Representation in XBRL of the Data Point Model

Documentation of XBRL taxonomy

Abstract

This document describes and explains the architecture of the public consultation version of the draft XBRL taxonomy for second level supervisory reporting developed by the European Banking Authority. In particular, it explains the semantics and syntax used to express the information requirements of the data point model in XBRL format, and presents modularisation of the taxonomy folder and files, naming conventions, and descriptive attributes used.

Date : 16/09/2013	

Table of Contents

1 Introduction

This document presents and explains the architecture of the XBRL taxonomy for second level reporting defined by the European Banking Authority.

The expected direct audience of this document are software developers working directly or indirectly for national competent authorities that will be required to pass supervisory data to the EBA using this taxonomy. This document is also useful for developers of software that produces or consumes instance documents following this taxonomy.

Additionally, given the possibility of this taxonomy forming, to some degree, the basis for reporting from credit institutions to some national competent authorities, it will also be of interest more widely to credit institutions and vendors of software involved in the regulatory reporting process.

2 Relation to other standards and documents

Comprehension of the Extensible Business Reporting Language (XBRL) 2.1 Specification and various other XBRL Specifications such as XBRL Dimensions 1.0, XBRL Formula 1.0, Generic Link 1.0 and Table Linkbase 1.0 (Public Working Drafts) is 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 EBA followed the approaches applied for various deliverables of the Eurofiling project[1].

In particular, the EBA applied the Data Point Modelling methodology and the Data Point Model [DPM] format to the description of the exchanged data[2].

The mapping of this DPM to and XBRL taxonomy follows the general architectural approach of the preliminary finrep taxonomies published on the Eurofiling website[3], an approach shared with the EIOPA Preparatory Solvency II taxonomy[4].

3 Data model

Prior to 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. Data

[1] 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). All deliverables of the Eurofiling project can be found on http://www.eurofiling.info

[2] Meta model of the DPM: see accompanying file DPM-Formal-Model.pdf

[3] http://www.eurofiling.info/finrepTaxonomy/EBA-DPM-XBRL-Mapping.pdf

[4] https://eiopa.europa.eu/publications/eu-wide-reporting-formats/index.html

models organize the data for communication purposes (e.g. between business and IT experts, or between various groups of business experts).

In the case of CRR reporting, the inputs for creation of the data model are Implementing Technical Standards, consisting of the main provisions covering the reporting requirements, the reporting templates, i.e. tabular representation of information requirements, the instructions associated with these templates, and the related validation formulae.

These templates, provisions, instructions and underlying regulations are analysed according to the Data Point Modelling methodology in order to create a Data Point Model format.

In the case of the ITS data model, the DPM format consists primarily of a structured Microsoft Access database, the content of which is also documented via two Microsoft Excel workbooks:

Dictionary - defining properties (and their classifications/breakdowns) that can be used to describe each exchanged piece of information, and hierarchical relations between them.

Table Layout and Data Point Categorisation - Annotated tables where each row/column/sheet is associated with a property or a set of properties defined in the dictionary.

As a result, the DPM database defines a set of reportable cells (data points) in tables by specifying all of the properties (according to the content of the dictionary) required to convey their full meaning.

The preparatory taxonomy was created by (automated) translation of the DPM database format into XBRL syntax based on the rules described in this document.

4 XBRL specifications compliance

Following the XBRL standard requirements, the EBA taxonomies, and any XBRL instance documents are compliant with the XBRL 2.1 specification as of December 31, 2003 with Errata Corrections up to January 25, 2012, 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 – 2011 and supporting specifications (Registry – 2009-2011, Generic Links – June 22, 2009).

Rendering of tables is created according to the Public Working Draft of the Table Linkbase specification published on 17 May 2013.

Due to unfortunate overlap between the implementation period of the EBA XBRL taxonomy, and the development of the table linkbase specification, it is sadly not expected for the table linkbase specification to reach recommendation status before the EBA taxonomy is finalised. As such, and in the interest of stability, the current intention is to continue to utilise the 17 May 2013 PWD version for initial

implementation, with any final recommendation version being adopted as part of a subsequent normal maintenance cycle.[5]

For convenience for reviewers, the taxonomy files provided contain technical files defined by various XBRL specifications and registries. They are placed in the folder www.xbrl.org. In addition shared files from www.eurofiling.org are also included. The inclusion of these files simplifies the use of the supplied taxonomy files offline if required.

The Taxonomy files reference these files in their official locations. As such mappings will usually be required to be configured in most XBRL software to utilise the local version of these files, rather than those at the official locations, if so desired.

In a primary production release of the taxonomy, by normal XBRL convention, these files would likely not be included, and arrangements should be made to utilise, at least notionally, the official copies of the files from the official locations.

[5] This position will of course be revisited as and when newer versions are released, with the cost/benefit balance of adopting any significant advance in the specification being considered on its merits, however non-adoption until a post introduction maintenance release is considered the most likely outcome.

5 Supporting concepts

This chapter describes some concepts to facilitate the definition of the mapping rules between the abstract data point model and XBRL taxonomies.

5.1 Owner

The owner represents an institution that defines concepts of the model. The owner is closely related to the idea of extensibility in XBRL. The main properties of the owner are:

- *Owner's namespace (ons) and owner's prefix (opre)*: the owner namespace is a URI used to establish the namespace of the concepts defined by that owner. This URI is generally built by adding the "xbrl" particle to the internet domain of the institution that the owner represents plus an optional particle ("crr" in the case of EBA). The use of this particle enables the definition in the future of new models covering different functional areas where the use of a common dictionary might not be considered convenient, or the possibility of creating a major version of an existing model with a completely renewed dictionary.

  The prefix is used as the basis to establish namespace prefixes in taxonomy files and for some short representations of the concepts. Namespace prefixes do not impose any constraints on instance files. Namespace prefixes are local to XML documents and XML elements, thus, instance files and taxonomy consumers should never presume any particular use of prefixes; XML documents consumption must be based on namespaces.

Owner	Internet domain	Namespace	Prefix
European Banking Authority	http://www.eba.europa.eu	http://www.eba.europa.eu/xbrl/crr	eba
Eurofiling[6]	http://www.eurofiling.info	http://www.eurofiling.info/xbrl	eu
Banco de España	http://www.bde.es	http://www.bde.es/xbrl	es

- *Official location (oloc)*: URL used to specify the location where taxonomy files associated to that owner are to be published. Different owners must have different official locations, even owners with the same internet domain / same namespace. The official location is generally built by adding three particles to the internet domain of the institution: one that represents the geographical

[6] For concepts shared with other European supervisors

area covered by the institution, plus two fixed ones: “fr” (for financial reporting) and “xbrl”:

Owner	Official location
European Banking Authority	http://www.eba.europa.eu/eu/fr/xbrl/crr
Eurofiling	http://www.eurofiling.info/eu/fr/xbrl
Banco de España	http://www.bde.es/es/fr/xbrl

- *Copyright*: text used as a header in every taxonomy file published by its owner.
- *Supported languages*: list of languages used in taxonomy files defined by an institution. It is used to deduce the location of label linkbases in a certain language given the owner of the concept. This enables the addition of labels to concepts imported from other taxonomies.

5.2 Model supporting schema

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.* Throughout this document, the prefix “*model*” will be used to make reference to this schema namespace.

5.3 Namespaces

The following table shows the prefixes used throughout this document as an abbreviated reference to namespaces:

Prefix	Namespace
xbrli	http://www.xbrl.org/2003/instance
xbrldt	http://xbrl.org/2005/xbrldt
link	http://www.xbrl.org/2003/linkbase
xl	http://www.xbrl.org/2003/XLink
gen	http://xbrl.org/2008/generic
iso4217	http://www.xbrl.org/2003/iso4217
nonnum	http://www.xbrl.org/dtr/type/non-numeric

num	http://www.xbrl.org/dtr/type/numeric
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/functions/interval-arithmetics
variable	http://xbrl.org/2008/variable

6 Public elements

Public elements are 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.

Public elements include two attributes to reflect the creation date of the element (*model:creationDate*) and the date when it was last modified (*model:modificationDate*).

Language specific information is represented using label resources (generic ones for concepts represented as XLink resources and standard ones for concepts represented as XBRL items). The default role (http://www.xbrl.org/2003/role/link) will be used for the extended links containing this information. The following roles must be used for label resources:

Property	Generic label role	Standard label role
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[7]	http://www.xbrl.org/2008/role/documentation	http://www.xbrl.org/2003/role/documentation

The labels of the concepts of a schema file are represented together in label linkbases by language, in the same folder as its corresponding schema file. The naming convention for these linkbases is:

[7] 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.

{main-file}-lab-{lang}.xml

Where *{main-file}* corresponds to the name of the schema or linkbase file where the concept is defined without extension, and {lang} corresponds to the ISO 639-1 code of the language (lowercase). In case of needing any region or country code to identify more specifically the language, the following notation shall be used:

{main-file}-lab-{lang}-{country}.xml

Where {country} corresponds to the ISO 639-2 code of the region or country (lowercase).

In addition to this, some concepts of the dictionary may contain a special linkbase to represent codes needed for different purposes. More specifically, the codes given to the columns and rows of tables are represented using this mechanism. The name of this linkbase is as follows:

{main-file}-lab-codes.xml

The labels for these codes will be represented as resources with the following role, as defined in the model schema:

http://www.eurofiling.info/xbrl/role/rc-code

Extensions might use this mechanism to add their own application specific codifications using different roles.

7 Dictionary of concepts

The core concepts of the dictionary are metrics, dimensions, domains and domain members. Secondary concepts are families and perspectives (auxiliary concepts meant to group dimensions for presentation purposes).

All the concepts in the dictionary are public elements. In addition to the properties and language specific information of public elements, dictionary elements include two optional attributes that establish its currency period: the starting date of the period interval (*model:fromDate* attribute) and its end date (*model:toDate* attribute). If the "fromDate" attribute is not included, then the concept is assumed to be current for any period prior to the "toDate" attribute. If the "toDate" attribute is not included, then the concept is assumed to be current for any period after the "fromDate" attribute. If neither "fromDate" nor "toDate" attributes are included, then the concept is assumed to be current for any period of time. The first versions of the dictionary won't include this attribute. As new versions are released and some concepts become obsolete and replaced by others, these attributes will be updated. These attributes

don’t have any impact on the reporting process itself; they are meant to make easier the management of the concepts of the dictionary.

All files in the dictionary of concepts are placed under the folder “*dict*” in the official location of its owner. Its namespace is obtained by adding a suffix that depends on the type of element to the namespace of the owner. The prefix to represent that namespace is obtained by adding a predefined suffix to the prefix of its owner:

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 of domain	{oloc}/dict/dom/{dc}/mem.xsd	{ons}/dict/dom/{DC}	{opre}_{DC}
Families	{oloc}/dict/dim/fam.xsd	{ons}/dict/fam	{opre}_fam
Perspectives	{oloc}/dict/dim/pers.xsd	{ons}/dict/pers	{opre}_pers

Where {oloc} represents the official location of taxonomy files of the owner of the concepts, {ons} its base namespace, {opre} the prefix of its base namespace, and {dc}/{DC} the code of a domain in lower and capital case. In the case of the dictionary of concepts of the EBA:

Dictionary concept	Official location	Target namespace	Prefix
Metrics	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/met/met.xsd	http://www.eba.europa.eu/xbrl/crr/dict/met	eba_met
Dimensions	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dim/dim.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dim	eba_dim
Explicit domains	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/exp.xsd	http://www.eba.europa.eu/xbrl/crr/dict/exp	eba_exp
Typed domains	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/typ.xsd	http://www.eba.europa.eu/xbrl/crr/dict/typ	eba_typ
Explicit domain members (domain CP)	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/cp/cp.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/CP	eba_CP
Families	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dim/fam.xsd	http://www.eba.europa.eu/xbrl/crr/dict/fam	eba_fam
Perspectives	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dim/pers.xsd	http://www.eba.europa.eu/xbrl/crr/dict/pers	eba_pers

7.1 Metrics

Metrics define the nature of the measure to be performed. Metrics determine the data type, the period type (instant / duration) plus additional semantics of their corresponding data points. Metrics are represented in XBRL as primary items.

All the contexts in an instance document are expected to include an xbrli:period element with the same value: the reference period[8] in the case of metrics of duration type, or the end of the reference period (for metrics of instant type). The variations from this reference period in certain data points are expressed with the Reference Period (RF) dimension. This approach has been introduced in order to overcome the difficulty of defining time constraints for multiple periods in the table and definition linkbases.

The local name of base items is composed of three parts:

- A letter that represents the data type in lower case (see data types table below):

Model data type	XBRL data type	Local name codification letter	Reporting unit
Monetary (currency)	xbrli:monetaryItemType	m	Adequate currency using ISO 4217 codification (e.g.: iso4217:EUR)
Percent	num:percentItemType	p	xbrli:pure
Decimal	xbrli:decimalItemType	p	xbrli:pure
Integer	xbrli:integerItemType	i	xbrli:pure
Date	xbrli:dateItemType	d	No unit
Boolean (true/false or 0/1)	xbrli:booleanItemType	b	No unit
Text	xbrli:stringItemType	s	No unit
Explicit domain	xbrli:qnameItemType	e	No unit
Typed domain	Domain corresponding data type, codification letter and reporting unit		

[8] Reference period is defined as the period that starts at the beginning of the accounting year and ends at the reference date.

- A letter that represents the period type (i: instant, d: duration).
- A number that corresponds to the numeric code in the model (no zero padding or predetermined length).

In the case of domain based data types, an additional attribute (*model:domain*) is included to identify the qualified name of the domain (explicit or typed). Where the acceptable set of values for such a metric is a subset of the full set of values within an explicit domain, an additional attribute (*model:hierarchy*) is included to identify the URI of the role of a hierarchy containing the acceptable subset of domain values.

The id of the element (necessary for XLink locators) is composed like this:

{opre}_{name}

Where {opre} represents the prefix of the base namespace of the owner of the base item and {name} represents the name described above. Some examples follow:

Owner	Data / period type	Code	Name	Id	Namespace	Prefix
EBA	Monetary / Instant	7	mi7	eba_mi7	http://www.eba.europa.eu/xbrl/crr/dict/met	eba_met
EBA	Text / Instant	7	si7	eba_si7	http://www.eba.europa.eu/xbrl/crr/dict/met	eba_met
BdE	Boolean / duration	3	bd3	es_bd3	http://www.bde.es/xbrl/dict/met	es_met
BdE	Monetary / duration	7	md7	es_md7	http://www.bde.es/xbrl/dict/met	es_met

7.2 Dimensions

Dimension items are represented in XBRL as XDT dimensions. The local name of each dimension corresponds to its code in the model: a short sequence of capital case letters (usually two, but it is not limited to two letters).

The id of the element (necessary for XLink locators) is composed like base items:

{opre}_{name}

Where {opre} represents the prefix of the base namespace of the owner of the dimension and {name} represents the name described above. Some examples follow:

Owner	Code	Name	Id	Namespace	Prefix
EBA	CP	CP	eba_CP	http://www.eba.europa.eu/xbrl/crr/dict/dim	eba_dim
EBA	MC	MC	eba_MC	http://www.eba.europa.eu/xbrl/crr/dict/dim	eba_dim
BdE	DPC	DPC	es_DPC	http://www.bde.es/xbrl/dict/dim	es_dim
BdE	XP	XP	es_XP	http://www.bde.es/xbrl/dict/dim	es_dim

Dimension schemas include a reference to a definition linkbase whose file name is "dim-def.xml" and is placed in the same folder as the schema file. This linkbase includes the following information about explicit dimensions:

- Reference to the domain associated to the dimension by means of a dimension-domain relationship (with xbrldt:usable attribute equal to false).
- Reference to the default member of that dimension by means of a dimension-default relationship. Note that though the model defines default members at domain level, the dimensions XBRL specification establishes this relationship at dimension level. Thus, each dimension using a domain with a default member must include this relationship.

These relationships are defined in an extended whose role is the standard one (http://www.xbrl.org/2003/role/link).

7.3 Domains

Explicit domains are represented using XBRL abstract items of domain type ("*model:explicitDomainType*") in the schema file ("*exp.xsd*"). 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*")[9].

The local name of each domain corresponds to its code in the model model ({dom-code}): a short sequence of capital case letters (usually two, but not limited to two letters). The id of the element (necessary for XLink locators) is composed like base items:

{opre}_{name}

Where {opre} represents the prefix of the base namespace of the owner of the domain and {name} represents the name described above. Some examples follow:

[9] Explicit domains are xbrli:items 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. As some tools in the market do not support a single file with two different extended links, these items have been split into two different schemas.

Owner	Code	Element Name	Type	Id	Namespace	Prefix
EBA	CO	CO	Explicit	eba_CO	http://www.eba.europa.eu/xbrl/crr/dict/exp	eba_exp
EBA	MI	MI	Typed	eba_MI	http://www.eba.europa.eu/xbrl/crr/dict/typ	eba_typ
BdE	DPC	DPC	Explicit	es_DPC	http://www.bde.es/xbrl/dict/exp	es_exp
BdE	AP	AP	Typed	es_AP	http://www.bde.es/xbrl/dict/typ	es_typ

Though the namespace of explicit and typed domains is different, different local names should be used to avoid any confusion.

7.3.1 Explicit domain members and hierarchies

Explicit domain members are represented using XBRL abstract items of domain item type ("*domainItemType*" is defined in the non numeric set of types of XII's type registry). The default domain member of a domain (usually the one with code 0) is marked with an attribute: *model:isDefaultMember* = "true".

The local name of each explicit domain member corresponds to its numeric code in the model preceded by a lower case "x"[10]. If the concept represented has already a widely accepted standard codification, like ISO codes, the local name will match the existing codification in lower case. More specifically, the following ISO codes are used:

- ISO 4217: standard currency codes composed of three alphabetical characters
- ISO 3166-1 alpha-2: standard country codes composed of two alphabetical characters

The id of explicit domain members follows the general rule:

{opre}_{name}

The schema file that represents explicit members is placed in a folder with the name of its corresponding domain. The schema file for explicit domain members is called "mem.xsd":

Owner	Domain code	Domain members schema	Namespace	Prefix
EBA	CO	http://www.eba.europa.eu/xbrl/crr/dict/dom/co/mem.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/CO	eba_CO
EBA	MI	http://www.eba.europa.eu/xbrl/crr/dict/dom/mi/mem.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/MI	eba_MI
BdE	AP	http://www.bde.es/xbrl/dict/dom/ap/mem.xsd	http://www.bde.es/xbrl/dict/dom/AP	eba_AP

[10] Local names are XML schema tokens and thus, are not allowed to start with a numeric character.

Hierarchies are represented using XBRL extended link roles whose role is built following this pattern:

{ons}/role/dict/dom/{dom-code}/{hierarchy-code}

Where {ons} represents the namespace of the owner, {dom-code} represents the code of the domain and {hierarchy-code} the numeric code of the hierarchy. The id of these roles is composed following the pattern:

{opre}_r{code}

Owner	Domain code	Hierarchy Code	Role	Id
EBA	CO	1	http://www.eba.europa.es/xbrl/crr/role/dict/dom/CO/1	eba_r1
EBA	MI	1	http://www.eba.europa.es/xbrl/crr/role/dict/dom/MI/1	eba_r1
BdE	DCP	1	http://www.bde.es/xbrl/role/dict/dom/DCP/1	es_r1
BdE	AP	5	http://www.bde.es/xbrl/role/dict/dom/AP/5	es_r5

The schema file that represents hierarchies is placed in the same folder as members and it is called “hier.xsd”:

Owner	Domain code	Hierarchies schema	Namespace	Prefix
EBA	CO	http://www.eba.europa.eu/xbrl/crr/dict/dom/co/hier.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/CO/hier	eba_CO_h
EBA	MI	http://www.eba.europa.eu/xbrl/crr/dict/dom/mi/hier.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/MI/hier	eba_MI_h
BdE	AP	http://www.bde.es/xbrl/dict/dom/ap/hier.xsd	http://www.bde.es/xbrl/dict/dom/AP/hier	eba_AP_h

In addition to labels, these schemas include three additional linkbases with information about hierarchies:

- A presentation linkbase (hier-pre.xml), which represents the hierarchical disposition of members in hierarchies using parent-child relationships.
- A definition linkbase (hier-def.xml), which enables the inclusion of the members of a hierarchy in dimensional combinations using domain-member relationships.
- 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 or equal than the addition of its child members in the hierarchy: partial-breakdown relationships.
- A member is less or equal than the addition of its child members in the hierarchy: superset-breakdown relationships.

These arc roles are defined in the model schema:

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

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 EBA by Banco de España, we would have:

Code	Extending domain members schema	Namespace	Prefix
CO	http://www.bde.es/xbrl/dict/dom/eba_co/mem.xsd	http://www.bde.es/xbrl/dict/dom/eba_CO	es_eba_CO
AP	http://www.bde.es/xbrl/dict/dom/eba_ap/mem.xsd	http://www.bde.es/xbrl/dict/dom/eba_AP	es_eba_AP

These arcs (calculation arcs) include a weight attribute to indicate whether the child member contributes to the aggregation positively (+1) or negatively (-1). The roles that represent these calculation relationships are defined in the schema that supports the model. The root member of the definition and presentation relationship networks is the domain item defined in the schema.

7.3.2 Families and perspectives

Neither families nor perspectives are used in the consultation taxonomy.

8 Reporting requirements layer

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

All the files that correspond to this layer are placed under the folder "*fws*" in the official location of its owner. 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.

8.1 Frameworks

Frameworks are public elements represented using XBRL abstract items of framework type ("*model:frameworkType*") in the schema file "*fws.xsd*". The local name of each framework element corresponds to its code in the model and its id follows the general pattern.

Schema property	Value
Official location	{oloc}/fws/fws.xsd
Target namespace	{ons}/fws
Target namespace prefix[11]	{opre}_fws
Element local name	{framework }
Element id	{opre}_{framework }

In the case of the EBA:

Schema property	Value
Official location	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/fws.xsd
Target namespace	http://www.eba.europa.eu/xbrl/crr/fws
Target namespace prefix	eba_fws
Local names	finrep, corep
Element ids	eba_finrep, eba_corep

[11] Target namespace prefixes are not strictly necessary. Moreover, schemas like frameworks define names that are not used in the exchange of information between supervisors and supervised entities. However, as some XBRL tools raise warnings whenever they find a schema with no prefix defined. So, prefixes have been included to avoid misleading the users of these tools.

Each framework has a folder where the files of its taxonomies are placed. This folder has the name of its code in the model:

Description	Framework folder
Common Reporting	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep
Financial Reporting	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/finrep

8.2 Taxonomies

Taxonomies are public elements represented using XBRL abstract items of 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 and another subfolder with the date of its version[12], using the ISO 8601 codification.

Thus, the file "tax.xsd" includes a single element. Its local name corresponds to its code in the model and its id uses the general pattern:

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

To facilitate the specification of additional taxonomy resources, we will refer by {taxonomy-loc} to the URL *"{oloc}/fws/{framework}/{normative}/{vers-date}*" and by {taxonomy-ns} to the URI "{ons}/fws/{framework}/{normative}/{vers-date}".

The taxonomy folders in the consultation taxonomy are:

Description	Taxonomy folder
Common Reporting	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/its-2013-02/2013-09-03
Financial Reporting	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/finrep/its-2013-02/2013-09-03

[12] Ideally, this version date should correspond to the date where the corresponding normative is published or the date when a new version is released.

The folder of a taxonomy includes three folders for tables (*tab*), modules (*mod*) and validations (*val*).

8.3 Tables

The table folder includes a schema file (tab.xsd), a generic linkbase with the hierarchy of table groups and tables (tab-pre.xml) and a label linkbase for table groups (tab-lab-en.xml). The schema includes the definition of table groups (if any), which are represented using XBRL abstract items of table group type ("*model:tableGroupType*"). Its name is composed by adding the prefix "tg" to the code in the model. The linkbase with the hierarchy of tables is not referenced in schema; otherwise, all the modules defined in a taxonomy would include indirect links to all the tables in the taxonomy.

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}_{local-name}

Arcs with role "group-table" are used to establish the link between a table group and other table groups or tables in the presentation linkbase. This arc role is defined in the schema that supports the model.

Table groups are used to link numerous tables resulting from normalization of templates or if an original templates is composed by two or more physical tables. In other words, table groups represent those templates that consist of more than one table. In addition table groups are used more generally to group related tables into a subject area, for example Capital Adequacy or Credit Risk.

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:

Schema property	Value
Official location	{taxonomy-loc}/tab/{table}/{table}.xsd

Target namespace	{taxonomy-bns}/tab/{table}
Target namespace prefix	{opre}_tab_{table}
Element local name	N/A (elements defined as resources in linkbases)
Element id	{opre}_{table} (element defined as a resource in the rendering linkbase)

In addition to label linkbases, this schema includes a table linkbase ({table}-rend.xml) and a definition linkbase ({table}-def.xml).

The table linkbase includes the definition of the table according to the last table specification released. The relationships of each table are placed in an extended link whose role is built following this pattern:

{ons}/role/fws/{framework}/{normative}/{pub-date}/tab/{table}

In this linkbase, the different components of tables are represented using resources. The "id" of these resources is based on the code of the model plus a prefix to obtain a unique code in the context of the linkbase file:

Model class	Table linkbase resource	Id
Table	table	{opre}_t{code}
Predefined axis	ruleAxis (abstract = true)	{opre}_a{code}
Variable axis	filterAxis	{opre}_a{code}
Coordinate	ruleAxis	{opre}_c{code}
Base items hierarchy reference	conceptRelationshipAxis	{opre}_h{code}
Dimension hierarchy reference	dimensionRelationshipAxis	{opre}_h{code}

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

The definition linkbase 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 the algorithm described in Annex 1.

Each extended link role contains a set of primary items and a single hypercube[13]. In case of multiple primary items, the first one will be used to group the rest and reduce the number of "all" arcs. The domain element will be used as target of dimension-domain arcs to avoid cycles. The @xbrldt:targetRole attribute might be necessary in the case of hypercubes with dimensions sharing the same domain.

The roles of the extended links necessary to express these combinations are built adding numeric suffixes to the role previously defined for the table. For example:

{ons}/role/fws/{framework}/{normative}/{pub-date}/tab/{table}/1

{ons}/role/fws/{framework}/{normative}/{pub-date}/tab/{table}/2

...

The label linkbase file for a table contains labels for Table Linkbase nodes. In addition to the standard label, a table:table node, also contains a documentation label which defines a code to be used on filing indicators (see next section of this document).

The link between table groups and individual tables is established in the tab-pre.xml linkbase file as well as in linkbase files of modules (as described below).

8.4 Modules

Modules are represented using XBRL abstract items of module type ("*model:moduleType*"). Each module is stored in a different schema file whose name module file is the same as the code of the module in the model plus the extension ".xsd". These schema files imports the schemas of all the tables imported by that module:

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

In addition to label linkbases, each module includes a presentation linkbase ("{module}-pre.xml") where the relationship between modules and tables / table

[13] The model schema includes a hypercube element to be used. There is no need to define hypercube elements in each table or taxonomy.

groups is expressed using group-table arcs whose source is the module element and target is the table / group of tables element. Furthermore, table groups link to individual tables via a group-table relation.

The module schema also imports the formula linkbases and optionally, the linkbases with the preconditions on filing indicators.

Modules in the consultation taxonomy serve as entry points, defining the potential tables in each individual instance file that can be reported. As such there are five conceptual modules:

Conceptual Module	Description
corep	Common reporting own funds and leverage
corep_le	Common reporting large exposures
corep_lcr	Common reporting liquidity coverage ratio
corep_nsfr	Common reporting net stable funding ratio
finrep	Financial reporting

To aid practical implementations of reporting scheduling, the modules (and hence the schemaRef values of instance files) also indicate the consolidation approach and accounting standard used to prepare figures, so that there are the following modules:

Module	Description
corep_ind	CoRep own funds and leverage, individual basis
corep_conpru	CoRep own funds and leverage, consolidated on prudential basis
corep_le_ind	CoRep large exposures, individual basis
corep_le_conpru	Common reporting large exposures, consolidated on prudential basis
corep_lcr_ind	CoRep liquidity coverage ratio, individual basis
corep_lcr_conpru	CoRep liquidity coverage ratio, consolidated on prudential basis
corep_nsfr_ind	CoRep net stable funding ratio, individual basis
corep_nsfr_conpru	CoRep net stable funding ratio, consolidated on prudential basis
finrep_ind	Financial reporting, individual basis, any accounting standard
finrep_ind_gaap	Financial reporting, individual basis, national GAAP

finrep_ind_ifrs	Financial reporting, individual basis, IFRS
finrep_conacc	Financial reporting, consolidated on accounting basis, any accounting standard
finrep_conacc_gaap	Financial reporting, consolidated on accounting basis, national GAAP
finrep_conacc_ifrs	Financial reporting, consolidated on accounting basis, IFRS
finrep_conpru	Financial reporting, consolidated on prudential (CRR) basis, any accounting standard
finrep_conpru_gaap	Financial reporting, consolidated on prudential (CRR) basis, national GAAP
finrep_conpru_ifrs	Financial reporting, consolidated on prudential (CRR) basis, IFRS

Each of these modules contains validation rules restricting the descriptive values of table 00.01 to appropriate values.

8.5 Filing indicators

Filing indicators serve the purpose of communicating the scope of the reported data based on templates. The main purposes of filing indicators are to:

- provide hints to applications using the taxonomy, when processing instance files, on which templates are included in the filing and, for example, shall be displayed to users,

- trigger 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 defined 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 in every taxonomy module. Throughout this document, the prefix *"find"* will be used to make reference to this schema 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 a module, it is assumed that a filing contains no information on this template. In some case 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 template with code C_01.00 and no information (explicitly stated) on template C_07.00:

```
<find:fIndicators>
  <find:filingIndicator contextRef=”ctx”>C_01.00</find:filingIndicator>
  <find:filingIndicator contextRef=”ctx” filed=”false”>C_07.00</find:filingIndicator>
</find:fIndicators>
```

Contexts to which facts representing find:filingIndicator element refer must identify the reporting entity and use the end date of the reporting period as the instant date.

Identification of templates on find:filingIndicator facts is made using codes. These codes are the documentation label of a table:table resource (in case a template is reflected by a single individual table) or the common label of a set of tables.

8.6 Validation rules

8.6.1 Assertion patterns

Validations are expressed using XBRL assertions. Assertions are be identified by a unique code, which is the same as that used to identify the corresponding validation rule expressed in the ITS documentation.

There are several common patterns of validations implemented in the taxonomy, explained hereafter, which are:

- Hierarchy checks (Dimensional aggregation)
- Sign checks
- “Manual” or general value checks
- Enumerated value checks
- Module specific value restriction checks

It will be the case that some (most) assertions are not applicable to all modules of a taxonomy. Each entry point will include, in its DTS, all assertions that are applicable in its context.

Each assertion is associated to a description, appearing as a label, which indicates which check is performed, in business / form-centric terms.

Each assertion may also, in future taxonomies, be associated to two attributes: *model:fromDate* and *model:toDate* which may be used to express a period of validity, in term of reporting date ("as of").

8.6.1.1 Hierarchy checks (Dimensional Aggregation)

Derived from information in the data point model, the Hierarchy check (dimensional aggregation) pattern corresponds to the validation of an aggregation of a business concept, or a set of business concepts, along a dimension. In other words the rolling up of component parts of a breakdown along a particular aspect.

These rules have the suffix "_h", e.g. v0150_h. This rule, expressed in the ITS as "Table: C 02.00, Column: 010, Formula: {r490} = +{r500} + {r510}", is derived from the hierarchy with code PL2, which indicates a (fairly obvious) relationship between three possible values for the Portfolio dimension:

```
Banking and trading book =   Banking book
                           + Trading book
```

These three different values for the Portfolio dimension are the distinguishing factor of rows 490, 500 and 510 on table C02.00, so this validation rule asserts that these rows should be related in the way the hierarchy indicates.

8.6.1.2 Sign checks

Many cells (data points) to be reported are required to be positive numbers or amounts (and conversely many are required to be negative). Where this is the case this is enforced using sign check assertions, with the suffix "_s", which are also derived from information in the DPM.

E.g. v2468_s, which checks that the values in column 050 and rows 010, 020 and 090 of table C 05.02 are negative (or zero).

Note that where a range of both rows and columns are checked for a particular sign, the table centric formula of these rules may initially appear strange, e.g. v2028_s "F 46.00 (r010;040;210, c090;110) : {F 46.00} <=0". This does not indicate, as the formula might suggest at first glance, that the table as whole is somehow less than or equal to zero, but that the (six) cells at the intersections of rows 010,040 and 210 and columns 090 and 110 must be.

8.6.1.3 "Manual" or general value checks

Moving beyond the information captured in a structured form in the DPM, and the validation rules that can be inferred from it, there are many additional business checks between data points. These have been specified individually by subject matter experts, have the suffix "_m", and involve a wide variety of formulae, e.g. v0219_m "{C 03.00, r020,c010} = {C 01.00, r020,c010} - {C 02.00, r010,c010} * 4.5%", or v0284_m "{C 06.00, c180} >= {C 06.00, c200}"[14].

8.6.1.4 Enumerated value checks

Data type checks simply ensure that submitted data is of the correct nature, i.e. that monetary values are entered when requested, or that if a value is supposed to be a percentage that it is entered as a number between 0 and 1[15].

---

[14] Or even v1037_m "sum({F 31.01, r120, (c010-050)}) <= {F 10.00, r290,c030} - sum({F 10.00, c030, (r050-060, r110-120, r170-180)}) + {F 11.01, r500,c030} - sum({F 11.01, c030, (r040-050, r090-100, r140-150, r270-280, r320-330, r370-380)}) + {F 11.02, r230,c010} - sum({F 11.02, c010, (r040-050, r090-100, r140-150)})" !

[15] In the XBRL instance files, percentages should be represented as a decimal number between 0 and 1, with four decimal digits.

These rules have the suffix "_t", e.g. v2685_t which specifies "[Type of securitisation] IN {[Securitisation],[Re-securitisation]]"[16] , i.e. that there are only two allowed values for the "Type of securitisation" metric.

This data type information is conveyed in the DPM in a structured form (linking the metric to the domain and hierarchy from which its values must be drawn), and as such these validation rules are not present in the ITS validation rule list. They are present in the taxonomy merely as technical artefact required in XBRL to enforce the restriction to the appropriate values, since the underlying data type of the metric in XBRL is simply a xbrli:qnameItemType. At present XBRL lacks a standardised mechanism to more precisely define the allowed values for such an item, so it must be enforced post-hoc in validation using assertions[17].

8.6.1.5 Module specific value restriction checks

As described in section, some or all of the values in the general information table 00.01 are determined by the module a particular instance represents (i.e. the schemaRef used). Each module includes validation rules (or more precisely, includes parameters that control the execution of validation rules) to ensure these data points are consistent with the nature of the module.

These rules have the suffix "_r", e.g. v2710_r "{F 00.01, r010 , c010 } = (IFRS)". Again these rules are XBRL technical artefacts, not present in the ITS validation rule list.

8.7 Assertion sets

Validations are grouped into assertion sets that correspond to the tables they are to be applied. In the context of a table, not reported or nil numeric values will be assumed to be zero; consequently, fallback values are used in their corresponding assertion definitions.

The link between an assertion set and the table (or tables[18]) it applies is represented using applies-to-table arcs from the assertion set to the resource that corresponds to the table. The URI of this arc is http://www.eurofiling.info/xbrl/arcrole/applies-to-table

If an assertion applies to multiple tables individually or to multiple sets of tables, then it will be associated to different assertion sets.

Ex.#	*Assertion example (textual description)*	*Assertion sets*	*Tables*
1	$a > 0 (where $a represents data in table 1)	assertion set 1	table1
2	$a > 0 (where $a represents data in tables 1, 2 and 3)	assertion set 1	table1

[16] Or in the xpath of the XBRL assertion test "$a = (xs:QName('eba_UE:x14'), xs:QName('eba_UE:x15'))"

[17] see also the description of the *model:hierarchy* attribute in §7.1 for a non-XBRL standard extension used to indicate this information for any tooling that wishes to make use of it prior to/outside the validation stage, such as for data entry or display.

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

		assertion set 2	table 2
		assertion set 3	table 3
3	$a = $b (where $a represents data in table 1 whereas $b represents data in table 2)	assertion set 1	table 1 table 2
4	$a = $b (where in some cases, $a represents data in table 1 and $b data in table 2; in other cases, $a represents data in table 3 and $b represents data in table 4)	assertion set 1	table 1 table 2
		assertion set 2	table 3 table 4

Assertion sets resources might include the attributes fromDate and toDate to constraint the reference date where their associate assertions should be applied.

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.

8.7.1 Preconditions and filing indicator parameters

Each value assertion defined is associated to a precondition[19] 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 are 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 optimizations), 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).

[19] 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.

There is a filing indicators parameters defined for each table defined in the framework. These parameters are defined in the namespace of the filing indicators schema and have a name according to the following convention:

t{table-code}

where table-code represents the code of the corresponding table. Thus, the definition of one of these parameters would look like this:

```
<variable:parameter
       name="find:t{table-code}"
       select="//find:fIndicators/find:fIndicator = ‘{template-code}’"
       as="xs:boolean" …/>
```

Where ‘template-code’ represents the code of the template

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 or expression is composed of a sequence of and expressions on the tables involved:

```
“$find:t{c1.1} and $find:t{c1.2} and …

or $find:t{2.1} and $find:t{2.2} and …

or …”
```

Some examples:

Expression	***Explanation***
$find:t1	Assertion applies only to table 1
$find:t1 and $find:t2	Assertion crosses information between tables 1 and 2
$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)
$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.

8.7.2 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.

Wherever possible, value assertions will be used instead of existence assertions. The consultation taxonomy contains no existence assertions.

Though unlikely, there might be the case of validations that cannot be (effectively or efficiently) defined using value assertions. If such rules were required, the “id” of such assertions would follow a predefined naming convention to help applications not relying on validation sets to discard such evaluations:

Id for existence assertions: “e{code}”

Id for value assertions: “v{code}”

8.7.3 Interval Arithmetic

In order 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 and placed in the following location:

Namespace:

- http://www.eurofiling.info/xbrl/func/interval-arithmetics

Official location:

- http://www.eurofiling.info/eu/fr/xbrl/func/interval-arithmetics.xml

Some example functions are:

- iaf:numeric-equal(arg1, arg2): true if two values are equal or are within the tolerance interval derived from its reported precision.

- iaf:numeric-less-than(arg1, arg2): checks whether arg1 is less than arg2, considering their precision.

An entry point for these functions and additional ones that could be provided in the future is placed in the following location:

- http://www.eurofiling.info/eu/fr/xbrl/func/functions.xsd

Variables used are defined in no namespace; this way, there is a clear separation between variables and filing indicator parameters and the pivot-variable. The naming convention for variables is lower camel case notation.

8.7.4 Notation

Assertions are be identified by a unique code, to enables the identification of errors in a validation process with the corresponding definition. It must be noted that an XBRL assertion might produce several evaluations covering different sets of data points. Assertions might include a description and custom error messages, as defined by business experts.

Existence assertions shall only be used, where absolutely necessary, to detect errors in the case of data that should have been reported[20]. Whenever it is possible, value assertions shall be used instead of existence assertion, as the former enable more comprehensive error messages and makes possible the usage of preconditions on filing indicators.

The files that define assertions and assertion sets are grouped into files depending on their scope. These files are placed in the "val" folder of the corresponding taxonomy, together with files to define preconditions and filters[21] of common use shared by different assertions in the taxonomy and parameters:

Resource description	File location
Assertions location that apply to a single table (example 1)	{taxonomy-loc}/val/val-{tab1}.xml
Assertions location that apply to multiple tables individually (example 2)	{taxonomy-loc}/val/val-{tab1}.{tab2}.xml
Assertions location that cross information in a set of tables (example 3)	{taxonomy-loc}/val/val-{tab1}_{tab2}.xml
Assertions that cross information in a multiple sets of tables (example 4)	{taxonomy-loc}/val/val-{tab1}_{tab2}.{tab3}_{tab4}.xml
Assertion sets location that apply to a single table (example 1)	{taxonomy-loc}/val/aset-{tab1}.xml
Assertion sets location that apply to multiple tables	{taxonomy-loc}/val/aset-{tab1}.xml

[20] As noted the consultation taxonomy contains no existence assertions

[21] These filters and preconditions should be independent of the assertion they apply to, and thus, should not depend on the variables defined by specific assertions.

individually (example 2)	{taxonomy-loc}/val/aset-{tab2}.xml
Assertion sets location that cross information in a set of tables (example 3)	{taxonomy-loc}/val/aset-{tab1}_{tab2}.xml
Assertion sets that cross information in a multiple sets of tables (example 4)	{taxonomy-loc}/val/aset-{tab1}_{tab2}.xml {taxonomy-loc}/val/aset-{tab3}_{tab4}.xml
Parameters	{taxonomy-loc}/val/params.xml
Filters common to multiple assertions in the taxonomy	{taxonomy-loc}/val/filt.xml
Preconditions common to multiple assertions in the taxonomy	{taxonomy-loc}/val/prec.xml
Preconditions on filing indicators plus variable-set-precondition arcs	{taxonomy-loc}/mod/{module}-find-prec.xml
Filing indicators parameters	{taxonomy-loc}/val/find-params.xml

Any of these linkbases can have its corresponding set of label linkbases, following the convention defined in this document. In the cases of assertions, an additional set of linkbases might be included for error messages expressed in different languages:

{assertions-file}-err-{lang}.xml

or

{assertions-file}-err-{lang}-{country}.xml

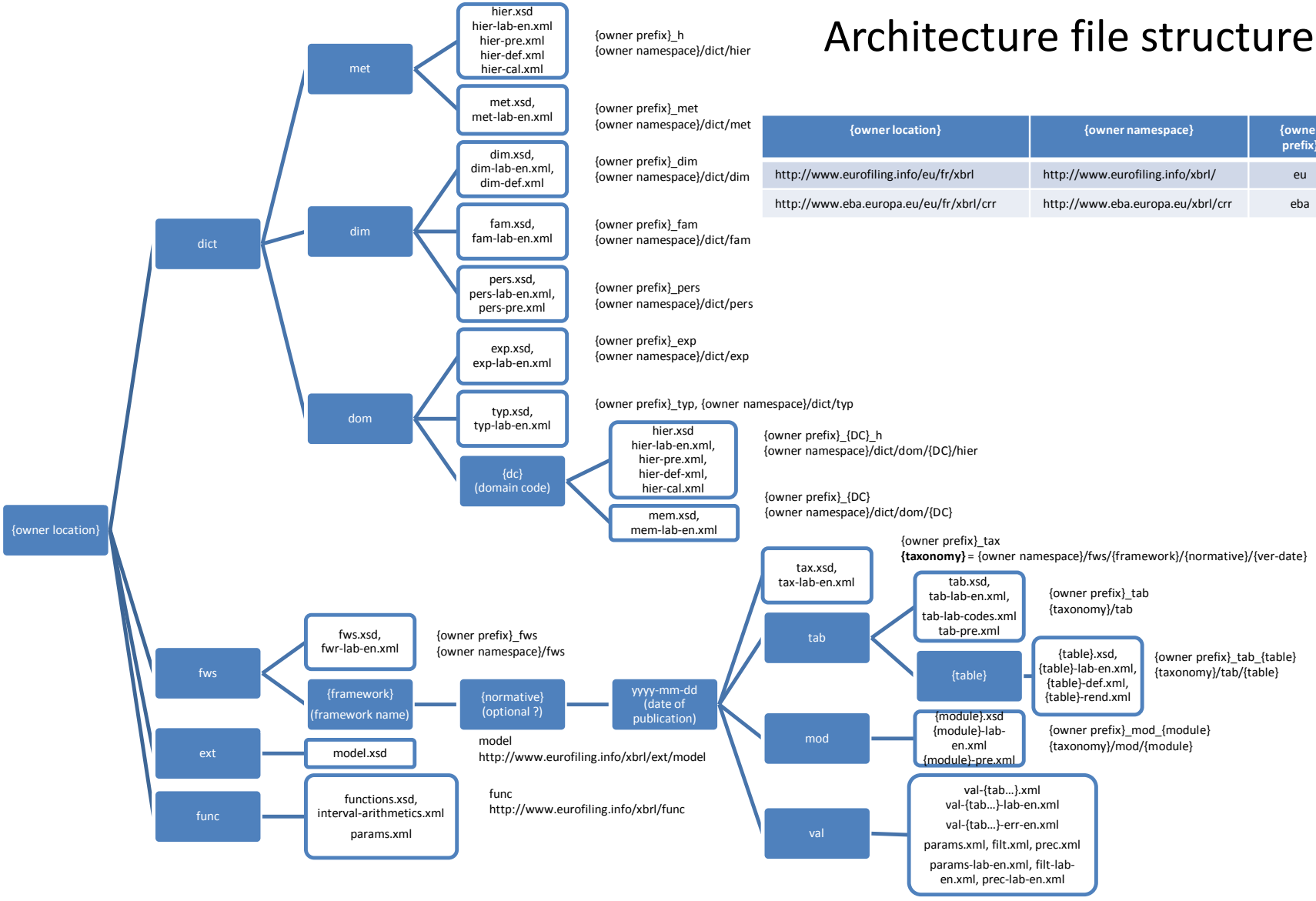
Where {assertions-file} corresponds to the name of the file with the assertions whose error message are described, without the extension.

These files will be included by the modules defined in the taxonomy.

9 Hypercubes

It is important to remark that the XBRL hypercubes in the taxonomy are validation artefacts (essentially just indicating grey cells) and should not be used by external systems for the automatic creation of database structures. The hypercubes in the taxonomy are generated automatically by an algorithm, and do not obey to any kind of business criteria. These hypercubes might be dramatically modified with any cfuture change to the reported information in a table, with the only consideration being the reduction of the final set of hypercubes and performing more efficiently with XBRL market tools.

Architecture file structure



{owner location}	{owner namespace}	{owner prefix}
http://www.eurofiling.info/eu/fr/xbrl	http://www.eurofiling.info/xbrl/	eu
http://www.eba.europa.eu/eu/fr/xbrl/crr	http://www.eba.europa.eu/xbrl/crr	eba