PD CEN/TS 16735:2015

BSI Standards Publication

Postal services — Extensible Common Structure and Representation for Postal Rates — EPR

...making excellence a habit.™

National foreword

This Published Document is the UK implementation of CEN/TS 16735:2015.

The UK participation in its preparation was entrusted to Technical Committee SVS/4, Postal services.

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

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

Published by BSI Standards Limited 2015

ISBN 978 0 580 86034 8
ICS 03.240

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

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 March 2015.

Amendments/corrigenda issued since publication

Date	Text affected

TECHNICAL SPECIFICATION
SPÉCIFICATION TECHNIQUE
TECHNISCHE SPEZIFIKATION

CEN/TS 16735

March 2015

ICS 03.240

English Version

Postal services - Extensible Common Structure and Representation for Postal Rates - EPR

Services postaux - Structure commune extensible et représentation pour les tarifs postaux

Postalische Dienstleistungen - Erweiterungsfähige gemeinsame Struktur und Repräsentation von Posttarifen - EPR

This Technical Specification (CEN/TS) was approved by CEN on 30 September 2014 for provisional application.

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

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

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

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

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

Ref. No. CEN/TS 16735:2015 E

Contents

Page

Foreword

This document (CEN/TS 16735:2015) has been prepared by Technical Committee CEN/TC 331 “Postal services”, the secretariat of which is held by NEN.

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

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

Introduction

Standards are important prerequisites for effective postal operations and for interconnecting the global network. Significant efficiency can be gained by implementing standards that define interconnectivity between postal operators and their customers. The postal sector is now facing the most challenging period in its long history. On the one hand, up to a few years ago the postal sector in its letter mail market has not faced competition from new entrants as a result of the historic reserved areas and letterbox monopoly. On the other hand it is confronting formidable competition from outside of its traditional boundaries (this type of competition is frequently referred to as “electronic substitution” or “electronic diversion”). One of the most effective responses to competitive pressure is innovation. Innovation involving an extensive use of modern information technology has proven to be particularly effective in many fields. However, information technology, while deeply permeating many aspects of the postal system, still lacks significant penetration when it is applied to postal product innovation, marketing and distribution. To alleviate this shortcoming UPU in collaboration with CEN/TC 331 developed and published the standard UPU S54. This standard is aimed at making a significant step forward in enabling more effective use of information technology in these areas. The standard however does not fully address effective computer-aided management of postal rates by both postal operators and their customers. This Technical Specification has been developed to provide postal operators, carriers, their customers and mailing industry at large with a tool designed for computerized creation, management and distribution of postal rates.

Mail communication involves a complex process of mail item production, induction, processing, transportation and delivery. It consists of a sequence of information and physical activities. The physical activities (required for mail item production, induction, sort, transportation and delivery) are supported by information processing activities that play an increasingly important role in the overall system. Complexity of the mail communication system also stems from the fact that different processes and activities (both physical and informational) are performed by different parties that are only loosely integrated and thus generally do not make optimal use of available information. In order to achieve a higher level of integration and better meet increasingly demanding requirements of customers, the mail communication system requires that different parties, entities and processes within the system should be integrated together via a set of standardized interfaces.

One of the most important interfaces is the set of rates for postal products. This is viewed in this Technical Specification as an information interface between posts (postal operators) and their customers, both mail senders and mail recipients. More specifically, from an information viewpoint, postal rates are data files that are created and managed by postal operators and carriers and distributed/shared with their customers.

By defining postal rates using a computerized description resulting in a standardized electronic document (postal rate file), postal operators can enable its customers to more effectively review and choose products suitable for their communication needs as well as provide for automated computing and application of postal rates during mail generation and finishing process.

The criteria that the postal rate file (as an information object) should meet are completeness, non-redundancy, consistency and unambiguous interpretation by mailers’ equipment.

In order to meet these criteria it is necessary to make rate information communications and processing computerized and automated, thus achieving ubiquity, consistency, speed and error tolerance typical for internet-based information exchanges.

This Technical Specification is an information description and presentation document. It is intended as a powerful and flexible tool capable of supporting both human and machine descriptions of postal rates with arbitrary complexity. In accordance with this Technical Specification, postal rate data files are messages that are structured in accordance with a specialized data representation language named Extensible Common Structure and Representation for Postal Rates (EPR). The EPR is organized as an application-specific version of the XML. It consists of flexible and adaptable constructs (elements) and thus by its very definition is extensible.

The EPPML standard (UPU S54-2) only briefly touches postal product prices (postal rates) and their use and communication. It defines rates communication by post and their acquisition by postal customers in a flexible manner leaving all technical details and structures of rate data outside of its scope. This Technical Specification is aimed at filling this gap.

Postal operators, their suppliers, mailing industry and mailing community at large developed many ad hoc rates structures and communication mechanisms designed to work in specific postal environments. In many instances the process of creating, updating and communicating postal rates (tables) is costly and time-consuming while there is a growing need to have fast (e.g. at the speed of the internet) distribution of rates and rate updates from postal operators to their customers (mailers and in some instances also mail recipients). Traditional methods as they are practiced today by a vast majority of postal operators involve distribution of postal rates to customers via physical documents or electronic documents that are not designed for automated interpretation and processing by computers. In case of commercial mailers who mail from tens to millions of mail unit per day this results in a manual process of rate data translation into data structures suitable for computer processing by mail generation and finishing equipment. This process is most frequently performed by trained professionals employed by the mailing industry (typically manufacturers of just mentioned mailing equipment), at a large cost and inconvenience to mailers and postal operators. However, an examination of postal rate data (typically rate tables) and their communication mechanisms as practiced by a majority of technically advanced postal operators revealed mostly common structure, features and approaches. This suggests that there has to exist a common, flexible and extensible data structure suitable for use as a basis for automated postal rate management and communication in all mailing environments, from SMEs to large enterprises. Modern general data representation language such as XML can be adapted for the purpose of defining such a structure, which is referred below as the Extensible Common Structure and Representation for Postal Rates or the EPR.

This Technical Specification presents detailed and practical description of the EPR while leaving decisions and implementation of actual postal product pricing (e.g. valuation) to postal operators, their regulators (if any) and the mailing industry. Postal operators and mailing industry can use this Technical Specification, as a guidance document for the development of their own software tools that, when implemented, will allow quick and cost effective postal rate management process.

This Technical Specification should be viewed by postal operators, their suppliers and mailing industry at large as a valuable data representation standard and a tool that will facilitate all stages of postal rates life-cycle processing and management from creation by postal operators to installation and use in customer mailing equipment and ultimate disposal.

“Postal products” are referred to below simply as “products”. The reader is referred to the UPU S54 standard for detailed discussion and definition of various aspects and attributes of postal products. Postal products are always associated with postal rates, which is a specific established term for postal product prices. When a class of the mail unit (i.e. letter, flat or parcel) is not mentioned it usually implies any suitable class. This Technical Specification uses the established definition and terminology for the “mail unit” as a physically constrained unit consisting of one or more mail items and/or lower level mail units, together with the means of constraint (as defined in the UPU Standards glossary). The terms “mailer” and “sender” are used interchangeably.

The notion of the postal product is a complex one. Unlike common products (food, clothing, books), that are bought and consumed every day, postal product consumption requires significant cooperation between product producers (postal operators and carriers) and consumers (mailers and recipients). An important part of cooperation involves computation of the rate for a postal product that is to be paid to a postal operator offering this product for sale. This computation is commonly known as a “rating” or “rating process” and most frequently performed during mail unit generation and finishing process that takes place in a mailer’s environment (this is due to prepayment requirement imposed by majority of postal operators or simply because of a need of the mailer to know the charges before the product is purchased). The rating process requires knowledge of three fundamental components, namely (a) measurable, computable or stored attributes of the mail unit to be mailed and the product to be used by the mailer in association with this mail unit, (b) prices or rates for postal products (typically postal rates tables), and (c) an algorithm that inputs (a) and (b) and outputs the rate for the mail unit and the associated product chosen by the mailer. The most common computational algorithm is known as a “look-up table” algorithm, which refers to a traditional human method of price determination when rates tables for each category of products are indexed by the attributes of mail unit and postal product (such as weight, distance from origination to destination, level of presort or worksharing, etc.). The algorithm is a simple search in a table whose rows and columns are indexed. The algorithm is normally stable and known in advance to both mailers and postal operators. Thus, for every act of mail rating, the detailed description of the algorithm does not need to be shared between mailers and postal operators. In this manner, the rate computation for an individual mail unit requires obtaining information from two sources, one controlled by the mailer (the mail unit itself and the product to be used in conjunction with the mail unit) and one controlled by the postal operator (postal rates tables). The main purpose of this Technical Specification is codification of a broad variety of postal rates tables in a standardized manner suitable for effective (without human involvement) computer processing in a mailer’s environment.

Postal customers are traditionally divided into several categories depending on the volume of mail units they produce and send and the level of automation and equipment they use for mail production.

There are individual users who create a few mail units using manual process. For the individual users many postal operators provide access to a postal website where such users can rate their mail using a feature typically known under the name ”postal calculator” or “calculator”. In this case while being connected to a postal website the user enters all relevant parameters of the mail unit and product and obtains the price for the service (rate). Individual users will have little benefit from this Technical Specification and thus are not a main target for its implementation.

Then there are small and medium size enterprises and businesses (SMEs), which typically produce and rate from tens to a few hundreds of mail units per day and use computerized mail finishing equipment such as mailing and franking machines. These users depend on manufacturers of mailing equipment for having up to date postal pricing (rate) information since rating in this environment is done automatically by the mailing machine. These users buy a broad variety of postal products (from letter mail to express and parcel products) and are expected to greatly benefit from this Technical Specification. For example pricing for postal products marketed to this class of customers can be updated and communicated to their mailing equipment instantly providing for a great speed, flexibility and convenience, which is very desirable.

Finally there are large enterprises that produce thousands and even millions of mail units per day using sophisticated high speed mail generation and finishing equipment such as inserters, mail assembly conveyors and the like. This type of customers frequently uses the so-called “negotiated rates” that are not published. The number of such customers is relatively small and they frequently have technical professionals involved in mailing (and rating) process. They are also expected to benefit from this Technical Specification since it will allow for quicker and more efficient rate updates and communication. For example a nimble and effective dynamic capacity management by postal operators involves stimulation of sales by lowering product prices where there is an excessive unutilized capacity for delivery of such products. This Technical Specification is aimed at enabling this capability by providing means for computer-to-computer communication and processing of postal rates.

With implementation of this Technical Specification postal operators and carriers are expected to achieve increased flexibility of new products introductions, nimble rates management commensurate with market and capacity conditions and will be able to introduce and verify new pricing models. Note that introduction of new products require many activities other than rating. These activities are outside of the scope of this Technical Specification.

This Technical Specification is applicable to all categories of postal products without restrictions.

Graphically, traditional rate management process can be represented by the following figure:

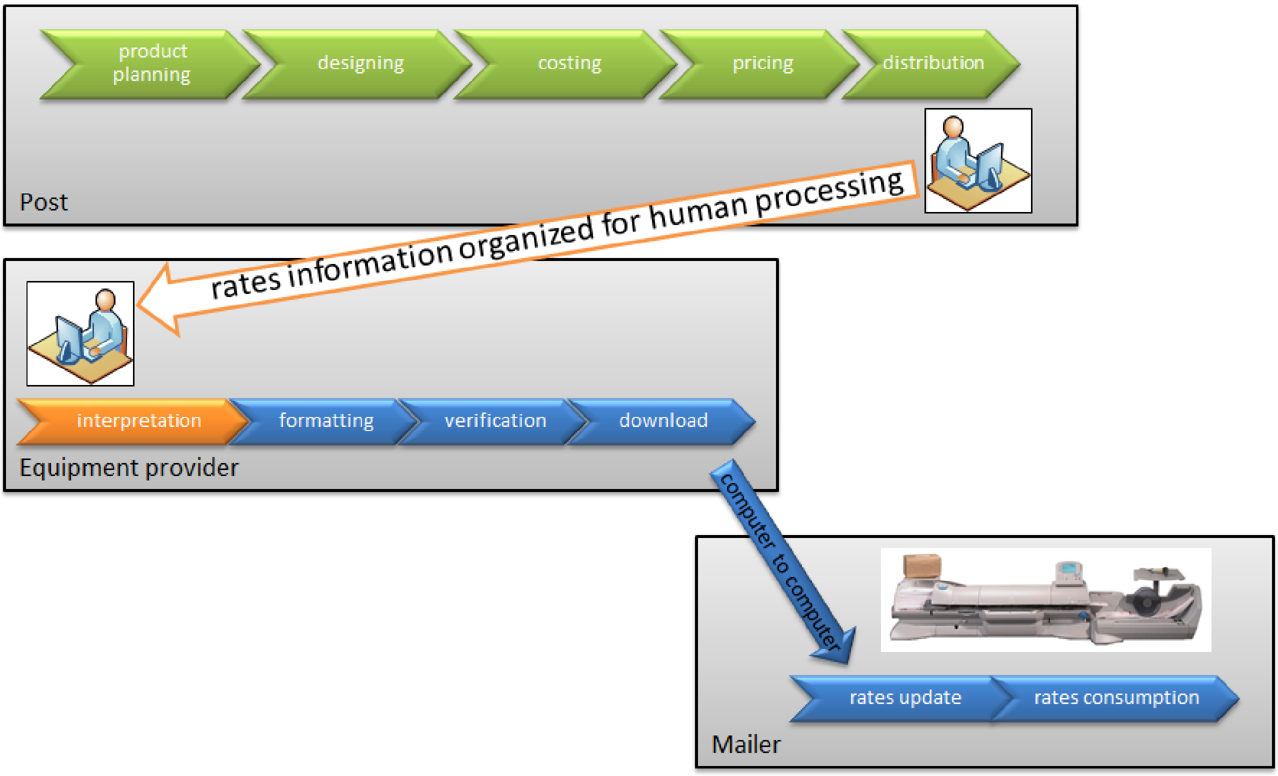


Figure 1 — Manual postal rates management and distribution process

Activities that take place in the postal environment (product planning, designing, costing and pricing) end up with the distribution of postal rates (predominantly organized as tables) to customers and their suppliers (mailing equipment manufacturers). In the traditional rates management process the rates data are organized for human consumption and requires quite significant manual (programming) effort to be converted into computer processable data at the major expense and inconvenience to mailers and postal operators. This Technical Specification is designed to provide a description of a rate data structure that when implemented will allow to replace the traditional postal rates management and distribution process by a new process graphically represented by the following figure:

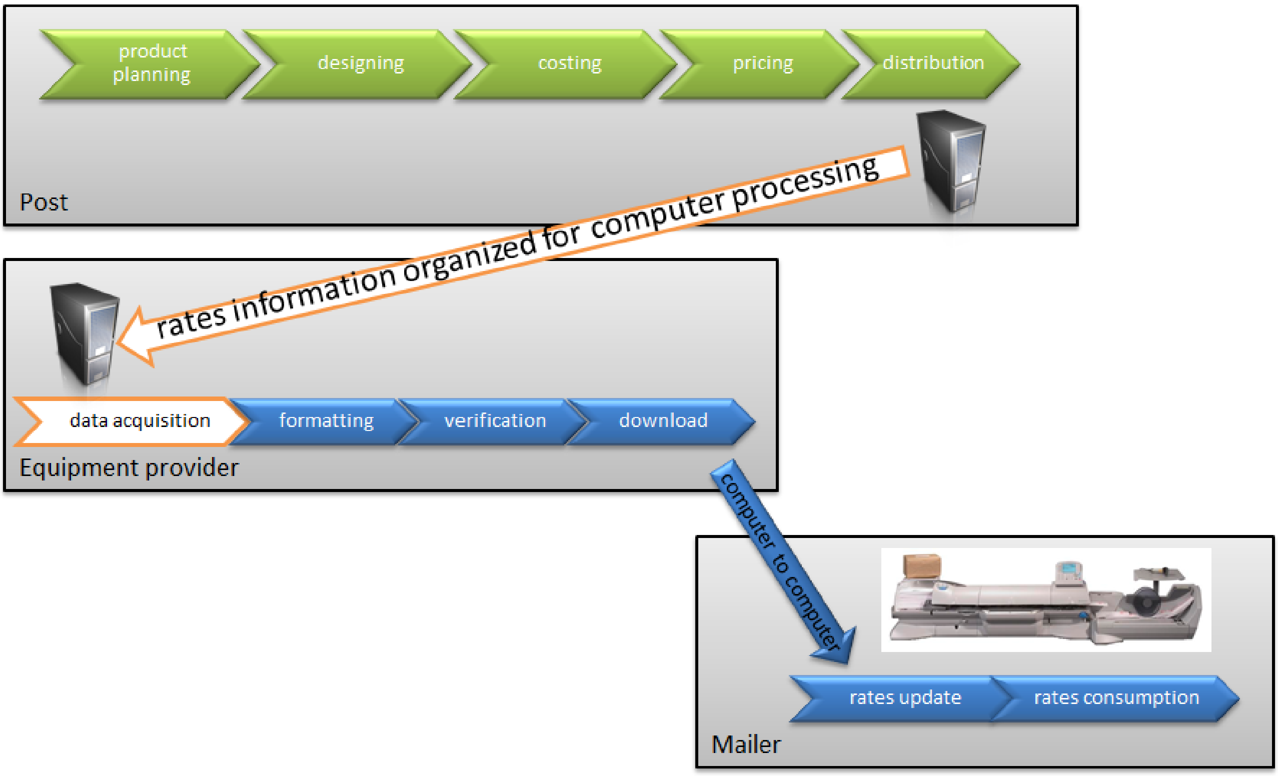


Figure 2 — Automated postal rates management and distribution process

In the automated rates management process postal rates data are organized for computer-to-computer communication. Rates computation in Figure 1 and Figure 2 implies mail rating algorithm, which automatically inputs and processes EPR data structured in accordance with the present Technical Specification.

The main body of this Technical Specification contains six substantive clauses and five informative annexes:

General Concepts: explains concepts and describes the application of EPR in defining postal rates by postal operators and processing them by mailers' computerized equipment.

EPR XML Schema: defines the EPR schema using subclauses each dedicated to a specific element of the schema. Whenever possible each element is provided with a detailed description, a motivation for its specified organization and an example.

However, the structure of the EPR schema enables significant flexibility in the representation of postal rates. Postal product pricing experts are encouraged to explore this flexibility of representations of postal rates supported by EPR and to identify those that are most suitable to their environment that is by necessity post-specific.

Expected readership of this Technical Specification includes postal financial, marketing and product development experts, and IT professionals as well as technical professionals in the mailing industry. Technical implementation of this Technical Specification requires proficiency with XML concepts and constructs and thus implementation related information in this document is intended primarily for IT professionals. Experts in other fields (e.g. finance, marketing and product development and inspection) will find it useful to read the Introduction and General Concept clauses and are encouraged to work in concert with IT professionals toward building an effective EPR-based rate management process.

1 Scope

This Technical Specification defines a uniform structure and meaning for the information that fully represents postal rates for a broad variety of postal products in all mail categories. The postal rates definition is viewed as an important interface between posts and their customers and as such will benefit from standardization. Such representation of postal rates allows automated systems to uniformly use postal rates as they are introduced for new products or updated for existing postal products by postal operators. A postal rate file (PRF) is an XML document, which fully describes postal products rates. It contains all necessary and sufficient information for both postal operators and their customers to efficiently create, respectively acquire, update and process postal product rates. The structure, types and constraints of XML elements in an XML document are defined by an XML schema. The Extensible Postal Rates (EPR) schema is the XML schema that governs Postal Rate Files.

This Technical Specification contains a complete description of the EPR schema, its hierarchical structure, information types and semantics of its elements.

This Technical Specification neither defines nor constraints how postal rates are created by postal operators but rather provide a powerful and flexible tool that supports efficient rates definition, management and distribution.

This Technical Specification does not define communication protocols that can be used by posts to distribute postal rates files to their customers. Suitable communication protocols are typically well known and already standardized (for example: Web Services, File Transport Protocol, email, etc.). The standard also does not define valuation of postal products as applied by postal operators and their customers.

2 Normative references

Not applicable.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE A number of common terms used in this document are defined in documents which are referred to in the Bibliography. Definition of frequently used or particularly important terms as well as other terms introduced in this document are given below.

3.1
dimensional (volumetric) weight
value that is produced as a result of calculation of a mail unit weight based on the International Air Transport Association (IATA) volumetric conversion factor and used for the purpose of rate calculation instead of mail unit actual measurable weight

Note 1 to entry: Using IATA conversion factor dimensional (volumetric) weight is computed by multiplying the length, the height and the width of the mail unit and dividing the result by 166 in^3/lbs (if mail unit dimensions are measured in inches) or by 6,000 cm^3/kg (if mail unit dimensions are measured in centimeters).

Note 2 to entry: Dimensional weight takes into account the density of the mail unit, which is the amount of space the mail unit occupies in relation to its measurable weight.

Note 3 to entry: The conversion factor established by IATA is somewhat arbitrary and can be revised by individual postal operators and carriers depending on the cost of transportation (air, ground, sea) and other factors.

3.2
calculated weight
value indicative of the weight of the mail unit that is used for the purpose of rate calculation

Note 1 to entry: The calculated weight is computed by using specified algorithms which may include selection and thresholding.

3.3
zone-based distance
value indicative of the distance between mail unit origin and destination that is computed using zone codes assigned to its origin and destination

Note 1 to entry: The zone codes are established by dividing the geographical area served by the postal network into smaller subareas, each of which is assigned a code value.

3.4
calculated distance
value indicative of the distance between origin and destination that is used is for the purpose of rate calculation

Note 1 to entry: The calculated distance is computed by using specified algorithms which may include selection and thresholding.

3.5
base amount
amount that represents a component of the overall postal rate that is computed by accounting for the weight, product and distance dependent elements of the postal rate

Note 1 to entry: The base amount roughly reflects contributions to the postal rate that are based on the cost of delivering the postal product for a given mail unit.

3.6
distance and weight-based fee
component of the postal rate that is a function of the calculated distance between origin and destination and the calculated weight of the mail unit

Note 1 to entry: The distance and weight-based fee represents a surcharge imposed on heavyweight mail units and/or mail units destined for destinations associated with additional delivery cost compared to other destinations.

3.7
over-dimension fee
component of the postal rate that is a function of the mail unit dimensions

Note 1 to entry: The over-dimension fee represents a surcharge imposed on mail units having one or several of their dimensions outside of a pre-determined range and reflects an extra cost associated with processing, transportation and delivery of such units.

3.8
value-based fee
component of the postal rate that is a function of the value of the mail unit content

3.9
quantity-based fee
component of the postal rate that is a function of the number of mail units inducted into the postal distribution network as a part of mailing submission

Note 1 to entry: The quantity-based fee can be positive, when the number of mail pieces in the submission is below a pre-determined threshold or negative (discount) when this number is above the threshold.

3.10
amount fee
component of the postal rate that is a function of the base amount

3.11
date/time-based fee
component of the postal rate that is dependent on the date (including time) of the mail unit(s) induction

3.12
value-added service-based fee
component of the postal rate that is dependent on value-added services provided in addition to the services typical for basic products

Note 1 to entry: The value-added service-based fee reflects the view that postal products consists of basic delivery services and additional services such as track and trace and proof of deposit or delivery.

3.13
generic fee
component of the postal rate that is dependent on a number of possible cost, competitive or regulatory considerations and factors

Note 1 to entry: The generic fee could be positive or negative and is included for extensibility purposes. For example a generic fee may be imposed depending on the type of mail unit recipient (e.g. business or residential) or on the type of mailer (e.g. non-profit or commercial).

3.14
algorithm ID
identifier that uniquely references an algorithm involved in the postal rate computation

Note 1 to entry: The Algorithm ID provides for unambiguous understanding between postal operators and customers concerning the method and steps involved in the postal rate computation for a given product and mail unit.

3.15
range descriptor
formal mechanism that identifies an interval containing the value of a parameter involved in the postal rate computation

Note 1 to entry: The range descriptor consists of one or two boundary values that determine minimum and/or maximum value that a given parameter is allowed to take, for example a range of weights or distances.

3.16
zone code
value assigned to a geographic area

Note 1 to entry: The zone codes are established by dividing the entire geographical area served by the postal network into smaller subareas, each of which is assigned with a single code.

Note 2 to entry: Zone codes can be expressed in a multitude of ways (integers, alphanumeric codes, etc.). Different postal product providers typically employ different representation for the zone codes.

3.17
mailer category
input to the postal rate tax computation that is dependent on the type of the sender (mailer)

Note 1 to entry: The mailer category allows distinguishing between different types of mailers (e.g. social or business) for the purpose of tax imposition.

3.18
content category
input to the customs fees computation that is a function of the mail unit content

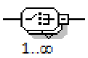
Note 1 to entry: The content category allows distinguishing between dutiable and non-dutiable merchandise (e.g. merchandise that is intended for commercial sales as opposed to the merchandise indented for personal consumption).

4 Symbols and abbreviations

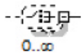
The following graphic conventions are used throughout the document:

 This symbol indicates that a sequence containing all descendant elements is expected.

 This symbol indicates a choice of one element from its descendants.



This symbol indicates a choice of one or more elements from its descendants. The text refers to it as *(one…many)*.

The use of dotted lines indicates that the choice is optional, for example 

means zero or more descendants are allowed. The text refers to it as *(zero…many)*.The descriptor *“(one…many)”* designates the cardinality of an element when one or more instances of this element are allowed. The descriptor “*(zero…many)*” is used to indicate an optional element that can be present in multiple instances.

The symbol 

indicates an algorithm. The text inside the rounded rectangle indicates the name of the algorithm.

5 General Concepts

5.1 Extensible Common Structure and Representation for Postal Rates (EPR)

As an informational object, postal rates can be defined by a formal (computerized) information structure consisting of elements organized as an extensible collection of attributes, their values and relations. When described in EPR, postal rates form an input to a rate computation that takes place during mail generation/finishing process. The diagram in Figure 3 depicts interactions between mailers and postal operators that take place in the context of mail rating. The rate computation for a given mail unit (MU) is preceded by the process of selection of the postal product by the mailer which takes as inputs the MU attributes, mailer requirements and the postal product description (e.g. postal product definition file). Following product selection the mailer computing system calculates the rate by taking the selected product, the postal rate file, the MU attributes and mailer requirements as inputs.

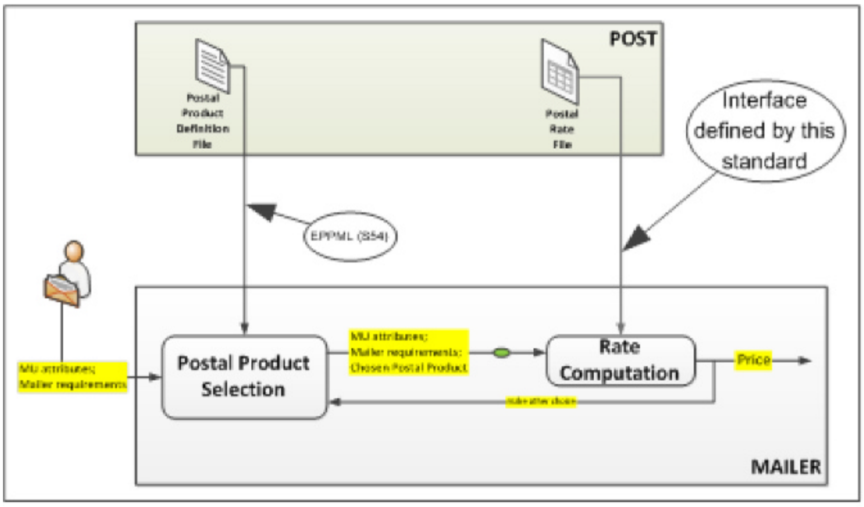


Figure 3 — Interaction between mailers and postal operator

NOTE While the process of product selection by mailers can be aided by a formal XML product description (as specified in the UPU S54 standard (EPPML)), the use of this document is not predicated on the use of the UPU S54 standard. When a product description is provided through traditional means (i.e. postal documentation) the product selection is done by a human operator.

The rate computation can be conveniently decomposed into four calculation categories, namely calculation of the base amount, fees, taxes and customs charges. Addition of the four values resulting from these calculations produces the final rate for the mail unit and its associated product. It should be noted that only base rate computation is always required while three other categories are optional. Similarly, within each calculations category (base rate, fees, taxes and customs charges) some computations are optional depending on the product. The only mandatory computation is the base amount computation (see Figure 4 below). This is the case for example with the so-called ‘flat rate” products. The method of rate calculation based on decomposition of the overall computation into several categories reflects the traditional costing/pricing approach as practiced by most postal operators and has its roots in a regulated environment when postal operators had to provide justification for their prices. It also reflects cost-based pricing methods that are at the core of pricing methodology for majority of postal operators and carriers. The following diagram depicts the overall rate computation:

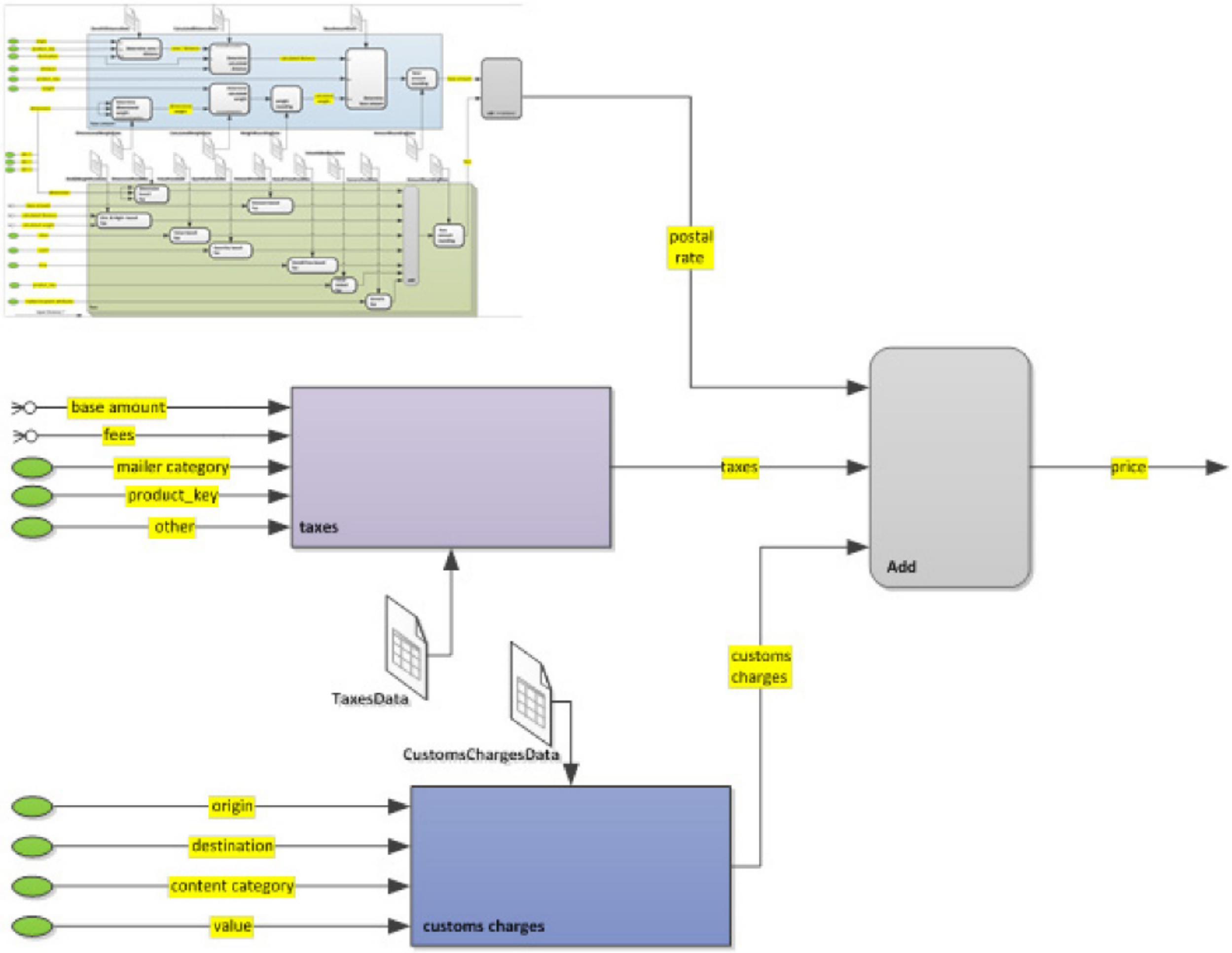


Figure 4 — Overall rate computation process

Taxes and customs charges are added to the computed postal rate (the sum of the base amount and fees) to arrive at the total price due from the mailer for a product. The algorithm for the tax computation is the “look up table” algorithm. It requires five inputs (shown on the left), two resulting from base amount and fees computations described below as well as three other parameters (mailer category, product key and “other”, that is included to enable additional input as it may be required). These parameters are selected by mailer before rating process commences. Another input to the tax computation process is a data structure (typically one or two dimensional table) shown as “TaxesData”. TaxesData are supplied by the postal operator as a part of the EPR data file. This and similar data structures represented in XML are the main goal of this Technical Specification. They are described in detail in the main body of this document. Similarly if custom charges are paid by the mailer, the look up table algorithm for computing customs’ charges in the diagram requires input of origin, destination, content category (e.g. commercial merchandise or merchandise for personal use) and the value of the content. These inputs are combined with a data structure “CustomsChargesData” supplied by the postal operator (or customs) that contain customs-defined parameters stored in one, two or multi-dimensional table necessary for computing custom duties. This data structure represented in XML is described in detail in the main body of this document.

The upper portion of the diagram in Figure 4 represents the core postal rate computation process. It is shown in detail in the following diagram:

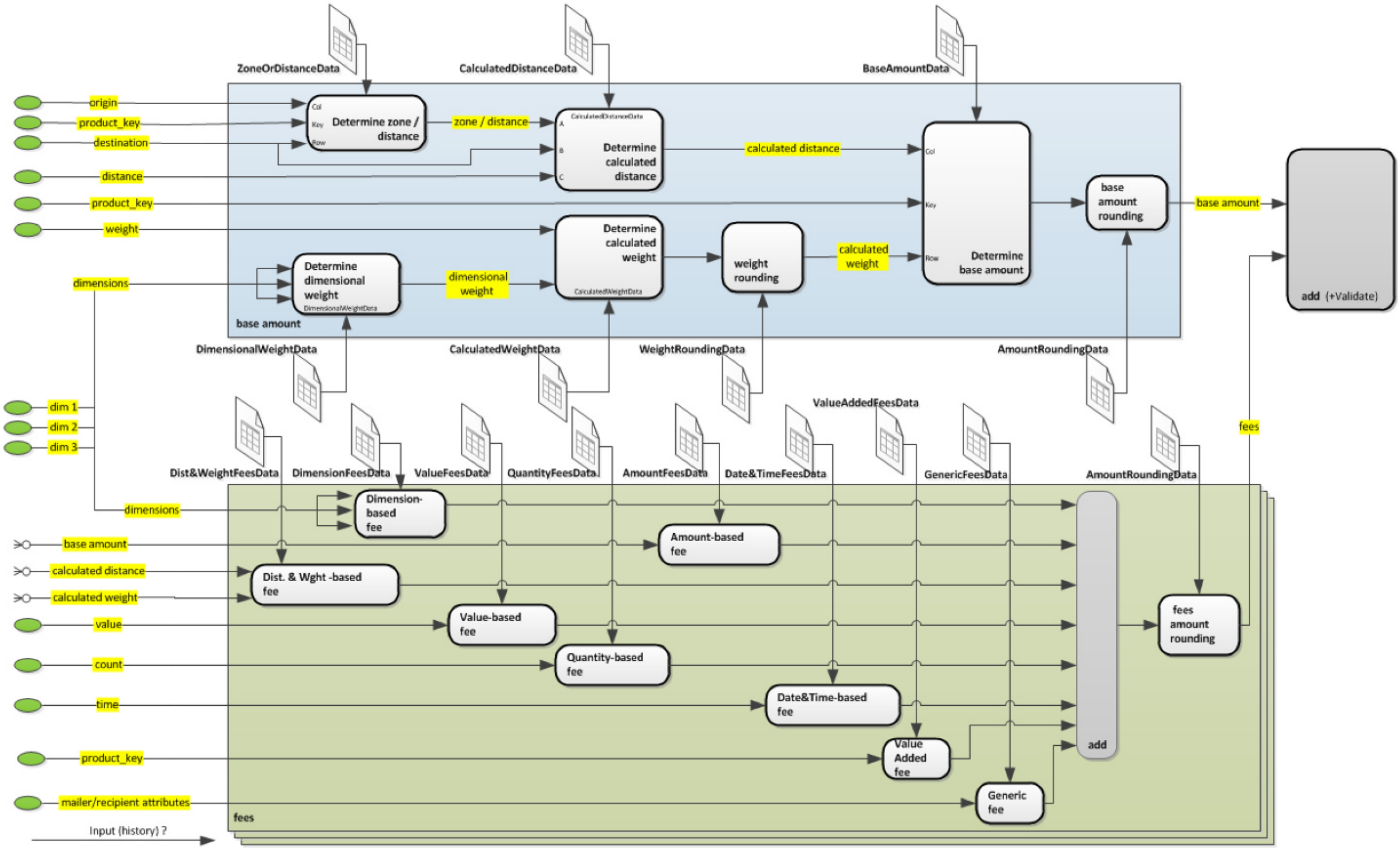


Figure 5 — Postal rate computation process

In the diagram of Figure 5 the upper portion represents the calculation of the base amount. The rounded rectangles in the diagram represent algorithms. Basic algorithms (zone-based distance, dimensional weight/index, calculated weight and distance, base amount and weight/amount rounding computations) are depicted in the middle of the rectangular shaded area. These basic algorithms are simple, well-known algorithms, which are explicitly understood without ambiguity and need not be described in any detail. These algorithms are “look up table”, thresholding, input selection and rounding algorithms. In this document they are identified by their unique identifiers (AlgorithmID). The two-dimensional data table is a widely used data structure that supports search by lookup table algorithms. The data in the table is assumed to be organized in rows and columns. The first row contains the headings of all the columns, and similarly, the first column contains the headings of all the rows. The result of the search into the two dimensional table is found by first finding matches for the two input values in the first row and the first column. The first input value is matched against the column headings (search into the first row) resulting in the column index and the second input value is matched against the row headings (search into the first column which is made up by the first element of all the rows) resulting in the row index. If matches for two input values are found and expressed as the row index and the column index, they identify the value in the table that is the result of the search.

The description of the basic algorithms for postal rate calculation can be defined formally in any suitable computer language or pseudo language. Such descriptions are outside the scope of this document because interpretations of the basic algorithms by mailers and posts do not cause ambiguity or misinterpretation.

The sequence of execution of basic algorithms involved in the overall rate computation is presented in the diagrams of Figures 4 and 5. This sequence uniquely determines the result of the overall rate computation.

There are two categories of inputs to the base amount computation. Inputs that are shown on the left with small ovals represent mail unit’ and product’ attributes that are measured and/or selected and automatically or manually entered into the base amount computation process by the mailer. Most of these inputs are self-explanatory (their definitions are given in Clause 3). The “Product Key “ input is an index that allows conversion of any multi-dimensional table into a collection of two-dimensional tables optimally suitable for look up table search algorithm. Each of these two-dimensional tables is indexed (identified) by the Product Key. The Product Key captures multiple diverse inputs involved in the product costing and pricing such as machinability of the mail unit (a term used for identifying objects sortable by automatic mail sorting machines), a level of mail preparation and induction, value added services and similar. The level of mail preparation (mail pre-sort performed by the mailer and a type of facility where the mail is inducted into the postal stream) affects the cost of mail sorting and transportation and serve as a basis for various work-sharing discounts prevalent in North America and some European countries. Value-added services such as mail tracking and tracing or mail redirection also affect the cost of the service and also reflected in the Product Key index. Generally all fees are dependent on the product and thus the product key serves as an input to fees calculation. The Product Key takes a number of values that is defined by the number of possible combinations of inputs (e.g. number of different pre-sort schemes, types of induction facilities, value-added features and other parameters that serve as inputs into Product Key determination). The postal product provider performs computing of the product key during a set up phase. It is done by assigning an identifier to each of the two-dimensional tables resulting from conversion of multidimensional tables into collections of two-dimensional tables. The conversion process is straightforward and is outside the scope of this document. The product keys are expected to be electronically supplied to mailers and mailing equipment manufacturers by the postal product provider as a part of the postal rates data structured in accordance with this document.

The concept of service code as defined in UPU S25 may be used as a surrogate for the product key. Service codes are stored in publicly accessible code tables by UPU. Note however that service codes were defined primarily for exchanges of data between postal operators and were not intended for use in conjunction with communication of rate information to postal customers. Therefore, the application of service codes as product keys in the context of this document should be used with caution.

It should be noted that the separation of core postal rate computation into two categories (base amount and fees) is somewhat arbitrary and it is done for the convenience of adaptation from human to computer-centric rate management. For example, base amount computation as described above involves the Product Key that reflects worksharing discounts and can be also moved into the fees computation. It should be emphasized however that this separation neither affects the results of the computation nor it restricts freedom of software architects to design software tools for EPR-based rate management optimized for custom requirements of postal operators, customers and mailing industry.

The other category of inputs are various one or two or multi-dimensional tables provided to mailers by the postal operator or the carrier. The structure and meaning of the elements in these tables form the object of this document. In the diagram they are identified by an icon (diamond) of a multi or two-dimensional table (which is what they mostly are) and are marked as “DimensionalWeightData”, “CalculatedWeightData”, “ZoneOrDistanceData”, “CalculatedDistanceData”, “BaseAmountData”, …,” AmountRoundingData”. They are defined and constructed using XML schema whose structure, semantics and elements are described in detail in the main body of this document. Similarly the bottom portion the diagram of Figure 5 represent calculation of fees associated with various corrections applied to the base amount. The fees can be positive (surcharges e.g. for over size) or negative (discounts e.g. volume discount) depending on the type of correction to be applied. Just like the case of the algorithm for the base amount computation, algorithms required for fees’ calculation are shown in the middle of the shaded area and identified by their names (dimension-based fee, distance and weight based fee, value and quantity based fees, amount, date/time based fees and generic fee). The generic fee is a placeholder for any type of fees (positive or negative) that postal product provider could wish to impose in the future. It is added here for extensibility purposes. Similarly to the base amount computation, inputs to the fee calculation algorithm are twofold. Input from the mailer is depicted on the left as dimensions, base amount, product, calculated distance and weight, value, count, date/time, mailer/recipient attributes (e.g. residential or commercial) and others such as history of mailer-postal product provider relations (implying that certain past behaviour of the mailer could affect rates that mailer shall pay for certain products). The other inputs are rate tables “DistanceAndWeightFeesData” through “GenericFeesData” that are supplied by the postal product provider. Just as before, they are described in detail in the main body of this document. Finally, Figure 5 also depicts how the base amount and fees are added together to arrive at the postal rate. This addition process may also require input from the postal rate data shown as “AmountRoundingData” (for example rounding), which is again further described in the main body of this document.

5.2 Using EPR for design, communication and processing of Postal Rates

Rate design (creation) process (or pricing) takes place in the postal product providers’ environment. The rate design process resulting in postal rate data is assumed to be at the total discretion of postal operators and carriers and is outside of the scope of the present document. In the EPR-enabled environment the last step in the rate design process prior to distribution of the rates to customers is a conversion of the postal rate data into a XML document (an EPR file) structured in accordance with this document. This conversion is envisioned to be performed with a specialized software tool (e.g. an EPR editor), architecture and design of which is also outside the scope of this document. Postal operators will be encourage to design and implement software tools suitable for working with their rates data within their IT environments. Once the EPR file has been created it is made accessible to postal customers and their suppliers via a number of existing and widely used electronic communication means and protocols (websites, file transfer protocols, e-mail, etc.). These communication means and protocols are equally outside of the scope of this document.

The EPR file downloaded into customers’ computers or computer-driven mailing equipment is automatically processed and adapted for mail rating process as described in the Introduction. This also can be done with the help of software tools that are expected to be developed by the mailing and software industries. Design and implementation of such tools is at the sole discretion of the mailing equipment manufacturers and software tools providers and is also outside the scope of this document. Thus, this document specifies only EPR data structure shared between postal product providers, their customers and suppliers.

5.3 Time

The measurement and referencing of time is a critical component in the context of this document. The definition of date in the UPU Standards glossary includes a specification of the local time offset from a standard reference time such as UTC or GMT. Consequently, in this document, the date is expressed as an XML string structured in accordance with the UPU Standards glossary definition of Date.

The default value of time is the local time for the locality where the rating process takes place.

EXAMPLE The preparation and rating of the mail unit.

Since location information (origin) is always included, this allows for unambiguous interpretation of timing for the rating and for all parties involved regardless of their location.

6 EPR XML Schema

6.1 Introduction

According to W3C consortium, XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents [1].

A postal rate file (PRF) is an XML document which contains all necessary data to calculate the price of a given mail unit. The attributes of the given mail unit used in the determination of the price include the postal product, physical and informational attributes of the mail unit and mailers options.

The structure and meaning of the XML elements in a PRF are defined by an XML schema, named EPR schema to distinguish its purpose and use. As such, the syntax of postal rate files is governed by the EPR schema. The main purpose of this document is to describe the EPR schema.

This document describes the hierarchical structure of PRF according to the EPR schema. Diagrams of XML schema elements illustrate the hierarchy of concepts in EPR. Each subclause contains such diagrams followed by descriptions of the XML elements, their content and semantics. Most subclauses contain examples.

The elements of the EPR schema described in the following subclauses govern the structure and meaning of data used by algorithms shown in Figure 5. Each subclause describes a top element of the EPR schema. Each top element of the EPR schema corresponds to data identified in Figure 5. For example, Subclause 6.6 Base Amount corresponds to the BaseAmount data in Figure 5 that serves as input to the algorithm for determination of the base amount.

6.2 Structure of EPR Type

6.2.1 General

Product rate files are XML documents that are valid according to the EPR schema. Its root element is named PostalRate. The PostalRate element is of type PostalRateType. Accordingly, the XML schema described in this document has the PostalRateType at the top level of its hierarchy, as shown in Figure 6. The PostalRateType contains all elements necessary to define postal rate data. The term mail unit is used throughout this document to indicate either a single mail item or a mail unit created through the aggregation of smaller mail units including individual mail items. A mail item is a special case of a mail unit. Each mail unit is subject to one and only one postal product.

Figure 6 shows the elements at the top level of the postal rate file (PRF). The following table provides an overview of the postal rate elements while the PRF attributes are specified in Clause 6.

Postal Rates – top level elements			
Element name	**Type**	**Meaning of XML element**	**Comments and Examples**
DimensionalWeight	Complex Type	Data supplied by the postal product provider necessary to determine the dimensional weight of a mail unit according to the rules for a given postal product.	Determination of dimensional weight requires information stored in the element DimensionalWeight and knowledge of actual dimensions of the mail unit. Dimensions of the mail unit are provided by the mailer. See 6.3 about DimensionalWeight.
CalculatedWeight	Complex Type	Data supplied by the postal product provider necessary to determine the calculated weight of the mail unit.	Determination of calculated weight requires information stored in the element CalculatedWeight and information derived from mail unit attribute values.
WeightRounding	Complex Type	Data supplied by the postal product provider necessary to determine the weight to be used in computing postal rate.	Example: Rounding to the next integer value (always linked to the unit of measurement).
ZoneOrDistance	Complex Type	Data supplied by the postal product provider necessary to determine the zone (or distance) to be used in computing postal rate.	In case when distance is required, the origin and destination data shall be expressed in appropriate form.
CalculatedDistance	Complex Type	Data supplied by the postal product provider necessary to determine the calculated distance to be used in computing postal rate.	When the postal rate is a function of distance one of the following three representations of distance is used: codified distance (zone), destination code or distance.
BaseAmount	Complex Type	Data supplied by the postal product provider necessary to determine the base amount.	The determination of the base amount requires (in addition to BaseAmount data) the calculated distance, the product key and the calculated weight.
AmountRounding	Complex Type	Data supplied by the postal product provider necessary for rounding the amount to be used in computing postal rate.	EXAMPLE Rounding to next integer value.
DistanceAndWeightFees	Complex Type	Data supplied by the postal product provider necessary to determine the fees that are dependent of distance and/or weight.	The determination of the distance and weight fees requires the calculated distance and/or the calculated weight.

Postal Rates – top level elements			
Element name	**Type**	**Meaning of XML element**	**Comments and Examples**
DimensionFees	Complex Type	Data supplied by the postal product provider necessary to determine the fees that are dependent of mail unit dimensions.	The determination of the dimensions-based fee amount requires (in addition to DimensionFees data) the mail unit dimensions provided by the mailer.
ValueFees	Complex Type	Data supplied by the postal product provider necessary to determine the fees that are dependent of mail unit value as declared by the mailer.	The determination of the value-based fee amount requires (in addition to ValueFees data) the mail unit value provided by the mailer.
QuantityFees	Complex Type	Data supplied by the postal product provider necessary to determine the fees that are dependent of the number of mail units inducted into postal operator/carrier stream within a given period of time.	The determination of the quantity-based fee amount requires (in addition to QuantityFees data) the number of mail units to be inducted inputted into rating process by the mailer.
AmountFees	Complex Type	Data supplied by the postal product provider necessary to determine the fees that are dependent on the base amount.	The determination of the amount-based fee amount requires (in addition to AmountFees data) the knowledge of the base amount.
DateTimeFees	Complex Type	Data supplied by the postal product provider necessary to determine the fees that are dependent on date and time of the mail unit induction.	The determination of the date and time-based fee amount requires (in addition to DateAndTimeFees data) the date and/or time data provided by the mailer.
ValueAddedFees	Complex Type	Data supplied by the postal product provider necessary to determine value-added fees.	The determination of the value-added fee amount requires (in addition to ValueAddedFees data) knowledge of the product chosen by the mailer for the given mail unit.
GenericFees	Complex Type	Data supplied by the postal product provider necessary to determine generic fees.	The generic fee is a placeholder for any type of fees (positive or negative) that the postal product provider could wish to impose in the future.

Postal Rates – top level elements			
Element name	**Type**	**Meaning of XML element**	**Comments and Examples**
FeesCompatibilityAndDependency	Complex Type	Data supplied by the postal product provider necessary to validate compatibility and dependency of different types of fees.	The validation algorithm requires FeesCompatibilityAndDependencyData and the fees values.
Taxes	Complex Type	Data supplied by the postal product provider necessary to determine taxes for postal services.	The determination of the tax amount requires (in addition to Taxes data) the product chosen by the mailer, the mailer category, the base amount and fees amount for the given mail unit.
CustomsCharges	Complex Type	Data supplied by the postal product provider necessary to determine customs charges.	The determination of the customs charges requires (in addition to CustomsCharges data) the origin, destination, content category and the value for the given mail unit.
RatesValidityPeriod	Complex Type	Data supplied by the postal product provider necessary to determine the time interval(s) when the postal rates are valid.	The time interval is understood as a continuous period of time with defined beginning and ending. The rates may be defined to be valid during one or more intervals.
TermsAndAgreements	Complex Type	Data supplied by the postal product provider necessary to retrieve applicable legal documents.	EXAMPLE Legal document containing Terms and Agreements that shall be presented to (and/or acknowledged by) the mailer.
InfoForHumanConsumption	Complex Type	Data supplied by the postal product provider that improves the interaction with the post	This info supports the creation of a complete user interface for use of the postal product. It fulfills some legal requirements, too. Examples: customer service contacts, preparation instructions.

Most top-level elements of the EPR schema contain other elements that are of type Table2D or AlgorithmAndValue. Both are described in detail in the Foundation Types subclause (6.21).

The following figure shows the top-level schema for the postal rates data (EPR schema).

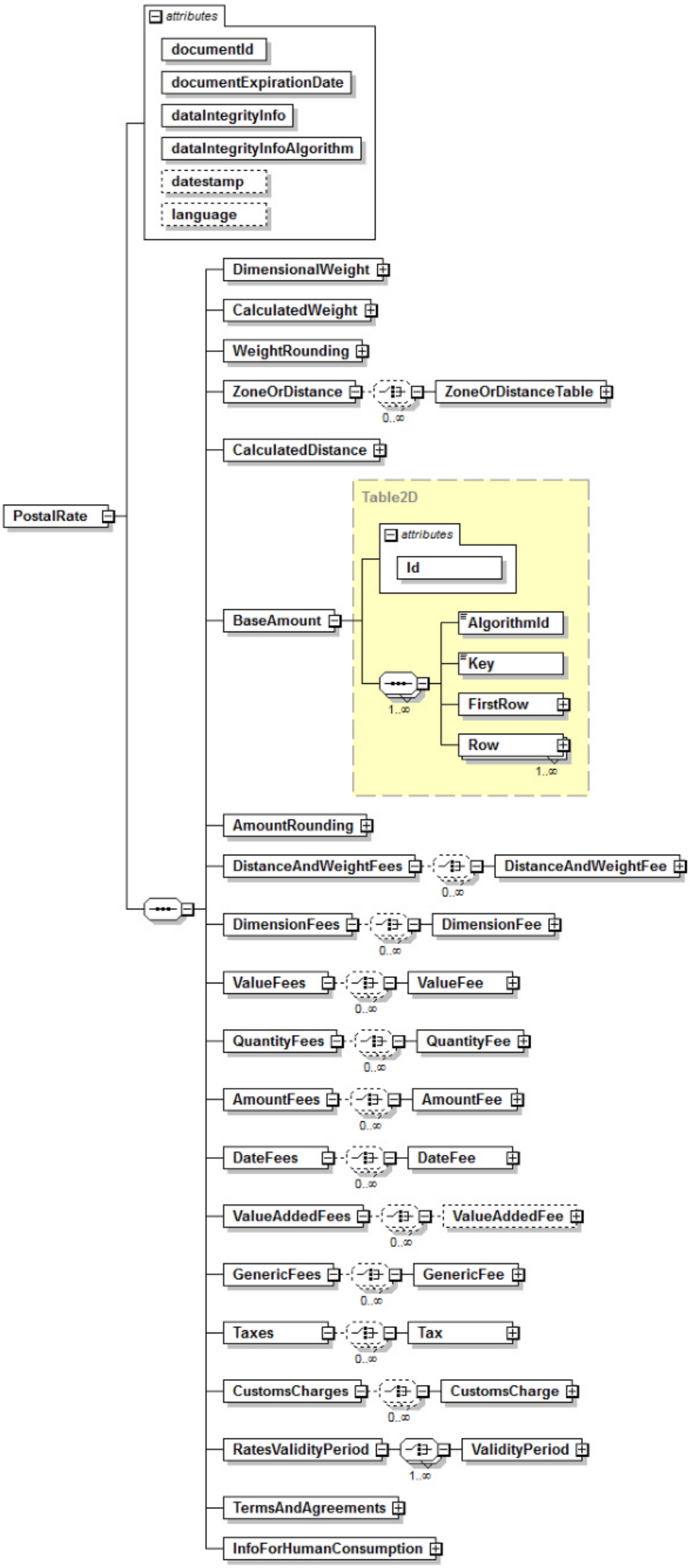


Figure 6 — Structure of the PostalRate Type

Each top level element of the EPR schema is identified by an attribute *Id* that allows postal product providers to manage versions of postal rates at different levels of granularity. This information applies to XML schema elements described in every subclause below.

6.2.2 EPR attributes

6.2.2.1 Introduction

There are several attributes of the root element of the EPR. They apply to the postal rates definition file as a whole. These are: *documentExpirationDate, documentId, dataIntegrityInfo, dataIntegrityInfoAlgorthm, datestamp, and language.*

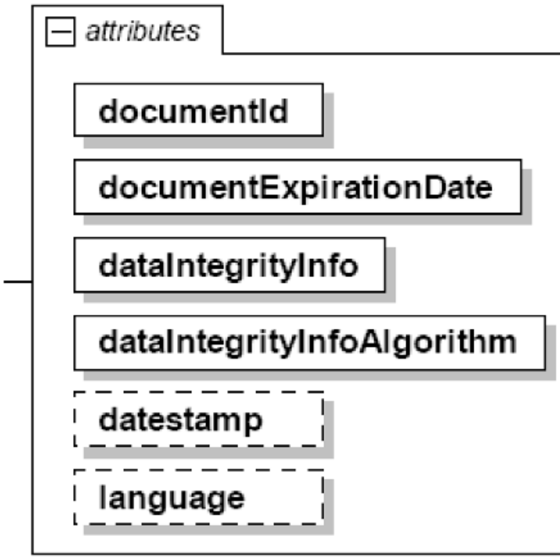


Figure 7 — EPR attributes

6.2.2.2 documentExpirationDate

Each postal rate definition file is required to have an expiration date. The attribute documentExpirationDate of the element PostalRate is of type xs:date. It contains the date after which the postal rate definition file cannot be used. The purpose of the attribute documentExpirationDate is to enable postal product providers to manage and control postal rates. The postal rate element ValidityPeriod is distinct from attribute documentExpirationDate. For example, a postal rate can be valid each month of December for indefinite period of time, while the expiration date for the corresponding postal rate document indicates when the document cannot be used and an update may be required.

6.2.2.3 documentId

Each postal rate file is required to have a unique identifier. The attribute *documentId* of the element *PostalRate* is a string that contains the unique identifier of each valid postal rate file. The scope of uniqueness for the attribute *documentId* shall be determined taking into account the possibility of future cooperation between postal organizations and other parties presently not involved in the postal sector. As a result, a globally unique identifier (GUID) is the best choice. One recommended method of generation of the GUID is in accordance with ISO/IEC 15459 (all parts).

The *documentId* attribute should always be present.

6.2.2.4 dataIntegrityInformation

In the context of information exchanges between multiple parties there is frequently a need to preserve the integrity of the information being exchanged, thus assuring recipients that the information they have received has not been corrupted during transmission process.

Data integrity information is a string that contains the value created by using a data integrity algorithm. The algorithm takes as input the entire text file of the postal rate file with the exception of its own value (the value of this attribute) which is initially set to an empty string for the purpose of this computation. The data integrity information is computed using one of several known algorithms such as CRC (cyclic redundancy codes), hash functions, message authentication codes (MAC) and digital signatures, depending on the security requirements. Specifically, if there is no danger of malicious modification of the postal rate file by an adversary inclined to damage relations between postal operators and their customers, any of the traditional data integrity checks (such as CRC or hash functions) are sufficient. If, on the other hand, the postal rate file shall be protected against an active adversary, a cryptographic data integrity and source authentication mechanism should be used. The recommended mechanisms include MACs and digital signatures and are described in detail in a number of national and international standards (see, for example, UPU S36-4, UPU S43-3, EN 14615 and the ISO/IEC 10118, ISO/IEC 9797 and ISO/IEC 14888 series). Specific choice of the algorithm for *dataIntegrityInfo* computation in this case depends on the desired complexity of the cryptographic key management system and it is outside of the scope of this document.

NOTE When the information exchange takes place over a public communication network (such as the Internet) there is also a persistent need to authenticate both the sender and the recipient of the information. Mutual authentication of sender and recipient is accomplished via well-known cryptographic algorithms and therefore it is out of the scope of this document.

The *dataIntegrityInfo* attribute is mandatory. It enables parties using postal rate file to verify the integrity of the data contained in the postal rate file. The identity of the algorithm and the values of its required parameters used to compute the dataIntegrityInfo are specified in the attribute dataIntegrityInfoAlgorithm.

6.2.2.5 dataIntegrityInfoAlgorithm

Data integrity information algorithm attribute is a string that contains the name of the algorithm used to create the value of dataIntegrityInfo and the values of all necessary parameters.

6.2.2.6 dateStamp

The *datestamp* is an attribute of type *date* that contains the date when the postal rate document was created.

The *datestamp* attribute is optional.

6.2.2.7 Language

The optional attribute *language* indicates the language of the XML element values. The type of *language* attribute is xs:language. The use of the ‘language’ attribute is left at the discretion of posts. Best Current Practices (BCP) document provides guidance regarding the specification of the language used by XML documents. It was created by IETF (Internet Engineering Task Force) and consists of RFC 4646: Tags for Identifying Languages and RFC 4647: Matching of Language Tags [2].

6.3 Dimensional weight

This subclause describes the element of the EPR schema that defines the structure of the data (supplied by the postal product provider) necessary to determine the dimensional weight. Figure 8 represents the structure of the “*DimensionalWeight*” element of the EPR schema.

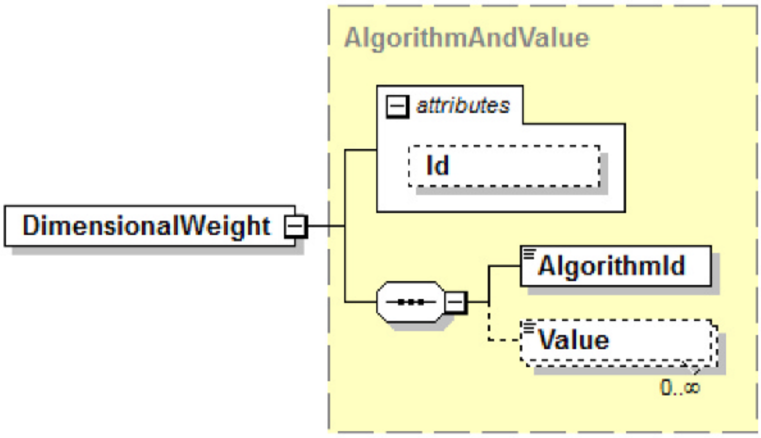


Figure 8 — Structure of DimensionalWeight element

DimensionalWeight is an XML element of AlgorithmAndValue Type. AlgorithmAndValue Type is described in 6.21.2. The computation of the dimensional weight uses (in addition to *DimensionalWeightData*) the dimensions of the mail unit (provided by the mailer) as an input to the dimensional weight algorithm shown in the following diagram.

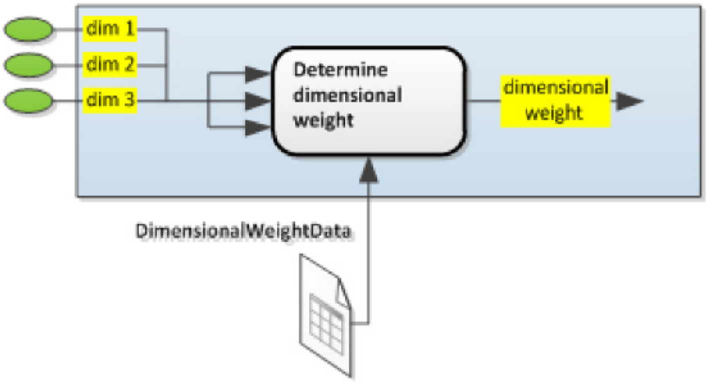


Figure 9 — Inputs and output of dimensional weight calculation algorithm

Postal services providers frequently take into account the amount of space taken by mail units. When mail units are unusually large for their weight (i.e. when they have low density) postal services providers incur a processing cost that exceeds the cost associated only with the mail unit's weight. Dimensional weight value provides a compensation mechanism that takes into account additional processing and transportation cost attributable to mail units with relatively large dimensions. Postal operators define algorithms for computing dimensional weight based on specific conditions (e.g. space in an aircraft or in railroad car available for transportation of mail units). Most common algorithms for dimensional weighing calculate the volume of the mail item, compare it to a threshold and when the threshold is exceeded the volume is multiplied by a conversion factor and rounded it to the next larger integer to arrive at the dimensional weight. The dimensional weight is then used in subsequent rate calculation steps instead of the mail item physical weight. The algorithm is identified by the value stored in the XML element *AlgorithmId* and all necessary parameters (e.g. conversion factors) are stored in the XML elements *Value* (zero or more such elements are allowed).

An example of dimensional weighing and rounding is “divide the cubic size in inches by 166 to determine dimensional weight in pounds. Increase any fraction to the next whole pound.” [5]

6.4 Calculated weight

6.4.1 General

This subclause describes the CalculatedWeight element of the EPR schema that defines the structure of the data (supplied by the postal product provider) necessary to determine the calculated weight. Figure 10 represents the structure of the "*CalculatedWeight*" element of the EPR schema.

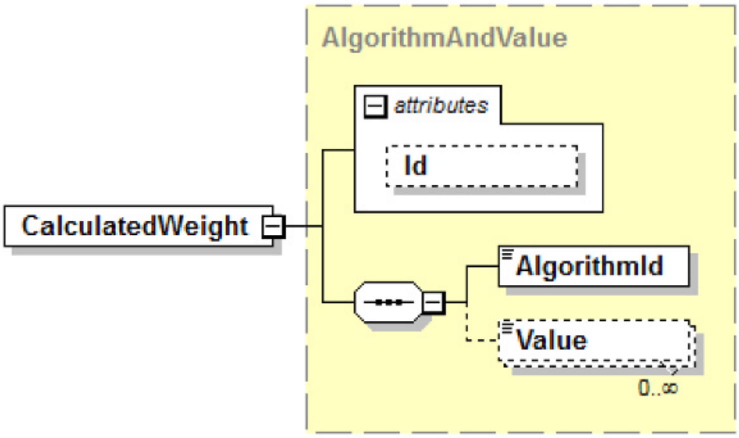


Figure 10 — Structure of CalculatedWeight element

CalculatedWeight is an XML element of AlgorithmAndValue Type. The AlgorithmAndValue Type is described in 6.21.2. The computation of the calculated weight uses (in addition to *CalculatedWeightData*) the dimensional and measured weight of the mail unit (provided by the mailer) as an input to the calculated weight algorithm shown in the following diagram.

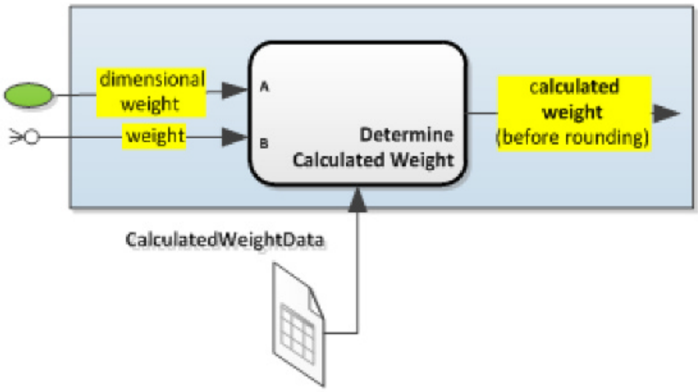


Figure 11 — Inputs and output of calculated weight calculation algorithm

Calculated weight is computed using a selection algorithm making choice between the physical (measured) weight of the mail unit and the computed dimensional weight (described in 6.3) depending on which weight is the larger. For most postal products and associated mail units the calculated weight is equal to the physical weight.

An example of the use of calculated weight is: "if the cubic size of the package in inches is 5,184 or larger, divide the cubic size by 166 to determine dimensional weight in pounds. If the cubic size in inches is less than 5,184, use the actual weight of the package." [5]

6.4.2 Weight rounding

This subclause describes the WeightRounding element of the EPR schema that defines the structure of the data (supplied by the postal product provider) necessary to determine the rounding of the weight value to be used in computing the postal rate. Figure 12 represents the structure of the "*WeightRounding*" element of the EPR schema.

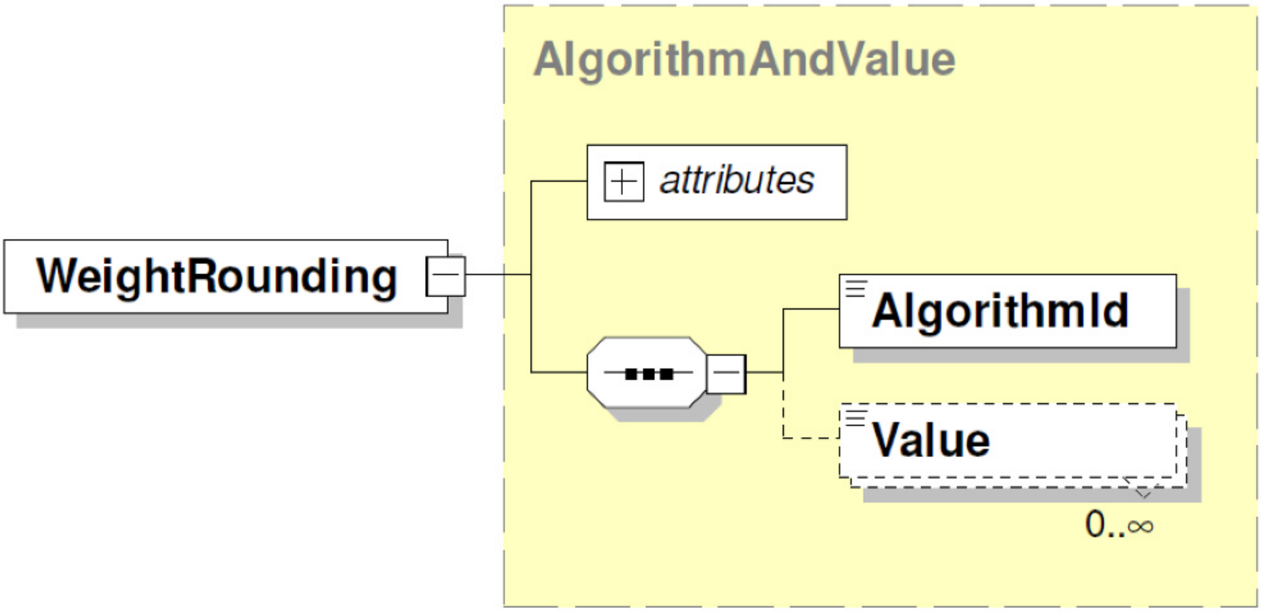


Figure 12 — Structure of WeightRounding element

WeightRounding is an XML element of AlgorithmAndValue Type. The AlgorithmAndValue Type is described in 6.21.2. The weight rounding uses (in addition to *WeightRoundingData*) the calculated weight of the mail unit (provided by the mailer) before rounding as an input to the weight rounding algorithm shown in the following diagram.

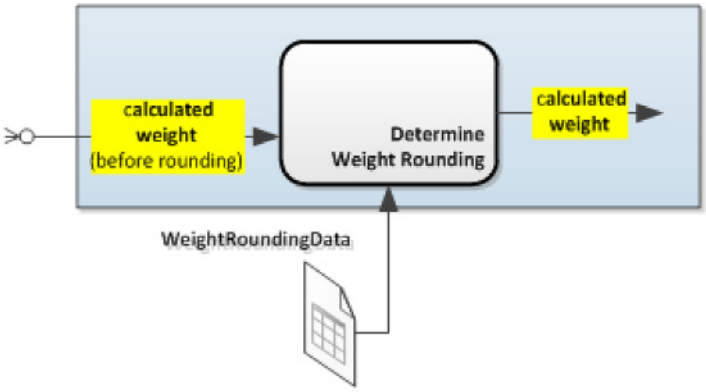


Figure 13 — Inputs and output of weight rounding calculation algorithm

Examples of rounding algorithms are "rounding up to the next integer", "rounding down to the next integer", "rounding to the nearest next integer". In weight rounding the value of "integer" is always linked to the unit of measurement (e.g. gram, kilogram, ounce, and pound).

6.5 Zone or Distance

This subclause describes the ZoneOrDistance element of the EPR schema that defines the structure of the data (supplied by the postal product provider) necessary to determine the zone (or distance) to be used in computing postal rate. Figure 14 represents the structure of the "*ZoneOrDistance*" element of the EPR schema.

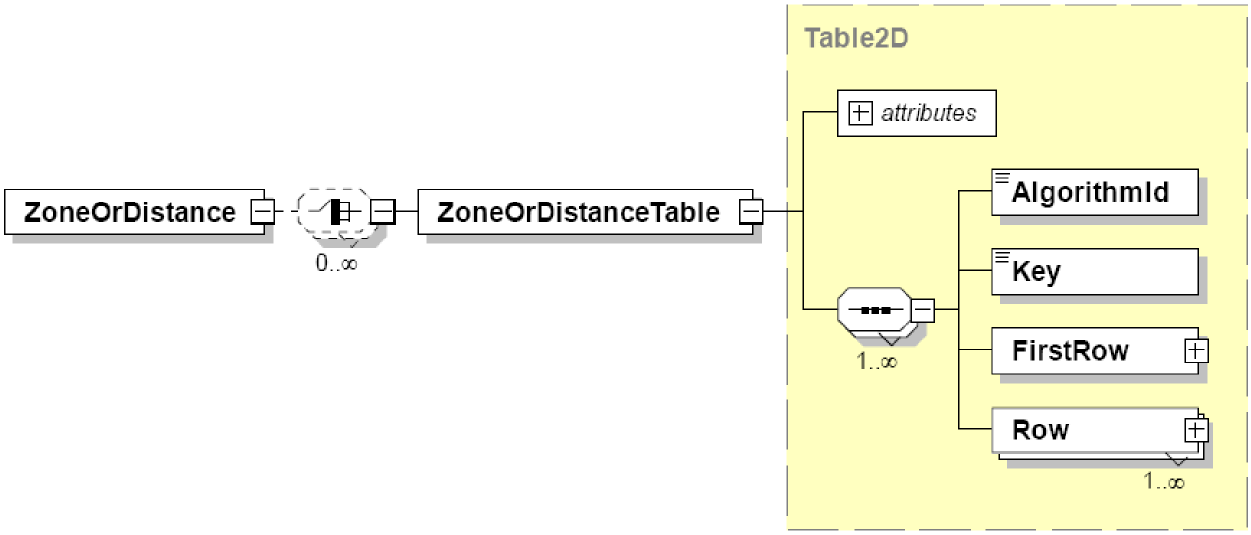


Figure 14 — Structure of ZoneOrDistance element

ZoneOrDistance is an XML element that includes zero or more XML elements *ZoneOrDistanceTable*. *ZoneOrDistanceTable* is an XML element of type Table2D which is described in 6.21.1. The computation of the zone/distance uses (in addition to *ZoneOrDistanceData*) the origin, destination of the mail unit and the product key (provided by the mailer) as an input to the zone/distance algorithm as shown in the following diagram.

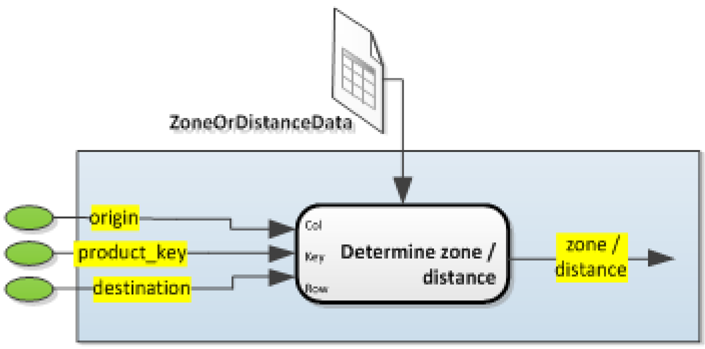


Figure 15 — Inputs and output of zone/distance determination algorithm

The cost of transporting the mail unit between the origin and destination locations represents a portion of the overall postal product cost. Postal operators and carriers use zone-based postal rates in order to reflect the cost of transportation in product pricing. The data representing zone codes is normally organized as a collection of tables. As shown in the Figure 16 structure, each table for rating purposes is searched using a specific algorithm identified by the value of the *AlgorithmId* element. The *Key* element contains an identifier for the product (*product_key*).

The *product_key* may depend on the status of the sender (e.g. non-profit organization) or other attributes.

EXAMPLE 1: The figure below shows an UPS table that is used to determine the zone code from the origin locations (5 digit ZIP codes from 786 to 01 to 787–99), product (Ground, 3 day select, etc.) and destination locations identified by 3 digit ZIP codes (004–005, 006–007, etc.). [3]

```
Untitled - Notepad
File  Edit  Format  View  Help
For shipments originating in ZIP Codes 786-01 to 787-99.  To determine zone
take the first three digits of the receiver's ZIP Code and refer to the chart below:

ZONES
Dest. ZIP   Ground   3 Day Select    2nd Day Air   2nd Day Air A.M. Next Day Air Saver    Next Day Air

004-005       7         307             207            247              137                 107
006-007       -          -              225             -                -                  125
008           -          -              [1]             -                -                  [1]
009           -          -              225             -                -                  125
010-043       7         307             207            247              137                 107
044           8         308             208            248              138                 108
045           7         307             207            247              137                 107
046-048       8         308             208            248              138                 108
049-089       7         307             207            247              137                 107
```

Figure 16 — Example of determination of zone code from the product_key, origin and destination locations (UPS)

EXAMPLE 2: The figure below shows a fragment of the USPS zone chart for the origin identified by 3 digit ZIP code “064” and multiple destinations identified by various ranges of 3 digit ZIP codes from “005” to “741”. [4]

Zip Code	Zone	Zip Code	Zone	Zip Code	Zone	Zip Code	Zone
005	1	270---286	4	500---503	6	735---736	7
006---009	7	287---294	5	504	5	737	6
010---034	2*	295	4	505	6	738---739	7
035---037	3*	296	5	506---507	5	740---741	6

Figure 17 — Example of determination of zone code from the origin and destination locations (USPS)

6.6 Calculated distance

This subclause describes the *CalculatedDistance* element of the EPR schema that defines the data (supplied by the postal product provider) necessary to determine the calculated distance to be used in computing postal rate. Figure 18 represents the structure of the XML element “*CalculatedDistance*” of the EPR schema.

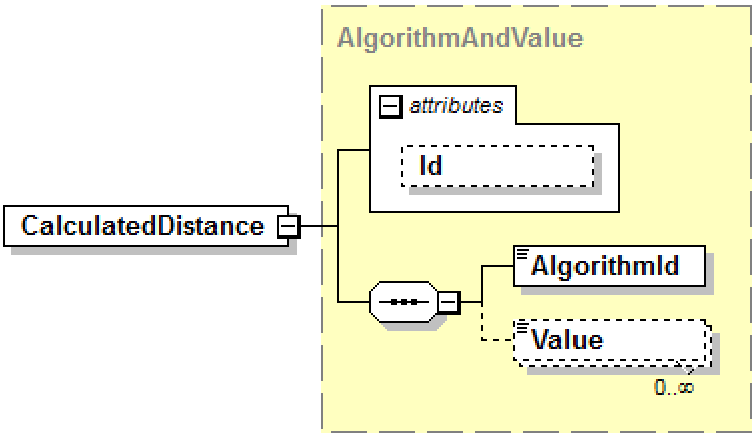


Figure 18 — Structure of CalculatedDistance element

CalculatedDistance is a XML element of AlgorithmAndValue Type. The AlgorithmAndValue Type is described in 6.21.2. The computation of the calculated distance uses (in addition to *CalculatedDistanceData*) the zone/distance, destination and geographical distance (between origin and destination) as an input to the zone/distance algorithm as shown in the following diagram.

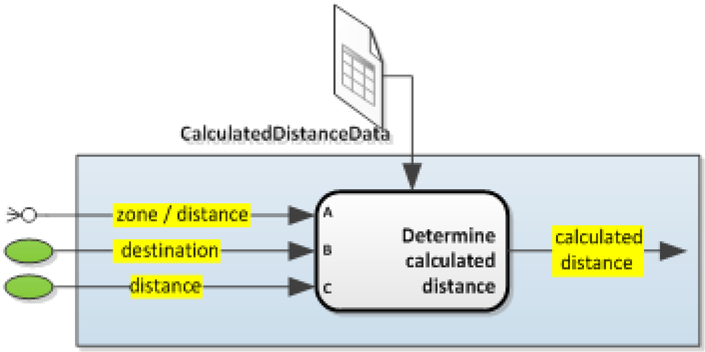


Figure 19 — Inputs and output of zone/distance selection algorithm

When the postal rate is a function of the distance between the origin and the destination, one of the following three representations of distance is used: codified distance (zone), only destination zone code or a geographical distance. The algorithm shown above is a selection of one of these three inputs as it is specified in the product description. The chosen input is determined either by the unique value of *AlgorithmId* or a by combination of *AlgorithmId* and *Value* (see 6.21.2).

6.7 Base amount

6.7.1 General

This subclause describes the BaseAmount element of the EPR schema that defines the structure of the data supplied by the postal product provider necessary to determine the base amount. Clause 5 provides more detailed description of the concept of base amount and its relation to the rate of postal product for a given the mail unit. Figure 20 represents the structure of the XML element "*BaseAmount*" of the EPR schema.

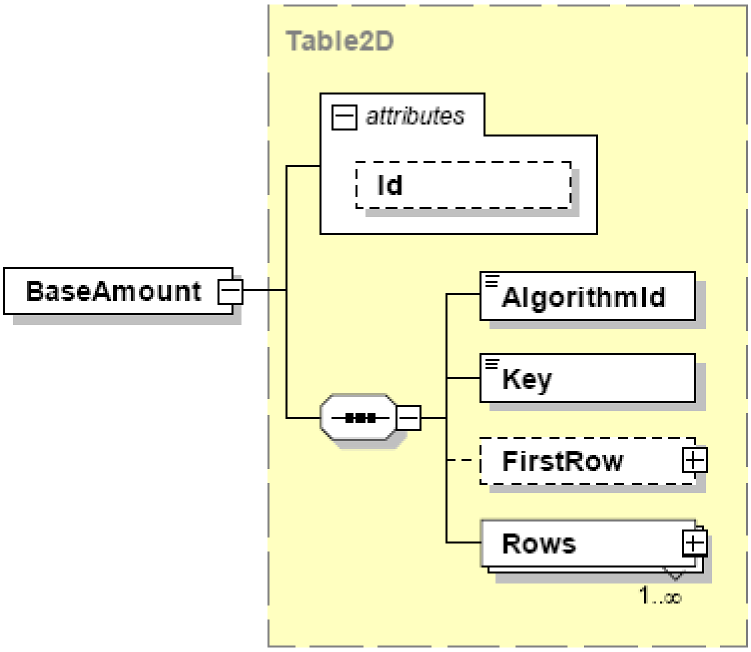


Figure 20 — Structure of BaseAmount element

The BaseAmount element of the EPR schema is of type Table2D (see description of Table2D structure in 6.21.1). Each two-dimensional table contains the data organized in rows and columns. The element Table2D contains one or more two-dimensional tables each corresponding to a product defined by the postal product provider and uniquely identified by the product key.

The computation of the base amount uses (in addition to *BaseAmountData*) the calculated weight and distance and the product key as an input to the base amount algorithm as shown in the following diagram.

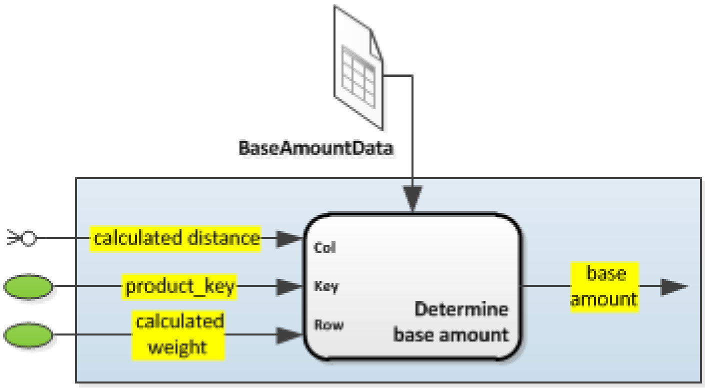


Figure 21 — Inputs and output of base amount calculation algorithm

The algorithm inputs correspond to the data in the 2D Table XML elements as shown in the following table:

Algorithm input value	Algorithm input name	Table2D organization
Calculated distance	col	First Row
Product key	key	Key
Calculated weight	row	First column

The BaseAmount element is designed to represent a broad variety of postal rates, from simple ones (e.g. rates for Royal Mail Large letter first class product) to complex ones (e.g. rates for USPS workshare products).

EXAMPLE The base amount for Royal Mail first class large letter is determined by a rate for one of the two products "first class large letter stamps" or "first class large letter franking" each identified by its own product key. The rates for each product are shown in the table below.

First Class		
	Stamps or SmartStamp	Franking or Account
Weight	Price	Price
Letter		
0–100g	£0.41	£0.36
Large Letter		
0–100g	£0.66	£0.50
101–250g	£0.96	£0.72
251–500g	£1.32	£1.04
501–750g	£1.87	£1.51

Figure 22 — Example of base amount calculation using product key input (Royal Mail)

NOTE The difference between "Stamp or Smart stamp" and "Franking or Account" represents a discount provided to Franking and Account users and it is reflected in the choice of the product key when base amount is computed. This difference can also be reflected in the product key input to multiple fees calculations as described below. This indicates (as was already mentioned in Clause 5) that the distinction between base amount and fees is somewhat arbitrary and postal operators in determining prices for their products can select a convenient representation mechanism for fees and discounts. This document is designed to accommodate various such mechanisms for marketing and convenience purposes.

An example of more complex postal rates is USPS Commercial-Standard mail shown in the following table.

Commercial—Standard Mail

Standard Mail Regular

COMMERCIAL LETTERS—CARRIER ROUTE & AUTOMATION

	Entry Discount	Carrier Route[1,2]			Automation[2]			
		Saturation	High Density	Basic	5-Digit	3-Digit	AADC	Mixed AADC
Letters weighing 3.3 oz. or less per piece price	None	$0.182	$0.193	$0.262	$0.233	$0.251	$0.253	$0.270
	DBMC	0.148	0.159	0.228	0.199	0.217	0.219	0.236
	DSCF	0.139	0.150	0.219	0.190	0.208	0.210	---
	DDU	---	---	---	---	---	---	---
more than 3.3 oz.[3] per pound price	None	0.609	0.649	0.696	0.725	0.725	0.725	0.725
	DBMC	0.446	0.486	0.533	0.562	0.562	0.562	0.562
	DSCF	0.401	0.441	0.488	0.517	0.517	0.517	---
	DDU	---	---	---	---	---	---	---
+ per piece price		+ 0.056[4]	+ 0.059[4]	+ 0.118[4]	+ 0.083[4]	+ 0.101[4]	+ 0.103[4]	+ 0.120[4]

Figure 23 — Example of base amount data table (USPS)

The USPS rates for Commercial-Standard is expressed as a collection of tables for single postal products such as “Entry discount DBMC with 3-digit automation presort”, “Entry discount DBMC with 5-digit automation presort”, etc. DBMC, DSCF, DDU are abbreviated classifications for different types of mail induction facilities. Thus, the base amount for Commercial-Standard mail in US depends on the level of presort and the type of postal processing centre where the mail is inducted representing overall level of work sharing between mailers and USPS.

All products are uniquely identified by their product keys and the base amount for each product is stored in a 2D table.

Note that the example in Figure 23 shows a document created for human consumption. It contains a formula for the calculation of base amount. When rates are communicated from a computer to a computer they are expressed as tables, not formulas. An example of such table is in Annex B.

6.7.2 Amount rounding

This subclause describes the *AmountRounding* element of the EPR schema that defines the structure of the data supplied by the postal product provider necessary to determine the amount to be used in computing postal rate. Figure 24 represents the structure of the XML element “*AmountRounding*” of the EPR schema.

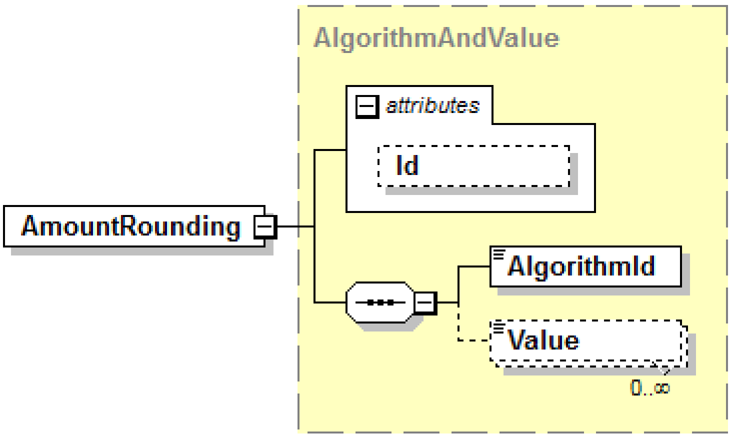


Figure 24 — Structure of AmountRounding element

AmountRounding is an XML element of AlgorithmAndValue Type. The AlgorithmAndValue Type is described in 6.21.2. The amount rounding uses (in addition to *AmountRoundingData*) the base amount before rounding as an input to the amount rounding algorithm shown in the following diagram.

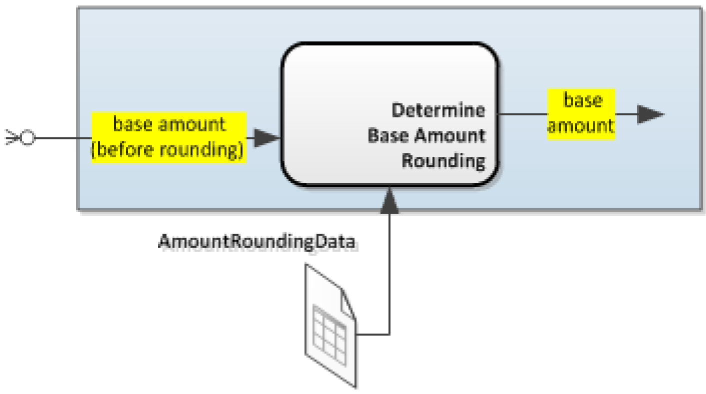


Figure 25 — Inputs and output of base amount rounding calculation algorithm

Similar to the weight rounding described in 6.4.2 “AmountRounding” element can express all practically encountered choices for rounding value, namely “rounding up to the next integer”, “rounding down to the next integer”, “rounding to the nearest next integer” and rounding precision. Each integer value in the amount rounding is linked to the selected unit of measurement, e.g. monetary unit (Euro, Dollar and Cent).

6.8 Distance and weight based fees

This subclause describes the *DistanceAndWeightFees* element of the EPR schema that defines the structure of the data supplied by the postal product provider necessary to determine the fees that are dependent of distance and/or weight.

Figure 26 represents the structure of the XML element “*DistanceAndWeightFees*” of the EPR schema.

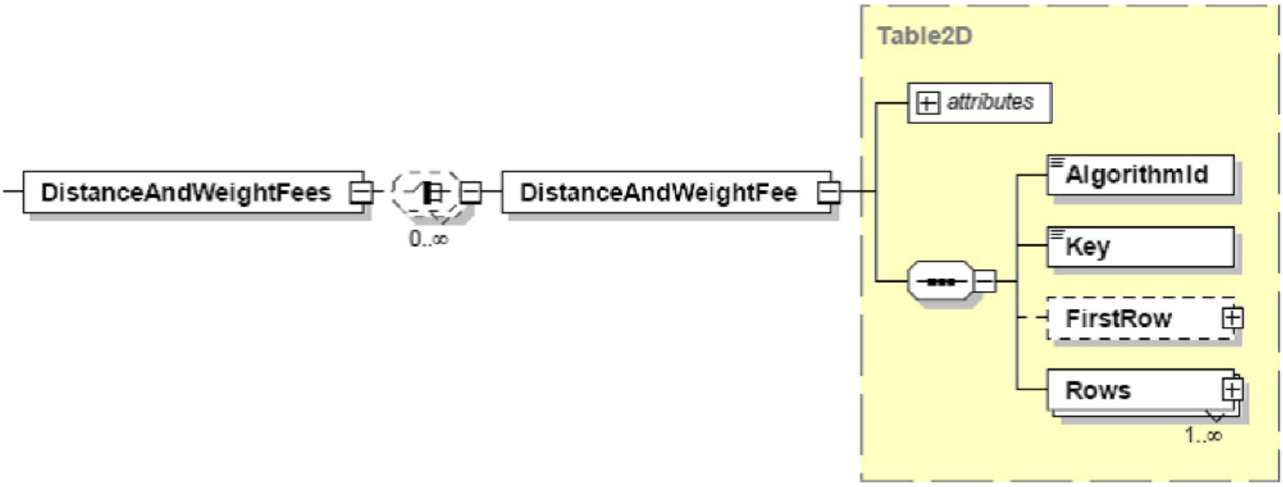


Figure 26 — Structure of DistanceAndWeightFees element

DistanceAndWeightFees is an XML element that includes zero or more XML elements *DistanceAndWeightFee*. *DistanceAndWeightFee* is an XML element of type Table2D which is described in 6.21.1.

The distance and weight fees are computed using the calculated distance and/or the calculated weight as inputs to the algorithm shown in the following diagram.

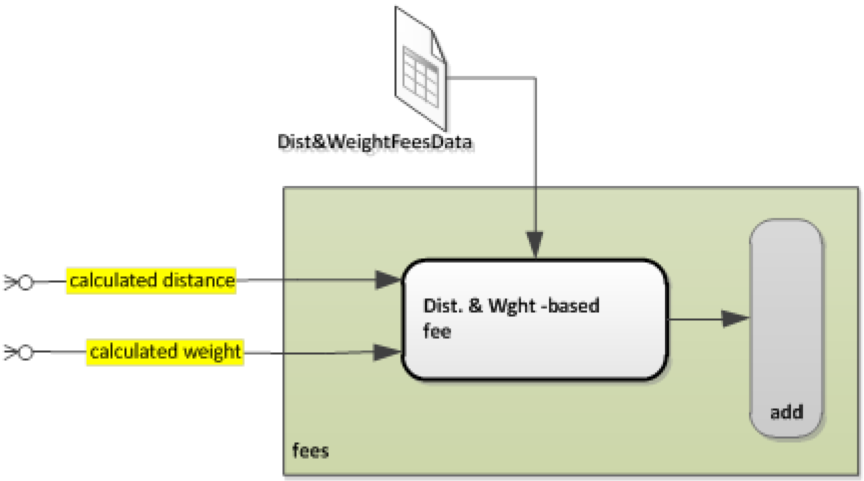


Figure 27 — Inputs and output of distance and weight fees calculation algorithm

The resulting fee is an additive component of the amount of (total) fees that are computed by adding (or subtracting) all fees associated with the mail unit and the postal product selected by the mailer.

6.9 Dimension-based fees

Dimension-based fees are the data supplied by the postal product provider that is necessary to determine the fees that are dependent of mail unit dimensions. Figure 28 represents the structure of the XML element "*DimensionFees*" of the EPR schema.

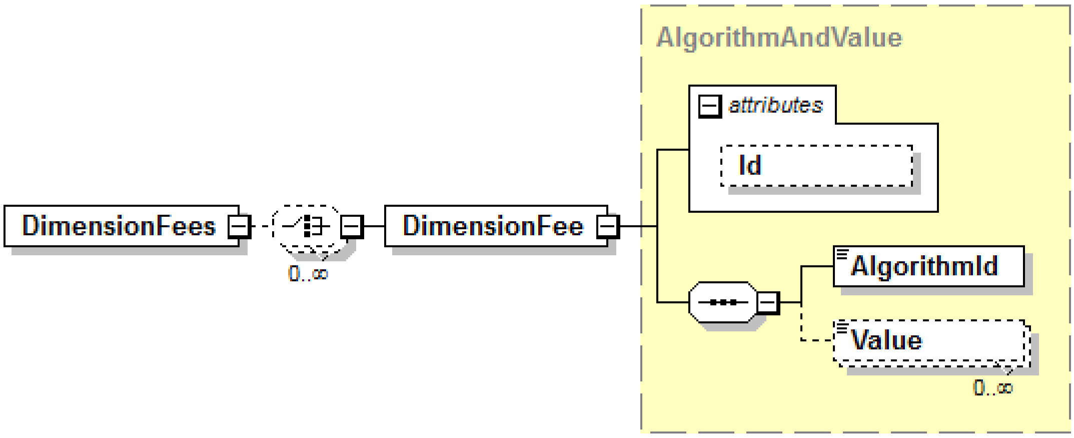


Figure 28 — Structure of DimensionFees element

DimensionsFees element includes zero or more XML elements *DimensionsFee*.

DimensionsFee is an XML element of type "AlgorithmAndValue" which is described in 6.21.2.

The dimension-based fees are computed using dimensions of the mail unit and "DimensionsFeesData" (supplied by the postal product provider) as inputs to the dimension-based fees algorithm shown in the following diagram.

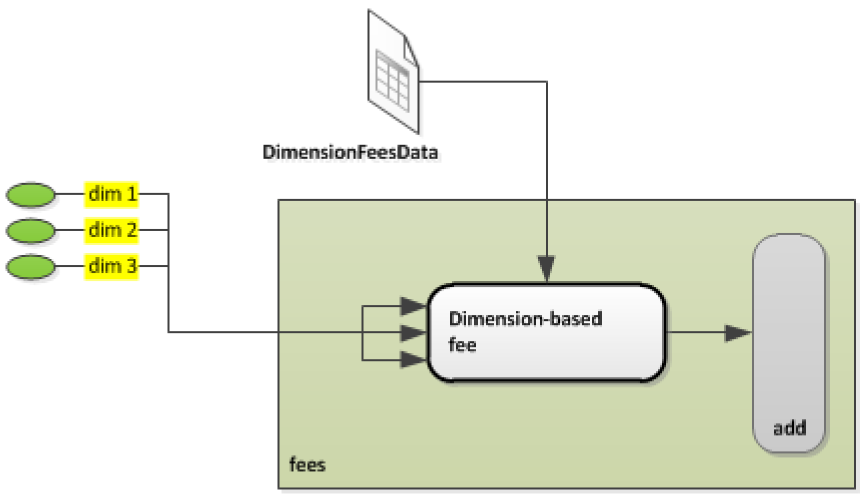


Figure 29 — Inputs and output of dimension-based fees calculation algorithm

The resulting fee is an additive component of the amount of (total) fees that are computed by adding (or subtracting) all fees associated with the mail unit and the postal product selected by the mailer.

6.10 Value-based fees

This subclause describes Value-based fees data element that is supplied by the postal product provider. This element is necessary to determine the fees that are dependent of mail unit value as declared by the mailer. Figure 30 represents the structure of the "*ValueFees*" element of the EPR schema.

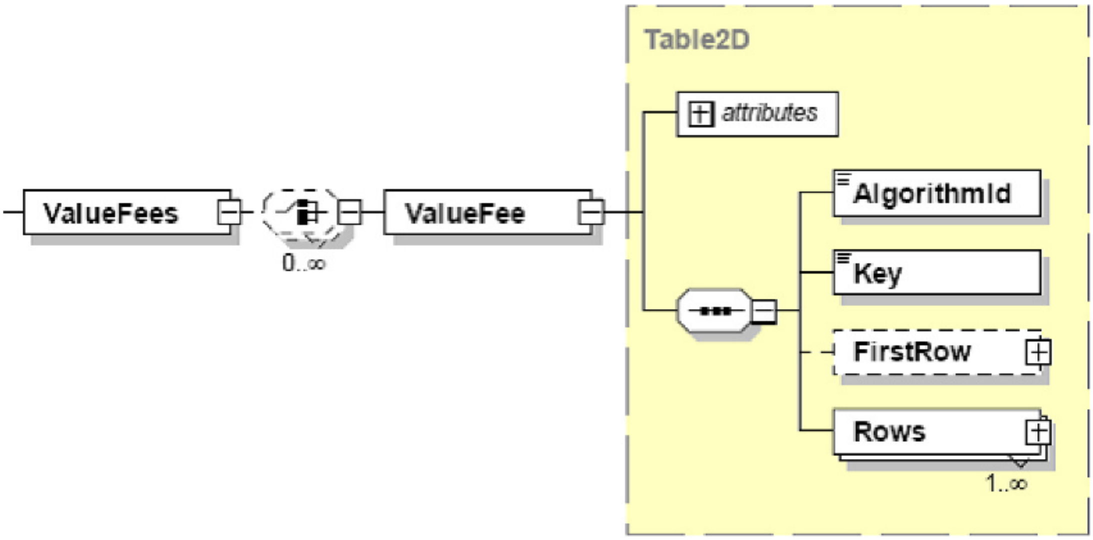


Figure 30 — Structure of ValueFees element

ValueFees is an XML element that includes zero or more XML elements *ValueFee*.

ValueFee is an XML element of type Table2D which is described in 6.21.1.

The determination of the value-based fee amount requires (in addition to ValueFees data) the mail unit value provided by the mailer as shown in the following diagram.

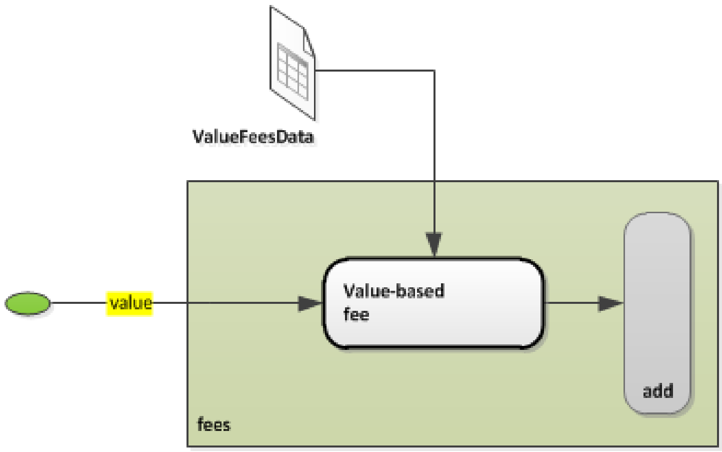


Figure 31 — Inputs and output of value-based fee calculation algorithm

The resulting fee is an additive component of the amount of (total) fees.

6.11 Quantity-based fees

Quantity-based fees represents data supplied by the postal product provider that is necessary to determine the fees that are dependent of the number of mail units (associated with a given postal product) that are submitted by the mailer in the act of single or multiple inductions. Figure 32 represents the structure of the "*QuantityFees*" element of the EPR schema.

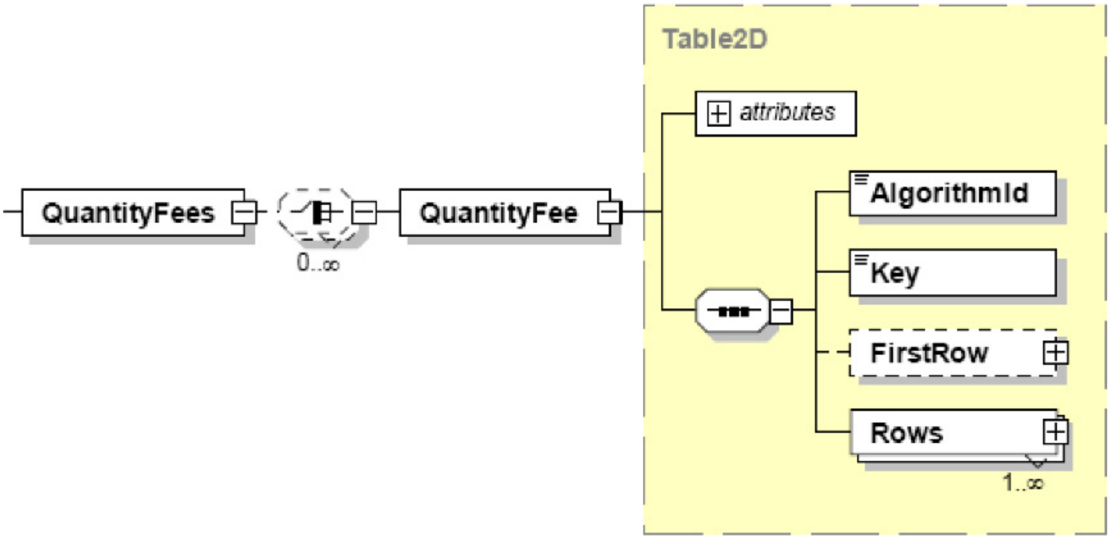


Figure 32 — Structure of QuantityFees element

QuantityFees is an XML element that includes zero or more XML elements *QuantityFee*.

QuantityFee is an XML element of type Table2D which is described in 6.21.1.

The computation of the quantity-based fee uses (in addition to *"QuantityFees"* data) the number of mail units (provided by the mailer) as an input to the quantity fees algorithm shown in the following diagram.

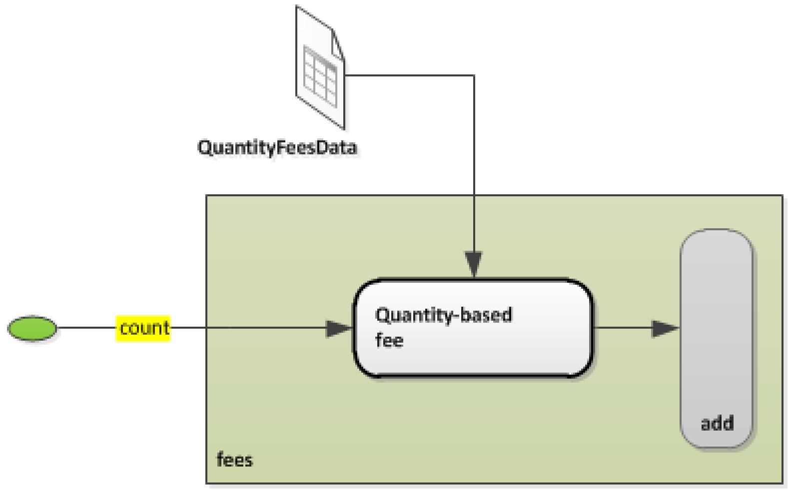


Figure 33 — Inputs and output of quantity-based fee calculation algorithm

The Quantity-based fee is an additive component of the amount of (total) fees. Quantity based fee can have a positive or negative value depending on the definition of the postal product.

6.12 Amount-based fees

Amount-based fees represents data supplied by the postal product provider that is necessary to determine the fees that are dependent on the base amount. Figure 34 represents the structure of the "*AmountFees*" element of the EPR schema.

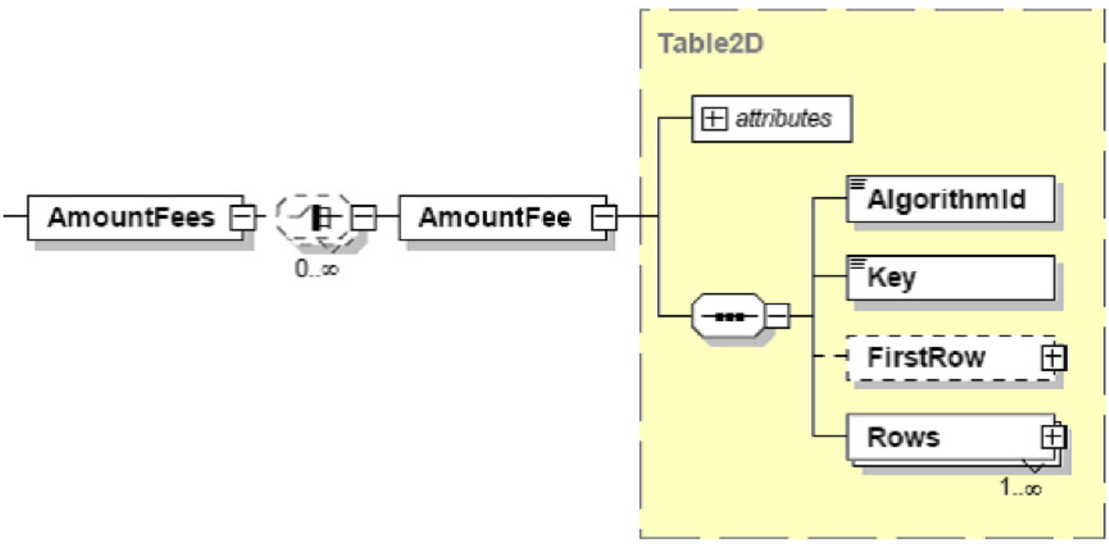


Figure 34 — Structure of AmountFees element

AmountFees is an XML element that includes zero or more XML elements *AmountFee*.

AmountFee is an XML element of type Table2D which is described in 6.21.1

The computation of the amount-based fee uses (in addition to AmountFees data) the value of the base amount as an input as shown in the following diagram.

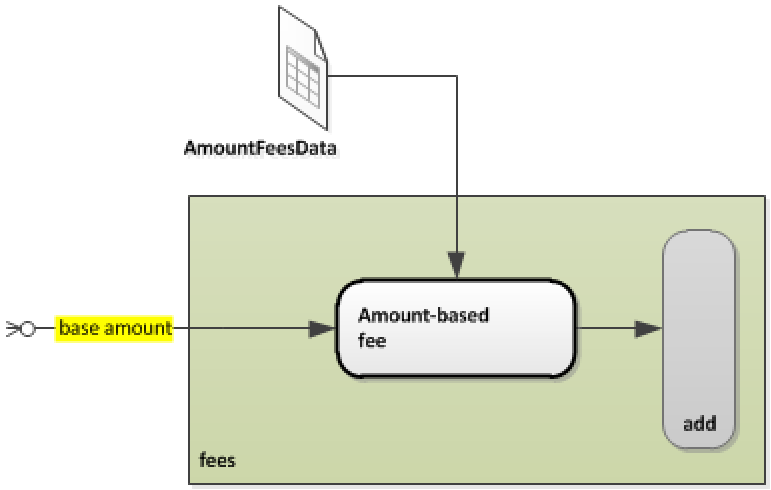


Figure 35 — Inputs and output of amount-based fee calculation algorithm

The amount-based fee is an additive component of the amount of (total) fees.

6.13 Date-based fees

The date-based fees represents the data supplied by the postal product provider that is necessary to determine the fees that are dependent on date and time of induction of the mail unit (or units). These fees enable postal product providers to achieve cost efficiency in mail processing and delivery by incentivizing mailers to deposit mail during time periods desired by postal operators and carriers. Figure 36 represents the structure of the "*DateFees*" element of the EPR schema.

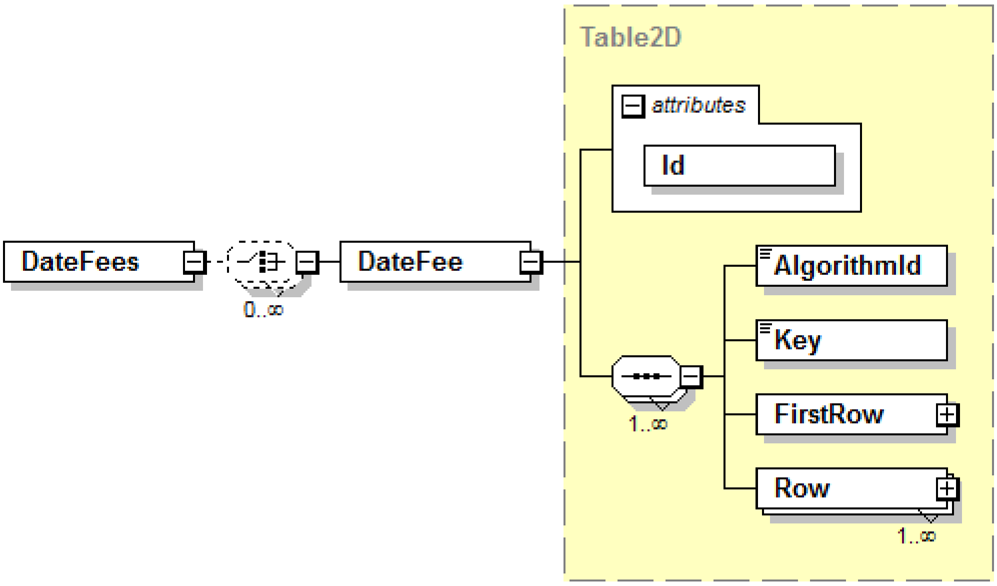


Figure 36 — Structure of DateFees element

DateFees is an XML element that includes zero or more XML elements *DateFee*.

DateFee is an XML element of type Table2D which is described in 6.21.1.

The computation of the date/time-based fee amount uses (in addition to DateFees data) the date/time data provided by the mailer as an input to the Date-based fees algorithm as shown in the following figure.

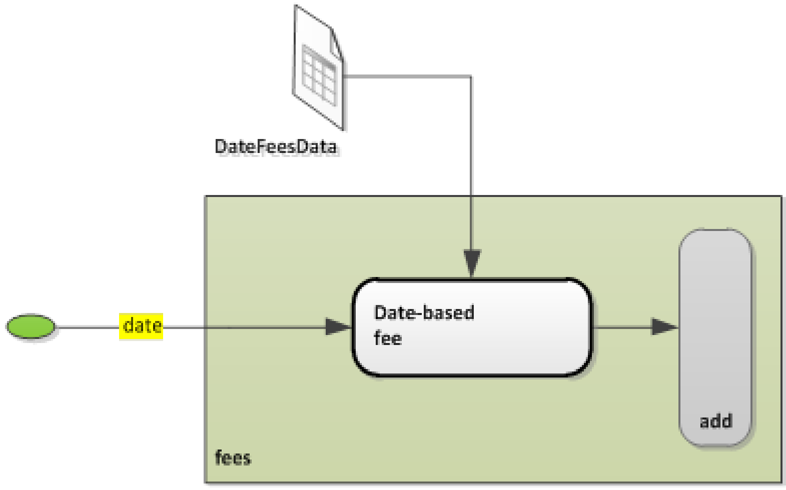


Figure 37 — Inputs and output of date/time-based fee calculation algorithm

The Date-based fee is an additive component of the amount of (total) fees.

6.14 Value added fees

Value added fees represent data supplied by the postal product provider that is necessary to determine value-added fees. Figure 38 represents the structure of the "*ValueAddedFees*" element of the EPR schema.

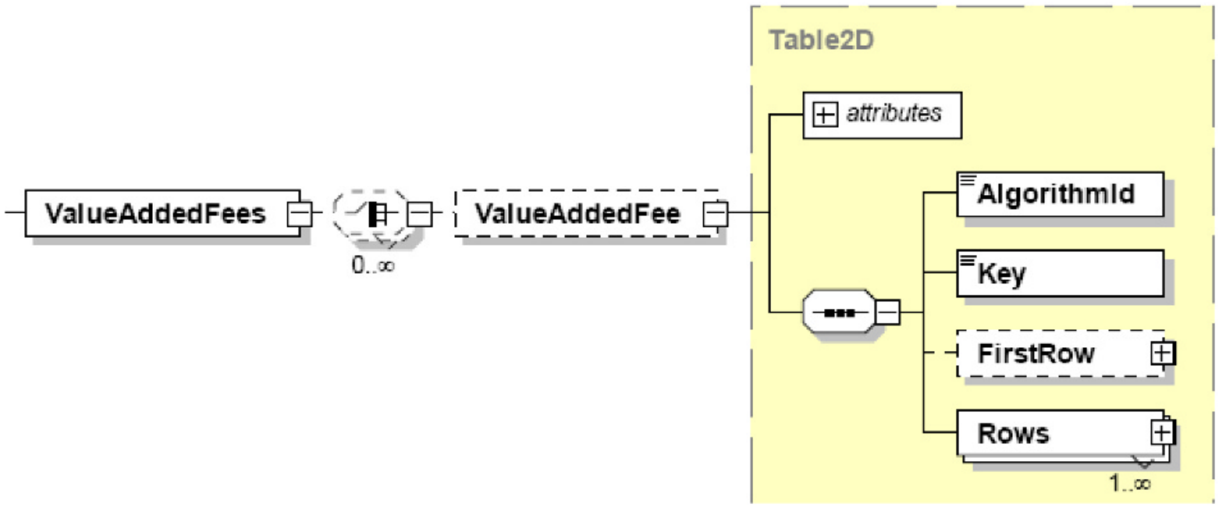


Figure 38 — Structure of ValueAddedFees element

ValueAddedFees is an XML element that includes zero or more XML elements *ValueAddedFee*.

ValueAddedFee is an XML element of type Table2D which is described in 6.21.1.

The computation of the value-added fee amount uses (in addition to ValueAddedFees data) the product selected by the mailer (and represented by the *product_key* information) as an input data to Value Added fee algorithm shown in the following figure.

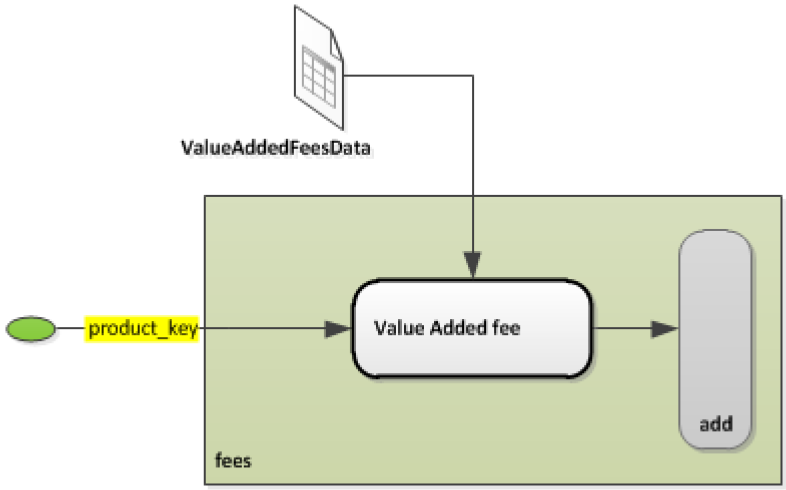


Figure 39 — Inputs and output of value-added fee calculation algorithm

Value Added fee is an additive component of the amount of (total) fees.

6.15 Generic fees

Generic fees represents data supplied by the postal product provider that is necessary to determine generic fees.

The generic fees element is a placeholder for any type of fees (positive or negative) that the postal product provider would wish to impose. The generic fees may be introduced when the postal product provider wants or is required to distinguish between different classes of mailers and recipients. Figure 40 represents the structure of the "*GenericFees*" element of the EPR schema.

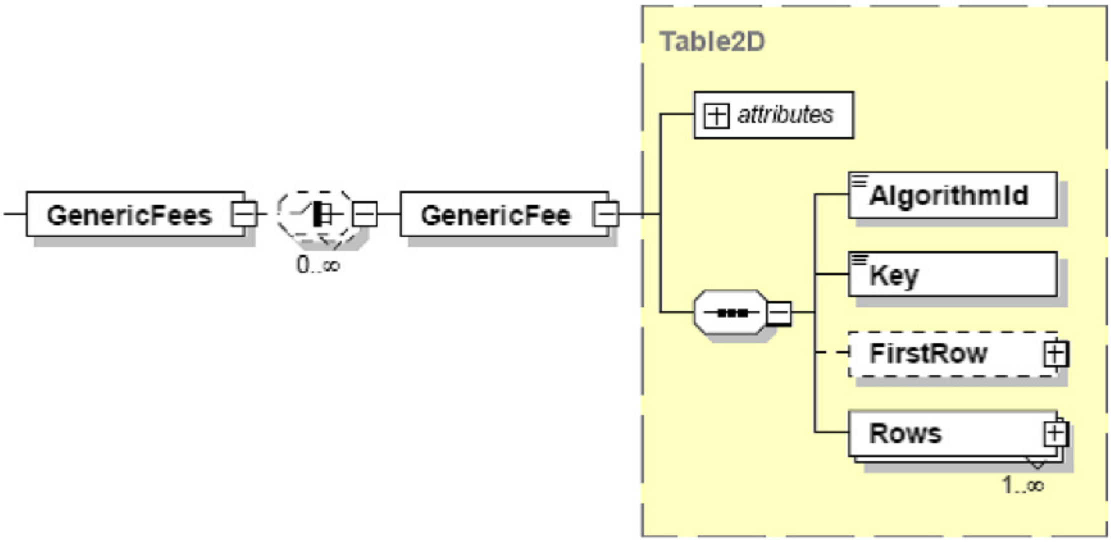


Figure 40 — Structure of GenericFees element

GenericFees is an XML element that includes zero or more XML elements *GenericFee*.

GenericFee is an XML element of type Table2D which is described in 6.21.1.

The computation of the generic fee amount uses (in addition to GenericFees data) mailer and recipients attributes provided by the mailer as an input to the Generic fee algorithm as shown in the following figure.

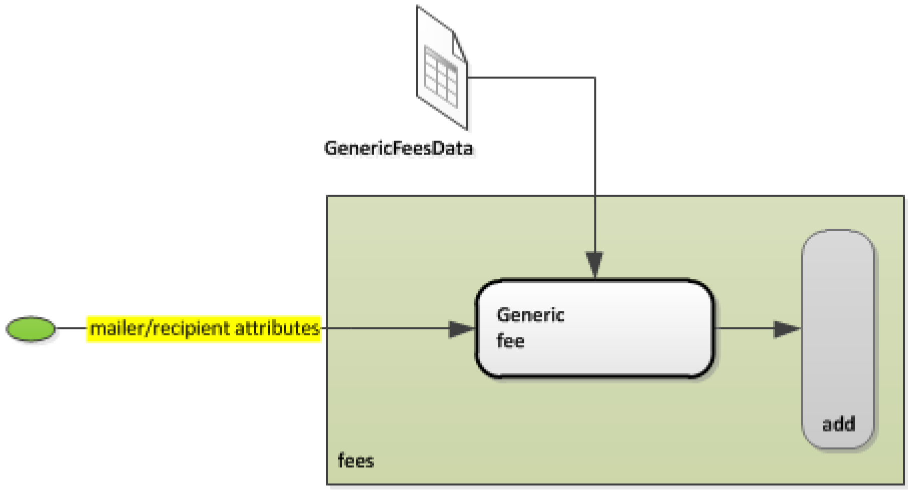


Figure 41 — Inputs and output of generic fee calculation algorithm

The Generic fee is an additive component of the amount of (total) fees.

6.16 Taxes

Taxes represent data supplied by the postal product provider that is necessary to determine taxes for the postal product. Figure 42 represents the structure of the element "*Taxes*" of the EPR schema.

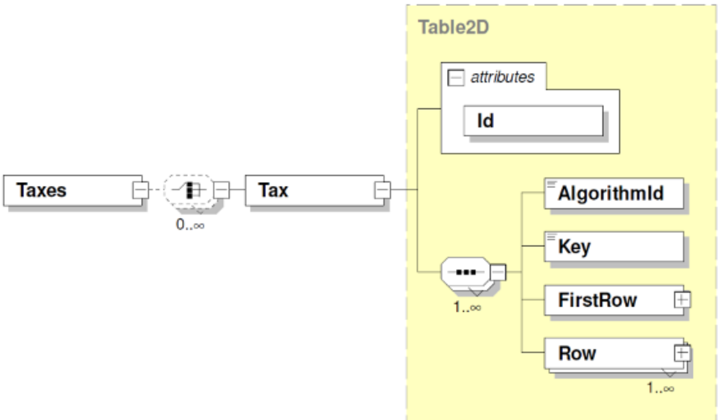


Figure 42 — Structure of Taxes element

Taxes is an XML element that includes zero or more XML elements *Tax*.

Tax is an XML element of type Table2D which is described in 6.21.1.

The computation of the tax amount uses (in addition to Taxes data supplied by the postal product provider) a number of additional parameters as inputs to the Taxes algorithm shown in the following figure. These parameters include the product selected by the mailer (*product key*), the mailer category, the base amount and fees amount for the given mail unit. Other parameters can also be included to provide for flexibility and extensibility of the Taxes element in case when tax regulations are change by authorities.

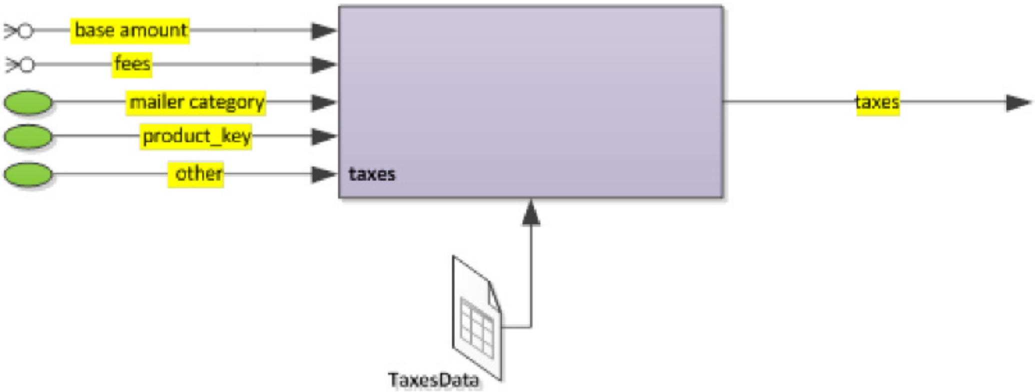


Figure 43 — Inputs and output of tax calculation algorithm

Taxes are added to other charges to arrive at the overall postal product rate.

6.17 Customs charges

Customs charges represent data supplied by the postal product provider that is necessary to determine customs charges. Figure 44 represents the structure of the element “*CustomCharges*” of the EPR schema.

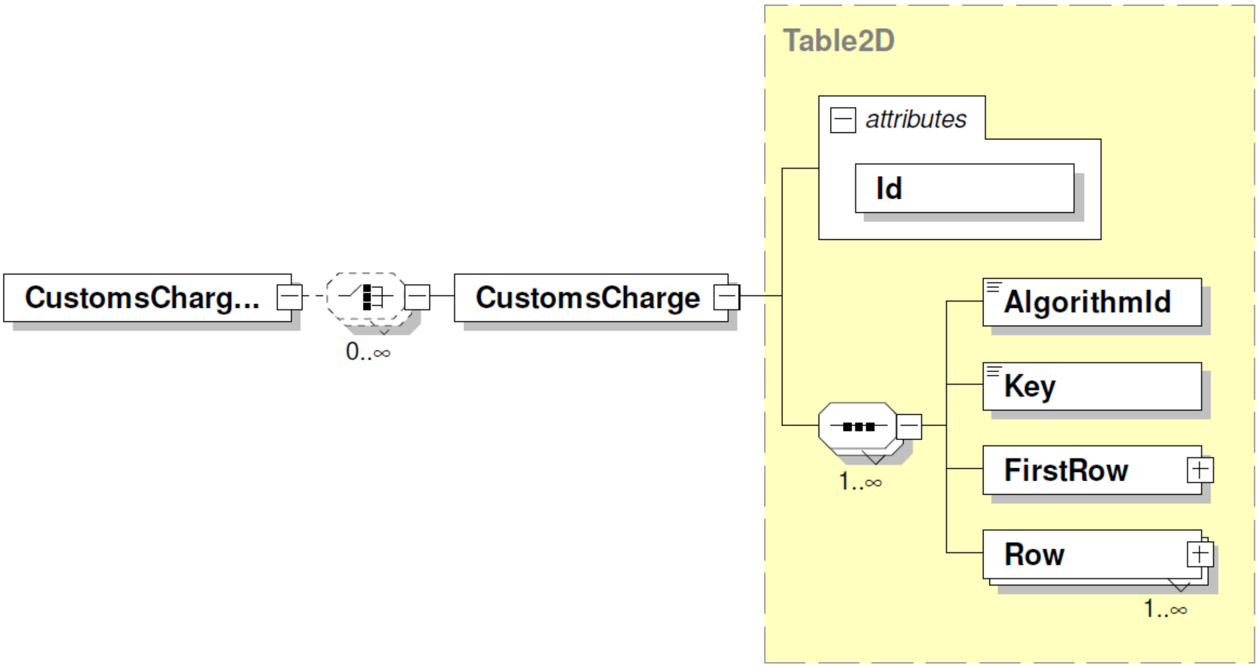


Figure 44 —Structure of CustomCharges element

CustomsCharges is an XML element that includes zero or more XML elements *CustomsCharge*.

CustomsCharge is an XML element of type Table2D which is described in 6.21.1.

Computation of the customs charges uses (in addition to CustomsCharges data) the origin, destination, content category and the declared value for the given mail unit as an input to the Custom Charges algorithm shown in the following figure.

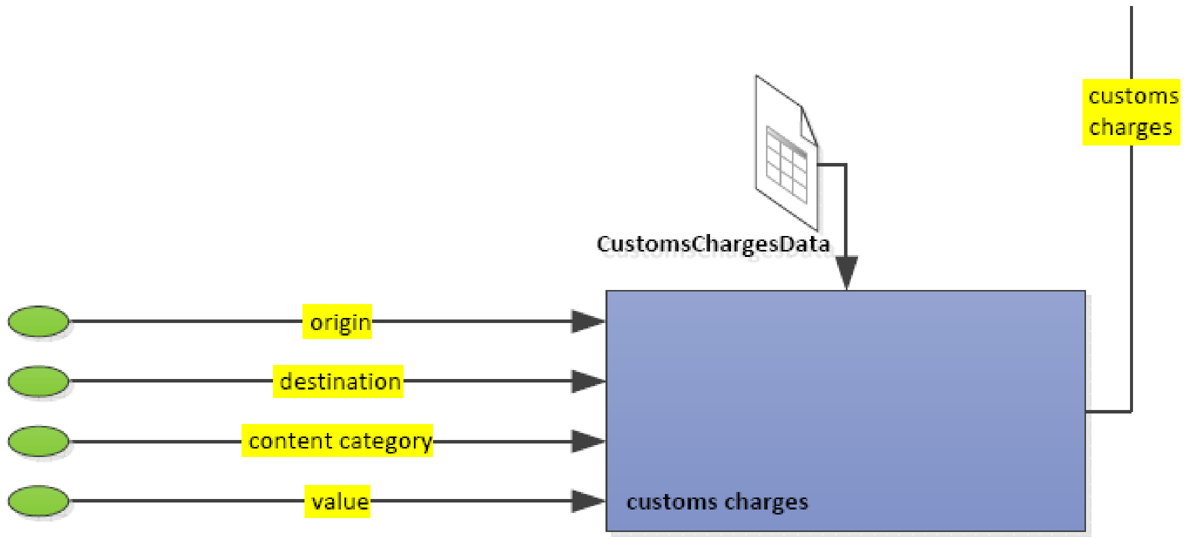


Figure 45 — Inputs and output of customs charges calculation algorithm

Customs charges are added to other charges to arrive at the overall postal product rate.

6.18 EPR validity

EPR validity represents data supplied by the postal product provider that is necessary to determine the time interval(s) when the postal rates are valid. The time interval is represented as a continuous period of time with defined beginning (StartDate) and ending (EndDate). The rate may be defined to be valid during one or more time intervals. Figure 46 represents the structure of the element "*RatesValidityPeriod*" of the EPR schema.

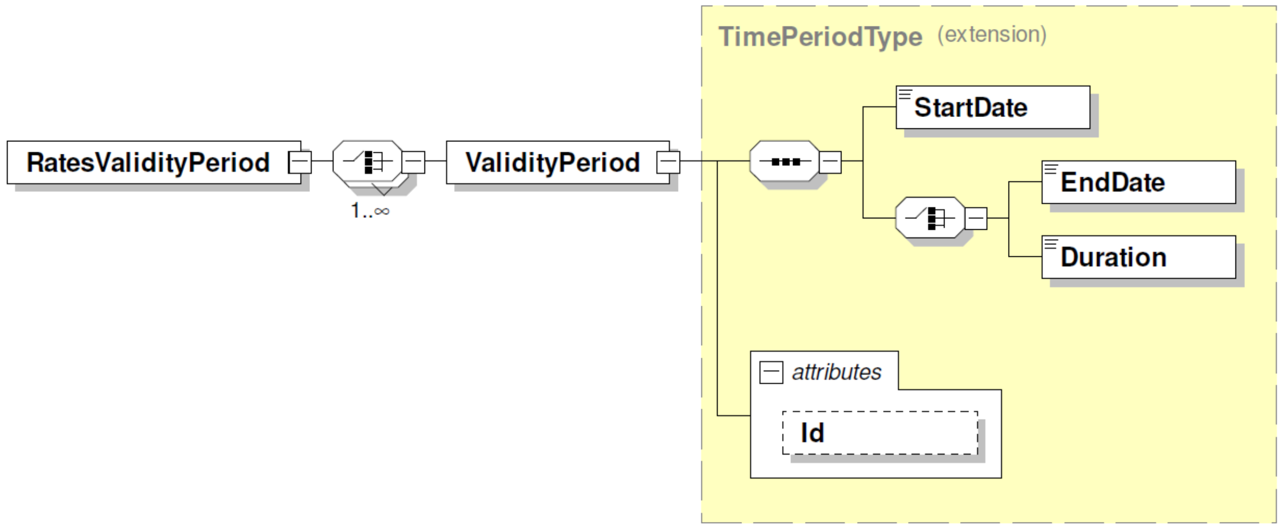


Figure 46 — Structure of RatesValidityPeriod element

RatesValidityPeriod is an XML element that includes zero or more XML elements *ValidityPeriod*.

ValidityPeriod is an ancillary XML element of type TimePeriodType.

The postal rate's validity includes one or more validity intervals. The XML element that defines rate's validity is a collection of elements each specifying a time interval when the rate is valid. Such structure supports a flexible validity policy, allowing postal operators to introduce product rates which are valid only for a limited time (e.g. during the holiday season or when discounting is needed to stimulate demand).

As shown in Figure 46, each rate file shall have at least one validity interval. The validity interval is defined from a start date. The end of validity interval may be defined either as an end date or by defining the duration of the validity period. By convention if the *Duration* of the validity interval equals zero, then the rate has no specified end date. The rate validity interval is distinct from the validity of the rate definition file describing the product. Both the rate validity and the rate definition file make use of the same XML construct (TimePeriodType). The rate definition file has its own expiration date (see 6.2.2.2) for the convenience of computer files management.

6.19 Terms and agreements

Terms and agreements represents data related to postal rates supplied by the postal product provider that is necessary to reference applicable legal documents. The following figure represents the structure of the element "*TermsAndAgreements*" of the EPR schema.

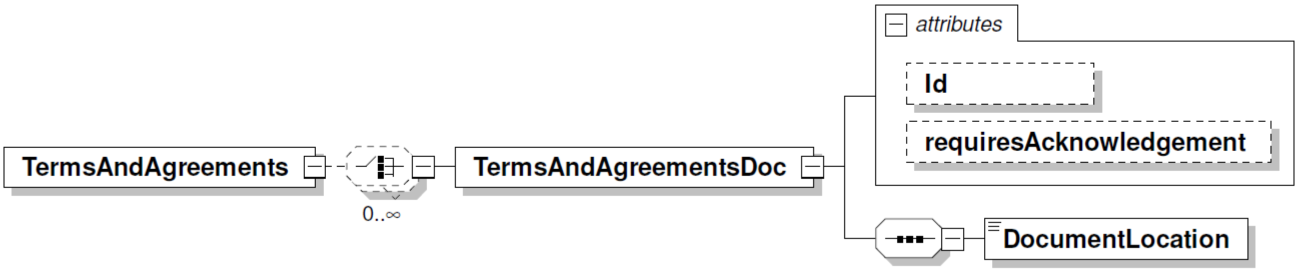


Figure 47 — Structure of TermsAndAgreements element

TermsAndAgreements element is a collection of references to zero or more legal documents (*TermsAndAgreementsDoc*). These references are necessary to create a complete environment for selection and use of the postal product that fulfills applicable legal requirements. An example of legal requirements is the requirement that the mailer shall provide an accurate statement of mailing submission (SMS) in order to buy a given postal product. It is frequently required that that the discrepancy between actual mailing data and the information describing the mailing provided in the SMS should not exceed a pre-specified threshold. For example the number of mail units in the mailing and in the SMS should not differ by more than 3 %. The element TermsAndAgreements requires "Acknowledgement" indicative of the fact that the mailer shall acknowledge a legal document as a pre-condition of mail acceptance by a postal operator or a carrier.

6.20 Information for human consumption

Information for the human consumption represents data supplied by the postal product provider that is designed to improve the interaction between mailers and postal operators and carriers. This information may explain various product features and enhancements and facilitates better use of postal products. This info also supports the creation of a complete user interface for use with new and existing postal products. Examples of information for human consumption include customer service contacts and mail preparation instructions.

Figure 48 represents the structure of the element "*InfoForHumanConsumption*" of the EPR schema.

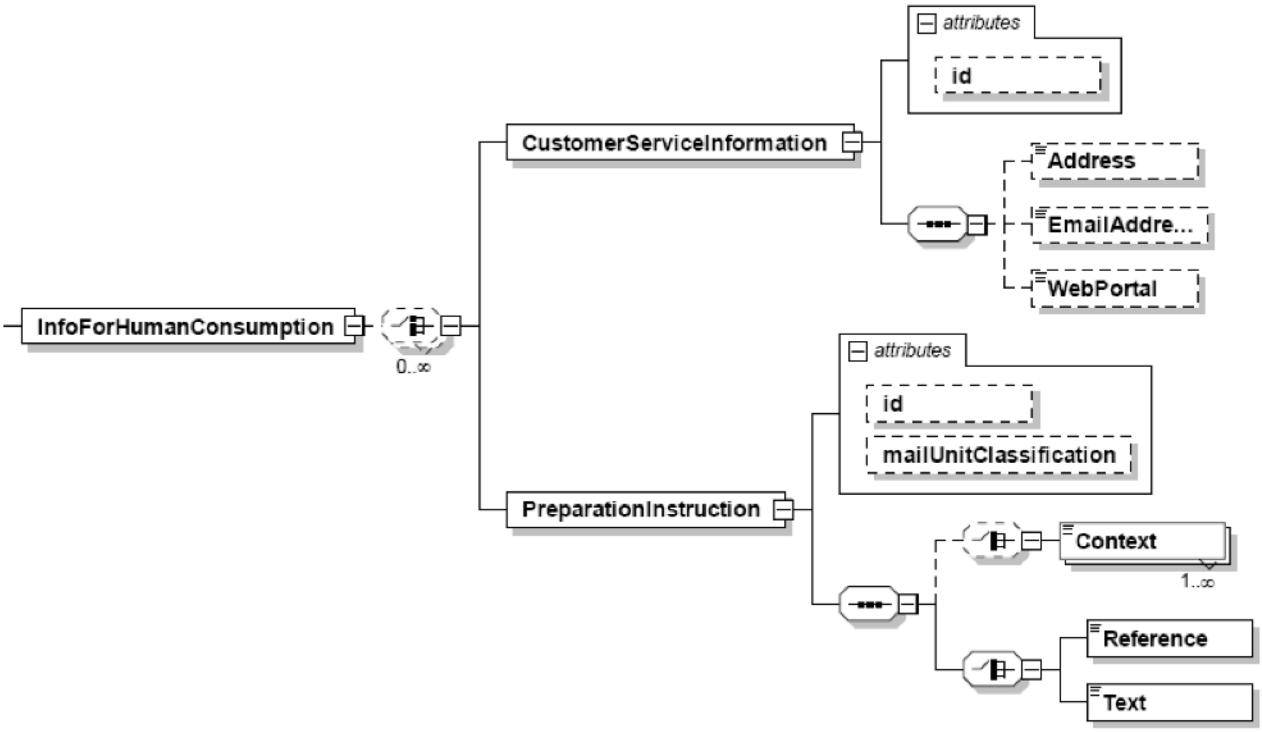


Figure 48 — Structure of InfoForHumanConsumption element

The element InfoForHumanConsumption may include various elements such as references to web resources, documents and Email or physical addresses where preparation instructions, customer service contacts and other information can be found.

6.21 Foundation types

6.21.1 Table2D type

Postal rates are functions of multiple parameters that can be represented for computer processing as multidimensional data structures (multidimensional tables). By convention, postal operators and carriers publish postal rates as two dimensional tables providing a set of conditions, rules and parameters designed to facilitate the conversion of the multidimensional tables into a set of two-dimensional (2D) tables. These conditions, rules and parameters are used by humans for navigating the multidimensional tables. In this document the XML Table2D type is the structure that represents a multidimensional table as a searchable collection of two-dimensional table. Each 2D table in the collection is uniquely identified by a parameter stored in the XML element Key.

The Table2D element contains an attribute (Id) and a collection of sequences, each of which consists of four elements (AlgorithmId, Key, FirstRow and Row) as represented in the following figure:

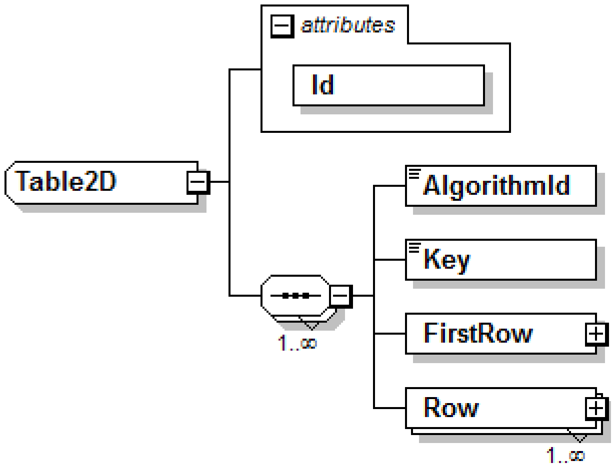


Figure 49 — Structure of Table2D type – top level

The first element in each sequence is the AlgorithmId, followed by the element Key that identifies the 2D table in the collection of rows which is part of the same sequence. This allows unambiguous identification of 2D tables that are stored as a collection. The AlgorithmId element is used to identify variations of the search algorithm to be used with the data contained in the two-dimensional table.

In order to accommodate search in 2D table the first row and the remaining rows are separated in two elements as shown in the diagram. The function of first row is to store the values of the table columns headings. The remaining rows store the rest of data in the table in XML elements of type Row. The table's data values are stored in XML elements of type DataElement.

At its core, Table2D is a collection of data elements organized in one or more rows (XML type RowType).

To facilitate searches, the first data element of each row is stored in a separate element, FirstDataElement.

The Table2D element is represented in its entirety in the following figure:

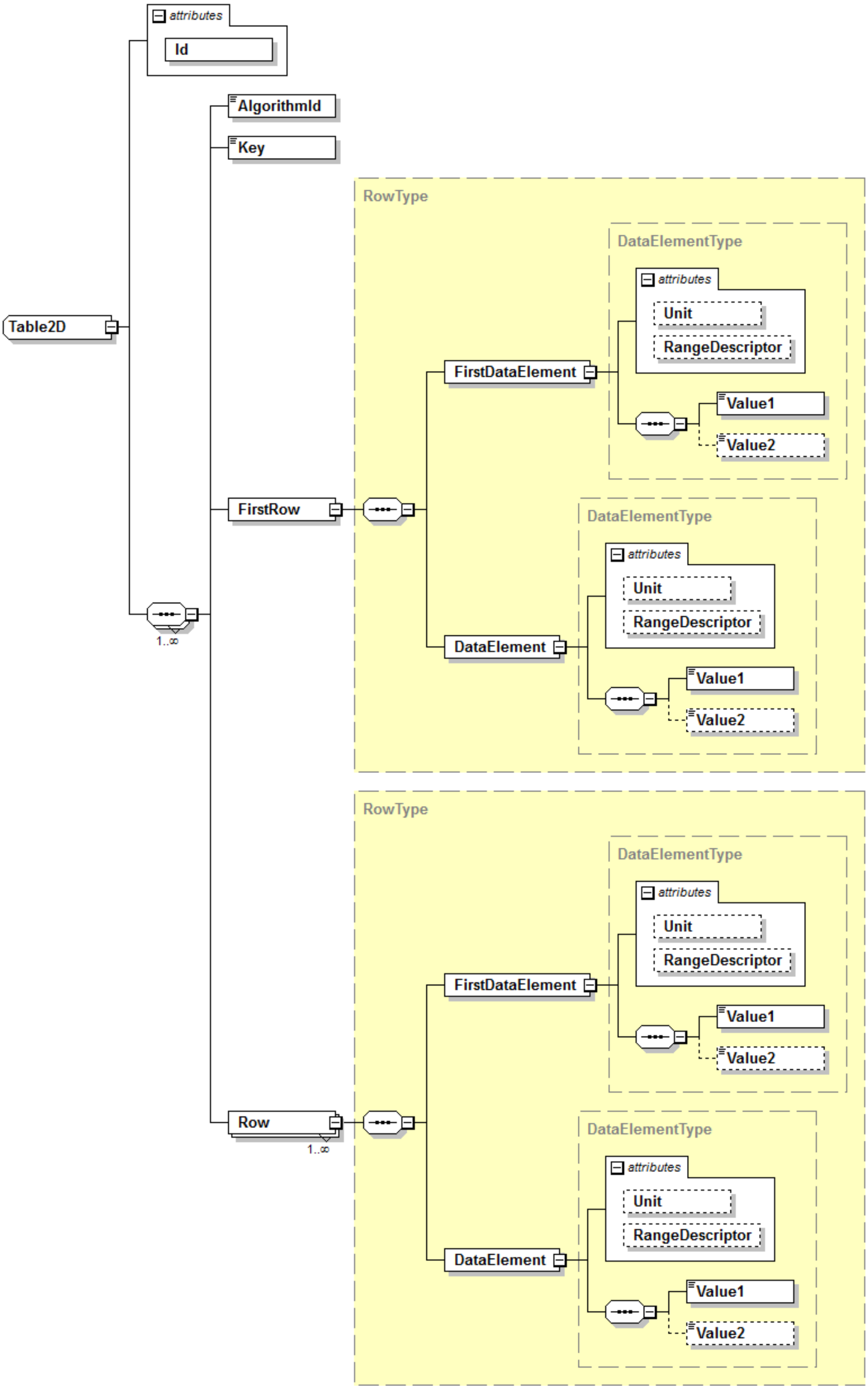


Figure 50 — Structure of Table2D type - detailed

The Table2D element is organized as a collection of one or more sequences, each sequence containing an AlgorithmID element, a Key element, a FirstRow element and one or more Row elements.

The FirstRow element is of the same type as the Row elements (RowType). It is distinguished from other rows because it is intended to contain Unit and RangeDescriptor elements with different values.

Detailed examples of usage of elements of Table2D type can be found in Annexes A, B and C. In each of the examples, the BaseAmount element is implemented as a Table2D type. In the Annex C the element **Id** has the value “*RM_2011_10_03_FirstClass”*, the element **AlgorithmId** has value “LookupTable2D”, etc

6.21.2 RowType type

The RowType is an XML element that is the main building block for the Table2D type (see above). It is a sequence of two elements, both of type DataElementType (see below). The separation of the first data element from all the other data elements facilitates searches through the elements of the row.

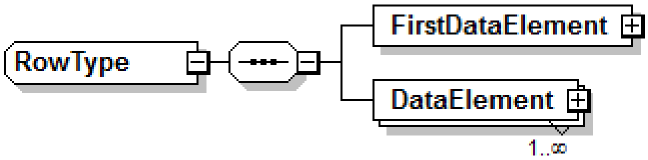


Figure 51 — Structure of RowType type

Detailed examples of usage of the element RowType can be found in Annexes A, B and C in representation of the element **BaseAmount**..

6.21.3 DataElementType type

The data element type contains individual data elements of the rate table. The Value1 element stores the mandatory data value. Value2 element stores a second, optional value when appropriate.

The optional Unit attribute contains description of the unit of measurement applicable to the value stored in the element Value1 and (optionally) Value2. Value1 and Value2 support representation of the value intervals (for example when a given rate applies to a range of mail unit weights). The optional RangeDescriptor attribute provides codified indication for interpretation of Value1 and Value2 when they define an interval. The interval may be open or close ended depending on whether Value1 and/or Value2 are included into the interval.

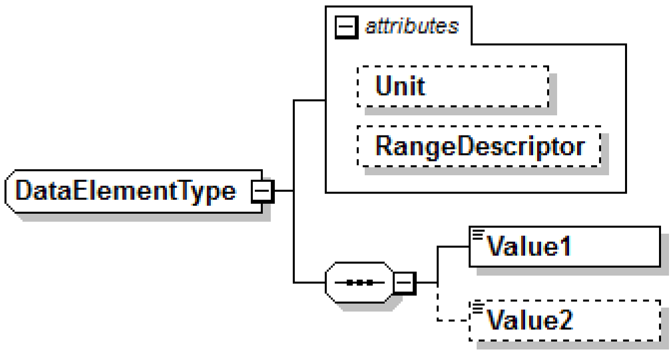


Figure 52 — Structure of DataElementType type

Detailed examples of usage of the element of the type DataElement can be found in Annexes A, B and C in representation of the element **BaseAmount**..

6.21.4 AlgorithmAndValue type

The postal rate calculation process shown in Figure 5 makes use of several types of simple, well known and widely used algorithms. These algorithms are thresholding (threshold above value, threshold below a value) and selection of one of several possible inputs. When the algorithm requires knowledge of one or more parameters (e.g. the value of the threshold or the selected input) then the values of these parameters are stored in the optional element Value.

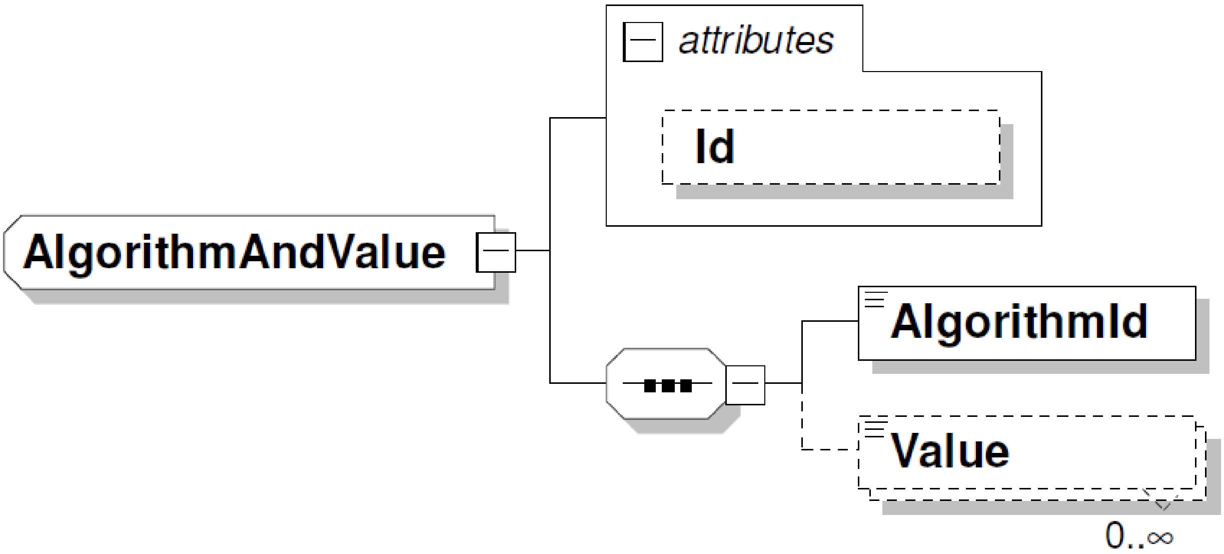


Figure 53 —Structure of AlgorithmAndValue type

All algorithms discussed in this document are uniquely identified by the value stored in the AlgorithmId element and are shared between postal operators, mailers and providers of mailing equipment and software as further explained in Clause 5.

An example of usage of an element of AlgorithmAndValue type is discussed in 6.3. In the XML schema fragment below, the algorithm identifier is "*CalculatedWeightAlgo342"* and the parameters of the calculated weight are the numbers 5184 and 166:

```
<CalculatedWeight>
   <AlgorithmId>CalculatedWeightAlgo342</AlgorithmId>
   <Value>5184</Value>
   <Value>166</Value>
</CalculatedWeight>
```

6.21.5 TimePeriodType type

The TimePeriod type supports the representation of time periods either as an interval defined by its beginning and end dates or as the beginning date and duration.

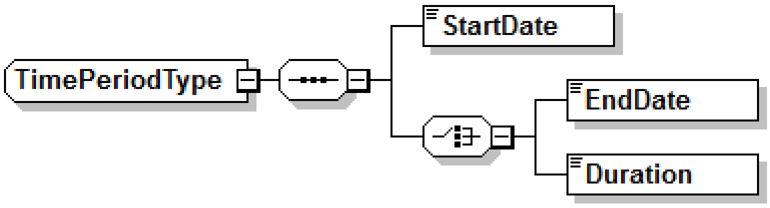


Figure 54 — Structure of TimePeriod type

The StartDate and EndDate elements are of UpuDate type.

6.21.6 UpuDate type

UpuDate type is a string. According to the UPU Glossary [6], the date is a characteristic of an event which defines, to an appropriate level of accuracy, the point in time at which it occurred or is or was forecast to occur. The *date* includes both date and time information expressed as a string using a modified UTC notation. The UPU format is different from standard XML representation of date and time. For example, the string “20020823” represents local date 23rd August 2002 (the recipient is assumed to know the time zone and the offset). See UPU Glossary [6] for additional examples and explanations.

6.21.7 DataIntegrityInfoAlgorithm type

Data integrity algorithms verify that the information received is exactly the same as the information sent.

DataIntegrityInfoAlgorthm contains only one element, the name of the integrity algorithm. Examples of data integrity algorithms are SHA1 and MD5.

The EPR schema defines two possible sources for the name of the data integrity algorithm. The name of the data integrity algorithm can be either included in the DataIntegrityInfoAlgorithmEnum list (see below) or it can be taken from a code list referenced using a format defined by CodeListType (see below). As such, the DataIntegrityInfoAlgorithm is a union of DataIntegrityInfoAlgorithmEnum and CodeListType.

The following fragment of the XML schema is an example that defines the element DataIntegrityInfoAlgorithm:

```
<xs:simpleType name=”DataIntegrityInfoAlgorithmType” >
   <xs:union memberTypes=”DataIntegrityInfoAlgorithmEnum CodeListType”/ >
</xs:simpleType>
```

The following fragment demonstrates how the element DataIntegrityInfoAlgorithm is used:

```
<PostalRate documentExpirationDate=”” DataIntegrityInfoAlgorithm = ”CRC32”
documentId = ”” dataIntegrityInfo = ””
xmlns:xsi = ”http://www.w3.org/2001/XMLSchema-instance”
xsi:noNamespaceSchemaLocation = ”EPR_Schema%20v25.xsd” >
```

Where the value of the DataIntegrityInfoAlgorithm attribute is “CRC32” XML parsers will verify that the string “CRC32” is one of the strings contained in the enumeration DataIntegrityInfoAlgorithmEnum described in detail in 6.21.8.

6.21.8 DataIntegrityInfoAlgorithmEnum enumeration

The data integrity information enumeration is a placeholder that includes the names of a few well known data integrity algorithms (SHA-2, SHA-1, MD5, CRC64 and CRC32). The inclusion of these algorithms in the list does not represent an endorsement. The selection of the data integrity algorithms for the transmission of postal rates is the responsibility of the system architects and designers.

The following fragment of the XML schema for EPR shows the implementation of the enumeration:

```
<xs:simpleType name=”DataIntegrityInfoAlgorithmEnum” >
   <xs:restriction base=”xs:string” >
      <xs:enumeration value=”SHA-2”/ >
      <xs:enumeration value=”SHA-1”/ >
      <xs:enumeration value=”MD5”/ >
      <xs:enumeration value=”CRC64”/ >
      <xs:enumeration value=”CRC32”/ >
   </xs:restriction>
</xs:simpleType>
```

6.21.9 CodeListType type

The data integrity algorithms can be referenced by name from a specified code list. The name of the code list is expressed as a string with restrictions. The format of the pattern is as follows:

CodeList:Source:.+;Id:.+;(Value:.+;)*(URI:.+;)?

See XML schema standard [1] for a detailed description of constraints on string content.

The UPU S54 standard (EPPML) describes the CodeListType as follows:

The EPPML schema may constrain the content of some of its elements. In some cases this can be achieved by using enumerations included in the EPPML schema. In other cases, the content of the element needs to reflect the vocabulary used by the postal organization which defines postal product. The mechanism employed by EPPML to reflect vocabulary specific to a postal organization (a.k.a. Lexicon) is the code list.

The *CodeListType* contains a template that allows posts to define terms in addition to enumerations already included in the schema. The code list is a string containing its source, an identifier and, optionally, one or more values from the code list. This allows for the selection of specific elements from a code list. Also, the template for code list may include the URI of the code list to facilitate automated validation of PPDFs (Postal Product Definition Files).

The CodeListType is a XML Simple Type which contains a string (xs:string) with restrictions. The restrictions are defined by the following template (Regular Expression):

CodeList:Source:.+;Id:.+;(Value:.+;)*(URI:.+;)?

The template comprises a prefix ‘CodeList:’, followed by the prefix ‘Source:’, followed by one or more occurrences of any Unicode character (indicating the name of the source of the code list), a separator ‘;’, followed by the prefix ‘Id:’, followed by one or more occurrences of any Unicode character (indicating the identity of the code list) and a separator ‘;’, followed by two optional strings. The optional strings contain zero or more values and the location of the code list expressed as an URI.

XML elements of type CodeListType may be used in several ways:

1) to identify a code list. For example, UPU code list number 106 contains Event Reason Codes. An element of a PPDF that refers to it has the value: ‘CodeList:Source:UPU;Id106;’

2) to provide one or more values from a specified code list. For example, events 45, 55, 56 and 63 from UPU code list 106 are expressed as: ‘CodeList:Source:UPU;Id106;Value:45;Value:55;Value:56;Value:63;’

3) to provide the URI of the code list. For example, UPU code list number 106 is published by UPU at http://www.upu.int/standards/106.txt. This can be expressed as: ‘CodeList:Source:UPU;Id106;URI: http://www.upu.int/standards/106.txt.

Annex A
(informative)

Example of EPR definition file – La Poste COLISSIMO OUTRE-MER

Examples in this Technical Specification are provided only for illustrative purposes. They are intended to facilitate the understanding of the postal rate data structures represented using EPR standard. The examples described in this Technical Specification should not be used verbatim in the distribution of postal rates and preparation of mail units.

The example postal rate file, as an XML document, is composed of element tags and element values. It is worth noting that most of the text of the XML document shown is generated automatically by software tools designed to work with the EPR schema. The creation of a postal rate file involves specifying values where appropriate (e.g. '11', '3.5') or selection from a list of available choices defined in the EPR schema (e.g. string 'SHA2' from a list of data integrity algorithms). The XML is quite verbose, but the process of creating the actual document file (such as a postal rates file) is well supported by commercially available and open source tools. The development of a software application for the editing of postal rates files is within the normal capabilities of most IT organizations.

The following example modelled after La Poste COLISSIMO OUTRE-MER product has the following characteristics:

— The postal product is COLISSIMO OUTRE-MER. The identifier chosen by the postal operator is COLL-O-M. The value of the product identifier is stored in the XML element "*key*".

— The rate data are stored in a 2D table. This is indicated by the value of the *AlgorithmId* element of the *BaseAmount* which is "*2D-lookup".*

— The first data row contains the headings for the columns: Weight, ZoneOM1 and ZoneOM2. The first column has the heading Weight and it contains values for the upper limit of weight ranges.

— The product contains rates for two zones: ZoneOM1 and ZoneOM2. The values ZoneOM1 and ZoneOM2 in the first row indicate that the rates for ZoneOM1 are in the second column and the rates for ZoneOM2 are in the third column.

— In each row, the first element is the value of the upper limit of the weight range. The prices for the ZoneOM1, respectively ZoneOM2 are in the second and third values.

— This example shows only the first four lines of the rates table.

- XML
  - = version: 1.0
  - = encoding: UTF-16
- PostalRate
  - = documentExpirationDate
  - = dataIntegrityInfoAlgorithm: SHA-2
  - = documentId
  - = dataIntegrityInfo
  - = xmlns:*xsi*: http://www.w3.org/2001/XMLSchema-instance
  - = *xsi*:noNamespaceSchemaLocation: EPR_Schema.xsd
  - DimensionalWeight
    - () AlgorithmId
  - CalculatedWeight
    - () AlgorithmId
  - WeightRounding
    - () AlgorithmId
  - () ZoneOrDistance
  - CalculatedDistance
    - () AlgorithmId
  - BaseAmount
    - = Id
    - () AlgorithmId: 2D-Lookup
    - () Key: COLISSIMO OUTRE-MER
    - FirstRow
      - FirstDataElement
        - () Value1: Weight(kg)
      - DataElement
        - () Value1: ZoneOM1(Eu...
      - DataElement
        - () Value1: ZoneOM2(Eu...
    - Row
      - FirstDataElement
        - () Value1: 0.5
      - DataElement
        - () Value1: 8.45
      - DataElement
        - () Value1: 10.10
    - Row
      - FirstDataElement
        - () Value1: 1
      - DataElement
        - () Value1: 12.70
      - DataElement
        - () Value1: 15.20
    - Row
      - FirstDataElement
        - () Value1: 2
      - DataElement
        - () Value1: 17.35
      - DataElement
        - () Value1: 26.80
    - Row
      - FirstDataElement
        - () Value1: 3
      - DataElement
        - () Value1: 22
      - DataElement
        - () Value1: 38.40
  - AmountRounding

	{} AlgorithmId	
{} DistanceAndWeightFees		
{} DimensionFees		
{} ValueFees		
{} QuantityFees		
{} AmountFees		
{} DateFees		
{} ValueAddedFees		
{} GenericFees		
{} Taxes		
{} CustomsCharges		
RatesValidityPeriod		
	ValidityPeriod	
		{} StartDate
		{} EndDate
{} TermsAndAgreements		
{} InfoForHumanConsumption		

The following document represents human readable postal rates document corresponding to the EPR document above.

COLISSIMO OUTRE-MER

Affranchissement au poids avec preuve de dépôt, livraison sans signature, suivi[1].
Indemnisation intégrée en cas de perte ou avarie de 23 €/kg (frais de port compris).

Poids jusqu'à	Tarifs nets **Zone OM 1**	Tarifs nets **Zone OM 2**
0,5 kg	8,45 €	10,10 €
1 kg	12,70 €	15,20 €
2 kg	17,35 €	26,80 €
3 kg	22,00 €	38,40 €
4 kg	26,65 €	50,00 €
5 kg	31,30 €	61,60 €
6 kg	35,95 €	73,20 €
7 kg	40,60 €	84,80 €
8 kg	45,25 €	96,40 €
9 kg	49,90 €	108,00 €
10 kg	54,55 €	119,60 €
15 kg	77,75 €	177,60 €
20 kg	100,95 €	235,60 €
25 kg	124,15 €	293,60 €
30 kg	147,35 €	351,60 €

Annex B
(informative)

Example of EPR definition file – USPS First Class Single Piece

The following example modelled after USPS first class letter presort product has the following characteristics:

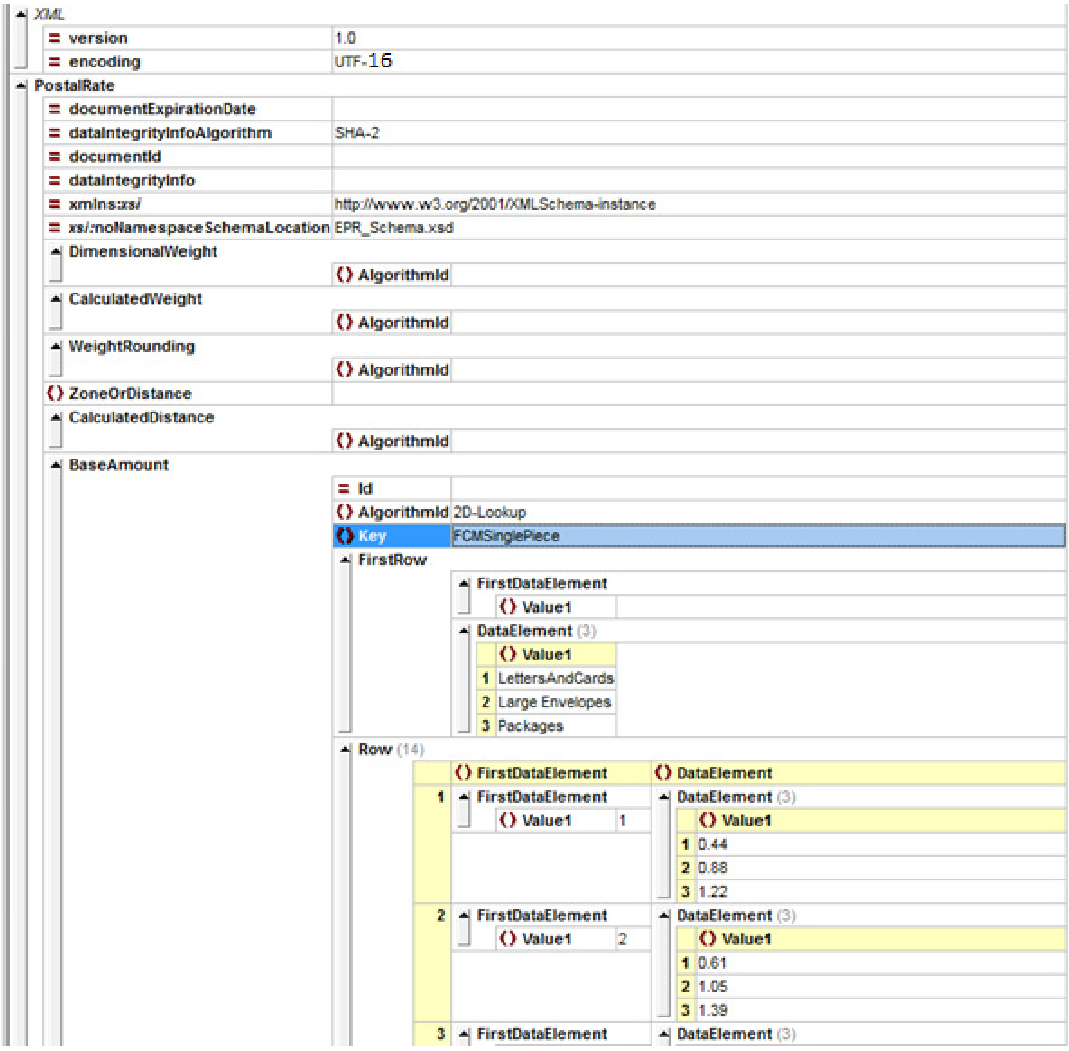
— The postal product is First Class Mail Single Piece. The identifier chosen by the postal operator for this product is *FCMSinglePiece*. The value of the product identifier is stored in the XML element “*key”*.

— The rate data are stored in a 2D table. This is indicated by the value of the *AlgorithmId* element of the *BaseAmount* which is “*2D-lookup”*.

— The table contains data for three types of mail units: Letters and Cards, Large Envelopes and Packages. The first row of the 2D table as described in 6.21.1 stores the column headings with names corresponding to each type of mail unit. The table contains 14 rows each of which stores four values. The first value is the upper value of the weight range for mail unit and the next three values are the base amount for each type of mail having the corresponding weight.

— The validity period for the rate starts on 01 January 2011 and ends on 31 December 2011 and there is only one validity period.

The following three figures are screen shots of the example rate table.


XML
version
1.0
encoding
UTF-16
PostalRate
documentExpirationDate
dataIntegrityInfoAlgorithm
SHA-2
documentId
dataIntegrityInfo
xmlns:xsi
http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation
EPR_Schema.xsd
DimensionalWeight
AlgorithmId
CalculatedWeight
AlgorithmId
WeightRounding
AlgorithmId
ZoneOrDistance
CalculatedDistance
AlgorithmId
BaseAmount
Id
AlgorithmId
2D-Lookup
Key
FCMSinglePiece
FirstRow
FirstDataElement
Value1
DataElement (3)
Value1
1 LettersAndCards
2 Large Envelopes
3 Packages
Row (14)
FirstDataElement
DataElement
1
FirstDataElement
Value1
1
DataElement (3)
Value1
1 0.44
2 0.88
3 1.22
2
FirstDataElement
Value1
2
DataElement (3)
Value1
1 0.61
2 1.05
3 1.39
3
FirstDataElement
DataElement (3)

	FirstDataElement		DataElement (3)		
	Value1		Value1		
			1	2	3
	Value1	3	0.78	1.22	1.56
4	Value1	3.5	0.95	NA	NA
5	Value1	4	NA	1.39	1.73
6	Value1	5	NA	1.56	1.90
7	Value1	6	NA	1.73	2.07
8	Value1	7	NA	1.90	2.24
9	Value1	8	NA	2.07	2.41
10	Value1	9	NA	2.24	2.58
11	Value1	10	NA	2.41	

			3 | 2.75
12	FirstDataElement Value1 | 11	DataElement (3)	Value1 1 | NA 2 | 2.58 3 | 2.92
13	FirstDataElement Value1 | 12	DataElement (3)	Value1 1 | NA 2 | 2.75 3 | 3.09
14	FirstDataElement Value1 | 13	DataElement (3)	Value1 1 | NA 2 | 2.92 3 | 2.07

- AmountRounding
  - AlgorithmId
- DistanceAndWeightFees
- DimensionFees
- ValueFees
- QuantityFees
- AmountFees
- DateTimeFees
- ValueAddedFees
- GenericFees
- Taxes
- CustomsCharges
- RatesValidityPeriod
  - ValidityPeriod
    - StartDate: 01-01-2011
    - EndDate: 31-12-2011
- TermsAndAgreements
- InfoForHumanConsumption

Annex C
(informative)

Example of XML schema for base amount calculation using product key input

The following example shows a portion of the XML document (rates file) that implements the base amount for the Royal Mail First Class Letter products (Figure 22). There are four postal products, each identified by its own key: First Class Letter evidenced with Stamps or SmartStamp, First Class Letter evidenced with Franking or Account, First Class Large Letter evidenced with Stamps or SmartStamp and First Class Large Letter evidenced with Franking or Account.

To improve the readability of the example, the keys were chosen to be long strings that can be interpreted by the human reader. Each postal operator may chose identifiers that are already used in their information systems for each product.

Below is a screen shot of the rates table for these four products and the corresponding XML document:

- BaseAmount
  - Id: RM_2011_10_03_FirstClass
  - AlgorithmId: LookupTable2D
  - Key: FirstClassLetterStamp
  - FirstRow
    - FirstDataElement
      - Value1: Weight(g)
    - DataElement
      - Value1: Price(Euro)
  - Row (1)

	FirstDataElement	DataElement
1	FirstDataElement: Value1 100	DataElement: Value1 0.41

  - AlgorithmId: LookupTable2D
  - Key: FirstClassLetterFranking
  - FirstRow
    - FirstDataElement
      - Value1: Weight(g)
    - DataElement
      - Value1: Price(Euro)
  - Row (1)

	FirstDataElement	DataElement
1	FirstDataElement: Value1 100	DataElement: Value1 0.36

  - AlgorithmId: LookupTable2D
  - Key: FirstClassLargeLetterStamp
  - FirstRow
    - FirstDataElement
      - Value1: Weight(g)
    - DataElement
      - Value1: Price(Euro)
  - Row (4)

	FirstDataElement	DataElement
1	FirstDataElement: Value1 100	DataElement: Value1 0.66
2	FirstDataElement: Value1 250	DataElement: Value1 0.96
3	FirstDataElement: Value1 500	DataElement: Value1 1.32
4	FirstDataElement: Value1 750	DataElement: Value1 1.87

  - AlgorithmId: LookupTable2D
  - Key: FirstClassLargeLetterFranking
  - FirstRow
    - FirstDataElement
      - Value1: Weight(g)
    - DataElement
      - Value1: Price(Euro)
  - Row (4)

	FirstDataElement	DataElement
1	FirstDataElement: Value1 100	DataElement: Value1 0.50
2	FirstDataElement: Value1 250	DataElement: Value1 0.72
3	FirstDataElement: Value1 500	DataElement: Value1 1.04
4	FirstDataElement: Value1 750	DataElement: Value1 1.51

```
<BaseAmount Id="RM_2011_10_03_FirstClass" >
   <AlgorithmId>LookupTable2D</AlgorithmId>
   <Key>FirstClassLetterStamp</Key>
   <FirstRow>
      <FirstDataElement>
         <Value1>Weight(g)</Value1>
      </FirstDataElement>
      <DataElement>
         <Value1>Price(Euro)</Value1>
      </DataElement>
   </FirstRow>
   <Row>
      <FirstDataElement>
         <Value1>100</Value1>
      </FirstDataElement>
      <DataElement>
         <Value1>0.41</Value1>
      </DataElement>
   </Row>
   <AlgorithmId>LookupTable2D</AlgorithmId>
   <Key>FirstClassLetterFranking</Key>
   <FirstRow>
      <FirstDataElement>
         <Value1>Weight(g)</Value1>
      </FirstDataElement>
      <DataElement>
         <Value1>Price(Euro)</Value1>
      </DataElement>
   </FirstRow>
   <Row>
      <FirstDataElement>
         <Value1>100</Value1>
      </FirstDataElement>
      <DataElement>
         <Value1>0.36</Value1>
      </DataElement>
   </Row>
   <AlgorithmId>LookupTable2D</AlgorithmId>
   <Key>FirstClassLargeLetterStamp</Key>
   <FirstRow>
      <FirstDataElement>
         <Value1>Weight(g)</Value1>
      </FirstDataElement>
      <DataElement>
         <Value1>Price(Euro)</Value1>
      </DataElement>
   </FirstRow>
   <Row>
      <FirstDataElement>
         <Value1>100</Value1>
      </FirstDataElement>
      <DataElement>
         <Value1>0.66</Value1>
      </DataElement>
   </Row>
   <Row>
      <FirstDataElement>
```

```
        <Value1>250</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>0.96</Value1>
    </DataElement>
</Row>
<Row>
    <FirstDataElement>
        <Value1>500</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>1.32</Value1>
    </DataElement>
</Row>
<Row>
    <FirstDataElement>
        <Value1>750</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>1.87</Value1>
    </DataElement>
</Row>
<AlgorithmId>LookupTable2D</AlgorithmId>
<Key>FirstClassLargeLetterFranking</Key>
<FirstRow>
    <FirstDataElement>
        <Value1>Weight(g)</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>Price(Euro)</Value1>
    </DataElement>
</FirstRow>
<Row>
    <FirstDataElement>
        <Value1>100</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>0.50</Value1>
    </DataElement>
</Row>
<Row>
    <FirstDataElement>
        <Value1>250</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>0.72</Value1>
    </DataElement>
</Row>
<Row>
    <FirstDataElement>
        <Value1>500</Value1>
    </FirstDataElement>
    <DataElement>
        <Value1>1.04</Value1>
    </DataElement>
</Row>
<Row>
```

```
        <FirstDataElement>
            <Value1>750</Value1>
        </FirstDataElement>
        <DataElement>
            <Value1>1.51</Value1>
        </DataElement>
    </Row>
</BaseAmount>
```

Annex D
(informative)

EPR schema files

The full text will be distributed with the electronic version.

```
<?xml version="1.0" encoding = "UTF-16"? >
<!- edited with XMLSpy v2011 sp1 (http://www.altova.com) by Andrei Obrea (Pitney
Bowes) - >
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault = "qualified" attributeFormDefault = "unqualified" >
   <xs:element name="PostalRate" >
      <xs:annotation>
         <xs:documentation>information necessary and sufficient to calculate the
price of a mail unit subject to an identified postal product when using a given
calculation model</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element name="DimensionalWeight" type = "AlgorithmAndValue"
minOccurs = "1" >
               <xs:annotation>
                  <xs:documentation>Data_11
Indicates which method is used to  calculate the dimensional weight, when
applicable.</xs:documentation>
               </xs:annotation>
            </xs:element>
            <xs:element name="CalculatedWeight" type = "AlgorithmAndValue"
minOccurs = "1" >
               <xs:annotation>
                  <xs:documentation>Data_12
Choice between two data sources. The output takes the value of the first data
input if it is below a threshold.
The threshold is contained in the data file.</xs:documentation>
               </xs:annotation>
            </xs:element>
            <xs:element name="WeightRounding" type = "AlgorithmAndValue"
minOccurs = "1" >
               <xs:annotation>
                  <xs:documentation>Data_16
</xs:documentation>
               </xs:annotation>
            </xs:element>
            <xs:element name="ZoneOrDistance" >
               <xs:complexType>
                  <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                     <xs:element name="ZoneOrDistanceTable" type = "Table2D" >
                        <xs:annotation>
                           <xs:documentation>Data_13
Lookup into a 2D table for the value representing the zone or the value of the
distance between origin and destination. Tables may be product dependent, so the
same origin/destination pair may result into different expressions or values for
```

```
the distance. The product 'key' input is used to identify the postal
product.</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name=”CalculatedDistance” type = ”AlgorithmAndValue”/ >
            <xs:element name=”BaseAmount” type = ”Table2D” >
               <xs:annotation>
                  <xs:documentation>Data_15</xs:documentation>
               </xs:annotation>
            </xs:element>
            <xs:element name=”AmountRounding” type = ”AlgorithmAndValue”
minOccurs = ”1” >
               <xs:annotation>
                  <xs:documentation>Data_17
</xs:documentation>
               </xs:annotation>
            </xs:element>
            <xs:element name=”DistanceAndWeightFees” >
               <xs:complexType>
                  <xs:choice minOccurs=”0” maxOccurs = ”unbounded” >
                     <xs:element name=”DistanceAndWeightFee” type = ”Table2D” >
                        <xs:annotation>
                           <xs:documentation>Data_22
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name=”DimensionFees” >
               <xs:complexType>
                  <xs:choice minOccurs=”0” maxOccurs = ”unbounded” >
                     <xs:element name=”DimensionFee”
type = ”AlgorithmAndValue” >
                        <xs:annotation>
                           <xs:documentation>Data_23
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name=”ValueFees” >
               <xs:complexType>
                  <xs:choice minOccurs=”0” maxOccurs = ”unbounded” >
                     <xs:element name=”ValueFee” type = ”Table2D” >
                        <xs:annotation>
                           <xs:documentation>Data_24
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
```

```
            <xs:element name="QuantityFees" >
               <xs:complexType>
                  <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                     <xs:element name="QuantityFee" type = "Table2D" >
                        <xs:annotation>
                           <xs:documentation>Data_25
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="AmountFees" >
               <xs:complexType>
                  <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                     <xs:element name="AmountFee" type = "Table2D" >
                        <xs:annotation>
                           <xs:documentation>Data_26
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="DateFees" >
               <xs:complexType>
                  <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                     <xs:element name="DateFee" type = "Table2D" >
                        <xs:annotation>
                           <xs:documentation>Data_27
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="ValueAddedFees" >
               <xs:complexType>
                  <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                     <xs:element name="ValueAddedFee" type = "Table2D"
minOccurs = "0" >
                        <xs:annotation>
                           <xs:documentation>Data_28
</xs:documentation>
                        </xs:annotation>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="GenericFees" >
               <xs:complexType>
                  <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                     <xs:element name="GenericFee" type = "Table2D" >
                        <xs:annotation>
                           <xs:documentation>Data_28
</xs:documentation>
                        </xs:annotation>
```

```
                    </xs:element>
                 </xs:choice>
              </xs:complexType>
           </xs:element>
           <xs:element name="Taxes" >
              <xs:complexType>
                 <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                    <xs:element name="Tax" type = "Table2D"/ >
                 </xs:choice>
              </xs:complexType>
           </xs:element>
           <xs:element name="CustomsCharges" >
              <xs:complexType>
                 <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                    <xs:element name="CustomsCharge" type = "Table2D"/ >
                 </xs:choice>
              </xs:complexType>
           </xs:element>
           <xs:element name="RatesValidityPeriod" >
              <xs:annotation>
                 <xs:documentation>Time intervals when the product may be
used</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                 <xs:choice maxOccurs="unbounded" >
                    <xs:element name="ValidityPeriod" >
                       <xs:complexType>
                          <xs:complexContent>
                             <xs:extension base="TimePeriodType" >
                                <xs:attribute name="Id" >
                                   <xs:simpleType>
                                      <xs:restriction base="xs:string" >
                                         <xs:pattern
value="ValidityInterval:\d+(\.?\d+)*"/ >
                                      </xs:restriction>
                                   </xs:simpleType>
                                </xs:attribute>
                             </xs:extension>
                          </xs:complexContent>
                       </xs:complexType>
                    </xs:element>
                 </xs:choice>
              </xs:complexType>
           </xs:element>
           <xs:element name="TermsAndAgreements" >
              <xs:annotation>
                 <xs:documentation>Collection of references to legal
documents</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                 <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                    <xs:element name="TermsAndAgreementsDoc" >
                       <xs:annotation>
                          <xs:documentation>Reference to legal
documents</xs:documentation>
                       </xs:annotation>
                       <xs:complexType>
```

```
                              <xs:sequence>
                                 <xs:element name="DocumentLocation" 
type = "xs:anyURI"/ >
                              </xs:sequence>
                              <xs:attribute name="Id" >
                                 <xs:simpleType>
                                    <xs:restriction base="xs:string" >
                                       <xs:pattern 
value="TermsAndAgreements:\d+(\.?\d+)*"/ >
                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:attribute>
                              <xs:attribute name="requiresAcknowledgement" 
type = "xs:boolean"/ >
                           </xs:complexType>
                        </xs:element>
                     </xs:choice>
                  </xs:complexType>
               </xs:element>
               <xs:element name="InfoForHumanConsumption" >
                  <xs:annotation>
                     <xs:documentation>Information that improves the interaction 
with the post</xs:documentation>
                  </xs:annotation>
                  <xs:complexType>
                     <xs:choice minOccurs="0" maxOccurs = "unbounded" >
                        <xs:element name="CustomerServiceInformation" >
                           <xs:complexType>
                              <xs:sequence>
                                 <xs:element name="Address" type = "xs:string" 
minOccurs = "0"/ >
                                 <xs:element name="EmailAddress" type = "xs:string" 
minOccurs = "0"/ >
                                 <xs:element name="WebPortal" type = "xs:anyURI" 
minOccurs = "0"/ >
                              </xs:sequence>
                              <xs:attribute name="Id" >
                                 <xs:simpleType>
                                    <xs:restriction base="xs:string" >
                                       <xs:pattern 
value="CustomerServiceInformation:\d+(\.?\d+)*"/ >
                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:attribute>
                           </xs:complexType>
                        </xs:element>
                        <xs:element name="PreparationInstruction" >
                           <xs:annotation>
                              <xs:documentation>Text describing actions to be taken 
by mailer in preparation for submitting mail item. Includes options, packaging 
material, etc. The structure and the information supports context-sensitive help 
for mailer's system</xs:documentation>
                           </xs:annotation>
                           <xs:complexType>
                              <xs:sequence>
                                 <xs:choice minOccurs="0" >
```

```
                                 <xs:element name="Context" type = "xs:string"
maxOccurs = "unbounded" >
                                    <xs:annotation>
                                       <xs:documentation>Id of elements uniquely
identified in other elements of the PPDF</xs:documentation>
                                    </xs:annotation>
                                 </xs:element>
                              </xs:choice>
                              <xs:choice>
                                 <xs:element name="Reference"
type = "xs:anyURI"/ >
                                 <xs:element name="Text" type = "xs:string"/ >
                              </xs:choice>
                           </xs:sequence>
                           <xs:attribute name="Id" >
                              <xs:simpleType>
                                 <xs:restriction base="xs:string" >
                                    <xs:pattern
value="PreparationInstruction:\d+(\.?\d+)*"/ >
                                 </xs:restriction>
                              </xs:simpleType>
                           </xs:attribute>
                        </xs:complexType>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
         <xs:attribute name="documentId" type = "xs:string" use = "required" >
            <xs:annotation>
               <xs:documentation>unique identifier of the postal product
definition file </xs:documentation>
            </xs:annotation>
         </xs:attribute>
         <xs:attribute name="documentExpirationDate" type = "UpuDate"
use = "required"/ >
         <xs:attribute name="dataIntegrityInfo" type = "xs:string"
use = "required" >
            <xs:annotation>
               <xs:documentation>value created using a data integrity
algorithm</xs:documentation>
            </xs:annotation>
         </xs:attribute>
         <xs:attribute name="dataIntegrityInfoAlgorithm"
type = "DataIntegrityInfoAlgorithmType" use = "required"/ >
         <xs:attribute name="datestamp" type = "UpuDate" >
            <xs:annotation>
               <xs:documentation>date when the document was
created</xs:documentation>
            </xs:annotation>
         </xs:attribute>
         <xs:attribute name="language" type = "xs:language"/ >
      </xs:complexType>
   </xs:element>
   <xs:complexType name="Table2D" >
      <xs:sequence maxOccurs="unbounded" >
         <xs:element name="AlgorithmId" minOccurs = "1"/ >
```

```
         <xs:element name="Key" >
            <xs:annotation>
               <xs:documentation>Identifies the 2D table in a postal rate
context. Intended to support 'multidimensional' data sets built from 2D
tables.</xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="FirstRow" type = "RowType" minOccurs = "1"/ >
         <xs:element name="Row" type = "RowType" maxOccurs = "unbounded" >
            <xs:annotation>
               <xs:documentation>Second and all subsequent
rows.</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:sequence>
      <xs:attribute name="Id" use = "required"/ >
   </xs:complexType>
   <xs:complexType name="RowType" >
      <xs:sequence>
         <xs:element name="FirstDataElement" type = "DataElementType"
minOccurs = "1" >
            <xs:annotation>
               <xs:documentation>Value in the first column of the current row.
Most application search this data element for the value or interval matching an
input.</xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="DataElement" type = "DataElementType"
maxOccurs = "unbounded" >
            <xs:annotation>
               <xs:documentation>Data elements in the current row corresponding
to the  second, third, ... and so on columns.</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="AlgorithmAndValue" >
      <xs:sequence>
         <xs:element name="AlgorithmId"/ >
         <xs:element name="Value" minOccurs = "0" maxOccurs = "unbounded"/ >
      </xs:sequence>
      <xs:attribute name="Id"/ >
   </xs:complexType>
   <xs:complexType name="TimePeriodType" >
      <xs:sequence>
         <xs:element name="StartDate" type = "UpuDate"/ >
         <xs:choice>
            <xs:element name="EndDate" type = "UpuDate"/ >
            <xs:element name="Duration" type = "xs:duration" >
               <xs:annotation>
                  <xs:documentation>Durations must be in a form compliant with
the ISO 8601 data standard, e.g., PnYmMpD - where n=number of years, m=number of
months, and p = number of days.</xs:documentation>
               </xs:annotation>
            </xs:element>
         </xs:choice>
      </xs:sequence>
```

```
   </xs:complexType>
   <xs:simpleType name=”UpuDate” >
      <xs:annotation>
         <xs:documentation>String containg date and time expressed according to
UPU definition of Date. It may be useful to express the UPU definition through
restrictions.</xs:documentation>
      </xs:annotation>
      <xs:restriction base=”xs:string”/ >
   </xs:simpleType>
   <xs:simpleType name=”DataIntegrityInfoAlgorithmEnum” >
      <xs:restriction base=”xs:string” >
         <xs:enumeration value=”SHA-2”/ >
         <xs:enumeration value=”SHA-1”/ >
         <xs:enumeration value=”MD5”/ >
         <xs:enumeration value=”CRC64”/ >
         <xs:enumeration value=”CRC32”/ >
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name=”DataIntegrityInfoAlgorithmType” >
      <xs:union memberTypes=”DataIntegrityInfoAlgorithmEnum CodeListType”/ >
   </xs:simpleType>
   <xs:simpleType name=”CodeListType” >
      <xs:restriction base=”xs:string” >
         <xs:pattern value=”CodeList:Source:.+;Id:.+;(Value:.+;)*(URI:.+;)?”/ >
      </xs:restriction>
   </xs:simpleType>
   <xs:complexType name=”DataElementType” >
      <xs:sequence>
         <xs:element name=”Value1”/ >
         <xs:element name=”Value2” minOccurs = ”0”/ >
      </xs:sequence>
      <xs:attribute name=”Unit”/ >
      <xs:attribute name=”RangeDescriptor”/ >
   </xs:complexType>
</xs:schema>
```

Annex E
(informative)

Intellectual property

The following US patents were brought to the attention of the editorial committee as potentially relevant:

— US 5,606,508 Assignee: Frankotyp Postalia;

— US 6,615,196 Assignee: Frankotyp Postalia;

— US 7,577,617 Assignee: Frankotyp Postalia;

— US 6,286,009 Assignee: Pitney Bowes;

— US 6,910,047 Assignee: Pitney Bowes;

— US 6,041,319 Assignee: Pitney Bowes.

These patents claim systems which make use of a postal rate table or data structures from which one may assume that they are postal rate tables.

With the exception of the US patent 6,286,009 all these patents can be considered to describe applications of the EPR language. The US patent 6,286,009 describes a platform independent rate structure. For the sake of promoting the acceptance of the EPR standard, Pitney Bowes submitted to CEN a commitment letter offering royalty free, perpetual license to US patent 6,286,009 to entities practicing this Technical Specification.

Other patents that were brought to the attention of the editorial committee are:

— US5710706 Assignee: Frankotyp Postalia;

— US7103583 Assignee: Frankotyp Postalia;

— US7739205 Assignee: Frankotyp Postalia;

— US4763271 Assignee: Pitney Bowes;

— US5448641 Assignee: Pitney Bowes;

— US5778348 Assignee: Pitney Bowes;

— WO9857305 Assignee: Pitney Bowes;

— US2009 171861 Assignee: Pitney Bowes;

— US6018725 Assignee: Pitney Bowes;

— US6078889 Assignee: Pitney Bowes;

— US6873978 Assignee: Pitney Bowes;

- US2003074324 Assignee: ASCOM HASLER MAILING SYSTEMS (US);
- US2004220889 Assignee: NEOPOST IND. (FR);
- US2006155937 Assignee: NEOPOST IND. (FR).

Bibliography

[1] W3C XML Schema, http://www.w3.org/TR/xmlschema-1/

[2] IETF BCP 47 “Internet Engineering Task Force – Best Current Practices 47” ftp://ftp.isi.edu/in-notes/bcp/bcp47.txt

[3] UPS Zone chart at http://www.ups.com/using/software/currentrates/zone-txt/786.txt (as of May 9, 2011)

[4] USPS Zone charts http://postcalc.usps.gov/Zonecharts/ (as of May 9, 2011)

[5] UPS Air Services on-line manual as of May 9, 2011– “How to determine billable weight” (http://www.ups.com/content/us/en/resources/prepare/oversize.html

[6] UPU Standards Glossary http://www.upu.int/nc/en/activities/standards/standards-documents.html (as of July 29, 2011)

[7] ISO/IEC 15459 (all parts), *Information technology — Automatic identification and data capture techniques — Unique identification*

[8] UPU S54, *Extensible Postal Product Model and Language (EPPML)*

[9] UPU S25, *Data constructs for the communication of information on postal items, batches and receptacles*

[10] EN 14615, *Postal services — Digital postage marks — Applications, security and design*

[11] ISO/IEC 10118 (all parts), *Information technology — Security techniques — Hash-functions*

[12] ISO/IEC 9797 (all parts), *Information technology — Security techniques — Data integrity mechanism using a cryptographic check function employing a block cipher algorithm*

[13] ISO/IEC 14888 (all parts), *Information technology — Security techniques — Digital signatures with appendix*

[14] UPU S36-4, *Digital Postage Marks (DPM) — Applications, security and design*

[15] UPU S43-3, *Secured electronic postal services (SePS) interface specification*

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™