EIOPA XBRL Taxonomy Documentation

LAST UPDATE 23/11/2026

Contents

I Abstract 4

II Introduction 4

III Relation to the data model 5

IV Relation to XBRL specifications 6

V Publication and distribution 7

VI Supporting concepts 8

VI.1 Model supporting schema and other technical files 8

VI.2 Public elements 10

VI.2.1 Standard labels 10

VI.2.2 Specific labels 11

VII Logical taxonomy architecture 12

VII.1 Owners 12

VII.2 Dictionary layer 13

VII.2.1 Core concepts 14

VII.2.2 Metrics 15

VII.2.3 Domains 18

VII.2.4 Explicit domain members and their relationships 19

VII.2.5 Typed domain values 22

VII.2.6 Dimensions 23

VII.3 Information requirements layer 24

VII.3.1 Frameworks 24

VII.3.2 Taxonomies 25

VII.3.3 Tables 26

VII.3.4 Modules 32

VII.3.5 Filing indicators 34

VII.3.6 Validations 36

VIII ..... Comments and processing instructions 46

IX Mapping between MD and HD properties 47

Annex 1. EIOPA XBRL Taxonomy: Owners, Folders, Files, Namespaces and Prefixes 49

Annex 2. Multiple values for a fact..... 50

I Abstract

This document describes and explains the architecture of the XBRL (eXtensible Business Reporting Language) taxonomy for second-level supervisory reporting developed by the European Insurance and Occupational Pensions Authority [EIOPA].

The aim of this document is to explain the semantics and syntax used to express the information requirements of the Data Point Model [DPM] in the XBRL format and present the structure and modularisation of the XBRL taxonomy as well as the manner of its publication and distribution.

II Introduction

This document describes the architectural approach applied in the EIOPA XBRL taxonomy.

The expected audience of this document are software developers working directly or indirectly for national competent authorities [NCAs], who will be required to pass supervisory data to EIOPA using this taxonomy. Additionally, given the possibility that this taxonomy will, to some degree, form the basis for reporting by undertakings to certain NCAs, it will also be of broader interest to firms and vendors of software involved in the regulatory reporting process (in particular, developers of tools and applications that produce or consume XBRL instance documents following this taxonomy).

Comprehension of the XBRL 2.1 Specification and various other XBRL specifications[1] such as XBRL Dimensions 1.0, XBRL Formula 1.0, Generic Link 1.0, Table Linkbase 1.0, and Extensible Enumerations 1.0 are required to understand the content of this document.

For modelling of data (in terms of methodology and format) as well as physical representation in XBRL syntax, the EIOPA followed the approaches applied for various

[1] https://specifications.xbrl.org/specifications.html

deliverables of the Eurofiling project[2]. In particular, EIOPA applied the Data Point Modelling methodology and the Data Point Model format to describe the exchanged data[3].

The mapping of this DPM to an XBRL taxonomy follows the general architectural approach as published on the EIOPA website[4], an approach shared also with the EBA taxonomies[5] and similar solutions developed by various NCAs.

III Relation to the data model

Before the development of an XBRL taxonomy (which is a technical format used for data exchange), information requirements need to be identified by specifying reportable pieces of information. This is usually done in the form of data models, which aim to organise the information for communication purposes (e.g. between business and IT experts, or between various groups of business experts).

In case of EIOPA reporting, the inputs for creation of the data model are the Implementing Technical Standards [ITS] and Guidelines, consisting of the main provisions covering the information requirements - the templates, i.e. tabular representation of information requirements, the instructions associated with these templates (LOG), and the related validation formulae.

Templates, provisions, instructions, and underlying regulations are analysed according to the DPM methodology in order to create a DPM model.

The EIOPA DPM consists primarily of three Microsoft Excel workbooks:

– Dictionary - defining properties (and their classifications/breakdowns) that can be used to describe each exchanged piece of information,

---

[2] Eurofiling is an open joint initiative in collaboration with the EBA, EIOPA, and XBRL Europe, as well as stakeholders like central and commercial banks, supervisors over banking systems, data exchange solutions providers, and others. Deliverables of the Eurofiling project can be found on https://www.eurofiling.info.

[3] Description of the DPM meta model, as well as EIOPA DPM, can be found in the accompanying file "EIOPA DPM Documentation.pdf"

[4] https://www.eiopa.europa.eu/tools-and-data/supervisory-reporting-dpm-and-xbrl_en

[5] https://www.eba.europa.eu/risk-and-data-analysis/reporting/reporting-frameworks

- Annotated templates - tables where each row/column/sheet is associated with a property or a set of properties defined in the dictionary,
- Validations - a list of quality checks to be performed on data, defined in a form-centric manner (i.e. based on templates' row/column/sheet codes).

The XBRL taxonomy is output created from the single data modelling platform, where EIOPA defines and manages its DPM models starting from version 2.8.0. The process is fully automated, and the XBRL taxonomy itself is just one of the formats obtained in this way, alongside the DPM Database or various Excel files (including DPM Dictionary and Annotated Templates).

IV Relation to XBRL specifications

EIOPA taxonomies are compliant with the XBRL 2.1 specification as of December 31, 2003, with Errata Corrections up to February 20, 2013, and the Dimensions 1.0 specification as of September 18, 2006 with errata corrections up to January 25, 2012.

The business rules layer in the form of linkbase files is defined according to the XBRL Formula specification 1.0 (2009 - 2016) and supporting specifications (Registry – 2009 - 2013, Generic Links - June 22, 2009). Assertion test may also use XPath/XQuery and XBRL Functions.

Rendering of tables is created according to the Table Linkbase 1.0 specification from March 18, 2014.

For enumerated metrics' dropdowns, the taxonomy utilises the Extensible Enumerations 1.0 specification from October 29, 2014.

For clarity of this document, XBRL technical constructs referenced in various sections are identified by their qualified names [QNames]. Prefixes applied in these QNames to abbreviate the namespaces follow the canonical namespace prefixes as presented in *Table 1*.

Table 1. Prefixes and namespaces of the XBRL technical files referenced in this document.

Prefix	Namespace
df	http://xbrl.org/2008/filter/dimension
enum	http://xbrl.org/2014/extensible-enumerations
gen	http://xbrl.org/2008/generic
iso4217	http://www.xbrl.org/2003/iso4217
label	http://xbrl.org/2008/label

link	http://www.xbrl.org/2003/linkbase
nonnum	http://www.xbrl.org/dtr/type/non-numeric
num	http://www.xbrl.org/dtr/type/numeric
sev	http://xbrl.org/2016/assertion-severity
table	http://xbrl.org/2014/table
tp	http://xbrl.org/2016/taxonomy-package
variable	http://xbrl.org/2008/variable
xbrldi	http://xbrl.org/2006/xbrldi
xbrldt	http://xbrl.org/2005/xbrldt
xbrli	http://www.xbrl.org/2003/instance
xlink	http://www.w3.org/1999/xlink
xs	http://www.w3.org/2001/XMLSchema
xfi	http://www.xbrl.org/2008/function/instance
xff	http://www.xbrl.org/2010/function/formula

V Publication and distribution

In the past, all taxonomy files were stored in their official location in subfolders of http://eiopa.europa.eu/eu/xbrl/. However, users are requested to stop using xBRL taxonomy canonical files by 31 December 2025.

For convenience, the taxonomy is distributed as a package according to the Taxonomy Packages 1.0 specification (as of April 19, 2016). This allows users to quickly identify relevant entry points and enables software to automatically configure the necessary remappings.

Starting from version 2.3, XBRL specifications and registries, as well as shared files defined by the Eurofiling project, are not included in the official EIOPA XBRL taxonomy package and may be downloaded from the XBRL taxonomies registry[6].

Figure 1 presents the folder structure of the EIOPA XBRL taxonomy package.

[6] https://taxonomies.xbrl.org/

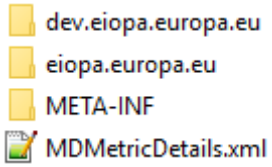


Figure 1. Folders of the taxonomy package.

Additionally, starting from version 2.4.0, EIOPA publishes a taxonomy package containing all XBRL specification, registry, and taxonomy files as well as the referenced Eurofiling files. This enables loading of the taxonomy without accessing any remote files. It is important to note that this package is valid at the moment of publication of the taxonomy and will not be maintained for any changes in the referenced files.

In a production environment, by normal XBRL convention, these files would likely not be used, and arrangements should be made to utilise, at least notionally, the official copies of the files from their official locations, which shall be made available by EIOPA for each final release after its publication.

VI Supporting concepts

This chapter describes some concepts to facilitate the definition of the mapping rules between the Data Point Model and XBRL taxonomies.

VI.1 Model supporting schema and other technical files

The XBRL representation of the model makes use of some schema definitions in the namespace *http://www.eurofiling.info/xbrl/ext/model*. The official location of this schema file is *http://www.eurofiling.info/eu/fr/xbrl/ext/model.xsd*[7]. Throughout this document, the prefix *model* will be used to refer to this schema namespace (see *Table 2*).

EIOPA extends the Eurofiling model schema with a few additional concepts hosted in *http://eiopa.europa.eu/xbrl/ext/model* namespace (official location *http://eiopa.europa.eu/eu/xbrl/ext/model.xsd*). These are, for example, definitions of dimensional constructs and linkbase placeholders to increase validation of reports for superfluous, unwanted content (in particular to prevent default use of metrics (i.e. when

[7] It can also be accessed through the XBRL Taxonomies Registry: https://taxonomies.xbrl.org/taxonomy/165

not explicitly allowed) and block scenario and segment for filing indicators) and custom roles applied in the EIOPA XBRL taxonomy.

Apart from *model.xsd* schema, *http://www.eurofiling.info/eu/fr/xbrl/ext* folder also includes other technical files explained in the next sections of this document. One of these files is *filing-indicators.xsd* schema (see – *I.1.1 Filing indicators*), where filing indicators are assigned with an empty hypercube to block the use of *xbrli:segment* and *xbrli:scenario* in the context they refer to and assertions ensuring that filing indicators are declared in the report and they are used in the required tuple structure. Starting with the 2.8.0 release, EIOPA is not extending this schema anymore and relies completely on the functions defined in the Eurofiling *filing-indicators.xsd* schema.

Another construct defined in the referenced *http://www.eurofiling.info/eu/fr/xbrl/ext* folder is a pivot variable declared in *pivot-variable.xml* that supports the definition of existence checks (see VII.3.6.5 Existence assertions).

A set of applied XBRL custom functions' definitions (for example, *http://www.eurofiling.info/eu/fr/xbrl/func/interval-arithmetics.xml, https://www.xbrl.org/taxonomy/int/lei/2020-07-02/lei-formula-functions.xsd,* http://www.eurofiling.info/eu/fr/xbrl/func/func.xsd*)* applied to every module (see VII.3.4). XBRL Formula assertions may also use constructs defined in linkbases placed in *http://www.eurofiling.info/xbrl/eu/fr/val* folder.

Table 2. Prefixes and namespaces of the model supporting schema and other technical files used in this document.

Prefix	Namespace
model	http://www.eurofiling.info/xbrl/ext/model
find	http://www.eurofiling.info/xbrl/ext/filing-indicators
pvar	http://www.eurofiling.info/xbrl/ext/pivot-variable
iaf	http://www.eurofiling.info/xbrl/func/interval-arithmetics
ifn	http://www.eurofiling.info/xbrl/func/isin
cmf	http://www.eurofiling.info/xbrl/func/custom-margin
efn	http://www.eurofiling.info/xbrl/func/extra-functions
mfn	http://www.eurofiling.info/xbrl/func/math
lei-fn	http://www.xbrl.org/taxonomy/int/lei/2020-07-02/functions

Schema and linkbase files described in this section are imported or referred from/to various EIOPA XBRL taxonomy files.

VI.2 Public elements

Public elements are all concepts of the model that are identified by a code in a certain scope and may include some additional information, such as readable labels, definitions, and legal references in different languages.

Declarations of public elements may include also attributes supporting management of the model. These are *model:creationDate* that reflects the date when the element was defined and *model:modificationDate* with the date when it was last modified (following the values assigned to these properties in the DPM dictionary).

VI.2.1 Standard labels

Language-specific information of public elements is represented using the following label resources:

- XBRL 2.1 labels (*link:label*) for *xbrli:item*s (or derived) public elements,
- generic labels (*label:label*) for public elements represented as XLink resources or other constructs (e.g. *link:roleType*s).

In general, the default (standard) role (*http://www.xbrl.org/2003/role/link*) is used for the extended links containing the label resources; however some specific labels may also be assigned in different extended link roles (e.g. domain member labels specific to hierarchies as explained in *VII.2.4 Explicit domain members and their relationships*).

The role types used as roles for generic and standard label resources are presented in *Table 3*.

Table 3. Role types used as roles for generic and standard label resources.

Property	Generic label role	Standard label role
Standard name	http://www.xbrl.org/2008/role/label	http://www.xbrl.org/2003/role/label
Definition	http://www.xbrl.org/2008/role/verboseLabel	http://www.xbrl.org/2003/role/verboseLabel
Legal references[8]	http://www.xbrl.org/2008/role/documentation	http://www.xbrl.org/2003/role/documentation

The labels for the concepts of a schema or a linkbase file are placed in a separate label linkbase file for each distinct language, located in the same folder as its corresponding schema or linkbase file. The naming convention for these label linkbase files is: *{base file name}-lab-{lang}.xml,* where *{base file name}* is the name of the schema or linkbase file where the concept is defined (without extension) and the *{lang}* component is the ISO 639-1 code of the language (lowercase). The primary language for the EIOPA XBRL taxonomy is English (ISO 639-1 code "en").

VI.2.2 Specific labels

In addition, some concepts may contain a special linkbase to represent specific labels needed for different purposes (e.g. codes to be used as filing indicators' values). The names of these linkbase files are constructed as follows: *{base file name}-lab-{lang}-codes.xml* or *{base file name}-lab-codes.xml*

The labels for these codes are represented as resources with a custom role. In particular, the role defined in the Eurofiling *model.xsd* schema for resources representing codes for filing indicators is *http://www.eurofiling.info/xbrl/role/filing-indicator-code,* while the role for resources representing the table row/column/sheet codes is *http://www.eurofiling.info/xbrl/role/rc-code*.

EIOPA *model.xsd* schema defines *http://eiopa.europa.eu/eu/xbrl/role/key-column-type* role that serves the purpose of providing hints of the type of key columns in open tables (following the description in the DPM Annotated Templates). Starting from version 2.8 of the EIOPA Taxonomies, the above information is reflected using

[8] Current references are described in plain English; as a consequence, labels are a better solution than reference linkbases. In the future, a structured approach for legal references could be undertaken.

http://www.xbrl.org/2008/role/verboseLabel role applied for labels for the Table linkbase resources.

Hierarchy nodes' specific labels are defined in the hierarchy extended link role in a separate file for each domain.

Extensions might use the same mechanism to add their own application-specific codifications using different roles.

VII Logical taxonomy architecture

This section describes in detail the components and content of the taxonomy. The diagram provided in *Annex 1. EIOPA XBRL Taxonomy: Owners, Folders, Files, Namespaces and Prefixes* may be helpful for the comprehension of this section.

VII.1 Owners

The owner represents a location and a namespace in which a set of related concepts is defined. The owner is closely related to the idea of extensibility in XBRL. The main properties of the owner are:

- namespace (*{ons}*),
- prefix (*{opre}*), and
- official location (*{oloc}*).

The owner's namespace is a URI used to define the namespace used by the concepts.

The prefixes associated with the namespaces in the taxonomy's files and the associated documentation are called the "canonical prefixes". Items of the DPM and the taxonomy are referenced by their QName, using their canonical prefix.

The official location is a URL used to specify the location where taxonomy files associated with that owner are to be published. Different owners must have different official locations, even if owners share a single internet domain. The official location of the taxonomy should be built from the internet domain of the institution, plus a component representing the geographical area covered by the institution (as *eu* for EIOPA artefacts), followed by the identification of the type of standard used to express information requirements (e.g. *xbrl*).

Examples of owner namespaces and locations are presented in *Table 4*.

Table 4. Examples of owner namespaces and locations.

Owner	Namespace	Official location
Eurofiling	http://www.eurofiling.info/xbrl	http://www.eurofiling.info/eu/fr/xbrl
EIOPA	http://eiopa.europa.eu/xbrl	http://eiopa.europa.eu/eu/xbrl/
NCA's extension	http://www.nca.xx//xx/xbrl	http://{internet domain name}/{country code}/xbrl

In addition to the above, EIOPA utilises the notion of owners to express the groups of concepts defined in the model. According to the underlying DPM, the EIOPA model is defined in two versions: highly dimensional (HD) and moderately dimensional (MD). In general, the difference between the two is the definition of metrics that in the HD version represent very basic data types, while in the MD version includes additional dimensional information. Other dimensional properties are shared (reused in both versions). To resemble this situation on the technical level, EIOPA defines three owners:

- Solvency 2 common (s2c),
- Solvency 2 highly dimensional (s2hd),
- Solvency 2 moderately dimensional (s2md).

Solvency 2 common items (s2c) are all dimensions, domains, and members defined in the DPM dictionary.

Furthermore, s2hd and s2md define DPM dictionary metrics for each version and may resemble information requirements in form of frameworks (see the next section of this document: *VII.3 Information requirements layer*). It is important to note that the **EIOPA information requirements are defined in the XBRL taxonomy only in the MD approach**.

Table 5 contains owners (location, namespace, and prefixes) used in the EIOPA XBRL taxonomy.

Table 5. EIOPA XBRL taxonomy owners, namespaces, and prefixes.

Owner official location	Owner namespace	Owner prefix
http://eiopa.europa.eu/eu/xbrl/s2c	http://eiopa.europa.eu/xbrl/s2c	s2c
http://eiopa.europa.eu/eu/xbrl/s2hd	http://eiopa.europa.eu/xbrl/s2hd	s2hd
http://eiopa.europa.eu/eu/xbrl/s2md	http://eiopa.europa.eu/xbrl/s2md	s2md

VII.2 Dictionary layer

The dictionary layer contains the definition of business properties identified in the DPM Dictionary. The properties can subsequently be used in the identification of currently requested information requirements.

VII.2.1 Core concepts

The core concepts of the dictionary are metrics, dimensions, domains, domain members, and hierarchies. All of these concepts are public elements (see *VI.2 Public elements*).

To cope with the management of the dictionary, all core concepts include two optional attributes that establish the currency period: the starting date of the period interval (*model:fromDate* attribute) and the end date (*model:toDate* attribute). If the *model:fromDate* attribute is not included, then the concept is assumed to be valid for any period prior to the *model:toDate* attribute. If the *model:toDate* attribute is not included, then the concept is assumed to be valid for any period after the *model:fromDate* attribute. If neither *model:fromDate* nor *model:toDate* attributes are included, then the concept is assumed to be current for any period of time. These attributes do not have any impact on the reporting process itself - they are meant to support the management of the concepts of the dictionary.

The core concepts are never deleted[9]. As a result the dictionary will grow in time as the new concepts are added and the obsolete are disabled using the attribute defined in the previous paragraph.

All files in the dictionary of concepts are placed under the folder *dict* in the official location *{oloc}* of its owner (see *Annex 1. EIOPA XBRL Taxonomy: Owners, Folders, Files, Namespaces and Prefixes*). Its namespace is obtained by adding a suffix that depends on the type of element to the namespace of the owner *{ons}*. The prefix to represent that namespace is obtained by adding a predefined suffix to the prefix of its owner *{opre}* (as presented in *Table 6*) where *{oloc}*, *{ons}* and *{opre}* are defined as in VII.1 Owners, and *{dc}/{DC}* is the code of a domain in lower and upper case respectively.

[9] However, concepts that are used in production reporting may be deleted.

Table 6. Pattern for location, target namespace, and its canonical prefix for dictionary concepts.

Dictionary concept	Official location	Target namespace	Namespace prefix
Metrics	{oloc}/dict/met/met.xsd	{ons}/dict/met	{opre}_met
Dimensions	{oloc}/dict/dim/dim.xsd	{ons}/dict/dim	{opre}_dim
Explicit domains	{oloc}/dict/dom/exp.xsd	{ons}/dict/exp	{opre}_exp
Typed domains	{oloc}/dict/dom/typ.xsd	{ons}/dict/typ	{opre}_typ
Explicit domain members	{oloc}/dict/dom/{dc}/mem.xsd	{ons}/dict/dom/{DC}	{opre}_{DC}

Examples of location, target namespace, and its prefix for dictionary concepts are presented in *Table 7*.

Table 7. Examples of location, target namespace, and its prefix for dictionary concepts.

Dictionary concept	Prefix	Target namespace	Official location
Common dimensions	s2c_dim	http://eiopa.europa.eu/xbrl/s2c/dict/dim	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dim/dim.xsd
Common explicit domains	s2c_exp	http://eiopa.europa.eu/xbrl/s2c/dict//exp	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/exp.xsd
Common typed domains	s2c_typ	http://eiopa.europa.eu/xbrl/s2c/dict/typ	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/typ.xsd
Common explicit domain members example (domain CG)	s2c_CG	http://eiopa.europa.eu/xbrl/s2c/dict/dom/CG	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/cg/mem.xsd
MD version metrics	s2md_met	http://eiopa.europa.eu/xbrl/s2md/dict/met	http://eiopa.europa.eu/eu/xbrl/s2md/dict/met/met.xsd

VII.2.2 Metrics

In general, metrics define the nature of the measure to be performed by doing the following:

- indicating the data type, i.e. the expected type of value that should be reported for a data point,
- determining the period type, i.e. whether a fact corresponding to a data point is reported for a single date (instant) or a period of time (duration),
- expressing certain semantics.

There is a different treatment of metrics between HD and MD (for more information, see the associated *EIOPA DPM Documentation* and section *IX Mapping between MD and HD properties*). Neither version applies period type differentiation of metrics - in both

versions, period type is set to instant[10] (in some cases, the duration of a data point may be expressed using certain dimensional properties).

Technically, metrics are represented in the taxonomy as XBRL primary items and defined in schema files named *met.xsd* (in *{oloc}/dict/met/* folder location) that reference label linkbase file *met-lab-{lang}.xml* (providing human-readable labels as defined in the DPM; for representation in syntax see *VI.2.1 Standard labels*) and definition linkbase file *met-def.xml* (defining XBRL Dimensions relationships that constrain the use of metrics in reports[11]).

The code (*{name}*) for each metric is composed of three components:

- a letter that represents the data type in lowercase (for available options, see *Table 8* below),
- a letter that represents the period type characteristics (*i* for instant and *d* for duration, which, as explained above, is always *i* in the EIOPA taxonomy),
- a number that corresponds to the numeric code in the model (no zero padding or predetermined length).

[10] This approach has been introduced in order to overcome the difficulty of defining time constraints for multiple periods in the table, definition and XBRL Formula specification based linkbases.

[11] In order to prevent from unrequested content in filings, all metrics are prohibited from being reported (in the dictionary) unless they are subsequently used in hypercubes of tables referenced from a module (see next sections of this document).

Table 8. Model and XML data type for metrics, local name codification letter and reporting unit.

Model data type	**XML data type**	**Local name codification letter**	**Reporting unit**
Monetary (currency)	xbrli:monetaryItemType	m	adequate currency using ISO 4217 codification (e.g.: iso4217:EUR)
Percent or Ratio	num:percentItemType	p	xbrli:pure
Decimal[12]	xbrli:decimalItemType	r	xbrli:pure
Integer	xbrli:integerItemType	i	xbrli:pure
Text	xbrli:stringItemType	s	not applicable
Date	xbrli:dateItemType	d	not applicable
Boolean	xbrli:booleanItemType	b	not applicable
True	restriction of xbrli:booleanItemType to "true"	t	not applicable
URI	xbrli:anyURIItemType	u	not applicable
Explicit domain	enum:enumerationItemType	e	not applicable
Typed domain	typed domain corresponding data type, e.g. xbrli:stringItemType if a typed domain is xs:string	depending on typed domain type	depending on typed domain type, usually xbrli:pure, if numeric

For domain-based data types, additional attributes (as defined in the Extensible Enumerations specification) are included on declaration of metrics to identify the allowable members.

The *id* of the metric element (necessary for XLink locators) is composed as: *{opre}_{name}.*

Table 9 contains a few examples of metrics declared in the taxonomy.

[12] There are a few cases where decimal metrics' codes start with letter *p* rather than *r*. They were used in preparatory phase reporting where the naming codification was different (both percent/ratio and other decimal items were using *p* code letter).

Table 9. Examples of metrics.

Owner	Data type	Code	Name	Id	Namespace	Prefix
MD	Decimal	17	pi17	s2md_pi17	http://eiopa.europa.eu/xbrl/s2md/dict/met	s2md_met
MD	Integer	31	ii31	s2md_ii31	http://eiopa.europa.eu/xbrl/s2md/dict/met	s2md_met
MD	Monetary	43	mi43	s2md_mi43	http://eiopa.europa.eu/xbrl/s2md/dict/met	s2md_met
HD	Explicit domain	17	ei17	s2hd_ei17	http://eiopa.europa.eu/xbrl/s2hd/dict/met	s2hd_met

Although currently not used by the EIOPA taxonomy, metrics (similarly to domain members, as explained in the next section) can be arranged in hierarchies.

VII.2.3 Domains

Explicit domains are represented using XBRL abstract items of domain type *model:explicitDomainType* in the schema file *exp.xsd* with namespace *{ons}/dict/exp* and prefix *{opre}_exp*.

Typed domains are represented as XML elements that are not in the substitution group of *xbrli:item*. These elements are defined in the schema file *typ.xsd* with namespace *{ons}/dict/typ* and prefix *{opre}_typ*.

Both schema files are placed in *{oloc}/dict/dom/* folder location.

The code (*{name}*) of each domain corresponds to its code in the model (which is a short sequence of uppercase letters, usually two).

Value of the *id* attribute of a domain (necessary for XLink locators) is composed according to the following pattern: *{opre}_{name}.*

A few examples of domain items are presented in *Table 10*.

Table 10. Examples of domain items.

Owner	Code	Element Name	Type	Id	Namespace	Prefix
Common	BC	BC	Explicit	s2c_BC	http://eiopa.europa.eu/xbrl/s2c/dict/exp	s2c_exp
	ID	ID	Typed	s2c_ID	http://eiopa.europa.eu/xbrl/s2c/dict/typ	s2c_typ

Domain schema files reference label linkbase files[13] *exp-lab-{lang}.xml* and *typ-lab-{lang}.xml* (providing human readable labels as defined in the DPM; for representation in syntax see *VI.2.1 Standard labels*).

VII.2.4 Explicit domain members and their relationships

Explicit domain members are represented using XBRL abstract items of domain item type, as defined in the non-numeric set of types of the XBRL International type registry: *nonnum:domainItemType*.

The code (*{name}*) of each explicit domain member corresponds to its code assigned in the DPM Dictionary which in general it starts with a lowercase letter *x* (due to XML naming restrictions disallowing digit as the starting character) followed by a sequential number. However, if the concept represented already has a widely accepted standard codification like ISO codes or NACE code list, the name will aim to match the existing codification, for example:

- ISO 4217: standard currency codes composed of three alphabetical characters,
- ISO 3166-1 alpha-2: standard country codes composed of two alphabetical characters,
- ISO 639-1: standard language codes composed of two alphabetical characters,
- NACE 2.1 codes: http://ec.europa.eu/competition/mergers/cases/index/nace_all.html (with dots).

The default member of a domain (usually, but not necessarily, the one with code *x0*) is marked with an attribute: *model:isDefaultMember="true"*.

The value of the *id* attribute of explicit domain members follows the general rule: *{opre}_{name}*.

[13] Explicit domains are of *xbrli:item* substitution group whereas typed domains are not. Because of this, labels for the former ones are defined using standard label links and labels for the latter using generic label links.

The schema file that represents explicit members is placed in a folder with the name of its corresponding domain according to the following pattern: *{oloc}/dict/dom/{dc}* where *{dc}* is domain code in lowercase. The schema file for explicit domain members is called *mem.xsd*, its namespace is constructed based on the following pattern: *{ons}/dict/dom/{DC}* while prefix consist of *{opre}_{DC}* where *{DC}* is domain code in the uppercase. Examples of schema files defining explicit domain members are presented *Table 11*.

Table 11. Examples of schema files defining explicit domain members.

Owner	Domain code	Domain members schema	Namespace	Prefix
Common	CM	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/cm/mem.xsd	http://eiopa.europa.eu/xbrl/s2c/dict/dom/CM	s2c_CM
Common	GA	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/ga/mem.xsd	http://eiopa.europa.eu/xbrl/s2c/dict/dom/GA	s2c_GA

This schema file references linkbases defining labels (*mem-lab-{lang}.xml*) for domain members (according to the DPM dictionary) and a definition linkbase file (*mem-def.xml*) where all members are connected to the domain item using *domain-member arcrole* of XBRL Dimensions.

Relationships of domain members defined in the DPM dictionary are represented using XBRL extended link roles whose role type URI is built according to the following pattern: *{ons}/role/dict/dom/{DC}/{relationship code}* where *{DC}* represents the code of the domain in uppercase and *{relationship code}* is the numeric code of the hierarchy. The value of the *id* attribute of these roles is composed following the pattern: *{opre}_{relationship code}*. Examples of extended link roles used for hierarchies of domain members are presented in *Table 12*.

Table 12. Extended link roles used for domain member relationships.

Owner	Domain code	Relationship code	Relationship role URI	Role id
Common	CM	1	http://eiopa.europa.eu/xbrl/s2c/role/dict/dom/CM/1	s2c_1
Common	GA	4	http://eiopa.europa.eu/xbrl/s2c/role/dict/dom/GA/4	s2c_4

The schema file that represents relationships (defining role types and referring to linkbases) is placed in the same folder as the schema defining members and it is named *hier.xsd*. Its namespace is constructed based on the following pattern: *{ons}/dict/dom/{DC}/hier,* while prefix consist of *{opre}_{DC}_h* where *{DC}* is domain code in the uppercase. Examples of such schema files, their namespaces and prefixes are presented in *Table 13*.

Table 13. Examples of schema files defining extended links for the relationships domain members.

Owner	Domain code	Relationships schema	Namespace	Prefix
Common	CM	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/cm/hier.xsd	http://eiopa.europa.eu/xbrl/s2c/dict/dom/CM/hier	s2c_CM_h
Common	GA	http://eiopa.europa.eu/eu/xbrl/s2c/dict/dom/ga/hier.xsd	http://eiopa.europa.eu/xbrl/s2c/dict/dom/GA/hier	s2c_GA_h

These schema files refer to a linkbase file containing hierarchy specific labels for members (*hier-lab-mem-{lang}.xml*) and three linkbase files with information about relationships between members:

- a presentation linkbase (*hier-pre.xml*), which represents the hierarchical disposition of members using XBRL 2.1 *parent-child* relationships,
- a definition linkbase (*hier-def.xml*), which enables the inclusion of the members of a hierarchy in dimensional combinations or applying them as enumerations for metrics (using *domain-member* relationships of XBRL Dimensions 1.0, and taking into account the *xbrldt:usable* attribute to identify "grouping" members),
- a calculation linkbase (*hier-cal.xml*), which establishes some basic arithmetical relationships between a member of the hierarchy and its children:
    - a member is equal to the addition of its child members in the hierarchy: *complete-breakdown* relationships,
    - a member is greater than or equal (in absolute value) to the addition of its child members in the hierarchy: *partial-breakdown* relationships,
    - a member is less than or equal (in absolute value) to the addition of its child members in the hierarchy: *superset-breakdown* relationships.

These calculation arcs include a *weight* attribute to indicate whether the child member contributes to the aggregation positively (*+1*) or negatively (*-1*)[14]. The roles representing these calculation relationships are defined in the Eurofiling *model.xsd* schema, and are presented in *Table 14*.

[14] Other decimal values are also allowed but currently not used.

Table 14. Arc roles defined in the Eurofiling model.xsd schema, reflecting different forms of aggregations of members.

Arc role id	Arc role URI
complete-breakdown	http://www.eurofiling.info/xbrl/arcrole/complete-breakdown
partial-breakdown	http://www.eurofiling.info/xbrl/arcrole/partial-breakdown
superset-breakdown	http://www.eurofiling.info/xbrl/arcrole/superset-breakdown

The root member of the definition and presentation relationship networks is the domain item, as defined in the *exp.xsd* schema associated with the owner.

Some hierarchies of members are used to constrain the values of metrics with means of XBRL Extensible Enumerations specification. In this case, the labels applicable to members in a particular enumeration may differ from the standard labels of these members. This requirement is addressed by defining member labels (using a standard generic label role) in an extended link role specific to a hierarchy. Examples of such cases are provided in *Table 15*.

Table 15. Examples of hierarchy-specific labels.

Member QName	Standard label	Hierarchy specific label ELR	Hierarchy specific label
s2c_CN:x1	Reported	http://eiopa.europa.eu/xbrl/s2c/role/dict/dom/CN/20	1 - Reported
s2c_CN:x20	Not reported as no off-balance sheet items		2 - Not reported as no off-balance sheet items
s2c_CN:x2	Not reported other reason		0 - Not reported other reason (in this case special justification is needed)

Domain members that extend the domain of another owner are placed in a folder preceded by the prefix of the extended owner. For instance, in the case of extensions of domains of the EIOPA by an NCA, the domain code would be preceded by the prefix *s2c,* for example, *http://www.nca.gov/{country code}/xbrl/dict/dom/s2c_ga/mem.xsd* for file location, *http://www.nca.gov/xbrl/dict/dom/s2c_GA* for namespace, and *nca_s2c_GA* for prefix.

VII.2.5 Typed domain values

Values of typed domains are neither listed as XBRL items with labels nor arranged in hierarchies. The content of typed domains is restricted by XML data type constraints (as these domains, according to the XBRL Dimensions specification, are XML constructs).

In most cases, a typed domain would be represented by an XML element with a simple data type (e.g. *xs:string* or *xs:decimal*), though further restrictions are technically

possible (also with means of business rules defined according to XBRL Formula specification).

Typed domains may be *nillable="true",* which means that they can be reported as *xsi:nil="true"* (and no value). This construct is used in the reporting of optional open table columns modelled as typed dimensions.

VII.2.6 Dimensions

The representation of dimension items in XBRL is defined in the XBRL Dimensions 1.0 specification.

The schema file defining dimension items is placed in *{oloc}/dict/dim/* folder and named *dim.xsd* with namespace *{ons}/dict/dim* and *{opre}_dim* prefix.

The local name (*{name}*) of each dimension corresponds to its code in the model: a short sequence of uppercase letters (usually two).

The value of the *id* attribute of the element representing a dimension item (necessary for XLink locators) is composed according to the following pattern: *{opre}_{name}*.

A few examples of dimension items declarations are presented in *Table 16*.

Table 16. Examples of dimension items.

Owner	Code	Name	Id	Namespace	Prefix
Common	VL	VL	s2c_VL	http://eiopa.europa.eu/xbrl/s2c/dict/dim	s2c_dim
Common	VG	VG	s2c_CG	http://eiopa.europa.eu/xbrl/s2c/dict/dim	s2c_dim

The schema file defining dimensions includes references to the label linkbase file *dim-lab-{lang}.xml* and a definition linkbase named *dim-def.xml* (placed in the same folder as the schema file).

This definition linkbase includes the following information about explicit dimensions:

- reference to the domain associated with the dimension by means of a *dimension-domain* relationship (with an *xbrldt:usable* attribute equal to *"false"*) pointing to a domain item defined in either the *exp.xsd* or *typ.xsd* schema file of any referenced or defined owner,

– reference to the default member of that dimension by means of *a dimension-default* relationship[15].

These relationships associating a dimension with a domain and its default members are defined in the standard extended link role (*http://www.xbrl.org/2003/role/link*).

VII.3 Information requirements layer

Frameworks, taxonomies, tables, modules, validations, and other concepts constitute the layer of the model where actual information requirements are specified with the support of the concepts defined in the dictionary.

All of the files that correspond to this layer are placed under the folder *fws* in the official location of its owner (i.e. *{ons}/fws/*). Its namespace is obtained by adding the suffix *fws* to the base namespace of the owner, plus some additional suffixes that depend on the type of concept represented.

In the EIOPA XBRL taxonomy, frameworks are defined for the MD modelling approach.

VII.3.1 Frameworks

Frameworks are public elements represented using XBRL abstract items of the framework type (*model:frameworkType*) in the schema file *fws.xsd*. General framework properties are presented in *Table 17*.

Table 17. Framework properties.

Schema property	Value
Official location	{oloc}/fws/fws.xsd
Target namespace	{ons}/fws
Target namespace prefix[16]	{opre}_fws
Element local name	{framework}

[15] Note that although the model defines default members at the domain level, the XBRL Dimensions specification establishes this relationship at dimension level; thus, each dimension using a domain with a default member must include this relationship.

[16] Target namespace prefixes are not strictly necessary. Moreover, schemas like frameworks, taxonomies, table groups and tables define names that are not used in the exchange of information between supervisors and supervised entities. However, as

Element id	{opre}_{framework}

The local name of each framework element corresponds to its code in the model (*{name}*) and its *id* follows a general pattern (*{opre}_{name}*). Each framework has a folder in which the taxonomies are placed. Examples of frameworks are presented in *Table 18*.

Table 18. Examples of frameworks.

Owner	Schema property	Value
MD version	Official location	http://eiopa.europa.eu/eu/xbrl/s2md/fws/fws.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/s2md/fws
	Target namespace prefix	s2md_fws
	Local name example	solvency
	Element id example	s2md_solvency
	Element label (English)	Solvency II (MD version)
	Framework folder	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/

VII.3.2 Taxonomies

Taxonomies are public elements represented using XBRL abstract items of the taxonomy type (*model:taxonomyType*). These elements are stored in the schema file *tax.xsd* under the folder of its framework, a subfolder that corresponds to its normative code or publication date of the legislation (*{normative code}*), and another subfolder with the publication date[17] of this version of the taxonomy (*{taxonomy publication date}*).

Thus, the taxonomy schema file includes a single element. Its local name corresponds to its code (*{name}*) in the model, and the value of its *id* attribute is constructed according to the general pattern (*{opre}_{name}*). General taxonomy properties are presented in *Table 19*.

some XBRL tools raise warnings whenever they find a schema with no prefix defined, prefixes have been included to avoid misleading the users of these tools.

[17] Using the ISO 8601 codification.

Table 19. Taxonomy properties.

Schema property	Value
Official location	{oloc}/fws/{framework}/{normative code }/{taxonomy publication date}/tax.xsd
Target namespace	{ons}/fws/{framework}/{normative code}/{{taxonomy publication date}
Target namespace prefix	{opre}_tax
Element local name	{taxonomy}
Element id	{opre}_{taxonomy}

To facilitate the specification of additional taxonomy resources in this document, the following abbreviations will be applied from here onwards:

- *{taxonomy-loc}* represents the URL *{oloc}/fws/{framework}/{normative code}/{taxonomy release date}*,
- *{taxonomy-ns}* represents the URI *{ons}/fws/{framework}/{normative code}/{taxonomy release date}*.

Examples of taxonomy folders, items, namespaces and prefixes are presented in *Table 20*.

Table 20. Examples of taxonomies.

Owner	Schema property	Value
Solvency II MD version	Official location	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/tax.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15
	Target namespace prefix	s2md_tax
	Local name example	solvency2
	Element id example	s2md_solvency2
	Element label (English)	Solvency II, 2016-07-15
	Taxonomy folder	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/

The taxonomy folder may include subfolders for:

- tables (*tab*),
- modules (*mod*) and
- validations (*val*).

VII.3.3 Tables

The table folder includes a schema file (*tab.xsd*). The schema file contains the definitions of table groups (e.g. template variants), which are represented using XBRL abstract items of the table group type (*model:tableGroupType*). The name ({name}) of a table group item composed by adding the prefix *tg* to the code (*{table group code}*) of a

table group in the model (e.g. *tgS.01.01.01*)". General properties of table groups are presented in *Table 21*.

Table 21. General table group properties.

Schema property	Value
Official location	{taxonomy-loc}/tab/tab.xsd
Target namespace	{taxonomy-ns}_tab
Target namespace prefix	{opre}_tab
Element local name	tg{table group code}
Element id	{opre}_{name}

Examples of table group definitions are presented in *Table 22*.

Table 22. Examples of table groups definitions.

Owner	Schema property	Value
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/tab/tab.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15_tab
	Target namespace prefix	s2md_tab
	Local name example	tgS.01.01.01
	Element id example	s2md_tgS.01.01.01
	Element label (English)	S.01.01.01 Appendix I: Quantitative reporting templates
	Tables folder	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/tab/

Table group schema file references a label linkbase for table groups (*tab-lab-{lang}.xml*).

In the same folder, there is also a linkbase file *tab-pre.xml* linking all table groups to tables that they gather using *group-table* relationship of the Eurofiling model schema. This linkbase is not linked from any of the taxonomy files in order to prevent the discovery of all tables from modules (that refer to table groups). It can, however, be used by tools to visualise the full scope of the taxonomy content from table groups and tables' perspective. The link between the table groups and individual tables is established in the linkbase files of modules (see *VII.3.4 Modules*).

The files that define the content of each table are placed in a folder whose name corresponds to the code of the table in the model (*{table code}*) in lowercase. Location and general properties of these files are presented in *Table 23*.

Table 23. General properties of table files.

Schema property	Value
Official location	{taxonomy-loc}/tab/{table code}/{table code}.xsd – table code in lowercase
Target namespace	{taxonomy-ns}/tab/{table code} – table code in uppercase
Target namespace prefix	{opre}_tab_{table code}
Element local name	N/A - elements defined as resources in linkbases
Element id	N/A, {opre}_{table code} is a table resource id in the table linkbase

Examples of table definitions are presented in *Table 24*.

Table 24. Examples of tables definitions.

Owner	Schema property	Value
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/tab/ s.01.01.01.01/s.01.01.01.01.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/tab/S.01.01.01.01
	Target namespace prefix	s2md_tab_S.01.01.01.01
	Local name example	N/A
	Element id example	s2md_tS.01.01.01.01 (table resource id in the table linkbase)
	Element label (English)	S.01.01.01.01
	Table folder	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/tab/ s.01.01.01.01/

A schema file for a table refers to:

a. a table linkbase (*{table code}-rend.xml*),

b. a definition linkbase (*{table code}-def.xml*),

c. a generic label linkbase with table headers text (*{table code}-lab-{lang}.xml*)[18],

d. a generic label linkbase with table header codes (*{table code}-lab-codes.xml*),

e. until version 2.7.0 (inclusive), an optional generic label linkbase file identifying types of key columns in case of open tables (*{table code}-lab-keys.xml*), from version 2.8.0 (inclusive) onwards, this information is included in the generic label linkbase file indicated in point c. as label resources with role *http://www.xbrl.org/2008/role/verboseLabel*.

The table linkbase (*a*) includes the definition of the table according to the Table Linkbase specification. The relationships of each table are placed in an extended link whose role is built according to the following pattern: *{ons}/role/fws/{framework}/{normative code}/{taxonomy publication date}/tab/{table code}* with id *role*. For example, table linkbase relationships for table S.01.01.01.01 are defined in extended link role

http://eiopa.europa.eu/xbrl/s2md/role/fws/solvency/solvency2/2016-07-15/tab/S.01.01.01.01.

In this linkbase, the different components of the tables are represented using resources. The value of the *id* attribute of these resources is based on the code or sequential number, plus a prefix to obtain a unique code in the context of the linkbase, as presented in the examples in *Table 25*.

Table 25. Pattern and examples of ids for table linkbase resources.

Resource	Id pattern	Example
table:table	{opre}_t{table code (uppercase)}	s2md_tS.01.01.01.01
table:breakdown (predefined or variable axis)	{opre}_a{sequential number}	s2md_a1
table:conceptRelationshipNode; table:dimensionRelationshipNode	{opre}_r{sequential number}	s2md_r6
top level abstract table:ruleNode	{opre}_a{sequential number}.root	s2md_a1.root
table:ruleNode	{opre}_c{sequential number}	s2md_c2
filer, e.g. df:explicitDimension	{opre}_a{sequential number}.root.filter	s2md_a3.root.filter

According to the Table Linkbase specification, aspect rules are used to specify the concepts represented in predefined axes.

Although not strictly requested by the Table Linkbase specification, *link:roleRef* is included in the table linkbase files, pointing to an extended link role when resources relate to domain member relationships defined in the dictionary.

The definition linkbase file (*b*) includes dimensional relationships valid in the context of the table. Valid combinations are defined using only positive (*all*) closed hypercubes obtained from the set of valid cells of the table, following an optimisation algorithm[19].

[19] It is important to remark that XBRL hypercubes in the definition linkbase of tables are validation artefacts and should not be used by external systems for the automatic creation of database structures. The hypercubes produced by the algorithm do not adhere to any business criteria. These hypercubes might be modified with the addition of new information to tables, with the only purpose of reducing the final set of hypercubes and performing more efficiently with XBRL market tools.

Each extended link role contains one or more metrics (primary items) and a single hypercube[20]. Where there are multiple metrics, the first one will be used to group the rest and reduce the number of all arcs. The domain element will be used as the target of *dimension-domain* arcs to avoid cycles. The *@xbrldt:targetRole* attribute might be necessary in the case of hypercubes with dimensions that share the same domain.

The roles of the extended links necessary to express these combinations are built by adding sequential numeric suffixes to the role previously defined for the table, i.e. *{ons}/role/fws/{framework}/{normative code}/{taxonomy publication date}/tab/{table code}/{sequential number}* (e.g. *http://eiopa.europa.eu/xbrl/s2md/role/fws/solvency/solvency2/2016-07-15/tab/S.01.01.01.01/1*). Role ids follow the pattern *role{sequential number},* e.g. *role1*.

The generic label linkbase file (*c*) for a table headers text contains labels for Table Linkbase nodes. In addition to the standard label representing table title, a *table:table* node also contains a verbose label concatenating both the table code and the title of a table.

Another (separate) generic label linkbase file (*d*) referenced from the table schema file contains codes. These are row/column/sheet/table codes (*http://www.eurofiling.info/xbrl/role/rc-code*) for table rule nodes (e.g. *C0010*, *R0070, S.01.01.01.01*) and filing indicator code (using http://www.eurofiling.info/xbrl/role/filing-indicator-code role) for table resource identifying a value to be included on a filing indicator when a template (which a table is part of) is reported or explicitly not reported (e.g. *S.02.01*, see section *VII.3.5)*.

Until version 2.7 (inclusive), open tables included (if applicable), optional generic label linkbase file (*e*) identifying key column types using *http://eiopa.europa.eu/eu/xbrl/role/key-column-type* role. Starting from version 2.8 (inclusive), this information is included in the generic label linkbase file indicated in point c. as label resources with role *http://www.xbrl.org/2008/role/verboseLabel*.

VII.3.3.1 Root hierarchy node in semi-open axes

EIOPA XBRL taxonomies include tables with semi-open axes. For these axes, the reporting entity must select a member of the hierarchy included in the axis. Usually, the

[20] The Eurofiling *model.xsd* schema includes a hypercube element to be used therefore it is no needed to define hypercube elements in each table or taxonomy.

hierarchies start with a default member (designated by the code *x0*), but this member is not eligible to be used by the undertaking.

An example of such a solution is table S.04.05.21.04 "Top 5 countries (by amount of gross premiums written): Life insurance and reinsurance obligations", from the Solvency II 2.10.0 XBRL taxonomy. This table includes a semi-open X axis "Countries", represented by the hierarchy with the code 43 of the domain Geographical Area (GA).

As presented in the *Figure 2*, this hierarchy starts with a root member *x0* "Not applicable/All geographical areas".

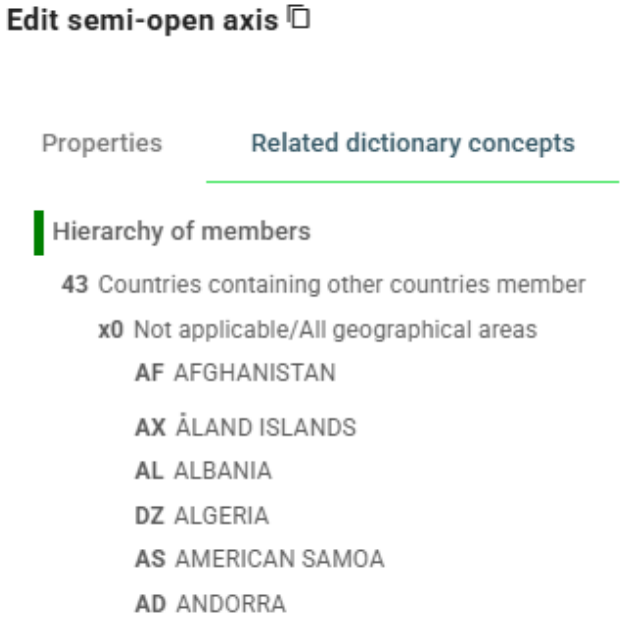


Figure 2. Structure of the GA_43 hierarchy.

This member however cannot be used by the undertaking, as it is specified in the rendering linkbase of this table (/eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2025-08-11/tab/s.04.05.21.04/s.04.05.21.04-rend.xml) where only the hierarchy members descendants of *x0* can be used. As presented in the snippet below.

```
<df:explicitDimension id="s2md_a2.root.filter" xlink:label="s2md_a2.root.filter" xlink:type="resource">
    <df:dimension>
        <df:qname>s2c_dim:LR</df:qname>
    </df:dimension>
    <df:member>
        <df:qname>s2c_GA:x0</df:qname>
        <df:linkrole>http://eiopa.europa.eu/xbrl/s2c/role/dict/dom/GA/43</df:linkrole>
        <df:arcrole>http://xbrl.org/int/dim/arcrole/domain-member</df:arcrole>
        <df:axis>descendant</df:axis>
```

```
        </df:member>
    </df:explicitDimension>
```

Another mechanism provided by XBRL is to restrict the usability of some members of the hierarchy from a dictionary point of view. Following the definition linkbase of the hierarchy used for the previous example, i.e. /eiopa.europa.eu/eu/xbrl/s2c/dict/dom/ga/hier-def.xml, the attribute "usable" is set to "false" for the arc between the member of the hierarchy (GA) and the root member *x0*. The snippet below represents this situation.

```
    <link:definitionLink xlink:role="http://eiopa.europa.eu/xbrl/s2c/role/dict/dom/GA/43"
xlink:type="extended">
        <link:loc xlink:href="../exp.xsd#s2c_GA" xlink:label="loc_s2c_GA" xlink:type="locator" />
        <link:loc xlink:href="mem.xsd#s2c_x0" xlink:label="loc_s2c_x0" xlink:type="locator" />
        <link:definitionArc xlink:type="arc" xlink:arcrole="http://xbrl.org/int/dim/arcrole/domain-member"
    order="1.0" xlink:from="loc_s2c_GA" xlink:to="loc_s2c_x0" xbrldt:usable="false" />
        …
    </link:definitionLink>
```

VII.3.4 Modules

Modules serve as entry points to subsets of information requirements that shall be used for filing (the only files referenced from XBRL instance documents) depending on the reporting scenario (reporting frequency, solo or group data, etc.) as defined in the underlying DPM Annotated Templates.

Modules are represented using XBRL abstract items of the module type (*model:moduleType*). Each module is stored in a different schema file whose name is the same as the code of the module in the model (*{module code}*) plus the extension *.xsd*. General properties of a module are presented in *Table 26*.

Table 26. General properties of a module.

Schema property	Value
Official location	{taxonomy-loc}/mod/{module code}.xsd
Target namespace	{taxonomy-bns}/mod/{module code}
Target namespace prefix	{opre}_mod_{module}
Element local name	mod_{module}
Element id	{opre}_mod_{module}

Examples of module declarations are presented in *Table 27*.

Table 27. Example of a module declaration.

Owner	Schema property	Value
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/mod/ars.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/s2md/fws/solvency/solvency2/2016-07-15/mod/ars
	Target namespace prefix	s2md_mod_ars
	Local name example	ars
	Element id example	s2md_ars
	Element label (English)	Annual Solvency II reporting Solo

Module schema files import the schemas of all the tables required by that module (from the *tab* folder of the taxonomy; determining the subset of information requirements for a particular reporting scenario defined by a module). They also import the filing indicators schema file and a schema file referring to the custom functions definition and implementation (see *VI.1 Model supporting schema and other technical files*).

In addition to these imports, the module schema file references some linkbase files:

- label linkbase file with label for a module (*{module code}-lab-{lang}.xml*),
- presentation linkbase (*{module code}-pre.xml*) where the relationships between modules, table groups, and tables are expressed using the legacy *group-table* arcs (defined in the Eurofiling *model.xsd* schema file),
- a linkbase file defining precondition tests for filing indicators (*{module code}-find-prec.xml*),
- value assertion definition (*{module code)-find-check.xml*), label (*{module code)-find-check-lab-{lang).xml*) and error message (*{module code)-find-check-err-{lang).xml*) checking and documenting the values of filing indicators applicable to a module (see *VII.3.5*),
- assertion sets and value assertions (validation rules) definitions, labels, and error messages declared in the *val* folder of the taxonomy in scope that applies to the tables imported by a module (see *VII.3.6 Validations*),
- linkbase files that are used to deactivate assertions as described in http://eurofiling.info/portal/taxonomiesmechxml-blacklist/; these includes 1) deactivations resulting from rules not applying to a module despite the fact that it refers all tables affected by these rules (e.g. ECB add on rules that shall not apply to regular EIOPA reporting) and 2) rules that are deactivated after the taxonomy was published; both files are named according to the following pattern *{module code)-ignore-val.xml* where the first one is stored in the *mod* folder while the latter in the folder as described in VII.3.6.8 Deactivation of rules.

VII.3.5 Filing indicators

The principle of proportionality stipulates that an entity's reporting burden should be proportional to its size. It allows a filer to report less information if it satisfies certain criteria. For example, this principle allows a smaller organisation to file less information if it is not active in some domains or if some figures are under a given threshold.

The simplest technical solution to this business requirement is to define a module (an entry point) for each reporting scenario. Each entry point exposes only the subset of the model and validation checks specific to the reporting scenario in question. However, if several characteristics and/or thresholds are defined to cope with the proportionality principle, a different entry point must be defined for each and every valid combination of characteristics. This complicates:

- the filing process, where the filer must choose an appropriate entry point from a potentially large selection that differ in subtle ways,
- the taxonomy, where several entry points must be defined, tested, and assured with added complexity if some assertions are shared between entry points and some are not (which is typically the case),
- the submission handling process, where the received instances must be processed against one of many different taxonomies,
- the maintenance of the taxonomy, where every time a new characteristic or threshold is introduced for proportionality, the number of entry points could be as much as doubled.

The idea of a filing indicator enables a single entry point to be shared between different similar reporting scenarios. The content of each entry point is notionally split into several components, and every component (typically corresponding to a template) that is reported in an instance is accompanied by an explicit indication that the reporting unit has been filed.

Filing indicators, therefore serve the purpose of communicating the scope of the reported data based on templates. Their main purposes are:

- to provide hints to applications handling instance documents as to which templates are included in the filing and, for example, shall be displayed to users,
- to trigger the execution of business rules (XBRL assertions) to be run on a filing to check its correctness depending on the reported scope of data.

In technical terms, filing indicators are facts included as part of an instance document where the filer provides information about the reported templates (within the scope defined by a module that the filing is created against, see previous section on *Modules*).

The elements and attributes used to communicate filing information are defined in the namespace *http://www.eurofiling.info/xbrl/ext/filing-indicators*. The official location of this schema file is *http://www.eurofiling.info/eu/fr/xbrl/ext/filing-indicators.xsd*. This schema file is imported by its EIOPA counterpart (*http://eiopa.europa.eu/eu/xbrl/ext/filing-indicators.xsd,* see *VI.1*) which in turn is imported by each taxonomy module. As described in *Table 2*, throughout this document, the prefix *find* is used to make reference to *http://www.eurofiling.info/xbrl/ext/filing-indicators* namespace.

Each reported template is represented as an instance fact of the item *find:filingIndicator* under the *find:fIndicators* tuple element. If there is no filing indicator for a template included in an instance document, it is assumed that a filing contains no information on this template. In some cases, however, it may be necessary that filers explicitly identify unreported templates, usually with a reason explaining this situation/choice. To cater for this situation, a *find:filingIndicator* fact relating to the template identification can have a *find:filed* attribute set to boolean *"false"*.

The following instance excerpt represents a filing with information about a template with code S.02.01 and no information (explicitly stated) on template S.03.02:

```
<find:fIndicators>
  <find:filingIndicator contextRef="ctx">S.02.01</find:filingIndicator>
  <find:filingIndicator contextRef="ctx" find:filed="false">S.03.02</find:filingIndicator>
</find:fIndicators>
```

Contexts to which *find:filingIndicator* facts refer must identify the reporting entity and use the end date of the reporting period as the instant date[21].

Identification of templates in *find:filingIndicator* facts uses codes. These codes can be found as a label resource associated with the *table:table* element in an extended link using the standard role; the label resource uses the role *http://www.eurofiling.info/xbrl/role/filing-indicator-code* (as defined in the Eurofiling *model.xsd* supporting schema).

[21] EIOPA defines a dimensional construct in *http://eiopa.europa.eu/eu/xbrl/ext/filing-indicators-def.xml* (referred to from *http://eiopa.europa.eu/eu/xbrl/ext/filing-indicators.xsd*) that prohibits use of *xbrli:segment* or *xbrli:scenario* in *xbrli:context* referred to from filing indicator elements.

Note that multiple tables belonging to one template have the same filing indicator code, but the filing indicator does not need to be repeated if multiple such tables are submitted.

EIOPA information requirements include a *Content template* for each module that details which templates have been included in a filing and if a template has been omitted, why. Filing indicators may appear to serve the same purpose as the content templates but filing indicators are a technical mechanism (using XBRL tuples to distinguish from other facts and simplify referring from preconditions on assertions) which has been used to align with the EBA, and the content templates satisfy a business requirement for reasoning behind the inclusion (or not) of templates in a report. There is a series of assertions that ensure that entries in the content table and values of filing indicators are consistent.

Additionally, there is a validation rule associated with each module to check if the reported values of filing indicators match the required codes. This rule is defined as an assertion in *{module code}-find-check.xml* file. The test expression is, for example, *$filingIndicator = ('S.01.01','S.02.01')*. Documentation (label) and error message for this check is defined in *{module code}-find-check-lab-{lang}.xml* and *{module code}-find-check-err-{lang}.xml* files, respectively.

To block the use of *xbrli:scenario* and *xbrli:segment* on contexts that filing indicator elements refer to, EIOPA extended the Eurofiling schema defining filing indicators with a definition linkbase where filing indicators are linked to a closed hypercube with no dimensions attached (files *filing-indicators.xsd* and *filing-indicators-def.xml* in *http://eiopa.europa.eu/eu/xbrl/ext/* folder). Additionally, this folder includes also two XBRL assertions (*filing-indicators-check.xml*, *filing-indicators-check-lab-en.xml*, *filing-indicators-check-err-en.xml*):

- existence assertion (*filingIndicatorsExistanceAssertion*) checking if filing indicators are present in a report,
- value assertion (*filingIndicatorOutsidefIndicatorsTupleAssertion*) checking if filing indicator elements (*find:filingIndicator*) are declared in *find:fIndicators* tuple.

VII.3.6 Validations

Data checks are created according to the XBRL Formula Specification 1.0. In particular, they are defined as assertions which are gathered in assertion sets, use filing indicators on preconditions, and apply interval arithmetic functions on test expressions as described in the next sections.

This chapter relates to validations defined in *val* folder of the framework (see *Annex 1. EIOPA XBRL Taxonomy: Owners, Folders, Files, Namespaces and Prefixes*). For other validations included in the taxonomy (files containing *check* component in their name) see section on *VII.3.4 Modules* (checks for correct codes of filing indicators) and *VII.3.5* (existence checks for filing indicators and their proper representation in a tuple).

VII.3.6.1 Assertions

Validations are expressed using XBRL assertions. In general, assertions are identified by a unique code (*{rule code}*), which is the same as that code used to identify the corresponding validation rule expressed in the EIOPA validations documentation.

Each assertion is defined in a separate file whose name starts with the prefix *vr-* followed by the *{rule code}* and *.xml* extension (for example, *vr-bv38-1.xml*). For documentation purposes, each assertion is associated with a description, appearing as a generic label in *vr-{rule code}-lab-{lang}.xml* (e.g. *vr-bv38-1-lab-en.xml*), which indicates the performed check in business/form-centric terms. Error message (according to the Generic messages 1.0 specification) to appear in case of unsatisfied evaluation is defined in a file whose name follows the pattern *vr-{rule code}-err-{lang}.xml* (e.g. *vr-bv38-1-err-en.xml*). Both the generic label and message resources may use different roles to identify different type of documentation or notes.

Business rules are, in most cases, expressed as value assertions (for potential exceptions, see *VII.3.6.5 Existence assertions*). They refer to variables (usually fact variables, however, the use of generic variables is also allowed) which until the 2.8.0 taxonomy release, were named after the letters of the alphabet (i.e. *a, b, c, d*, …). Currently, variables are given numbers (v1, v2, v3, …). Both the assertions and fact variables may refer to filters (that can be complemented if necessary). Variables may bind as a sequence and contain fallback values.

Assertion resource *xlink:label* and *id* attributes follow the pattern *{opre}_{RULE CODE}* (e.g. *s2md_BV38-1*). *xlink:label* and *id* of variable resources are constructed by adding a variable name to the assertion code, i.e. *{opre}_{RULE CODE}.{variable name}* (e.g. *s2md_BV38-1.a*). Filter resources *xlink:label* and *id* are constructed according to the pattern: *{opre}_{rule code}.f{sequential number}* (e.g. *s2md_BV38-1.f1*).

Each assertion may also be associated with two attributes: *model:fromDate* and *model:toDate* which may be used to express a period of validity, in terms of reporting date ("as of"). Additionally, the list of currently applicable assertions is listed in the Excel file published on the website, together with the taxonomy.

Not all assertions apply to every module. Each module includes, in its DTS, all assertions that are applicable in its context (see *VII.3.4 Modules*).

VII.3.6.2 Assertion sets

Validations are grouped into assertion sets that correspond to the tables to which they are to be applied. In the context of a table, not reported or nil numeric values are assumed to be zero (unless differently indicated in the business rules definition); consequently, fallback values are used in their corresponding assertion definitions.

The link between an assertion set and the table (or tables[22])it applies is represented using *applies-to-table* arcs (defined in the Eurofiling model.xsd supporting schema) from the assertion set to the resource that corresponds to the table.

If an assertion applies to multiple tables individually or to multiple sets of tables, then it will be associated to different assertion sets. Examples of assertion sets application to tables are presented in *Table 28*.

Table 28. Examples of assertion sets application to tables.

Assertion example (textual description)	**Assertion sets**	**Tables**
$a > 0 (where $a represents a data point in table 1)	assertion set 1	table 1
$a > 0 (where $a represents data point in tables 1, 2 and 3)	assertion set 1	table 1
	assertion set 2	table 2
	assertion set 3	table 3
$a = $b (where $a represents a data point in table 1 whereas $b represents a data point in table 2)	assertion set 1	table 1 table 2
$a = $b (where in some cases, $a represents a data point in table 1 and $b a data point in table 2; in other cases, $a represents a data point in table 3 and $b represents a data point in table 4)	assertion set 1	table 1 table 2
	assertion set 2	table 3 table 4

Example of code where an assertion set links to tables and a validation rule is provided below:

```
<gen:link xlink:type="extended" xlink:role="http://www.xbrl.org/2003/role/link">
    <validation:assertionSet xlink:type="resource" xlink:label="assertionSet" id="assertionSet1" />
    <model:properties xlink:type="resource" xlink:label="properties" id="properties" />
    <gen:arc xlink:type="arc" xlink:arcrole="http://www.eurofiling.info/xbrl/arcrole/model-properties"
        xlink:from="assertionSet" xlink:to="properties" />
    <link:loc xlink:type="locator"
```

[22] In the case of assertions that cross information represented in different tables.

```
        xlink:href="../tab/s.02.01.01.01/s.02.01.01.01-rend.xml#s2md_tS.02.01.01.01"
        xlink:label="loc_s2md_tS.02.01.01.01" />
    <gen:arc xlink:type="arc" xlink:arcrole="http://www.eurofiling.info/xbrl/arcrole/applies-to-table"
        xlink:from="assertionSet" xlink:to="loc_s2md_tS.02.01.01.01" />
    <link:loc xlink:type="locator"
        xlink:href="../tab/s.12.01.01.01/s.12.01.01.01-rend.xml#s2md_tS.12.01.01.01"
        xlink:label="loc_s2md_tS.12.01.01.01" />
    <gen:arc xlink:type="arc" xlink:arcrole="http://www.eurofiling.info/xbrl/arcrole/applies-to-table"
        xlink:from="assertionSet" xlink:to="loc_s2md_tS.12.01.01.01" />
    <link:loc xlink:type="locator"
        xlink:href="../tab/s.17.01.01.01/s.17.01.01.01-rend.xml#s2md_tS.17.01.01.01"
        xlink:label="loc_s2md_tS.17.01.01.01" />
    <gen:arc xlink:type="arc" xlink:arcrole="http://www.eurofiling.info/xbrl/arcrole/applies-to-table"
        xlink:from="assertionSet" xlink:to="loc_s2md_tS.17.01.01.01" />
    <link:loc xlink:type="locator" xlink:href="vr-bv138-1.xml#s2md_BV138-1"
        xlink:label="loc_s2md_BV138-1" />
    <gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/assertion-set"
        xlink:from="assertionSet" xlink:to="l loc_s2md_BV138-1/>
</gen:link>
```

Assertion sets resources might include the attributes *model:fromDate* and *model:toDate* to constrain the reference date where their associated assertions should be applied.

Assertion sets are defined in *val* folder. The file name starts with *aset* prefix followed by codes of tables (in lower case) divided by underscore ("_"), for example, *aset-s.01.02.04.01_s.02.01.01.01_s.35.01.04.01.xml*.

As suggested by the XBRL specification, assertion sets can be used as a mechanism to control the set of assertions to be evaluated in a validation process. Following this approach, an application processing a certain filing would configure the processor to skip all those assertion sets that are linked to a table that is not reported. However, currently, the XBRL specifications do not provide a standard API to pass this information to XBRL processors, neither a standard way for the filer to indicate that only a subset of all the tables in an entry point is being submitted. To overcome this situation, a mechanism based on preconditions and filing indicators is provided as described in the next section of this document.

VII.3.6.3 Preconditions for filing indicator parameters

Each value assertion defined is associated with a precondition[23] on filing indicators. To avoid XBRL instance syntactic dependencies, rather than including directly an XPath expression, preconditions include a reference to a filing indicator parameter (no *variableset-variable* arc is required). The default value of this parameter is an XPath expression to obtain the information from the filing indicators in the instance document. This way, there is no need to provide externally a value to the processor (the value from the instance is used), the parameter is guaranteed to be only evaluated once (providing more chances for processors to perform optimisations), precondition expressions are simpler, and it makes possible, for more advanced uses, to override this value at application level (for instance, if the filing requirements of a credit institution are known, an application could override the values for filing indicator parameters rather than accepting the values provided by the filter).

There are filing indicator parameters defined for each template of the framework. These parameters are defined in *find-params.xml* file of the *val* folder in the namespace of the filing indicators schema and have a name according to the following convention: *t{template code}*. Thus, the definition of one of these parameters would be as follows:

```
<variable:parameter name="find:tS.02.01"
select="find:fIndicators/find:filingIndicator[not(@find:filed) or @find:filed != false()] = 'S.02.01'"
as="xs:boolean" xlink:type="resource" xlink:label="S.02.01" id="S.02.01" />
```

Each precondition is composed as a sequence of or expressions that correspond to each set of tables where the validation is to be applied. Each expression is composed of a sequence of tables involved, e.g. *test="$find:tS.02.01 and $find:tS.04.01 or $find:tS.02.02 and $find:tS.04.02 …"*. More examples of application of filing indicators to tables are provided in *Table 29*.

[23] Assertions might have additional preconditions as required by the logic of the assertion to be tested. But these additional preconditions do not depend on filing indicators.

Table 29. Examples of filing indicators application to tables.

Precondition test expression	Explanation
$find:t1	Assertion applies only to table 1. It will be evaluated if table 1 is marked as reported.
$find:t1 and $find:t2	Assertion crosses information between tables 1 and 2. It will be evaluated if table 1 and table 2 are marked as reported.
$find:t1 or $find:t2	Assertion applies to both table 1 and table 2, but considered in an individual way (there are no cross checks). It will be evaluated if table 1 or table 2 or both are reported.
$find:t1 and $find:t2 or $find:t3 and $find:t4	Assertion performs cross-checks between information in table 1 and table 2 on the one hand. On the other hand, it cross-checks information between table 3 and 4. It will be evaluated if table 1 and table 2 is reported or if table 3 and table 4 is reported or when all tables (1, 2, 3 and 4) are reported.

These preconditions are defined for each module in *mod* folder file *{module code}-find-prec.xml* from where they refer to respective assertions in *val* folder, for example:

```
<variable:precondition xlink:type="resource" xlink:label="findPrec" test="$find:tS.03.01" />
<link:loc xlink:type="locator" xlink:href="../val/vr-138.xml#s2md_138" xlink:label="loc_s2md_138" />
<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
xlink:from="loc_s2md_138" xlink:to="findPrec" />
```

VII.3.6.4 Preconditions for specific concepts' parameters

Until version 2.7.0, there were specific concepts included in the taxonomy (in particular in the basic information template) that identified if a submission was under a regular or non-regular reporting scenario. Similarly to filing indicators, these facts taking specified values are declared on parameters and used on preconditions in order to trigger rules with different severity (ERROR or WARNING) – see section VII.3.6.6 (Assertions severity). These are also declared in *find-params.xml*

```
<variable:parameter xlink:type="resource" xlink:label="regularReporting" name="regularReporting"
select="s2md_met:ei1677 = xs:QName('s2c_CS:x35') or s2md_met:ei2503 =
xs:QName('s2c_CS:x35')" as="xs:boolean" id=" regularReporting" />
<variable:parameter xlink:type="resource" xlink:label="nonRegularReporting"
name="nonRegularReporting" select="s2md_met:ei1677 != xs:QName('s2c_CS:x35') or
s2md_met:ei2503 != xs:QName('s2c_CS:x35')" as="xs:boolean" id="nonRegularReporting" />
```

and subsequently applied in *{module code}-find-prec.xml* files:

```
<variable:precondition xlink:type="resource" xlink:label="rp" test="$regularReporting" />
<variable:precondition xlink:type="resource" xlink:label="nrp" test="$nonRegularReporting" />
<link:loc xlink:type="locator" xlink:href="../val/vr-bv6-1.xml#s2md_BV6-1"
xlink:label="loc_s2md_BV6-1" />
```

<link:loc xlink:type="locator" xlink:href="../val/vr-bv6-1_w.xml#s2md_BV6-1_W"
xlink:label="loc_s2md_BV6-1_W" />
<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
xlink:from="loc_s2md_BV6-1" xlink:to="rp" />
<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
xlink:from="loc_s2md_BV6-1_W" xlink:to="nrp" />

VII.3.6.5 Existence assertions

Existence assertions are not compatible with the precondition-based control schema proposed in the previous chapter. Existence assertions perform a test on the number of evaluations of a set of variables. Preconditions restrict the number of evaluations of the assertion, but not the evaluation of the assertion itself. Consequently, existence assertions are always evaluated (unless controlled using assertion sets); if a filing indicator precondition is added to an existence assertion, it will raise false errors.

However, most existence assertions can be re-defined as value assertions using, in addition, the "pivot variable" - a fact variable that matches data in the instance document known to be reported always (it is defined once as a sequence variable that matches the filing indicators and uses aspect cover filters to avoid any interference with other variables). The rest of the variables in the original existence assertion are included with a fallback value (a value given to the variable if the fact is not found in the instance document).

The pivot-variable is defined in the namespace *http://www.eurofiling.info/xbrl/ext/pivot-variable*. The official location of this schema file is *http://www.eurofiling.info/eu/fr/xbrl/ext/pivot-variable.xsd*.

Though unlikely, there might be cases of validations that cannot be (effectively or efficiently) defined using value assertions. If such rules were required, the *id* attribute value of such assertions would follow a predefined naming convention (to be established when such a situation occurs) to help applications not relying on validation sets to discard such evaluations.

VII.3.6.6 Assertions severity

Each assertion is assigned with severity as defined in the XBRL Assertions Severity specification (19 April 2016). Locator for severity resource and an arc linking it to an assertion is defined in the same file as the assertion itself (i.e. *vr-* followed by the *{rule code}* and *.xml*). Currently, the taxonomy applies ERROR and WARNING severity levels. In case of ERROR, the submission is blocked. WARNING does not block the request but allows to provide information about possible discrepancies.

As described in section VII.3.6.4 (Preconditions for specific concepts' parameters) until version 2.7.0, there were specific concepts in the basic information template that could have an impact on the severity of assertions. In particular, for non-regular reporting, all assertions with ERROR severity levels are downgraded to WARNING. In order to achieve that, these assertions were duplicated in the taxonomy with a suffix "_W" e.g. "vr-bv6-1.xml" with ERROR severity has a counterpart "vr-bv6-1_w.xml" with severity WARNING. This solution was considered to replace this a dynamic assertion severity mechanism as per the recently published XBRL specification: https://www.xbrl.org/Specification/assertion-severity/PR-2022-02-02/assertion-severity-PR-2022-02-02.html. Starting from 2.8.0, it was decided to eliminate the validation rules for non-regular reporting, resulting in each assertion having only one static severity.

VII.3.6.7 Evaluation of validation rules, interval arithmetic and custom margin functions

Any arithmetic comparison may not be exact due to rounding of figures and their representation. For example, in a simple expression *A = B / C* where *B = 1000*, *C =3000* the result of division is *0.333333*…. If *A* is reported as *0.33,* then compared to the result would raise an error.

To handle the error margin caused by the imprecision of input data, assertions make use of a set of functions implemented according to the Custom Functions Implementation specification. These functions use the same name as the ones defined in the XPath 2.0 Functions specifications, but are defined in the following namespace *http://www.eurofiling.info/xbrl/func/interval-arithmetics* (with canonical prefix *iaf*) and placed in the location: *http://www.eurofiling.info/eu/fr/xbrl/func/interval-arithmetics.xml*. An entry point (referred from taxonomy modules) for these functions and additional ones that could be provided in the future is placed in the *http://www.eurofiling.info/eu/fr/xbrl/func/func.xsd*.

In interval arithmetic, each reported number is converted to an interval based on its expression (reported value) and precision (*@decimals* attribute[24]) as exemplified in *Table 30*.

[24] Or *@precision* attribute which is however prohibited by the EIOPA XBRL Filing Rules published on https://www.eiopa.europa.eu/tools-and-data/supervisory-reporting-dpm-and-xbrl_en.

Table 30. Examples of intervals.

Reported number	@decimals	Precision	Interval
123 456.789	-3	in thousands (+/- 500)	(122 956.789; 123 956.789)
	0	in units (+/- 0.5)	(123 456.289; 123 457,289)
	2	to two digits after decimal point (+/-0.005)	(123 456.784; 123 456.794)
	INF	exact (+/- 0)	(123 456.789; 123 456.789)

Following that conversion, the interval arithmetic functions use basic operations (as implemented in *http://www.eurofiling.info/eu/fr/xbrl/func/interval-arithmetics.xml*) to compute the resulting intervals after applying mathematical operations. For instance, in the case of the addition of two numbers *A* and *B*, where *A* is an interval of *(A1;A2)*, *B* is an interval of *(B1;B2)* the result is an interval of *(A1+B1;A2+B2)*. If the interval of the reported number overlaps with the computed interval, the assertions is satisfied. An example in *C* = *A* + *B*, where:

- *A* is reported as 1499 with precision in units (*@decimals = 0*) hence the resulting range is (1498.5;1499.5),
- B is reported as 1502 with precision in units (*@decimals = 0*) hence the resulting range is (1501.5;1502.5),
- C is reported as = 3000 with precision in units (*@decimals = 0*) hence the resulting range is (2999.5;3000.5).

Following the basic operations, the computed tolerance interval for *A* + *B* is (1498.5+1501.5;1499.5+1502.5) = (3000;3002). As presented in *Figure 3* there is an overlap (marked in orange) between the interval of *C* (in blue) and the interval of *A* + *B* (in green). As a result, the assertion is satisfied.

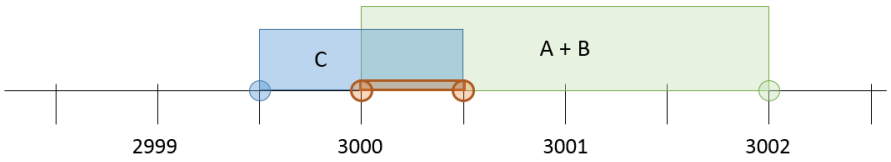


Figure 3. Overlap of intervals

If *C* was reported as 2999, the resulting interval (with precision in units) would be (2998.5;2999.5). With no overlap (see *Figure 4*) the assertion would not be satisfied, and an error would be raised.

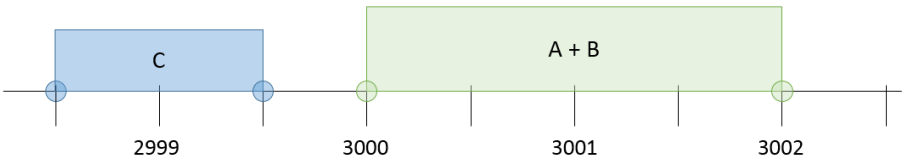


Figure 4. No overlap of intervals

Implementation of interval arithmetic defines the following functions:

- *iaf:sum*,
- *iaf:numeric-equal*,
- *iaf:numeric-less-than, iaf:numeric-less-equal-than*,
- *iaf:numeric-greater-than, iaf:numeric-greater-equal-than,*
- *iaf:numeric-add,*
- *iaf:numeric-subtract,*
- *iaf:numeric-divide,*
- *iaf:numeric-multiply,*
- *iaf:abs, iaf:min, iaf:max,*

where for example:

- *iaf:numeric-equal(arg1, arg2)*: returns *true* if two values are equal or are within the tolerance interval derived from their reported precision,
- *iaf:numeric-less-than(arg1, arg2)*: checks whether *arg1* is less than *arg2*, considering their precision.

In more complex expressions, functions are nested following the order of their executions. For example, *a = ((b – c) / (d * e)) + b* would be defined as *iaf:numeric-equal($a,iaf:sum((iaf:numeric-divide(iaf:numeric-subtract($b,$c),iaf:numeric-multiply($d,$e))),$b))*.

For simple comparison of a value with a constant interval arithmetic is typically not applied.

An alternative approach to interval arithmetic has been implemented by EIOPA in some validation rules starting on the Solvency II 2.10.0 XBRL taxonomy. In this approach to handle the error margin caused by the imprecision of input data, the threshold is applied to the difference between the left-hand side and the right-hand side of the comparison operator. Two different variants have been implemented, considering the nature of the threshold:

A. Relative error for a percentage threshold, and

B. Absolute error for an absolute threshold.

For instance, in the case of the addition of two numbers, B and C, where B is 499000, C is 502000, and the result of the addition is A with a value of 1000000.

The relative error variant requires to define a threshold (allowed margin) as a percentage. In the presented example it is assumed that this allowed margin is 1%. The relative error variant is going to check that the absolute value of ([hand-left part of

equality] divided by [hand-right part of equality] minus 1) must be less than the allowed margin. For the example above, it would be:

abs((A/(B+C)) – 1) < threshold

abs((1000000/(499000+502000)) – 1) < 1%

0.1% < 1% → Validation satisfied

For the absolute error variant, a threshold must be defined (allowed margin) as an absolute number. For this example, that allowed margin is 5000 monetary units.

The absolute error variant is going to check that the absolute value of [hand-left part of equality] minus the [hand-right part of equality] must be less than or equal to the allowed margin. For the example above, it would be:

abs(A-(B+C)) <= threshold

abs(1000000 – 1002000) <= 5000

2000 <= 5000 → Validation satisfied

VII.3.6.8 Deactivation of rules

Each module references a linkbase file, which serves the purpose of deactivating assertions. The name of the file follows the pattern *{module code}-ignore-val.xml*. The mechanism is described in http://eurofiling.info/portal/taxonomiesmechxml-blacklist/.

The folder with files used to deactivate the assertions can be found in the following location: https://dev.eiopa.europa.eu/Taxonomy/Full/deactivations/{taxonomy version code}.

VIII Comments and processing instructions

Top lines of each taxonomy file contain processing instructions and comments:

- First line contains declaration of XML version and encoding, which is always 1.0 and UTF-8, respectively.
- The second line is a comment with copyright information and descriptive information on the taxonomy.
- The third line is a processing instruction with the official URI of a file.
- Fourth line is a processing instruction identifying taxonomy version.
- Fifth line is a processing instruction with the taxonomy release date.

An example of the first five lines of code in every taxonomy file is presented below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--(C) EIOPA - XBRL Taxonomy-->
<?officialURI http://eiopa.europa.eu/eu/xbrl/s2md/fws/solvency/solvency2/2024-10-15/tab/s.01.01.01.01/s.01.01.01.01.xsd?>
<?taxonomy-version 2.8.2?>
<?taxonomy-date 2024-10-15?>
```

IX Mapping between MD and HD properties

According to the underlying DPM and the approach taken to simplify the description of information requirements, reporting in XBRL is made based on the MD version of the model, where metrics also include some dimensional properties defined in the HD version.

To provide a structured reconciliation between the MD and HD metrics, the taxonomy is associated with an XML file *MDMetricDetails.xml* that decomposes an MD metric into an HD metric and the remaining HD dimensions[25].

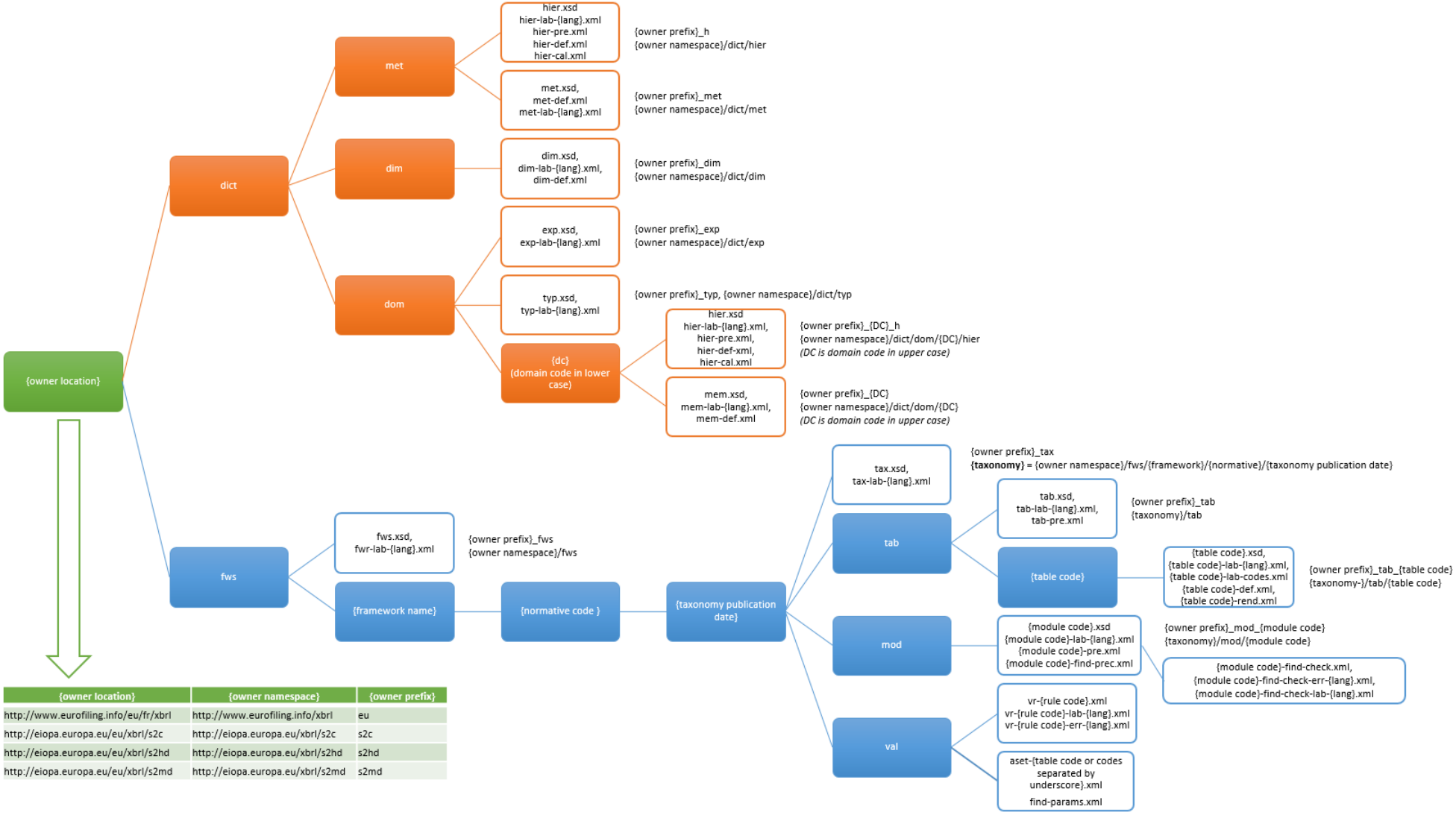
Each MD metric is defined as a *metric* element with *@label* attribute whose value is a standard label of an MD metric and *@name* attribute reflecting its QName (according to the canonical namespace prefixes declared on the root *metrics* element). The content of *metric* element is an *HDMetric* element and a set of *dimension* elements with *domainMember* subelements. All these elements contain @label and @name attributes that correspond to a metric/dimension/member standard label and QName respectively as exemplified below:

```
<metric label="Metric: Monetary|BC/Assets|AS/Corporate Bonds|IO/Investment" name="s2md_met:mi211">
    <HDMetric label="Monetary" name="s2hd_met:mi1"/>
    <dimension label="Basic concepts" name="s2c_dim:BC">
        <domainMember label="Assets" name="s2c_BC:x3"/>
    </dimension>
    <dimension label="Type of assets" name="s2c_dim:AS">
        <domainMember label="Corporate Bonds" name="s2c_MC:x176"/>
    </dimension>
    <dimension label="Investment or own use" name="s2c_dim:IO">
        <domainMember label="Investment" name="s2c_PU:x7"/>
    </dimension>
```

[25] Note that *MDMetricDetails.xml* includes only these MD metrics that are used in the framework (i.e. are applied in any of the tables of the published taxonomy).

</metric>

Annex 1. EIOPA XBRL Taxonomy: Owners, Folders, Files, Namespaces and Prefixes



{owner location}	{owner namespace}	{owner prefix}
http://www.eurofiling.info/eu/fr/xbrl	http://www.eurofiling.info/xbrl	eu
http://eiopa.europa.eu/eu/xbrl/s2c	http://eiopa.europa.eu/xbrl/s2c	s2c
http://eiopa.europa.eu/eu/xbrl/s2hd	http://eiopa.europa.eu/xbrl/s2hd	s2hd
http://eiopa.europa.eu/eu/xbrl/s2md	http://eiopa.europa.eu/xbrl/s2md	s2md

Annex 2. Multiple values for a fact

Some facts in EIOPA XBRL taxonomies represent predefined lists of options, i.e. the LOGs identify the set of allowed values to be reported in a cell. These are modelled in the EIOPA XBRL Taxonomies using the XBRL Extensible Enumerations specification[26], i.e. the allowed options are represented as domain members gathered in a hierarchy (definition linkbase relations in a given extended link role [ELR]), and a metric representing such facts refers to this hierarchy (ELR), optionally identifying also the starting member and if it is included as one of the allowed values.

An example of such a metric is *"Metric: Simplifications - spread risk - bonds and loans"* (*"s2md_met:ei2390"*[27]) which refers to hierarchy number 17 of AP domain listing two members: *"Simplifications used"* (*"s2c_AP:x33"*) and *"Simplifications not used"* (*"s2c_AP:x34"*). In the hierarchy-specific ELR, these members have hierarchy specific labels (in this case *"1 – Simplifications used"* and *"2 – Simplifications not used"* respectively, as in other enumerations the same member can be assigned different labels; also, these labels, ultimately presented to the users, may be provided in various languages).

As a result, the representation of such a fact (reported, for example, in table S.26.01.01.03, R0010, C0010) in an XBRL instance document may look as follows:

```
<s2md_met:ei2390 contextRef="c1">s2c_AP:x33</s2md_met:ei2390>
```

In some cases, however, such as in column C0090 of S.30.02.01.01 representing *"Activity code broker"*, **the value of a cell may include one or more options from a given set defined** in the LOGs. As such a case is not addressed by the XBRL Extensible Enumerations specification, this document explains a few considerations and alternatives for the representation of such facts in the EIOPA XBRL taxonomies.

---

[26] http://www.xbrl.org/Specification/ext-enumeration/REC-2014-10-29/ext-enumeration-REC-2014-10-29.html

[27] Throughout this document the canonical namespace prefixes are applied in element qualified names.

Currently this case is approached by assigning in the LOGs a sequential number to each option. For example, C0090 of S.30.02.01.01 is described in the LOGs as:

Representing the activities of the broker involved, as considered by the undertaking. In case the activities are combined all activities must be mentioned separated be a ",":. 1 - Intermediary for placement 2 - Underwriting on behalf of 3 - Financial services

Although not explicitly addressed by the Filing Rules, it was assumed that an undertaking shall report a pattern based on the numbers assigned to each option in ascending order and separated by commas, for example, *"1"* or *"1,2"* or *"1,3"*, *"2,3"*, *"1,2,3"*, …

As a result, the element *"Metric: String|TS/Activity code broker"* (*"s2md_met:si1858"*) is modelled as a string data type, and its representation in the XBRL instance document may look as follows:

```
<s2md_met:si1858 contextRef="c1">1,2,3</s2md_met:si1858>
```

The pattern has also not been checked by the XBRL taxonomy version 2.0.1.

The topic has been discussed by the EIOPA Taxonomy Working Group during the time of preparation for 2.1.0 XBRL Taxonomy. It was identified that:

- from the data exchange standpoint, it should be possible to validate if the content of such facts is correct (i.e. defined according to the required pattern/set of enumerated values and consistent across reports),
- from the user perspective it is important that the solution to this issue documents and describes the allowed options identified in the LOGs which can be used as hints or tips in the tools when editing or viewing the data. It is also required that it is multilingual or at least language agnostic.

Based on these requirements, several options were identified to address this issue.

From the validation perspective, it was considered to:

1. Add a value assertion to check if the value reported for such a fact follows the allowed pattern/subset of values[28] (such an assertion would list all allowed combinations[29]),
2. apply the Extensible Enumerations (as described in the introduction) and allow for duplicated facts[30] in an instance document i.e., the fact would be reported one or more time[31] with different member QNames, for example:

   ```
   <s2md_met:eiNNNN contextRef="c1">s2c_XX:x1</s2md_met:eiNNNN>
   <s2md_met:eiNNNN contextRef="c1">s2c_XX:x2</s2md_met:eiNNNN>
   ```

3. list all combinations as separate values and apply Extensible Enumerations specification (it may be impractical as in some cases the number of combinations may be hundreds and their labels quite long),
4. customise the Extensible Enumerations to allow many QNames as fact value, e.g.:

   ```
   <s2md_met:eiNNNN contextRef="c1">s2c_XX:x1, s2c_XX:x2</s2md_met:eiNNNN>
   ```

The latter was discussed within the XBRL International Working Group but was pushed out of scope for the first version of the Extensible Enumerations specification, largely because there wasn't any good and clean way to implement it. The ideal solution would be an *xsd:list* of *xsd:QNames*, but unfortunately, the XBRL 2.1 specification disallows the necessary type derivation. Otherwise, types are no longer native *xsd:QNames*, meaning that use of invalid prefixes wouldn't get caught by the XML Schema validation

---

[28] It is also possible to check such patterns on XML Schema level however all validations in EIOPA XBRL taxonomies are preferably defined as XBRL Formula assertions (to benefit from such features as labels, etc.).

[29] This also provides a mechanism to identify invalid combinations (e.g. if *"2,3"* is not allowed to be reported, it is excluded from the list of allowed combinations).

[30] An attribute on such metric declaration could flag it as potentially multiple occurrence facts.

[31] Duplicated facts are however discouraged in the filing rules and an exception for this rule would have to be defined for specific enumerated metrics.

and implementations are forced to decode the prefixes themselves, which is not trivial[32]. However, during meetings in June 2016 in Frankfurt am Main, the XBRL International Working Group decided to draft the specification including the new data type *enum:enumeration**s**ItemType* that would work similarly to the current *enum:enumerationItemType* (applying also *@domain*, *@linkrole* and *@headUsable* attributes) but do the same checks as *xsd:list* of *xsd:QNames* for validation. The Extensible Enumerations 2.0 specification was released as a Recommendation on February 12, 2020[33] and is considered for implementation in one of subsequent releases of the Solvency II, Pension Funds and Pan-European Personal Pension Product XBRL taxonomies.

From the user perspective standpoint there are at least two options in XBRL to document the description in the LOGs in order to provide hints for filers on the allowed options.

One (A) is to provide a more descriptive label containing the text from the LOGs (as in the grey box above). It may be achieved using one of the standard labels (e.g. documentation label) or a custom label.

Another option (B) is to define in the dictionary the domain members representing each option, link them in a hierarchy and refer to from the metric definition (similarly to how it is done in case of Extensible Enumerations, using similar attributes).

As these cases usually appear in open tables another general approach considered was to split this information in individual columns (one per each option). This, however, would change the look of templates, which was undesired.

After analysing the issue, it was decided that for releases later than 2.0.1 (i.e. 2.1.0, 2.2.0 and 2.3.0), the allowed options (represented by sequential numbers) are checked by value assertions (added as Technical Validations). Additionally, the domain members and hierarchies are added in the DPM dictionary in order to be applied in the future by means of the updated XBRL Extensible Enumerations specification 2.0, potentially starting with the EIOPA adopting the new DPM Refit XBRL taxonomy architecture.

---

[32] Full discussion on this thread can be found here: http://lists.xbrl.org/mailman/private/int-spec/2014-March/004741.html

[33] https://www.xbrl.org/Specification/extensible-enumerations-2.0/REC-2020-02-12/extensible-enumerations-2.0-REC-2020-02-12.html