CEN/TC 278

Date: 2013-03

TC 278 WI 00278xxx

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 information tarifaie pour le transport public

ICS:

Descriptors:

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

C:\Data\NeTEx\Part 3 - V4 kb-nk\Part 3 - Word document\prCEN TS 278xxx FV (E)-part 3 -v5.doc STD Version 2.4a

Contents

Page

Foreword

This document (TC 278 WI 00278xxx) 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.[1]

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

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

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

— To avoid confusion with usual wording, Transmodel terms are in capital letters (JOURNEY PATTERN for example).

— To avoid confusion with usual wording, attributes names are in bold/italic style and use camelcase style with no spaces (***JourneyPattern*** for example).

— To avoid confusion with usual wording, attributes types are in italic style and use camelcase style with no spaces (*TypeOfEntity* for example).

[1] Currently under development

Introduction

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

This European Technical Specification specifies a Network and Timetable Exchange (NeTEx) standard 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.

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

This standard will improve a number of features of public transport information and service management: Interoperability – the standard will facilitate interoperability between information processing systems of the transport operators by: (i) introducing common architectures for message exchange; (ii) introducing a modular set of compatible information services for real-time vehicle information; (iii) using common data models and schemas for the messages exchanged for each service; and (iv) introducing a consistent approach to data management.

Technical advantages include the following: 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[2]) 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 Many NeTEx concepts are taken directly from Transmodel and IFOPT; the definitions and explanation of these concepts are extracted directly from the respective standard and reused in NeTEx, sometimes with adaptions 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. 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 complex conditions that may attach to particular fares, for example if restricted to specific groups of users, or subject to temporal restrictions.

(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' and allows fare data to be managed and integrated journey planning and network data in public facing information systems. It is complementary to and distinct from the 'downstream' to the ticketing and retail systems that sell fares and 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 and air journeys, but there has not been any specific consideration of any additional requirements that apply specifically to air transport.

[2] Under development

1.4 Compatibility with existing standards and recommendations

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

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

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

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

2 Normative references

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

EN 15531-1, *Public transport - Service interface for real-time information relating to public transport operations - Part1: Context and framework*[3]

EN 15531-2, *Public transport - Service interface for real-time information relating to public transport operations - Part2: Communications infrastructure*[4]

EN 15531-3, *Public transport - Service interface for real-time information relating to public transport operations - Part3: Functional service interfaces*[5]

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] Under development (WI 00278340)

[4] Under development (WI 00278341)

[5] Under development (WI 00278342)

3 Terms and definitions

For the purposes of this document, the terms and definitions given in CEN/TS 00278307 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
(Access Rights 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 PASSENGER 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 fare calculation. May or may not be a SCHEDULED STOP POINT.

3.6
CHARGING MOMENT
(Fare Product MODEL)
A classification of FARE PRODUCTs according to the payment time and 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.

3.7
COMMERCIAL PROFILE
(Usage Parameter 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.

3.8
COMPANION PROFILE
(Usage Parameter MODEL)
The number and characteristics of persons entitled to travel in a group or as companions to another USER PROFILE.

3.9
DISCOUNTABLE PRICE
(Fare Price MODEL)

A price for which a discount can be offered.

3.10
COMPOSITE ACCESS RIGHT PARAMETER ASSIGNMENT
(Access Rights Parameters MODEL)
An ACCESS RIGHT PARAMETER ASSIGNMENT made up of several other ACCESS RIGHT PARAMETER ASSIGNMENT combined with a logical operator to specify a complex condition.

3.11
CONTROLLABLE ELEMENT
(Access Rights Parameters MODEL)
The smallest controllable element of public transport consumption, all along which any VALIDITY PARAMETER ASSIGNMENT remains valid.

3.12
CONTROLLABLE ELEMENT IN SEQUENCE
(Access Rights Parameters 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.

3.13
CONTROLLABLE ELEMENT PRICE
(Access Rights Parameters MODEL)
A set of all possible price features of a CONTROLLABLE ELEMENT: default total price, discount in value or percentage, etc.

3.14
CUSTOMER
(Fare Contract MODEL)
An identified person or organisation involved in a fare process. There may be a PASSENGER CONTRACT between the CUSTOMER and the OPERATOR or the AUTHORITY ruling the consumption of services.

3.15
DISTANCE MATRIX ELEMENT
(Fare Structure 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.

3.16
DISTANCE MATRIX ELEMENT PRICE
(Fare Structure MODEL)
A set of all possible price features of a DISTANCE MATRIX ELEMENT: default total price, etc.

3.17
DISTRIBUTION ASSIGNMENT
(Fare Sales MODEL)
An assignment of the COUNTRY and/or DISTRIBUTION CHANNEL through which a product may or may not be distributed.

3.18
DISTRIBUTION CHANNEL
(Fare Sales MODEL)
A type of outlet for selling a product.

3.19
EXCHANGING
(Usage Parameter MODEL)
Whether and how the product may be exchanged for another product o

3.20
FARE DAY TYPE
(Access Rights 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.

3.21
FARE POINT IN JOURNEY 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.

3.22
FARE PRICE
(Fare Price MODEL)
Price features DEFINED BY DEFAULT characterizing different PRICE GROUPs.

3.23
FARE PRODUCT
(Fare Product MODEL)
An immaterial marketable element (access rights, discount rights etc), specific to a CHARGING METHOD.

3.24
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.

3.25
FARE SCHEDULED STOP POINT
(Fare Zone MODEL)
A specialisation of SCHEDULED STOP POINT describing a stop with fare accounting and routing characteristics.

3.26
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.

3.27
FARE STRUCTURE ELEMENT
(Fare Structure MODEL)
A sequence or set of CONTROLLABLE ELEMENTs to which rules for limitation of access rights and calculation of prices (fare structure) are applied.

3.28
FARE STRUCTURE ELEMENT IN SEQUENCE
(Fare Structure 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.

3.29
FARE STRUCTURE ELEMENT PRICE
(Fare Structure MODEL)
A set of all possible price features of a FARE STRUCTURE ELEMENT: default total price, discount in value or percentage, etc.

3.30
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, ACCESS RIGHT PARAMETER ASSIGNMENT, CLASS OF USE, OPERATOR, VEHICLE MODE, FARE PRODUCT.

3.31
FREQUENCY OF USE
(Usage Parameter MODEL)
The limits of usage frequency for a FARE PRODUCT (or one of its components) or a SALES PACKAGE during a specific VALIDITY PERIOD. There may be different tariffs depending on how often the right is consumed during the period.

3.32
FULFILLMENT METHOD
(Fare Sales MODEL)
The means by which the ticket is delivered to the Customer. e.g. online, collection, etc.

3.33
GENERIC PARAMETER ASSIGNMENT
(Access Rights 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).

3.34
GEOGRAPHICAL INTERVAL
(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.

3.35
GEOGRAPHICAL INTERVAL PRICE
(Fare Structure MODEL)
A set of all possible price features of a GEOGRAPHICAL INTERVAL: default total price, etc.

3.36
GEOGRAPHICAL STRUCTURE FACTOR
(Fare Structure MODEL)
The value of a GEOGRAPHICAL INTERVAL or a DISTANCE MATRIX ELEMENT expressed by a GEOGRAPHICAL UNIT.

3.37
GEOGRAPHICAL UNIT
(Fare Structure MODEL)
A unit for calculating geographical graduated fares.

3.38
GEOGRAPHICAL UNIT PRICE
(Fare Structure MODEL)
A set of all possible price features of a GEOGRAPHICAL UNIT: default total price, etc.

3.39
GROUP OF DISTANCE MATRIX ELEMENTS
(Fare Structure MODEL)
A grouping of DISTANCE MATRIX ELEMENTs. May be used to provide reusable Origin / Destination pairs (and associate them a PRICE).

3.40
GROUP OF SALES PACKAGES
(Fare Sales MODEL)
A grouping of SALES PACKAGEs, possibly sharing ACCESS RIGHTs and NOTICE

3.41
GROUP TICKET
(Usage Parameter MODEL)
The number and characteristics of persons entitled to travel in addition to the holder of an access right.

3.42
INTERCHANGING
(Usage Parameter MODEL)
Limitations on making changes within a trip.

3.43
LUGGAGE ALLOWANCE
(Usage Parameter MODEL)
The number and characteristics (weight, volume) of luggage that a holder of an access right is entitled to carry.

3.44
MINIMUM STAY
(Usage Parameter MODEL)
Details of any minimum stay at the destination required to use the product.

3.45
PASSENGER CONTRACT
(Fare Contract MODEL)
A contract with a particular (but possibly anonymous) customer, ruling the consumption of transport services (and joint services). A PASSENGER CONTRACT may be designed for a fixed SALES PACKAGE (e.g. ticket) or to allow successive purchases of SALES PACKAGEs.

3.46
PASSENGER CONTRACT EVENT
(Fare Contract MODEL)
A log entry describing an event referring to the life of a PASSENGER CONTRACT: initial contracting, sales, validation entries,, etc. A subset of a PASSENGER CONTRACT EVENT is often materialised on a TRAVEL DOCUMENT.

3.47
PRE-ASSIGNED FARE PRODUCT
(Fare Product MODEL)
A FARE PRODUCT consisting of one or several VALIDABLE ELEMENTs, specific to a CHARGING METHOD.

3.48
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.

3.49
PRICEABLE OBJECT
(Fare Price MODEL)
An element which may be given an associated price.

3.50
PRICING SERVICE
(Fare Price MODEL)
A web service used to provide prices dynamically at time of booking or purchase.

3.51
PRICE UNIT
(Fare Price MODEL)
A unit to express prices: amount of currency, abstract fare unit, ticket unit or token, etc.

3.52
PURCHASE WINDOW
(Usage Parameter MODEL)
Period in which the product must be purchased.

3.53
QUALITY STRUCTURE FACTOR
(Fare Structure MODEL)
A factor influencing access rights definition or calculation of prices, based on the quality: traffic congestion threshold, early/late reservation, etc.

3.54
REFERENCE PRICE
(Fare Price MODEL)
A fundamental price from which other prices can be derived.

3.55
RETAIL CONSORTIUM
(Retail MODEL)
A group of ORGANISATIONs formally incorporated as a retailer of fare products. The consortium may share security and validation processes.

3.56
RETAIL DEVICE
(Retail MODEL)
A retail device used to sell fare products. Its identity be used to record fulfilment and support security processes.

3.57
REFUNDING
(Usage Parameter MODEL)
Whether and how the product may be refunded.

3.58
REPLACING
(Usage Parameter MODEL)
Whether and how the product may be replaced if lost or stolen.

3.59
REQUIRED ENTITLEMENT
(Usage Parameter MODEL)
Entitlement necessary to be able to buy or use a FARE PRODUCT.

3.60
RESELLING
(Usage Parameter MODEL)
Common resale conditions (i.e. for exchange or refund) attaching to the product.

3.61
RESERVING
(Usage Parameter MODEL)
Limitations on making changes within a trip.

3.62
ROUND TRIP
(Usage Parameter MODEL)
Properties relating to single or return trip use of a fare.

3.63
ROUNDING
(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.

3.64
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.

3.65
ROUTING
(Usage Parameter MODEL)
Limitations on routing of a fare.

3.66
SALE DISCOUNT RIGHT
(Fare Product MODEL)
A FARE PRODUCT allowing a customer to benefit from discounts when purchasing SALES PACKAGEs.

3.67
SALE TRANSACTION
(Fare Sales MODEL)
A SALE OF a FIXED PACKAGE or a SALE OF a RELOADABLE PACKAGE.

3.68
SALES NOTICE ASSIGNMENT
(Fare Sales MODEL)
The assignment of a NOTICE to a SALES PACKAGE or a GROUP OF SALES PACKAGEs.

3.69
SALES PACKAGE
(Fare Sales 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.

3.70
SALES PACKAGE ELEMENT
(Fare Sales MODEL)
The assignment of a FARE PRODUCT to a TYPE OF TRAVEL DOCUMENT in order to define a SALES 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.

3.71
SALES PACKAGE PRICE
(Fare Sales MODEL)

A set of all possible price features of a SALES PACKAGE: default total price, etc.

3.72
SALES PACKAGE SUBSTITUTION
(Fare Sales MODEL)
Information on the preferred substitution of packages with other package if quota restricted product is no longer available.

3.73
SEAT CLASS
(Access Rights Parameters MODEL)
A parameter indicating the quality of transport (e.g. 1st class or 2nd class).

3.74
SERIES CONSTRAINT
(Fare Zone MODEL)
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 (value in km, number of fare units etc), constrained to specific routes. SERIES CONSTRAINT are mainly used for rail fares.

3.75
SPECIFIC PARAMETER ASSIGNMENT
(Access Rights Parameters 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).

3.76
STANDARD FARE TABLE
(Fare Table MODEL)
A predefined grouping of four prices (1st /Second Class for each Single / Return) that may be associated with a DISTANCE MATRIX ELEMENT, FARE STRUCTURE ELEMENT or GROUP OF ACCESS PARAMETERs. It is a simplification of a FARE TABLE, to limit the number of required relations.

3.77
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.

3.78
TARIFF
(Fare Structure MODEL)
A particular tariff, described by a combination of parameters.

3.79
TIME INTERVAL
(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.

3.80
TIME INTERVAL PRICE
(Fare Structure MODEL)
A set of all possible price features of a TIME INTERVAL, e.g. default total price, etc.

3.81
TIME STRUCTURE FACTOR
(Fare Structure MODEL)
The value of a TIME INTERVAL expressed by a TIME UNIT.

3.82
TIME UNIT
(Fare Structure MODEL)
A unit for calculating time-based graduated fares.

3.83
TIME UNIT PRICE
(Fare Structure MODEL)
A set of all possible price features of a TIME UNIT: default total price, etc.

3.84
TRANSFERABILITY
(Usage Parameter MODEL)
The number and characteristics of persons entitled to use the public transport service instead of the original customer.

3.85
TRAVEL DOCUMENT
(Fare 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 PASSENGER CONTRACT liabilities or a combination of those.

3.86
TRAVEL SPECIFICATION
(Fare Contract 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).

3.87
TYPE OF ACCESS RIGHT ASSIGNMENT
(Access Right Assignment MODEL)
A classification of access right assignment by purpose.

3.88
TYPE OF CONCESSION
(Fare Price MODEL)
A classification of users by eligibility for different types of discount., e.g. child, student, old age pensioner

3.89
TYPE OF FARE PRODUCT
(Access Right Assignment MODEL)
A classification of fare product according to nature.
3.90
TYPE OF PASSENGER CONTRACT
(Fare Contract MODEL)
A classification of PASSENGER CONTRACTs to express the different types of contract.

3.91
TYPE OF PASSENGER CONTRACT EVENT
(Fare Contract MODEL)
A classification of PASSENGER CONTRACT EVENTs to express the different types of event.

3.92
TYPE OF RETAIL DEVICE
(Retail MODEL)
A classification of RETAIL DEVICE to express the different types of device used to sell products.

3.93
TYPE OF SALES PACKAGE
(Sales Package MODEL)
A classification of SALES PACKAGEs to express the different types of sales offering.

3.94
TYPE OF TARIFF
(Fare Structure MODEL)
A classification of TARIFFs to express the different classes of fares.

3.95
TYPE OF TRAVEL DOCUMENT
(Fare 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.

3.96
TYPE OF USAGE PARAMETER
(Usage Parameter MODEL)
A classification of USAGE PARAMETERs to express the nature parameter.

3.97
USAGE DISCOUNT RIGHT
(Fare Product MODEL)
A FARE PRODUCT allowing a customer to benefit from discounts when consuming VALIDABLE ELEMENTs.

3.98
USAGE PARAMETER
(Usage Parameter MODEL)
A parameter used to specify the use of a SALES PACKAGE or a FARE PRODUCT.

3.99
USAGE PARAMETER PRICE
(Usage Parameter MODEL)
A set of all possible price features of a USAGE PARAMETER: discount in value or percentage, etc.

3.100
USAGE VALIDITY PERIOD
(Usage Parameter MODEL)
A time limitation for validity of a FARE PRODUCT or a SALES PACKAGE. It may be composed of a standard duration (e.g. 3 days, 1 month) and/or fixed start/end dates and times.

3.101
USER PROFILE
(Usage Parameter 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.

3.102
VALIDABLE ELEMENT
(Fare Product MODEL)

A sequence or set of FARE STRUCTURE ELEMENTs, grouped together to be validated in one go.

3.103
VALIDABLE ELEMENT PRICE
(Fare Product MODEL)
A set of all possible price features of a VALIDABLE ELEMENT : default total price, discount in value or percentage, etc.

3.104
VALIDITY PARAMETER ASSIGNMENT
(Access Rights 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 PACKAGE.

4 Symbols and abbreviations

For the purposes of this document, the symbols and abbreviations given in CEN/TS 00278307 apply.

5 Use Cases for Fare Exchange

5.1 Actors and Use Case Types

5.1.1 Actors

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.

[TO DO USE CASES?]

Table 1 — NeTEx Part3 actors

Systems	Producer	Consumer	Organisations
	TO DO		

5.1.2 Delivery Use Cases

Refer to Part1 & 2 use case document.

- FARE-01X: (Re)define a fare policy (fare structure, products)
- FARE-02X: Organise usage (fare policy)

5.1.3 Fare Use Cases

Refer to Part1 & 2 use case document.

5.1.4 Security Use Cases

Refer to Part1 & 2 use case document.

5.1.5 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	Source	Reason for exclusion
Provision of Fare information to validation devices		
Provision of fare information to sale devices		
Management of fare product and applications		
Certification, registration and identification		
Purchasing and fulfilment		
Provision of management information about tariff and product usage		

[TO DO USE CASES?]

5.2 Use Cases

5.2.1 Requirements Table

NOTE Version filter means a filter by version and validity condition (e.g. date, time from, until)

[TO DO USE CASES?]

Use Case	Focus on	Filtering	Comments

NOTE Data Producers should support a “capability request” telling which filters it supports.

It is recommended that data producers support filter by version, line and mode at least

5.2.2 Preamble to Use Cases

This section presents use cases for a variety of public transport activities that could benefit from using the NeTEx data exchange interface. Discussion focuses mainly on the data content that is to be exchanged (content use cases).

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.)

5.2.2.1 Provide fare-related information

- FARE-011: (Re)define a fare policy (fare structure)
- FARE-012: (Re)define a fare policy (fare products)
- FARE-013 (Re)define a fare policy (dynamic prices)
- FARE-021: Organise usage for publication
- FARE-022: Organise usage for journey planning
- FARE-023: Organise usage for fare product sales
- FARE-024: Organise usage for operational support

Fare structure: set of parameters that determine the basic tariffs.

Fare product:

1) definition of access rights
2) definition of product templates that are used to market access rights

5.2.2.2 Introduction to terminology

The terms *Tariff*, *Cost* and *Price* are distinguished as follows in the discussion of use cases.

Fare:
From the customer perspective: the amount that a customer has to pay for a journey or for acquiring a product.

Tariff:

From a planner perspective: the set of discrete elements to be used according to the fare calculation rules to calculate the fare.

Price:

Value of fare or tariff.

The following key terms are used in the discussion of use cases.

Fare product:

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

Product instance

A particular selection of specific features of a fare product as might be acquired by a user buying a ticket or other product, e.g. single way ticket from Amsterdam to Paris (also called in NeTEx a travel specification).

Sales packages:

A marketed product that a passenger can buy on a travel document, i.e. paper ticket or electronic card.

Prepaid ticketing

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

Post-paid ticketing

The user is charged sometime after using the transport service (detailed description of process see below).

5.2.2.3 Description of fare calculation and charging processes

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

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.

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 some time later to incorporate rebates like a fare cap.

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.

5.2.2.3.1 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.

5.2.2.3.2 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).

5.2.3 Use Case descriptions

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 the practical use 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.

5.2.3.1 Dynamic prices

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.

— **Transparent fare structure with 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).

— **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 after- sales 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 organization 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.

5.2.3.2 (Re)define a fare policy

5.2.3.2.1 Fare Structure and Fare Product

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 the information regarding the Fare Structure. A distinction is made between distance, geo unit and time based fares. This could 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. 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.

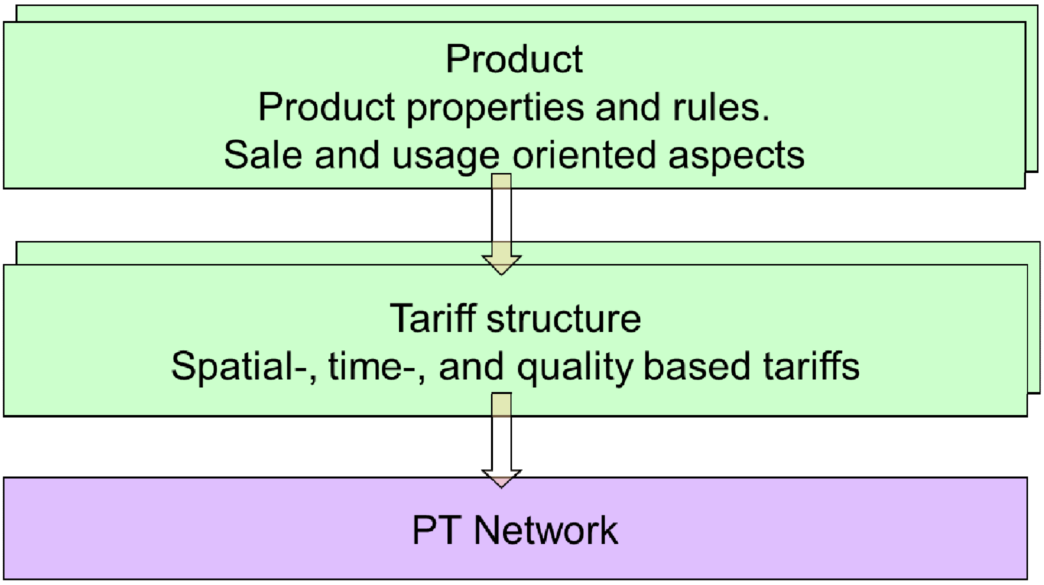


Figure 1 — PT fare overview

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 and 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.

5.2.3.2.2 Separation of Fare prices

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.

5.2.3.2.3 Features of Fare products

The features that determine the price of fare product to the user can be divided into two main groups. These groups have many interdependencies.

— Feature relating to the scope of access rights as to extent and time of travel.

  The choice of access features of a fare product can range from the one-time access to very specific features of the fare structure, as for a single or return ticket between a nominated start and destination, to repeated access to a wide range of features, as in a period pass to a several zones of a Network.

— Features relating to the discounting and commercial conditions.

  The price can also be affected by other factors such as the number and type of user (e.g. a discount for seniors or a disabled companion), the conditions governing exchange and refund, the flexibility of travel . Some features can allow specific reductions on the standard tariff as defined by the Fare Structure.

There are subtle differences between moment of calculation, moment and method of charge that will be described in the respective use-cases.

As mentioned, the above two categories do have a high level of interaction. A product that is pre specified and where the right to travel is commuted charges a sale price but can offer free travel within the area of validity (Access Right). Or a default product (e.g. easy trip) is free of charge at sale but charged the entire tariff while traveling. The following table shows some simplified examples.

Table 3 — Fare examples

Product	Access right	Sale	Travel (use)
Easy trip	Nation-wide	Free of charge	Full charge conform tariff in fare structure.
“Star” subscription	Central tariff zone and zone radius	Sale price depending on zone radius	Free of charge within access right.
Train always reduction	All train operators	Fixed sale price	Reduction in off-peak hours and weekend, full charge conform tariff structure otherwise.

[TO DO WHAT DOES THE ABOVE MEAN – DON’T UNDERSTAND!]

A fundamental distinction can be made between a product that actually gives **access rights** to travel and a product that gives **the right to buy or consume other access rights** at a discounted rate – but that doesn’t of itself permit travel (for example a rail card that allows tickets to be purchased at a discount).

An access right defines the validity of a product in both geographical and temporal dimensions.

Depending on law or agreed working procedures, several organisations can be made responsible for the definition of a (part of a) fare policy. E.g.

- The fare basics in the fare structure are defined by the authority,
- Common national or regional fare products that are defined by the authority,
- Specific action products that are defined by PT Operators,
- Etc.

5.2.3.2.4 Business context

A fare planning process will consider the effects of different possible prices and products on yields, a process that requires modelling 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 relevance of the 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 underling 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”).

5.2.3.2.5 (Re)define a fare policy, fare structure

The process of creating and maintaining the 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 fare structure typically contains base tariffs optionally enhanced with tariff 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 matrices 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 electronic form.

The use cases are described to reflect a workflow (or process) that can be used for the definition of a fare structure. Certain special use cases , for example the definition of fare zones for the use of statistical analysis , are relevant on a stand-alone basis.

A typical workflow (or process) for definition of a fare structure is represented by the sequence of use-cases as described below.

[Is the following roughly what we are covering?

1. choose Validable element: e.g. bus ride, train ride, metro trip, metro ride etc,
2. Choose spatial structure for fares : (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
3. Choose temporal structure e.g. 1 hour 2 hour day pass, etc.
4. Choose limitation parameters (i) roundtrip, parameters (ii) user profiles and group tickets

Add basic prices]

5. Decide Charging moments and Choose products as combinations of the above
6. Test these against predicted yields and iterate 1-6
7. Choose travel documents and distribution
8. Add deriving prices
9. Choose sales packages

Derive final prices]

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. 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. 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.

[TO DO – Use cases should be heading levels so they can be comprehended at an outline level – have temporarily added as separate headings – can we change so that heading in box is a level?]

5.2.3.2.6 Use Case: FARE-001-001-Determination of basic fare structure network scope (#1)

Use Case: FARE-001-001-Determination of basic fare structure network scope (#1)	
Name	Determination of basic fare structure network scope.
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Tariffs are consistent within a certain network scope that is defined by PT Operator, domain, line, group of lines, modality, tariff zones, etc. The scope is defined as a set of scope defining validity conditions.

	Define the scope for the tariff structure in terms of: • **Geography**: network domain, group of lines, line. • **Modality**: the modality or modalities the tariff is valid for within the geographical scope. Define the scope in terms of existing PT network characteristics. The scope can be defined at alternative abstraction levels: • 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.
NeTEx contribution	NeTEx provides a means to exchange the actual state and different versions of it.
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, TYPE OF SERVICE, TYPOE OF PRODUCT CATEGORY. Part3: FARE ZONE, BORDER POINT, DISTANCE MATRIX ELEMENT.

Some examples of validity scope are given below:

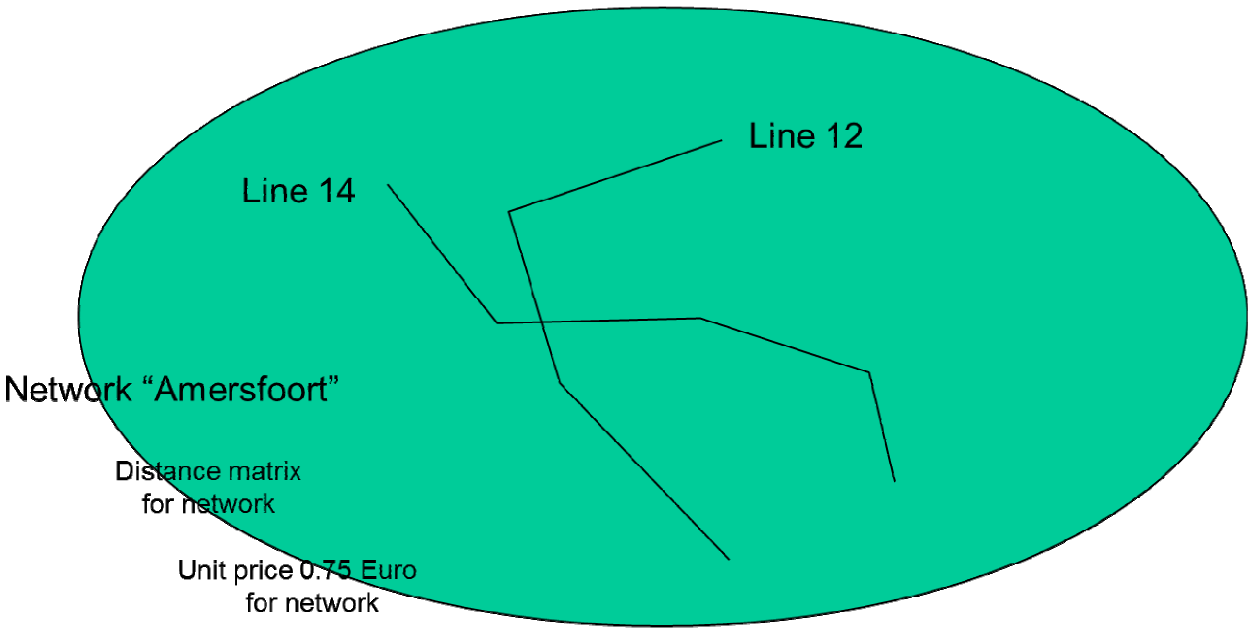


Figure 2 — Example of a fare structure that contains an OD matrix and a unit price

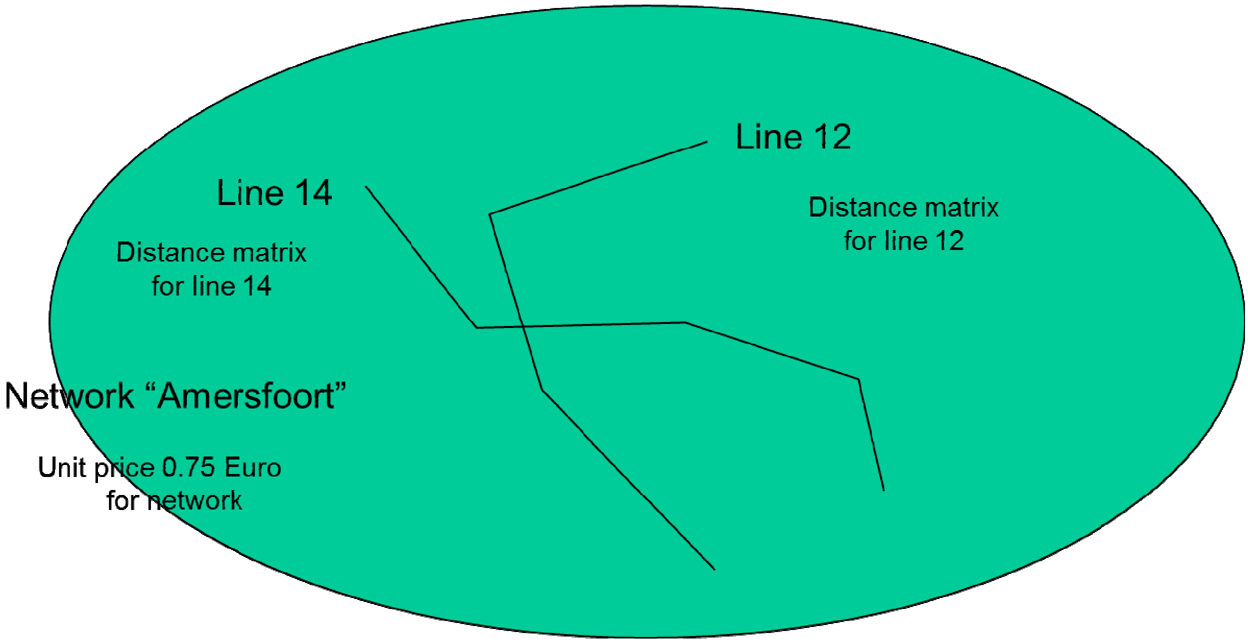


Figure 3 — Example with OD matrices for each line within and a unit price for the entire network

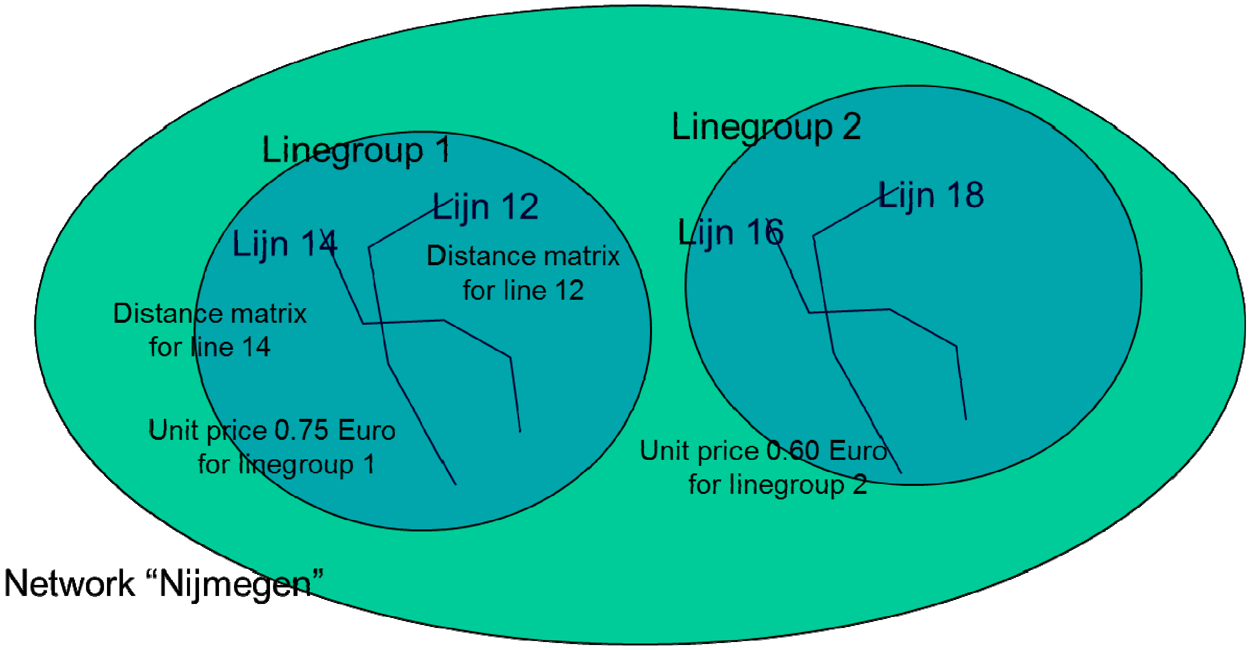


Figure 4 — Example with line groups, OD matrices for each line and a unit price for each line group.

[

5.2.3.2.6.1 Use Case: FARE-001-002- Determination of the basic fare structure factors (#2)

Use Case: FARE-001-002- Determination of the basic fare structure factors (#3)	
Name	Determination of the basic fare structure factors
Source	Transmodel, FareXChange, BISON, NeTEx
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 an 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 actual factors .
Main actors	Fare planning systems.
Main objects	GEOGRAPHICAL FARE FACTOR, GEOGRAPHICAL INTERVAL, GEOGRAPHICAL UNIT, TIME STRUCTURE FACTOR, TIME ITERVAL, TIME UNIT, QUALITY STRUCTURE FACTOR, TIME DEMAND FACTOR. Usage Parameters: ROUND TRIP, ROUTING, FREQUENCY OF USE, INTERCHANGING, USAGE VALIDITY PERIOD.

5.2.3.2.6.2 Use Case: FARE-001-XXX- Demarcating Fare stages or Fare zones / honeycombs

Use Case: FARE-001-XXX- Demarcating Fare stages or Fare zones / honeycombs	
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 a 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.

	Fare zones and combs: A comb is equivalent to a zone for the level of definition that is required here [6]. 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. ***Fare stages:*** Fare stages are defined using a set of rules. Stages are geographically based factors that are relative to the start of the passenger trip. Define the Fare stages in terms of the points in the network at which the fare boundaries take place.: [TO DO from bart Please enhance, I'm not very familiar with this concept.]
NeTEx contribution	NeTEx provides a means to exchange the network (stops, stations, lines) work and fare definition element s entities and different versions of it. Zones, sections and stages will be defined in the NeTEx network layer with a fare related purpose definition attached to them.
Main actors	Fare planning systems.
Main objects	Part1 : SCHEDULED STOP POINT, TARIFF ZONE, LINE, LINE SECTION Part3: FARE ZONE, FARE SECTION, FARE STRUCTURE ELEMENT, BORDER POINT, FARE SCHEDULED STOP POINT.

5.2.3.2.6.3 Use Case: FARE-001-xxxx- Additional use case for projection of patterns / links on zones. (#4

[TO DO] Additional use case for projection of patterns / links on zones.

5.2.3.2.6.4 Use Case: FARE-001-004- Determination of price currency and unit. (#5

Use Case: FARE-001-004- Determination of price currency and unit. (#6)	
Name	Determination of price units and amounts
Source	Transmodel, FareXChange, BISON, NeTEx

[6] 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 goups of points or polygons and the fact that the tariff is flat within a zone or comb is also a common feature. Is this always the case?

Description	Price currency and units define the way tariffs are calculated and represented. • Define the **currency** or currencies the prices are represented in. E.g. EURO, GBP. • Define the **unit** the prices are represented in. E.g. 0.01, 1 The currency and unit can be defined at alternative abstraction levels: 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.

5.2.3.2.6.5 Use Case: FARE-001-005- Determination of the basic tariffs (#7)

[TO DO Shouldn't this discuss modelling fare structure against expected traffic to arrive at proposed fares?]

Use Case: FARE-001-005- Determination of the basic tariffs (#8)	
Name	Determination of the basic fare structure tariffs
Source	Transmodel, FareXChange, BISON, NeTEx
Description	The basic tariffs represent the base tariff without concessionary reductions and other rules applied to it. The tariffs are defined for categories or intervals of space (stops, zones or stages), travel time or combination. The basic tariff can be represented in several way's. • 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. or • Define a tier table that maps a consumed amount of factor units to the due tariff. E.g. examples below. and / or 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.
NeTEx contribution	NeTEx provides a means to exchange the actual tariff and different versions of it.
Main actors	Fare planning systems.

Main objects	GEOGRAPHICAL FARE FACTOR, GEOGRAPHICAL INTERVAL PRICE, TIME FARE FACTOR, TIME INTERVAL PRICE, DISTANCE MATRIX ELEMENT, DISTANCE MATRIX PRICE.

Examples of tier 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.	

5.2.3.2.6.6 Use Case: FARE-001-xxxx- Setting rounding and calculation factors(#xx)

[TO DO] Additional use case for setting rounding and calculation factors.

Use Case: FARE-001-xxxx- Setting rounding and calculation factors (#9)	
Name	Setting rounding and calculation factors
Source	FareXChange, BISON, NeTEx
Description	Where fare calculations take place 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.

5.2.3.2.6.7 Use Case: FARE-001-006- Determination of tariff differentiation based on Network properties (#10)

Use Case: FARE-001-006- Determination of tariff differentiation based on Network properties (#11)	
Name	Determination of differentiation of basic tariffs
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Derivative tariffs can be based on network properties, e.g. mode of transport, line, PTO, travel class. Derivative tariffs usually are a simple multiplier of the standard or base fare • 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

5.2.3.2.6.8 Use Case: FARE-002-xxxx Market segmentation and user eligibility. criteria (#xx

Use Case: FARE-002-xxx Market segmentation and user eligibility criteria (# (#XX)	
Name	Market segmentation and determination of 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 .

Main actors	Marketing planners, Fare planning systems
Main objects	USER PROFILE, GROUP TICKET, COMMERCIAL PROFILE, COMPANION OR GROUP MEMBER, TYPE OF CONCESSION.

5.2.3.2.6.9 Use Case: FARE-001-007- Addition of user profile dependant tariffs (#12)

Use Case: FARE-001-007- Addition of user profile dependant tariffs (#13)	
Name	Addition of user profile dependant tariffs
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Derivative tariffs can be based on user profile properties, e.g. reduction for the elderly or reduction for travellers with certain disabilities. The derivative tariffs with the conditions that define their application can be contained by the tariff structure. The conditions are defined by user profile categories according to ISO [TO DO] Derivative tariffs usually are a simple multiplier of the standard or base fare • Define[BaWo1] the user profile categories 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 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 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	USER PROFILE, COMPANION OR GROUP MEMBER, USAGE PARAMETER PRICE

5.2.3.2.6.10 Use Case: FARE-001-008- Addition of temporal derivative tariffs (#14)

Use Case: FARE-001-008- Addition of temporal derivative tariffs (#15)	
Name	Addition of temporal derivative tariffs
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Derivative tariffs can be based on time of day and day type, e.g. Surtax during rush hour on weekdays or 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 user 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. Note: Depending on national laws or PT contract conditions, several alternatives to define temporal derived tariffs are possible. Derivative tariffs can be defined in the tariff structure for all temporal categories for which one is relevant. It is also possible to define the temporal derived tariff as a product attribute that is used to derive the tariff using the base tariff from the tariff structure. Which tariff derivative is actually applied during travelling is determined by the actual time of travel.
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 ELEMENT, FARE DEMAND FACTOR, FARE DEMAND FACTOR PRICE

5.2.3.2.6.11 Use Case: FARE-001-009- Addition of boarding charges and deposits (#16)

Use Case: FARE-001-009- Addition of boarding charges and deposits (#17)	
Name	Addition of boarding charges and deposits
Source	FareXChange, BISON, NeTEx
Description	Depending on national law, PT contract or concessionary conditions, it is sometimes required to enhance the base or derived tariff with additional fee's. Boarding charge. This charge is due as a result of boarding the PT service. It is charged according conditions, e.g. at the start of a trip. Deposit. When travelling with a post specified (or specified while travelling) product, sometimes a deposit is charged at the beginning of a trip that is returned when finishing the trip. Define the boarding charge currency, unit and value

	Define the deposit currency, unit and value. Note: Depending on national laws or PT contract conditions, several alternatives to define additional fees are possible. 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 fee's .are charged are always part of the product definition.
NeTEx contribution	
Main actors	
Main objects	[TO DO HOW DO WE HANDLE THESE?]

5.2.3.2.6.12 Use Case: FARE-001-010- Define Flat (including zero) fares (#18)

[TO DO This or another use case should still be concerned with pricing this travel – concessionary travel is not "free" typically it is priced and paid for by government. Therefore a use case should to model the operators cost as well.

Use Case: FARE-001-010- Define Flat (including zero) fares a19)	
Name	Define Flat (including zero) fares
Source	Transmodel, FareXChange, BISON, NeTEx
Description	Concessionary fares for elderly and disabled people can be set to a standard flat value (for example 50p for any journey in Greater Manchester in 2005/06). A special case is zero fares as now apply for off-peak travel for elderly and disabled people at least in their local areas in all parts of Great Britain. In some areas the concession extends into the peak and 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.
NeTEx contribution	NeTEx provides a means to exchange the actual tariff and different versions of it.
Main actors	Fare planning systems.
Main objects	USER PROFILE, FARE STRUCTURE ELEMENTs

5.2.3.2.6.13 Use Case: FARE-001-011- Define routes and transfer points (#20)

Use Case: FARE-001-011- Define routes and transfer points (#21)	
Name	Define routes and transfer points
Source	Transmodel, FareXChange, BISON, NeTEx
Description	For distance and geo unit fares, facilitation for tariff calculation in case there are

	alternative routes and / or alternative transfer points. E.g. selection of the shortest route or selection of a specific route. For the fare structure these may be specified as constraints on point to point and other fares
NeTEx contribution	NeTEx provides a means to exchange the actual routes and transfer points and different versions of it.
Main actors	Fare planning systems.
Main objects	SERIES CONSTRAINT, ROUTING usage parameter

5.2.3.2.7 (Re)define a fare policy, fare products

Right to access and use PT transport can be marketed and obtained by using fare products.

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

A fare product is immaterial, which means that the same fare product can be materialised on various travel documents. For instance, a monthly pass may be incorporated on a specific paper document or stored on an electronic card.

5.2.3.2.7.1 Use Case: FARE-002-001 Determination of the product access rights (#22

Use Case: FARE-002-001 Determination of the product access rights (#23)	
Name	Determination of the product access rights
Source	Transmodel, FareXChange, BISON, NeTEx
Description	A traveller that needs 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 several basic forms and as combinations of these basic forms: **Spatial access rights.** Access to stages, zones, stop areas, domains, routes (one-way, retour),, etc. These access rights must be based spatial or geographical entities as present in the PT network definition[BaWo2]. **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.
NeTEx contribution	NeTEx provides a means to exchange the actual product access rights and different versions of it.
Main actors	
Main objects	ACCESS RIGHT PARAMETER ASSIGNMENT.

5.2.3.2.7.2 Use Case: FARE-002-002 Determination of the product price and concessionary parameters. (#25)

Use Case: FARE-002-002 Determination of the product price and concessionary parameters. (#24)	
Name	Determination of the product price
Source	Transmodel, FareXChange, BISON, NeTEx
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. determination of sale and usage prices. 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) etc. A price can be defined as an amount (e.g. sale price of a month reduction pass is 30,- 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 access rights 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	
Main objects	TARIFF, FARE PRICEs, PRICING RULEs, PRICING PARAMETER SET

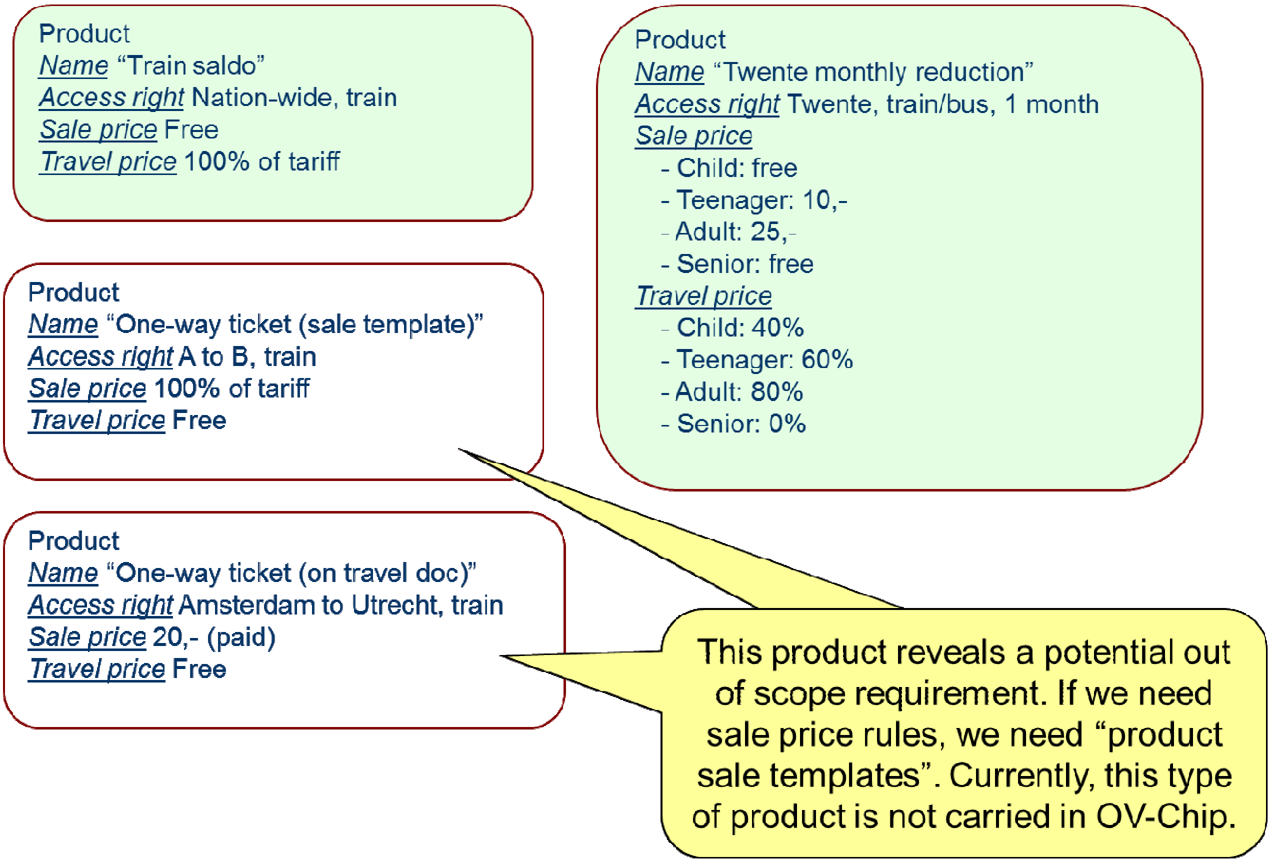


Figure 5 — Product pricing examples.

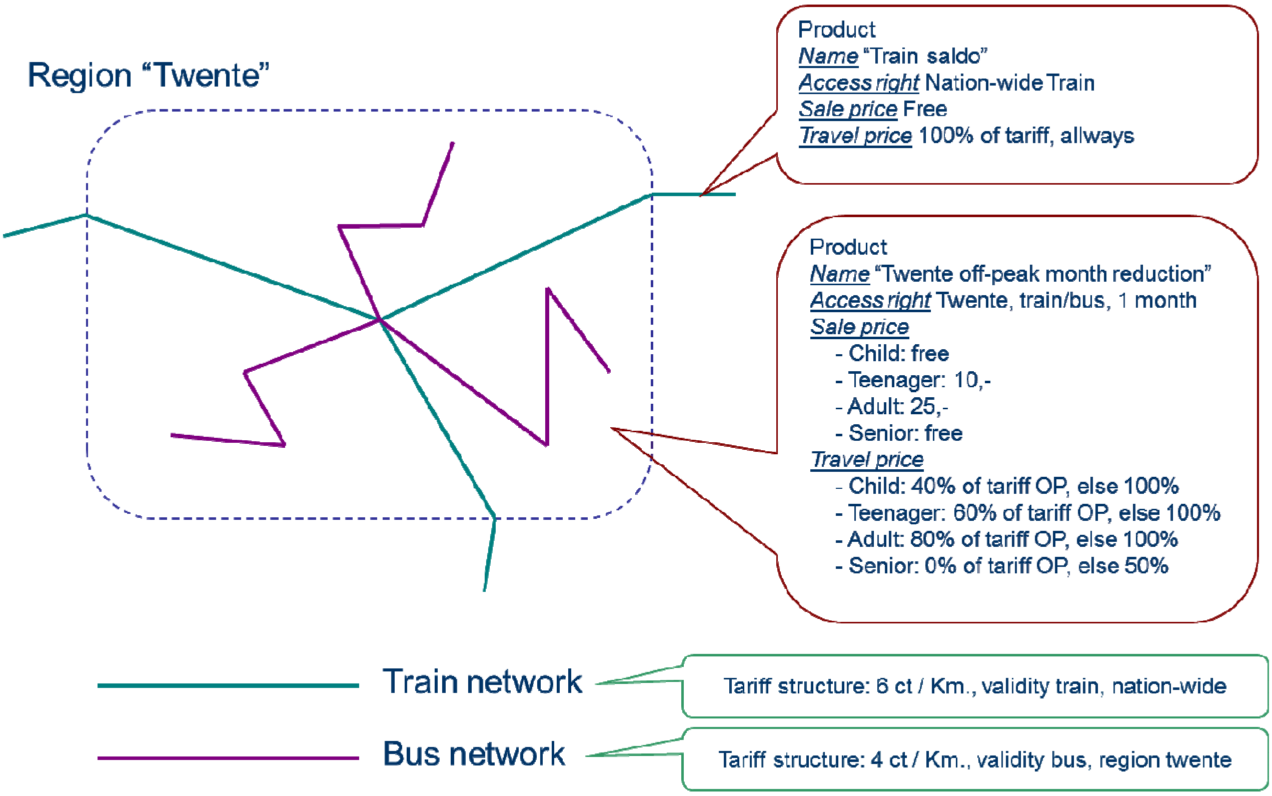
Calendar	
Daytype	Off-Peak
Sunday	00:00 – 23:59
Monday	00:00 – 07:00, 09:00 – 23:59
Tuesday	00:00 – 07:00, 09:00 – 23:59
Wednesday	00:00 – 07:00, 09:00 – 23:59
Thursday	00:00 – 07:00, 09:00 – 23:59
Friday	00:00 – 07:00, 09:00 – 23:59
Saturday	00:00 – 23:59

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 6 — Product pricing example

5.2.3.2.7.3 Use Case: FARE-002-005 Determination of product interoperability (#25s

Use Case: FARE-002-004 Determination of product boarding charges and deposits (#26)	
Name	Determination of product boarding charges and deposits
Source	FareXChange, BISON, NeTEx
Description	Depending on national law, PT contract or concessionary conditions, it is sometimes required to enhance the base or derived tariff with additional fee's. Boarding charge. This charge is due as a result of boarding the PT service. It is charged according conditions, e.g. at the start of a trip. Deposit. When travelling with a post specified (or specified while travelling) product, sometimes a deposit is charged at the beginning of a trip that is returned when finishing the trip. Define the boarding charge currency, unit and value Define the deposit currency, unit and value. Note: Depending on national laws or PT contract conditions, several alternatives to define additional fees are possible. 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 fee's .are charged are always part of the product definition.
NeTEx contribution	NeTEx provides a means to exchange the actual product information and different versions of it.
Main actors	
Main objects	[TO DO HOW DO WE MODEL A BOARDING CHARGE?]



Use Case: FARE-002-005 Determination of product interoperability (#27)	
Name	Determination of product interoperability
Source	FareXChange, BISON, NeTEx
Description	Determination of product interoperability, e.g. validity at joint PTO's or cross boundary validity and behaviour. This is a further refinement of the of the "Determination of the product access rights" use-case. Product interoperability is defined by the product access rights. Within a certain scope were definition of interoperability of products is important, relevant PT network properties should be defined uniquely. Stop identifiers should be defined uniquely within this scope since these mark the begin and end of a trip. Next, networks and lines should be defined uniquely since these are used to define the scope of tariff structures. Furthermore, all the spatial access rights of products should refer to uniquely identifiable PT network properties. Besides being a technical issue, this is foremost an organisational issue. The technical part, offering sufficient mechanisms to define unique id's of PT network properties and objects, is within the scope of influence of NeTEx. Also in the scope of NeTEx are business rules stating that tariff structure scope and product access rights should be defined sufficiently complete to be uniquely identifiable within the required scope. What determines the required scope of uniqueness and completeness? E.g., for a nation-wide journey planner this scope is the entire nation, for a validator in a bus this the line it operates. Domains…? Are uniquely defined stops already sufficient…?
NeTEx contribution	

Main actors	
Main objects	BORDER POINT, SERIES CONSTRAINT.

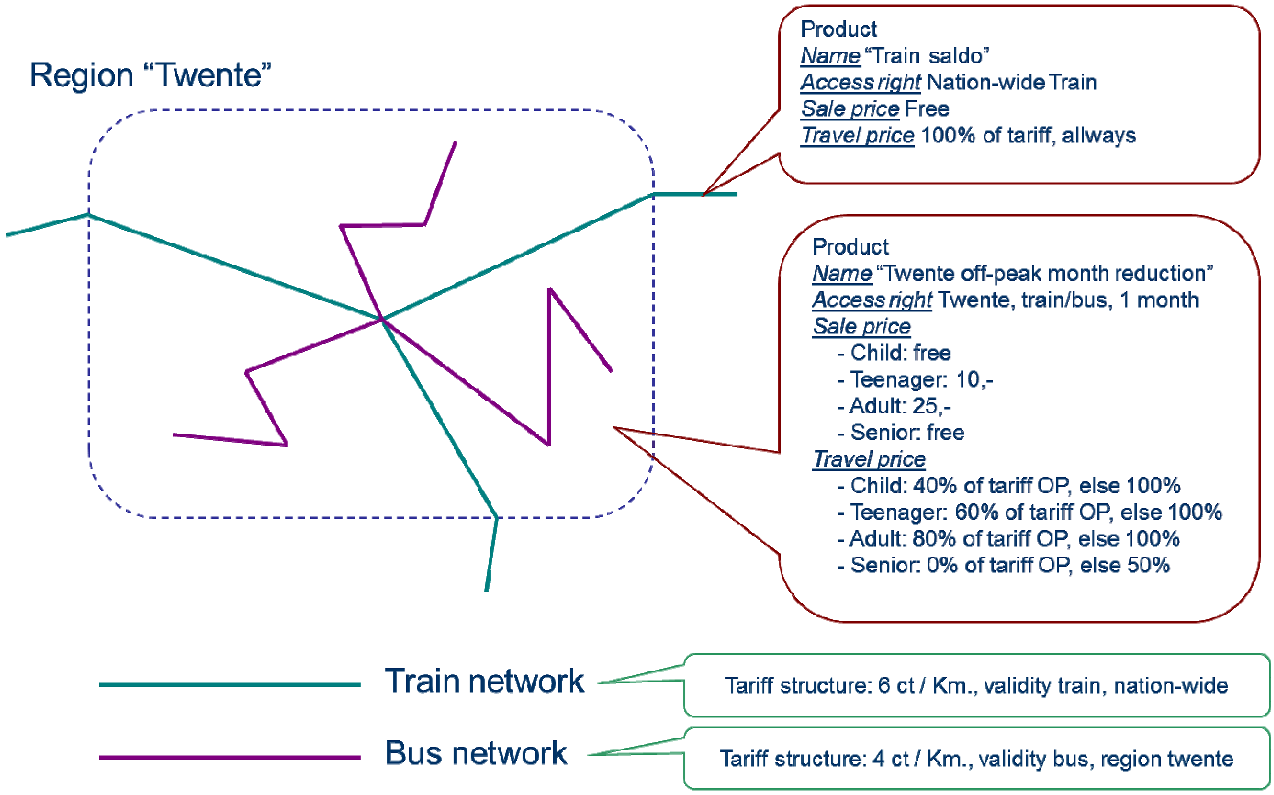


Figure 7 — Region "Twente" example

5.2.3.2.7.4 Use Case: FARE-002-006 Determination of product transfer rights (#28)

5.2.3.2.7.5 Use Case: FARE-002-007 Determination of combination fares (#29)

Use Case: FARE-002-006 Determination of product transfer rights (#30)	
Name	
Source	
Description	Determination of product transfer rights and related fares, e.g. no boarding charge within 35 minutes after last travel
NeTEx contribution	
Main actors	
Main objects	FARE STRUCTURE ELEMENTs, USAGE VALIDITY PERIOD.

5.2.3.2.7.6 Use Case: FARE-002-008 Determination of product pre- and after- sale conditions (#31)

Use Case: FARE-002-007 Determination of combination fares (#32)	
Name	
Source	
Description	Determination of product combination fares (including non-transport entities or with add-ons, such as sleeping accommodation Combination of PT Products
NeTEx contribution	
Main actors	
Main objects	SALES PACKAGE ELEMENT, SALES PACKAGE, FARE TABLE.

5.2.3.2.7.7 Use Case: FARE-002-xxx Re-price products for periodic increase (#xx)

Use Case: FARE-002-008 Determination of product pre- and after- sale conditions (#33)	
Name	
Source	
Description	Commercial conditions for purchasing and using the fare products need to be described These may include • When bookings must be made • Conditions for exchanging and refund • Minimum Stay • the amount of luggage allowed,, etc. etc
NeTEx contribution	Conditions can be specified and associated with products and sales packages.
Main actors	Marketing process
Main objects	USAGE PARAMETERs: PURCHASE WINDOW, MINIMUM STAY, RESERVING EXCHANGING, REFUNDING, LUGGAGE ALLOWANCE. SUMMARY CONDITION

5.2.3.3 Distribute fare information)

Use Case: FARE-002-xxx Re-price products for periodic increase (#xx)	
Name	
Source	
Description	Existing fare structures ar typically re-priced annually or at other intervals to account for inflation and cost changes. New prices need to be supplied for the sae fare structure.
NeTEx contribution	Fare sets can be exchanged with specified

Main actors	Fare planning process
Main objects	Part1: VALIDITY CONDITION Part3: FARE FRAME, PRICE GROUP, FARE TABLE.

5.2.3.4

The Distribute use cases describe the organisation and distribution of dare data to downstream systems such as journey planners

5.2.3.4.1 Distribute Fare Information, publication

5.2.3.4.1.1 Use Case: FARE-003-001- To provide information about sales channels (#34)

5.2.3.4.1.2 Use Case: FARE-003-002- To provide information on fare products (#35

Use Case: FARE-003-001- To provide information about sales channels (#36)	
Name	To provide information about sales channels
Source	
Description	To provide information about links (web sites) and places (retailers) where to get fare information and where to buy tickets / products
NeTEx contribution	
Main actors	
Main objects	DISTRIBUTION CHANNEL, FULLFILMENT METHOD, DISTRIBUTION ASSIGNMENT, SALES PACKAGE

5.2.3.4.1.3 Use Case: FARE-003-003- To provide fare information to on-line systems (#37))

Use Case: FARE-003-002- To provide information on fare products (#38)	
Name	
Source	
Description	To provide information on 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	
Main actors	
Main objects	FARE PRODUCT, SALES PACKAGE, USAGE PARAMETER, DISTRIBUTION CHANNEL, NOTICE ASSIGNMENT, DISTRIBUTION ASSIGNMENT

5.2.3.4.1.4 Use Case: FARE-003-005- To provide zones for maps (#39)

Use Case: FARE-003-003- To provide fare information to on-line systems (#40)	
Name	
Source	
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
NeTEx contribution	
Main actors	
Main objects	FARE FRAME with all OBJECTs

5.2.3.4.1.5 Use Case: FARE-003-006- To provide an overview of fare products / tariffs (#41

Use Case: FARE-003-005- To provide zones for maps (#42)	
Name	
Source	
Description	To show fare zones on topographical and schematic maps
NeTEx contribution	
Main actors	
Main objects	Part1: TARIFF ZONE, SHCEDULED STOP POINT, LINE, LINE SECTION, NETWORK. Part3: FARE ZONE

5.2.3.4.2 Organise fare information, journey plannin)

Use Case: FARE-003-006- To provide an overview of fare products / tariffs (#43)	
Name	
Source	
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	
Main actors	
Main objects	SALES PACKAGE, FARE PRODUCT, CONDITION SUMMARY, NOTICE.

5.2.3.4.2.1 Use Case: FARE-004-002- To provide up to date fare parameters for price calculation[BaWo3] (#44)g

Use Case: FARE-004-001- To provide up to date fare parameters for price calculation[BaWo4] (#45)	
Name	
Source	
Description	There should be somewhere a description of the sequence of operation for a journey planner : 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	
Main actors	
Main objects	FARE QUERY, TRAVEL SPECIFICATION

5.2.3.4.2.2 Use Case: FARE-004-003- To provide information for a recurring trip / travel (#46

Use Case: FARE-004-002- To provide up to date fare parameters for price calculation[BaWo5] (#47)	
Name	
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[BaWo6]
NeTEx contribution	
Main actors	
Main objects	Part2 : VEHICL JOURNEY Part3 FARE TABLE, PRICE GROUP, FARE PRICE, TRAVEL SPECIFICATION

Provide all alternative products that are available for a travel)

Use Case: FARE-004-003- To provide information for a recurring trip / travel (#48)	
Name	
Source	
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	
Main actors	
Main objects	
Use Case: FARE-004-004- optimize journey plan for price (#49)	
Name	
Source	
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	
Main actors	
Main objects	FARE QUERY, SALES PACKAGE, TRAVEL SPECIFCATION.

Provide the price of a travel (also through price calculation engine e.g. yield managed)

Information about loyalty programs

5.2.3.4.3 Organise Fare Information, Sales

Provide fare information “between systems”.
Twee aanpakken:
- fare structure, open, static
- market prijzen, dynamisch maar “gesloten”, wel pre- en after- sale conditions
- vraag is: voor dynamische prijzen ook prijs structuur uitwisselen off-line? Indien ja, wat is dan de status, indicatief?
Sales conditions set of use cases toevoegen.

- *travel documents*
- *market prices (was: yield managed fares)*
- *sales package*
- *sales conditions (omruilen, inleveren, e.d.)*

It is probably better to talk about Market Price than Yield Managed fares. Yield Managed fares usually refer to the fact that N seats are at price P, N' at price P', N" at P",, etc. these values being possibly dynamically changed, and some prices being only available on some specific channels (usually the lowers only on the operator's channel).

5.2.3.4.4 Organise Fare Information, Operational suppor

5.2.3.4.4.1 Use Case: FARE-006-002- Plan fare information for service providers (#50t

Use Case: FARE-006-001- Exchange information between rail and local transport (#51)	
Name	
Source	
Description	To exchange fare information between long distance (i.e. heavy rail) and local public transport
NeTEx contribution	
Main actors	
Main objects	FARE FRAME with FARE PRODUCTs, USAGE PARAMETERs, DISTANCE MATRIX ELEMENTs, FARE TABLEs

5.2.3.4.4.2 Use Case: FARE-006-003- Provide fare information that applies to a PT Contract[BaWo7] (#52)

Use Case: FARE-006-002- Plan fare information for service providers (#53)	
Name	
Source	
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	
Main actors	
Main objects	FARE FRAME with FARE PRODUCTs, FARE PRICEs, FARE TABLEs

5.2.3.4.4.3 Use Case: FARE-006-004- Provide information about interoperable products (#54)

Use Case: FARE-006-003- Provide fare information that applies to a PT Contract[BaWo8] (#55)	
Name	
Source	
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	
Main actors	
Main objects	FARE FRAME with FARE PRODUCTs, FARE PRICEs, FARE TABLEs

Provide fare information for checking (authorities, customer organisations))

Use Case: FARE-006-004- Provide information about interoperable products (#56)	
Name	
Source	
Description	To provide information about inter-operable products properties and parameters
NeTEx contribution	
Main actors	
Main objects	

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 Extensions to NeTEx Part1 Framework

NeTEx Part3 uses the same uniform model for versioning and managing elements as is used by Part1 and Part2, including grouping elements into VERSION FRAMEs for the purposes of exchange. Part3 also integrates seamlessly with underlying concepts from Part1 & Part2 such as stops, timetables and facilities and equipment) reusing existing elements where needed.

7.1 Introduction

7.2 Enhancement to Responsibility Model

The Part3 model introduces some minor refinements to the existing NeTEx framework model.

7.2.1 Organisation delegation relationship

An explicit delegates to relationship is added to the Part1 ORGANISATION model to indicate that a specific RESPONSIBILITY, as defined by a RESPONSIBILITY SET is delegated to another organization.

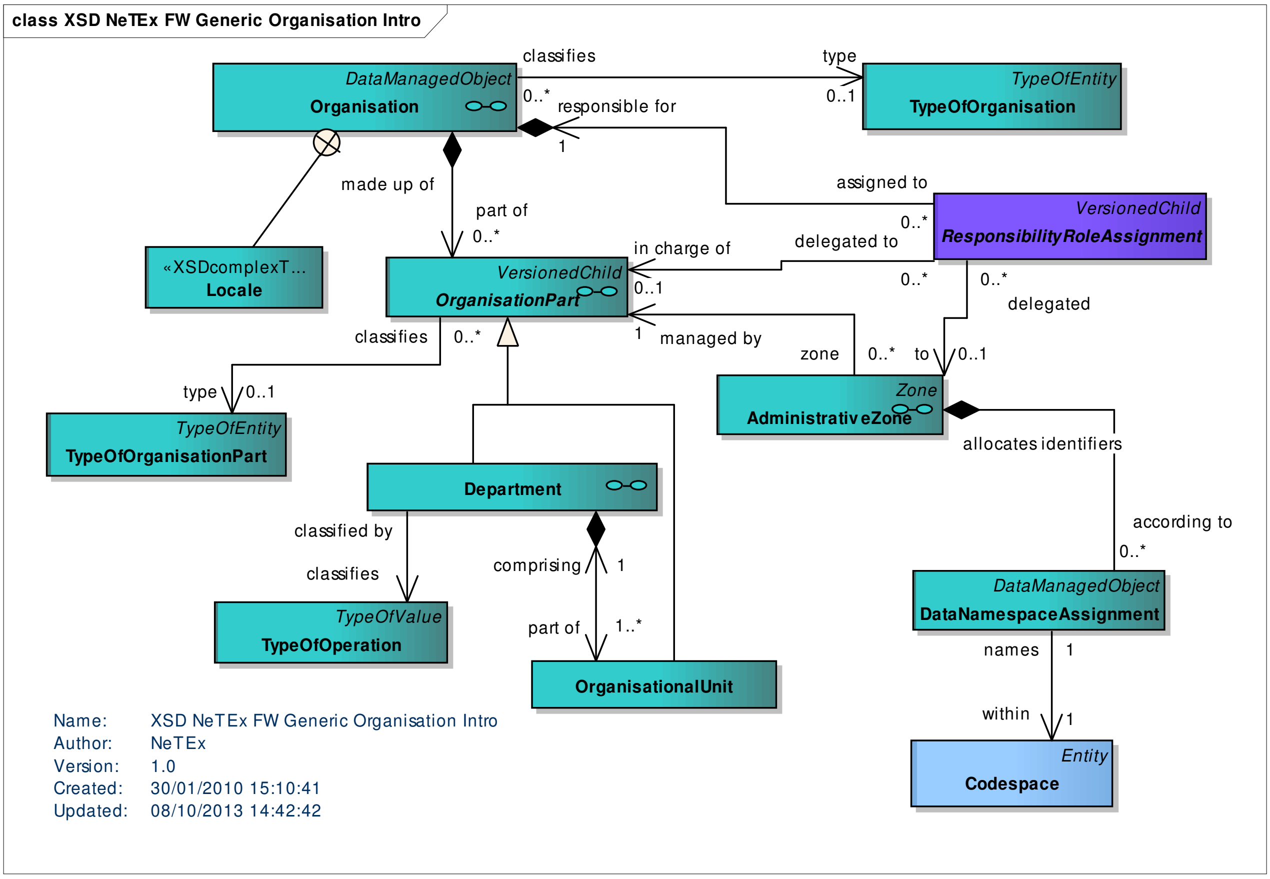


Figure 8 — Organisation (Part1 Framework) – Physical Model

7.2.2 Alternative Name Package

The concept of ALTERNATIVE NAME is supported for many Part3 elements so is generalised and moved to a separate Alternative Name package in the NeTEx reusable components package.

7.2.2.1 Alternative Name – Physical model

The following diagram shows detailed attributes of the FARE ZONE model

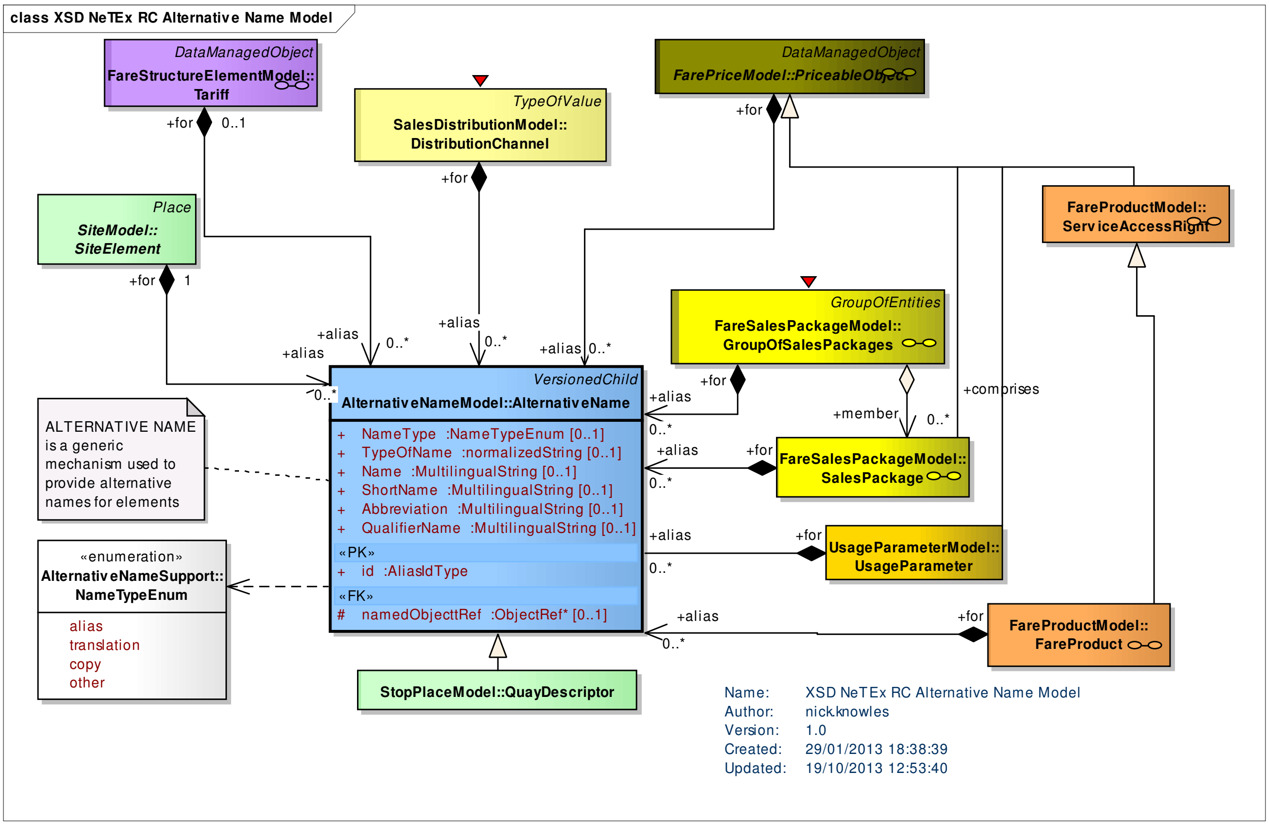


Figure 9 — Alternative Name (Part1 Framework) – Physical Model

7.2.3 Booking Arrangements

The concept of BOOKING ARRANGEMENTs is generalised for reused by Part3 elements and is added to the ACCESS RIGHT TYPEs package in the NeTEx reusable components package.

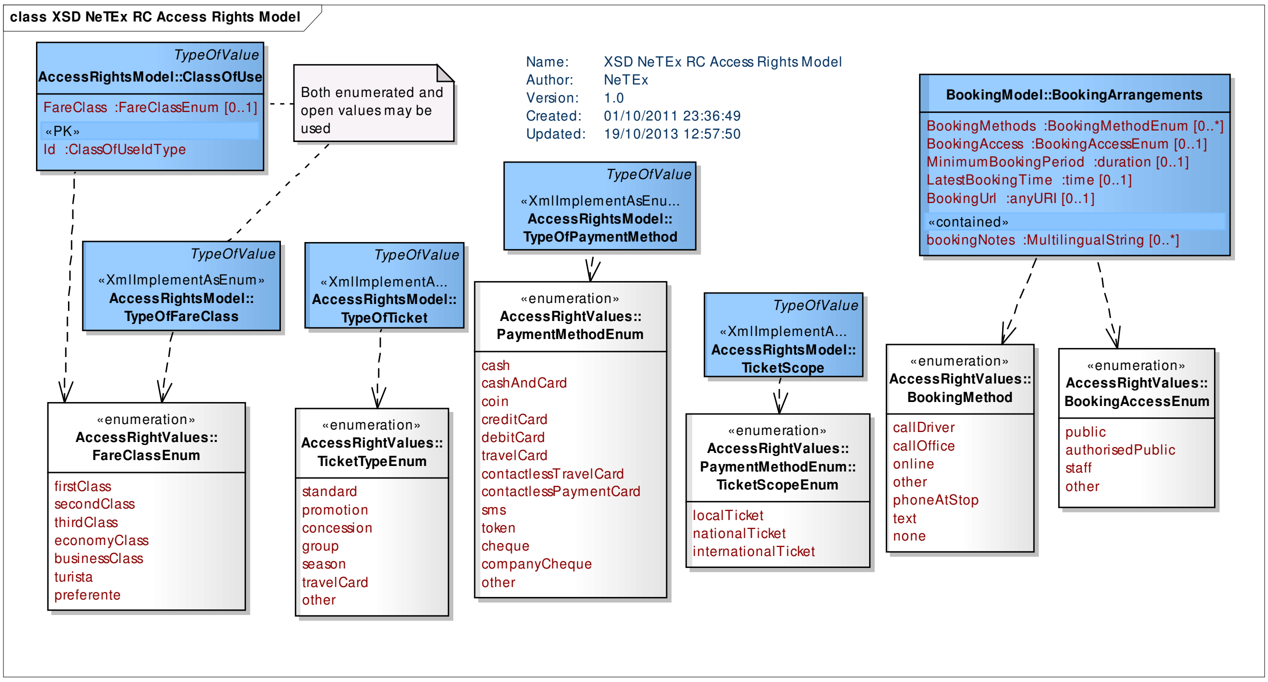


Figure 10 — Access Rights (Part1 Framework) – Physical Model

8 Public transport fares – Conceptual and physical data model

8.1 Introduction

8.2 Conceptual Model overview

8.2.1 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.

“Fare policy specification” and “Provision of information on fares” are in the scope of NeTEx.

“Pricing”, limited to “pricing parameters specification” are as well part of NeTEx and represent an extension to Transmodel V5.1,

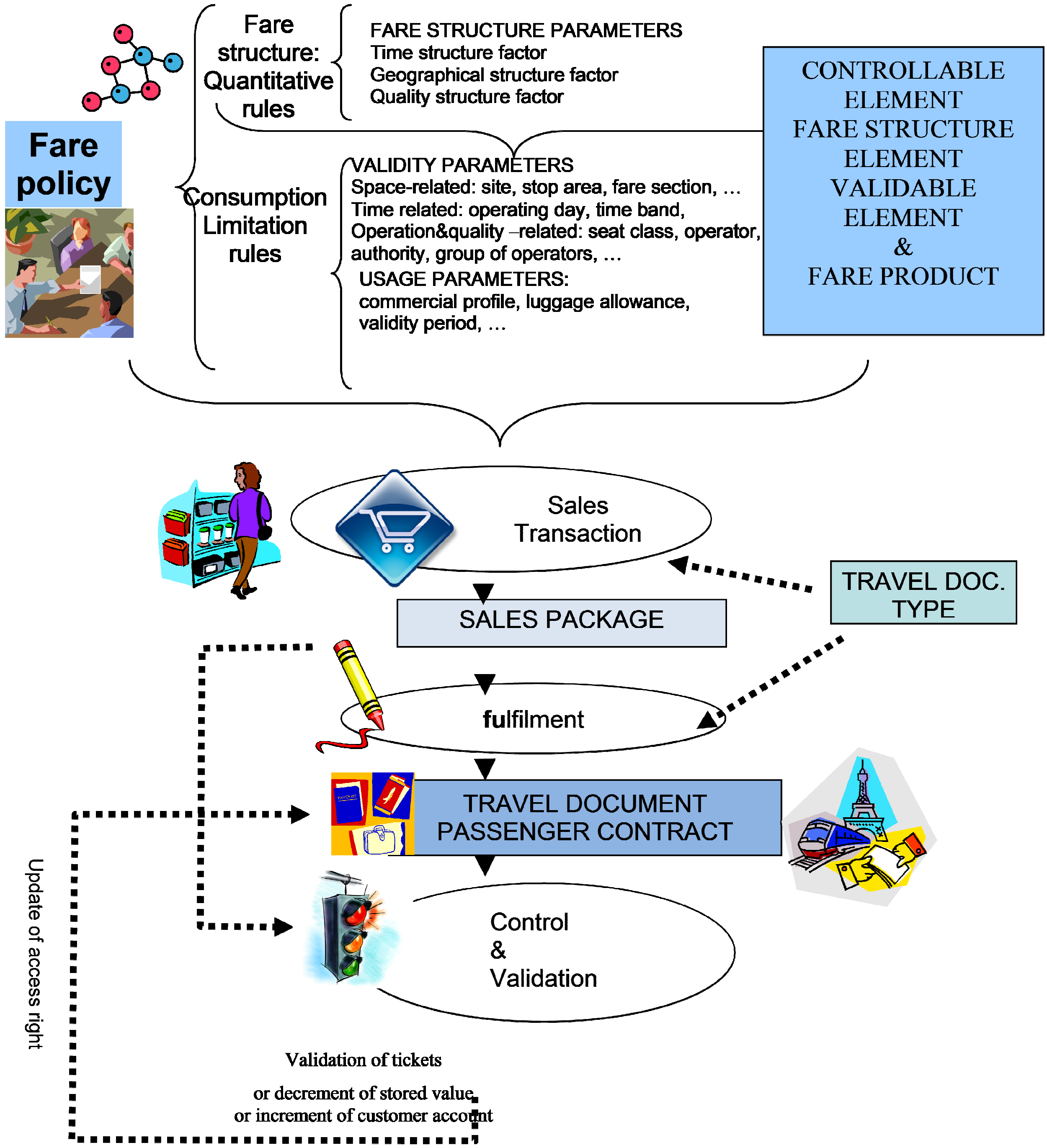


Figure 11 — Fare model overview

8.2.2 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 simple ticket or multiple trips during one month" granted by a monthly pass), which are marketable sets of access rights. One or several fare products may be associated to a "travel document" (e.g. a 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 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 contacts, 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 packages, in order to calculate the end price to be paid by the user.

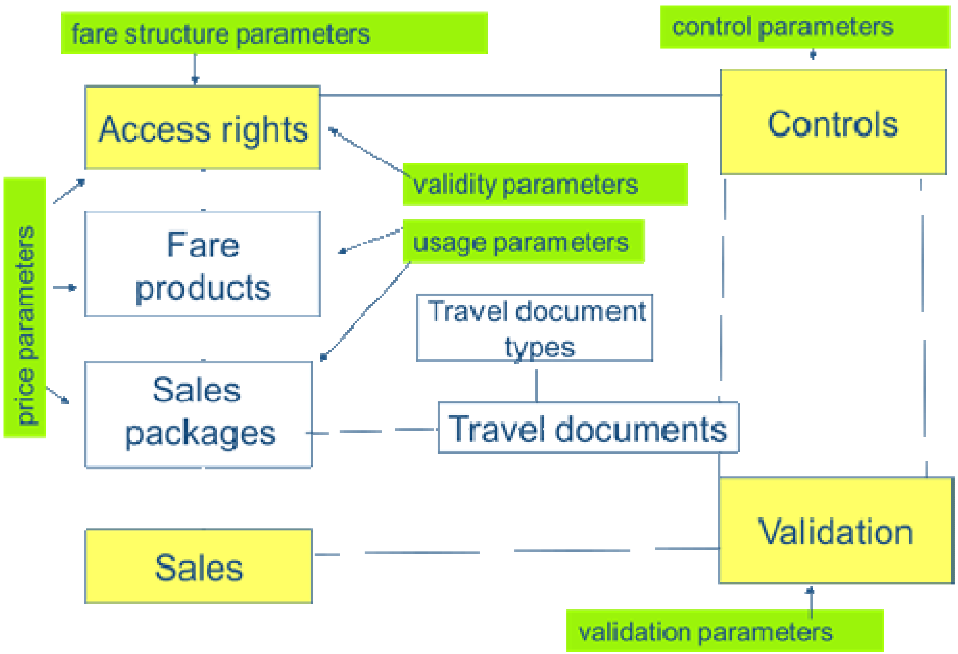


Figure 12 — Fare model main processes and concepts[KB9]

8.2.3 Main Concepts

8.2.3.1 Access Rights overview

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

— elements of the service offered for consumption through a set of quantitative rules determined by space- or time-related parameters (e.g. the right to consume a ride, i.e. the possibility to carry out a ride on a service journey from one stop point to another on one single vehicle, accessing a closed system such as a metro network, entering a car park, spending some time in a 1st class waiting room, etc.). Such elementary access rights may be combined in various marketable combinations;

— the definition of validity limitation rules, specifying the specific conditions to consume these elements with use of validity parameters (mainly time or space limitations such as “valid only for a journey”) 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, etc), 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.).

8.2.3.2 Fare Structure

The definition of a fare structure is based on generic quantitative rules that influence the access rights regulating the consumption, as well as the price a passenger has to pay for a specific consumption: 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 (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:

— Quantitative rules, determined by “fare structure parameters”, used to limit the validity of a range of access rights or to form 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.);

— Access rights (i.e. service consumption) limitation rules which consist in assigning certain “limiting parameters” to specific access rights. 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);
  — “usage parameters”, which affect the use of access rights (user profile, frequency of use, transferability, etc.).

The description of access rights is organised in a hierarchy of three levels:

— CONTROLLABLE ELEMENTs,
— FARE STRUCTURE ELEMENTs
VALIDABLE ELEMENTs.

8.2.3.3 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 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.

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

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 sum up: the different access rights components are defined through rules (quantitative and consumption limitation rules). Quantitative rules are determined through fare structure parameters, sometimes combined with validity limitation rules; validity limitation rules are determined through validity and usage parameters. The figure below presents schematically this situation:

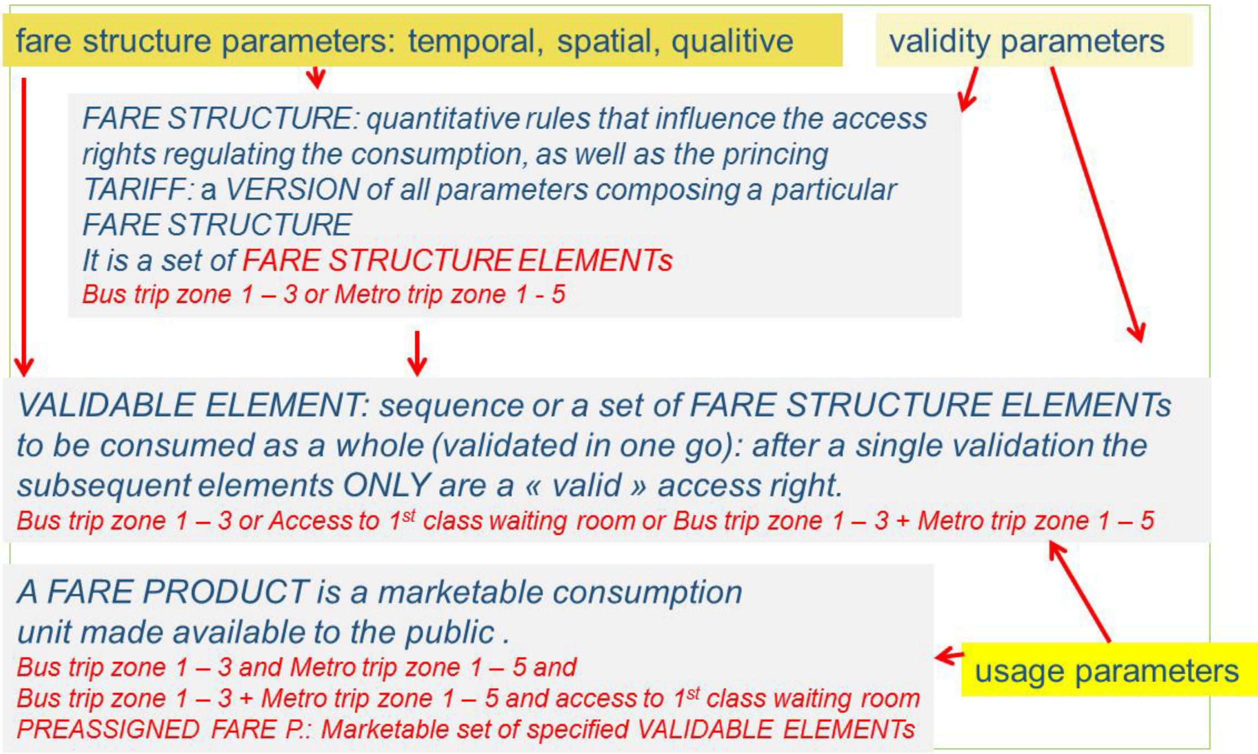


Figure 13 — Overview of the different parameters determining the components of access rights[KB10][KB11]

8.2.3.4 Travel Documents and Sales Packages[KB12]

Summary to be provided

8.2.3.5 Sales Transactions

Summary to be provided

8.2.3.6 Prices

Summary to be provided

END OF the Overview

8.3 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 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.

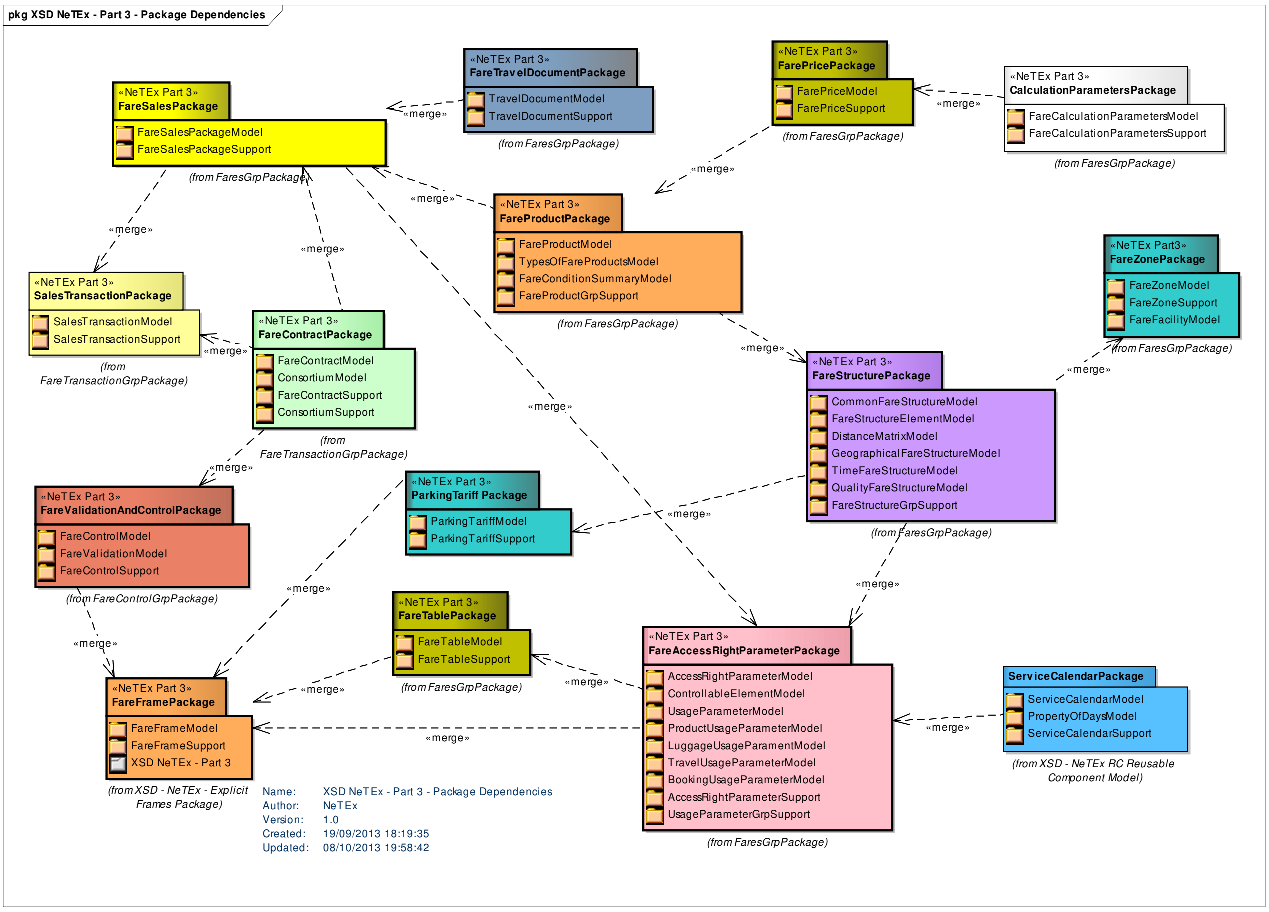


Figure 14 — Fare Package Dependencies

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

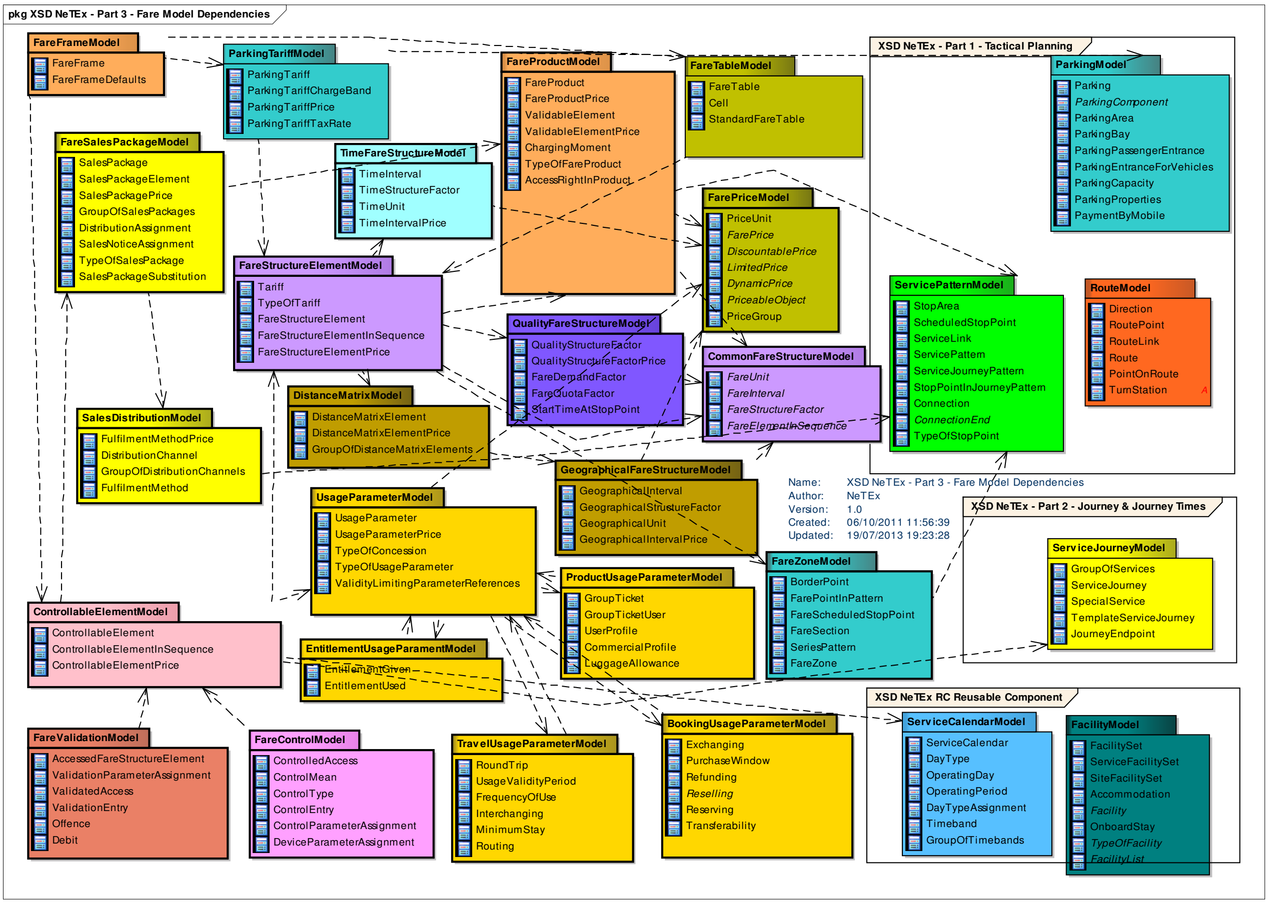


Figure 15 — Fare Model Dependencies

8.3.1 Fare Frame

8.3.1.1 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 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 FRfAMEs 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.

8.3.1.2 Fare Frame – Physical Model

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

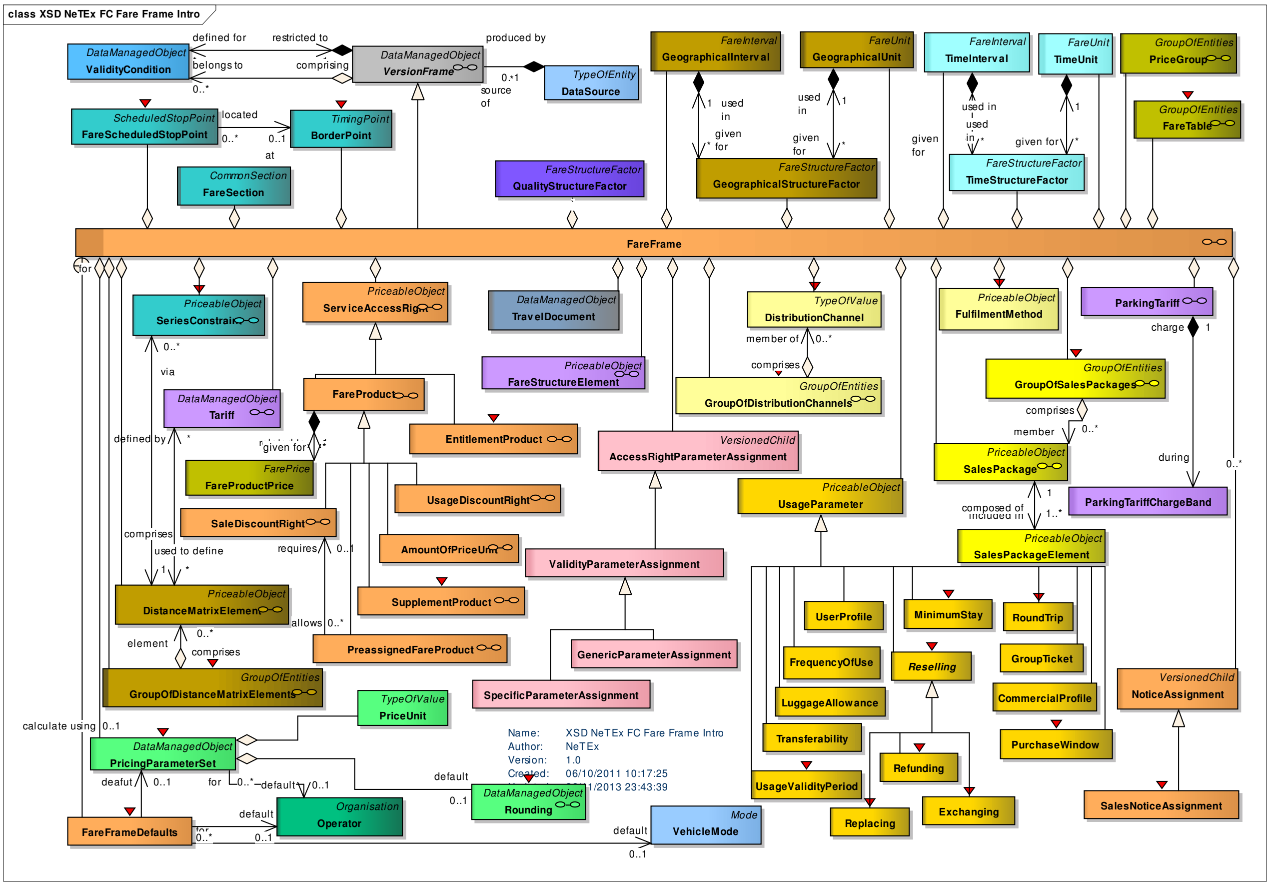


Figure 16 — Fare Frame Contents – Physical Model (UML)

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

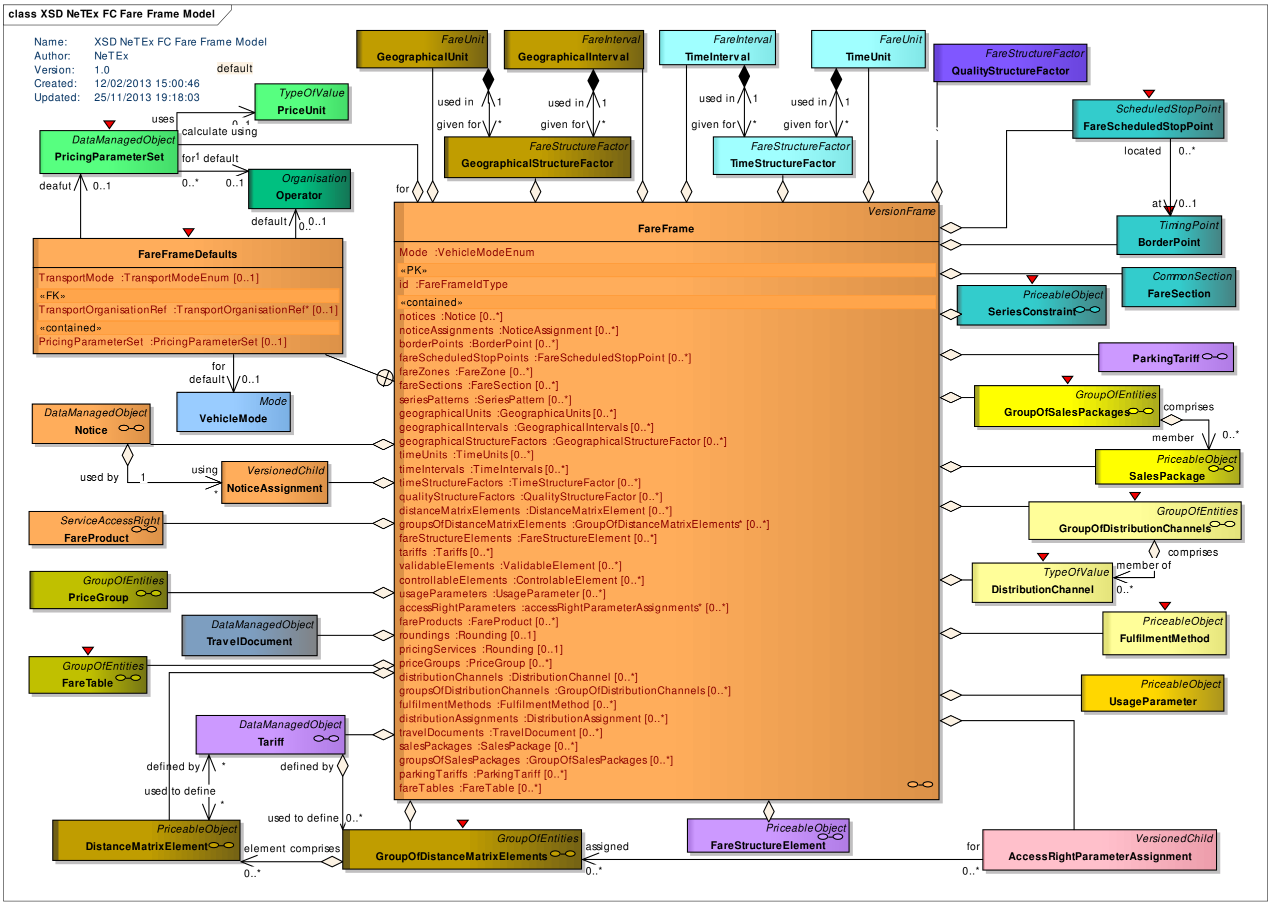


Figure 17 — Fare Frame – Physical Model Detail (UML)

8.3.1.3 Fare Frame — Attributes and XSD

FARE FRAMEs group together sets of fare data for exchange.

8.3.1.3.1 FareFrame – Model Element

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

Table 4 – *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***	*ServiceCalendarIdType*	1:1	Reference to SERVICE CALENDAR for FARE FRAME.
GROUP	***FareDefaultsGroup***	*FareDefaultsGroup*	0:1	Default values for pricing calculations - see below.
GROUP	***NoticesInFrame-Group***	*NoticesInFrameGroup*	0:1	NOTICE elements used in frame. See below.

GROUP	***FareRoutingIn-FrameGroup***	*FareRoutingInFrame-Group*	0:1	FARE ROUTING elements used in frame. See below.
GROUP	***FareSeriesInFrame-Group***	*FareSeriesInFrameGroup*	0:1	FARE SERIES CONSTRAINT elements used in frame. See below.
GROUP	***DistanceMatrix-ElementsInFrame-Group***	*DistanceMatrix-ElementsInFrameGroup*	0:1	DISTANCE MATRIX ELEMENT elements used in frame. See below.
GROUP	***FareStructure-ElementsInFrame-Group***	*FareStructureElements-InFrameGroup*	0:1	FARE STRUCTURE elements used in frame. See below.
GROUP	***ValidableElements-InFrameGroup***	*ValidableElementsIn-FrameGroup*	0:1	VALIDABLE ELEMENT elements used in frame. See below.
GROUP	***FareProduct-ElementsInFrame-Group***	*FareProductElements-InFrameGroup*	0:1	FARE PRODUCT elements IN FRAME GROUP elements used in frame. See below.
GROUP	***FarePriceElements-InFrameGroup***	*FarePriceElements-InFrameGroup*	0:1	FARE PRICE elements IN FRAME GROUP elements used in frame. See below.
GROUP	***SalesDistribution-InFrameGroup***	*SalesDistribution-ElementsInFrameGroup*	0:1	SALES DISTRIBUTION elements IN FRAME GROUP elements used in frame. See below.
GROUP	***TravelDocuments-InFrameGroup***	*TravelDocuments-InFrameGroup*	0:1	TRAVEL DOCUMENTs elements IN FRAME GROUP elements used in frame. See below.
GROUP	***SalesPackage-InFrameGroup***	*SalesPackage-InFrameGroup*	0:1	SALES PACKAGE elements IN FRAME GROUP elements used in frame. See below.
GROUP	***ParkingTariff-InFrameGroup***	*ParkingTariffInFrame-Group*	0:1	PARKING TARIFF elements IN FRAME GROUP elements used in frame. See below.

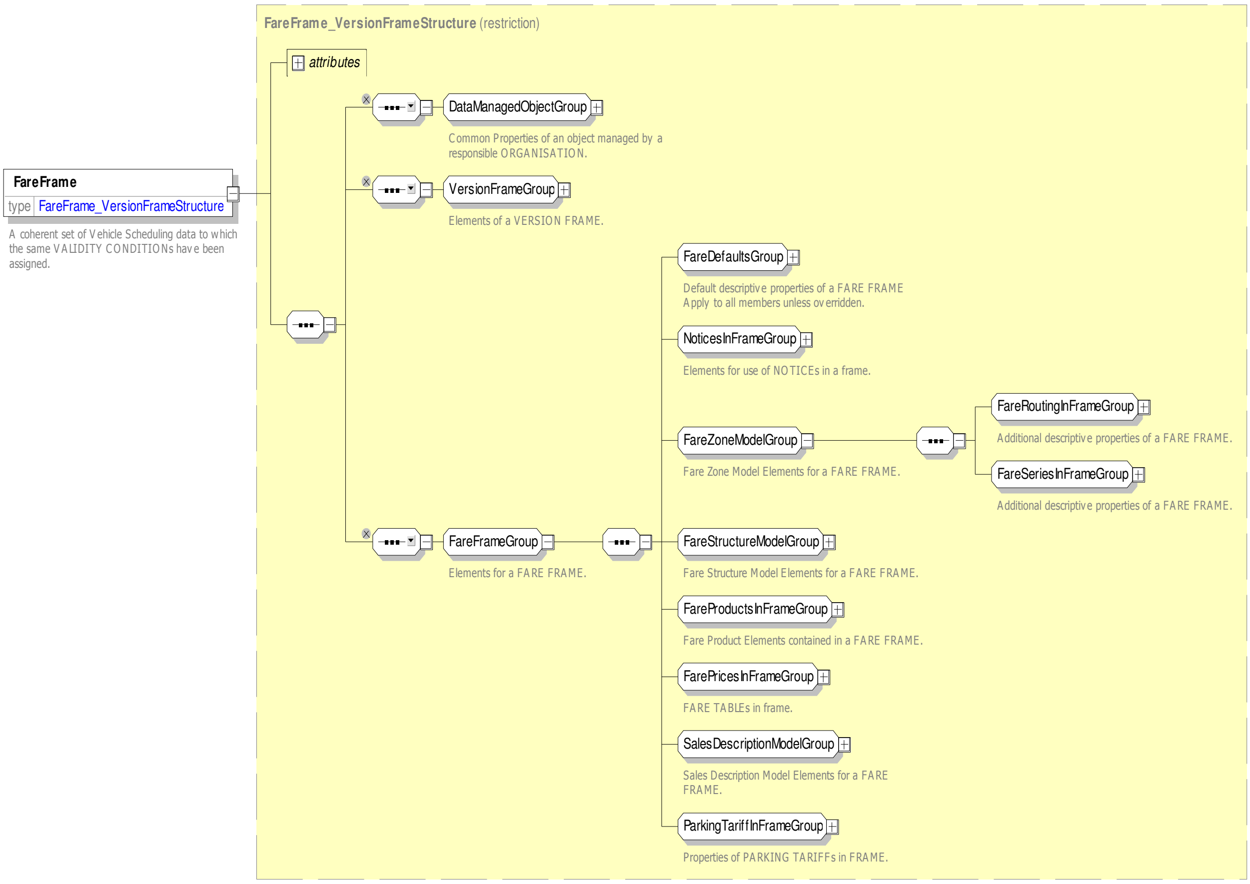


Figure 18 — *FareFrame* — XSD

8.3.1.3.2 *FareDefaultsGroup — 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 5 – *FareFrameDefaults* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
«FK»	**Mode**	*VehicleModeEnum*	0:1	Default vehicle mode for FARE FRAME.
«FK»	***Transport-OrganisationRef***	*Transport-OrganisationRef*	0:1	Default organisation for FARE FRAME.
«FK»	***Pricing-ParameterSet***	*PricingParameterSet*	0:1	PRICING PARAMETER SET associated with FARE FRAME. See later.

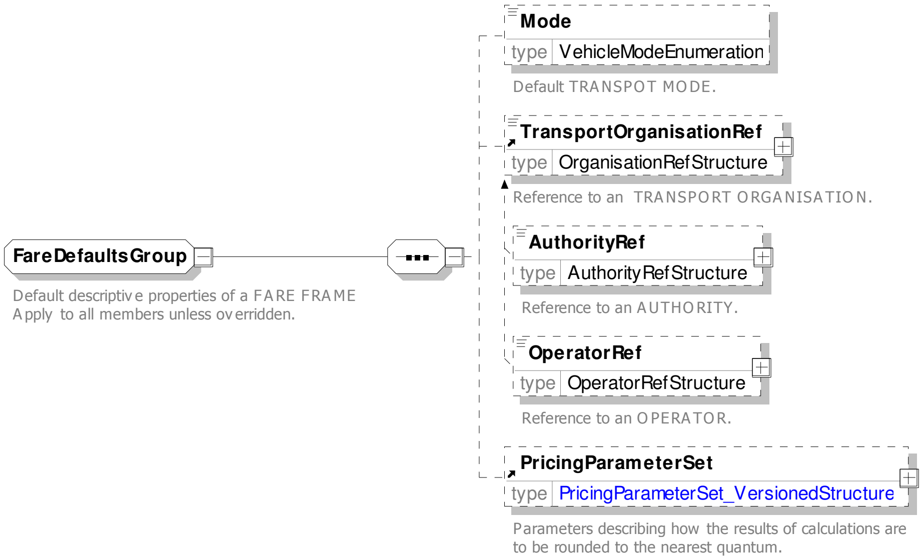


Figure 19 — *FareDefaultsGroup* — XSD

8.3.1.3.3 *NoticesInFrameGroup — Group*

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

Table 6 – *NoticesInFrameGroup* – Group

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

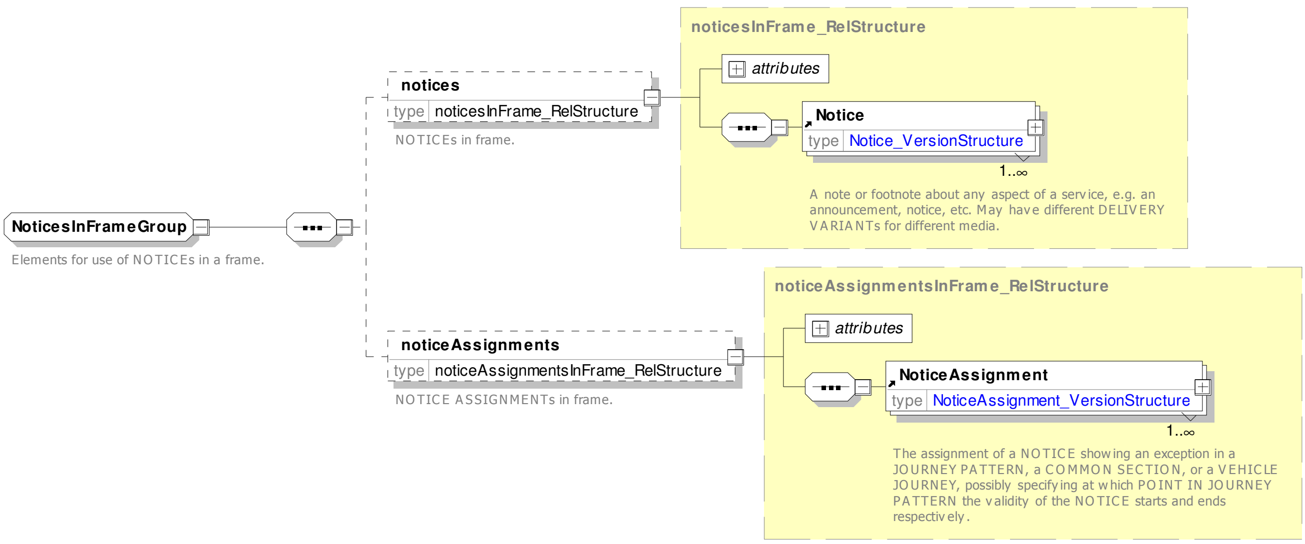


Figure 20 — *NoticesInFrameGroup* — XSD

8.3.1.3.4 Fare Zone — *Groups*

8.3.1.3.4.1 *FareRoutingInFrameGroup — Group*

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

Table 7 – *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.

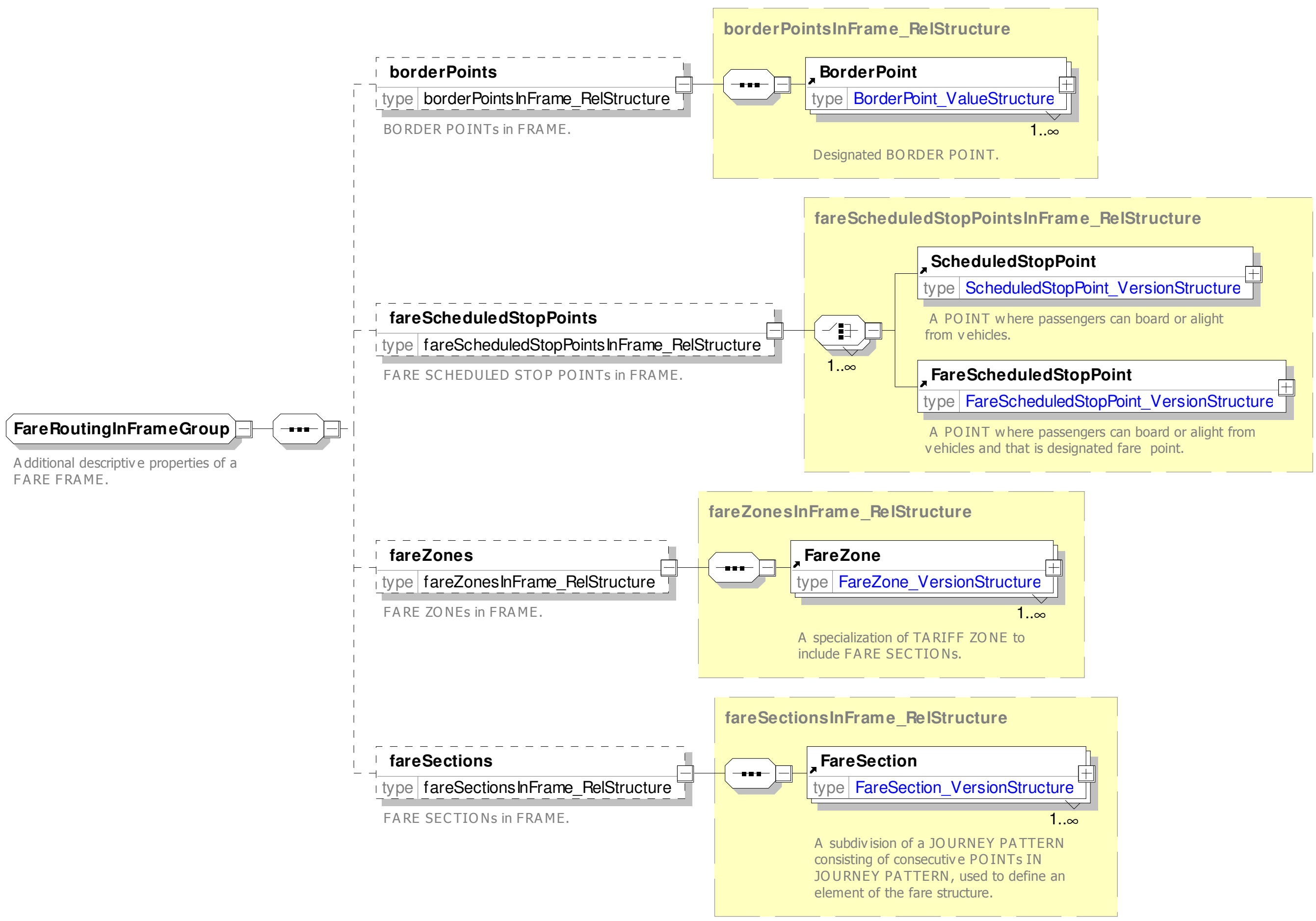


Figure 21 — *FareRoutingInFrameGroup* — XSD

8.3.1.3.4.2 *FareSeriesInFrameGroup — Group*

The ***FareSeriesInFrameGroup*** holds the SERIES CONSTRAINT elements for the frame.

Table 8 – *FareSeriesInFrameGroup* – Group

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

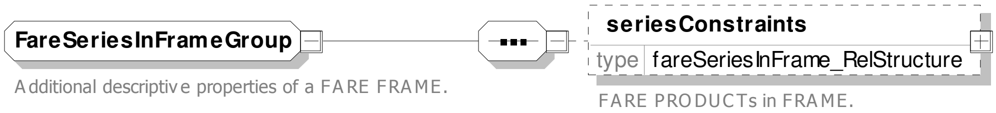


Figure 22 — *FareSeriesInFrameGroup* — XSD

8.3.1.3.5 Fare Structure — *Groups*

8.3.1.3.5.1 *FareStructureModelGroup — Group*

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

Table 9 – *FareStructureModelGroup* – Group

Classifi-cation	Name	Type	Cardin-ality	Description
[GROUP]	***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.

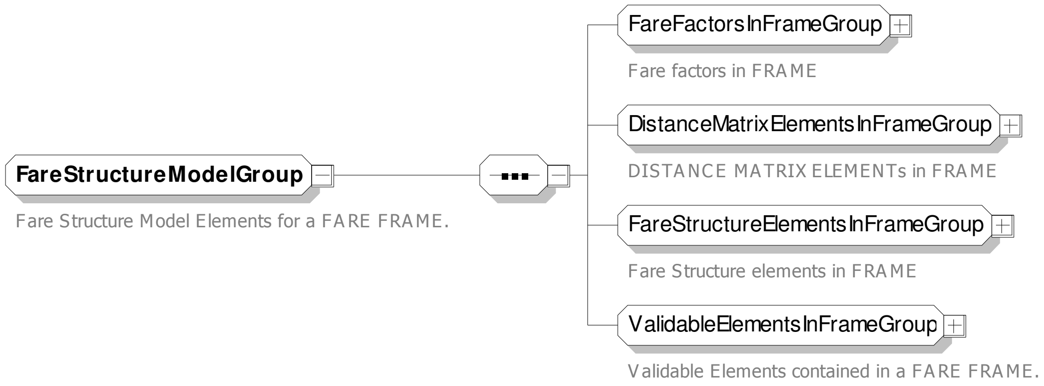


Figure 23 — *FareStructureModelGroup* — XSD

8.3.1.3.5.2 *FareFactorsInFrameGroup* — *Group*

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

Table 10 – *FareFactorsInFrameGroup* – Group

Classifi-cation	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.

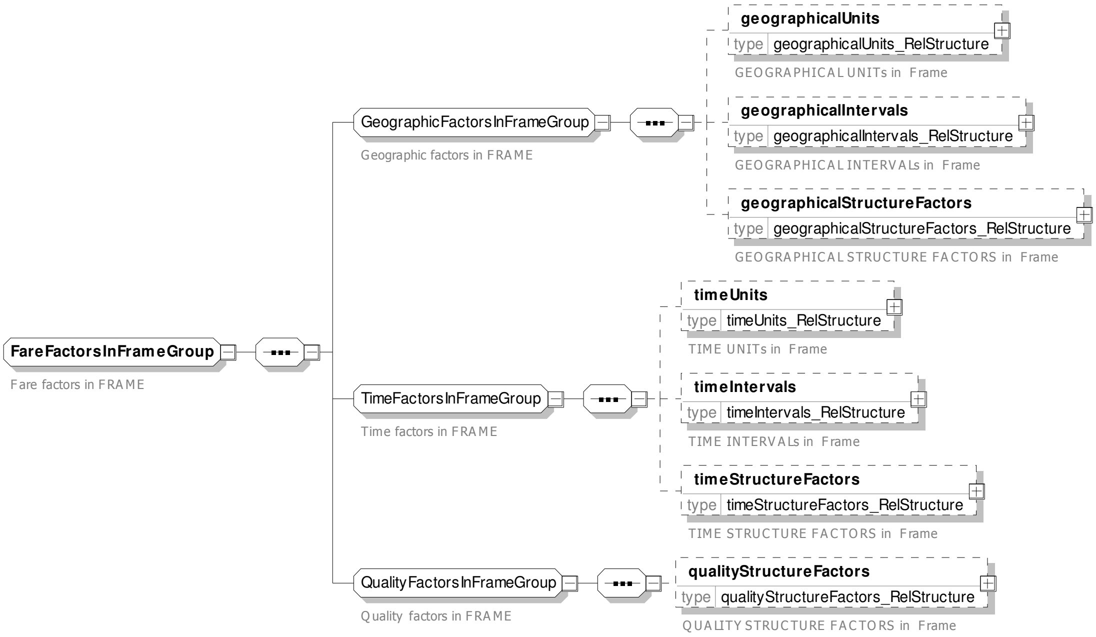


Figure 24 — *FareFactorsInFrameGroup*— XSD

8.3.1.3.5.3 *DistanceMatrixeElementsInFrameGroup — Group*

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

Table 11 – *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.

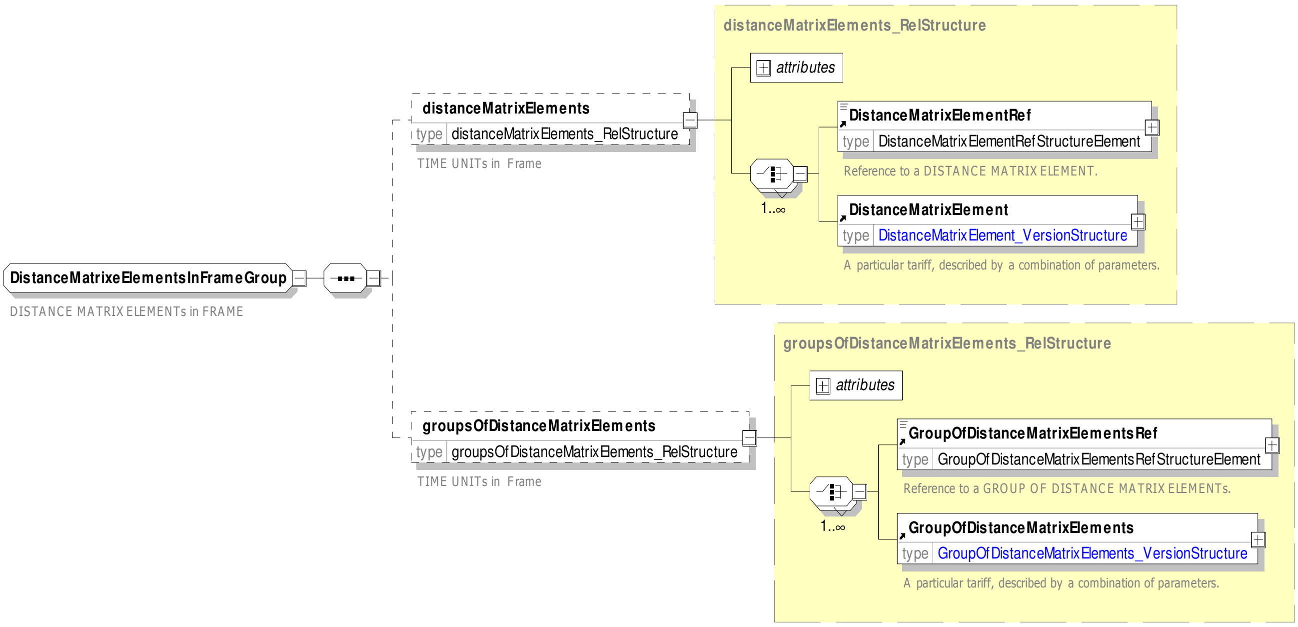


Figure 25 — *DistanceMatrixeElementsInFrameGroup* — XSD

8.3.1.3.5.4 *FareStructureElementsInFrameGroup — Group*

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

Table 12 – *FareStructureElementsInFrameGroup* – Group

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

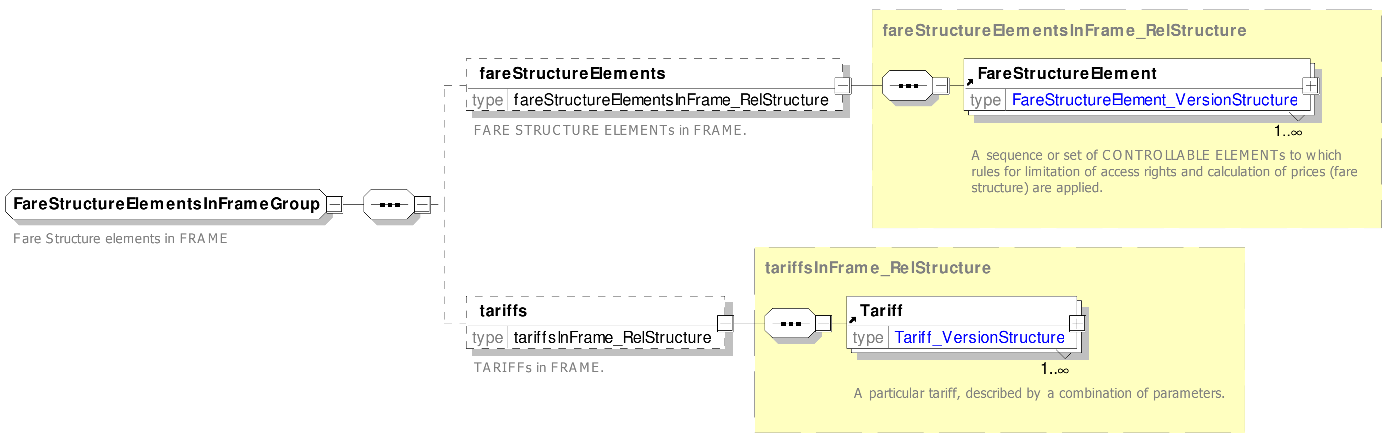


Figure 26 — *FareStructureElementsInFrameGroup* — XSD

8.3.1.3.5.5 *ValidableElementsInFrameGroup — Group*

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

Table 13 – *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.

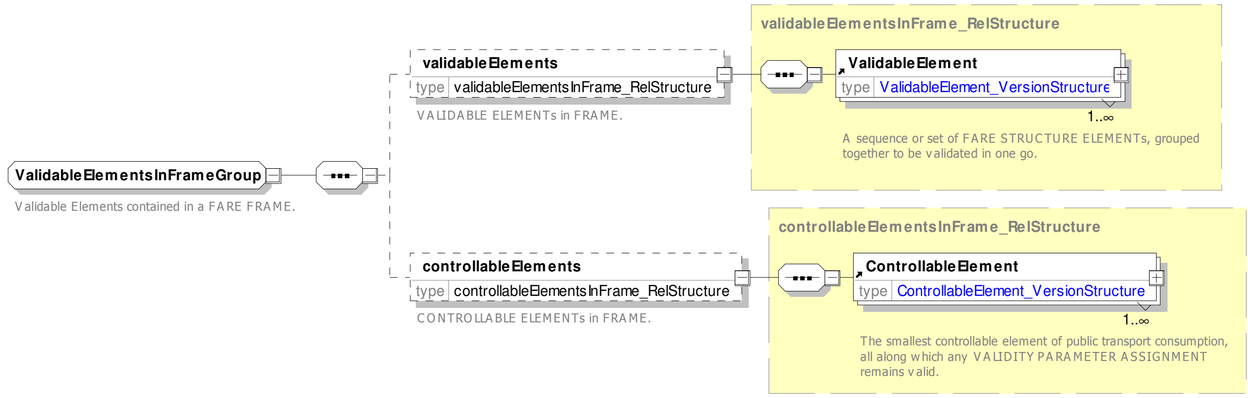


Figure 27 — *ValidableElementsInFrameGroup* — XSD

8.3.1.3.6 Fare Products — *Groups*

8.3.1.3.6.1 *FareProductsInFrameGroup — 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 14 – *FareProductsInFrameGroup* – Group

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

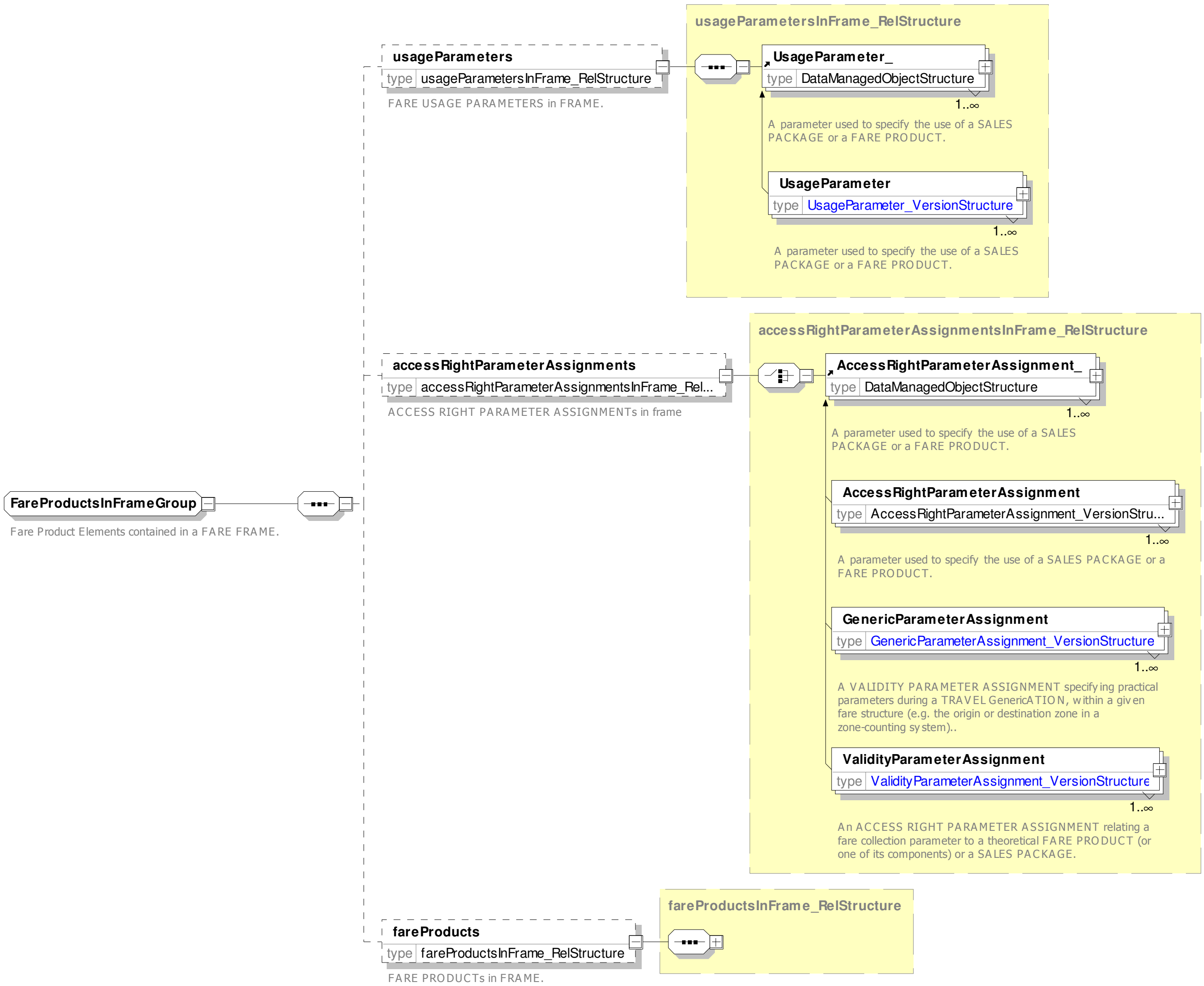


Figure 28 — *FareProductsInFrameGroup* — XSD

8.3.1.3.7 *FarePricesInFrameGroup — Group*

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

Table 15 – *FarePricesInFrameGroup* – Group

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

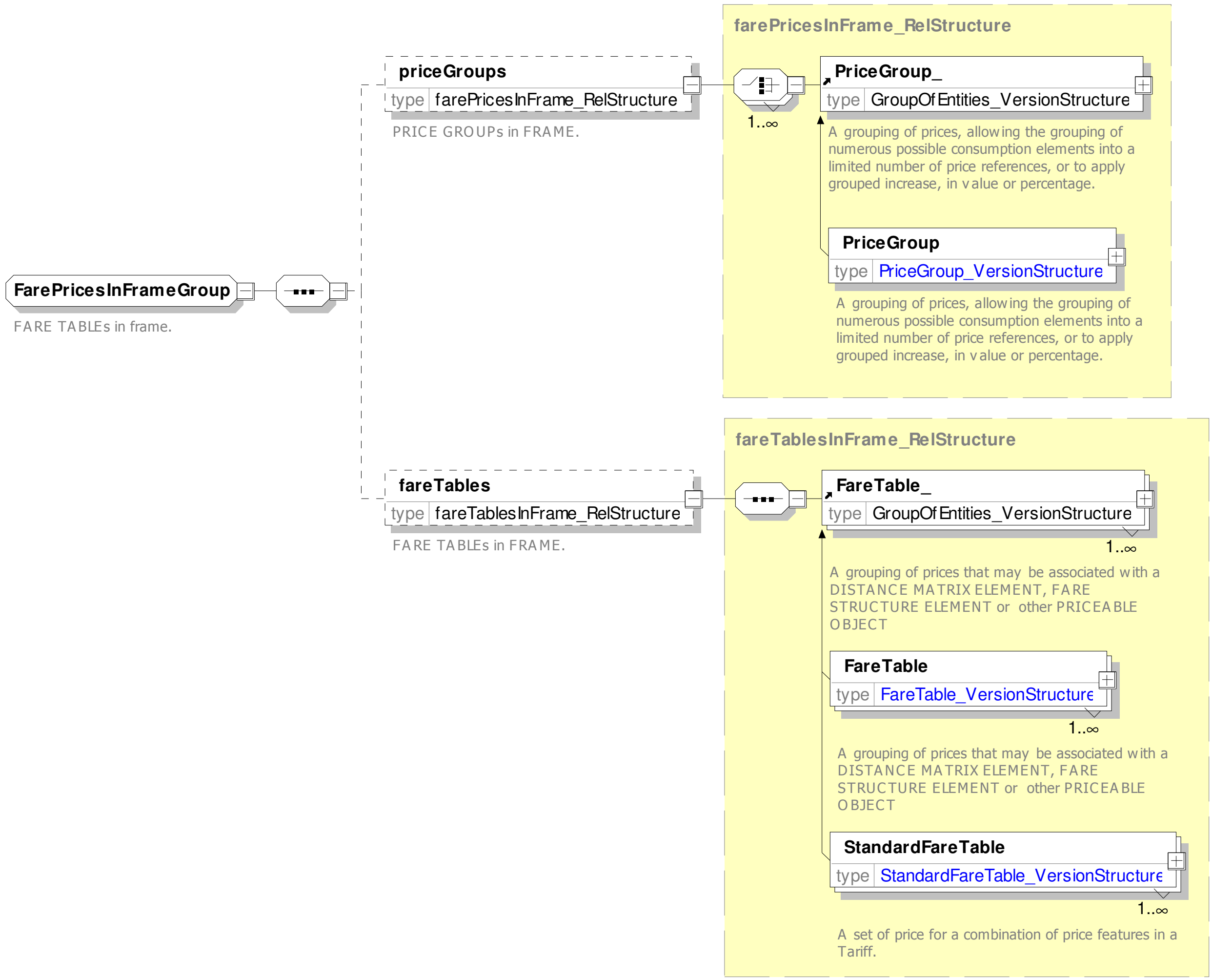


Figure 29 — *FarePricesInFrameGroup* — XSD

8.3.1.3.8 Sales Description — *Groups*

8.3.1.3.8.1 *SalesDistributionInFrameGroup* — *Group*

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

Table 16 – *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.

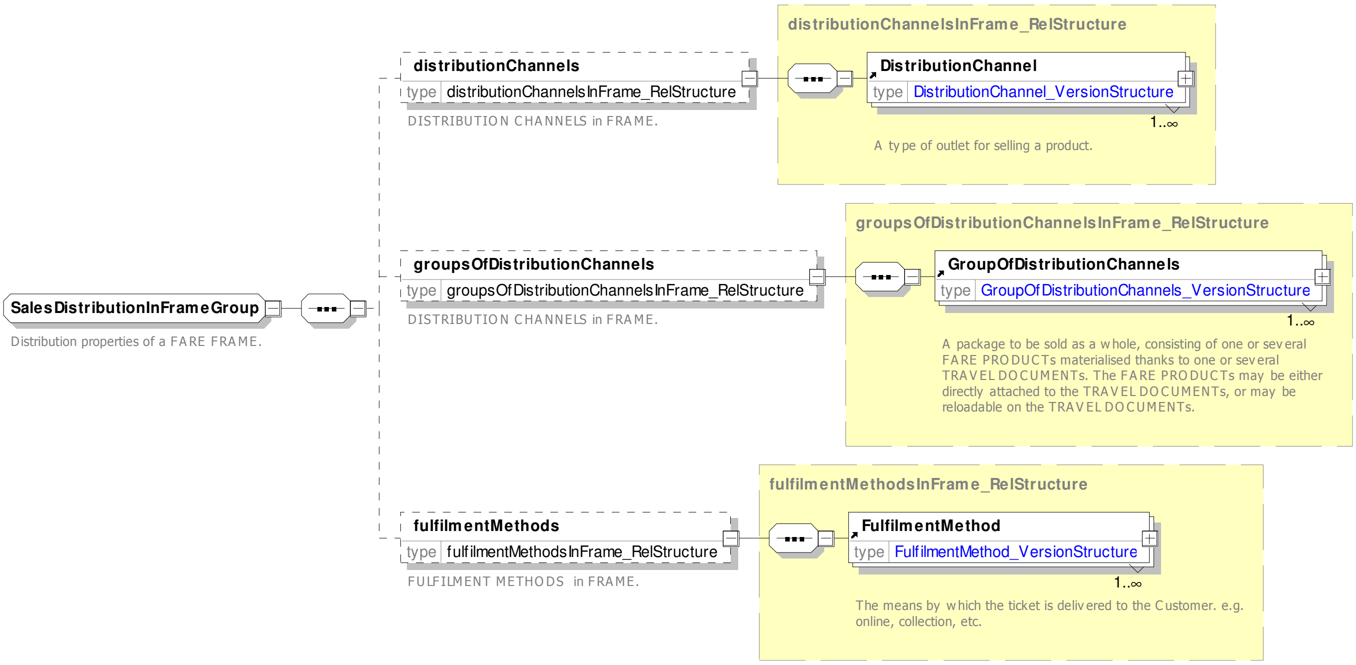


Figure 30 — *SalesDistributionInFrameGroup* — XSD

8.3.1.3.8.2 *TravelDocumentsInFrameGroup — Group*

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

Table 17 – *TravelDocumentsInFrameGroup* – Group

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

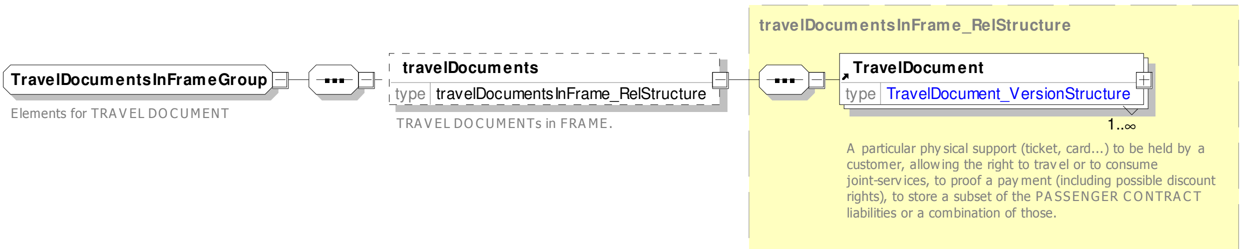


Figure 31 — *TravelDocumentsInFrameGroup* — XSD

8.3.1.3.8.3 *SalesPackageInFrameGroup — Group*

The ***SalesPackageInFrameGroup*** holds the SALES PACKAGE elements for the frame.

Table 18 – *SalesPackageInFrameGroup* – Group

Classifi-	Name	Type	Cardinality	Description

cation				
“cntd»	***salesPackages***	*SalesPackage*	0:*	SALES PACKAGEs in FARE FRAME.
“cntd»	***groupsOfSales-Packages***	*GroupOfSalesPackages*	0:*	GROUPs OF SALES PACKAGES in FARE FRAME.
“cntd»	***distribution-Assignments***	*DistributionAssignment*	0:*	DISTRIBUTION ASSIGNMENTs in the FARE FRAME.

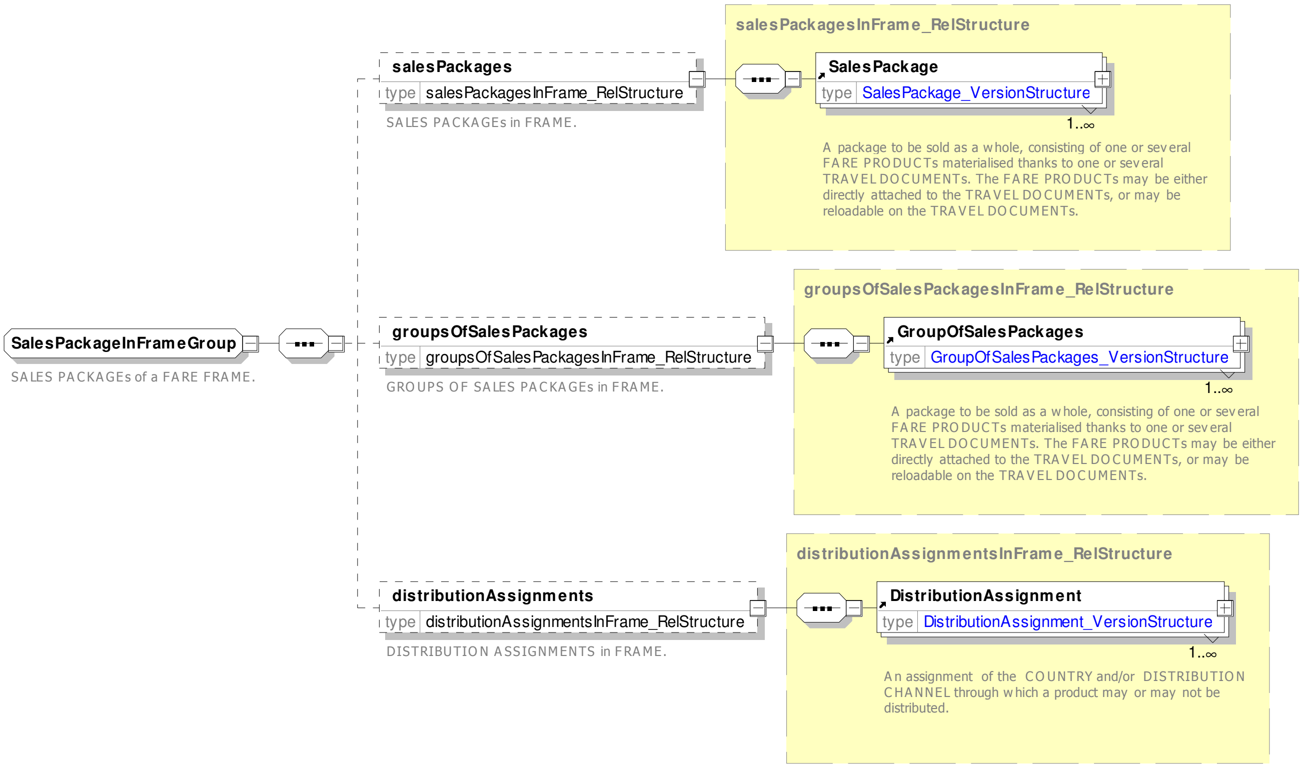


Figure 32 — *SalesPackageInFrameGroup* — XSD

8.3.1.3.9 ParkingTariff — *Groups*

8.3.1.3.9.1 *ParkingTariffInFrameGroup— Group*

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

Table 19 – *ParkingTariffInFrameGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
“cntd»	***parkingTariffs***	*ParkingTariff*	0:*	PARKING TARIFFs in FARE FRAME.

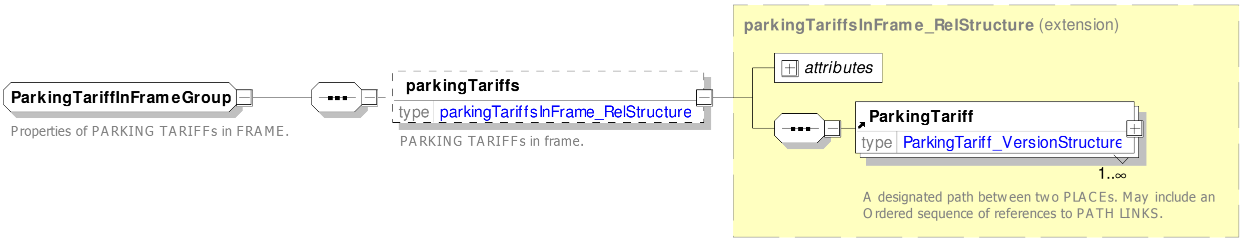


Figure 33 — *ParkingTariffInFrameGroup*— XSD

8.4 Reusable Fare Components

8.4.1 Fare Zone

8.4.1.1 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. The NeTEx Part3 FARE ZONE MODEL presents additional concepts relating to the network that can be used 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.

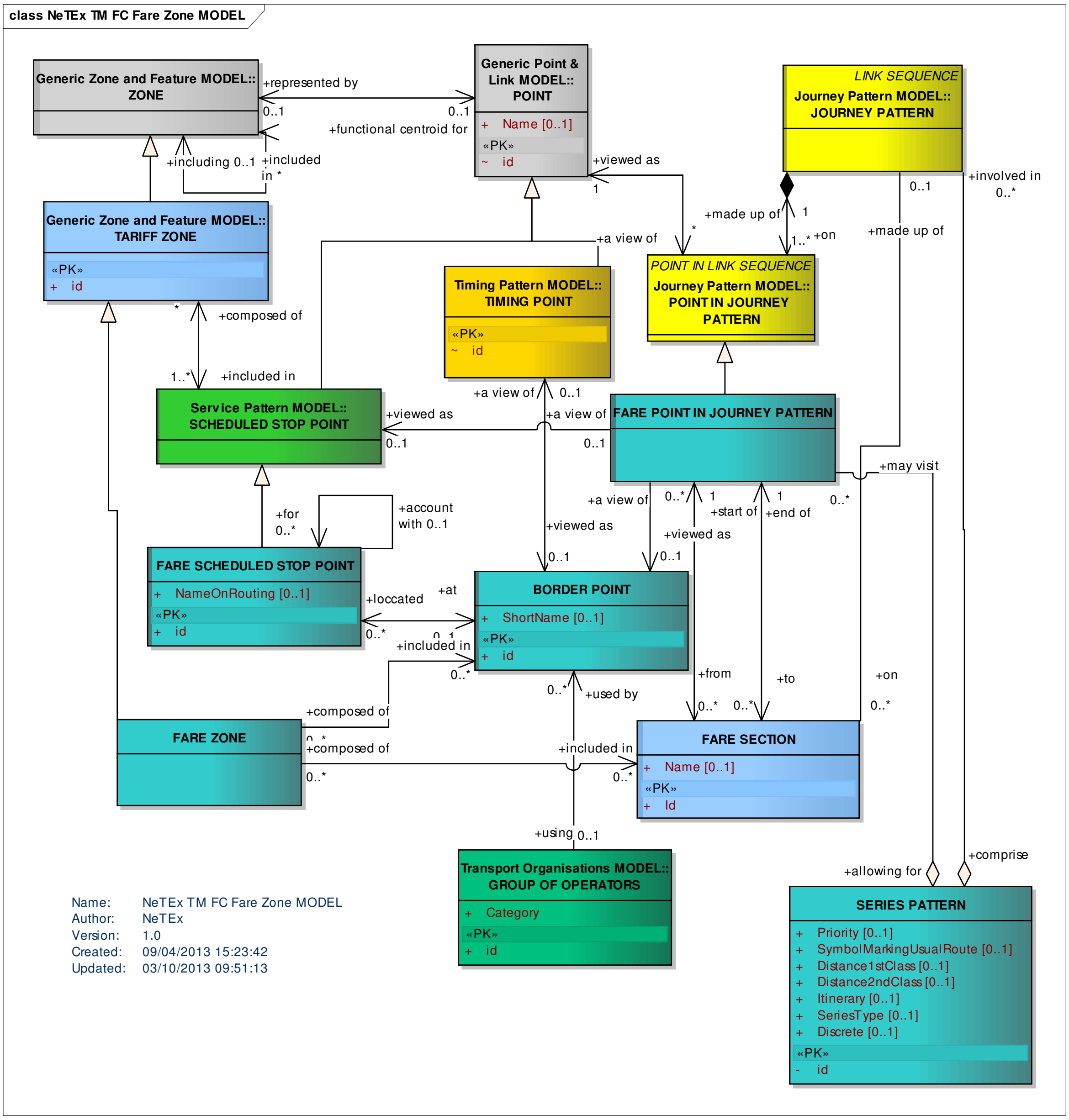


Figure 34 — Fare Zone – Conceptual Model

8.4.1.2 Fare Zone – Conceptual Examples

8.4.1.2.1 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 36) or (ii) *[A * C * D]* (Figure 36), 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 37), and (iv) (Figure 38), . Thus a ticket [A * D] with SERIES CONSTRAINT (B/C) would allow travel along any of the following four routes:

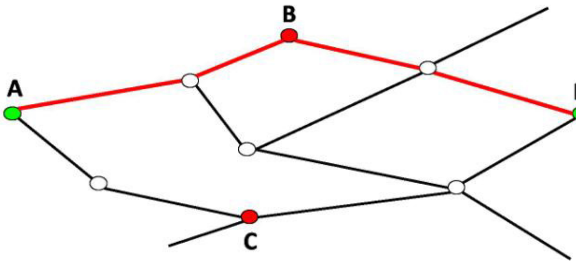


Figure 35 — Series Example (i) [A * B * D]

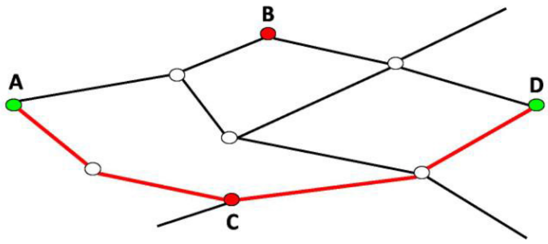


Figure 36 — Series Example (ii) [A * C * D]

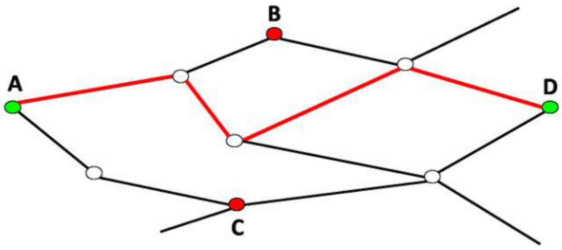


Figure 37 — Series Example (iii) [A * (B/C) * D]

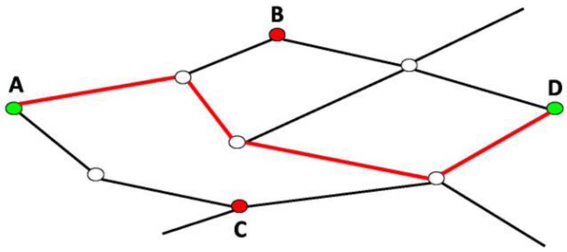


Figure 38 — Series Example (iv) [A * (B/C) * D]

8.4.1.2.1.1 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 20 – 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 39) 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 *optionalShowLeft* ("2" in Figure 39)

— If, however, it is to the right, it is given the value *optionalShowRight (* "3" in Figure 39).

— Where there is no station at all, *noStation* is entered ("0" in Figure 39).

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

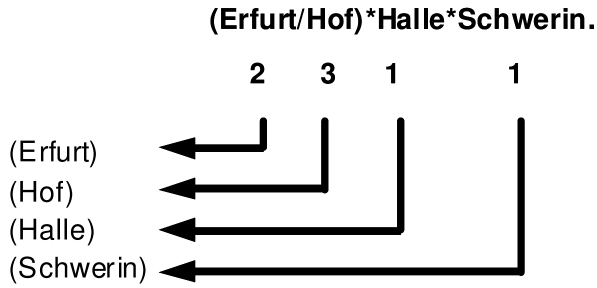


Figure 39 — Example: Use of

8.4.1.2.2 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 networks does not always correspond to its geographical location.

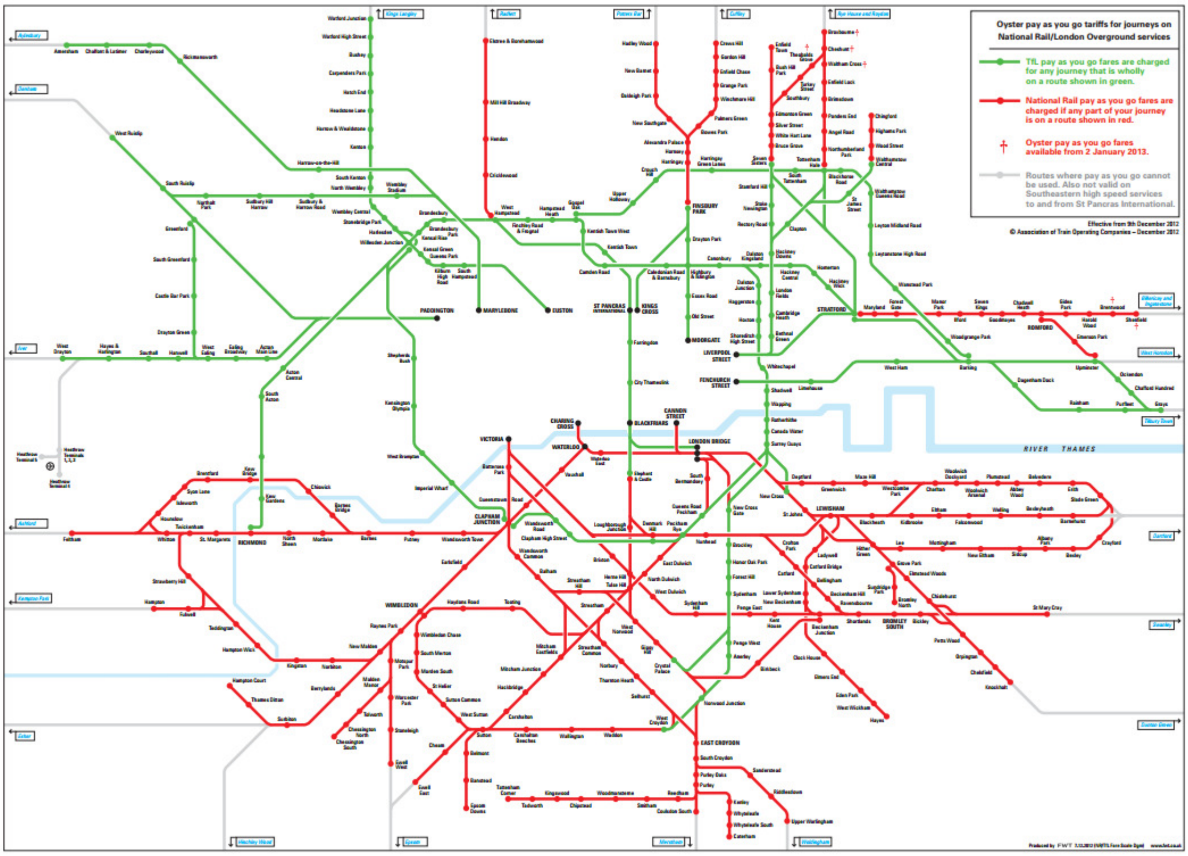


Figure 40 — Example: Transport For London rail fare sections

8.4.1.2.3 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 43) shows a BORDER POINT “0847” defined for “*Brest”* that corresponds to stations either side of the border

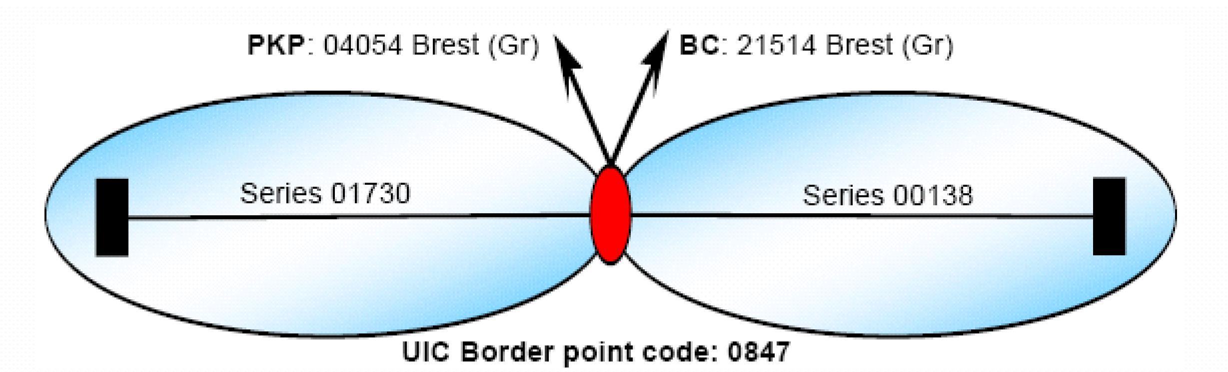


Figure 41 — Example: Border Point for Brest (from Tap/TSI B1)

8.4.1.3 Fare Zone – Physical model

The following diagram shows detailed attributes of the FARE ZONE model

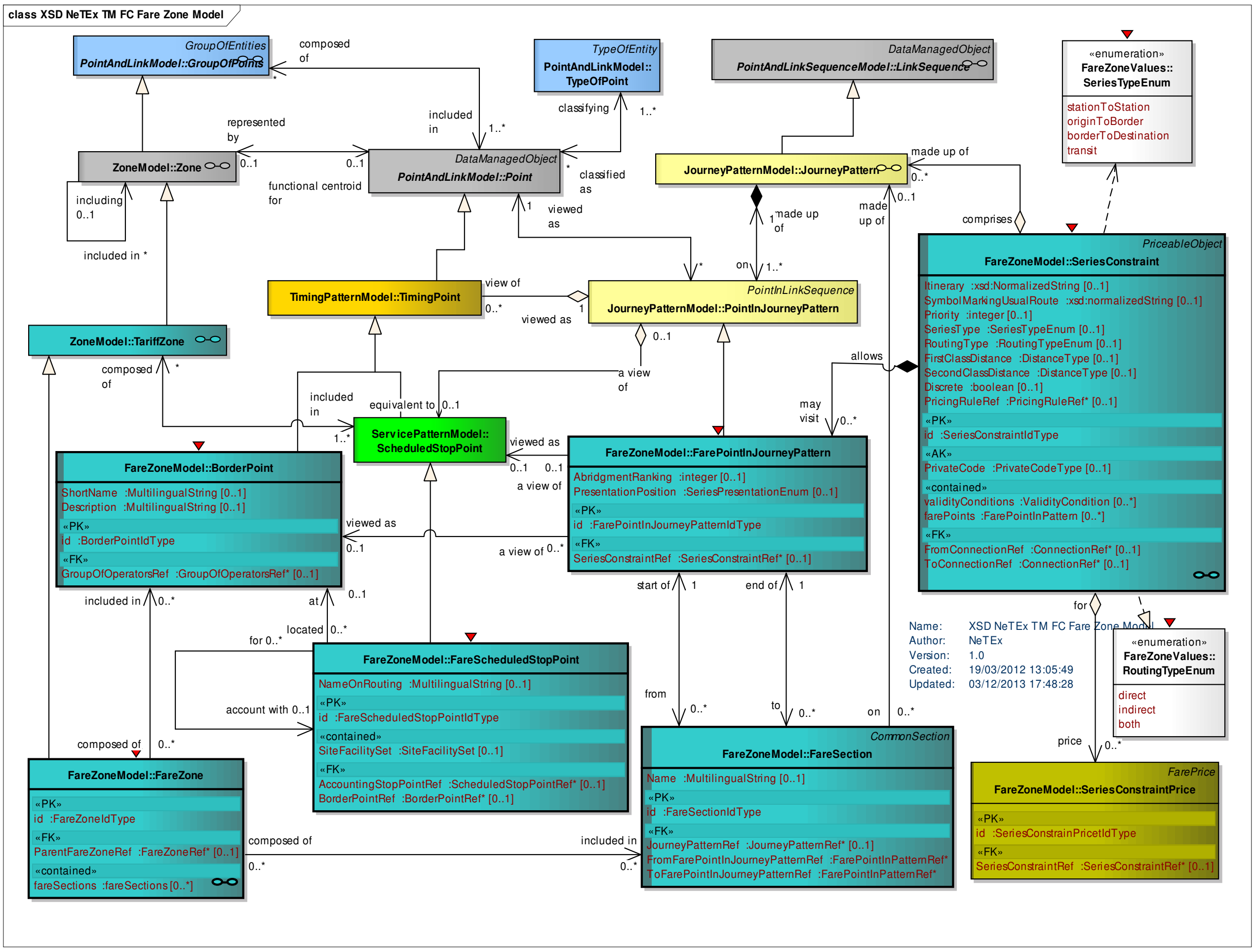


Figure 42 — Fare Zone – Physical Model

8.4.1.4 Fare Zone Model – Attributes and XSD

8.4.1.4.1 BorderPoint – Model Element

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

Table 21 – *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.

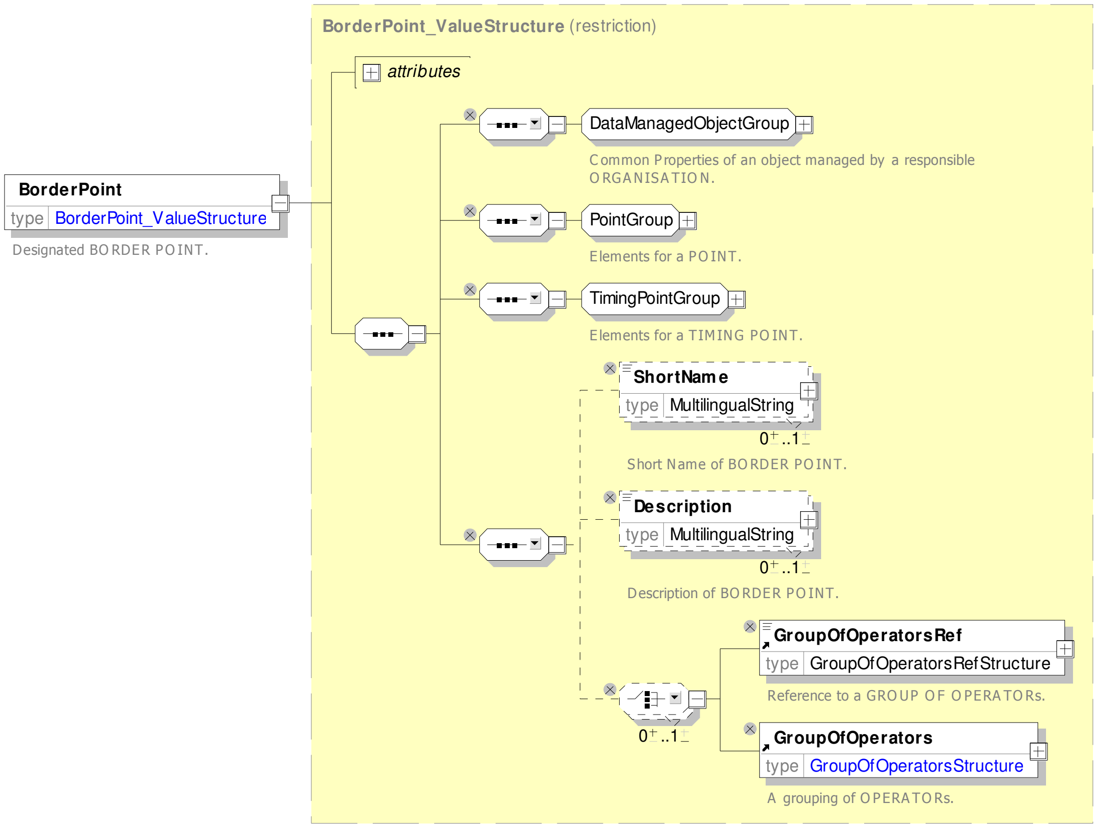


Figure 43 — *BorderPoint* — XSD

8.4.1.4.2 FareScheduledStopPoint – Model Element

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

Table 22 – *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.

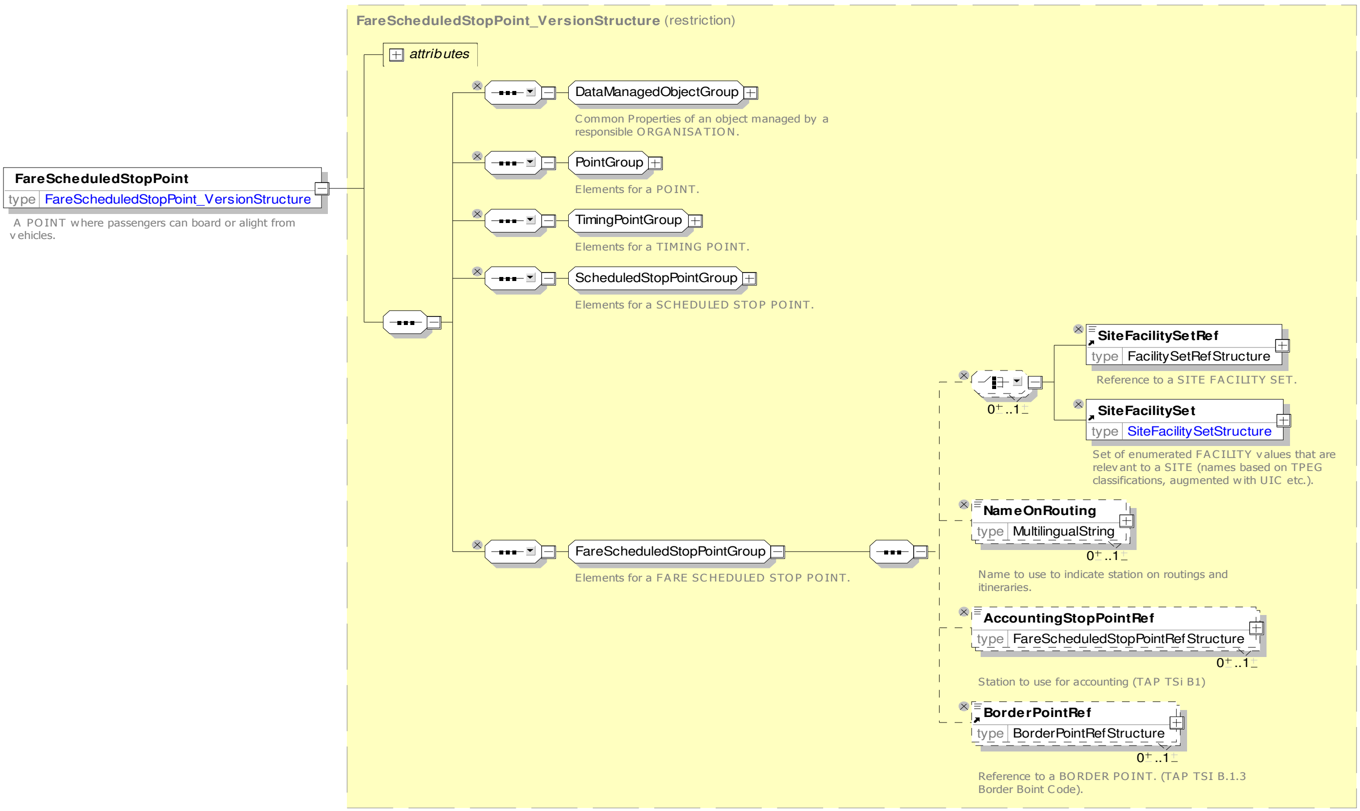


Figure 44 — *FareScheduledStopPoint* — XSD

8.4.1.4.3 FareZone – Model Element

A specialization of TARIFF ZONE to include designated FARE SECTIONs.

Table 23 – *FareZone* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TariffZone*	::>	FARE ZONE inherits from TARIFF ZONE. See NeTEx Part1.
«PK»	***id***	*FareZoneIdType*	1:1	Identifier of FARE ZONE.
«FK»	***FareZoneRef***	*FareZoneRef*	0:1	Parent FARE ZONE of which this is part.
“cntd»	***fareSections***	*fareSections*	0:*	FARE SECTIONs in FARE ZONE.

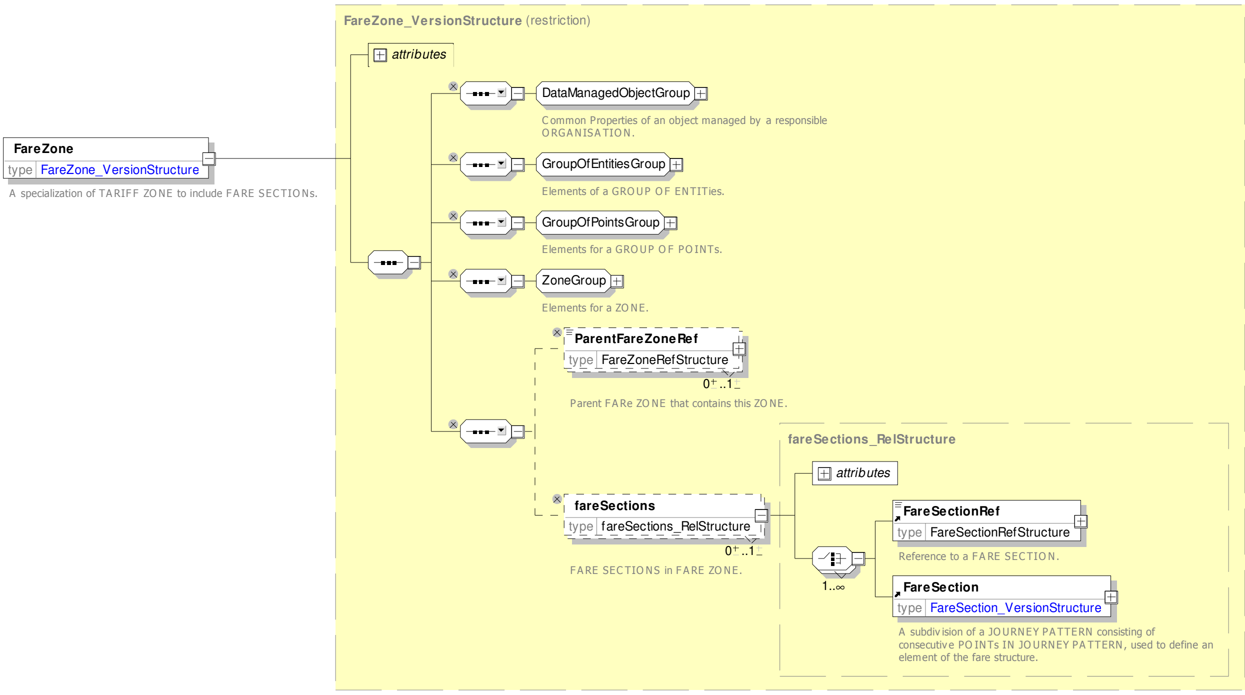


Figure 45 — *FareZone* — XSD

8.4.1.4.4 FareSection – Model Element

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

Table 24 – *FareSection* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*CommonSection*	::>	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***	*FarePointInJourney-PatternRef*	1:1	Reference to FARE POINT IN JOURNEY PATTERN at which FARE SECTION starts.
«FK»	***ToFarePointRef***	*FarePointInJourney-PatternRef*	1:1	Reference to FARE POINT IN JOURNEY PATTERN at which FARE SECTION ends.

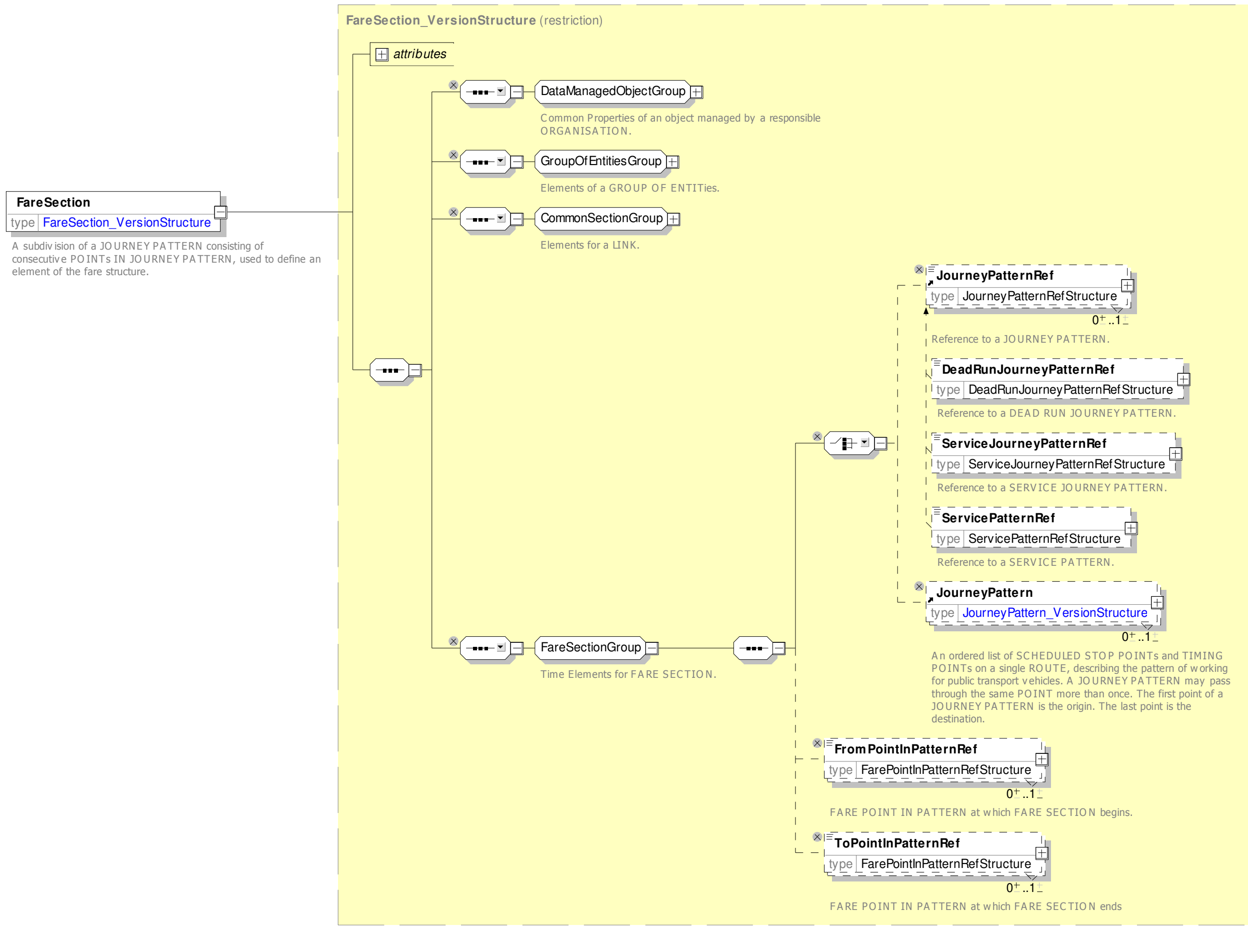


Figure 46 — *FareSection* — XSD

8.4.1.4.5 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 25 – *SeriesConstraint* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	SERIES CONSTRAINT inherits from PRICEABLE OBJECT.
«PK»	***id***	*SeriesConstraintIdType*	1:1	Identifier of SERIES CONSTRAINT.
	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.
	Priority	*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).
	SeriesType	*SeriesTypeEnum*	0:1	Classification of SERIES CONSTRAINT. Default is station to station. See allowed values below.
	RoutingType	*RoutingTypeEnum*	0:1	Whether this is a direct i.e. no changes required point to point or indirect. 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-JourneyPattern***	*FarePointInJourneyPattern*	0:*	FARE POINTs IN PATTERN in SERIES CONSTRAINT.
"cntd»	***JourneyPatterns***	*JourneyPattern*	0:*	JOURNEY PATTERN or patterns. equivalent to the SERIES CONSTRAINT.
"cntd»	***replaces***	*SeriesConstraintRef*	0:*	Replaces the specified SERIES. (Needed for TAP TSI)

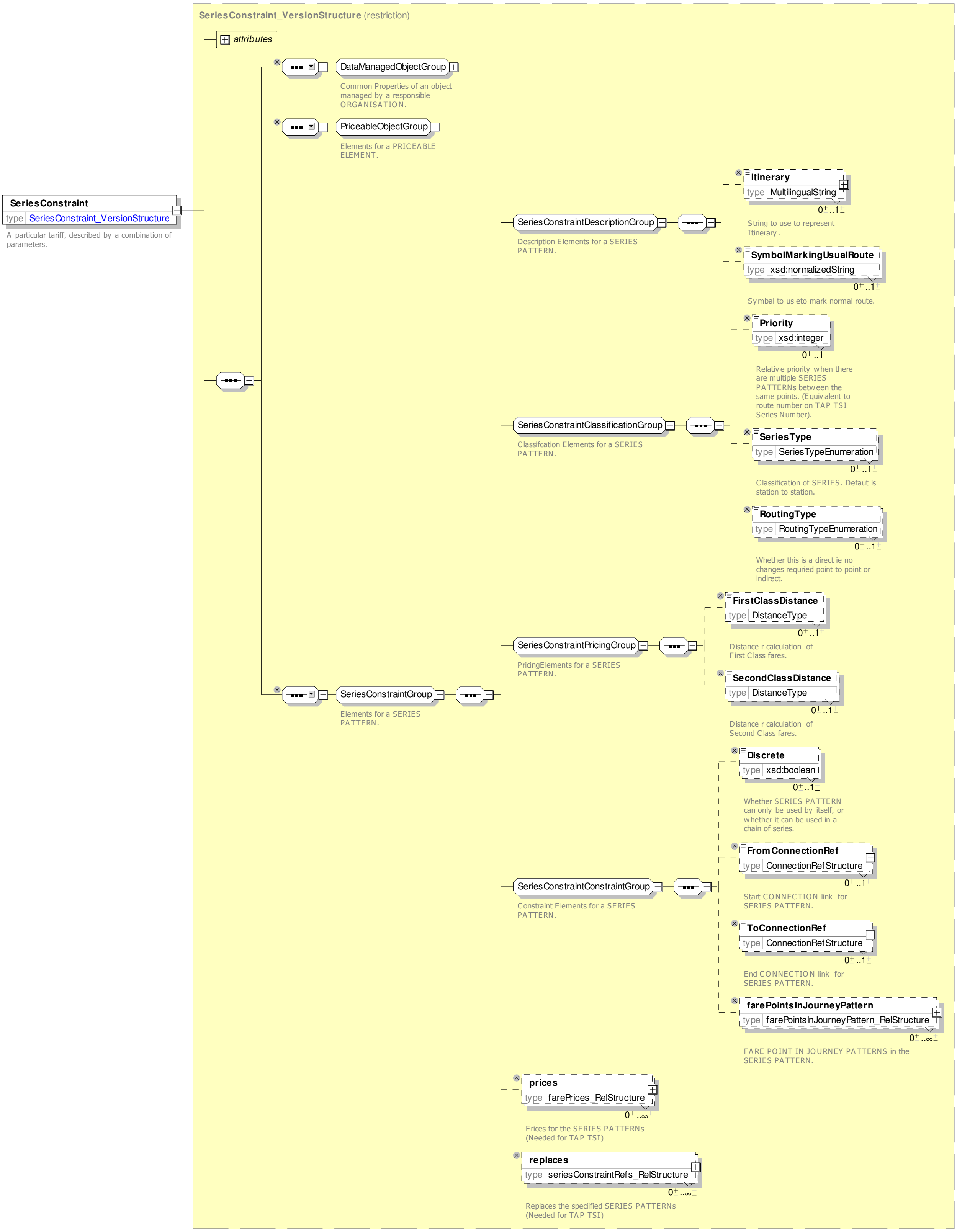


Figure 47 — *SeriesConstraint* — XSD

8.4.1.4.5.1 SeriesType – Allowed values

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

Table 26 – *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.

8.4.1.4.5.2 RoutingType – Allowed values

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

Table 27 – *RoutingType* – Allowed values

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

8.4.1.4.6 FarePointInJourneyPattern – Model Element

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

Table 28 – *FarePointInJourneyPattern* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PointInJourneyPattern*	::>	FARE POINT IN JOURNEY PATTERN inherits from POINT IN JOURNEY PATTERN. See NeTEx Part1.
«PK»	***id***	*FaresPointInPattern-IdType*	1:1	Identifier of a FARE POINT IN JOURNEY 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 position for showing this FARE POINT IN JOURNEY PATTERN in itinerary when there is a choice according to rail conventions. For example, *(A /B) * C* versus *(B/A) * C*.
	Presentation-Position	*SeriesPresentationEnum*	0:1	Relative position for showing this FARE POINT IN JOURNEY PATTERN in itinerary when there is a choice according to rail conventions. For example, *(A /B) * C* versus *(B/A) * C*. See allowed values below.

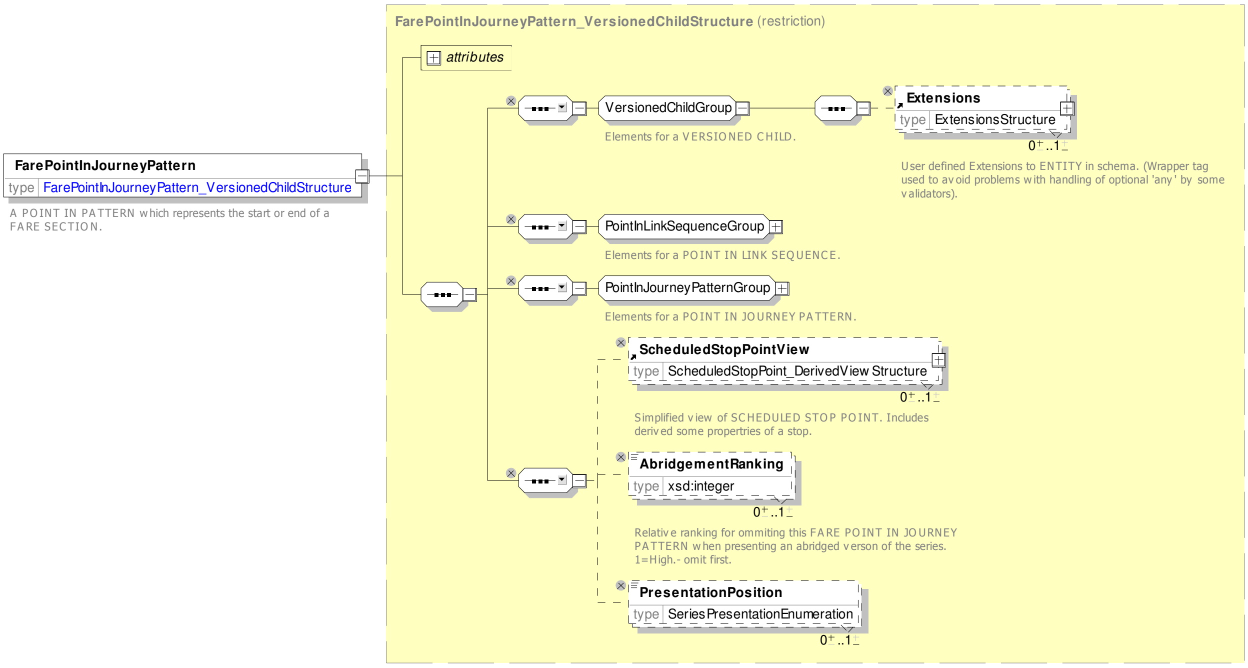


Figure 48 — *FarePointInJourneyPattern* — XSD

8.4.1.4.6.1 PresentationPosition – Allowed values

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

Table 29 – *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.
optionalShowLeft	In an itinerary, show as an optional station - first choice on left.
optionalShowRight	In an itinerary, show as an optional station - second choice on right.

8.4.1.4.7 SeriesConstraintPrice – Model Element

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

Table 30 – *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»	***GroupOf-SeriesConstraint sRef***	*GroupOf-SeriesConstraintsRef*	0:1	GROUP OF SERIES CONSTRAINT for which this is the price.

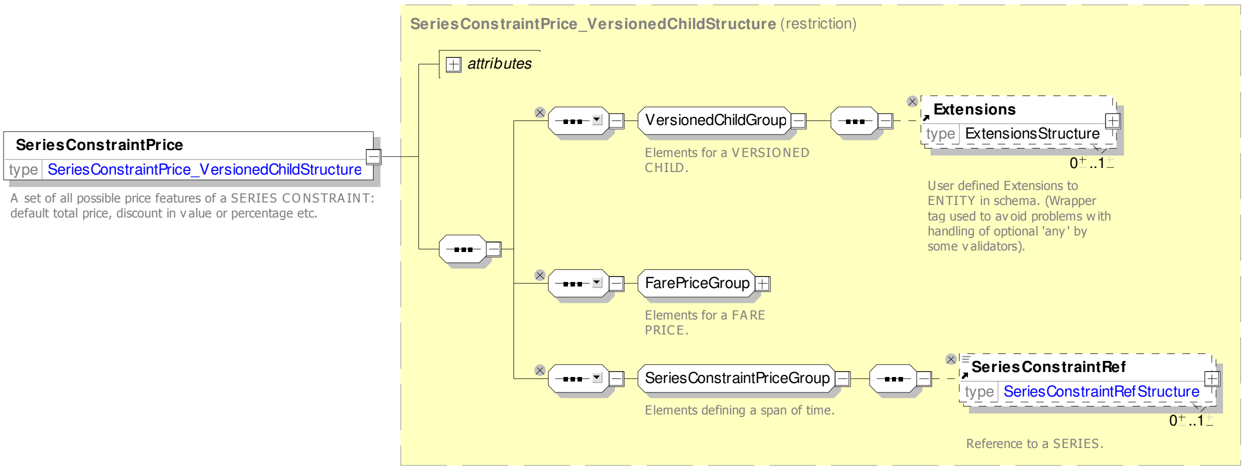


Figure 49 — *SeriesConstraintPrice* — XSD

8.4.1.5 Fare Zone – XML examples

8.4.1.5.1 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=":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>
```

```xml
    <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>
```

8.4.1.5.2 Fare Zone: XML Example of Border Points

The following code fragment shows two BORDER POINT definitions.

For EXAMPLE:

```xml
<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:

```xml
<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>
```

8.4.1.5.3 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:

```xml
<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>
                <FarePointInJourneyPattern id="tap::55501" version="01" order="1">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap::16043"/>
                        <Name>Erfurt</Name>
                    </ScheduledStopPointView>
                    <AbridgementRanking>2</AbridgementRanking>
                    <PresentationPosition>optionalLeft</PresentationPosition>
                </FarePointInJourneyPattern>
                <FarePointInJourneyPattern id="tap::55502" version="01" order="2">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap::26002"/>
                        <Name>Hof</Name>
                    </ScheduledStopPointView>
                    <AbridgementRanking>2</AbridgementRanking>
                    <PresentationPosition>optionalRight</PresentationPosition>
                </FarePointInJourneyPattern>
                <FarePointInJourneyPattern id="tap::55503" version="01" order="3">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap::23002"/>
                        <Name>Halle</Name>
                    </ScheduledStopPointView>
                    <PresentationPosition>required</PresentationPosition>
                </FarePointInJourneyPattern>
                <FarePointInJourneyPattern id="tap::55504" version="01" order="4">
                    <ScheduledStopPointView>
                        <ScheduledStopPointRef ref="tap::27359"/>
                        <Name>Schwerin</Name>
                    </ScheduledStopPointView>
                    <AbridgementRanking>2</AbridgementRanking>
                    <PresentationPosition>required</PresentationPosition>
                </FarePointInJourneyPattern>
            </farePointsInPattern>
        </SeriesConstraint>
    </seriesConstraints>
</DistanceMatrixElement>
```

8.4.2 Fare Facility

8.4.2.1 Fare Facility – Conceptual model

[TO DO DO WE NEED THIS?]

Figure 50 — Fare Facility – Conceptual Model

8.4.2.2 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 ACCOMODATION elements that can be associated with access rights.

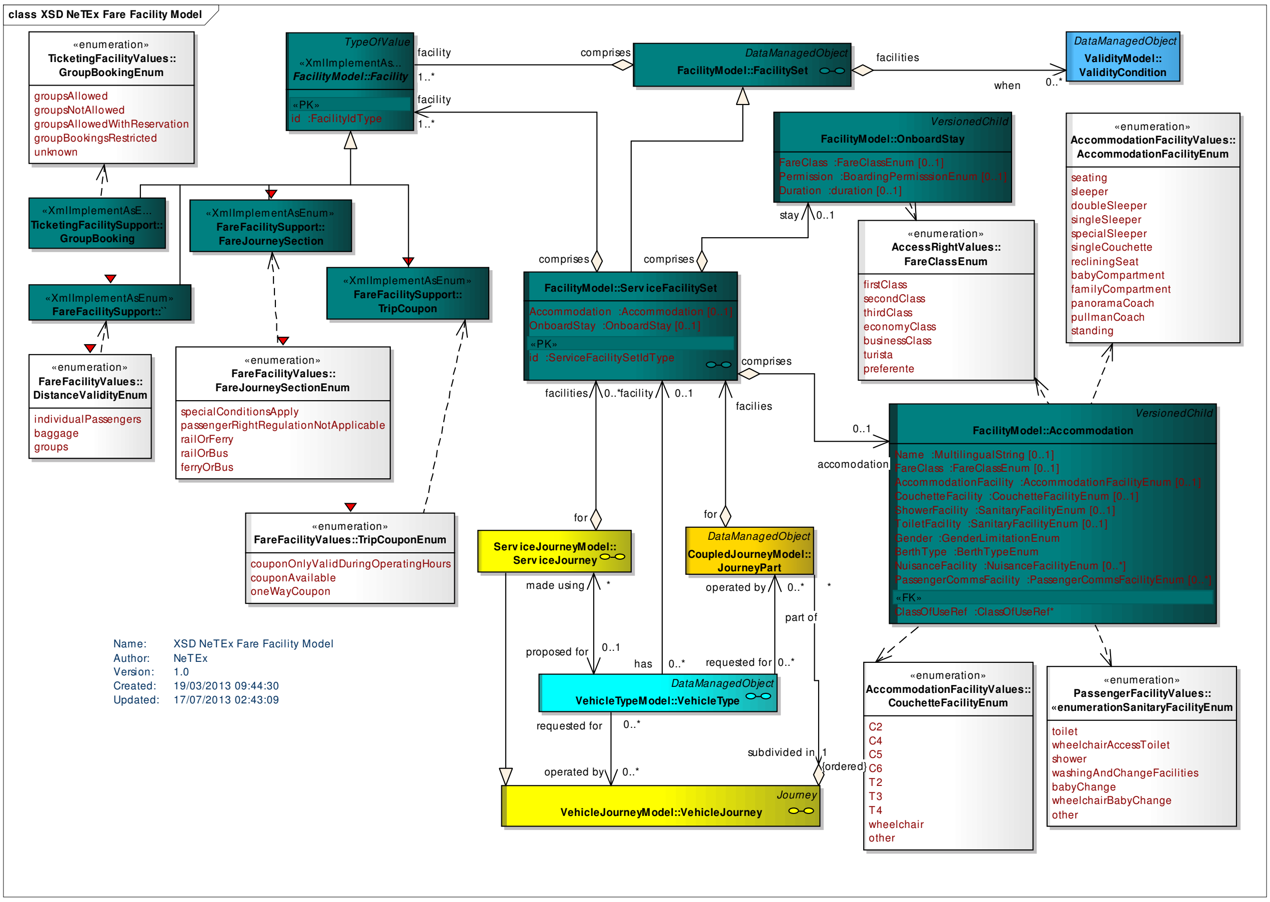


Figure 51 — Fare Facility– Physical Model

8.4.2.3 Fare Facility – Attributes and XSD

8.4.2.3.1.1 DistanceValidity – Allowed values

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

Table 31 – *DistanceValidity* – Allowed values

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

8.4.2.3.1.2 FareJourneySection – Allowed values

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

Table 32 – *FareJourneySection* – Allowed values

Value	Description
specialConditionsApply	Special conditions made available by Railway Undertaking.
passengerRightRegulationNotApplicable	Journey section for which the PPR 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.

8.4.2.3.1.3 TripCoupon – Allowed values

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

Table 33 – *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.

8.5 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.

8.5.1 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.

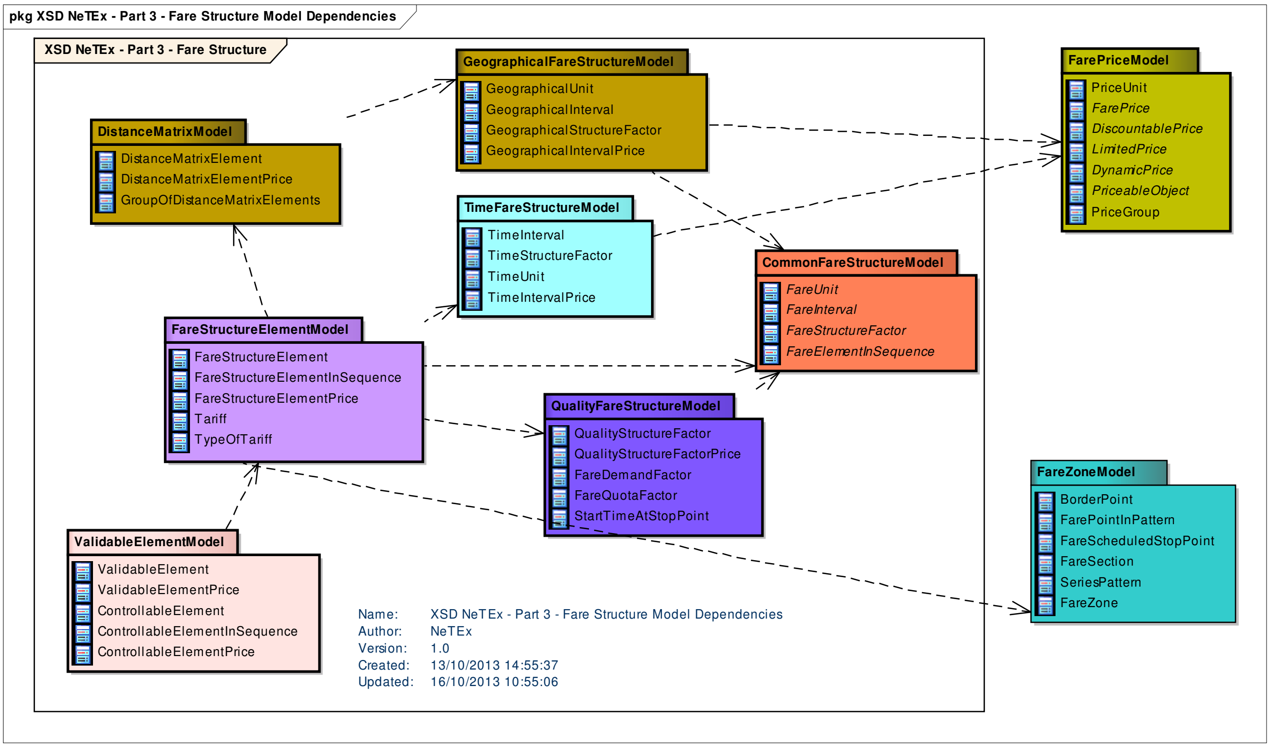


Figure 52 — Fare Structure Model Dependencies

8.5.2 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.

8.5.2.1 Common Fare Structure: Fare Structure Factors – Conceptual model

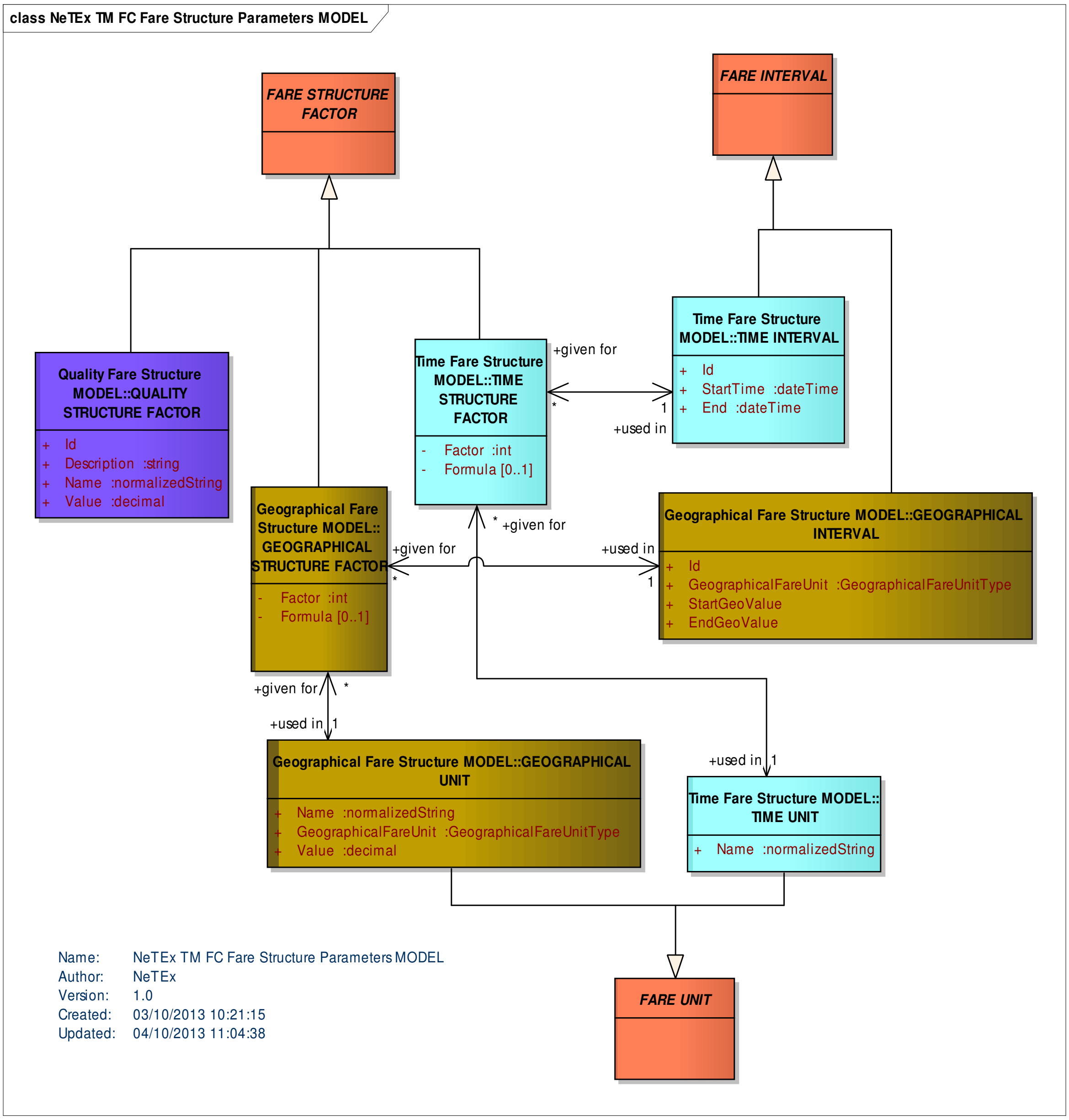


Figure 53 — Fare Structure Parameters – Conceptual Model

8.5.2.2 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.

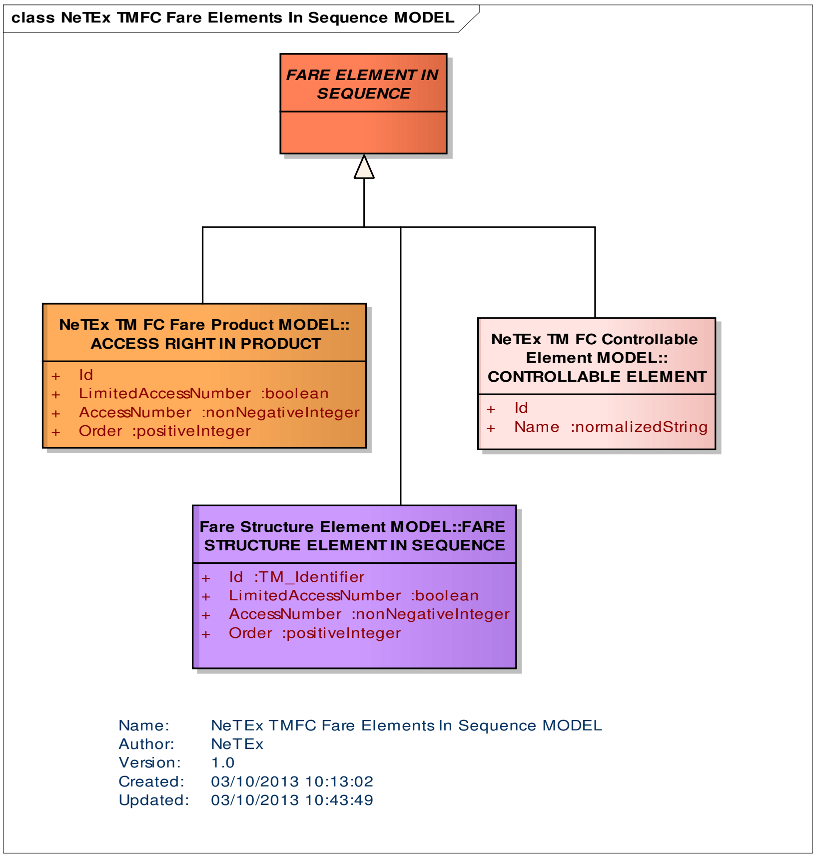


Figure 54 — Fare Elements In Sequence – Conceptual Model

8.5.2.3 Common Fare Structure – Physical model

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

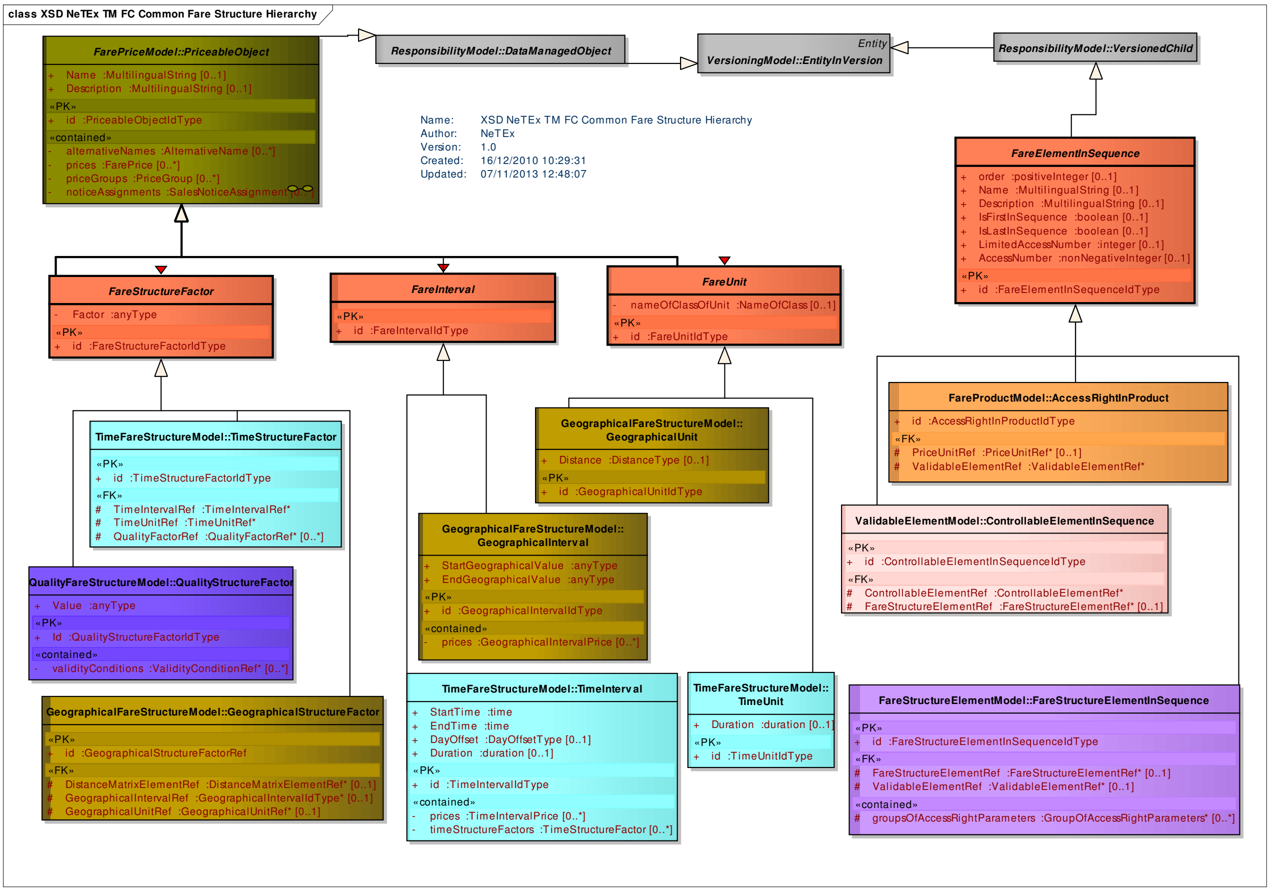


Figure 55 — Common Fare Structure Model – Physical Model

8.5.2.4 Common Fare Structure – Attributes and XSD

8.5.2.4.1 FareUnit – Model Element

A unit associated with a FARE STRUCTURE FACTOR.

Table 34 – *FareUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	FARE UNIT inherits from PRICEABLE OBJEC.
«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.

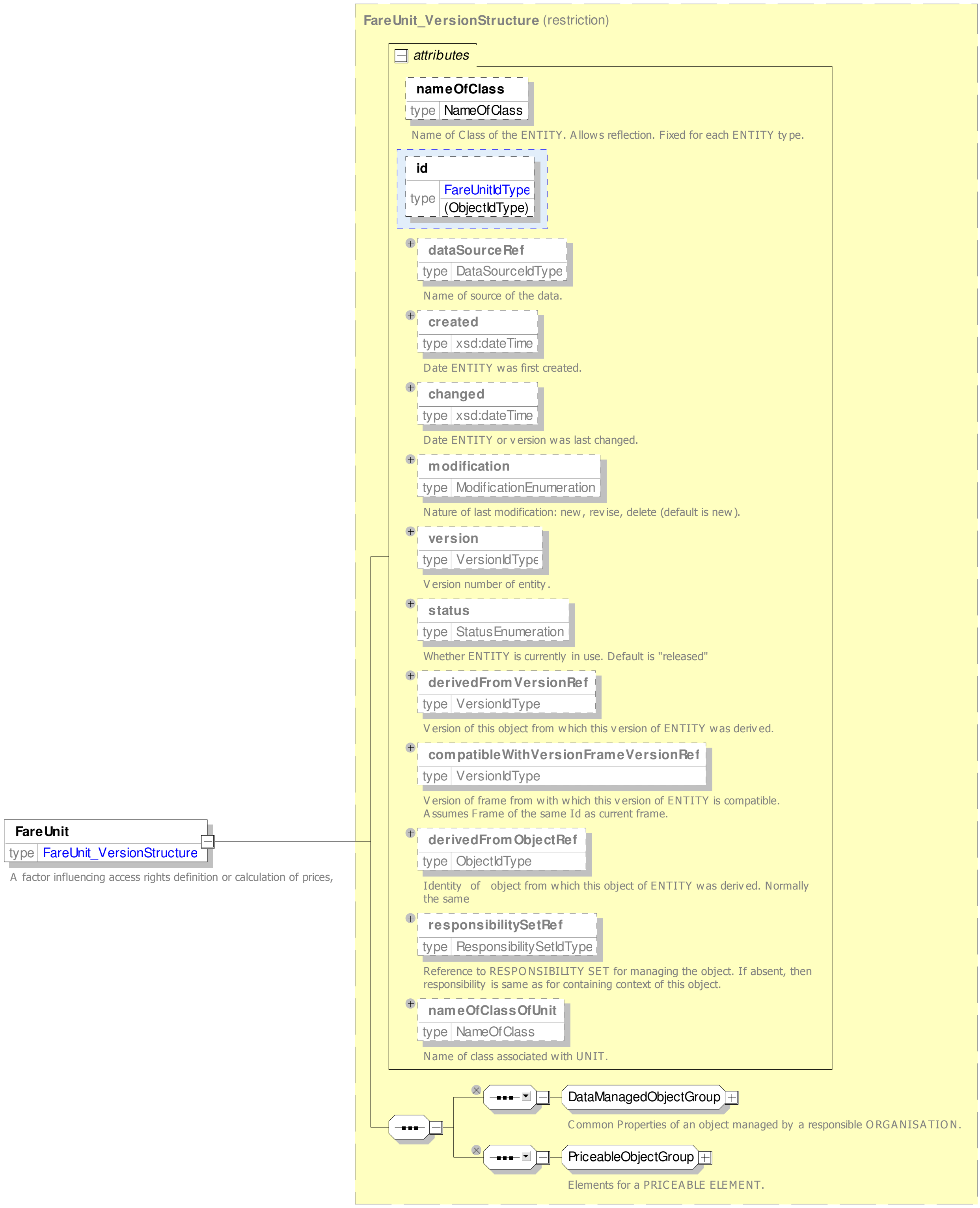


Figure 56 — *FareUnit* — XSD

8.5.2.4.2 FareInterval – Model Element

An interval based aspect of the fare structure.

Table 35 – *FareInterval* – Element

Classifi-	Name	Type	Cardinality	Description

cation				
::>	::>	*PriceableObject*	::>	FARE INTERVAL inherits from PRICEABLE OBJECT.
«PK»	***id***	*FareIntervalIdType*	1:1	Identifier of FARE INTERVAL.

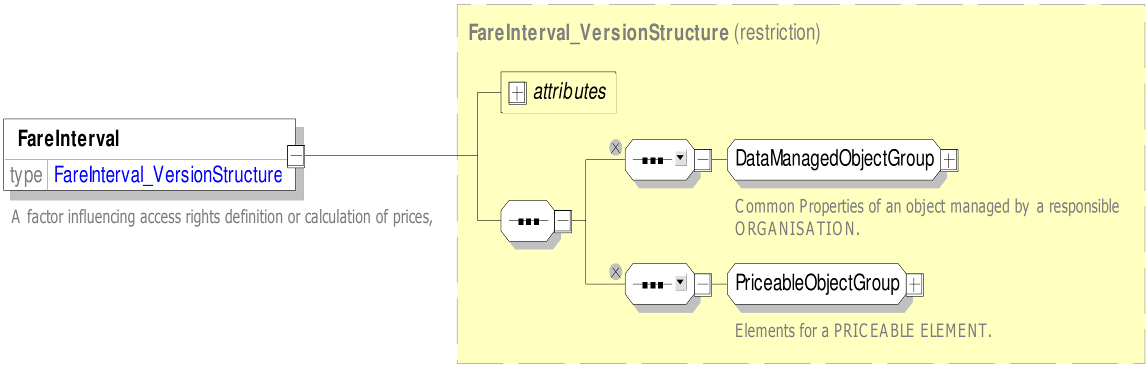


Figure 57 — *FareInterval* — XSD

8.5.2.4.3 FareStructureFactor – Model Element

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

Table 36 – *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.
	Factor	*xsd:anyType*	1:0	Arbitrary values associated with factor.

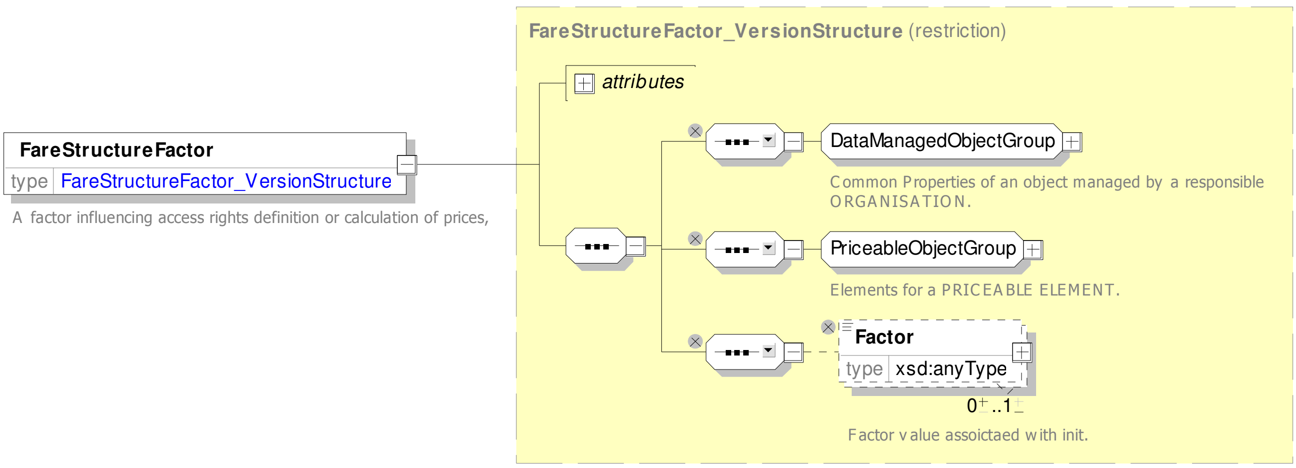


Figure 58 — *FareStructureFactor* — XSD

8.5.2.4.4 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 37 – *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.
	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.
	IsFirstIn-Sequence	*xsd:boolean*	0:1	Whether element is the first in the sequence.
	IsLastIn-Sequence	*xsd:boolean*	0:1	Whether element is the last in the sequence.
	Access-NumberIsLimited	*xsd:boolean*	0:1	Whether access number is limited.
	AccessNumber	*xsd:nonNegativeInteger*	0:1	Access number in sequence.

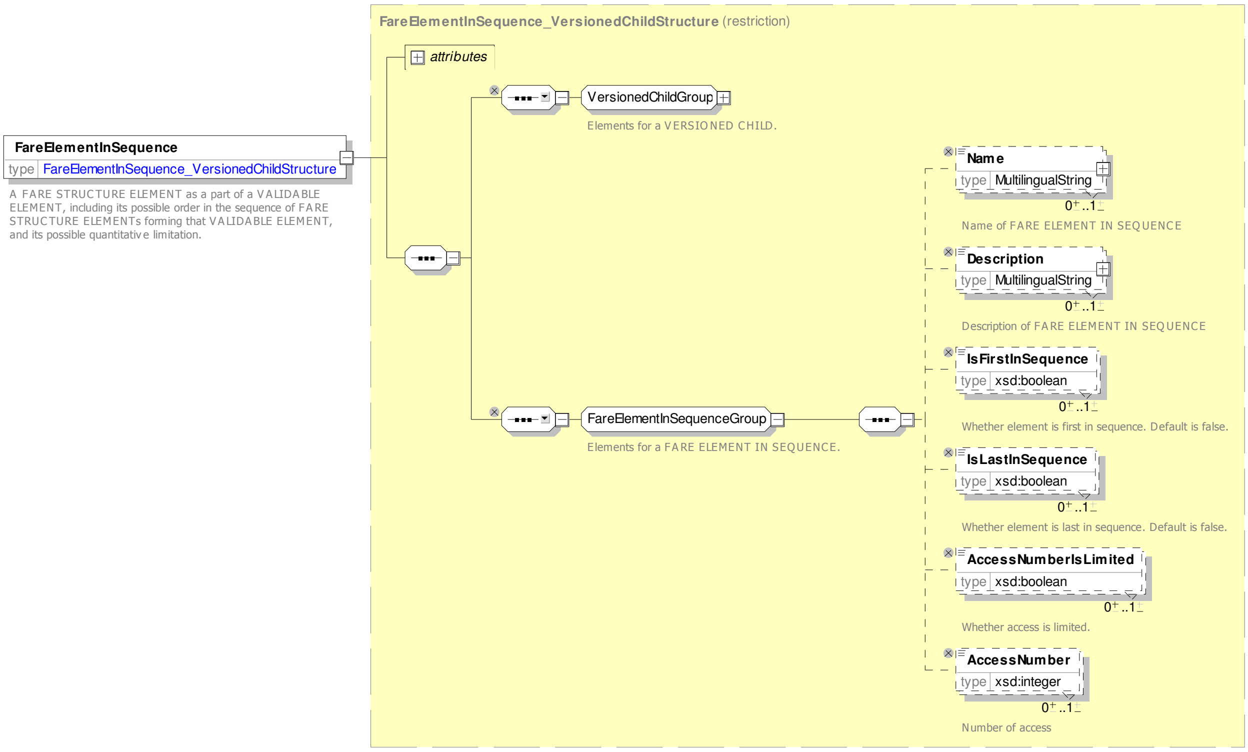


Figure 59 — *FareElementInSequence* — XSD

8.5.3 Geographical Fare Structure

8.5.3.1 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.

8.5.3.1.1 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 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 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 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 STOP POINTs.

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

8.5.3.1.2 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.

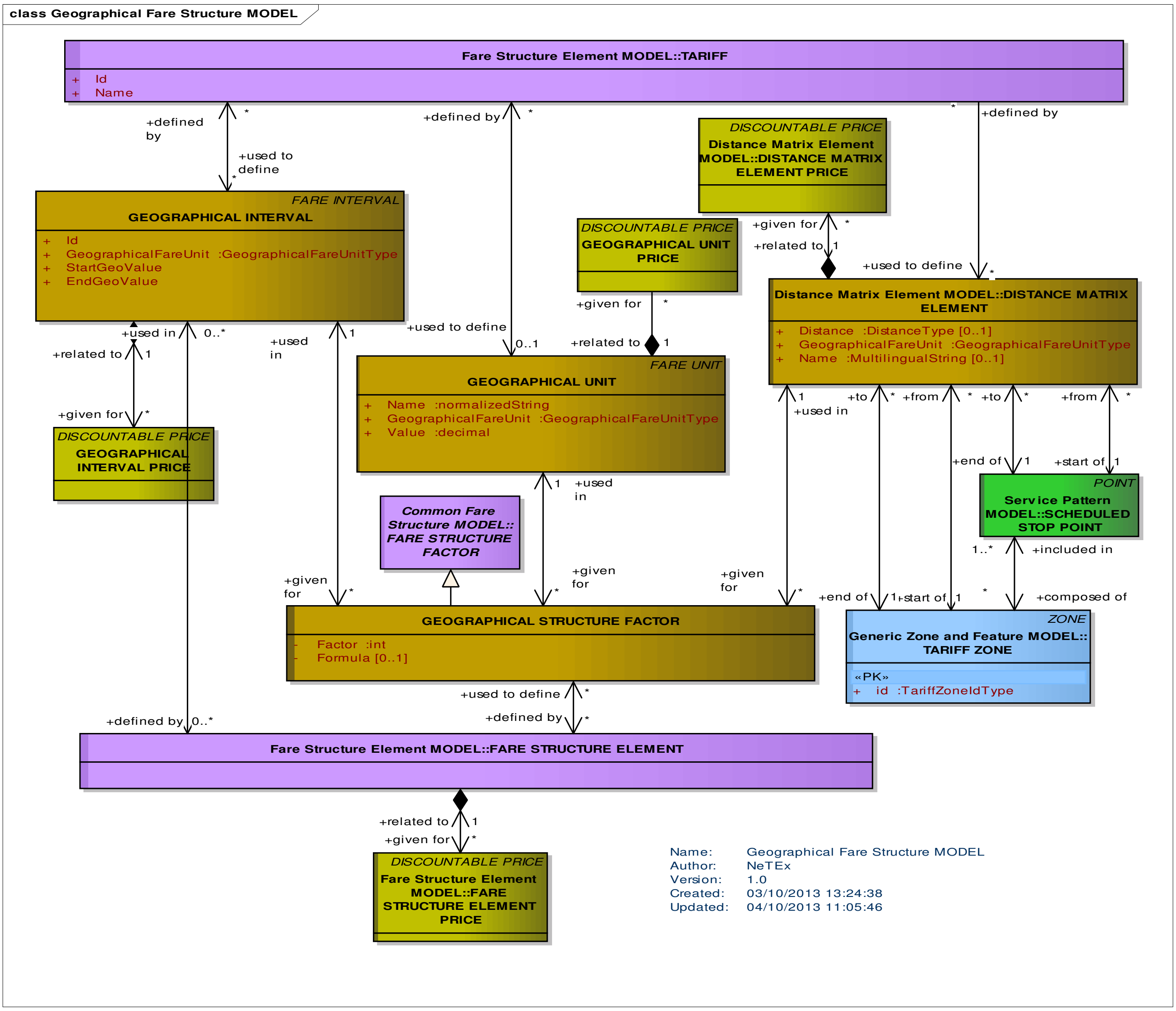


Figure 60 — Geographical Fare Structure – Conceptual Model

8.5.3.2 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.

[TO DO ADD Examples]

8.5.3.3 Geographical Fare Structure – Physical model

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

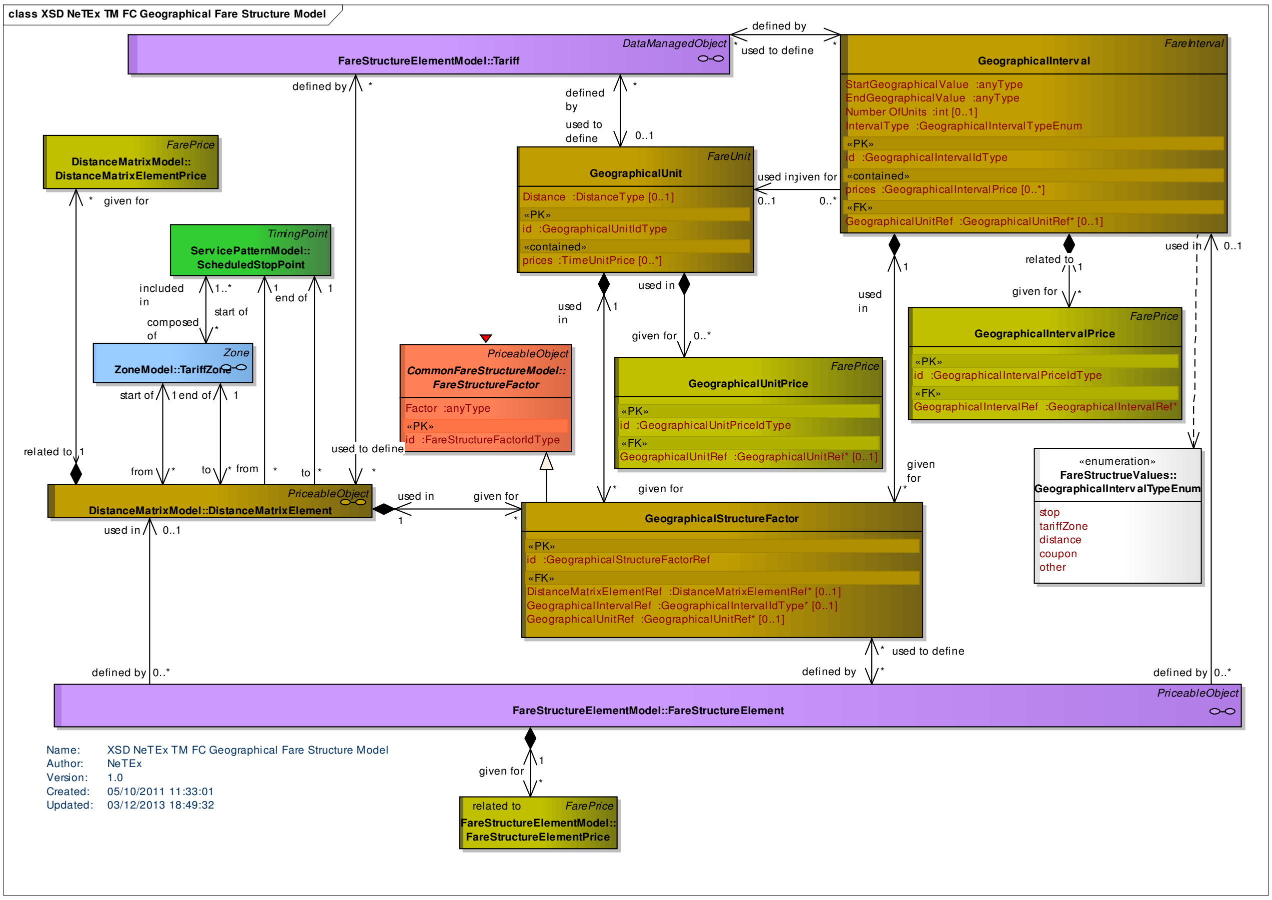


Figure 61 — Geographical Fare Structure – Physical Model

8.5.3.4 Geographical Fare Structure – Attributes and XSD

8.5.3.4.1 GeographicalUnit – Model Element

A unit for calculating geographical graduated fares.

Table 38 – *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.
	prices	*GeographicalUnitPrice*	0:*	Prices associated with GEOGRAPHICAL UNIT

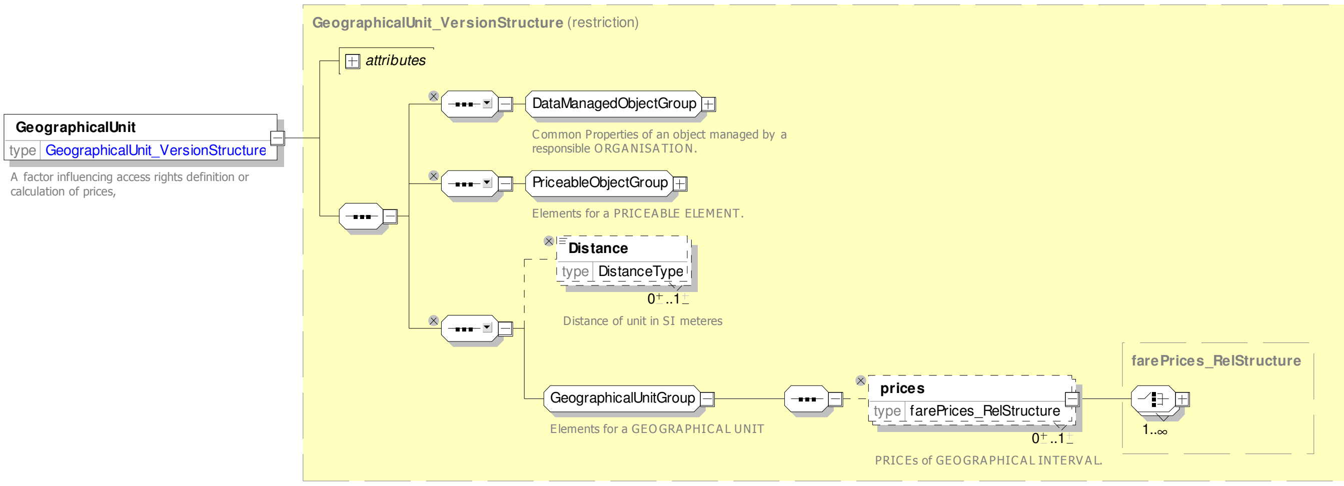


Figure 62 — *GeographicalUnit* — XSD

8.5.3.4.2 GeographicalUnitPrice – Model Element

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

Table 39 – *GeographicalUnitPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	GEOGRAPHICAL UNIT PRICE inherits from FARE PRICE
«PK»	***id***	*GeographicalUnitPrice IdType*	1:1	Identifier of GEOGRAPHICAL UNIT PRICE.
«FK»	***Geographical-UnitRef***	*GeographicalUnitRef*	1:1	Reference to GEOGRAPHICAL UNIT for which this is the price.

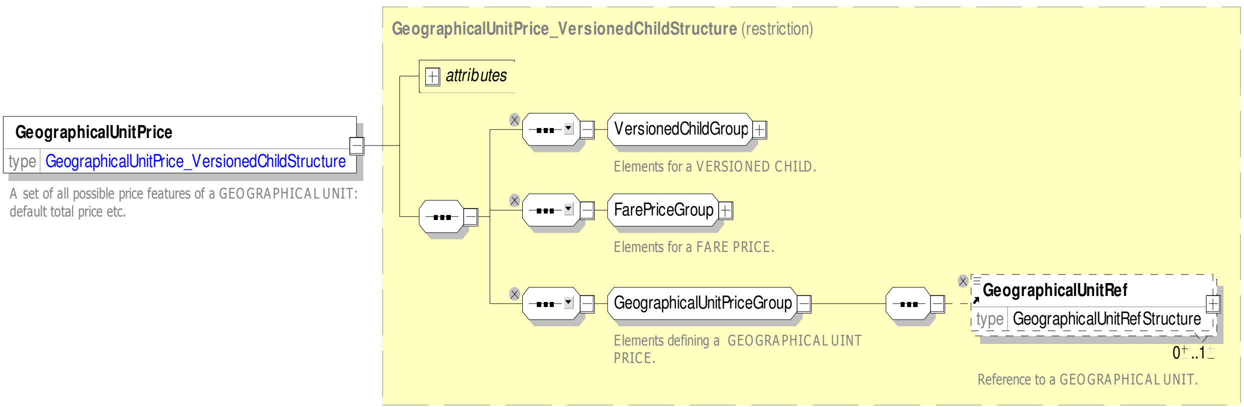


Figure 63 — *GeographicalUnitPrice* — XSD

8.5.3.4.3 GeographicalInterval – Model Element

A geographical interval specifying access rights for the FARE STRUCTURE ELEMENTs within the range of this interval: "*20-5 km", "4-6 zones*", etc.

Table 40 – *GeographicalInterval* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareInterval*	::>	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.
	IntervalType	*IntervalkTypeEnum*	0:1	Classification of interval type. See allowed values below.
«FK»	***Geographical-UnitRef***	*GeographicalUnitRef*	0:1	GEOGRAPHICAL UNIT for interval.
"cntd»	***prices***	*Geographical-IntervalPrice*	0:*	Prices for the GEOGRAPHIC INTERVAL.

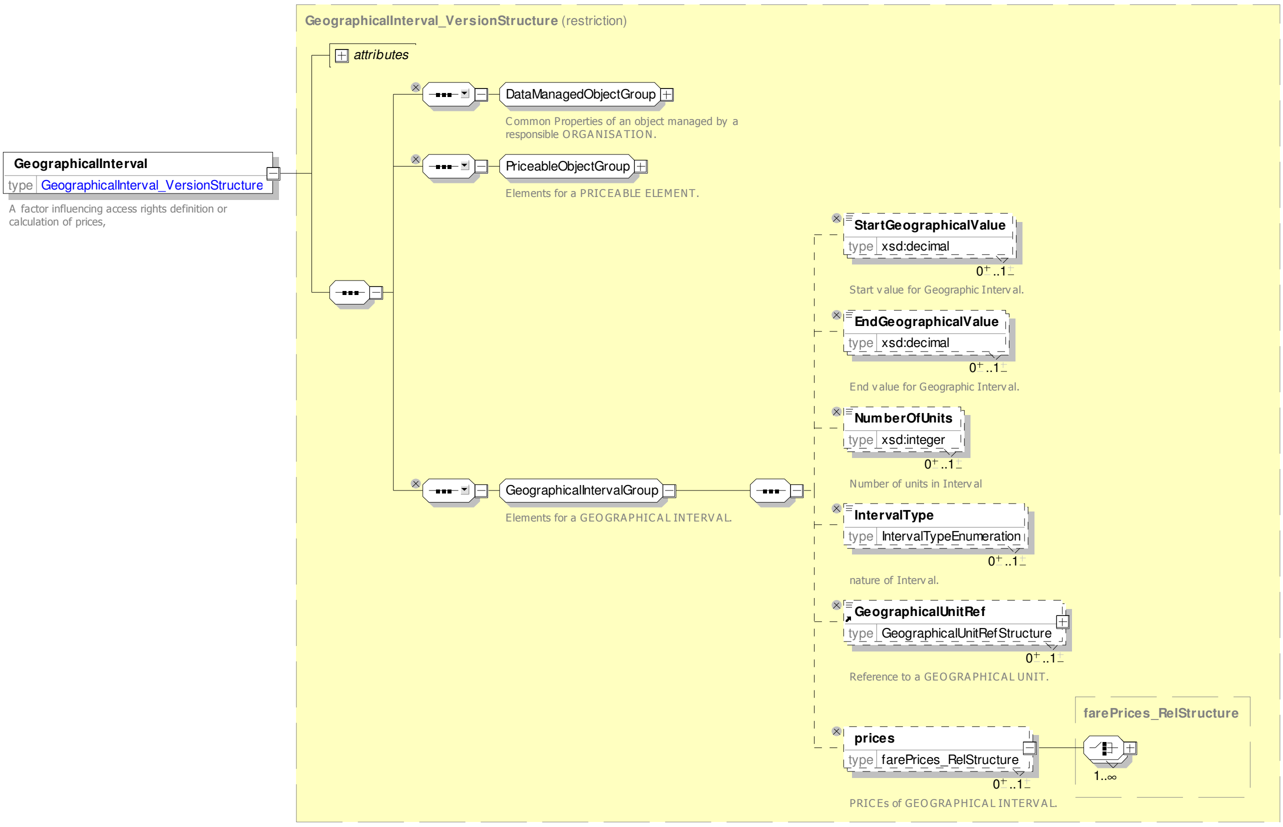


Figure 64 — *GeographicalInterval* — XSD

8.5.3.4.3.1 IntervalType – Allowed values

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

Table 41 – *IntervalType* – Allowed values

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

8.5.3.4.4 GeographicalIntervalPrice – Model Element

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

Table 42 – *GeographicalIntervalPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description

::>	::>	*FarePrice*	::>	GEOGRAPHICAL INTERVAL PRICE inherits from FARE PRICE
«PK»	***id***	*GeographicalIntervalPrice IdType*	1:1	Identifier of GEOGRAPHICAL INTERVAL PRICE.
«FK»	***Geographical-IntervalRef***	*GeographicalIntervalRef*	1:1	Reference to GEOGRAPHICAL INTERVAL for which this is the price.

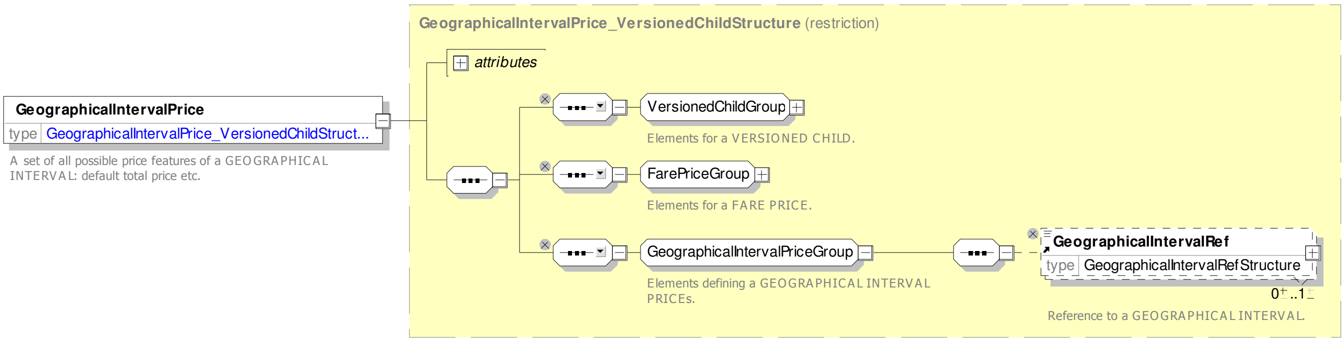


Figure 65 — *GeographicalIntervalPrice* — XSD

8.5.3.4.5 GeographicalStructureFactor – Model Element

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

Table 43 – *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 Factor

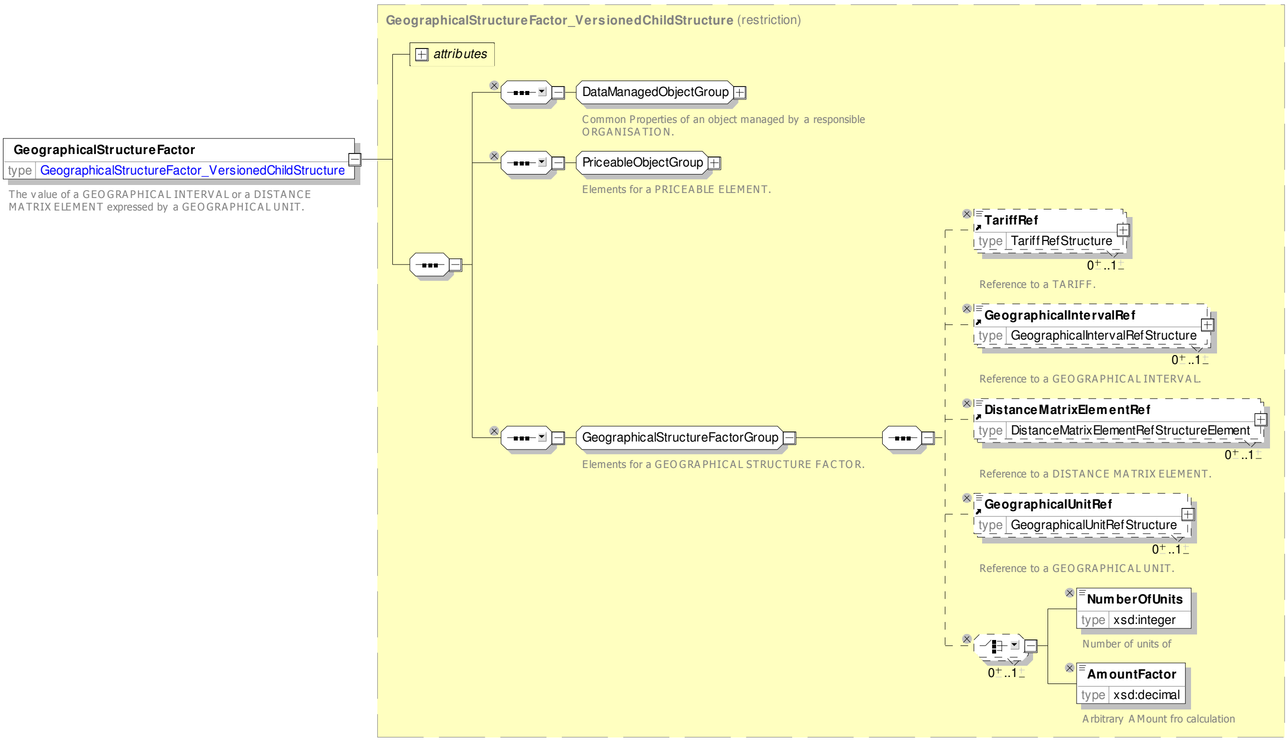


Figure 66 — *GeographicalStructureFactor* — XSD

8.5.3.5 Geographical Fare Structure – XML examples

8.5.3.5.1 Geographical Fare Structure: XML Example of stepped geographical intervals

The following code fragment shows a 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; 50o 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::upTo50Kilometre" version="any">
            <Name>Price per Kilometre up to 50 kilometres</Name>
            <StartGeographicalValue>1</StartGeographicalValue>
            <EndGeographicalValue>100</EndGeographicalValue>
            <prices>
                <GeographicalIntervalPrice id="tap::upTo50Kilometer" version="any">
                    <Amount>1.30</Amount>
                </GeographicalIntervalPrice>
            </prices>
        </GeographicalInterval>

        <GeographicalInterval id="tap::50To100Kilometre" version="any">
            <Name>Price per Kilometre 50 to 150 Kilometres </Name>
            <StartGeographicalValue>51</StartGeographicalValue>
            <EndGeographicalValue>150</EndGeographicalValue>
            <prices>
                <GeographicalIntervalPrice id="tap::50To100Kilometer" version="any">
                    <Amount>1.00</Amount>
                </GeographicalIntervalPrice>
            </prices>
```

```xml
        </GeographicalInterval>

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

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

8.5.4 Time Fare Structure

8.5.4.1 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.

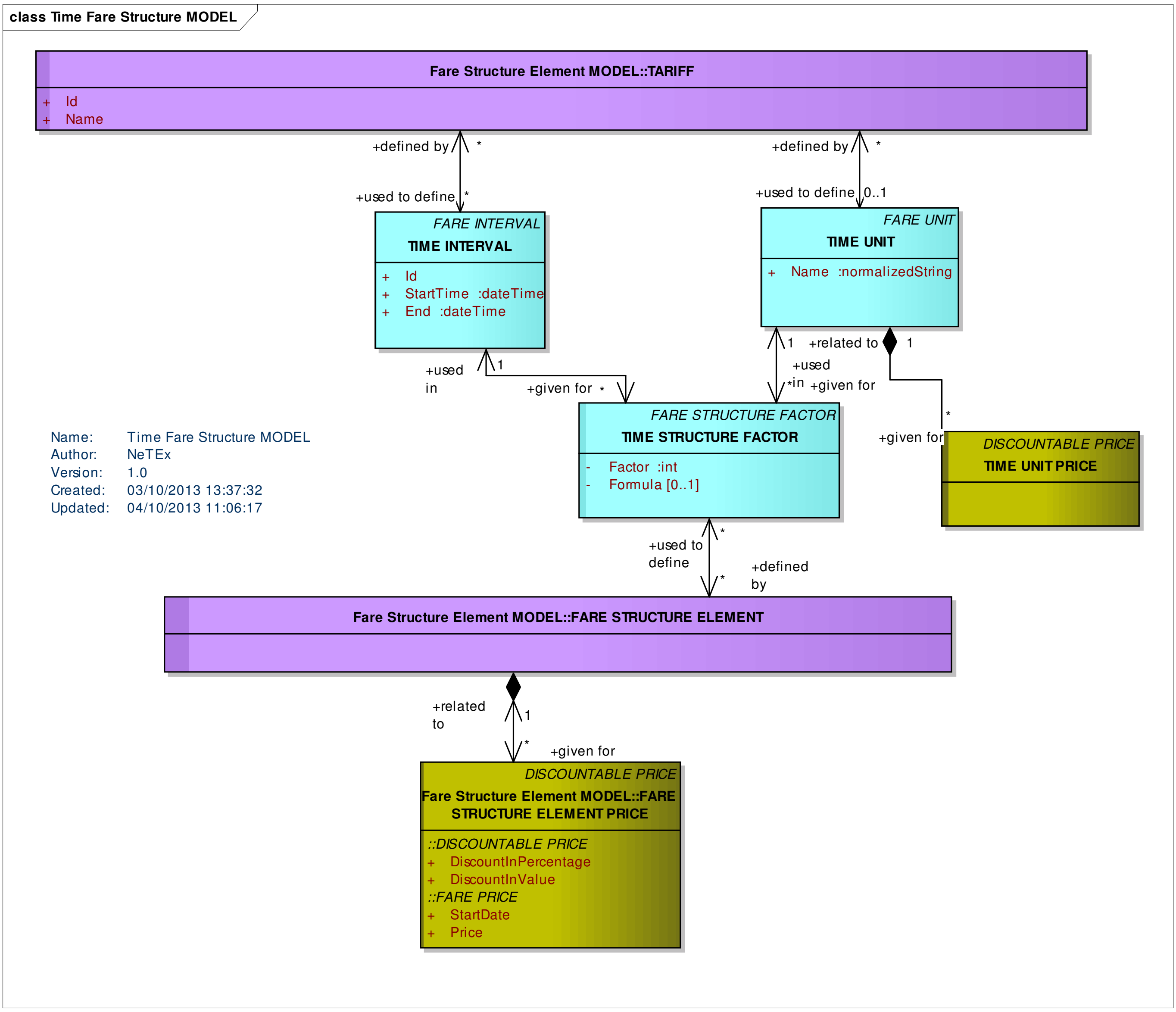


Figure 67 — Time Fare Structure – Conceptual Model

8.5.4.2 Time Fare Structure – Physical model

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

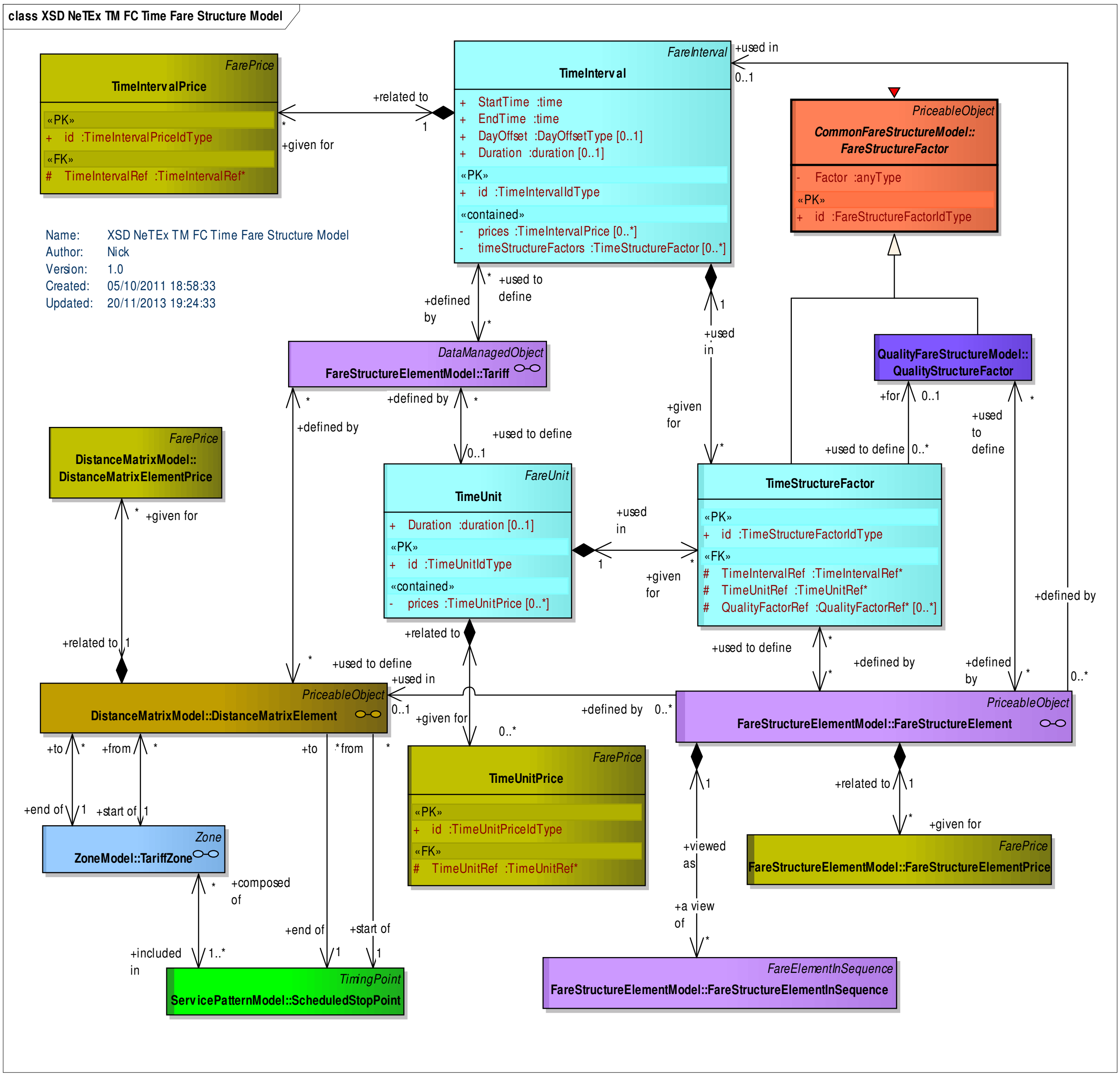


Figure 68 — Time Fare Structure – Physical Model

8.5.4.3 Time Fare Structure – Conceptual Examples

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

[TO DO ADD CONCEPT EXAMPLES]

8.5.4.4 Time Fare Structure – Attributes and XSD

8.5.4.4.1 TimeUnit – Model Element

A unit for calculating time-based graduated fares.

Table 44 – *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

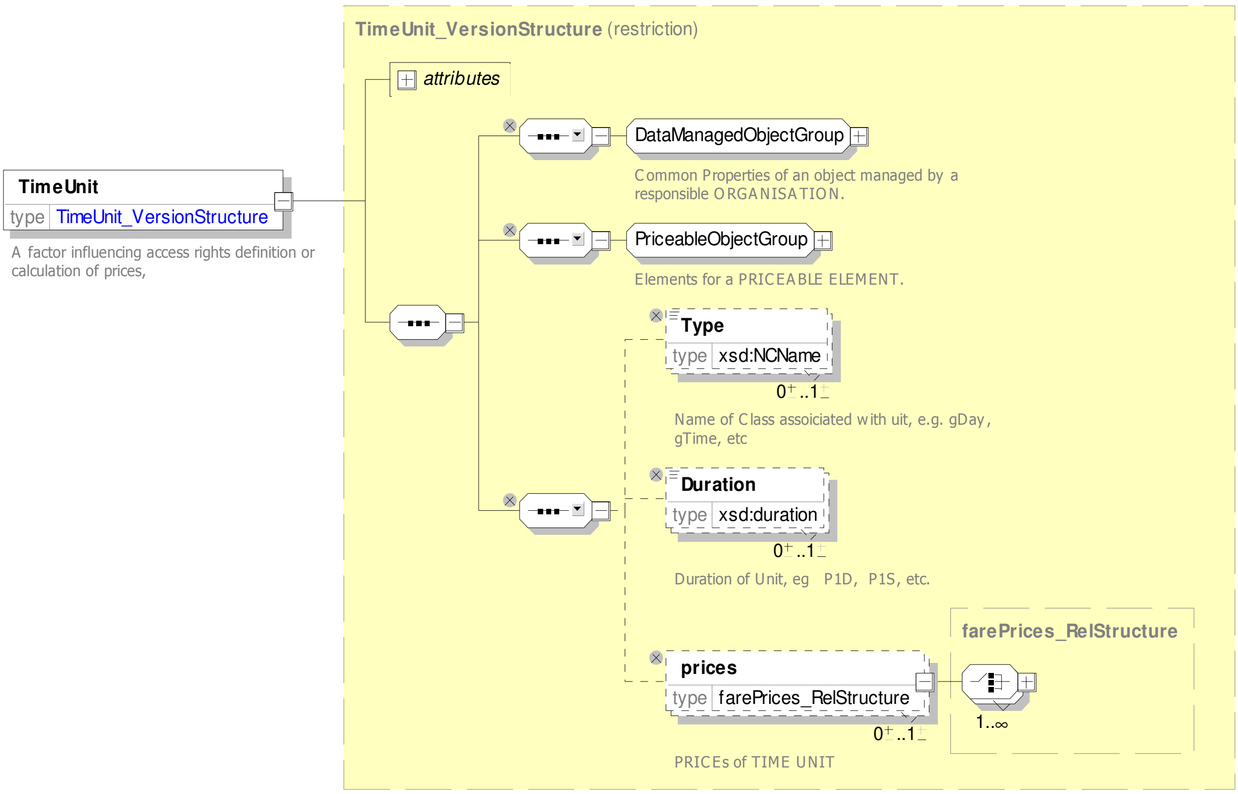


Figure 69 — *TimeUnit* — XSD

8.5.4.4.2 TimeUnitPrice – Model Element

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

Table 45 – *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*	1:1	Reference to TIME UNIT for which this is the price.

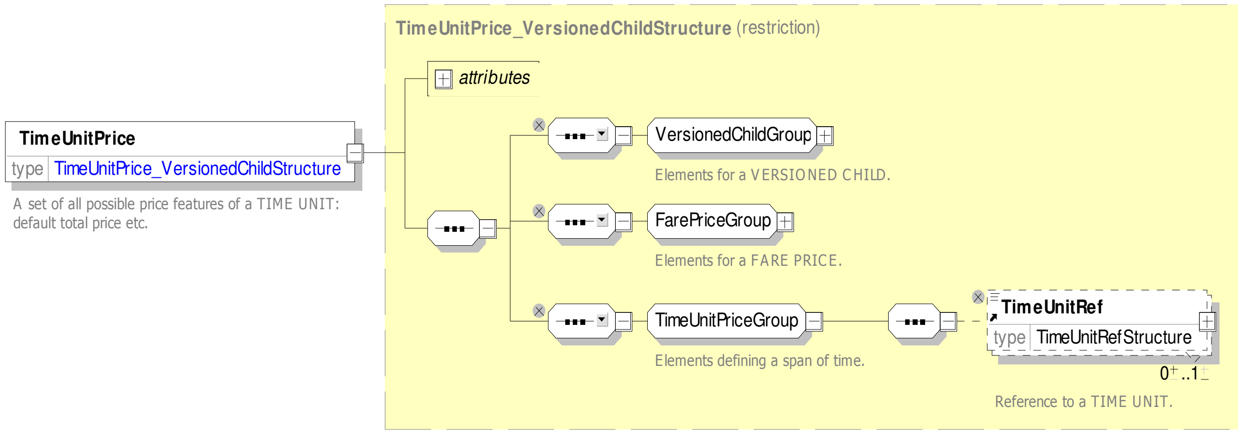


Figure 70 — *TimeUnitPrice* — XSD

8.5.4.4.3 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 46 – *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*	1:1	Start of TIME INTERVAL.
	EndTime	*xsd:time*	1:1	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
"cntd»	***prices***	*TimeIntervalPrice*	0:*	Prices for the TIME INTERVAL.
"cntd»	***timeStructure-Factors***	*TimeStructureFactor*	0:*	TIME STRUCTIURE FACTORs using the TIME INTERVAL.

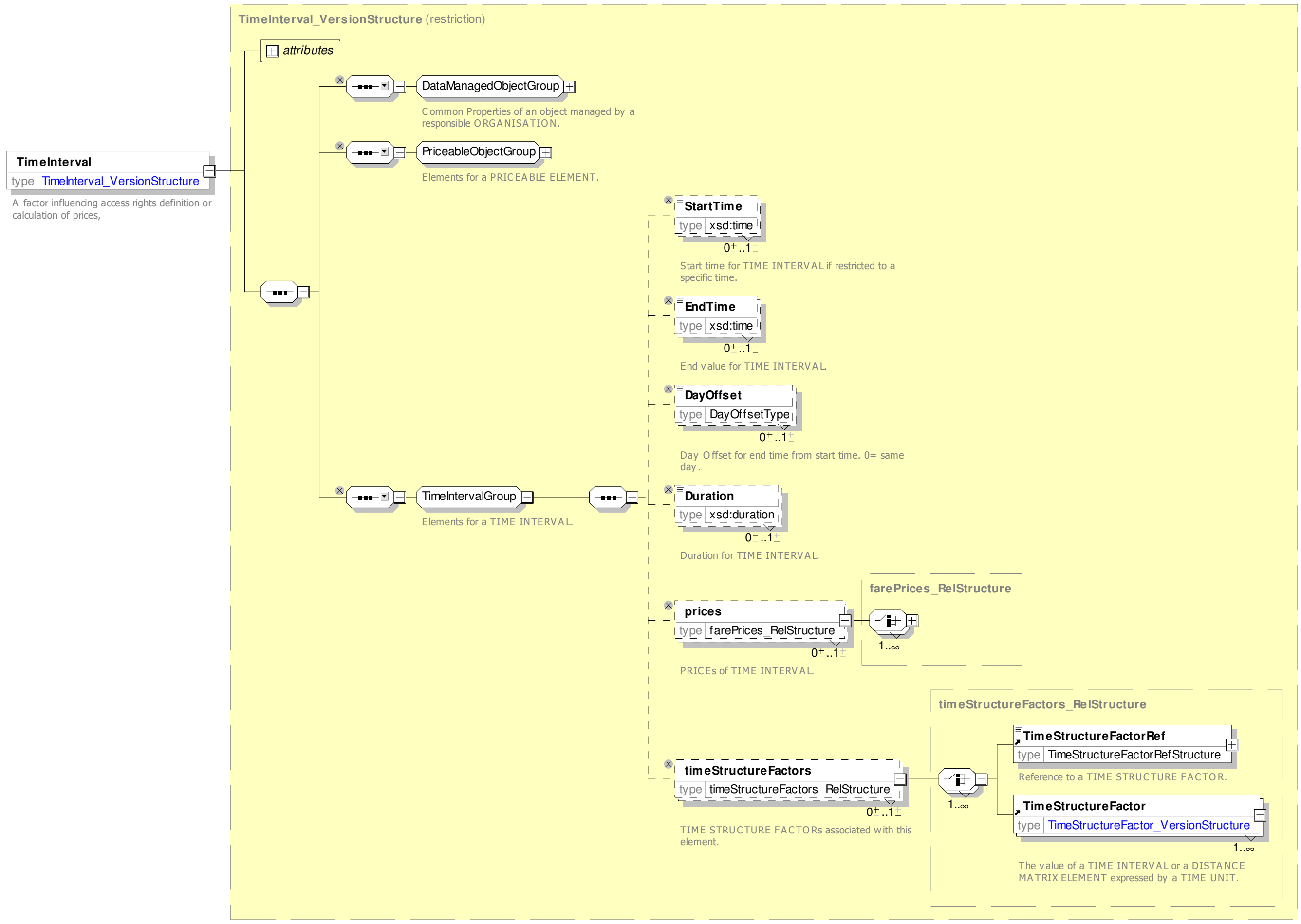


Figure 71 — *TimeInterval* — XSD

8.5.4.4.4 TimeIntervalPrice – Model Element

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

Table 47 – *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*	1:1	Reference to TIME INTERVAL for which this is the price.

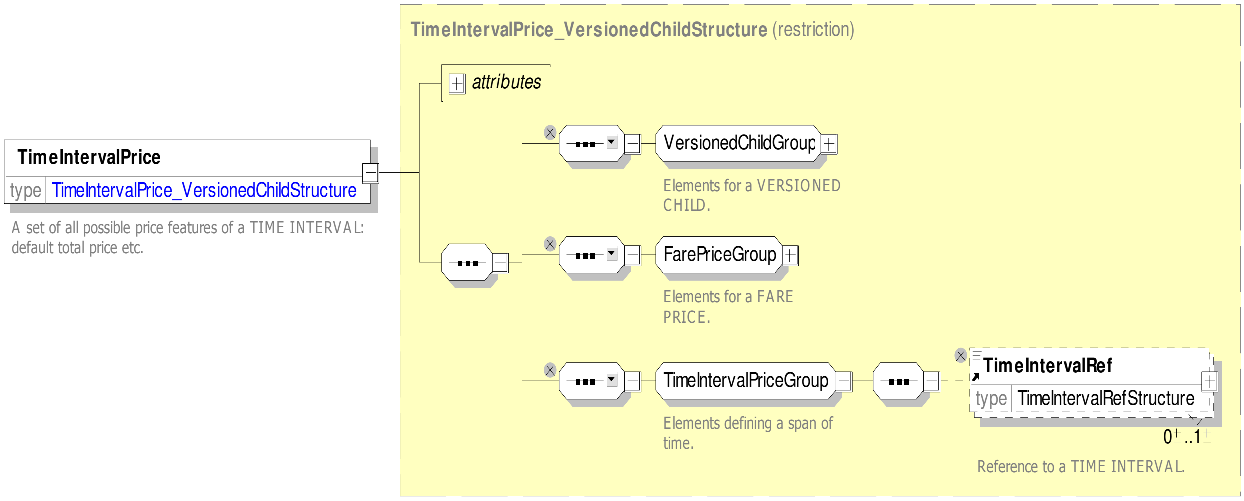


Figure 72 — *TimeIntervalPrice* — XSD

8.5.4.4.5 TimeStructureFactor – Model Element

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

Table 48 – *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»	***TimeIntervalRef***	*TimeIntervalRef*	1:1	Reference to TIME INTERVAL associated with factor.
«FK»	***TimeUnitRef***	*TimeUnitRef*	1:1	Reference to TIME UNIT associated with factor.
«FK»	***QualityStructure-FactorRef***	*QualityStructure-FactorRef*	0:*	QUALITY FACTOR associated with the TIME STRUCTURE FACTOR.

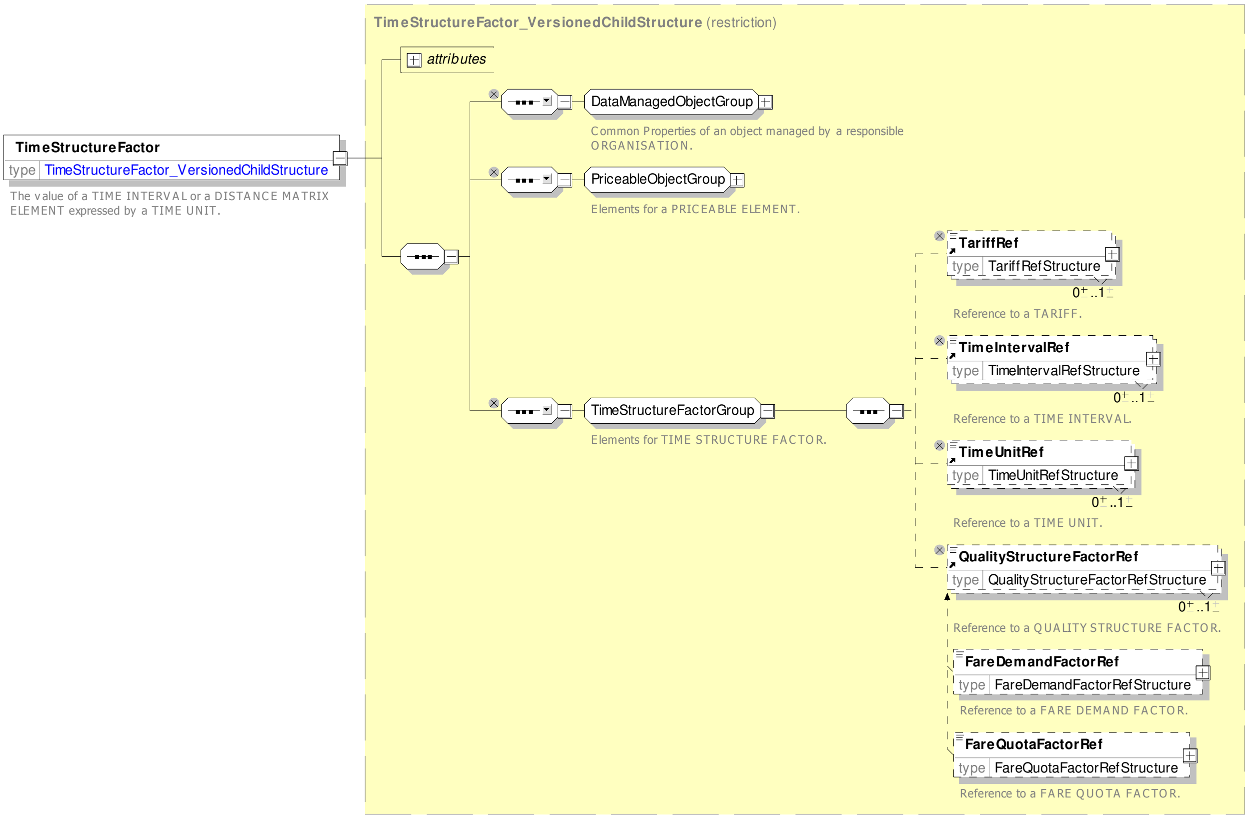


Figure 73 — *TimeStructureFactor* — XSD

8.5.4.5 Time Fare Structure – XML examples

8.5.4.5.1 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::TimeInterval">
    <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>
```

```xml
                </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>
```

8.5.5 Quality Fare Structure

8.5.5.1 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.

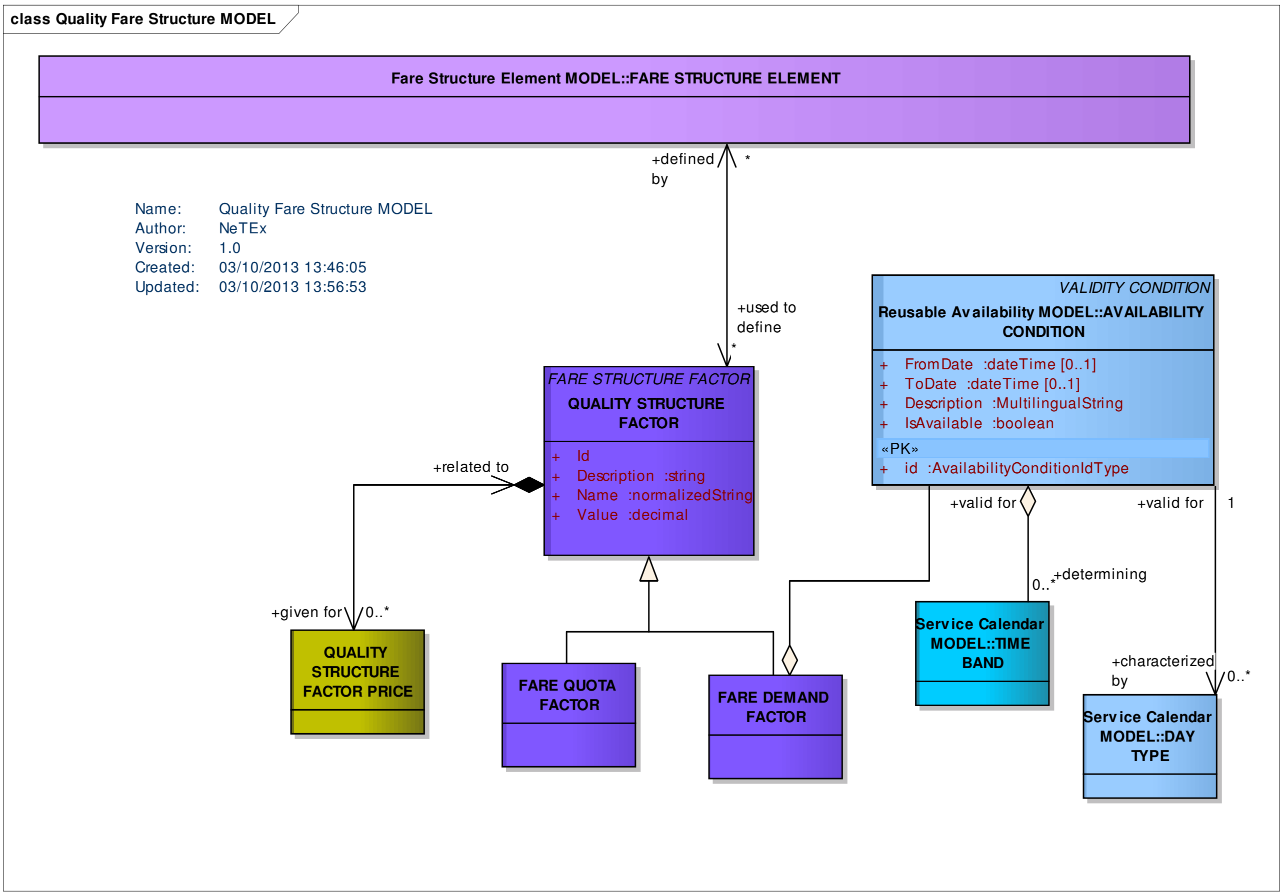


Figure 74 — Quality Fare Structure – Conceptual Model

8.5.5.2 Quality Fare Structure – Physical model

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

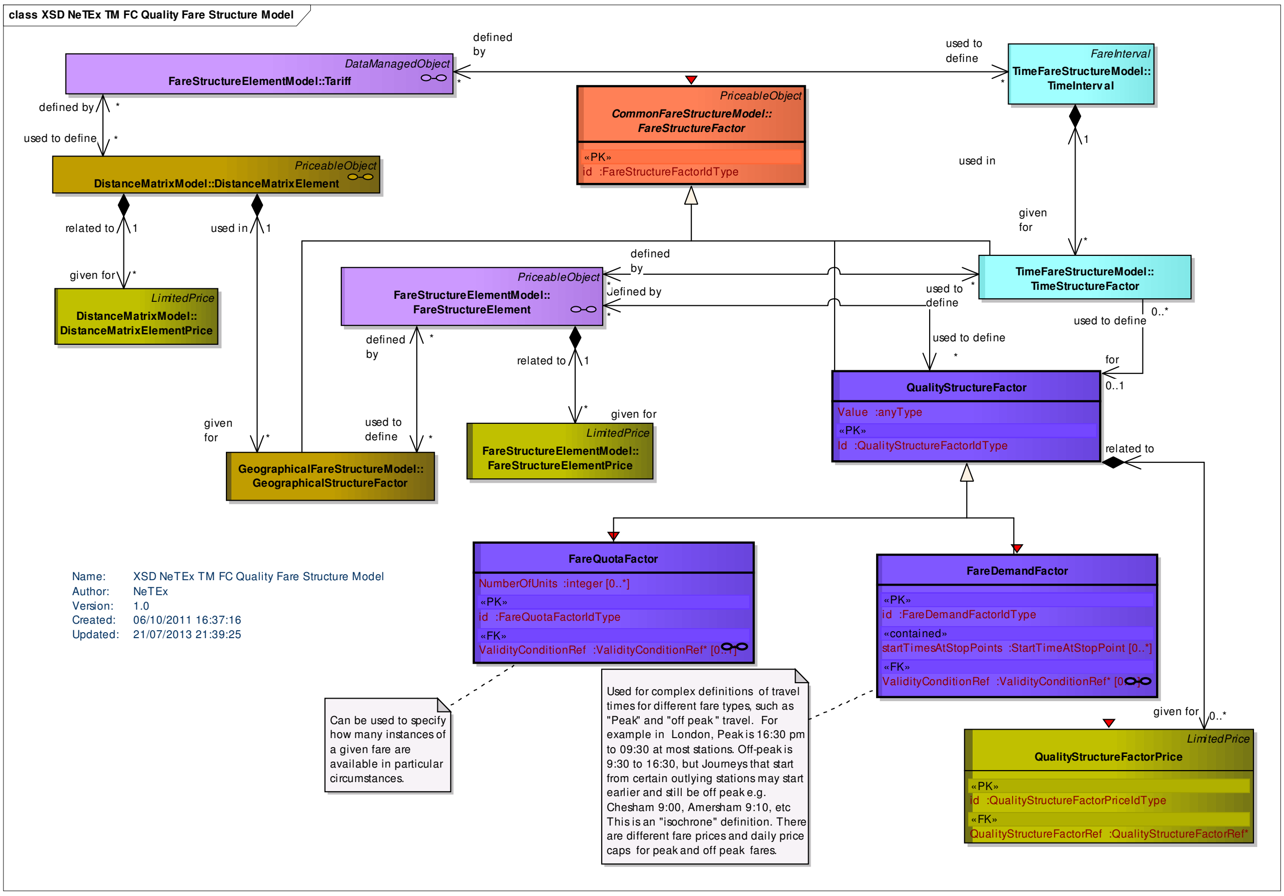


Figure 75 — Quality Fare Structure – Physical Model

8.5.5.3 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.

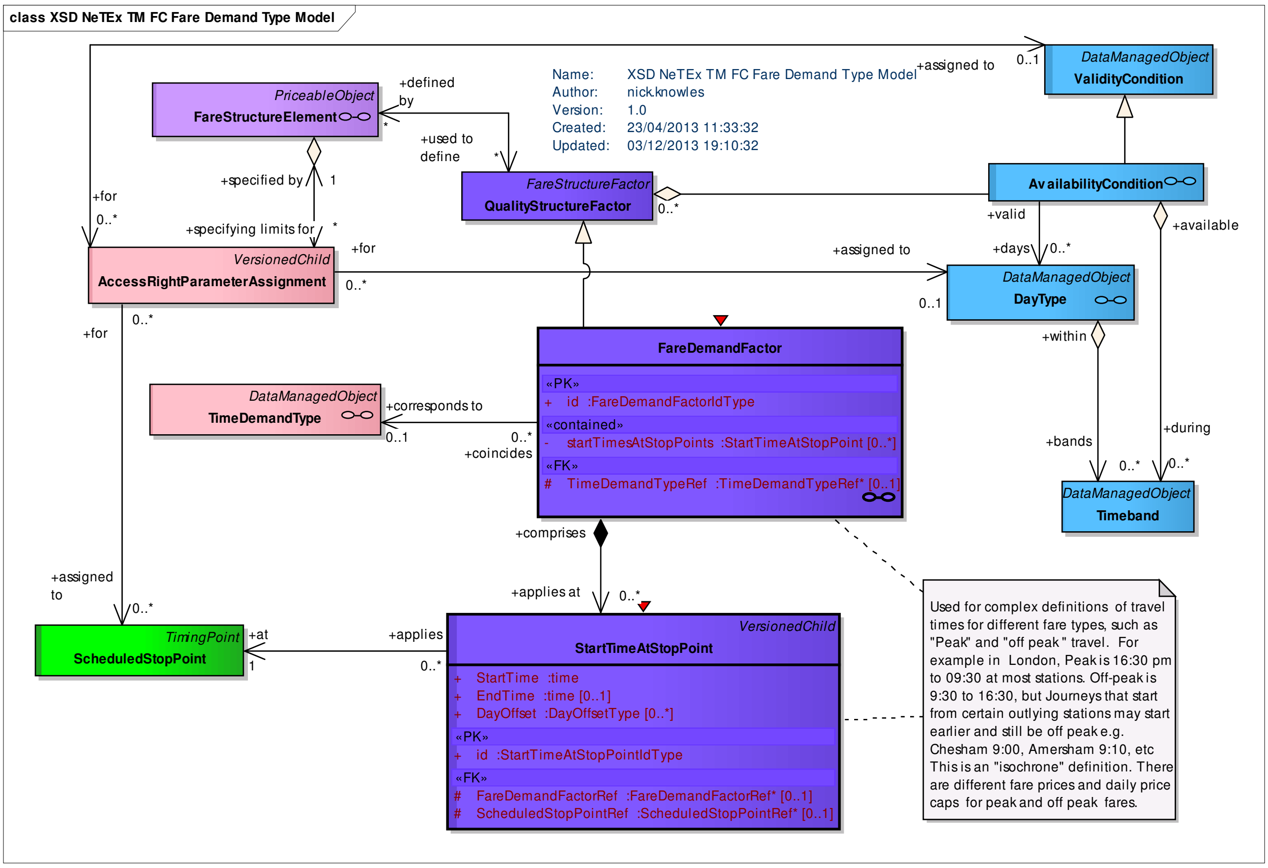


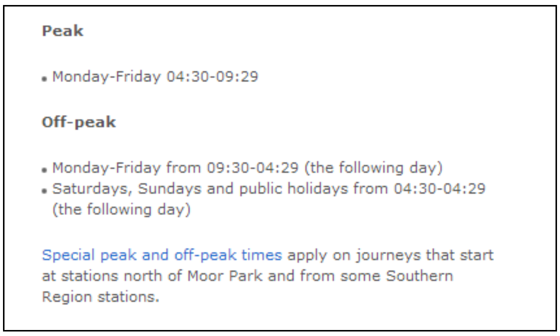
Figure 76 — Fare Demand Factor– Physical Model

8.5.5.4 Quality Fare Structure – Examples

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

8.5.5.4.1 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 77 — *Fare Demand Factor* — TfL metro Example

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 78 — *Start Times At Stop Point* — TfL Metro Example

[TO DO ADD OTHER EXAMPLES]

8.5.5.5 Quality Fare Structure – Attributes and XSD

8.5.5.5.1 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 49 – *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.
“cntd»	***validity-Conditions***	*ValidityConditionRef*	0:*	VALIDITY CONDITIONs governing FARE DEMAND FACTOR. See NeTEx Part1.
	Value	*xsd:anyType*	1:1	Quantitative quality value.
	prices	*QualityStructureFactorPrice*	0:*	Price for quality factor.

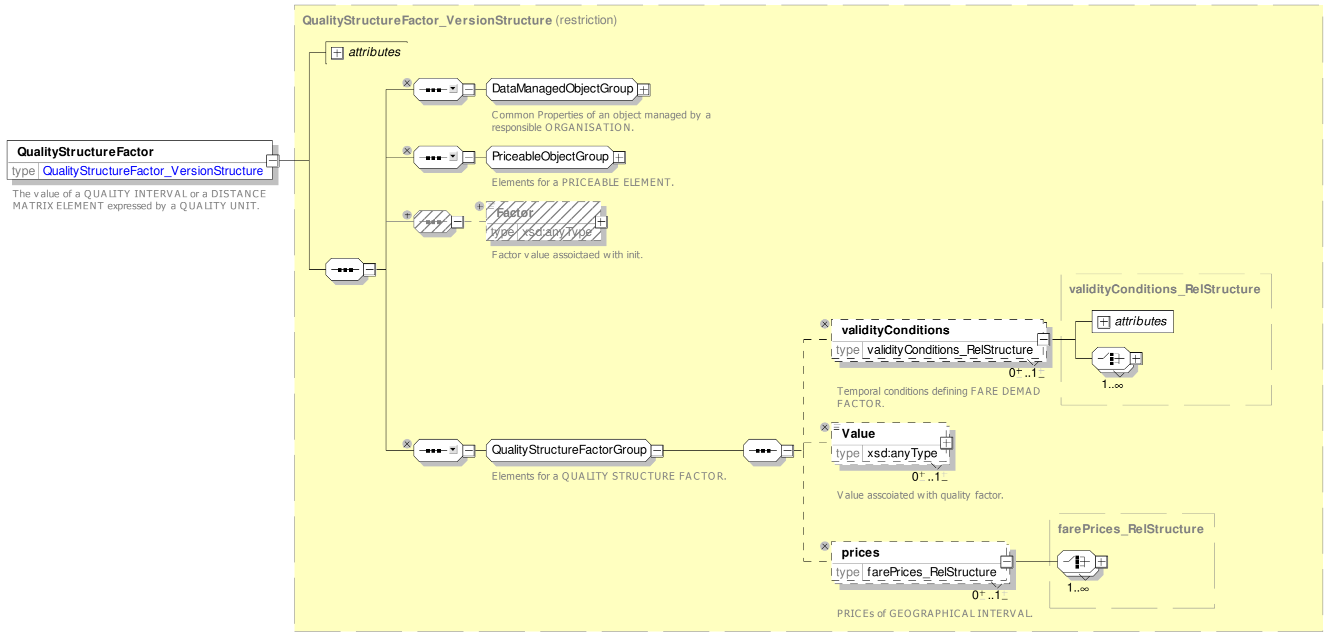


Figure 79 — *QualityStructureFactor* — XSD

8.5.5.5.2 QualityStructureFactorPrice – Model Element

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

Table 50 – *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*	1:1	Reference to QUALITY STRUCTURE FACTOR for which this is the price.

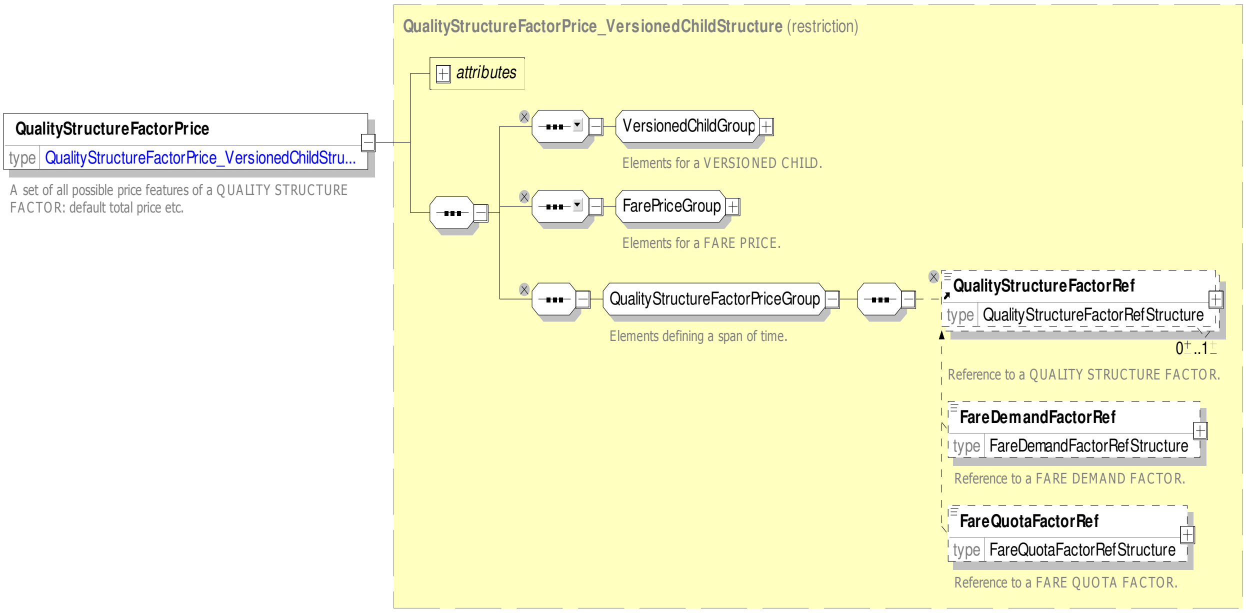


Figure 80 — *QualityStructureFactorPrice* — XSD

8.5.5.5.3 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 51 – *FareDemandFactor* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*QualityStructureFactor*	::>	FARE DEMAND FACTOR inherits from QUALITY STRUCTURE FACTOR.
«PK»	***id***	*FareDemandFacto-rIdType*	1:1	Identifier of a FARE DEMAND FACTOR.
	FareDemand-Type	*FareDemand-TypeEnum*	0:1	TIME DEMAND TYPE corresponding to FARE DEMAND FACTOR.
«FK»	***TimeDemand-TypeRef***	*TimeDemandTypeRef*	0:1	TIME DEMAND TYPE corresponding to FARE DEMAND FACTOR. See NeTEx Part2.
"cntd»	***startTimesAt-StopPoints***	*StartTimeAtStopPoint*	0:*	Start times at SCHEDULED STOP POINTs for FARE DEMAND TYPE.

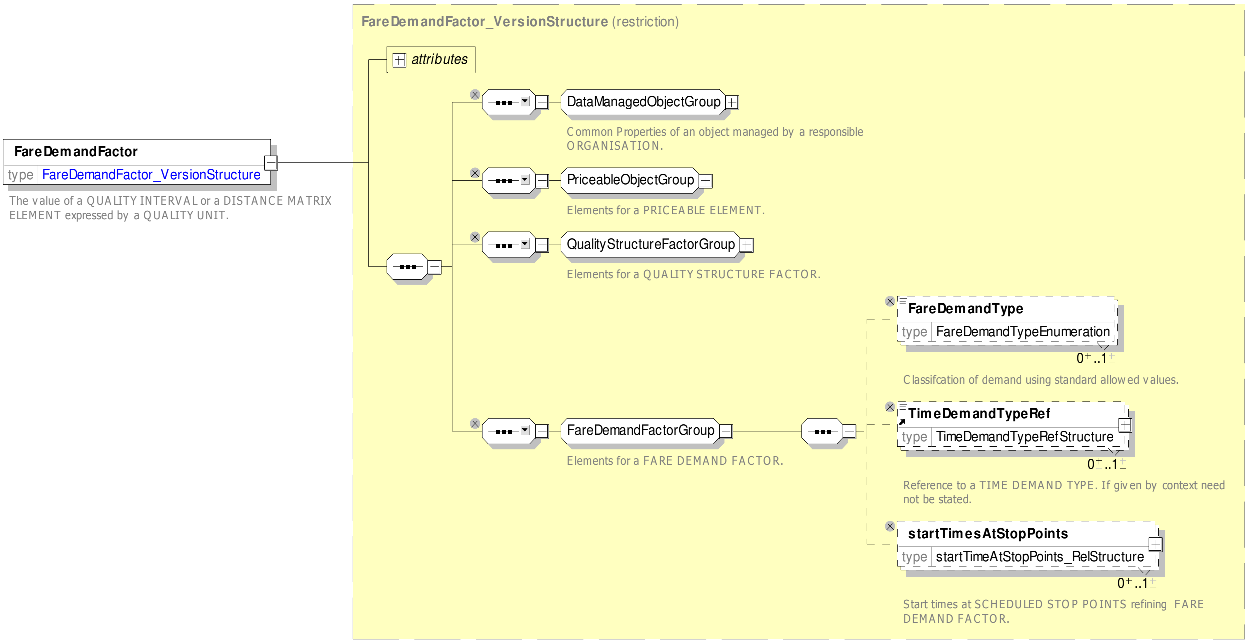


Figure 81 — *FareDemandFactor* — XSD

8.5.5.5.3.1 FareDemandType – Allowed values

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

Table 52 – *FareDemandType* – Allowed values

Value	Description
peak	Peak travel time.
middle	Between peak and off-peak travel time.
offPeak	Off Peak travel time.
night	Night travel time.

8.5.5.5.4 StartTimeAtStopPoint – Model Element

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 53 – *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-***	*FareDemandFactorRef*	0:1	FARE DEMAND FACTOR for which start time

	FactorRef			applies.
«FK»	***ScheduledStop-PointRef***	*ScheduledStopPointRef*	0:1	Station at which time band start applies.
	StartTime	*xsd:time*	1:1	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.

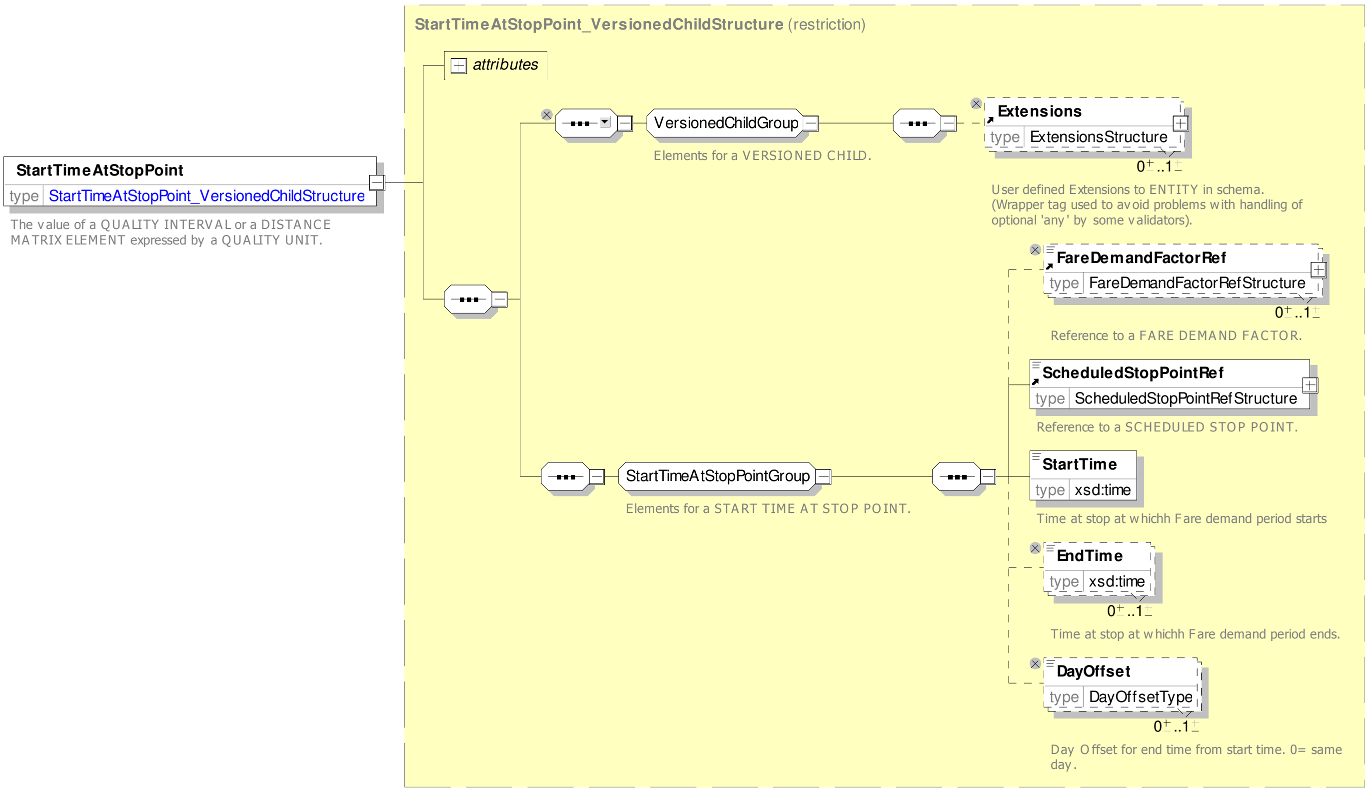


Figure 82 — *StartTimeAtStopPoint* — XSD

8.5.5.5.5 FareQuotaFactor – Model Element

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

Table 54 – *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.

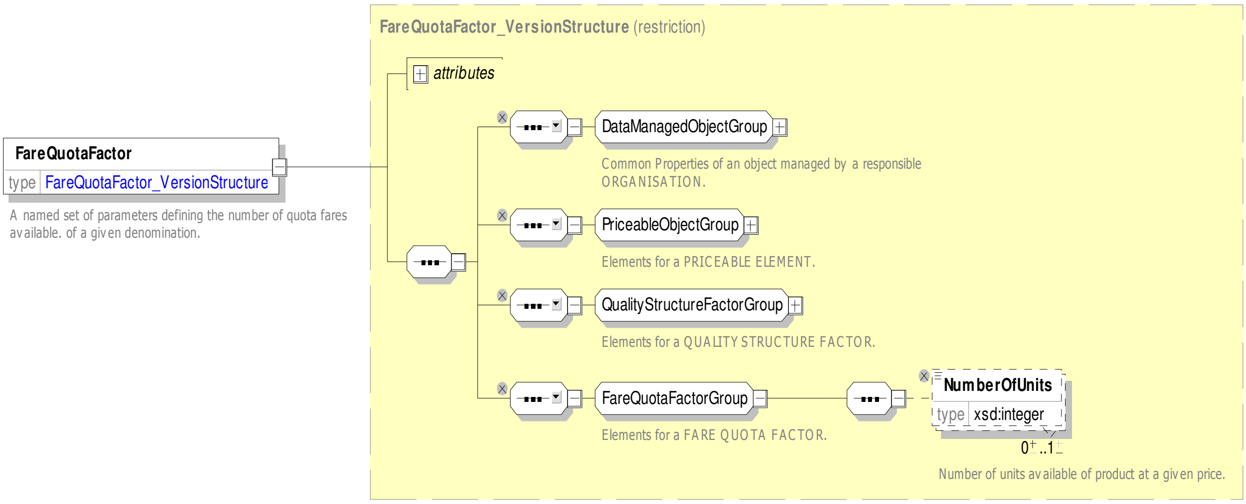


Figure 83 — *FareQuotaFactor* — XSD

8.5.5.6 Quality Fare Structure – XML examples

8.5.5.6.1 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::anyTime">
            <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 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
                </Description>
            <validityConditions>
                <AvailabilityConditionRef ref="tfl:AvailabilityCondition: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">
```

```xml
                    <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>
```

8.5.6 Fare Structure Element

8.5.6.1 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.

8.5.6.2 Controllable Elements[KB13]

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 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 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.

8.5.6.3 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).

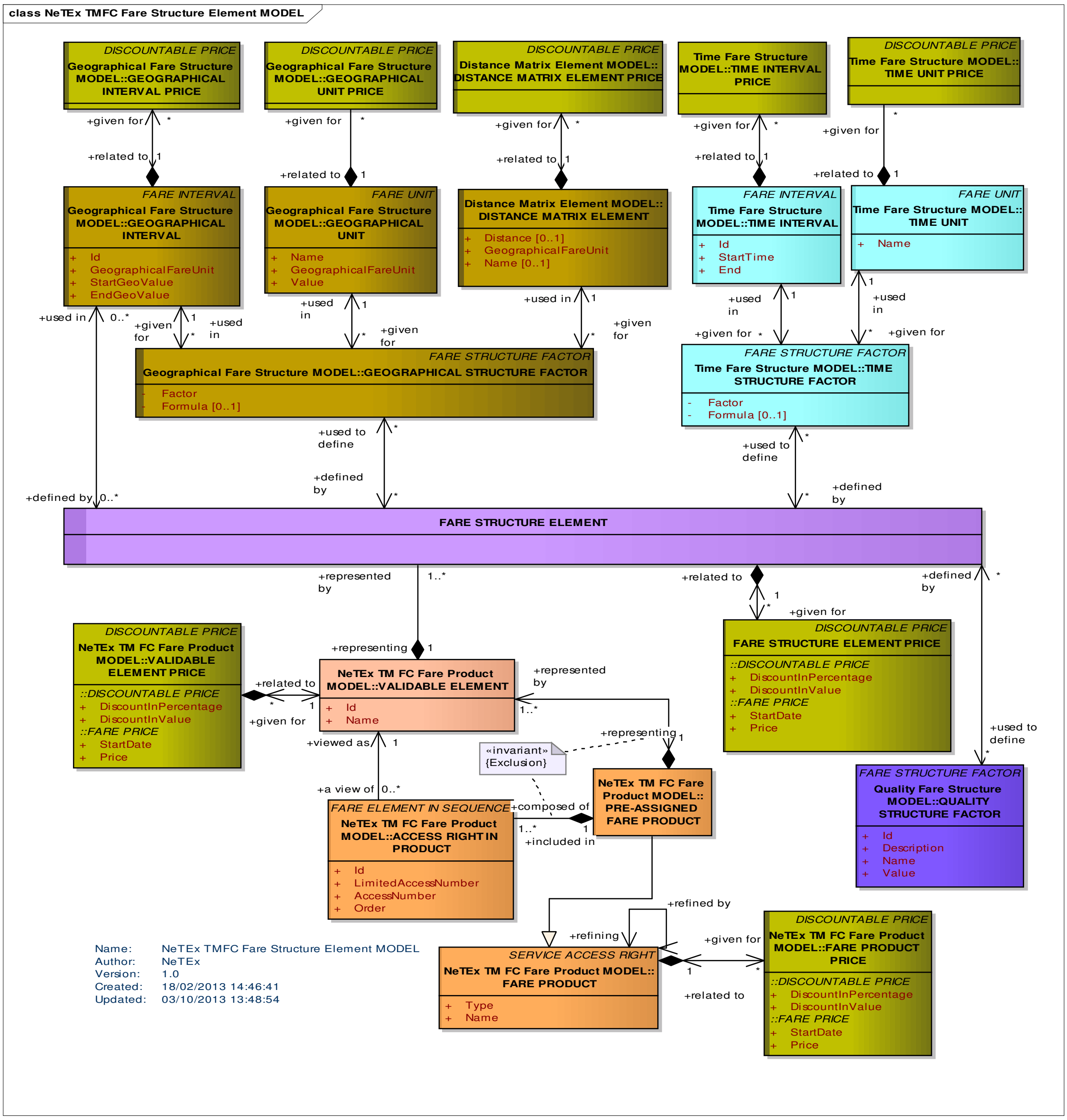


Figure 84 —Fare Structure Element– Conceptual Model

8.5.6.4 Tariff – Conceptual model

The FARE STRUCTURE ELEMENT MODEL also describes TARIFFs – groupings of FARE STRUCTURE ELEMENTs subject to particular availability[KB14] 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.

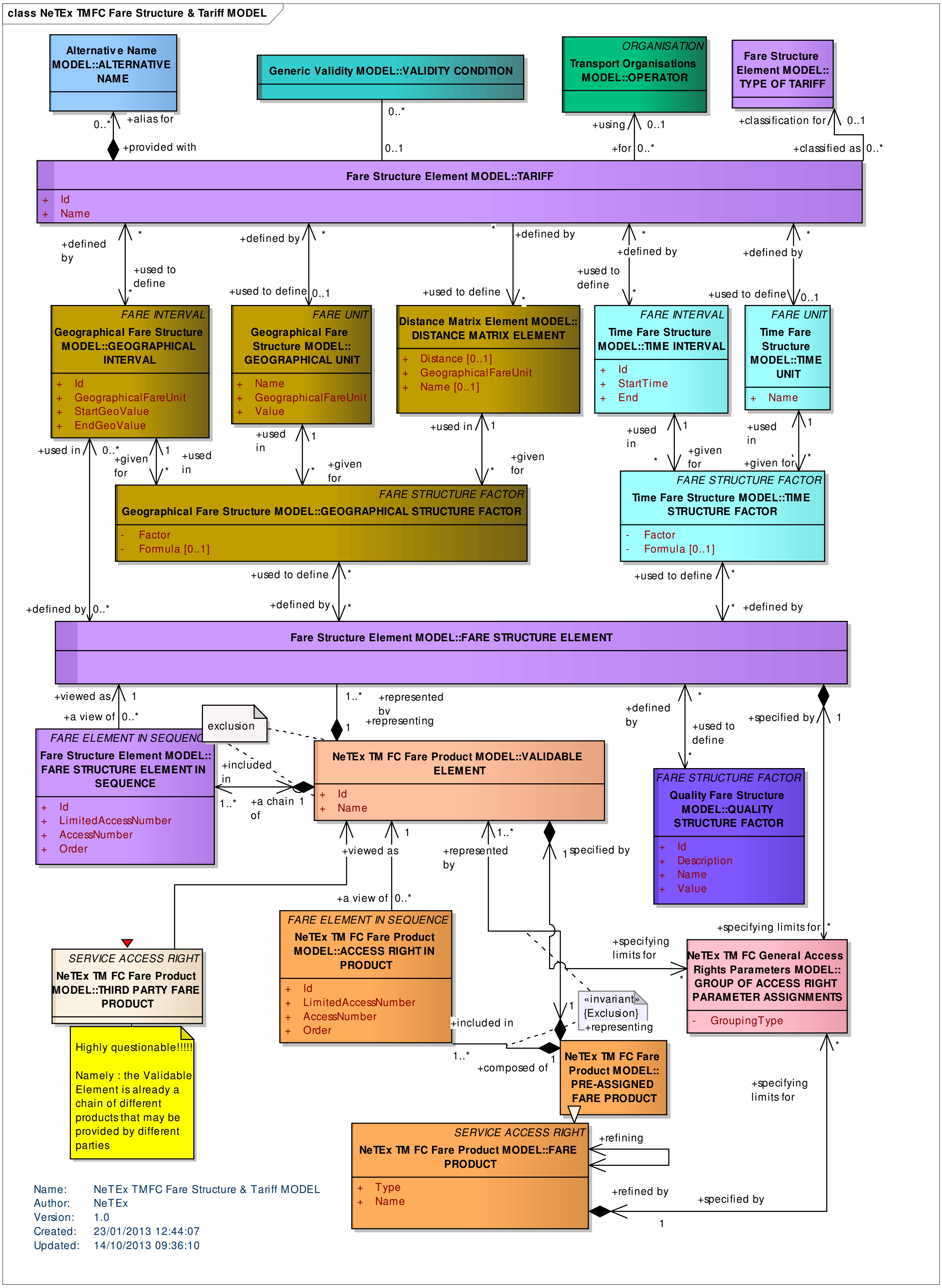


Figure 85 —Tariff Element– Conceptual Model

8.5.6.5 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 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 ELMENTs, that is, a matrix of point to point fares – see use cases for fare preparation.

Table 55 – 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.)	Example?
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	VALIDABLE ELEMENT + FARE STRUCTURE ELEMENT GEOGRAPHICAL FARE FACTOR	

	considered similar)	+ 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 + TARIFF ZONE	Paris ??
Honeycomb zones		VALIDABLE ELEMENT + ???	TO DO? Or is this a "Zonal far"
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	TO DO
Stage Fare -	Fare is charged for according to the specific sections reached	VALIDABLE ELEMENT + DISTANCE MATRIX ELEMENT + ZONE SECTION	TO DO
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	Paris ??
[OTHER MORE COMPLEX ? TO DO			

8.5.6.5.1 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

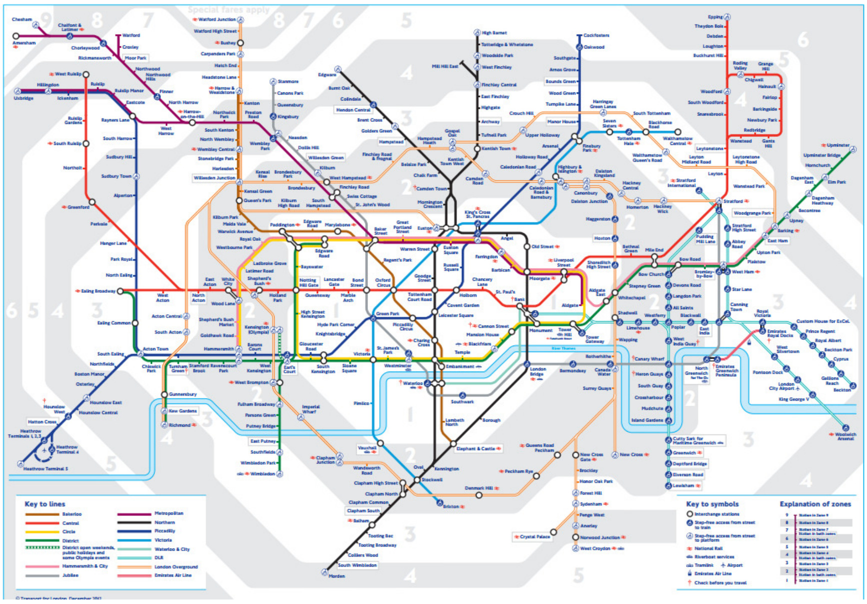


Figure 86 —Example Zone to Zone Fares – London Underground Map

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 87 —Example Zone to Zone Fares – London Underground Prices

{TO DO ADD MORE EXAMPLES]

8.5.6.6 Fare Structure Element – Physical model

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

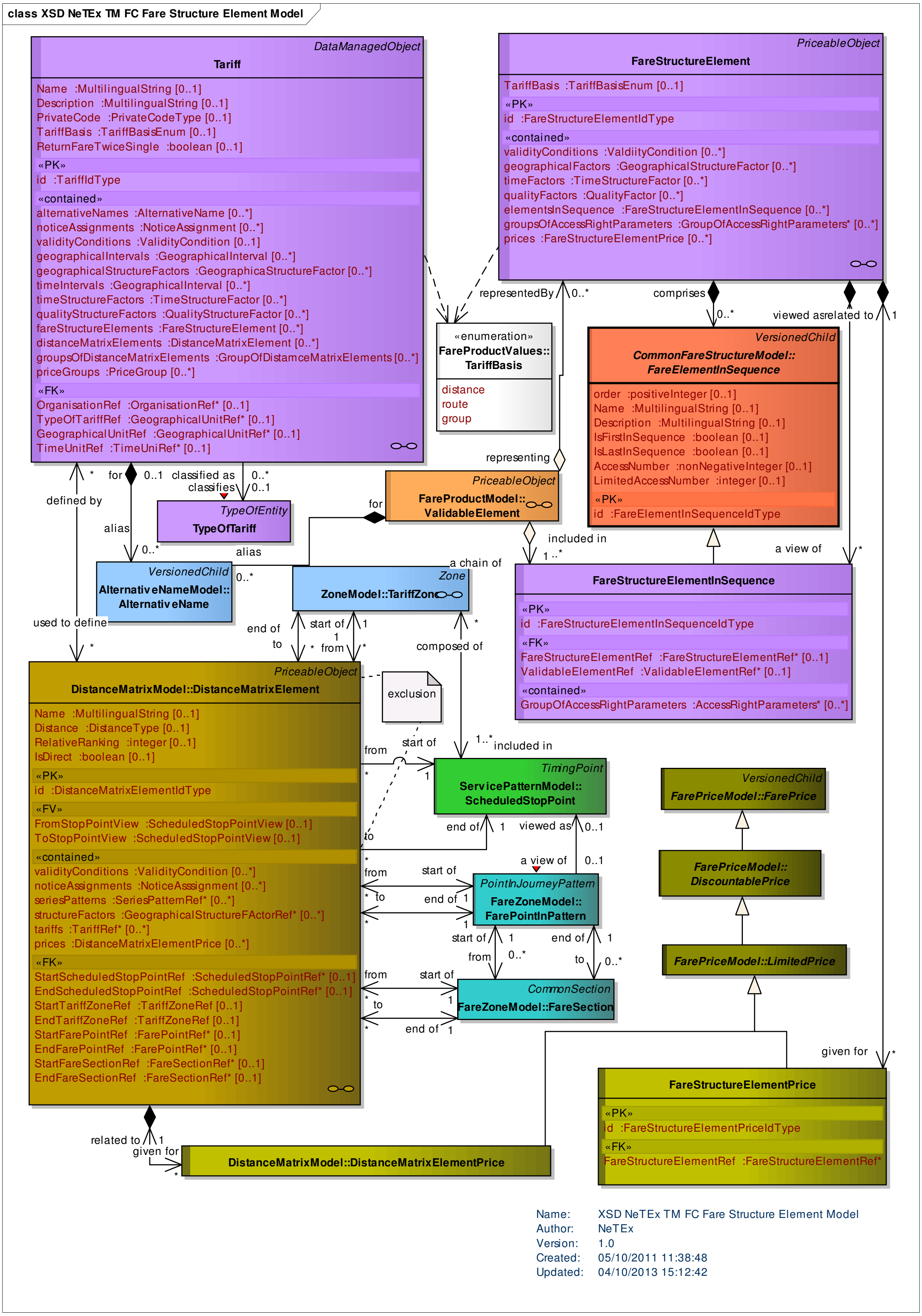


Figure 88 —Fare Structure Element– Physical Model

8.5.6.7 Fare Structure Element – Attributes and XSD

8.5.6.7.1.1 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 56 – *FareStructureElement* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	FARE STRUCTURE ELEMENT inherits from PRICEABLE OBJECT.
«PK»	***id***	*FareStructureElement-IdType*	1:1	Identifier of FARE STRUCTURE ELEMENT.
“cntd»	***validity-Conditions***	*ValidityConditions*	0:*	VALIDITY CONDITIONs that apply to FARE STRUCTURE ELEMENT.
	TariffBasis	*TariffBasisEnum*	0:1	TARIFF BASIS to be used for the element. See allowed values below
GROUP	***FareStructure-ElementFactor-Group***	*FareStructure-ElementFactorGroup*	1:1	FARE STRUCTURE FACTORs associated with the FARE STRUCTURE ELEMENT.
GROUP	***FareStructure-Component-Group***	*FareStructureComponent Group*	1:1	FARE STRUCTURE components associated with the FARE STRUCTURE ELEMENT.
“cntd»	***prices***	*FareStructureElement-Price*	0:*	Prices for the FARE STRUCTURE ELEMENT.

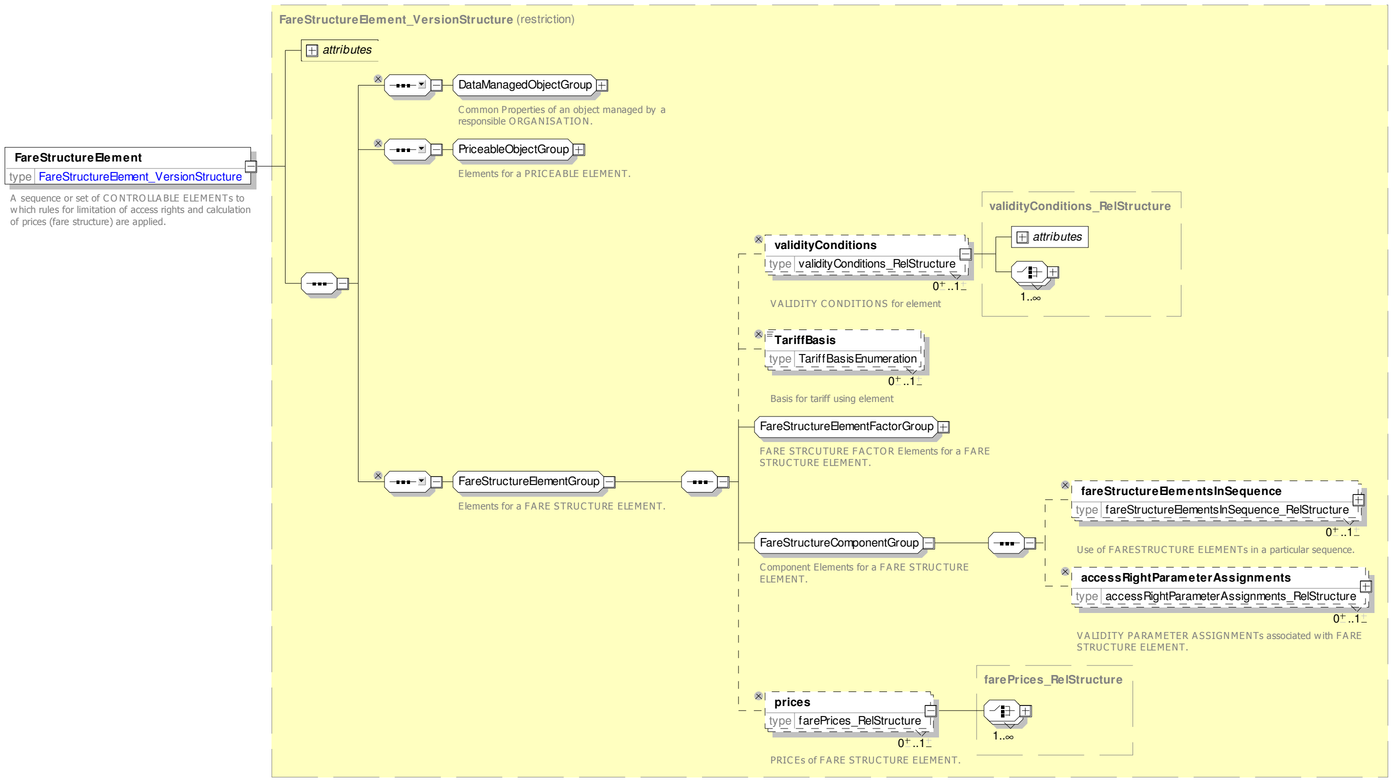


Figure 89 — *FareStructureElement* — XSD

8.5.6.7.1.2 FareStructureFactorGroup – Group

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

Table 57 – *FareStructureFactorGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***Geographical-IntervalRef***	*GeographicalIntervalRef*	1:1	Reference to GEOGRAPHICAL INTERVAL associated with FARE STRUCTURE ELEMENT.
“cntd»	***geographical-Factors***	*GeographicalStructure-Factor*	0:*	GEOGRAPHICAL FACTORs associated with the FARE STRUCTURE ELEMENT.
«FK»	***TimeIntervalRef***	*TimeIntervalRef*	1:1	Reference to TIME INTERVAL associated with FARE STRUCTURE ELEMENT.
“cntd»	***timeFactors***	*TimeStructureFactor*	0:*	TIME FACTORs associated with the FARE STRUCTURE ELEMENT.
“cntd»	***qualityFactors***	*QualityFactor*	0:*	QUALITY FACTORs associated with the FARE STRUCTURE ELEMENT.
«FK»	***DistanceMatrix-ElementIRef***	*DistanceMatrix-ElementIRef*	1:1	Reference to DISTANCE MATRIX ELEMENT associated with FARE STRUCTURE ELEMENT.

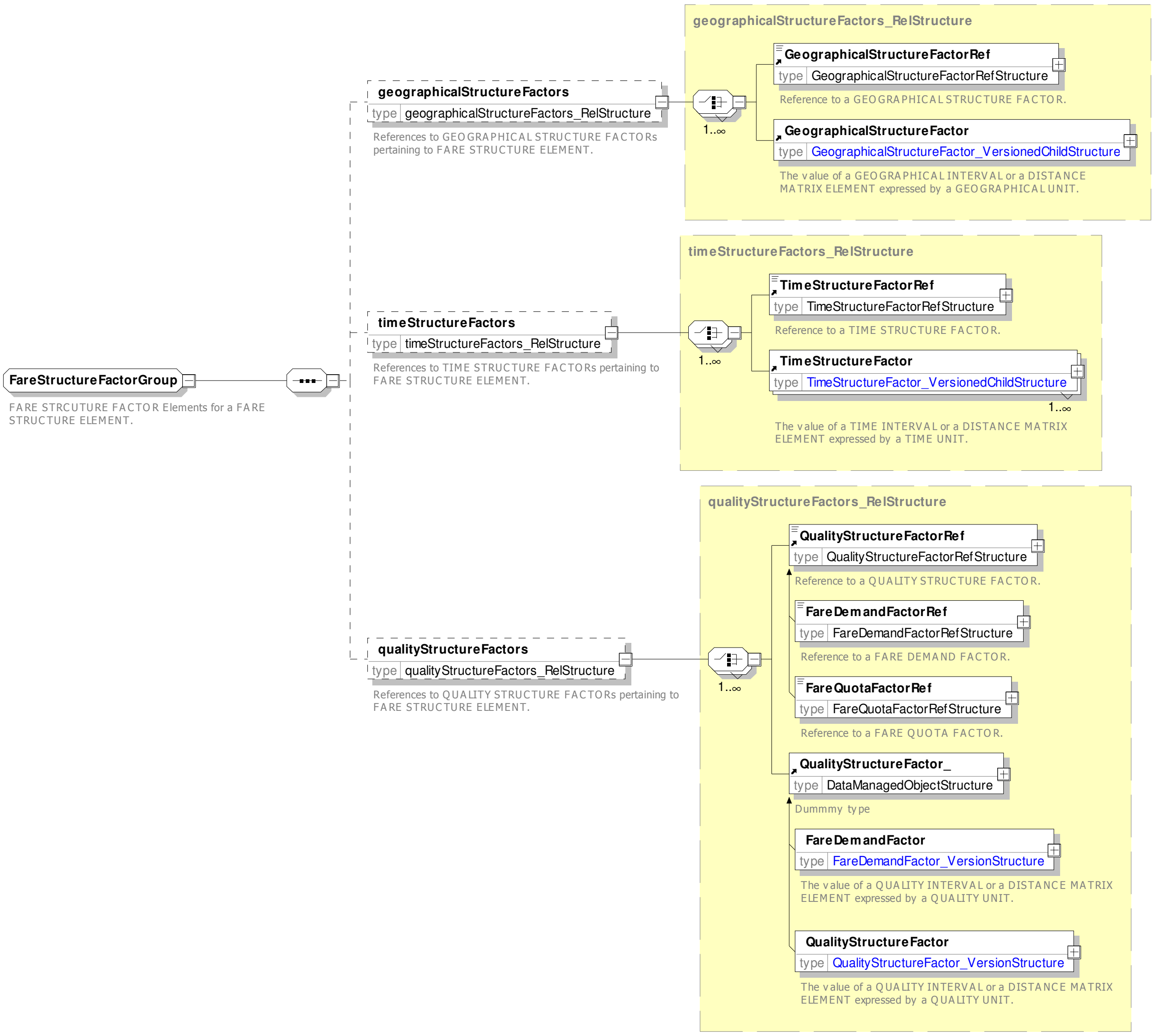


Figure 90 — *FareStructureFactorGroup* — XSD

8.5.6.7.1.3 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 58 – *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.
“cntd»	***accessRight-Parameter-***	*AccessRightParameter-*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs associated with the FARE STRUCTURE

	Assignments	*Assignments*		ELEMENT.

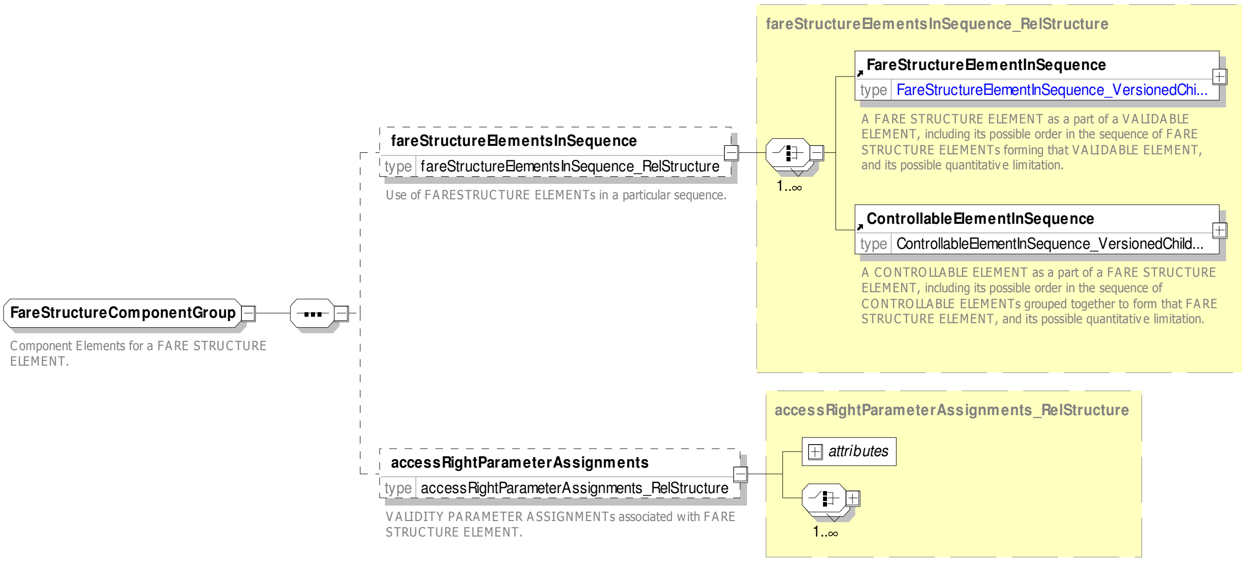


Figure 91 — *FareStructureComponentGroup* — XSD

8.5.6.7.2 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 59 – *FareStructureElementInSequence* – Element

Classification	Name	Type	Cardinality	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***	*FareStructure-ElementRef*	0:1	Reference to a FARE STRUCTURE ELEMENT.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	0:1	Reference to a VALIDABLE ELEMENT.
“cntd»	***validity-Parameter-Assignments***	*ValidityParameter-Assignment*	0:*	VALIDITY PARAMETER ASSIGNMENTs associated with the ELEMENT IN SEQUENCE.

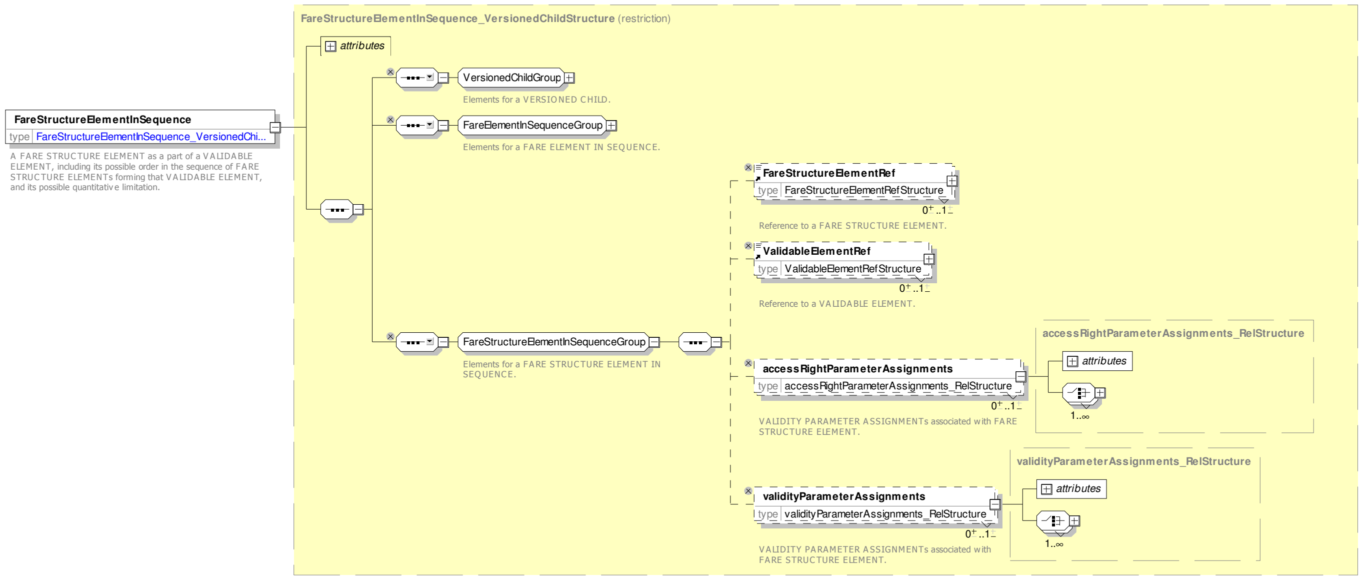


Figure 92 — *FareStructureElementInSequence* — XSD

8.5.6.7.3 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 60 – *FareStructureElementPrice* – Element

Classifi-cation	Name	Type	Cardinality	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***	*FareStructureElement-Ref*	1:1	Reference to a FARE STRUCTURE ELEMENT for which this is the price.

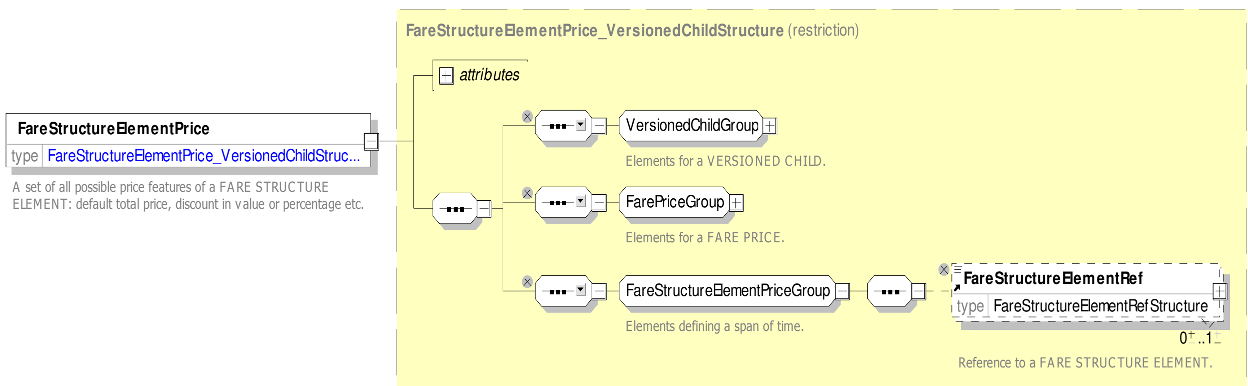


Figure 93 — *FareStructureElementPrice* — XSD

8.5.6.7.4 Tariff – Model Element

A particular tariff, described by a combination of parameters.

Table 61 – *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.
	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.
	PrivateCode	*PrivateCodeType*	0:1	Alternative identifier of an entity; can be used to associate with legacy systems.
“cntd»	***ValidityConditions***	*ValidityCondition*	0:1	VALIDITY CONDITIONs of TARIFF.
GROUP	***TariffOrganisation-Group***	*TariffOrganisationGroup*	1:1	Elements for ORGANISATIONs offering TARIFF.
GROUP	***TariffCalculation-Group***	*TariffCalculationGroup*	0:1	Fare calculation elements for TARIFF.
GROUP	***TariffGeographical-Group***	*TariffGeographicalGroup*	1:1	GEOGRAPHICAL STRUCTURE FACTOR elements for TARIFF.
GROUP	***TariffTimeGroup***	*TariffTimeGroup*	1:1	TIME STRUCTURE FACTOR calculation elements for TARIFF.
GROUP	***TariffQualityGroup***	*TariffQualityGroup*	1:1	TARIFF QUALITY elements for TARIFF.
GROUP	***TariffStructure-Group***	*TariffStructureGroup*	1:1	TARIFF STRUCTURE elements for TARIFF.
GROUP	***TariffPricesGroup***	*TariffPricesGroup*	1:1	TARIFF PRICE elements for TARIFF.

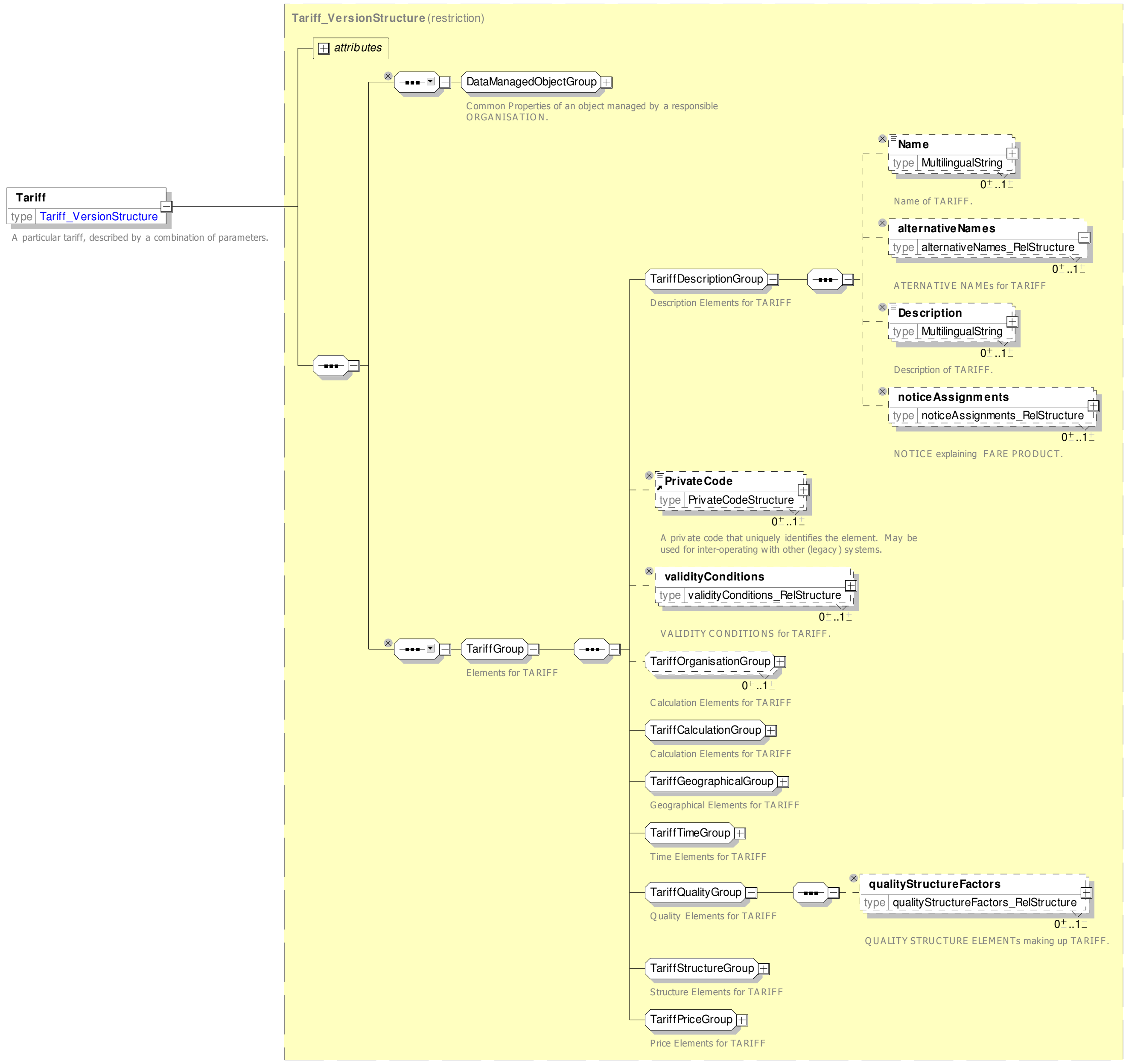


Figure 94 — *Tariff* — XSD

8.5.6.7.4.1 TariffOrganisationGroup – Group

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

Table 62 – *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 ORGANISATION to which TARIFF applies.

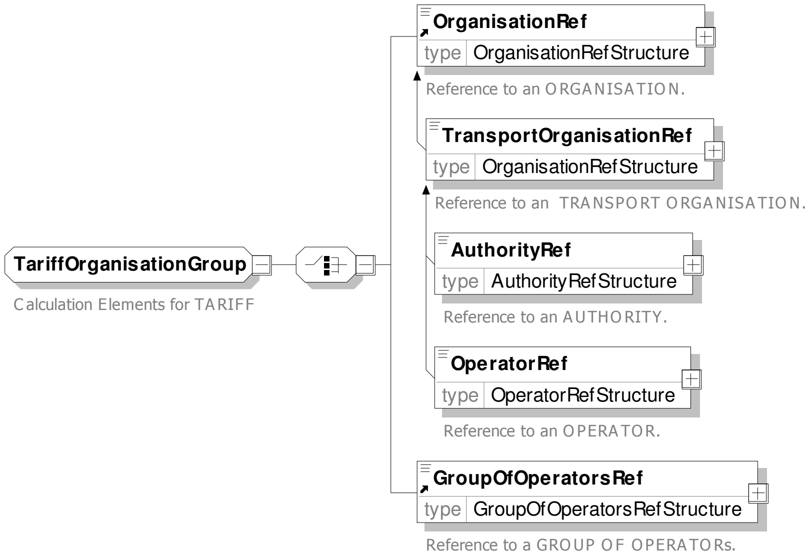


Figure 95 — *TariffOrganisationGroup* — XSD

8.5.6.7.4.2 TariffCalculationGroup – Group

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

Table 63 – *TariffCalculationGroup* – Group

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

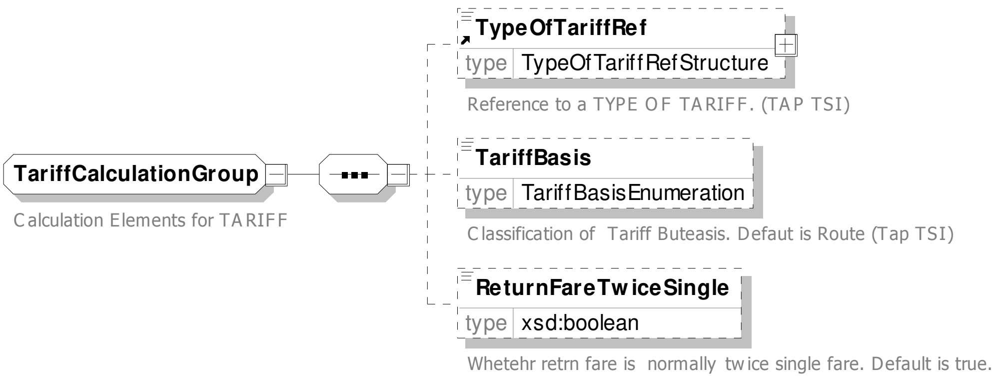


Figure 96 — *TariffCalculationGroup* — XSD

8.5.6.7.4.3 TariffGeographicalGroup – Group

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

Table 64 – *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.

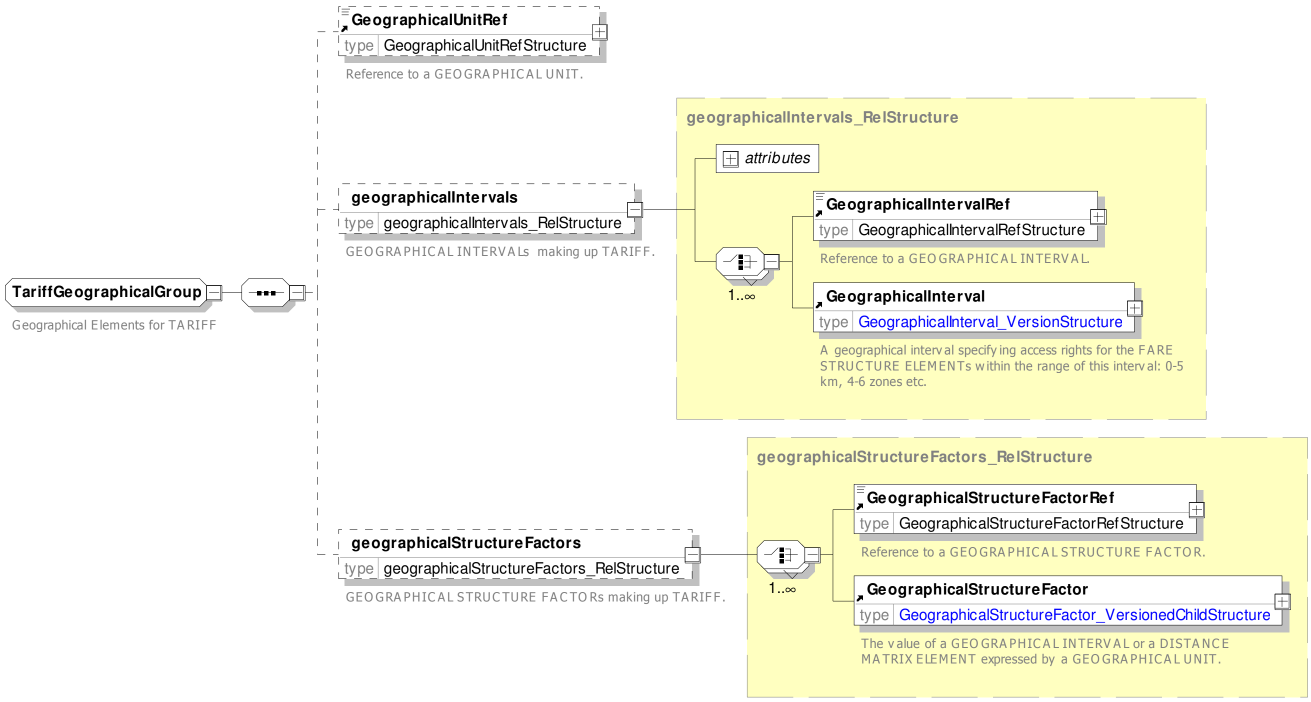


Figure 97 — *TariffGeographicalGroup* — XSD

8.5.6.7.4.4 TariffTimeGroup – Group

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

Table 65 – *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.

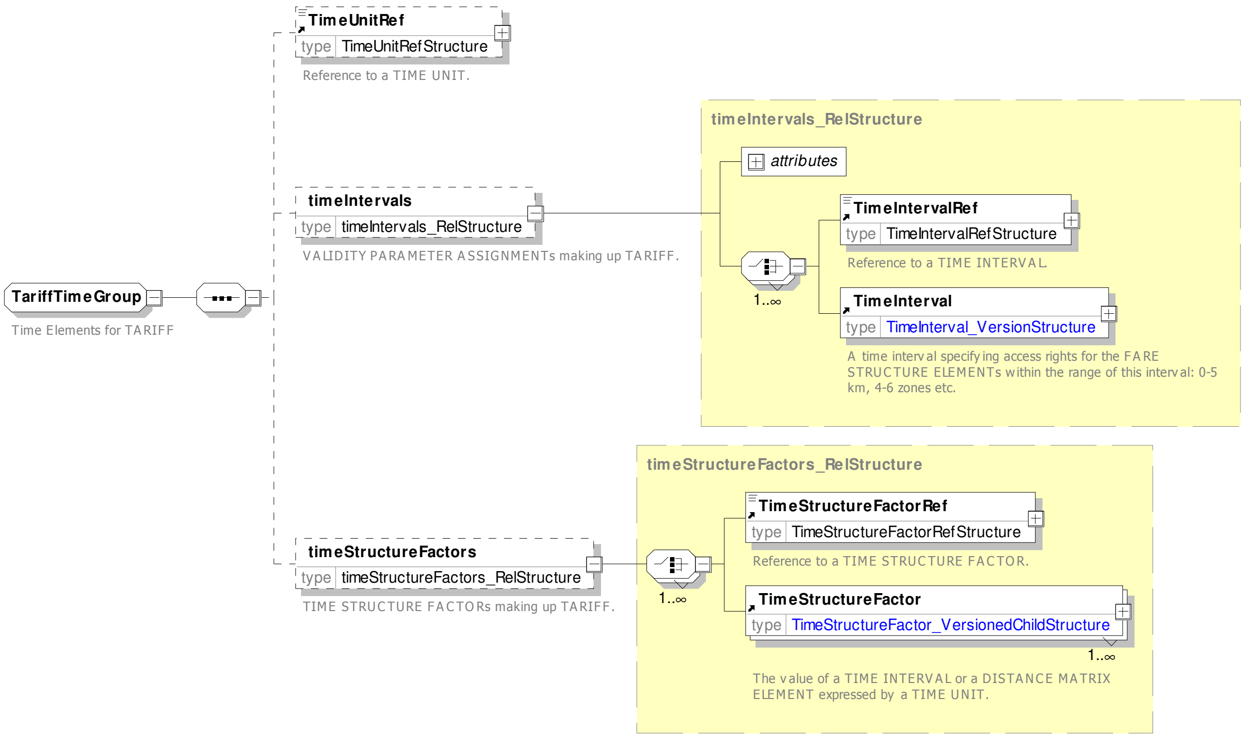


Figure 98 — *TariffTimeGroup* — XSD

8.5.6.7.4.5 TariffQualityGroup – Group

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

Table 66 – *TariffQualityGroup* – Group

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

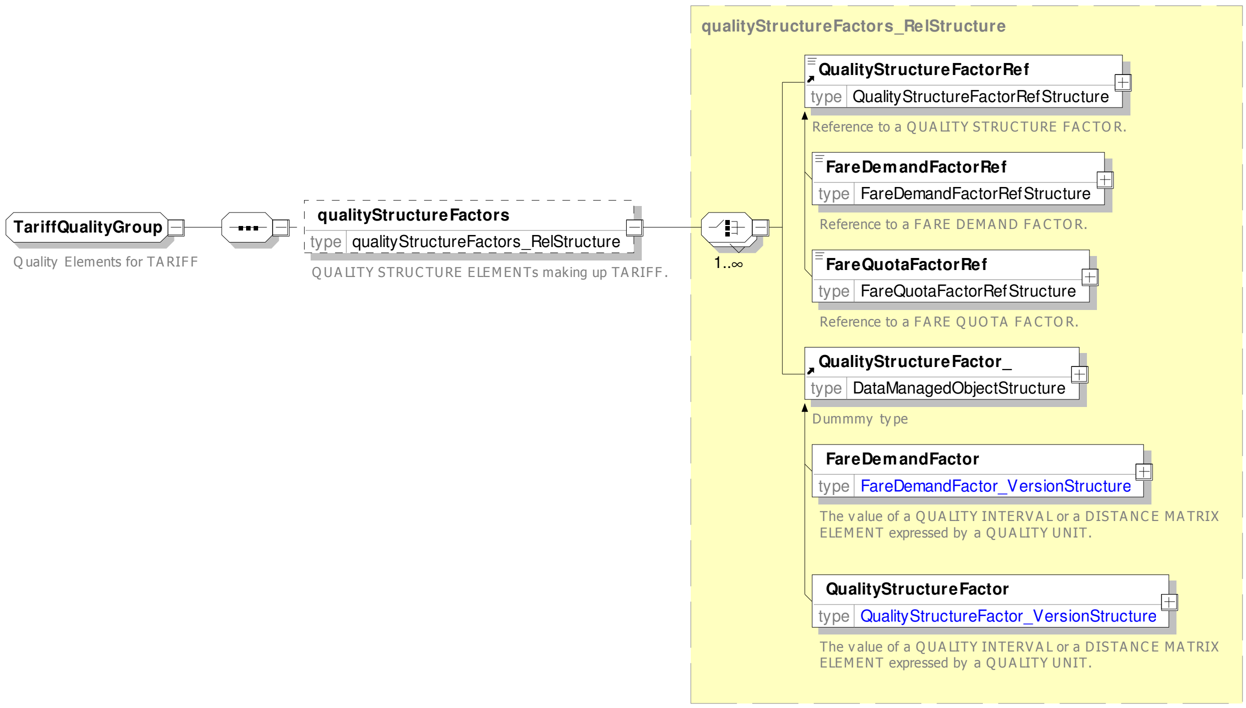


Figure 99 — *TariffQualityGroup* — XSD

8.5.6.7.4.6 TariffStructureGroup – Group

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

Table 67 – *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:*	GROUP OF DISTANCE MATRIX ELEMENTs associated with TARIFF.

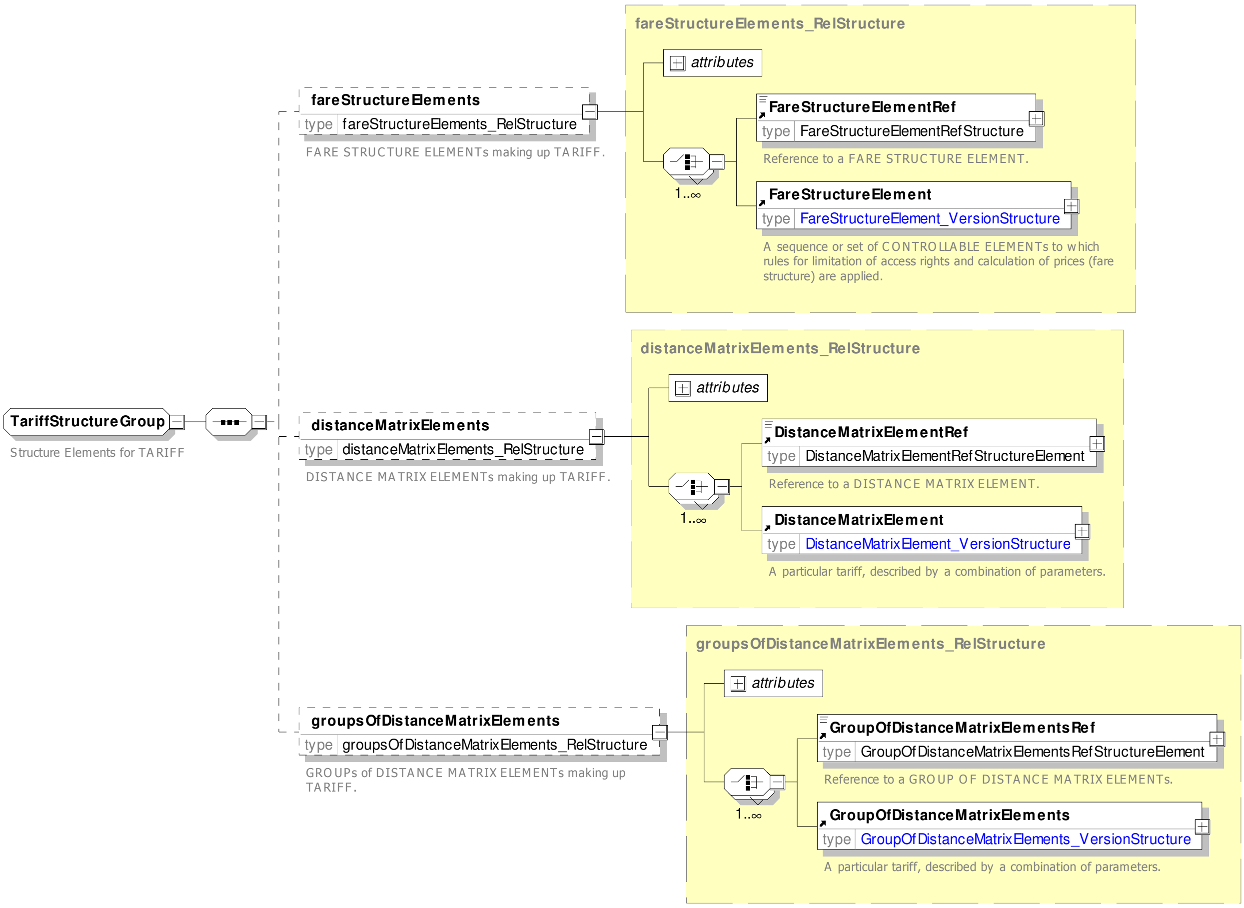


Figure 100 — *TariffStructureGroup* — XSD

8.5.6.7.4.7 TariffPricesGroup – Group

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

Table 68 – *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.

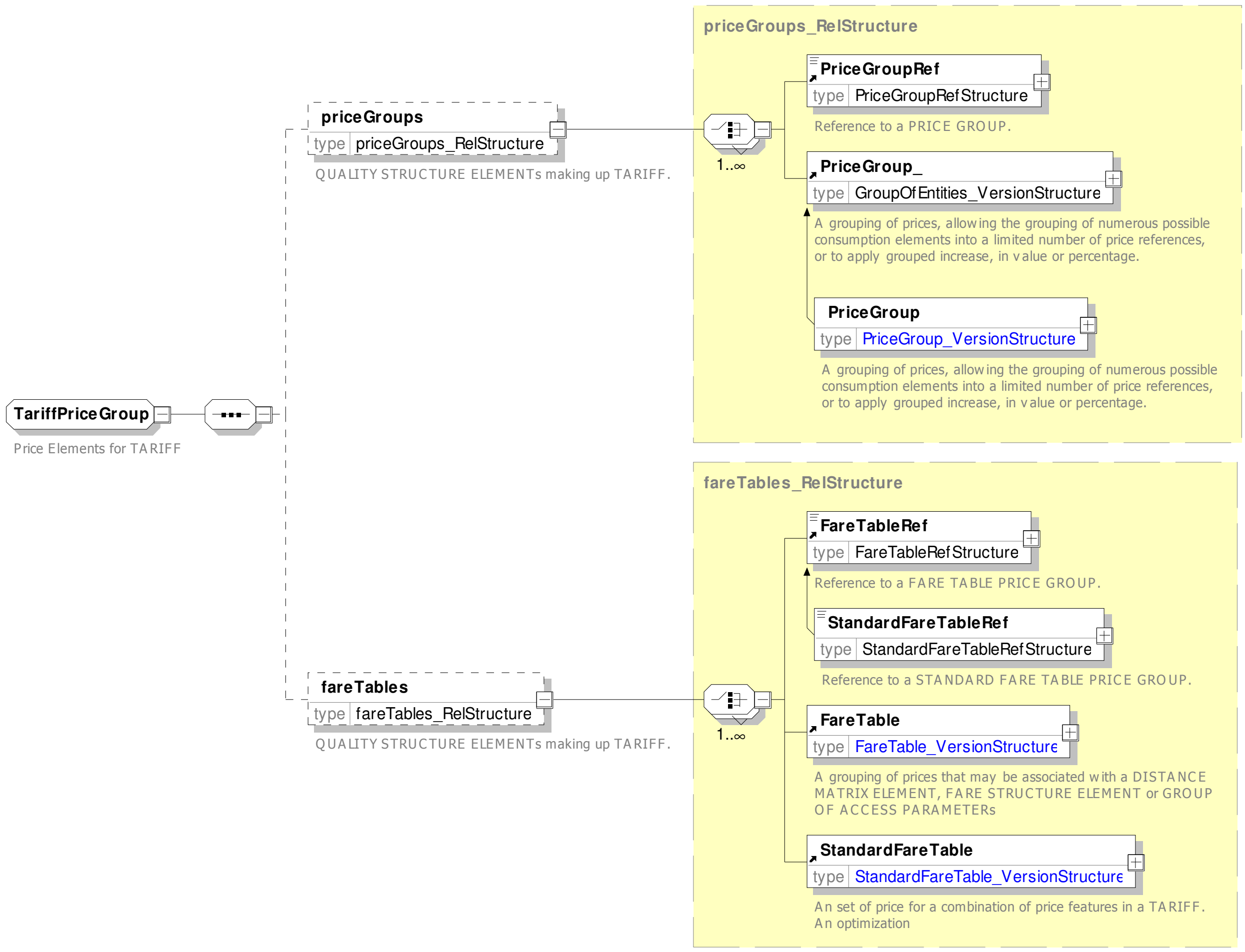


Figure 101 — *TariffPricesGroup* — XSD

8.5.6.7.5 TypeOfTariff – Model Element

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

Table 69 – *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.

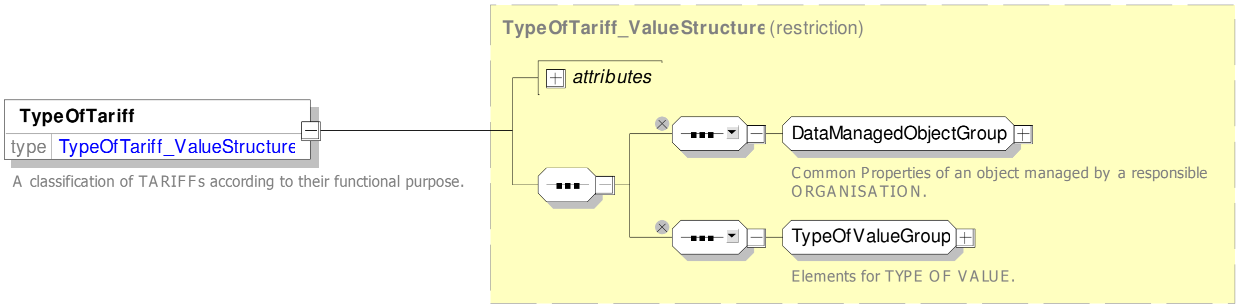


Figure 102 — *TypeOfTariff* — XSD

8.5.6.8 Fare Structure Element – XML examples

8.5.6.8.1 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

b) A TARIFF Definition

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

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

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

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

g) A FARE STRUCTURE ELEMENT defining the allowed SERIES CONSTRAINT's 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>
        </AvailabilityCondition>
    </validityConditions>
    <OperatorRef ref="tap::0106" version="any"/>
    <TypeOfTariffRef ref="tap::B.1.1:01" version="any"/>
    <TariffBasis>route</TariffBasis>
    <ReturnFareTwiceSingle>true</ReturnFareTwiceSingle>
    <fareStructureElements>
```

```xml
        <FareStructureElementRef ref="tap::NrtProduct@roundTrips" version="01"/>
        <FareStructureElementRef ref="tap::NrtProduct@seatClasses" 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@roundTrips" version="01">
        <Name>Single or return tickets are available</Name>
        <accessRightParameterAssignments>
            <GenericParameterAssignment id="tap::NrtProduct@roundTrip" version="01">
                <PreassignedFareProductRef ref="tap::NrtProduct" version="01"/>
                <LimitationGroupingType>OR</LimitationGroupingType >
                        <for>
                            <RoundTripRef ref="tap::single" version="any"/>
                            <RoundTripRef ref="tap::return" version="any"/>
                        </for>
                    </GenericParameterAssignment>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
    </FareStructureElement>

    <FareStructureElement id="tap::NrtProduct@seatClasses" version="01">
        <Name>First of second class tickets are available </Name>
        <accessRightParameterAssignments>
            <GenericParameterAssignment id="tap::NrtProduct@seatClass" version="01">
                <GroupingType>OR</GroupingType>
                <PreassignedFareProductRef ref="tap::NrtProduct" version="01"/>
                <includes>
                    <GenericParameterAssignment id="tap::NrtProduct@firstClass" version="01">
                        <Scope><SeatClass>firstClass</SeatClass></Scope>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment id="tap::NrtProduct@secondClass" version="01">
                        <Scope><SeatClass>secondClass</SeatClass></Scope>
                    </GenericParameterAssignment>
                </includes>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
    </FareStructureElement>

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

```
                        <for>
                            <GroupTicketRef ref="tap::groupTicket" version="any"/>
                        </for>
                    </GenericParameterAssignment>
                </includes>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
    </FareStructureElement>

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

…. etc, etc
```

8.5.6.8.2 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">
            <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>
        <accessRightParameterAssignments>
            <GenericParameterAssignment version="any" id="mygtfsxm::1">
                <Scope><TariffZoneRef version="any" ref="mygtfsxm::1"/></Scope>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
    </FareStructureElement>
    <FareStructureElement version="any" id="mygtfsxm::2">
        <Name>Zone 2</Name>
        <accessRightParameterAssignments>
            <GenericParameterAssignment version="any" id="mygtfsxm::2">
                <Scope><TariffZoneRef version="any" ref="mygtfsxm::2"/></Scope>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
    </FareStructureElement>
    <FareStructureElement version="any" id="mygtfsxm::3">
        <Name>Zone 3</Name>
        <accessRightParameterAssignments>
            <GenericParameterAssignment version="any" id="mygtfsxm::3">
                <Scope><TariffZoneRef version="any" ref="mygtfsxm::3"/></Scope>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
    </FareStructureElement>

…. etc, etc
```

[TO DO MORE EXAMPLES]

8.5.7 Distance Matrix Element

8.5.7.1 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.

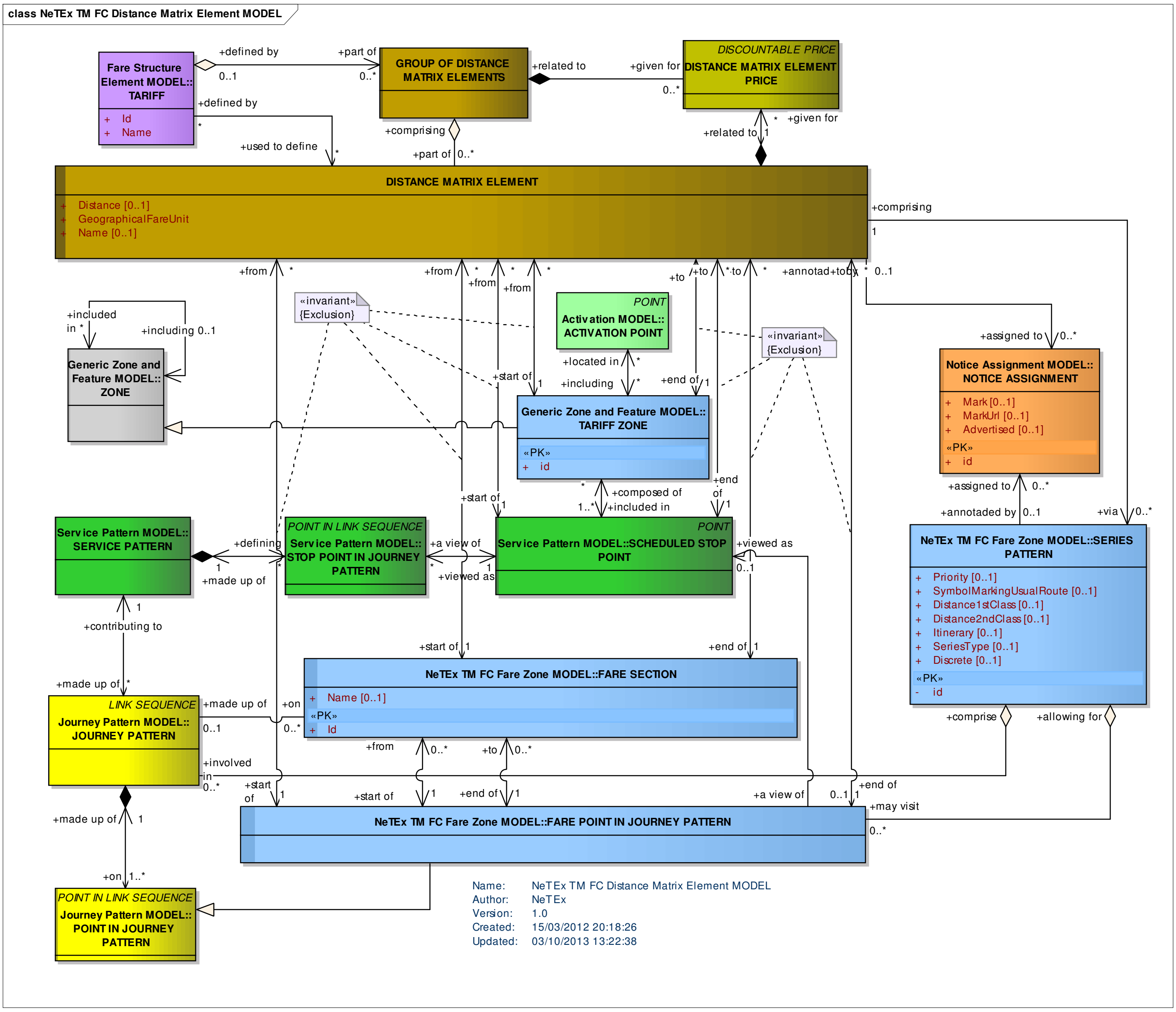


Figure 103 — Distance Matrix Element – Conceptual Model

8.5.7.2 Distance Matrix Element – Conceptual Examples

8.5.7.2.1 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

Figure 104 — Example: Distance Matrix Element – Triangular Fare table with absolute Prices

8.5.7.2.2 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

8.5.7.2.3 Example – Zone to Zone fare

The following partial table shows the zone to zone based fares for travel by rail in the London area. 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 105 — Example: Distance Matrix Element – Zone to Zone fare on London

8.5.7.3 Distance Matrix Element – Physical model

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

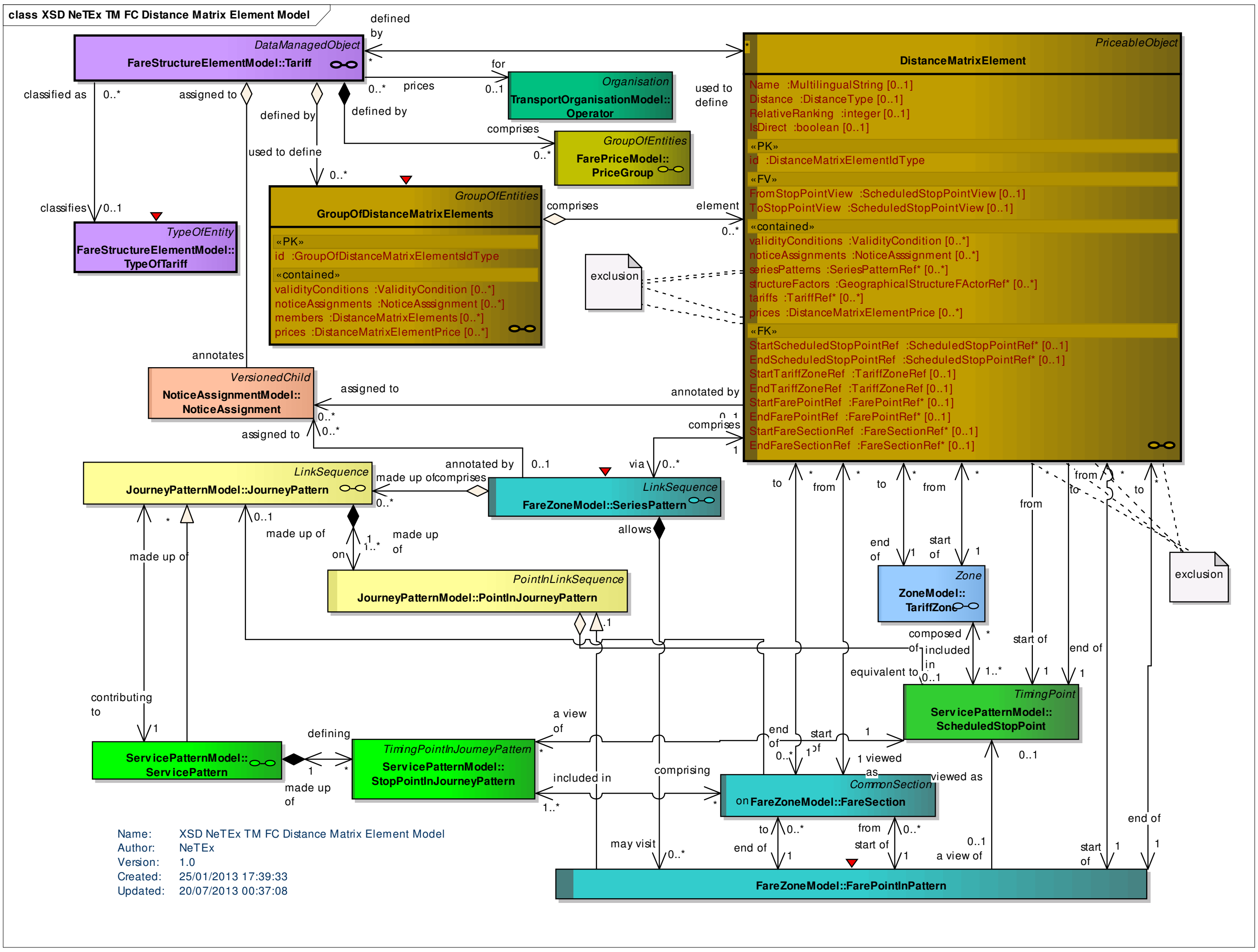


Figure 106 — Distance Matrix Element – Physical Model

8.5.7.4 Distance Matrix Element – Attributes and XSD

8.5.7.4.1 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 70 – *DistanceMatrixElement* – Element

Classifi-cation	Name	Type	Cardinality	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.

GROUP	***DistanceMatrix-Element-DetailsGroup***	*DistanceMatrixElement-DetailsGroup*	0:1	Detailed property elements for DISTANCE MATRIX ELEMENT.
GROUP	***DistanceMatrix-Element-ODGroup***	*DistanceMatrixElement-ODGroup*	1:1	Origin and Destination elements for DISTANCE MATRIX ELEMENT.
GROUP	***DistanceMatrix-Element-Component-Group***	*DistanceMatrixElement-ComponentGroup*	0:1	Component elements for DISTANCE MATRIX ELEMENT.
» GROUP	***DistanceMatrix-ElementPricing-Group***	*DistanceMatrixElement-ComponentGroup*	0:1	Pricing elements for DISTANCE MATRIX ELEMENT.

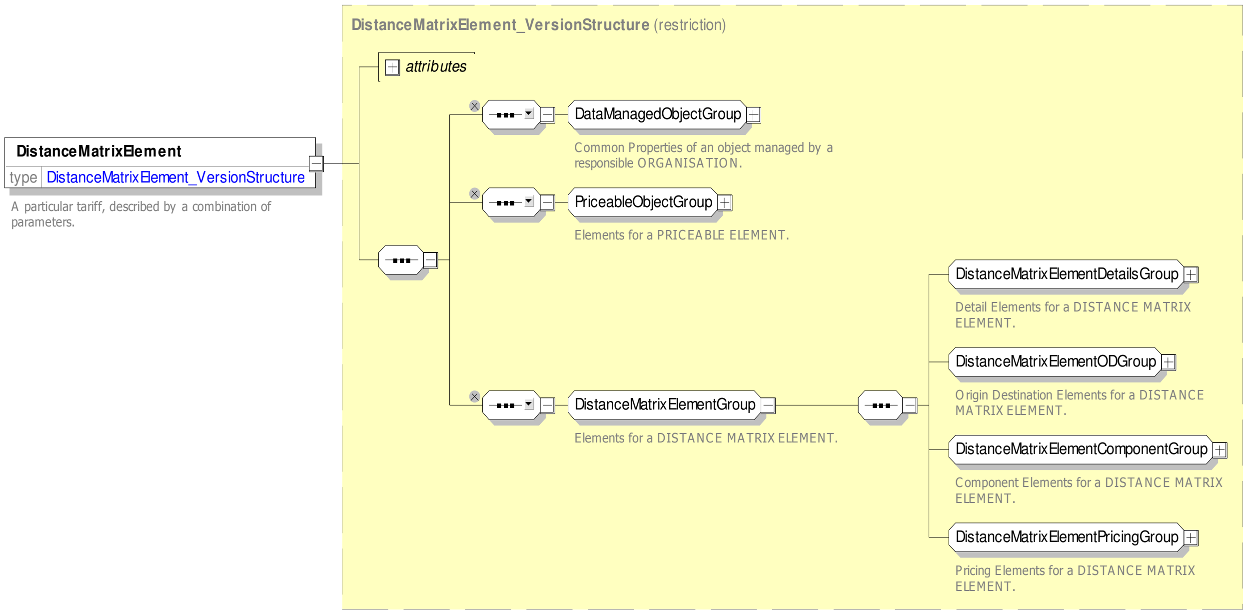


Figure 107 — *DistanceMatrixElement* — XSD

8.5.7.4.1.1 *DistanceMatrixElementDetailsGroup* – Group

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

Table 71 – *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.
“cntd»	***validity-Conditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs for element.

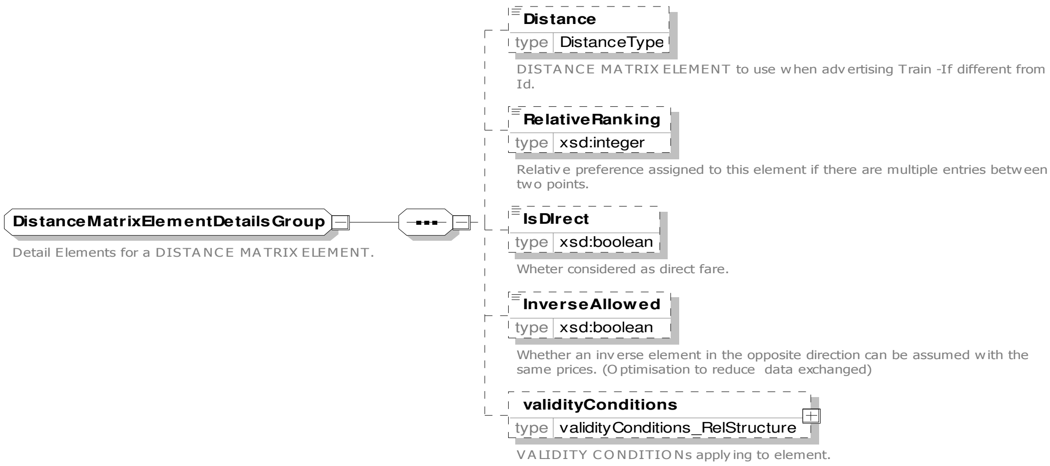


Figure 108 — *DistanceMatrixElementDetailsGroup* — XSD

8.5.7.4.1.2 *DistanceMatrixElementODGroup* – Group

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

Table 72 – *DistanceMatrixElementODGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
		Choice	1:1	Origin of DISTANCE MATRIX ELEMENT
«FK»	***Start-StopPointRef***	*ScheduledStopPointRef*	0:1	Start SCHEDULED STOP POINT at which a DISTANCE MATRIX ELEMENT begins.
«FV»	***Start-StopPointView***	*ScheduledStopPointView*	0:1	Details of origin SCHEDULED STOP POINT.
«FK»	***Start-TariffZoneRef***	*TariffZoneRef*	0:1	Start TARIFF ZONE at which a DISTANCE MATRIX ELEMENT begins.
«FV»	***Start-TariffZoneView***	*TariffZoneView*	0:1	Details of origin TARIFF ZONE.
«FK»	***StartFare-SectionRef***	*FareSectionRef*	0:1	Start FARE SECTION at which a DISTANCE MATRIX ELEMENT begins.
«FK»	***StartFarePoint-***	*FarePointIn-*	0:1	Start FARE POINT In PATTERN at which a DISTANCE MATRIX ELEMENT begins. (Handles

	InPatternRef	*JourneyPatternRef*		case of repeated visits)
		Choice	1:1	Destination of DISTANCE MATRIX ELEMENT.
«FK»	***EndStopPointRef***	*ScheduledStopPointRef*	0:1	End SCHEDULED STOP POINT at which a DISTANCE MATRIX ELEMENT ends.
«FV»	***End-StopPointView***	*ScheduledStopPointView*	0:1	Details of destination SCHEDULED STOP POINT
«FK»	***End-TariffZoneRef***	*TariffZoneRef*	0:1	Final TARIFF ZONE at which a DISTANCE MATRIX ELEMENT ends.
«FV»	***End-TariffZoneView***	*TariffZoneView*	0:1	Details of origin TARIFF ZONE.
«FK»	***EndFare-SectionRef***	*FareSectionRef*	0:1	End FARE SECTION at which a DISTANCE MATRIX ELEMENT ends.
«FK»	***EndFarePoint-InPatternRef***	*FarePointIn-JourneyPatternRef*	0:1	End FARE POINT IN JOURNEY PATTERN at which a DISTANCE MATRIX ELEMENT ends. (Handles case of repeated visits).

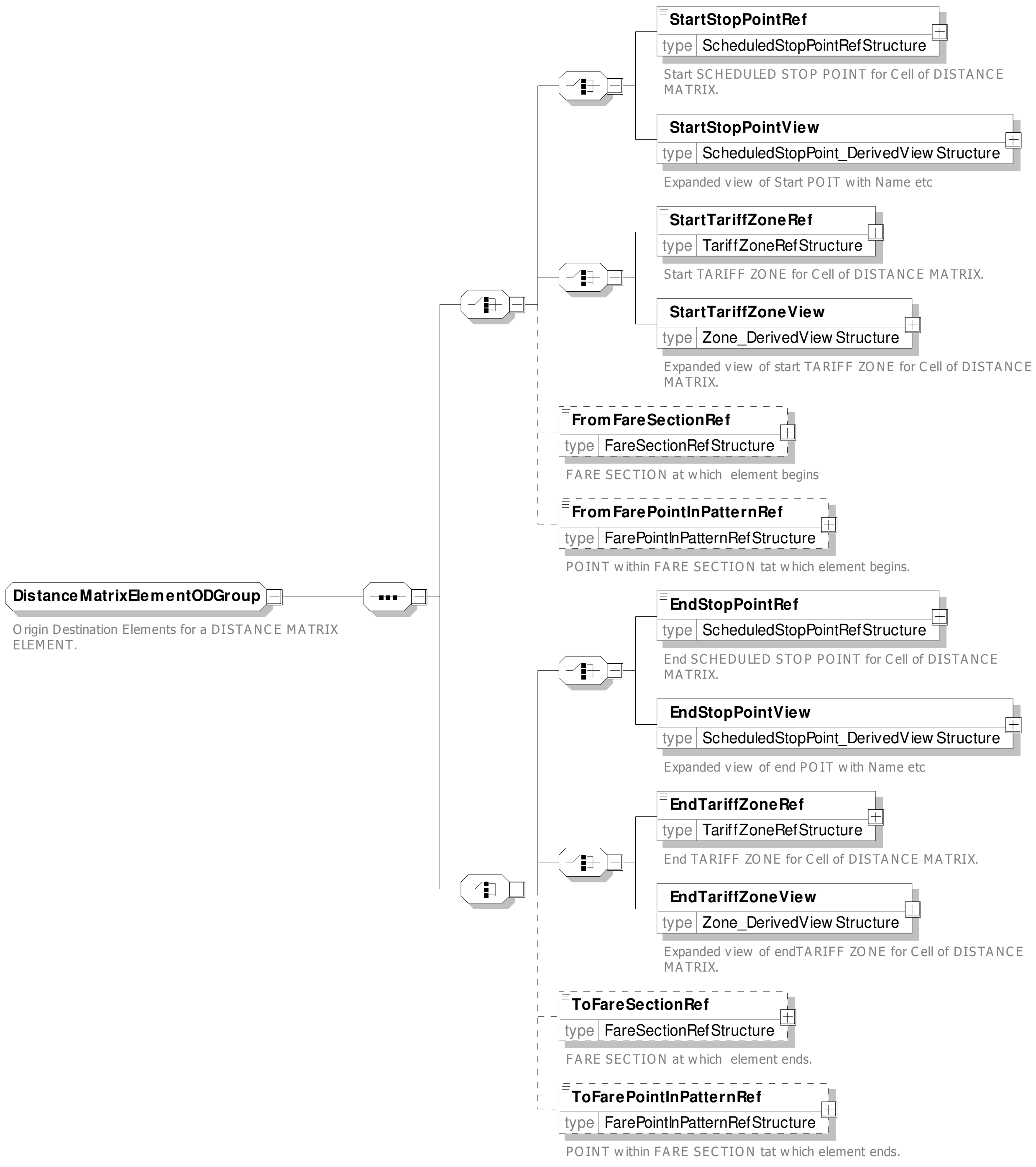


Figure 109 — *DistanceMatrixElementODGroup* — XSD

8.5.7.4.1.3 *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 73 – *DistanceMatrixElementComponentGroup* – Group

Classifi-	Name	Type	Cardinality	Description

cation				
“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.

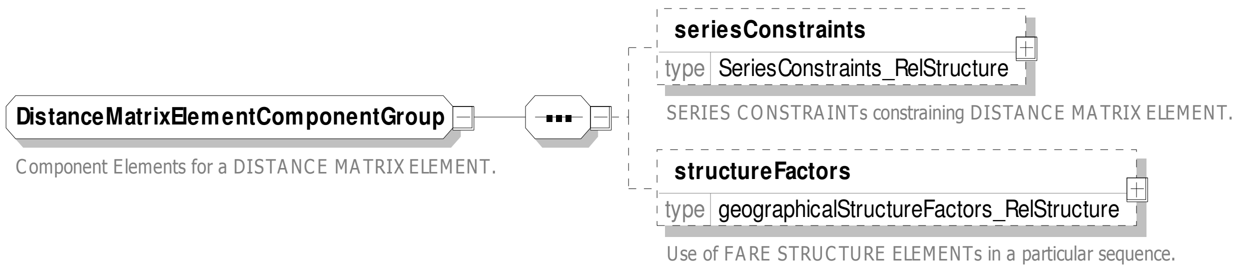


Figure 110 — *DistanceMatrixElementComponentGroup* — XSD

8.5.7.4.1.4 *DistanceMatrixElementPricingGroup* – Group

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

Table 74 – *DistanceMatrixElementPricingGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	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.
“cntd»	***Validity-Conditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs for element.
“cntd»	***Notice-Assignments***	*NoticeAsssignment*	0:*	NOTICE ASSIGNMENTs for DISTANCE MATRIX ELEMENT.
“cntd»	***tariffs***	*TariffRef*	0:*	TARIFFs for the DISTANCE MATRIX ELEMENT.
FK	***FareTableRef***	*FareTableRef*	0:1	Primary FareTable for the DISTANCE MATRIX ELEMENT.
“cntd»	***fareTables***	*FareTableRef*	0:*	FARE TABLEs for the DISTANCE MATRIX ELEMENT.
“cntd»	***prices***	*DistanceMatrix-ElementPrice*	0:*	Prices for the DISTANCE MATRIX ELEMENT.

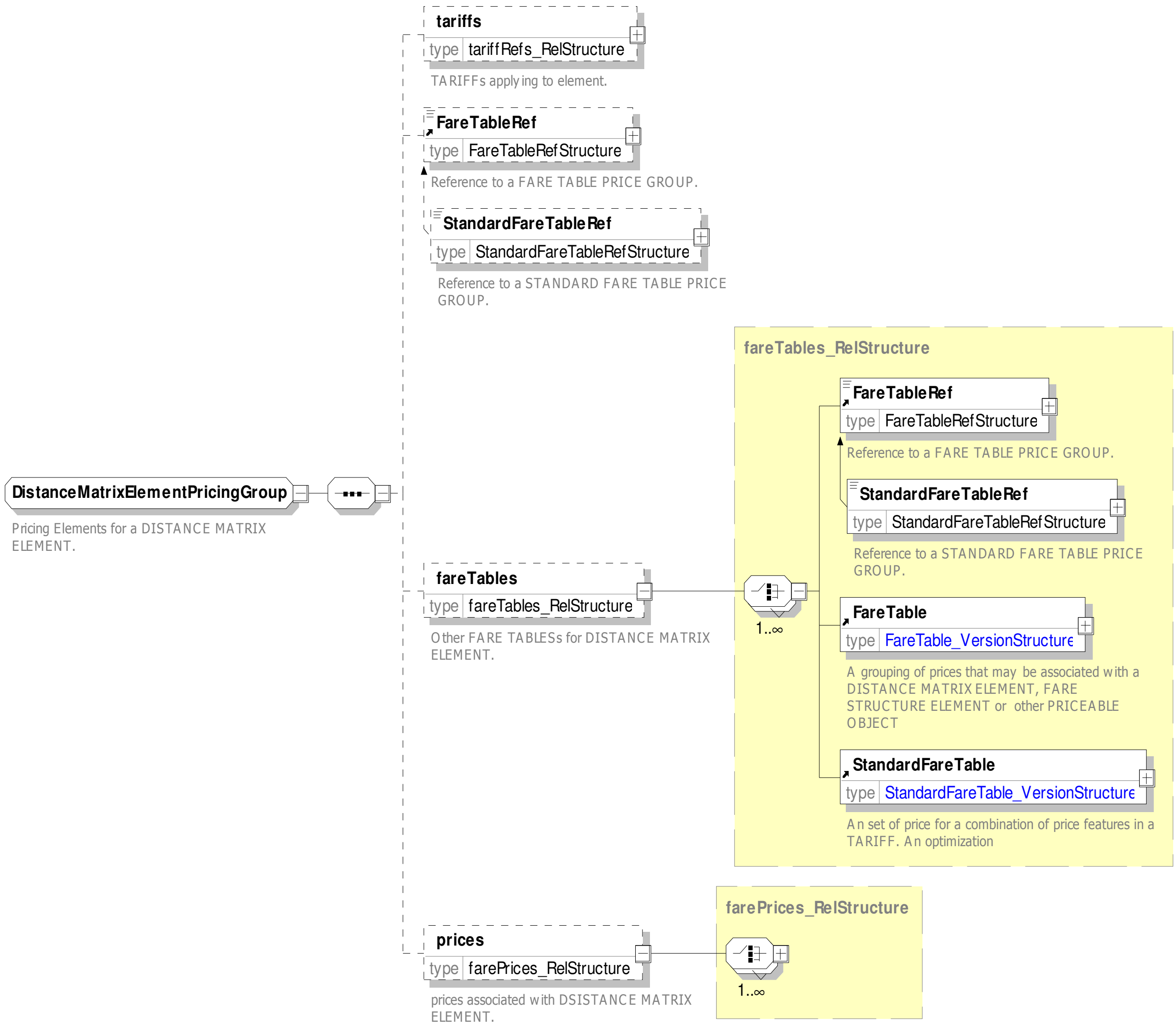


Figure 111 — *DistanceMatrixElementPricingGroup* — XSD

8.5.7.4.2 DistanceMatrixElementPrice – Model Element

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

Table 75 – *DistanceMatrixElementPrice* – Element

Classification	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»	***DistanceMatrix-ElementRef***	*DistanceMatrixElementRef*	0:1	DISTANCE MATRIX ELEMENT for which this is the price.
«FK»	***GroupOf-DistanceMatrix-***	*GroupOf-DistanceMatrixElements*	0:1	GROUP OF DISTANCE MATRIX ELEMENT for

	ElementsRef	*Ref*		which this is the price.

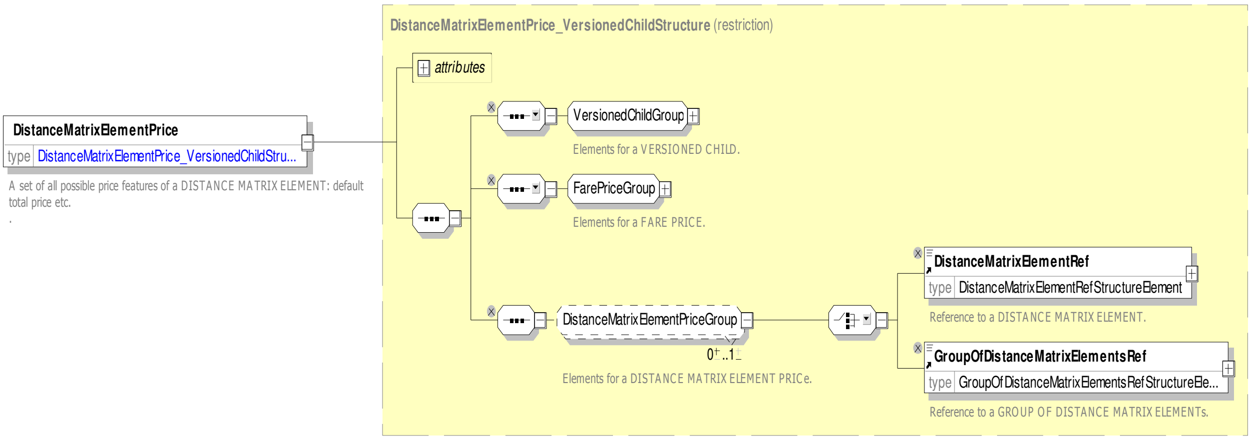


Figure 112 — *DistanceMatrixElementPrice* — XSD

8.5.7.4.3 GroupOfDistanceMatrixElements – Model Element

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

Table 76 – *GroupOfDistanceMatrixElements* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*GroupOfEntities*	::>	GROUP of DISTANCE MATRIX ELEMENT inherits from GROUP OF ENTITIES
«PK»	***id***	*GroupOfDistance-MatrixElementsIdType*	1:1	Identifier of GROUP of DISTANCE MATRIX ELEMENTs.
	Distance	*DistanceType*	0:1	Distance between origins and destinations of a DISTANCE MATRIX ELEMENT GROUP.
“cntd»	***validity-Conditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs for GROUP OF DISTANCE MATRIX ELEMENTs.
“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***	*DistanceMatrix-ElementPrice*	0:*	Prices for the GROUP OF DISTANCE MATRIX ELEMENTs.

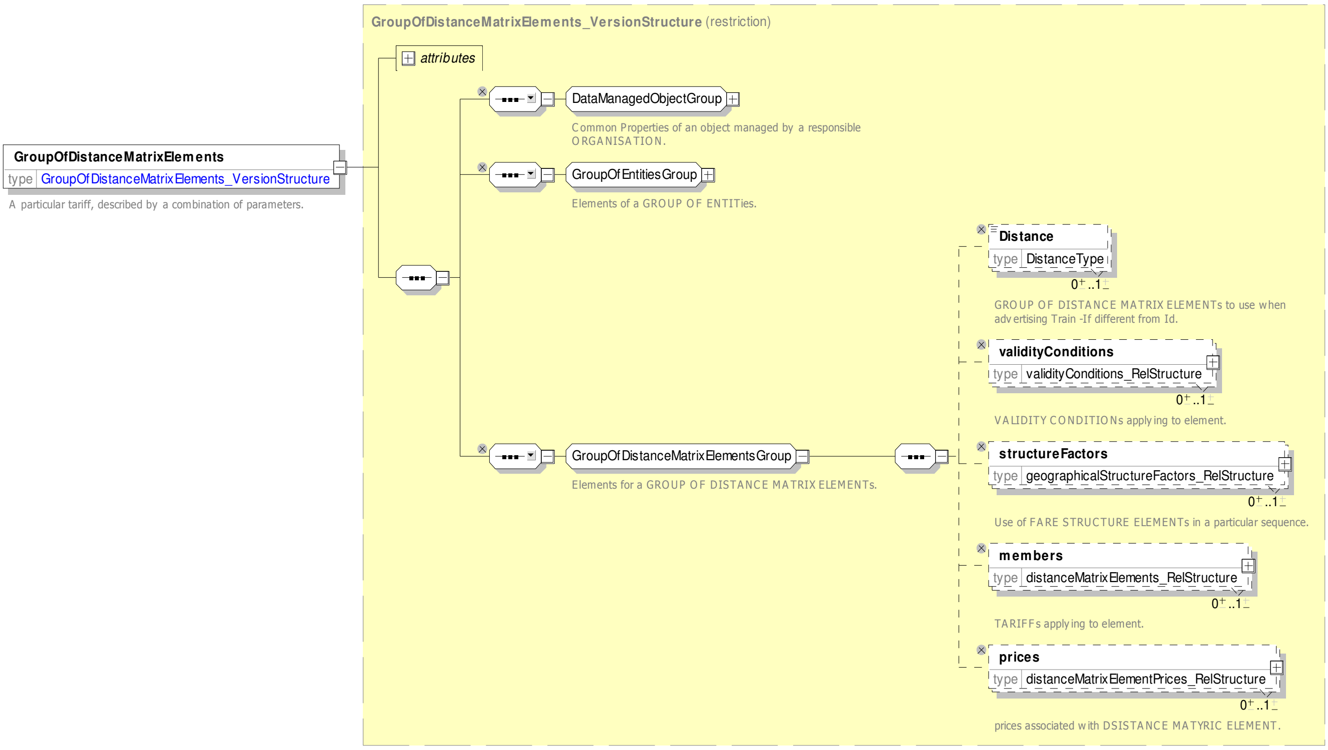


Figure 113 — *GroupOfDistanceMatrixElements* — XSD

8.5.7.5 Distance Matrix Element – XML examples

8.5.7.5.1 Distance Matrix Element: XML Example of Distance matrix elements

The following code fragment shows a GROUP OF DISTANCE MATRIX ELEMENTS with two DISTANCE MATRIX ELWMENTS 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>
```

8.5.7.5.2 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>
```

8.5.8 Validable & Controllable Elements

8.5.8.1 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 workds, 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 (see section 8.6.5.2).

8.5.8.2 Validable & Controllable Elements – Physical model

The following figure shows the physical model for VALIDABLE & CONTROLLABEL ELEMENTs.

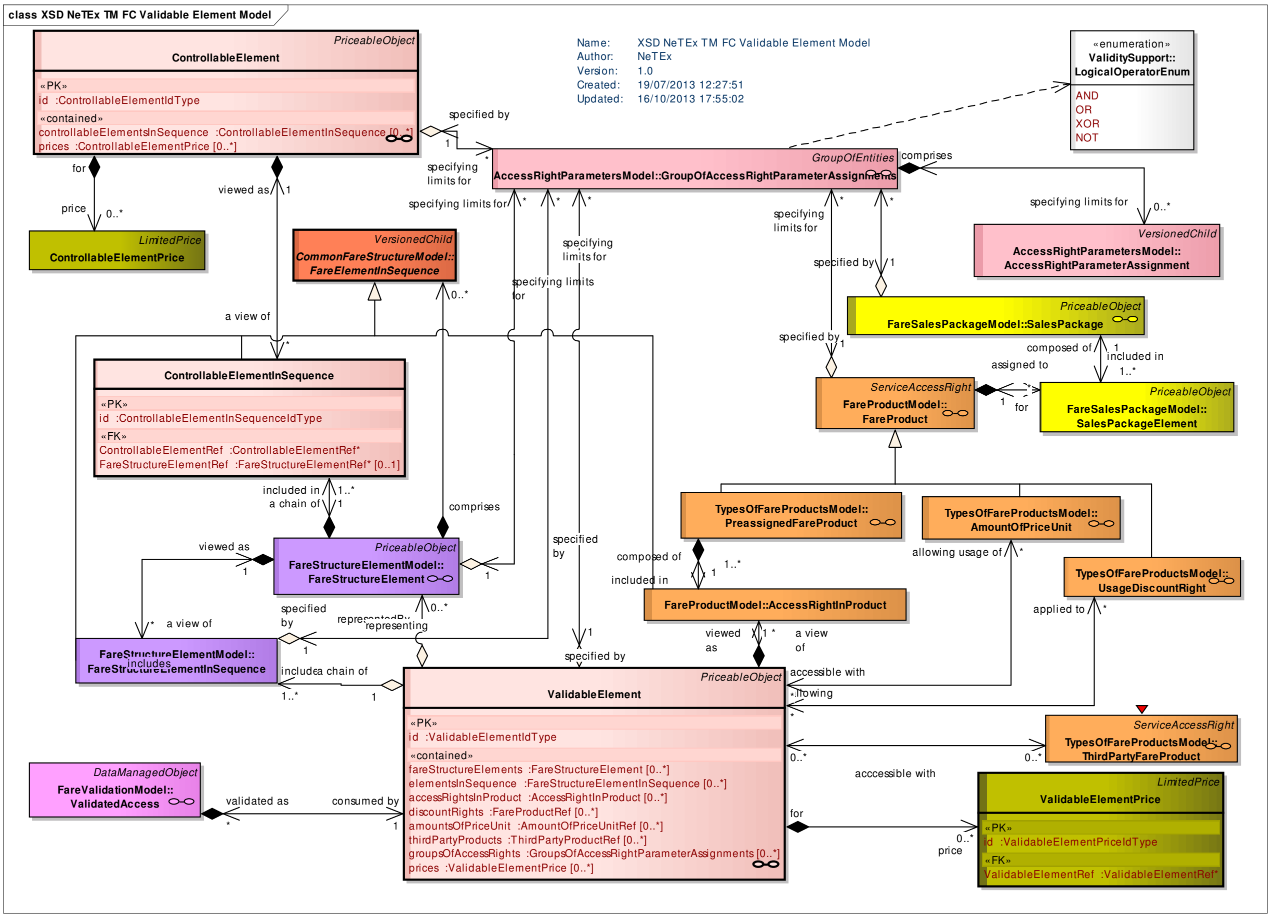


Figure 114 — Validable Element – Physical Model

8.5.8.3 Validable Element – Attributes and XSD

8.5.8.3.1 ValidableElement – Model Element

A sequence or set of FARE STRUCTURE ELEMENTs, grouped together to be validated in one go.

Table 77 – *ValidableElement* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	VALIDABLE ELEMENT inherits from PRICEABLE OBJECT***.***
«PK»	***id***	*ValidableElementIdType*	1:1	Identifier of VALIDABLE ELEMENT.
GROUP	***Validable-Element-StructureGroup***	*ValidableElement-StructureGroup*	1:1	Structure elements making up VALIDABLE ELEMENT.
GROUP	***Validable-ElementProduct Group***	*ValidableElement-ProductGroup*	1:1	Product elements making up VALIDABLE ELEMENT.
“cntd»	***prices***	*ValidableElementPrice*	0:*	VALIDABLE ELEMENT PRICEs for element.

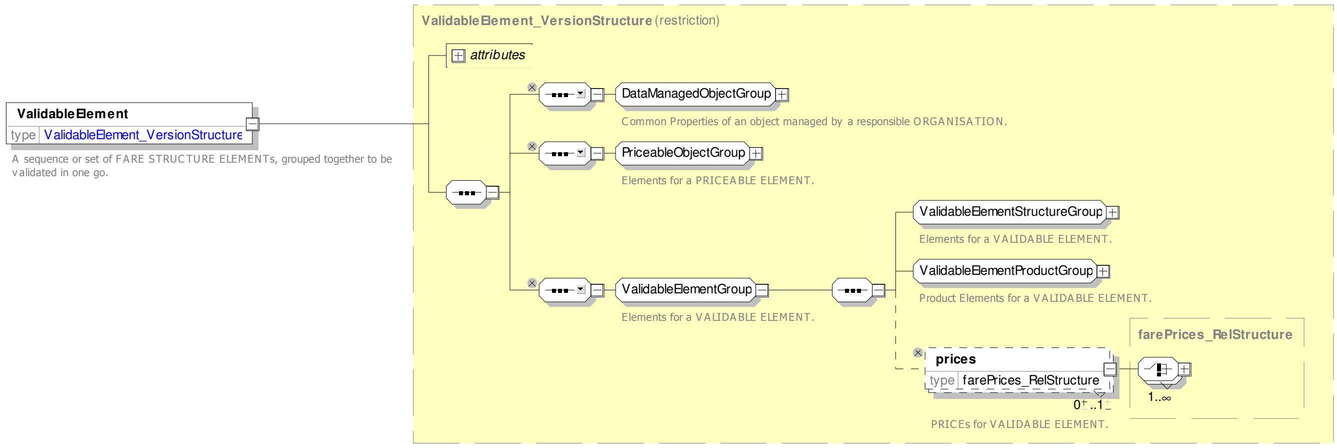


Figure 115 — *ValidableElement* — XSD

8.5.8.3.1.1 ValidableElementStructureGroup – Group

The ***ValidableElementStructureGroup*** define**s** fare structure elements comprising a VALIDABLE ELEMENT.

Table 78 – *ValidableElementStructureGroup* – Group

Classifi-cation	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.

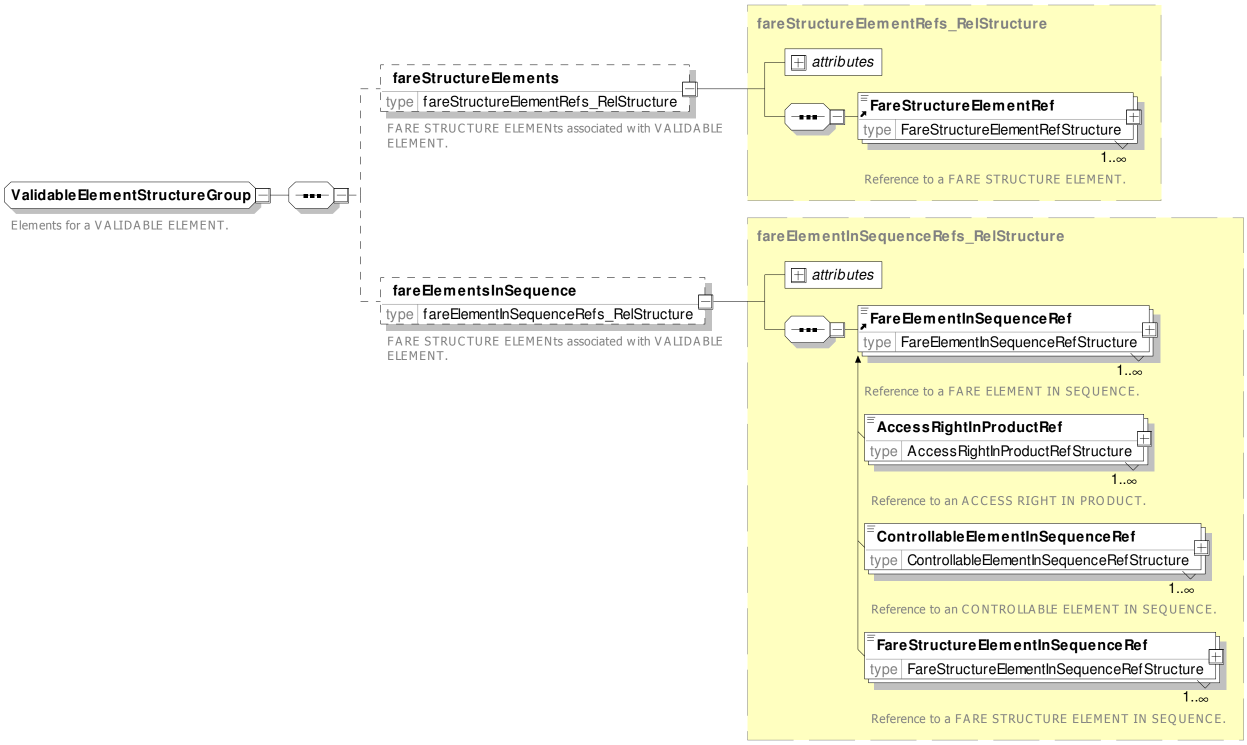


Figure 116 — *ValidableElementStructureGroup* — XSD

8.5.8.3.1.2 ValidableElementProductGroup – Group

The ***ValidableElementProductGroup*** define**s** FARE PRODUCT elements associated with a VALIDABLE ELEMENT.

Table 79 – *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-Unit**s*	*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.

*

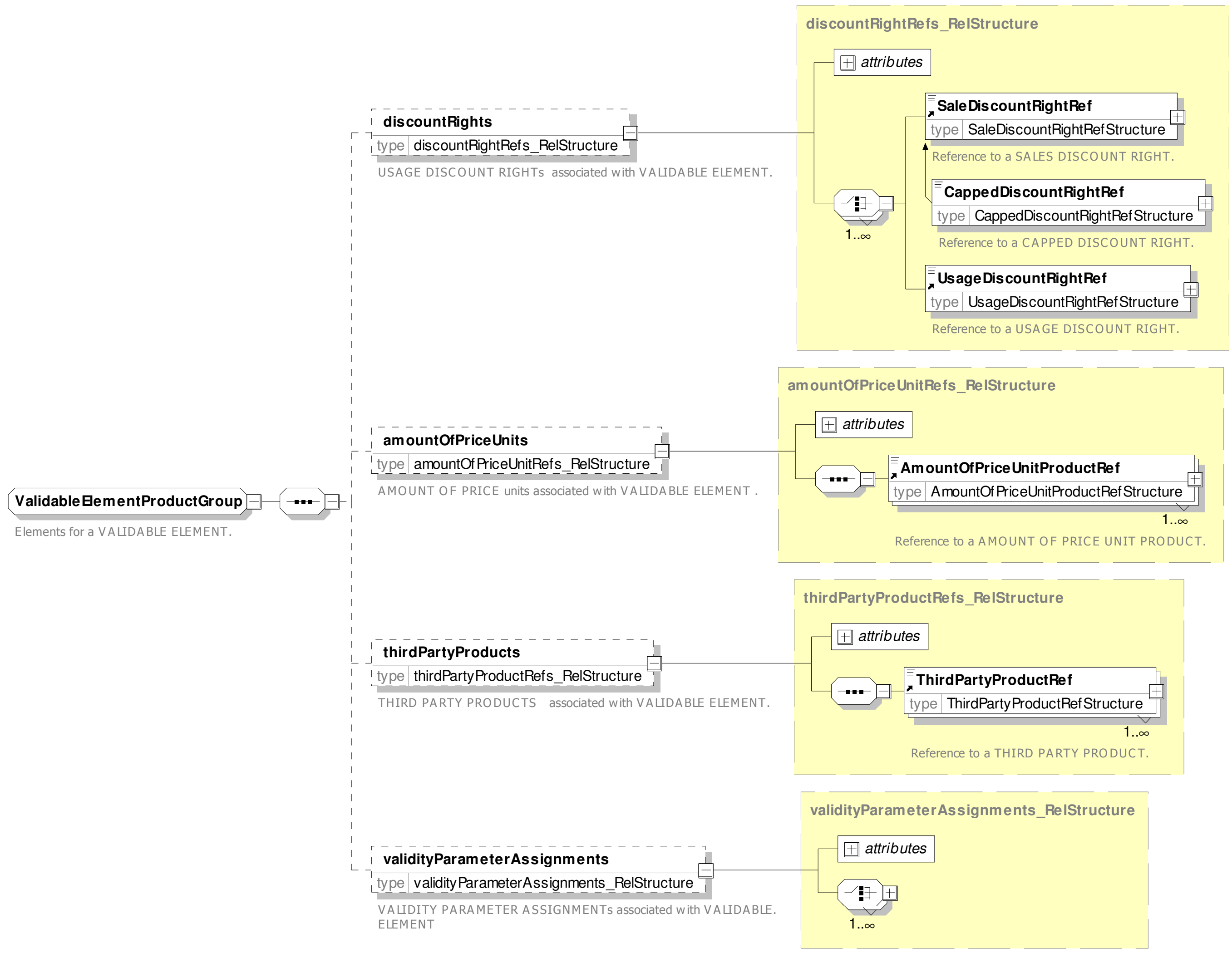


Figure 117 — *ValidableElementProductGroup* — XSD

8.5.8.3.2 ValidableElementPrice – Model Element

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

Table 80 – *ValidableElementPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	VALIDABLE ELEMENT PRICE inherits from FARE PRICE
«PK»	***id***	*ValidableElement-PriceIdType*	1:1	Identifier of VALIDABLE ELEMENT PRICE.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	1:1	Reference to a VALIDABLE ELEMENT.

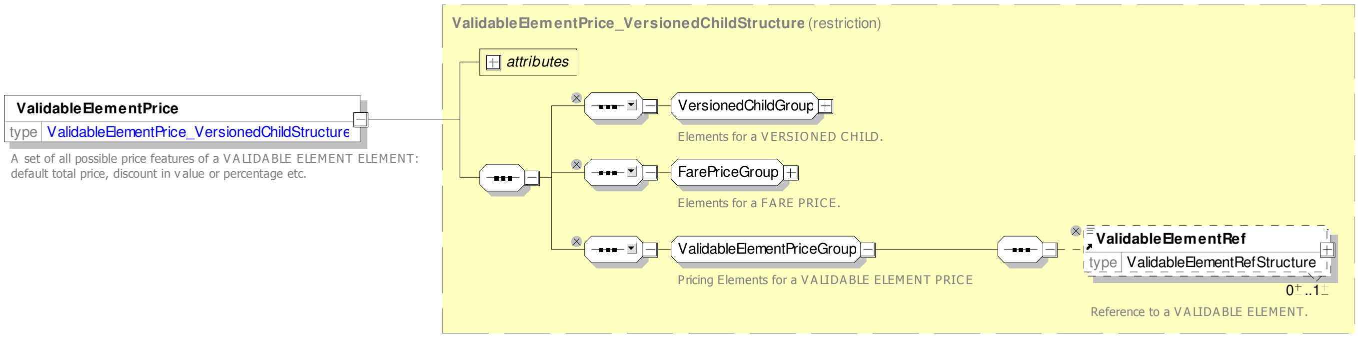


Figure 118 — *ValidableElementPrice* — XSD

8.5.8.3.3 ControllableElement – Model Element

The smallest controllable element of public transport consumption, all along which any VALIDITY PARAMETER ASSIGNMENT remains valid.

Table 81 – *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***	*AccessRightParameter-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.

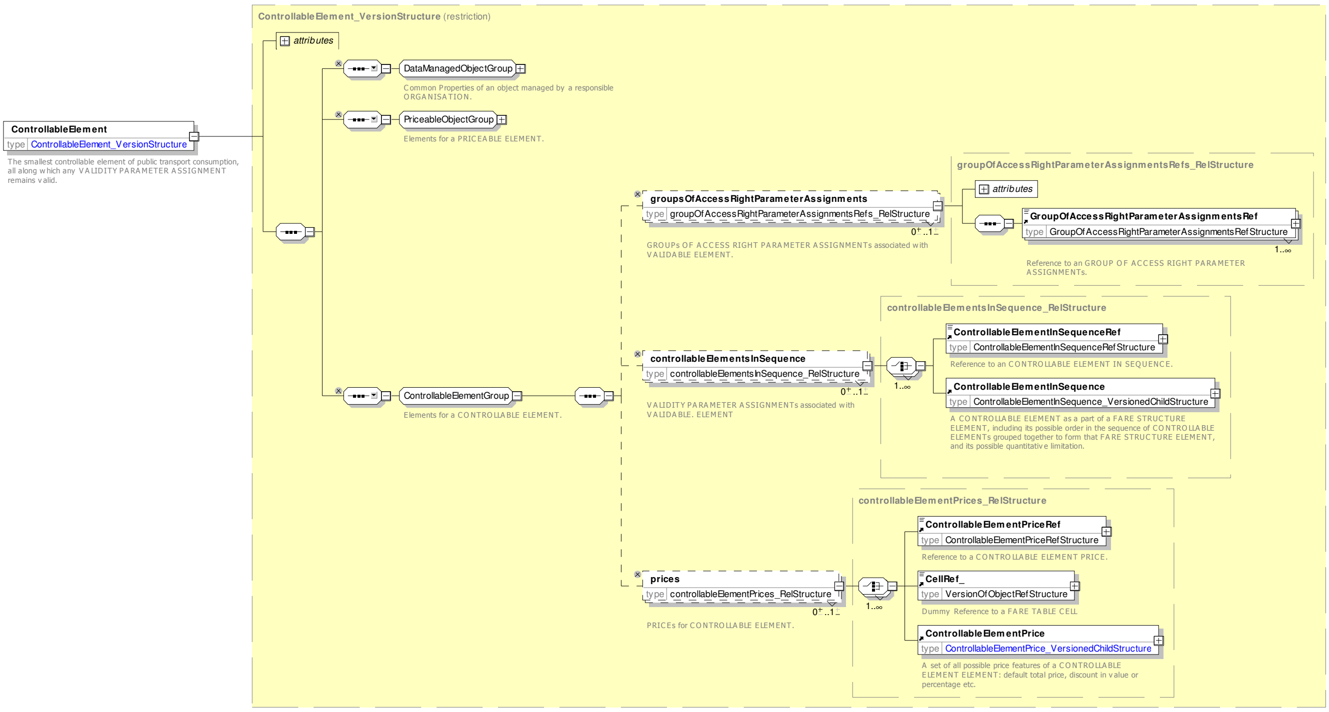


Figure 119 — *ControllableElement* — XSD

8.5.8.3.4 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 82 – *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***	*AccessRightParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs associated with CONTROLLABLE ELEMENT IN SEQUENCE.

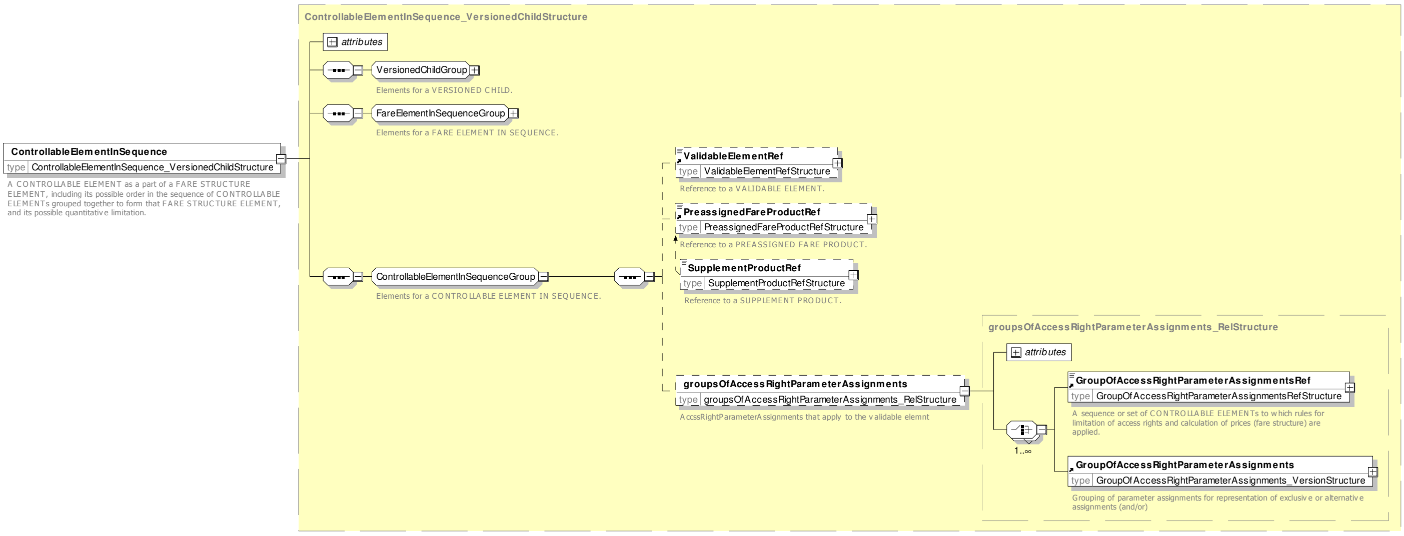


Figure 120 — *ControllableElementInSequence* — XSD

8.5.8.3.5 ControllableElementPrice – Model Element

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

Table 83 – *ControllableElementPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	CONTROLLABLE ELEMENT PRICE inherits from FARE PRICE
«PK»	***id***	*ControllableElement-PriceIdType*	1:1	Identifier of CONTROLLABLE ELEMENT PRICE.
«FK»	***Controllable-ElementRef***	*ControllableElement-IdType*	0:1	Reference to a CONTROLLABLE ELEMENT.

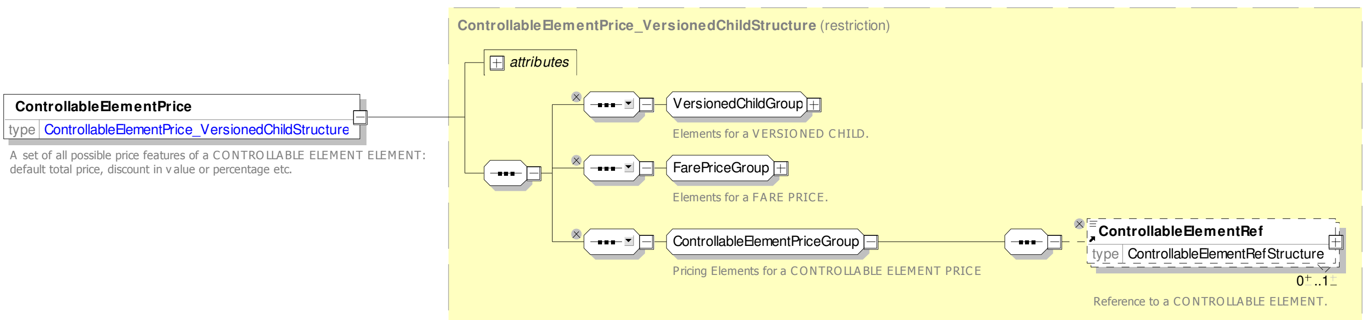


Figure 121 — *ControllableElementPrice* — XSD

8.5.8.4 Validable & Controllable Elements – XML examples

8.5.8.4.1 Validable & Controllable Elements: XML Example of a Tariff for a sequence of Fare Structure Elements

The following code fragment shows some basic VALIABLE ELEMENTs for different modes.

For EXAMPLE:

```
<validableElements>
    <ValidableElement version="any" id="lul::metroTrip">
        <Name>A metro Trip</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="lul::metroTrip">
                <Scope><VehicleModes>metro</VehicleModes></Scope>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </ValidableElement>
    <ValidableElement version="any" id="lul::metroPartTrip">
        <Name>A metro Trip between validators - use Pink reader</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="lul::metroPartTrip">
                <Scope><VehicleModes>metro</VehicleModes></Scope>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </ValidableElement>

    <ValidableElement version="any" id="nr::suburbanRailTrip">
        <Name>A rail Trip</Name>
        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="nr::suburbanRailTrip">
                <Scope>
                    <VehicleModes>rail</VehicleModes>
                        <TransportSubmode>
                            <RailSubmode>local</RailSubmode>
                        </TransportSubmode>
                </Scope>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </ValidableElement>
```

{TO DO ADD MORE COMPLEX EAMPLES]

8.6 Access Rights Description

The following diagram shows the models making up the Access Right Description submodel.

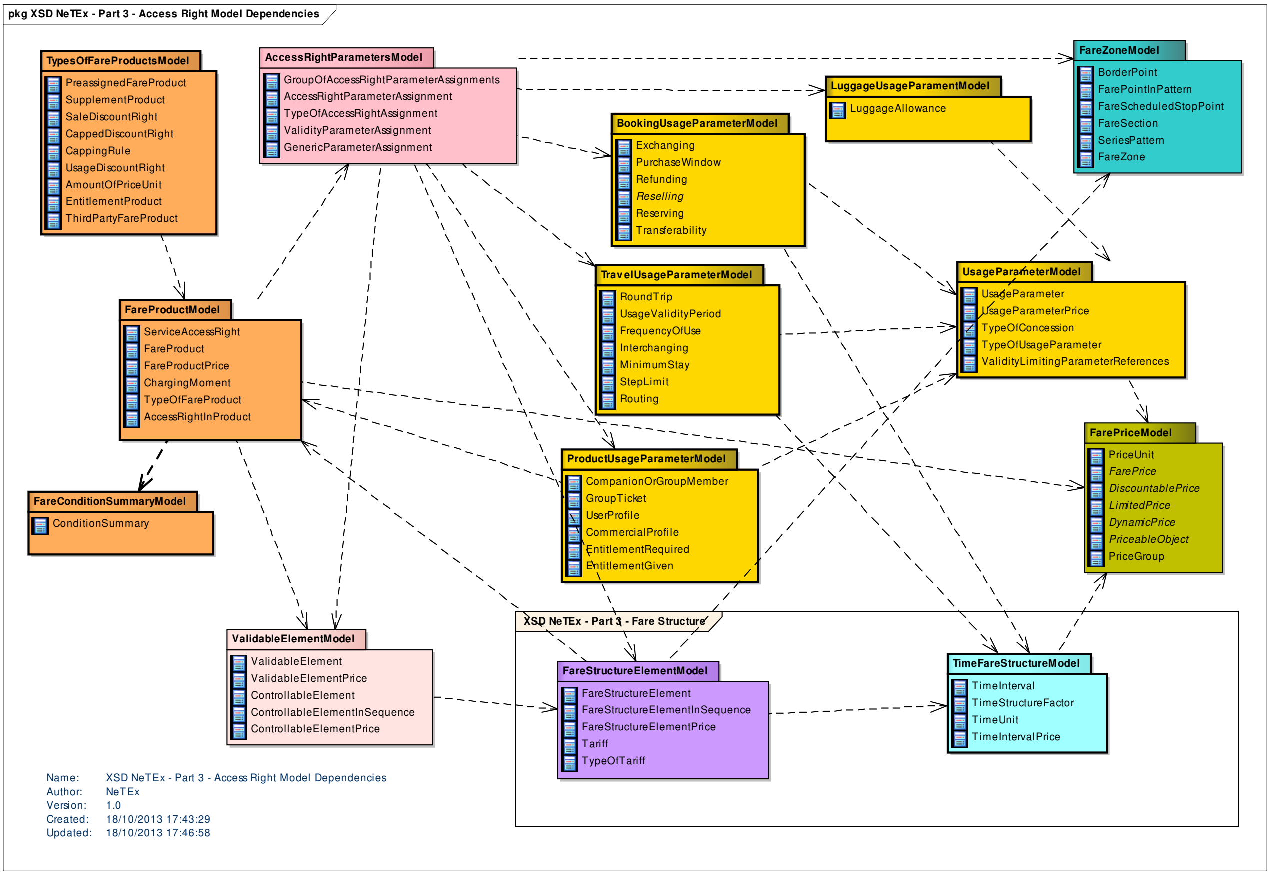


Figure 122 — Access Right Parameters – Model Dependencies

8.6.1 Access Right Parameters

8.6.1.1 Access Rights Parameters

8.6.1.1.1 Access Rights Parameters – Conceptual model

The ACCESS RIGHT PARAMETERs Model allows specific ACCESS RIGHTs to be associated with a group of parameters. The core element is the ACCESS RIGHT PARAMETER ASSIGNMENT which assigns a set of limitation 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, FARE PRODUCT, SALES PACKAGE, VALIDABLE ELEMENT, or CONTROLLABLE ELEMENT.

8.6.1.2 Limiting Fare Parameters and Their Assignment

Apart from the quantitative parameters used in the fare structure, other parameters may be used by a fare system in order to limit the validity of some access rights. The definition of theoretical access rights or consumption controls, for instance, uses parameters referring to physical characteristics.

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, conceptually (and correspond to the real world process). The assignment of such parameters to an element of the fare system is therefore described using a generic entity ACCESS RIGHT PARAMETER ASSIGNMENT.

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 includes two subtypes:

— A GENERIC PARAMETER ASSIGNMENT, which attaches a fixed parameter to a certain class of rights and

— A SPECIFIC PARAMETER ASSIGNMENT, which assigns a limiting parameter to a particular right, within a certain fare structure.

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. They express that the compared characteristic, e.g.:

— is greater than or equal to the parameter ("start" parameter), e.g. the consumption should start from the SCHEDULED STOP POINT "Central station" or from any further SCHEDULED STOP POINT in the JOURNEY PATTERN;

— is equal or smaller than the parameter ("end" parameter), e.g. the consumption has to end before "11.00 p.m.";

— 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.

— 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.'

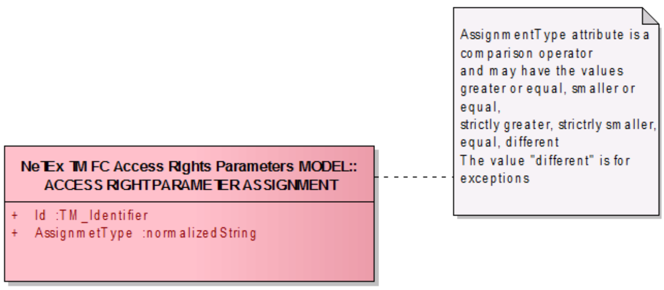


Figure 123 — ACCESS RIGHT ASSIGNMENTs[KB15][KB16]

In general, access right limitation rules are 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).

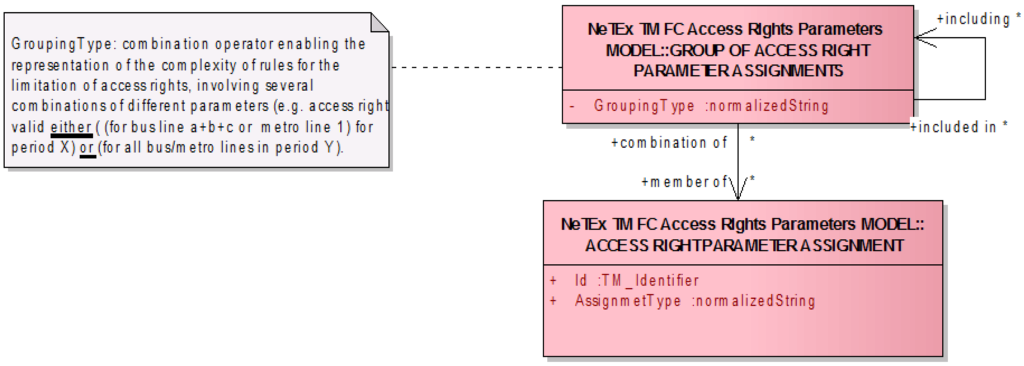


Figure 124 — GROUP OF ACCESS RIGHT ASSIGNMENTS[KB17]

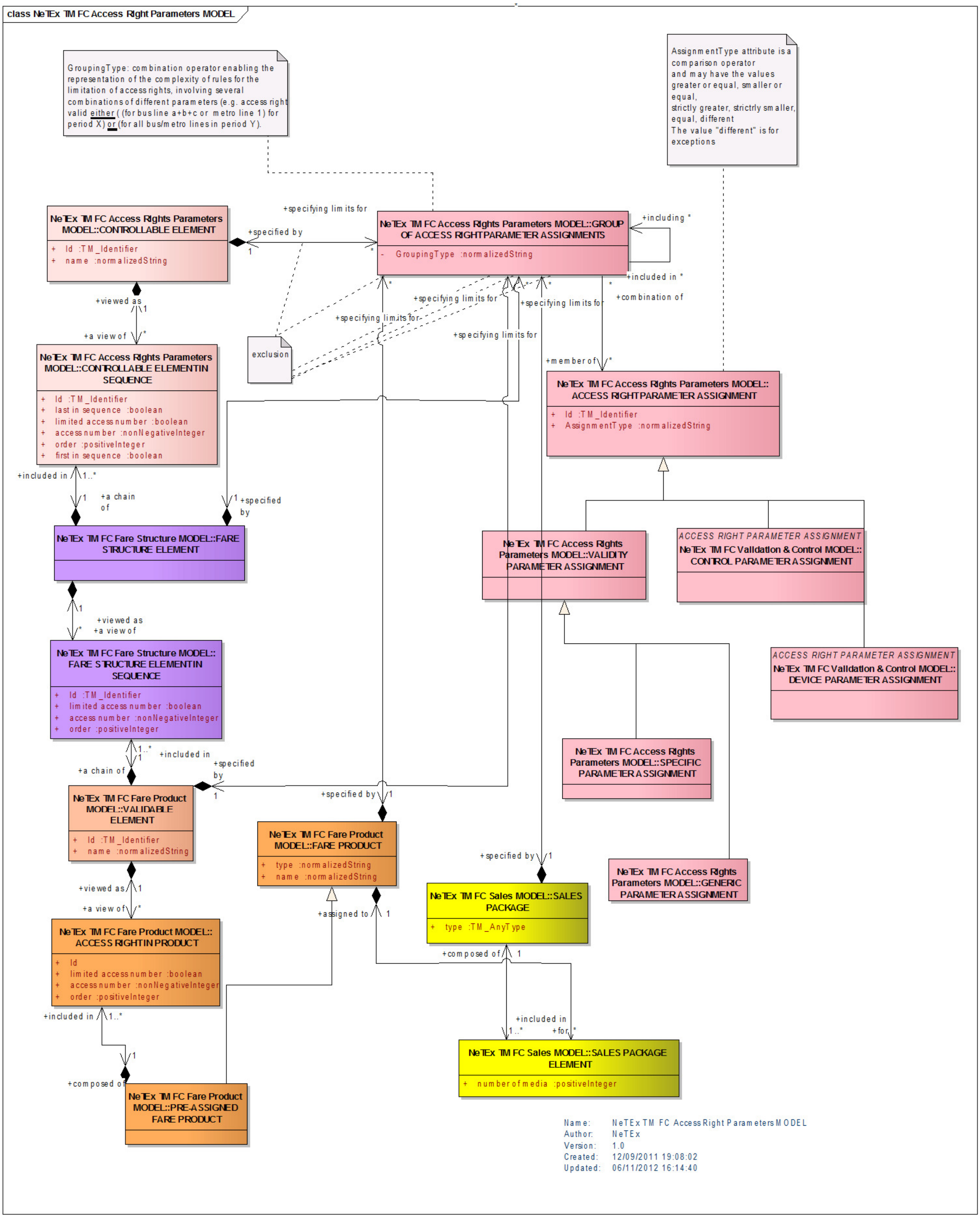


Figure 125 —ACCESS RIGHT PARAMETER ASSIGNMENT

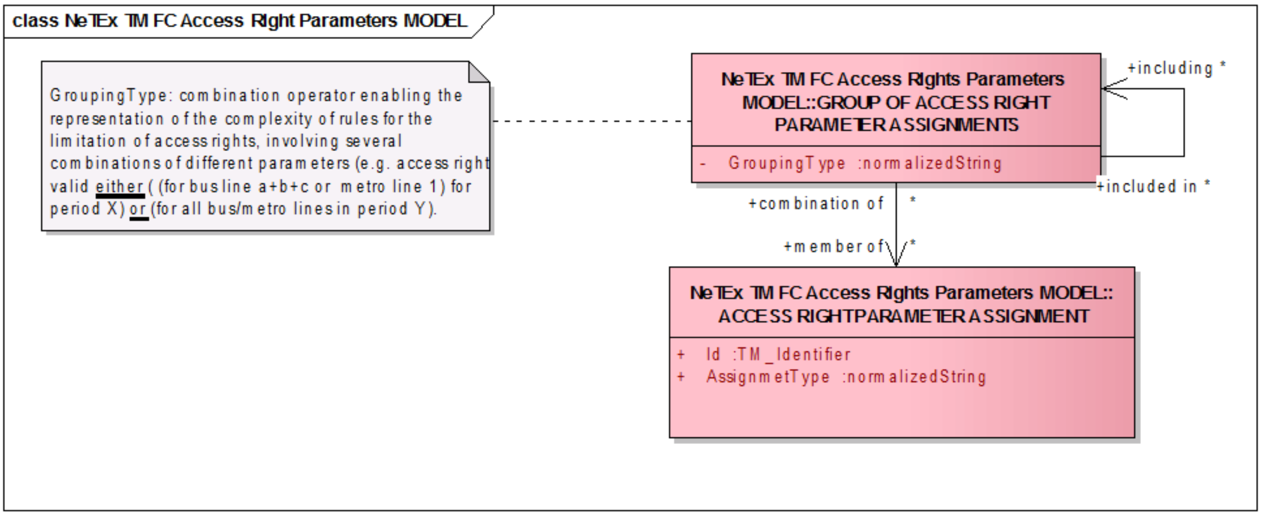


Figure 126 —ACCESS RIGHTs[KB18]

8.6.1.2.1 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 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.

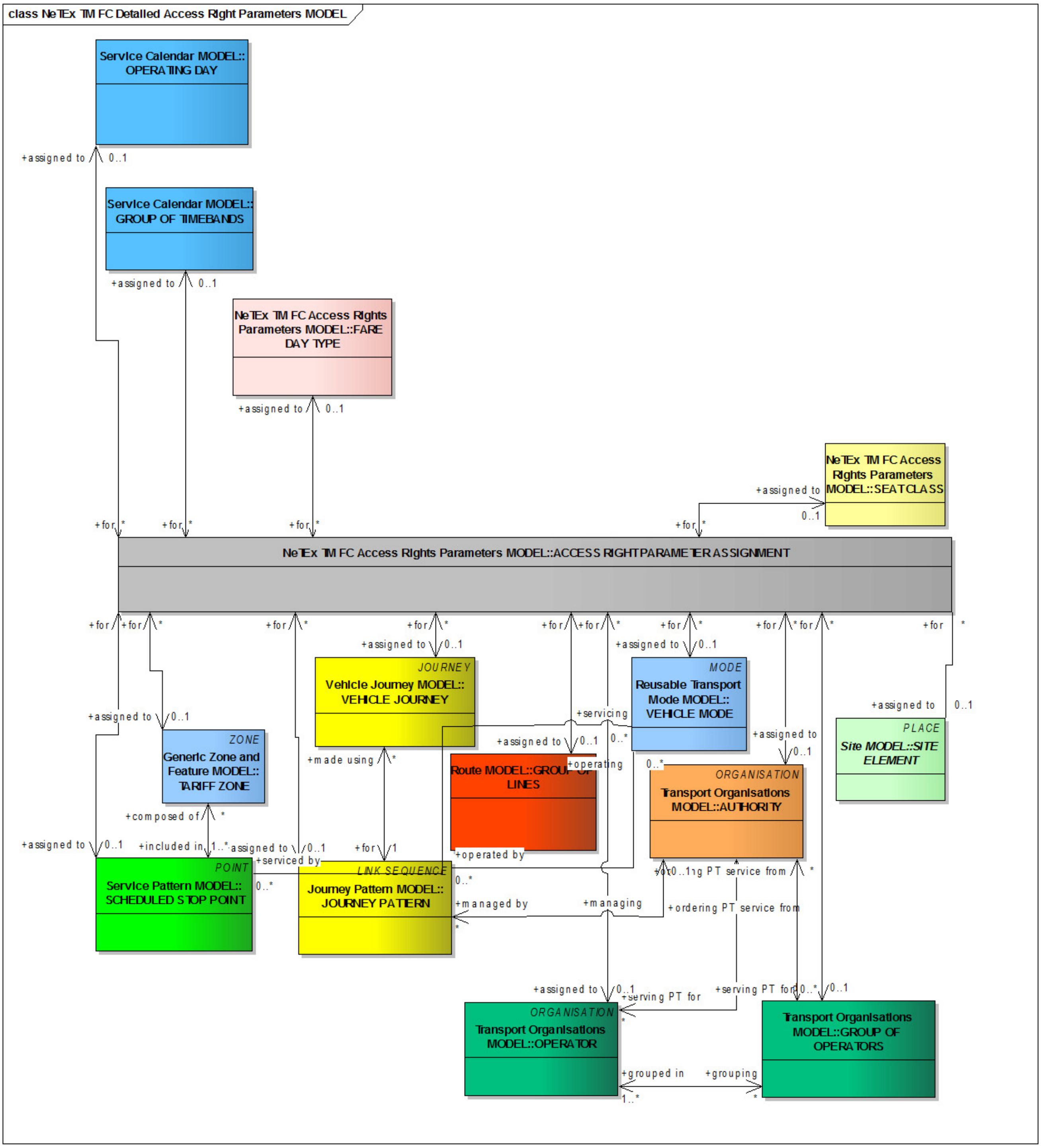


Figure 127 — Validity Parameters

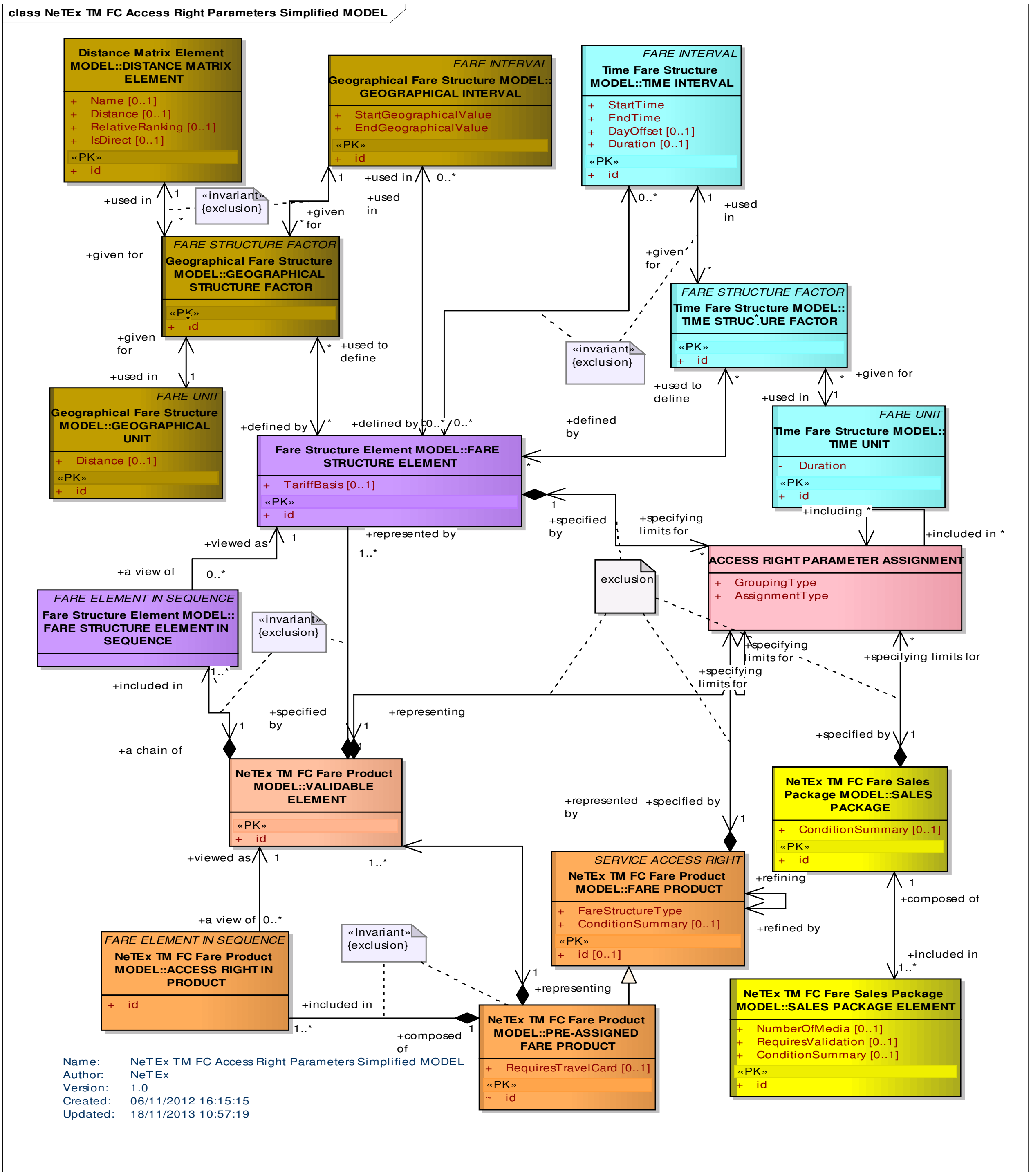


Figure 128 — Access Right Parameters Simplified – Conceptual Model[KB19]

8.6.1.2.1.1 Access Rights Parameter Assignment – Conceptual model

The ACCESS RIGHT PARAMETER ASSIGNMENTs may involve the compound assignment of several assignments either ANDed or ORed together.

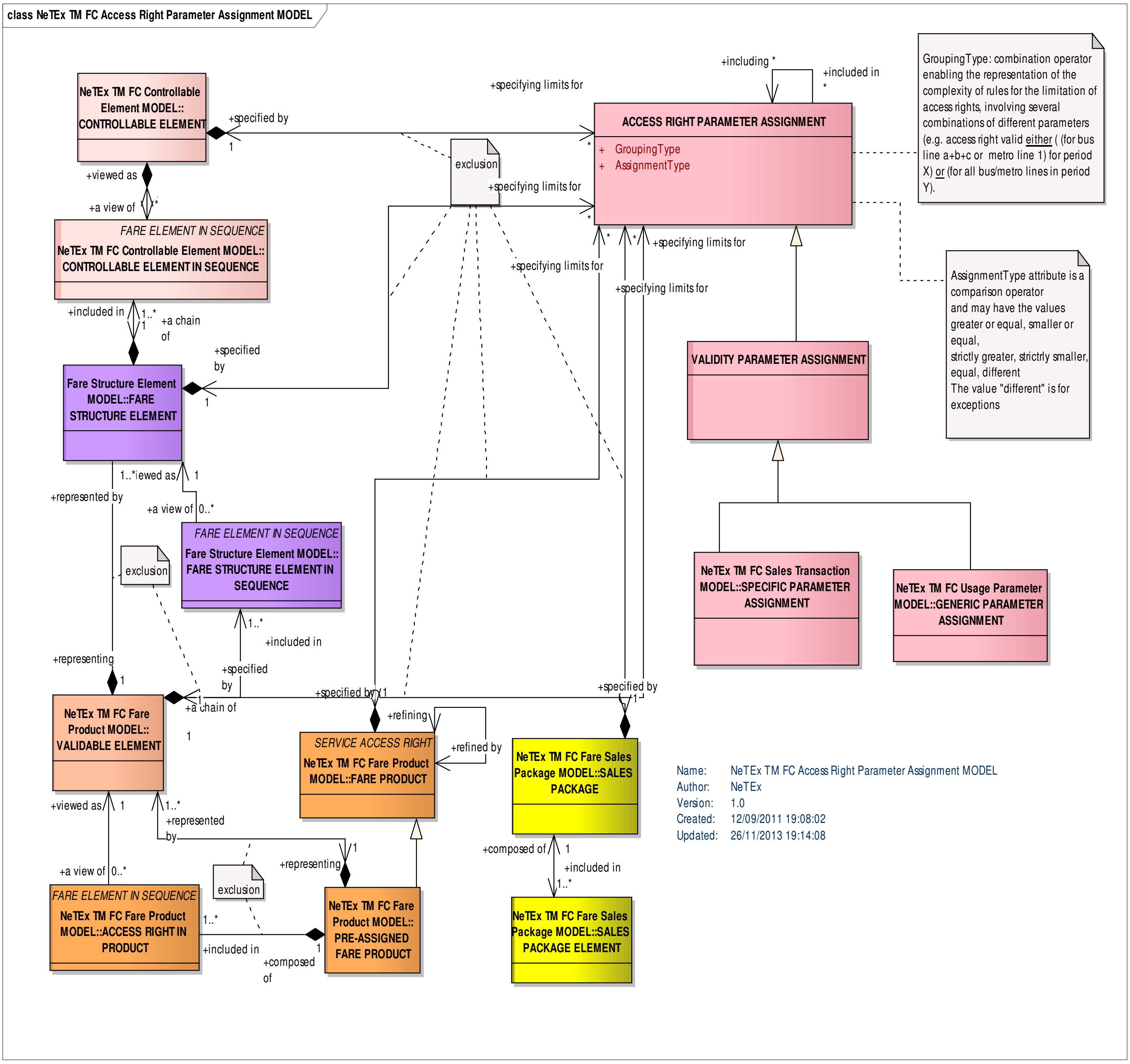


Figure 129 — Access Right Parameter Assignment – Conceptual Model[KB20]

8.6.1.2.1.2 Access Rights Parameter Assignment Scope– Conceptual model

The ACCESS RIGHT PARAMETER ASSIGNMENTs may assign to a large variety of different scope elements assignment of several assignments either ANDed or ORed together.

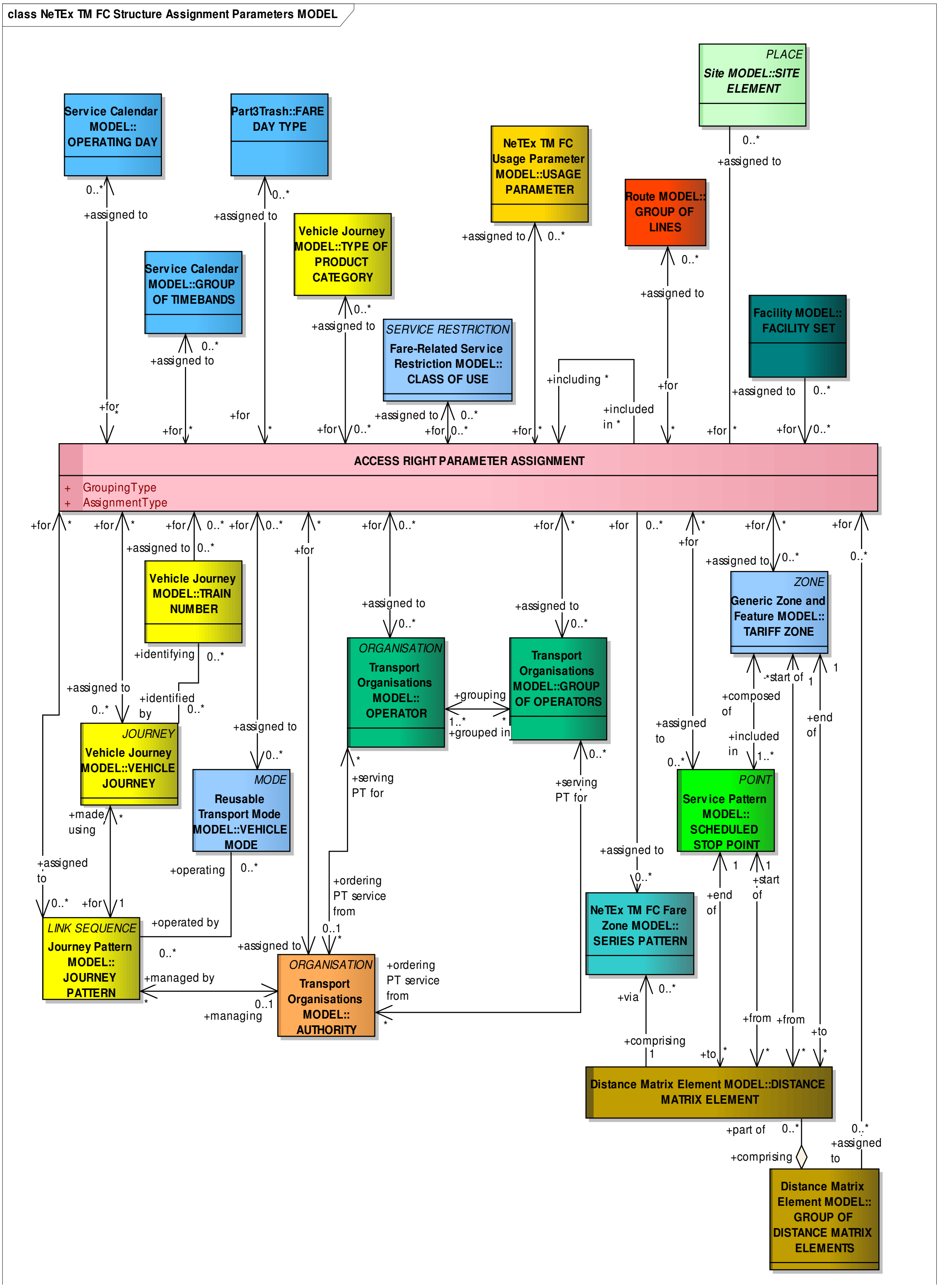


Figure 130 —Validity Parameters – Conceptual Model[KB21]

8.6.1.2.2 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 scope") (b) when they may be used ("temporal scope") (c) The conditions and limitations on use (USAGE PARAMETERs).

Assignments may be made to any of FARE STRUCTURE ELEMENTs, FARE STRUCTURE ELEMENTs in SEQUENCE, VALIDABLE ELEMENTs, CONTROLLABLE ELEMENTs, FARE PRODUCTs and SALES PACKAGEs.

Each assignments may restrict the access rights to a particular temporal scope (DAY TYPE, VALIDITY CONDITION, etc) and to one or more access rights to the network or services (LINE, TARIFF ZONEs, SCHEDULED STOP POINTs, VEHICLE JOURNEYS, 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 types of ACCESS RIGHT ASSIGNMENT for use in specific circumstances.

— A GENERIC PARAMETER ASSIGNMENT is used in a FARE PRODUCT and SALES PACKAGE to indicate a 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 used in an actual materialization of a fare product as a SALES TRANSACTION.

— DEVICE and CONTROL PARAMETER ASSIGNMENTs are used in FARE COLLECTION systems.

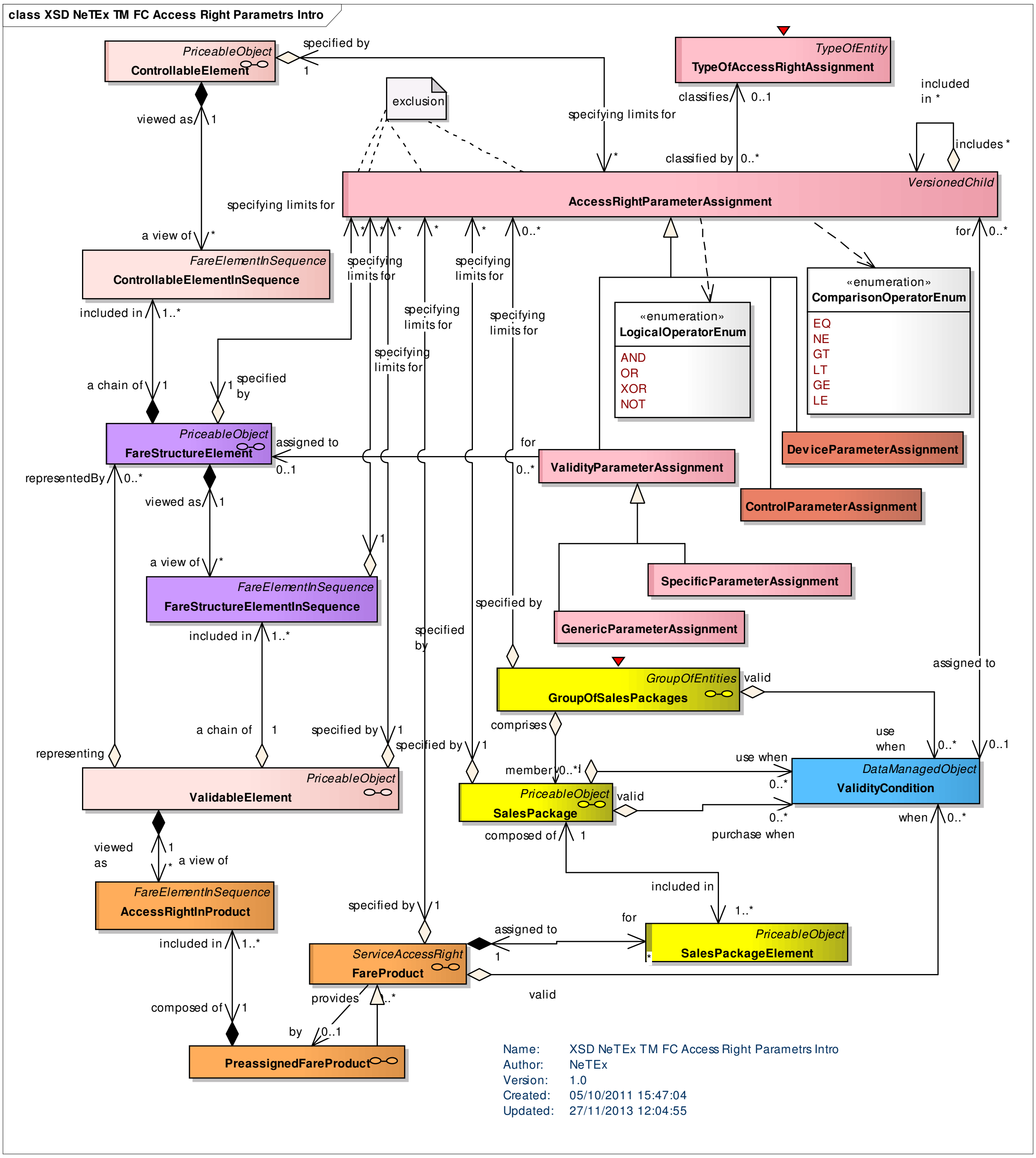


Figure 131 — Access Right Parameters – Physical Model

8.6.1.2.2.1 Access Rights Parameters: Detail – Physical model

The following figure summarises the basic physical model for ACCESS RIGHT PARAMETER ASSIGNMENTs. It omits the scope details (see below).

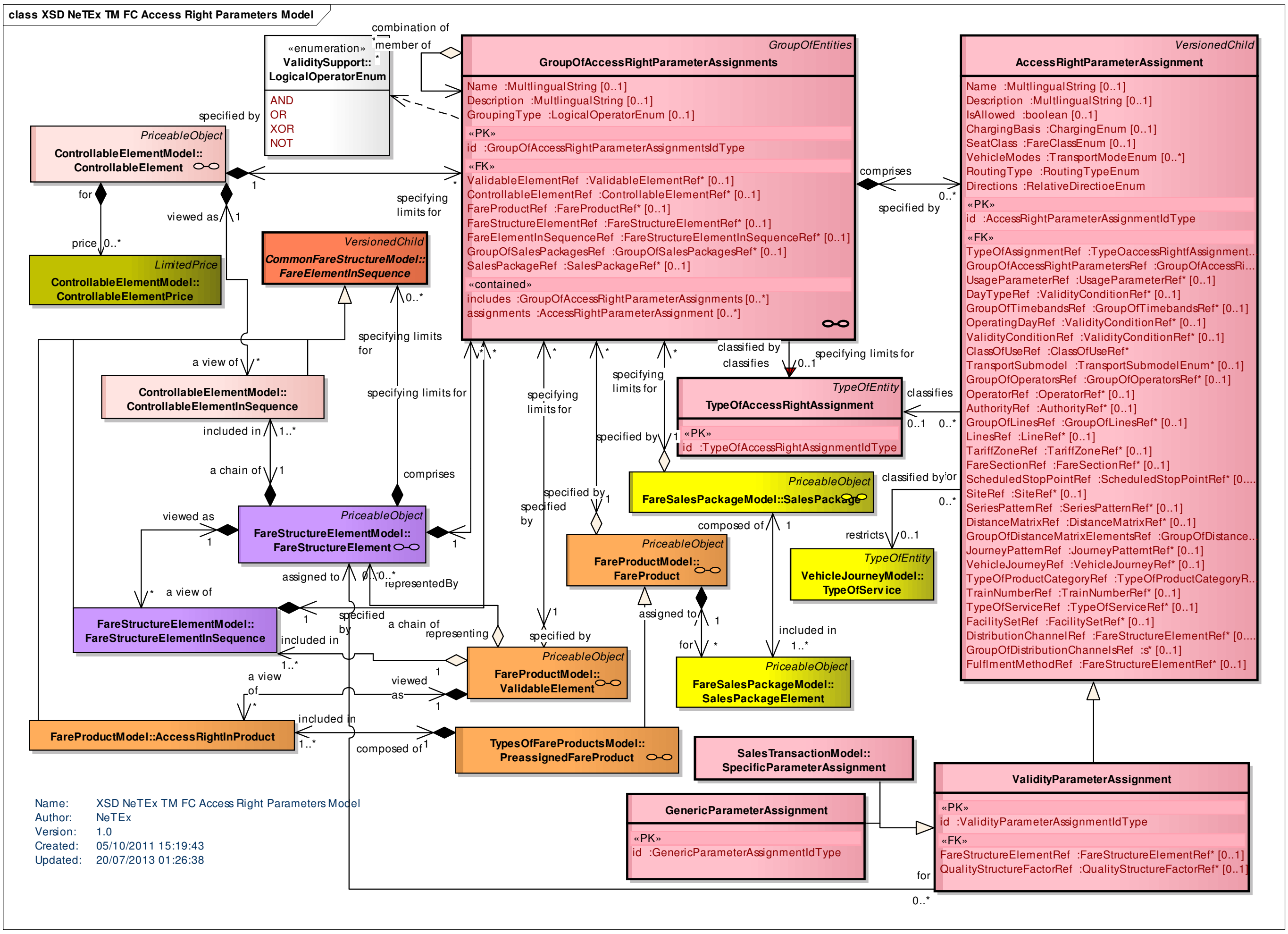


Figure 132 — Access Right Parameters: Detail – Physical Model

8.6.1.2.2.2 Access Rights Parameters: Scope Summary – Physical model

The following figure summarises the scope parameters which may be assigned using an ACCESS RIGHT PARAMETER ASSIGNMENT. These may be grouped into two may sets: a temporal assignment scope (DAY TYPE, VALIDITY CONDITION etc) and a network assignment scope (CLASS OF USE, Network, OPERATOR, DISTANCE MATRIX ELEMENT, SERIES CONSTRAINT, etc,. etc.).

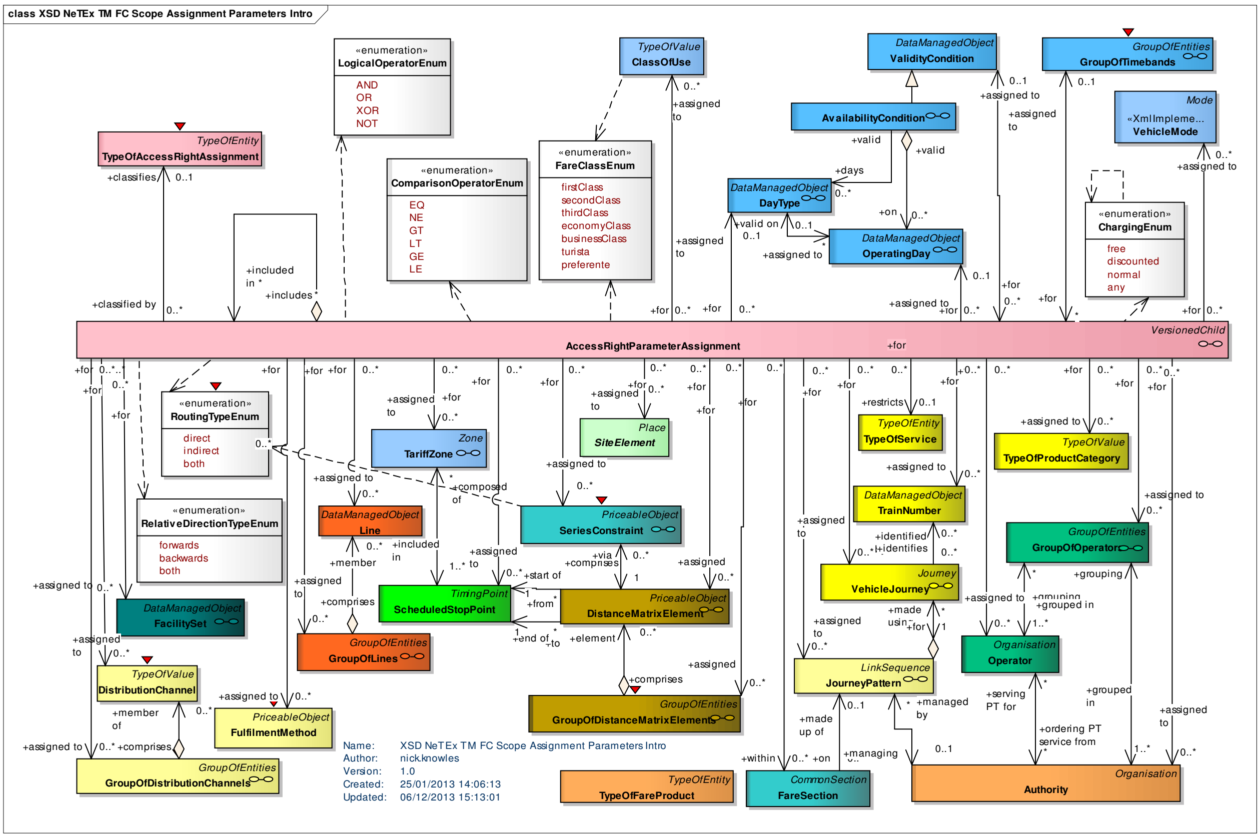


Figure 133 — Access Right Parameters: Scope Parameters – Physical Model

8.6.1.2.2.3 Access Rights Parameters: Network Scope – Physical model

The following figure shows the network scope 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.

- Which OPERATORs or GROUPs of OPERATORs may be used.
- Which LINEs, GROUPS OF LINEs or NETWORKs, and VEHICLE MODEs may be used.
- Which TARIFF ZONEs and or which SCHEDULED STOP POINTs may be used.
- Which point to point journeys (DISTANCE MATRIX ELEMENT) may be used and any SERIES CONSTRAINTs on the routes between them.
- 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.
- Which DISTRIBUTION CHANNELs and FULFILMENT METHODs may be used.

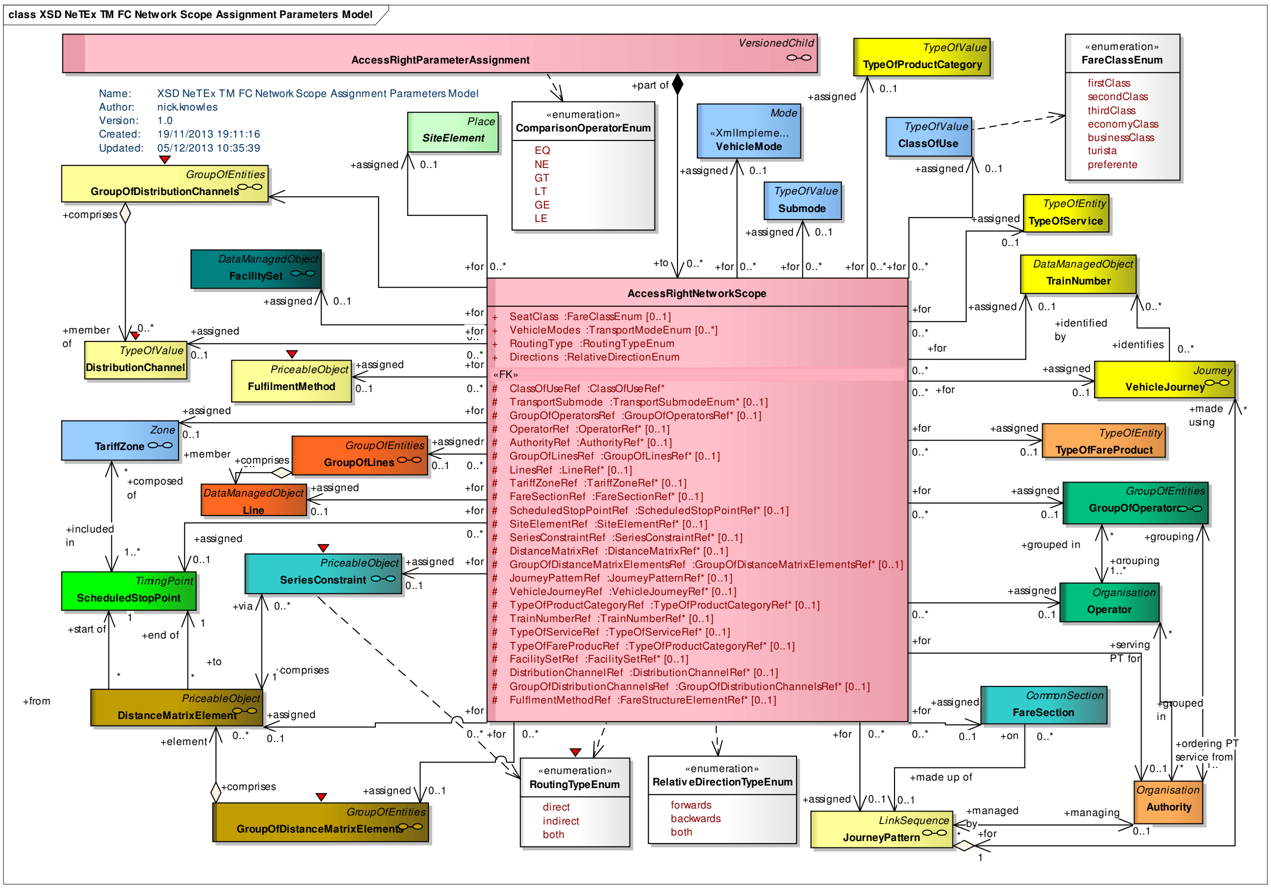


Figure 134 — Access Right Parameters: Network Scope Parameters – Physical Model

8.6.1.2.2.4 Access Rights Parameters: Temporal Scope – Physical model

The following figure shows the temporal scope 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 VALDITY CONDITION or AVAILABILITY CONDITION restricting the assignment.

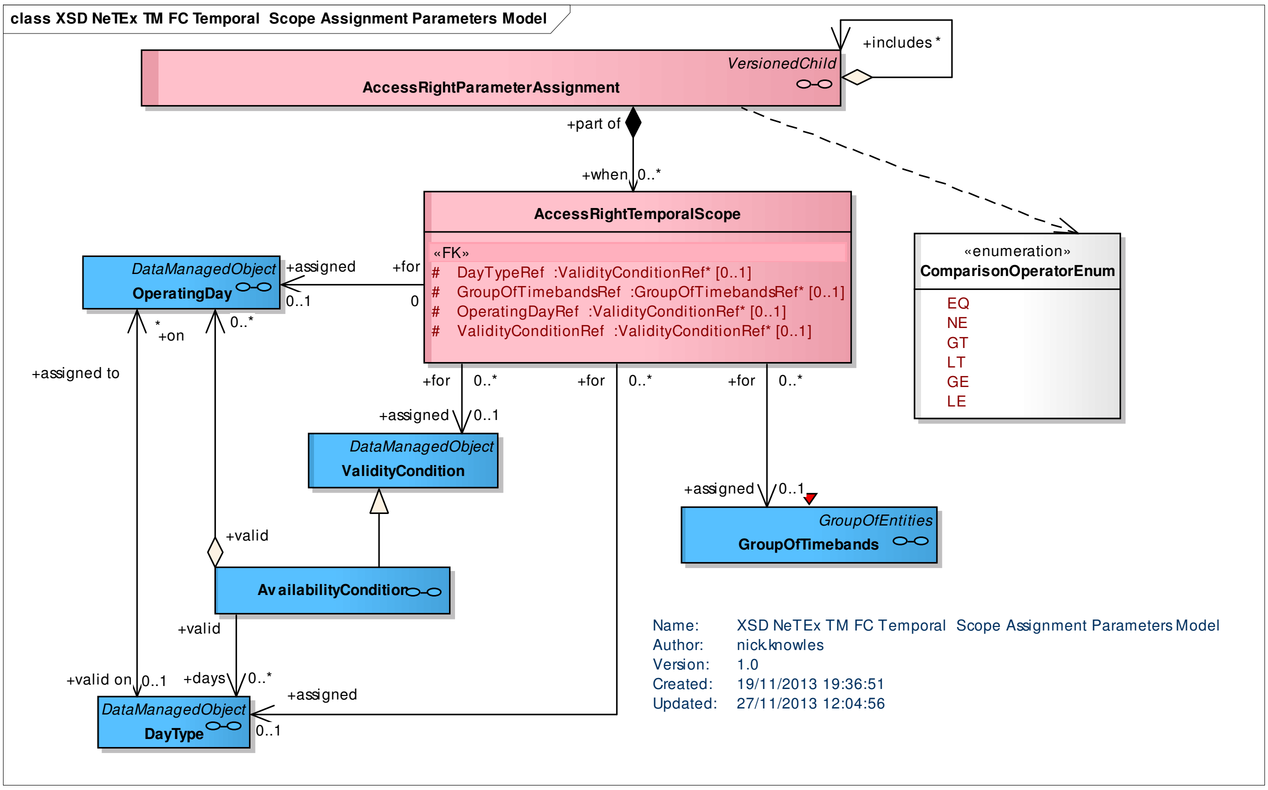


Figure 135 — Access Right Parameters: Temporal Scope Parameters – Physical Model

8.6.1.2.2.5 Access Rights Parameters: Limitations – Physical model

The following figure shows the limitation ie USAGE PARAMETERs 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.

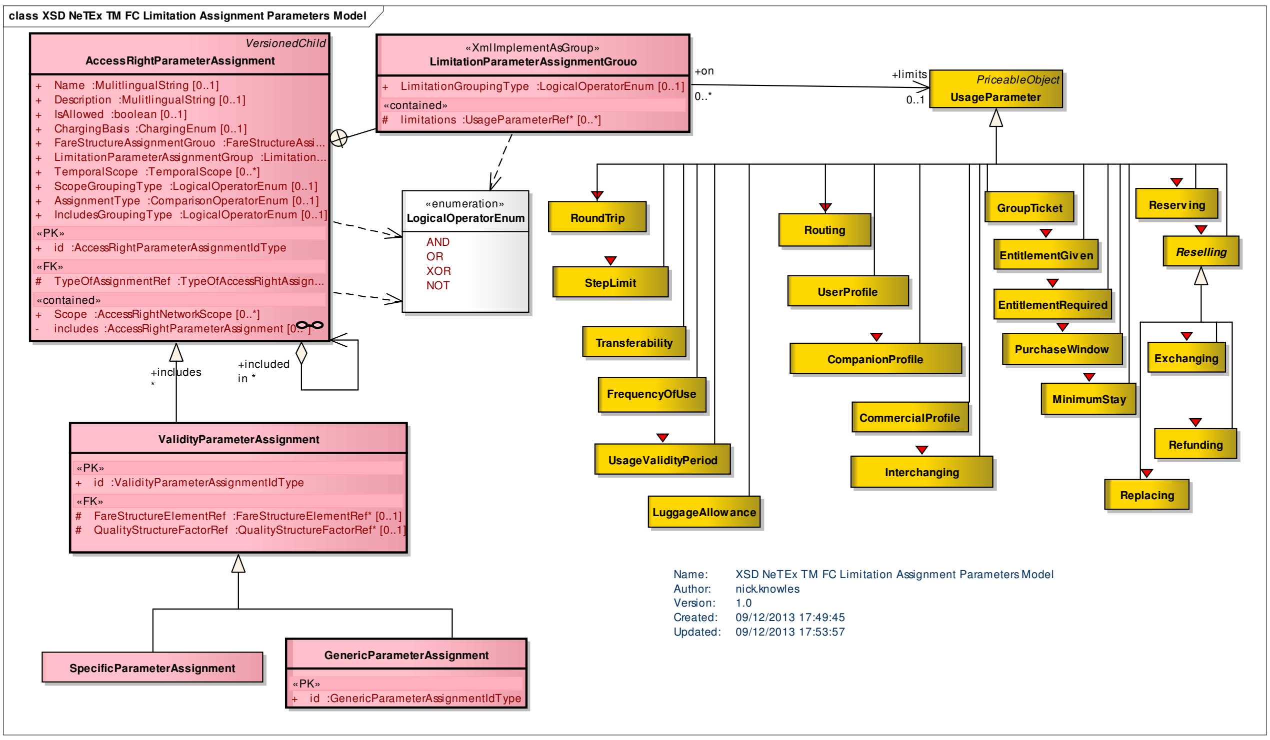


Figure 136 — Access Right Parameters: Limitations – Physical Model

8.6.1.2.3 Access Rights Parameters – Attributes and XSD

8.6.1.2.3.1 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 84 – *AccessRightParameterAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	ACCESS RIGHT PARAMETER ASSIGNMENT inherits from VERSIONED CHILD. See NeTEx Part1.
«PK»	***id***	*AccessRightParameter-AssignmentIdType*	1:1	Identifier of ACCESS RIGHT PARAMETER ASSIGNMENT.
	Name	*MultilingualString*	0:1	Name of ACCESS RIGHT PARAMETER ASSIGNMENT.
	Description	*MultilingualString*	0:1	Description of ACCESS RIGHT PARAMETER ASSIGNMENT.
GROUP	***AccessRight-Parameter-Assignment-PropertiesGroup***	*AccessRight-Parameter-Assignment-Properties¬Grou*	1:1	General properties of an ACCESS RIGHT PARAMETER ASSIGNMENT.
GROUP	***AccessRight-Parameter-AssignmentFare-StructureGroup***	*AccessRight-Parameter-Assignment--Fare¬Structure-Group*	1:1	The FARE STRUCTURE elements to which the ACCESS RIGHT PARAMETER ASSIGNMENT is made.
GROUP	***AccessRight-Parameter-Assignment-LimitationsGroup***	*AccessRight-Parameter-Assignment-Limitations¬Group*	1:1	USAGE PARAMETER by which ACCESS RIGHT PARAMETER ASSIGNMENT is limited.
GROUP	***AccessRight-ScopeGroup***	*AccessRightScopel-Group*	1:1	Access right scope parrameters by ACCESS RIGHT PARAMETER ASSIGNMENT is restricted.
GROUP	***AccessRight-IncludesGroup***	*AccessRight-Includes¬¬Grou*	1:1	Elements used to create composite ACCESS RIGHT PARAMETER ASSIGNMENTs.

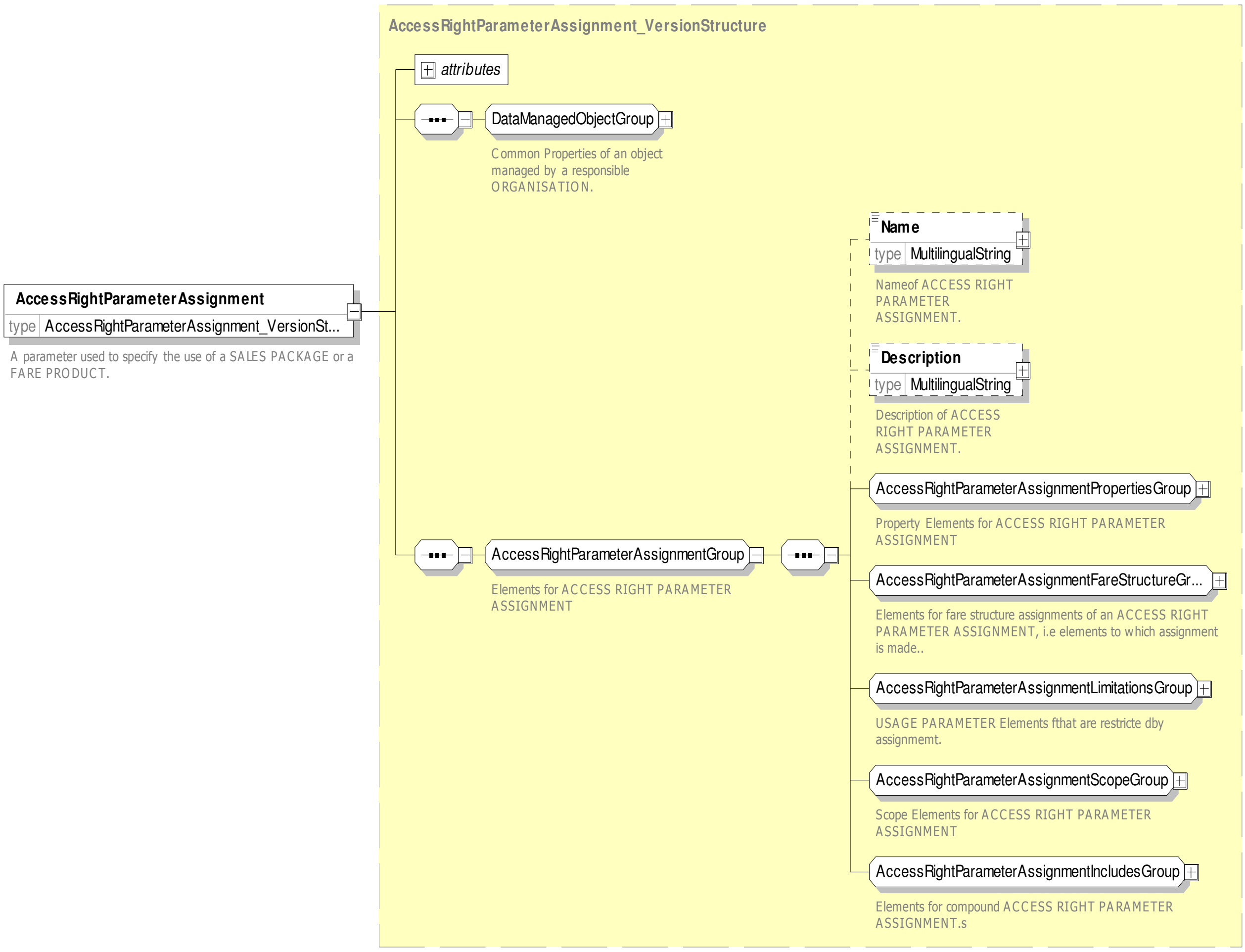


Figure 137 — *AccessRightParameterAssignment* — XSD

8.6.1.2.3.2 AccessRightParameterAssignmentPropertiesGroup – Group

The ***AccessRightParameterAssignmentPropertiesGroup*** define**s** basic properties an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 85 – *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.
	ChargingBasis	*ChargingBasisEnum*	0:1	Whether the specified assignment is for charged access, discounted access or free access. See allowed values below.

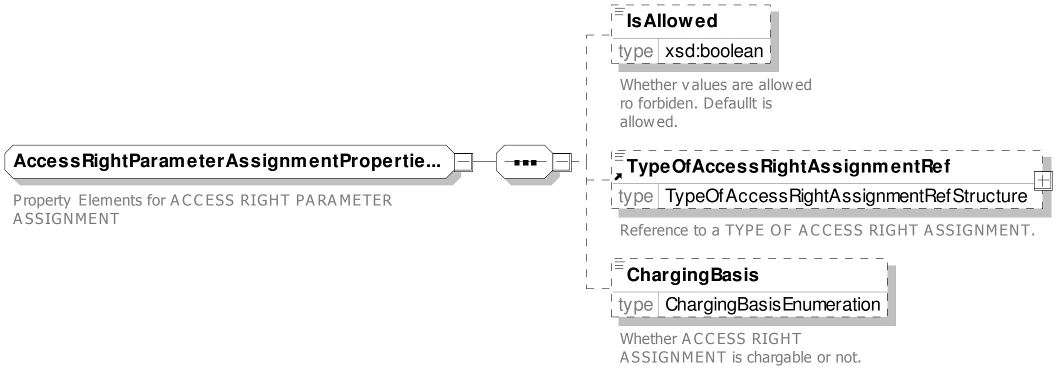


Figure 138 — *AccessRightParameterAssignmentPropertiesGroup* — XSD

8.6.1.2.3.2.1 ChargingBasis – Allowed values

The following table shows the allowed values for ***ChargingBasis*** (*ChargingBasisEnum*).

Table 86 – *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

8.6.1.2.3.3 AccessRightParameterAssignmentFareStructureGroup – Group

The ***AccessRightParameterAssignmentFareStructureGroup*** species the fare structure elements to which an assignment is made by an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 87 – *AccessRightParameterAssignmentFareStructureGroup* – 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»	***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»	***SalesPackageRef***	*SalesPackageRef*	0:1	SALES PACKAGE to which assignment is made.

«FK»	***GroupOfSales-PackagesRef***	*GroupOfSales-PackagesRef*	0:1	GROUP OF SALES PACKAGEs to which assignment is made.

Figure 139 — *AccessRightParameterAssignmentFareStructureGroup* — XSD

8.6.1.2.3.4 AccessRightParameterAssignmentLimitationsGroup – Group

The ***AccessRightParameterAssignmentLimitationsGroup*** species the limitation conditions made by an ACCESS RIGHT PARAMETER ASSIGNMENT by reference one ro more usage parameters.

Table 88 – *AccessRightParameterAssignmentLimitationsGroup* – Group

Classifi-	Name	Type	Cardinality	Description

cation				
	Limitations-GroupingType	*BooleanOperatorEnum*	0:1	Logical operator for combining USAGE PARAMETERs elements. Default is AND. OR and XOR should only be used if parameters are all of the same type.
«FK»	***limitations***	*UsageParameter¬Ref*	0:*	References to USAGE PARAMETERs defining limitations made by ACCESS RIGHT PARAMETER ASSIGNMENT.

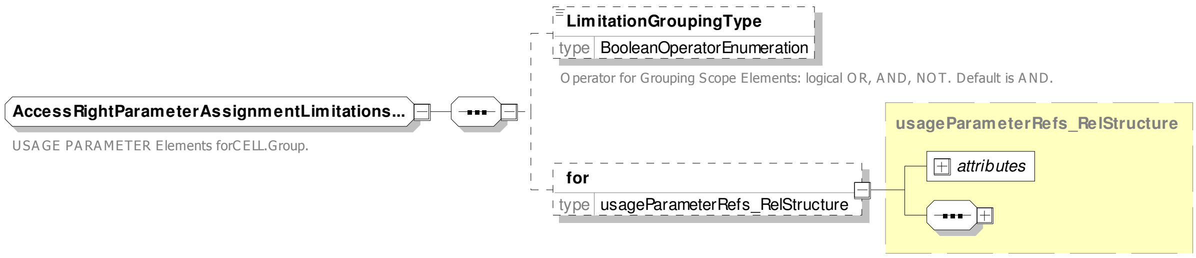


Figure 140 — *AccessRightParameterAssignmentLimitationsGroup* — XSD

8.6.1.2.3.5 AccessRightParameterAssignmentScopeGroup – Group

The ***AccessRightParameterAssignmentScopeGroup*** species the network and temporal access right restrictions made by an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 89 – *AccessRightParameterAssignmentScopeGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***AssignmentType***	*Comparison-OperatorEnum*	0:1	Comparison operator for matching scope values.
GROUP	***AccessRight-ScopeTemporal-Group***	*AccessRight-ScopeTemporalGroup*	1:1	Time related properties assigned by ACCESS RIGHT PARAMETER ASSIGNMENT.
	Scope-GroupingType	*BooleanOperatorEnum*	0:1	Logical operator for combining network scope elements, e.g. AND, OR, XOR.
«cntd»	***Scope***	*Scope*	0:1	Scope properties assigned by ACCESS RIGHT PARAMETER ASSIGNMENT.

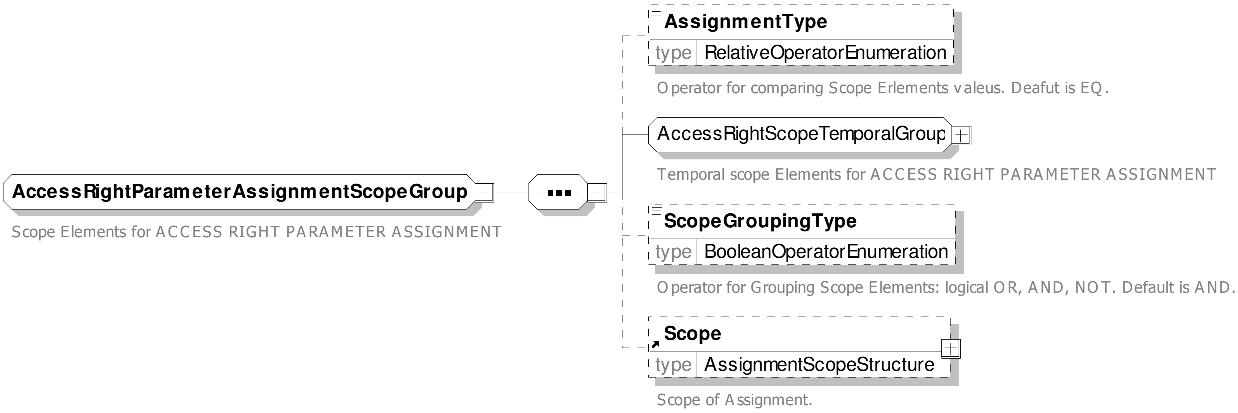


Figure 141 — *AccessRightParameterAssignmenScopeGroup* — XSD

8.6.1.2.3.5.1 AssignmentType – Allowed values

The following table shows the allowed values for ***AssignmentType*** (*ComparisonOperatorEnum*)

Table 90 – *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.

8.6.1.2.3.5.2 ScopeGroupingType – Allowed values

The following table shows the allowed values for ***ScopeGroupingType*** (*BooleanOperatorEnum*)

Table 91 – *GroupingType* – Allowed values

Value	Description
AND	Logical And
OR	Logical Or
XOR	Exclusive OR
NOT	Logical not

8.6.1.2.3.6 AccessRightScopeTemporalGroup – Group

The ***AccessRightScopeTemporalGroup*** species the network and temporal access right restrictions made by an ACCESS RIGHT PARAMETER ASSIGNMENT. See NeTEx Part1 for Furtehr details on these elements.

Table 92 – *AccessRightScopeTemporalGroup* – 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***	*ValidityConditionRef*	0:1	OPERATING DAY to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***Validity-ConditionRef***	*ValidityConditionRef*	0:1	VALIDITY CONDITION to which ACCESS RIGHT PARAMETER is assigned.

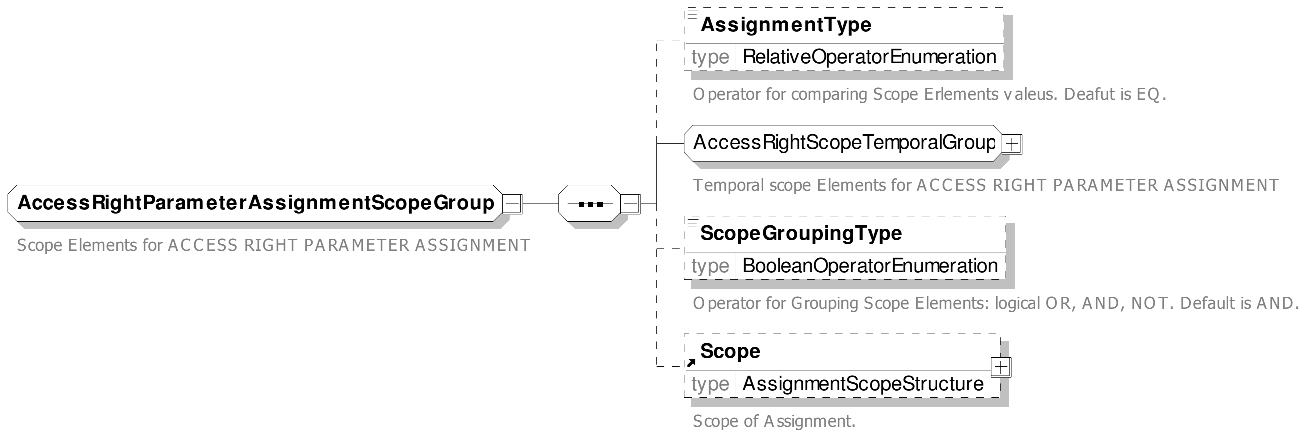


Figure 142 — *AccessRightScopeTemporalGroup* — XSD

8.6.1.2.3.7 Scope – Model Element

Limiting values for the assignment. Multiple values are Logically ANDed together. For example LINE “22“ and SCHEDULED STOP POINT “4563” means that the assignment applies specifically to stop “4563” of LINE “22”.

Table 93 – Scope – Element

Classifi-cation	Name	Type	Cardin-ality	Description
GROUP	***AccessRight-Scope-ClassOfUse-Group***	*AccessRightScope-ClassOfUseGroup*	1:1	CLASS OF USE elements for assignment SCOPE.
GROUP	***AccessRight-ScopeGlobal-Group***	*AccessRightScopeGlobal-Group*	1:1	Global service elements for assignment SCOPE.
GROUP	***AccessRight-ScopeNetwork-Group***	*AccessRightScope-NetworkGroup*	1:1	Network elements for assignment SCOPE.

GROUP	***AccessRight-ScopeSiteGroup***	*AccessRightScopeSite-Group*	1:1	SITE elements for assignment SCOPE.
GROUP	***AccessRight-ScopeRoute-Group***	*AccessRightScopeRoute-Group*	1:1	Route elements for assignment SCOPE.
GROUP	***AccessRight-ScopeService-Group***	*AccessRightScope-ServiceGroup*	1:1	SERVICE elements for assignment SCOPE.
GROUP	***AccessRight-ScopeFacility-Group***	*AccessRightScopeFacility-Group*	1:1	FACILITY elements for assignment SCOPE.
GROUP	***AccessRight-Scope-Distribution-Group***	*AccessRightScope-DistributionGroup*	1:1	DISTRIBUTION elements for assignment SCOPE.

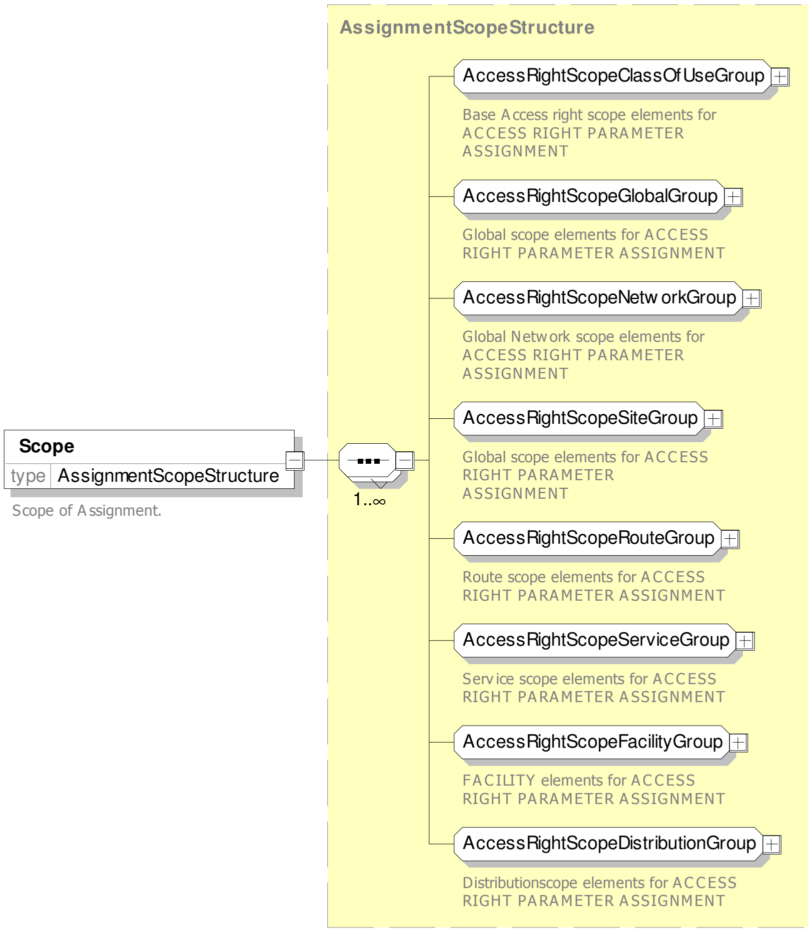


Figure 143 — *Scope* — XSD

8.6.1.2.3.7.1 AccessRightScopeClassOfUseGroup – Group

The ***AccessRightScopeClassOfUseGroup*** species the restrictions on class of use made by an ACCESS RIGHT PARAMETER ASSIGNMENT. Either predefined standard *SeatClass* or an arbitrary CLASS OF USE may be used.

Table 94 – *AccessRightScopeClassOfUseGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***ClassOfUseRef***	*ClassOfUseRef*	1:1	Reference to a CLASS OF USE (Seat Class)
	SeatClass	*FareClassEnum*	0:1	FARE CLASS to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1 for allowed values.

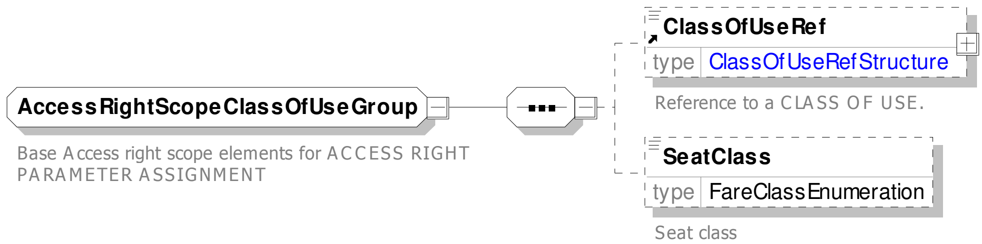


Figure 144 — *AccessRightScopeClassOfUseGroup* — XSD

8.6.1.2.3.7.2 AccessRightScopeGlobalGroup – Group

The ***AccessRightScopeGlobalGroup*** species general access rights for MODE and ORGANISATION for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 95 – *AccessRightScopeGlobalGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	VehicleModes	*TransportModeEnum*	0:*	TRANSPORT MODEs to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1 for allowed values. See NeTEx Part1.
«FK»	***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.
«FK»	***OperatorRef***	*OperatorRef*	0:1	OPERATOR to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***AuthorityRef***	*AuthorityRef*	0:1	AUTHORITY to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.

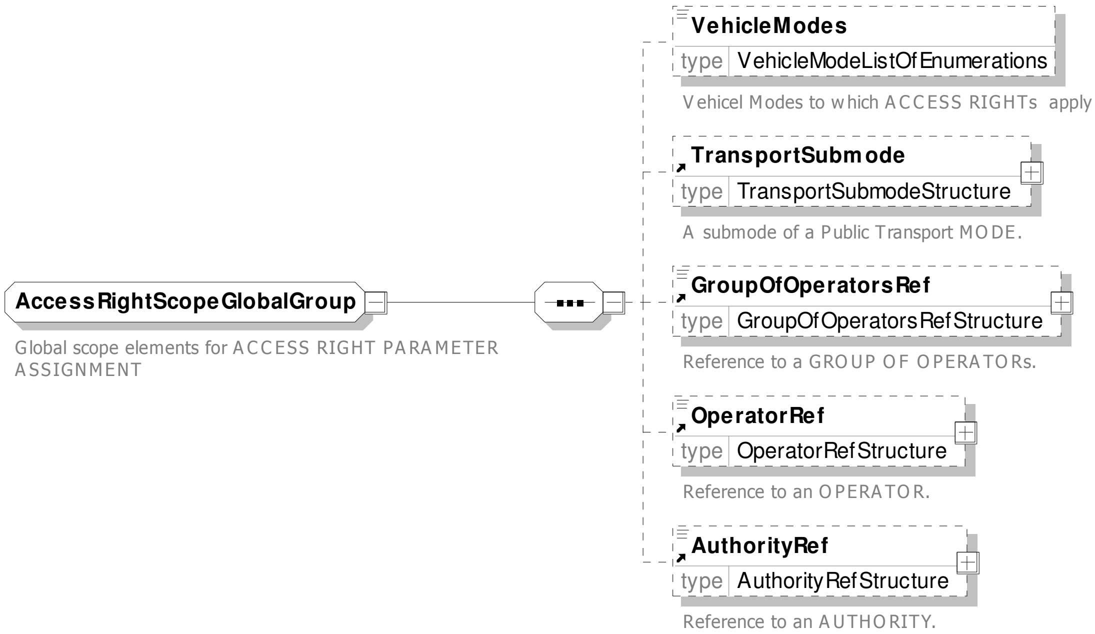


Figure 145 — *AccessRightScopeGlobalGroup* — XSD

8.6.1.2.3.7.3 AccessRightScopeNetworkGroup – Group

The ***AccessRightScopeNetworkGroup*** species access rights to network elements such as LINE, TARIFF ZONE and SCHEDULED STOP POINT for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 96 – *AccessRightScopeNetworkGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***GroupOfLinesRef***	*GroupOfLinesRef*	0:1	GROUP OF LINEs to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.
«FK»	***LinesRef***	*LineRef*	0:1	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.

Figure 146 — *AccessRightScopeNetworkGroup* — XSD

8.6.1.2.3.7.4 AccessRightScopeSiteGroup – Group

The ***AccessRightScopeSiteGroup*** species 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.

Table 97 – *AccessRightScopeSiteGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***SiteElementRef***	*SiteElementRef*	0:1	SITE ELEMENT to which ACCESS RIGHT PARAMETER is assigned. See NeTEx Part1.

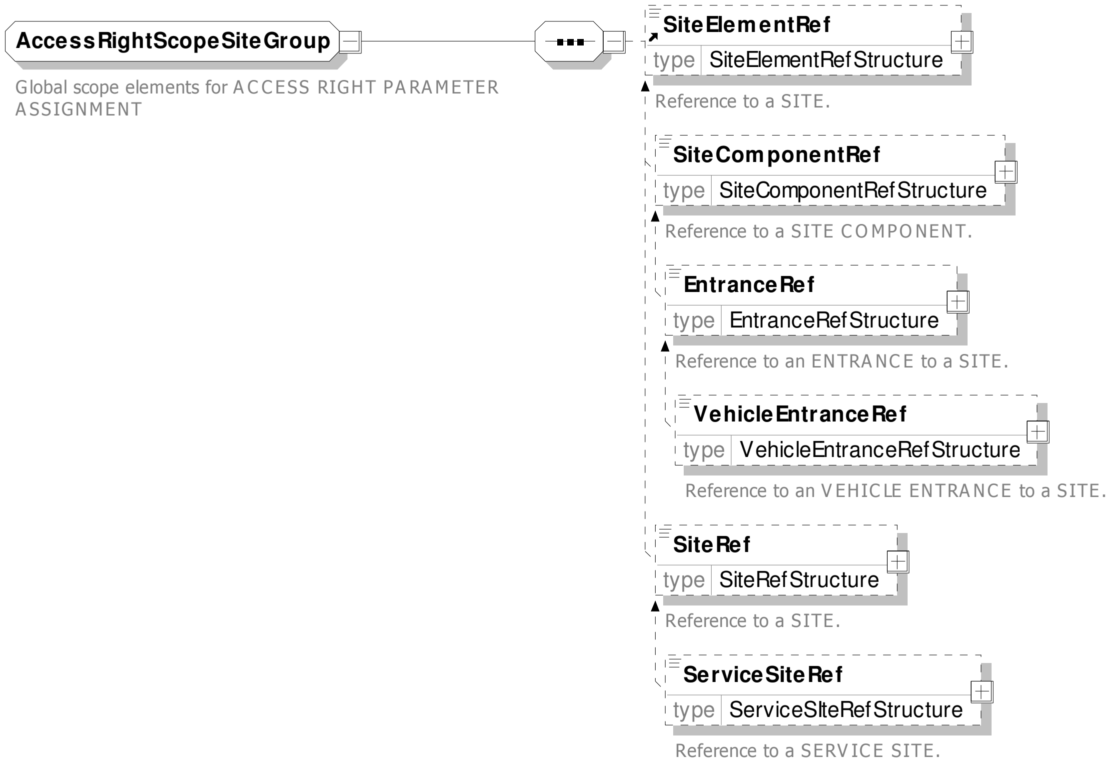


Figure 147 — *AccessRightScopeSiteGroup* — XSD

8.6.1.2.3.7.5 AccessRightScopeRouteGroup – Group

The ***AccessRightScopeRouteGroup*** species access rights to particular routes (as in effect specified by DISTANCE MATRIX and SERIES CONSTRAINT elements) for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 98 – *AccessRightScopeRouteGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	RoutingType	*RoutingTypeEnum*	1:1	Type of routing to which assignment applies. See allowed values earlier under FARE ZONE model.
«FK»	***SeriesConstraint Ref***	*SeriesConstraintRef*	0:1	SERIES CONSTRAINT to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***Distance-MatrixRef***	*DistanceMatrixRef*	0:1	DISTANCE MATRIX ELEMENT to which ACCESS RIGHT PARAMETER is assigned.
	Directions	*RelativeDirectioeEnum*	1:1	Directions in which assignment applies. See NeTEx Part1.
«FK»	***GroupOf-DistanceMatrix-ElementsRef***	*GroupOfDistanceMatrix-ElementsRef*	0:1	GROUP OF DISTANCE MATRIX ELEMENTs to which ACCESS RIGHT PARAMETER is assigned.

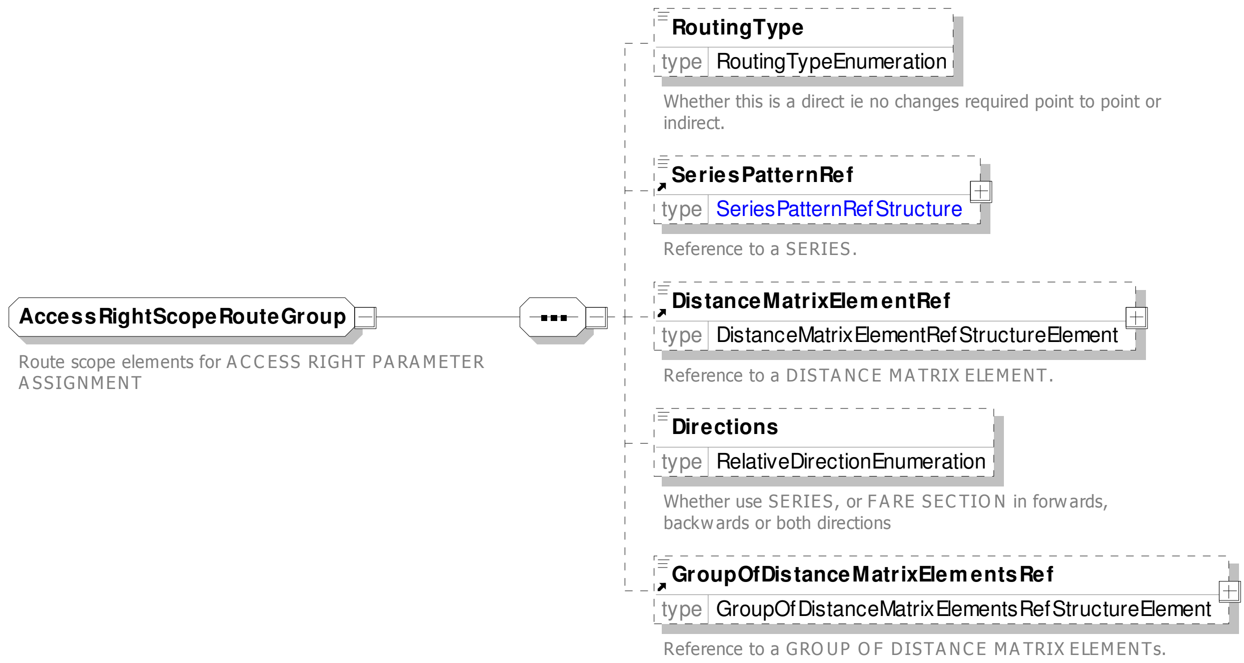


Figure 148 — *AccessRightScopeRouteGroup* — XSD

8.6.1.2.3.7.6 AccessRightScopeServiceGroup – Group

The ***AccessRightScopeServiceGroup*** species access rights to particular services or types of service for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 99 – *AccessRightScopeServiceGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***ServiceJourney-PatternRef***	*ServiceJourney-PatternRef*	0:1	JOURNEY PATTERN to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***Service-JourneyRef***	*ServiceJourneyRef*	0:1	VEHICLE JOURNEY to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***TypeOfProduct-CategoryRef***	*TypeOfProduct-CategoryRef*	0:1	Type of PRODUCT CATEGORY to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***TrainNumberRef***	*TrainNumberRef*	0:1	TRAIN NUMBER to which ACCESS RIGHT PARAMETER is assigned.
«FK»	***TypeOf-ServiceRef***	*TypeOfServiceRef*	0:1	TYPE OF SERVICE to which assignment is made, for example whether the assignment is a night train.
«FK»	***TypeOf-FareProductRef***	*TypeOfFareProductRef*	0:1	Type of FARE PRODUCT to which assignment is made, for example special excursion.

Figure 149 — *AccessRightScopeServiceGroup* — XSD

8.6.1.2.3.7.7 AccessRightScopeFacilityGroup – Group

The ***AccessRightScopeFacilityGroup*** species access rights to particular facilities (for example first class lounge, types of accommodation, etc) for an ACCESS RIGHT PARAMETER ASSIGNMENT.

Table 100 – *AccessRightScopeFacilityGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***FacilitySetRef***	*FacilitySetRef*	0:1	Facility Set provided or available for fare

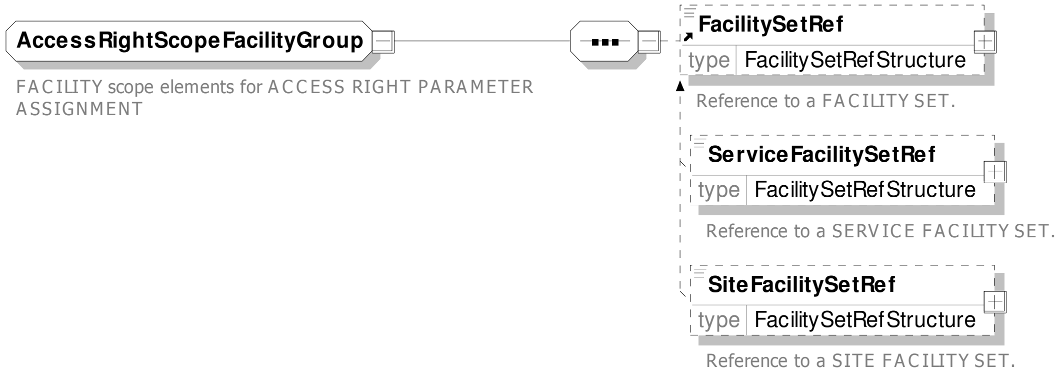


Figure 150 — *AccessRightScopeFacilityGroup* — XSD

8.6.1.2.3.7.8 AccessRightScopeDistributionGroup – Group

The ***AccessRightScopeFacilityGroup*** species 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 DISTRINUTION CHANNEL.

Table 101 – *AccessRightScopeDistributionGroup* – 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»	***Fulflment-MethodRef***	*FareStructureElement-Ref*	0:1	Reference to a FULFILMENT METHOD to which the ACCESS RIGHT PARAMETER ASSIGNMENTs applies.

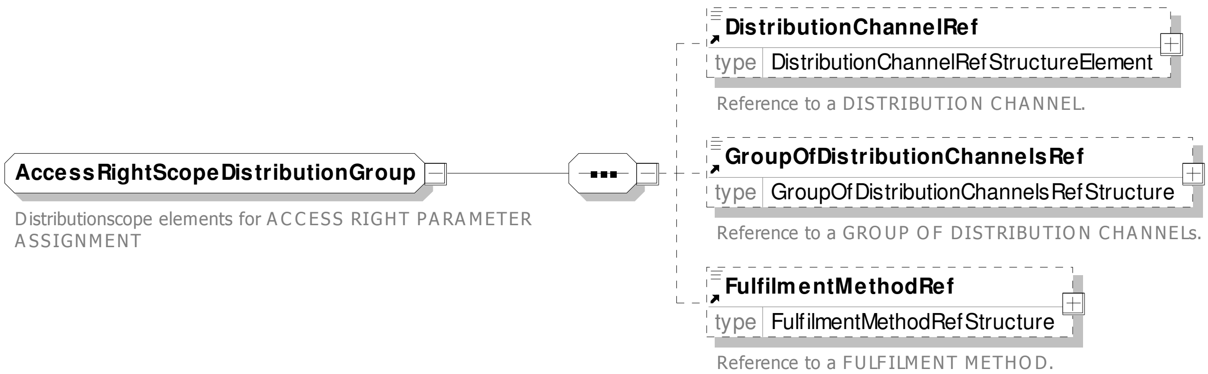


Figure 151 — *AccessRightScopeDistributionGroup* — XSD

8.6.1.2.3.8 AccessRightParameterAssignmentIncludesGroup – Group

The ***AccessRightScopeFacilityGroup*** species rules for creating composite ACCESS RIGHT PARAMETER ASSIGNMENTs that combine simple assignments into complex conditions. A logical operator can be specified.

Table 102 – *AccessRightParameterAssignmentIncludesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Includes-GroupingType	*BooleanOperatorEnum*	0:1	Logical operator for combining included elements. Default is OR.
“cntd»	***includes***	*AccessRightParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs making up a composite ACCESS RIGHT PARAMETER ASSIGNMENT.

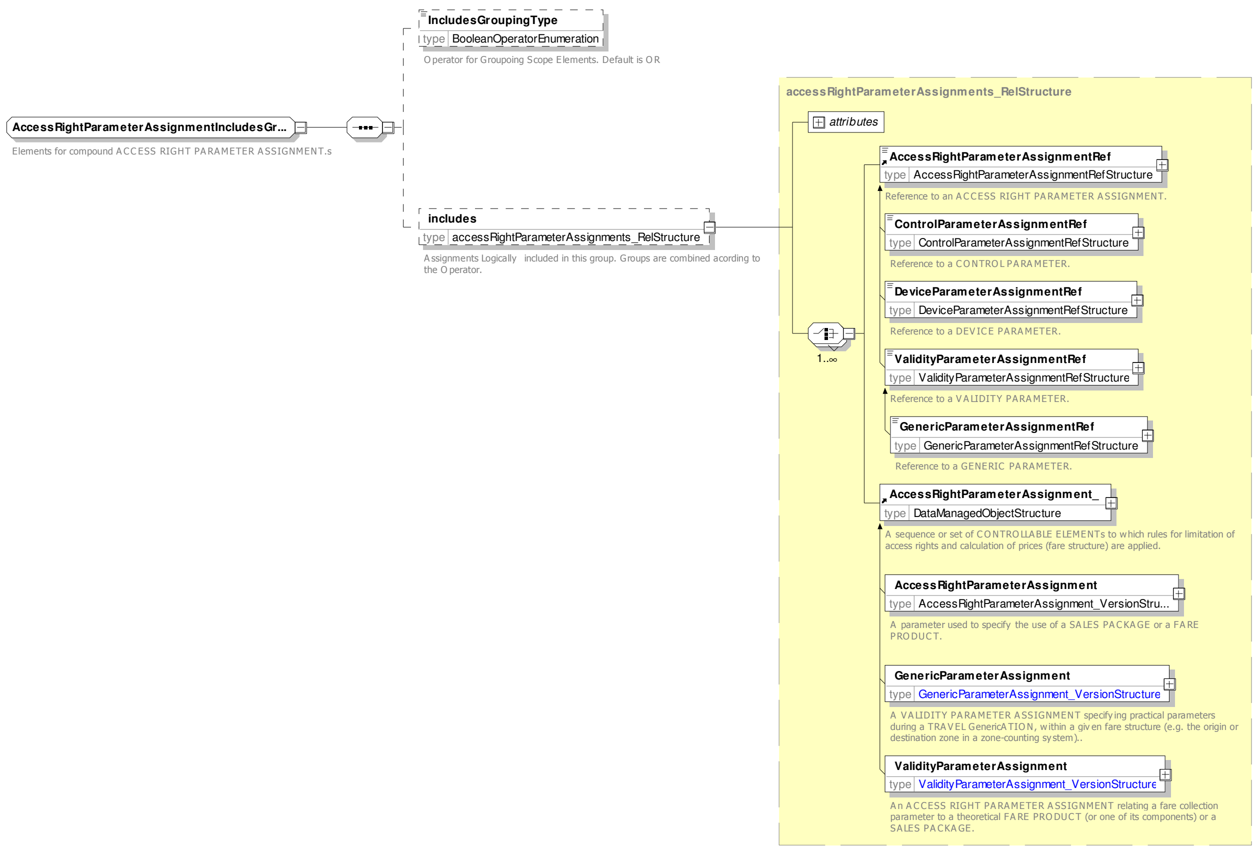


Figure 152 — *AccessRightParameterAssignmentIncludesGroup* — XSD

8.6.1.2.3.8.1 IncludesGroupingType – Allowed values

The following table shows the allowed values for the ***IncludesGroupingType*** (BooleanOperatorEnum)

Table 103 – *GroupingType* – Allowed values

Value	Description

AND	Logical And
OR	Logical Or
XOR	Exclusive OR
NOT	Logical not

8.6.1.2.3.9 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 PACKAGE.

Table 104 – *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.

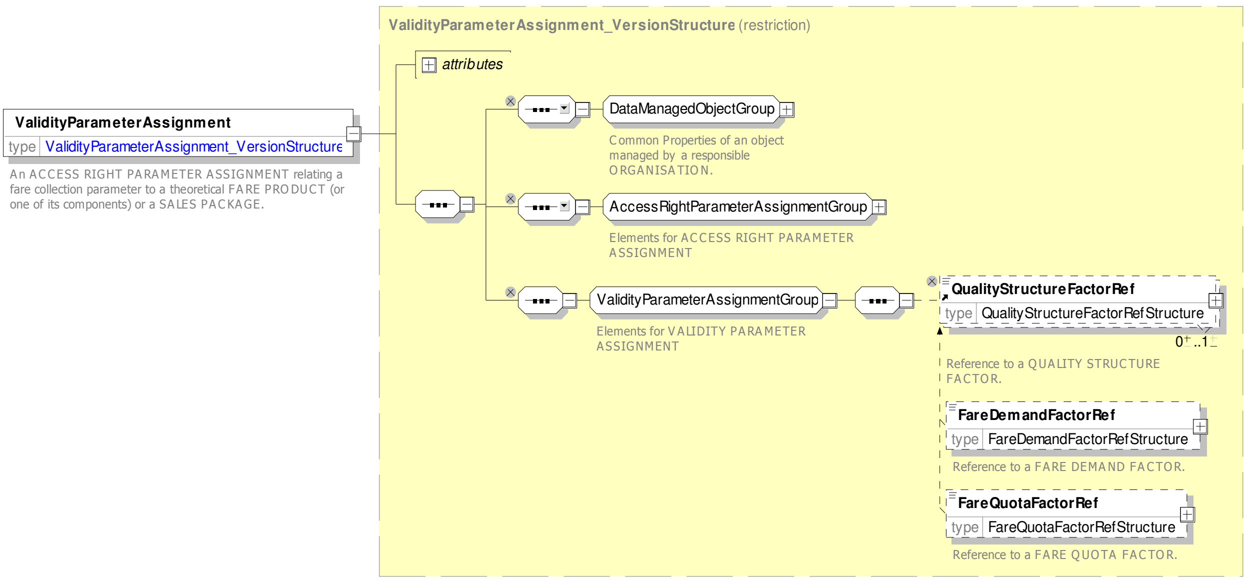


Figure 153 — *ValidityParameterAssignment* — XSD

8.6.1.2.3.10 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).

Table 105 – *GenericParameterAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*ValidityParameter-Assignment*	::>	GENERIC PARAMETER ASSIGNMENT inherits from VALIDITY PARAMETER ASSIGNMENT
«PK»	***id***	*GenericParameter-AssignmentIdType*	1:1	Identifier of GENERIC PARAMETER ASSIGNMENT.

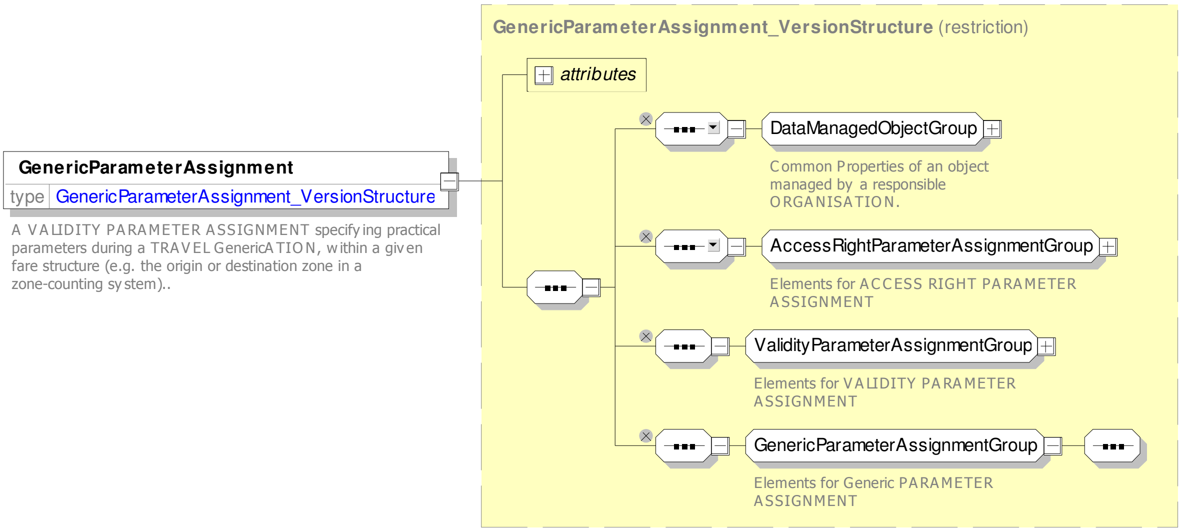


Figure 154 — *GenericParameterAssignment* — XSD

8.6.1.2.3.11 TypeOfAccessRightAssignment – Model Element

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

Table 106 – *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.

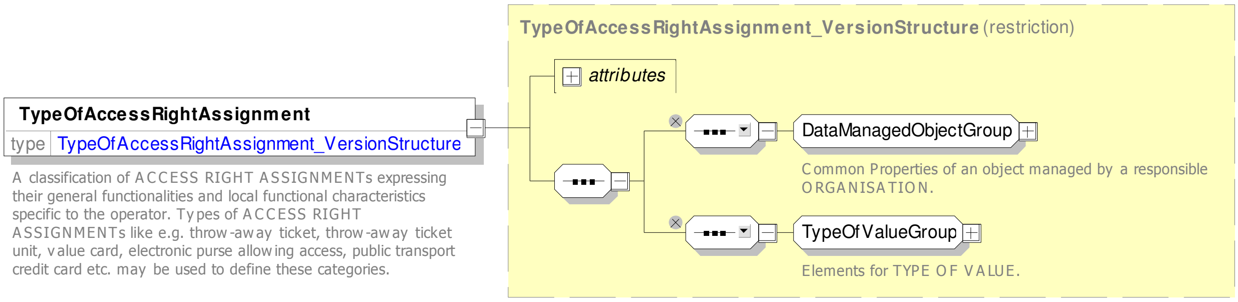


Figure 155 — *TypeOfAccessRightAssignment* — XSD

8.6.1.2.4 Access Right Assignment – XML examples

[TO DO]

8.6.1.3 Usage Parameter

8.6.1.3.1 Usage Parameter – 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[KB22].

  — 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 relating to single or return trip use of an access right[KB23].

- **Product** USAGE PARAMETERs specify limitations on who can use a product such as USER PROFILE, GROUP TICKET, COMPANION OR GROUP MEMBER, COMMERICAL 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;
  - 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[KB24].
- **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[KB25] the product
    - EXCHANGING indicating whether and how the access right may be exchanged for another access right[KB26]
    - REFUNDING indicating whether and how the purchased access right may be refunded[KB27].
    - REPLACING indicating whether and how the access right may be replaced[KB28]
    - RESERVING indicating whether the access right requires reservation[KB29].

USAGE PARAMETERs may have one or more USAGE PARAMETER PRICE and be classified with a TYPE OF USER PARAMETER.

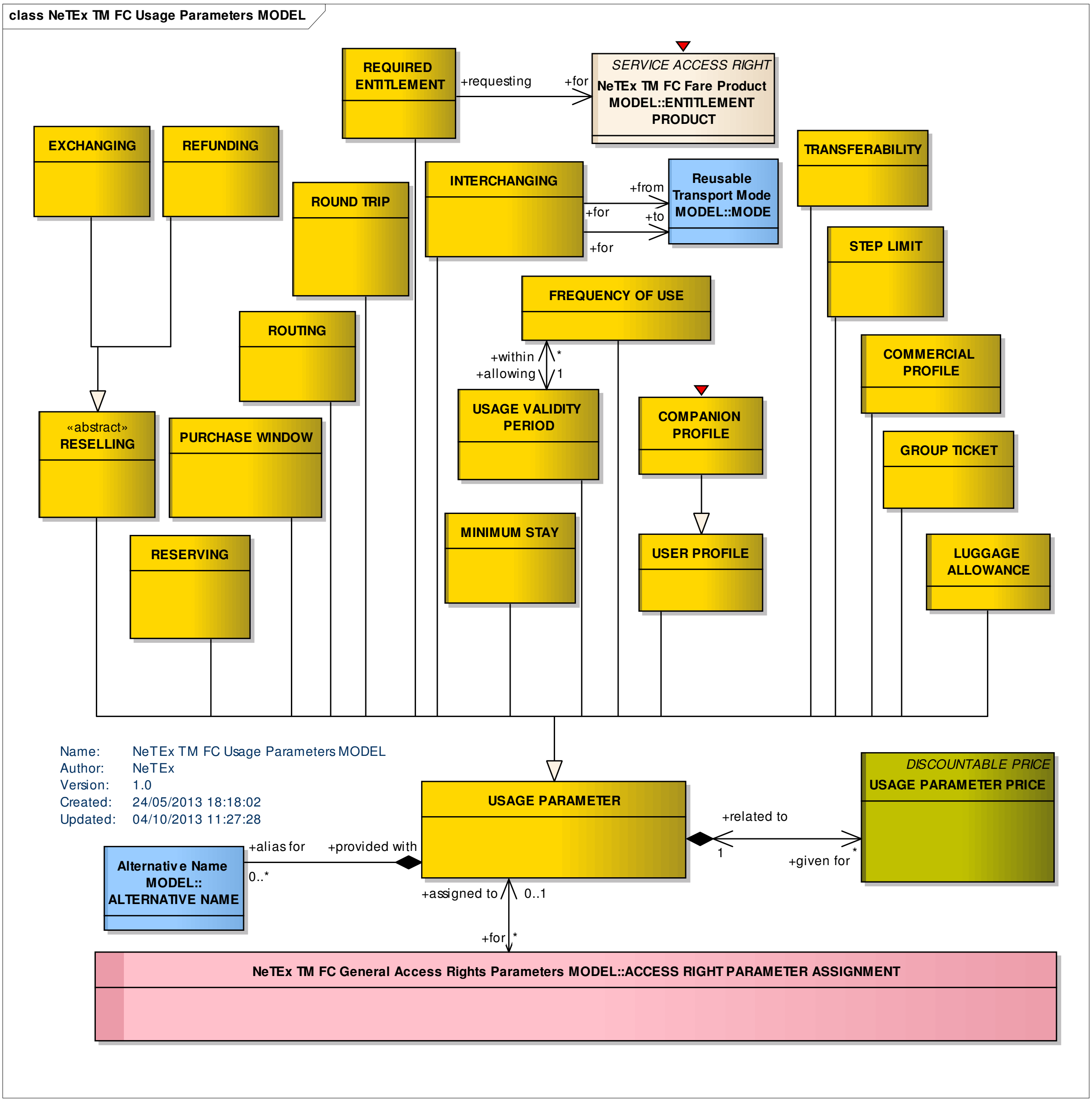


Figure 156 —Usage Parameters – Conceptual Model

8.6.1.3.2 Usage Parameter – Conceptual Examples

8.6.1.3.2.1 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 TERMS

Please find below a summary of the conditions that apply to your selected ticket(s).	
CHEAP DAY	
Train Operator	Most Train Operating Companies
Booking Deadlines	No deadline - walk UP fare
Discounts	Discounts are available for all railcard holders
Refunds	Full refund if wholly unused minus cancellation fee of GBP7.50 if processed online, or if processed at call centre.
Changes To Travel Plans	GBP10.00 admin charge plus upgrade to next appropriate walk up fare.
Conditions	Reservations are not essential but are recommended on certain services. Return journey must be made on same day.
Break Of Journey	Valid for break of journey on outward and return portion of ticket
Availability	Available on most off-peak journeys on any days.
Validity	Only valid for off peak travel on date shown on ticket. Not valid for travel on some Monday to Friday peak services, especially to/from London. Definition of peak period is dependent on route.

Figure 157 —Usage Parameters – Example of Rail conditions. (National Rail Enquiries)

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 VALIDTY PARAMETER ASSIGNMENTs. Peak and off peak periods can be defined using FARE DEMAND FACTOR can be used to specify.

Fare conditions can also be summarised in the CONDITION SUMMARY on a FARE PRODUCT or SALES 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>
```

[TOD DO MORE EXAMPLES]

8.6.1.3.2.2 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:
Fare structure parameters:
GEOGRAPHICAL INTERVAL: 1 – 4, GEOGRAPHICAL UNIT: zone
TIME INTERVAL: 0-1, TIME UNIT: hour
and
Limiting parameters (validity parameters)
OPERATOR= N and MODE = bus and (LINE different 278 or LINE different 66))
and
TIMEBAND different 2 a.m.- 4a.m.
and
Limiting parameters (usage parameters)
USER PROFILE: student[KB30]

8.6.1.3.3 Usage Parameter – Physical model

The following figure introduces the General physical model for USAGE PARAMETERs. The various different physical

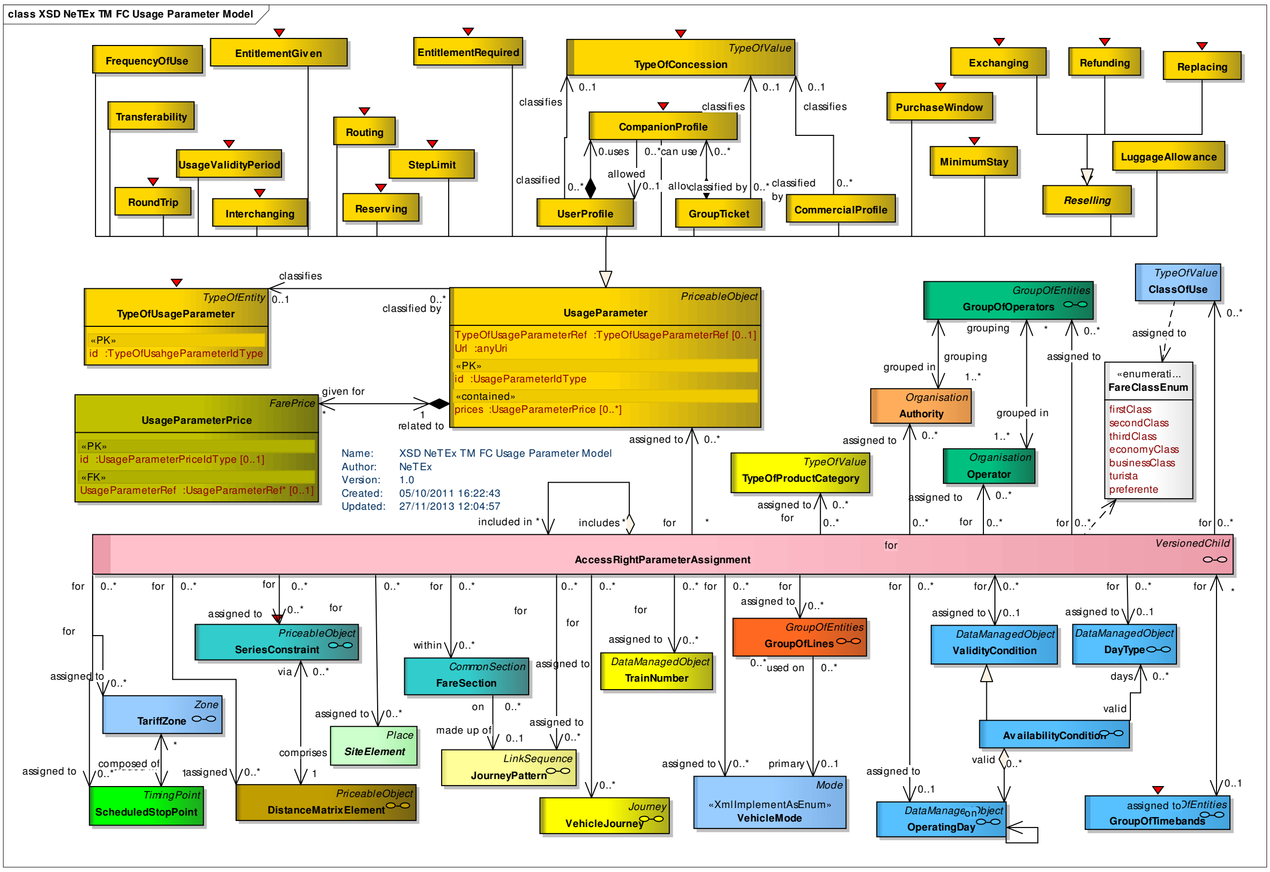


Figure 158 —Usage Parameters Summary – Physical Model

8.6.1.3.3.1 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 FREQUENCY OF USE parameter specifies some restrictions on ROUTING (See also SERIES CONSTRAINT).

— The USAGE VALIDITY PERIOD parameter specifies any limitations on how long the passenger may travel.

— The MINIMUM STAY parameter specifies if there is a restriction on how long the passenger must stay at the destination before returning.

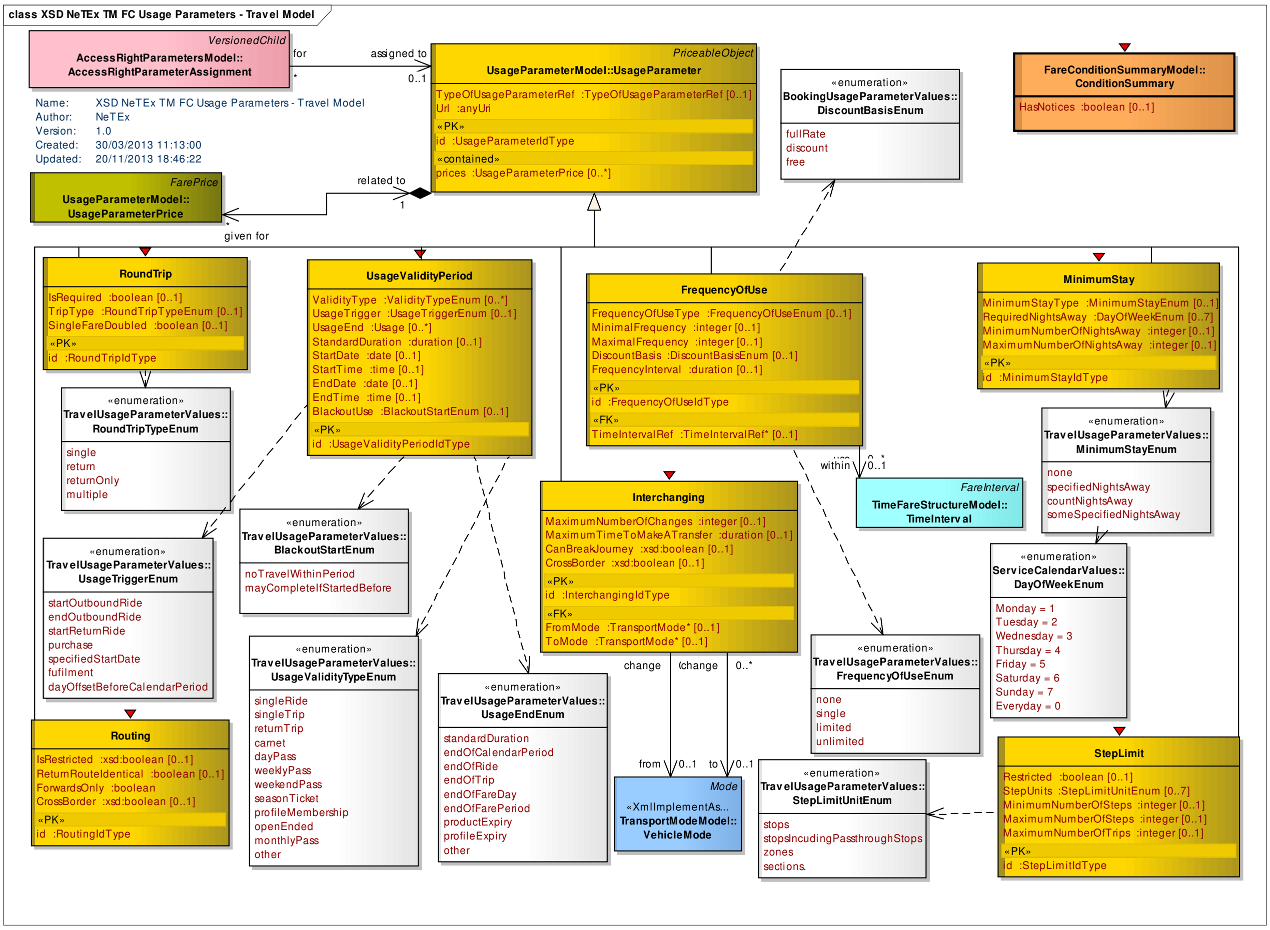


Figure 159 —Usage Parameters: Travel – Physical Model

8.6.1.3.3.2 Usage Parameter: Product – Physical model

The following figure shows the physical model for USAGE PARAMETERs describing limits on eligibility for products.

- The GROUP TICKET parameter specifies properties of group tickets and who may use them.
- The USER PROFILE parameter specifies who may use a product. Residential restrictions may be specified by a RESIDENTIAL QUALFICATION.
- 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.
- The ENTITLEMENT REQUIRED parameter specifies any prerequisite products for purchasing the product.
- The ENTITLEMENT GIVEN parameter specifies any entitlements given by the product.
- TYPE OF CONCESSION describes classes of users who may be eligible for GROUP TICKETs or USER PROFILEs.

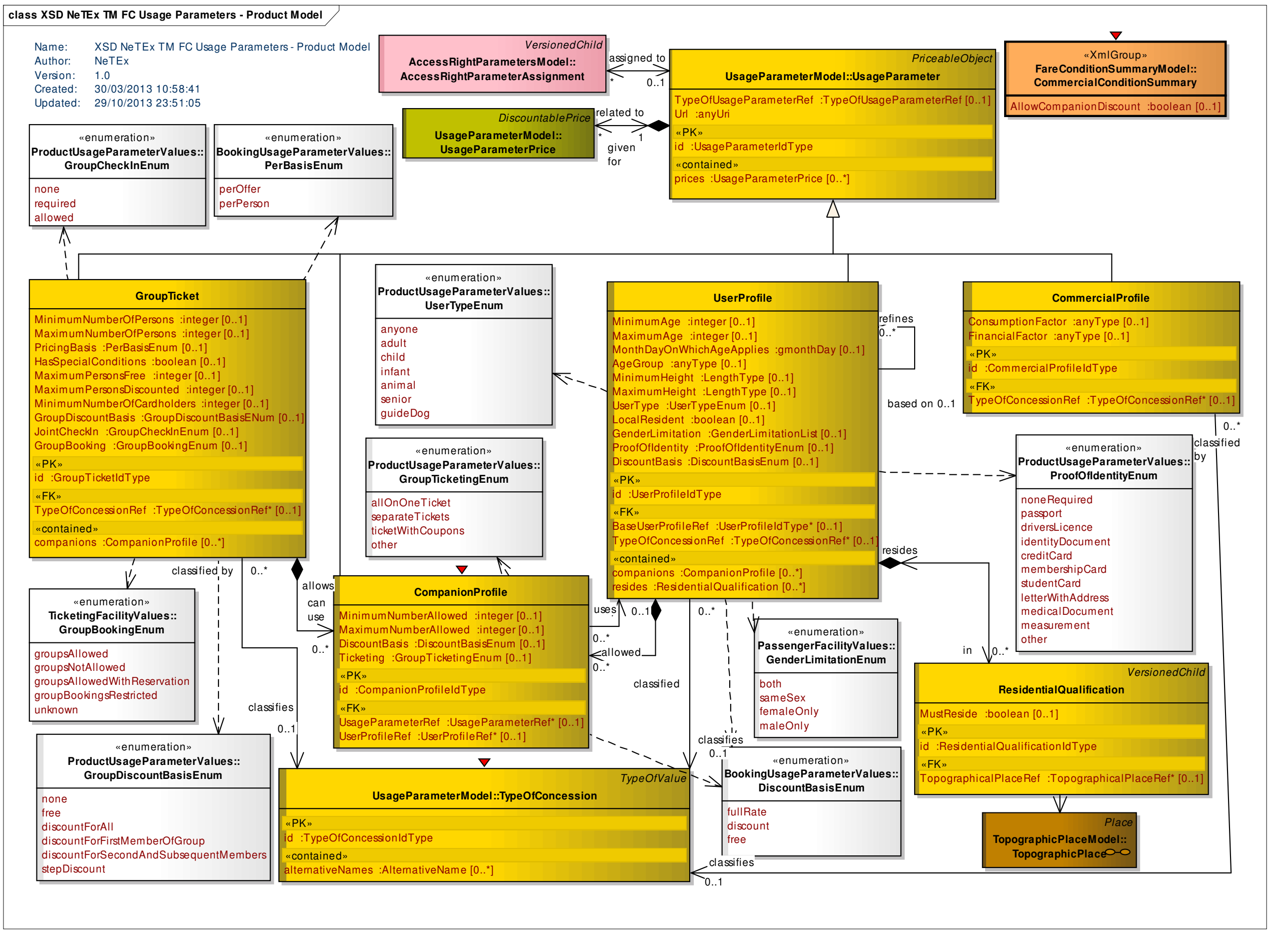


Figure 160 —Usage Parameters: Product – Physical Model

8.6.1.3.3.3 Usage Parameter: Product 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.

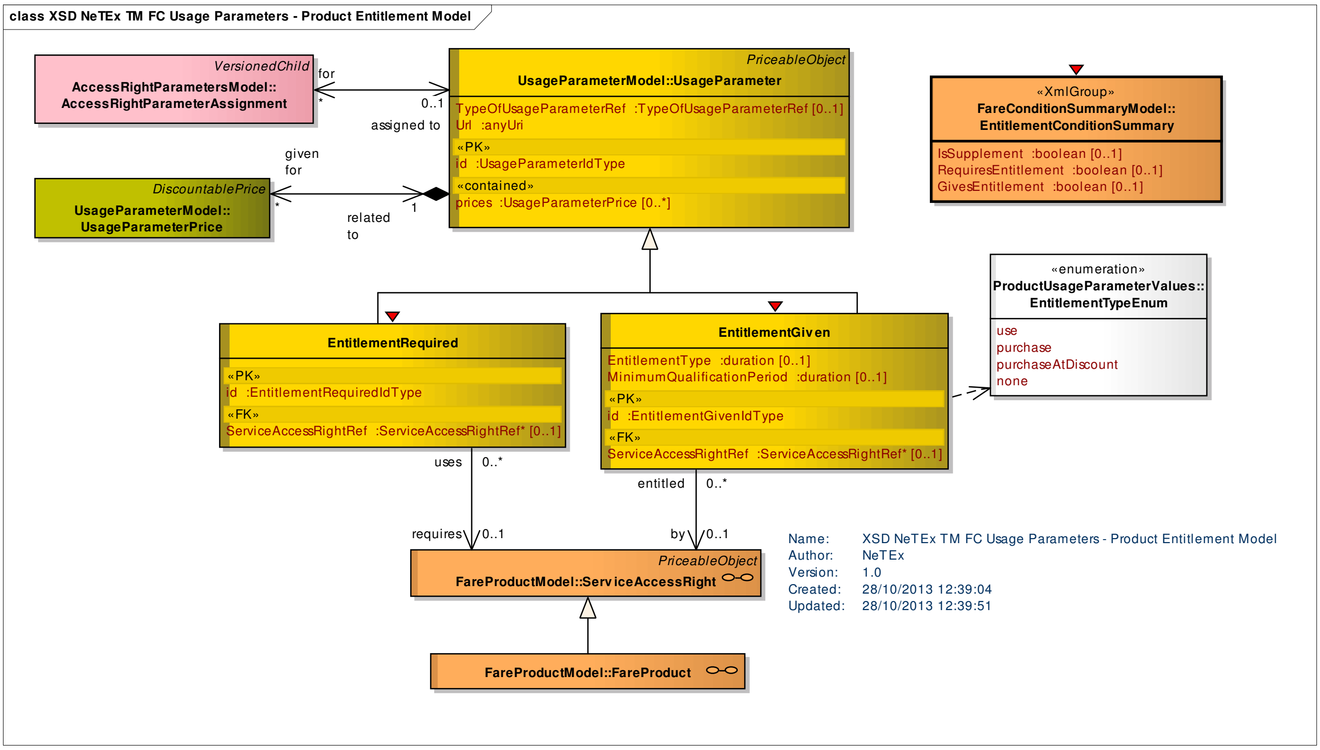


Figure 161 —Usage Parameters: Product Entitlement– Physical Model

8.6.1.3.3.4 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.

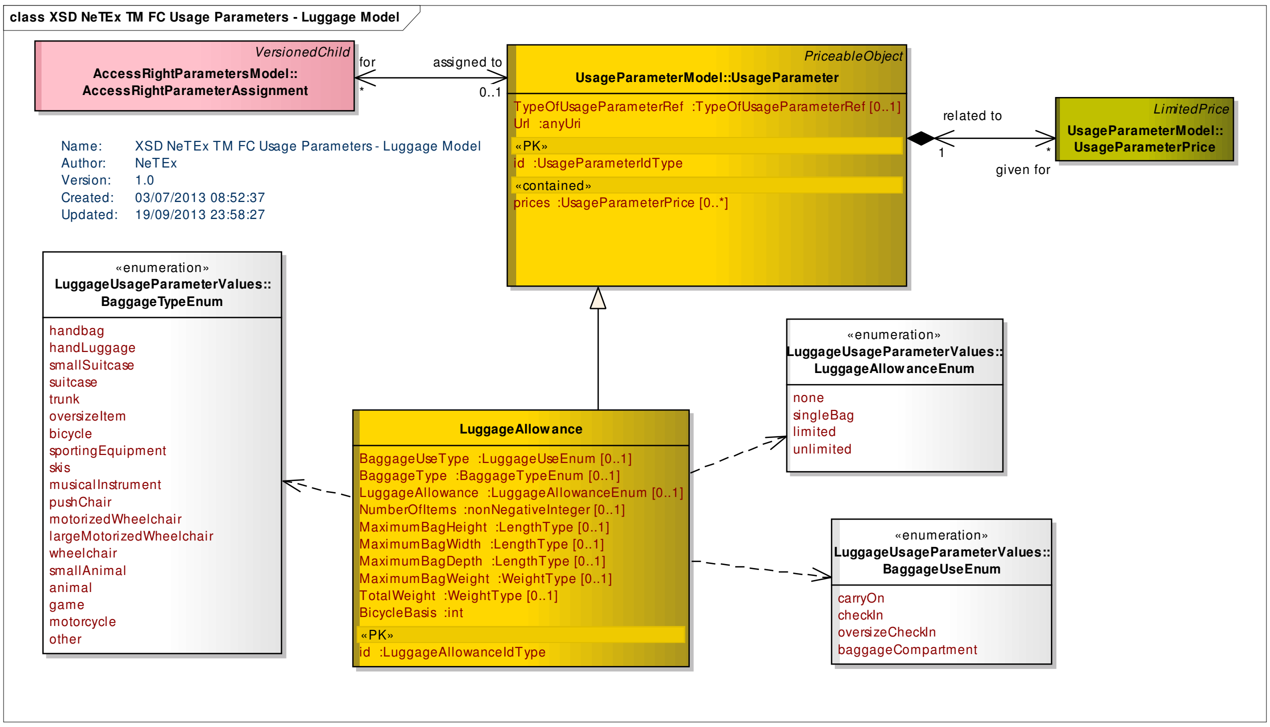


Figure 162 —Usage Parameters: Luggage – Physical Model

8.6.1.3.3.5 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 TRANSFERABILITY parameter specifies limitations on transferring a ticket to someone else.

— The REFUNDING parameter specifies limitations on refunds for a product and other resale properties.

— The EXCHANGING parameter specifies limitations on exchanging tickets for a other tickets.

— The REPLACING parameter specifies whether the product can be replaced if lost or stolen.

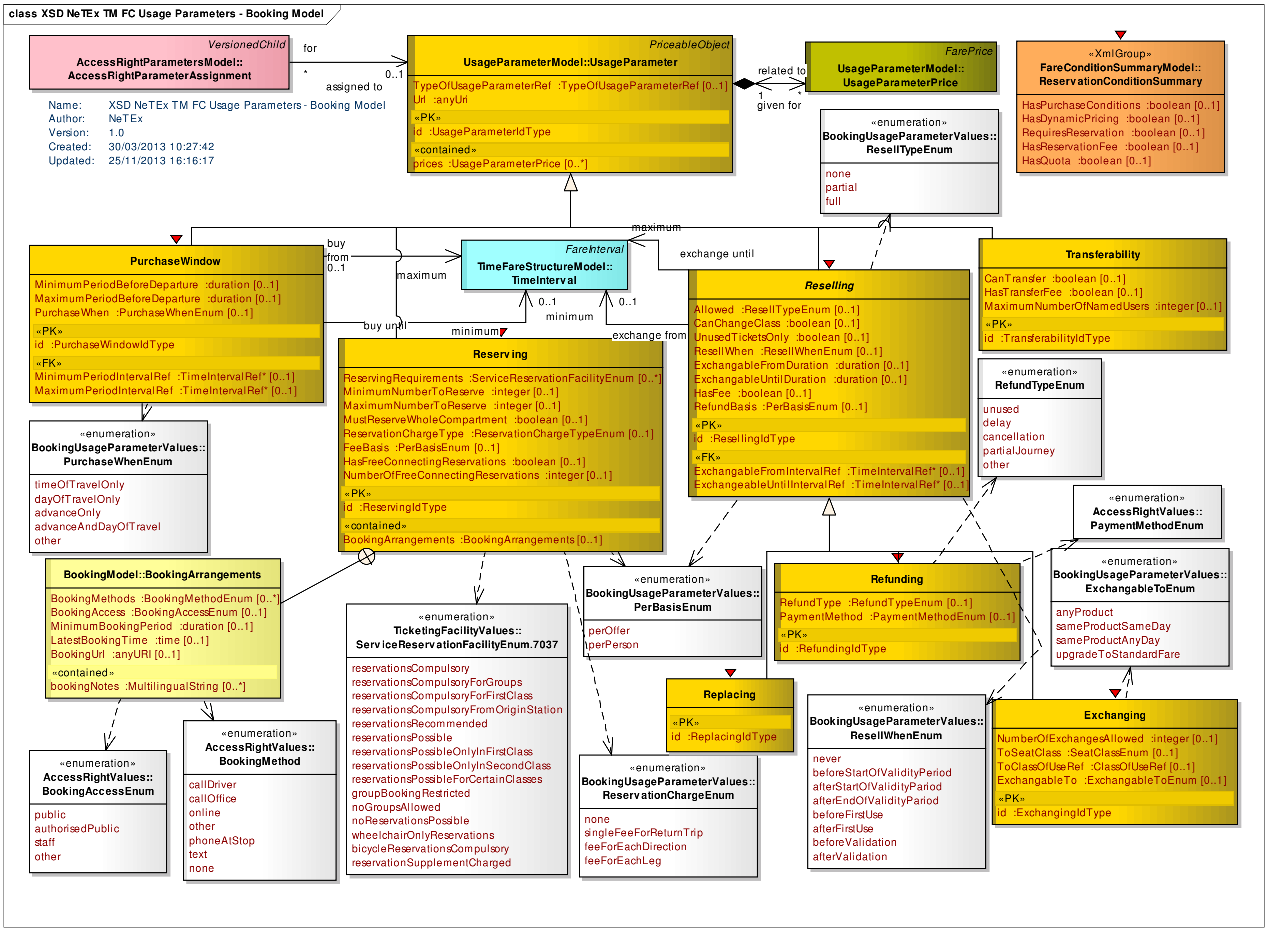


Figure 163 —Usage Parameters: Booking – Physical Model

8.6.1.3.4 Usage Parameter: General – Attributes and XSD

8.6.1.3.4.1 UsageParameter – Model Element

A parameter used to specify the use of a SALES PACKAGE or a FARE PRODUCT.

Table 107 – *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*	1:1	Url associated with parameter.
	TypeOf-UsageParameter-Ref	*TypeOf-UsageParameterRef*	0:1	Type of USAGE PARAMETER.
“cntd»	***priceTables***	*FareTable*	0:*	FARE TABLEs for the USAGE PARAMETER.
“cntd»	***prices***	*UsageParameterPrice*	0:*	USAGE PARAMETER PRICEs for the USAGE PARAMETER.

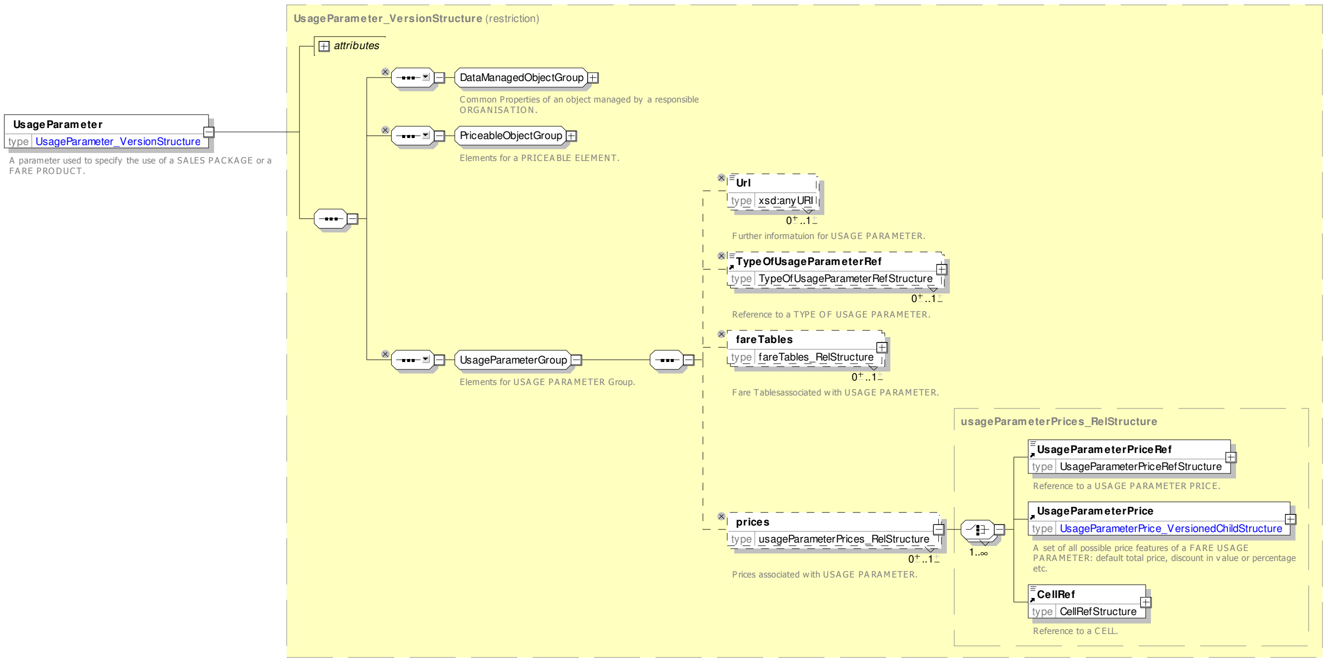


Figure 164 — *UsageParameter* — XSD

8.6.1.3.4.2 UsageParameterPrice – Model Element

A set of all possible price features of a USAGE PARAMETER: discount in value or percentage etc.

Table 108 – *UsageParameterPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	USAGE PARAMETER PRICE inherits from FARE PRICE

«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.

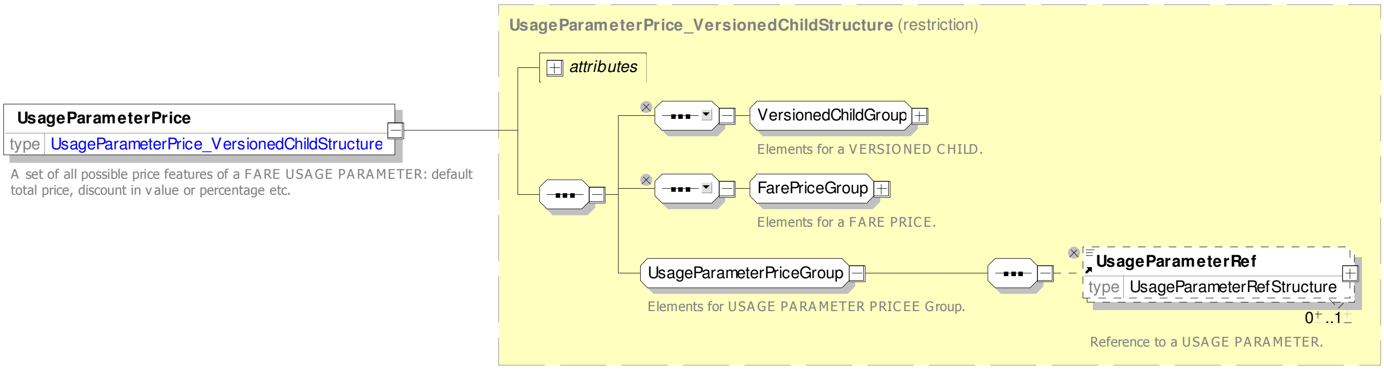


Figure 165 — *UsageParameterPrice* — XSD

8.6.1.3.4.3 TypeOfUsageParameter – Model Element

A classification of USER PROFILE by type of person eligible to use it

Table 109 – *TypeOfUsageParameter* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF USAGE PARAMETER inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*TypeOfUsahgeParameterIdType*	1:1	Identifier of TYPE OF USAGE PARAMETER.

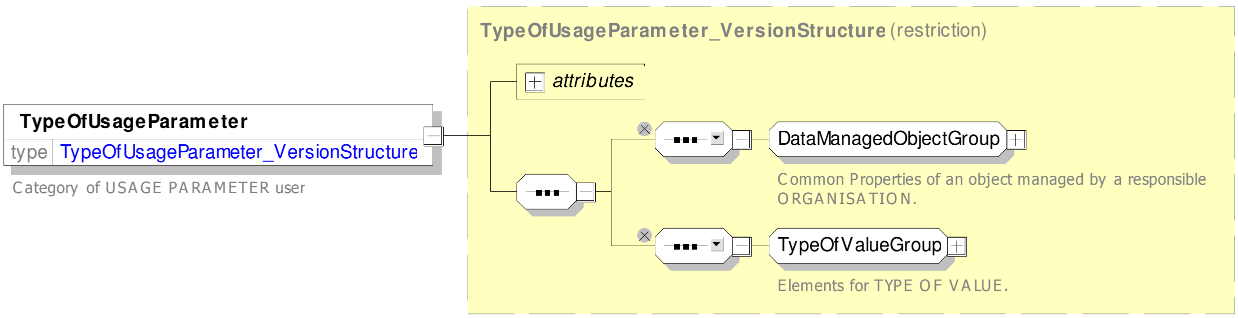


Figure 166 — *TypeOfUsageParameter* — XSD

8.6.1.3.5 Usage Parameter: Travel – Attributes and XSD

8.6.1.3.5.1 RoundTrip – Model Element

Properties relating to single or return trip use of a fare.

Table 110 – *RoundTrip* – Element

Classification	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.

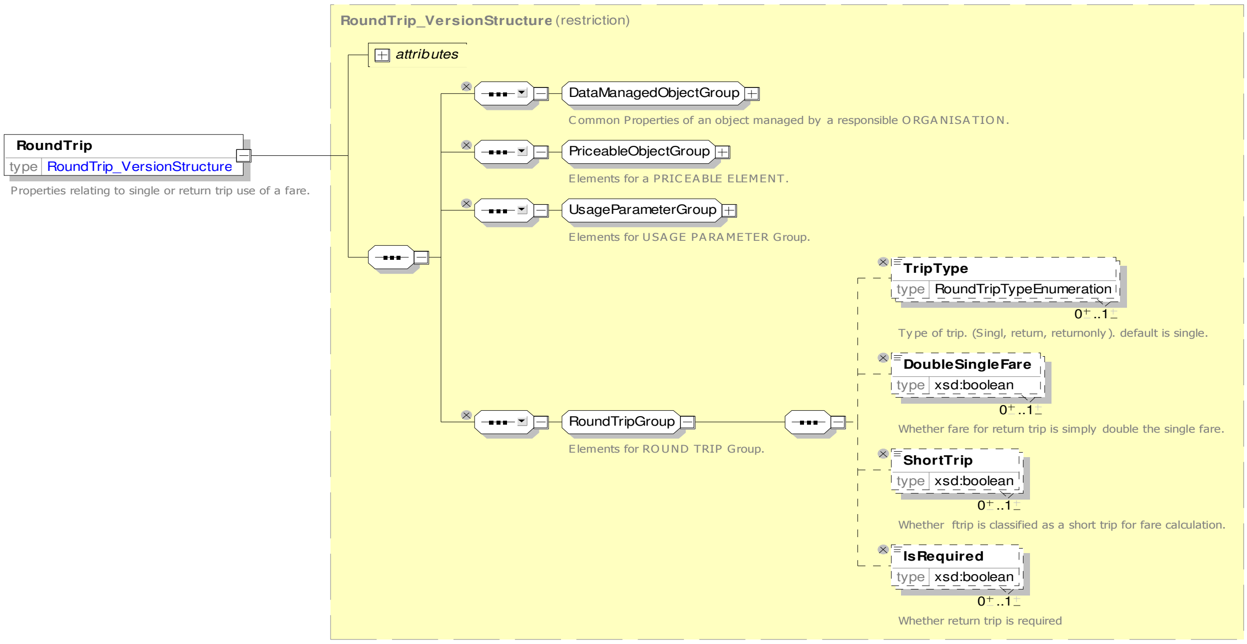


Figure 167 — *RoundTrip* — XSD

8.6.1.3.5.1.1 *RoundTripType* – Allowed values

The following table shows the allowed values for ***RoundTripType*** *(RoundTripTypeEnum)*.

Table 111 – *RoundTripType* – Allowed values

Value	Description
single	Single trip.

return	Outbound and return trip.
returnOnly	returnOnly
multiple	mMultiple trip.

8.6.1.3.5.2 Routing – Model Element

Limitations on routing of a fare.

Table 112 – *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	Return route must be same as outbound route.
	ForwardsOnly	*xsd:boolean*	0:1	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:xsd:boolean*	0:1	Whether the routing is across a border.

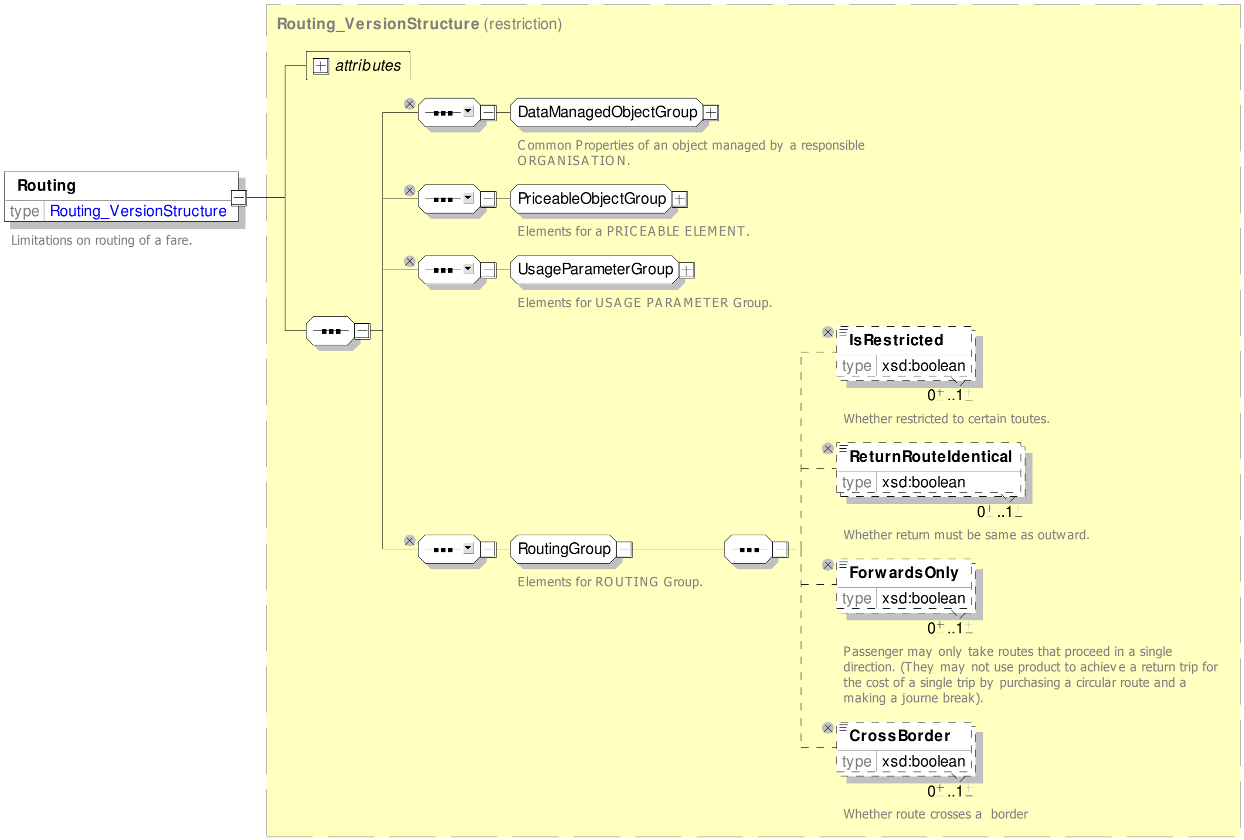


Figure 168 — *Routing* — XSD

8.6.1.3.5.3 FrequencyOfUse – Model Element

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

Table 113 – *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.
	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.
	DiscountBasis	*DiscountBasisEnum*	0:1	Nature of discount for number of journeys. See allowed values below.

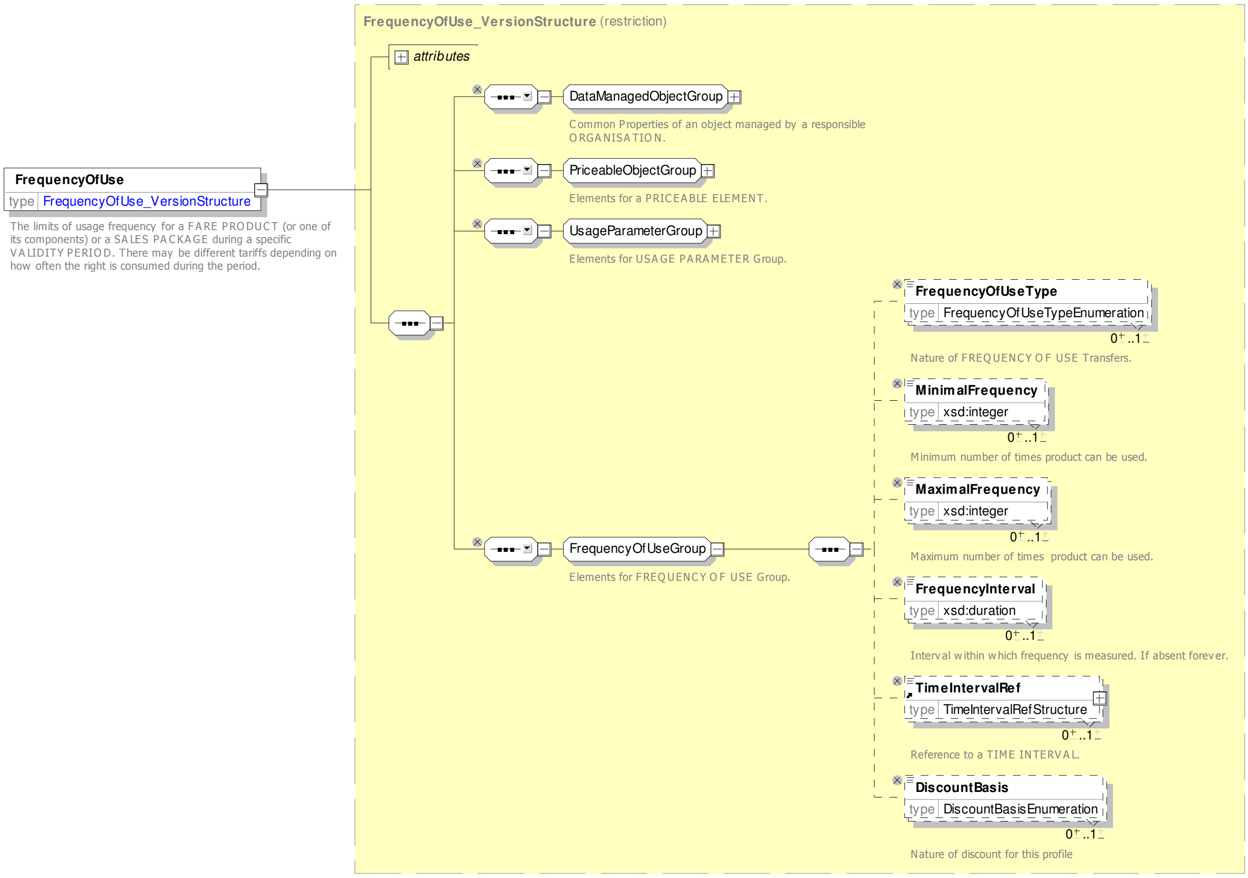


Figure 169 — *FrequencyOfUse* — XSD

8.6.1.3.5.3.1 FrequencyOfUseType – Allowed values

The following table shows the allowed values for ***FrequencyOfUseType*** (*FrequencyOfUseEnum*).

Table 114 – *FrequencyOfUseType* – Allowed values

Value	Description
none	No changes allowed
single	One change allowed.
limited	Limited number of changes allowed
unlimited	Unlimited number of changes allowed

8.6.1.3.5.3.2 DiscountBasis – Allowed values

The following table shows the allowed values for ***DiscountBasis*** (*DiscountBasisEnum*).

Table 115 – *DiscountBasis* – Allowed values

Value	Description
none	No companion allowed.
free	Companion allowed for free
discount	Companion allowed at discount

8.6.1.3.5.4 Interchanging – Model Element

Limitations on making changes within a trip.

Table 116 – *Interchanging* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	INTERCHANGING inherits from USAGE PARAMETER.
«PK»	***id***	*InterchangingIdType*	1:1	Identifier of INTERCHANGING.
	FromMode	*VehicleModeEnum*	0:1	TRANSPORT MODE from which user is interchanging. See Netex Part1 for allowed values.
	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:xsd:boolean*	0:1	Whether the journey can be broken at an interchange point.
	CrossBorder	*xsd:xsd:boolean*	0:1	Whether the interchange is across a border.

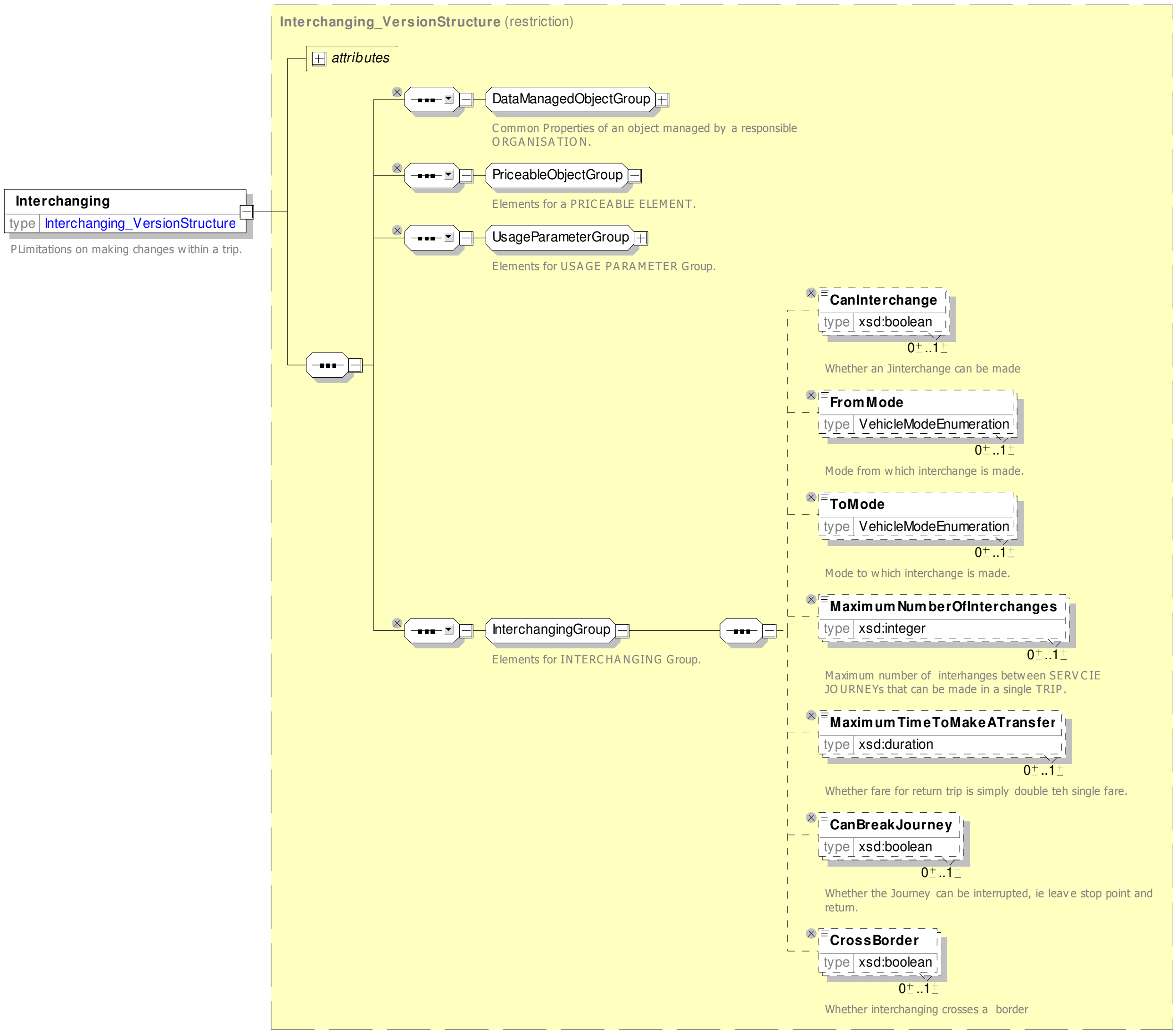


Figure 170 — *Interchanging* — XSD

8.6.1.3.5.5 UsageValidityPeriod – Model Element

A time limitation for validity of a FARE PRODUCT or a SALES PACKAGE. It may be composed of a standard duration (e.g. 3 days, 1 month) and/or fixed start/end dates and times.

Table 117 – *UsageValidityPeriod* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	USAGE VALIDITY PERIOD inherits from USAGE PARAMETER.
«PK»	***id***	*UsageValidity-PeriodIdType*	1:1	Identifier of USAGE VALIDITY PERIOD.
	ValidityType	*ValidityTypeEnum*	0:*	Type of USAGE VALIDITY PERIOD. See allowed values below.

	UsageTrigger	*UsageTriggerEnum*	0:1	Trigger event that starts validity period. See allowed values below.
	UsageEnd	*Usage*	0:*	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
	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.
	BlackoutUse	*BlackoutStartEnum*	0:1	When start of travel restriction applies. See allowed values below.

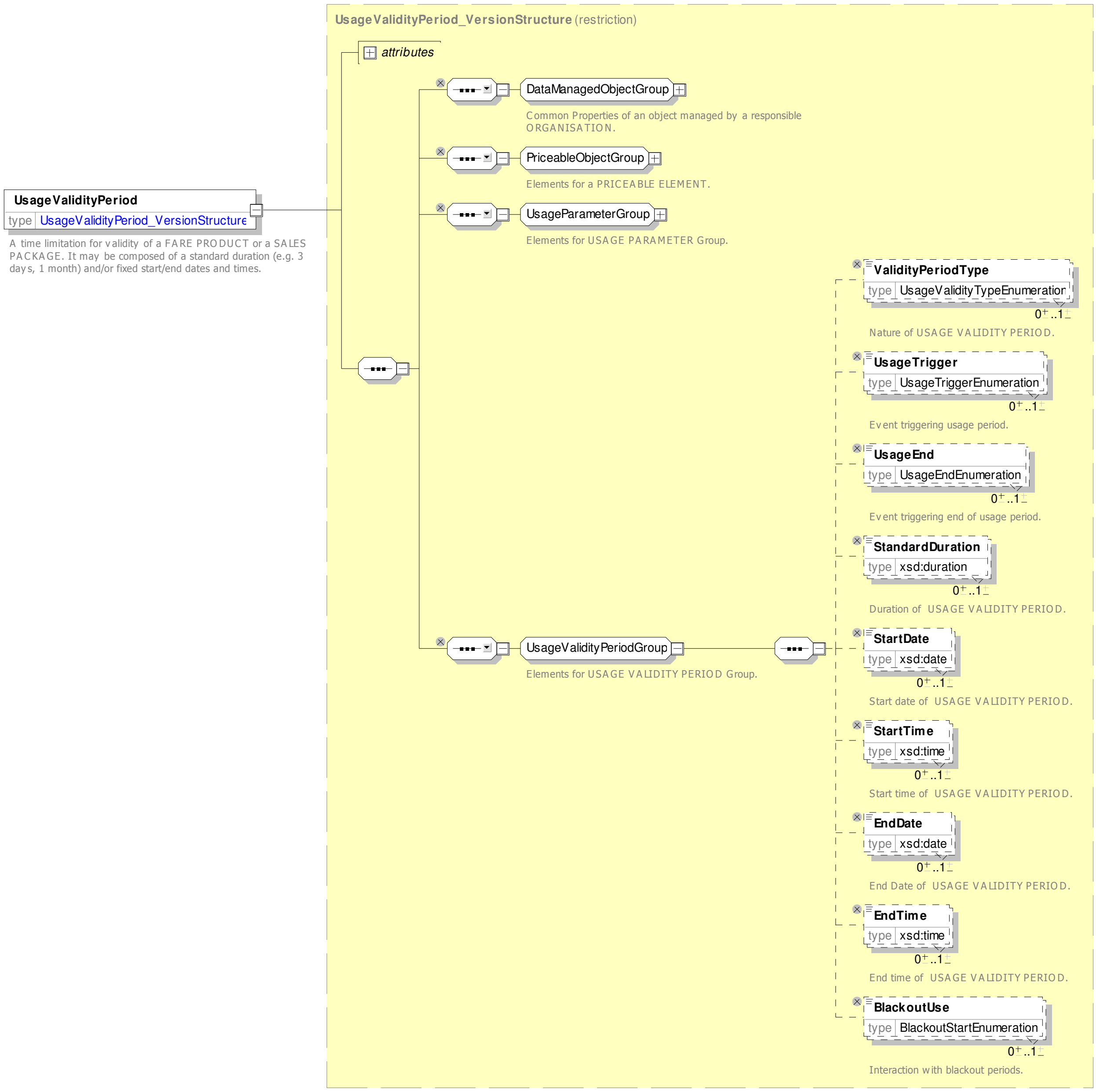


Figure 171 — *UsageValidityPeriod* — XSD

8.6.1.3.5.5.1 UsageValidityType – Allowed values

The following table shows the allowed values for ***UsageValidityType*** (*UsageValidityTypeEnum).*

Table 118 – *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.
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

8.6.1.3.5.5.2 UsageTrigger – Allowed values

The following table shows the allowed values for ***UsageTrigger*** (*UsageTriggerEnum*).

Table 119 – *UsageTrigger* – Allowed values

Value	Description
startOutboundRide	Start of outbound trip.
endOutboundRide	End of outbound trip.
startReturnRide	Start of return trip
purchase	Starts on purchase.
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.

8.6.1.3.5.5.3 UsageEnd – Allowed values

The following table shows the allowed values for ***UsageEnd*** (*UsageEndEnum*).

Table 120 – *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	Ticket valid to end of product - for a travel card withe expiry date.
profileExpiry	Ticket valid while member of a profile. Stops when ends.
other	Other Validity period.

8.6.1.3.5.5.4 BlackoutStart – Allowed values

The following table shows the allowed values for ***BlackoutStart*** condition (*BlackoutStartEnum*).

Table 121 – *BlackoutStart* – Allowed values

Value	Description
noTravelWithinPeriod	No travel permitted within exclusion period
mayCompleteIfStartedBefore	Outward and return journeys may be completed if started before exclusion period

8.6.1.3.5.6 MinimumStay – Model Element

Details of any minimum stay at the destination required to use the product.

Table 122 – *MinimumStay* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	MINIMUM STAY inherits from USAGE PARAMETER.
«PK»	***id***	*MinimumStayIdType*	1:1	Identifier of MINIMUM STAY parameter.
	Minimum-StayType	*MinimumStayEnum*	0:1	Nature of Minimum stay requirements. See allowed values below.
	RequiredNightsAway	*DayOfWeekEnum*	0:7	Specific nights which must be spent away. See NeTEx Part1.
	Minimum-NumberOfNightsAway	*xsd:integer*	0:1	Minimum number of nights that must be spent away.
	Maximum-NumberOfNightsAway	*xsd:integer*	0:1	Minimum number of nights that can be spent away.

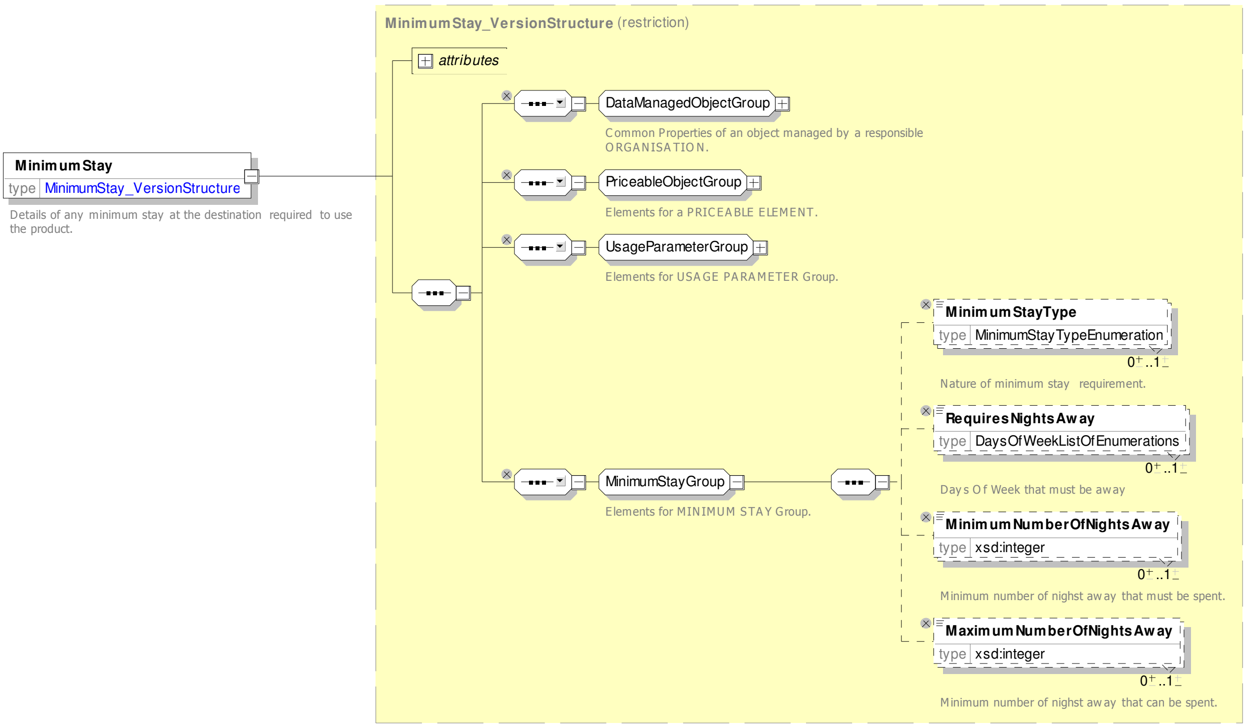


Figure 172 — *MinimumStay* — XSD

8.6.1.3.5.6.1 MinimumStay – Allowed values

The following table shows the allowed values for ***MinimumStay*** (*MinimumStayEnum*).

Table 123 – MinimumStay – Allowed values

Value	Description
none	Starts on purchase.
specifiedNightsAway	Must spend specified nights away
countNightsAway	Must spend the specified number of nights away
someSpecifiedNightsAway	Must spend the specified number of nights away which must be from the specified nights.

8.6.1.3.6 Usage Parameter: Product – Attributes and XSD

8.6.1.3.6.1 GroupTicket – Model Element

The number and characteristics of persons entitled to travel in addition to the holder of an access right.

Table 124 – *GroupTicket* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	GROUP TICKET inherits from USAGE PARAMETER.

«PK»	***id***	*GroupTicketIdType*	1:1	Identifier of GROUP TICKET.
«FK»	***TypeOf-ConcessionRef***	*TypeOfConcessionRef*	0:1	Type of concession to which this group applies.
GROUP	***GroupTicket-SizeGroup***	*GroupTicketSizeGroup*	0:1	Elements relating to size of group.
	PricingBasis	*PerBasisEnum*	0:1	Basis on which pricing is done - per whole group or per member. See allowed values below.
	Maximum-PersonsFree	*xsd:integer*	0:1	Number of persons allowed free on ticket.
	Maximum-Persons-Discounted	*xsd:integer*	0:1	Number of persons for which a group is discount allowed.
	DiscountOnly-ForFirstPerson	*xsd:boolean*	0:1	Whether there is only a discount for the first person in the group
	JointCheckIn	*GroupCheckInEnum*	0:1	Whether joint check in is required. See allowed values below.
	GroupBooking-Facility	*GroupBookingEnum*	0:1	Type of Group Booking allowed. See NeTEx Part1.

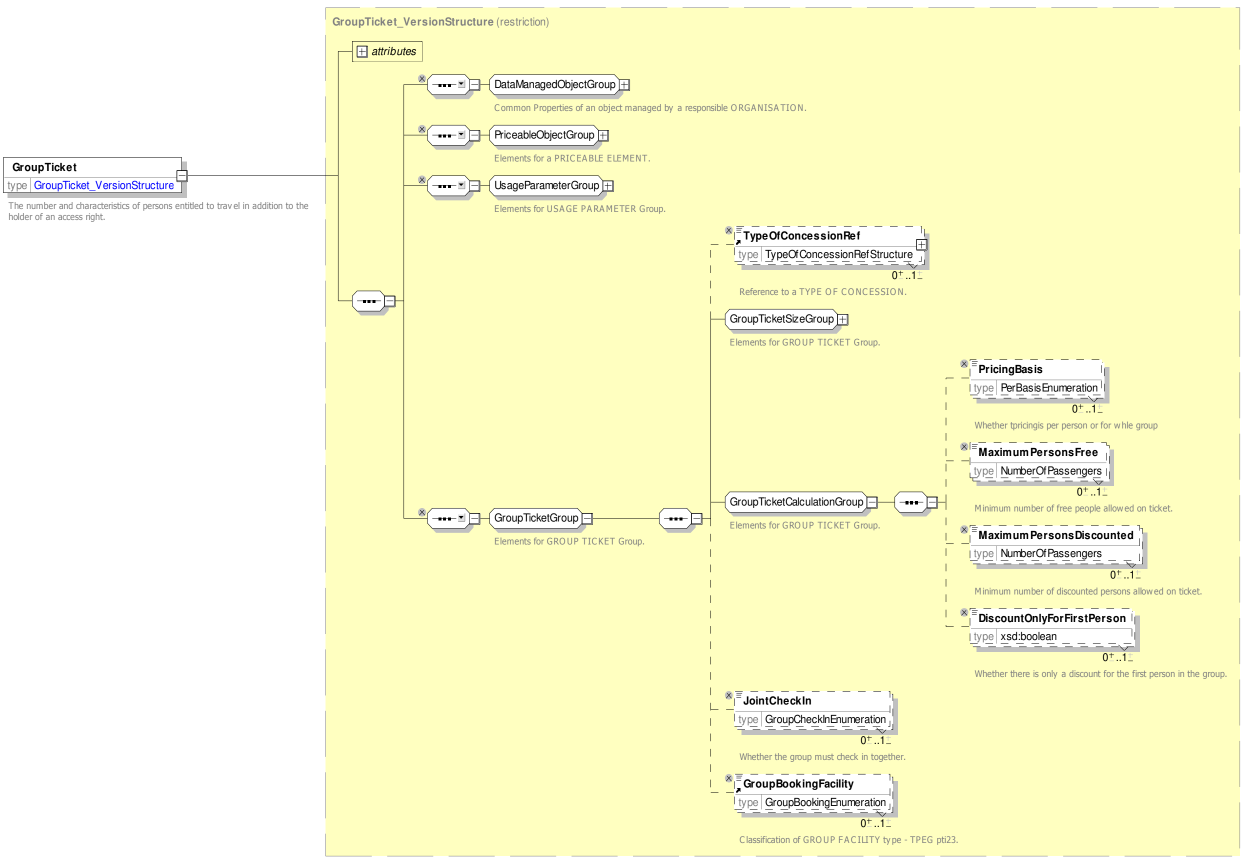


Figure 173 — *GroupTicket* — XSD

8.6.1.3.6.2 GroupTicketSizeGroup – Group

The ***GroupTicketSizeGroup*** specifies the number and characteristics of persons entitled to travel in addition to the holder of an access right.

Table 125 – *GroupTicketSizeGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Minimum-NumberOf-Persons	*xsd:integer*	0:1	Minimum number of persons overall allowed on GROUP TICKET.
	Maximum-NumberOf-Persons	*xsd:integer*	0:1	Maximum number of persons overall allowed on GROUP TICKET.
	Minimum-NumberOf-CardHolders	*xsd:integer*	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.

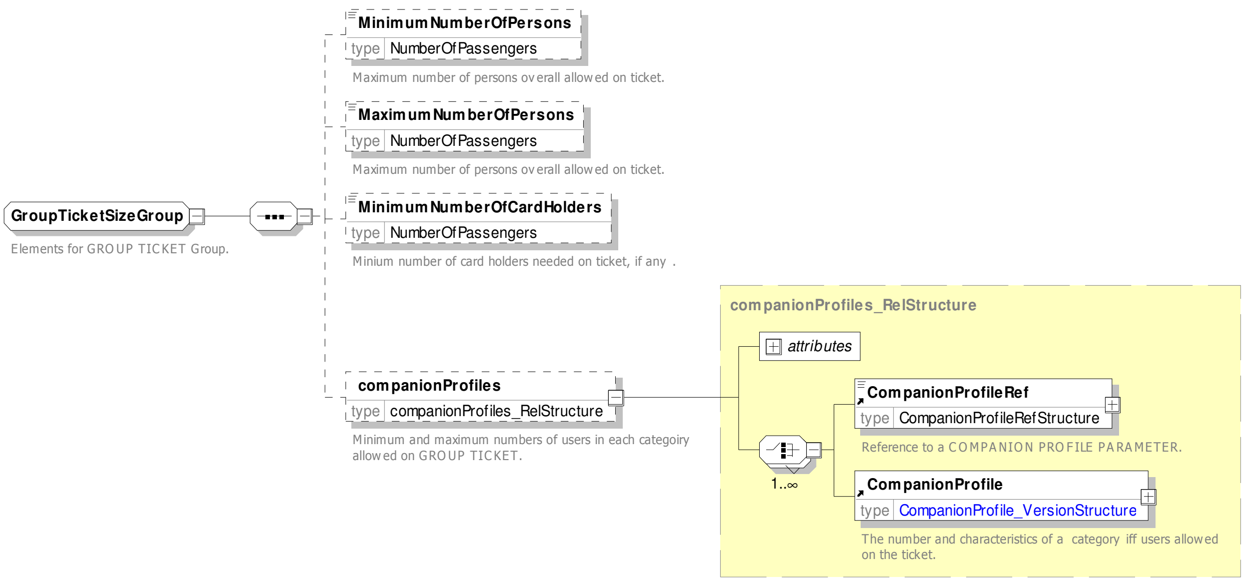


Figure 174 — *GroupTicketSizeGroup* — XSD

8.6.1.3.6.2.1 GroupCheckIn – Allowed values

The following table shows the allowed values for ***GroupCheckIn*** (*GroupCheckInEnum*).

Table 126 – *GroupCheckIn* – Allowed values

Value	Description

none	No group check in
required	Passengers must check in together
allowed	Passengers may check in together.

8.6.1.3.6.2.2 GroupDiscountBasis – Allowed values

The following table shows the allowed values for group discount ***GroupDiscountBasis***. (*GroupDiscount-BasisEnum*).

Table 127 – *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.

8.6.1.3.6.3 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 128 – *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.
	UserType	*UserTypeEnum*	0:1	Classification of user type.
GROUP	***UserProfile-Qualification-Group***	*UserProfile-QualificationGroup*	0:1	Elements describing eligibility conditions for user.

	GenderLimitation	*GenderLimitationList*	0:1	Gender required by USER PROFILE. Relevant for single sex accommodation products.
	ProofRequired	*ProofOfIdentityEnum*	0:1	Proof required for type of user.
	DiscountBasis	*DiscountBasisEnum*	0:1	Nature of discount for this type of user. See earlier.
	companion-Profiles	*CompanionProfile*	0:*	COMPANION PROFILEs describing users who may travel with user.

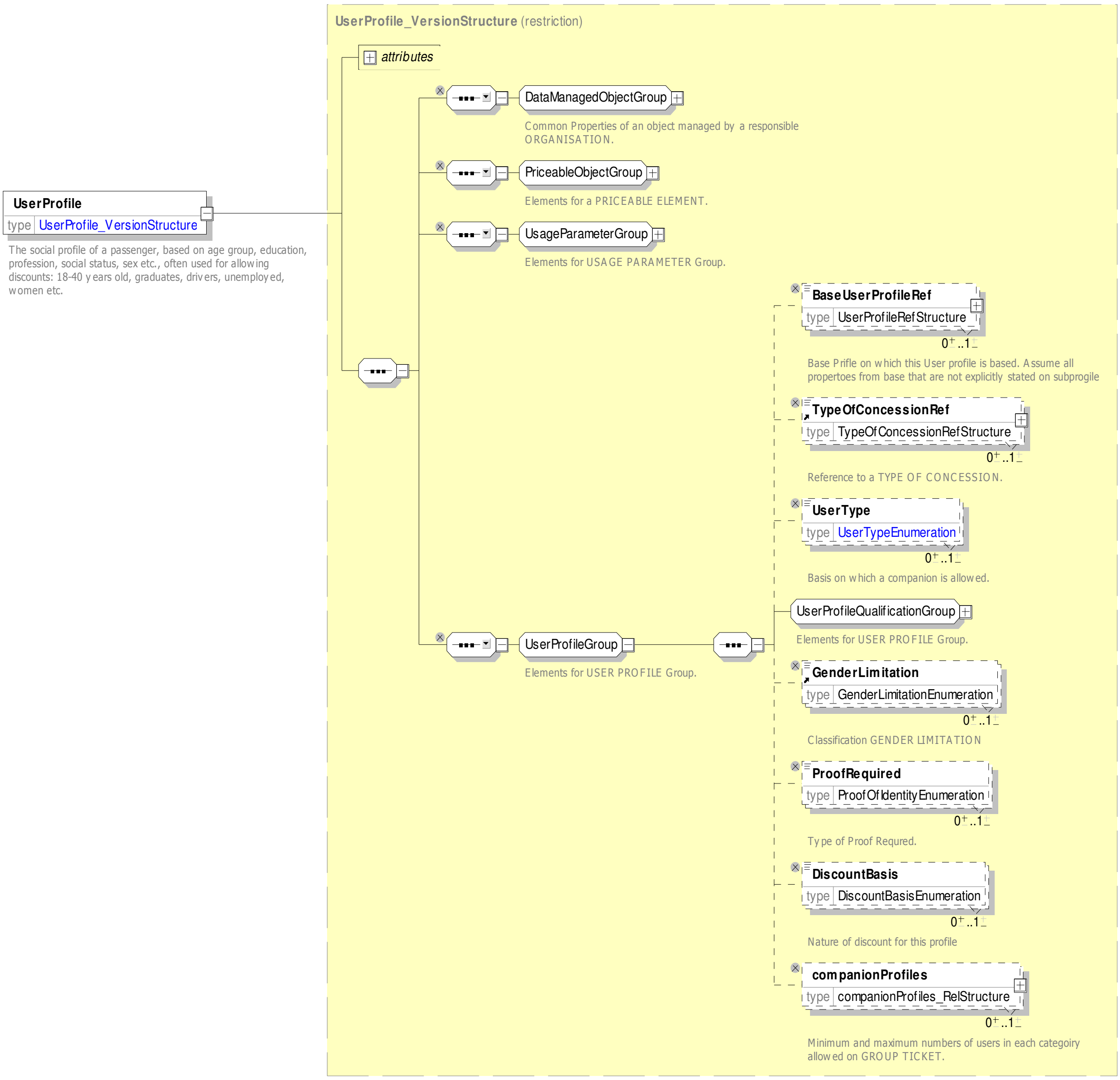


Figure 175 — *UserProfile* — XSD

8.6.1.3.6.3.1 *UserProfileQualificationGroup* – Group

The ***UserProfileQualificationGroup*** specifies attributes describing the eligibility of a user to belong to a USER PROFILE.

Table 129 – *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 helath 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. Default is true.
	resides	*ResidentialQualification*	0:*	RESIDENTIAL QUALIFICATIONs for USER PROFILE – if more than one, these will be logically ORed together.

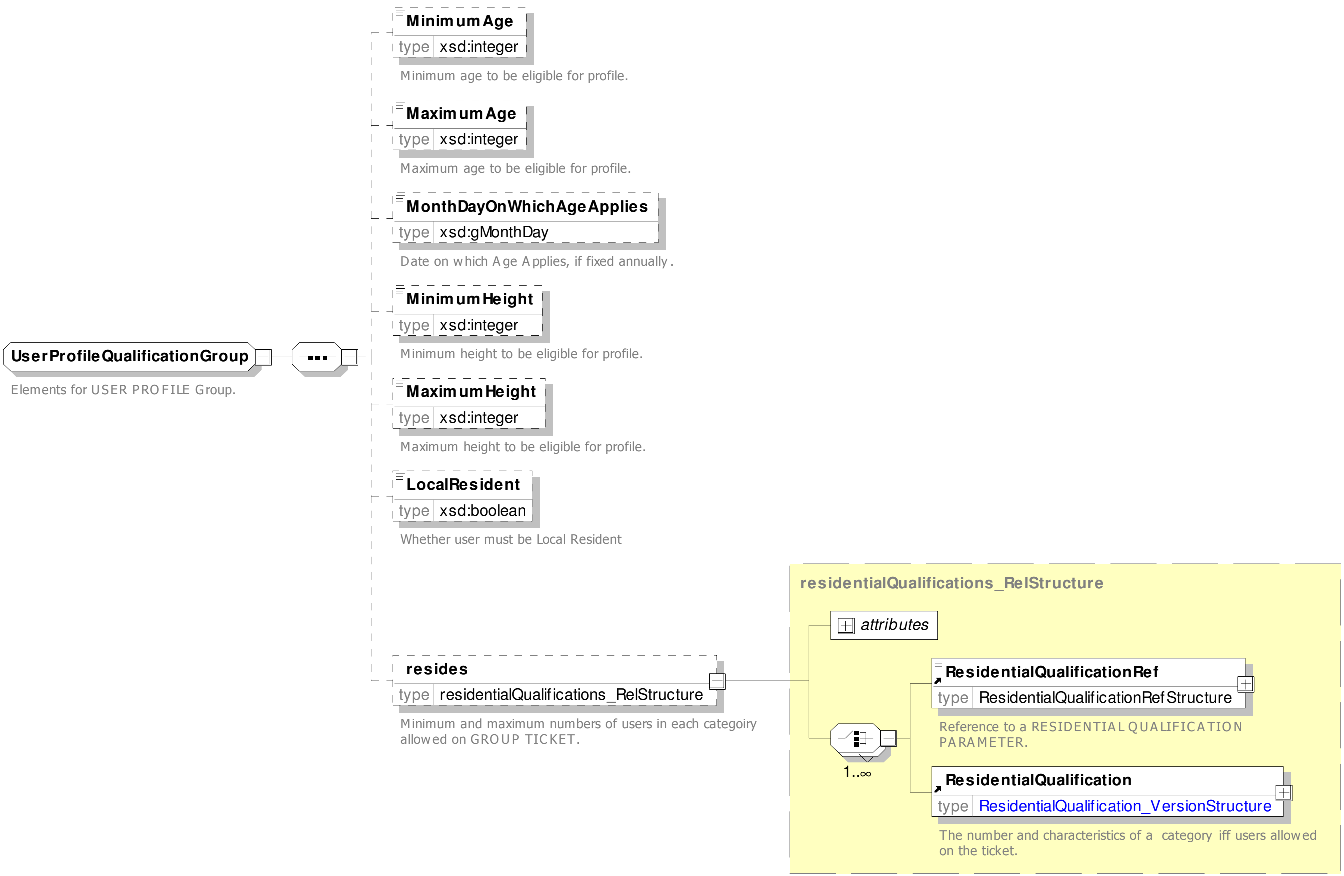


Figure 176 — *UserProfileQualificationGroup*— XSD

8.6.1.3.6.3.2 ResidentialQualification – Model Element

The RESIDENTIAL QUALIFICATION element describes a requirement to live in a certain area.

Table 130 – *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.
	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.
	Topographical-PlaceRef	*TopographicalPlaceRef*	0:1	TOPOGRAPHIC PLACE for which residency rule applies. See NeTEx Part1.

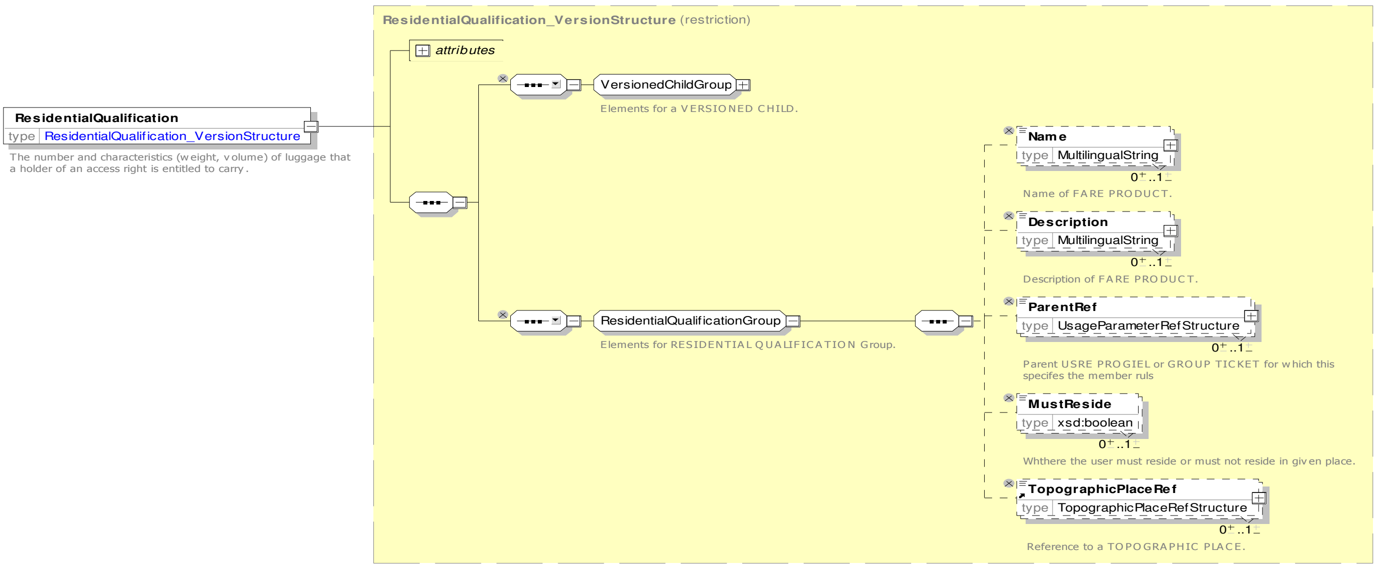


Figure 177 — *ResidentialQualification* — XSD

8.6.1.3.6.3.3 ProofRequired – Allowed values

The following table shows the allowed values for The following table shows the allowed values for ***ProofRequired*** (*ProofOfIdentityEnum*).

Table 131 – *ProofRequired* – Allowed values

Value	Description
noneRequired	No proof required.
passport	Proof is to show a passport.
drivingLicence	Proof is to show an driving licence.
membershipCard	Proof is to show an Identify document. such as a passport or driving licence.
studentCard	Proof is to show an student card.
identityDocument	Proof is to show an Identify document.
creditCard	Proof is to show a credit card.
medicalDocument	Proof is to show an medical document or letter from a medical authority.
letterWIthAddress	Proof is to show a letter or bill from a an organisation to applicant's address.
measurement	Height measurement.
other	Other proof.

8.6.1.3.6.4 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 132 – *CompanionProfile* – Element

Classifi-cation	Name	Type	Cardinality	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.
	MinimumNumberOfPersons	*xsd:integer*	0:1	Minimum number of persons overall allowed of this type.
	MaximumNumberOfPersons	*xsd:integer*	0:1	Maximum number of persons overall allowed of this type.
	DiscountBasis	*DiscountBasisEnum*	0:1	Nature of discount for this type of user. See allowed values earlier.

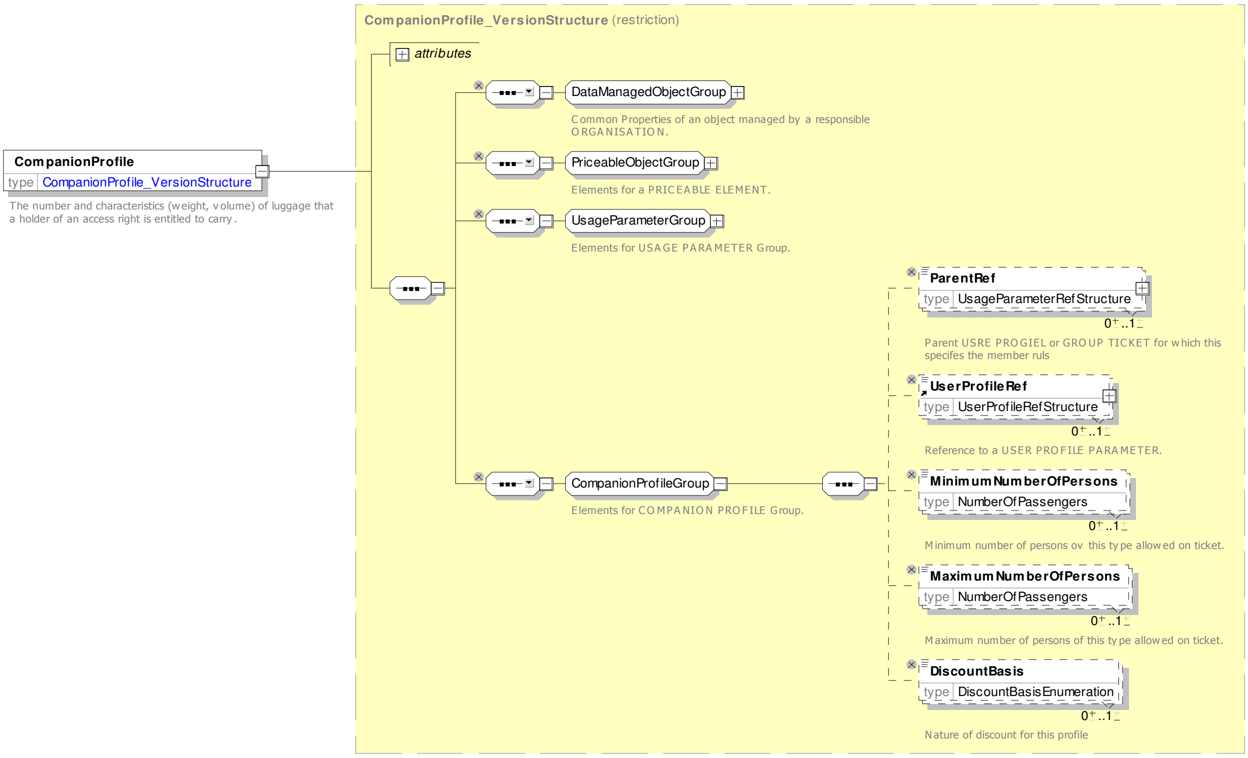


Figure 178 — *CompanionProfile* — XSD

8.6.1.3.6.5 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 133 – *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-Amountr	*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*		GROUP OF ORGANISATIONs or other entities associated with the COMMERCIAL PROFILE.

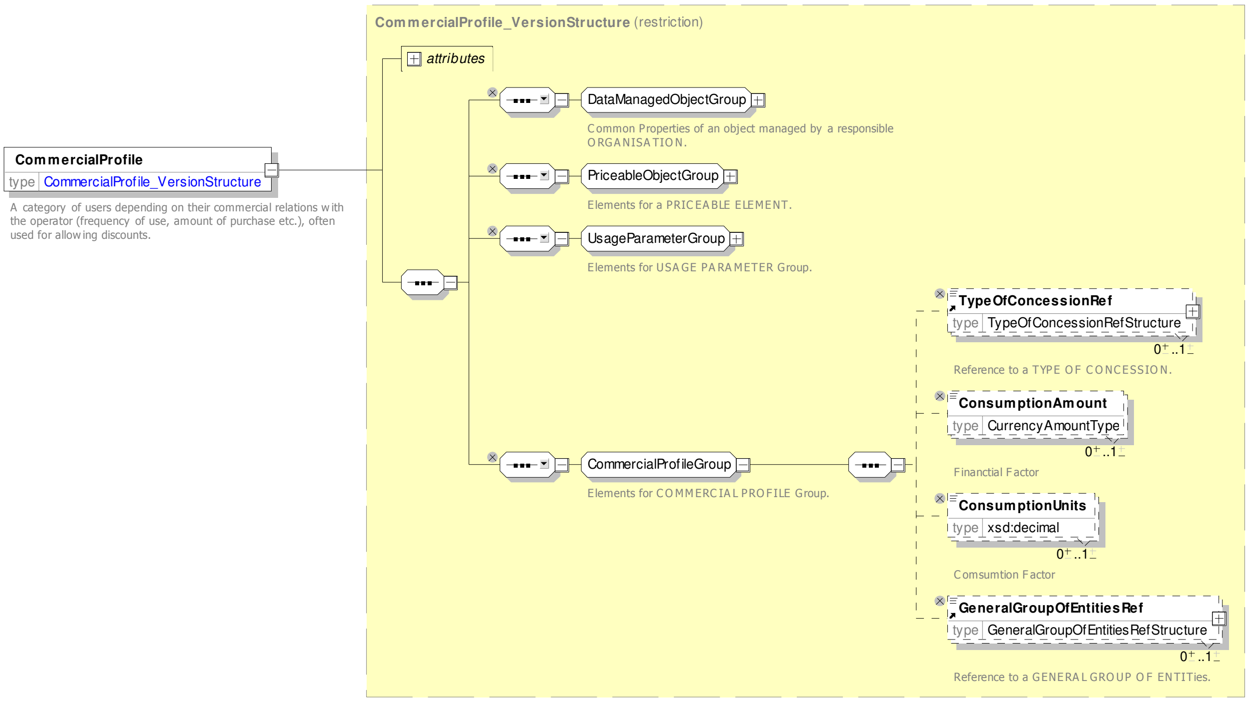


Figure 179 — *CommercialProfile* — XSD

8.6.1.3.6.6 EntitlementRequired – Model Element

Receiving of entitlement from another FARE PRODUCT .

Table 134 – *EntitlementRequired* – Element

Classifi-cation	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:*	Minimum period that required product must be held in order to be eligible.

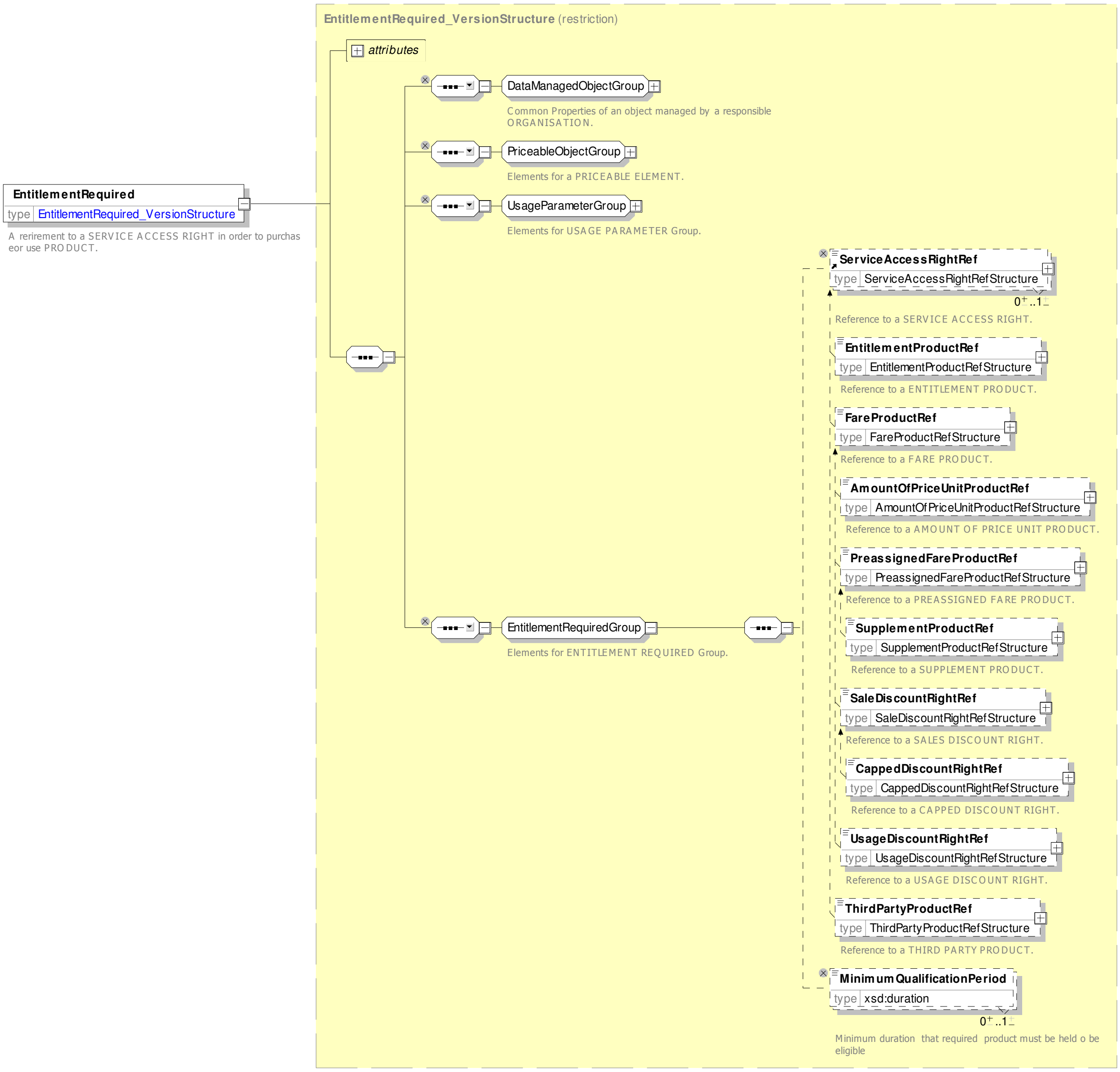


Figure 180 — *EntitlementRequired* — XSD

8.6.1.3.6.7 EntitlementGiven – Model Element

Granting of entitlement to another FARE PRODUCT.

Table 135 – *EntitlementGiven* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	ENTITLEMENT GIVEN inherits from USAGE PARAMETER.
«PK»	***id***	*EntitlementGivenIdType*	1:1	Identifier of ENTITLEMENT GIVEN.
«FK»	***ServiceAccess-***	*ServiceAccessRightRef*	0:1	Entitlement comes from the referenced FARE

	RightRef			PRODUCT.
	EntitlementType	*EntitlementTypeEnum*	0:1	Type of entitlement. See allowed values below.
	Minimum-Qualification-Period	*xsd:duration*	0:1	Minimum period that product must be held for entitlement to be granted.

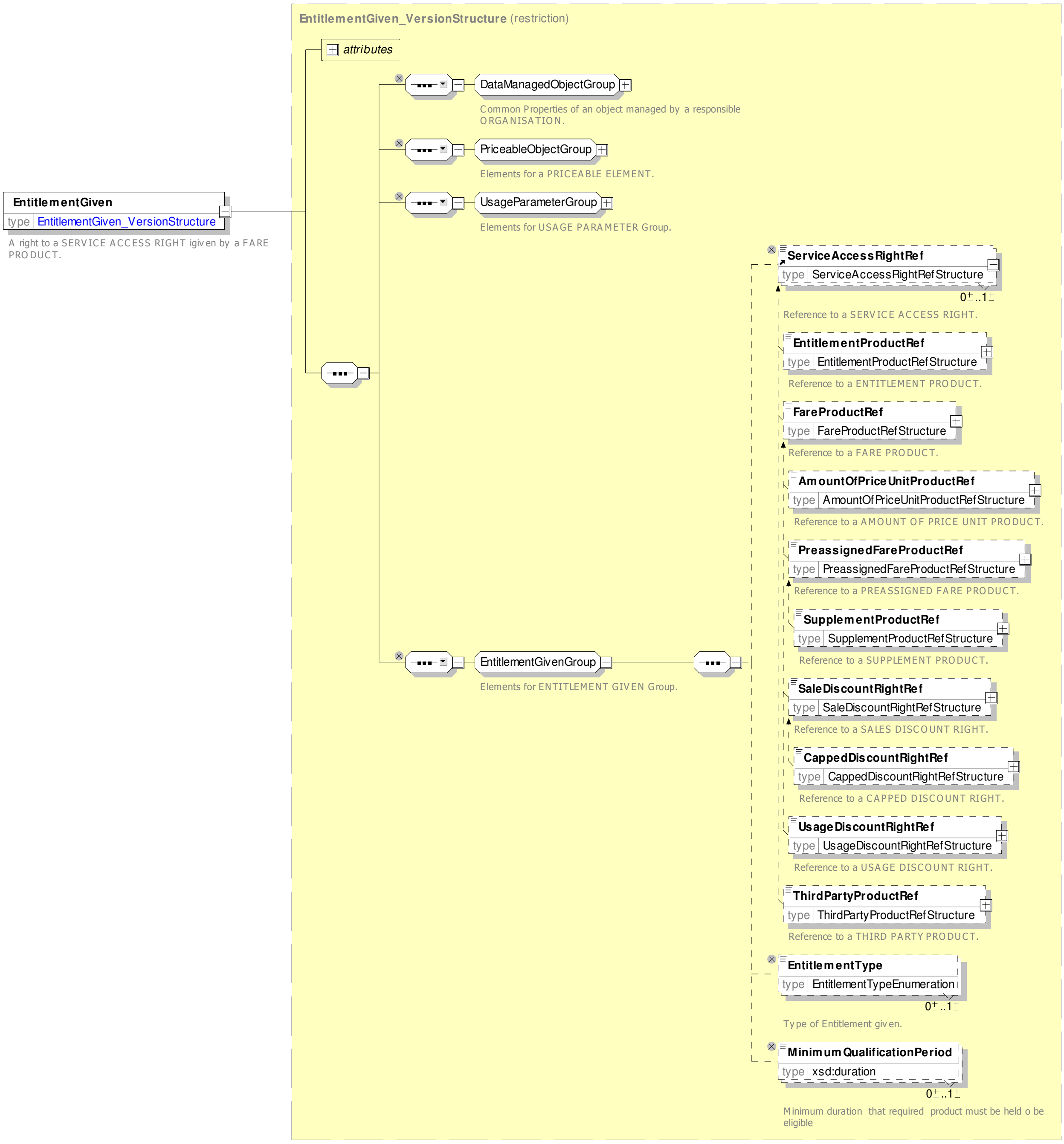


Figure 181 — *EntitlementGiven* — XSD

8.6.1.3.6.7.1 EntitlementType – Allowed values

The following table shows the allowed values for ***EntitlementType*** (*EntitlementTypeEnum*)

Table 136 – *EntitlementType* – Allowed values

Value	Description
use	Entitlement is to use product.
purchase	Entitlement is to purchase product.
none	No entitlement.

8.6.1.3.6.8 TypeOfConcession – Model Element

A classification of USER PROFILE by type of person eligible to use it

Table 137 – *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»	***Alternative-Names***	*AlternativeName*	0:*	Alternative names for VALUE.

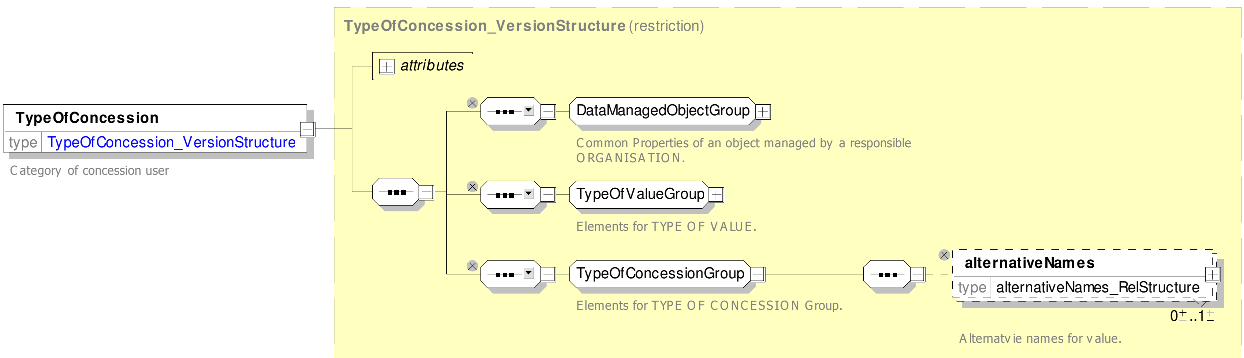


Figure 182 — *TypeOfConcession* — XSD

8.6.1.3.7 Usage Parameter: Luggage – Attributes and XSD

8.6.1.3.7.1 LuggageAllowance – Model Element

The number and characteristics (weight, volume) of luggage that a holder of an access right is entitled to carry.

Table 138 – *LuggageAllowance* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	LUGGAGE ALLOWANCE inherits from USAGE PARAMETER.
«PK»	***id***	*LuggageAllowance-IdType*	1:1	Identifier of LUGGAGE ALLOWANCE.
	BaggageUse-Type	*BaggageUseEnum*	0:1	Use of baggage covered by the allowance. See allowed values below.
	BaggageType	*LuggageUseEnum*	0:1	Type of baggage covered by the allowance. See allowed values below.
	Luggage-AllowanceType	*LuggageAllowanceEnum*	0:1	Classification of allowance type. See allowed values below.
	Maximum-NumberOfItems	*xsd:nonNegativeInteger*	0:1	Number of bags allowed.
	Weight	*WeightType*	0:1	Total Weight limit of LUGGAGE ALLOWANCE.
	Maximum-BagHeight	*LengthType*	0:1	Maximum bag height.
	Maximum-BagWidth	*LengthType*	0:1	Maximum bag width.
	Maximum-BagDepth	*LengthType*	0:1	Maximum bag depth.
	Maximum-BagWeight	*WeightType*	0:1	Maximum bag weight.

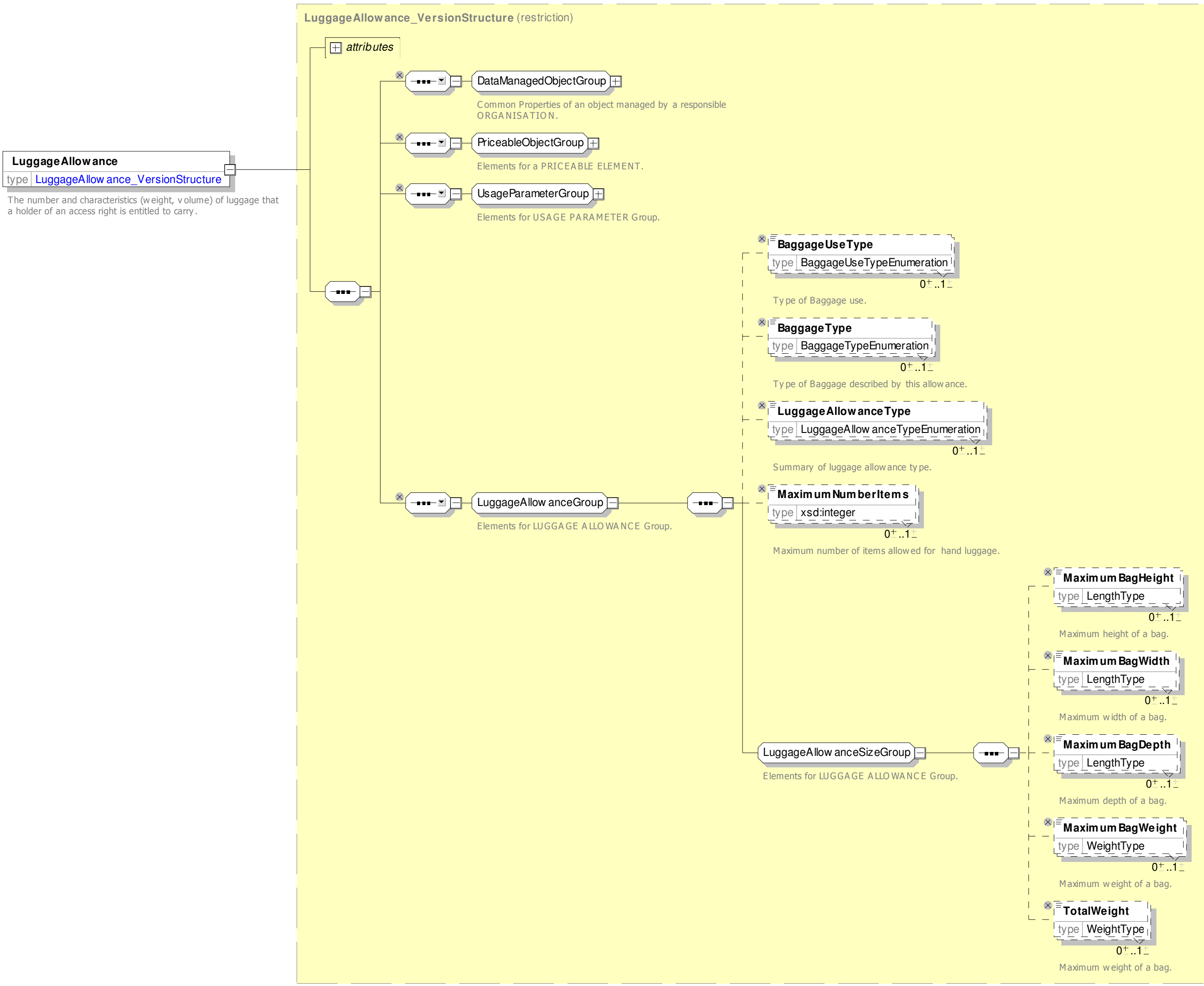


Figure 183 — *LuggageAllowance* — XSD

8.6.1.3.7.1.1 BaggageType – Allowed values

The following table shows the allowed values for ***BaggageType*** (*BaggageTypeEnum).*

Table 139 – *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.

8.6.1.3.7.1.2 LuggageAllowance – Allowed values

The following table shows the allowed values for ***LuggageAllowance*** (*LuggageAllowanceEnum).*

Table 140 – *LuggageAllowance* – Allowed values

Value	Description
none	Luggage is to carry on.
unlimited	Unlimited baggage allowance.
single	Single bag allowed.
limited	Baggage limited by restriction.

8.6.1.3.7.1.3 LuggageUse – Allowed values

The following table shows the allowed values for ***LuggageUse*** (*LuggageUseTypeEnum*)

Table 141 – *LuggageUseType* **– Allowed values**

Value	Description
carryOn	Luggage is to carry on.

checkIn	Luggage is to check in.
oversizeCheckIn	Oversize bag check in.

8.6.1.3.8 Usage Parameter: Booking – Attributes and XSD

8.6.1.3.9 PurchaseWindow – Model Element

Period in which the product must be purchased.

Table 142 – *PurchaseWindow* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	PURCHASE WINDOW inherits from USAGE PARAMETER.
«PK»	***id***	*PurchaseWindowIdType*	1:1	Identifier of PURCHASE WINDOW.
	PurchaseWhen	*PurchaseWhenEnum*	0:1	When purchase may be made. See allowed values below.
	MinimumPeriod-BeforeDeparture	*xsd:duration*	0:1	Minimum duration before departure that ticket may be purchased.
	MaximumPeriod-BeforeDeparture	*xsd:duration*	0:1	Maximum duration before departure that ticket may be purchased.

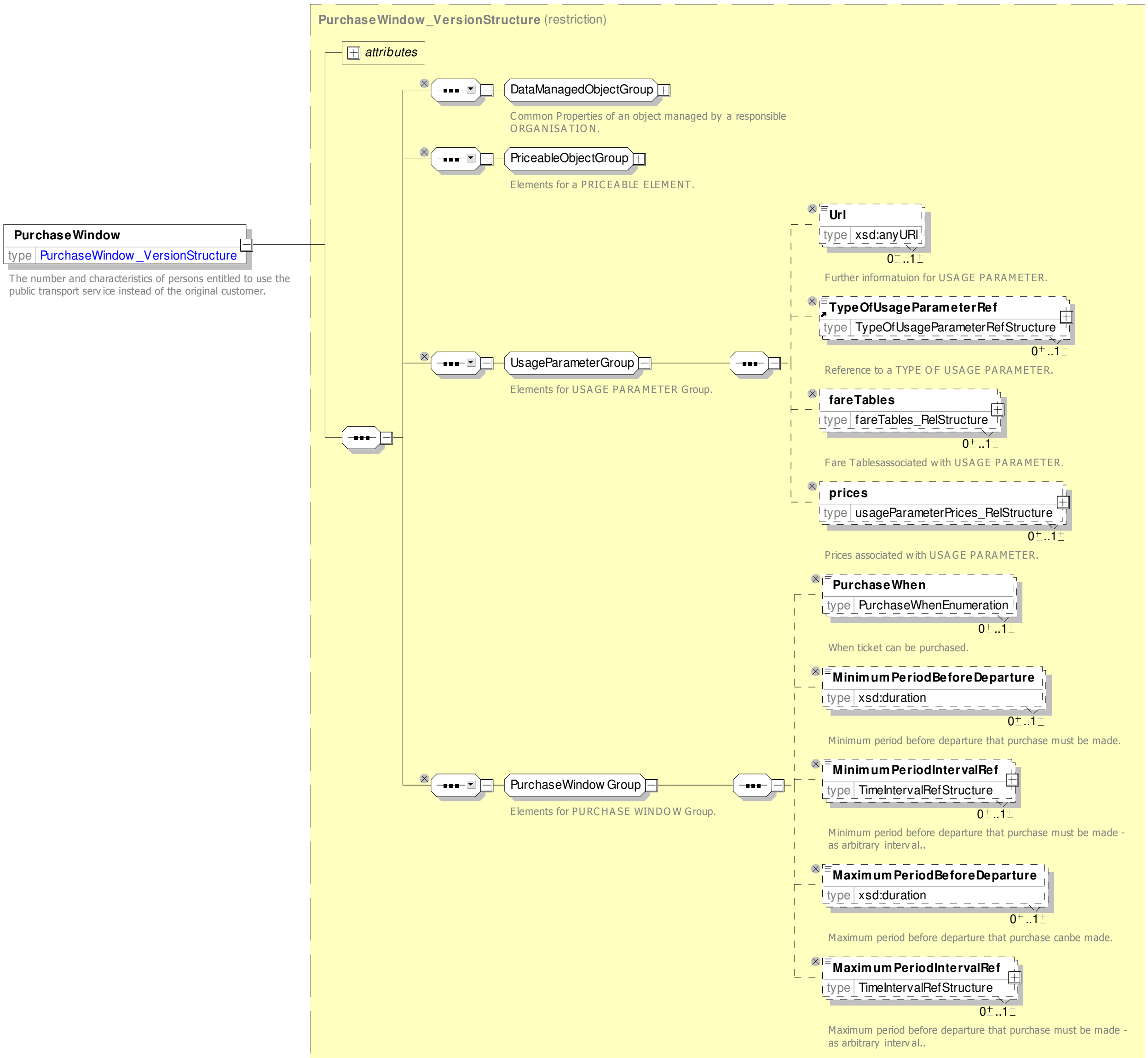


Figure 184 — *PurchaseWindow* — XSD

8.6.1.3.9.1.1 PurchaseWhen – Allowed values

The following table shows the allowed values for ***PurchaseWhen*** (*PurchaseWhenEnum*).

Table 143 – *PurchaseWhen* – Allowed values

Value	Description
timeOfTravelOnly	Purchase may only be made at time of travel.
dayOfTravelOnly	Purchase may only be made on day of travel.
advanceOnly	Purchase may only be made in advance.
advanceAndDayOfTravel	Purchase may be made in advance or on day of travel.

other	Other limitation on who may make e a booking

8.6.1.3.9.2 Transferability – Model Element

The number and characteristics of persons entitled to use the public transport service instead of the original customer.

Table 144 – *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-NamedUsers	*xsd:integer*	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.

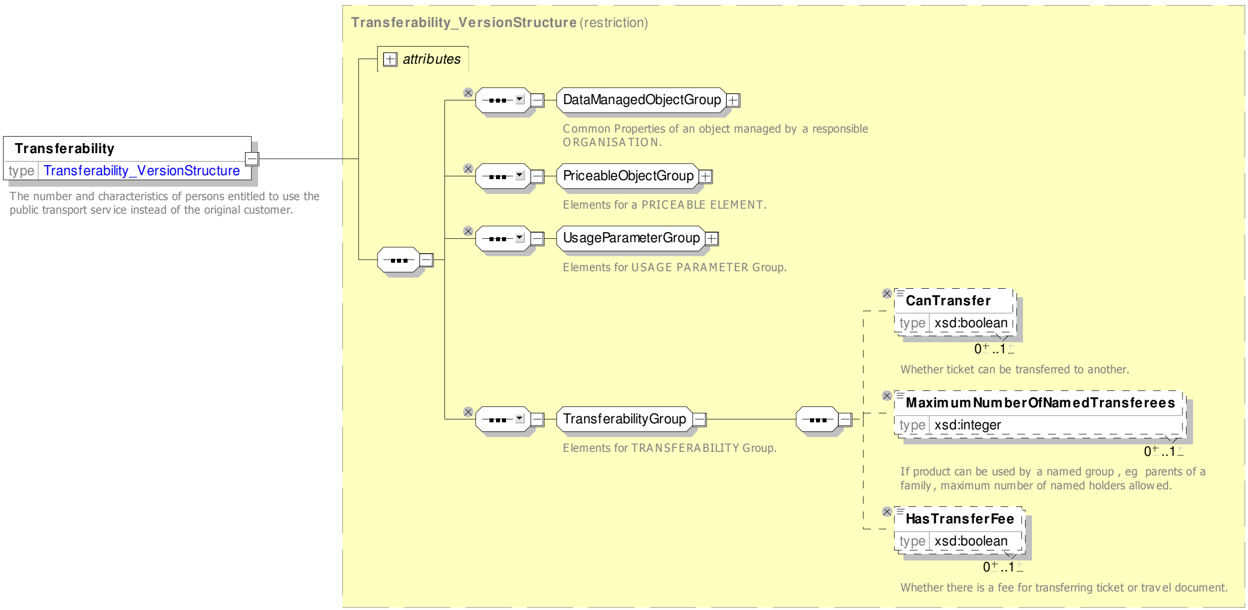


Figure 185 — *Transferability* — XSD

8.6.1.3.9.3 Reselling – Model Element

Common resale conditions (i.e. for exchange or refund) attaching to the product.

Table 145 – *Reselling* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	RESELLING inherits from USAGE PARAMETER.
«PK»	***id***	*ResellingIdType*	1:1	Identifier of RESELLING.
	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.
	ResellWhen	*ResellWhenEnum*	0:1	Event marking when the is exchangeable status of the ticket changes. See allowed values below.
	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) or that ticket may be exchanged or refunded
	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) that ticket may be exchanged or refunded
	HasFee	*xsd:boolean*	0:1	Whether these is a fee for a refund or exchange.
	RefundBasis	*PerBasisEnum*	0:1	Basis on which refund is made. See allowed values below.

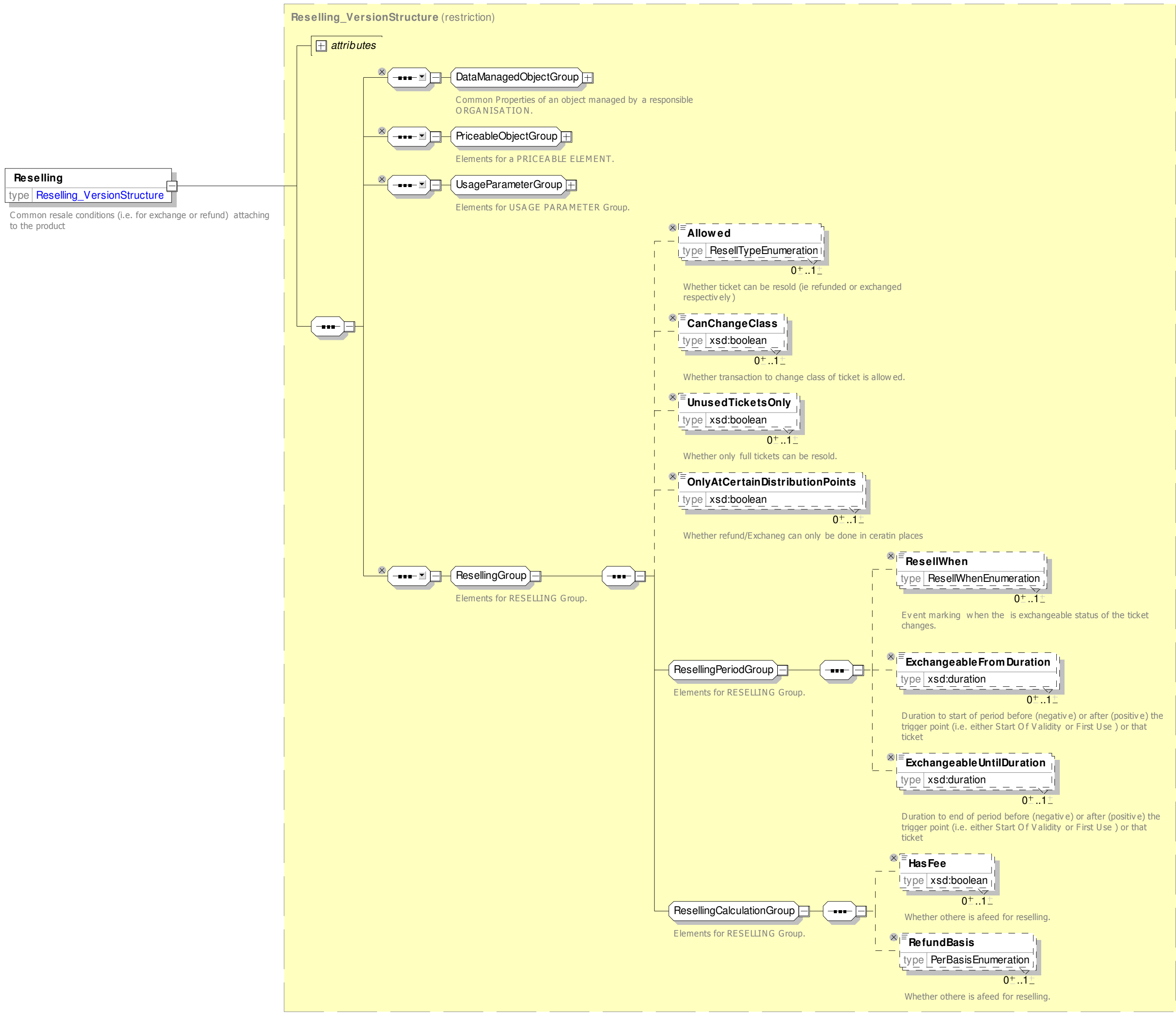


Figure 186 — *Reselling* — XSD

8.6.1.3.9.3.1 ResellWhen – Allowed values

The following table shows the allowed values for ***ResellWhen*** (*ResellWhenEnum*).

Table 146 – *ResellWhen* – Allowed values

Value	Description
never	No transaction allowed, i.e. Ticket can never be exchanged or refunded.
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.

8.6.1.3.9.3.2 ResellType – Allowed values

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

Table 147 – ResellTypeEnum – Allowed values

Value	Description
none	Ticket can never be exchanged or refunded.
partial	Partial refund or exchange allowed.
full	Full refund allowed.

8.6.1.3.9.4 Exchanging – Model Element

Whether and how the product may be exchanged for another product o

Table 148 – *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.
	ToSeatClass	*SeatClassEnum*	0:1	Seat class to which can be exchanged. See NeTEx Part1. (From class would be expression as the Seat class on an ACCESS RIGHT PARAMETER ASSIGNMENT)
	ToClass-OfUseRef	*ClassOfUseRef*	0:1	Seat class to which can be exchanged.
	ExchangableTo	*ExchangableToEnum*	0:1	Type of exchange allowed. Default is to any other fare.

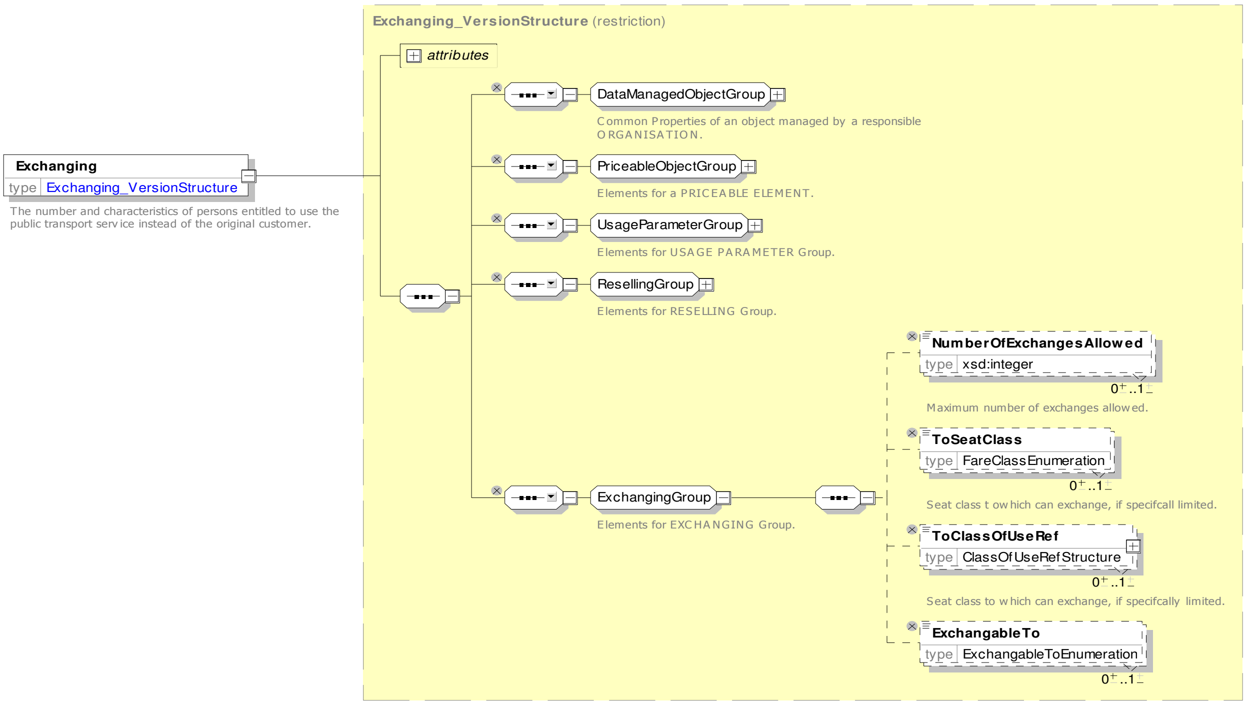


Figure 187 — *Exchanging* — XSD

8.6.1.3.9.4.1 ExchangableTo – Allowed values

The following table shows the allowed values for ***ExchangableTo*** *(ExchangableToEnum)*.

Table 149 – *ExchangableTo* – Allowed values

Value	Description
anyFare	Can exchange to any other fare.
sameFareSameDay	Can exchange to fares of the same type for travel on the same date.
sameFareAnyDay	Can exchange to fares of the same type for travel on the any date.
upgradeToStandardFare	Can exchange as upgrade to full standard fare.

8.6.1.3.9.5 Refunding – Model Element

Whether and how the product may be refunded.

Table 150 – *Refunding* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Reselling*	::>	REFUNDING inherits from RESELLING**.**
«PK»	***id***	*RefundingIdType*	1:1	Identifier of REFUNDING.

	RefundType	*RefundingTypeEnum*	1:1	Classifcation of REFUNDING. See allowed values below.
	PaymentMethod	*PaymentMethod-Enumeration*	0:*	PAYMENT METHODS THAT May be used for REFUNDING.

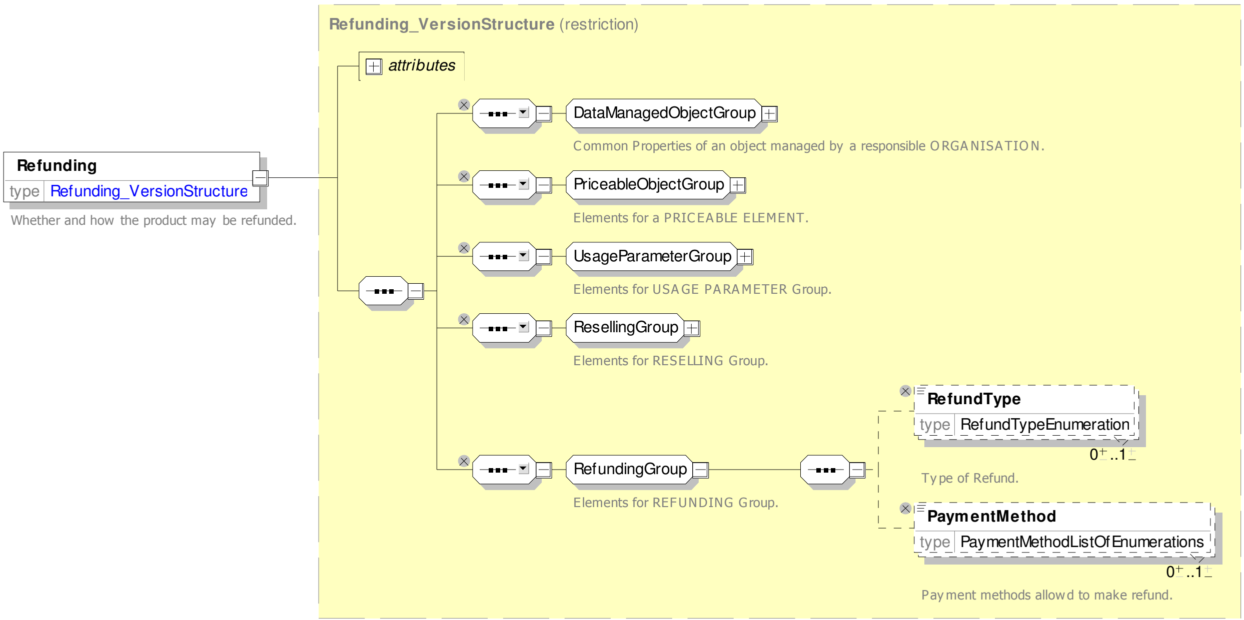


Figure 188 — *Refunding* — XSD

8.6.1.3.9.5.1 RefundType – Allowed values

The following table shows the allowed values for ***RefundType*** *(RefundTypeEnum)*.

Table 151 – *RefundType* – Allowed values

Value	Description
unused	Refund is because the product was unused.
delay	Refund is because the passengers trip was delayed.
cancellation	Refund is because the passenger journey was cancelled.
partialJourney	Refund is because the product was only party unused.
unused	Refund is because of some other reason.

8.6.1.3.9.6 Replacing – Model Element

Whether and how the product may be replaced if lost or stolen.

Table 152 – *Replacing* – Element

Classifi-	Name	Type	Cardinality	Description

cation				
::>	::>	*Reselling*	::>	REPLACING inherits from RESELLING.
«PK»	***id***	*ReplacingIdType*	1:1	Identifier of REPLACING.

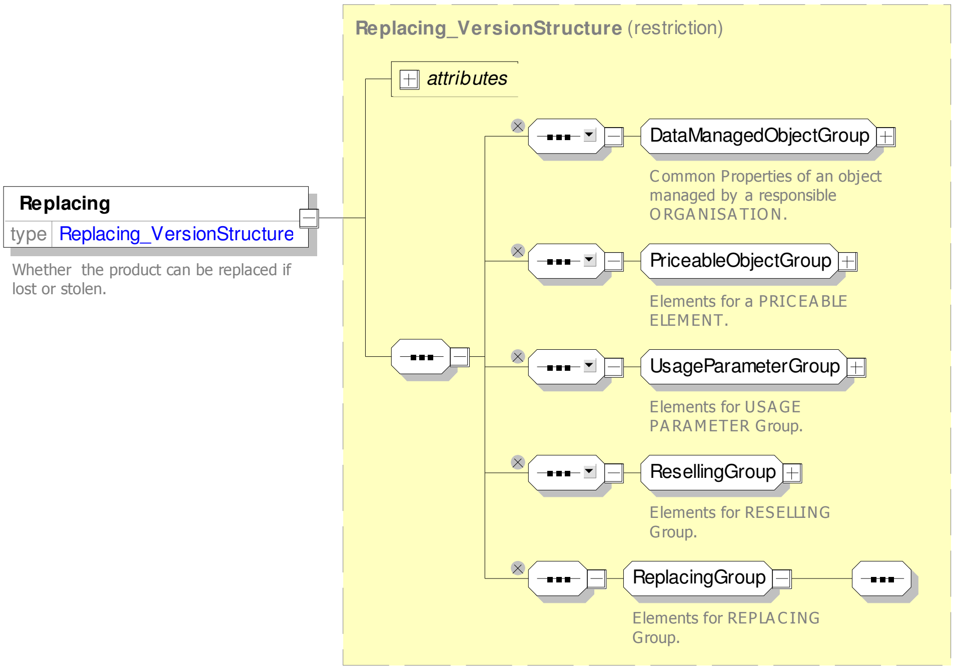


Figure 189 — *Replacing* — XSD

8.6.1.3.9.7 Reserving – Model Element

Limitations on making changes within a trip.

Table 153 – *Reserving* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*UsageParameter*	::>	RESERVING inherits from USAGE PARAMETER.
«PK»	***id***	*ReservingIdType*	1:1	Identifier of RESERVING.
	Reserving-Requirements	*ServiceReservation-FacilityEnum*	0:*	Nature of reservations required. See NeTEx Part1 for allowed values.
	Minimum-NumberToReserve	*xsd:integer*	0:1	Minimum number of persons allowed on a reservation.
	MaximumNumber-ToReserve	*xsd:integer*	0:1	Minimum number of persons allowed on a reservation.
	MustReserve-Whole-	*xsd:boolean*	0:1	Whether a whole compartment must be reserved.

	Compartment.			
	Reservation-ChargeType	*Reservation-ChargeTypeEnum*	0:1	Nature of reservation fee. See allowed values below.
	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.
“cntd»	***Booking-Arrangements***	*BookingArrangements*	0:1	Booking arrangements. See below.

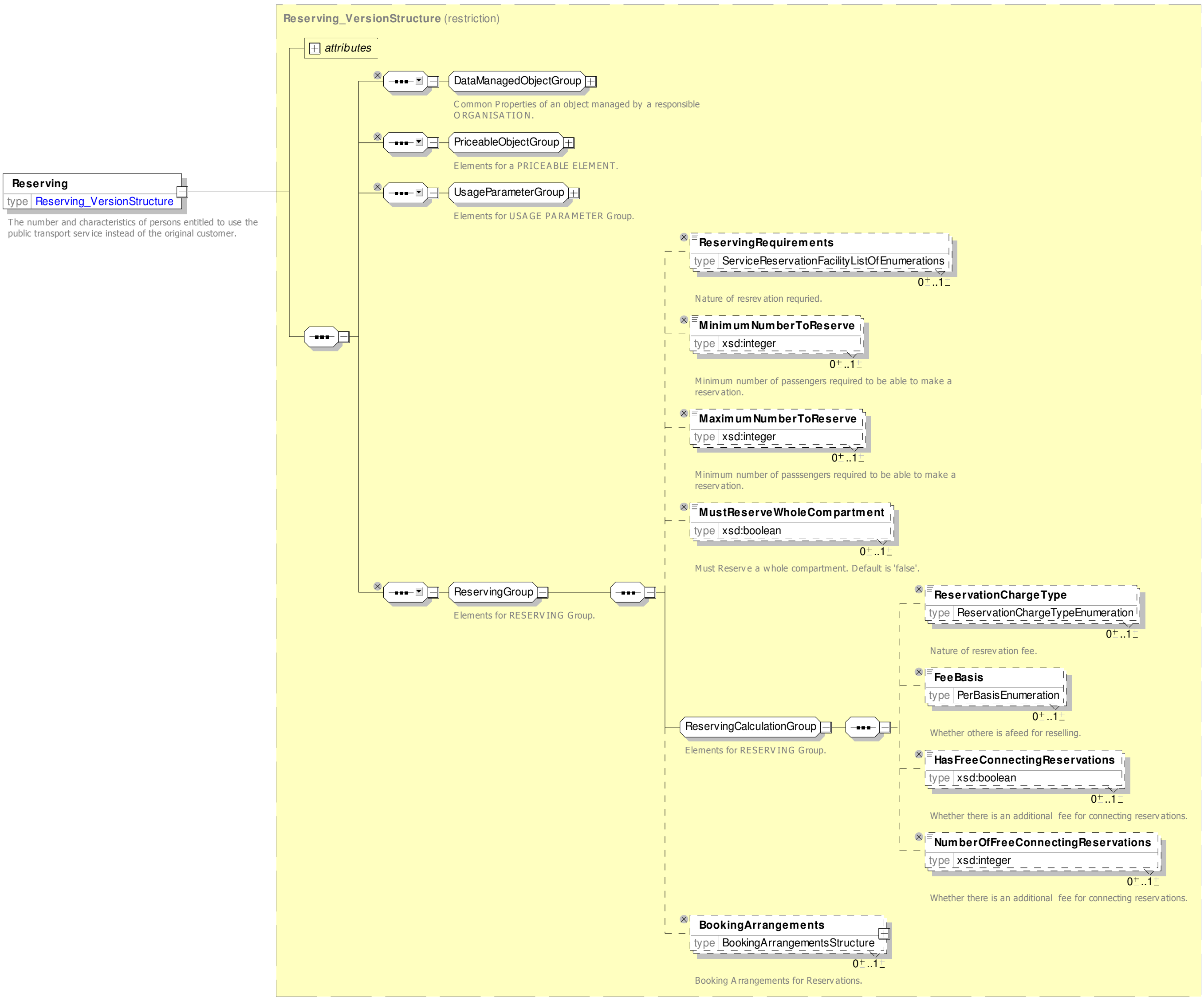


Figure 190 — *Reserving* — XSD

8.6.1.3.9.8 ReservationCharge – Allowed values

The following table shows the allowed values for ***ReservationCharge*** (*ReservationChargeEnum*).

Table 154 – ReservationCharge – Allowed values

Value	Description
none	No reservation fee.
singleFeeForReturnTrip	Refund is per person
feeForEachDirection	Separate reservation fee is for each direction of travel.
feeForEachLeg	Separate reservation fee is for each leg.

8.6.1.3.9.8.1 FeeBasis – Allowed values

The following table shows the allowed values for ***FeeBasis*** (*PerBasisEnum*).

Table 155 – PerBasis – Allowed values

Value	Description
perOffer	Refund is per offer
perPerson	Refund is per person

8.6.1.3.9.9 Booking Arrangements – Model Element

Details of the booking arrangements for a service.

Table 156 – *BookingArrangements* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
	BookingMethods	*BookingMethodEnum*	0:*	Booking method. See allowed values below.
	BookingAccess	*BookingAccessEnum*	0:1	Who can make a Booking. See allowed values below.
	LatestBooking-Time	*xsd:time*	0:1	Latest time in day that booking can be made.
	BookingUrl	*xsd:anyURI*	0:1	Booking url if different from info url (i.e. Contact URL).
	Minimum-BookingPeriod	*xsd:duration*	0:1	Minimum interval in advance of departure day or time that service may be ordered.

“cntd»	***bookingNotes***	*MultilingualString*	0:*	Additional notes about booking.

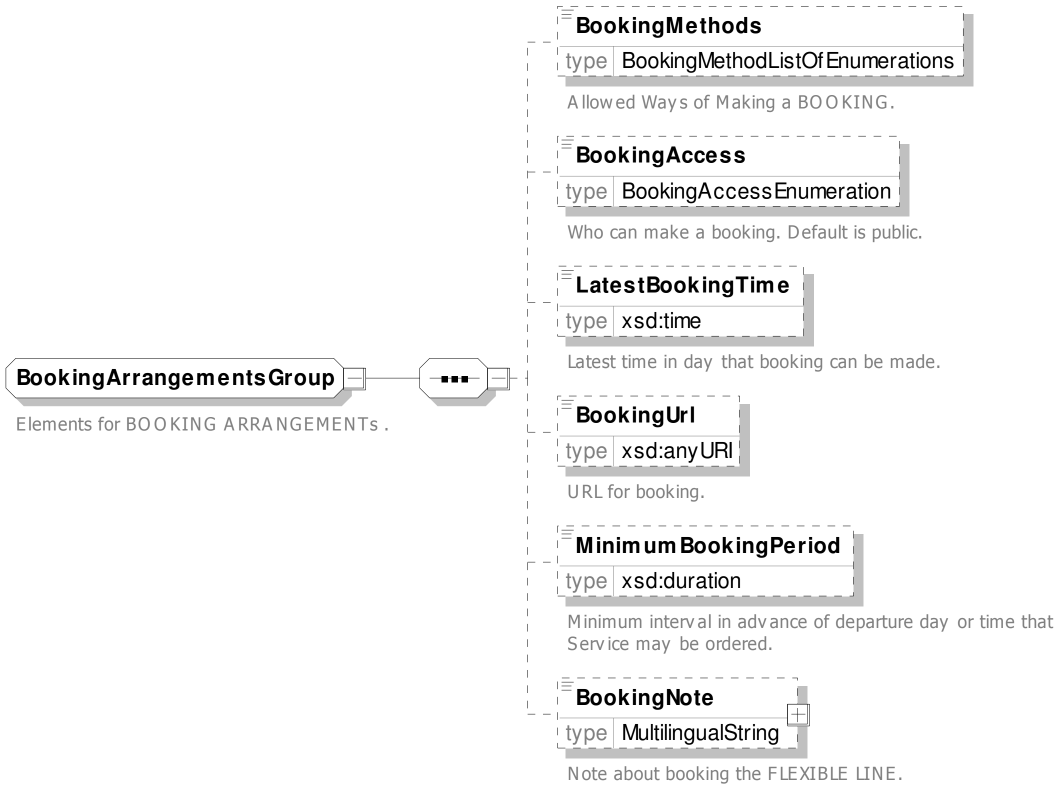


Figure 191 — *BookingArrangements* — XSD

8.6.1.3.9.9.1 BookingMethod– Allowed values

The following table shows the allowed values for ***BookingMethod*** (*BookingMethodEnum*)

Table 157 – *BookingMethod* – Allowed values

Value	Description
callDriver	Call the driver to book a journey.
callOffice	Call an office to book a journey.
online	Book a journey online.
other	Book a journey by other means.
phoneAtStop	Book by using phone at stop.
text	Book by sending a text message.
none	No booking method.

8.6.1.3.9.9.2 BookingAccess – Allowed values

The following table shows the allowed values for ***BookingAccess*** (*BookingAccessEnum*)

Table 158 – *BookingAccess* – Allowed values

Value	Description
public	Public may make a booking
authorisedPublic	Certain members of public may make a booking
staff	Staff may make a booking
other	Other limitation on who may make e a booking

8.6.1.3.10 Usage Parameter:Travel – XML examples

8.6.1.3.10.1 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>
```

8.6.1.3.10.2 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::returnSameWay">
    <IsRestricted>true</IsRestricted>
    <ReturnRouteIdentical>true</ReturnRouteIdentical>
</Routing>

<Routing version="any" id="tap::international">
    <Name>International</Name>
    <CrossBorder>true</CrossBorder>
</Routing>
```

8.6.1.3.10.3 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::oneTrip">
    <Name>Single use </Name>
    <FrequencyOfUseType>single</FrequencyOfUseType>
    <MinimalFrequency>1</MinimalFrequency>
    <MaximalFrequency>1</MaximalFrequency>
```

```xml
</FrequencyOfUse>

<FrequencyOfUse version="any" id="tfl::oneTrip">
    <Name>Unlimited use pass</Name>
    <FrequencyOfUseType>unlimited</FrequencyOfUseType>
</FrequencyOfUse>
```

8.6.1.3.10.4 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:

```xml
<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>
```

8.6.1.3.10.5 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:

```xml
<UsageValidityPeriod version="any" id="tfl::endOfRide">
    <Name>Valid only until end of a single ride complete</Name>
    <UsageEnd>endOfRide</UsageEnd>
</UsageValidityPeriod>

<UsageValidityPeriod version="any" id="tfl::endOfReturn">
    <Name>Valid only until end of a return ride complete</Name>
    <UsageTrigger>startOutboundRide</UsageTrigger>
    <UsageEnd>endOfTrip</UsageEnd>
</UsageValidityPeriod>

<UsageValidityPeriod version="any" id="tfl::endOfFareDay">
    <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:

```xml
<UsageValidityPeriod version="any" id="tfl::TravelCardValidityPeriod@1DOffPeak">
    <Name>Travel Card valid 1 day off peak </Name>
    <UsageTrigger>specifiedStartDate</UsageTrigger>
    <UsageEnd>productExpiry</UsageEnd>
    <StandardDuration>P1D</StandardDuration>
    <BlackoutUse>noTravelWithinPeriod</BlackoutUse>
</UsageValidityPeriod>

<UsageValidityPeriod version="any" id="tfl::TravelCardValidityPeriod@1DAnyTime">
    <Name>Travel Card valid 1 day </Name>
    <UsageTrigger>specifiedStartDate</UsageTrigger>
    <UsageEnd>productExpiry</UsageEnd>
    <StandardDuration>P1D</StandardDuration>
    <BlackoutUse>mayTravelAnytime</BlackoutUse>
</UsageValidityPeriod>
```

```xml
<UsageValidityPeriod version="any" id="tfl::TravelCardValidityPeriod@6MAnyTime">
    <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::TravelCardValidityPeriod@18Plus1Y">
    <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>
```

8.6.1.3.10.6 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::3nightsAway" 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>specifiedNightsAway</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>
```

8.6.1.3.11 Usage Parameter: Product – XML examples

8.6.1.3.11.1 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:

```xml
<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::disabledPerson">
        <Name>Disabled Person Fare</Name>
        <BaseUserProfileRef version="any" ref="tfl::concession"/>
        <TypeOfConcessionRef version="any" ref="tfl::disabledPerson"/>
        <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::child11To15@11to15OysterCardHolder">
                <members>
                    <FareProductPrice version="any" id="tfl::child11To15@TravelCardOnOyster">
                        <DiscountAsPercentage>50.00</DiscountAsPercentage>
                        <PreassignedFareProductRef version="any" ref="tfl::PayAsYouGoFare"/>
                    </FareProductPrice>
                    <FareProductPrice version="any" id="tfl::child11To15@Oyster">
                        <Name> 50 % discount for Oyster travel</Name>
                        <DiscountAsPercentage>50.00</DiscountAsPercentage>
                        <PreassignedFareProductRef version="any" ref="tfl::PayAsYouGoFare"/>
                    </FareProductPrice>
                </members>
                <EntitlementRequiredRef version="any" ref="tfl::OysterPayAsYouGoRightHolder"/>
            </FareTable>
            <FareTable version="any" id="tfl::child11To15@studentRailCard">
                <members>
                    <FareProductPrice version="any" id="tfl::child11To15@studentRailCard">
                        <Name>34 % discount for RailCard travel</Name>
                        <DiscountAsPercentage>34.00</DiscountAsPercentage>
                        <PreassignedFareProductRef version="any" ref="tfl::PrepaidFare"/>
                    </FareProductPrice>
                </members>
                <UserProfileRef version="any" ref="tfl::railCardHolder"/>
            </FareTable>
        </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::seniorResident">
        <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>
```

8.6.1.3.11.2 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:

```
    <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::familyGroup">
        <Name>up to 4 children allowed with adult Oyster holders</Name>
        <MinimumNumberOfCardHolders>1</MinimumNumberOfCardHolders>
        <members>
            <CompanionProfile version="any" id="tfl::familyGroup@adult">
                <UserProfileRef version="any" ref="tfl::adult"/>
                <MinimumNumberOfPersons>1</MinimumNumberOfPersons>
                <MaximumNumberOfPersons>2</MaximumNumberOfPersons>
            </CompanionProfile>
            <CompanionProfile version="any" id="tfl::familyGroup@children">
                <UserProfileRef version="any" ref="tfl::child"/>
                <MinimumNumberOfPersons>1</MinimumNumberOfPersons>
                <MaximumNumberOfPersons>4</MaximumNumberOfPersons>
            </CompanionProfile>
        </members>
        <MaximumPersonsFree>4</MaximumPersonsFree>
    </GroupTicket>
```

8.6.1.3.11.3 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:

```
    <EntitlementGiven version="any" id="tfl::TravelCardSaleDiscount">
        <Name>GIVING ENTITLEMENT use discount from a TravelCard</Name>
        <SaleDiscountRightRef version="any" ref="tfl::TravelCardSaleDiscount"/>
        <EntitlementType>purchase</EntitlementType>
    </EntitlementGiven>

    <EntitlementGiven version="any" id="tfl::OysterPayAsYouGoRight">
        <Name>GIVING ENTITLEMENT right to buy Oyster discounted fares</Name>
        <CappedDiscountRightRef version="any" ref="tfl::OysterPayAsYouGoRight"/>
        <EntitlementType>purchase</EntitlementType>
    </EntitlementGiven>

    <EntitlementGiven version="any" id="tfl::DiscountedPrepaidFare">
        <Name>GIVING ENTITLEMENT a pay as you go fare</Name>
        <PreassignedFareProductRef version="any" ref="tfl::PrepaidFare"/>
        <EntitlementType>purchaseAtDiscount</EntitlementType>
    </EntitlementGiven>

    <EntitlementGiven version="any" id="tfl::AnnualPassGoldCard">
        <Name>GIVING ENTITLEMENT Gold card because has annual pass </Name>
```

```xml
        <SaleDiscountRightRef version="any" ref="tfl::GoldCard"/>
        <EntitlementType>use</EntitlementType>
        <MinimumQualificationPeriod>P1Y</MinimumQualificationPeriod>
    </EntitlementGiven>
```

8.6.1.3.11.4 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:

```xml
    <EntitlementRequired version="any" id="tfl::OysterPayAsYouGoRightHolder">
        <Name>Entitlement from having Right to use Oyster pay as You go</Name>
        <CappedDiscountRightRef version="any" ref="tfl::OysterPayAsYouGoRight"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="tfl::TravelCardHolder">
        <Name>Entitlement from having a TravelCard</Name>
        <PreassignedFareProductRef version="any" ref="tfl::TravelCard"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="tfl::GoldCardHolder">
        <Name>Entitlement from having a GoldCard</Name>
        <SaleDiscountRightRef version="any" ref="tfl::GoldCard"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="nr::railCardHolder">
        <Name>Entitlement from having a RailCard</Name>
        <SaleDiscountRightRef version="any" ref="nr::RailCard"/>
    </EntitlementRequired>

    <EntitlementRequired version="any" id="tfl::FreedomPassHolder">
        <Name>Entitlement from having a Freedom Pass</Name>
        <SaleDiscountRightRef version="any" ref="tfl::FreedomPass"/>
    </EntitlementRequired>
```

8.6.1.3.11.5 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.

For EXAMPLE:

[TO DO ADD EXAMPLE]

8.6.1.3.12 Usage Parameter: Luggage Allowance – XML examples

8.6.1.3.12.1 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:

```xml
    <LuggageAllowance version="any" id="tap::carryon">
        <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>
```

```xml
<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>
```

8.6.1.3.13 Usage Parameter: Booking – XML examples

8.6.1.3.13.1 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:

```xml
<PurchaseWindow version="any" id="tfl::TimeOfTravel">
    <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::SameDay">
    <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::UpTo4DaysInAdvance">
    <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::18PlusUpTo4WeeksInAdvance">
    <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::atLeastTwoMonths">
    <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>
```

8.6.1.3.13.2 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:

```xml
<Transferability version="any" id="tfl::CanTransfer">
    <CanTransfer>true</CanTransfer>
    <HasTransferFee>true</HasTransferFee>
</Transferability>
```

8.6.1.3.13.3 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 PACKAGE) and there is a NOTICE ASSIGNMENT to an associated NOTICE element (not shown) with refund conditions.

For EXAMPLE:

```
<Refunding version="any" id="thx::remboursement2H_50percent">
    <Description lang="fr">PremsOui jusqu_à H (2) du voyage aller</Description>
    <noticeAssignments>
        <NoticeAssignment id="thx::remboursement2H_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::exchange2H_50percent">
            <DiscountAsPercentage>50</DiscountAsPercentage>
        </UsageParameterPrice>
    </prices>
    <Allowed>partial</Allowed>
    <OnlyAtCertainDistributionPoints>true</OnlyAtCertainDistributionPoints>
    <ResellWhen>beforeStartOfValidity</ResellWhen>
    <ExchangeableUntilDuration>PT2H</ExchangeableUntilDuration>
</Refunding>
```

8.6.1.3.13.4 Usage Parameter: Booking: XML Example of Reserving parameter

The following code fragment shows various RESERVING usage parameters.

For EXAMPLE:

```
<Reserving version="any" id="thx::mustReserve">
    <Name lang="fr">La réservation rest e obligatoire</Name>
    <ReservingRequirements>reservationsCompulsory</ReservingRequirements>
</Reserving>

<Reserving version="any" id="tap::groupRequired">
    <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::mustReserveWholeCompartment">
    <ReservingRequirements>reservationsCompulsory</ReservingRequirements>
    <MinimumNumberToReserve>4</MinimumNumberToReserve>
    <MaximumNumberToReserve>4</MaximumNumberToReserve>
    <MustReserveWholeCompartment>true</MustReserveWholeCompartment>
</Reserving>
```

8.6.2 Fare Product

8.6.2.1 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 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 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.

[TO DO RENAME CHARGING METHOD TO CHARGING MOMENT]

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 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 an 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.

— an 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.

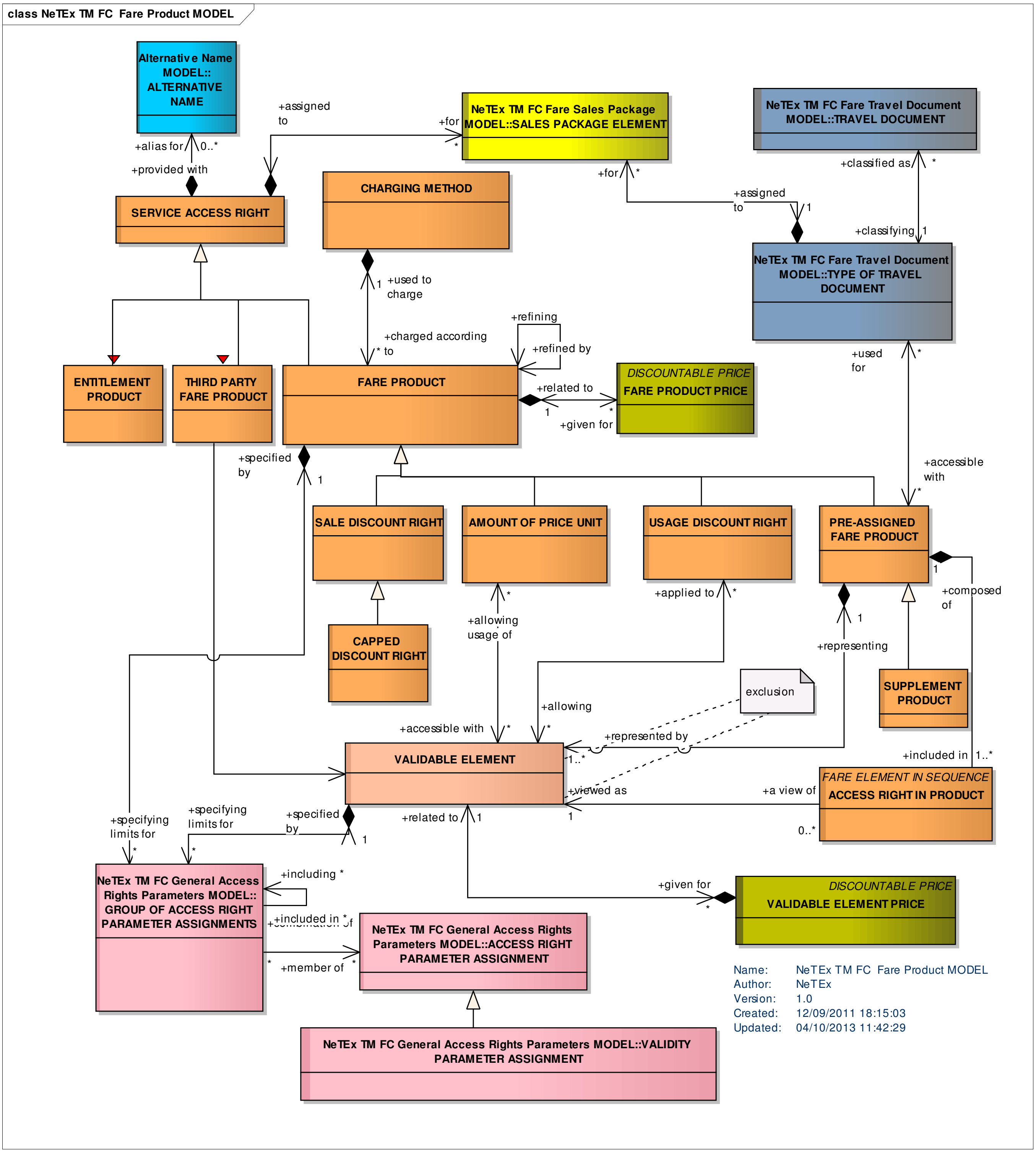


Figure 192 — Fare Product – Conceptual Model

8.6.2.2 Fare Product – Conceptual model examples

[MORE TO DO]

8.6.2.2.1 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

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 193 — Fare Table – Example Rail Products and Fare Conditions

8.6.2.3 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 represent products that are not for actual travel but which are required in order to purchase or use a FARE PRODUCT.

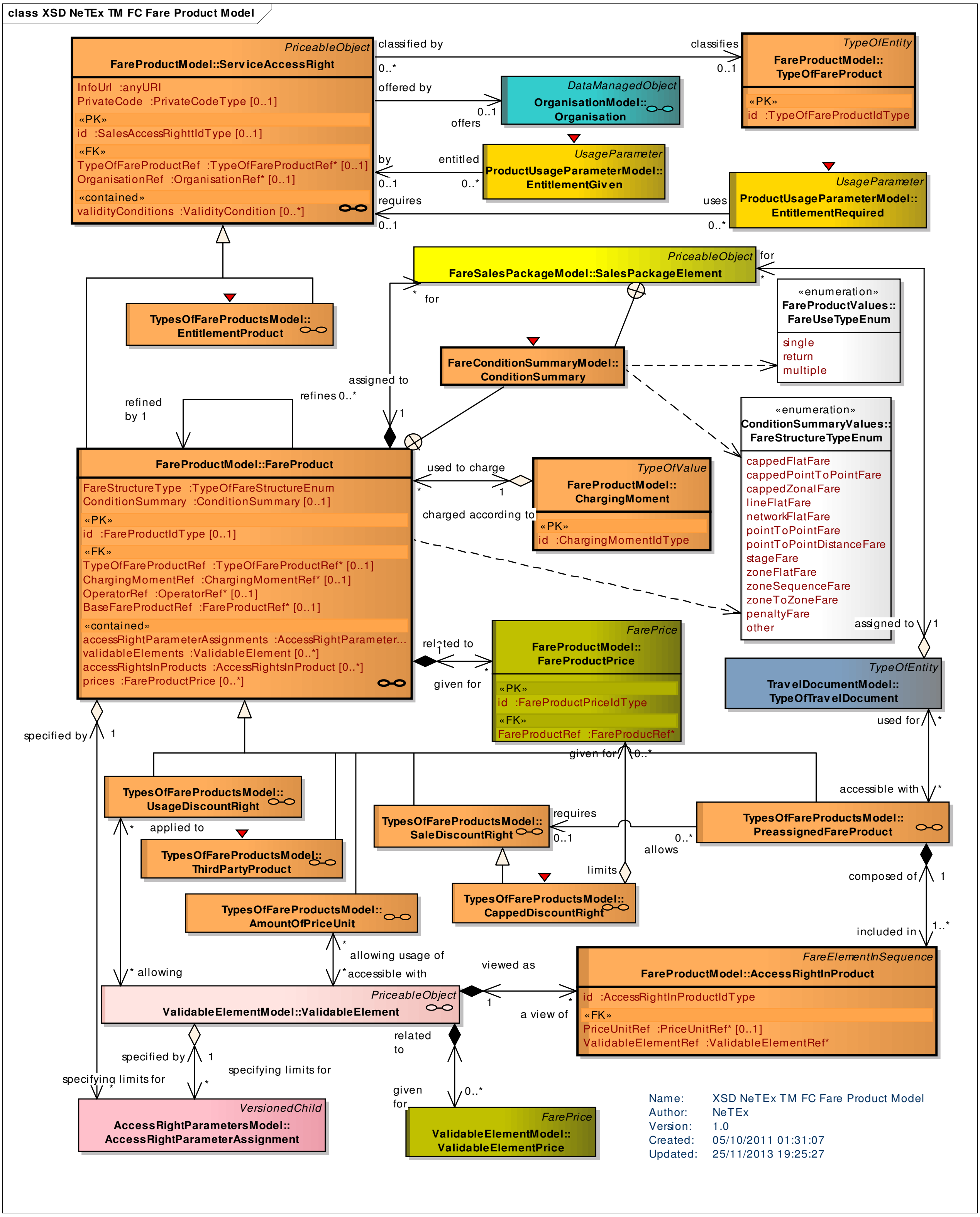


Figure 194 — Fare Product – Physical Model

8.6.2.3.1 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 PRDUCT – 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 an 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.
- an THIRD PARTY PRODUCT: A product from another organisation not fully described by the systems.

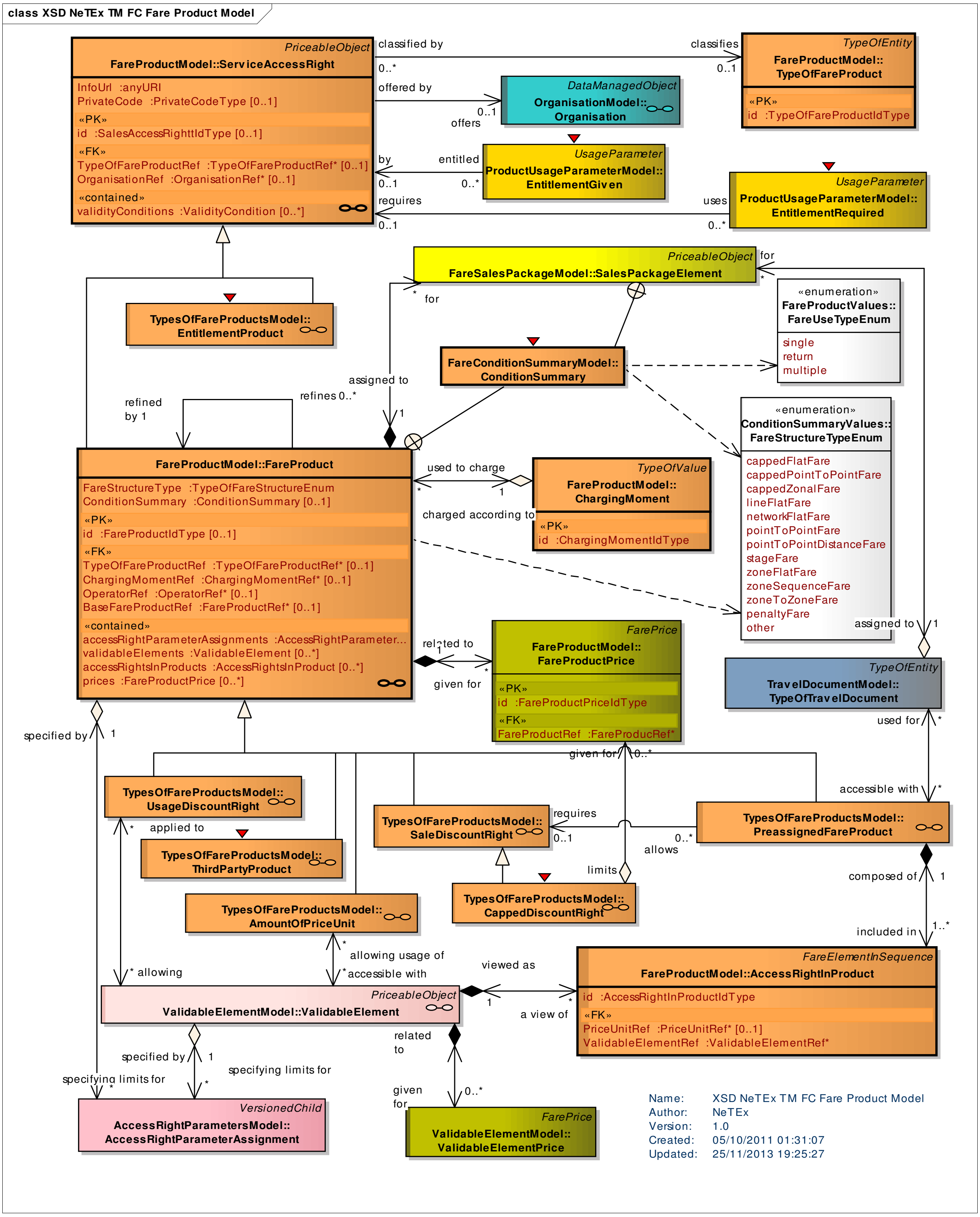


Figure 195 — Types of Fare Product – Physical Model

8.6.2.3.2 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.

There may be different caps for different VALIDABLE ELEMENTs, e.g. metro Trip, river trip etc.

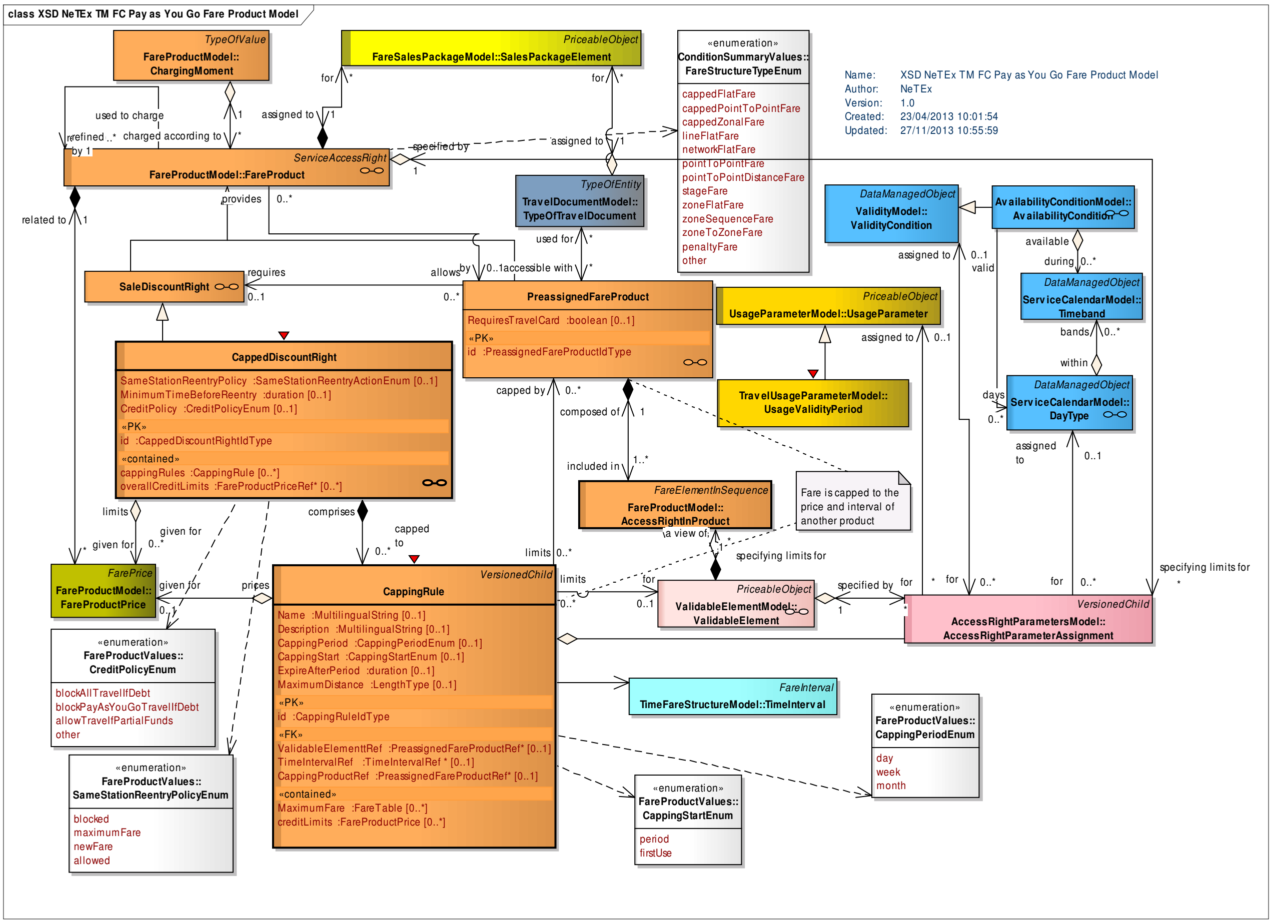


Figure 196 — Pay as You Go Fare Product – Physical Model

8.6.2.3.3 Fare Condition Summary – Physical model

The following figure shows the physical model for the FARE CONDITION SUMMARY. This can be used to provide a high level description of a product for product comparison 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.

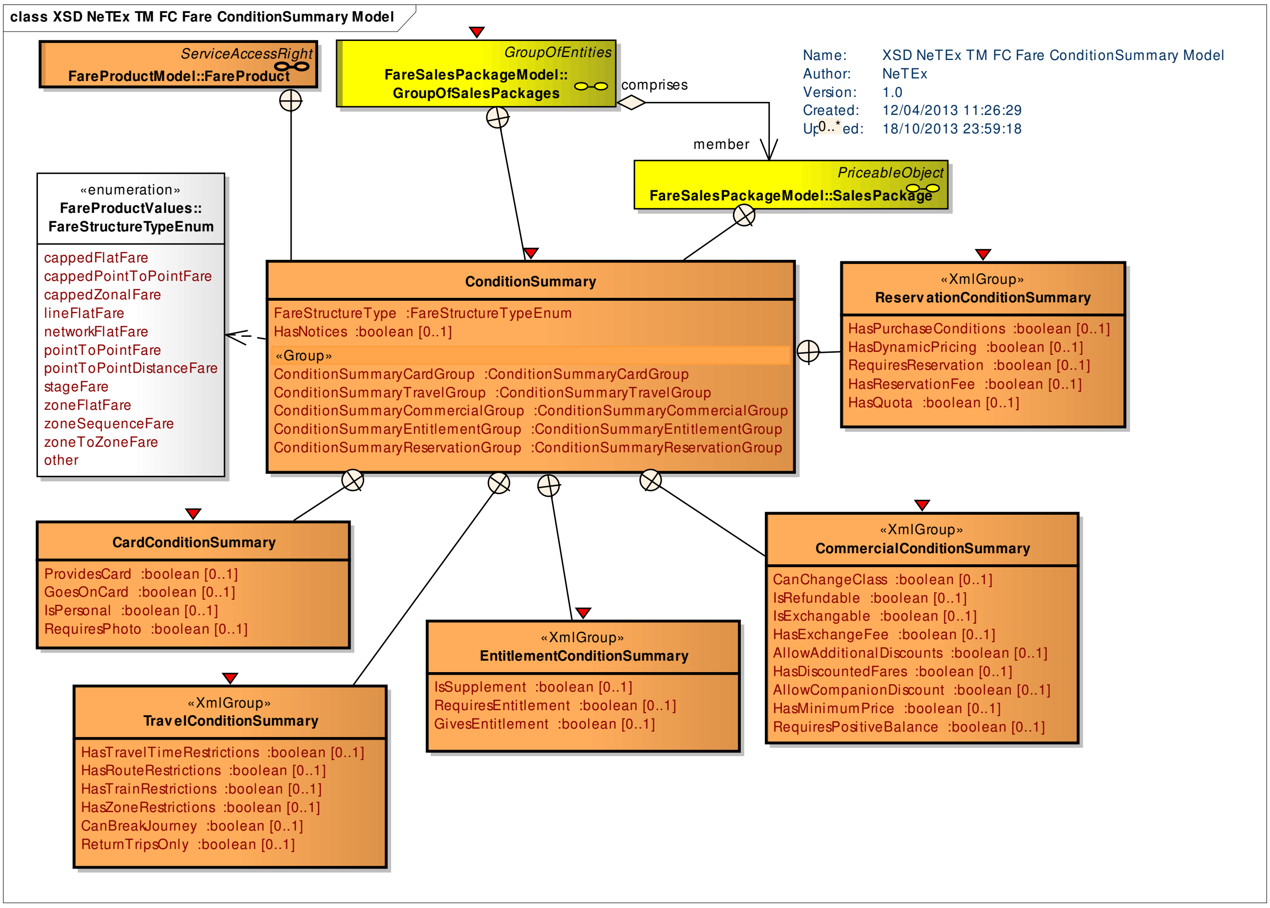


Figure 197 — Fare Condition Summary– Physical Model

8.6.2.4 Fare Product – Attributes and XSD

8.6.2.4.1 ServiceAccessRight – Model Element

An immaterial marketable element (access rights, discount rights etc).

Table 159 – *ServiceAccessRight* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PriceableObject*	::>	SERVICE ACCESS RIGHT inherits from PRICEABLE OBJECT.
«PK»	***id***	*ServiceAccessRightIdType*	0:1	Identifier of SERVICE ACCESS RIGHT.
"cntd»	***validityConditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs specifying specific availability of FARE PRODUCT for use.
	PrivateCode	*PrivateCodeType*	0:1	Alternative identifier of an entity; can be used to associate with legacy systems.
	InfoUrl	*xsd:anyURI*	1:1	Link for product information.

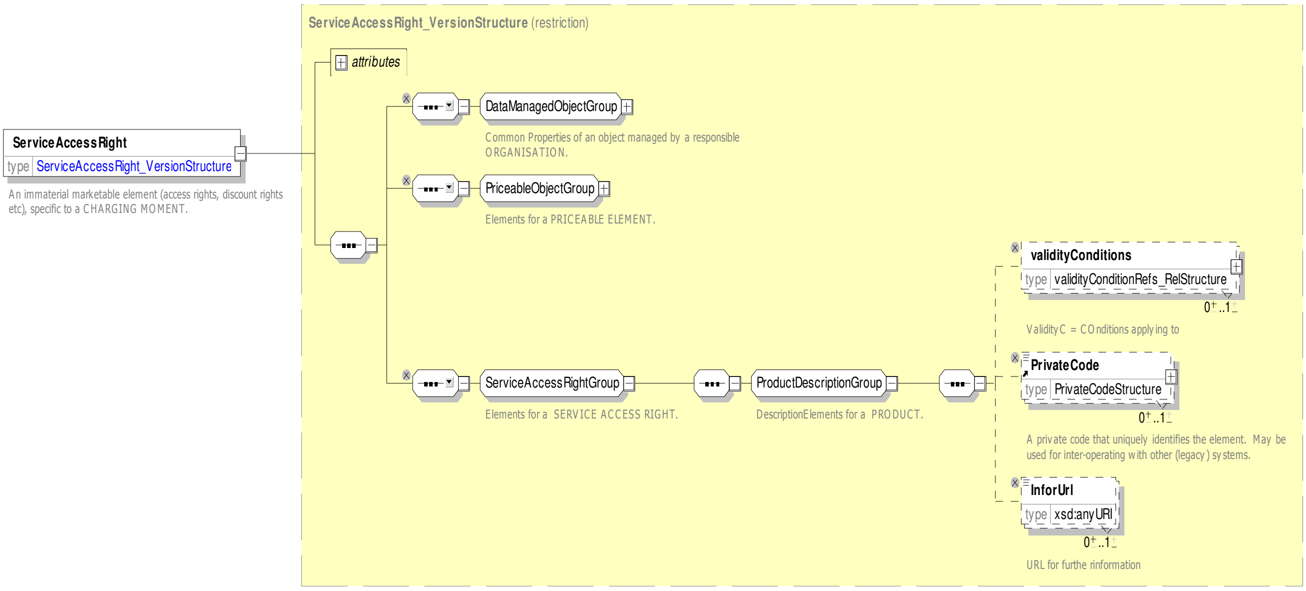


Figure 198 — *ServiceAccessRight* — XSD

8.6.2.4.2 FareProduct – Model Element

An immaterial marketable element (access rights, discount rights etc), specific to a CHARGING MOMENT.

Table 160 – *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***	*FareProductRef*	0:1	Reference to a CHARGING MOMENT for product.
«FK»	***TypeOfFare-ProductRef***	*TypeOfFareProductRef*	0:1	Type of FARE PRODUCT.
	ConditionSummary	*ConditionSummary*	0:1	Summary description of conditions on FARE PRODUCT.
«FK»	***OperatorRef***	*OperatorRef*	0:1	OPERATOR to which ACCESS RIGHT PARAMETER is assigned.
GROUP	***FareProduct-RelationGroup***	*FareProduct-RelationGroup*	0:1	Elements relating to association of FARE PRODUCT.
GROUP	***FareValidityGroup***	*FareValidityGroup*	0:1	Elements relating to validity of FARE PRODUCT.
GROUP	***FareProductPricing Group***	*FareProduct-PricingGroup*	0:1	Elements relating to pricing of FARE PRODUCT.

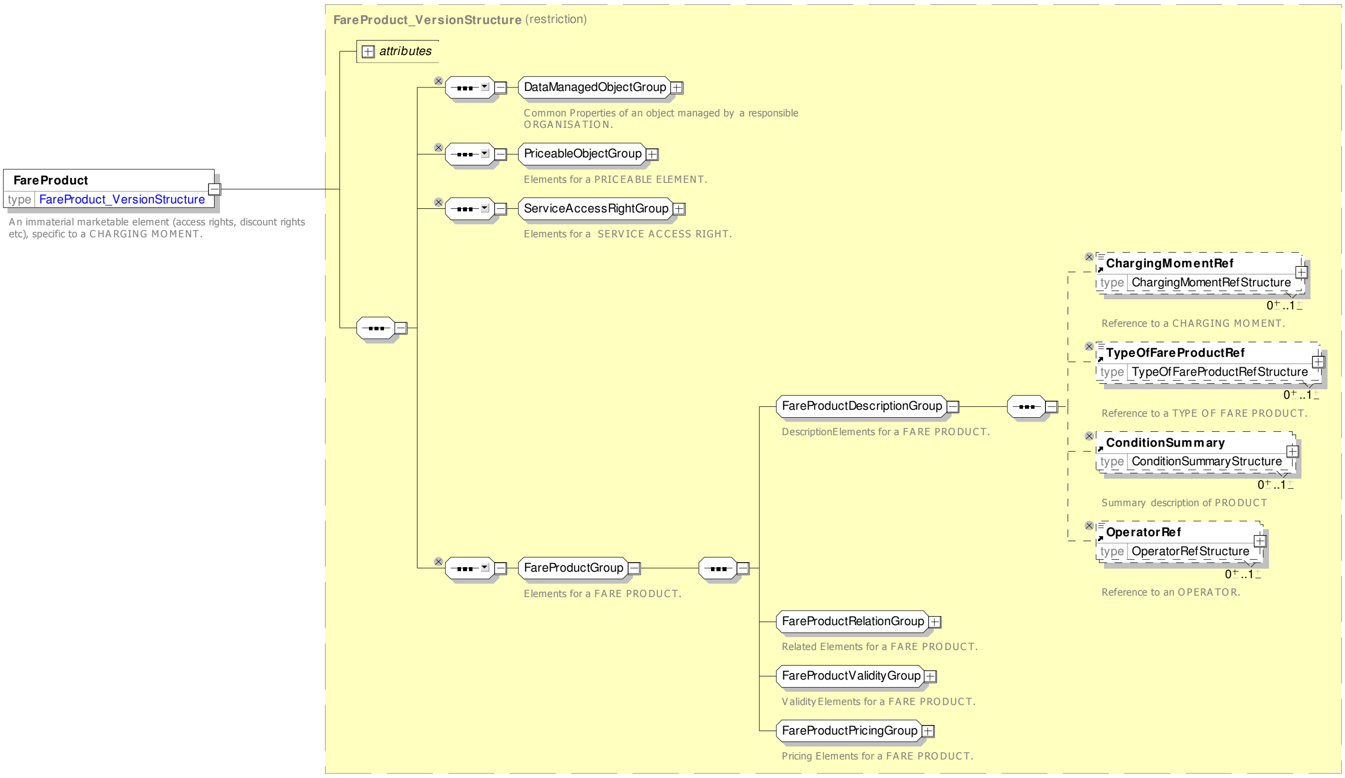


Figure 199 — *FareProduct* — XSD

8.6.2.4.2.1 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 161 – *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.

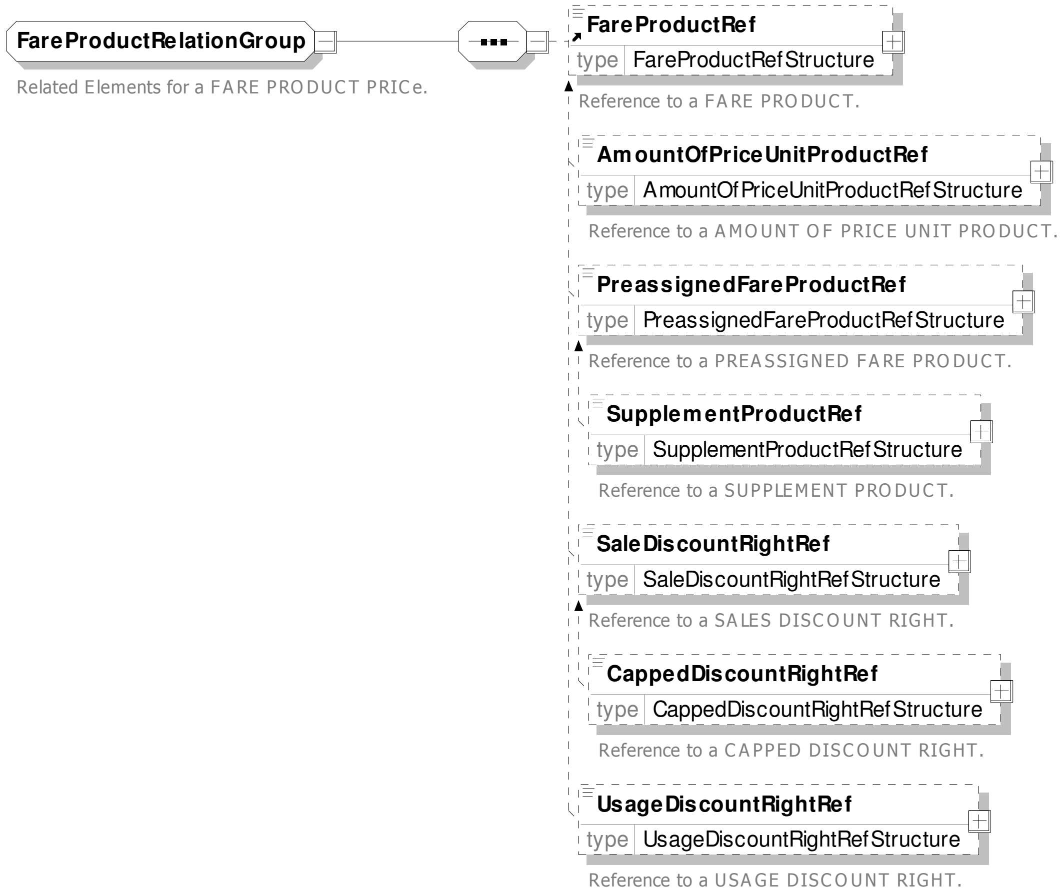


Figure 200 — *FareProductRelationGroup* — XSD

8.6.2.4.2.2 *FareProductValidityGroup* – Group

The ***FareProductValidityGroup*** specifies attributes describing the validity of a FARE PRODUCT.

Table 162 – *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.

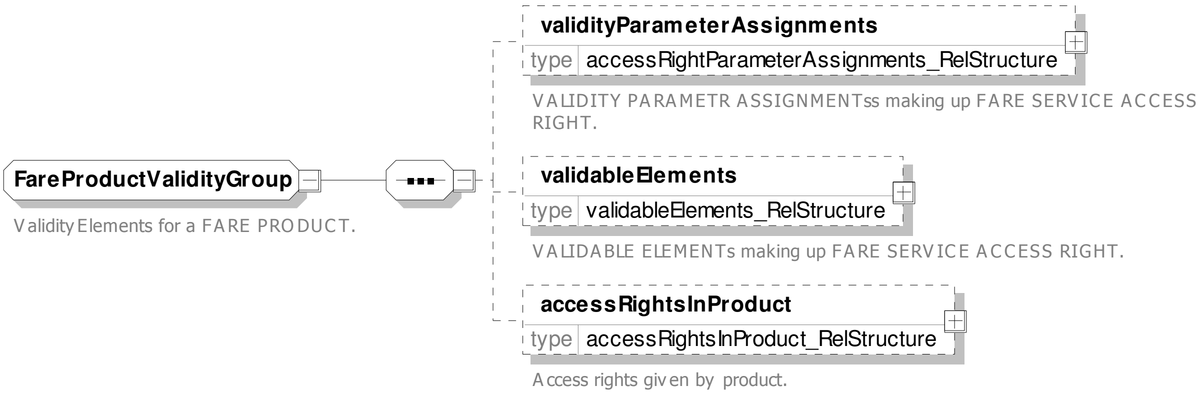


Figure 201 — *FareProductValidityGroup* — XSD

8.6.2.4.2.3 FareProductPricingGroup – Group

The ***FareProductPricingGroup*** specifies pricing properties of a FARE PRODUCT.

Table 163 – *FareProductPricingGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
“cntd»	***priceGroups***	*PriceGroup*	0:*	FARE PRODUCT PRICEs in PRICE GROUP.
“cntd»	***prices***	*FareProductPrice*	0:*	FARE PRODUCT PRICEs in PRICE GROUP.

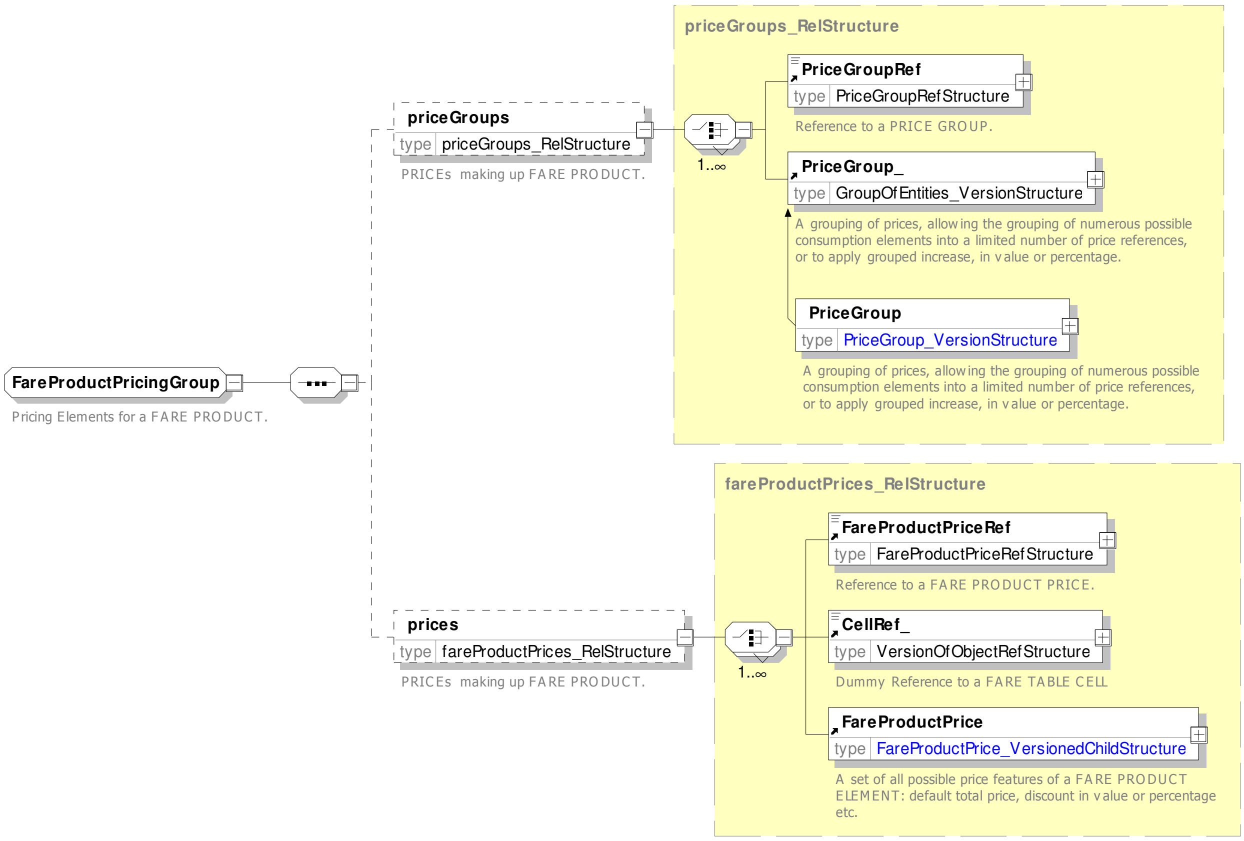


Figure 202 — *FareProductPricingGroup* — XSD

8.6.2.4.3 FareProductPrice – Model Element

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

Table 164 – *FareProductPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	FARE PRODUCT inherits from FARE PRICE
«PK»	***id***	*FareProductPriceIdType*	1:1	Identifier of FARE PRODUCT PRICE.
«FK»	***FareProductRef***	*FareProducRef*	1:1	FARE PRODUCT for which this is the price.

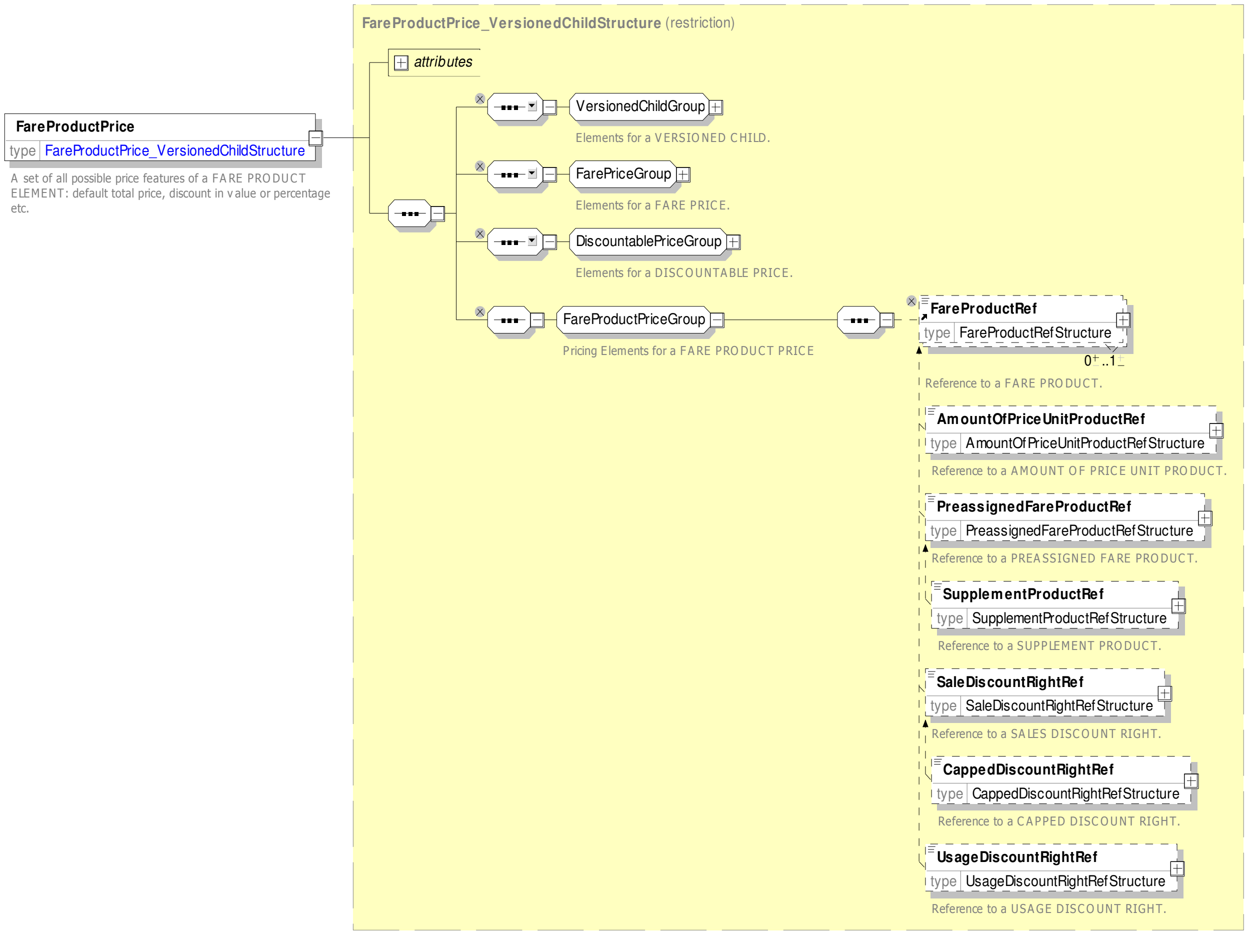


Figure 203 — *FareProductPrice* — XSD

8.6.2.4.4 AccessRightInProduct – Model Element

tA 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 165 – *AccessRightInProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareElementInSequence*	::>	ACCESS RIGHT IN PRODUCT inherits from FARE ELEMENT IN SEQUENCE.
	id	*AccessRightInProduct-IdType*	1:1	Identifier of ACCESS RIGHT IN PRODUCT.
«FK»	***Validable-ElementRef***	*ValidableElementRef*	1:1	Identifier of ACCESS RIGHT IN PRODUCT.
«FK»	***Preassigned-FareProducRef***	*PreassignedFareProduc Ref*	0:1	Identifier of PRE ASSIGNED FARE PRODUCT.
“cntd»	***accessRight-Assignments***	*AccessRightParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs relating to ACCESS RIGHT IN PRODUCT.

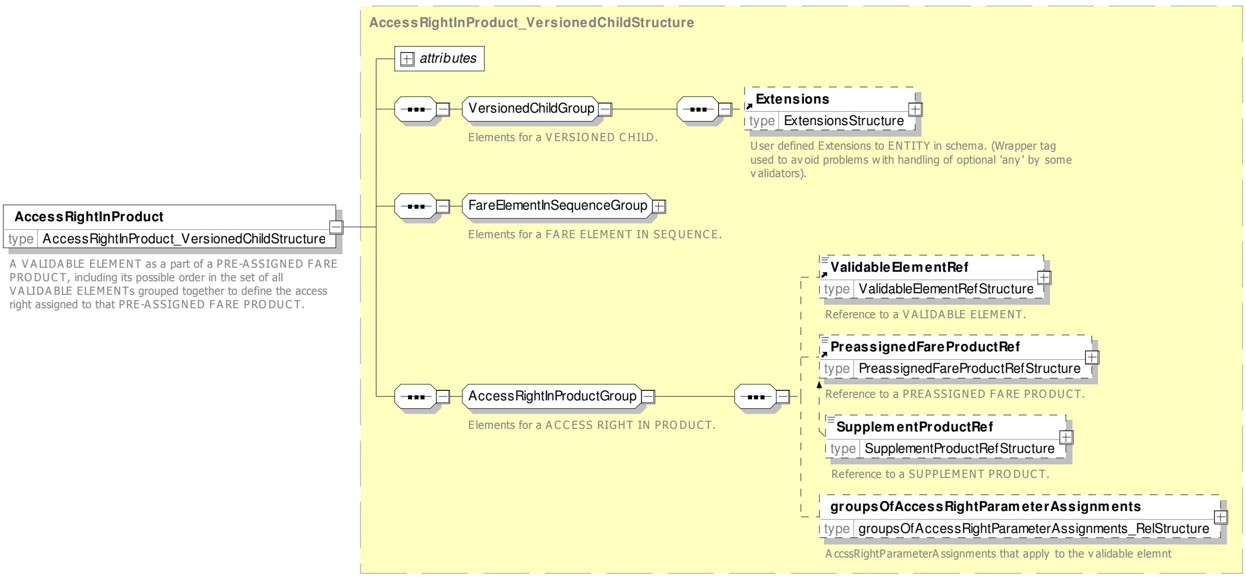


Figure 204 — *AccessRightInProduct* — XSD

8.6.2.4.5 ConditionSummary – Model Element

A summary of the properties of a FARE PRODUCT or PACKAGE that can be used to generate passenger information.

Table 166 – *ConditionSummary* – Element

Classifi-cation	Name	Type	Cardinality	Description
	FareStructure-Type	*FareStructureTypeEnum*	1:1	Classification of fare type.
	HasNotices	*xsd:boolean*	0:1	Whether there are notices associated with the product.
GROUP	***Condition-SummaryCard-Group***	*ConditionSummaryCard-Group*	1:1	Elements relating to cards on CONDITION SUMMARY.
GROUP	***Condition-Summary-EntitlementGroup***	*ConditionSummary-EntitlementGroup*	1:1	Elements relating to entitlement conditions on CONDITION SUMMARY.
GROUP	***Condition-Summary-TravelGroup***	*ConditionSummary-TravelGroup*	1:1	Elements relating to travel conditions on CONDITION SUMMARY.
GROUP	***Condition-Summary-Commercial-Group***	*ConditionSummary-CommercialGroup*	1:1	Elements relating to commercial conditions on CONDITION SUMMARY.
GROUP	***Condition-Summary-Reservation-Group***	*ConditionSummary-ReservationGroup*	1:1	Elements relating to reservation conditions on CONDITION SUMMARY.

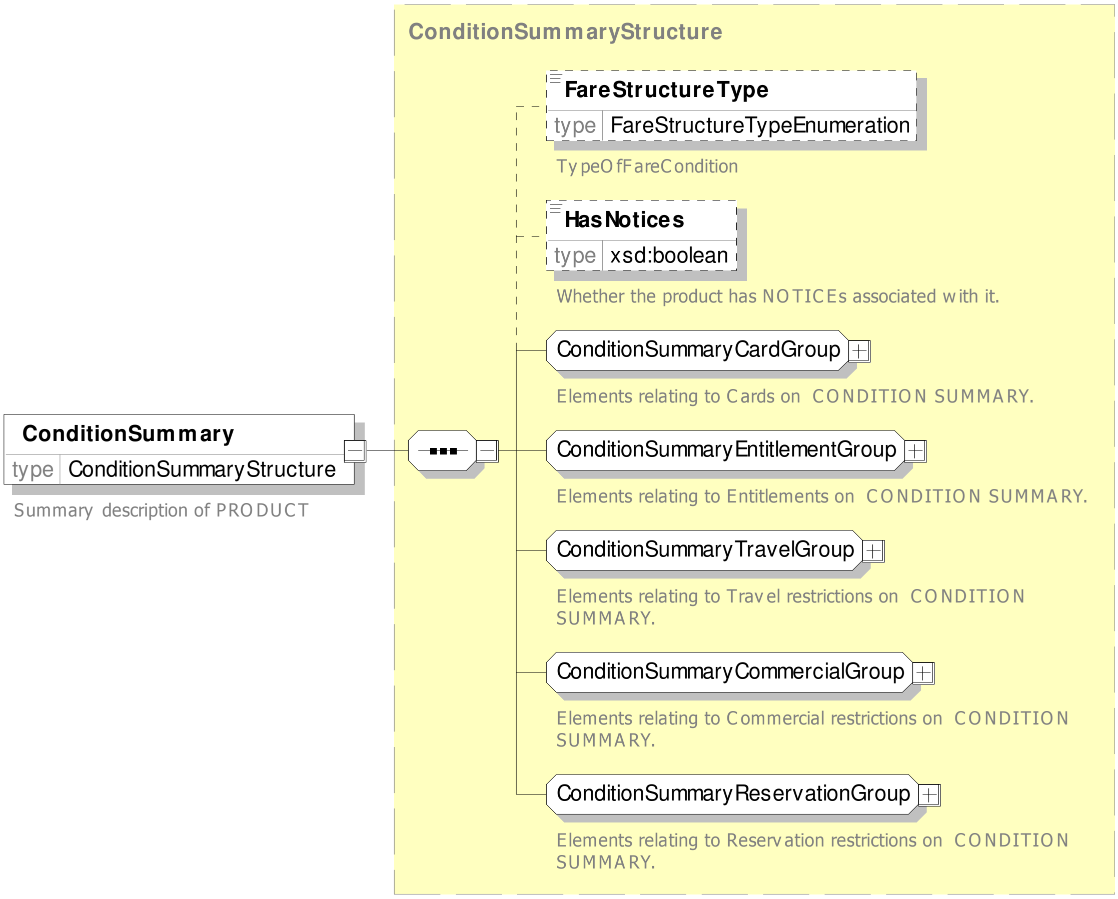


Figure 205 — *ConditionSummary* — XSD

8.6.2.4.5.1 FareStructureType – Allowed values

The following table shows the allowed values for Fare Structure Type.

Table 167 – FareStructureTypeEnum – Allowed values

Value	Description
networkFlatFare	Flat fare applying to whole network.
lineFlatFare	Flat fare applying to group of lines or line.
zoneToZoneFare	Zone to zone fare.
zoneSequenceFare	Zones in sequence fare.
pointToPointFare	Point to point fare.
stageFare	Fare stage fare.
zoneFlatFare	Zonal flat fare.
cappedZonalFare	Capped zonal fare.

cappedFlatFare	Capped flat fare.
other	Other fare.

8.6.2.4.5.2 ConditionSummaryCardGroup – Group

The ***ConditionSummaryCommercialGroup*** summarises the conditions on travel cards associated with a FARE PRODUCT and /or SALES PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs.

Table 168 – *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.

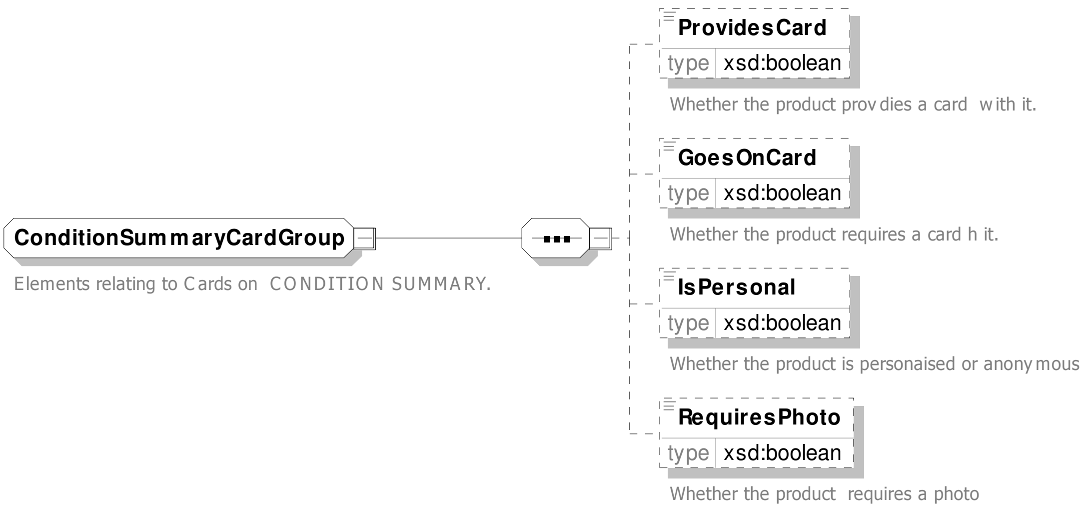


Figure 206 — *ConditionSummaryCardGroup* — XSD

8.6.2.4.5.3 ConditionSummaryEntitlementGroup – Group

The ***ConditionSummaryCommercialGroup*** summarises the entitlement conditions associated with a FARE PRODUCT and /or SALES PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs.

Table 169 – *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.

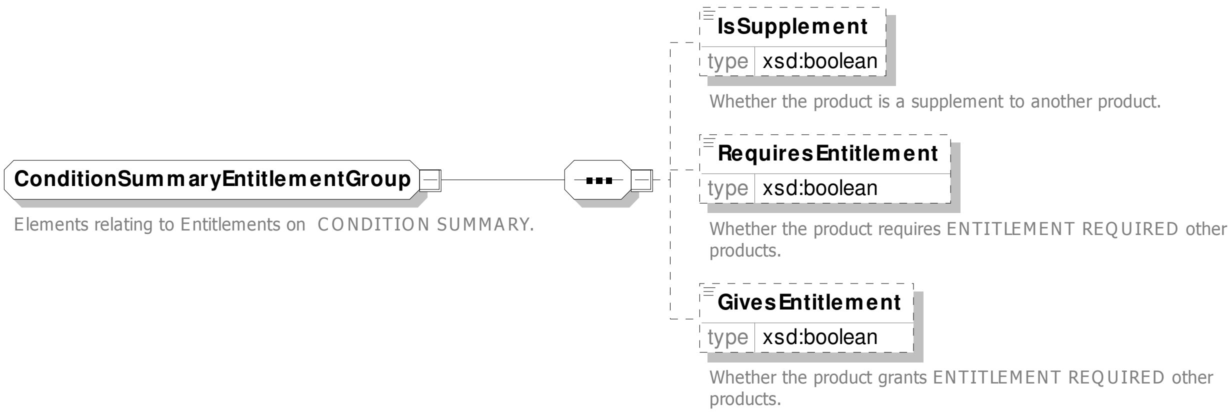


Figure 207 — *ConditionSummaryEntitlementGroup* — XSD

8.6.2.4.5.4 ConditionSummaryTravelGroup – Group

The ***ConditionSummaryTravelGroup*** summarises whether there are restrictions on travel associated with a FARE PRODUCT and /or SALES PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs and ACCESS RIGHT ASSIGNMENTs.

Table 170 – *ConditionSummaryTravelGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Operator-Restrictions	*Operator RestrictionEnum*	0:1	Limitations as to which OPERATOR's services may be used.
	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.
	TrainRestrictions	*TrainRestrictionEnum*	0:1	Limitations as to which trains may be used.
	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.

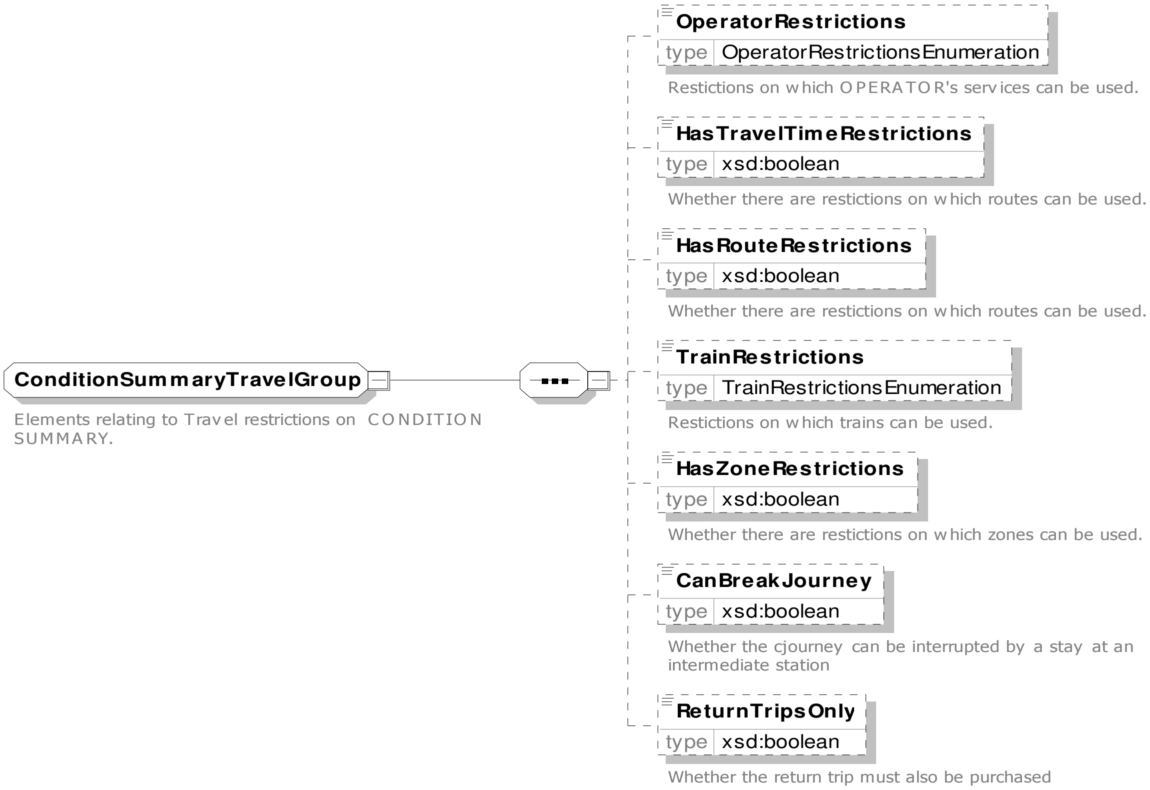


Figure 208 — *ConditionSummaryTravelGroup* — XSD

8.6.2.4.5.4.1 OperatorRestrictions – Allowed values

The following table shows the allowed values for ***OperatorRestrictions*** (*OperatorRestrictionsEnum*).

Table 171 – *OperatorRestrictions* – Allowed values

Value	Description
anyTrain	Can travel on any OPERATORs' services.
restricted	Restricted to certain OPERATORs' services.
specifiedOperatorOnly	Can travel only on a specific OPERATOR’s services.

8.6.2.4.5.4.2 TrainRestrictions – Allowed values

The following table shows the allowed values for ***TrainRestrictions*** (*TrainRestrictionsEnum*).

Table 172 – *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.

8.6.2.4.5.5 ConditionSummaryCommercialGroup – Group

The ***ConditionSummaryCommercialGroup*** summarises the commercial conditions associated with a FARE PRODUCT and /or SALES PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs, FARE PRICEs and ACCESS RIGHT ASSIGNMENTs.

Table 173 – *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 209 — *ConditionSummaryCommercialGroup* — XSD

8.6.2.4.5.6 ConditionSummaryReservationGroup – Group

The ***ConditionSummaryReservationGroup*** summarises the restrictions on booking associated with a FARE PRODUCT and /or SALES PACKAGE. Detailed restrictions are specified by USAGE PARAMETERs, FARE PRICEs and ACCESS RIGHT ASSIGNMENTs.

Table 174 – *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-	*xsd:boolean*	0:1	Whether product has dynamic pricing.

	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.

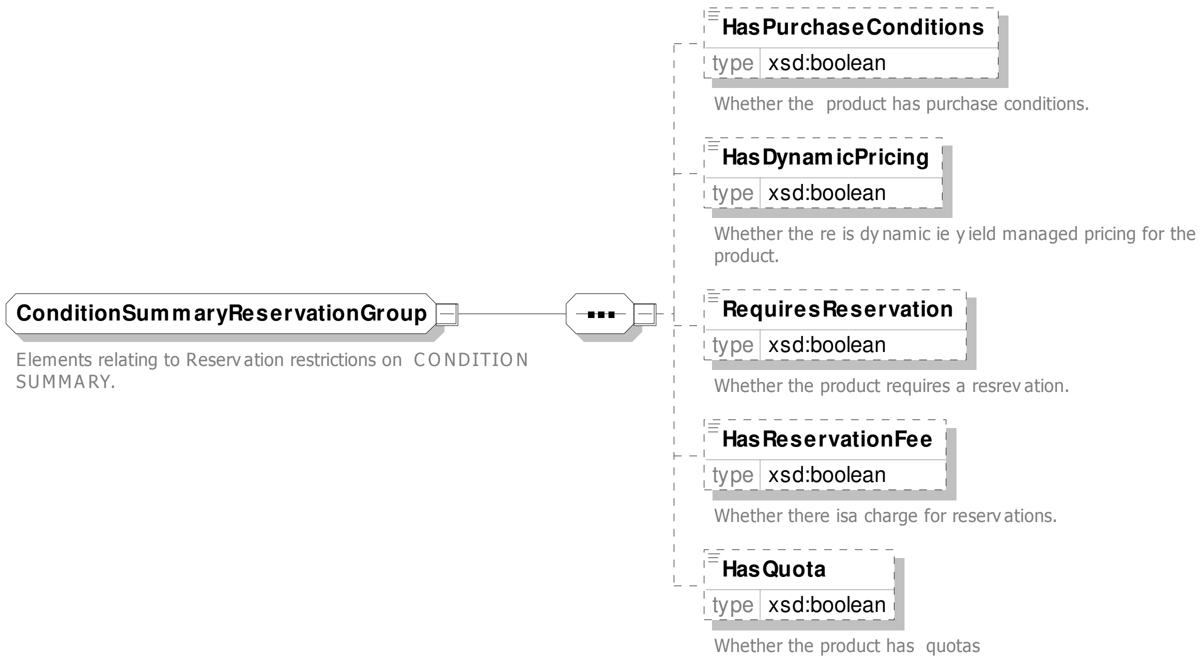


Figure 210 — *ConditionSummaryReservationGroup* — XSD

8.6.2.4.5.6.1 FareUseType – Allowed values

The following table shows the allowed values for ***FareUseType*** (FareUseTypeEnum).

Table 175 – *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.

8.6.2.4.5.6.2 TariffBasis – Allowed values

Allowed values for ***TariffBasis*** *(TariffBasisEnum*).

Table 176 – 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.

8.6.2.4.6 TypeOfFareProduct – Model Element

A classification of FARE PRODUCTs.

Table 177 – *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.

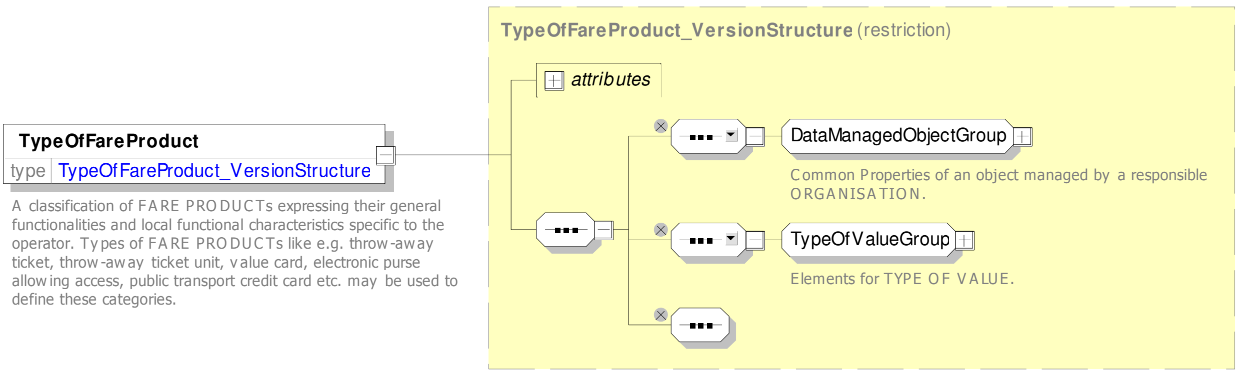


Figure 211 — *TypeOfFareProduct* — XSD

8.6.2.4.7 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 178 – *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.

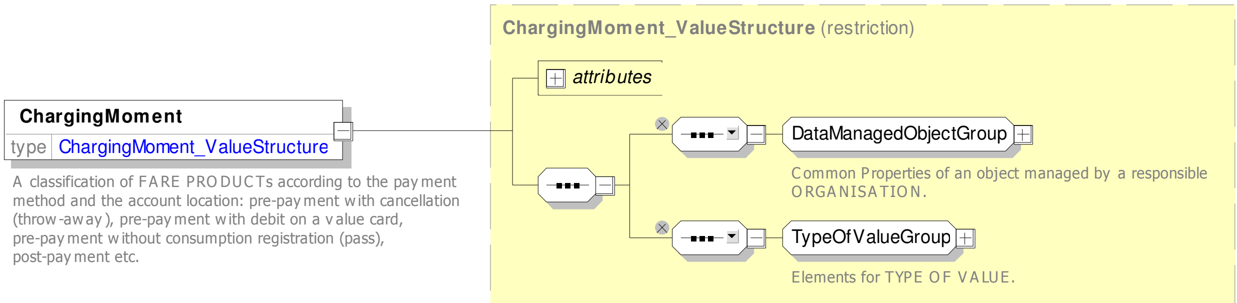


Figure 212 — *ChargingMoment* — XSD

8.6.2.5 Types of Fare Product – Attributes and XSD

8.6.2.5.1 PreassignedFareProduct – Model Element

A FARE PRODUCT consisting of one or several VALIDABLE ELEMENTs, specific to a CHARGING METHOD.

Table 179 – *PreassignedFareProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareProduct*	::>	PREASSIGNED FARE PRODUCT inherits from FARE PRODUCT.
«PK»	***id***	*Preassigned-FareProductIdType*	1:1	Identifier of PREASSIGNED FARE PRODUCT.

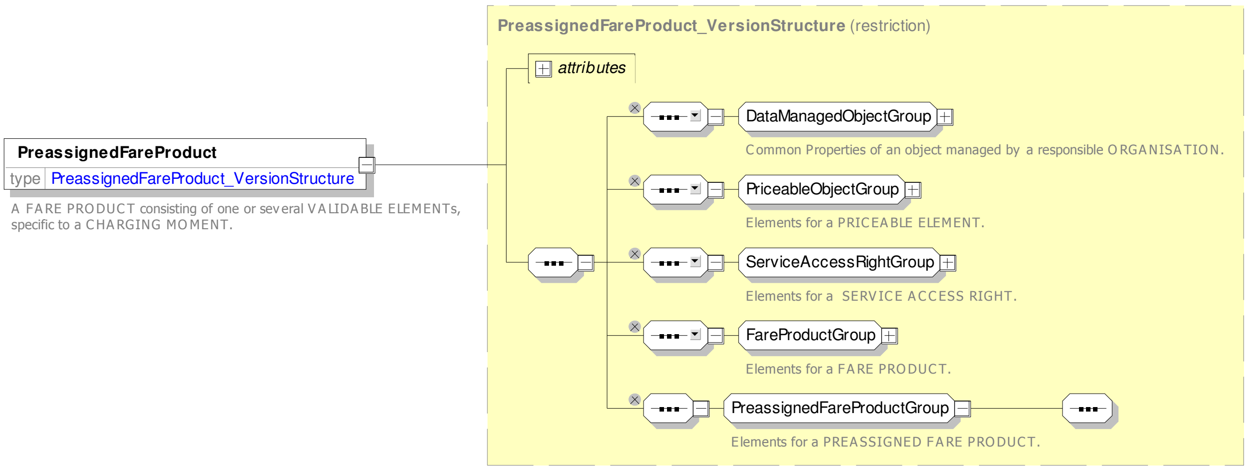


Figure 213 — *PreassignedFareProduct* — XSD

8.6.2.5.2 AmountOfPriceUnit – Model Element

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.

Table 180 – *AmountOfPriceUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareProduct*	::>	AMOUNT OF PRICE UNIT inherits from FARE PRODUCT.
«PK»	***id***	*AmountOfPriceUnit-IdType*	1:1	Identifier of AMOUNT OF PRICE UNIT.
	PriceUnitRef	*PriceUnitRef*	1:1	Reference to a PRICE UNIT.

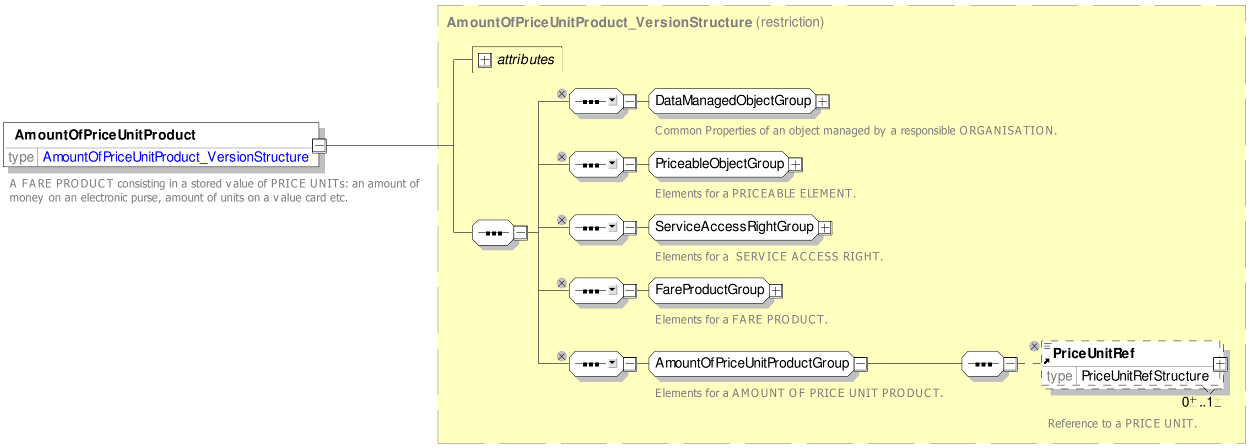


Figure 214 — *AmountOfPriceUnit* — XSD

8.6.2.5.3 UsageDiscountRight – Model Element

A FARE PRODUCT allowing a customer to benefit from discounts when consuming VALIDABLE ELEMENTs.

Table 181 – *UsageDiscountRight* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareProduct*	::>	USAGE DISCOUNT RIGHT inherits from FARE PRODUCT.
«PK»	***id***	*UsageDiscountIdType*	1:1	Identifier of USAGE DISCOUNT RIGHT.

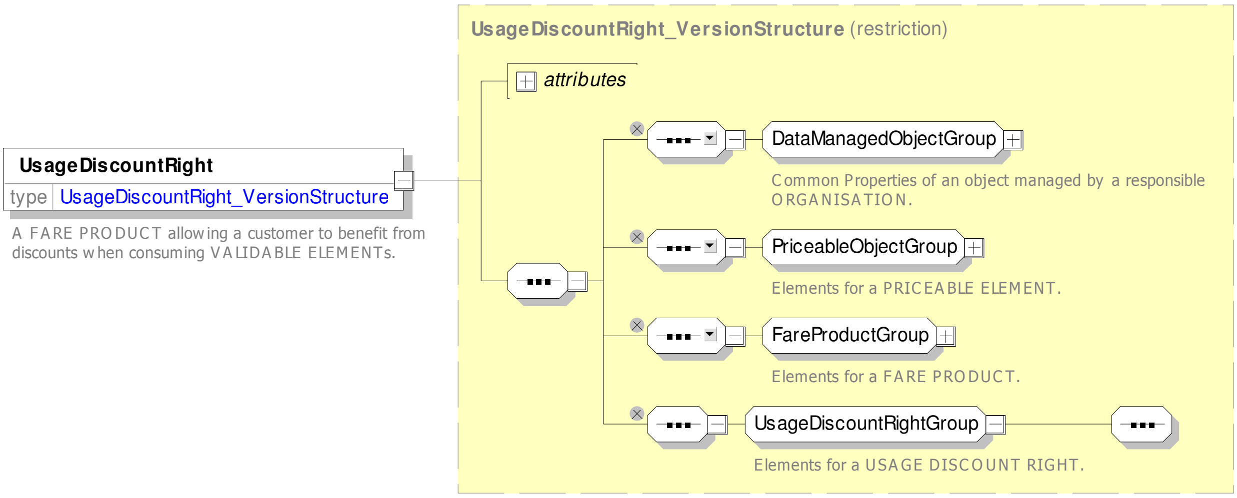


Figure 215 — *UsageDiscountRight* — XSD

8.6.2.5.4 SaleDiscountRight – Model Element

A FARE PRODUCT allowing a customer to benefit from discounts when purchasing SALES PACKAGEs.

Table 182 – *SaleDiscountRight* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareProduct*	::>	SALE DISCOUNT RIGHT inherits from FARE PRODUCT.
«PK»	***id***	*SaleDiscountRightIdType*	1:1	Identifier of SALE DISCOUNT RIGHT.

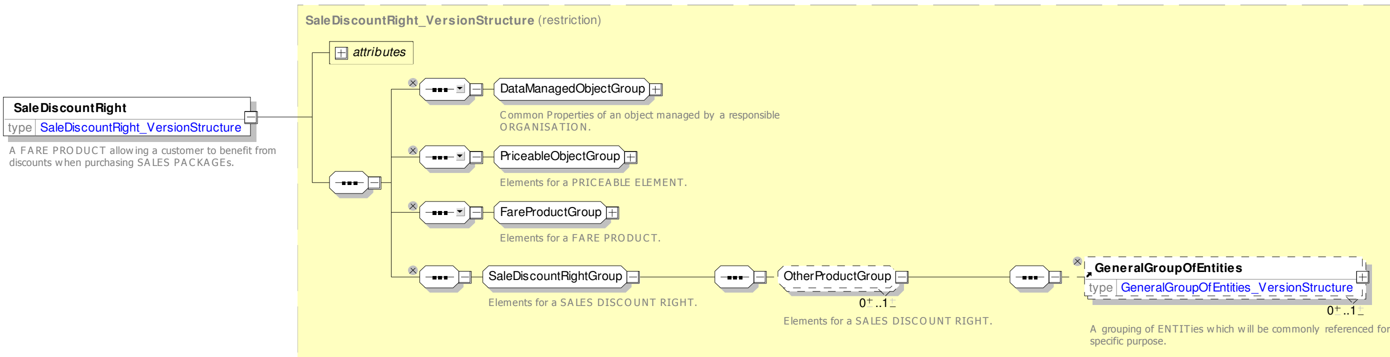


Figure 216 — *SaleDiscountRight* — XSD

8.6.2.5.5 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 183 – *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.

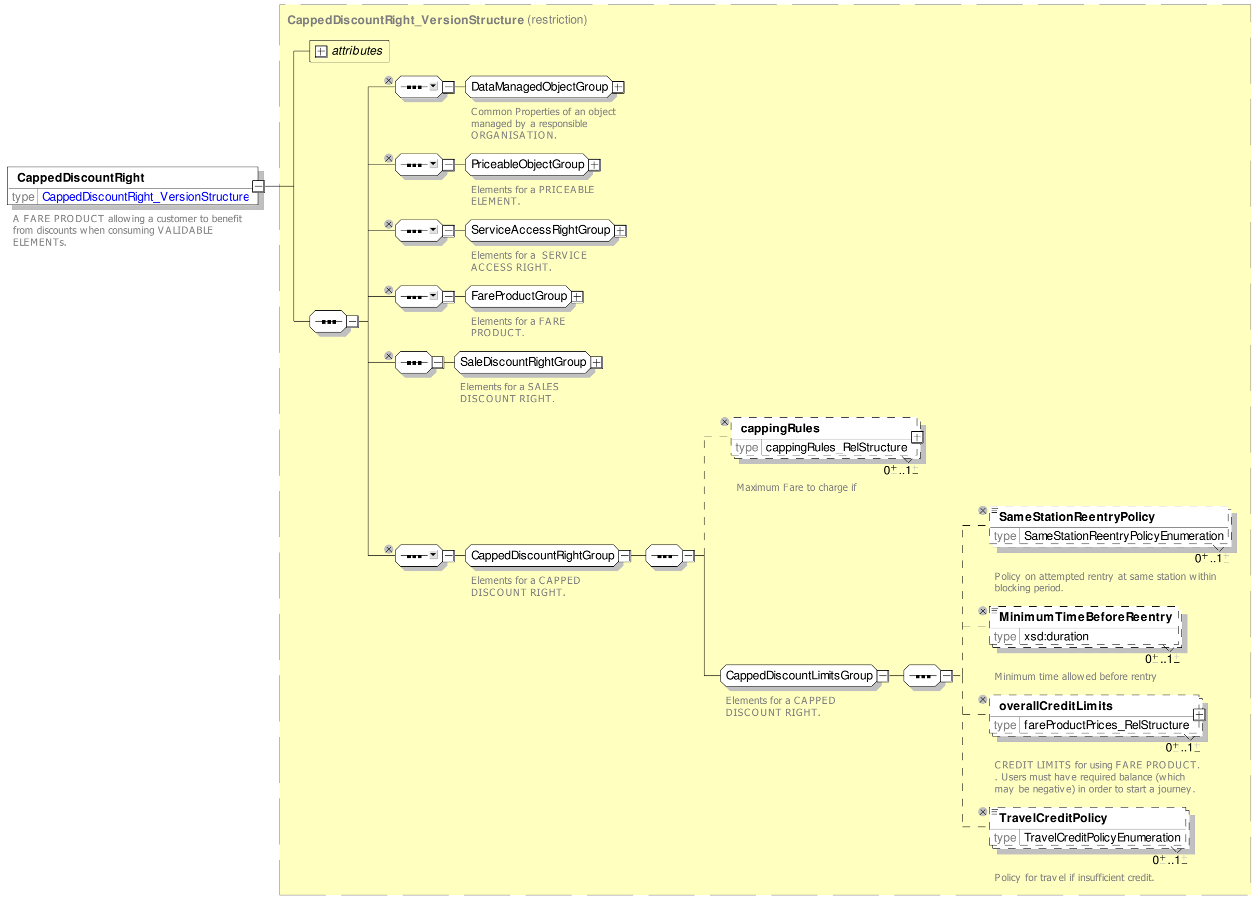


Figure 217 — *CappedDiscountRight* — XSD

8.6.2.5.5.1 CappedDiscountLimitsGroup – Group

The ***CappedDiscountLimitsGroup*** defines time and credit limit properties of the CAPPED DISCOUNT RIGHT.

Table 184 – *CappedDiscountLimitsGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	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 access at the same station not considered as re-entry.
	TravelCredit-Policy	*TravelCreditPolicy*	0:1	A set of parameters set a price cap on a product.
“cntd»	***overallCredit-Limits***	*PricingRule*	0:*	LIMITING RULE setting credit limits that apply to overall use of this product.

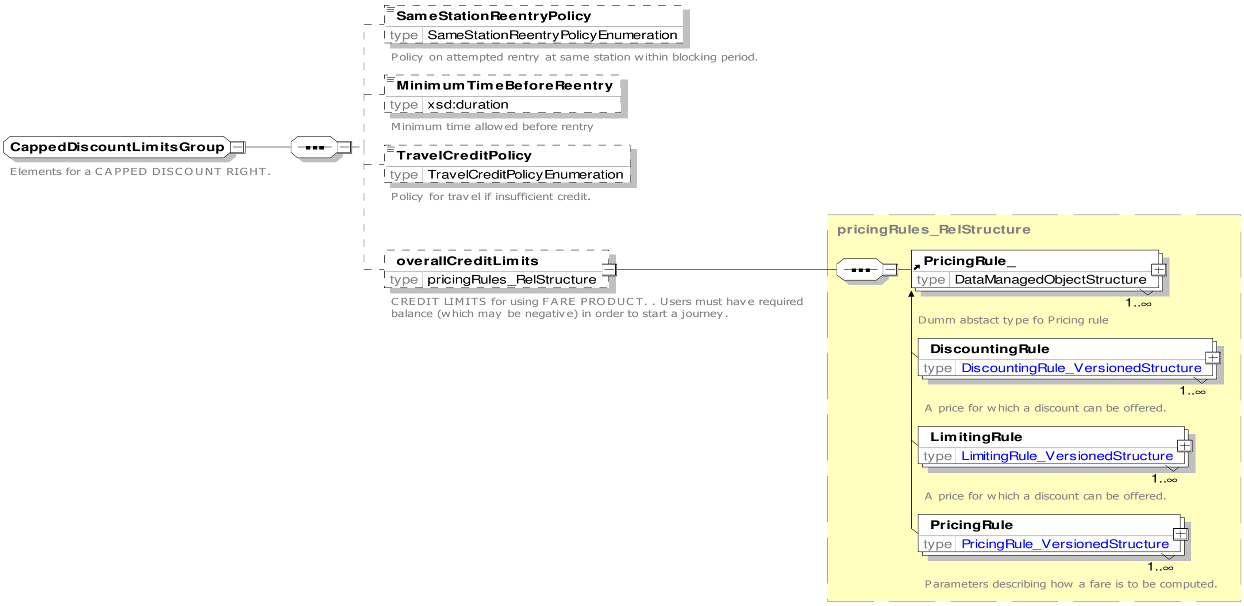


Figure 218 — *CappedDiscountLimitsGroup* — XSD

8.6.2.5.5.1.1 SameStationEntryPolicy – Allowed values

The following table shows the allowed values for ***SameStationEntryPolicy*** (*SameStationEntryPolicyEnum*).

Table 185 – *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.

8.6.2.5.5.1.2 TravelCreditPolicy – Allowed values

The following table shows the allowed values for ***TravelCreditPolicy*** (*TravelCreditPolicyEnum*).

Table 186 – *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

8.6.2.5.5.2 CappingStart – Allowed values

The following table shows the allowed values for *CappingStart* (*CappingStartEnum*.)

Table 187 – *CappingStart* – Allowed values

Value	Description
period	Capping period starts at day, week or month start.
firstUse	Capping period starts at first use.

8.6.2.5.6 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.

Table 188 – *CappingRule* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	CAPPING RULE inherits from VERSIONED CHILD. See NeTEx Part1.
«PK»	***id***	*CappingRuleIdType*	1:1	Identifier of CAPPING RULE.
	Name	*MultilingualString*	0:1	Name of CAPPING RULE.
	Description	*MultilingualString*	0:1	Description of CAPPING RULE.
GROUP	***CappingRule-Applicability-Group***	*CappingRule-ApplicabilityGroup*	0:1	Elements describing capping applicability.
GROUP	***CappingRule-PricingGroup***	*CappingRule-PricingGroup*	0:1	Elements describing capping prices.

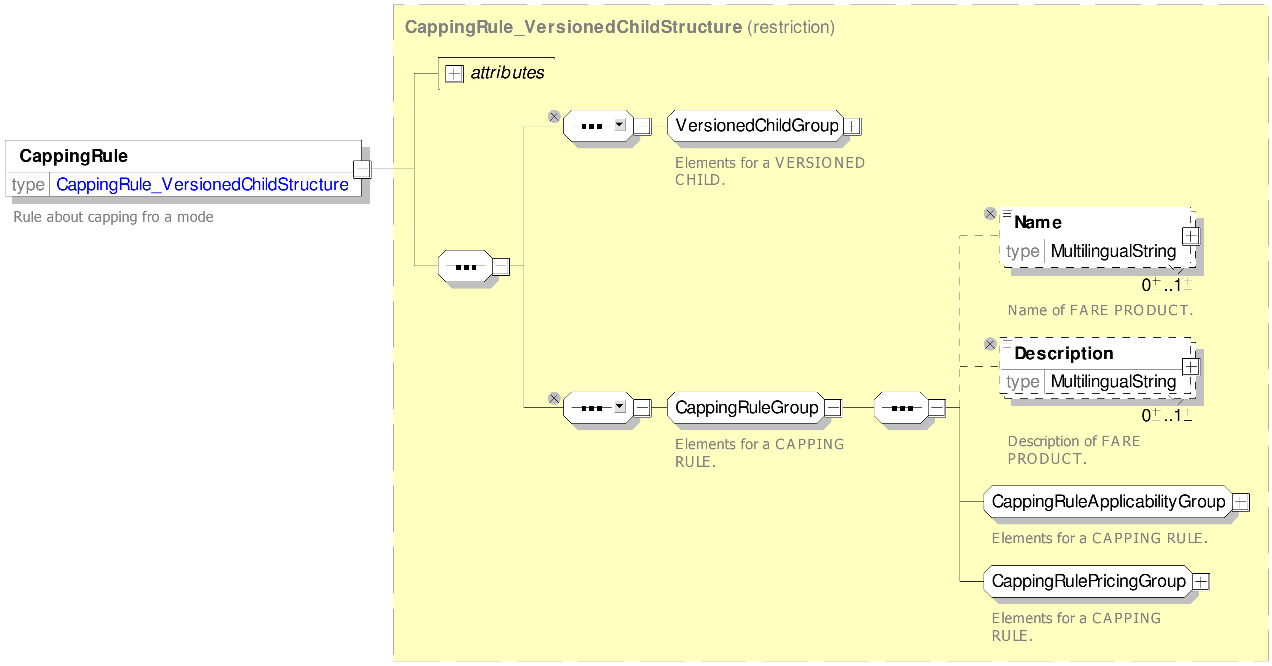


Figure 219 — *CappingRule* — XSD

8.6.2.5.6.1 CappingRuleApplicabilityGroup – Group

The ***CappingRuleApplicabilityGroup*** specifies the capping period for the CAPPED DISCOUNT RULE. Some capped products also have a limitation that the discount is issued as a rebate which must be collected within a certain time.

Table 189 – *CappingRuleApplicabilityGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	Validable-ElementRef	*ValidableElementRef*	0:1	VALIDABLE ELEMENT for which capping applies.
	CappingPeriod	*CappingPeriodEnum*	0:1	Period over which capping applies, e.g. daily. See allowed values below.
	CappingStart	*CappingStartEnum*	0:1	When does capping period start. See allowed values below.
«FK»	***TimeIntervalRef***	*TimeIntervalRef*	0:1	TIME INTERVAL over which capping applies.
	ExpireAfter-Period	*xsd:duration*	0:1	Period by which adjustment or rebate must be collected.

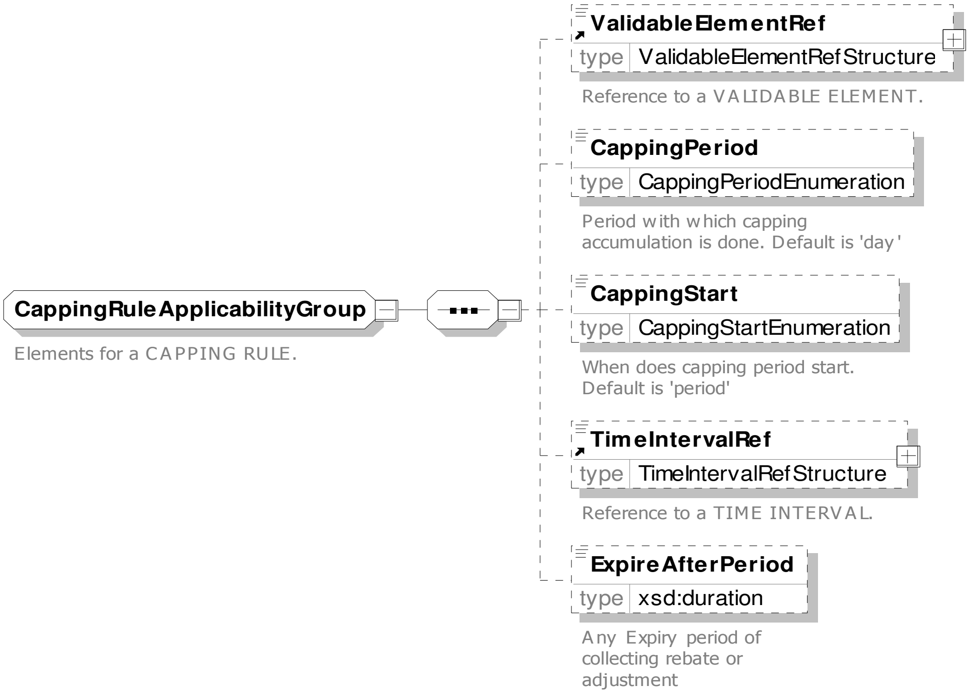


Figure 220 — *CappingRuleApplicabilityGroup* — XSD

8.6.2.5.6.2 CappingPeriod – Allowed values

The following table shows the allowed values for **CappingPeriod** (*CappingPeriodEnum*).

Table 190 – 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.

8.6.2.5.6.3 CappingStart – Allowed values

The following table shows the allowed values for **CappingStart** (*CappingStartEnum*).

Table 191 – CappingStart – Allowed values

Value	Description
period	Capping period starts at start of fare period, e.g. fare day.
firstUse	Capping period starts at time of first use.

8.6.2.5.6.4 CappingRulePricingGroup – Group

The ***CappingRulePricingGroup*** specifies pricing limits for a CAPPED DISCOUNT RULE.

Table 192 – *CappingRulePricingGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
“cntd»	***Maximum-Distance***	*LengthType*	0:*	Capping distance if distance based cap.
“cntd»	***maximumFares***	*FareTable*	0:*	Capping FARE PRICEs for this rule.
«FK»	***PreassignedFareProductRef***	*PreassignedFare-ProductRef*	0:1	PREASSIGNED FARE PRODUCT whose prices set cap for this product.
“cntd»	***creditLimits***	*FareProductPriceRef*	0:*	Credit limits that apply to use of this rule.

Figure 221 — *CappingRulePricingGroup* — XSD

8.6.2.5.7 SupplementProduct – Model Element

An additional FARE PRODUCT that may be used to describe additional purchases entitled by another product.

Table 193 – *SupplementProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PreassignedFare-Product*	::>	SUPPLEMENT PRODUCT inherits from PREASSIGNED FARE PRODUCT.
«PK»	***id***	*PreassignedFare-ProducOffertIdType*	1:1	Identifier of SUPPLEMENT PRODUCT.

«FK»	***FareProductRef***	*SupplementTo-FareProductRef*	1:1	Reference to base PRE ASSIGNED FARE PRODUCT OFFER for which this is a supplement.

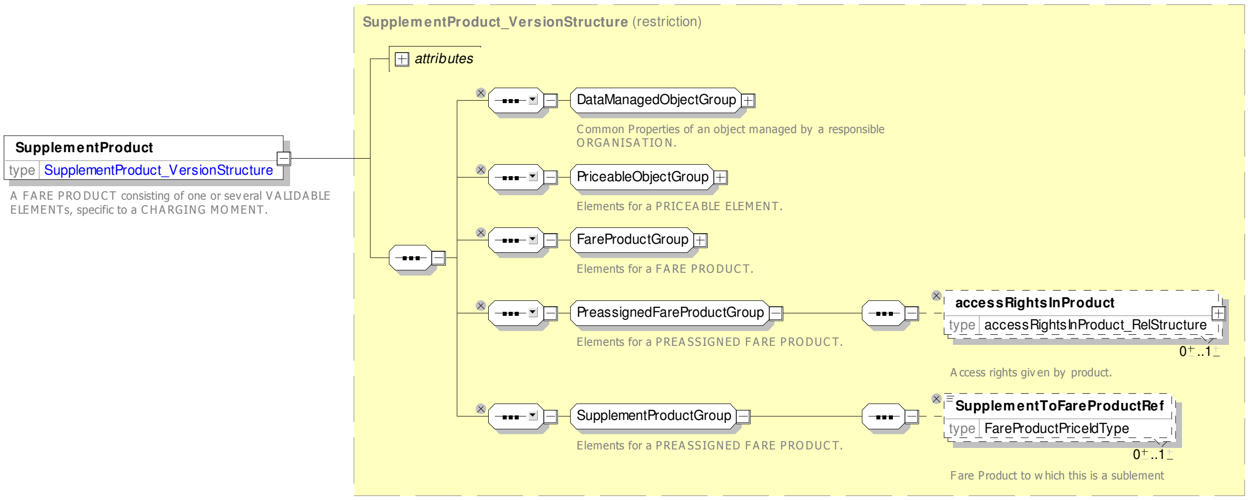


Figure 222 — *SupplementProduct* — XSD

8.6.2.5.8 EntitlementProduct – Model Element

An additional ACCESS RIGHT that gives entitlement to buy or use other products and may be a perquisite.

Table 194 – *EntitlementProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*ServiceAccessRight*	::>	ENTITLEMENT PRODUCT inherits from SERVICE ACCESS RIGHT
«PK»	***id***	*EntitlementProduct-IdType*	1:1	Identifier of ENTITLEMENT PRODUCT.
«FK»	***General-OrganisationRef***	*GeneralOrganisationRef*	1:1	Reference to ORGANISATION offering product.

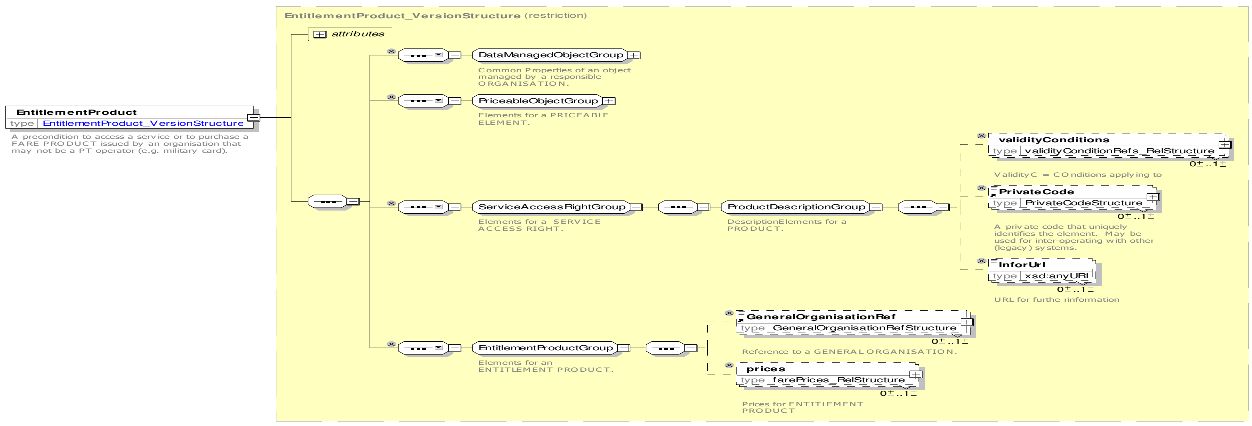


Figure 223 — *EntitlementProduct* — XSD

8.6.2.5.9 ThirdPartyProduct – Model Element

A FARE PRODUCT that is marketed together with a Public Transport Fare Product.

Table 195 – *ThirdPartyProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareProduct*	::>	THIRD PARTY PRODUCT inherits from FARE PRODUCT.
«PK»	***id***	*ThirdPartyProductIdType*	1:1	Identifier of THIRDPARTY PRODUCT.
"cntd»	***GeneralGroupOfEntities***	*GeneralGroupOfEntities*	0:*	GENERAL GROUP OF ENTITIES associated with Third PARTY product.

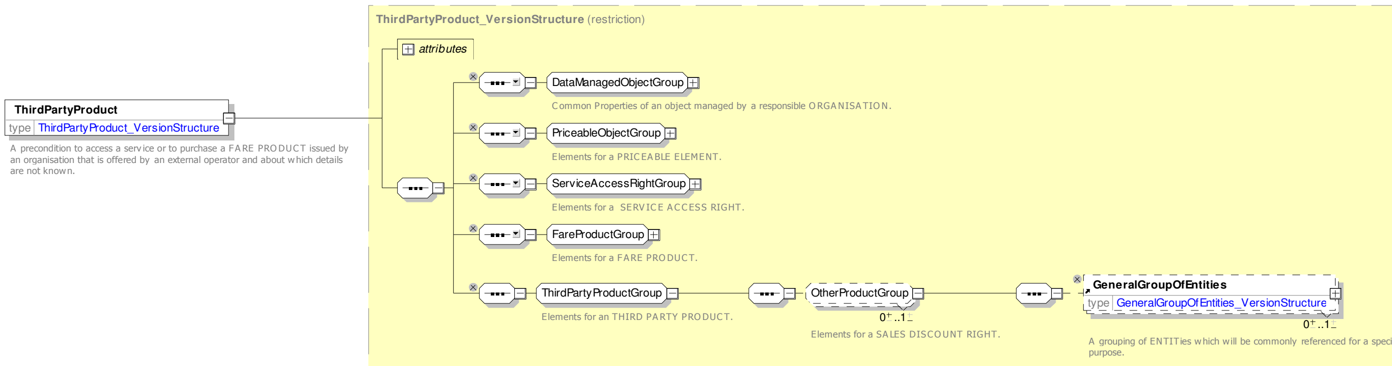


Figure 224 — *ThirdPartyProduct* — XSD

8.6.2.6 Fare Product – XML examples

[TO DO MORE EXAMPLES]

8.6.2.6.1 FareProduct: XML Example of Preassigned Fare product – Prepaid ticket

The following code fragment shows two PREASSIGNED FARE PRODUCT 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::PrepaidFare">
    <Name>Cash Ride Paid for before ride – </Name>
    <ChargingMomentRef version="any" ref="tfl::prepaid"/>

    <validityParameterAssignments>
        <GenericParameterAssignment version="any" id="tfl::PrepaidFare@singleRide">
            <Name>A single ride</Name>
            <FrequencyOfUseRef version="any" ref="tfl::oneTrip"/>
            <RoundTripRef version="any" ref="tfl::single"/>
            <UsageValidityPeriodRef version="any" ref="tfl:UsageValidityPeriod:endOfRide"/>
        </GenericParameterAssignment>

        <GenericParameterAssignment version="any" id="tfl::PrepaidFare@returnRide">
            <Name>A single ride</Name>
            <FrequencyOfUseRef version="any" ref="tfl::oneTrip"/>
            <RoundTripRef version="any" ref="tfl::return"/>
            <UsageValidityPeriodRef version="any" ref="tfl:UsageValidityPeriod:endOfRide"/>
        </GenericParameterAssignment>
    </validityParameterAssignments>

    <accessRightsInProduct>
        <AccessRightInProduct id="tfl::PrepaidFare@metroTrip" version="any">
            <Name>Metro trip </Name>
            <Description>from entry barrier to exit barrier </Description>
            <ValidableElementRef ref="lul::metroTrip" version="any"/>
            <accessRightParameterAssignments>
                <AccessRightParameterAssignmentRef version="any"
                    ref="tfl::CanInterchange_CannotBreakJourney"/>
            </accessRightParameterAssignments>
        </AccessRightInProduct>

        <AccessRightInProduct id="tfl::PrepaidFare@railTrip" version="any">
            <Name>Metro trip </Name>
            <ValidableElementRef ref="nr::railTrip" version="any"/>
            <accessRightParameterAssignments>
                <AccessRightParameterAssignmentRef version="any" ref="tfl::CanBreakJourney"/>
            </accessRightParameterAssignments>
        </AccessRightInProduct>
        <AccessRightInProduct id="tfl::PrepaidFare@busOrTramTrip" version="any">
            <Name>Bus or tram trip trip </Name>
            <ValidableElementRef version="any" ref="lbsl::busOrTramTrip"/>
            <accessRightParameterAssignments>
                <AccessRightParameterAssignmentRef version="any"
                    ref="tfl::NoInterchange_CannotBreakJourney"/>
            </accessRightParameterAssignments>
        </AccessRightInProduct>
        <AccessRightInProduct id="tfl::PrepaidFare@riverTrip" version="any">
            <Name>River trip </Name>
            <ValidableElementRef ref="lrs::riverTrip" version="any"/>
            <accessRightParameterAssignments>
                <AccessRightParameterAssignmentRef version="any"
                    ref="tfl::NoInterchange_CannotBreakJourney"/>
            </accessRightParameterAssignments>
        </AccessRightInProduct>
        <AccessRightInProduct id="tfl::PrepaidFare@riverHopOnOff" version="any">
            <Name>River trip </Name>
            <ValidableElementRef ref="lrs::riverHopOnOff" version="any"/>
            <accessRightParameterAssignments>
                <AccessRightParameterAssignmentsRef version="any"
                    ref="tfl::CanBreakJourney"/>
            </accessRightParameterAssignments>
        </AccessRightInProduct>
        <AccessRightInProduct id="tfl::PrepaidFare@cablewayTrip" version="any">
            <Name>River trip </Name>
            <ValidableElementRef ref="ea::cablewayTrip" version="any"/>
            <accessRightParameterAssignments>
                <AccessRightParameterAssignmentRef version="any"
                    ref="tfl::NoInterchange_CannotBreakJourney"/>
            </accessRightParameterAssignments>
```

```xml
            </AccessRightInProduct>
        </accessRightsInProduct>
    </PreassignedFareProduct>


    <PreassignedFareProduct version="any" id="tfl::PayAsYouGoFare">
        <Name>Cash Ride Paid for before ride - </Name>
        <ChargingMomentRef version="any" ref="tfl::payAsYouGo"/>

        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl::PayAsYouGoFare@Ride">
                <Name>Ticket Allows only one trip</Name>
                <for>
                    <FrequencyOfUseRef version="any" ref="tfl::oneTrip"/>
                    <UsageValidityPeriodRef version="any" ref="tfl::endOfRide"/>
                </for>
            </GenericParameterAssignment>
        </validityParameterAssignments>

        <accessRightsInProduct>
            <AccessRightInProductRef ref="tfl::PrepaidFare@metroTrip" version="any"/>
            <AccessRightInProductRef ref="tfl::PrepaidFare@railTrip" version="any"/>
            <AccessRightInProductRef ref="tfl::PrepaidFare@busOrTramTrip" version="any"/>
            <AccessRightInProductRef ref="tfl::PrepaidFare@riverTrip" version="any"/>
            <AccessRightInProductRef ref="tfl::PrepaidFare@riverHopOnOff" version="any"/>
            <AccessRightInProductRef ref="tfl::PrepaidFare@cablewayTrip" version="any"/>
        </accessRightsInProduct>
    </PreassignedFareProduct>
```

8.6.2.6.2 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:

```xml
    <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>

        <validityParameterAssignments>
            <GenericParameterAssignment version="any" id="tfl::TravelCard@give@TCDisco">
                <EntitlementGivenRef version="any" ref="tfl::TravelCardSaleDiscount"/>
                <EntitlementGivenRef version="any" ref="tfl::AnnualPassGoldCard"/>
            </GenericParameterAssignment>
            <GenericParameterAssignment version="any" id="tfl::TravelCard@periods">
                <Name>Travel Periods </Name>
                <LimitationGroupingType>OR</ LimitationGroupingType >
                <for>
                    <UsageValidityPeriodRef version="any"
                        ref="tfl::TravelCardValidityPeriod@1DAnyTime"/>
                    <UsageValidityPeriodRef version="any"
                        ref="tfl::TravelCardValidityPeriod@1DOffPeak"/>
                    <UsageValidityPeriodRef version="any"
                        ref="tfl::TravelCardValidityPeriod@1WAnyTime"/>
                </for>
            </GenericRightParameterAssignment>


            <GenericParameterAssignment version="any" id="tfl::TravelCard@Transferability">
                <Name>No Transfer by anyone.</Name>
                <for>
                    <TransferabilityRef ref="tfl::NoTransfer" version="any"/>
                    <UserProfileRef version="any" ref="tfl::anyone"/>
                </for>
            </GenericParameterAssignment>
```

```xml
            <GenericParameterAssignment version="any" id="tfl::TravelCard@UserProfile@anyone">
                <Name>Anyone can use at full fare.</Name>
                <ChargingBasis>normalFare</ChargingBasis>
                <for>
                    <UserProfileRef version="any" ref="tfl::anyone"/>
                </for>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any"
                        id="tfl::TravelCard@UserProfile@concession">
                <Name>Concession card holder</Name>
                <ChargingBasis>discounted</ChargingBasis>
                <for>
                    <UserProfileRef version="any" ref="tfl::concession"/>
                </for>
            </GenericParameterAssignment>
        </validityParameterAssignments>
    </PreassignedFareProduct>
```

8.6.2.6.3 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:AmountOfPriceUnitProduct:OysterTopUp">
        <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::OysterTopUp@Req@PAYG">
                <EntitlementRequiredRef version="any" ref="tfl::PayAsYouGoRightHolder"/>
            </GenericParameterAssignment>
        </validityParameterAssignments>
        <prices>
            <FareProductPrice version="any" id="tfl::OysterTopUp">
                <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>
```

8.6.2.6.4 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">
                <for>
```

```xml
                    <EntitlementRequiredRef version="any" ref="tfl:Req:TravelCardHolder"/>
                </for>
            </GenericParameterAssignment>

            <GenericParameterAssignment version="any"
                    id="tfl::TCSDiIsco@Give@DiscountedPayAsYouGoFare">
                <for>
                    <EntitlementGivenRef version="any" ref="tfl::PayAsYouGoFare"/>
                    <EntitlementGivenRef version="any" ref="tfl::DiscountedPrepaidFare"/>
                </for>
            </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>
                        <Scope>
                            <VehicleModes>water</VehicleModes>
                            <GroupOfLinesRef version="any" ref="lrs::RiverBus"/>
                        </Scope>
                    </GenericParameterAssignment>

                    <GenericParameterAssignment version="any"
                        id="tfl::TCSDiIsco@VehicleMode@cableway">
                        <ChargingBasis>discounted</ChargingBasis>
                        <Scope>
                            <VehicleModes>cableway</VehicleModes>
                        </Scope>
                    </GenericParameterAssignment>
                </includes>
            </GenericParameterAssignment >

            <GenericParameterAssignment version="any" id="tfl::TCSDiIsco@Transferability">
                <Name>Card can not be used by another adult</Name>
                <for>
                    <TransferabilityRef ref="tfl::NoTransfer" version="any"/>
                </for>
            </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::PrepaidFare"/>
            </FareTable>
        </priceGroups>

    </SaleDiscountRight>
```

8.6.2.6.5 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:

```xml
<!-- OYSTER PAY AS YOU GO CARD -->
    <CappedDiscountRight version="any" id="tfl:CappedDiscountRight: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">
                <for>
                    <EntitlementGivenRef version="any" ref="tfl::PAYGFare"/>
                </for>
            </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@VehicleModes@modes">
                        <Name>Can go on all modes   </Name>
                        <Scope><VehicleModes>metro tram bus cableway</VehicleModes></Scope>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment version="any" id="tfl::Oyster@rail@Overground">
                        <Name>Can go on overground Rail journeys</Name>
                        <Scope>
                            <VehicleModes>rail</VehicleModes>
                            <FareZoneRef version="any" ref="tfl::Overground"/>
                        </Scope>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment version="any"
id="tfl::Oyster@FareZone@rail@NROysterArea">
                        <Name>Can go on Rail journeys within Rail Oyster area</Name>
                        <Scope>
                            <VehicleModes>rail</VehicleModes>
                            <FareZoneRef version="any" ref="nr::NationalRailOysterArea"/>
                        </Scope>
                    </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>
                        <Scope><VehicleModes>water</VehicleModes>                     </Scope>
                    </GenericParameterAssignment>
                    <GenericParameterAssignment version="any" id="tfl::Oyster@canTakeChildren">
                        <for>
                            <GroupTicketRef version="any" ref="tfl::familyGroup"/>
                    </for>
                    </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>
                        <for>
                            <TransferabilityRef ref="tfl::CanTransfer" version="any"/>
                            <UserProfileRef version="any" ref="tfl::adult"/>
                        </for>
                    </GenericParameterAssignment>
```

```xml
                    <GenericParameterAssignment version="any"
id="tfl::Oyster@Transferability@concessions">
                        <Name>Disabled card holder</Name>
                        <ChargingBasis>free</ChargingBasis>
                        <for>
                            <TransferabilityRef ref="tfl::NoTransfer" version="any"/>
                            <UserProfileRef version="any" ref="tfl::concession"/>
                        </for>
                    </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">
                <CappingPeriod>day</CappingPeriod>
                <CappingStart>period</CappingStart>
                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1d"/>
                <maximumPriceGroups>
                    <FareTable id="tfl::Oyster@rail" version="any">
                        <members>
                            <Cell version="any" id="tfl:Cell:Oyster@MaxPrice@rail@adult">
                                <Price>
                                    <Name>Maximum daily price to charge</Name>
                                    <MaximumLimitPrice>8.50</MaximumLimitPrice>
                                </Price>
                                <ValidableElementRef version="any" ref="nr::railTrip"/>
                                <UserProfileRef version="any" ref="tfl::adult"/>
                                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1d"/>
                            </Cell>
                        </members>
                    </FareTable>
                </maximumPriceGroups>
                <PreassignedFareProductRef version="any" ref="tfl::TravelCardOnOyster"/>
            </CappingRule>
            <CappingRule id="tfl::Oyster@metro" version="any">
                <CappingPeriod>day</CappingPeriod>
                <CappingStart>period</CappingStart>
                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1d"/>
                <maximumPriceGroups>
                    <FareTable id="tfl::Oyster@metro" version="any">
                        <members>
                            <Cell version="any" id="tfl::Oyster@MaxPrice@metro@adult">
                                <Price>
                                    <Name>Maximum daily price to charge</Name>
                                    <MaximumLimitPrice>8.50</MaximumLimitPrice>
                                </Price>
                                <ValidableElementRef version="any" ref="lul::metroTrip"/>
                                <UserProfileRef version="any" ref="tfl::adult"/>
                                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1d"/>
                            </Cell>
                        </members>
                    </FareTable>
                </maximumPriceGroups>
                <PreassignedFareProductRef version="any" ref="tfl::TravelCardOnOyster"/>
            </CappingRule>
            <CappingRule id="tfl::Oyster@bus" version="any">
                <CappingPeriod>day</CappingPeriod>
                <CappingStart>period</CappingStart>
                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1d"/>
                <maximumPriceGroups>
                    <FareTable id="tfl::Oyster@bus" version="any">
                        <members>
                            <Cell version="any" id="tfl::Oyster@MaxPrice@Bus_Tram@adult">
                                <Price>
                                    <Name>Maximum daily price to charge</Name>
                                    <MaximumLimitPrice>4.40</MaximumLimitPrice>
```

```
                                </Price>
                                <ValidableElementRef version="any" ref="lbsl::busOrTramTrip"/>
                                <UserProfileRef version="any" ref="tfl::adult"/>
                                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1d"/>
                            </Cell>
                        </members>
                    </FareTable>
                </maximumPriceGroups>
                <PreassignedFareProductRef version="any" ref="lbsl::TravelCardOnOyster@Bus"/>
            </CappingRule>
            <CappingRule id="tfl::Oyster@river" version="any">
                <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>
                <maximumPriceGroups>
                    <FareTable id="tfl::Oyster@river" version="any">
                        <members>
                            <ValidableElementPrice version="any"
id="tfl:ValidableElementPrice:Oyster@MaxPrice@river@adult">
                                <IsAllowed>false</IsAllowed>
                                <ValidableElementRef version="any" ref="lrs::riverTrip"/>
                            </ValidableElementPrice>
                        </members>
                    </FareTable>
                </maximumPriceGroups>
            </CappingRule>

            <CappingRule id="tfl::Oyster@cableway" version="any">
                <Description>If you use Oyster to pay as you go for 5 or more Emirates Air Line
journeys in a calendar week (Sunday to Saturday), you'll get the 5+ Reward. The 5+ Reward is 50
per cent off the total price paid for 5 journeys or more during a week.

            How do I get the 5+ Reward?
            If you make 5 journeys or more in a calendar week, you'll pay the full pay as you go
fare for all the journeys you made. However, when you travel the following week, 50 per cent of
the total cost of your previous week's journeys will be credited back to your Oyster card.

            At the end of each week, we will work out how much to reward you and automatically
send this to the Emirates Air Line gates ready for you to collect from the following week.

            The 5+ Reward will be loaded on to your Oyster card as pay as you go credit when you
touch your Oyster card on a yellow card reader at the Emirates Air Line gates. The 5+ Reward can
only be collected as part of an Emirates Air Line journey.

            Your 5+ Reward will be available to collect from the next calendar week for up to 6
weeks; after this, it will expire.</Description>
                <CappingPeriod>week</CappingPeriod>
                <CappingStart>period</CappingStart>
                <ExpireAfterPeriod>P42D</ExpireAfterPeriod>
                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1w"/>
                <maximumPriceGroups>
                    <FareTable id="tfl::Oyster@cableway" version="any">
                        <members>
                            <Cell version="any" id="tfl:Cell:Oyster@MaxPrice@cableway@adult">
                                <Price>
                                    <Name>5+ discount  </Name>
                                    <LimitingRule>
                                        <DiscountAsPercentage>50</DiscountAsPercentage>
                                        <MinimumPriceAsMultiple>5</MinimumPriceAsMultiple>
                                    </LimitingRule>
                                </Price>
                                <ValidableElementRef version="any" ref="ea::cablewayTrip"/>
                                <UserProfileRef version="any" ref="tfl::adult"/>
                                <TimeIntervalRef version="any" ref="tfl:TimeInterval:1w"/>
                            </Cell>
                        </members>
                    </FareTable>
                </maximumPriceGroups>
            </CappingRule>
        </cappingRules>
    </CappedDiscountRight>
```

8.6.2.6.6 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:SupplementProduct:OysterAutoTopUp">
        <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::OysterAutoTopUp@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:: PayAsYouGoRightHolder"/>
            </GenericParameterAssignment>
            <GenericParameterAssignment version="any" id="tfl::OysterAutoTopUp@PurchaseWindow">
                <Description>You need to activate within eight days</Description>
                <for>
                    <PurchaseWindowRef version="any" ref="tfl::CollectWithin8Days"/>
                </for>
            </GenericParameterAssignment>
        </validityParameterAssignments>
        <prices>
            <FareProductPrice version="any" id="tfl::OysterAutoTopUp">
                <Name>Auto Top up must be between the following </Name>
                <MinimumPrice>20</MinimumPrice>
                <MaximumPrice>40</MaximumPrice>
                <RoundingRef ref="tfl:Rounding:TopUpIn20PoundSteps" version="any"/>
            </FareProductPrice>
        </prices>
    </SupplementProduct>
```

8.7 Pricing

8.7.1 Fare Calculation Parameters

8.7.1.1 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 PARAMETEs.

[TO DO] UPDATE MODEL

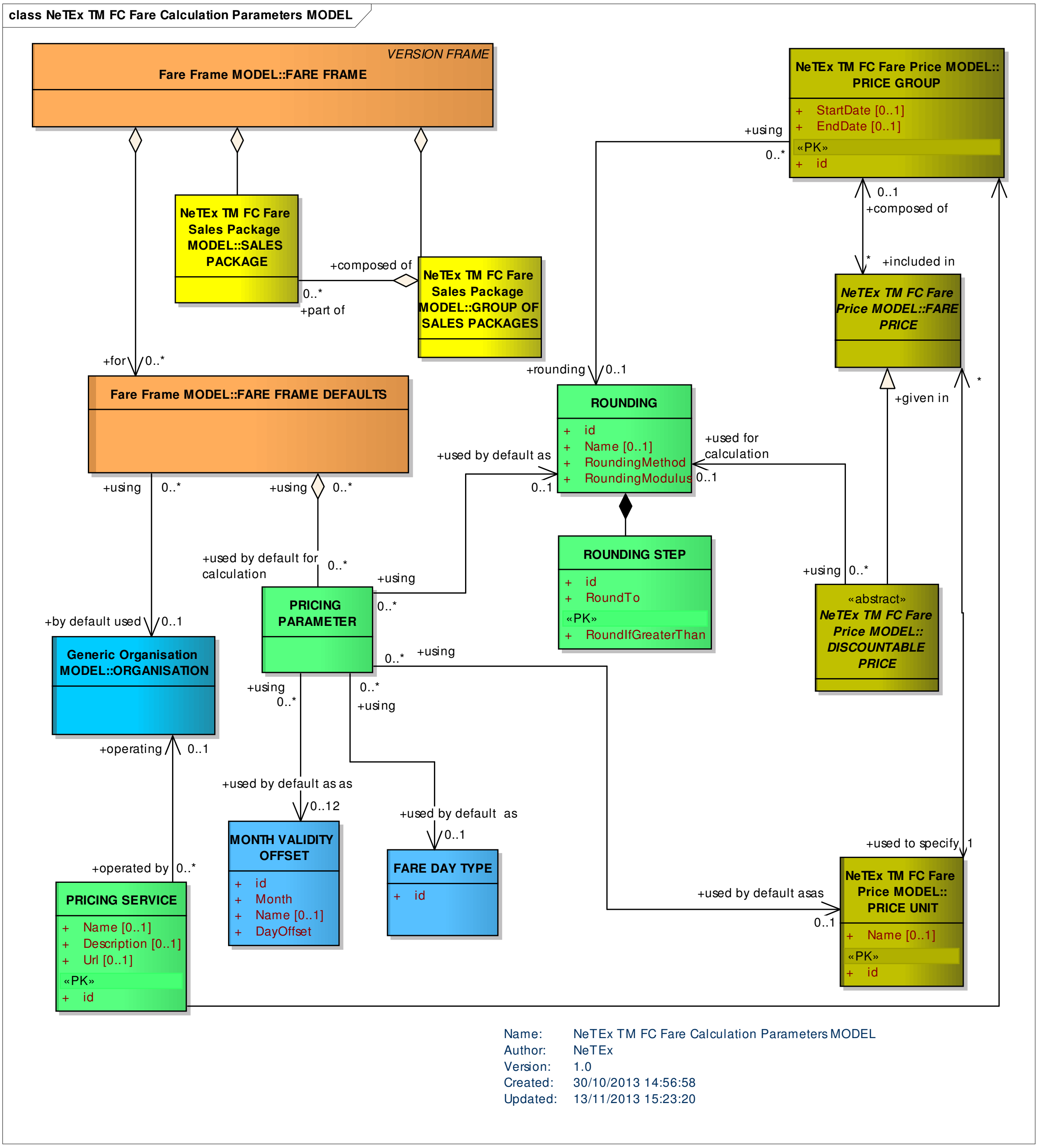


Figure 225 — Fare Calculation Parameters – Conceptual Model

8.7.1.2 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 calcualtions.

- ROUNDING defines rules for rounding the results of calculations. This may be by interval or to a prescribed rROUNDING STEP.

- FARE DAY TYPE defines the day end for fare prupsoes.
- PRICING SERVICE indicates any dynamic srevcie to use to fetch prices.

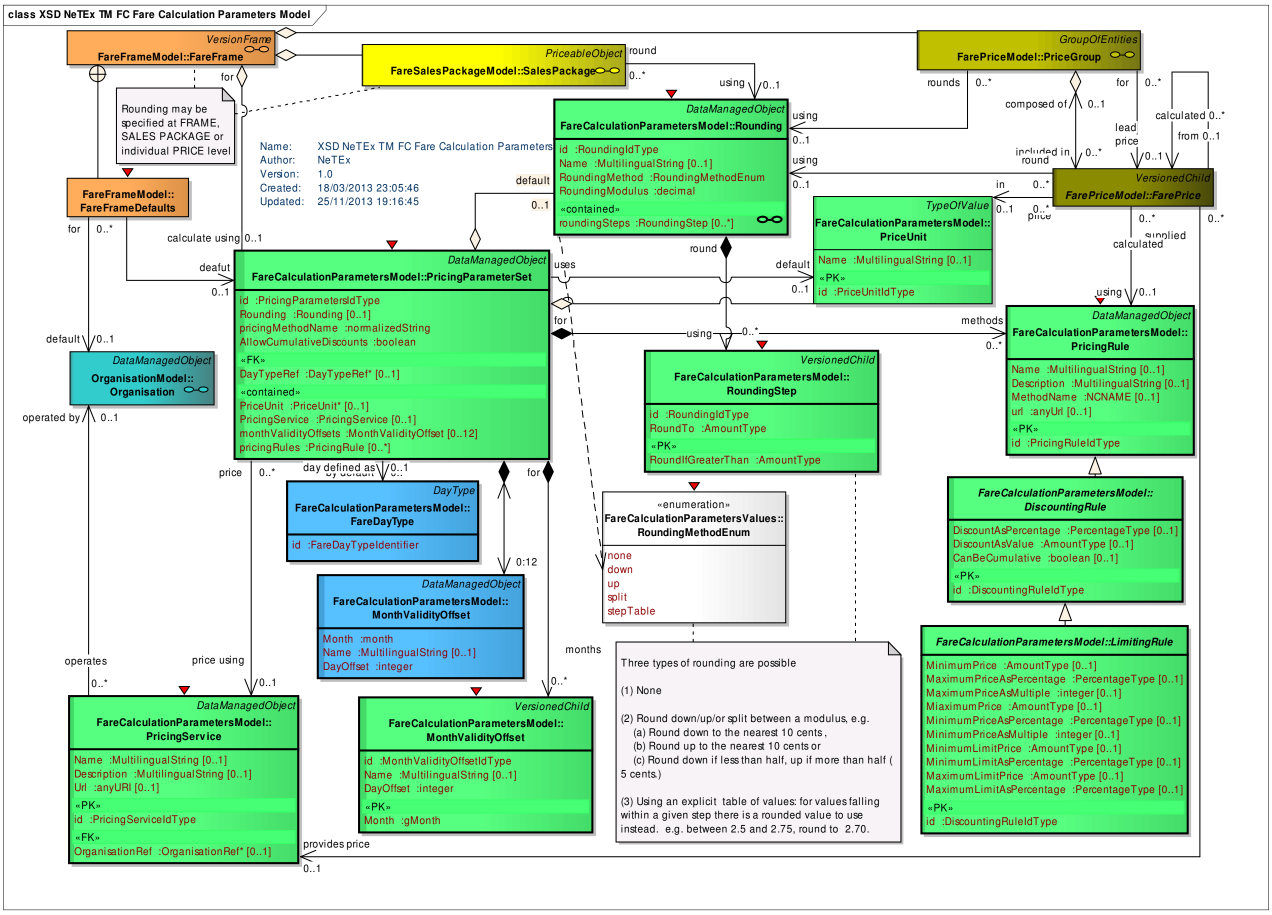


Figure 226 — Fare Calculation Parameters – Physical Model

8.7.1.3 Fare Calculation Parameters – Attributes and XSD

8.7.1.3.1 PricingParameterSet – Model Element

A set of reusable Pricing Parameters directing the rounding of values that are the result of calculations.

Table 196 – *PricingParameterSet* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	PRICING PARAMETER SET inherits from DATA MANAGED OBJECT. See NeTEx Part1.
	id	*PricingParameterSet-IdType*	1:1	Identifier of PRICING PARAMETERs.
	Name	*MultilingualString*	0:1	Name of PRICING PARAMETERs.
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a default PRICE UNIT.
«cntd»	***priceUnits***	*PriceUnit*	0:*	Available PRICE UNITs.

«cntd»	***pricingRules***	*PricingRule*	0:12	PRICING RULEs available to use in pricing.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Reference to a default ROUNDING.
«cntd»	***roundings***	*Rounding*	0:12	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.

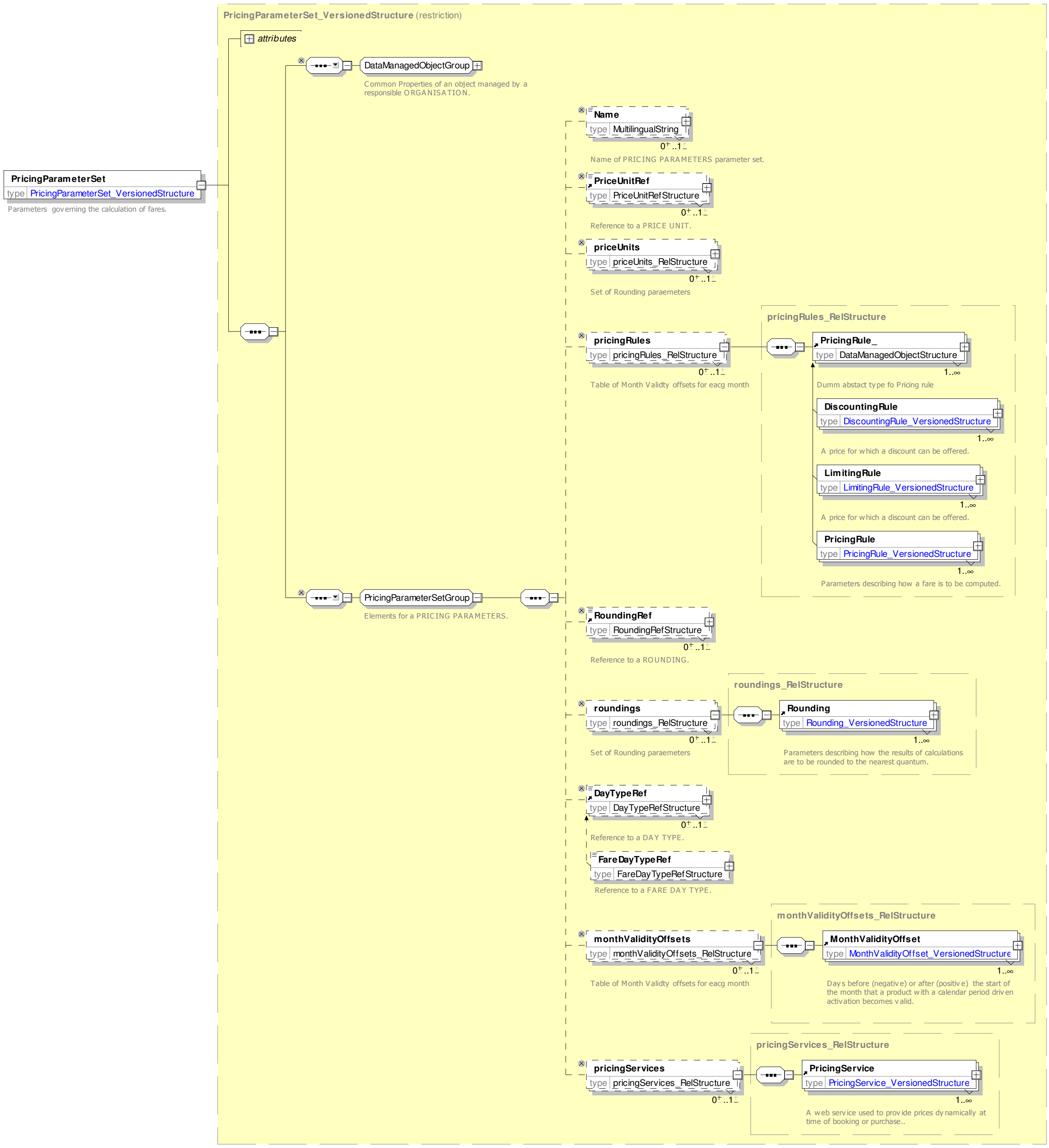


Figure 227 — *PricingParameterSet* — XSD

8.7.1.3.1.1 *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 197 – *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.

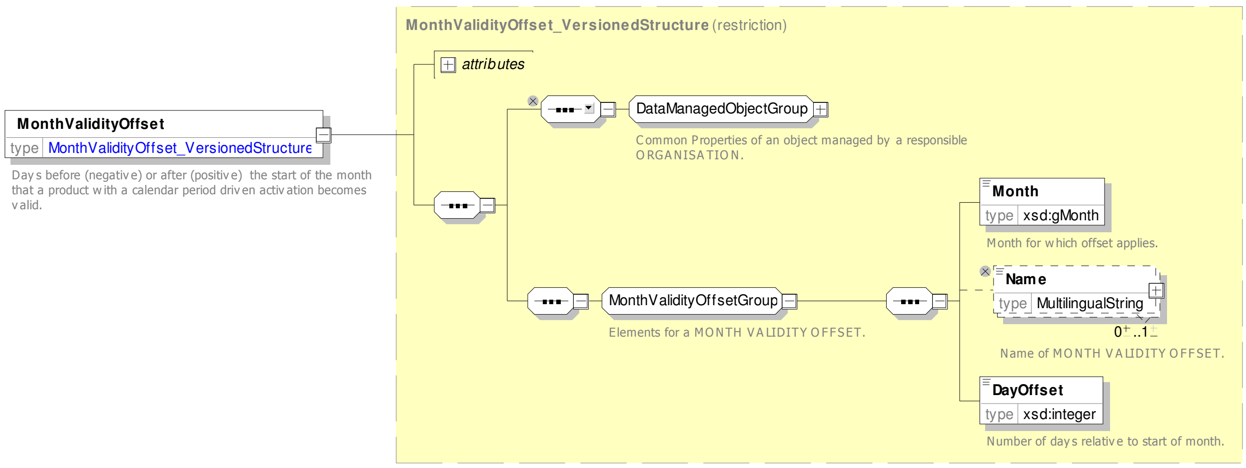


Figure 228 — *MonthValidityOffset* — XSD

8.7.1.3.2 PricingRule – Model Element

A named rule for compute one price from another price.

Table 198 – *PricingRule* – Element

Classifi-cation	Name	Type	Cardinality	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.
	Factor	*xsd:decimal*	0:1	Numeric factor associated with PRICING RULE.
	url	*xsd:anyURI*	0:1	url associated with PRICING RULE.

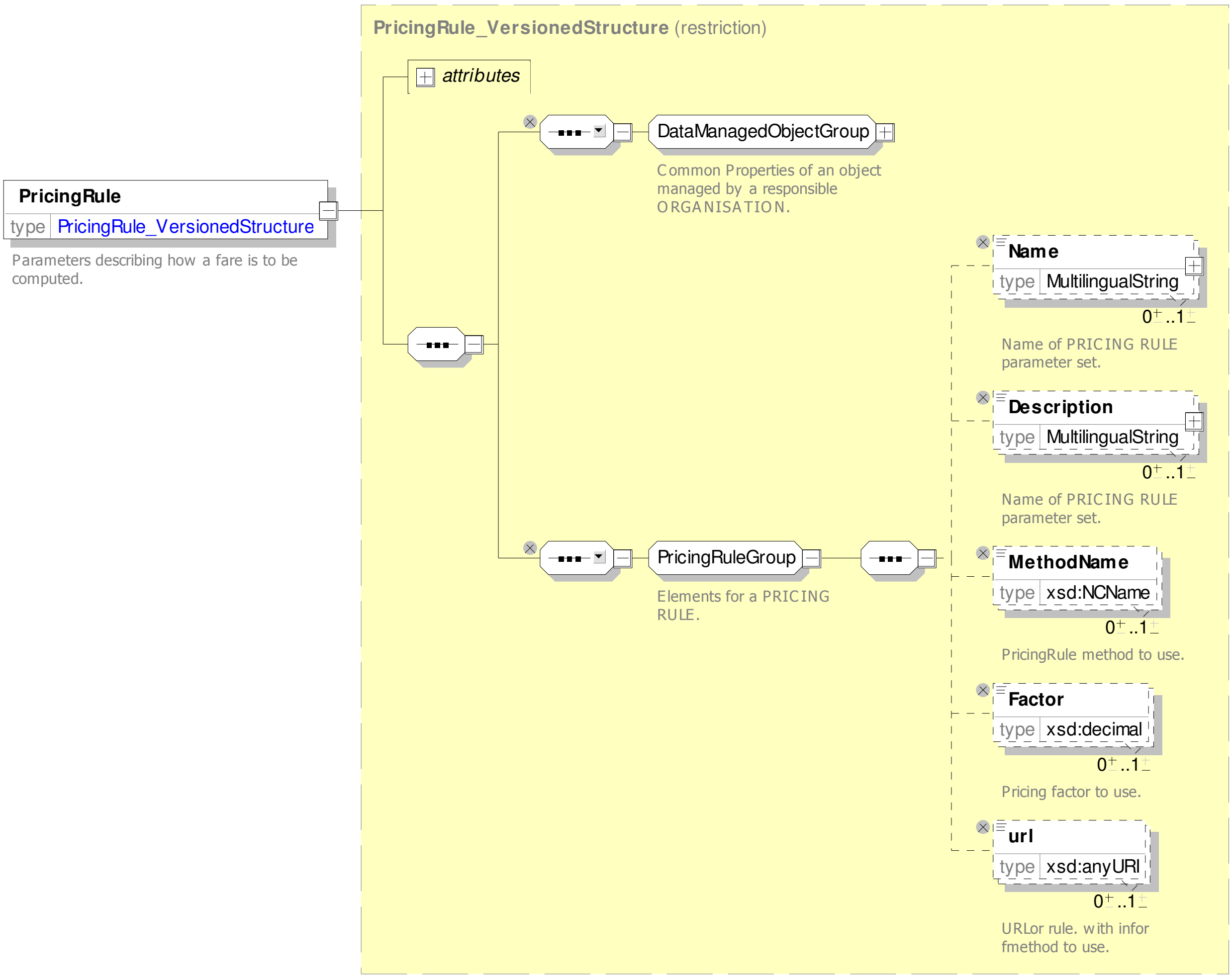


Figure 229 — *PricingRule* — XSD

8.7.1.3.3 DiscountingRule – Model Element

Parameters of a rule for computing a discounted price from another price.

Table 199 – *DiscountingRule* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PricingRule*	::>	DISCOUNTING RULE inherits from PRICING RULE.
«PK»	***id***	*DiscountingRuleIdType*	1:1	Identifier of FARE PRICE
	Discount-	*PercentageType*	0:1	Discount of PRICE as a percentage.

	AsPercentage			
	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.

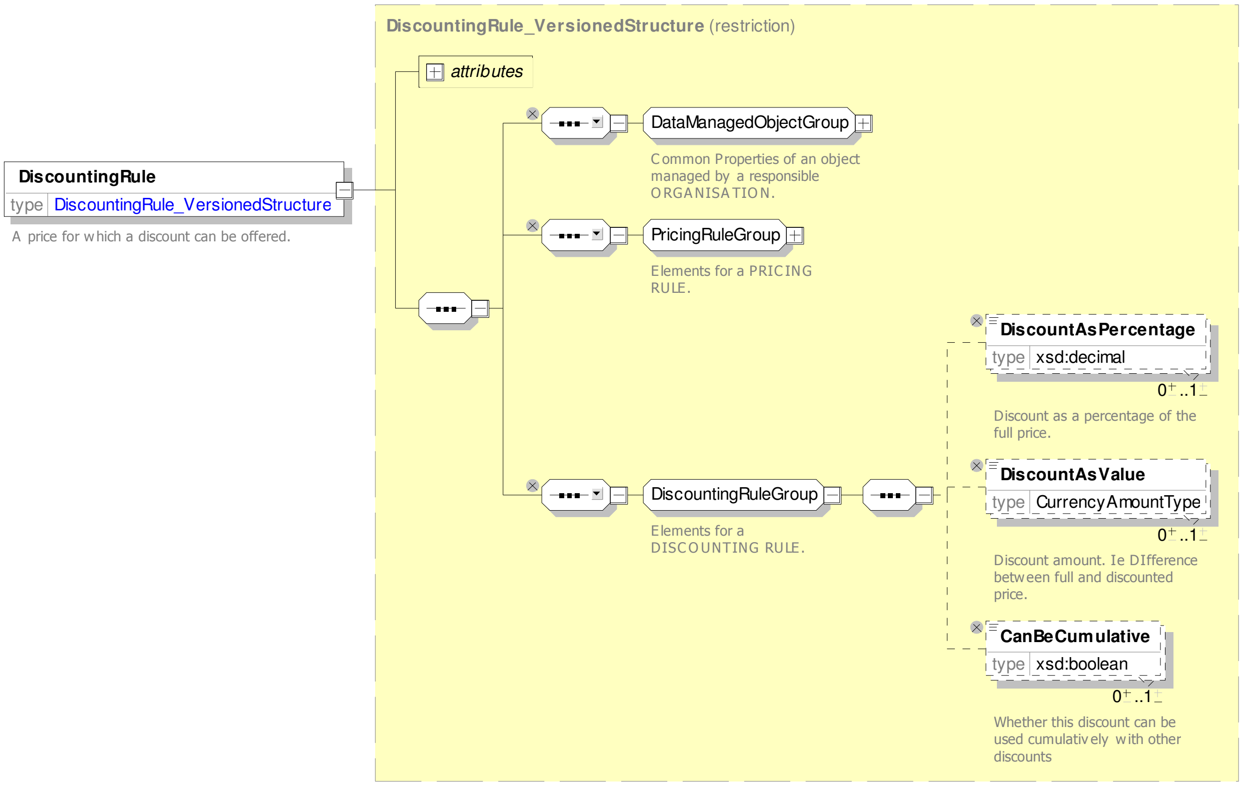


Figure 230 — *DiscountingRule* — XSD

8.7.1.3.4 LimitingRule – Model Element

Parameters of a rule for computing a price from another price subject to minima or maxima.

Table 200 – *LimitingRule* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DiscountingRule*	::>	LIMITING RULE from DISCOUNTING RULE.
«PK»	***id***	*DiscountingRule IdType*	1:1	Identifier of LIMITING RULE.

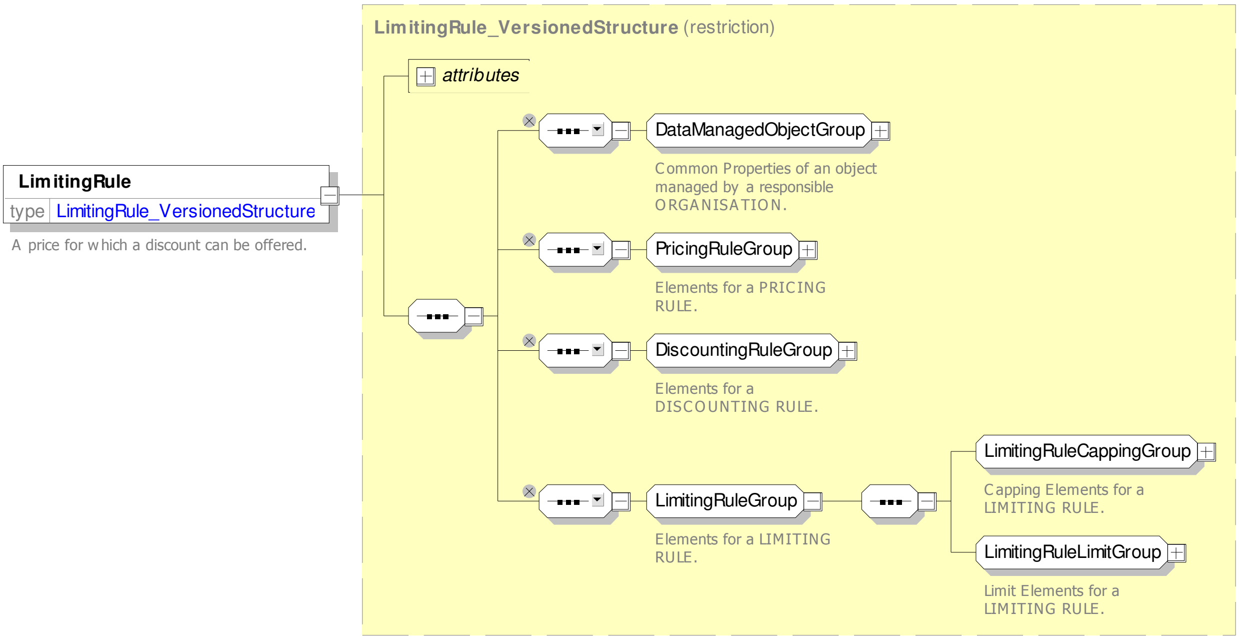


Figure 231 — *LimitingRule* — XSD

8.7.1.3.4.1 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 201 – *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.

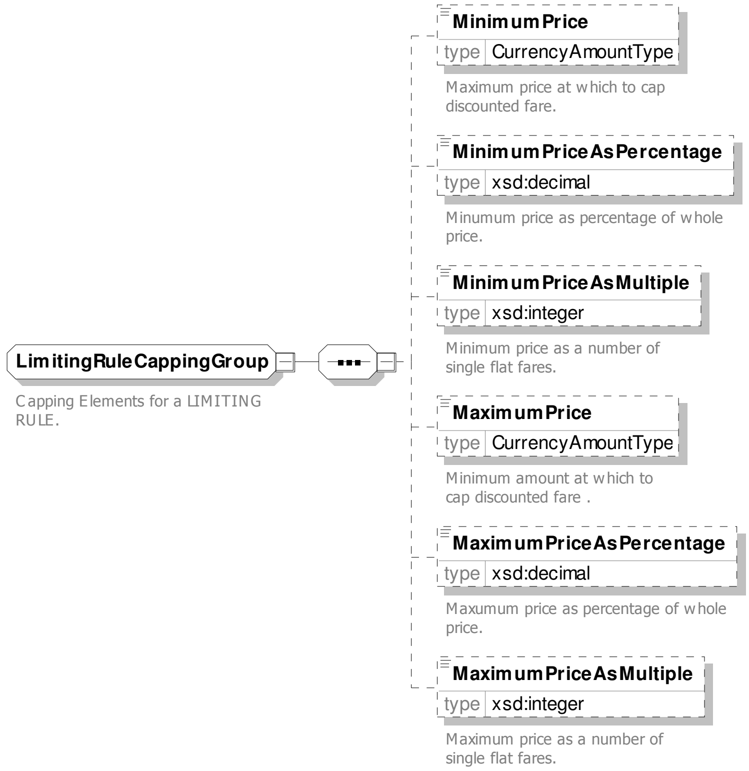


Figure 232 — *LimitingRuleCappingGroup* — XSD

8.7.1.3.4.2 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 202 – *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.

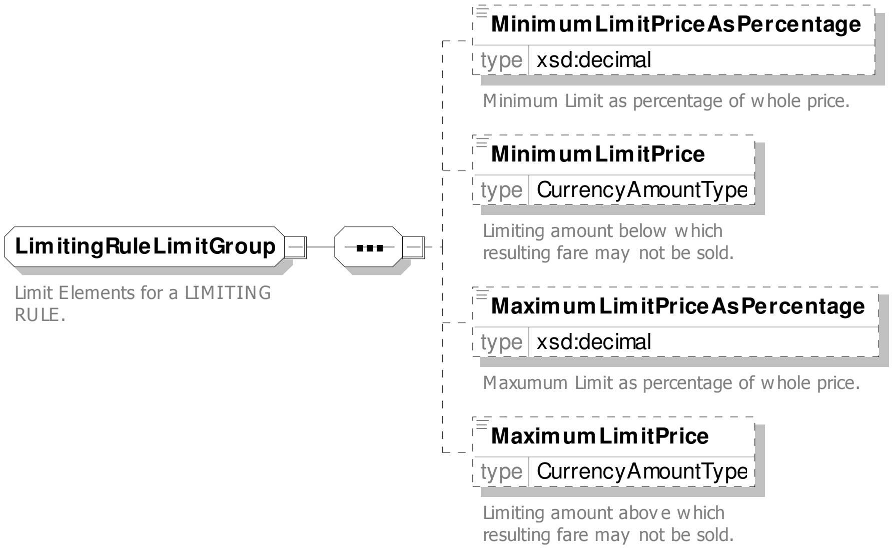


Figure 233 — *LimitingRuleLimitGroup* — XSD

8.7.1.3.5 Rounding – Model Element

Parameters directing the rounding of values that are the result of calculations.

Table 203 – *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.
	RoundingMethod	*RoundingMethodEnum*	1:1	Method to use to *round: down , up , split, none.* See allowed values below.
	Rounding-Modulus	*decimal*	1:1	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.

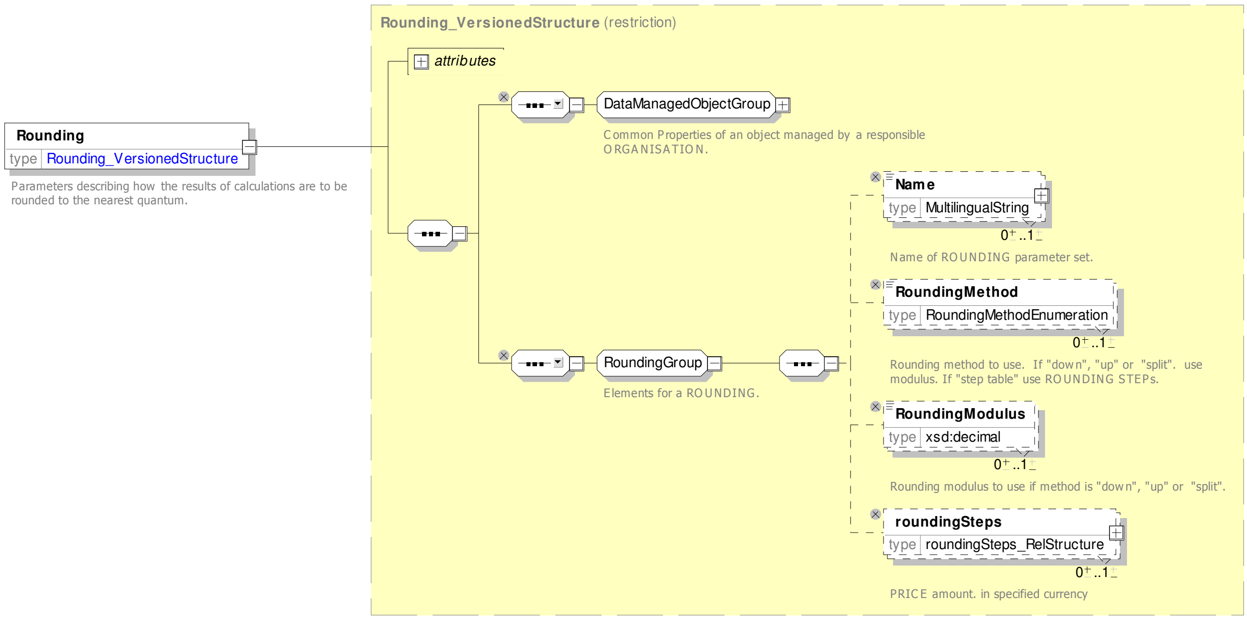


Figure 234 — *Rounding* — XSD

8.7.1.3.5.1 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 204 – *RoundingStep* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	ROUNDING STEP inherits from VERSIONED CHILD. See NeTEx Part1.
	id	*RoundingIdType*	1:1	Identifier of ROUNDING STEP.
«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.

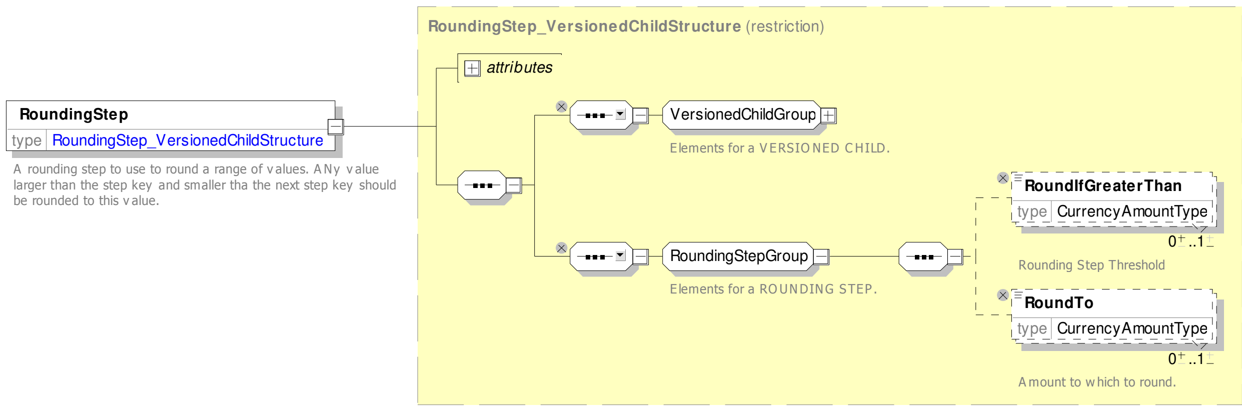


Figure 235 — *RoundingStep* — XSD

8.7.1.3.6 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 205 – *FareDayType* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DayType*	::>	FARE DAY TYPE inherits from DAY TYPE.
	id	*FareDayTypeIdentifier*	1:1	Identifier of FARE DAY TYPE.

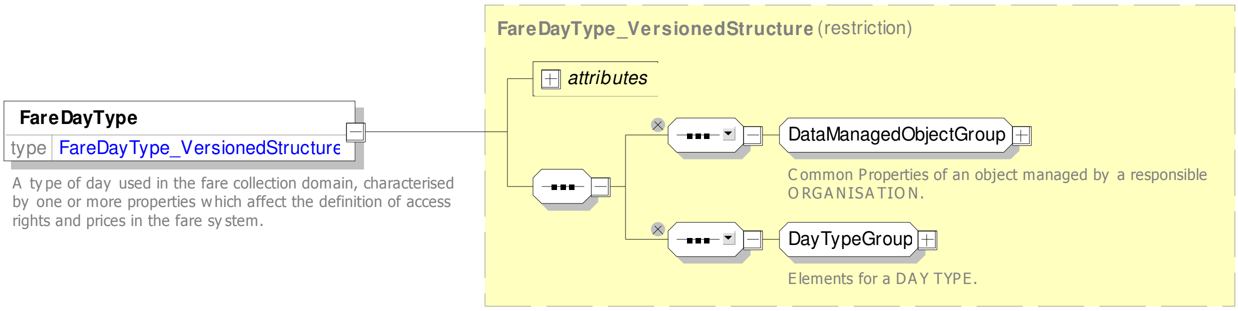


Figure 236 — *FareDayType* — XSD

8.7.1.3.7 PricingService – Model Element

A web service used to provide prices dynamically at time of booking or purchase.

Table 206 – *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.

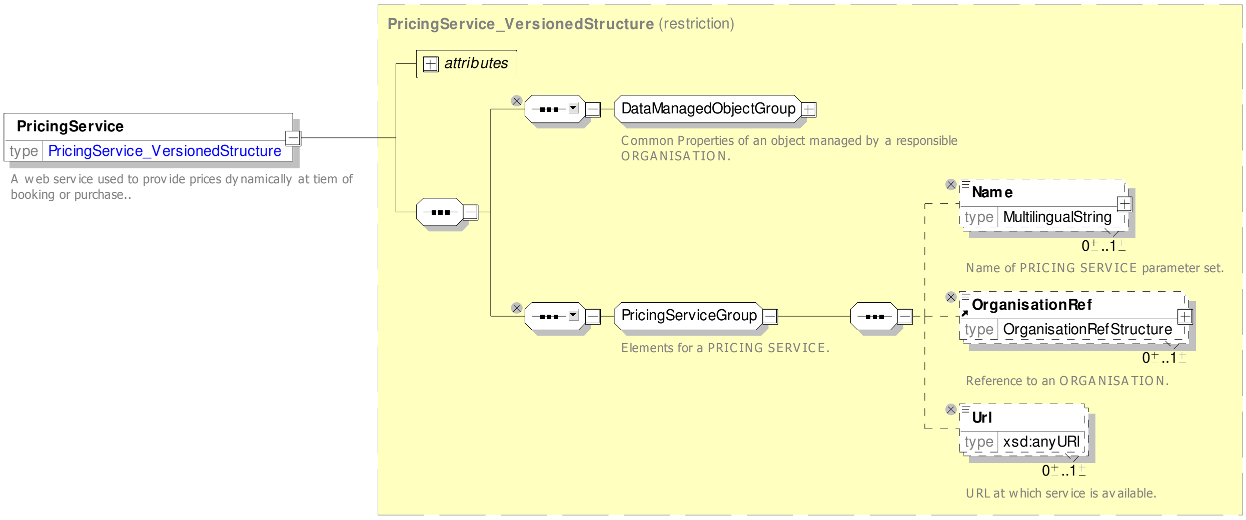


Figure 237 — *PricingService* — XSD

8.7.1.3.7.1 RoundingMethod – Allowed values

The following table shows the allowed values for ***RoundingMethod** (RoundingMethodEnum)*.

Table 207 – RoundingMethodEnum – Allowed values

Value	Description
none	No rounding.
down	Round down using rounding modulus.
up	Round up using rounding modulus.
split	Round down if below .5 up if above 0.5.
stepTable	Use a rounding table.

8.7.1.4 Fare Calculation Parameters – XML examples

8.7.1.4.1 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:

```
<!--- ====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>
```

8.7.1.4.2 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

```
<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>
```

8.7.2 Fare Price

8.7.2.1 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.

The following price entities are defined:

— 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 PACKAGE PRICE,
— FULFILMENT METHOD PRICE.

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.

[TO DO UPDATE DIAGRAM – NEED REFLEXIVE REL on FARE PRICE]

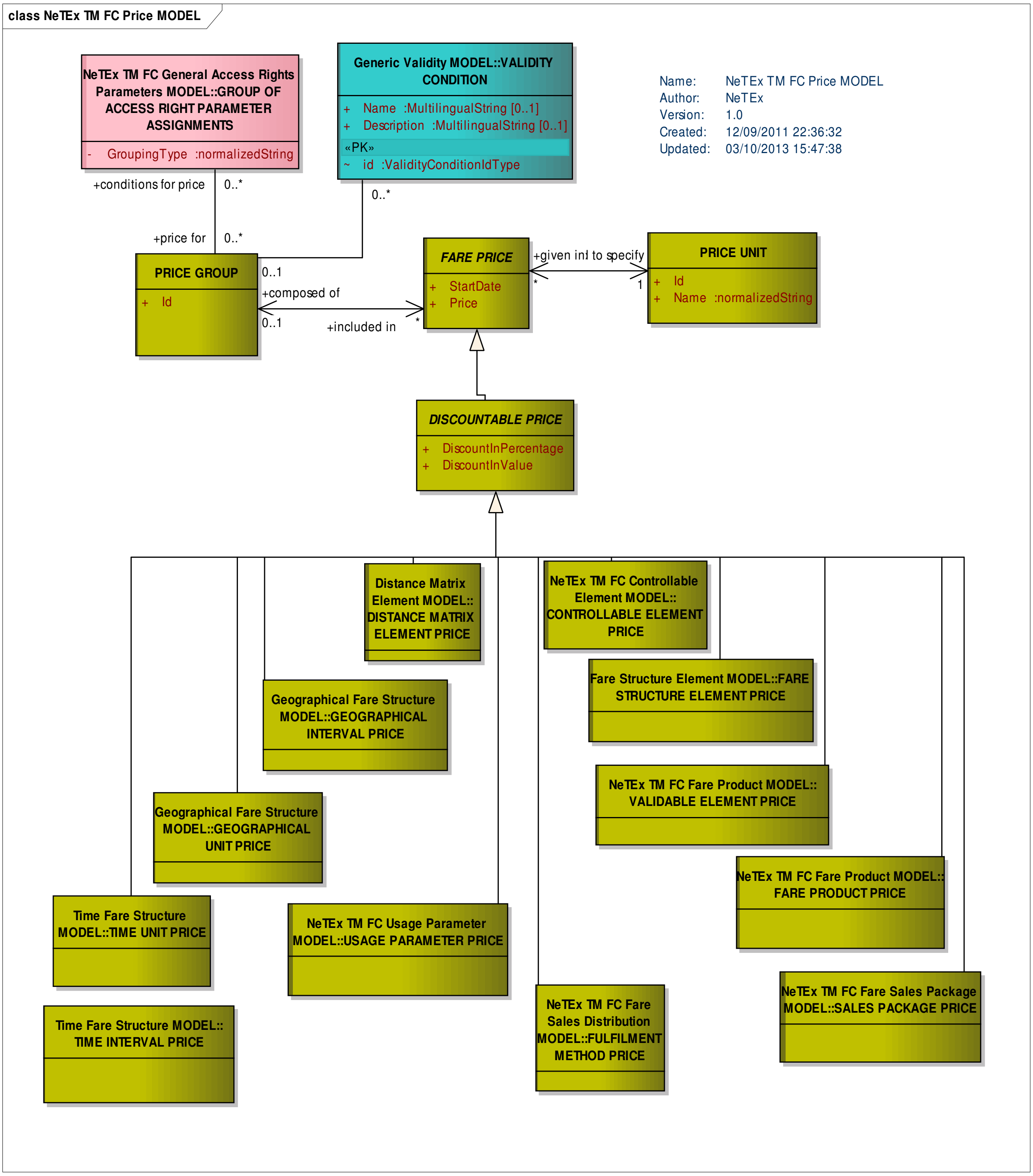


Figure 238 —Price – Conceptual Model

8.7.2.2 Fare Price – Physical model

The following figure shows the overall physical model for FARE PRICEs.

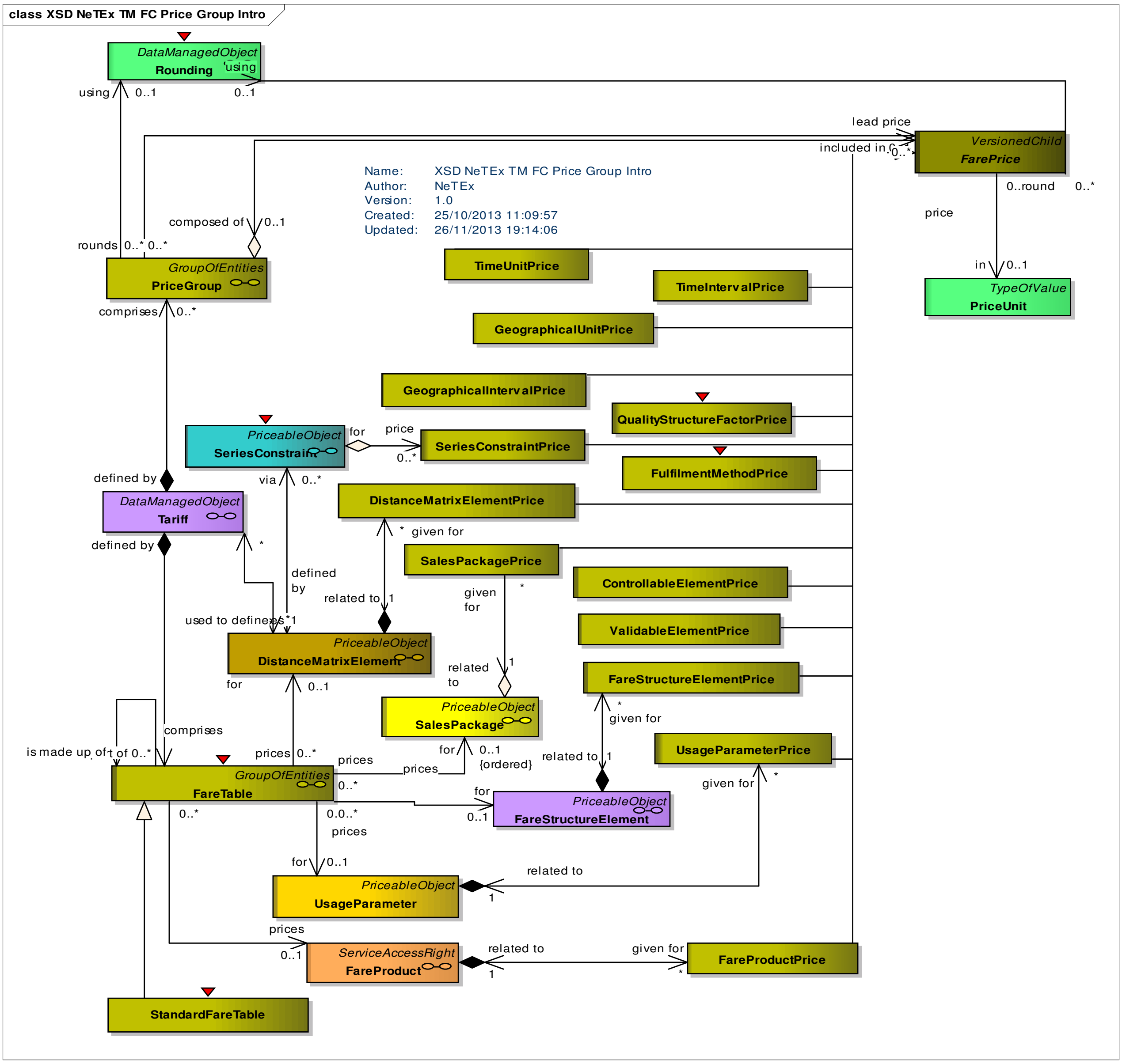


Figure 239 —Price Group Overview– Physical Model

8.7.2.3 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.

Fundamental prices are defined as REFERENCE PRICEs, for example TIME INTERVAL PRICE, GEOGRAPHICAL INTERVAL PRICE, ISTANCE MATRIX ELEMENT PRICE, etc

Discount and limit details are added in DISCOUNTED PRICE which is used to define prices which may also be derived from other prices.

.

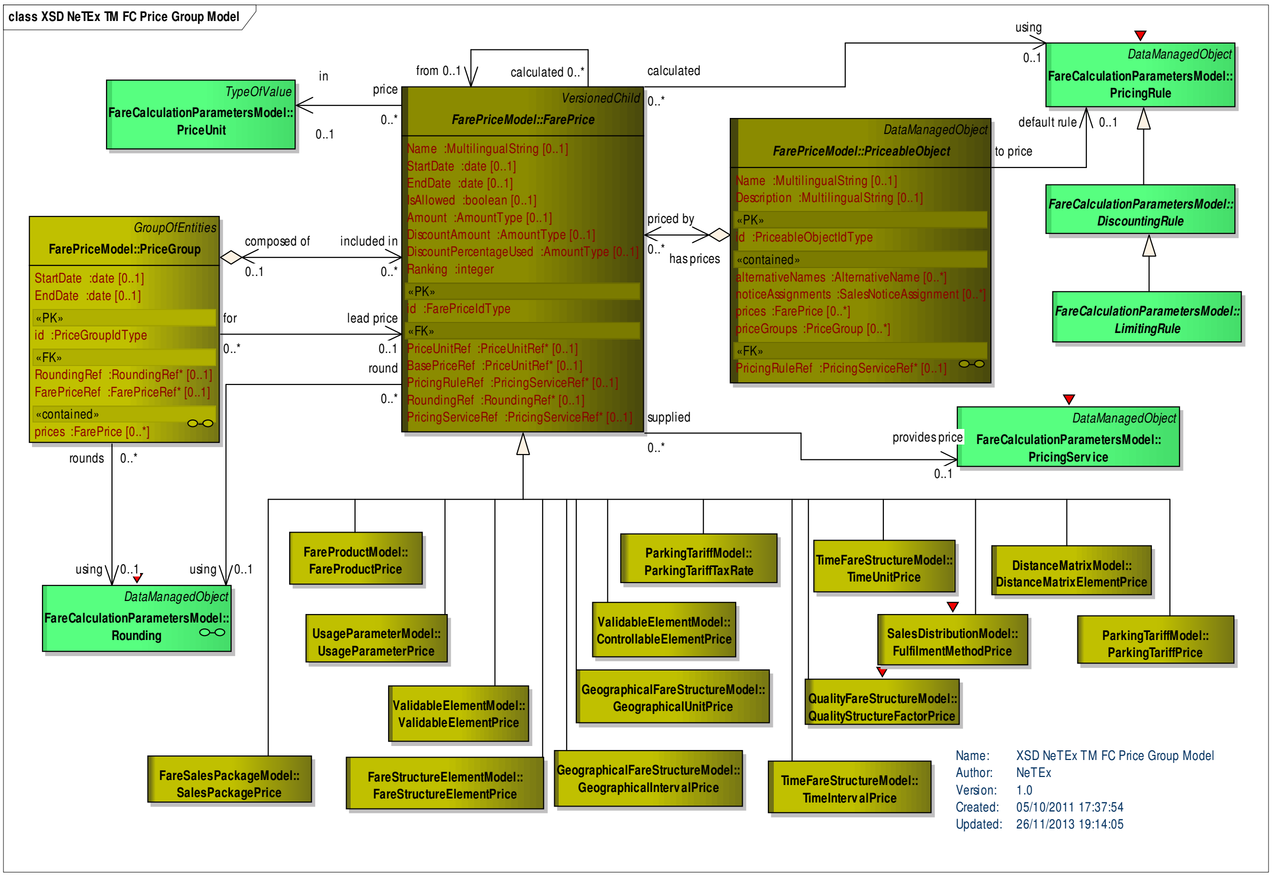


Figure 240 —Price Group – Physical Model

8.7.2.4 Fare Price – Attributes and XSD

8.7.2.4.1 PriceUnit – Model Element

A unit to express prices: amount of currency, abstract fare unit, ticket unit or token etc.

Table 208 – *PriceUnit* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	PRICE UNIT inherits from TYPE OF VALUE.
«PK»	***id***	*PriceUnitIdType*	1:1	Identifier of PRICE UNIT.
	Name	*MultilingualString*	0:1	Name of PRICE UNIT.

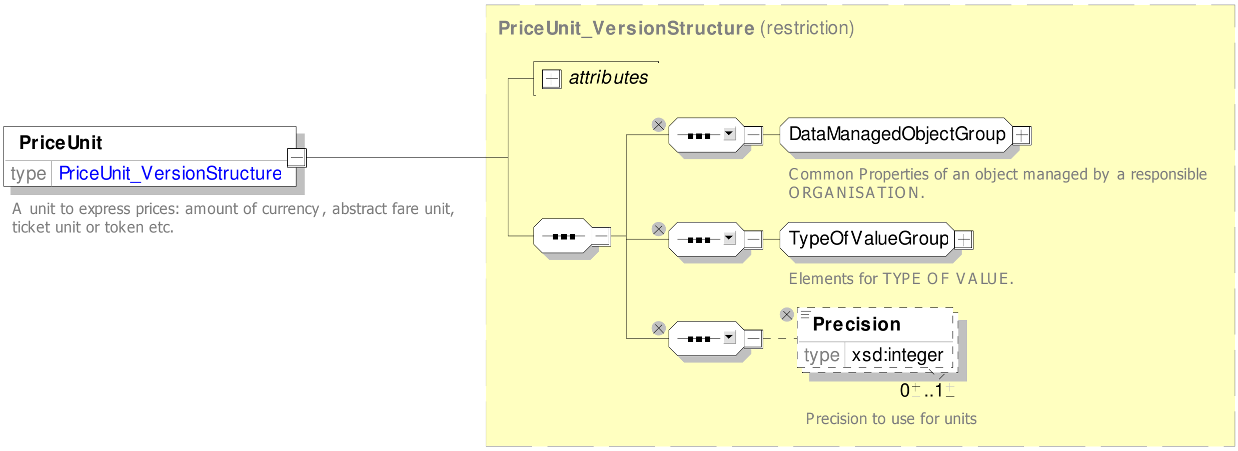


Figure 241 — *PriceUnit* — XSD

8.7.2.4.2 FarePrice – Model Element

A set of all possible price features for an Fare element.

Table 209 – *FarePrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	FARE PRICE inherits from VERSIONED CHILD
«PK»	***id***	*FarePriceIdType*	1:1	Identifier of FARE PRICE.
	Name	*MultilingualString*	0:1	Name of PRICE.
	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 $217 code (This in an optimization to allow PRICE UNITs to be omitted).
	Units	*xsd:decimal*	0:1	Amount in designated unit.
«FK»	***PriceUnitRef***	*PriceUnitRef*	0:1	Reference to a PRICE UNIT; may be a currency.
	IsAllowed	*xsd:boolean*	0:1	Whether the FARE PRICE is allowed. Default is true.
«FK»	***PricingService-Ref***	*PricingServiceRef*	0:1	Reference to a PRICE SERVICE which can provide / provided price.
[GROUP]	***FarePrice-Calculation-Group***	*FarePriceCalculation-Group*	0:1	Elements governing the calculation of prices.
	Ranking	*xsd:integer*	0:1	Relative ranking of price relative to other prices .

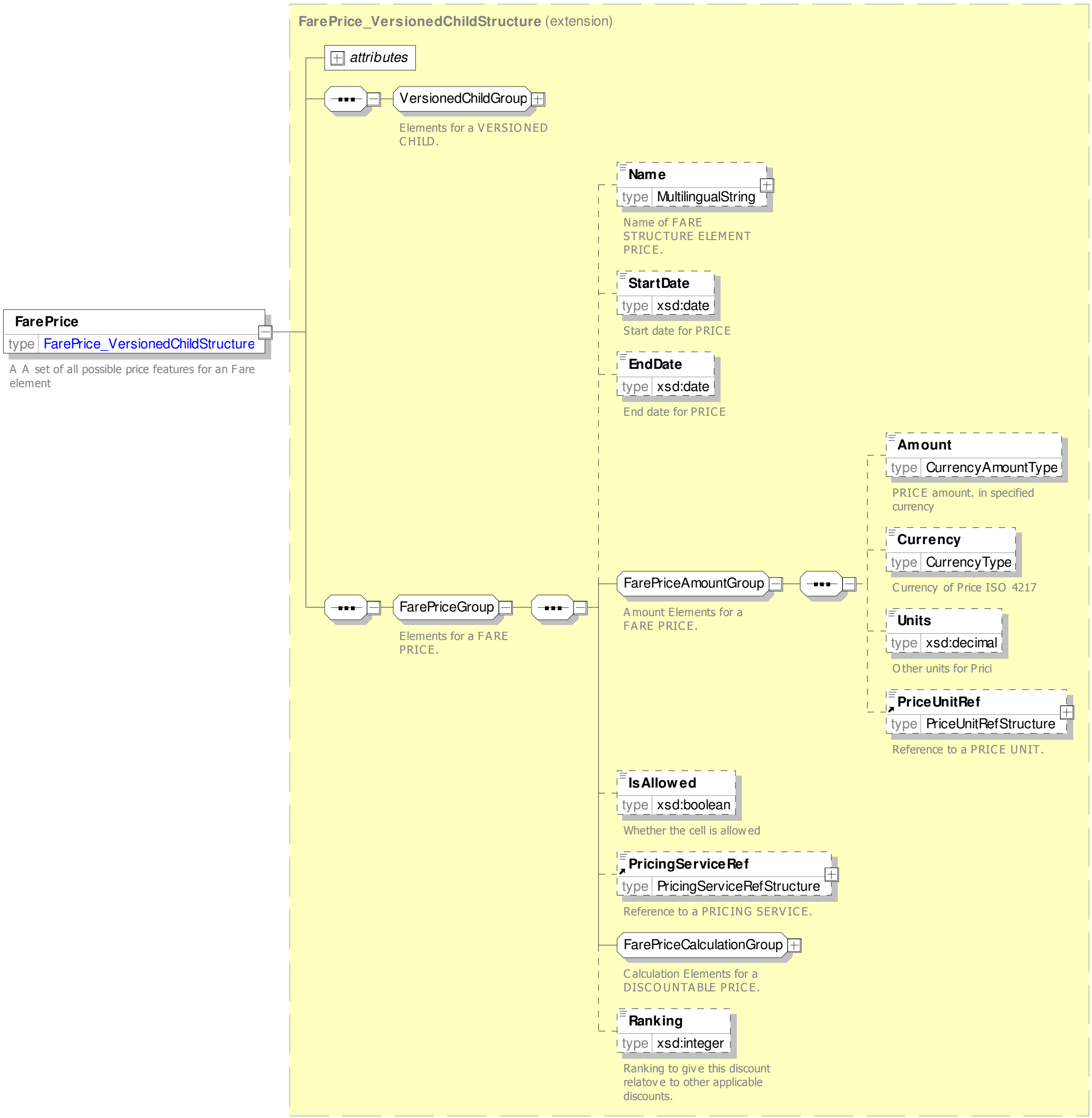


Figure 242 — *FarePrice* — XSD

8.7.2.4.2.1 FarePriceCalculationGroup – Group

The ***FarePriceCalculationGroup*** specifies attributes affecting the calculation of a price from another price.

Table 210 – *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		

	PricingRuleRef	*PricingRule*	0:1	Reference to a PRICING RULE used to derive price.
	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 243 — *FarePriceCalculationGroup* — XSD

8.7.2.4.3 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 211 – *PriceGroup* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*GroupOfEntities*	::>	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 validity for prices in the group. This overrides any date on an individual price in the group.
	EndDate	*xsd:date*	0:1	End date for PRICE validity for prices in the group. This overrides any date on an individual price in the group.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Rounding to use on calculations with prices in the group.
“cntd»	***members***	*FarePrice*	0:*	FARE PRICEs in PRICE GROUP.
«FK»	***FarePriceRef***	*FarePriceRef*	0:1	Lead FARE PRICE to follow if other prices in group follow, i.e. if the PRICE GROUP is a price band. All the other prices in the group will be derived from it
“cntd»	***validity-Conditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs applying to PRICE GROUP.

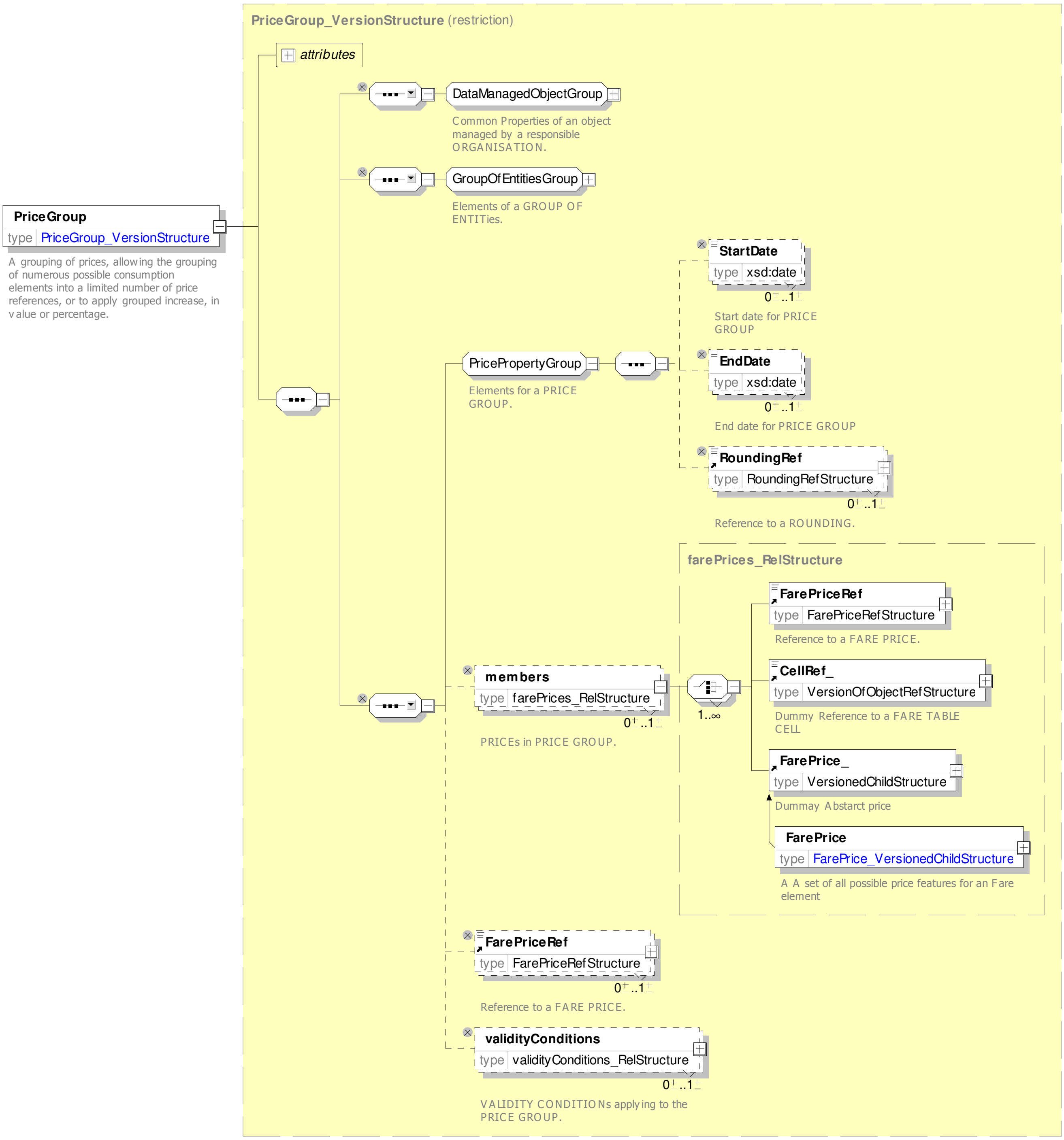


Figure 244 — *PriceGroup* — XSD

8.7.2.4.4 PriceableObject – Model Element

An element which may have a FARE PRICE.

Table 212 – *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.
“cntd»	***prices***	*FarePrice*	0:*	FARE PRICEs in PRICE GROUP.
	Name	*MultilingualString*	0:1	Name of FARE PRODUCT.
	Description	*MultilingualString*	0:1	Description of FARE PRODUCT.
«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»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for element.
“cntd»	***notice-Assignments***	*NoticeAssignment*	0:*	NOTICE ASSIGNMENTs associated with the element.

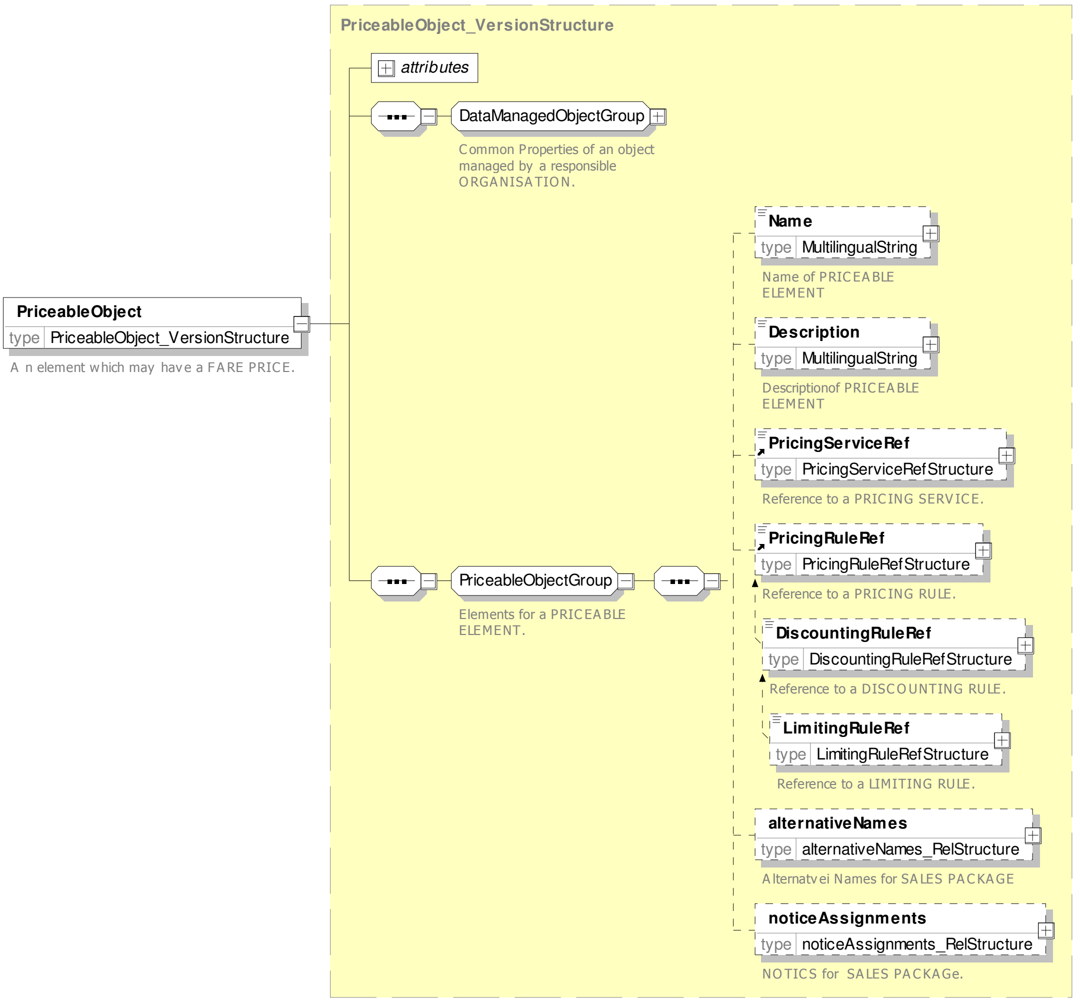


Figure 245 — *PriceableObject* — XSD

8.7.2.5 Fare Price – XML examples

8.7.2.5.1 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>
```

8.7.2.5.2 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@studentRailCard">
        <Amount>15.00</Amount>
        <UsageParameterPriceRef version="any" ref="nr:: studentRailCard@25pct"/>
        <SaleDiscountRightRef version="any" ref="nr::trip@StudentRailCard"/>
</FareProductPrice>


<UsageParameterPrice version="any" id="nr:studentRailCard@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 >
```

8.7.3 Fare Table

8.7.3.1 Fare Table – Conceptual model

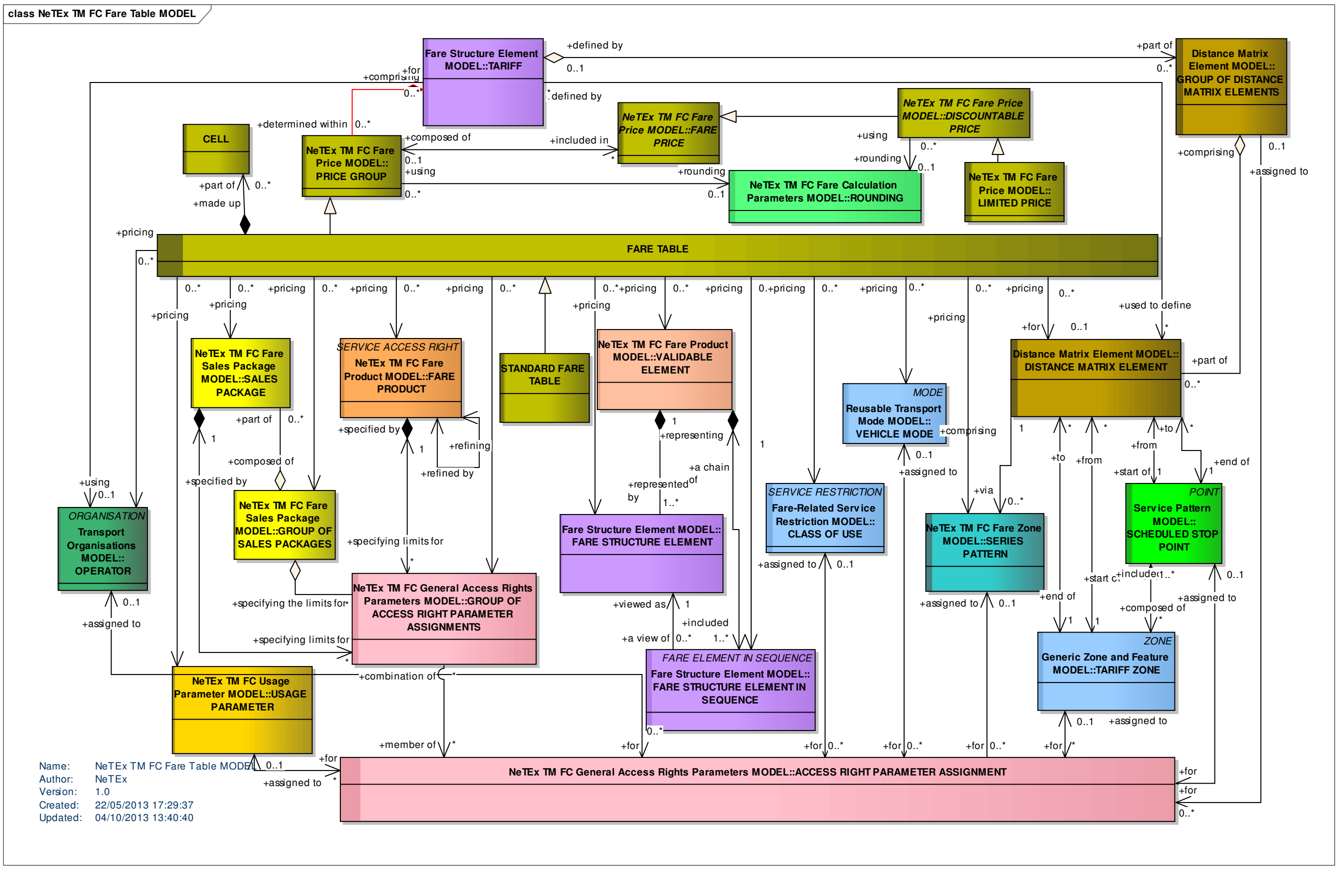


Figure 246 — Fare Table – Conceptual Model

8.7.3.2 Fare Table – Conceptual Examples

8.7.3.2.1 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 PRDUCT / 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		Double		Single		Double		T4				Sièges	
			Grande Classe		Grande Classe (5)		Affaires		Affaires (5)		Touriste				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		Double		Single		Double		T4				Sièges	
			Grande Classe		Grande Classe (5)		Affaires		Affaires (5)		Touriste				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 ».

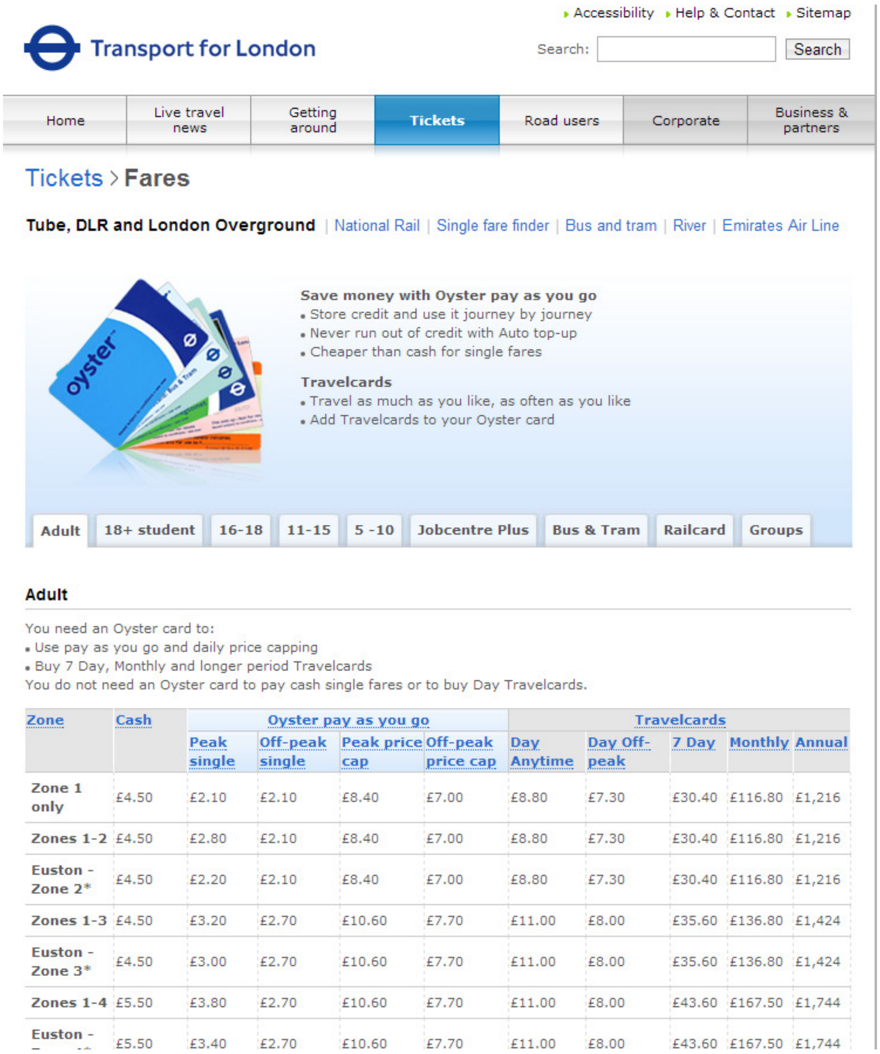
Le Trainhôtel est une offre internationale et il est interdit de descendre du train en dehors de sa gare de destination.

La réservation est ouverte à J-3 mois et jusqu'au départ du Trainhôtel. La réservation reste obligatoire.

Figure 247 — Fare Table – Example Rail Fare Table

8.7.3.2.2 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 248 — Fare Table – Example TfL Fare Table

8.7.3.3 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.

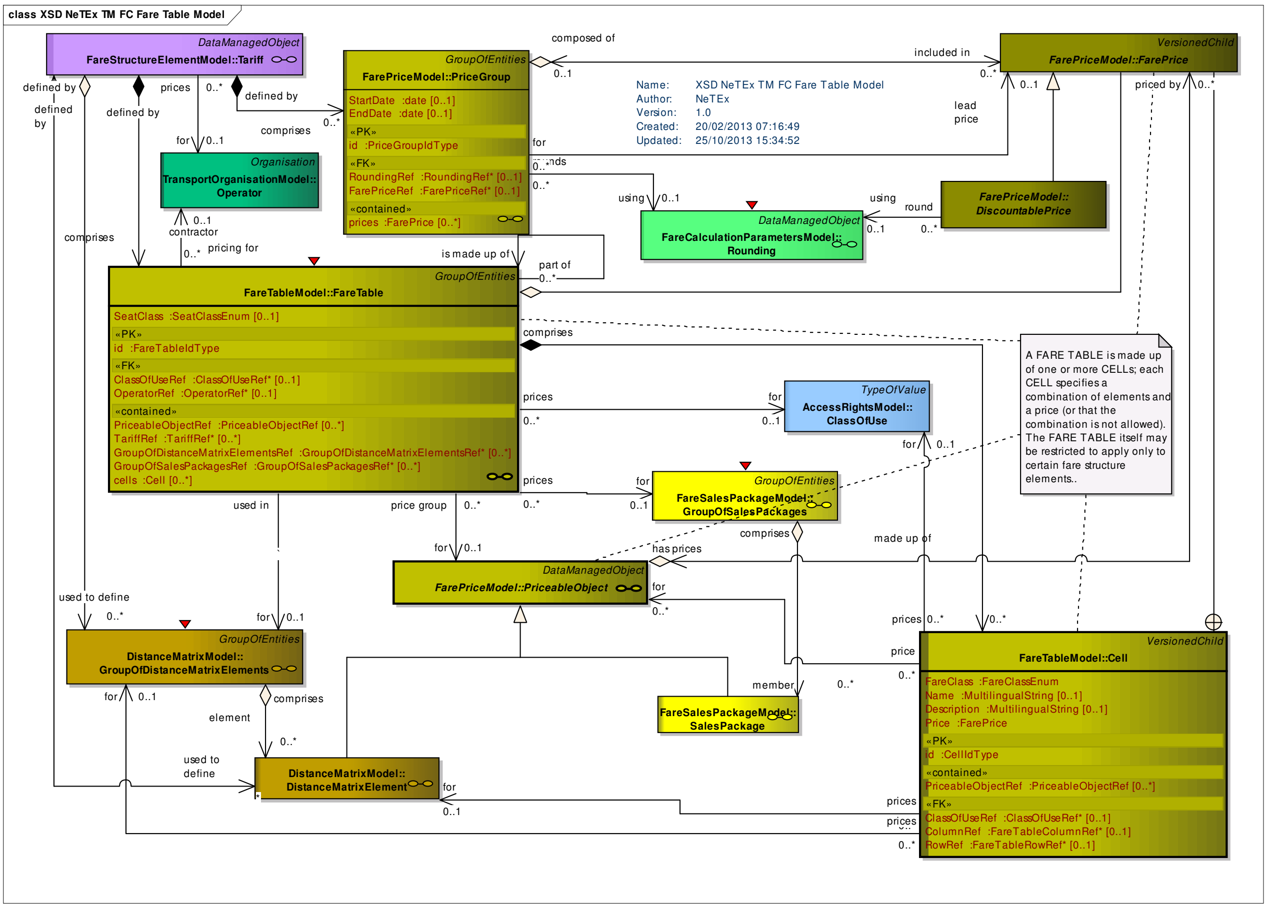


Figure 249 — Fare Table– Physical Model

8.7.3.3.1 Fare Table Cells – Physical model

The following figure shows the physical model for FARE TABLE CELLs. Each cell may reference one or more PRICEABLE OBJECTs as well as a CLASS OF USE. Cells may also be assigned to ROWs and Columns for presentation purposes.

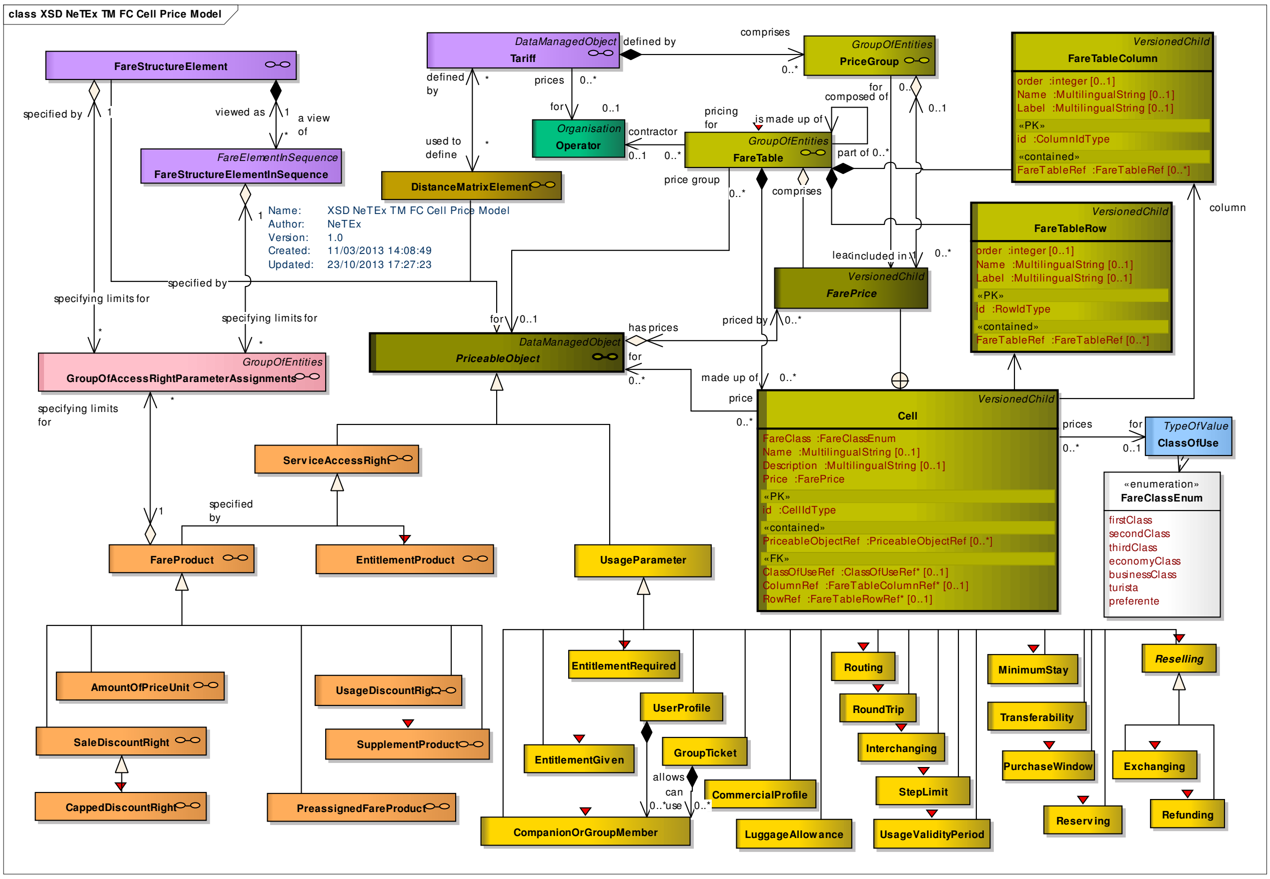


Figure 250 — Cell Price Model – Physical Model

8.7.3.3.2 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.

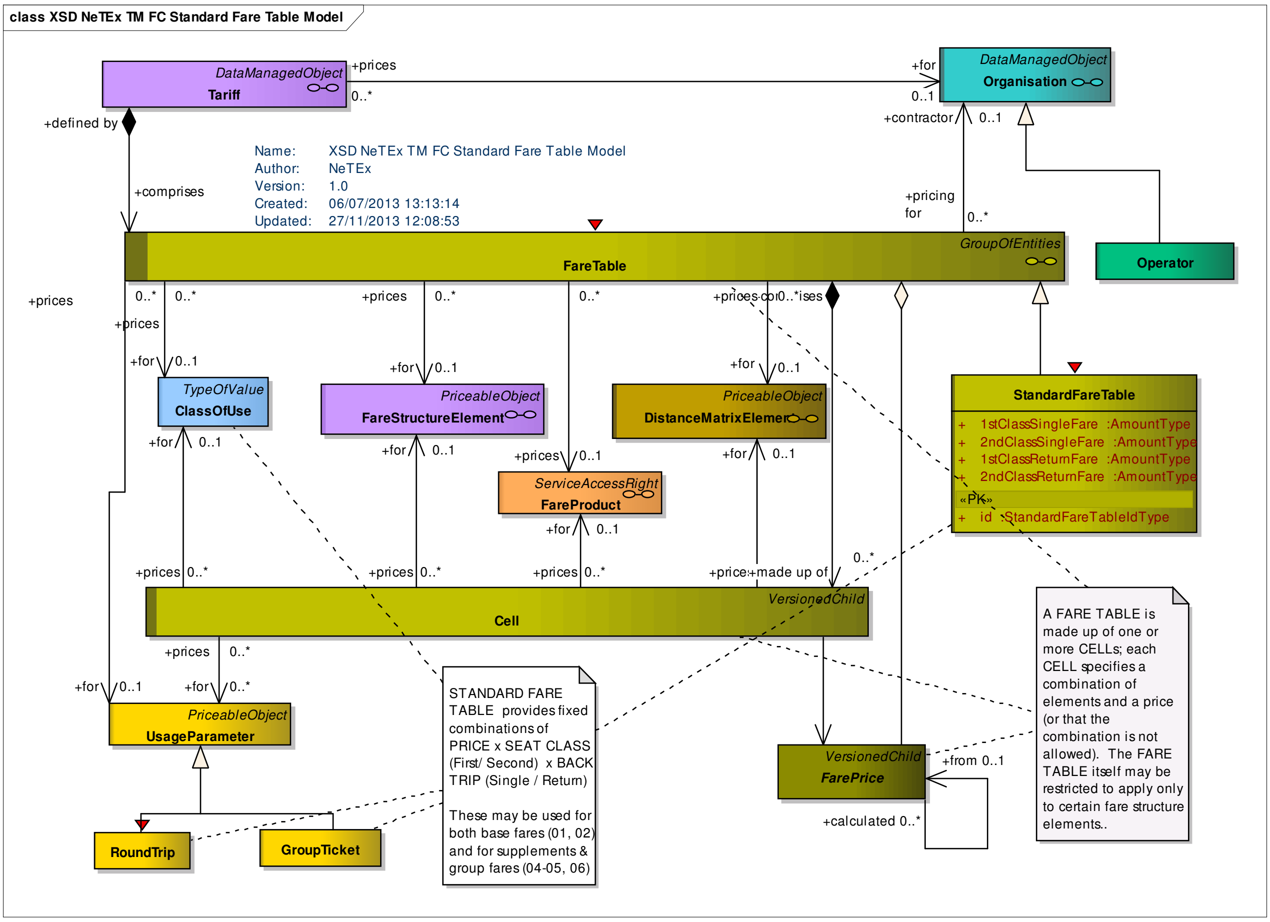


Figure 251 — Standard Fare Table – Physical Model

8.7.3.4 Fare Table – Attributes and XSD

8.7.3.4.1 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 INTERAVL, USAGE PARAMETER, etc.

Table 213 – *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.
	FareClass	*FareClassEnum*	0:1	CLASS OF USE associated with an individual CELL or FARE TABLE. See allowed values

«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	CLASS OF USE associated with an individual CELL or FARE TABLE.
GROUP	***FareTable-References-Group***	*FareTable-ReferencesGroup*	0:*	Fare structure elements which may be given a price and so associated with this CELL.
GROUP	***FareTable-HeadingsGroup***	*FareTable-HeadingsGroup*	0:*	Row and column headings for the table. CELLs may reference these. See below
"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.

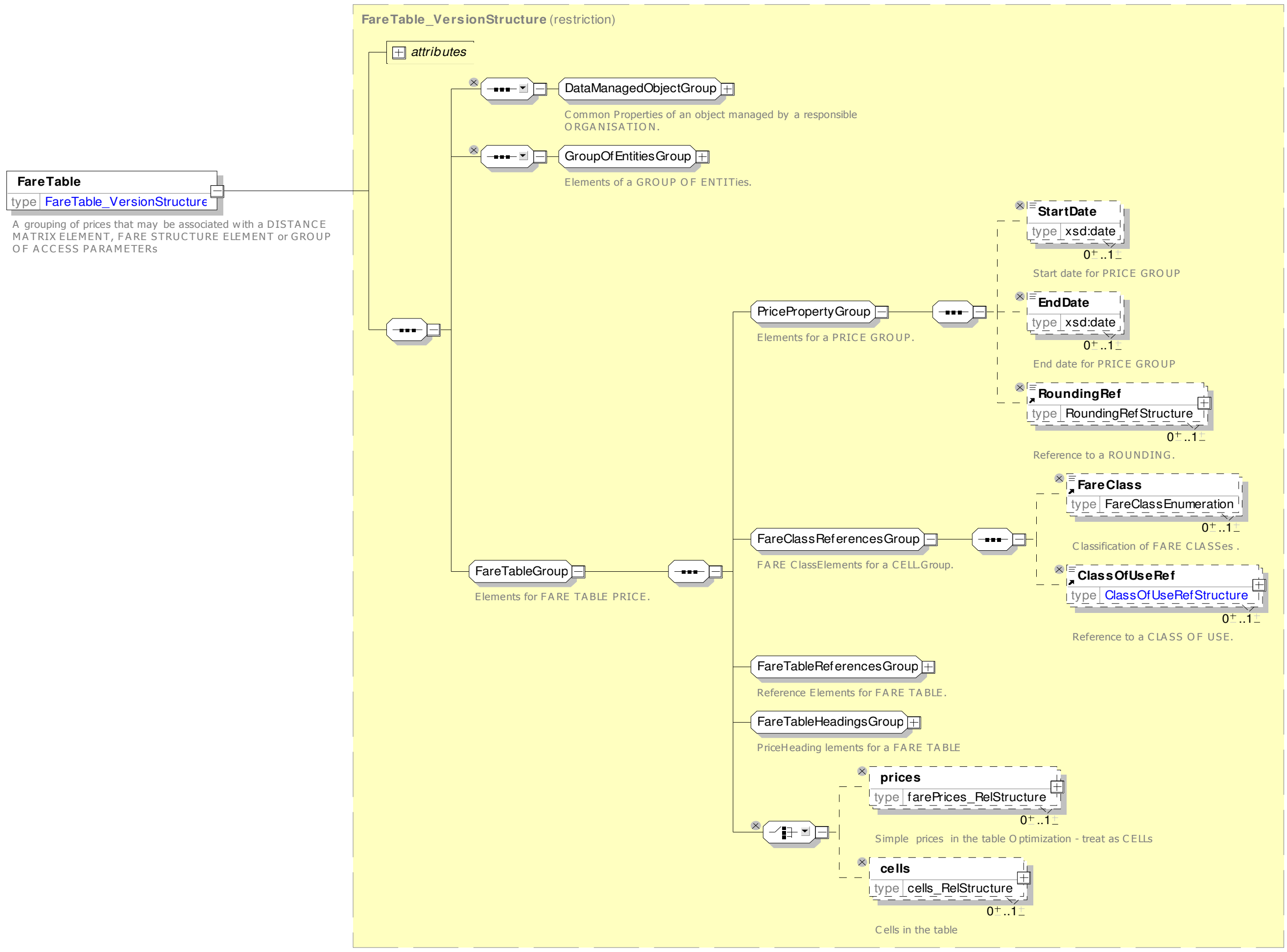


Figure 252 — *FareTable* — XSD

8.7.3.4.1.1 FareTableReferencesGroup – Group

The ***FareTableReferencesGroup*** specifies the associated fare structure elements for which prices are provided by the FARE TABLE.

Table 214 – *FareTableReferencesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
"cntd»	***Priceable-ObjectRef***	*PriceableObjectRef*	0:*	PRICEABLE OBJECT elements which may be given a price and so associated with this CELL.
«FK»	***Transport-OrganisationRef***	*Transport-OrganisationRef*	0:1	OPERATOR or AUTHORITY to which FARE PRICEs apply.
"cntd»	***usedIn***	*Choice*	0:1	A fare element associated
«FK»	***TariffRef***	*TariffRef*	1:*	TARIFF to which PRICEs of FARE TABLE apply.
«FK»	***GroupOf-DistanceMatrix-ElementsRef***	*GroupOfDistance-MatrixElementsRef*	1:*	GROUP OF DISTANCE MATRIX ELEMENTs associated with a FARE TABLE.
«FK»	***GroupOfSales-PackagesRef***	*GroupOfSales-PackagesRef*	1:*	GROUP OF SALES PACKAGEs associated with a FARE TABLE.

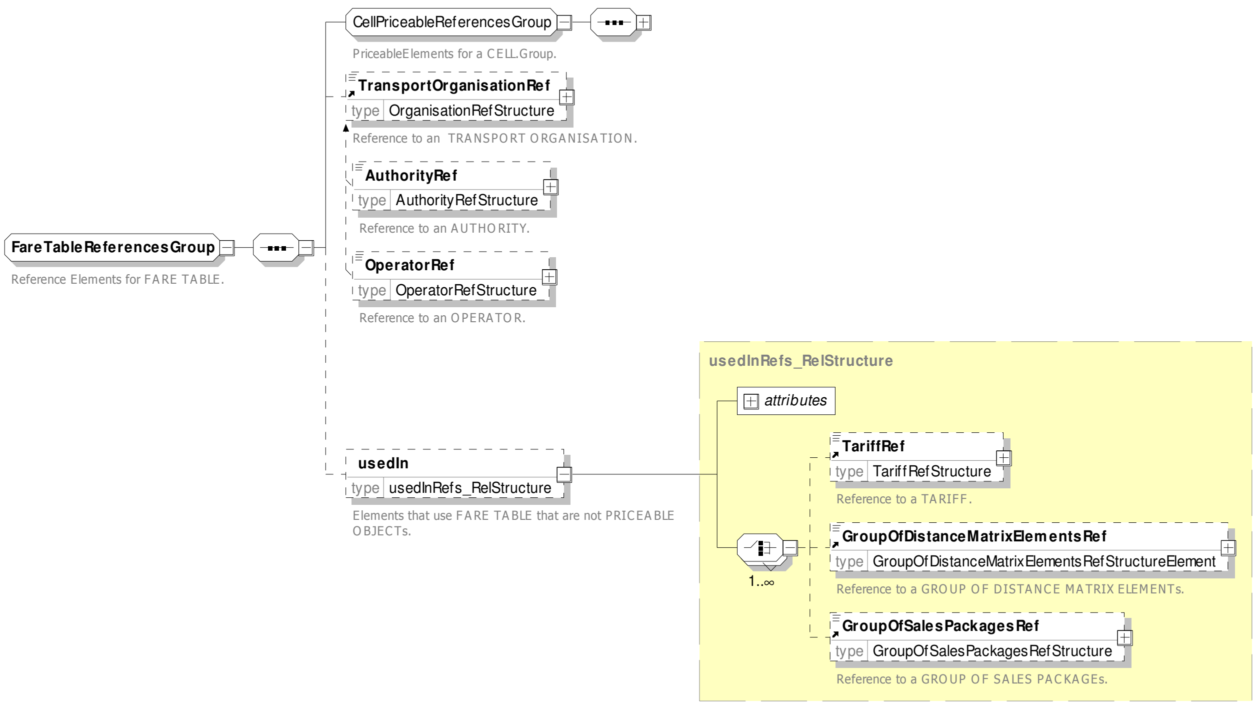


Figure 253 — *FareTableReferencesGroup* — XSD

8.7.3.4.1.2 FareTableHeadingsGroup – Group

The ***FareTableHeadingsGroup*** specifies any row and column headings which may be used to present the FARE TABLE.

Table 215 – *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.

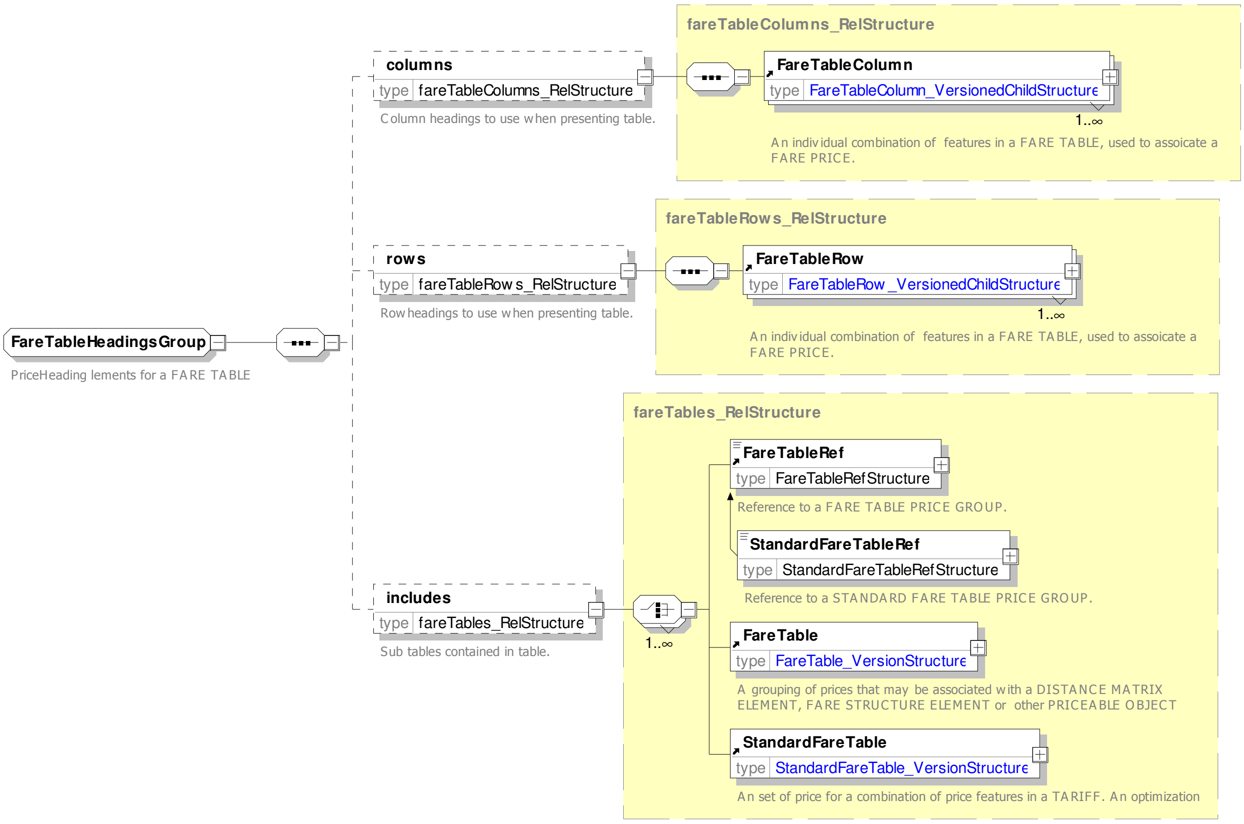


Figure 254 — *FareTableHeadingsGroup* — XSD

8.7.3.4.2 Cell – Model Element

An unique individual combination of features within a FARE TABLE, used to associate a FARE PRICE with a fare element.

Table 216 – *Cell* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	CELL inherits from VERSIONED CHILD
	FareClass	*FareClassEnum*	1:1	FARE CLASS associated with an individual CELL or FARE TABLE.
«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
a	***Price***	*Price*	1:1	Price in CELL.
b	***FarePriceRef***	*FarePriceRef*	1:1	Reference to another FARE PRICE providing price for CELL
c	***PriceGroupRef***	*PriceGroupRef*	1:1	Reference to a FARE GROUP providing price for CELL via its lead price.
	CellReferences-Group	*CellReferencesGroup*	0:1	Fare Structure which may be given a price and so be associated with this CELL.
	CellHeadings-Group	*CellHeadingsGroup*		Table headings associated with this CELL.

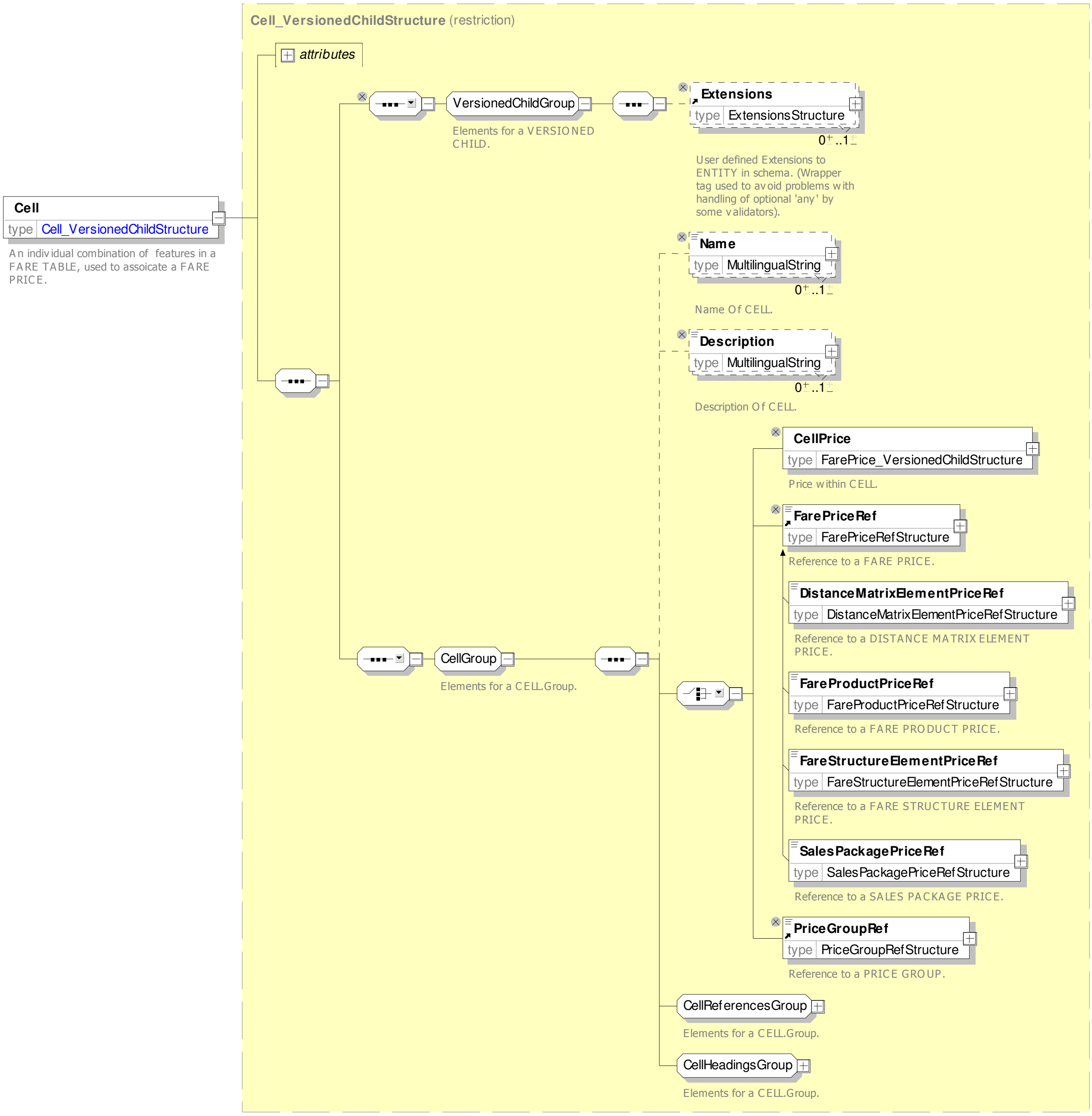


Figure 255 — *Cell* — XSD

8.7.3.4.2.1 CellReferencesGroup – Group

References to the elements for which the CELL provides a price. May be a combination of multiple elements.

Table 217 – *CellReferencesGroup* – Group

Classifi-cation	Name	Type	Cardinality	Description
	FareClass	*FareClassEnum*	1:1	FARE CLASS associated with an individual CELL or FARE TABLE.

«FK»	***ClassOfUseRef***	*ClassOfUseRef*	0:1	Reference to a CLASS OF USE (Seat Class) associated with an individual CELL or FARE TABLE.
“cntd»	***Priceable-ObjectRef***	*PriceableObjectRef*	0:*	Fare structure elements which may be given a price and so associated with this CELL.

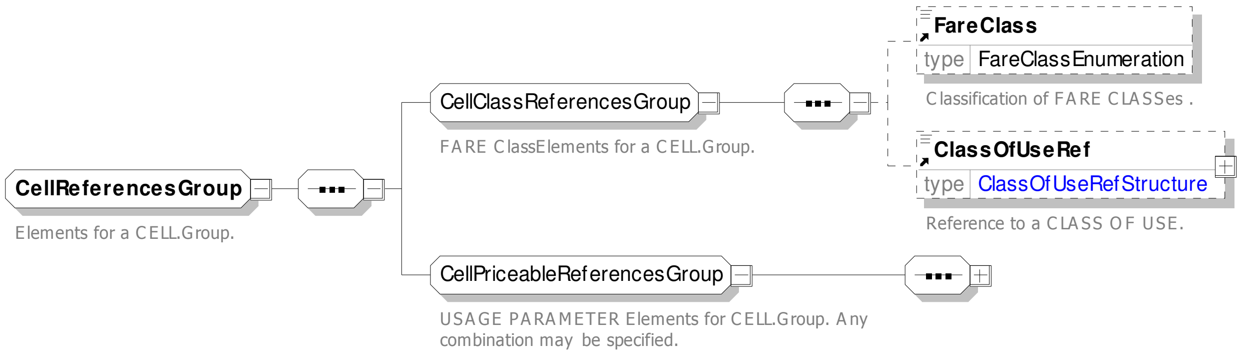


Figure 256 — *CellReferencesGroup* — XSD

8.7.3.4.2.2 CellHeadingsGroup – Group

References to the row and column of the faretable associated with CELL.

Table 218 – *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.

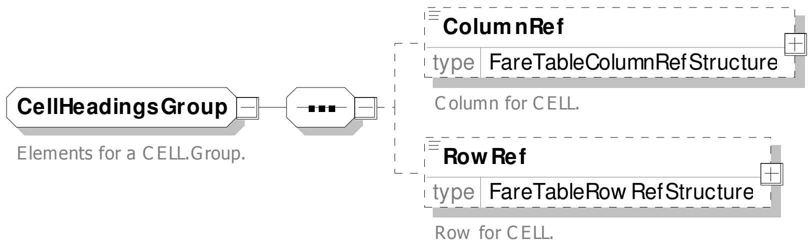


Figure 257 — *CellHeadingsGroup* — XSD

8.7.3.4.3 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 219 – *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.

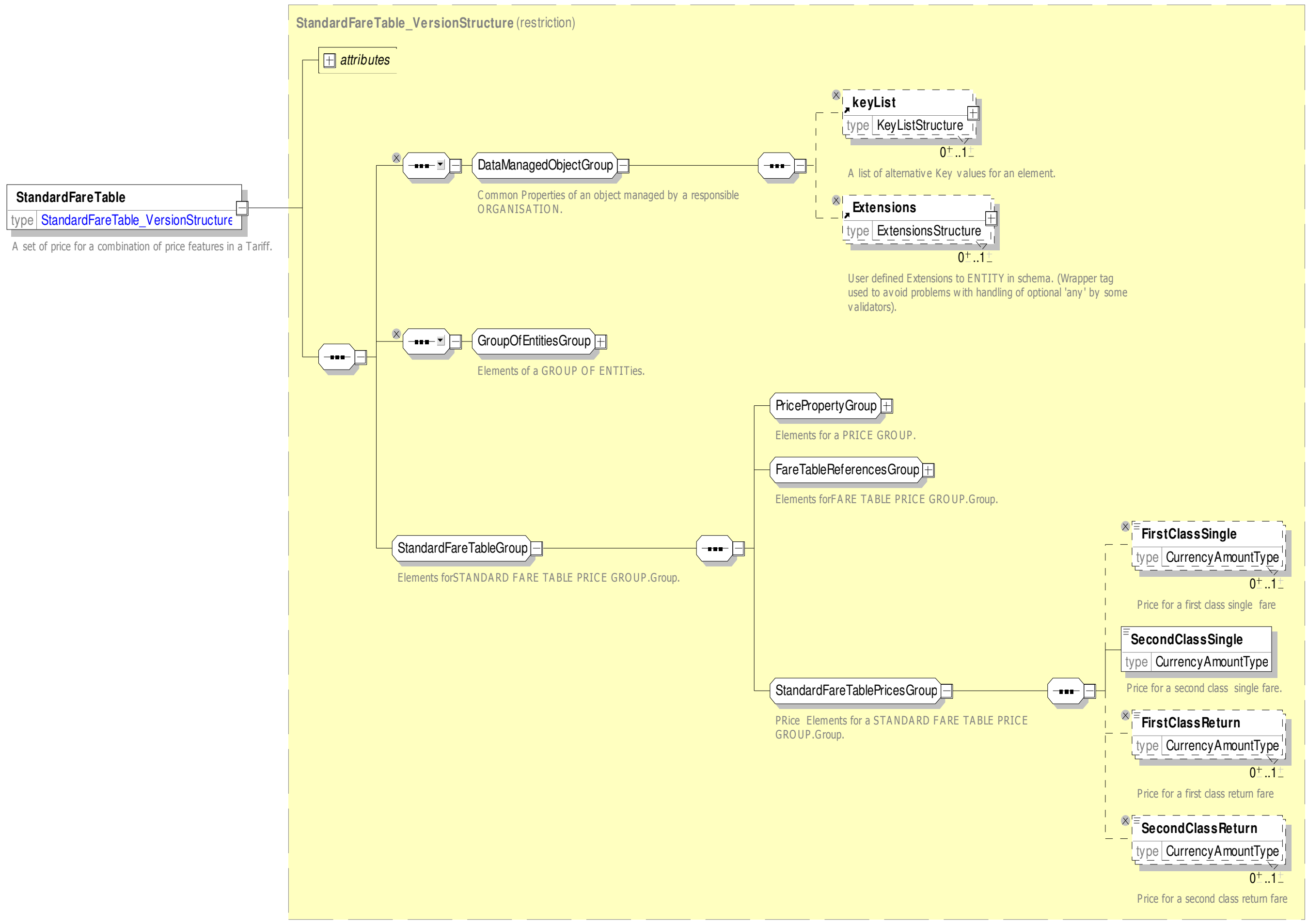


Figure 258 — *StandardFareTable* — XSD

8.7.3.5 Fare Table – XML examples

8.7.3.5.1 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).

```
<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::busOrTramTrip"/>
                <PreassignedFareProductRef version="any" ref="tfl::PrepaidFare"/>
            </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::PayAsYouGoFare"/>
            </Cell>
```

```
            <Cell version="any" id="lbsl::Bus_Tram@Oyster@Week@18Plus">
                <Price>
                    <Name>weekly TravelCardOnOyster 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::TravelCardOnOyster"/>
            </Cell>
            <Cell version="any" id="lbsl::Bus_Tram@Oyster@Monthly@18Plus">
                <Price>
                    <Name>Monthly TravelCardOnOyster 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::TravelCardOnOyster"/>
            </Cell>
        </cells>
    </FareTable>
```

8.7.3.5.2 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.

```
    <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">
                <UsageParameterPriceRef version="any" ref="tfl::concession@free"/>
                <ValidableElementRef version="any" ref="lbsl:ValidableElement:busOrTramTrip"/>
            </Cell>
            <Cell version="any" id="tfl::FreedomPass@Londoner@metro">
                <UsageParameterPriceRef version="any" ref="tfl::concession@free"/>
                <ValidableElementRef version="any" ref="lul:ValidableElement:metroTrip"/>
            </Cell>
            <Cell version="any" id="tfl::FreedomPass@Londoner@cableway">
                <UsageParameterPriceRef version="any" ref="tfl::concession@34pct"/>
                <ValidableElementRef version="any" ref="ea:ValidableElement:cablewayTrip"/>
            </Cell>
            <Cell version="any" id="tfl::FreedomPass@Londoner@water">
                <UsageParameterPriceRef version="any" ref="tfl::concession@34pct"/>
                <ValidableElementRef version="any" ref="lrs:ValidableElement:riverTrip"/>
            </Cell>
            <Cell version="any" id="tfl::FreedomPass@Londoner@overground">
                <UsageParameterPriceRef version="any" ref="tfl::concession@free"/>
                <ValidableElementRef version="any" ref="nr:ValidableElement:suburbanRailTrip"/>
            </Cell>
            <Cell version="any" id="tfl::FreedomPass@Londoner@nationalRail">
                <UsageParameterPriceRef version="any" ref="tfl::concession@34pct"/>
                <ValidableElementRef version="any" ref="nr:ValidableElement:railTrip"/>
            </Cell>
        </cells>
    </FareTable>
    <FareTable version="any" id="tfl:FareTable:FreedomPass@NonLondoner">
        <Name>Freedom Pass prices for Non Londoners</Name>
        <QualityStructureFactorRef version="any" ref="tfl::NonResident"/>
        <SaleDiscountRightRef version="any" ref="tfl::FreedomPass"/>
        <cells>
            <Cell version="any" id="tfl::FreedomPass@NonLondoner@bus">
                <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@NonLondoner@cableway">
                <UsageParameterPriceRef version="any" ref="tfl::concession@34pct"/>
                <ValidableElementRef version="any" ref="ea:ValidableElement:cablewayTrip"/>
            </Cell>
            <Cell version="any" id="tfl::FreedomPass@NonLondoner@water">
                <UsageParameterPriceRef version="any" ref="tfl::concession@34pct"/>
```

```
                <ValidableElementRef version="any" ref="lrs:ValidableElement:riverTrip"/>
            </Cell>
        </cells>
    </FareTable>
```

8.7.3.5.3 Fare Table: XML Example of Simple prices

The following partial table shows a simple fare prices used as sells for combinations of USER PROFILE, ROUN 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::PayAsYouGoFare"/>
    <prices>
        <DistanceMatrixElementPrice version="any"
id="nr::OysterRailOnly@Single@offPeak@Adult@WatfordJunction@Zone_1">
            <Name>Zone 1 to Watford Junction*</Name>
            <Amount>4.50</Amount>
            <DistanceMatrixElementRef version="any" ref="nr::WatfordJunction@Zone_1"/>
        </DistanceMatrixElementPrice>

        <DistanceMatrixElementPrice version="any"
id="nr::OysterRailOnly@Single@offPeak@Adult@WatfordJunction@Zone_2">
            <Name>Zone 2 to Watford Junction*</Name>
            <Amount>2.70</Amount>
            <DistanceMatrixElementRef version="any" ref="nr::WatfordJunction@Zone_2"/>
        </DistanceMatrixElementPrice>

        <DistanceMatrixElementPrice version="any"
id="nr::OysterRailOnly@Single@offPeak@Adult@WatfordJunction@Zone_3">
            <Name>Zone 3 to Watford Junction*</Name>
            <Amount>2.20</Amount>
            <DistanceMatrixElementRef version="any" ref="nr::WatfordJunction@Zone_3"/>
        </DistanceMatrixElementPrice>
    </prices>
```

8.8 Sales Description

8.8.1 Fare Sales Distribution

8.8.1.1 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 PACKAGEs can be restricted to specific DISTRIBUTION CHANNELs or GROUPs OF DISTRIBUTION CHANNELs using a DISTRIBUTION ASSIGNMENT (see SALES 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.

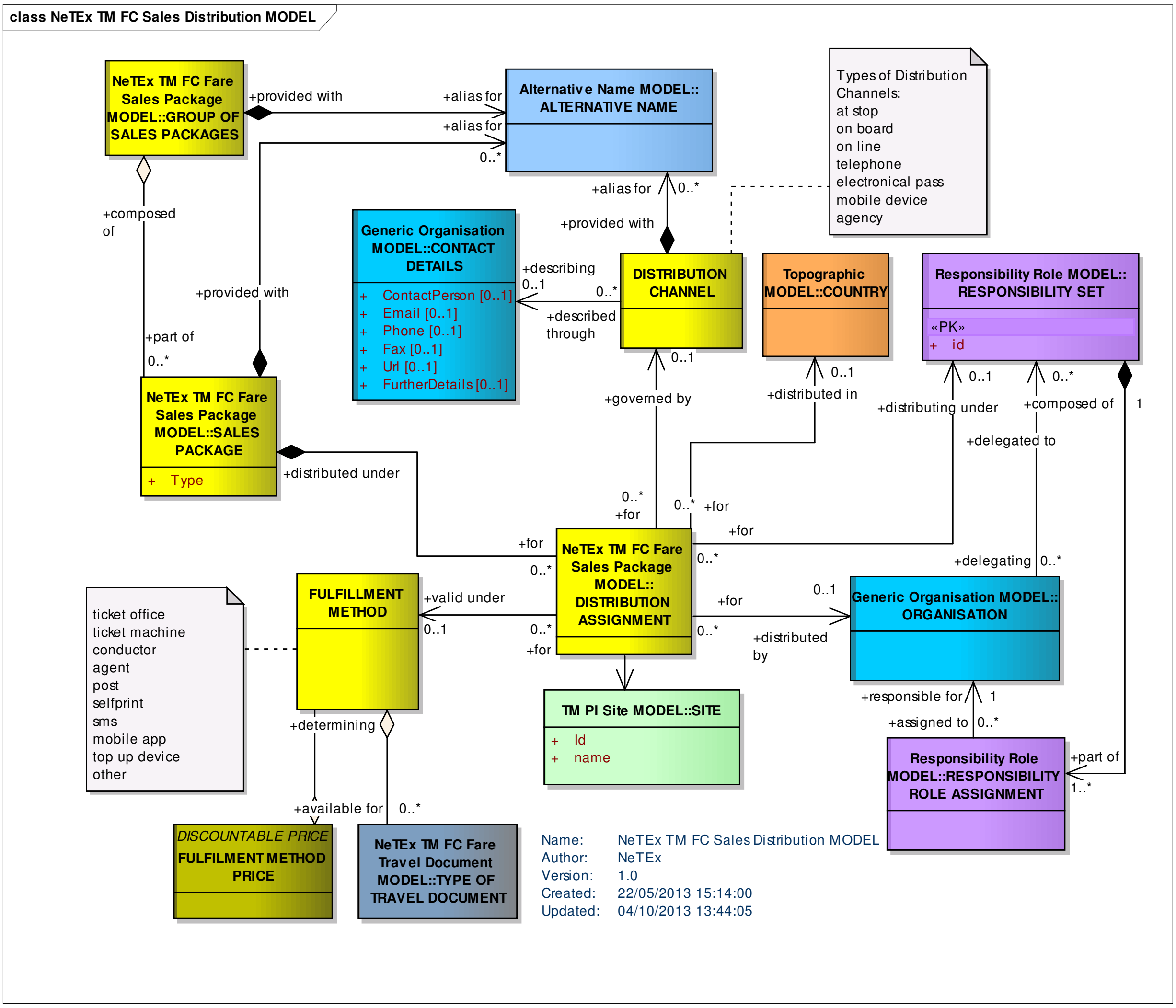


Figure 259 — Sales Distribution – Conceptual Model

8.8.1.2 Fare Sales Distribution – Physical model

The following figure shows the physical model for SALES DISTRIBUTION.

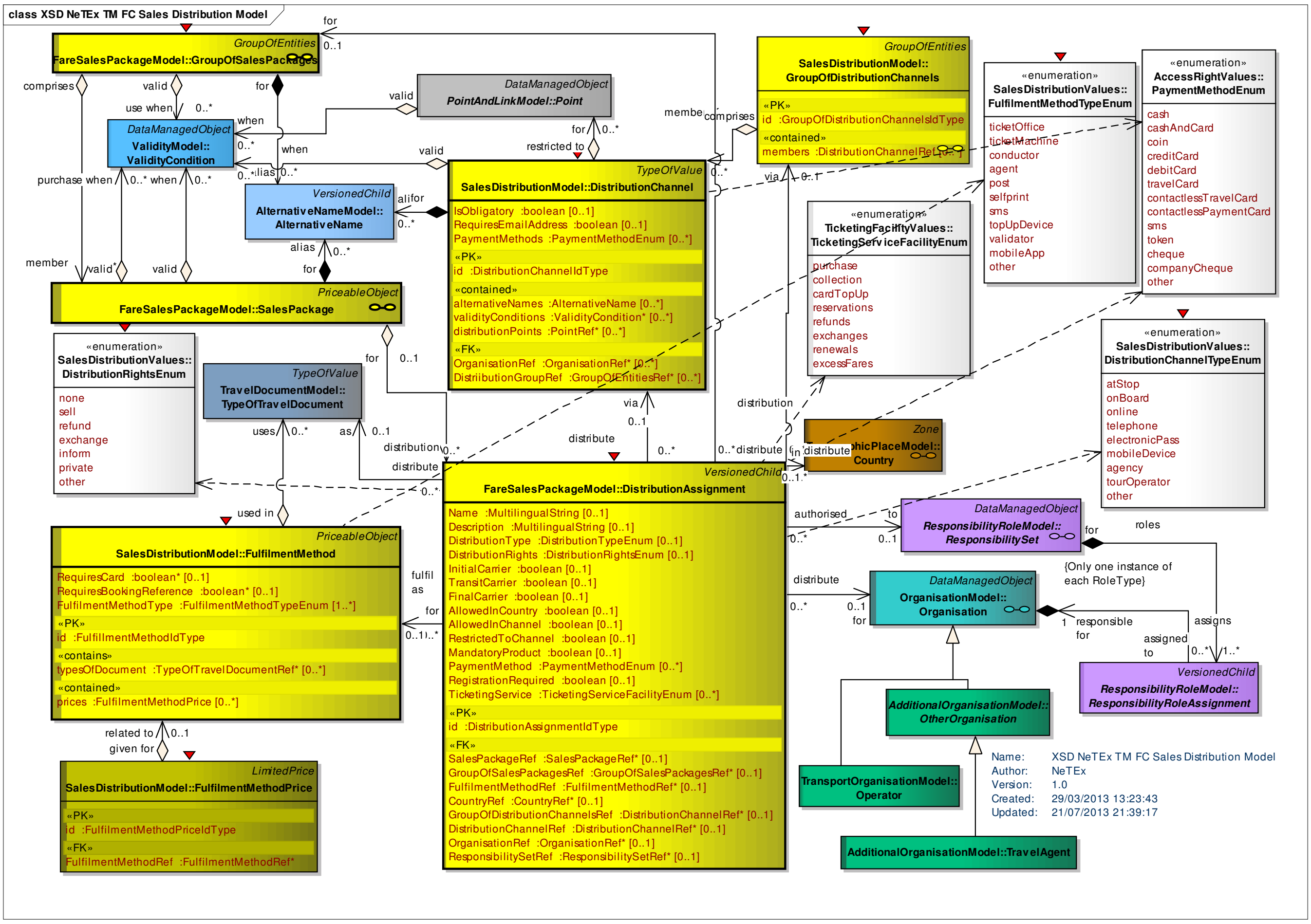


Figure 260 — Sales Distribution – Physical Model

8.8.1.3 Fare Sales Distribution – Attributes and XSD

8.8.1.3.1 DistributionChannel – Model Element

A type of outlet for selling a product.

Table 220 – *DistributionChannel* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	DISTRIBUTION CHANNEL inherits from TYPE OF VALUE. See NeTEx Part1.
«PK»	***id***	*DistributionChannel-IdType*	1:1	Identifier of a DISTRIBUTION CHANNEL.
“cntd»	***alternativeNames***	*AlternativeName*	0:*	Alternative names for DISTRIBUTION CHANNEL.
“cntd»	***validity-Conditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs specifying specific availability of DISTRIBUTION CHANNEL.
	Distribution-	*DistributionChannel-*	0:1	Type of DISTRIBUTION CHANNEL. See below

	ChannelType	*TypeEnum*		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:*	ORGANISATION associated with channel.
	PaymentMethods	*PaymentMethodEnum*	0:*	Payment methods supported on this distribution. See NeTEx Part1 for allowed values.
	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:*	GROUP OF ENTITIEs, egg 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.

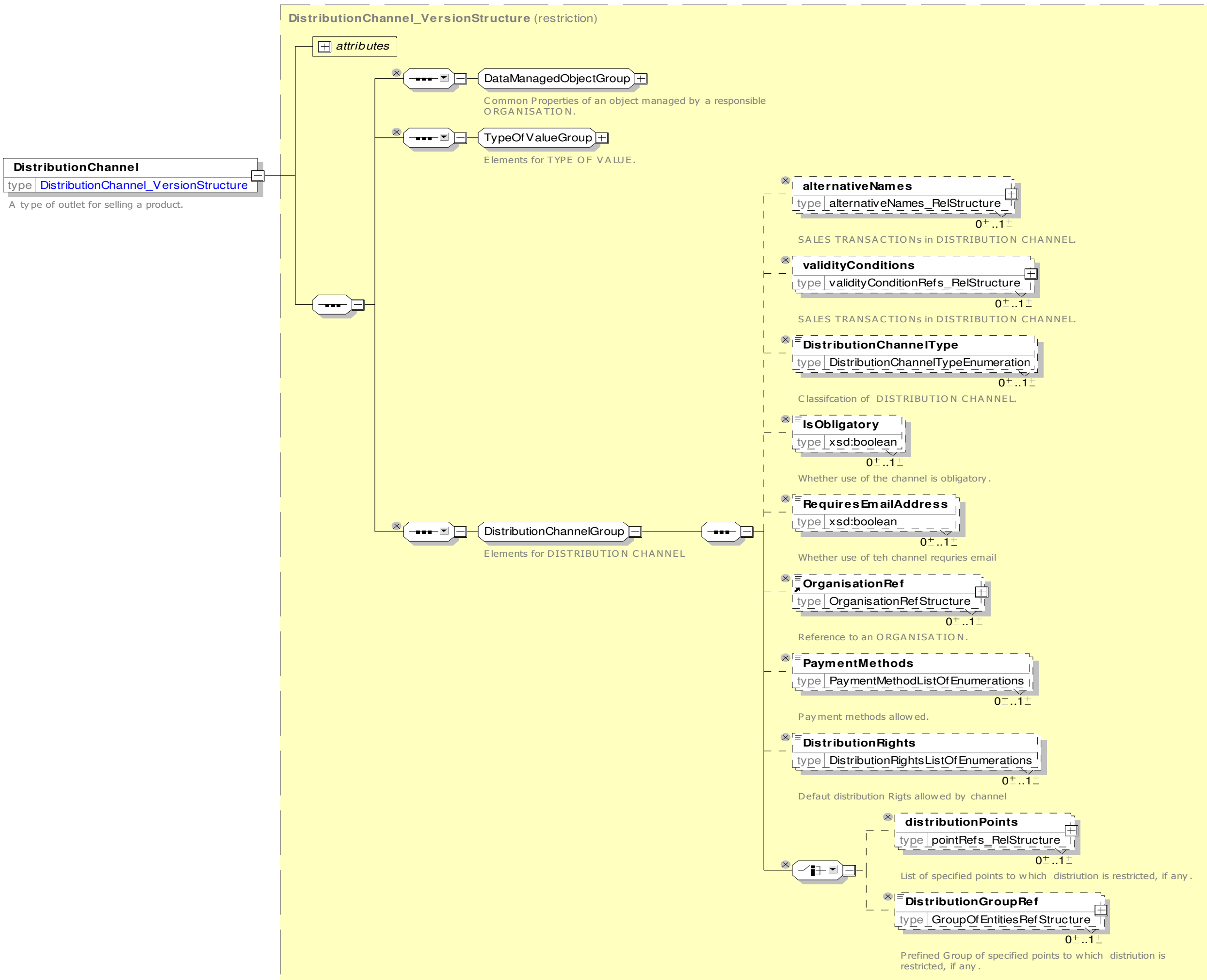


Figure 261 — *DistributionChannel* — XSD

8.8.1.3.1.1 DistributionChannelType – Allowed values

The following table shows the allowed values for ***DistributionChannelType***. (*DistributionChannelTypeEnum).*

Table 221 – *DistributionChannelType* – Allowed values

Value	Description
atStop	At stop sales.
onBoard	On-board sales.
online	On-line sales.
telephone	Telephone sales.
electronicPass	Electronic sale.
mobileDevice	Mobile device.
agency	Third party agency.

tourOperator	Tour operator.
other	Other channel.

8.8.1.3.1.2 DistributionRights – Allowed values

The following table shows the allowed values for ***DistributionRights*** (*DistributionRightsEnum*)

Table 222 – *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.

8.8.1.3.2 GroupOfDistributionChannels – Model Element

A grouping of DISTRIBUTION CHANNELs.

Table 223 – *GroupOfDistributionChannels* – Element

Classifi-cation	**Name**	**Type**	**Cardinality**	**Description**
::>	::>	*GroupOfEntities*	::>	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.

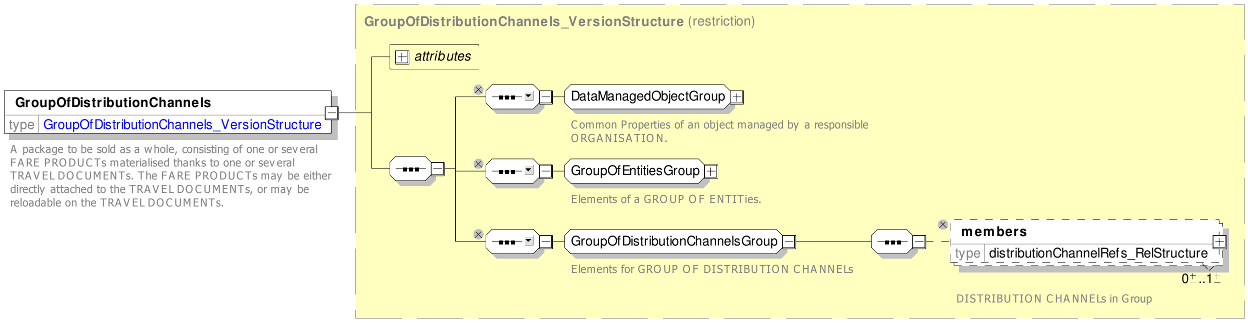


Figure 262 — *GroupOfDistributionChannels* — XSD

8.8.1.3.3 FulfilmentMethod – Model Element

The means by which the ticket is delivered to the Customer. e.g. online, collection, etc.

Table 224 – *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.
	Fulfilment-MethodType	*FulfilmentMethod-TypeEnum*	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.
«contains»	***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.

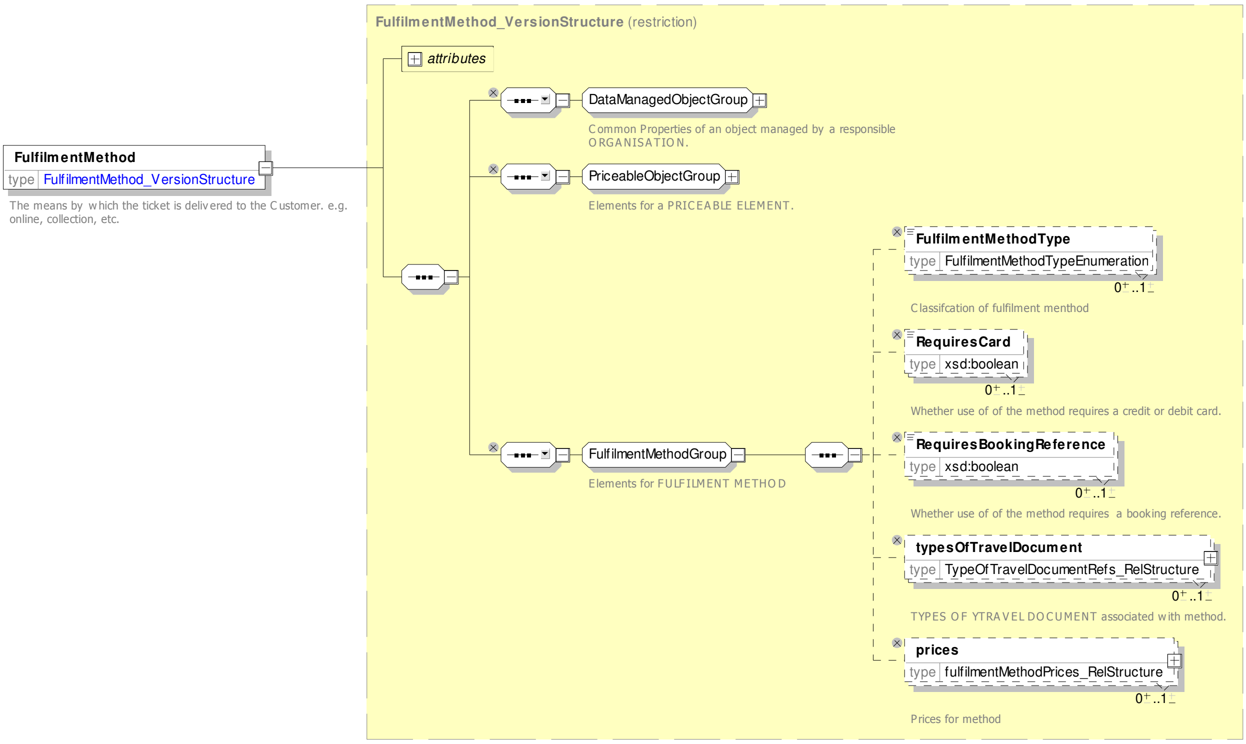


Figure 263 — *FulfilmentMethod* — XSD

8.8.1.3.3.1 FulfilmentMethodType – Allowed values

The following table shows the allowed values for ***FulfilmentMethodType*** (*FulfilmentMethodTypeEnum*).

Table 225 – *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.
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.

8.8.1.3.4 FulfilmentMethodPrice – Model Element

A set of all possible price features of a FULFILMENT METHOD, default total price etc.

Table 226 – *FulfilmentMethodPrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	FULFILMENT METHOD PRICE inherits from FARE PRICE
«PK»	***id***	*FulfilmentMethod-PriceIdType*	1:1	Identifier of FULFILMENT METHOD PRICE.
«FK»	***Fulfilment-MethodRef***	*FulfilmentMethodRef*	1:1	Reference to a FULFILMENT METHOD.

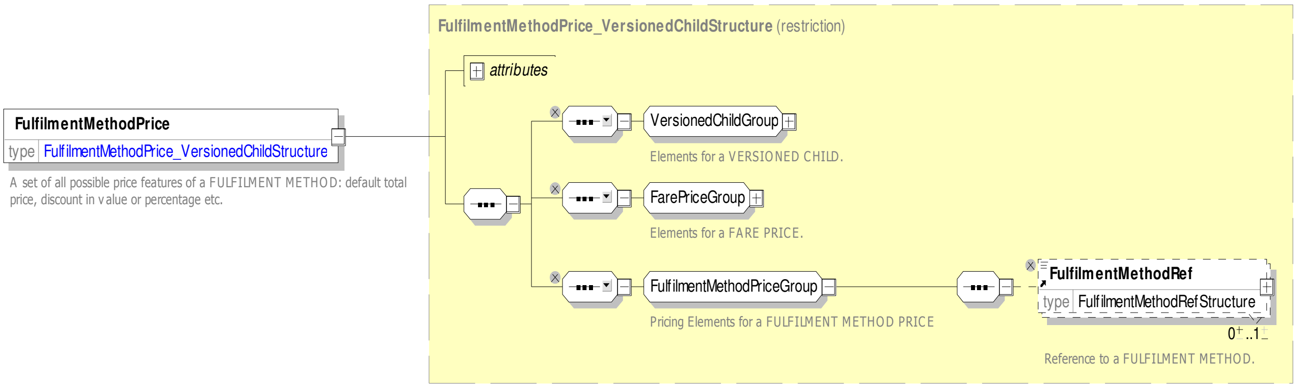


Figure 264 — *FulfilmentMethodPrice* — XSD

8.8.1.3.5 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::SelfServiceMachine">
            <Name>Product is delivered by machine</Name>
            <FulfilmentMethodType>ticketMachine</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
            </typesOfTravelDocument>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl::OverCounter">
            <Name>Product is delivered over counter</Name>
            <FulfilmentMethodType>ticketOffice</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
            </typesOfTravelDocument>
```

```xml
        </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::ElectronicCollection">
            <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::NormalPost">
            <Name>Product is delivered by normal post</Name>
            <FulfilmentMethodType>post</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl::membershipCard"/>
            </typesOfTravelDocument>
            <prices>
                <FulfilmentMethodPrice version="any" id="tfl::NormalPost">
            <Amount>0.00</Amount>
                </FulfilmentMethodPrice>
            </prices>
        </FulfilmentMethod>

        <FulfilmentMethod version="any" id="tfl::RegisteredPost">
            <Name>Product is delivered by registered post</Name>
            <FulfilmentMethodType>post</FulfilmentMethodType>
            <typesOfTravelDocument>
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
                <TypeOfTravelDocumentRef version="any" ref="tfl::membershipCard"/>
            </typesOfTravelDocument>
            <prices>
                <FulfilmentMethodPrice version="any" id="tfl::RegisteredPost">
            <Amount>5.65</Amount>
                </FulfilmentMethodPrice>
            </prices>
        </FulfilmentMethod>
</fulfilmentMethods>
```

8.8.1.3.6 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:

```xml
    <SalesPackage version="any" id="tfl:SalesPackage: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">
                <Name>Can o </Name>
                <DistributionChannelRef version="any" ref="tfl::VisitorCentres"/>
                <DistributionChannelType>online</DistributionChannelType>
                <AllowedInChannel>true</AllowedInChannel>
                <RestrictedToChannel>true</RestrictedToChannel>
                <TicketingServiceFacilityList>purchase</TicketingServiceFacilityList>
                <PaymentMethods>cardsOnly</PaymentMethods>
                <RequiresRegistration>true</RequiresRegistration>
                <FulfilmentMethodRef version="any" ref="tfl:FulfilmentMethod:NormalPost"/>
            </DistributionAssignment>
        </distributionAssignments>
        <SalesPackageRef version="any" ref="tfl::OnOysterCard"/>
    </SalesPackage>
```

8.8.2 Fare Travel Document

8.8.2.1 Fare Travel Document – Conceptual model

The TRAVEL DOCUMENT MODEL indicates the available materialisations of products as tickets on media.

The TRAVEL DOCUMENT entity describes an individual physical support, 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, only a few TRAVEL DOCUMENTs are individually managed in an operator database, namely those 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).

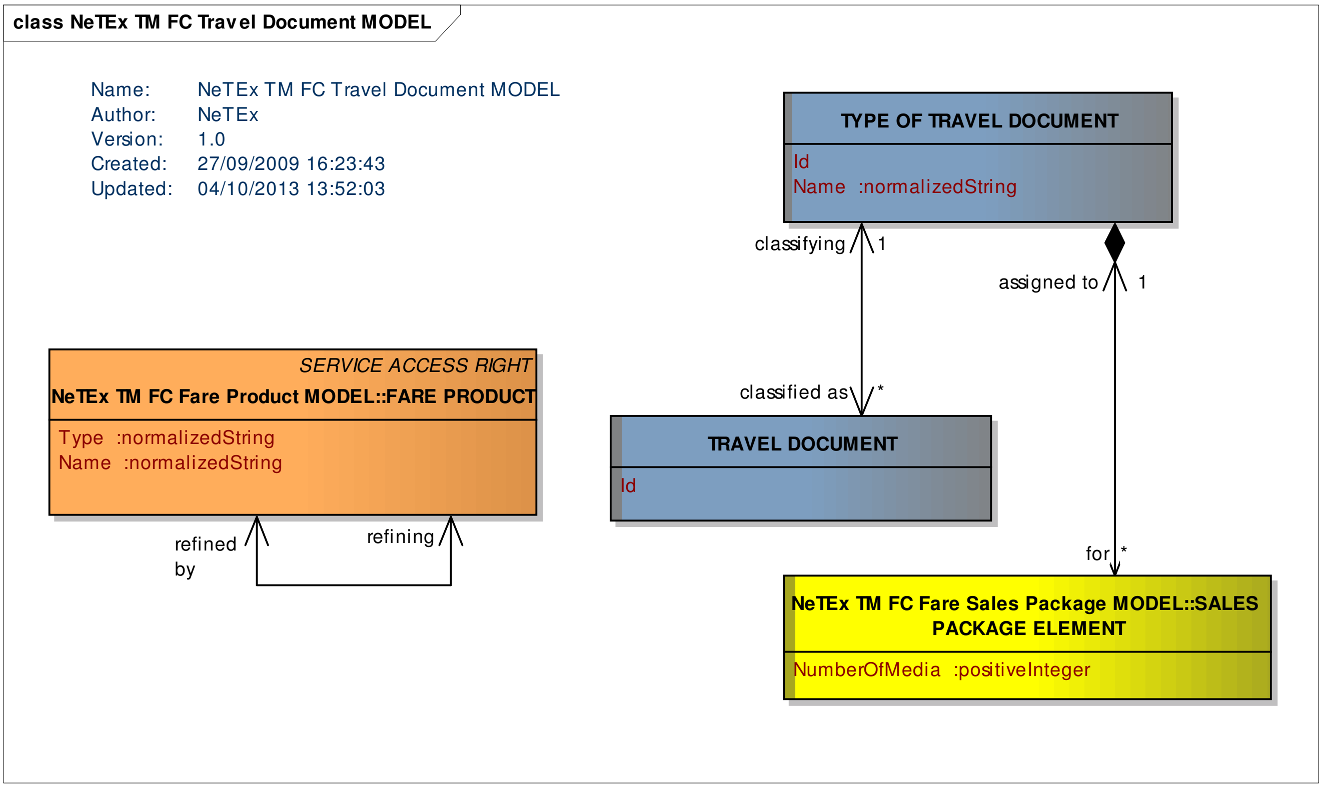


Figure 265 — Travel Document – Conceptual Model

8.8.2.2 Fare Travel Document – Physical model

The following figure shows the physical model for TRAVEL DOCUMENTs.

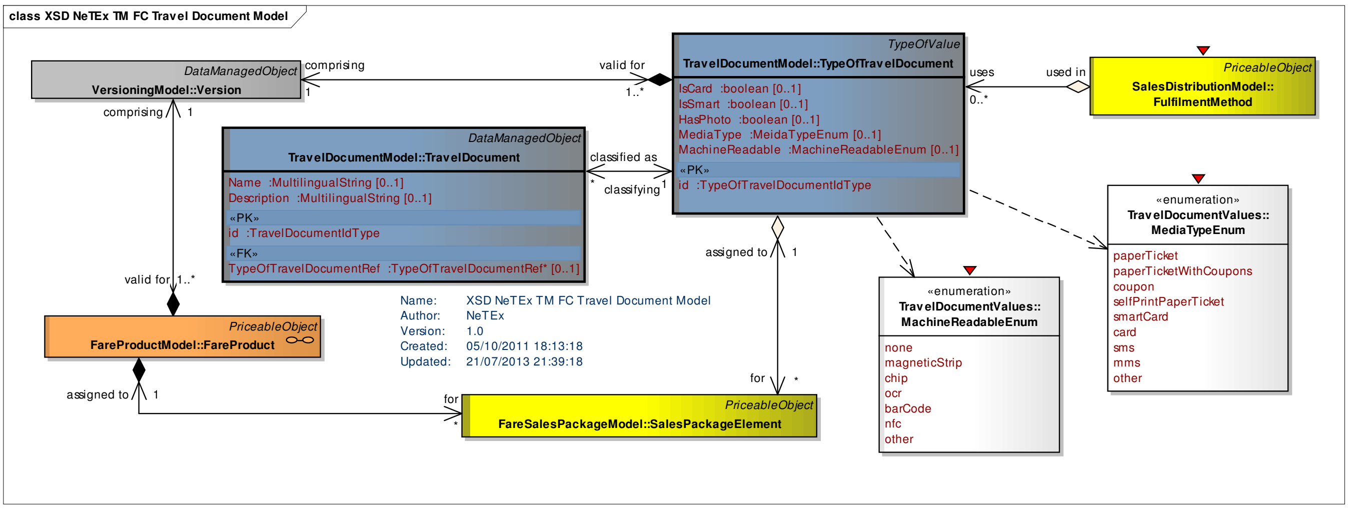


Figure 266 — Travel Document – Physical Model

8.8.2.3 Fare Travel Document – Attributes and XSD

8.8.2.3.1 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 227 – *TypeOfTravelDocument* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	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.
	IsSmart	*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.
	MediaType	*MediaTypeEnum*	0:1	Classification of the TRAVEL DOCUMENT by Media type. See allowed values below.
	Machine-Readable	*MachineReadableEnum*	0:1	Classification of the TRAVEL DOCUMENT by Machine Readable mechanism. See allowed values below.

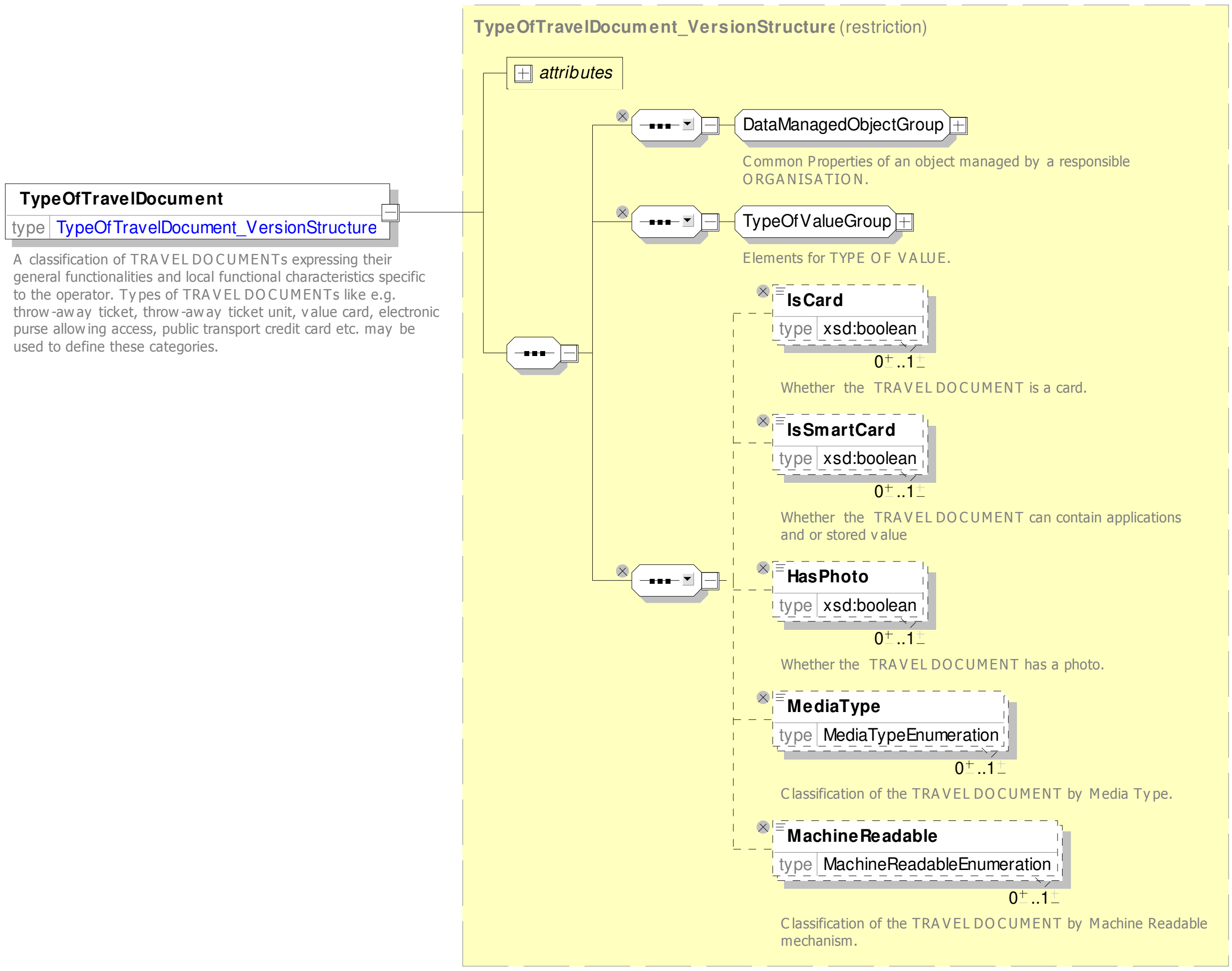


Figure 267 — *TypeOfTravelDocument* — XSD

8.8.2.3.1.1 MachineReadable – Allowed values

The following table shows the allowed values for ***MachineReadable*** *(MachineReadableEnum).*

Table 228 – *MachineReadable* – Allowed values

Value	Description
none	Not Machine readable.
magneticStrip	Readable by chip reader.
chip	Travel document is self-print paper ticket
ocr	Readable by Optical Character Recognition
barCode	Readable by magnetic strip.
nfc	Readable by Near Field Communication.
other	Other channel.

8.8.2.3.1.2 MediaType – Allowed values

The following table shows the allowed values for ***MediaType*** (*MediaTypeEnum*).

Table 229 – *MediaType* – Allowed values

Value	Description
paperTicket	Travel document is paper ticket.
paperTicketWithCoupons	Travel document is an SMS message.
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.

8.8.2.3.2 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 PASSENGER CONTRACT liabilities or a combination of those.

Table 230 – *TravelDocument* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	TRAVEL DOCUMENT inherits from DATA MANAGED OBJECT. See NeTEx Part1.
«PK»	***id***	*TravelDocumentIdType*	1:1	Identifier of TRAVEL DOCUMENT.
	Name	*MultilingualString*	0:1	Name of TYPE OF TRAVEL DOCUMENT.
	Description	*MultilingualString*	0:1	Description of TYPE OF TRAVEL DOCUMENT.
«FK»	***TypeOfTravel-DocumentRef***	*TypeOfTravel-DocumentRef*	0:1	Reference to a TYPE OF TRAVEL DOCUMENT.

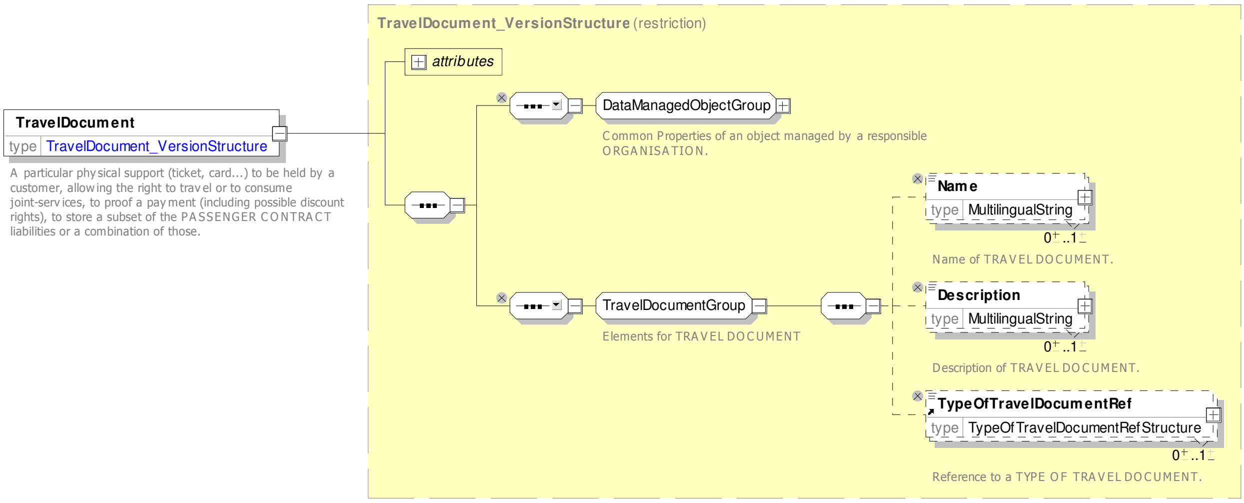


Figure 268 — *TravelDocument* — XSD

8.8.2.4 Fare Travel Document – XML examples

8.8.2.4.1 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::TypeOfTravelDocument" classOfValues="TypeOfTravelDocument">
    <Name>Types of Travel Document </Name>
    <values>
        <TypeOfTravelDocument version="any" id="tfl:TypeOfTravelDocument:paperTicket">
            <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>
```

8.8.2.4.2 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:

```
<travelDocuments>
    <TravelDocument version="any" id="tfl::Ticket">
        <Name>Ticket </Name>
        <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
    </TravelDocument>
    <Name>Oyster </Name>

    <TravelDocument version="any" id="tfl:TravelDocument:OysterPhotoCard">
        <Name>Oyster </Name>
        <TypeOfTravelDocumentRef version="any" ref="tfl::smartPhotoCard"/>
    </TravelDocument>
</travelDocuments>
```

8.8.2.4.3 Travel Document – XML Example of Sales Package

The following code fragment shows a reference to a TYPE OF TRAVEL DOCUMENT by a SALES PACKAGE.

For EXAMPLE:

```
<SalesPackage version="any" id="tfl::TravelCardOnOyster">
    <Name>TravelCard on Oyster</Name>
    <Description>Loaded onto card?</Description>
    <ConditionSummary>
        <ProvidesCard>false</ProvidesCard>
        <GoesOnCard>true</GoesOnCard>
        <IsRefundable>true</IsRefundable>
    </ConditionSummary>
    <salesPackageElements>
        <SalesPackageElement version="any" id="tfl::TravelCardOnOyster@TravelCard">
            <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
            <PreassignedFareProductRef version="any" ref="tfl::TravelCardOnOyster"/>
        </SalesPackageElement>
    </salesPackageElements>
</SalesPackage>
```

8.8.3 Fare Sales Package

8.8.3.1 Fare Sales Package – Conceptual model

The FARE SALES PACKAGE MODEL describes the products marketed to the user and available to purchase.

8.8.3.2 Sales Packages

The FARE PRODUCTs are associated with TRAVEL DOCUMENTs in order to form packages suitable for selling. A SALES 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 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 PACKAGEs are described by SALES 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 PACKAGEs must comprise only SALES 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 PACKAGE may sometimes be subject to a parameter limitation. For instance, a SALES 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 PACKAGE may be made up of one or more SALES PACKAGE ELEMENTs, each specifying a component of the PACKAGE that is purchased as a whole. A GROUP OF SALES 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 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 PACKAGE may comprise a number of different values for each feature of the fare structure – there is not necessarily a separate SALES 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 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 PACKAGE, assigned using a SALES NOTICE ASSIGNMENT.

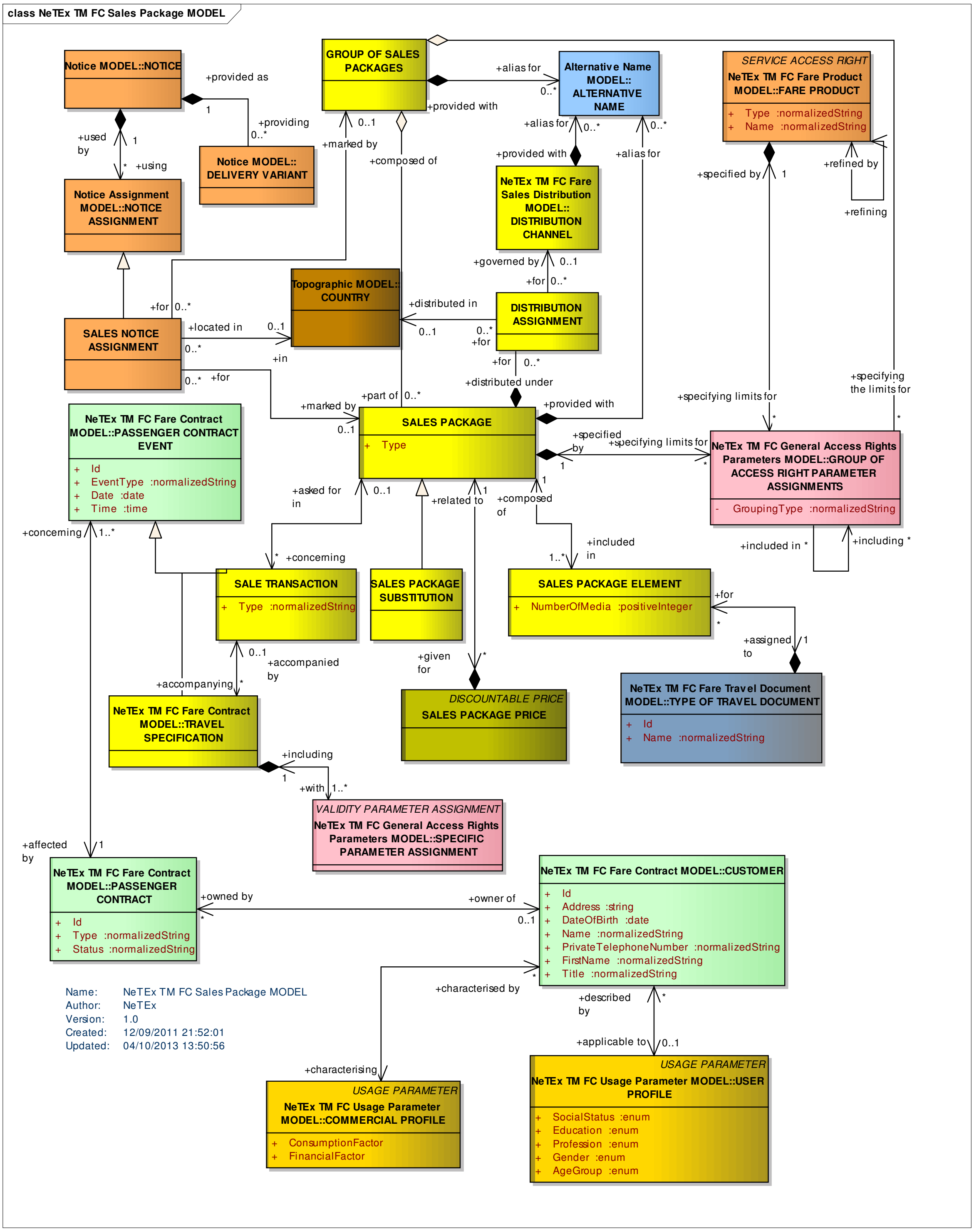


Figure 269 — Sales Package – Conceptual Model

8.8.3.3 Fare Sales Package – Physical model

The following figure shows the physical model for SALES PACKAGEs.

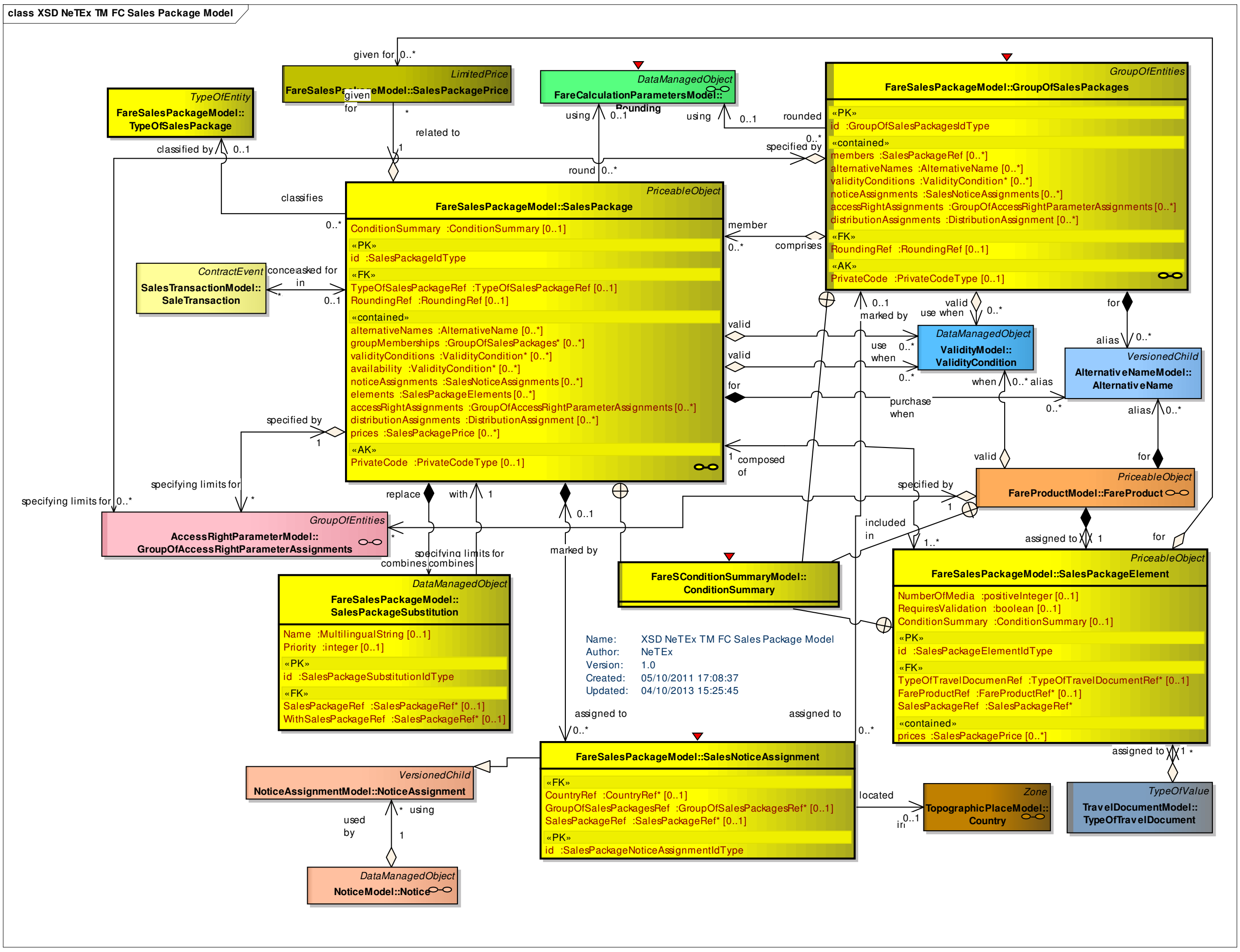


Figure 270 — Sales Package – Physical Model

8.8.3.3.1 Group of Fare Sales Package – Physical model

The following figure shows the physical model for a GROUP of SALES PACKAGEs.

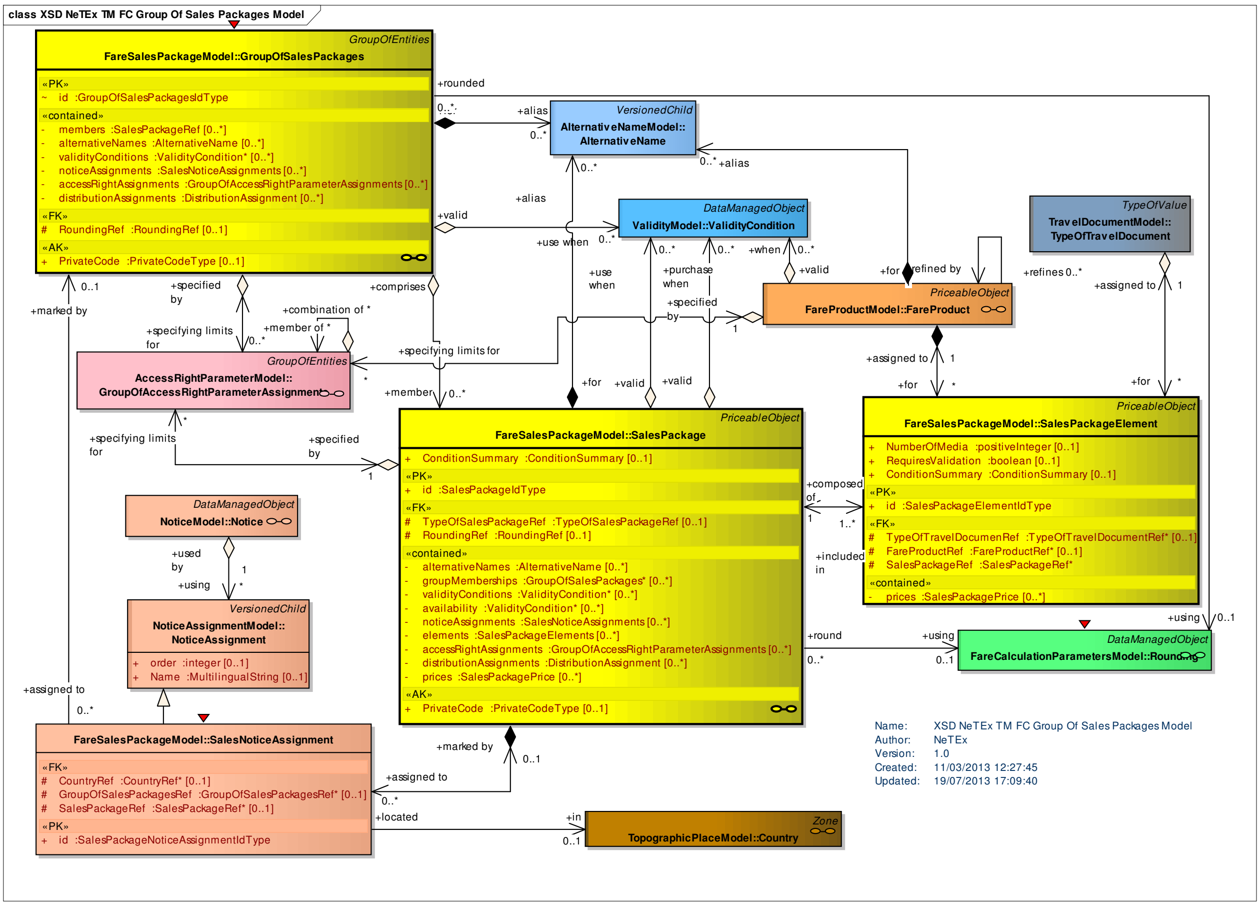


Figure 271 — Group Of Sales Packages – Physical Model

8.8.3.3.2 Distribution Assignment Sales Package – Physical model

The following figure shows the physical model for the DISTRIBUTION ASSIGNMENT of SALEs PACKAGEs.

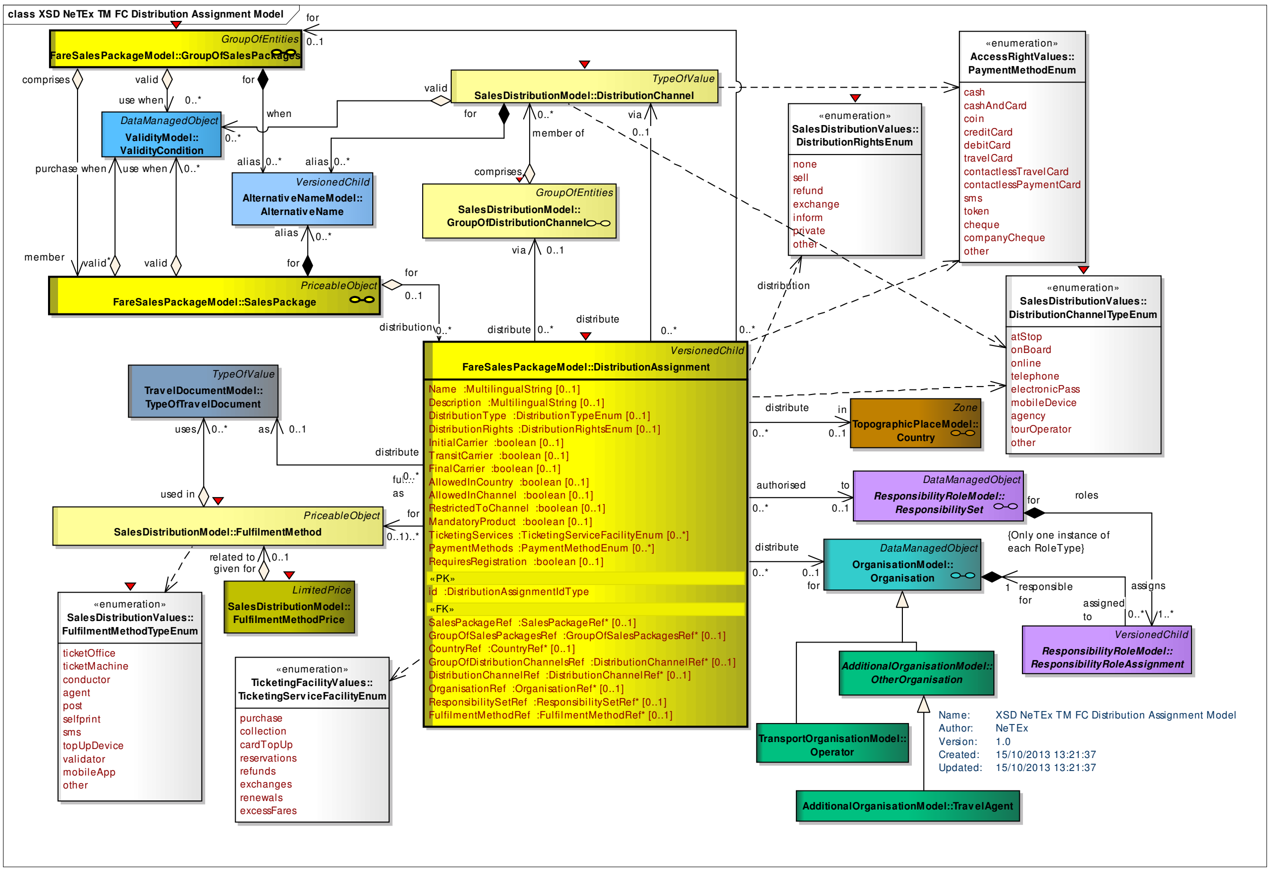


Figure 272 — Distribution Assignment – Physical Model

8.8.3.4 Fare Sales Package – Attributes and XSD

8.8.3.4.1 SalesPackage – 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 231 – *SalesPackage* – Model Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	SALES PACKAGE inherits from PRICEABLE OBJECT.
«PK»	***id***	*SalesPackageIdType*	1:1	Identifier of a SALES PACKAGE.
«AK»	***PrivateCode***	*PrivateCodeType*	0:1	Alternative identifier of an entity. can be used to associate with legacy systems.
«FK»	***TypeOfSales-PackageRef***	*TypeOfSalesPackageRef*	0:1	Type of SALES PACKAGE.
GROUP	***SalesPackage-CommonGroup***	*SalesPackageCommon-Group*	0:1	Common properties of SALES PACKAGE and GROUP OF SALES PACKAGES.
«FK»	***GroupOfSales-***	*GroupOfSales-*	0:1	GROUP OF SALES PACKAGES with which this

	PackagesRef	*PackageRef*		SALES PACKAGE shares common properties.
“cntd»	***salesPackage-Substitutions***	*SalesPackage-Substitution*	0:*	Allowed SALES PACKAGE SUBSITUTIONs for the SALES PACKAGE.

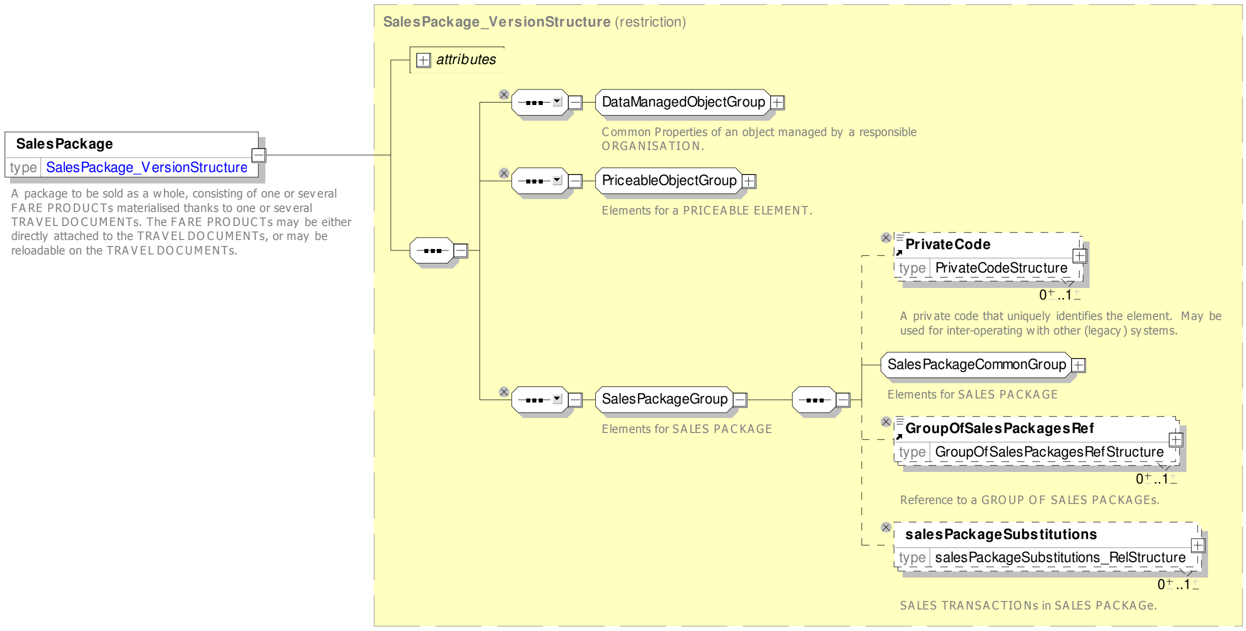


Figure 273 — *SalesPackage* — XSD

8.8.3.4.1.1 SalesPackageCommonGroup – Group

The ***SalesPackageCommonGroup*** specifies the common properties of SALES PACKAGE and GROUP OF SALES.

Table 232 – *SalesPackageCommonGroup* – Model Element– Element

Classifi-cation	Name	Type	Cardinality	Description
“cntd»	***validity-Conditions***	*ValidityCondition*	0:*	VALIDITY CONDITIONs specifying specific availability of SALES PACKAGE for use.
«FK»	***TypeOfSales-PackageRef***	*TypeOfSalesPackageRef*	0:1	Type of SALES PACKAGE.
“cntd»	***availabilityFor-Purchase***	*ValidityCondition*	0:*	VALIDITY CONDITIONs specifying specific availability of SALES PACKAGE for purchase.
	Condition-Summary	*ConditionSummary*	0:1	Summary description of conditions of a SALES PACKAGE that can be used to provide passenger information
“cntd»	***accessRight-Parameter-Assignments***	*AccessRightParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs associated with the SALES PACKAGE.

“cntd»	***distribution-Assignments***	*DistributionAssignment*	0:*	DISTRIBUTION ASSIGNMENTs for the SALES PACKAGE.
«FK»	***RoundingRef***	*RoundingRef*	0:1	Rounding to use on calculation
“cntd»	***prices***	*SalesPackagePrice*	0:*	SALES PACKAGE PRICEs associated with the FARE
“cntd»	***priceGroups***	*PriceGroup*	0:*	PRICE GROUPs associated with the FARE
“cntd»	***elements***	*SalesPackageElement*	0:*	SALES PACKAGE ELEMENTs associated with the SALES PACKAGE.

Figure 274 — *SalesPackageCommonGroup* — XSD

8.8.3.4.2 SalesPackageElement – Model Element

The assignment of a FARE PRODUCT to a TYPE OF TRAVEL DOCUMENT in order to define a SALES 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 233 – *SalesPackageElement* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PriceableObject*	::>	SALES PACKAGE ELEMENT inherits from

				PRICEABLE OBJECT.
«PK»	***id***	*SalesPackage-ElementIdType*	1:1	Identifier of SALES PACKAGE ELEMENT.
WHAT IS TH?	***NumberOfMedia***	*xsd:positiveInteger*	0:1	Number of media package in which package is available
	Requires-Validation	*xsd:boolean*	0:1	.Whether element requires validation before it can be used.
	Condition-Summary	*ConditionSummary*	0:1	Summary description of SALES PACKAGE properties.
«FK»	***SalesPackageRef***	*SalesPackageRef*	1:1	Reference to a SALES PACKAGE of which this is part.
«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 PACKAGE.
“cntd»	***accessRight-Parameter-Assignments***	*AccessRightParameter-Assignment*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs associated with the SALES PACKAGE ELEMENT.
“cntd»	***prices***	*SalesPackagePrice*	0:*	SALES PACKAGE PRICEs associated with the SALES PACKAGE ELEMENT.

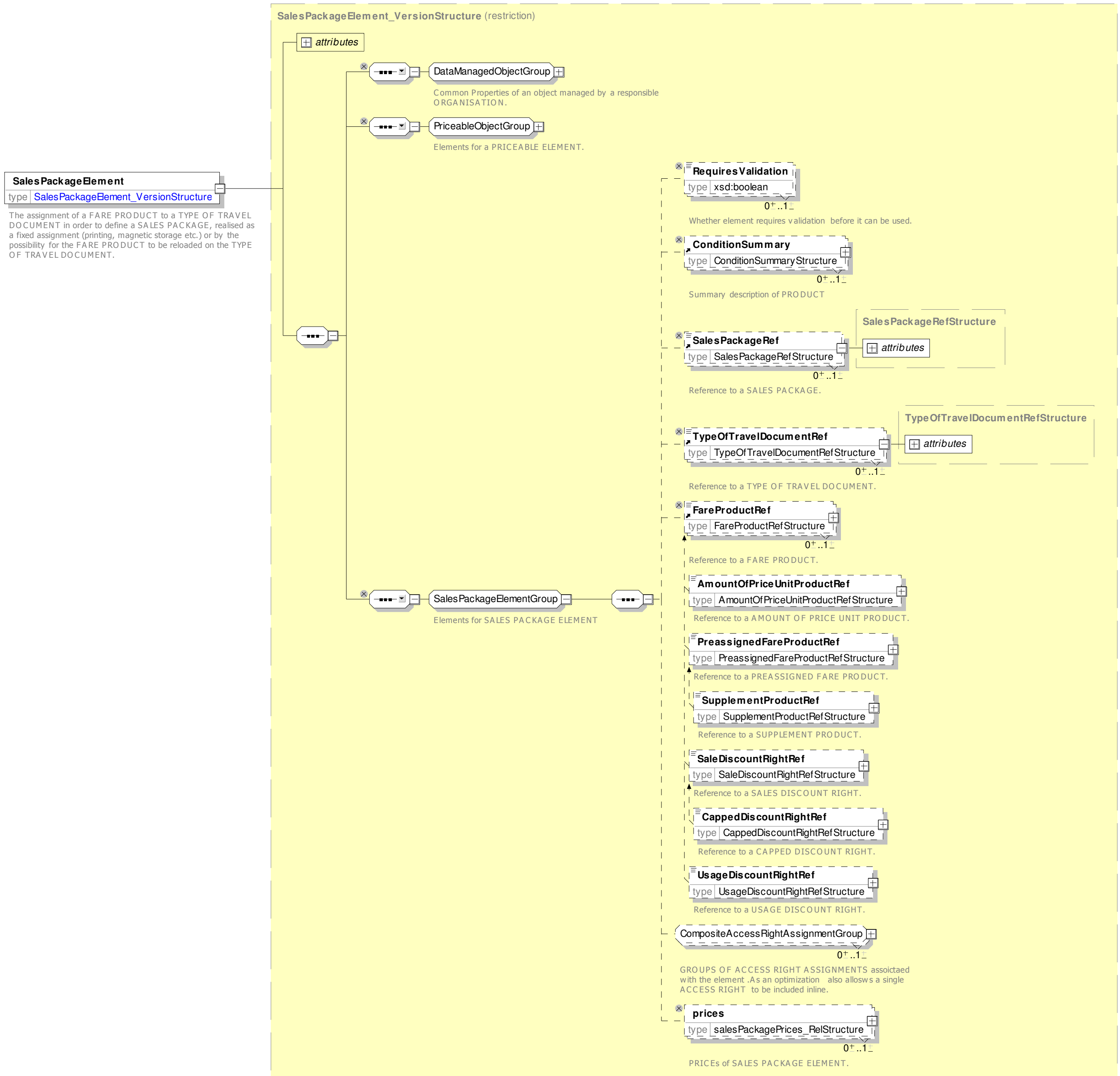


Figure 275 — *SalesPackageElement* — XSD

8.8.3.4.3 SalesPackagePrice – Model Element

A set of all possible price features of a SALES PACKAGE: default total price etc.

Table 234 – *SalesPackagePrice* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	SALES PACKAGE PRICE inherits from FARE PRICE
«PK»	***id***	*SalesPackage-PriceIdType*	1:1	Identifier of SALES PACKAGE PRICE.

		choice	1:1	Reference to element for which this is the price.
«FK»	***SalesPackageRef***	*SalesPackageIdType*	1:1	Reference to a SALES PACKAGE.
«FK»	***SalesPackage-ElementRef***	*SalesPackage-ElementIdType*	1:1	Reference to a SALES PACKAGE ELEMENT.

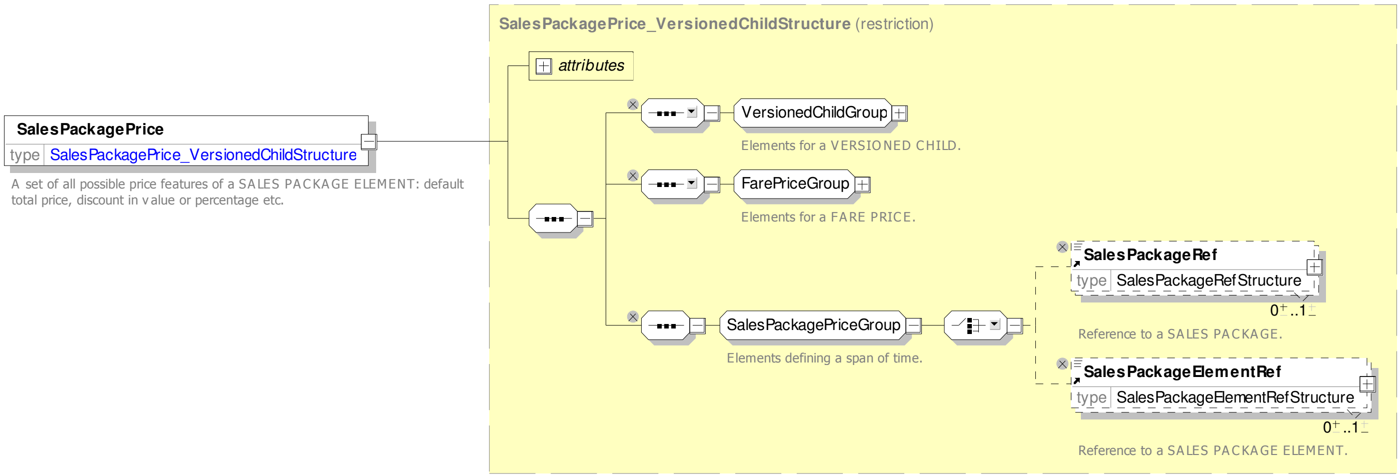


Figure 276 — *SalesPackagePrice* — XSD

8.8.3.4.4 GroupOfSalesPackages – Model Element

A grouping of SALES PACKAGEs.

Table 235 – *GroupOfSalesPackages* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*GroupOfEntities*	::>	GROUP of SALES PACKAGEs inherits from GROUP OF ENTITIES. See NeTEx Part1.
«PK»	***id***	*GroupOfSales-PackagesIdType*	1:1	Identifier of GROUP of SALES PACKAGEs.
“cntd»	***members***	*SalesPackageRef*	0:*	References to members of GROUP of SALES PACKAGEs. See above.
	SalesPackage-CommonGroup	*SalesPackageCommon-Group*	0:1	Common properties of SALES PACKAGE and GROUP OF SALES PACKAGES.

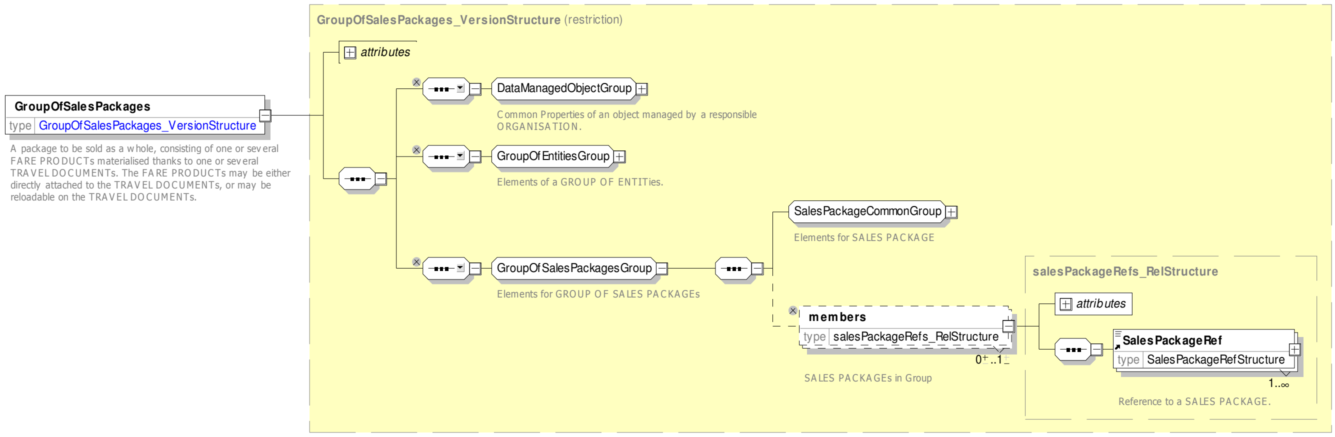


Figure 277 — *GroupOfSalesPackages* — XSD

8.8.3.4.5 DistributionAssignment – Model Element

An assignment of the COUNTRY and/or DISTRIBUTION CHANNEL through which a product may or may not be distributed.

Table 236 – *DistributionAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*VersionedChild*	::>	DISTRIBUTION ASSIGNMENT inherits from VERSIONED CHILD. See NeTEx Part1.
«PK»	***id***	*DistributionAssignmentId Type*	1:1	Identifier of a DISTRIBUTION ASSIGNMENT.
	Name	*MultilingualString*	0:1	Name of DISTRIBUTION ASSIGNMENT.
	Description	*MultilingualString*	0:1	Description of DISTRIBUTION ASSIGNMENT.
«FK»	***SalesPackageRef***	*SalesPackageRef*	0:1	SALES PACKAGE for which this specifies the DISTRIBUTION ASSIGNMENT.
«FK»	***GroupOfSales-PackagesRef***	*GroupOfSales-PackagesRef*	0:1	GROUP OF SALES PACKAGEs for which this specifies the DISTRIBUTION ASSIGNMENT.
	Distribution-Rights	*DistributionRightsEnum*	0:1	Rights allowed by this DISTRIBUTION ASSIGNMENT.

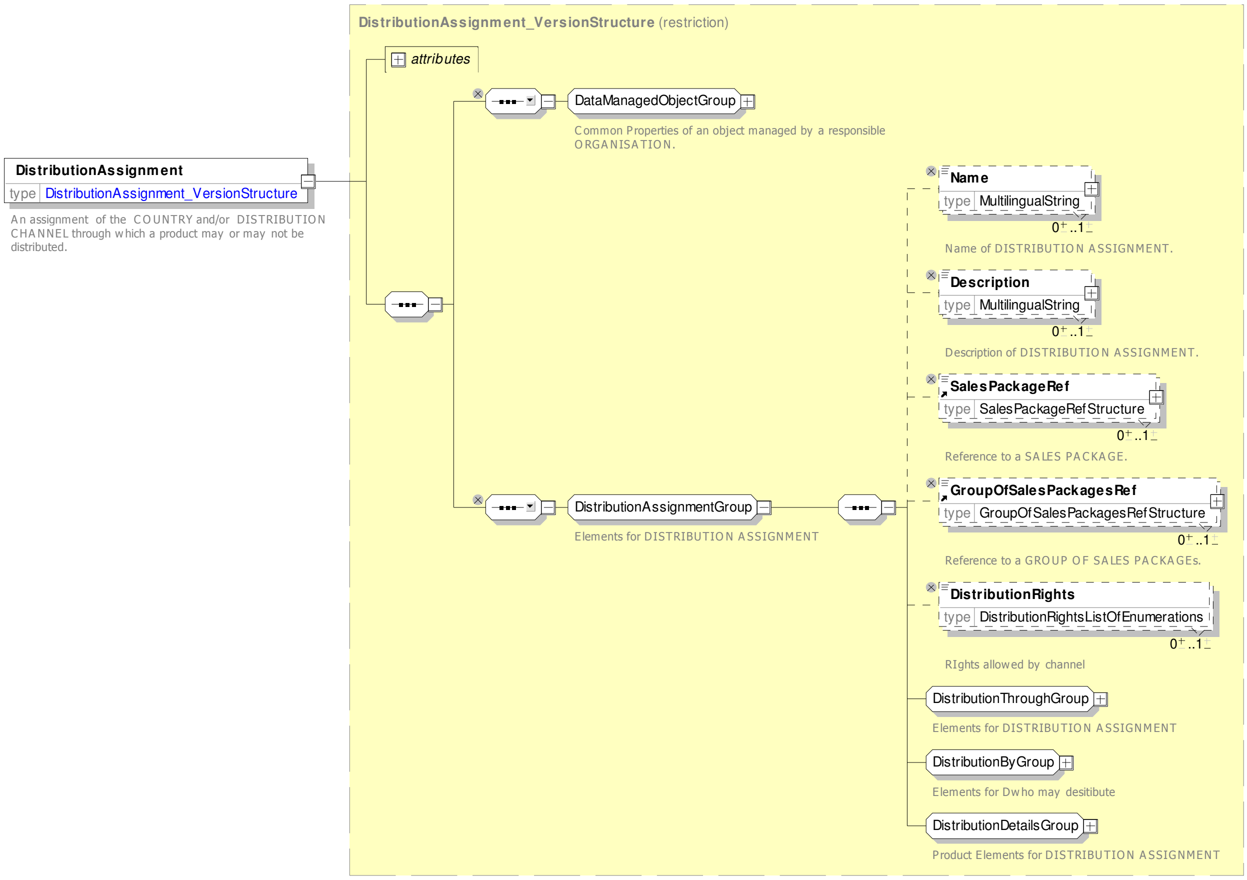


Figure 278 — *DistributionAssignment* — XSD

8.8.3.4.5.1 DistributionThroughGroup– Group

The ***DistributionThroughGroup*** group specifies the elements relating to the channels through which distribution of products can be made.

Table 237 – DistributionThroughGroup– Model Element– Element

Classifi-cation	Name	Type	Cardinality	Description
«FK»	***CountryRef***	*CountryRef*	0:1	COUNTRY for which this specifies the DISTRIBUTION ASSIGNMENT.
	AllowedIn-Country	*xsd:boolean*	0:1	Whether distribution is allowed or forbidden in the specified country.
«FK»	***GroupOf-Distribution-ChannelsRef***	*DistributionChannelRef*	0:1	GROUP OF DISTRIBUTION CHANNELs for which this specifies the DISTRIBUTION ASSIGNMENT.
«FK»	***Distribution-ChannelRef***	*DistributionChannelRef*	0:1	DISTRIBUTION CHANNEL for which this specifies the DISTRIBUTION ASSIGNMENT.
	Distribution-	*DistributionChannel-*	0:1	Type of DISTRIBUTION CHANNEL. See Earlier.

	ChannelType	*TypeEnum*		
	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 279 — *DistributionThroughGroup* — XSD

8.8.3.4.5.2 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 238 – *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
«FK»	***OrganisationRef***	*OrganisationRef*	0:1	ORGANISATION for which this specifies the DISTRIBUTION ASSIGNMENT.
«FK»	***Responsibility-SetRef***	*ResponsibilitySetRef*	0:1	RESPONSIBILITY SET describing the DISTRIBUTION ASSIGNMENT.

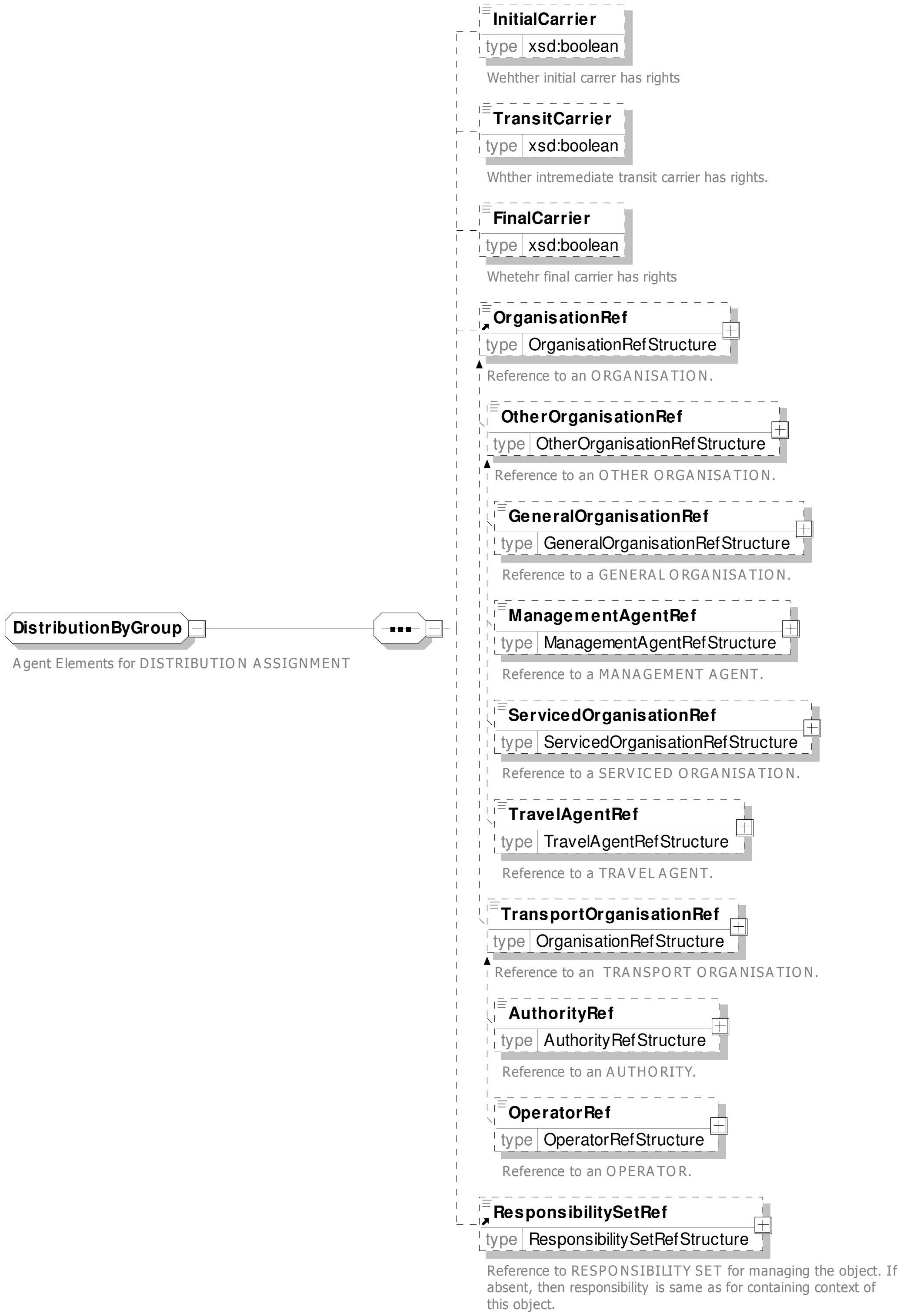


Figure 280 — *DistributionByGroup* — XSD

8.8.3.4.5.3 DistributionDetailsGroup– Group

The ***DistributionDetailsGroup*** group specifies the properties that can be set by a DISTRIBUTION ASSIGNMENT.

Table 239 – *DistributionDetailsGroup*– Model Element– Element

Classifi-cation	Name	Type	Cardinality	Description
	Ticketing-Services	*TicketingService-FacilityEnum*	0:*	Payment method supported on this distribution.
	PaymentMethods	*PaymentMethodEnum*	0:*	Payment method supported on this distribution.
	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.

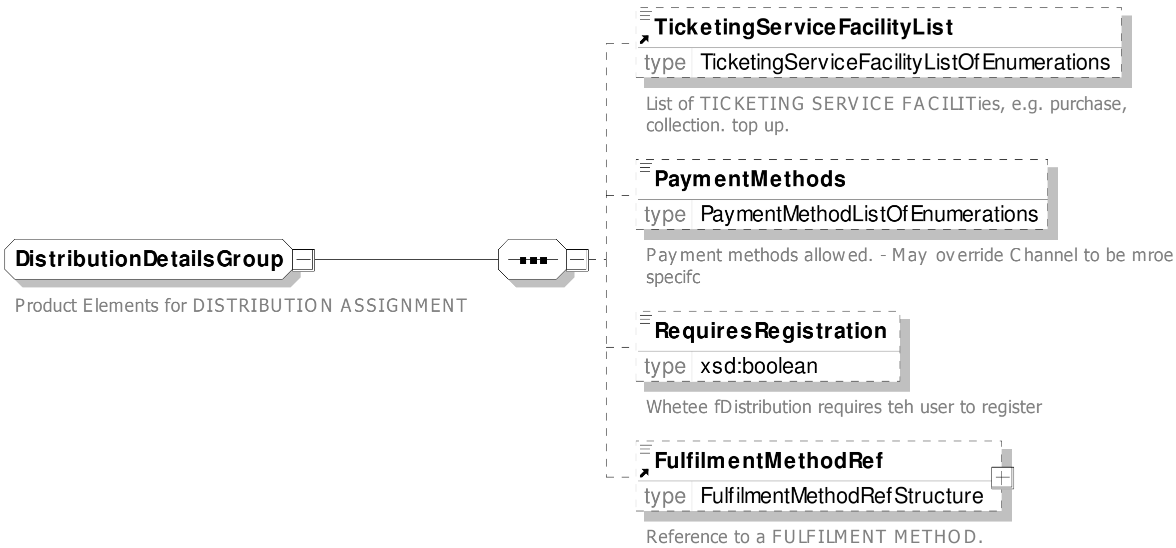


Figure 281 — *DistributionDetailsGroup*— XSD

8.8.3.4.6 SalesNoticeAssignment – Model Element

The assignment of a NOTICE to a SALES PACKAGE or a GROUP OF SALES PACKAGEs.

Table 240 – *SalesNoticeAssignment* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*NoticeAssignment*	::>	SALES NOTICE ASSIGNMENT inherits from NOTICE ASSIGNMENT
«FK»	***CountryRef***	*CountryRef*	0:1	Reference to an COUNTRY to which NOTICE applies.

«PK»	***id***	*SalesNotice-AssignmentIdType*	1:1	Identifier of a SALES NOTICE ASSIGNMENT.
«FK»	***GroupOfSales-PackagesRef***	*GroupOfSales-PackagesRef*	0:1	Reference to a GROUP of SALES PACKAGE to which assignment is made.
«FK»	***SalesPackageRef***	*SalesPackageRef*	0:1	Reference to a SALES PACKAGE of which this is part.

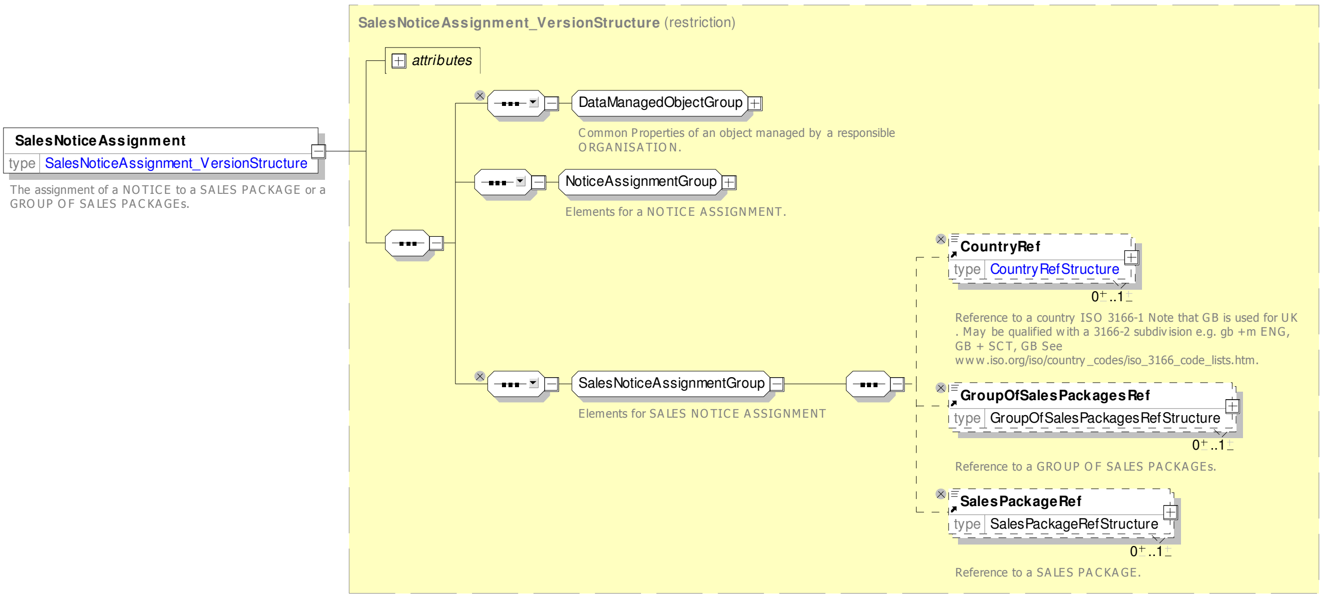


Figure 282 — *SalesNoticeAssignment* — XSD

8.8.3.4.7 SalesPackageSubstitution – Model Element

Information on the preferred substitution of packages with other package if quota restricted product is no longer available.

Table 241 – *SalesPackageSubstitution* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	SALES PACKAGE SUBSTITUTION inherits from DATA MANAGED OBJECT. See NeTEx Part1.
«PK»	***id***	*SalesPackage-SubstitutionIdType*	1:1	Identifier of SALES PACKAGE SUBSTITUTION.
	Name	*MultilingualString*	0:1	Name of SALES PACKAGE SUBSTITUTION.
«FK»	***SalesPackageRef***	*SalesPackageRef*	0:1	First SALES PACKAGE in combination.
«FK»	***WithSales-PackageRef***	*SalesPackageRef*	0:1	Second SALES PACKAGE in combination.
	Priority	*xsd:integer*	0:1	Relative priority for SALES PACKAGE

				SUBSTITUTION.

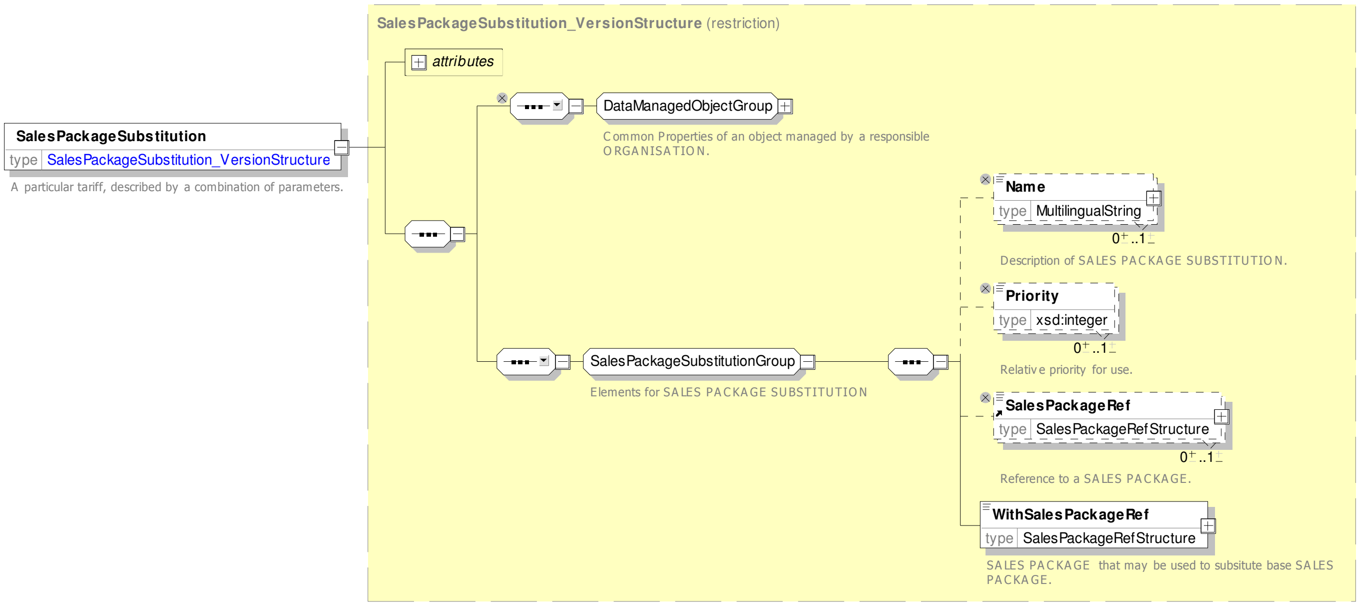


Figure 283 — *SalesPackageSubstitution* — XSD

8.8.3.4.8 TypeOfSalesPackage – Model Element

A classification of a SALES PACKAGE.

Table 242 – *TypeOfSalesPackage* – Element

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

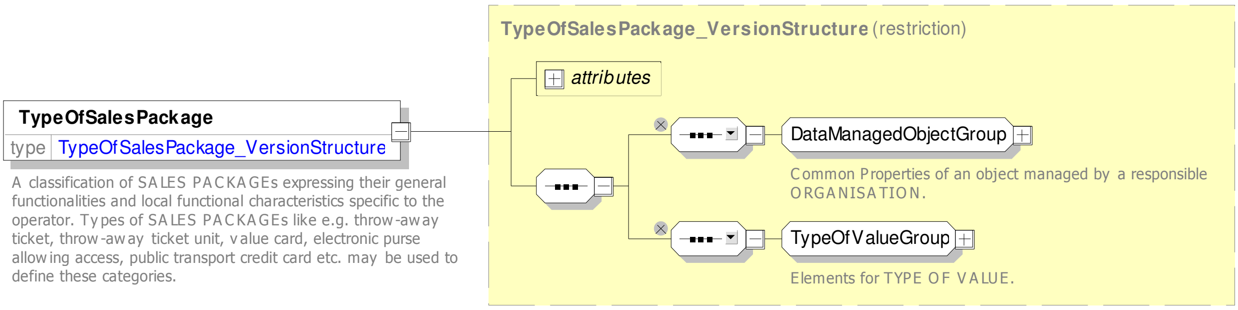


Figure 284 — *TypeOfSalesPackage* — XSD

8.8.3.5 Fare Sales Package – XML examples

8.8.3.5.1 Sales Package: XML Example of tickets

The following code fragment defines separate SALES PACKAGEs for paper ticket, group ticket (also on paper) and an electronic ticket.

For EXAMPLE:

```
<salesPackages>
    <SalesPackage version="any" id="tfl::Ticket">
        <Name>Mag or paper ticket is issued</Name>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesPackageElements>
            <SalesPackageElement version="any" id="tfl::Ticket@PrepaidFare@single">
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
                <PreassignedFareProductRef version="any" ref="tfl::PrepaidFare"/>
            </SalesPackageElement>
            <SalesPackageElement version="any" id="tfl::Ticket@PrepaidFare@return">
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
                <PreassignedFareProductRef version="any" ref="tfl::PrepaidFare"/>
            </SalesPackageElement>
        </salesPackageElements>
    </SalesPackage>

    <SalesPackage version="any" id="tfl::GroupDayTicket">
        <Name>Mag or paper ticket is issued</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>
        <salesPackageElements>
            <SalesPackageElement version="any" id="tfl::GroupDayTicket@FareProduct">
                <TypeOfTravelDocumentRef version="any" ref="tfl::paperTicket"/>
                <PreassignedFareProductRef version="any" ref="tfl::GroupDayTicket"/>
            </SalesPackageElement>
        </salesPackageElements>
    </SalesPackage>

    <!-- Other Ticket -->
    <SalesPackage version="any" id="tfl::PayAsYouGoFare">
        <Name>On Card ticket purchase</Name>
        <ConditionSummary>
            <ProvidesCard>false</ProvidesCard>
            <GoesOnCard>true</GoesOnCard>
            <IsRefundable>false</IsRefundable>
        </ConditionSummary>
        <salesPackageElements>
            <SalesPackageElement version="any" id="tfl::PayAsYouGoFare@single">
                <TypeOfTravelDocumentRef version="any" ref="tfl:TypeOfTravelDocument:smartCard"/>
                <PreassignedFareProductRef version="any" ref="tfl::PayAsYouGoFare"/>
            </SalesPackageElement>
        </salesPackageElements>
    </SalesPackage>
```

8.8.3.5.2 Sales Package: XML Example of card purchase

The following example (Based on the TfL Oyster card) shows a SALES PACKAGE for an electronic card that has separate SALES 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:

```
    <SalesPackage version="any" id="tfl::OnOysterCard">
        <Name>Product is carried on Oyster card </Name>
        <ConditionSummary>
            <ProvidesCard>true</ProvidesCard>
            <IsPersonal>false</IsPersonal>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesPackageElements>
            <SalesPackageElement version="any" id="tfl::OnOysterCard@OysterPayAsYouGoRight">
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                <CappedDiscountRightRef version="any" ref="tfl::OysterPayAsYouGoRight"/>
            </SalesPackageElement>
            <SalesPackageElement 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"/>
            </SalesPackageElement>
        </salesPackageElements>
    </SalesPackage>
```

8.8.3.5.3 Sales Package: XML Example of card Top up products

SALES PACKAGEs may be used for marketable transactions other than simple ticket purchase. The following code fragment defines separate SALES PACKAGEs to (a) top up an oyster card by an arbitrary amount; and (b) to register for automatic top up

For EXAMPLE:

```
        <SalesPackage version="any" id="tfl::OysterTopUp">
            <Name>Card Top up Put value on card</Name>
            <ConditionSummary>
                <ProvidesCard>false</ProvidesCard>
                <GoesOnCard>true</GoesOnCard>
                <IsRefundable>true</IsRefundable>
            </ConditionSummary>
            <salesPackageElements>
                <SalesPackageElement version="any" id="tfl::OysterTopUp@PrepaidFare@OysterTopUp">
                    <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                    <AmountOfPriceUnitProductRef version="any" ref="tfl::OysterTopUp"/>
                </SalesPackageElement>
            </salesPackageElements>
        </SalesPackage>

        <SalesPackage version="any" id="tfl::OysterAutoTopUp">
            <Name>Automatically top up when credit threshold drops to specified amount</Name>
            <ConditionSummary>
                <ProvidesCard>false</ProvidesCard>
                <GoesOnCard>true</GoesOnCard>
                <IsRefundable>true</IsRefundable>
            </ConditionSummary>
            <salesPackageElements>
                <SalesPackageElement version="any"
                    id="tfl::OysterAutoTopUp@PrepaidFare@OysterTopUp">
                    <TypeOfTravelDocumentRef version="any" ref="tfl::smartCard"/>
                    <SupplementProductRef version="any" ref="tfl::OysterAutoTopUp"/>
                </SalesPackageElement>
            </salesPackageElements>
        </SalesPackage>
```

8.8.3.5.4 Sales Package: XML Example of Group of Sale Packages to share properties

There may be many similar SALES PACKAGEs sharing a common attributes. The following code fragment shows separate SALES PACKAGEs for various types of personalised Oyster Photo cards, all of which share common base properties defined by a GROUP OF SALES PACKAGES.

For EXAMPLE:

```
<groupsOfSalesPackages>
    <GroupOfSalesPackages version="any" id="tfl::onOysterCardGroup">
        <Name>Current Products carried on Oyster card </Name>
        <ConditionSummary>
```

```xml
            <ProvidesCard>true</ProvidesCard>
            <IsRefundable>true</IsRefundable>
        </ConditionSummary>
        <salesPackageElements>
            <SalesPackageElement version="any" id="tfl::onOysterPhotoCardGroup@OysterPayAsYouGo">
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartPhotoCard"/>
                <CappedDiscountRightRef version="any" ref="tfl::OysterPayAsYouGoRight"/>
            </SalesPackageElement>
            <SalesPackageElement version="any"
                    id="tfl::onOysterPhotoCardGroup@TravelCardOnOyster">
                <TypeOfTravelDocumentRef version="any" ref="tfl::smartPhotoCard"/>
                <PreassignedFareProductRef version="any" ref="tfl::TravelCardOnOyster"/>
            </SalesPackageElement>
            <SalesPackageElement 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"/>
            </SalesPackageElement>
        </salesPackageElements>
            <members>
                <SalesPackageRef version="any" ref="tfl::60PlusLondonOysterPhotoCard"/>
                <SalesPackageRef version="any" ref="tfl::11-15ZipOysterPhotoCard"/>
                <SalesPackageRef version="any" ref="tfl::16PlusZipOysterPhotoCard"/>
                <SalesPackageRef version="any" ref="tfl::18PlusOysterPhotoCard"/>
                <SalesPackageRef version="any" ref="tfl::VeteransOysterPhotoCard"/>
                <SalesPackageRef version="any" ref="tfl::ApprenticeOysterPhotocard"/>
            </members>
    </GroupOfSalesPackages>
</groupsOfSalesPackages>

<salesPackages>
    <SalesPackage version="any" id="tfl::60PlusLondonOysterPhotoCard">
        <Name>60 + London Oyster Product is carried on Oyster card </Name>

        <accessRightParameterAssignments>
            <GenericParameterAssignment
                <id="tfl:: 60PlusLondonOysterPhotoCard@UserProfile">
                    <for>
                        <UserProfileRef version="any" ref="tfl:: senior"/>
                    </for>
            </GenericParameterAssignment>
        </accessRightParameterAssignments>
        <GroupOfSalesPackagesRef version="any" ref="tfl::onOysterPhotoCardGroup"/>
    </SalesPackage>

    <SalesPackage version="any" id="tfl::11-15ZipOysterPhotoCard">
            <Name>11-15 ZIP Oyster Product is carried
            <accessRightParameterAssignments>
                <GenericParameterAssignment  id="tfl::11-15ZipOysterPhotoCard@UserProfile">
                    <for>
                        <UserProfileRef version="any" ref="tfl::child11-15"/>
                    </for>
                </GenericParameterAssignment>
            </accessRightParameterAssignments>
            <GroupOfSalesPackagesRef version="any" ref="tfl::onOysterPhotoCardGroup"/>
        </SalesPackage>
Etc., etc
</salesPackages>
```

9 Sales Transactions

9.1 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 PASSENGER 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 PACKAGEs as specific selections of fare elements.

— The SALES TRANSACTION FRAME model describes the elements used to group data for exchange

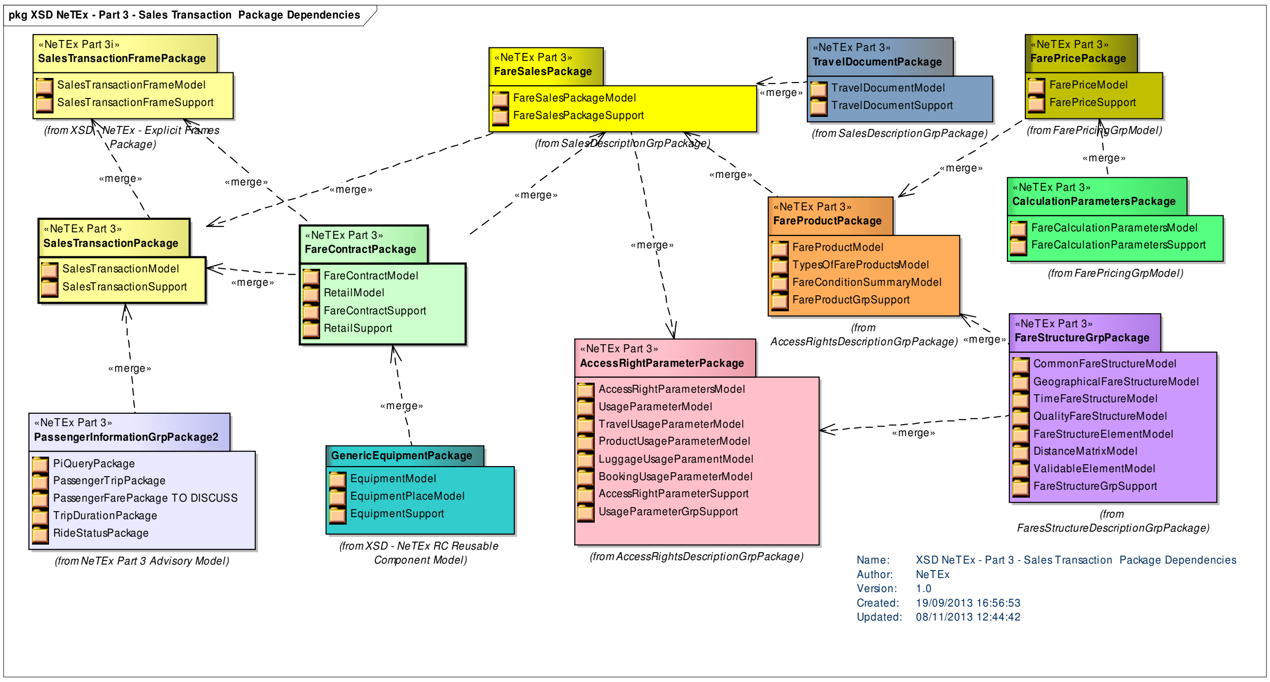


Figure 285 — Sales Transaction Package Dependencies

The following diagram shows the dependencies between the SALES TRANSACTION MODELs.

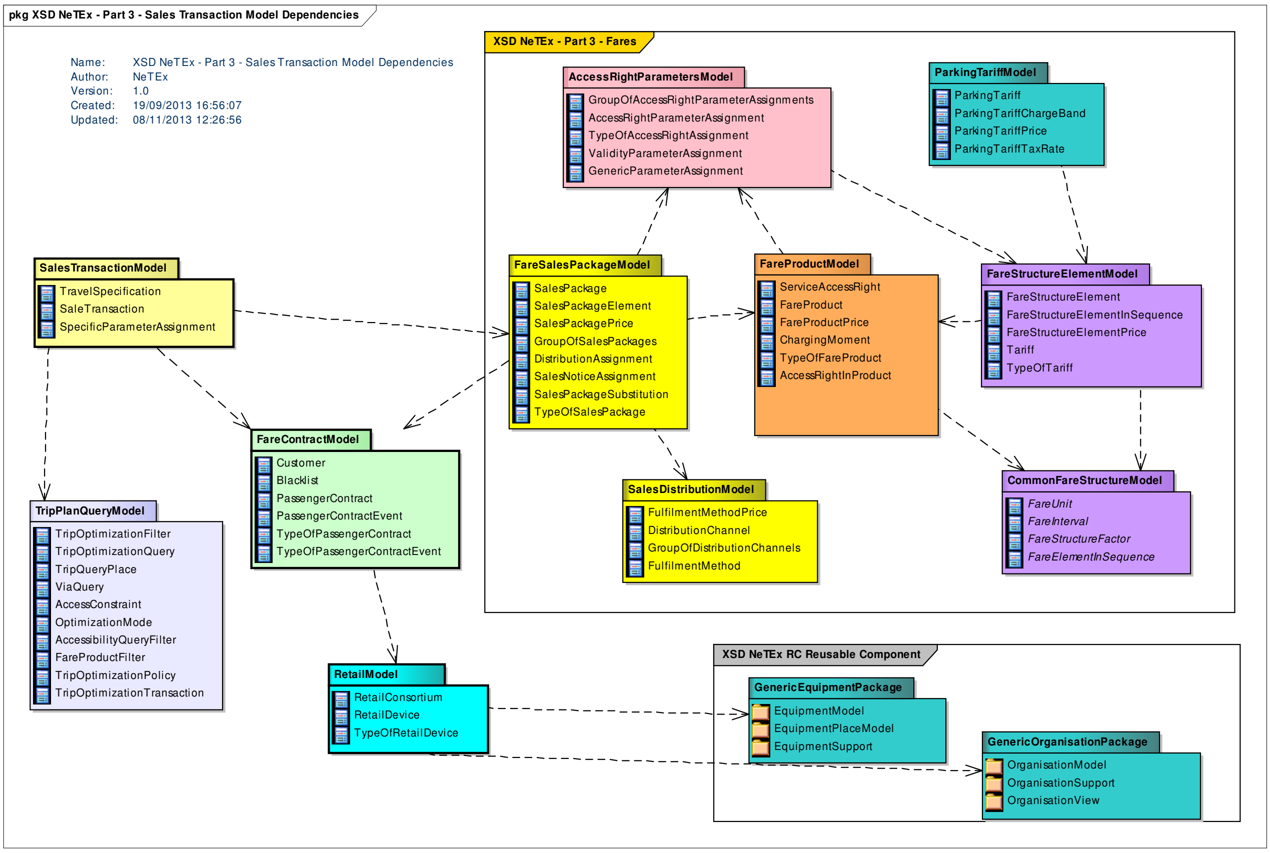


Figure 286 — Sales Transaction Model Dependencies

9.2 Sales Transaction Frame – Conceptual MODEL

The elements of the SALES TRANSACTION MODEL and the PASSENGER CONTRACT MODEL can be grouped with a SALES TRANSACTION FRAME, which holds a coherent set of fare sales related elements for data exchange. It can be used to exchange descriptiosn of customers and their purchases. See VERSION FRAME in the NeTEx Framework section for general concepts relating to version frames.

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 refrenced by the sales transactions. The components of a SALES TRANSACTION FRAME are described in detail in the following sections.

[TO DO ADD TO CONCEPTUAL MODEL?]

Figure 287 — Sales Transaction Frame – Conceptual MODEL (UML)

9.2.1 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.

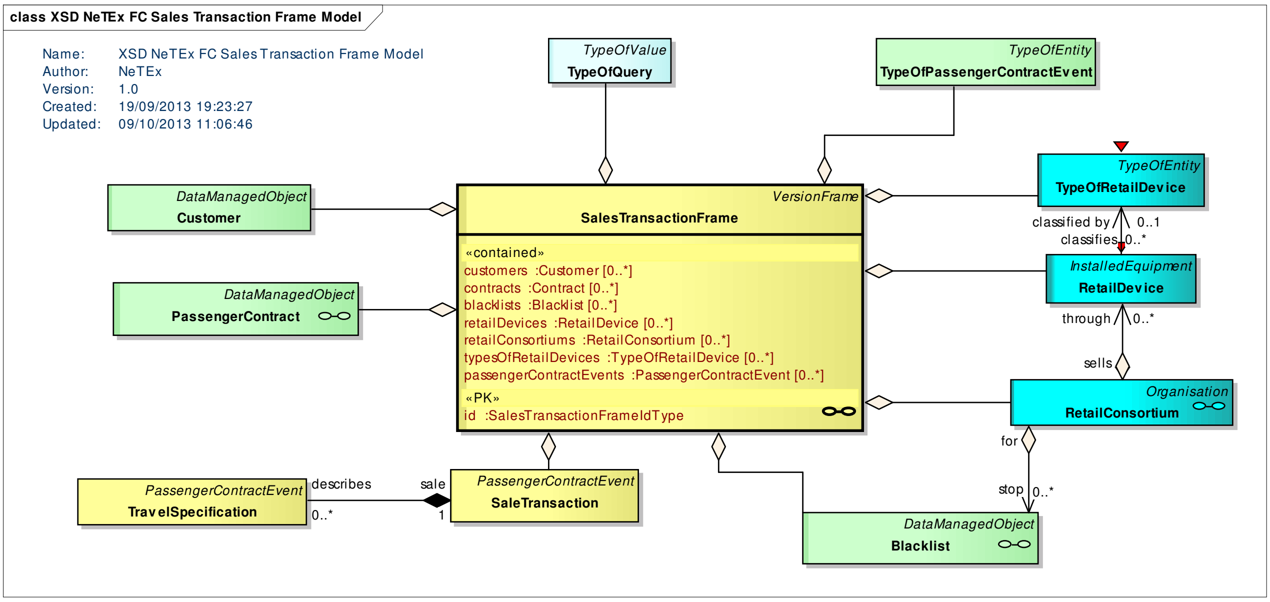


Figure 288 — Sales Transaction Frame – Physical Model Detail (UML)

9.2.2 Sales Transaction Frame – Attributes and XSD

9.2.2.1 SalesTransactionFrame – Model Element

A set of SALES TRANSACTION data elements (SALES TRANSACTIONs, etc.) to which the same VALIDITY CONDITIONs have been assigned.

Table 243 – *SalesTransactionFrame* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*VersionFrame*	::>	SALES TRANSACTION FRAME inherits from VERSION FRAME.
«PK»	***id***	*SalesTransactionFrame-IdType*	1:1	Identifier of SALES TRANSACTION FRAME.
“cntd»	***retail-Consortiums***	*RetailConsortium*	0:*	RETAIL CONSORTIUMs in SALES TRANSACTION FRAME.
“cntd»	***retailDevices***	*RetailDevice*	0:*	RETAIL DEVICEs in SALES TRANSACTION FRAME.
“cntd»	***customers***	*Customer*	0:*	CUSTOMERs in SALES TRANSACTION FRAME.
“cntd»	***passenger-Contracts***	*PassengerContract*	0:*	PASSENGER CONTRACTs in SALES TRANSACTION FRAME.
“cntd»	***blacklists***	*Blacklist*	0:*	BLACKLISTs in SALES TRANSACTION FRAME.
“cntd»	***travel-Specifications***	*TravelSpecification*	0:*	TRAVEL SPECIFCATIONs in SALES TRANSACTION FRAME.

“cntd»	***sales-Transactions***	*SalesTransaction*	0:*	SALES TRANSACTIONs in SALES TRANSACTION FRAME.

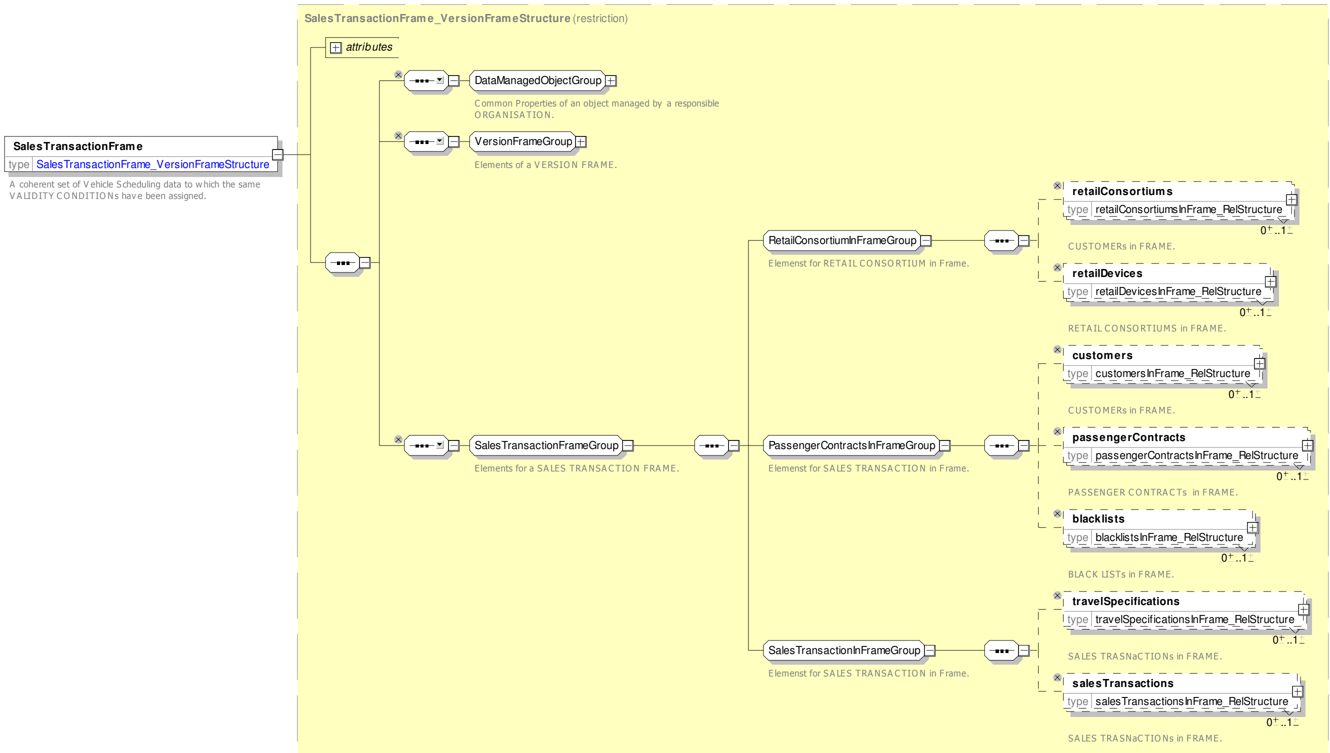


Figure 289 — *SalesTransactionFrame* — XSD

9.2.3 Fare Contract

9.2.3.1 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 EVENTs that take place such as booking or collecting a ticket, obtaining a refund, etc are subject to this contract.

NeTEx provides a minimal representation of this PASSENGER CONTRACT which can be used to organise and track ticket purchases. A TRAVEL SPECIFICATION is a specialisation of CONTRACT EVENT used to record a ticket PURCHASE.

CONTRACTs and CUSTOMERs may be placed on a BLACKLIST to identify them to systems for control purposes.

9.2.3.2 Passenger 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 PASSENGER CONTRACT entity. Various types of PASSENGER CONTRACTs may exist: single classical fare product, discount contract, identified card allowing the purchase of monthly passes, etc.

A PASSENGER CONTRACT is immaterial and is in principle materialised on a particular TRAVEL DOCUMENT. However, this relationship is not one-to-one:

— only a subset of a PASSENGER CONTRACT may be stored on a TRAVEL DOCUMENT;

— PASSENGER CONTRACT may be not (yet) stored on a TRAVEL DOCUMENT (e.g. in case of booking via Internet);

— a PASSENGER 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 PASSENGER CONTRACT may be replaced (in case of loss, failure, etc.);

— the same TRAVEL DOCUMENT may, in some cases, contain several PASSENGER CONTRACTs.

— This variability necessitates making both concepts independent.

9.2.3.3 Customers

Any PASSENGER 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 PASSENGER CONTRACT account. Such concepts, dealing with payment procedures, are not covered by the reference model.

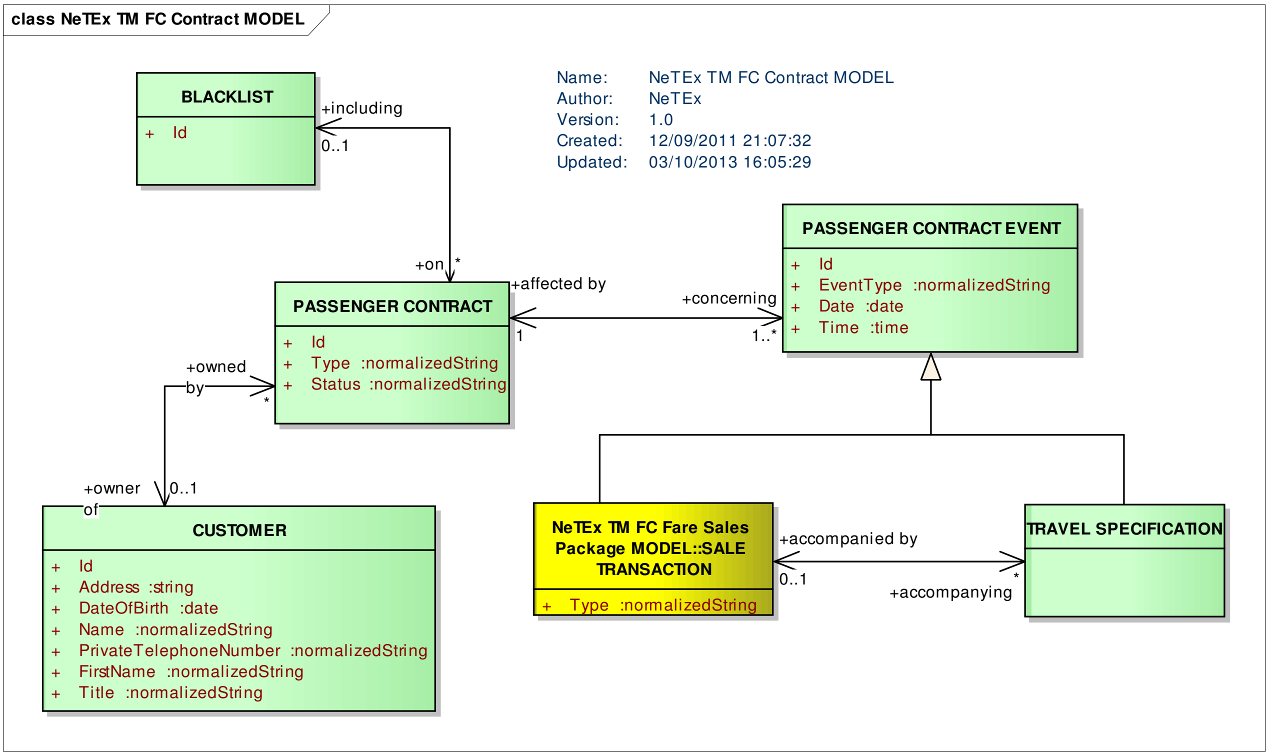


Figure 290 — Fare Contract – Conceptual Model

9.2.3.4 Fare Contract – Physical model

The following figure shows the physical model for FARE CONTRACTs.

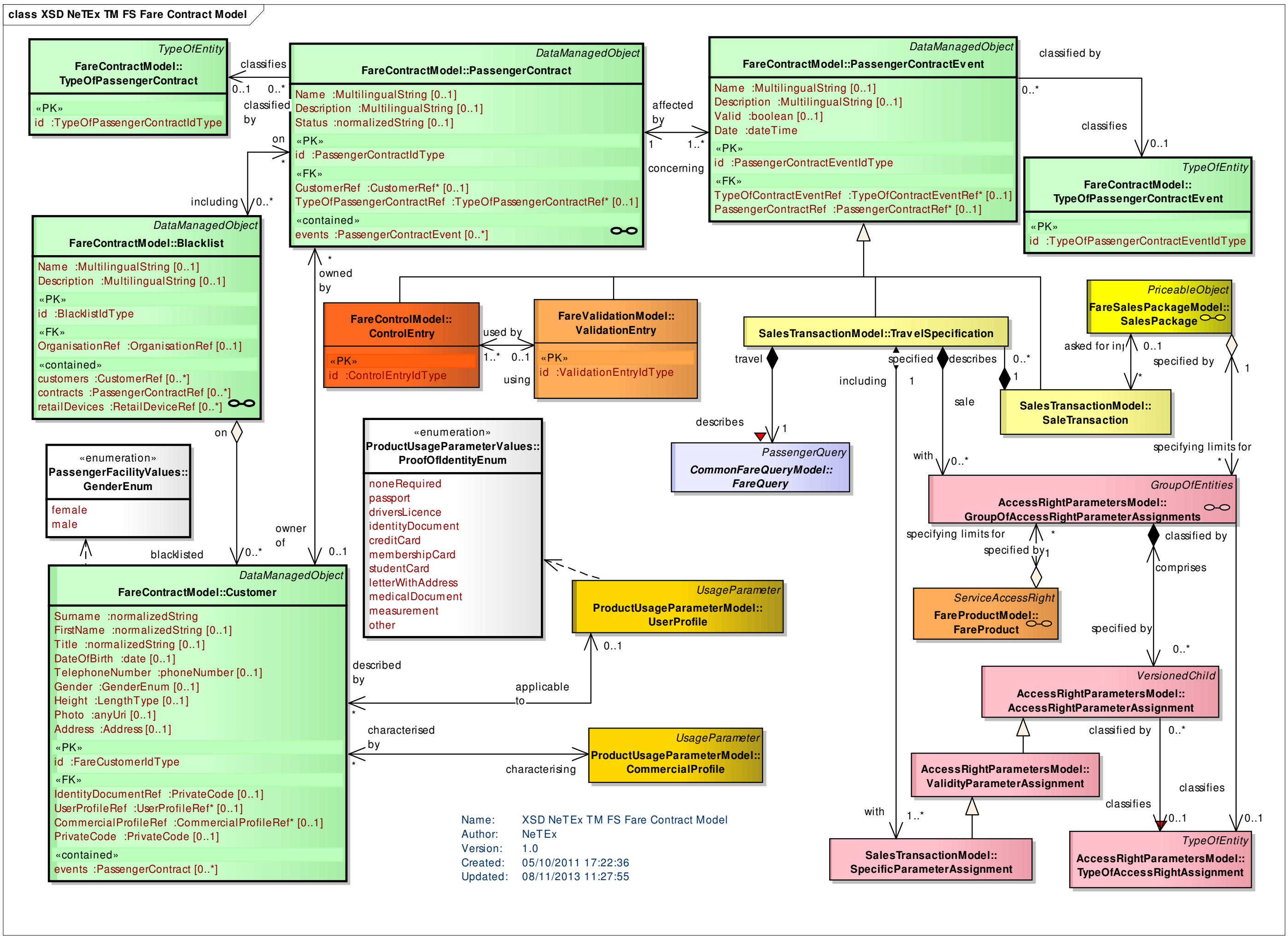


Figure 291 — Fare Contract – Physical Model

9.2.3.5 Fare Contract – Attributes and XSD

9.2.3.5.1 Customer – Model Element

An identified person or organisation involved in a fare process. There may be a PASSENGER CONTRACT between the CUSTOMER and the OPERATOR or the AUTHORITY ruling the consumption of services.

Table 244 – *Customer* – Element

Classifi-cation	Name	Type	Cardin ality	Description
::>	::>	*DataManagedObject*	::>	CUSTOMER inherits from DATA MANAGED OBJECT.
«PK»	***id***	*FareCustomerIdType*	1:1	Identifier of a CUSTOMER.
	Surname	*xsd:normalizedString*	1:1	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.
	Telephone-Number	*phoneNumber*	0:1	Telephone number of a CUSTOMER.
	Gender	*GenderEnum*	0:1	Gender of CUSTOMER.
	Height	*LengthType*	0:1	Height of CUSTOMER.
	Photo	*xsd:anyUri*	0:1	Photo of CUSTOMER.
	PostalAddress	*Address*	0:1	Postal address of a CUSTOMER.
«FK»	***Identity-DocumentRef***	*PrivateCode*	0:1	A document that identifies the CUSTOMER.
«FK»	***UserProfileRef***	*UserProfileRef*	0:1	Reference to a USER PROFILE to which the customer corresponds.
«FK»	***Commercial-ProfileRef***	*CommercialProfileRef*	0:1	Reference to a COMMERCIAL PROFILE to which the customer corresponds
«cntd»	***passenger-Contracts***	*PassengerContract*	0:*	PASSENGER CONTRACTs for the CUSTOMER.

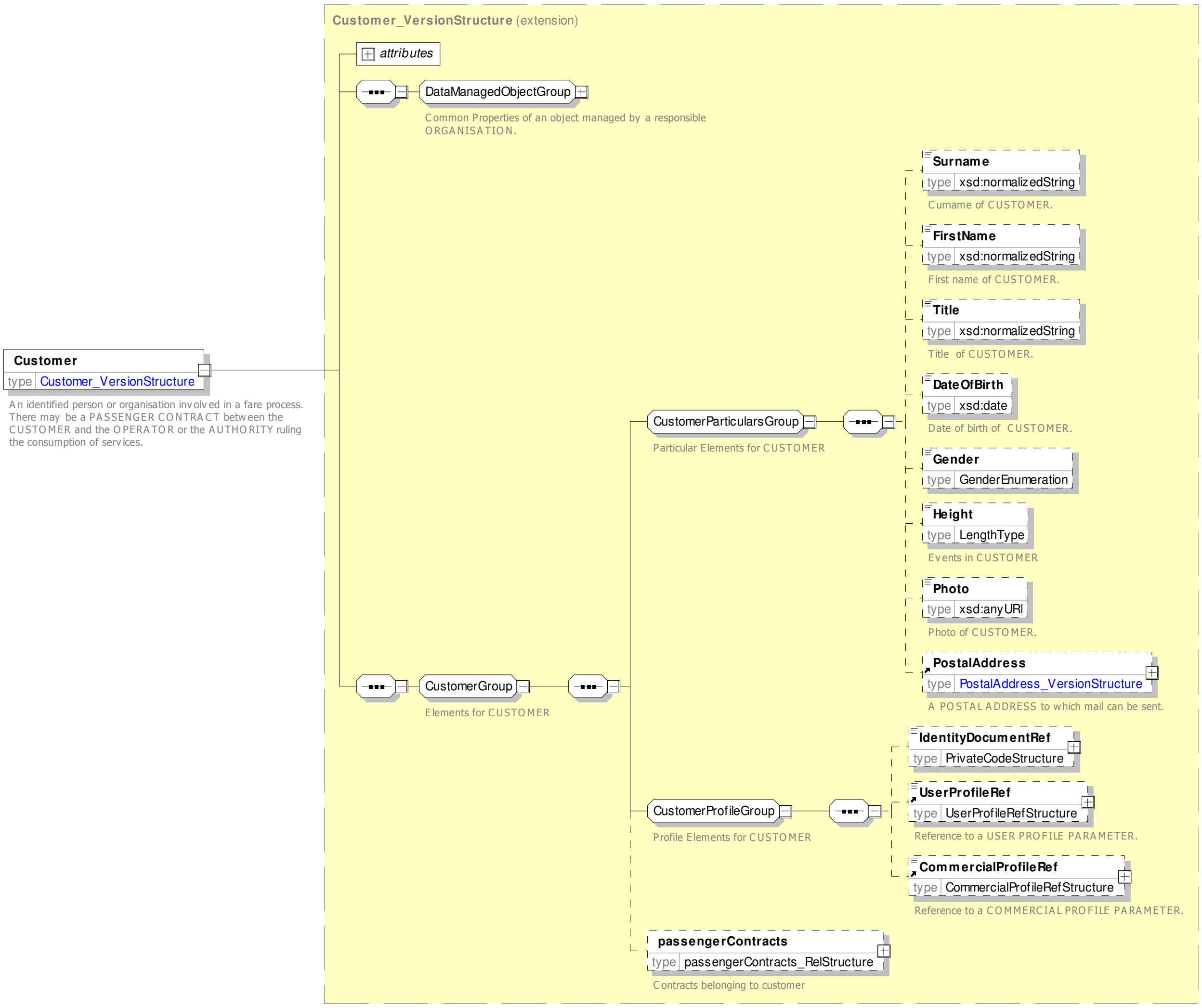


Figure 292 — *Customer* — XSD

9.2.3.5.2 PassengerContract – Model Element

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

Table 245 – *PassengerContract* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	PASSENGER CONTRACT inherits from DATA MANAGED OBJECT.
«PK»	***id***	*PassengerContractIdType*	1:1	Identifier of a PASSENGER CONTRACT.
	Name	*MultilingualString*	0:1	Name of PASSENGER CONTRACT.
	Description	*MultilingualString*	0:1	Description of PASSENGER 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 PASSENGER CONTRACT.
«FK»	***CustomerRef***	*CustomerRef*	0:1	Reference to the CUSTOMER for the PASSENGER CONTRACT.
«FK»	***TypeOf-Passenger-ContractRef***	*TypeOfPassenger-ContractRef*	0:1	Type of PASSENGER CONTRACT.
“cntd»	***passenger-ContractEvents***	*PassengerContractEvent*	0:*	PASSENGER CONTRACT EVENTs making up the PASSENGER CONTRACT.

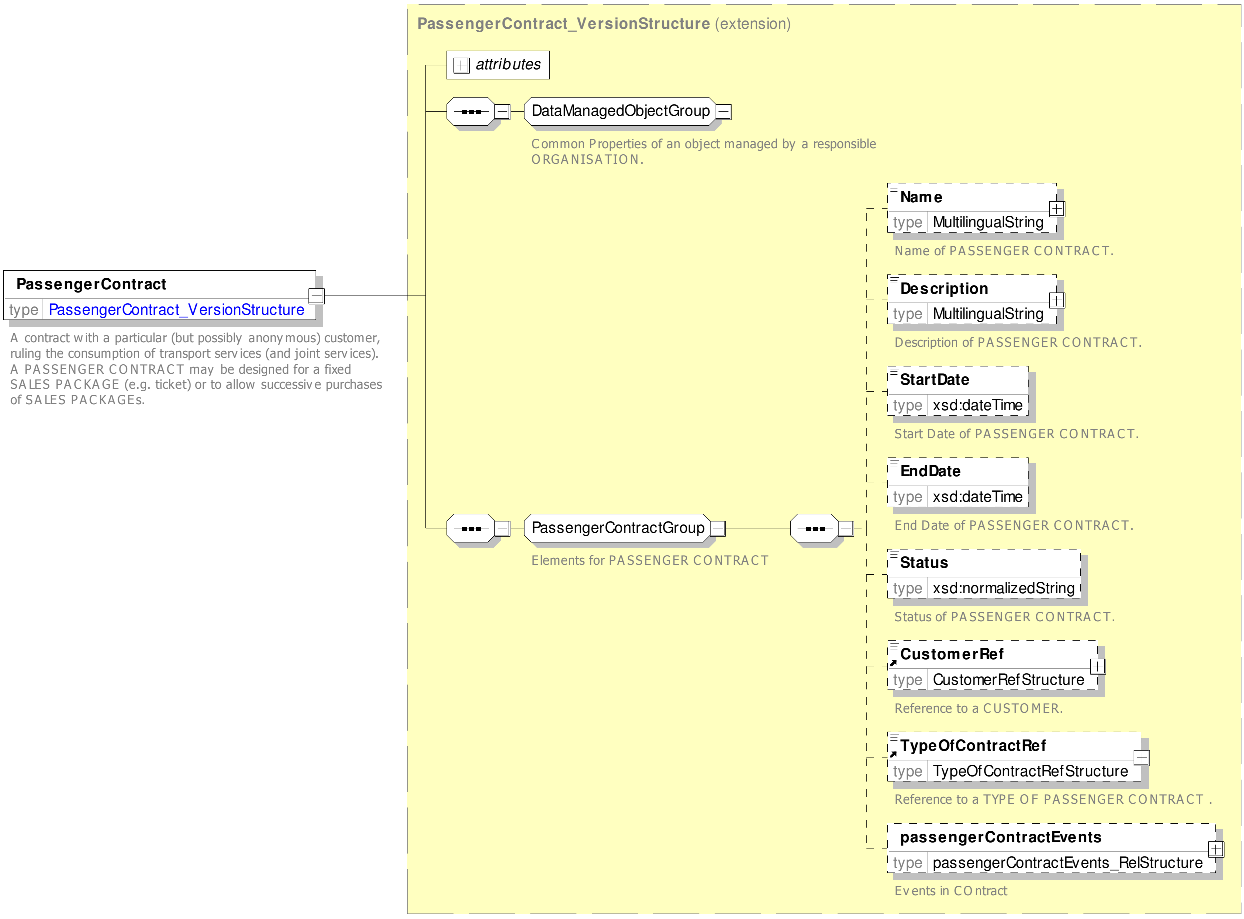


Figure 293 — *PassengerContract* — XSD

9.2.3.5.3 PassengerContractEvent – Model Element

A log entry describing an event referring to the life of a PASSENGER CONTRACT: initial contracting, sales, validation entries,, etc. A subset of a PASSENGER CONTRACT EVENT is often materialised on a TRAVEL DOCUMENT.

Table 246 – *PassengerContractEvent* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*DataManagedObject*	::>	PASSENGER CONTRACT EVENT inherits from DATA MANAGED OBJECT. See NeTEx Part1.
«PK»	***id***	*PassengerContract-EventIdType*	1:1	Identifier of PASSENGER CONTRACT EVENT.
	Name	*MultilingualString*	0:1	Name of PASSENGER CONTRACT EVENT
	Description	*MultilingualString*	0:1	Description of PASSENGER CONTRACT EVENT.
	IsValid	*xsd:boolean*	0:1	Whether the PASSENGER CONTRACT EVENT is valid or not.
	Date	*xsd:dateTime*	1:1	Date and time of PASSENGER CONTRACT EVENT.
«FK»	***TypeOfPassenger-ContractEventRef***	*TypeOfPassenger-ContractEventRef*	0:1	Type of PASSENGER CONTRACT EVENT.
«FK»	***Passenger-ContractRef***	*PassengerContractRef*	0:1	Reference to a PASSENGER CONTRACT to which the PASSENGER CONTRACT EVENT applies.

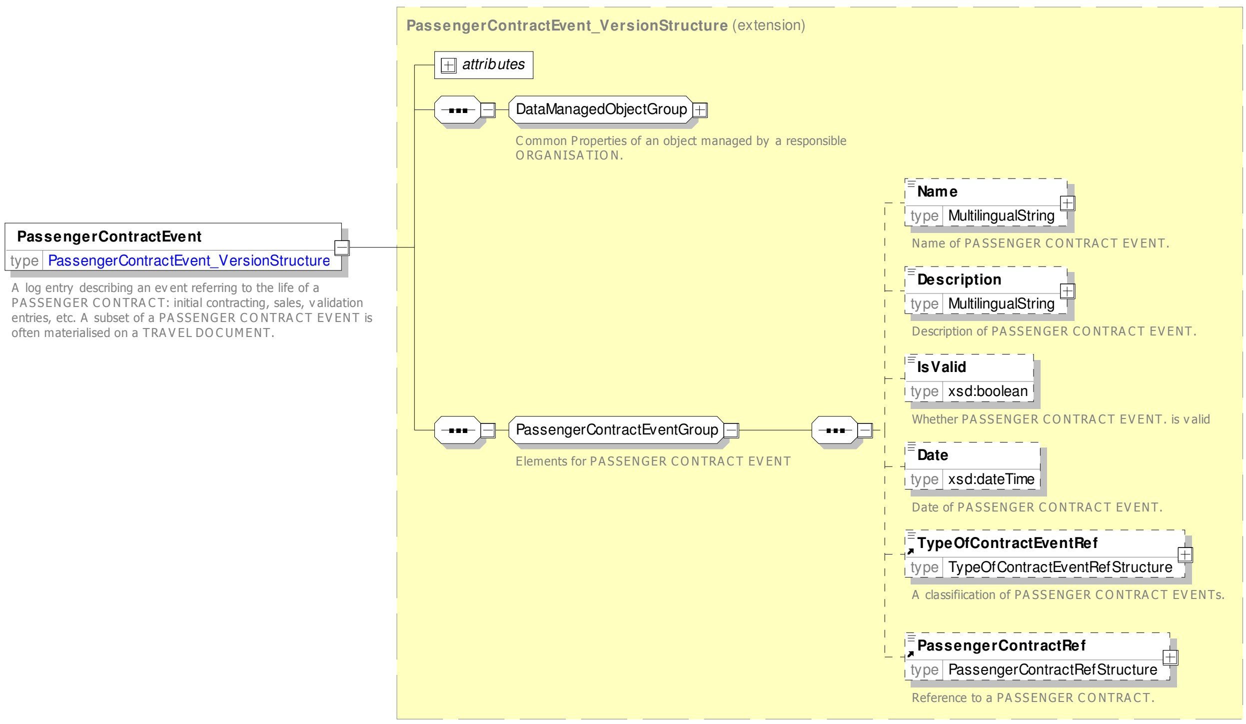


Figure 294 — *PassengerContractEvent* — XSD

9.2.3.5.4 TypeOfPassengerContract – Model Element

A classification of PASSENGER CONTRACT.

Table 247 – *TypeOfPassengerContract* – Element

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

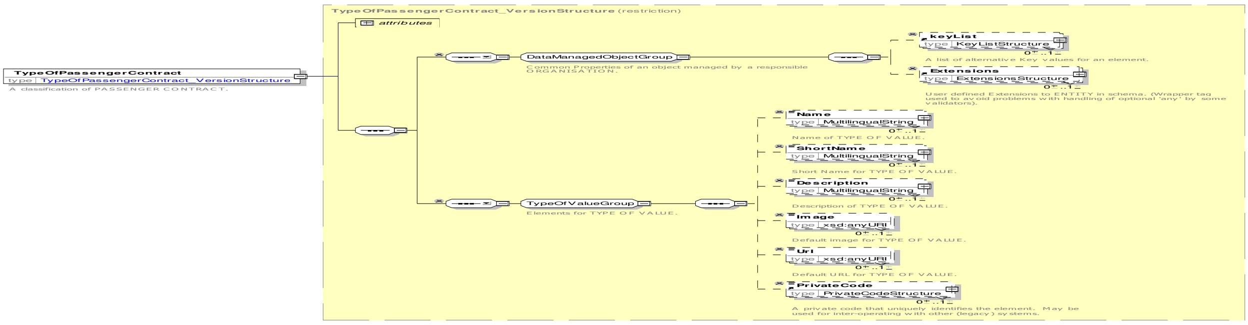


Figure 295 — *TypeOfPassengerContract* — XSD

9.2.3.5.5 TypeOfPassengerContractEvent – Model Element

A classification of PASSENGER CONTRACT EVENTs.

Table 248 – *TypeOfPassengerContractEvent* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfEntity*	::>	TYPE OF PASSENGER CONTRACT EVENT inherits from TYPE OF ENTITY. See NeTEx Part1.
«PK»	***id***	*TypeOfPassenger-ContractEventIdType*	1:1	Identifier of TYPE OF PASSENGER CONTRACT EVENT.

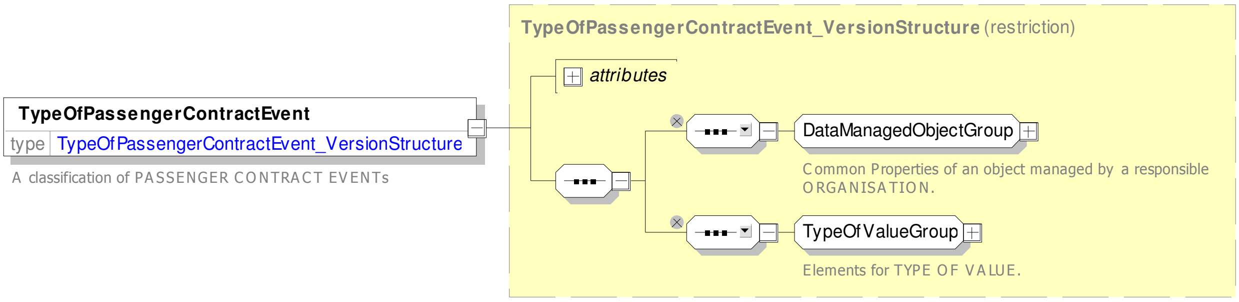


Figure 296 — *TypeOfPassengerContractEvent* — XSD

9.2.3.5.6 Blacklist – Model Element

A list of identified TRAVEL DOCUMENTs or PASSENGER 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.

Table 249 – *Blacklist* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	BLACKLIST inherits from DATA MANAGED OBJECT.
«PK»	***id***	*BlacklistIdType*	1:1	Identifier of a BLACKLIST.
	Name	*MultilingualString*	0:1	Name of BLACK LIST.
	Description	*MultilingualString*	0:1	Description of BLACK LIST.
«FK»	***OrganisationRef***	*OrganisationRef*	0:1	ORGANISATION associated with BLACK LIST.
“cntd»	***customers***	*CustomerRef*	0:*	CUSTOMERs on the BLACK LIST.
“cntd»	***passenger-Contracts***	*PassengerContractRef*	0:*	PASSENGER CONTRACTs on the BLACK LIST.
“cntd»	***retailDevices***	*RetailDevice*	0:*	RETAIL DEVICEs on the BLACK LIST.

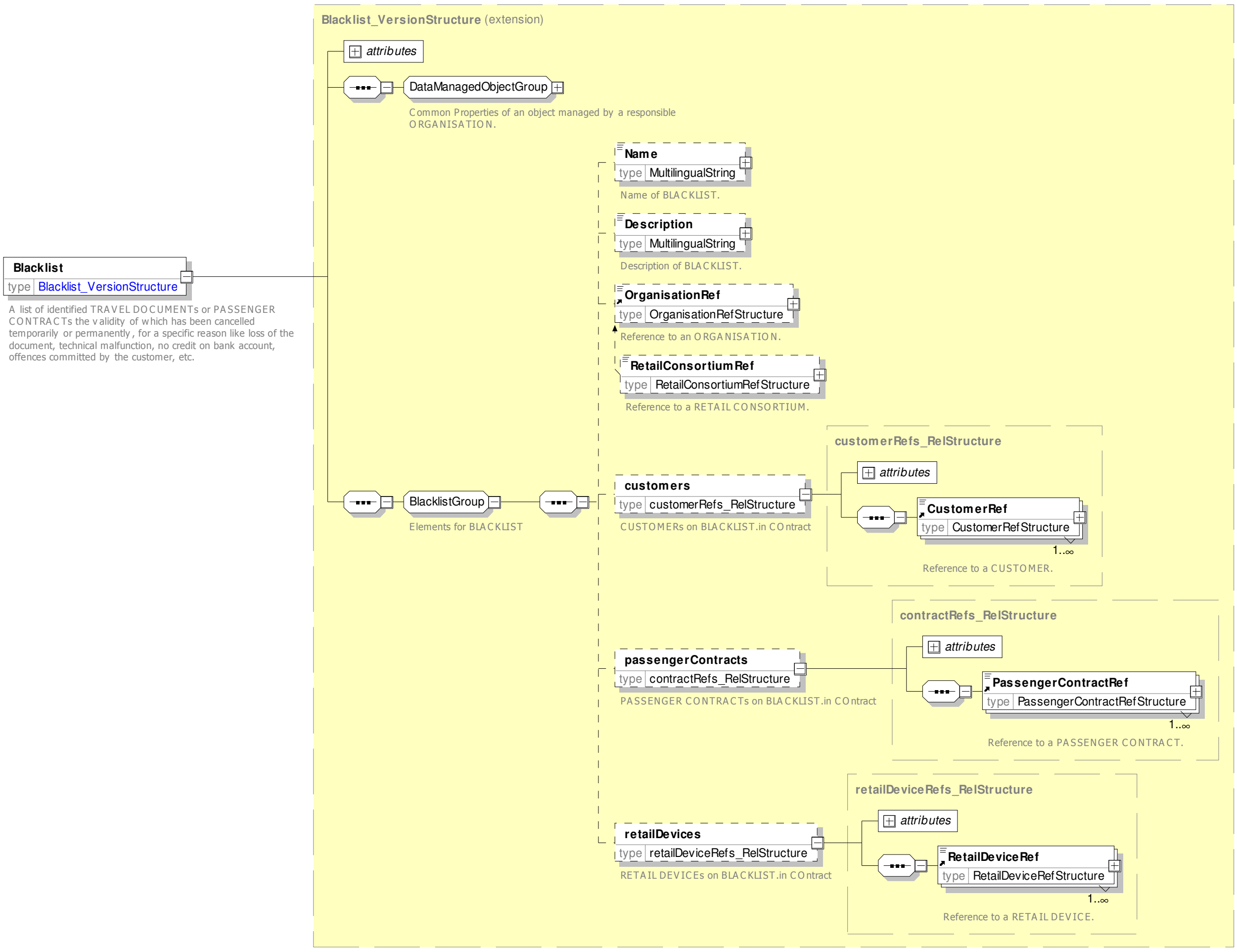


Figure 297 — *Blacklist* — XSD

9.2.3.6 Fare Contract – XML Examples.

9.2.3.6.1 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"/>
        <passengerContracts>
            <PassengerContract version="any" id="xpl::Oyster-12222">
                <Name> Registered Oyster card Oyster-122222 with Transactions </Name>
                <CustomerRef version="any" ref="xpl:Customer:122222"/>
                <passengerContractEvents>
                    <SalesTransaction version="any" etc., etc.

                </passengerContractEvents>
            </PassengerContract>
        </passengerContracts>
    </Customer>
</customers>
```

9.2.3.6.2 Fare Contract: XML Example of Blacklist

The following code fragment shows a BLACKLIST with a two PASSENGER CONTRACTs on it .

For EXAMPLE:

```
<blacklists>
    <Blacklist version="any" id="tfl::StolenCards">
    <Name>Stolen card list</Name>
    <passengerContracts>
        <PassengerContractRef version="any" ref="xpl::Oyster-12222"/>
        <PassengerContractRef version="any" ref="xpl::Oyster-75632"/>
    </passengerContracts>
    </Blacklist>
</blacklists>
```

9.2.4 Retail

9.2.4.1 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 CONTRCTs.

[TO DO ANY CONCEPTUAL]

Figure 298 — Retail – Conceptual Model

9.2.4.2 Retail – Physical model

The following figure shows the physical model for RETAIL.

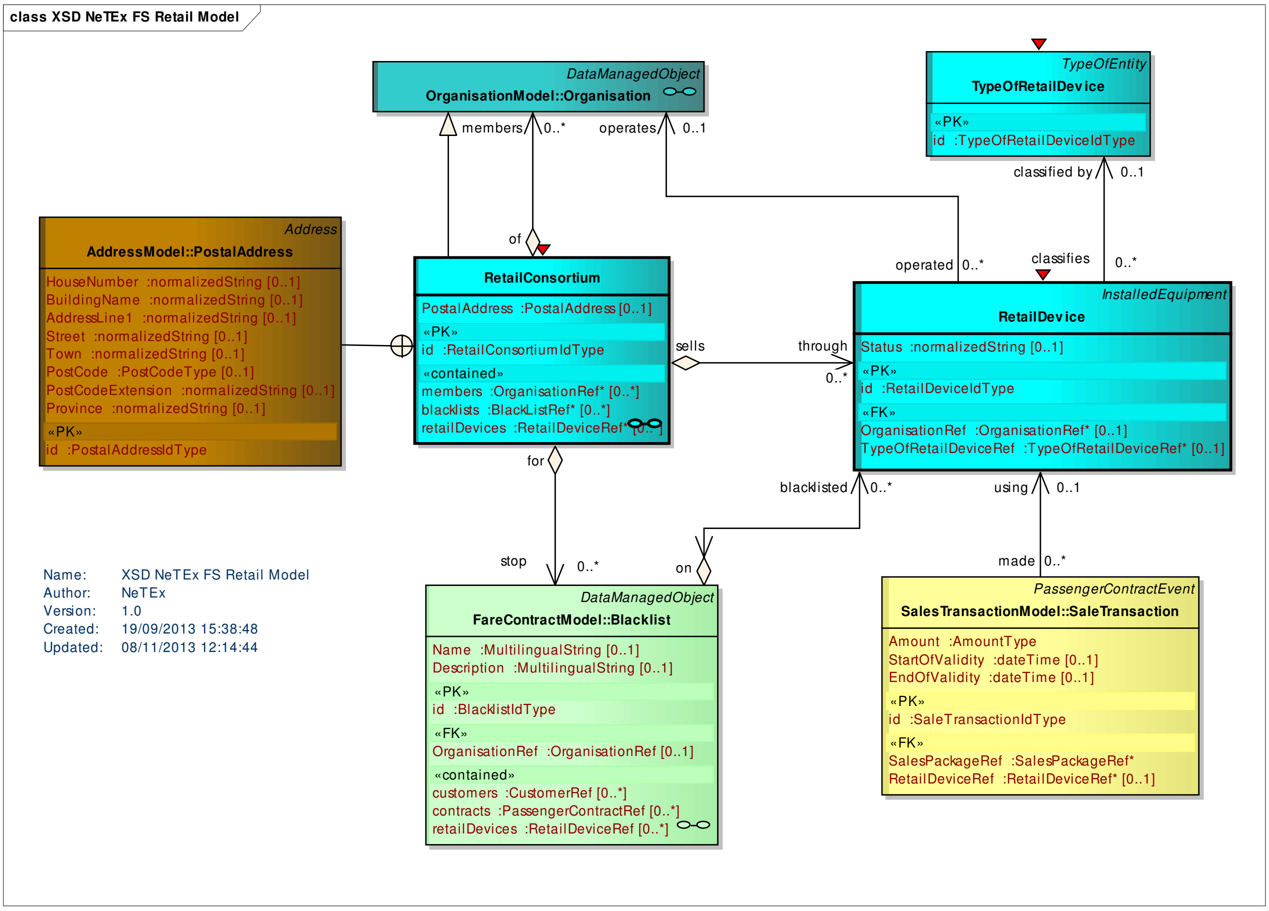


Figure 299 — Retail– Physical Model

9.2.4.3 Retail – Attributes and XSD

9.2.4.3.1 Retail Consortium – Model Element

A group of ORGANISATIONs formally incorporated a retail ORHGANISATION who are retailers of fare products and who share common security processes.

Table 250 – *RetailConsortium* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Organisation*	::>	RETAIL CONSORTIUM inherits from ORGANISATION. See NeTEx Part1.
«PK»	***id***	*RetailConsortiumIdType*	1:1	Identifier of a RETAIL CONSORTIUM.
	PostalAddress	*PostalAddress*	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.

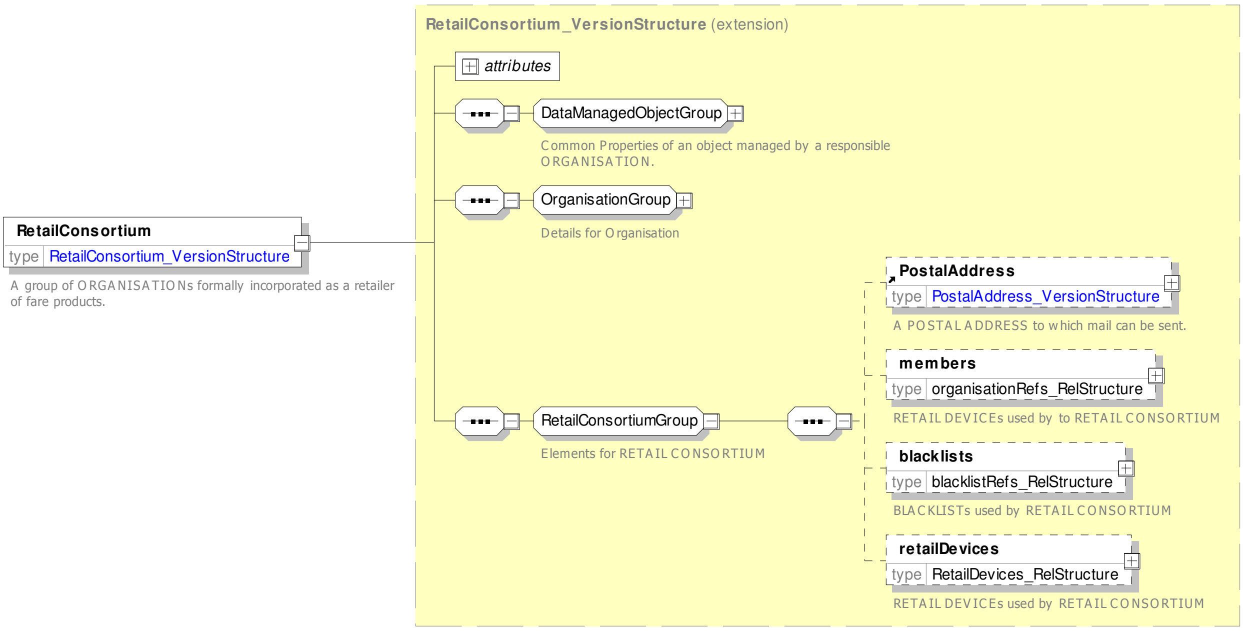


Figure 300 — *RetailConsortium* — XSD

9.2.4.3.2 Retail Device – Model Element

A retail device used to sell fare products. Can be used to record fulfilment.

Table 251 – *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.

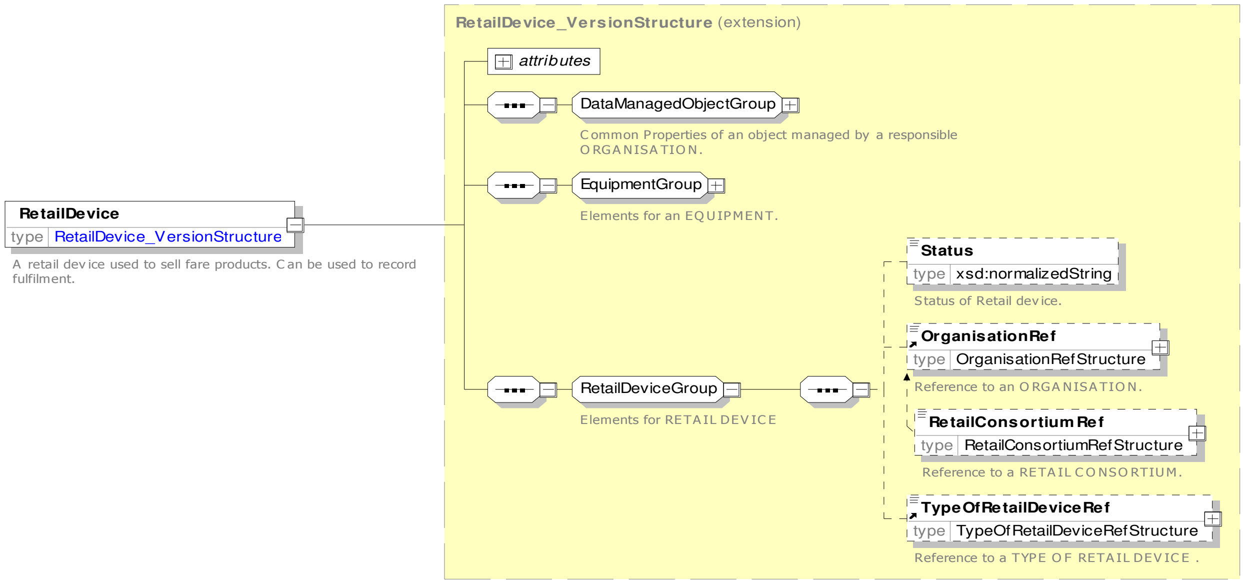


Figure 301 — *RetailDevice* — XSD

9.2.4.3.3 Type of Retail Device – Model Element

A classification of RETAIL DEVICEs.

Table 252 – *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.

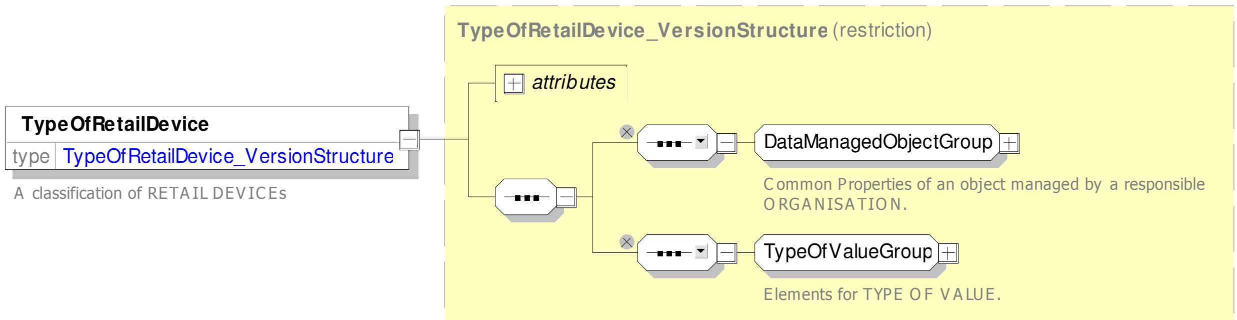


Figure 302 — *TypeOfRetailDevice* — XSD

9.2.4.4 Retail Consortium – XML examples

9.2.4.4.1 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:RetailConsortium:Oyster">
        <Name>Oyster plc</Name>
        <ShortName>Oyster</ShortName>
        <members>
            <OperatorRef version="any" ref="tfl:Operator:lbsl"/>
            <OperatorRef version="any" ref="tfl:Operator:lul"/>
            <OperatorRef version="any" ref="tfl:Operator:dlr"/>
            <OperatorRef version="any" ref="tfl:Operator:Emirates"/>
            <OperatorRef version="any" ref="nr:Operator:nr"/>
Etc., etc .
        </members>
        <retailDevices>
            <RetailDevice version="any" id="tfl:RetailDevice:4421">
                <Name>Ticket machine 2 at KingsX</Name>
            </RetailDevice>
            <RetailDevice version="any" id="tfl:RetailDevice:4422">
                <Name>Ticket machine3 at KingsX</Name>
            </RetailDevice>
Etc., etc .
        </retailDevices>
    </RetailConsortium>
</retailConsortiums>
```

9.2.5 Sales Transaction

9.2.5.1 Sales Transaction – Conceptual model

The SALES TRANSACTION model records product sales either to identified CUSTOMERs as events of a PASSENGER CONTRACT, or as anonymous transactions.

Each SALES TRANSACTION describes the purchase of a SALES 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.

9.2.5.2 Pre-consumption Specification

Numerous 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).

An action of specifying some parameters to be used by a customer when purchasing (or using) a generic access right is described by the TRAVEL SPECIFICATION entity. A TRAVEL SPECIFICATION is a use of PASSENGER CONTRACT EVENT, which implies the following properties:

— a TRAVEL SPECIFICATION refers to an individual PASSENGER CONTRACT;
— a TRAVEL SPECIFICATION is timed by the ‘date’ and ‘time’ attributes of PASSENGER CONTRACT EVENT;
— a TRAVEL SPECIFICATION may concern an identified CUSTOMER, through the relationship between PASSENGER CONTRACT EVENT and CUSTOMER;
— the parameters assigned by a TRAVEL SPECIFICATION are, in most cases, stored on an individual TRAVEL DOCUMENT, through the relationship between PASSENGER CONTRACT EVENT and TRAVEL DOCUMENT.

A TRAVEL SPECIFICATION often accompanies a SALE TRANSACTION, occurring at the same time or soon after.

A TRAVEL SPECIFICATION shall include at least one parameter assignment. Such an assignment is described by the SPECIFIC PARAMETER ASSIGNMENT entity, which is a subtype of VALIDITY PARAMETER ASSIGNMENT which is, in turn, subtype of ACCESS RIGHT PARAMETER ASSIGNMENT.

On the occasion of a SALE TRANSACTION, the customer has purchased a certain SALES PACKAGE, composed of generic access rights (e.g. a week pass based on a zone counting system). The specified parameter (e.g. the destination zone) is attached via SPECIFIC PARAMETER ASSIGNMENT to the concerned individual PASSENGER CONTRACT (i.e. the contract owning the concerned package) with reference to 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 a different nature:

— 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 sold FARE PRODUCT comprising this element shall comply with this limitation;
— SPECIFIC PARAMETER ASSIGNMENT assigns a parameter to only one particular instance of SALES PACKAGE purchased by a customer, with reference to a generic access right included in the package (e.g. the customer intends 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.

TO DO Any conceptual Model?

9.2.5.3 Sales Transaction – Physical model

The following figure shows the physical model for FARE CONTRACTs.

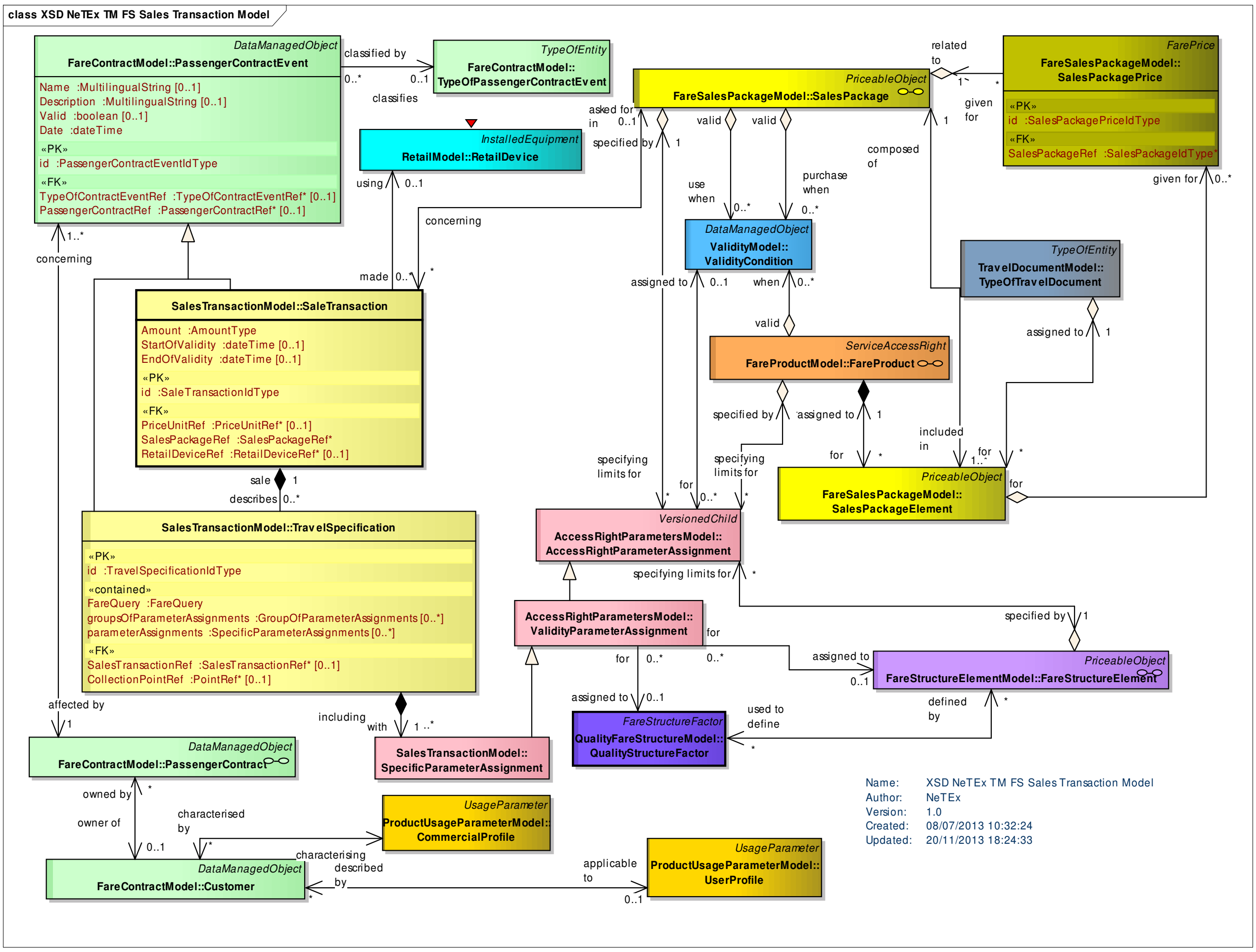


Figure 303 — Sales Transaction – Physical Model

9.2.5.4 Sales Transaction – Attributes and XSD

9.2.5.4.1 TravelSpecification – Model Element

The recording of a specification by a customer of parameters giving details of an intended consumption (e.g. origin and destination of a travel).

Table 253 – *TravelSpecification* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerContractEvent*	::>	TRAVEL SPECIFICATION inherits from PASSENGER CONTRACT EVENT
«PK»	***id***	*TravelSpecification-IdType*	1:1	Identifier of TRAVEL SPECIFICATION.
“cntd»	***FareQuery***	*FareQuery*	1:1	Travel specification as FARE QUERY parameters.
“cntd»	***groupsOf-Parameter-***	*GroupOfParameter-Assignments*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs relating to TRAVEL SPECIFICATION.

	Assignments			
“cntd»	***Parameter-Assignments***	*SpecificParameter-Assignments*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs relating to TRAVEL SPECIFICATION.
«FK»	***Sales-TransactionRef***	*SalesTransactionRef*	0:1	SALES TRANSACTION for TRAVEL SPECIFICATION.
«FK»	***Collection-PointRef***	*PointRef*	0:1	Point at which ticket will be collected.

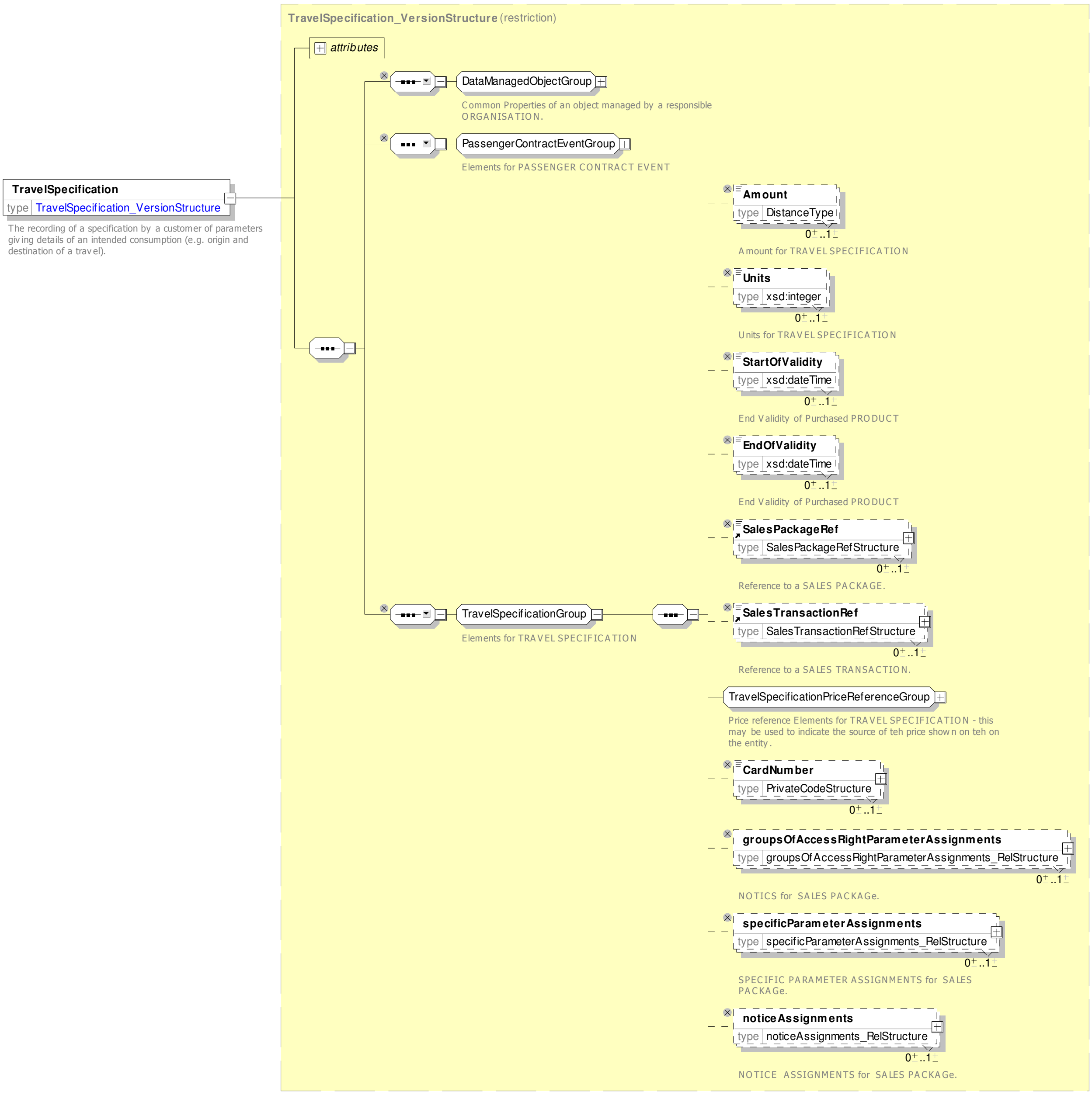


Figure 304 — *TravelSpecification* — XSD

9.2.5.4.2 SaleTransaction – Model Element

A SALE OF a FIXED PACKAGE or a SALE OF a RELOADABLE PACKAGE.

Table 254 – *SaleTransaction* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*PassengerContractEvent*	::>	SALES TRANSACTION inherits from PASSENGER CONTRACT EVENT.
«PK»	***id***	*SaleTransactionIdType*	1:1	Identifier of SALES TRANSACTION.
	Amount	*AmountType*	1:1	Amount of SALES TRANSACTION.
	PriceUnitRef	*PriceUnitRef*	0:1	PRICE UNIT of SALES TRANSACTION.
	StartOfValidity	*xsd:dateTime*	0:1	Stat time of validity of product.
	EndOfValidity	*xsd:dateTime*	0:1	End time of validity of product.
«FK»	***SalesPackageRef***	*SalesPackageRef*	1:1	Reference to a SALES PACKAGE of which this is a TRANSACTION.

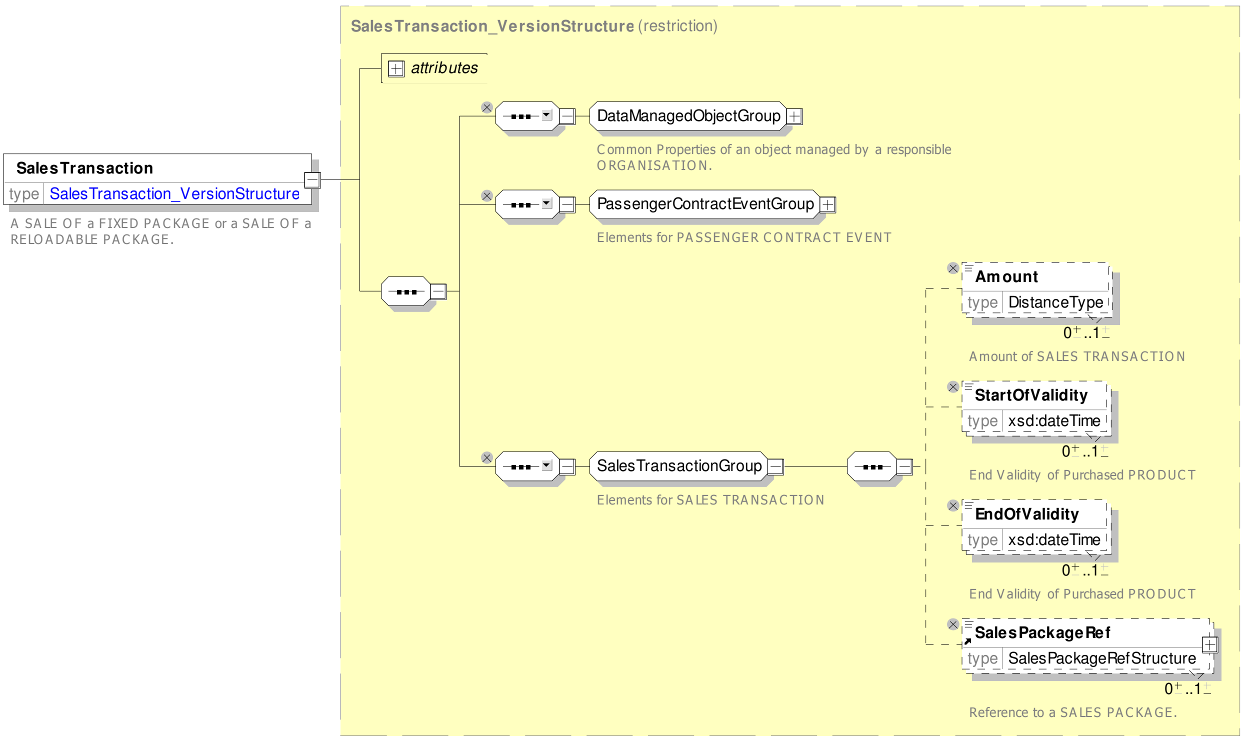


Figure 305 — *SaleTransaction* — XSD

9.2.5.4.3 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 255 – *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.

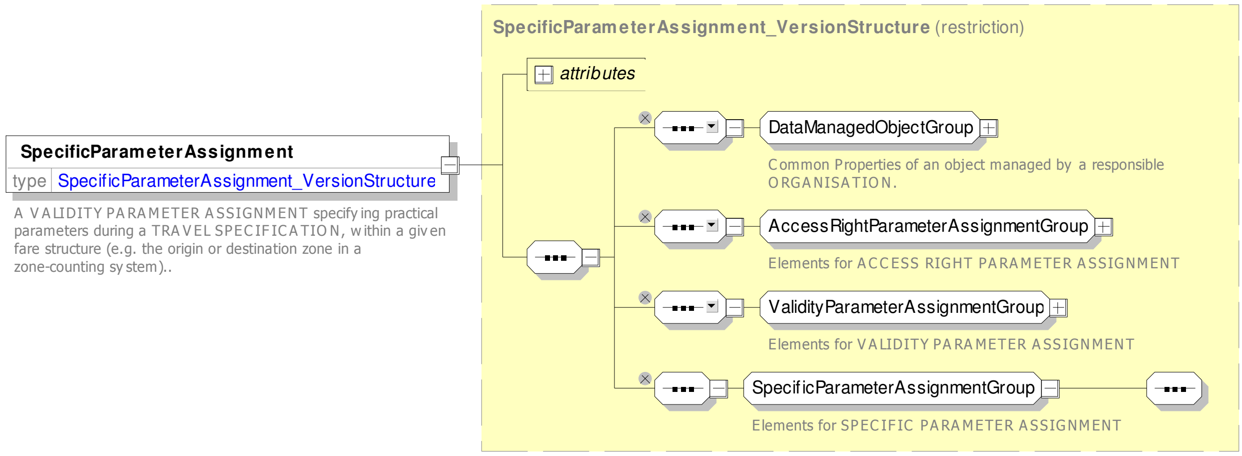


Figure 306 — *SpecificParameterAssignment* — XSD

9.2.5.5 Sales Transaction – XML examples

9.2.5.5.1 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 SPECIFCATION, grouped using a PASSENGER COBTRACT.

For EXAMPLE:

```
<PassengerContract version="any" id="xpl:Contract:Anon001">
    <Name> Zone 1 Paper ticket for Cash </Name>
    <passengerContractEvents>
        <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>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::purchase"/>
            <Amount>4.50</Amount>
            <StartOfValidity>2013-07-08T01:07:00</StartOfValidity>
            <SalesPackageRef version="any" ref="tfl::Ticket"/>
            <SalesTransactionRef version="any" ref="xpl::Anon001@trans001@purchaseTicket@Ticket"/>
            <accessRightParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl::Anon001@trans001@01">
                    <ValidableElementRef version="any" ref="lul::metroTrip"/>
                </SpecificParameterAssignment>
            </accessRightParameterAssignments>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl::Anon001@trans001@01">
```

```xml
        <Name>Fare zones 1 only</Name>
        <Description of purchase?</Description>
        <FareStructureElementRef version="any" ref="tfl::Zone_1_only"/>
        <FareDemandFactorRef version="any" ref="tfl::anyTime"/>
        <for>
            <FrequencyOfUseRef version="any" ref="tfl::oneTrip"/>
            <UserProfileRef version="any" ref="tfl::adult"/>
        </for>
        <Scope>
            <ScheduledStopPointRef ref="tfl::Kings_Cross" version="any"/>
            <DistributionChannelRef version="any" ref="tfl::SelfServiceMachine"/>
        </Scope>
    </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"/>
        <for>
            <RoundTripRef version="any" ref="tfl::single"/>
        </for>
```

```
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>
    </passengerContractEvents>
</PassengerContract>
```

9.2.5.5.2 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:

```
<PassengerContract version="any" id="xpl:Contract:Anon002">
    <Name> Day Group Paper Ticket </Name>
    <passengerContractEvents>
        <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>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::purchase"/>
            <Amount>4.50</Amount>
            <Units>4</Units>
            <SalesPackageRef 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>
                    <for>
                        <GroupTicketRef version="any" ref="tfl::groupDayTicket"/>
                        <UserProfileRef version="any" ref="tfl::adult"/>
                    </for>
                    <Scope>
                        <ScheduledStopPointRef ref="tfl::Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any"
                        ref="tfl::TubeStationTicketOffices"/>
                    </Scope>
                    <FareStructureElementRef version="any" ref="tfl::Zones_1-6"/>
                    <FareDemandFactorRef version="any" ref="tfl::anyTime"/>
                </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>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::purchase"/>
            <Amount>1.70</Amount>
            <Units>10</Units>
            <SalesPackageRef 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>
                    <for>
                        <GroupTicketRef version="any" ref="tfl::groupDayTicket"/>
                        <UserProfileRef version="any" ref="tfl::child"/>
                    </for>
                    <Scope>
                        <ScheduledStopPointRef ref="tfl::Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any"
                            ref="tfl::TubeStationTicketOffices"/>
```

```
                    </Scope>
                    <FareStructureElementRef version="any" ref="tfl::Zones_1-6"/>
                    <FareDemandFactorRef version="any" ref="tfl::anyTime"/>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>
    </passengerContractEvents>
</PassengerContract>
```

9.2.5.5.3 Sales Transaction: XML Example of Card Transactions

The following code fragment shows a series of PASSENGER CONTRACT EVENTs on an anonymous Oyster card (i.e. PASSENGER 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 -->
<PassengerContract version="any" id="xpl::Oyster-12345">
    <Name> Anonymous Oyster card Oyster-12345 with Transactions </Name>
    <passengerContractEvents>
        <SalesTransaction version="any" id="xpl::Oyster-12345@trans001@purchaseNewCard">
            <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@purchaseNewCardAdultOysterCard">
            <Name>Adult Oyster Card</Name>
            <Date>2013-07-08T01:07:00</Date>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::purchase"/>
            <Amount>5.00</Amount>
            <StartOfValidity>2013-07-08T01:07:00</StartOfValidity>
            <SalesPackageRef version="any" ref="tfl::AdultOysterCard"/>
            <SalesTransactionRef version="any" ref="xpl::Oyster-12345@trans001@purchaseNewCard"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl::Oyster-12345@trans001@01">
                    <for>
                        <UserProfileRef version="any" ref="tfl::adult"/>
                    </for>
                    <Scope>
                        <DistributionChannelRef version="any"
                            ref="tfl::TubeStationTicketOffices"/>
                    </Scope>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

        <TravelSpecification version="any" id="xpl::Oyster-
12345@trans001@purchaseNewCard@TravelCardOnOyster">
            <Name>TravelCardOnOyster 7 Day Travel card for zones 1 and 2 </Name>
            <Date>2013-07-08T01:07:00</Date>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::purchase"/>
            <Amount>30.40</Amount>
            <SalesPackageRef version="any" ref="tfl::TravelCardOnOyster"/>
            <SalesTransactionRef version="any" ref="xpl::Oyster-12345@trans001@purchaseNewCard"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any"
                id="xpl::Oyster-12345@trans001@Zones_1-2">
                    <FareStructureElementRef version="any" ref="tfl::Zones_1-2"/>
                    <for>
                        <UserProfileRef version="any" ref="tfl::adult"/>
                    </for>
                </SpecificParameterAssignment>
                <SpecificParameterAssignment version="any" id="xpl::Oyster-12345@trans001@@1Week">
                    <Name> 1 week pass </Name>
                    <FareStructureElementRef version="any" ref="tfl::1WeekPass"/>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>
```

```xml
        <!--     Prepay £10 O N an Travel Card Oyster Card with -->
        <SalesTransaction version="any" id="xpl::Oyster-12345@trans002@OysterTopUp">
            <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@OysterTopUp@OysterTopUp">
            <Name>Add 10 Pay as you go</Name>
            <Date>2013-07-08T01:07:10</Date>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::purchase"/>
            <Amount>10</Amount>
            <SalesPackageRef version="any" ref="tfl::OysterTopUp"/>
            <SalesTransactionRef version="any" ref="xpl::Oyster-12345@trans002@OysterTopUp"/>
            <specificParameterAssignments>
                <SpecificParameterAssignment version="any" id="xpl::Oyster-12345@trans002@01">
                    <Name>Top up ay at Kings Cross </Name>
                    <Scope>
                        <DistributionChannelRef version="any" ref="tfl::SelfServiceMachine"/>
                    </Scope>
                </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>
            <TypeOfPassengerContractEventRef 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>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::yellowTouchIn"/>
            <Amount>8.40</Amount>
            <SalesPackageRef version="any" ref="tfl::PayAsYouGoFare"/>
            <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"/>
                    <for>
                        <UserProfileRef version="any" ref="tfl::adult"/>
                    </for>
                    <Scope>
                        <ScheduledStopPointRef ref="tfl::Kings_Cross" version="any"/>
                        <DistributionChannelRef version="any" ref="tfl::Validator"/>
                    </Scope>
                </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>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::yellowTouchIn"/>
            <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>
            <TypeOfPassengerContractEventRef version="any" ref="tfl::yellowTouchOut"/>
            <Amount>0.00</Amount>
            <SalesPackageRef version="any" ref="tfl::PayAsYouGoFare"/>
            <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>
```

```
                        <FareStructureElementRef version="any" ref="tfl::Zone_1_only"/>
                    <for>
                        <UserProfileRef version="any" ref="tfl::adult"/>
                    </for>
                    <FareDemandFactorRef version="any" ref="tfl::peak"/>
                </SpecificParameterAssignment>
                <SpecificParameterAssignment version="any" id="xpl::Oyster-12345@trans004@02
                    <FareStructureElementRef version="any" ref="tfl::metroTrip"/>
                    <Name>Touch out at Oxford Circus . </Name>
                    <for>
                        <RoundTripRef version="any" ref="tfl:RoundTrip:single"/>
                    </for>
                    <Scope>
                        <ScheduledStopPointRef ref="tfl::Oxford_Circus" version="any"/>
                    </Scope>
                </SpecificParameterAssignment>
            </specificParameterAssignments>
        </TravelSpecification>

    </passengerContractEvents>
</PassengerContract>
</salesContracts>
```

Annex A
(informative)

Example of fares described with NeTEx

A.1 Introduction

[TO DO]

A.2 Simple examples

[TO DO]

A.3 Urban fares

[TO DO]

A.4 Rail fares

[TO DO]

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 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 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 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 PACKAGE + RESERVING + VALIDITY CONDITION + ALTERNATIVE NAME + FARE PRODUCT/SUPPLEMENT PRODUCT
B3	**OFCO**	Conditions of offer	SALES 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 PiQuery

C.1.1 PI Query dependencies

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

— The QUERY Package describe PI Queries.

— The PASSENGER TRIP PACKAGE describes the Passenger Trips returned by .

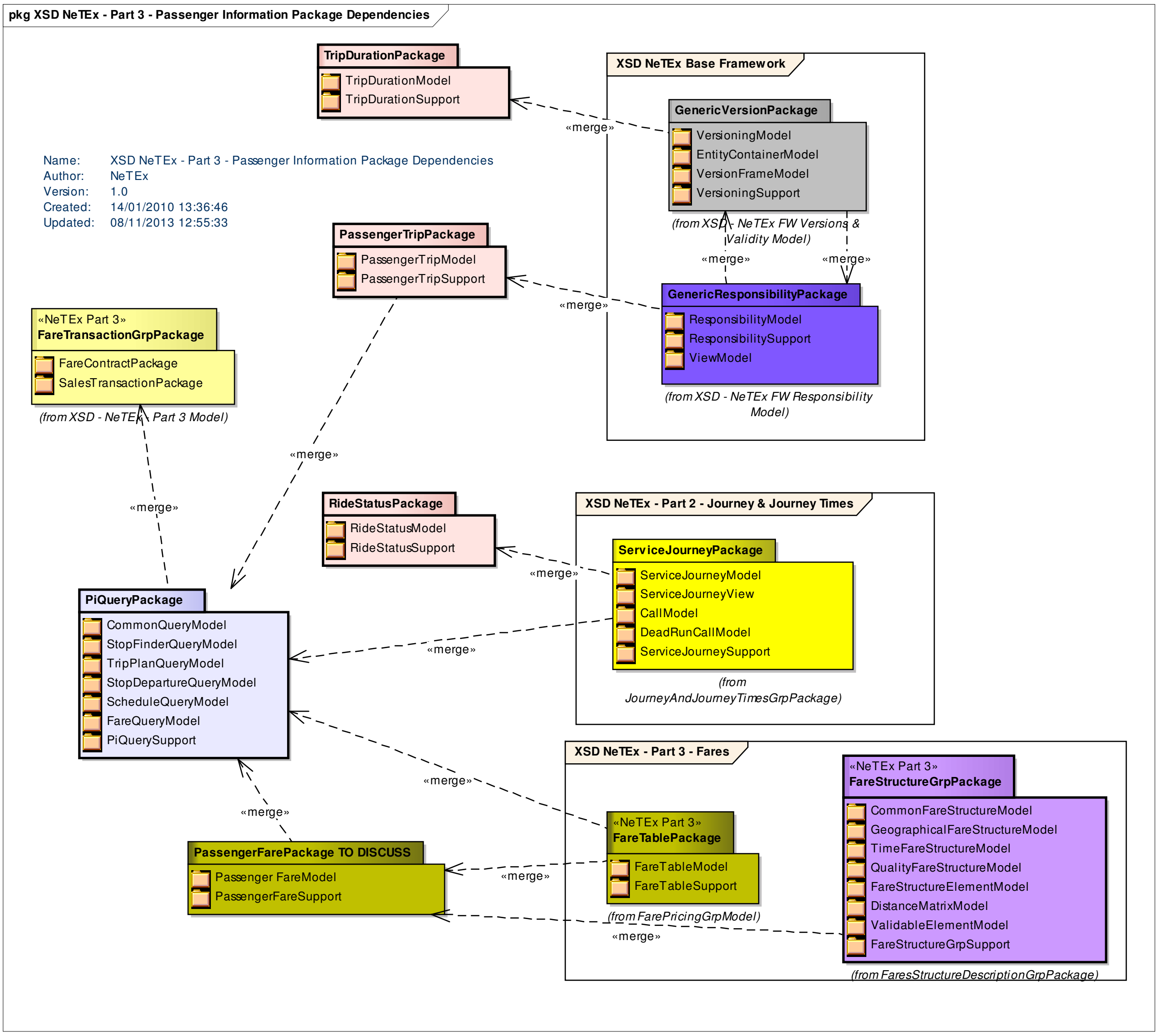


Figure 307 — PiQuery Package Dependencies

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

— The COMMON QUERY Model describe common query constructs used by all the different Query types.

— The STOP FINDER QUERY Model indicates the elements relevant for a query to find public transport stops.

— The TRIP PLANNER QUERY Model indicates the elements relevant for a query to find public transport journeys between given locations.

— The STOP DEPARTURE QUERY Model indicates the elements relevant for a query to find departures at a stop.

— The SCHEDULE QUERY Model indicates the elements relevant for a query to find public transport timetables.

— The FARE QUERY Model indicates the elements relevant for queries to find the fares for public transport journeys between given locations.

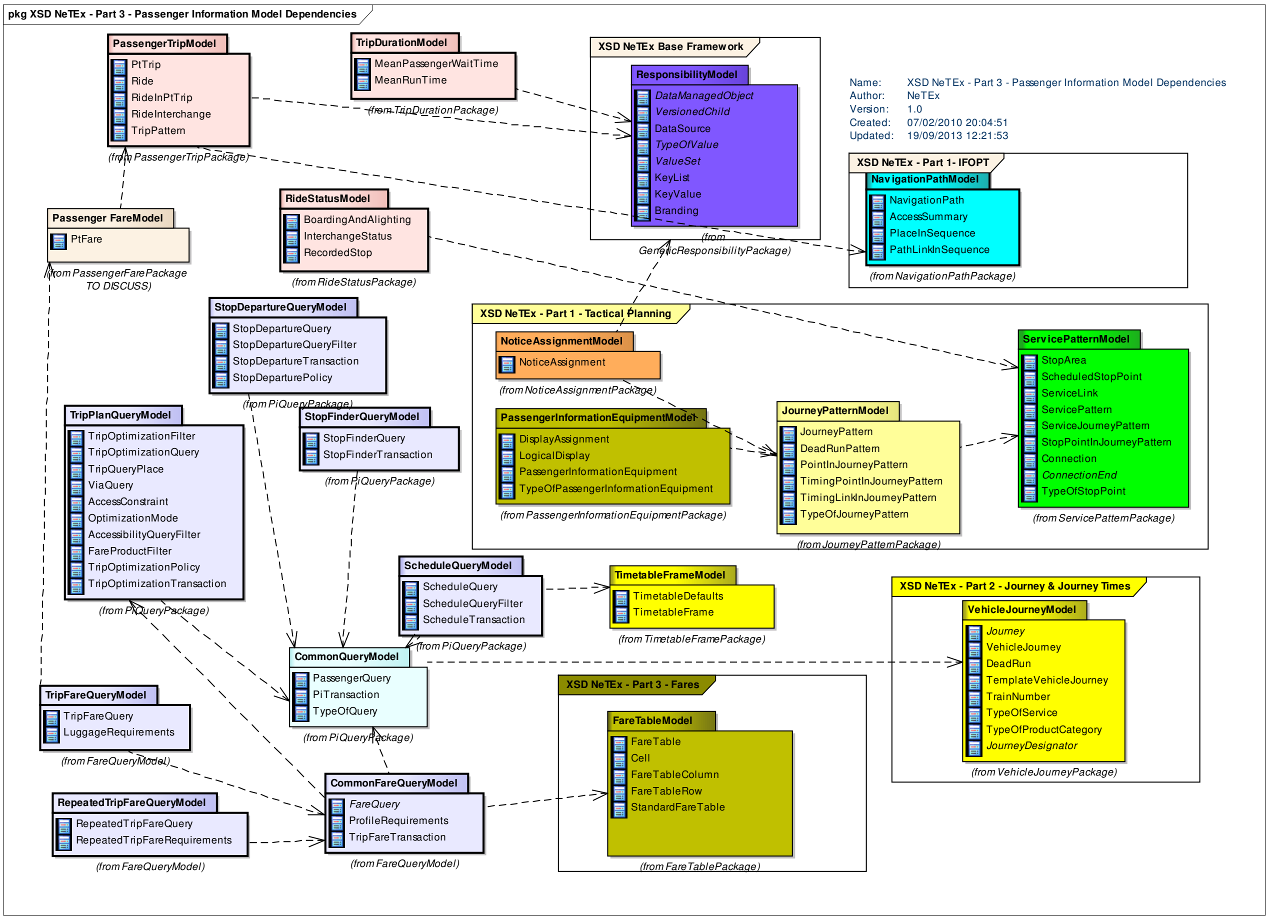


Figure 308 — PiQuery Model Dependencies

C.1.2 PiQuery

For information services that access passenger information through a service API, for example journey planners, fare queries, etc,

C.1.2.1 Common PI Query

C.1.2.1.1 PiQuery – Physical model – Overview

The following figure introduces the physical model for PI QUERYs.

Each PT TRANSACTION (corresponding to an http request or remote procedure call) may contain one or more PASSENGER QUERYs. There are a number of different types of concrete transaction (STOP FINDER TRANSACTION, TRIP PLAN TRANSACTION, etc.), each with its own corresponding query type (STOP FINDER QUERY, TRIP PLAN, etc.) and a specific result type.

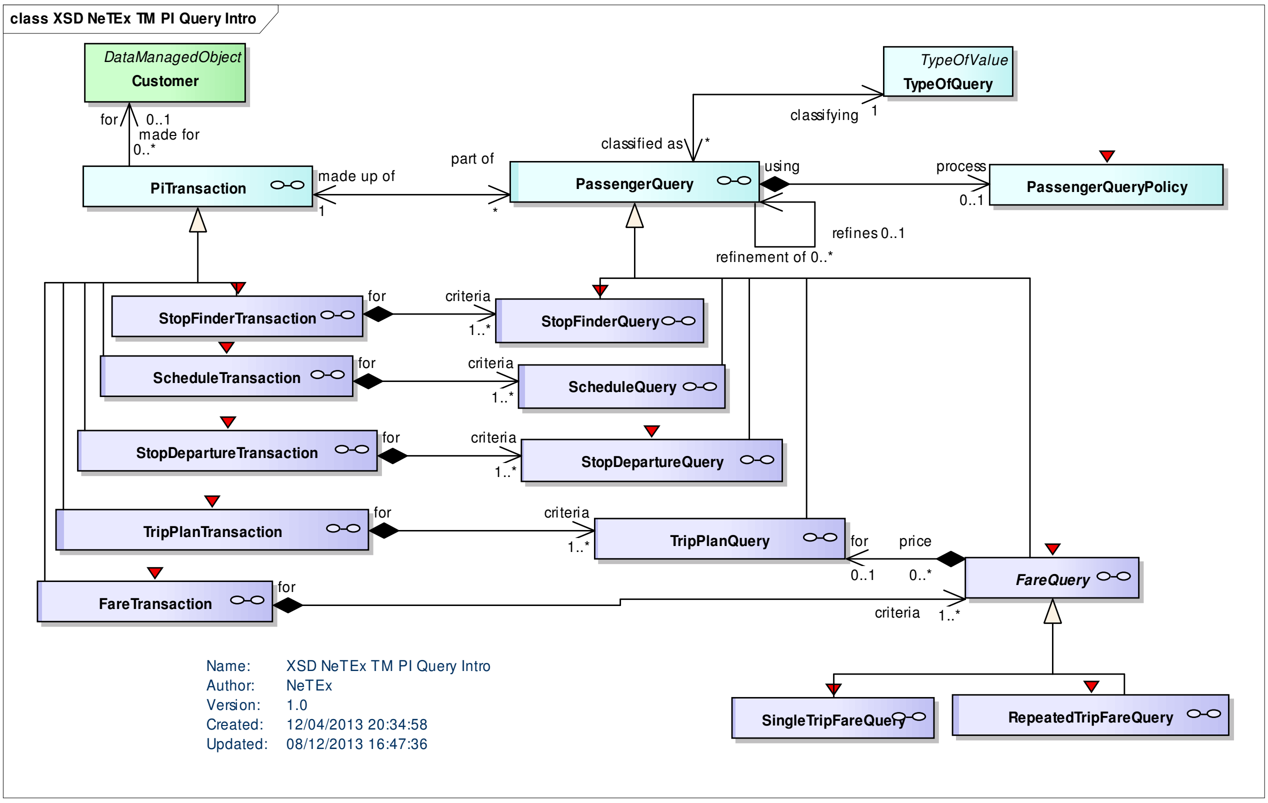


Figure 309 — PiQuery – Physical Model: Overview

C.1.2.1.2 PiQuery – Physical model – Details

The following figure shows the physical model for PI QUERYs.

The PT TRANSACTION provides a container. The QUERY specifies the actual search criteria.

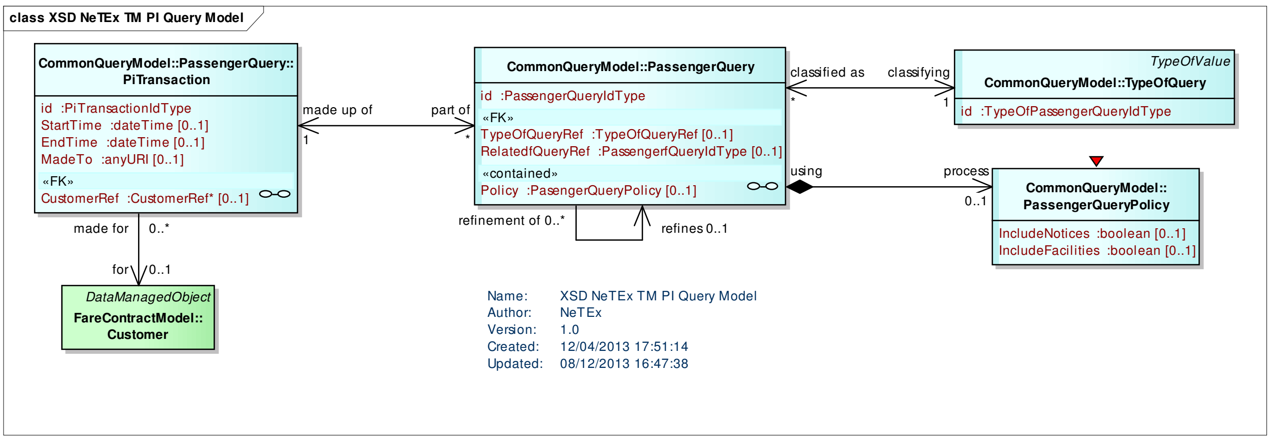


Figure 310 — PiQuery – Physical Model: Details

C.1.2.1.3 PiQuery – Attributes and XSD

C.1.2.1.3.1 PiTransaction – Model Element

A connection of a passenger to the operator information system, directly or via an employee, including one or several queries.

Table 256 – *PiTransaction* – Element

Classifi-cation	Name	Type	Cardinality	Description
	id	*PiTransactionIdType*	1:1	Identifier of PI TRANSACTION.
	StartTime	*dateTime*	0:1	Start time for transaction.
	EndTime	*dateTime*	0:1	Time when transaction was completed.
	MadeTo	*anyURI*	0:1	Uri to which transaction was made.
	UserProfileRef	*UserProfileRef*	0:1	Reference to USER PROFILE of user making query.
	CustomerRef	*CustomerRef*	0:1	Reference to CUSTOMER making query.

C.1.2.1.3.2 PassengerQuery – Model Element

A request for a specific information on public transport, expressed during a PI TRANSACTION.

Table 257 – *PassengerQuery* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
	id	*PassengerQueryIdType*	1:1	Identifier of a PASSENGER QUERY.
«FK»	***TypeOfQueryRef***	*TypeOfQueryRef*	0:1	Type of PASSENGER QUERY.
«FK»	***RelatedQueryRef***	*PassengerQueryIdType*	0:1	If refinement to previous query, identifier of related query.

C.1.2.1.3.3 PassengerQueryPolicy – Model Element

Optimisation criteria to be used to when computing and decorating the query results.

Table 258 – *PassengerQueryPolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
	IncludeNotices	*boolean*	0:1	Whether results should include NOTICEs associated with result elements.
	IncludeFacilities	*boolean*	0:1	Whether results should include information about facilities of result elements.
	IncludeAccessibility	*boolean*	0:1	Whether results should include ACCESSIBILITY ASSESSMENTs associated with result elements.

C.1.2.1.3.4 TypeOfQuery – Model Element

A classification of PASSENGER QUERies.

Table 259 – *PassengerQueryType* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TypeOfValue*	::>	TYPE OF PASSENGER QUERY inherits from TYPE OF VALUE.
	id	*TypeOfPassenger-QueryIdType*	1:1	Identifier of a TYPE OF PASSENGER QUERY.

C.1.2.2 Stop Finder Query

C.1.2.2.1 Stop Finder Query – Physical model – Overview

The following figure introduces the physical model for a STOP FINDER QUERY. A Stop FINDER QUERY 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 TRANSACTION (corresponding to an http request or remote procedure call) may contain one or more STOP FINDER QUERYs.)

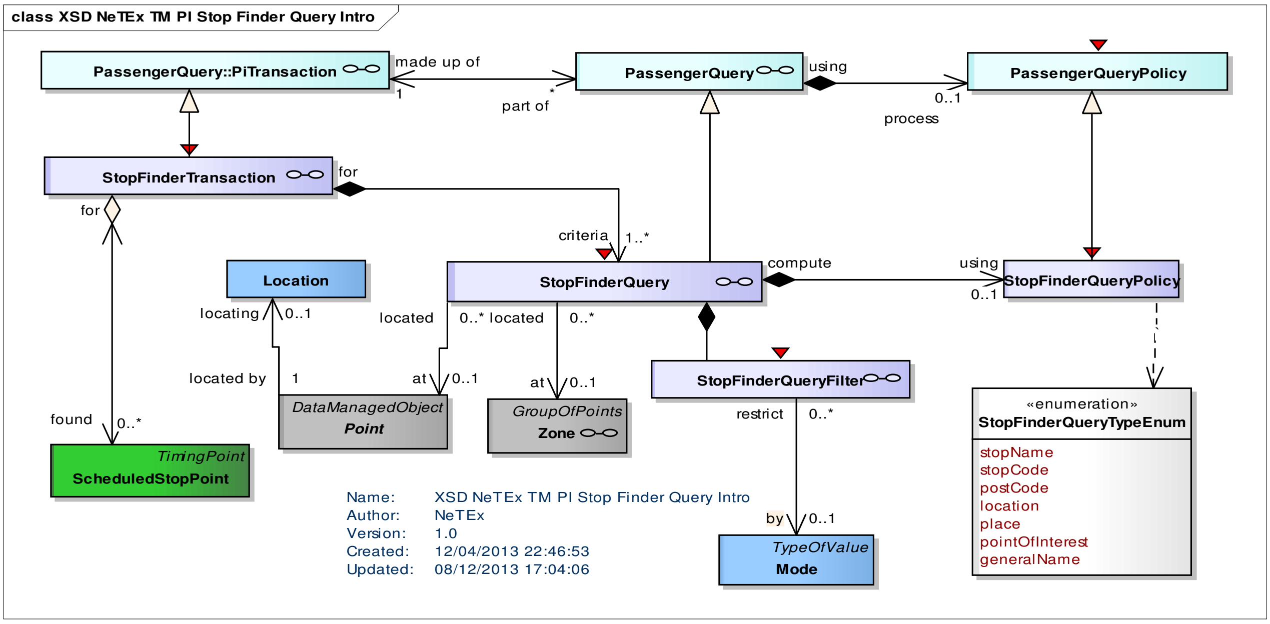


Figure 311 — Stop Finder Query– Physical Model: Overview

C.1.2.2.2 Stop Finder Query – Physical model – Details

The following figure shows the physical model for STOP FINDER QUERY.

A ZONE may be a STOP, POINT of INTEREST, PARKING or a TOPOGRAPHIC PLACE (See NeTEx part1).

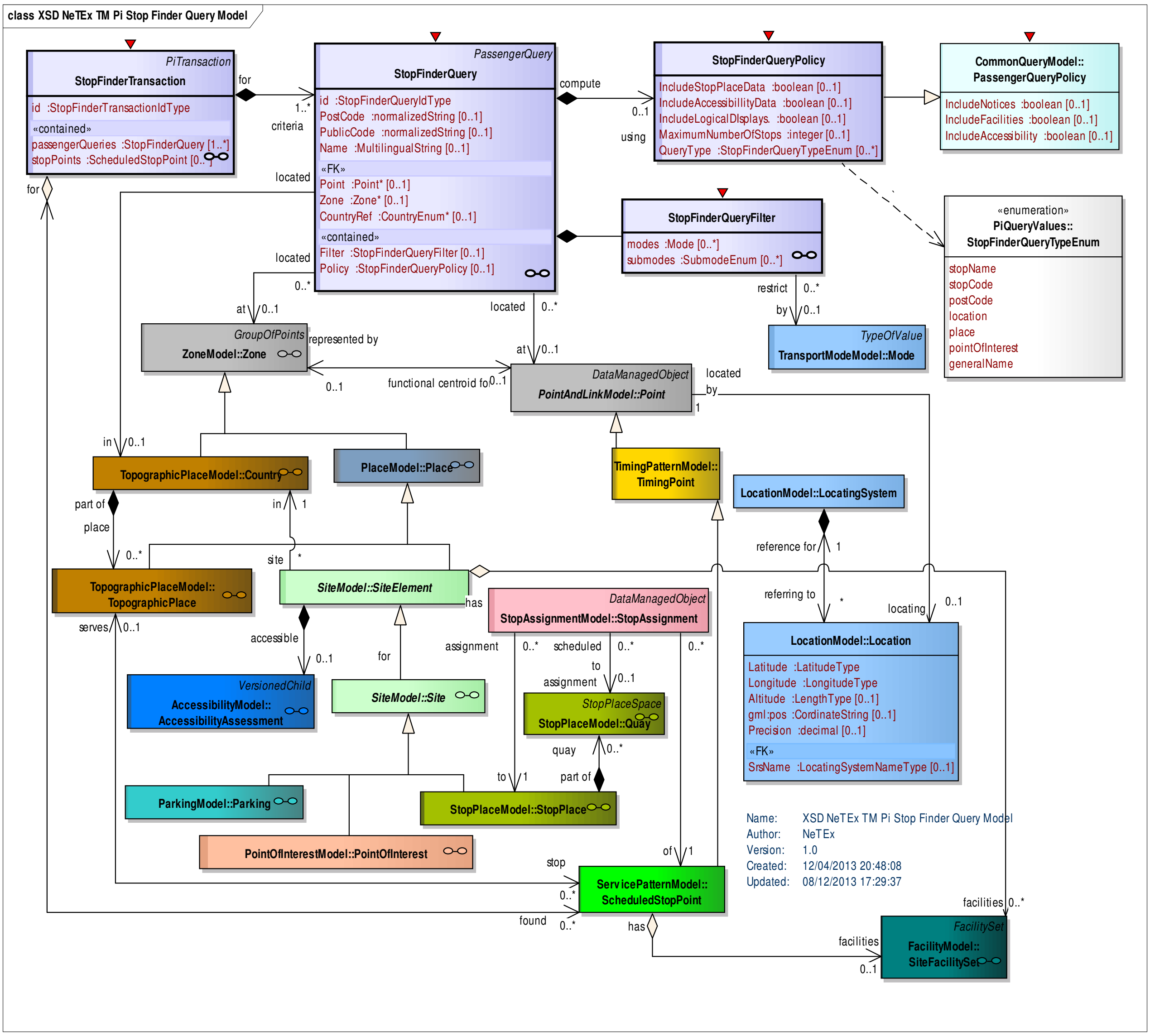


Figure 312 — Stop Finder Query – Physical Model: Details

C.1.2.2.3 Stop Finder– Attributes and XSD

C.1.2.2.3.1 StopFinderTransaction – Model Element

A specialization of PI TRANSACTION to make one or more STOP FINDER QUERies.

Table 260 – *StopFinderTransaction*– Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PiTransaction*	::>	STOP FINDER TRANSACTION inherits from PI TRANSACTION.
	id	*StopFinderTransaction-IdType*	1:1	Identifier of a STOP FINDER TRANSACTION.

“cntd»	***passenger-Queries***	*StopFinderQuery*	1:*	STOP FINDER QUERies used in the transaction.
“cntd»	***stopPoints***	*ScheduledStopPoint*	0:*	SCHEDULED STOP POINTs returned in response to query.

C.1.2.2.3.2 StopFinderQuery – Model Element

A PASSENGER QUERY to find a stop. Stops matching all of the given search criteria will be returned.

Table 261 – *StopFinderQuery*– Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PiQuery*	::>	STOP FINDER QUERY inherits from PI QUERY.
	id	*StopFinderQueryIdType*	1:1	Identifier of a STOP FINDER QUERY.
«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.
	QueryType	*StopFinderQuery-TypeEnum*	0:*	Type of STOP FINDER QUERY.
«contained»	***Filter***	*StopFinderQueryFilter*	0:1	Additional filter parameters.
«contained»	***Policy***	*StopFinderQueryPolicy*	0:1	Criteria for selecting and decorating stop data.

C.1.2.2.3.3 StopFinderQueryFilter – Model Element

Filter parameters used to limit the results of the query.

Table 262 – *StopFinderQueryFilter* – Element

Classifi-cation	Name	Type	Cardin-ality	Description

	modes	*Mode*	0:*	MODEs of transport to include in query.
	submodes	*SubmodeEnum*	0:*	submodes of transport to include in query.

C.1.2.2.3.4 StopFinderQueryPolicy – Model Element

Optimisation criteria to be used to when computing and decorating the query results.

Table 263 – *StopFinderQueryPolicy* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerQueryPolicy*	::>	STOP FINDER QUERY POLICY inherits from PASSENGER QUERY POLICY.
	Include-StopPlaceData	*boolean*	0:1	Whether results should include information about STOP PLACES, QUAYS and BOARDING POSITIONs.
	Include-AccessibilityData	*boolean*	0:1	Whether results should include accessibility data or stop.
	Include-LogicalDisplays.	*boolean*	0:1	Whether results should include information about LOGICAL DISPLAYS associated with stop. start and end.
	Maximum-NumberOfStops	*integer*	0:1	Maximum number of stops to include in results.
	QueryType	*StopFinderQueryType Enum*	0:*	Type of Stop Finder Query.

StopFinderQueryType – Allowed values

The following table shows the allowed values for ***QueryType***. (*StopFinderQueryTypeEnum).*

Table 264 – *QueryType* – Allowed values

Value	Description
stopName	Query by stop name.
stopCode	Query by stop code.
postCode	Query by post code.
location	Query by coordinates.
place	Query by place name.
pointOfInterest	Query by point of interest name.

generalName	Query by name; might be stop or place.

C.1.2.3 Trip Plan Query

C.1.2.3.1 Trip Plan Query – Physical model – Overview

The following figure introduces the physical model for a TRIP PLAN QUERY. A Trip Plan QUERY returns a list of PT TRIPs that satisfy the given search criteria.

Each TRIP PLAN TRANSACTION (corresponding to an http request or remote procedure call) may contain one or more TRIP PLAN QUERYs.

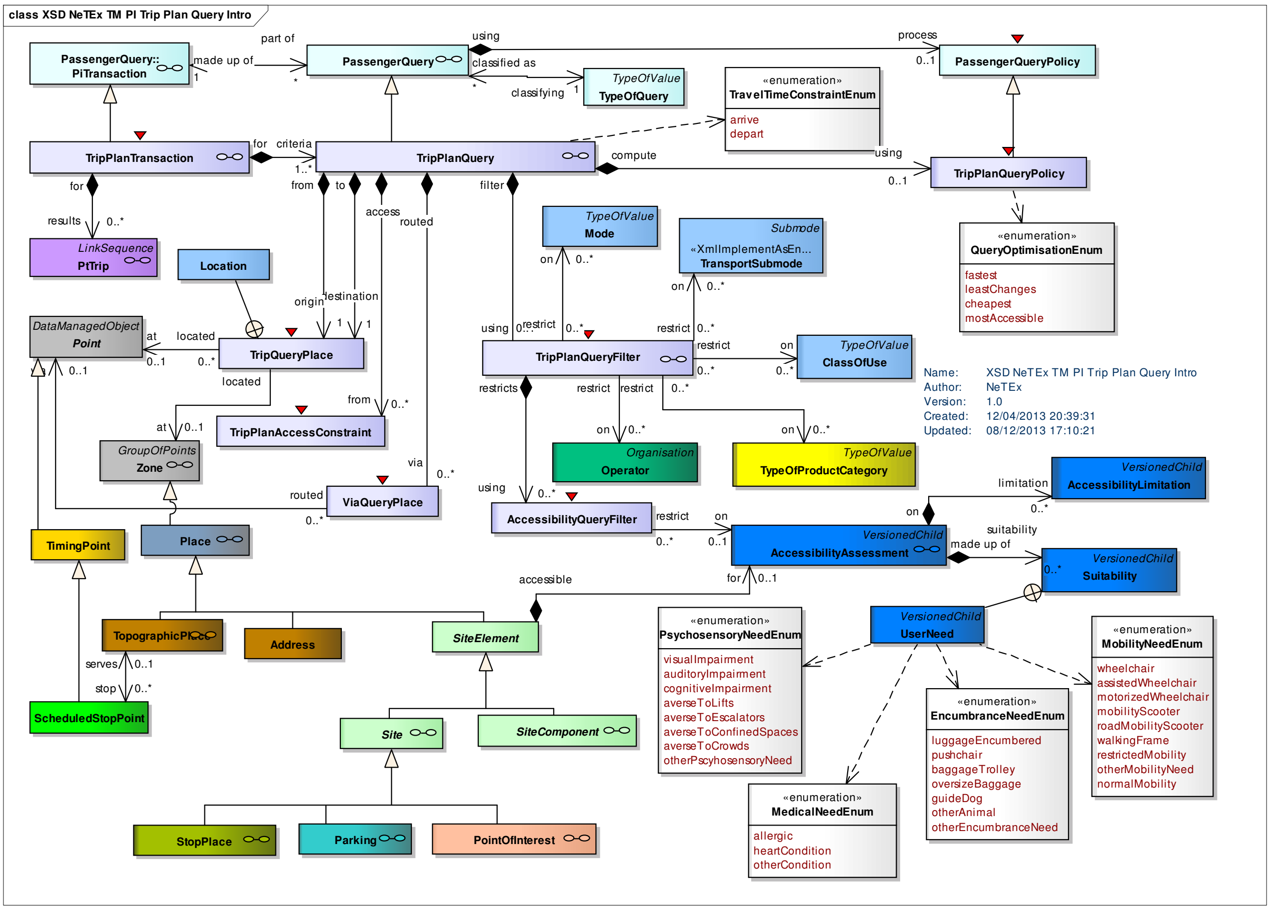


Figure 313 — Trip Plan Query– Physical Model: Overview

C.1.2.3.2 Trip Plan Query – Physical model – Details

The following figure shows the physical model for TRIP PLAN QUERY.

The fundamental criteria (Origin and destination TRIP QUERY PLACE, time of travel etc) must be supplied. Other additional criteria can also be specified.

- VIA QUERY 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 QUERY 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 QUERY FILTER can be used to set additional requirements for mobility.
- TRIP PLAN QUERY POLICY species parameters controlling the way the journey plan is computed.

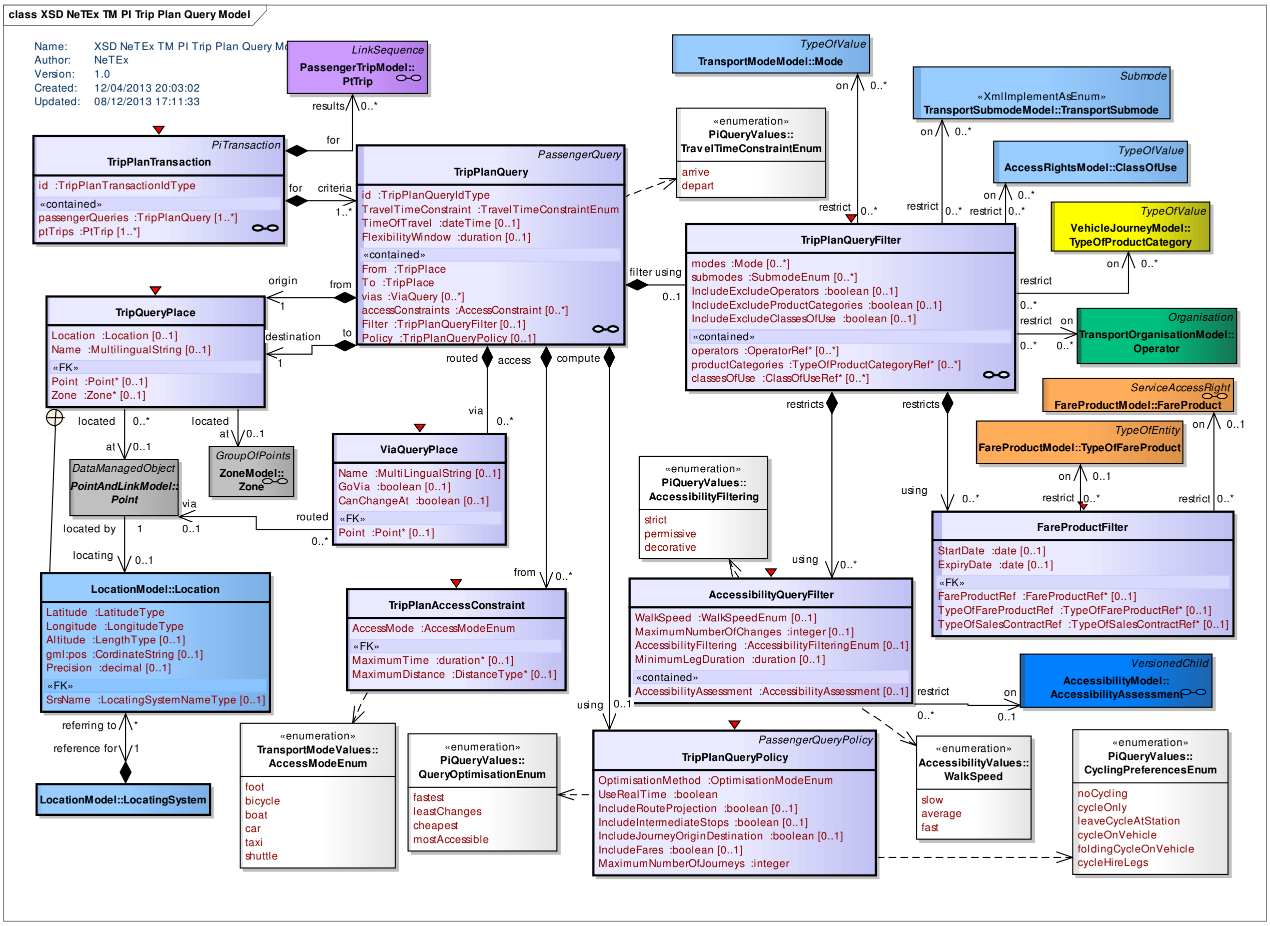


Figure 314 — Trip Plan Query – Physical Model: Details

C.1.2.3.3 Trip Plan Query – Attributes and XSD

C.1.2.3.3.1 TripPlanTransaction – Model Element

A specialization of PI TRANSACTION to make one or more TRIP PLAN QUERies.

Table 265 – *TripPlanTransaction* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PiTransaction*	::>	TRIP PLAN TRANSACTION inherits from PI TRANSACTION
	id	*TripPlanTransactionIdType*	1:1	Identifier of a TRIP PLAN TRANSACTION.

“cntd»	***passenger-Queries***	*TripPlanQuery*	1:*	TRIP PLAN queries used in transaction.
“cntd»	***ptTrips***	*PtTrip*	1:*	Passenger TRIPs returned in response to query.

C.1.2.3.3.2 TripPlanQuery – Model Element

A PASSENGER QUERY concerning an optimal trip proposal, according to a specified OPTIMISATION POLICY.

Table 266 – *TripPlanQuery* – Element

Classification	Name	Type	Cardinality	Description
::>	::>	*PiQuery*	::>	TRIP PLAN QUERY inherits from PI QUERY.
	id	*TripPlanQueryIdType*	1:1	Identifier of a TRIP PLAN QUERY.
“cntd»	***From***	*TripPlace*	1:1	Place at which desired TRIP starts.
“cntd»	***To***	*TripPlace*	1:1	Place at which desired TRIP ends.
	TravelTime-Constraint	*TravelTime-ConstraintEnum*	1:1	Constraint on how to use travel time: to arrival or to departure.
	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***	*ViaQuery*	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***	*TripPlanQueryFilter*	0:1	Additional filter parameters. See below.
“cntd»	***Policy***	*TripPlanQueryPolicy*	0:1	Additional processing parameters. See below.

TravelTimeConstraint– Allowed values

The following table shows the allowed values for ***TravelTimeConstraint***. (*TravelTimeConstraintEnum).*

Table 267 – *TravelTimeConstraint* – Allowed values

Value	Description
arrive	Plan trip to arrive at specified time.

depart	Plan trip to depart at specified time.

C.1.2.3.4 TripQueryPlace – 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 268 – *TripQueryPlace* – 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	Co-ordinates 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.1.2.3.4.1 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 269 – *TripPlanAccessConstraint* – Element

Classifi-cation	Name	Type	Cardinality	Description
	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.1.2.3.4.2 ViaQueryPlace – 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 270 – *ViaQueryPlace* – 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.1.2.3.4.3 TripPlanQueryFilter – Model Element

Filter parameters used to limit the MODEs, PRODUCT CATEGORIES and CLASS OF USE of the journeys returned.

Table 271 – *TripPlanQueryFilter* – Element

Classifi-cation	Name	Type	Cardinality	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.1.2.3.4.4 AccessibilityQueryFilter – Model Element

Criteria used to limit the ACCESSIBILITY properties of the journeys returned.

Table 272 – *AccessibilityQueryFilter* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
"cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	Default ACCESSIBILITY ASSESSMENT for JOURNEYS in the TIMETABLE, specifying. See NeTEx Part1.
	WalkSpeed	*WalkSpeedEnum*	0:1	Walk speed to use. See allowed values.
	MaximumNumberOf-Changes	*xsd:integer*	0:1	Maximum number of changes allowed.
	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***. (*AccessibilityFilteringEnum).*

Table 273 – *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.

C.1.2.3.4.5 FareProductFilter – Model Element

The FARE PRODUCTs to which to restrict resuts of a query.

Table 274 – *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.1.2.3.4.6 TripPlanQueryPolicy – Model Element

Optimisation criteria to be used to when computing and decorating trip plans.

Table 275 – *TripPlanQueryPolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
	Optimisation-Method	*Optimisation-MethodEnum*	1:1	How to optimize the query. See below.
	UseRealTime	*xsd:boolean*	1:1	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 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*	1:1	Maximum number of journeys to include in results.

OptimisationMethod – Allowed values

The following table shows the allowed values for ***OptimisationMethod***. (*OptimisationMethodEnum).*

Table 276 – *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***. (*CyclingPreferencesEnum)*.

Table 277 – *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.1.2.3.5 Stop Departure Query

C.1.2.3.6 Stop Departure Query – Physical model – Overview

The following figure introduces the physical model for a STOP DEPARTURE QUERY. A Stop Departure QUERY returns a list of CALLs that satisfy the given search criteria.

Each STOP DEPARTURE TRANSACTION (corresponding to an http request or remote procedure call) may contain one or more STOP DEPARTURE QUERYs.

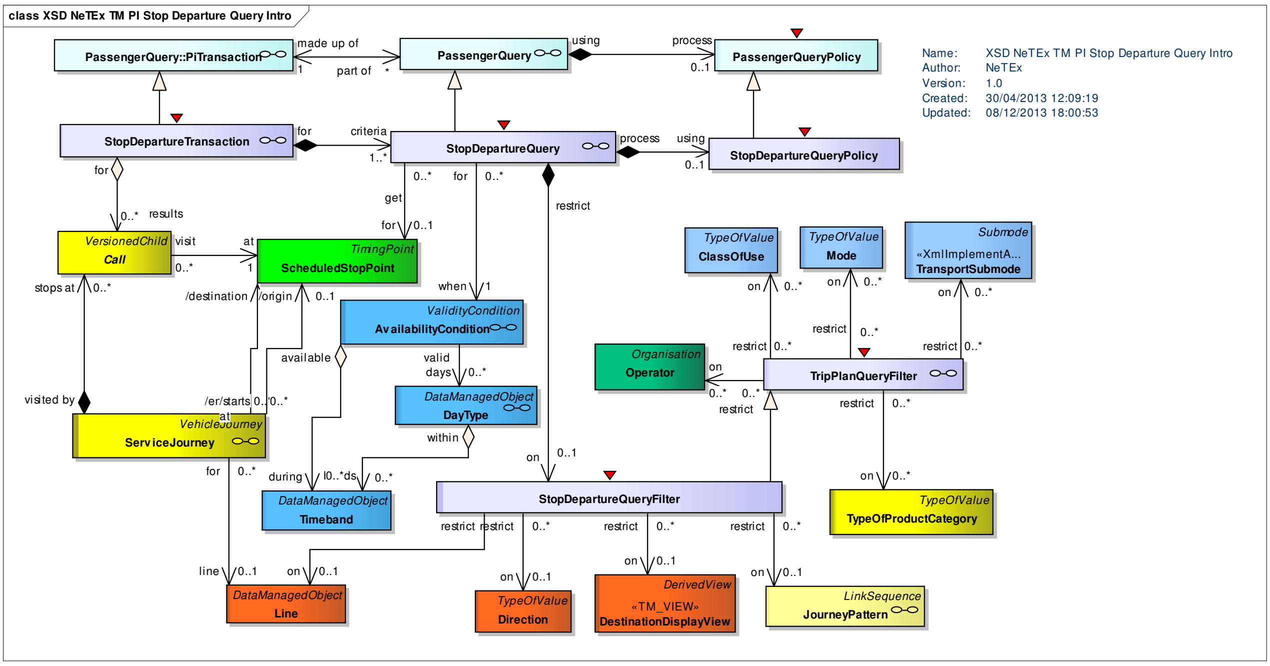


Figure 315 — Stop Departure Query– Physical Model: Overview

C.1.2.3.7 Stop Departure Query – Physical model – Details

The following figure shows the physical model for STOP DEPARTURE QUERY.

The fundamental criteria (SCHEDULED STOP POINT) must be supplied. Other additional criteria can also be specified.

- STOP DEPARTURE QUERY 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.

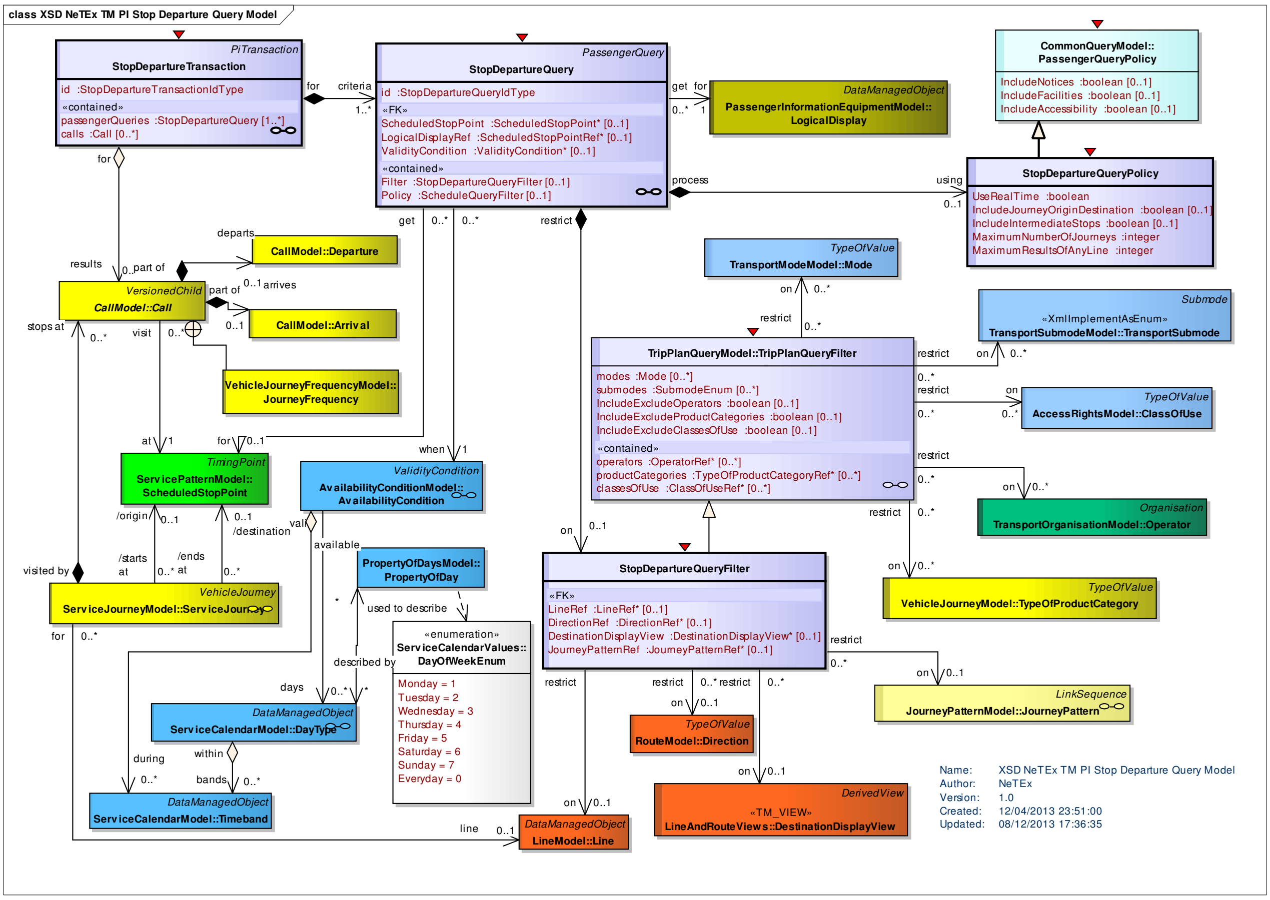


Figure 316 — Stop Departure Query – Physical Model: Details

C.1.2.3.8 Stop Departure Query – Attributes and XSD

C.1.2.3.8.1 StopDepartureTransaction – Model Element

A specialization of PI TRANSACTION to make one or more STOP DEPARTURE QUERies.

Table 278 – *StopDepartureTransaction* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PiTransaction*	::>	STOP DEPARTURE TRANSACTION inherits from PI TRANSACTION.
	id	*StopDeparture-TransactionIdType*	1:1	Identifier of a STOP DEPARTURE TRANSACTION .

“cntd»	***passenger-Queries***	*StopDepartureQuery*	1:*	STOP DEPARTURE Passenger QUERies.
“cntd»	***calls***	*Call*	0:*	CALLs returned in response to STOP DEPARTURE query.

C.1.2.3.8.2 StopDepartureQuery – Model Element

A PASSENGER QUERY about departures at a stop or LOGICAL DISPLAY at that stop.

Table 279 – *StopDepartureQuery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerQuery*	::>	STOP DEPARTURE QUERY inherits from PASSENGER QUERY.
	id	*StopDepartureQuery-IdType*	1:1	Identifier of a STOP DEPARTURE QUERY.
«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-QueryFilter*	0:1	Additional filter parameters. See below.
“cntd»	***Policy***	*StopDeparture-QueryFilter*	0:1	Additional policy parameters. See STOP DEPARTURE QUERY POLICY below.

C.1.2.3.8.3 StopDepartureQueryFilter – Model Element

Additional filter criteria for including VEHICLE JOURNEYs in a STOP DEPARTURE QUERY.

Table 280 – *StopDepartureQuery*Filter – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TripPlanQueryFilter*	::>	STOP DEPARTURE QUERY FILTER inherits from TRIP PLAN QUERY 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.1.2.3.8.4 StopDepartureQueryPolicy – Model Element

Criteria for computing results of a STOP DEPARTURE QUERY.

Table 281 – *StopDepartureQuery*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.
	MaximumNumberOf-Journeys	*xsd:integer*	1:1	Maximum number of journeys to include in results.
	MaxmimumResults-OfAnyLine	*xsd:integer*	1:1	Maximum number of CALLs for any line to be returned,

C.1.2.4 Schedule Query

C.1.2.4.1 Schedule Query – Physical model – Overview

The following figure introduces the physical model for a SCHEDULE QUERY. A Schedule QUERY returns a list of TIMETABLEs that satisfy the given search criteria.

Each SCHEDULE TRANSACTION (corresponding to an http request or remote procedure call) may contain one or more SCHEDULE QUERYs.

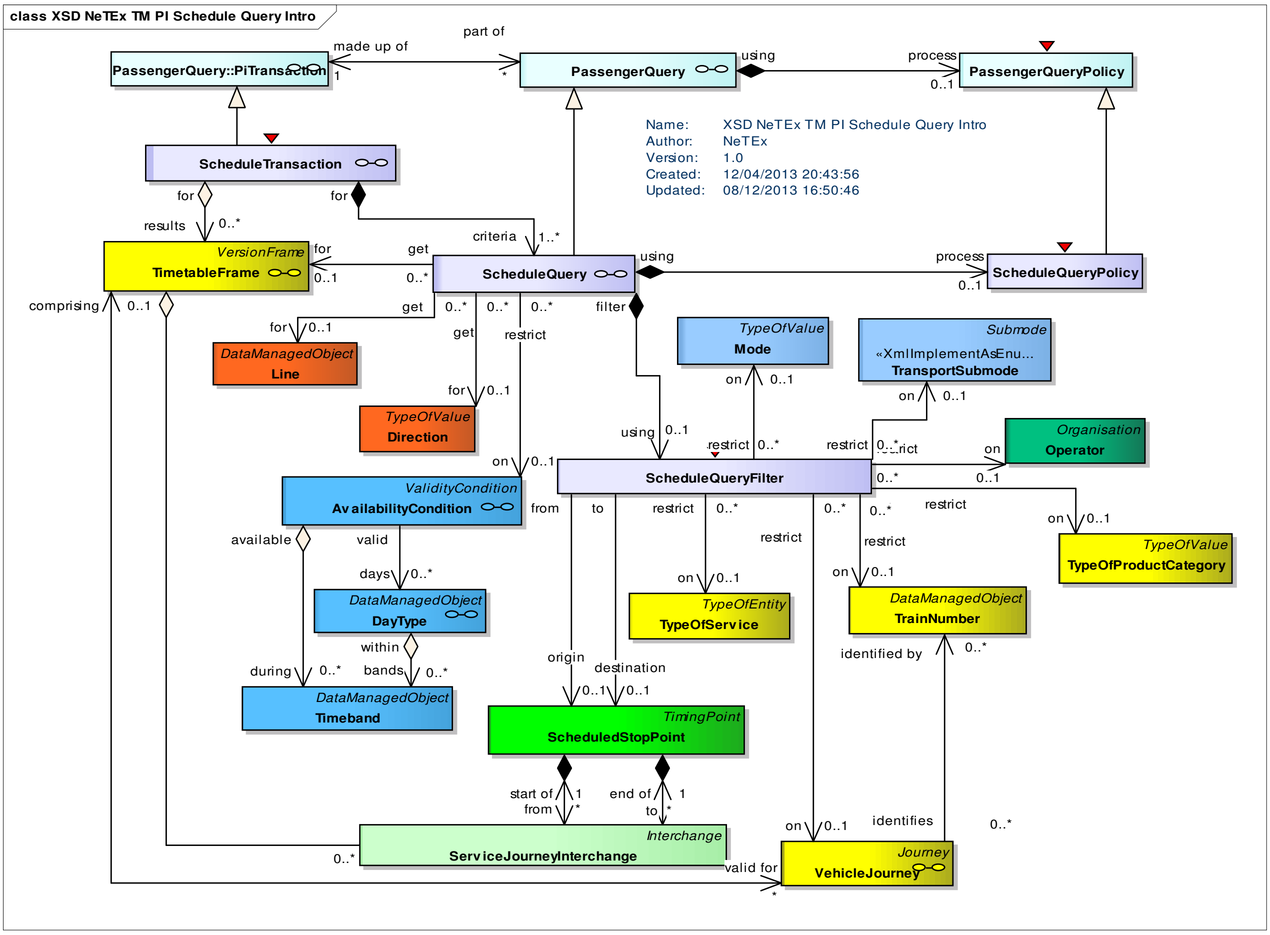


Figure 317 — Schedule Query– Physical Model: Overview

C.1.2.4.2 Schedule Query – Physical model – Details

The following figure shows the physical model for SCHEDULE QUERY.

Eitejr a specifc timetable may be specified, or paraeters to identify one.

- SCHEDULE QUERY FILTER can be used to limit the journeys to be included by MODE. LINE, DIRECTION etc. and to a partiucalr start and end point.
- SCHEDULE QUERY POLICY can be used to direct which information is to be included in the results.

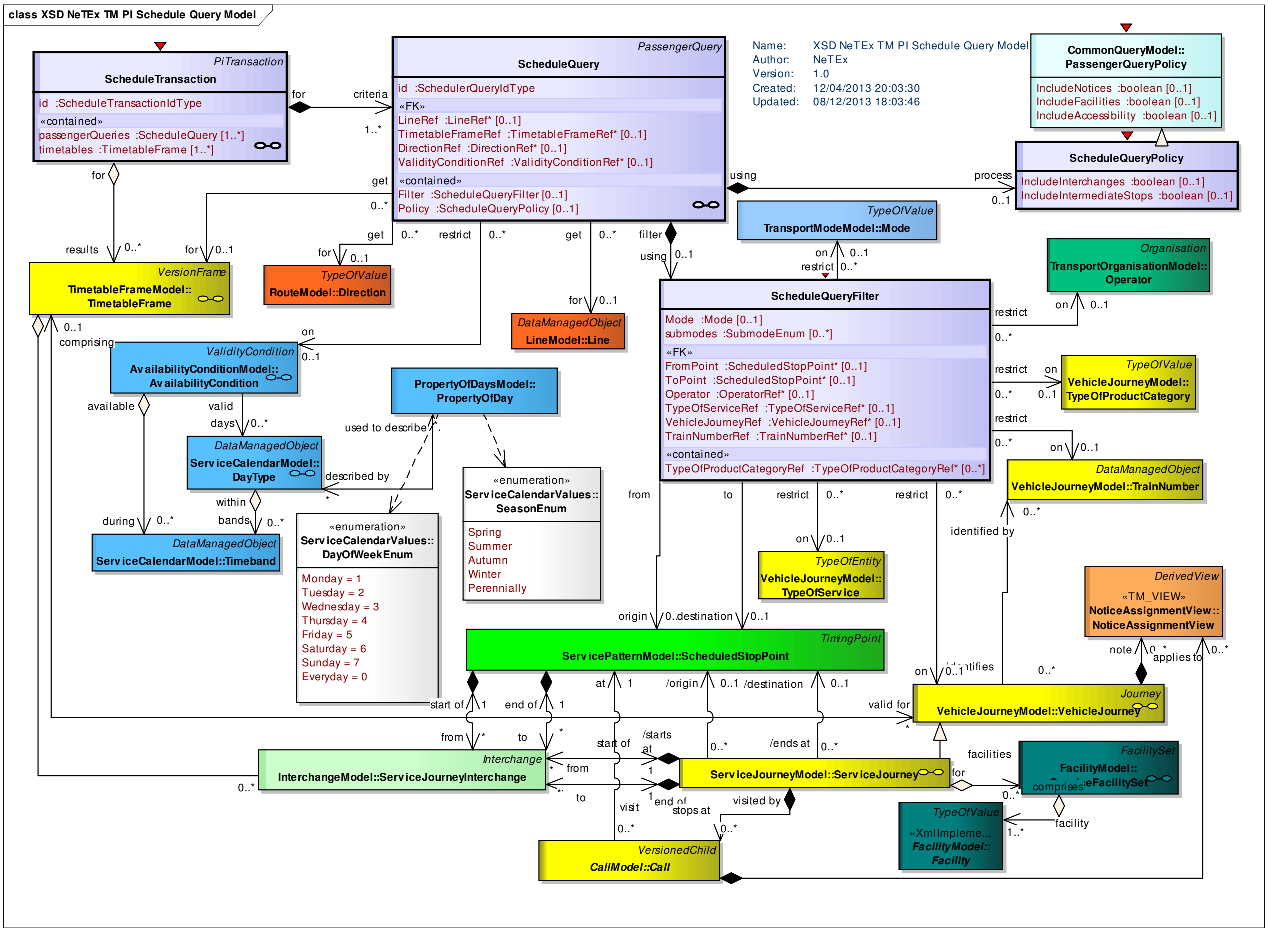


Figure 318 — Schedule Query – Physical Model: Details

C.1.2.4.3 Schedule Query – Attributes and XSD

C.1.2.4.3.1 ScheduleTransaction – Model Element

A specialization of PI TRANSACTION to make one or more SCHEDULE QUERies.

Table 282 – *ScheduleTransaction* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PiTransaction*	::>	SCHEDULE TRANSACTION inherits from PI TRANSACTION.
	id	*Schedule-TransactionIdType*	1:1	Identifier of a SCHEDULE TRANSACTION.
“cntd»	***passenger-Queries***	*ScheduleQuery*	1:*	SCHEDULE QUERIES for the transaction.
“cntd»	***timetables***	*TimetableFrame*	1:*	TIMETABLE FRAMEs with SERVICE JOURNEYS returned in response to query.

C.1.2.4.3.2 ScheduleQuery – Model Element

A PASSENGER QUERY about public transport timetables.

Table 283 – *ScheduleQuery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerQuery*	::>	SCHEDULE QUERY inherits from PASSENGER QUERY.
	id	*SchedulerQueryIdType*	1:1	Identifier of a SCHEDULE QUERY.
«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***	*ScheduleQueryFilter*	0:1	Additional filter parameters. See below.
“cntd»	Policy	*ScheduleQueryPolic*	0:1	Query Policy parameters. See below.

C.1.2.4.3.3 ScheduleQueryFilter – Model Element

Additional filter criteria for SCHEDULE QUERY, for example start and end stop. Only timetables that satisfy criteria will be returned.

Table 284 – *ScheduleQueryFilter* – 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-***	*TypeOfProduct-*	0:*	Type of PRODUCT CATEGORY on which to filter

	CategoryRef	*CategoryRef*		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.1.2.4.3.4 ScheduleQueryPolicy – Model Element

Criteria for computing results of a SCHEDULE QUERY.

Table 285 – *ScheduleQueryPolicy* – Element

Classifi-cation	Name	Type	Cardin-ality	Description
::>	::>	*PassengerQuery Policy*	::>	SCHEDULE QUERY POLICY inherits from PASSENGER QUERY POLICY.
	IncludeInterchanges	*boolean*	0:1	Whether results should include information about SERVICE JOURNEY INTERCHANGES for the SERVICE JOURNEYs in the timetable.
	IncludeIntermediate Stops	*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 Query

C.1.2.5.1 Common Fare Query Model

C.1.2.5.1.1 Common Fare Query – Physical model – Overview

Fare queries fetch the available fares for a trip. There are two types of fare query

- TRIP FARE QUERY: finds the FARE PRODUCTs suitable for a single trip.
- REPEATED FARE QUERY: finds the FARE PRODUCTs suitable for regularly repeated journeys such as season passes.

The following figure introduces the physical model for common fare query elements that are used by both types of query.

Each COMMON FARE TRANSACTION (corresponding to an http request or remote procedure call) may contain one or more COMMON FARE QUERYs.

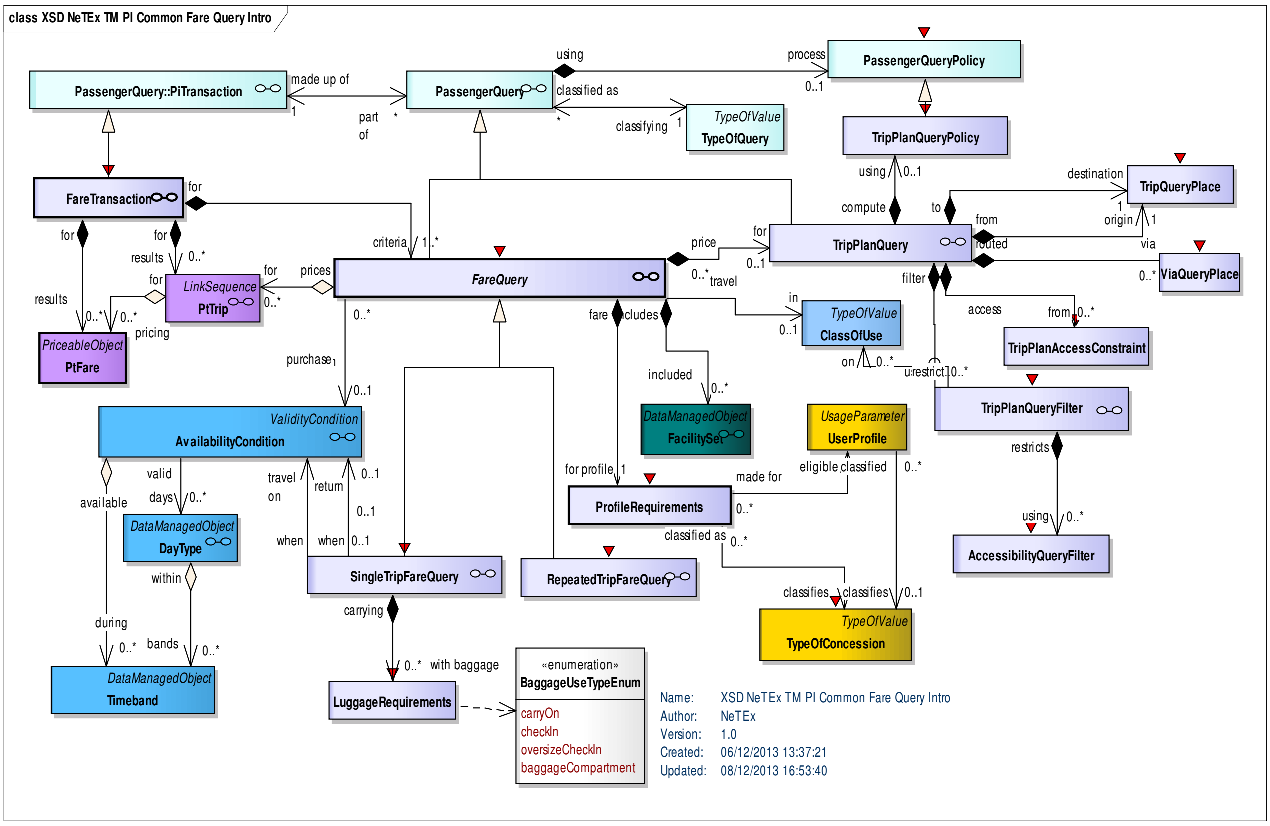


Figure 319 — Common Fare Query– Physical Model: Overview

C.1.2.5.1.2 Common Fare Query – Physical model – Details

The following figure shows the physical model for COMMON FARE QUERY.

The fundamental criteria (SCHEDULED STOP POINT) must be supplied. Other additional criteria can also be specified.

- COMMON FARE QUERY FILTER can be used to limit the journeys to be included. It can use a TRIP PLAN QUERY 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.

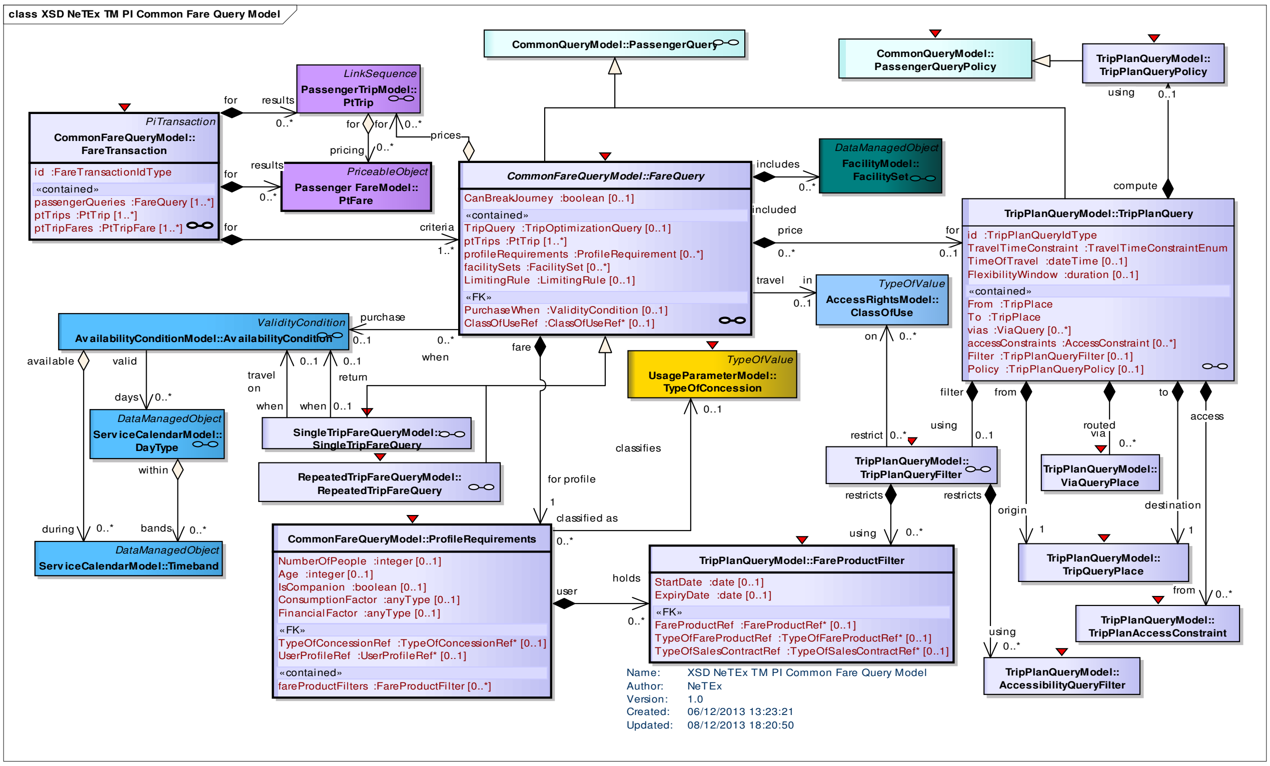


Figure 320 — Common Fare Query – Physical Model: Details

C.1.2.5.1.3 Common Fare Query – Attributes and XSD

FareTransaction – Model Element

A specialization of PI TRANSACTION to make one or more FARE QUERies.

Table 286 – *FareTransaction* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PiTransaction*	::>	FARE TRANSACTION inherits from PI TRANSACTION
	id	*FareTransaction-IdType*	1:1	Identifier of a FARE TRANSACTION.
“cntd»	***passenger-Queries***	*FareQuery*	1:*	FARE QUERies for PI TRANSACTION.
“cntd»	***ptTrips***	*PtTrip*	1:*	Passenger TRIPs in response to query.
“cntd»	***ptFares***	*PtTripFare*	1:*	Passenger FAREs in response to query

FareQuery – Model Element

A PASSENGER QUERY about fares.

Table 287 – *FareQuery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*PassengerQuery*	::>	FARE QUERY inherits from PASSENGER QUERY.
“cntd»	***TripQuery***	*TripPlanQuery*	0:1	Parameters describing a passenger trip for which fares are to be found. See TRIP PLAN QUERY.
«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 288 – *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-***	*FareProductFilter*	0:*	Products held by user that can be taken into

	Filters			account in query. See TRIP PLAN QUERY.

C.1.2.5.2 Single Trip Fare Query Model

C.1.2.5.2.1 Single Trip Fare Query – Physical model – Overview

The following figure introduces the physical model for a SINGLE TRIP FARE QUERY. A SINGLE TRIP FARE QUERY returns a list of PT FAREs for a PT TRIP that satisfy the given search criteria.

Each FARE TRANSACTION may contain one or more SINGLE TRIP FARE QUERies.

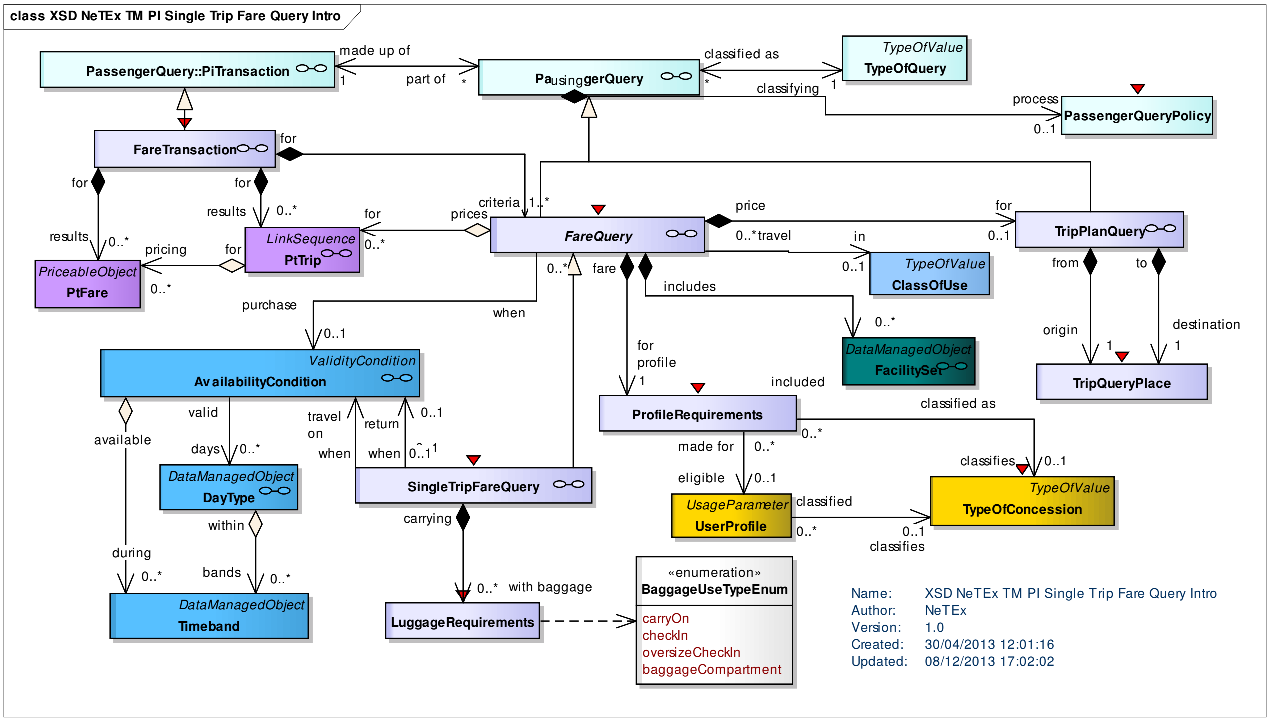


Figure 321 — Single Trip Fare Query– Physical Model: Overview

C.1.2.5.2.2 Single Trip Fare Query – Physical model – Details

The following figure shows the physical model for SINGLE TRIP FARE QUERY. 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 QUERY or the PT TRIPs resulting from a previously executed SINGLE TRIP QUERY.

- LUGGAGE REQUIREMENTs species details of luggage carrying requirements, e.g. a bicycle.

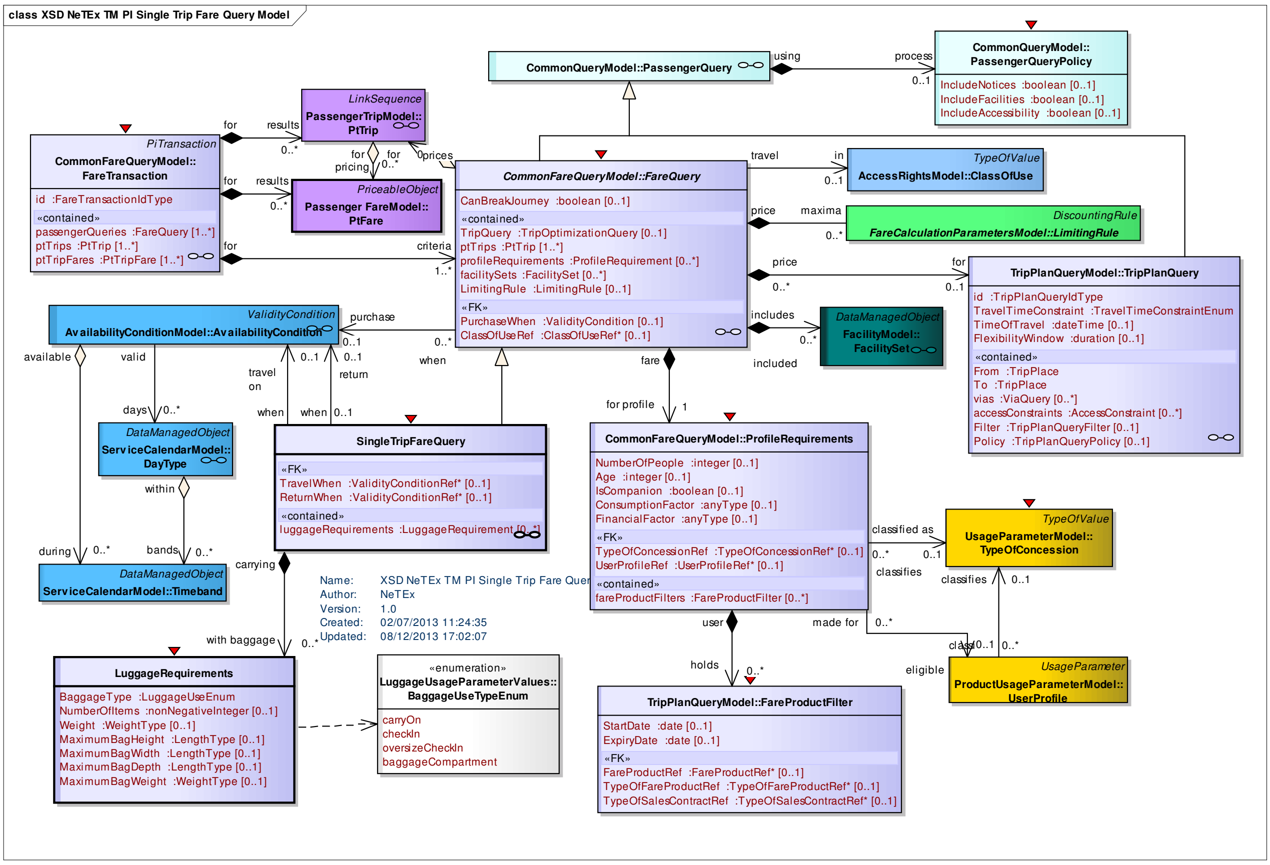


Figure 322 — Single Trip Fare Query – Physical Model: Details

C.1.2.5.2.3 Single Trip Fare Query – Attributes and XSD

SingleTripFareQuery – Model Element

A PASSENGER QUERY about the fares for a single trip or return trip.

Table 289 – *SingleTripFareQuery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareQuery*	::>	SINGLE TRIP FARE QUERY inherits from FARE QUERY
«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 290 – *LuggageRequirements* – Model Element

Classifi-cation	Name	Type	Cardin-ality	Description
	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 Query Model

C.1.2.5.3.1 Repeated Trip Fare Query – Physical model – Overview

The following figure introduces the physical model for a REPEATED TRIP FARE QUERY. A REPEATED TRIP FARE QUERY returns a list of PT FAREs for trips that satisfy the given search criteria.

Each FARE TRANSACTION may contain one or more REPEATED TRIP FARE QUERYs.

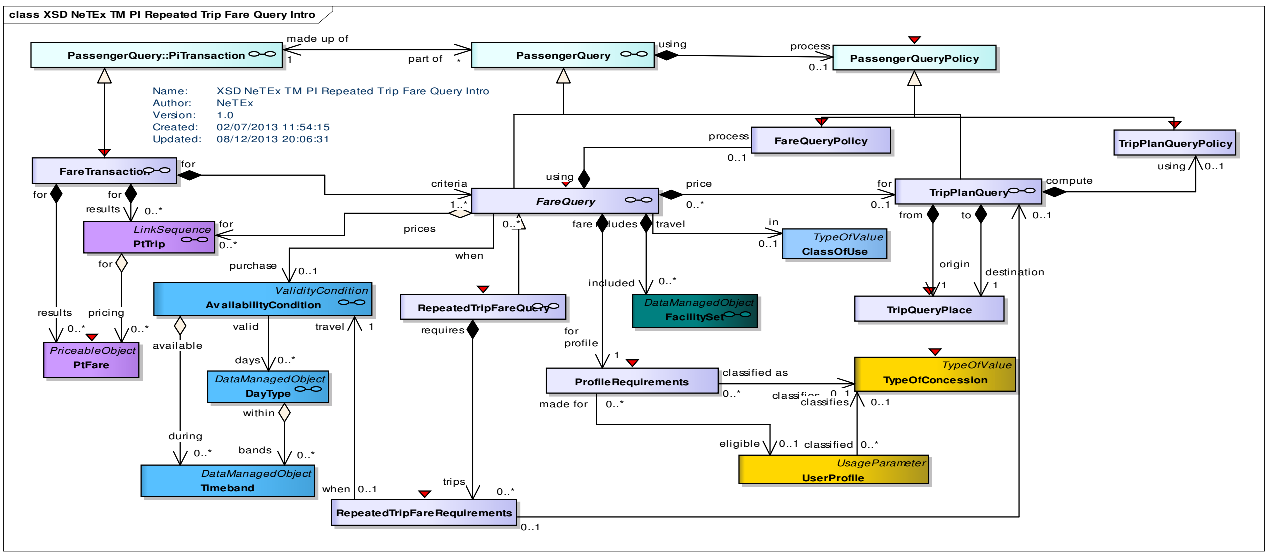


Figure 323 — Repeated Trip Fare Query– Physical Model: Overview

C.1.2.5.3.2 Repeated Trip Fare Query – Physical model – Details

The following figure shows the physical model for TRIP FARE QUERY. 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 QUERY or the PT TRIPs resulting from a previously executed TRIP QUERY.

- REPEATED TRIP FARE QUERY 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.

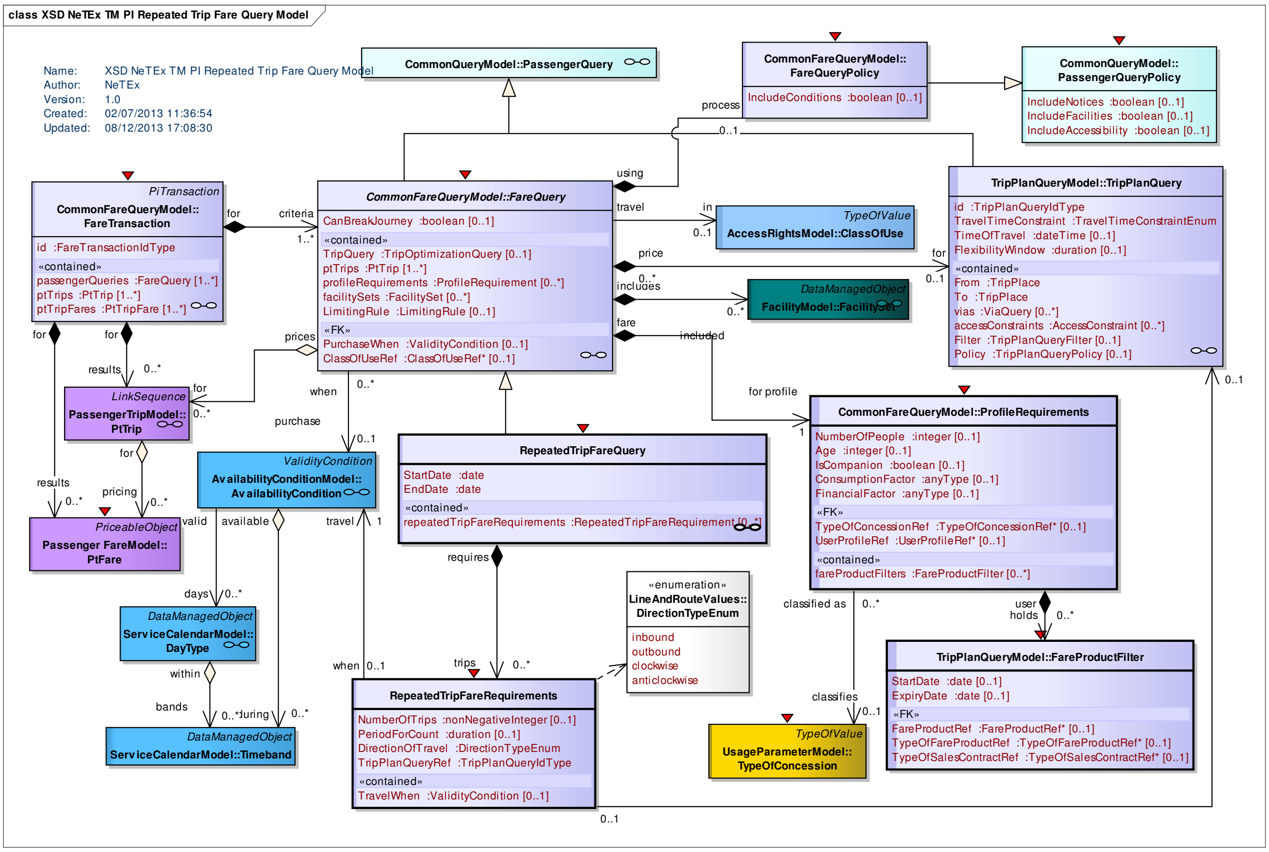


Figure 324 — Repeated Trip Fare Query – Physical Model: Details

C.1.2.5.3.3 Repeated Trip Fare Query – Attributes and XSD

RepeatedTripFareQuery – Model Element

A PASSENGER QUERY to find the best fares to use for repeated trips.

Table 291 – *RepeatedTripFareQuery* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FareQuery*	::>	REPEATED TRIP FARE QUERY inherits from FARE QUERY.
	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 292 – *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.
	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 specifc trip.

- The PASSENGER TRIP Model describes an individual passenger trip by public transport. It can be used to describe the results of a TRIP QUERY and also be used as an input to a FARE QUERY 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 QUERY. It can also be used on a SALES TRANSACTION may be used to record the inputs used to find the fare

D.1 Passenger Trip

D.1.1 Passenger Trip Model

D.1.1.1 Passenger Trip – Physical model – Overview

The following figure introduces the physical model for a PASSENGER TRIP. A Passenger Trip QUERY 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 RIDE INTERCHANGE 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 RIDE INTERCHANGEs as well as for the overall PT TRIP.

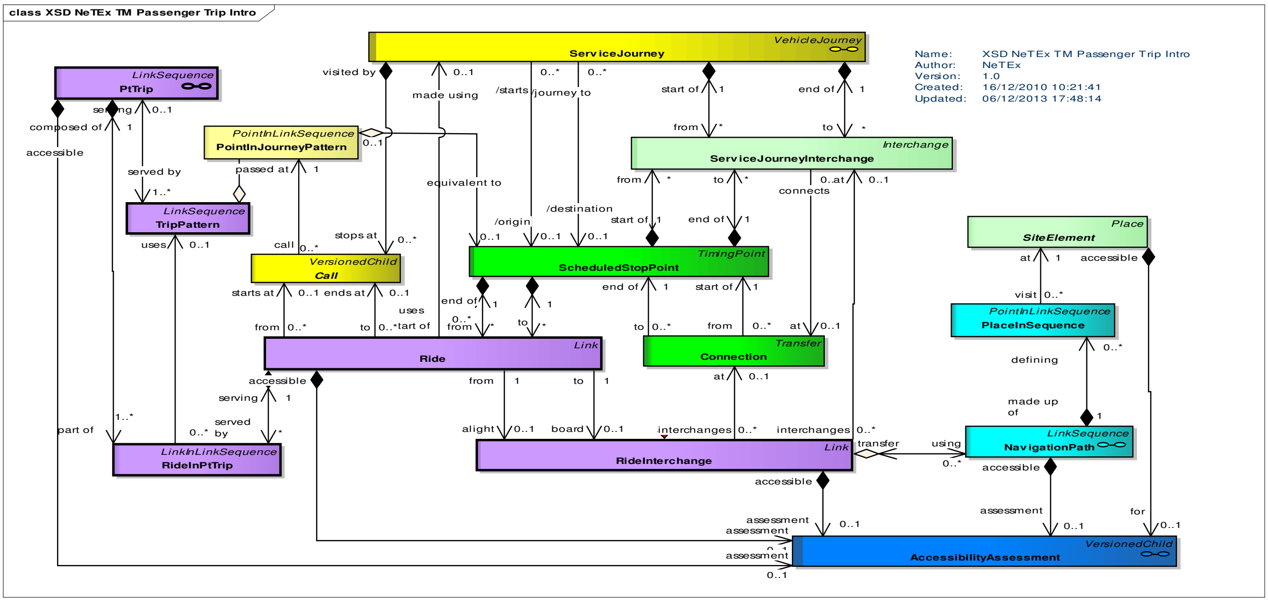


Figure 325 — Passenger Trip – Physical Model: Overview

D.1.1.2 Passenger Trip – Physical model – Details

The following figure shows the physical model for PASSENGER TRIP.

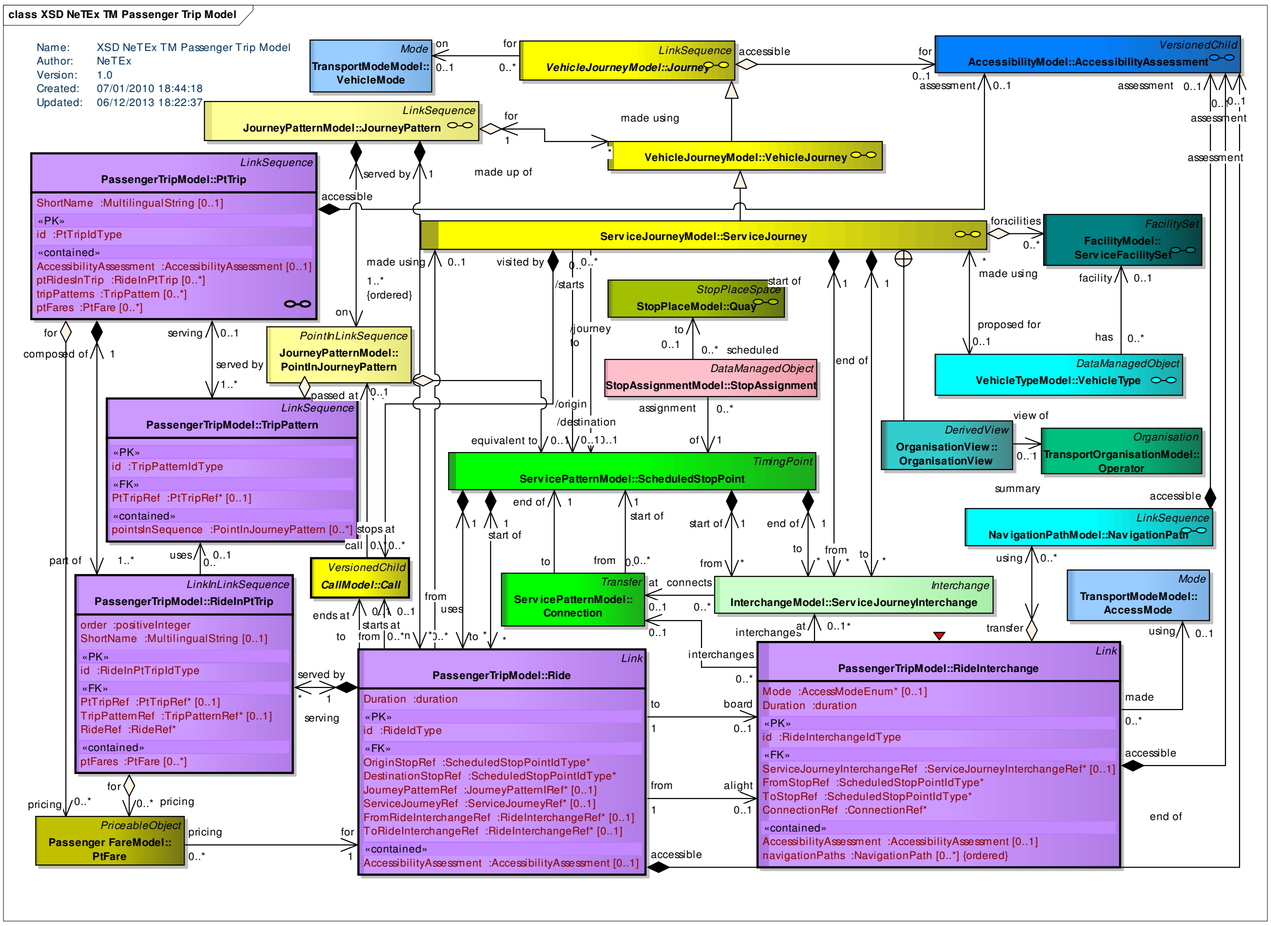


Figure 326 — Passenger Trip Query – Physical Model: Details

D.1.1.3 Passenger Trip – Attributes and XSD

D.1.1.3.1 PtTrip – Model Element

One or more RIDEs and the movements (usually walks) necessary to cover the corresponding CONNECTION links.

Table 293 – *PtTrip*– Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*LinkSequence*	::>	PT TRIP inherits from LINK SEEQUENCE.
«PK»	***id***	*PtTripIdType*	1:1	Identifier of PT PTRIP.
	ShortName	*MultilingualString*	0:1	Short Name of PT TRIP.
	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.
“cntd»	***ridesInPtTrip***	*RideInPtTrip*	0:*	RIDEs IN PT TRIP that make up PT TRIP.
“cntd »	***tripPatterns***	*TripPattern*	0:*	TRIP PATTERNs for PT TRIP.
“cntd »	***journeyProducts***	*JourneyProduct*	0:*	PJOURNEY PRODUCTs available for PT TRIP.
“cntd»	***ptFares***	*PtFare*	0:*	PT FAREs available for PT TRIP.

D.1.1.3.2 TripPattern – Model Element

The spatial pattern of a complete movement of a passenger (or another person, e.g. driver) from one PLACE of any sort to another. A trip may consist of one PT TRIP and the corresponding movements (usually walks) to cover the necessary ACCESS Links and CONNECTION Links, or of one walk only.

Table 294 –*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»	***PtTripRef***	*PtTripRef*	0:1	Reference to a PT TRIP for which this is a pattern.
“cntd»	***pointsIn-Sequence***	*PointInJourneyPattern*	0:*	POINTs IN JOURNEY PATTERN that make up TRIP PATTERN.

D.1.1.4 RideInPtTrip – Model Element

A RIDE in a PT TRIP with its order in that PT TRIP.

Table 295 –*RideInPtTrip* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*LinkInLinkSequence*	::>	RIDE IN PT TRIP inherits from LINK IN LINK SEQUENCE.
«PK»	***id***	*RideInPtTripIdType*	1:1	Identifier of RIDE IN PT TRIP.
	Order	*positiveInteger*	1:1	Order of ride within PT TRIP.
	ShortName	*MultilingualString*	0:1	Short Name of PT RIDE.
«FK»	***PtTripRef***	*PtTripRef*	0:1	Reference to a PT TRIP of which this is a ride.
«FK»	***RideRef***	*RideRef*	1:1	Reference to a PT RIDE.
FK»	***TripPatternRef***	*TripPatternRef*	0:*	TRIP PATTERN for PT RIDE.
"cntd»	***ptFares***	*PtFare*	0:*	PT FAREs for PT RIDE.

D.1.1.5 Ride – 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 296 –*Ride* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Link*	::>	RIDE inherits from LINK.
«PK»	***id***	*RideIdType*	1:1	Identifier of RIDE.
	Duration	*xsd:duration*	0:1	Duration of ride.
«FK»	***OriginStopRef***	*ScheduledStop-PointIdType*	**1:1**	Origin SCHEDULED STOP POINT of RIDE.
«FK»	***Destination-StopRef***	*ScheduledStopPoint-IdType*	**1:1**	Destination SCHEDULED STOP POINT of RIDE.
«FK»	***Journey-PatternRef***	*JourneyPatternIdType*	0:1	JOURNEY PATTERN followed by RIDE.
«FK»	***Service-JourneyRef***	*JourneyPatternIdType*	0:1	SERVICE JOURNEY used by RIDE.

«FK»	***FromCallRef***	*CallRef*	0:1	CALL at which RIDE starts.
«FK»	***ToCallRef***	*CallRef*	0:1	CALL at which RIDE ends.
«FK»	***FromRide-InterchangeRef***	*RideInterchangeRef*	0:1	RIDE INTERCHANGE from which RIDE starts.
«FK»	***ToRide-InterchangeRef***	*RideInterchangeRef*	0:1	RIDE INTERCHANGE at which RIDE ends.
“cntd»	***Accessibility-Assessment***	*AccessibilityAssessment*	0:1	ACCESSIBILITY ASSESSMENT for RIDE. See NeTEx Part1.

D.1.1.6 RideInterchange – 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 297 –*RideInterchange* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Link*	::>	RIDE INTERCHANGE inherits from LINK.
«PK»	***id***	*RideInterchangeIdType*	1:1	Identifier of RIDE INTERCHANGE.
	Duration	*xsd:duration*	0:1	Time to make interchange.
«FK»	***ServiceJourney-InterchangeRef***	*ServiceJourney-InterchangeRef*	0:1	SERVICE JOURNEY INTERCHANGE for RIDE INTERCHANGE.
«FK»	***FromStopRef***	*ScheduledStopPoint-IdType*	1:1	Arrival stop for RIDE INTERCHANGE.
«FK»	***ToStopRef***	*ScheduledStopPoint-IdType*	1:1	Departure stop for RIDE INTERCHANGE.
«FK»	***ConnectionRef***	*ConnectionRef*	0:1	CONNECTION link for RIDE INTERCHANGE.
	Mode	*AccessModeEnum*	0:1	Mode used to make RIDE INTERCHANGE.
“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 Model

D.1.2.1 Passenger Fare – Physical model – Overview

The following figure introduces the physical model for a PASSENGER FARE. A PT FARE describes a FARE PRODUCT and its PRICE available for a particular PT TRIP or PT RIDE.

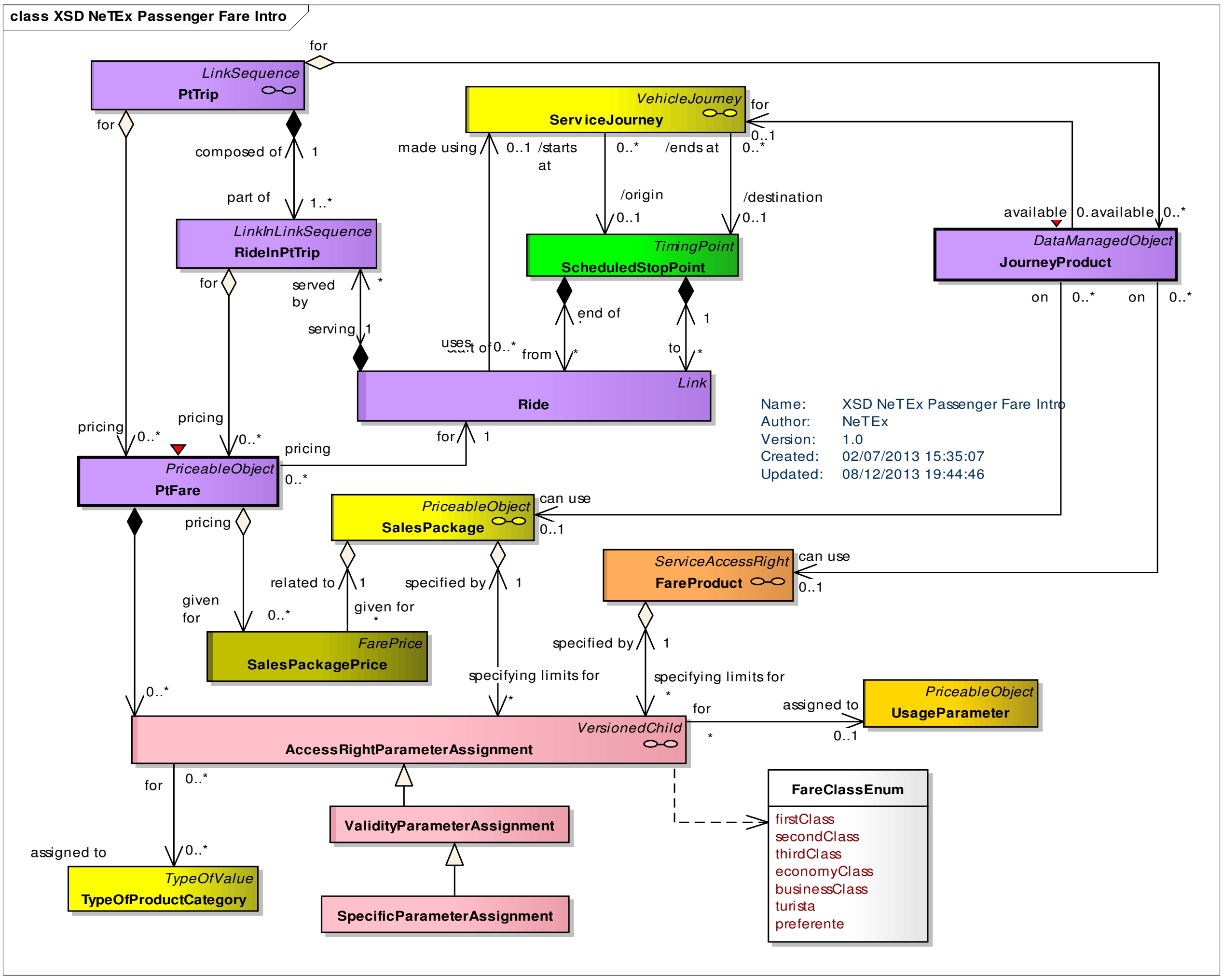


Figure 327 — Passenger Fare – Physical Model: Overview

D.1.2.2 Passenger Fare – Physical model – Details

The following figure shows the physical model for PASSENGER FARE.

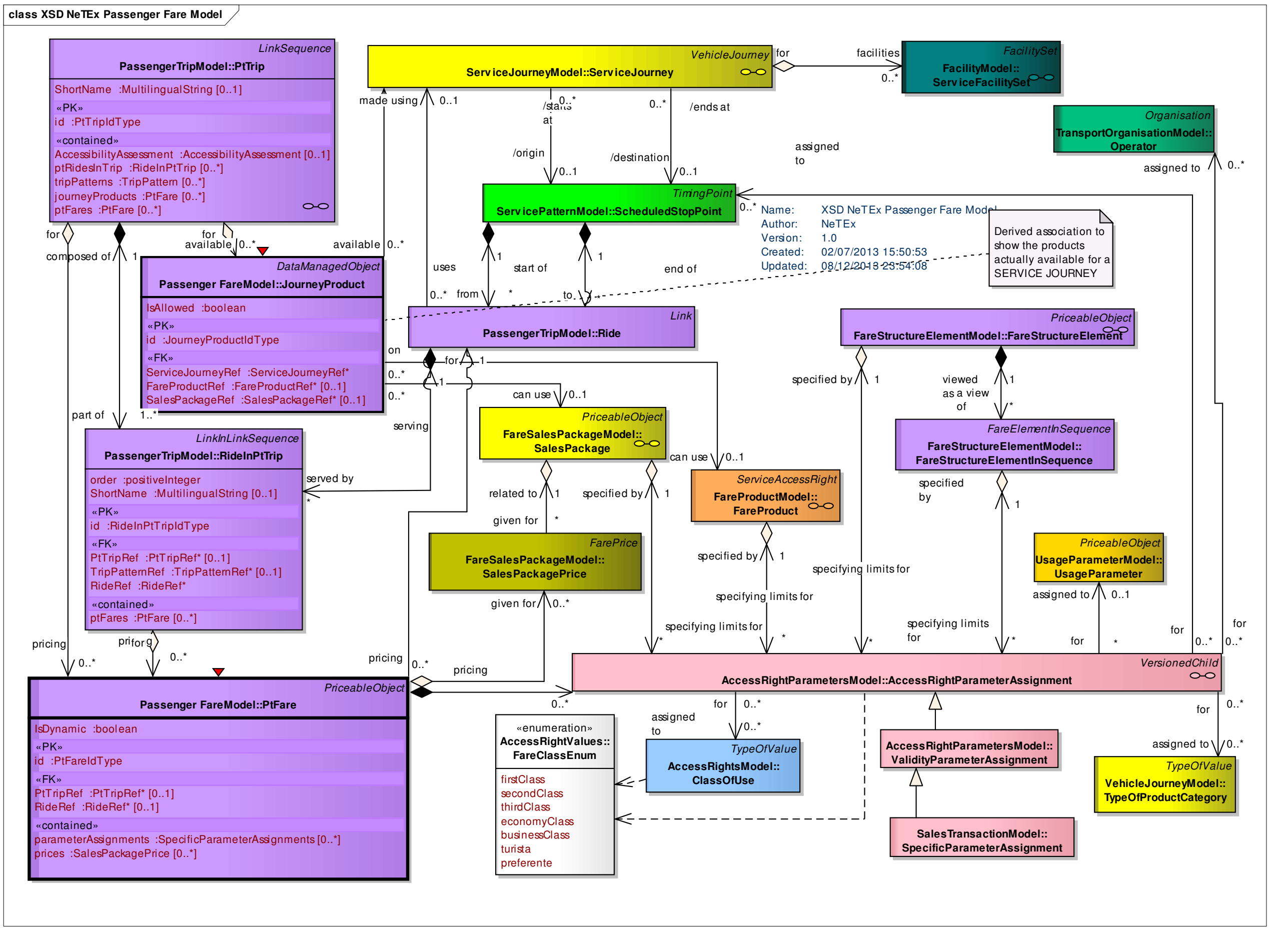


Figure 328 — Passenger Fare Query – Physical Model: Details

D.1.2.2.1 PtFare – Model Element

An available FARE for a specific a RIDE in a PT TRIP or for the whole trip. The fare may be restricted to particular products and conditions as described by an SPECIFIC PARAMETER ASSIGNMENT.

Table 298 – *PtFare* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	PT FARE inherits from FARE PRICE.
«PK»	***id***	*PrFareIdType*	1:1	Identifier of PT FARE.
	isDynamic	*boolean*	1:1	Whether the Fare was fetched dynamically.
«FK»	***PtTripRef***	*PtTripRef*	0:1	Reference to a PT TRIP of which this is a ride
«FK»	***RideRef***	*RideRef*	0:1	Reference to a PT RIDE.
“cntd»	***parameter-Assignments***	*SpecificParameter-Assignments*	0:*	ACCESS RIGHT PARAMETER ASSIGNMENTs relating to PT FARE.

“cntd»	***prices***	*SalesPackagePrice*	0:*	SALES PACKAGE PRICEs associated with the SALES PACKAGE.

D.1.2.2.2 JourneyProduct – Model Element

The availability of a FARE PACKAGE and or SALES PACKAGE for a SERVICE JOURNEY.

Table 299 – *JourneyProduct* – Element

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*DataManagedObject*	::>	JOURNEY PRODUCT inherits from DATA MANAGED OBJECT
«PK»	***id***	*JourneyProductIdType*	1:1	Identifier of JOURNEY PRODUCT.
«FK»	***Service-JourneyRef***	*ServiceJourneyRef*	1:1	SERVICE JOURNEY for which there is a FARE PRODUCT.
«FK»	***FareProductRef***	*FareProductRef*	0:1	FARE PRODUCT available for the SERVICE JOURNEY.
«FK»	***SalesPackageRef***	*SalesPackageRef*	0:1	SALES PACKAGE available for the SERVICE JOURNEY.

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 th 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.

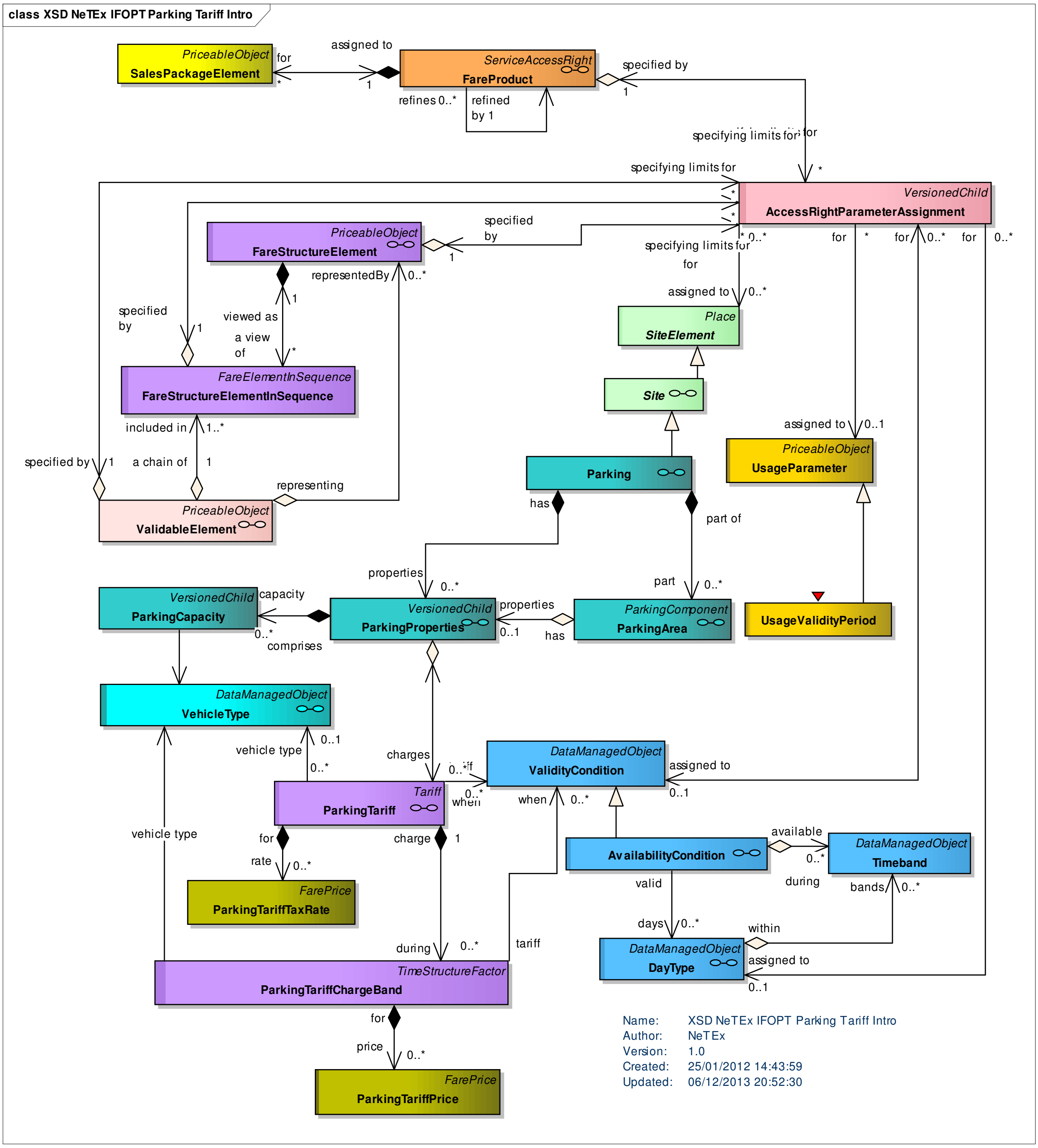


Figure 329 — Parking Tariff – Conceptual Model

E.1.1.1.1 Parking Tariff – Physical model

The following figure shows the physical model for PARKING TARIFFs.

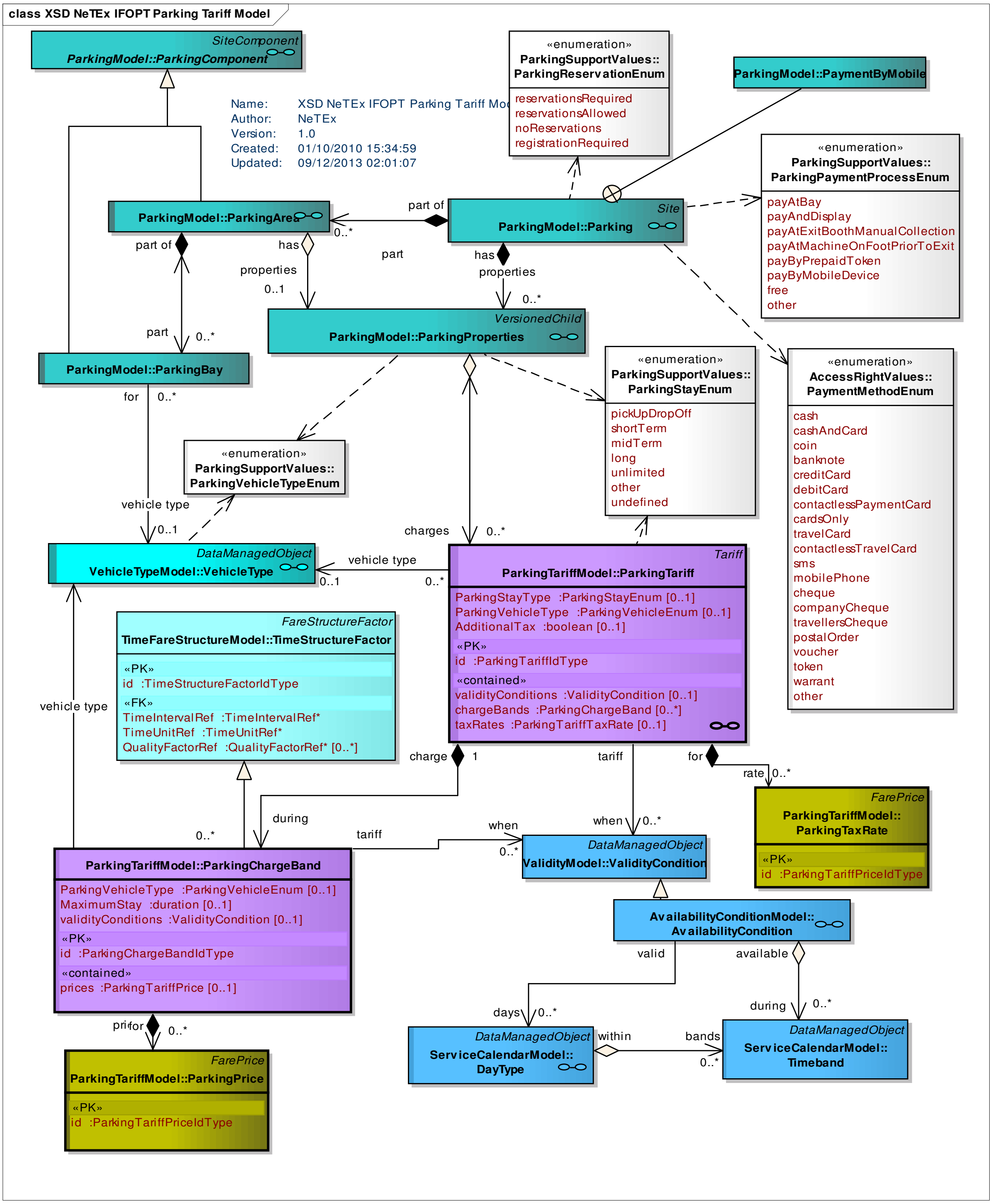


Figure 330 — Parking Tariff – Physical Model

E.1.1.1.2 Parking Tariff – Attributes and XSD

E.1.1.1.2.1 ParkingTariff

A set of parking CHARGE BANDS that describe the cost if using a PARKING or PARKING AREA.

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*Tariff*	::>	PARKING TARIFF inherits from TARIFF.
«PK»	***id***	*ParkingTariffIdType*	1:1	Identifier of PARKING TARIFF.
	ParkingStayType	*ParkingStayEnum*	0:1	Reference to PARKING STAY TYPE of PARKING TARIFF. See NeTEx Part1.
	Parking-VehicleType	*ParkingVehicleEnum*	0:1	Reference to VEHICLE TYPE of PARKING TARIFF. See NeTEx Part1
	AdditionalTax	*boolean*	0:1	Whether additional tax is charged on top of rates.
«contained»	***validity-Conditions***	*ValidityCondition*	0:1	VALIDITY CONDITIONS for PARKING TARIFF.
«contained»	***chargeBands***	*ParkingTariffChargeBand*	0:*	Charge bands of PARKING TARIFF.
«contained»	***taxRates***	*ParkingTaxRate*	0:1	Tax Rates for Parking.

E.1.1.1.2.2 ParkingChargeBand

Parking charges that describe the cost of using a PARKING or PARKING AREA for a given period.

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*TimeStructureFactor*	::>	PARKING CHARGE BAND inherits from TIME STRUCTURE FACTOR.
«PK»	***id***	*ParkingCharge-BandIdType*	1:1	Identifier of PARKING CHARGE BAND.
	Parking-VehicleType	*ParkingVehicleEnum*	0:1	Vehicle type for of PARKING CHARGE BAND.
	MaximumStay	*duration*	0:1	Maximum stay for PARKING CHARGE BAND.
«contained»	***prices***	*ParkingPrice*	0:1	Prices for PARKING CHARGE BAND.
	validity-Conditions	*ValidityCondition*	0:1	VALIDITY CONDITIONS of PARKING CHARGE BAND.

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	Cardinality	Description
::>	::>	*FarePrice*	::>	PARKING PRICE. inherits from FARE PRICE.
«PK»	***id***	*ParkingPriceIdType*	1:1	Identifier of PARKING PRICE.

E.1.1.1.2.4 ParkingTaxRate

A specialisation of FARE PRICE used to specify the tax rate on PARKING.

Classifi-cation	Name	Type	Cardinality	Description
::>	::>	*FarePrice*	::>	PARKING TAX RATE. inherits from FARE PRICE.
«PK»	***id***	*ParkingPriceIdType*	1:1	Identifier of PARKING TAX RATE.

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.