boolean*	0:1	Whether results should include information about the origin and destination of each VEHICLE JOURNEY, if they are before or beyond the start and end points of the trip.
	IncludeFares	*xsd:boolean*	0:1	Whether results should include information about fares.
	Maximum-NumberOf-Journeys	*xsd:integer*	0:1	Maximum number of journeys to include in results.

OptimisationMethod – Allowed values

The following table shows the allowed values for ***OptimisationMethod***. (*OptimisationMethodEnumeration).*

Table 376 – *OptimisationMethod* – Allowed values

Value	Description
fastest	Optimize trip to fastest trip.
leastChanges	Optimize trip for least changes.
cheapest	Optimize trip to cheapest trip.
mostAccessible	Optimize trip to best satisfy the accessibility criteria.

CyclingPreferences – Allowed values

The following table shows the allowed values for ***CyclingPreferences***. (*CyclingPreferencesEnumeration).*

Table 377 – *CyclingPreferences* – Allowed values

Value	Description
noCycling	No Cycling options.
cycleOnly	Only Cycle routes.

leaveCycleAtStation	Can cycle to a station (See access option for distance allowed).
cycleOnVehicle	Require journeys that can take cycle on journey.
foldingCycleOnVehicle	Assume passenger has cycle either end.
cycleHireLegs	Will consider cycle hire of available.

C.1.2.3.6 Stop Departure Request

A STOP DEPARTURE REQUEST is a PASSENGER REQUEST about departures from a given SCHEDULED STOP POINT.

C.1.2.3.7 Stop Departure Request – Conceptual MODEL

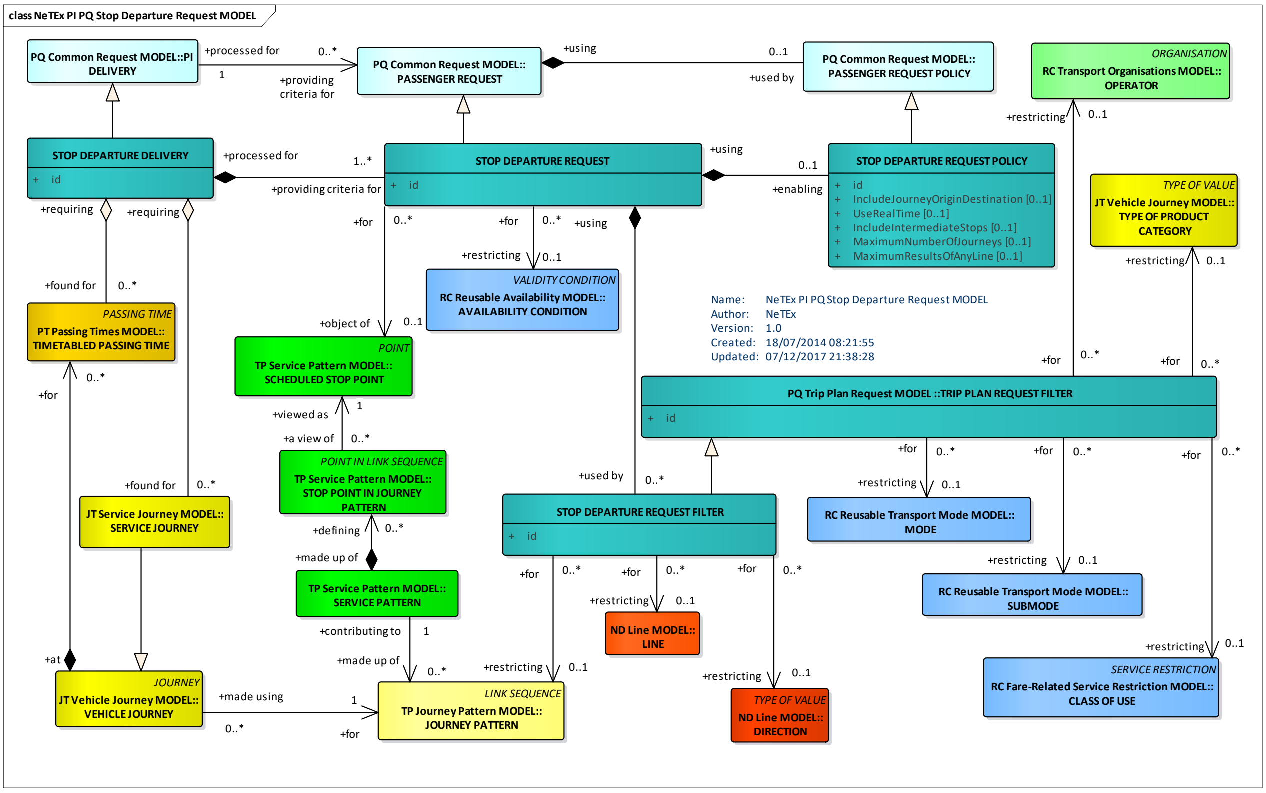


Figure 409 — Stop Departure Request– Conceptual Model

C.1.2.3.8 Stop Departure Request – Physical model – Overview

The following figure introduces the physical model for a STOP DEPARTURE REQUEST. A Stop Departure REQUEST returns a list of CALLs that satisfy the given search criteria.

Each STOP DEPARTURE DELIVERY (corresponding to an http request or remote procedure call) may contain one or more STOP DEPARTURE REQUESTs.

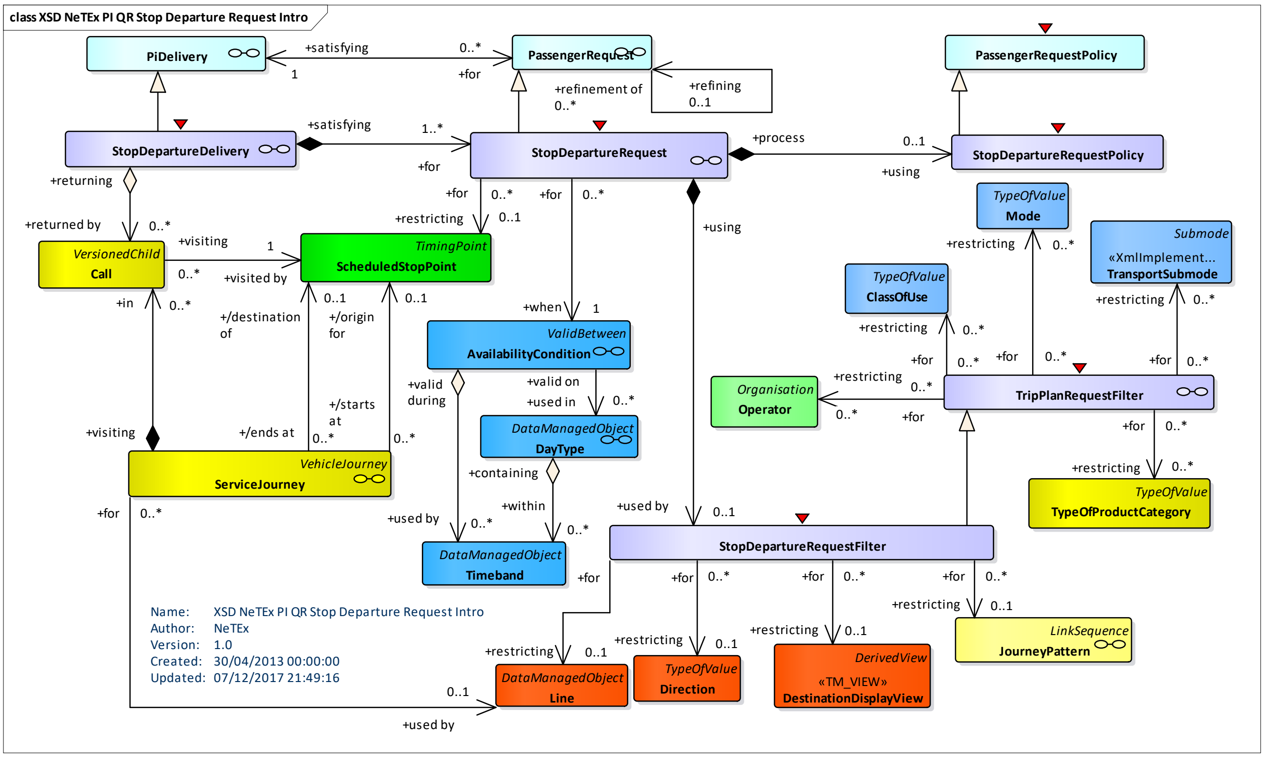


Figure 410 — Stop Departure Request– Physical Model: Overview

C.1.2.3.9 Stop Departure Request – Physical model – Details

The following figure shows the physical model for STOP DEPARTURE REQUEST.

The fundamental criteria (SCHEDULED STOP POINT) must be supplied. Other additional criteria can also be specified.

- STOP DEPARTURE REQUEST FILTER can be used to limit the journeys to be included by MODE. LINE, DIRECTION etc.
- STOP DEPARTURE POLICY species parameters controlling the way the departures are computed.

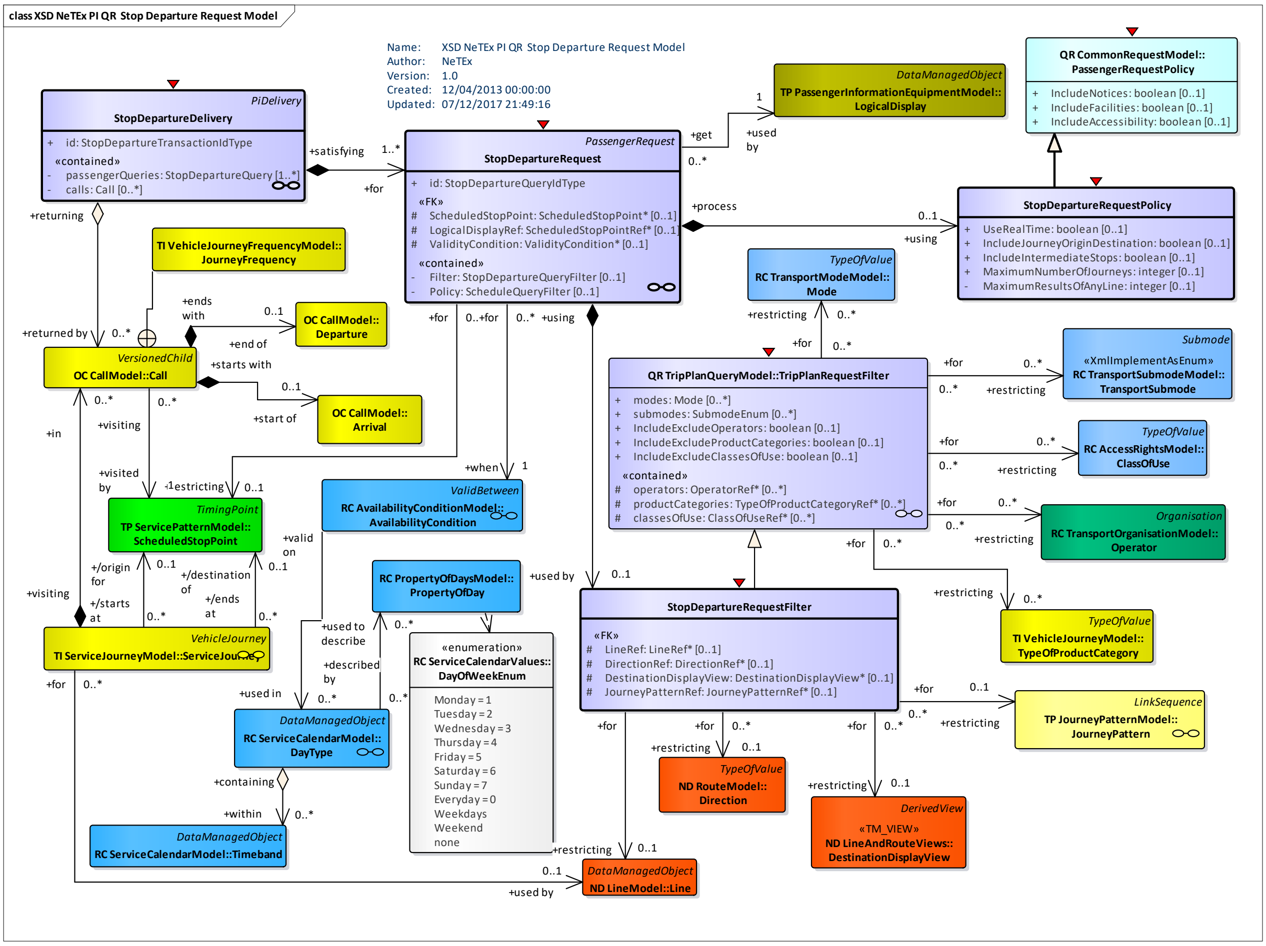


Figure 411 — Stop Departure Request – Physical Model: Details

C.1.2.3.10 Stop Departure Request – Attributes and XSD

C.1.2.3.10.1 StopDepartureDelivery – Model Element

A specialization of PI DELIVERY to make one or more STOP DEPARTURE QUERies.

Table 378 – *StopDepartureDelivery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PiDelivery*	::>	STOP DEPARTURE DELIVERY inherits from PI DELIVERY.
	id	*StopDeparture-DeliveryIdType*	1:1	Identifier of a STOP DEPARTURE DELIVERY.
«cntd»	***passenger-Queries***	*StopDepartureRequest*	1:*	STOP DEPARTURE Passenger QUERies.
«cntd»	***calls***	*Call*	0:*	CALLs returned in response to STOP DEPARTURE query.

C.1.2.3.10.2 StopDepartureRequest – Model Element

A PASSENGER REQUEST about departures at a stop or LOGICAL DISPLAY at that stop.

Table 379 – *StopDepartureRequest* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerRequest*	::>	STOP DEPARTURE REQUEST inherits from PASSENGER REQUEST.
	id	*StopDepartureRequest-IdType*	1:1	Identifier of a STOP DEPARTURE REQUEST.
«FK»	***Scheduled-StopPointRef***	*ScheduledStopPointRef*	0:1	Reference to SCHEDULED STOP POINT for which departures are to be shown.
«FK»	***LogicalDisplay-Ref***	*LogicalDisplayRef*	0:1	Reference to LOGICAL DISPLAY for which departure data is to be shown.
«FK»	***Validity-ConditionRef***	*ValidityConditionRef*	0:1	VALIDITY CONDITION for showing data.
«cntd»	***Filter***	*StopDeparture-RequestFilter*	0:1	Additional filter parameters. See below.
«cntd»	***Policy***	*StopDeparture-RequestFilter*	0:1	Additional policy parameters. See STOP DEPARTURE REQUEST POLICY below.

C.1.2.3.10.3 StopDepartureRequestFilter – Model Element

Additional filter criteria for including VEHICLE JOURNEYs in a STOP DEPARTURE REQUEST.

Table 380 – *StopDepartureRequest*Filter – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TripPlanRequestFilter*	::>	STOP DEPARTURE REQUEST FILTER inherits from TRIP PLAN REQUEST FILTER.
«FK»	***LineRef***	*LineRef*	0:1	Reference to a LINE by which to filter results.
«FK»	***DirectionRef***	*DirectionRef*	0:1	Reference to a DIRECTION by which to filter results.
«FK»	***Destination-DisplayRef***	*DestinationDisplayRef*	0:1	Reference to a DESTINATION DISPLAY by which to filter results.

«FK»	***JourneyPattern-Ref***	*JourneyPatternRef+*	0:1	Reference to a JOURNEY PATTERN by which to filter results.

C.1.2.3.10.4 StopDepartureRequestPolicy – Model Element

Criteria for computing results of a STOP DEPARTURE REQUEST.

Table 381 – *StopDepartureRequest*Policy – Element

Classifi-cation	Name	Type	Cardin ality	Description
	UseRealTime	*xsd:boolean*	1:1	Whether plan should use real-time data if available and if relevant.
	IncludeJourney-OriginDestination	*xsd:boolean*	0:1	Whether results should include information about origin and destination of vehicle journey if before or beyond trip start and end points.
	IncludeIntermediate-Stops	*xsd:boolean*	0:1	Whether results should include information about intermediate CALLs that the journey makes between the trip start and end.
	UseRealTime	*xsd:boolean*	0:1	Whether query should use real-time data if available and if relevant.
	MaximumNumberOf-Journeys	*xsd:integer*	0:1	Maximum number of journeys to include in results.
	MaxmimumResults-OfAnyLine	*xsd:integer*	0:1	Maximum number of CALLs for any line to be returned,

C.1.2.4 Schedule Request

A SCHEDULE REQUEST is a PASSENGER REQUEST about public timetables.

C.1.2.4.1 Schedule Request – Conceptual MODEL

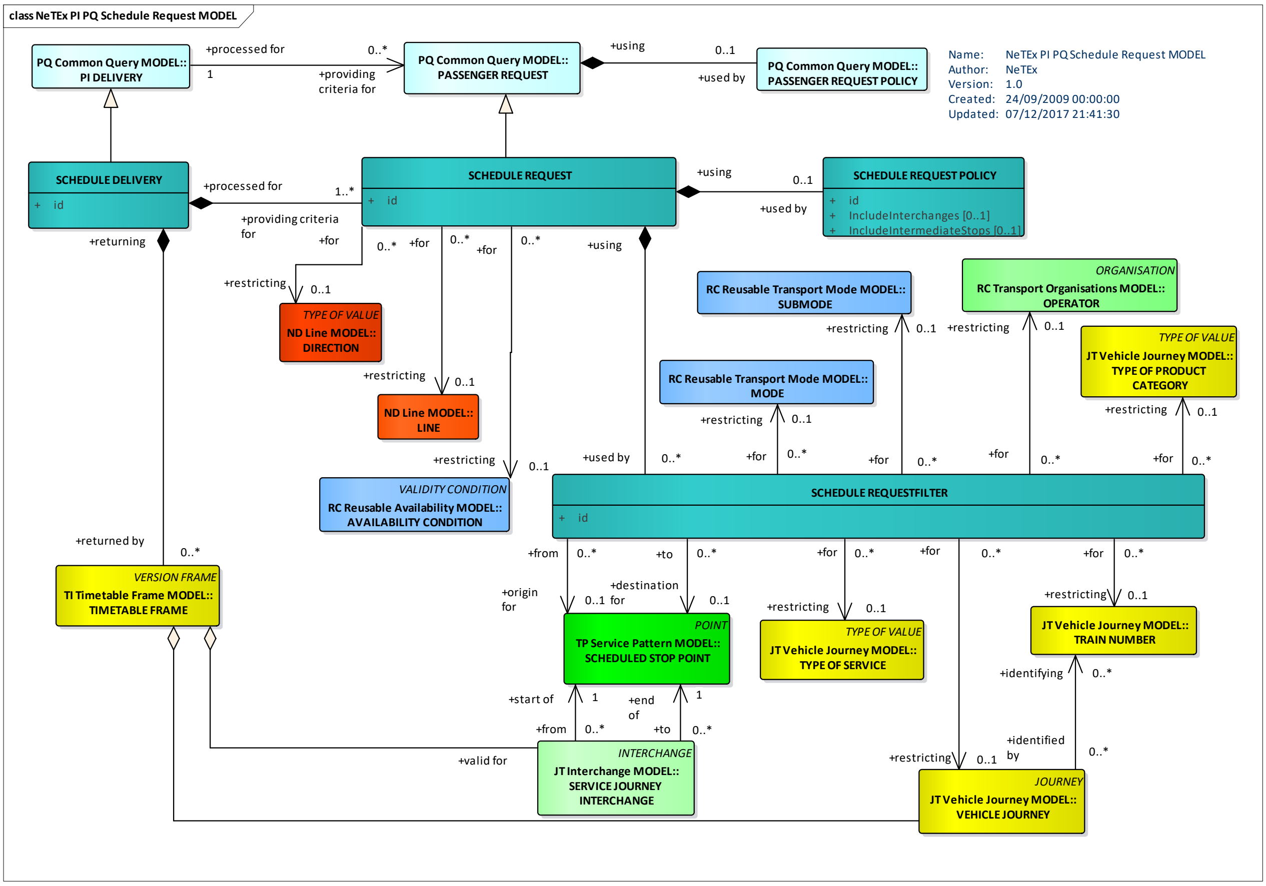


Figure 412 — Schedule Request– Conceptual Model

C.1.2.4.2 Schedule Request – Physical model – Overview

The following figure introduces the physical model for a SCHEDULE REQUEST. A Schedule REQUEST returns a list of TIMETABLEs that satisfy the given search criteria.

Each SCHEDULE DELIVERY (corresponding to an http request or remote procedure call) may contain one or more SCHEDULE REQUESTs.

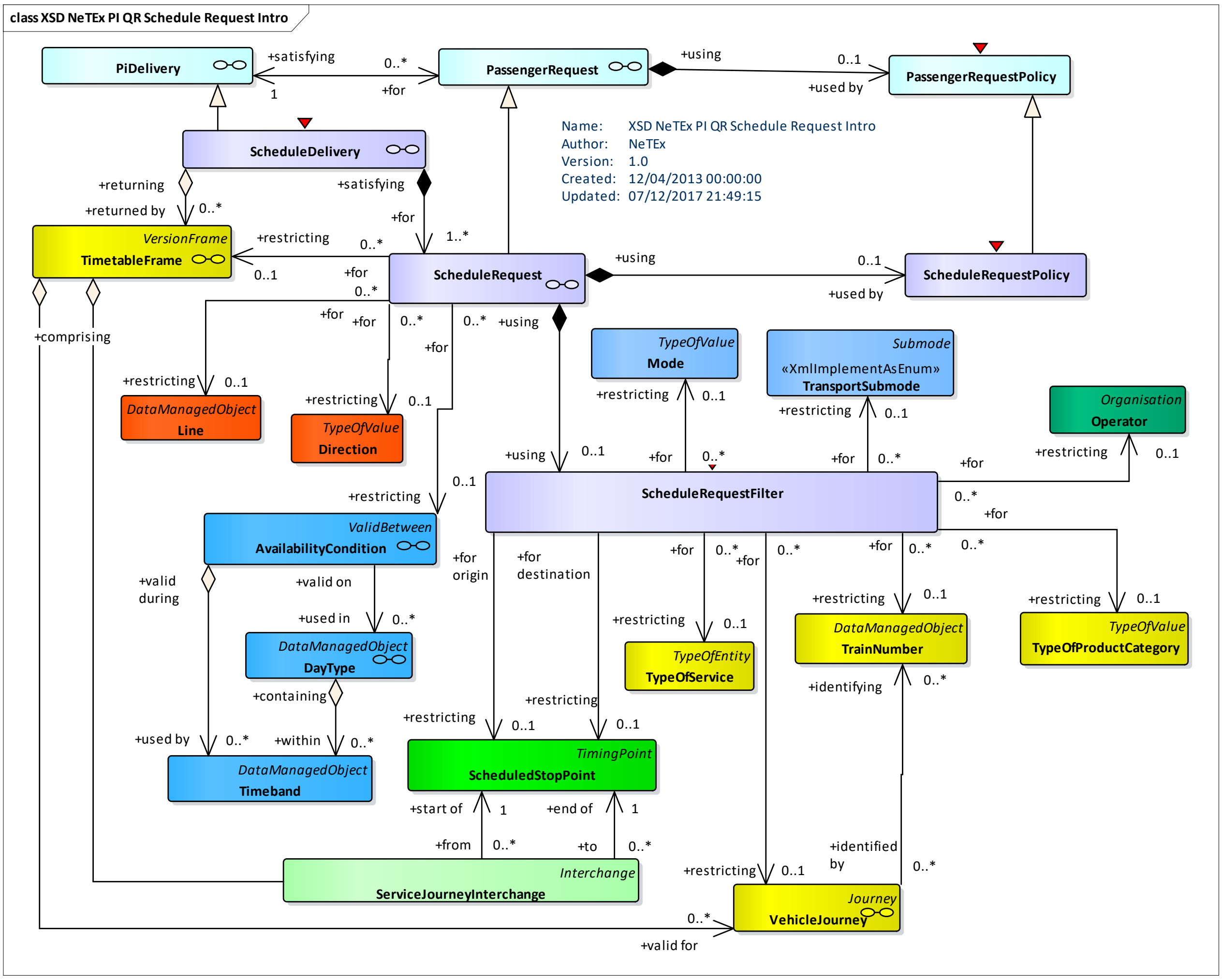


Figure 413 — Schedule Request– Physical Model: Overview

C.1.2.4.3 Schedule Request – Physical model – Details

The following figure shows the physical model for SCHEDULE REQUEST.

Eater a specific timetable may be specified, or parameters to identify one.

- SCHEDULE REQUEST FILTER can be used to limit the journeys to be included by MODE. LINE, DIRECTION etc. and to a particular start and end point.

- SCHEDULE REQUEST POLICY can be used to direct which information is to be included in the results.

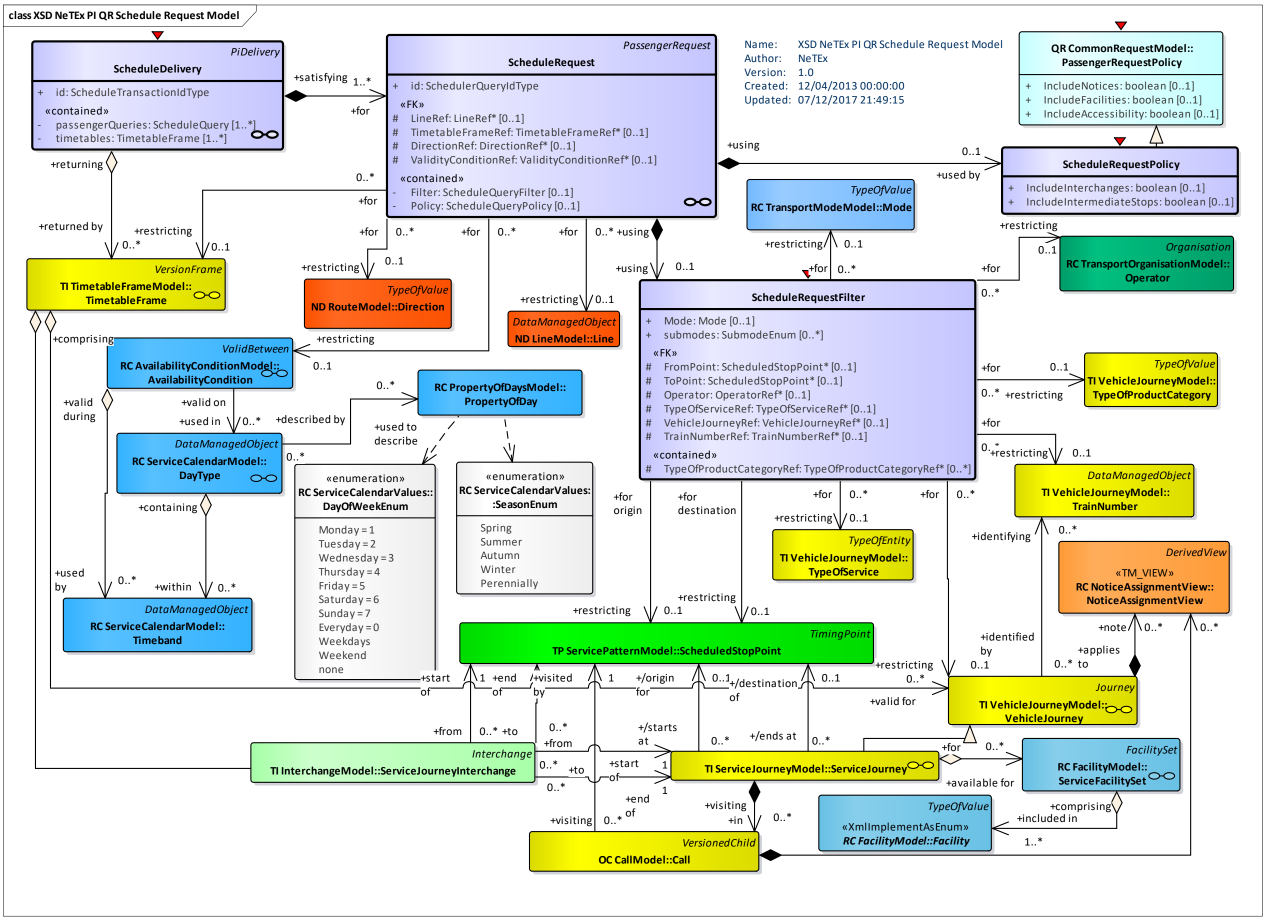


Figure 414 — Schedule Request – Physical Model: Details

C.1.2.4.4 Schedule Request – Attributes and XSD

C.1.2.4.4.1 ScheduleDelivery – Model Element

A specialization of PI DELIVERY to make one or more SCHEDULE QUERies.

Table 382 – *ScheduleDelivery* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PiDelivery*	::>	SCHEDULE DELIVERY inherits from PI DELIVERY.
	id	*ScheduleDeliveryIdType*	1:1	Identifier of a SCHEDULE DELIVERY.
«cntd»	***passenger-Queries***	*ScheduleRequest*	0:*	SCHEDULE QUERies for the Delivery.
«cntd»	***timetables***	*TimetableFrame*	0:*	TIMETABLE FRAMEs with SERVICE JOURNEYS returned in response to query.

C.1.2.4.4.2 ScheduleRequest – Model Element

A PASSENGER REQUEST about public transport timetables.

Table 383 – *ScheduleRequest* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerRequest*	::>	SCHEDULE REQUEST inherits from PASSENGER REQUEST.
	id	*SchedulerRequestIdType*	1:1	Identifier of a SCHEDULE REQUEST.
«FK»	***LineRef***	*LineRef*	0:1	Reference to a LINE for which SCHEDULE is to be retrieved.
«FK»	***Timetable-FrameRef***	*TimetableFrameRef*	0:1	TIMETABLE for which schedule is to be returned for specified timetable.
«FK»	***DirectionRef***	*DirectionRef*	0:1	Reference to a DIRECTION for which schedule is to be fetched.
«FK»	***Validity-ConditionRef***	*ValidityConditionRef*	0:1	VALIDITY CONDITION to which ACCESS RIGHT PARAMETER is assigned.
«cntd»	***Filter***	*ScheduleRequestFilter*	0:1	Additional filter parameters. See below.
«cntd»	Policy	*ScheduleRequestPolicy*	0:1	Request Policy parameters. See below.

C.1.2.4.4.3 ScheduleRequestFilter – Model Element

Additional filter criteria for SCHEDULE REQUEST, for example start and end stop. Only timetables that satisfy criteria will be returned.

Table 384 – *ScheduleRequestFilter* – Element

Classifi-cation	Name	Type	Cardinality	Description
	Mode	*Mode*	0:1	Mode of transport for which SCHEDULE is to be produced.
«FK»	***FromPoint***	*ScheduledStopPoint*	0:1	SCHEDULED STOP POINT at which desired TRIP starts.
«FK»	***ToPoin***t	*ScheduledStopPoint*	0:1	SCHEDULED STOP POINT at which desired TRIP ends.
«FK»	***Operator***	*OperatorRef*	0:1	OPERATOR for which to filter results.

«cntd»	***TypeOfProduct-CategoryRef***	*TypeOfProduct-CategoryRef*	0:*	Type of PRODUCT CATEGORY on which to filter results.
«FK»	***TypeOf-ServiceRef***	*TypeOfServiceRef*	0:1	TYPE OF SERVICE of JOURNEY on which to filter results.
«FK»	***VehicleJourney-Ref***	*VehicleJourneyRef*	0:1	Reference to a VEHICLE JOURNEY on which to filter results.
«FK»	***TrainNumberRef***	*TrainNumberRef*	0:1	TRAIN NUMBER on which to filter results.

C.1.2.4.4.4 ScheduleRequestPolicy – Model Element

Criteria for computing results of a SCHEDULE REQUEST.

Table 385 – *ScheduleRequestPolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*Passenger-RequestPolicy*	::>	SCHEDULE REQUEST POLICY inherits from PASSENGER REQUEST POLICY.
	IncludeInterchanges	*xsd:boolean*	0:1	Whether results should include information about SERVICE JOURNEY INTERCHANGEs for the SERVICE JOURNEYs in the timetable.
	IncludeIntermediate Stops	*xsd:boolean*	0:1	Whether results should include information about intermediate CALLs that the journey makes between the trip start and end.

C.1.2.5 Fare Request

C.1.2.5.1 Common Fare Request Model

Fare queries fetch the available fares for a trip. There are two types of fare query

- TRIP FARE REQUEST: finds the FARE PRODUCTs suitable for a single trip.
- REPEATED FARE REQUEST: finds the FARE PRODUCTs suitable for regularly repeated journeys such as season passes.

C.1.2.5.1.1 Common Fare Request – Conceptual MODEL

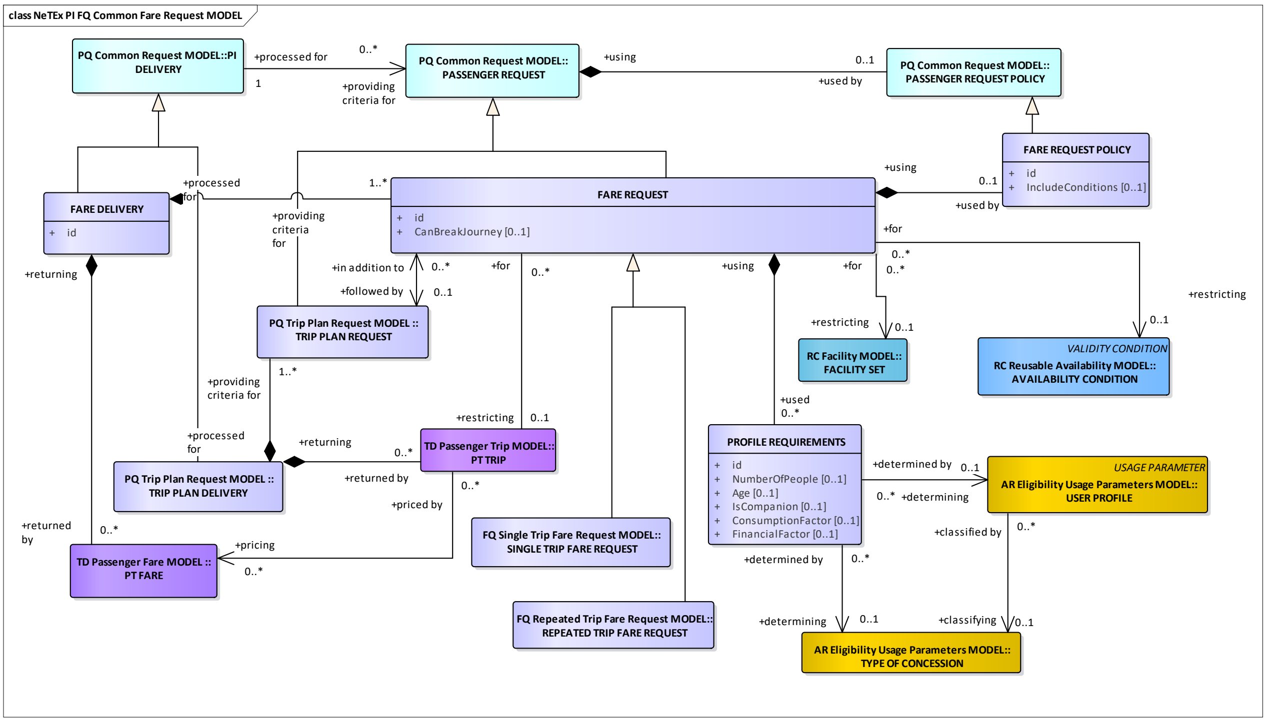


Figure 415 — Common Fare Request– Conceptual Model

C.1.2.5.1.2 Common Fare Request – Physical model – Overview

The following figure introduces the physical model for common fare query elements that are used by both types of query.

Each COMMON FARE DELIVERY (corresponding to an http request or remote procedure call) may contain one or more COMMON FARE REQUESTs.

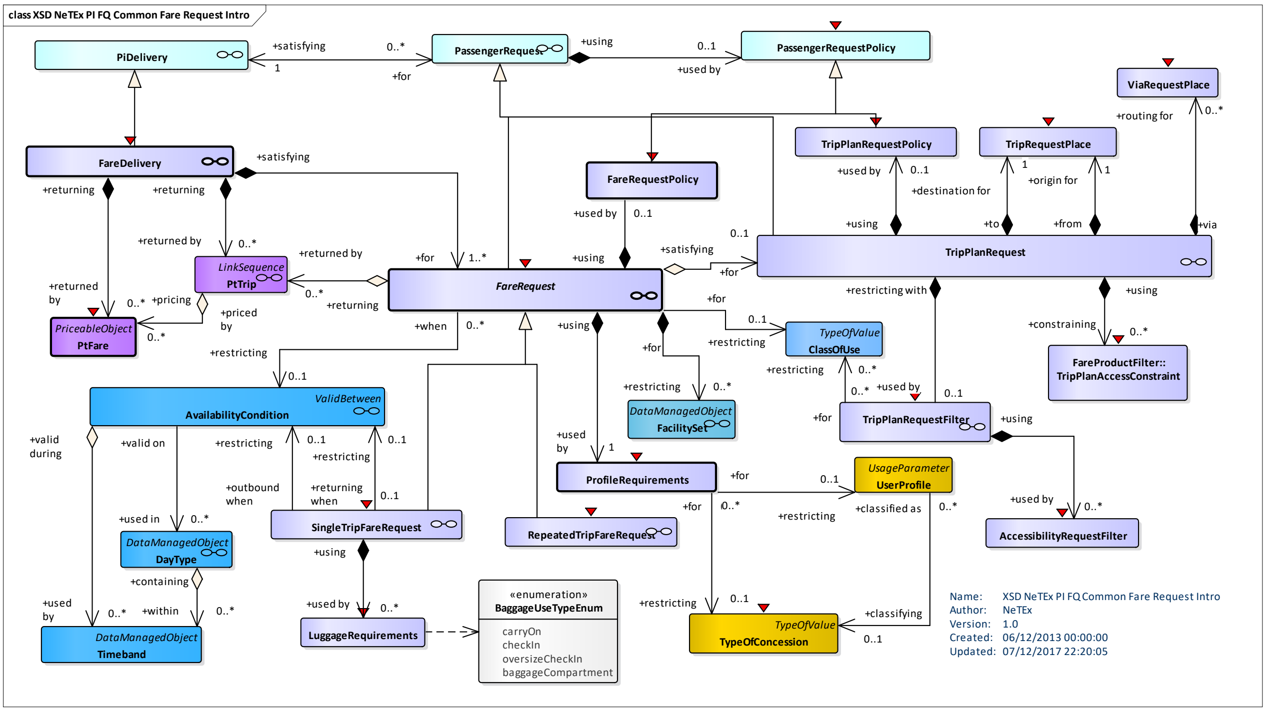


Figure 416 — Common Fare Request– Physical Model: Overview

C.1.2.5.1.3 Common Fare Request – Physical model – Details

The following figure shows the physical model for COMMON FARE REQUEST.

The fundamental criteria (SCHEDULED STOP POINT) must be supplied. Other additional criteria can also be specified.

- COMMON FARE REQUEST FILTER can be used to limit the journeys to be included. It can use a TRIP PLAN REQUEST to specify the journey.
- PROFILE REQUIREMENTs can be used to limit the fares to those that match the user's eligibility.
- COMMON FARE POLICY species parameters controlling the way the departures are computed.

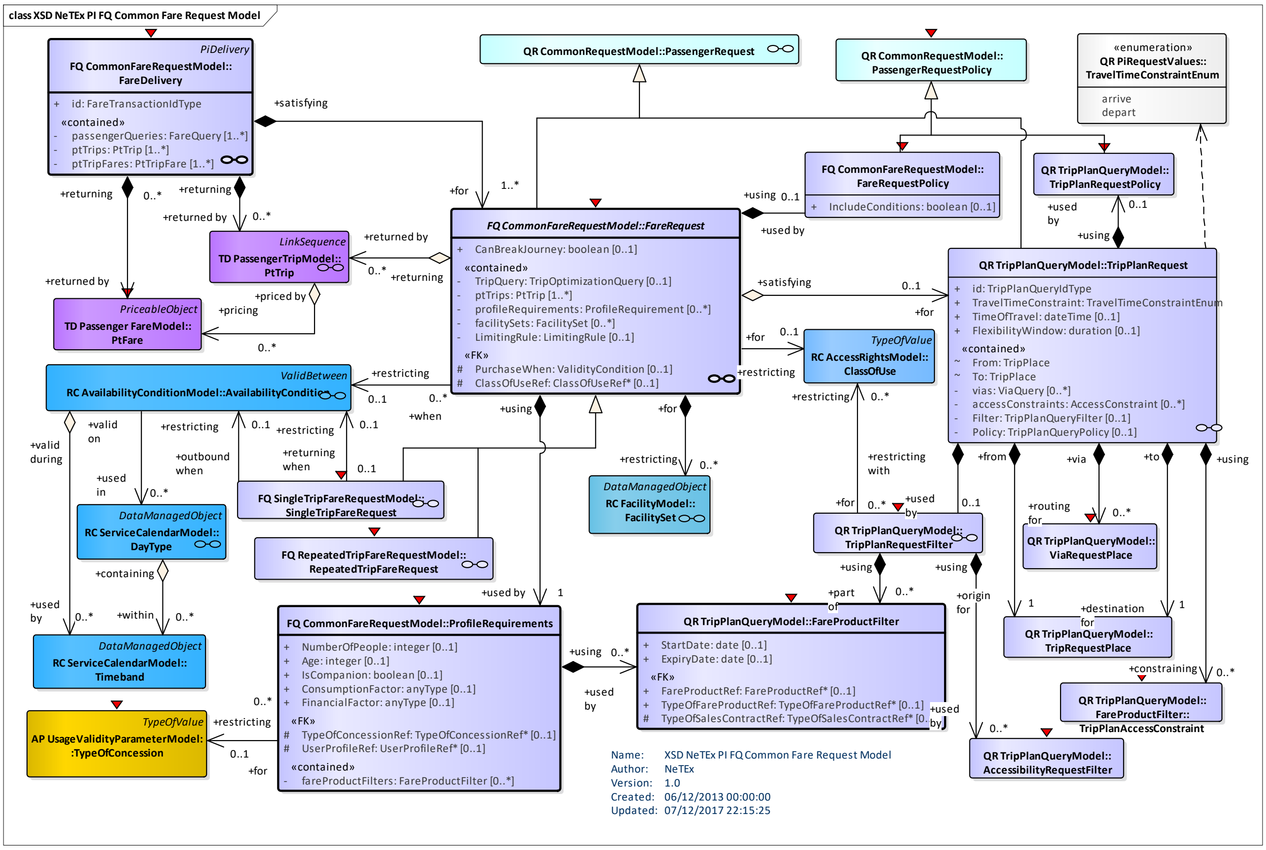


Figure 417 — Common Fare Request – Physical Model: Details

C.1.2.5.1.4 Common Fare Request – Attributes and XSD

FareDelivery – Model Element

A specialization of PI DELIVERY to make one or more FARE QUERies.

Table 386 – *FareDelivery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	PiDelivery	::>	FARE DELIVERY inherits from PI DELIVERY
	id	*FareDeliveryIdType*	1:1	Identifier of a FARE DELIVERY.
«cntd»	***passenger-Queries***	FareRequest	0:*	FARE QUERies for PI DELIVERY.
«cntd»	***ptTrips***	PtTrip	0:*	Passenger TRIPs in response to query.
«cntd»	***ptFares***	PtTripFare	0:*	Passenger FAREs in response to query

FareRequest – Model Element

A PASSENGER REQUEST about fares.

Table 387 – *FareRequest* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerRequest*	::>	FARE REQUEST inherits from PASSENGER REQUEST.
«cntd»	***TripRequest***	*TripPlanRequest*	0:1	Parameters describing a passenger trip for which fares are to be found. See TRIP PLAN REQUEST.
«FK»	***PurchaseWhen***	*ValidityCondition*	0:1	VALIDITY CONDITION specifying when fare is to be purchased. See NeTEx Part1.
	CanBreak-Journey	*xsd:boolean*	0:1	Whether user would like to break journey en route.
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	CLASS OF USE on which to filter results.
«cntd»	***profile-Requirements***	*ProfileRequirement*	0:*	PROFILE REQUIREMENTs for query. See below.
«cntd»	***facilitySets***	*(FacilitySet))*	0:*	Facilities desired on SERVICE JOURNEY.

ProfileRequirement – Model Element

The number and characteristics of a person wishing to travel.

Table 388 – *ProfileRequriement* – Element

Classifi-cation	Name	Type	Cardinality	Description
	NumberOfPeople	*xsd:integer*	0:1	Number of people of this type wanting to travel.
	Age	*xsd:integer*	0:1	Age of traveller.
	IsCompanion	*xsd:boolean*	0:1	Whether user is a companion or carer of someone else in the group.
	Consumption-Factor	*xsd:anyType*	0:1	Consumption factor associated with query, e.g. number of air miles.
	FinancialFactor	*xsd:anyType*	0:1	Financial factor associated with COMMERCIAL PROFILE.
«FK»	***TypeOf-ConcessionRef***	*TypeOfConcessionRef*	0:1	Type of concession user has.

«FK»	***UserProfileRef***	*UserProfileRef*	0:1	Reference to a USER PROFILE
«cntd»	***fareProduct-Filters***	*FareProductFilter*	0:*	Products held by user that can be taken into account in query. See TRIP PLAN REQUEST.

C.1.2.5.2 Single Trip Fare Request Model

A SINGLE TRIP FARE REQUEST is a PASSENGER REQUEST about fare for a single trip or return trip.

C.1.2.5.2.1 Single Trip Fare Request – Conceptual MODEL

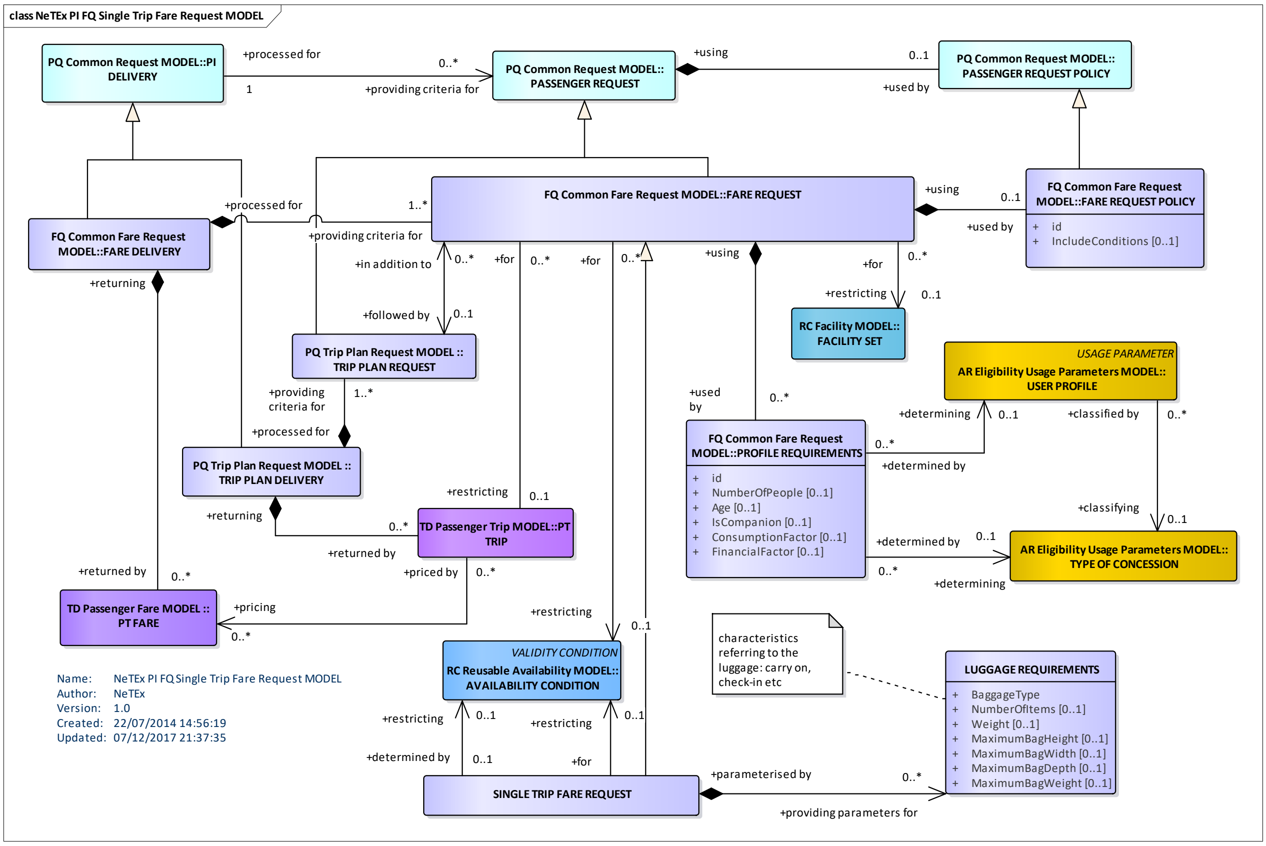


Figure 418 — Single Trip Fare Request– Conceptual Model

C.1.2.5.2.2 Single Trip Fare Request – Physical model – Overview

The following figure introduces the physical model for a SINGLE TRIP FARE REQUEST. A SINGLE TRIP FARE REQUEST returns a list of PT FAREs for a PT TRIP that satisfy the given search criteria.

Each FARE DELIVERY may contain one or more SINGLE TRIP FARE QUERies.

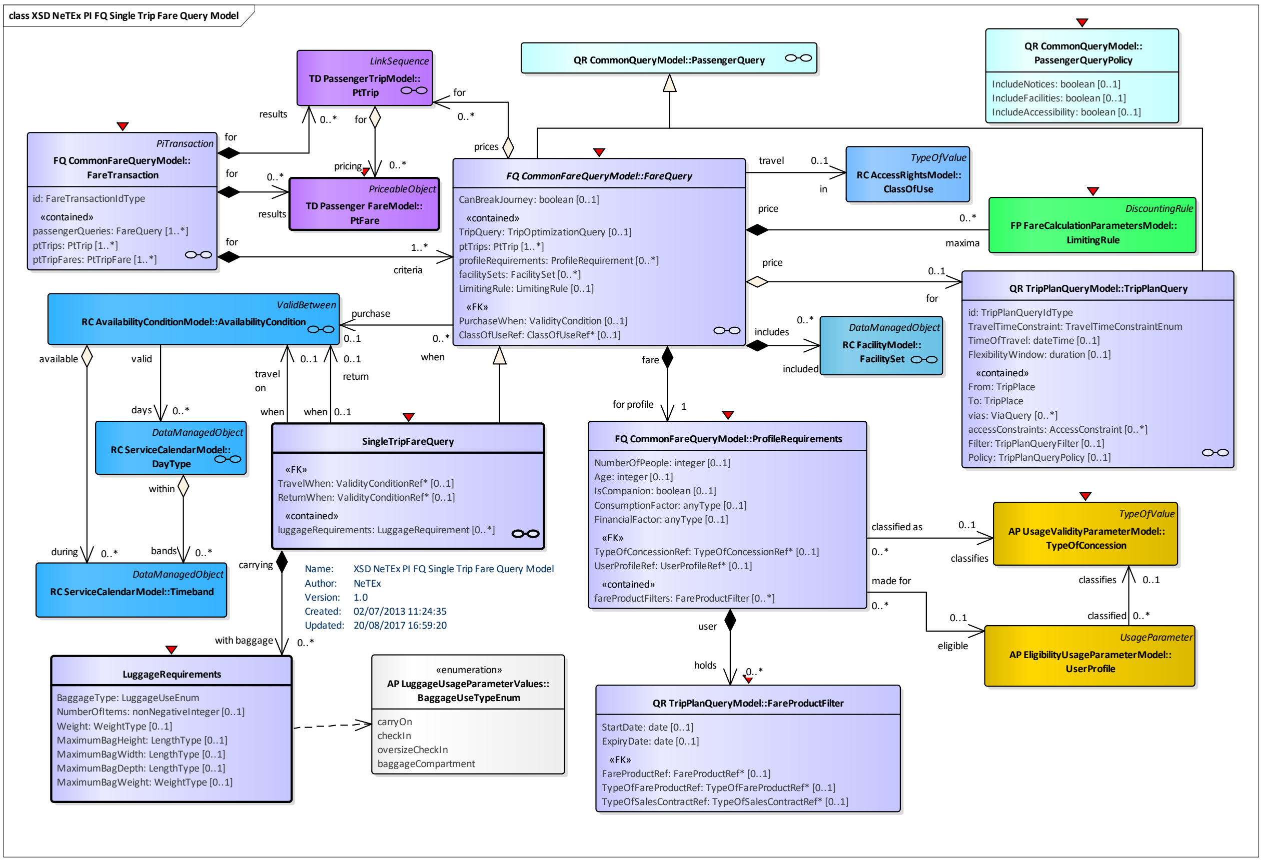


Figure 419 — Single Trip Fare Request– Physical Model: Overview

C.1.2.5.2.3 Single Trip Fare Request – Physical model – Details

The following figure shows the physical model for SINGLE TRIP FARE REQUEST. It returns the PT FAREs for specified trips.

To specify the journey for which fares are required the use may either include a new SINGLE TRIP REQUEST or the PT TRIPs resulting from a previously executed SINGLE TRIP REQUEST.

- LUGGAGE REQUIREMENTs species details of luggage carrying requirements, e.g. a bicycle.

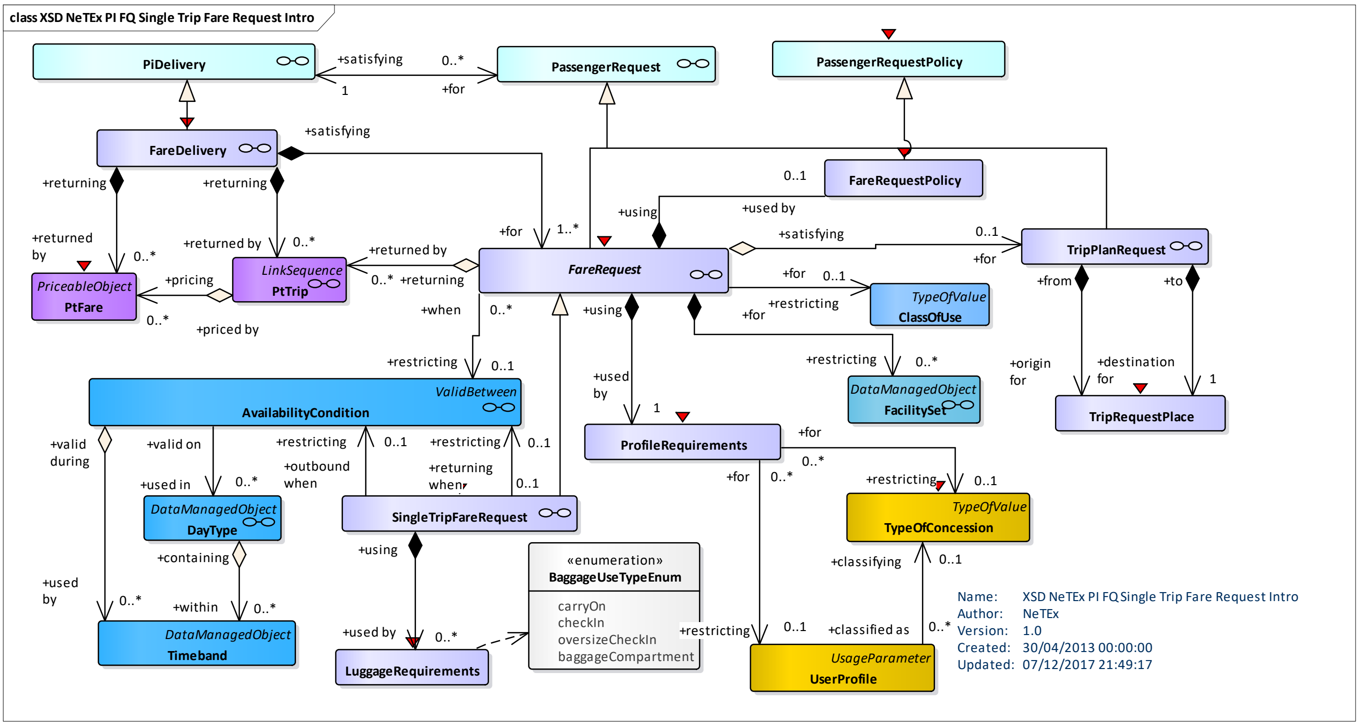


Figure 420 — Single Trip Fare Request – Physical Model: Details

C.1.2.5.2.4 Single Trip Fare Request – Attributes and XSD

SingleTripFareRequest – Model Element

A PASSENGER REQUEST about the fares for a single trip or return trip.

Table 389 – *SingleTripFareRequest* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	FareRequest	::>	SINGLE TRIP FARE REQUEST inherits from FARE REQUEST
«FK»	***TravelWhen***	ValidityCondition	0:1	VALIDITY CONDITION specifying when outbound travel is to take place.
«FK»	***ReturnWhen***	ValidityCondition	0:1	VALIDITY CONDITION specifying when return travel is to take place.
«cntd»	***luggage-Requirements***	LuggageRequirement	0:*	Luggage requirements for different types of bag.

LuggageRequirements – Model Element

The number and characteristics (weight, volume) of luggage that a passenger wishes to carry.

Table 390 – *LuggageRequirements* – Model Element

Classifi-cation	Name	Type	Cardin-ality	Description
«enum»	***BaggageType***	*LuggageUseEnum*	1:1	Type of bag described by this element.
	NumberOfItems	*xsd:nonNegativeInteger*	0:1	Number of bags allowed.
	Weight	*WeightType*	0:1	Total Weight limit of LUGGAGE ALLOWANCE.
	MaximumBagHeight	*LengthType*	0:1	Maximum bag height.
	MaximumBagWidth	*LengthType*	0:1	Maximum bag Width.
	MaximumBagDepth	*LengthType*	0:1	Maximum bag Depth.
	MaximumBagWeight	*WeightType*	0:1	Maximum bag weight.

C.1.2.5.3 Repeated Trip Fare Request Model

A REPEATED TRIP FARE REQUEST is A PASSENGER REQUEST about the best fare products to use for repeated similar trips.

C.1.2.5.3.1 Repeated Trip Fare Request – Conceptual MODEL

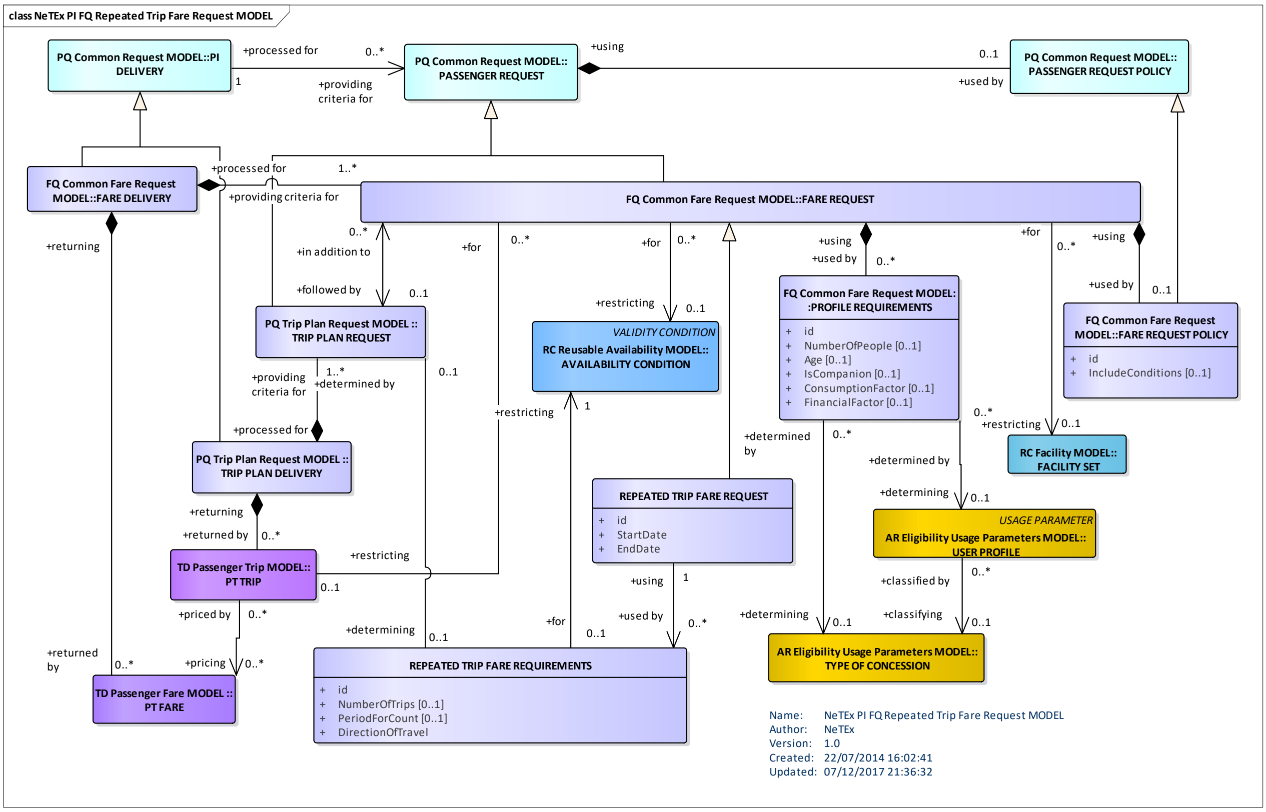


Figure 421 — Repeated Trip Fare Request– Conceptual Model

C.1.2.5.3.2 Repeated Trip Fare Request – Physical model – Overview

The following figure introduces the physical model for a REPEATED TRIP FARE REQUEST. A REPEATED TRIP FARE REQUEST returns a list of PT FAREs for trips that satisfy the given search criteria.

Each FARE DELIVERY may contain one or more REPEATED TRIP FARE REQUESTs.

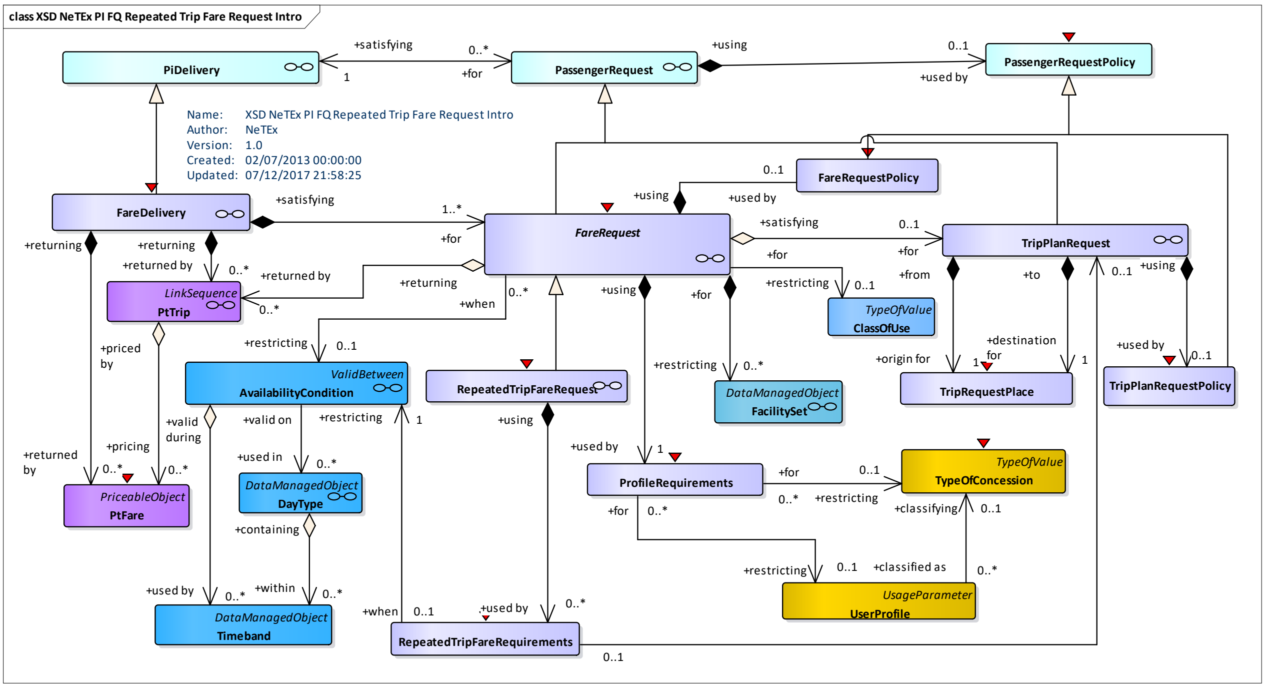


Figure 422 — Repeated Trip Fare Request– Physical Model: Overview

C.1.2.5.3.3 Repeated Trip Fare Request – Physical model – Details

The following figure shows the physical model for TRIP FARE REQUEST. It returns the PT FAREs for repeated trips to be u undertaken within a given period. To specify the journey for which fares are required the use may either include a new TRIP REQUEST or the PT TRIPs resulting from a previously executed TRIP REQUEST.

- REPEATED TRIP FARE REQUEST FILTER can be used to limit the period of travel to be considered.
- REPEATED TRIP REQUIREMENTs species the number of journeys to be made in the period.

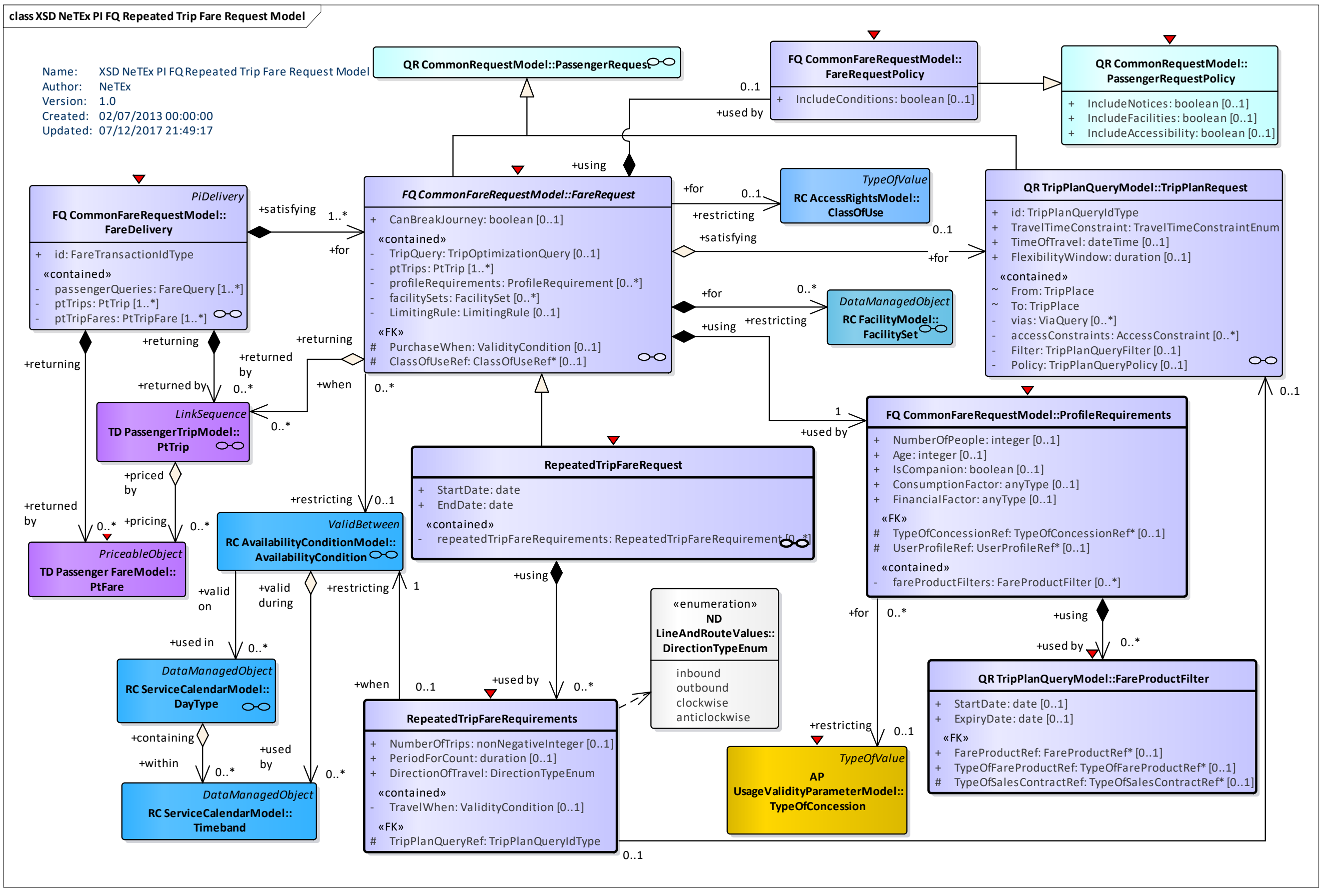


Figure 423 — Repeated Trip Fare Request – Physical Model: Details

C.1.2.5.3.4 Repeated Trip Fare Request – Attributes and XSD

RepeatedTripFareRequest – Model Element

A PASSENGER REQUEST to find the best fares to use for repeated trips.

Table 391 – *RepeatedTripFareRequest* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareRequest*	::>	REPEATED TRIP FARE REQUEST inherits from FARE REQUEST.
	StartDate	*xsd:date*	1:1	Start of period to travel.
	EndDate	*xsd:date*	1:1	End of period to travel.
«cntd»	***repeatedTrip-Fare-Requirements***	*RepeatedTripFare-Requirement*	0:*	Nature of repeated trips required in Period to compute optimum season fare product to use.

RepeatedTripFareRequirements – Model Element

The number and characteristics (weight, volume) of luggage that a passenger wishes to carry.

Table 392 – *RepeatedTripFareRequirements*– Element

Classifi-cation	Name	Type	Cardinality	Description
	NumberOfTrips	*xsd:nonNegativeInteger*	0:1	Number of trips to be made.
	PeriodForCount	*xsd:duration*	0:1	Period within which trip count is made, e.g. per day, if number is trips per day,
«cntd»	***TravelWhen***	*ValidityCondition*	0:1	VALIDITY CONDITION for time of day at which travel is to be made.
«enum»	***DirectionOf-Travel***	*DirectionTypeEnum*	1:1	Direction of travel. See NeTEx Part1.

Annex D
(informative)

How to go from a trip (from NeTEx Part1&2) to a fare ?

Most of NeTEx is concerned with representing and exchanging the journey and fare structures of a public transport system as coherent sets of data. The following PASSENGER TRIP models are useful for relating the NeTEx Elements to individual passenger trips made on the network, as required say by a information query for an individual passenger making a specific trip.

- The PASSENGER TRIP Model describes an individual passenger trip by public transport. It can be used to describe the results of a TRIP REQUEST and also be used as an input to a FARE REQUEST for which the fare is to be returned as

- The PASSENGER FARE Model describes the available fares for specific trips by public transport. It can be used to describe the results of a FARE REQUEST. It can also be used on a SALES DELIVERY may be used to record the inputs used to find the fare

D.1 Passenger Trip

D.1.1 Passenger Trip Model

Computer-aided tools assist potential travellers in preparing their trips, answering PASSENGER QUERies. Such a trip planning function identifies the places origin and destination of an intended trip and proposes one or several trip solutions. The proposition takes into account user's constraints or preferences, such as minimal trip duration, minimal number of interchanges, cheapest fare, etc. This involves an optimisation process using such parameters.

After specification of the trip origin and destination, the systems proposes a TRIP PATTERN composed of access or connection walks and “rides” on PT vehicles. Given a trip pattern proposal, it is possible to compute a precise duration of the trip (taking into account for instance the latest arrival time wished by the traveller) or a mean duration. It is possible as well to calculate the corresponding fare, the suitability for a restricted mobility person, etc.

D.1.1.1 Passenger Trip – Conceptual MODEL

The complete spatial movement of a passenger from a PLACE origin to a PLACE destination is described by a TRIP PATTERN. This entity describes the path between these two places, using public transport vehicles and possibly walking. The TRIP PATTERN may itself be made up of TRIPs, each describing a transition from one successive PLACE to another.

A TRIP PATTERN is made up of one or more LEGs, which is that part of a passenger trip taken on a single MODE. There are different specialisations of LEG for public transport use (PT RIDE LEG), access (ACCESS LEG) and making a connection (PT CONNECTION LEG) A PT RIDE LEG runs from one from one SCHEDULED STOP POINT to another. A PT RIDE LEG is carried out on only one JOURNEY PATTERN.

A PT TRIP is a part of a trip starting from the first boarding of a public transport vehicle to the last alighting from the public transport vehicle. A PT TRIP consists of one or more PT RIDE LEGs and the movements (usually walks) necessary to cover the corresponding CPT ONNECTIONs LEGs.

As long as the TRIP PATTERN is not complemented by a precise timing, any VEHICLE JOURNEY covering this JOURNEY PATTERN may support the proposed RIDE.

A part of a trip corresponding to the movement of a passenger to transfer from one SERVICE JOURNEY to another made at a CONNECTION from one SCHEDULED STOP POINT to another is represented byPT RIDE CONNECTION that may reference NAVIGATION PATHs.

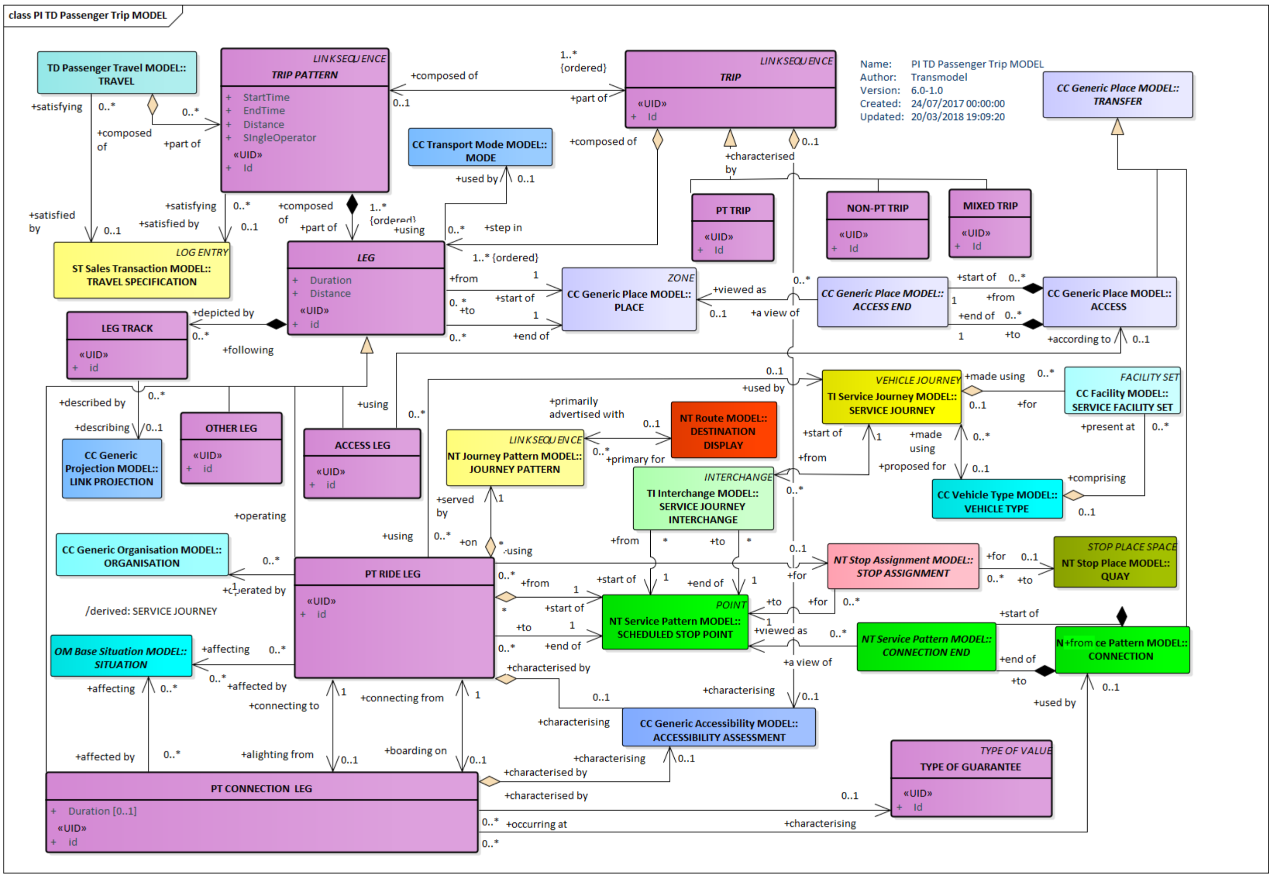


Figure 424 — Passenger Trip – Conceptual Model

D.1.1.2 Passenger Trip – Physical model – Overview

The following figure introduces the physical model for a PASSENGER TRIP. A Passenger Trip REQUEST describes a sequence of one or more RIDEs that a PASSENGER must take to complete a PT TRIP.

A PT TRIP is made up of a TRIP PATTERN and one or more PT RIDEs IN PT TRIP, each of which references a PT RIDE. The PT RIDE may indicate start and end times at a stop by reference to the CALLs of a SERVICE JOURNEY.

PT PT CONNECTION LEG elements may be used to describe details of the interchange, referencing the NAVIGATION PATHs between stops described in NeTExPart1.

ACCESSIBILITY ASSESSMENTs (See NeTEx Part1) can be associated with individual RIDEs and PT CONNECTION LEGs as well as for the overall PT TRIP.

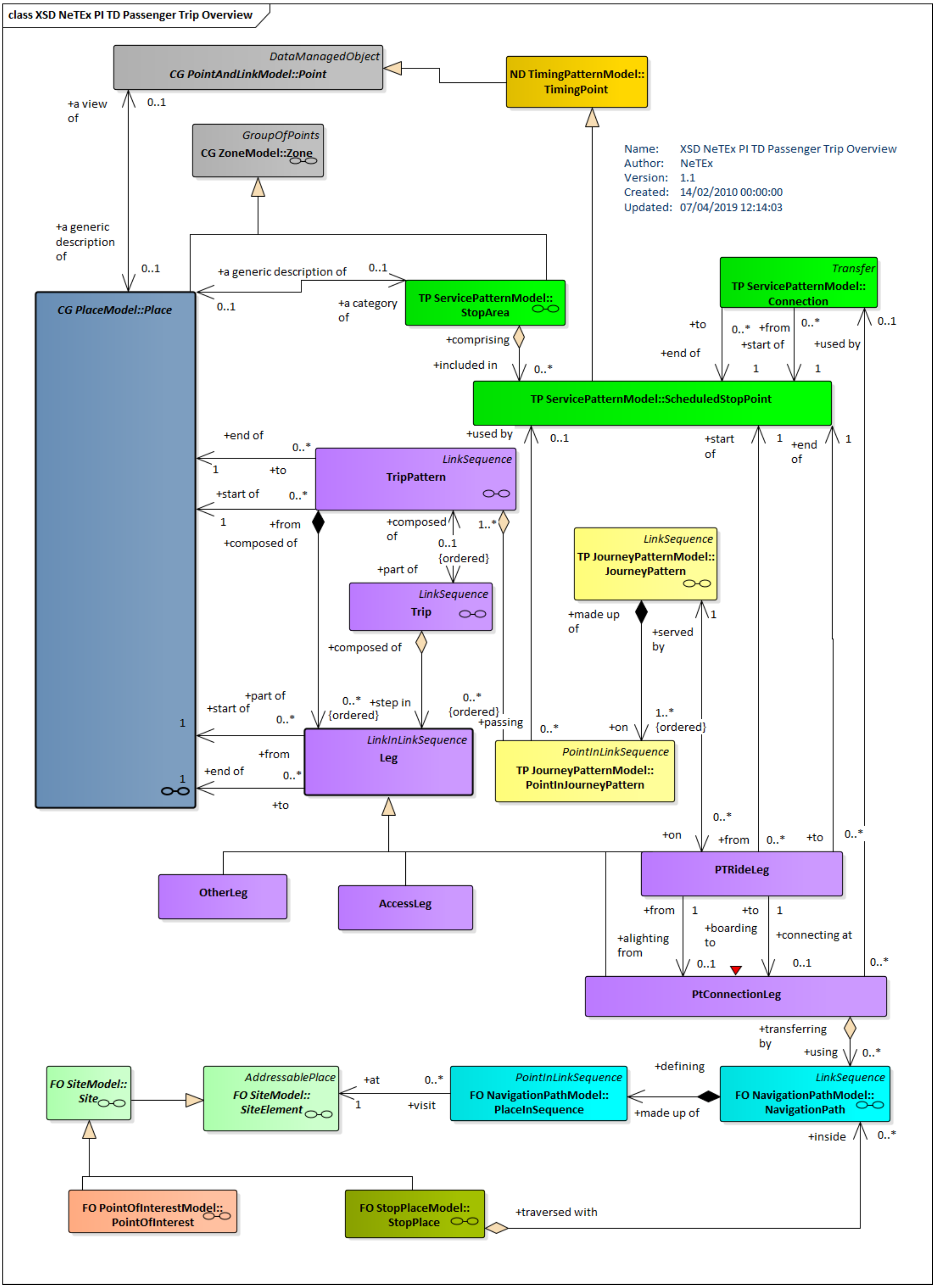


Figure 425 — Passenger Trip – Physical Model: Overview

D.1.1.3 Passenger Trip – Physical model – Details

The following figure shows the physical model for PASSENGER TRIP.

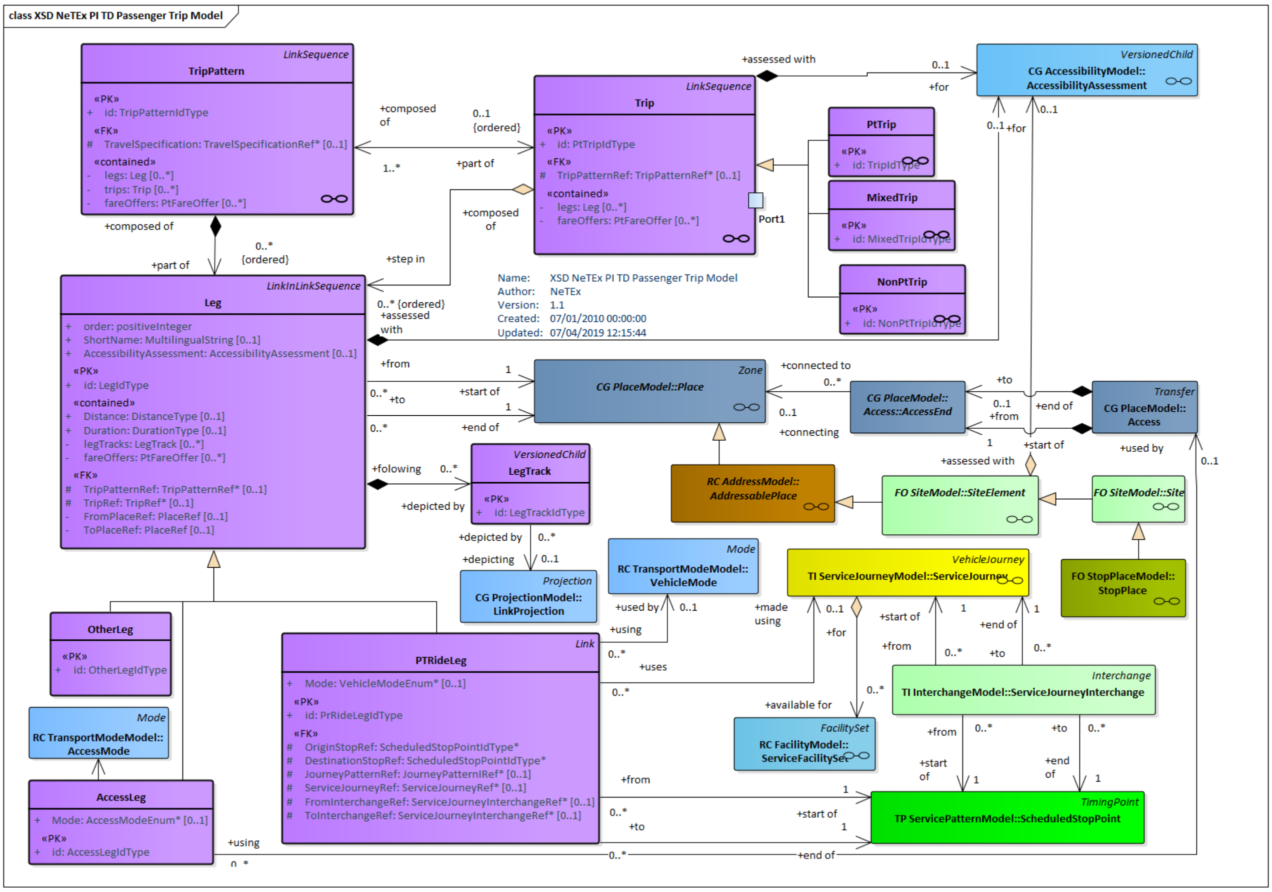


Figure 426 — Passenger Trip – Physical Model: Details

D.1.1.4 PT Connection Leg – Physical model – Details

The following figure shows the physical model for PT CONNECTION LEG

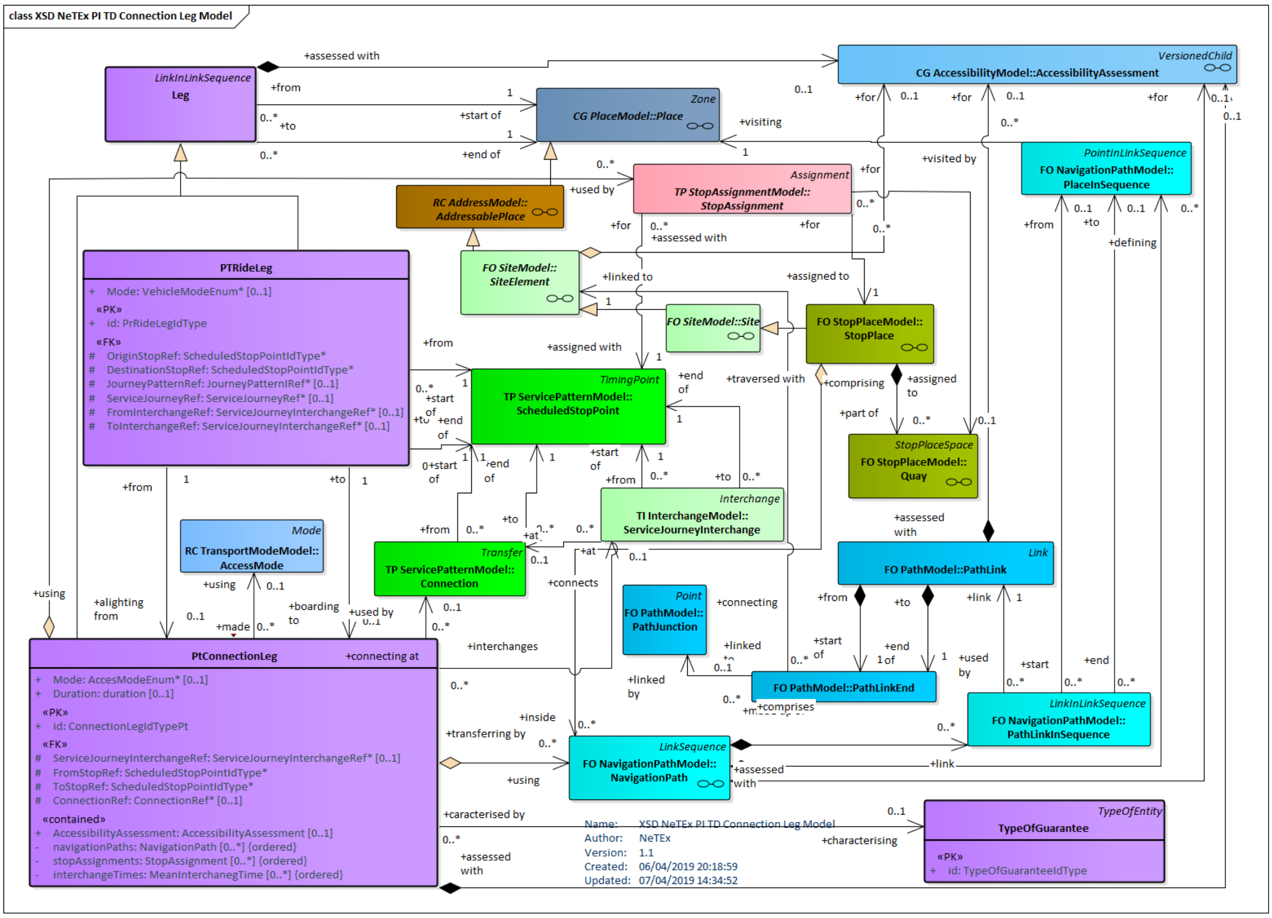


Figure 427 — PT Connection Leg – Physical Model: Details

D.1.1.5 Passenger Trip – Attributes and XSD

D.1.1.5.1 TripPattern – Model Element

A movement of a passenger (or another person, e.g. driver) from an origin to a destination PLACE, done for a specific TRIP REASON.

A TRIP PATTERN may consist of one or more consecutive TRIPs.

Table 393 –*TripPattern* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	TRIP PATTERN inherits from LINK SEQUENCE.
«PK»	***id***	*TripPatternIdType*	1:1	Identifier of TRIP PATTERN.
«FK»	***PtTravelSpecificationRef***	*PtTravelSpecificationRef*	0:1	Reference to a RAVEL SPECIFCATION dscribing the intended consumption.

«cntd»	***legs***	*Leg*	0:*	LEGs that make up TRIP PATTERN.
«cntd»	***trips***	*Trip*	0:*	TRIPs that make up TRIP PATTERN.
«cntd»	***fareOffers***	*PtFareOffer*	0:*	PT FARE OFFERs available for an overall TRIP PATTERN.

D.1.1.5.2 PtTrip – Model Element

A part of a TRIP PATTERN describing the movement of a passenger from one PLACE of any sort to another. A TRIP may consist of one or more consecutive LEGs having some common characteristics.

Table 394 – *PtTrip*– Element

Classification	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	PT TRIP inherits from LINK SEEQUENCE.
«PK»	***id***	*PtTripIdType*	1:1	Identifier of PT PTRIP.
	Duration	*xsd:duration*	0:1	Duration of PT TRIP.
«cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	Overall ACCESSIBILITY ASSESSMENT for the PT TRIP. See NeTEx Part1.
«FK»	***TripPatternRef***	*TripPatternRef*	0:1	TRIP PATTERNs for PT TRIP.
«cntd»	***legs***	*Leg*	0:*	LEGs that make up TRIP.
«cntd»	***fareOffers***	*PtFareOffer*	0:*	PT FARE OFFERs available for PT TRIP.

D.1.1.6 Leg – Model Element

A part of a TRIP PATTERN corresponding to the movement of a user in a single vehicle, or a pedestrian mode such as walking.

Table 395 –*Leg* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*LinkInLinkSequence*	::>	RIDE IN PT TRIP inherits from LINK IN LINK SEQUENCE.
«PK»	***id***	*LegIdType*	1:1	Identifier of LEG.
	order	*xsd:positiveInteger*	0:1	Order of LEG within PT TRIP.
	ShortName	*MultilingualString*	0:1	Short Name of LEG

	Distance	*DistanceType*	0:1	Duration of LEG.
	Duration	*xsd:duration*	0:1	Duration of LEG.
«FK»	***TripPatternRef***	*TripPatternRef*	0:1	TRIP PATTERNs for LEG.
«FK»	***TripRef***	*TripRef*	0:1	Reference to a TRIP of which this is a ride.
«FK»	***FromPlaceRef***	*PlaceRef*	**1:1**	Origin PLACE.
«FK»	***ToPlaceRef***	*PlaceRef*	**1:1**	Destination PLACE.
«cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	ACCESSIBILITY ASSESSMENT for LEG. See NeTEx Part1.
«cntd»	***legTracks***	*LegTrack*	0:*	LEG TRACKs describing spatial projection of LEG.
«cntd»	***fareOffers***	*PtFareOffer*	0:*	PT FARE OFFERs available for LEG.

D.1.1.7 PtRideLeg – Model Element

A part of a trip corresponding to the theoretical movement of a user (passenger, driver) on one and only one public transport vehicle, from one STOP POINT to another, on one JOURNEY PATTERN.

Table 396 –*PtRideLeg* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Leg*	::>	PT RIDE LEG inherits from LEG
«PK»	***id***	*PtRideLegIdType*	1:1	Identifier of a PT RIDE LEG.
«enum»	***VehicleMode***	*VehicleModeEnum*	0:1	TRANSPORT MODE of LEG. See allowed values
«FK»	***OriginStopRef***	*ScheduledStopPointRef*	**1:1**	Origin SCHEDULED STOP POINT of PT RIDE LEG.
«FK»	***Destination-StopRef***	*ScheduledStopPointRefe*	**1:1**	Destination SCHEDULED STOP POINT of PT RIDE LEG.
«FK»	***Journey-PatternRef***	*JourneyPatternRef+*	0:1	JOURNEY PATTERN followed by PT RIDE LEG.
«FK»	***Service-JourneyRef***	*ServiceJourneyRef*	0:1	SERVICE JOURNEY used by PT RIDE LEG.
«FK»	***PtTripRef***	*PtTripRef*	0:1	Reference to a PT TRIP for which this is a pattern.
«FK»	***FromCallRef***	*CallRef*	0:1	CALL at which PT RIDE LEG starts.

«FK»	***ToCallRef***	*CallRef*	0:1	CALL at which PT RIDE LEG ends.
«FK»	***From-InterchangeRef***	*ServiceJourney-InterchangeRef*	0:1	SERVICE JOURNEY INTERCHANGE from which PT RIDE LEG starts.
«FK»	***To-InterchangeRef***	*ServiceJourney-InterchangeRef*	0:1	SERVICE JOURNEY INTERCHANGE at which RIDE ends.

D.1.1.8 AccessLeg – Model Element

A part of a TRIP PATTERN corresponding to the movement of a passenger when not on a public transport vehicle, from an origin PLACE to a SCHEDULED STOP POINT, or a SCHEDULED STOP POINT to a destination PLACE. May reference an ACCESS.

Table 397 –*AccessLeg* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Leg*	::>	ACCESS LEGinherits from LEG.
«PK»	***id***	*AccessLegIdType*	1:1	Identifier of an ACCES LEG.
«enum»	***AccessMode***	*AcessModeEnum*	0:1	MODE of LEG. See allowed values.
«FK»	***AccessRef***	*AccessRef*	0:1	ACCESSS link followed by ACCESS LEG.

D.1.1.9 OtherLeg – Model Element

A part of a trip corresponding to the theoretical movement of a user (passenger, driver) on one and only one public transport vehicle, from one STOP POINT to another, on one JOURNEY PATTERN.

Table 398 –*OtherLeg* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Leg*	::>	ACCESS LEG inherits from LEG.
«PK»	***id***	*OtherLegIdType*	1:1	Identifier of OTHER LEG.
«enum»	***AccessMode***	*AccessModeEnum*	0:1	MODE of LEG. See allowed values.

Table 399 –*LegTrack* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	LEG TRACK inherits from VERSIONED CHILD

«PK»	***id***	*LegTrackIdType*	1:1	Identifier of v.
«cntd»	***Projection***	*LinkProjection*	0:1	LIN PROJETCION of

D.1.1.10 PtConnectionLeg – Model Element

A part of a trip corresponding to the movement of a passenger to transfer from one SERVICE JOURNEY to another made at a CONNECTION from one STOP POINT to another.

Table 400 –*RideInterchange* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*Link*	::>	PT CONNECTION LEG inherits from LINK.
«PK»	***id***	*PtConnectionLegIdType*	1:1	Identifier of PT CONNECTION LEG.
	Duration	*xsd:duration*	0:1	Time to make interchange.
«FK»	***ServiceJourney-InterchangeRef***	*ServiceJourney-InterchangeRef*	0:1	SERVICE JOURNEY INTERCHANGE for PT CONNECTION LEG.
«FK»	***FromStopRef***	*ScheduledStopPoint-IdType*	1:1	Arrival stop for PT CONNECTION LEG.
«FK»	***ToStopRef***	*ScheduledStopPoint-IdType*	1:1	Departure stop for PT CONNECTION LEG.
«FK»	***ConnectionRef***	*ConnectionRef*	0:1	CONNECTION link for PT CONNECTION LEG.
«enum»	***Mode***	*AccessModeEnum*	0:1	Mode used to make PT CONNECTION LEG.
«cntd»	***navigationPaths***	*NavigationPath*	0:*	NAVIGATION PATH by which to make interchange.
«cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	ACCESSIBILITY ASSESSMENT for use of interchange. See NeTEx Part1.

D.1.2 Passenger Fare Offer Model

The PT FARE OFFER model defines the available fares for a specific a PT LEG in a PT TRIP or for the whole TRIP PATTERN. The fare price may be restricted to particular products and conditions as described by a TRAVEL SPECIFICATION with SPECIFIC PARAMETER ASSIGNMENTS.

D.1.2.1 Passenger Fare Offer – Conceptual MODEL

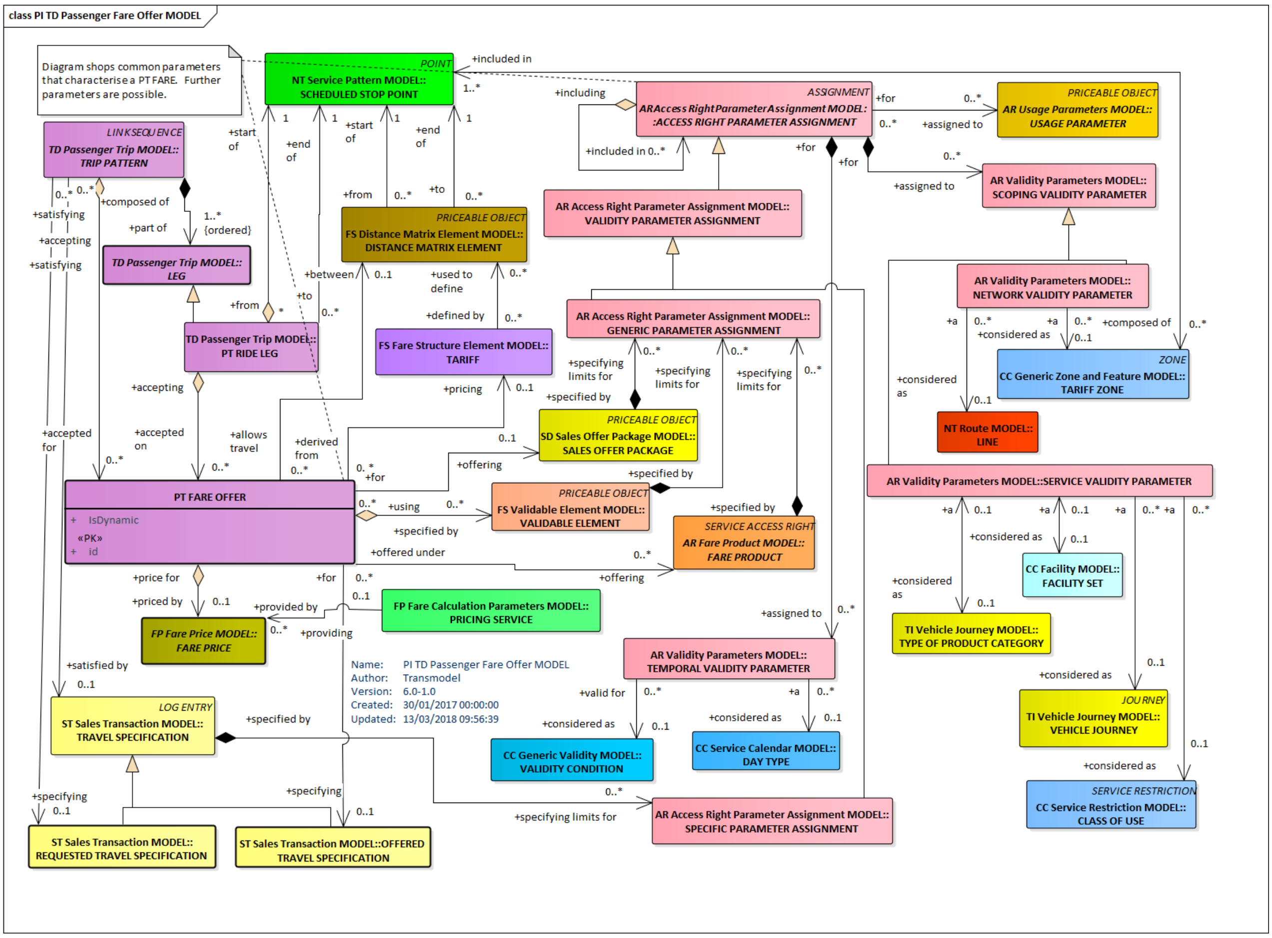


Figure 428 — Passenger Fare Offer – Conceptual Model

D.1.2.2 Passenger Fare Offer – Physical model – Overview

The following figure introduces the physical model for a PASSENGER FARE. A PT FARE OFFER describes the FARE PRODUCTs and their FARE PRICE that are available for a particular TRIP and / or PT RIDE LEG.

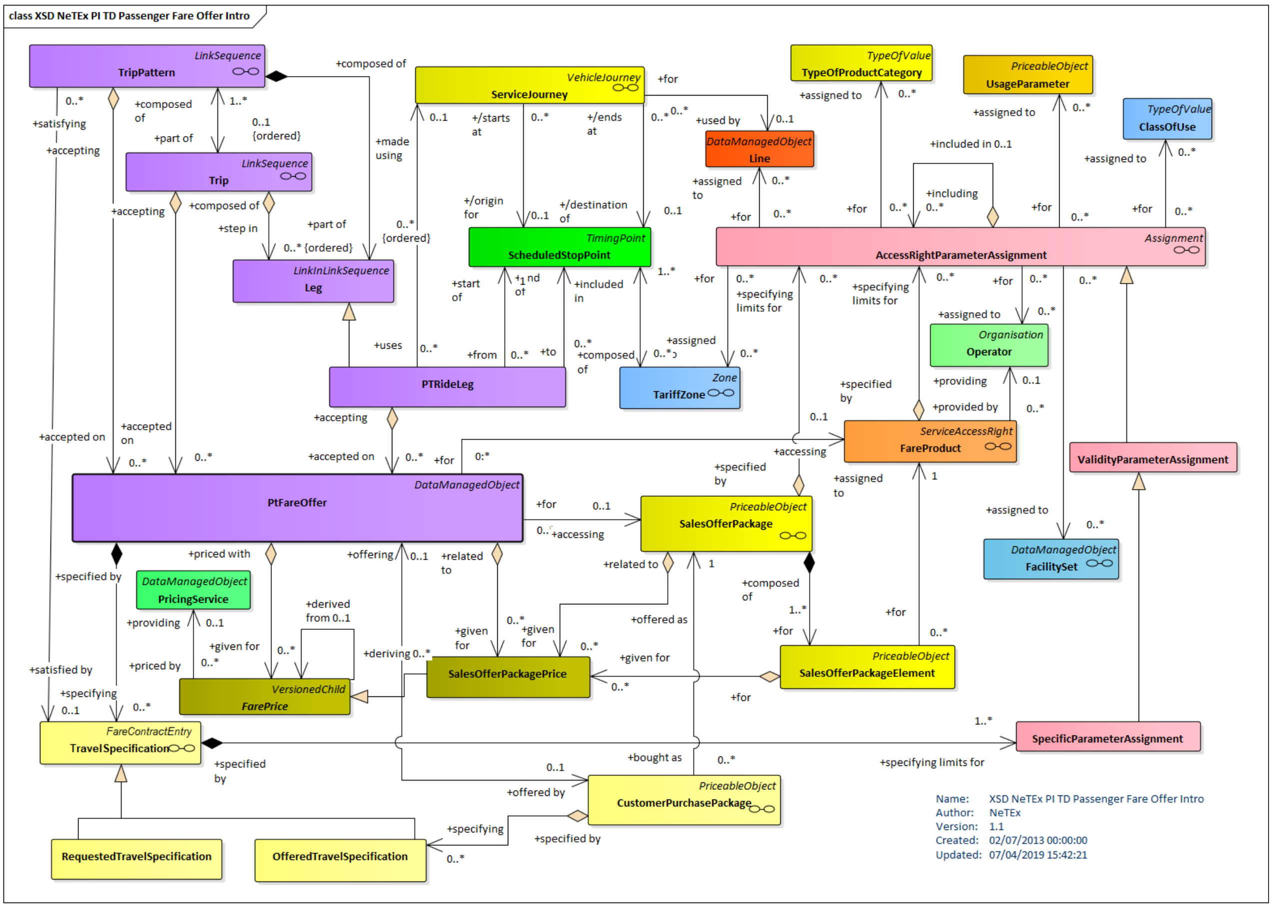


Figure 429 — Passenger Fare Offer – Physical Model: Overview

D.1.2.3 Passenger Fare Offer – Physical model – Details

The following figure shows the physical model for PASSENGER FARE OFFER.

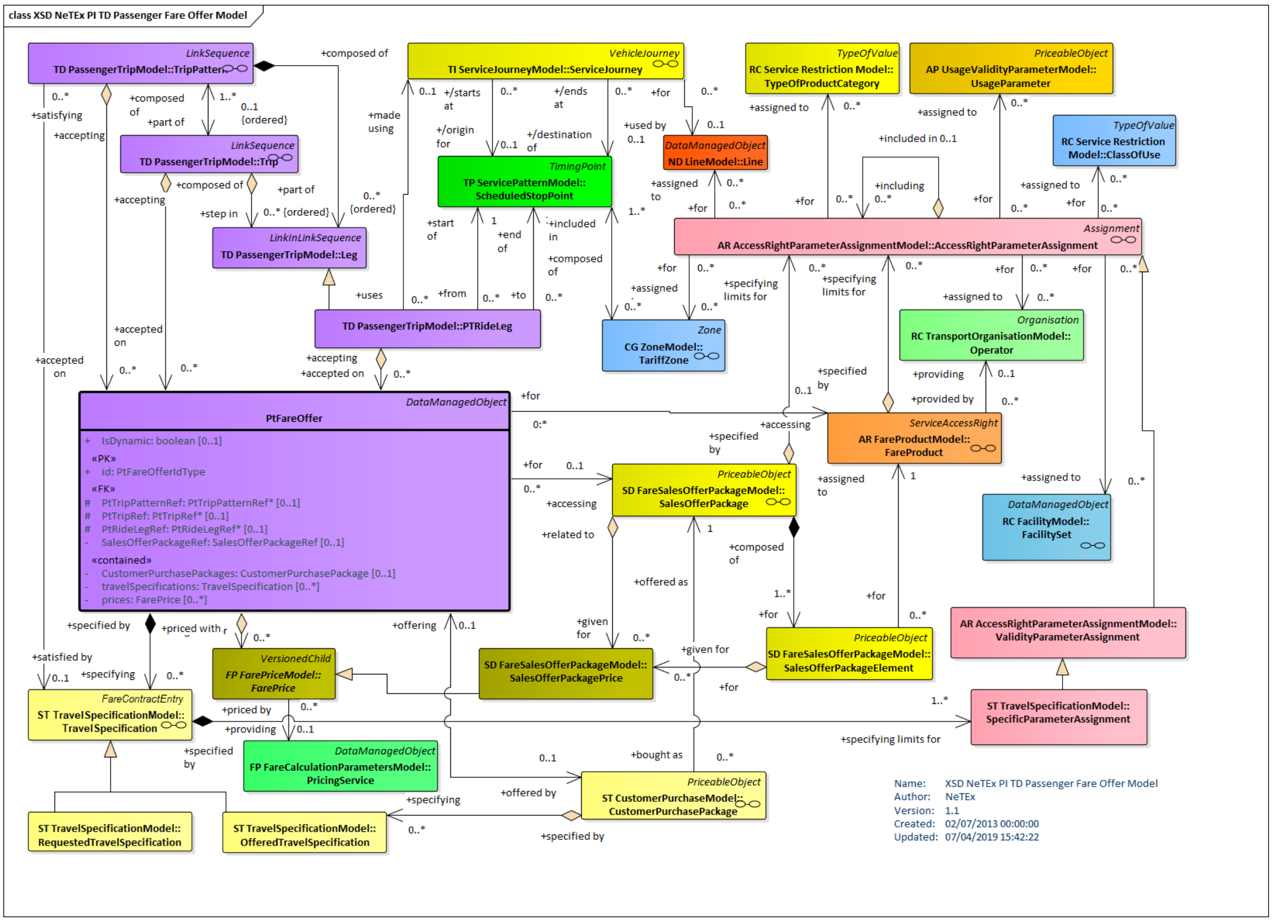


Figure 430 — Passenger Fare Offer – Physical Model: Details

D.1.2.3.1 PtFareOffer – Model Element

An available fare for a specific PT RIDE LEG in a PT TRIP or for the whole PT TRIP, or for the whole TRIP PATTERN. The fare may be restricted to particular products and conditions as described by SPECIFIC PARAMETER ASSIGNMENTS.

Table 401 – *PtFareOffer* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	PT FARE inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PrFareOfferIdType*	1:1	Identifier of PT FARE OFFER.
	isDynamic	*xsd:boolean*	0:1	Whether the fare was fetched dynamically.
«FK»	***TripPatternRef***	*TripPatternRef*	0:1	TRIP PATTERNs for PT FARE OFFER.

«FK»	***PtTripRef***	*PtTripRef*	0:1	Reference to a PT TRIP of which this is a ride
«FK»	***PtRideLegRef***	*PtRideLegRef*	0:1	Reference to a PT RIDE LEG.
«cntd»	***travel-Specifications***	*TravelSpecification+*	0:*	TRAVEL SPECIFICATIONs relating to PT FARE OFFER.
«cntd»	***Customer-Purchase-Package***	*Customer-Purchase-Package*	0:*	CUSTOMER PURCHASE PACKAGE.
«cntd»	***prices***	*SalesOfferPackagePrice*	0:*	SALES OFFER PACKAGE PRICEs associated with the PT FARE OFFER

Annex E
(informative)

Proposed model for Parking Tariff

Parking is often a component of a multimodal trip by car and public transport and some FARE PRODUCTs may include parking components; for example, parking at a station may be included in the overall price of a ticket. The following model is provided on an informative basis to show how PARKING TARIFFs can be included in NeTEx fare structures in a uniform manner that allows them to be included in products. It makes use of the PARKING element defined in NeTEx Part1 which describes the location and nature of a car as a specialisation of the SITE element.

The parking tariff model is intended to be interoperable with the DATEX2 model so that data can be exchanged with a straightforward transform.

E.1 Parking Tariff

E.1.1 Parking Tariff – Conceptual MODEL

The PARKING TARIFF MODEL describes tariffs for Parking.

PARKING TARIFFs may be specified for a PARKING and or its subdivision PARKING AREA using a PARKING PROPERTies element.

A PARKING TARIFF is made up of one or more PARKING CHARGE BANDs for a given set of VEHICLE TYPEs. Each CHARGE BAND can have a PARKING TARIFF PRICE.

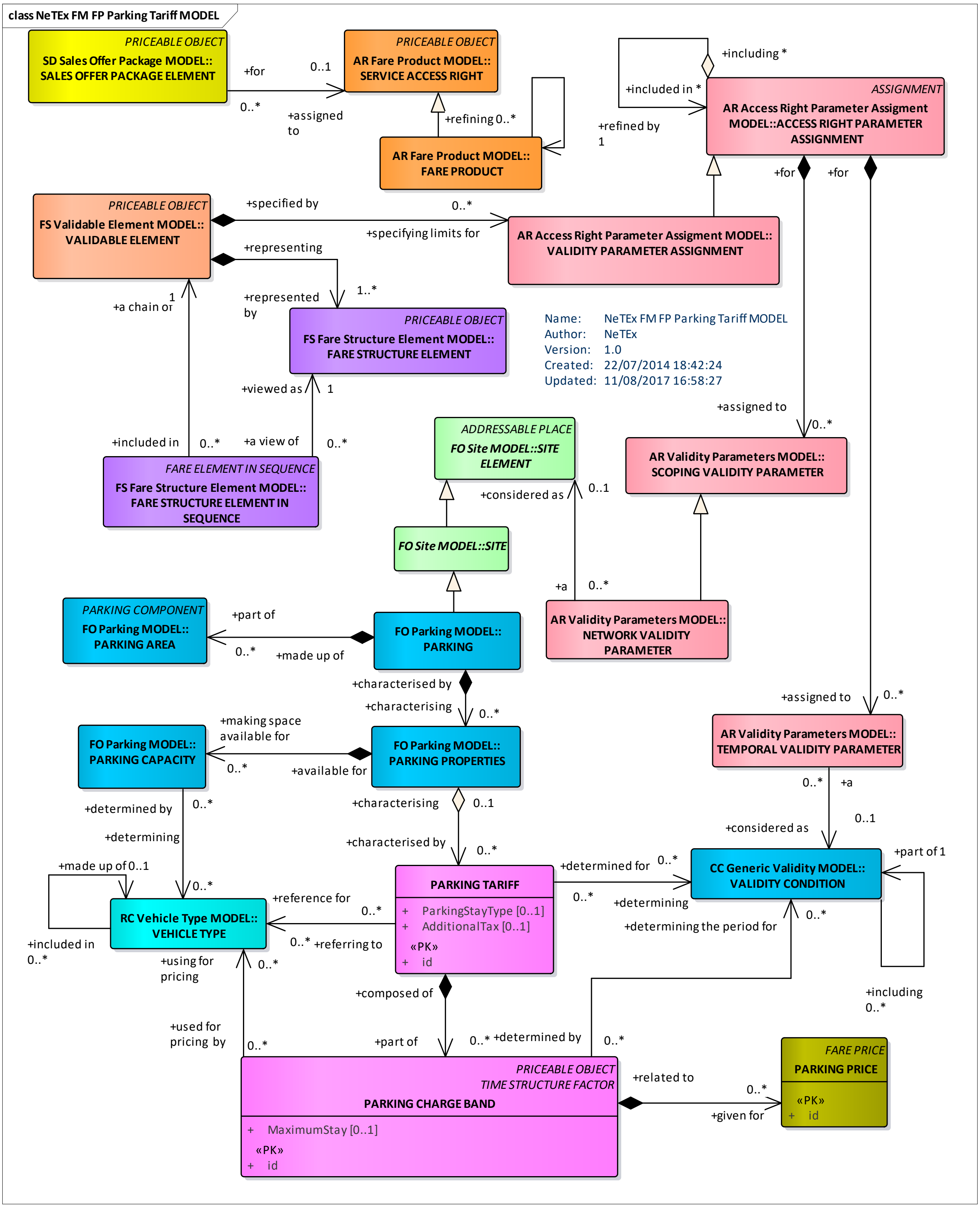


Figure 431 — Parking Tariff – Conceptual Model

E.1.1.1.1 Parking Tariff – Physical model

The following figure shows the physical model for PARKING TARIFFs.

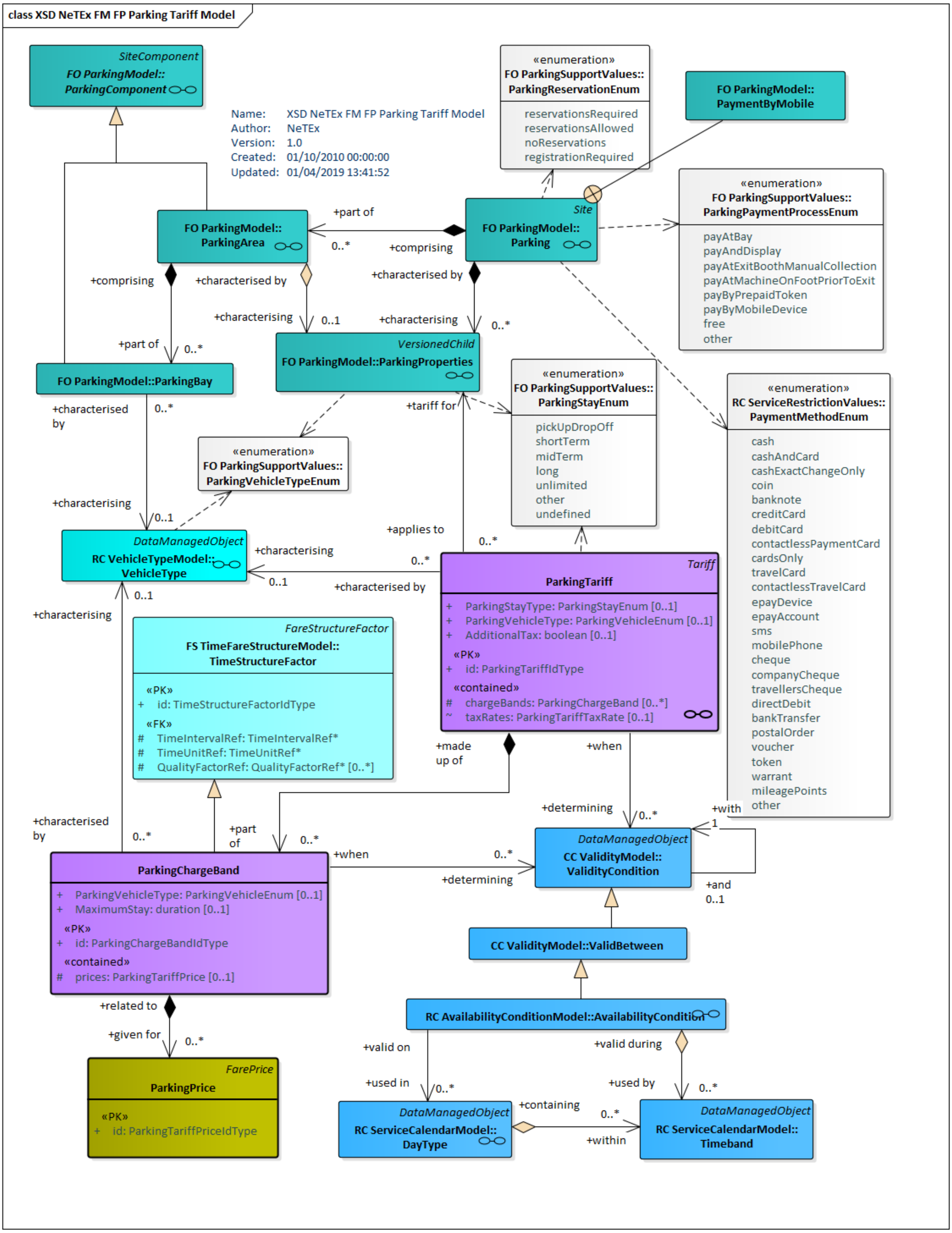


Figure 432 — Parking Tariff – Physical Model

E.1.1.1.2 Parking Tariff – Attributes and XSD

E.1.1.1.2.1 ParkingTariff – Model Element

A set of parking CHARGE BANDs that describe the cost if using a PARKING or PARKING AREA.

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*DataManagedObject*	::>	PARKING TARIFF inherits from TARIFF.
«PK»	***id***	*ParkingTariffIdType*	1:1	Identifier of PARKING TARIFF.
XGRP	***Tariff-Description-Group***	***xmlGroup***	0:1	See Tariff.
XGRP	***Tariff-Organisation-Group***	***xmlGroup***	0:1	See Tariff.
XGRP	***TariffTimeGroup***	***xmlGroup***	0:1	See Tariff.
XGRP	***TariffQuality-Group***	***xmlGroup***	0:1	See Tariff.
«enum»	***ParkingStayType***	*ParkingStayEnum*	0:1	Reference to PARKING STAY TYPE of PARKING TARIFF. See NeTEx Part1.
«enum»	***Parking-VehicleType***	*ParkingVehicleEnum*	0:1	Reference to VEHICLE TYPE of PARKING TARIFF. See NeTEx Part1.
	AdditionalTax	*xsd:boolean*	0:1	Whether additional tax is charged on top of rates.
«cntd»	***chargeBands***	*ParkingChargeBand*	0:*	Charge bands of PARKING TARIFF.
XGRP	***TariffPriceGroup***	***xmlGroup***	0:1	See Tariff.

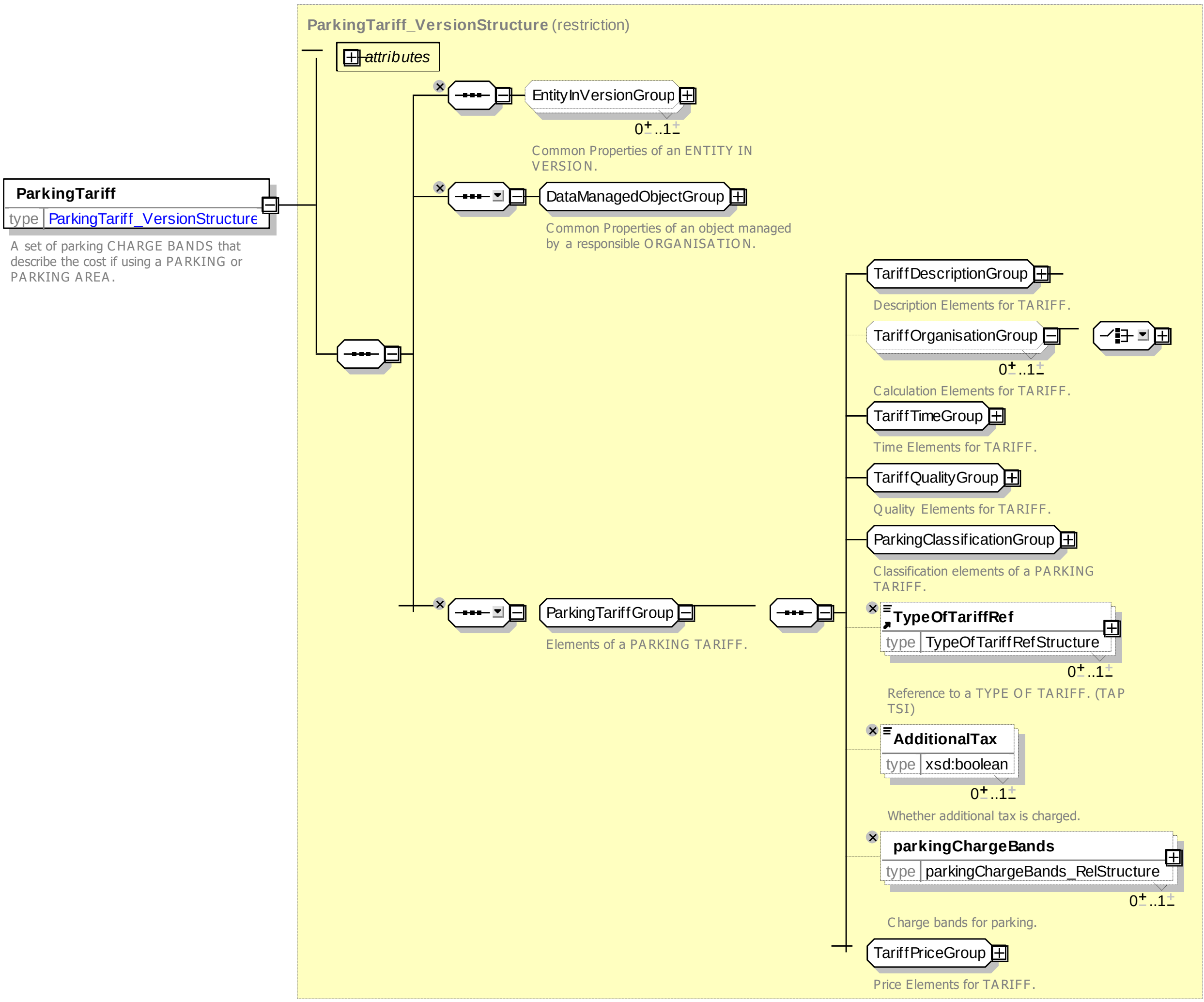


Figure 433 — *ParkingTariff* — XSD

E.1.1.1.2.2 ParkingChargeBand – Model Element

Parking charges that describe the cost of using a PARKING or PARKING AREA for a given period.

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*TimeStructureFactor*	::>	PARKING CHARGE BAND inherits from TIME STRUCTURE FACTOR.
«PK»	***id***	*ParkingChargeBandIdType*	1:1	Identifier of PARKING CHARGE BAND.
«enum»	***Parking-VehicleType***	*ParkingVehicleEnum*	0:1	Vehicle type for of PARKING CHARGE BAND.
	MaximumStay	*xsd:duration*	0:1	Maximum stay for PARKING CHARGE BAND.
«cntd»	***prices***	*ParkingPrice*	0:1	Prices for PARKING CHARGE BAND.

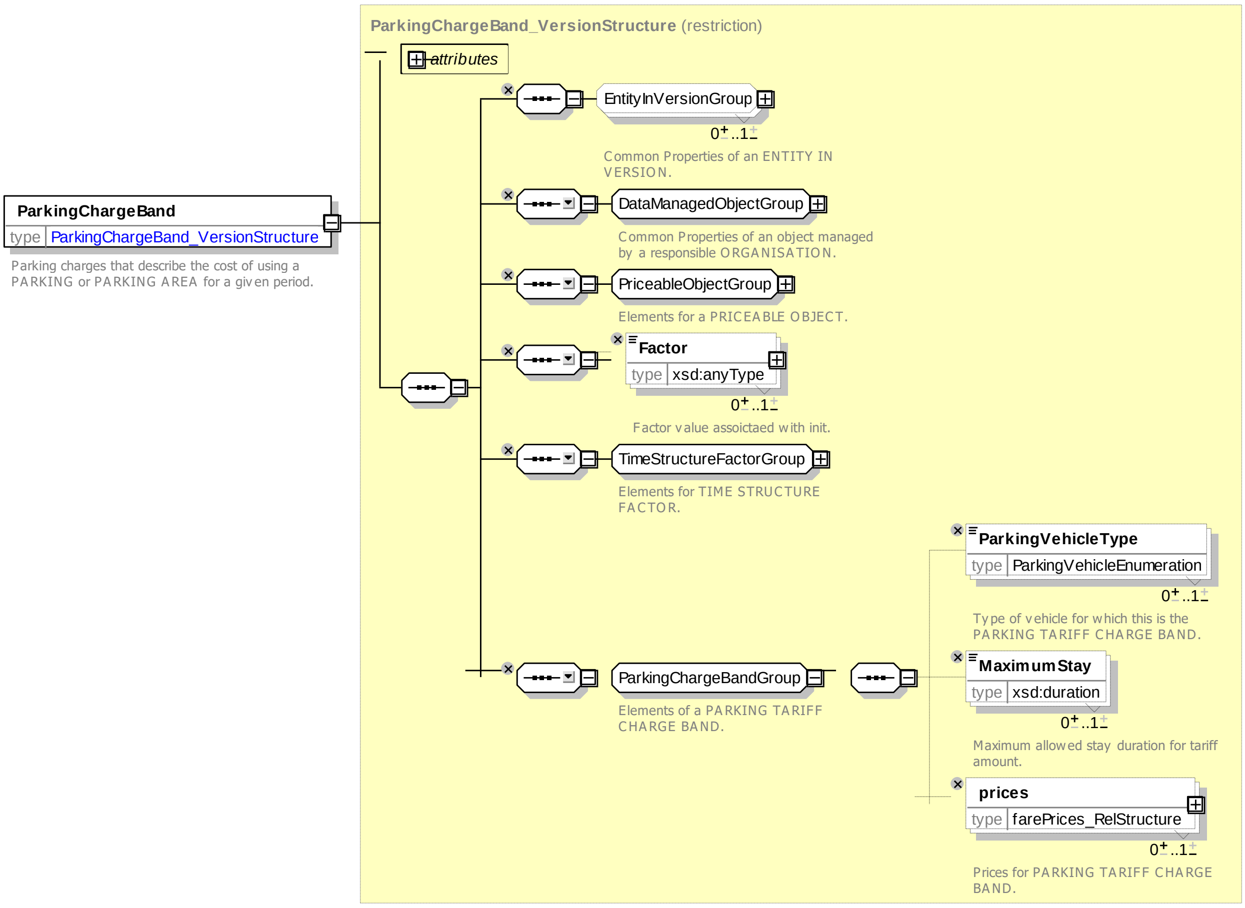


Figure 434 — *ParkingChargeBand* — XSD

E.1.1.1.2.3 ParkingPrice

A specialisation of FARE PRICE used to specify the price of a PARKING CHARGE BAND

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*FarePrice*	::>	PARKING PRICE. inherits from FARE PRICE.
«PK»	***id***	*ParkingPriceIdType*	1:1	Identifier of PARKING PRICE.
		Choice		Price for one of the following
«FK»	***ParkingTariffRef***	*ParkingTariffRef*	0:1	Reference to a PARKING TARIFF.
«FK»	***ParkingCharge-BandRef***	*ParkingChargeBandRef*	0:1	Reference to a PARKING CHARGE BAND.

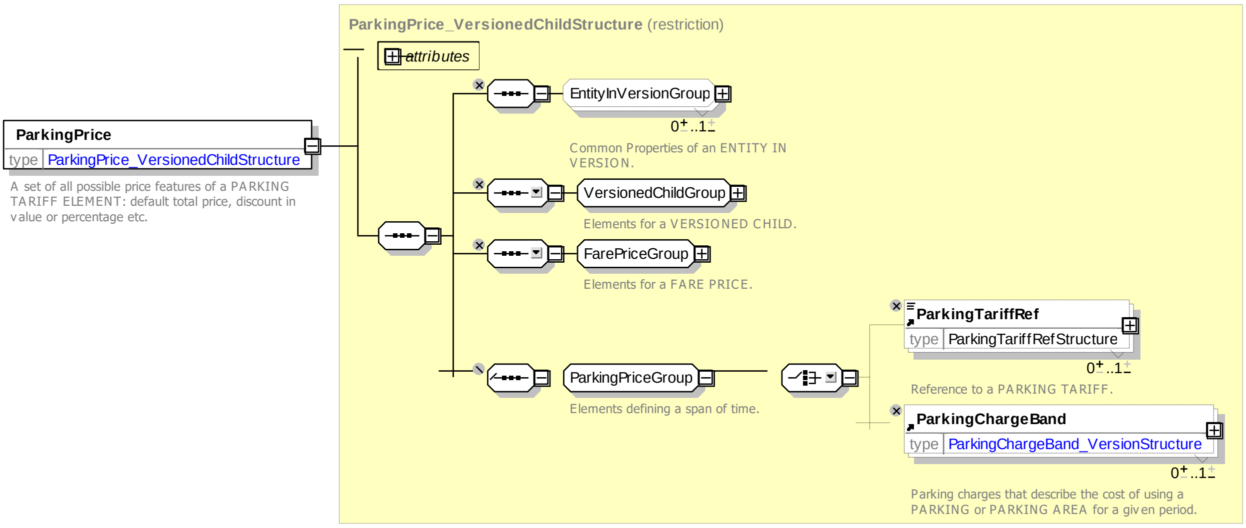


Figure 435 — *ParkingTarifffPrice* — XSD

Annex F
(informative)

Changes in Version 1.1

F.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.

F.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.

F.3 List of Changes

Optimisation: ***FareStructureElement*** add direct reference to ***TimeIntervals*** & ***geographical*** intervals, Also allow in-lining of ***DistanceMatrixElements***

CR0041 ***FrequencyOfUs: A***Add value "*twiceAday*".

CR0045 ***LogEntry***: Add.***FareContractEntry*** renamed from ***FareContractEvent***, make subtype of ***LogEntry***

- ***Blacklist***: Revise use of Blacklist. NB this is functionally, but not syntactically backwards compatible.

CR0047 ***FarePrice***: Add ***RuleStepResult*** to allow Tax, ***FareRule*** allow chaining

CR0049 Align with TM6 Changes

- ***CustomerAccount:*** Add ***CustomerAccountStatus***, ***TypeOfCustomerAccount***.
- Add ***CustomerSecurityListing, CustomerAccountSecurityListing FareContractSecurityListing, RetailDeviceSecurityListing, TravelDcoumentSecurityListing*** add
- Add ***CustomerPurchasePackage***, ***CustomerPurchasePackageAssignment***
- ***SecurityList*** – generalise ***WhiteList*** and ***BlackList.***
- ***TypeOfFareContractEntry*** renamed from ***TypeOfFareContractEvent***.
- ***FareTable:*** Add ***TypeOfTravelDocument***
- ***Customer***: Add ***CustomerEligibility, UserProfileEligibility, CommercialProfileEligibility, ResidentialQualificationEligibility.***
- TM Alignment: Rename ***PassengerContract*** ➔ ***FareContract.***
- TM Alignment: Rename ***PassengerContractEntry*** ➔ ***FareContractEntry.***
- TM Alignment: Rename ***PassengerContractSecurityListing*** ➔***FareContractSecurityListing.***
- TM Alignment: Rename ***TypeOfPassengerContract*** ➔ ***TypeOfFareContract.***
- TM Alignment: Rename ***TypeOfPassengerContractEntry*** ➔ ***TypeOfFareContractEntry***.

- TM Alignment: Rename ***SalesPackage* ➔ *SalesOfferPackage***.
- TM Alignment: Rename ***SalesPackageElement* ➔ *SalesOfferPackageElement***.
- TM Alignment: Rename ***SalesPackageSubstitition* ➔ *SalesOfferPackageSubstitition***.
- TM Alignment: Rename ***TypeOfSalesPackage* ➔ *TypeOfSalesOfferPackage***.
- TM Alignment: Rename ***SalesPackageSubstitition* ➔ *SalesOfferPackageSubstitition***.
- TM Alignment: Rename ***GroupOfSalesPackages* ➔ *GroupOfSalesOfferPackages***.
- TM Alignment: ***RenamePiQuery* ➔ *PiRequest***.

CR0051 ***Tariff***: Add ***LineRef***, ***documentLinks/ infoLink.***

CR0057 ***PriceableObject***: Add ***UML&/ Documentation***.

Fix: Align with UML

- ***VersionChild /alternativeNames:*** Add visibility on restriction many descendant elements.
- ***Accommodation*** and ***OnBoard***: Add missing attributes.
- ***Authority***: Add missingCountryRef.
- ***TypeOfFrame***: Add missing typesOfEntity /***TypeOfEntity***.
- ***TypeOfPassengerInformationEquipment***: Add missing attribute.
- ***SanitaryEquipment***: Add back missing ***NumberofToilets***.
- ***BoardingPermission***: Correct typo.

Fix: Support PLACE to PLACE travel.

- ***AccessRightParameter***: Add ***AddressRef***, ***TopoographicPlaceRef*** and ***PlaceUseEnumeration*** with values: *'startAt, endAt, via, retrictTo, other'.*

EURA00x ***DistanceMatrixInverseRef*** for backwards direction of reference. Revise constraints.

EURA0010 Improve ***CustomerPurchasePackage***.

- Allow inlining of ***CustomerPurchasePackages*** in a ***FareContract***.

EURA0029 Add ***EligibilityChangePolicy*** usage parameter:-

- ***EligibilityChangePolicy***: Add ***OnBecomingEnumeration*** with values*: "automatic, invite, noAction"*
- ***EligibilityChangePolicy*** Add O***nCeasingEnumeration*** with value: *"immediateTermination, useUntilExpiry, terminateAfterGracePeriod, automaticallySubstituteProduct, noAction, other*"

EURA0040 Add support for subscriptions:-

- Add ***Subscribing*** usage parameter with attributes ***SuspensionPolicy***, ***QualificationPeriod***, ***QualificationPercent***, ***MinimumSuspensionPeriod***, ***MaximumSuspensionPeriod***, ***MaximumNumberOfSuspensionsPerTerm***
    - ***Subscribing***: Add SubscriptionRenewalPolicy attribute with *values "automatic, manual, automaticOnConfirmation, none*
    - ***Subscribing***: Add ***SubscriptionTermEnumeration*** attribute with values: "*fixed, variable, openEnded."*
- ***DistributionChannelType*** Add values "*onlineAccount, postal".*
- ***PaymentMoment***: Add values "*subscriptionOnly, also onCheckIn, inAdvanceOnly, beforeBoarding, Only, onBoardingOnly".*

- ***FareProduct \ ConditionSummary***: Add ***PenaltyIfWithoutTicket***, ***AvailableOnSubscription***.

- ***Reselling***: Add ***typesOfPaymentMethods***/***TypeOfPaymentRef***

EURA0040 Parameter Assignment, add additional validity parameters;-

- ***AccessRightParameterAssignment*:** Add ***TypeOfTariffRef, TypeOfFareStructureFactor, TypeOfFares-StructureFactorRef, TypeOfPriceingRuleRef, ChargingMethodRef, TypeOfPaymentMethodRef, TypeOf-MachineReadability, TypeOfFareTableRef, TrainElementRef, TrainComponentLabelAssignmentRef.***

EURA0042 ***PricingRule***: Add ***Currency***.

EURA0043 ***FareZone***: Add ***AuthorityRef*** / ***OperatorRef***, ***GroupOfOperatorsRef***.

EURA0043 ***FarePointInPattern***: Add ***IsForbidden*** attribute.

EURA0050 ***PurchaseWindow***: Add ***PurchaseAction*** with values: *"purchase, reserve, orderWithoutPaying, payForPreviousOrder, other, seatMap and openSeating*".

- ***Reserving***: Add ***SeatAllocationMethod***

- ***Reserving***: Add ***ReservationExpiryPeriod***.

EURA0051 ***RoundTripType*** add *values "returnOut, returnBack*".

EURA0052, EURA40 Support suspension of a season pass.

- Add ***Suspending*** usage parameter, with attributes ***SuspensionPolicy, QualificationPeriod, QualificationPercent, MinimumSuspensionPeriod, MaximumSuspensionPeriod, MaximumNumberOfSuspensionsPerTerm***

- Add ***SuspensionPolicy*** with values "*none, forCertifiedIllness, forParentalLeave, forHoliday, forAnyReason".*

- ***Refunding*** / ***RefundType***: Add values *"unused, earlyTermination*".

EURA0054 Add a Seat reference to assignable parameters.

- Add a new package with ***PassengerSeatingRef***

EURA0068 Specify conditions for changing group size

- ***Exchanging*** / ***TypeOfExchange*** : add value "*changeGroupSize*."

- ***GroupTicket*** add new attribute GroupSiz***e***Changes with enum values noChanges, free, charge, steppedCharge.

- ***Refunding***/ ***RefundType*** aAdd value "changeOfGroupSize".

EURA0062 ***CompanionProfile***: Add ***CompanionRelationshipType*** with values "*anyone, grandparent, parent, child, grandchild, colleague, family, spouse, partner, colleague, teacher, pupil."*

EURA0065 ***Transferability***: Add ***SharedUsage*** attribute with values "*oneAtATime, severalAtATime, severalSpecifiedCompanionsAtATime."*

EURA0067 ***FulfilmentMethodType***: Add value "*courier*".

EURA0071 ***FareDemandType***: Add values "*superOffPeak, specialEvent".*

EURA0072 ***StartTimeAtStop***: Make ***StartTime*** optional.

- ***FareDemandType***: *Add StopUseConstraint with values arriving. departing, passingThrough.*

EURA0073 ***UsageValidityPeriod***: Add StartConstraintType with values fixed, variable, fixedWindow

EURA0075 ***ChargingPolicy***: Add ***TravelBillingPolicy*** with values; "*billAsYouGo, billOnThreshold, billAtFareDayEnd, billAtPeriodEnd "*

EURA0076 ***Reserving***: Add ***IsFeeRefundabl***".

EURA0077 ***FareProduct***: Add new relationship between and ***Tariff***.

- Add new ***tariffs***/***TariffRef*** attribute to ***FareProduct***.

EURA0078 ***FareProduct*** Allow more than one reference to a ***GroupsOfSalesOfferPackageRef*** (i.e. make relationship many-to-many.)

EURA0081 ***FareProduct*** Make relationship with ***TypeOfFareProduct*** many-to-many.

- ***AccessRightParamaterAssignment*** Add a new attribute ***PassengerSeatRef***
- ***SalesTransaction*** Add new ***TravelDocumentRef*** and ***RetailDeviceRef***.

EURA0085 ***UsageValidityPeriod***: Add ***ActivationMeans*** with values "*noneRequired, checkIn, useOfValidato useOfMobileDevice, automaticByTime, automaticByProximity, other"*

EURA0087 Support partial refunds of passes

- ***Reselling***:
  - ***RefundType*** add values "*unused, earlyTermination".*
  - Add ***ExchangableFromPercentUse*** and ***ExchangableUntilPercentUse***.
  - ***PurchaseWhen:*** Add value "*withinSpecifiedWindow".*
  - Add ***EffectiveFrom*** with values "*anytime, nextInterval, nextInstallment, never."*
  - Add ***NoticePeriod***
- ***Refunding***:
  - Add ***RefundPolicy*** with values "*illness, death, redundancy, maternity, other, etc".*
  - Add ***RefundBasis*** with values "*unusedDays, unusedWeeks,unusedMonths, other".*

EURA0088 Specify if start of validity is variable or fixed.

- ***UsageValidityPeriod*** Add ***StartConstraint*** with values "variable / fixed".
- ***UsageValidityPeriod***: Add ***startOnlyOn*** / ***DayType*** attribute so that any required day of week, day of month, month of year can be indicated.
- ***UsageValidityPeriod***: Add ***FixedStartWindow*** with attributes ***MaximumServicesBefore***. ***FlexiblePeriodBefore***, ***MaximumServicesAfter***, ***FlexiblePeriodAfter***.
- ***UsageTriggerEnumeration***: Add values "*enrolmen, t reservation*".
- ***UsageEndEnumeration***: Add value "*eligibilityExpiry*, *deregistration*".

EURA0089 ***USerProfile***: ***ProofOfIdentity*** Add new value "*birthCertificate*"

EURA0090 ***PenaltyPolicy***: Add a new attribute ***MaximumNumberOfFailToCheckOutEvents***.

EURA0091 ***Exchanging***/ ***TypeOfExchange*** Add values "*sameProductLongerJourney sameProducShorterJourney"*

Eura0093 ***InterChanging***: Add ***RegisterBreak*** with values *"none, markByStaff, markByValidator, markByMobileApp, other"*

EURA0094 ***StepLimit***: Add values "*networks, operators, countries* ".

UK0006 ***FareTable***: Add ***TariffZoneRef, LineRef, FareZoneRef, TariffRef, LineRef, ScheduledStopPointRef FareStructureElementInSequenceRef, SectionRef***

UK0007 ***FareTable*** Allow direct containment of ***FarePriceRef***, Allow ***xxPriceRefs*** directly in ***FareTable*** / cells.

- ***FareTableRow*** and ***FareTableColumn***: Add ***VersionOfObjectRef***.
- ***FareTable***: Allow nesting column headings and rows

UK0009 ***TravelSpecification***: Add ***TypeOfTariffRef*** and ***FareElementInSequenceRef***

UK0012 ***Tariff***: Add ***GroupOfOperatorRef***.

UK0013 ***FareZone***: Add ***ZoneTopology*** values *"annular, sequence, overlappingSequence*".

UK0014 ***FareZone***: Add ***ScopingMethod*** with values "*explicitStops, implicitSpatialProjection, implicitSpatialProjection.*"

UK0018 ***Inteval:*** Add ***TypeOfInterval***

UK0018 ***FarePointInPattern:*** Add ***IsFareStage***.

UK0020 ***FareZone***: Add ***contains*** relationship.

UK0021 Add ***SalesOfferEntitlementGiven*** and ***SalesOfferEntitlementRequired*** usage parameters.

UK0022 ***FarePrice***: Add ***PrivateCode***

- ***FareProductPrice***: Add ***Description***.
- ***PriceableObject***: Add ***InfoLinks***.

UK0023 ***FareTable*** / specifics : Add ***FareSectionRef***.

UK0027 Extend **CustomerPurchasePackage** implementation:-

- **TravelDocument**: Add ***PassengerSeatRef, TrainElementRef, CustomerPurchasePackageRef.***
- **SpecificParameterAssignment**: Add **AccessNumber.**
- **CustomerPurchasePackage** Add relationship ***travelDocuments \ TravelDucment.***
- ***CustomerPurchasePackage***: Add ***CustomerPurchasePackageStatus*** with values "*reserved,ordered, paidFor, unused, activated partiallyUsed, used, archived".*
- ***CustomerPurchasePackageElement***: Add ***MarkedAs*** to ***CustomerPurchasePackageElement***.
- ***CustomerPurchasePackageElement***: Add ***CustomerPurchasePackageElementAccess***.
- Modularise: Separate out *netex_typeOfravelDocumentPackage.xsd* from *netex_ travelDocumentPackag.xsd*

- Modularise: Move *netex_travelDocumentPackage.xsd* from \fares to *netex_salesTransaction.xsd*

UK0028 Extend ***FareContract*** support rfom TM6:-

- ***FareContract***: Add ***CustomerAccountRef***
- ***CustomerAccount***: Add ***AccountStatusType***.
- ***Customer***: Add ***Email*** attribute.
- ***CustomerAccount***: Add relationship ***fareContracts / FareContract***
- Remove ***fareContractEntries*** relationship from ***CustomerAccount*** : Use relationship on ***FareContract***. N.B. BREAKAGE!
- ***CustomerAccount***: Add relationship ***customerPurchasePackageRefs*** / ***CustomerPurchasePackage*** to ***CustomerAccount***.
- ***CustomerPurchasePackage***: Add ***CustomerRef***, ***CustomerAccountRef*** and ***FareCOntractRef***.
- Add ***PassengerSeatRef*** and ***TrainElementRef*** to ***TravelDocument***.
- Add ***PrivateCode*** to ***TravelDocument***
- ***TravelSpecification***: Add view element ***TravelSpecificationSummaryView***.
- ***CustomerPurchasePackage***: Add view element ***TravelSpecificationSummaryView***.

UK0032 ***ResidentialEligibility***: Add ***StartDate***, ***EndDate***.

UK0033 ***ResidenceQualification***: Add ***ResidenceType*** a with enumerated values; *"live, work, study, born".*

UK0034 ***TravelDocument*** remove from ***FareFrame***.

UK0038 ***FareStructureElementinSequence***: Add ***MinimumAccess*** and ***MaximumAccess***.

UK0040 Simplyiy Product definition:-

- **SupplementProduct**: Add ***SupplementProductType*** with values "*seatReservation, bicycle, dog, animal, meal, wifi, event, upgrade penalty, journeyExtension".*
- ***PreassignedFareProduct:*** Add ***ProductType*** with value "*; singleTrip, dayReturnTrip, periodReturnTrip, multTrip, dayPass, periodPass,shortTrip.*
- **AmountOfPriceUnit*:*** Add ***ProductType*** with values " *carnetPass, carnetTrip, storedValue."*
- ***UsageDiscountRight*** Add ***ProductType*** with values *"mileagePoints, usageRebate, other"*
- ***SaleDiscountRight****:* Add ***ProductType*** with *"values; travelCard, payAsYouGoDiscount, other."*
- **FareProduct** *Add* **ChargingMomentType** attribute to with values*: "beforeTravel, onStartOfTravel, beforeEndOfTravel, onStartThenAdjustAtEndOfTravel, onStarThenAdjustAtEndOfFareDay, onStartThenAdjustAtEndOfChargePeriod, atEndOfTravel, atEndOfFareDay, atEndOChargePeriod, free, other."*

UK0041 ***GenericParameterAssignment***: Add ***LimitationSelectionType*** operator with values "*oneOf / someOf/ allOf".*

UK0041 ***UserProfile*** : allow more than one enum values for ***ProofOfEligibilty***.

UK0044 ***GroupOfDistanceMatrixElements*** : Add ***UseToExclude***.

UK0045 ***EntitlementGiven***, ***EntitlmentRequired***: Add Constraints so that that supplements and dependent products have same parameters

UK0055 Add new ***TypeOfFareStructureElement***.

UK0089 Add new ***TypeOfFareStructureFactor***.

UK0057 ***TypeOfTravelDocument***: Allow list of ***MachineReadable*** values, Add open ended ***TypeOfMachineRedability***.

UK0069 ***FareTable***: Add ***TypeOfOfFareTable*** element.

UK0074 ***TariffBasis***; Add values *"zoneToZone, pointToPoint, discount".*

UK0080 ***GenericParameterAssignment*** Add ***TypeOfConcessionRef, TypeOfUsageParameterRef, VehicleTypeRef, TypeOfLineRef.***

NORWAY097 ***UserProfile*** / ***UserType***; Add values "*student, schoolPupil, youngPerson, military, disabled, disabledCompanion, employee, jobSeeker".*

NORWAY098 ***UsageValidityPeriod***/ ***UsageTrigger***: Add value "*activation"*

NORWAY099 ***SupplementProduct***: Change cardinality of ***supplemenTo / SupplementToFareProductRef*** from 0:1 to 0:.

NORWAY100 Support VAT (and other tax) categories.

- Add ***TypeOfPricingRule*** element.
- ***RuleStepResult***: add ***AdjustmentAmount***, ***AdjustmentUnits***, ***RoundingRef***, ***Narrative***.
- Also revise ***FarePrice*** elements to use ***AmountWithResultsGroup*** and FarePriceAmount groups to be clearer
- NB this revises current sense of ***PriceRuleStepResult \ Amount***
- ***SalesTransaction***: Add ***ruleStepResults \ RuleStep*** to

NORWAY101 ***Reserving***: Add ***ReservationType*** with values "*autoAssigned, seatMap and openSeating*".

NORWAY102 ***ExchangingType***: Add values "*upgradeToSpecifiedFare, downgradeToSpecifedFare, equivalentProduct.* changeGroupSize".

NORWAY105 ***TimeInterval***: Add ***MinimumDuration***.

Bibliography

[1] CEN/TS 15531-1, *Public transport - Service interface for real-time information relating to public transport operations - Part1: Context and framework*

[2] CEN/TS 15531-2, *Public transport - Service interface for real-time information relating to public transport operations - Part2: Communications infrastructure*

[3] CEN/TS 15531-3, *Public transport - Service interface for real-time information relating to public transport operations - Part3: 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] CEN/TS 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) – Part1: 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. B1 NRT Fares. B2 IRT Fares, B3 Special Fares.

[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.