20/12/2012
Public release v0.1

Representation in XBRL of the Data Point Model

Changes

1	12/04/2012	- Initial version
2	17/04/2012	- Added table of changes - Minor wording and format fixes - Corrected type in the target namespace of typed domains. - Changed the terms base namespace and base prefix to owner's namespace and owner's prefix - Other changes and fixes suggested by BRAG - Removed schema items for hierarchies because of redundancy with extended link roles. Different schemas to enable standard and generic labels.
3	30/04/2012	- Included the date in the "changes" table - Included clarification in naming convention rules of domain members that represent concepts in other standards like ISO currencies. - Included references to ISO 4217 and ISO 3166-1 alfa-2 - Corrected typo on group-table arc
4	08/05/2012	- Corrected type on the target namespace prefix of tables - Added missing information about namespace, official location and prefix of modules. - Corrected the name of the presentation arc in modules. - Corrected type in the name of the type for modules - Added reference algorithm for the creation of compact hypercubes. - Correction of some typos and some minor rewording. - Given some consistency to the references of the location, namespace and prefix of the owner. - Added a reference to the hypercube element in the model.
5	08/05/2012	- Solution for the inconsistency of types and base items: amount types and base items are swapped.
6	21/05/2012	- Corrected typo on the id of tables
7	24/05/2012	- Clarifications after BRAG questions - Added Eurofiling domain and location for shared elements at European level. Corrected the location of the model accordingly. - Included possibility of target roles in dimensional relationships (necessary if more than one dimension with the same domain is included in the same hypercube) - Fixed the description of hierarchies and members - Included basic information about validations
8	06/06/2012	- Minor wording and format corrections - Added a comment to the paragraphs to be fixed in case the approach of removing all business meaning from primary items is taken.
9	21/06/2012	- Included some wording / format enhancements suggested by BdF

		- Included id for tables (though tables are resources in a linkbase, they need an id so that they can be referenced by labels and other resources). - Included description of the representation of row and column codes - Included graphical representation of the set of files - Moved the description of time constraints in table linkbases to its proper chapter (previously described in the dictionary of concepts). - Added the linkbase params.xml with the definitions necessary for time reference constraints
10	29/06/2012	- Replaced angle brackets by curly brackets in the file structure diagram - Added detail on the definition of assertion-sets in formula linkbase and the naming convention for files - Removed reference to framework properties information. These properties are not currently defined. If needed in the future, they can be added without causing any harm to existing taxonomies. Similar references were removed in previous version of the document. This one went under the radar. - Some format corrections - Added paragraph to clarify BdI's questions on hypercubes
11	16/10/2012	- Removed comments about solved issues - Updated to include filing indicators and preconditions - Updates to describe the mechanism to convert existence assertions into value assertions
public release v1	20/12/2012	- Accepted changes corresponding to the decision taken during the last XBRL subgroup - Added attribute to enable comments on filing indicators when no tables are filed.

Table of Contents

1 Introduction

This document establishes the rules applied to represent financial models defined using the DPM approach in terms of XBRL taxonomies.

The expected audience of this document are XBRL taxonomy authors and reviewers undertaking the development of EBA taxonomies or any NSA extension of these taxonomies. This document is also useful for developers of software that consumes XBRL taxonomies following this approach, or software that produces instance documents.

2 Supporting concepts

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

2.1 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. 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.

European Banking Authority	http://www.eba.europa.eu	http://www.eba.europa.eu/xbrl	eba
Eurofiling[1]	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 area covered by the institution, plus two fixed ones: "fr" (for financial reporting) and "xbrl":

European Banking Authority	http://www.eba.europa.eu/eu/fr/xbrl
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.

[1] For concepts shared with other European supervisors

2.2 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.

2.3 Namespaces

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

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

3 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:

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[2]	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:

{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

[2] 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.

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

4 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:

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:

Metrics	http://www.eba.europa.eu/eu/fr/xbrl/dict/met/met.xsd	http://www.eba.europa.eu/xbrl/dict/met	eba_met
Dimensions	http://www.eba.europa.eu/eu/fr/xbrl/dict/dim/dim.xsd	http://www.eba.europa.eu/xbrl/dict/dim	eba_dim
Explicit domains	http://www.eba.europa.eu/eu/fr/xbrl/dict/dom/exp.xsd	http://www.eba.europa.eu/xbrl/dict/exp	eba_exp
Typed domains	http://www.eba.europa.eu/eu/fr/xbrl/dict/dom/typ.xsd	http://www.eba.europa.eu/xbrl/dict/typ	eba_typ

Explicit domain members (domain CP)	http://www.eba.europa.eu/eu/fr/xbrl/dict/dom/cp/cp.xsd	http://www.eba.europa.eu/xbrl/dict/dom/CP	eba_CP
Families	http://www.eba.europa.eu/eu/fr/xbrl/dict/dim/fam.xsd	http://www.eba.europa.eu/xbrl/dict/fam	eba_fam
Perspectives	http://www.eba.europa.eu/eu/fr/xbrl/dict/dim/pers.xsd	http://www.eba.europa.eu/xbrl/dict/pers	eba_pers

4.1 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.

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):

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		

- 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).

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:

EBA	Monetary / Instant	7	mi7	eba_mi7	http://www.eba.europa.eu/xbrl/dict/met	eba_met
EBA	Text / Instant	7	si7	eba_si7	http://www.eba.europa.eu/xbrl/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

4.2 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:

EBA	CP	CP	eba_CP	http://www.eba.europa.eu/xbrl/dict/dim	eba_dim
EBA	MC	MC	eba_MC	http://www.eba.europa.eu/xbrl/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).

A special dimension, the **base item**, includes an additional attribute (@balance) to specify the balance nature of the financial concept represent (credit or debit).

4.2.1 Families

Families are placed in the same folder as dimensions. Families are represented as XBRL abstract items of family type (“*model:familyType*”). The local name of each family corresponds to its numeric

code in the model preceded by a lower case “f”. The id of family elements is composed like base items:

{opre}_{name}

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

EBA	1	f1	eba_f1	http://www.eba.europa.eu/xbrl/dict/fam	eba_fam
EBA	2	f2	eba_f2	http://www.eba.europa.eu/xbrl/dict/fam	eba_fam
BdE	1	f1	es_f1	http://www.bde.es/xbrl/dict/fam	es_fam
BdE	5	f5	es_f5	http://www.bde.es/xbrl/dict/fam	es_fam

4.2.2 Perspectives

Perspectives are placed in the same folder as dimensions and families. Perspectives are represented as extended link roles that are used in presentation linkbases, where the relationships between dimensions and families are formalized.

The name of these roles is built according to the following pattern:

{ons}/role/dict/pers/{code}

Where {ons} represents the base namespace of the owner and {code} represents the numeric code given to the perspective in the model.

The id of “roleType” elements is composed similarly to other concepts of the dictionary:

{opre}_p{code}

Where {opre} represents the prefix of the base namespace of the owner of the perspective and {code} represents the numeric code of the perspective in the model. Some examples follow:

EBA	1	http://www.eba.europa.eu/xbrl/role/dict/pers/1	eba_p1
EBA	2	http:// www.eba.europa.eu/xbrl/role/dict/pers/2	eba_p2
BdE	1	http://www.bde.es/xbrl/role/dict/pers/1	es_p1
BdE	5	http://www.bde.es/xbrl/role/dict/pers/5	es_p5

The schema of perspectives imports the schemas of families and dimensions, and includes a generic linkbase with their labels and a presentation linkbase named “pers-pre.xml”, both in the same folder. The presentation linkbase includes extended links that correspond to each perspective. The arcs in these extended links have families as source and dimensions as target nodes.

4.3 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*")[3].

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:

EBA	CO	CO	Explicit	eba_CO	http://www.eba.europa.eu/xbrl/dict/exp	eba_exp
EBA	MI	MI	Typed	eba_MI	http://www.eba.europa.eu/xbrl/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.

4.3.1 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"[4]. 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":

EBA	CO	http://www.eba.europa.eu/xbrl/dict/do	http://www.eba.europa.eu/xbrl/	eba_

[3] 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.

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

		m/co/mem.xsd	dict/dom/CO	CO
EBA	MI	http://www.eba.europa.eu/xbrl/dict/dom/mi/mem.xsd	http://www.eba.europa.eu/xbrl/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

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}

EBA	CO	1	http://www.eba.europa.es/xbrl/role/dict/dom/CO/1	eba_r1
EBA	MI	1	http://www.eba.europa.es/xbrl/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":

EBA	CO	http://www.eba.europa.eu/xbrl/dict/dom/co/hier.xsd	http://www.eba.europa.eu/xbrl/dict/dom/CO/hier	eba_CO_h
EBA	MI	http://www.eba.europa.eu/xbrl/dict/dom/mi/hier.xsd	http://www.eba.europa.eu/xbrl/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:

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:

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.

5 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.

5.1 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.

Official location	{oloc}/fws/fws.xsd
Target namespace	{ons}/fws
Target namespace prefix[5]	{opre}_fws
Element local name	{framework }
Element id	{opre}_{framework }

In the case of the EBA:

Official location	http://www.eba.europa.eu/eu/fr/xbrl/fws/fws.xsd
Target namespace	http://www.eba.europa.eu/xbrl/fws
Target namespace prefix	eba_fws
Local name example	finrep
Element id example	eba_finrep

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

5.2 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[6], 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:

[5] 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.

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

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

5.2.1 Tables

The table folder includes a schema file (tab.xsd) that references 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.

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.

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:

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:

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. In the case of duration type metrics, the size of the period to be reported is constrained using time aspect rules in the context of a table. The default value is "reference period[7]"; other possible values are month, quarter, year… In case of data points that correspond to an instant or period of time prior to the reference period, temporal aspect rules include a temporal offset relative to the reference date.
The reference date and the reference period shall be defined in terms of two parameters:

- refPeriodEndDate: reference date and end date of the reference period
- refPeriodStartDate: starting date of the reference period

These two parameters are defined in the linkbase:

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

Reference date	instant = refPeriodEndDate
Reference period	duration.end = refPeriodEndDate duration.start = refPeriodStartDate
Beginning of the reference period	instant = refPeriodStartDate
A quarter ending at the reference date	duration.end = refPeriodEndDate duration.start = refPeriodEndDate - P3M + P1D
Previous quarter	duration.end = refPeriodEndDate - P3M duration.start = refPeriodEndDate - P6M + P1D

Time reference examples

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

[8] For the sake of simplicity, the actual aspect rules syntax is not included. As an example, "A quarter ending at the reference date" using the syntax of the the working draft released in December 2011 would be defined like this:

```
<formula:period>
        <formula:duration
            end="$refPeriodEndDate"
            start="$refPeriodEndDate – xs:yearMonthDuration('P3M') + xs:dayTimeDuration('P1D')" />
</formula:period>
```

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[9]. 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
...

5.2.2 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:

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 groups is expressed using group-table arcs whose source is the module element and target is the table / group of tables element.
The module schema also imports the formula linkbases and optionally, the linkbases with the preconditions on filing indicators.

5.2.3 Validation rules

Validations are expressed using XBRL assertions. In order to handle the error margin caused by the imprecision of input data, assertions can 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

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

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 (see below). The naming convention for variables is lower camel case notation. Whenever an expression involves a certain number of facts used uniformly, sequence variables shall be used in order to improve the readability of the expression and the performance of the processor.

5.2.3.1 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 shall be used in their corresponding assertion definitions.
The link between an assertion set and the table (or tables[10]) 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 as follows:

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.

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
		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.

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

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, at the time of the writing of this document, 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.

5.2.3.2 Filing indicators

Filing indicators are facts included as part of an instance document where the filer estates which tables are being reported. Each table is represented as an instance fact of the item "table" under the "fIndicators" element. These elements 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.* Throughout this document, the prefix "*find*" will be used to make reference to this schema namespace.

The following instance excerpt represents a filing with information about tables with code 100 and 200:

```
<find:fIndicators>
        <find:table contextRef="ctx">100</find:table>
        <find:table contextRef="ctx">200</find:table>
</find:fIndicators>
```

This approach enables a clear separation between business facts (under the xbrli:xbrl element) and additional data required in the reporting process. Moreover, it does not require the addition of filing indicators to the dictionary of concepts, as filing indicators are defined in a generic way in its own schema.

Filing indicators include a Boolean attribute named "find:filed" with default value "true". Tables not filed can be omitted from the list of filing indicators or have an entry with the "find:filed" attribute equal to "false". The latter is especially useful if a footnote explaining the reasons for not filing a table want to be included.

5.2.3.3 Preconditions and filing indicator parameters

Each value assertion defined is associated to a precondition[11] on filing indicators. To avoid XBRL instance syntactic dependencies[12], 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

[11] Assertions might have additional precondition as required by the logic of the assertion to be performed. But these additional preconditions do not depend on filing indicators.
[12] This approach is quite convenient, as there is a new line of work in the Formula WG to define a subset of validations considered to be portable. This will enable the possibility of having XBRL processors performing validations on data in a database, rather than information on a single XML document.

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).
Filing indicators 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:table = '{table-code}'" as="xs:boolean" …/>
```

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:

$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.

5.2.3.4 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.[13]

However, most existence assertions can be re-defined as value assertions using the following pattern:

- The assertion includes a "pivot-variable": a fact variable that matches data in the instance document known to be reported always. The variable uses aspect cover filters to avoid any interference with the rest of variables. This variable is defined once as a sequence variable that matches the filing indicators.

[13] Preconditions are not a mechanism designed to control the evaluation of assertions, but to restrict the set of data where a validation is applied. In fact, there is a new line of work in the Formulae WG to add preconditions at variable-set level. This way, preconditions could be used to control the evaluation of the assertions in the scope of a certain assertion set. However, this new and better approach is not expected to be ready in time for the release of EBA taxonomies. Nevertheless, the approach proposed in this document is ready to be adapted to the new standard in future releases of EBA taxonomies.

- The rest of 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.* The pivot variable is linked to the assertion using the name "pvar:pivot" in the pivot-variable namespace.

Though unlikely, there might be the case of validations that cannot be defined using value assertions. For instance, checking that the number of rows (in a table with a variable number of rows) that verify a certain condition on several columns is greater than a certain number can be easily implemented using an existence assertion. The equivalent rule using a value assertion is extremely contrived and will have a very negative impact on the performance of the processor. If such a kind of rule were required, it should be implemented using an existence assertion. The "id" of such assertions follows 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}"

5.2.3.5 Notation

Assertions will be identified by a unique code, so that it 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 to detect errors in the case of data that should have been reported. Whenever it is possible, value assertion 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[14] of common use shared by different assertions in the taxonomy and parameters:

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

[14] 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.

location that cross information in a multiple sets of tables (example 4)	
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}/val/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.

Annex 1. Compact hypercubes production algorithm

This chapter describes the way a small number of hypercubes can be produced from an initial set of unicellular hypercubes using an efficient algorithm. This has the benefit of a reduction in the size of the taxonomies obtained and a better performance in the dimensional validation process.

The algorithm is defined in terms of dimensions in a generic way. This means that primary items and typed[15] dimensions must be handled in a similar way. The algorithm starts from an initial set of unicellular hypercubes[16]. Each iteration replaces some sets of hypercubes with a single one and leaves some hypercubes unmodified. The resulting hypercubes of an iteration are the input to the next one.

The algorithm performs a single iteration per dimension in the table, starting with the dimension with a highest number of different members and finishing with the one with lowest number[17]. Let's use the notation $\{D_1, D_2 \dots D_n\}$ to refer to the dimensions in the order they are processed by the algorithm and $\{d_{i,1}, d_{i,2} \dots d_{i,n}\}$ to the members of dimension "i" used in the table.

Given the hypercubes h_1, h_2, $h_3 \dots h_m$, during the i^{th} iteration, they are grouped by equal values of all dimensions, except for D_i. For example:

Group	Hypercube	Dimensions / members					
		D_1	D_2	D_3	D_i	…	D_n
G1	H1	$d_{1,1}$,$d_{1,2}$	$d_{2,3}$	$d_{3,7}$,$d_{3,9}$	$d_{i,1}$	…	$d_{n,1}$
	H2				$d_{i,4}$		
G2	H3	$d_{1,2}$	$d_{2,2}$	$d_{3,4}$	$d_{i,5}$	…	$d_{n,6}$
G3	H4	$d_{1,1}$,$d_{1,2}$,$d_{1,5}$	$d_{2,2}$, $d_{2,3}$	$d_{3,5}$	$d_{i,3}$	…	$d_{n,2}$
	H5				$d_{i,3}$,$d_{i,5}$		
	H6				$d_{i,7}$		

Each group that contains more than one hypercube is replaced by a single hypercube whose dimensions are assigned to the members of the hypercubes in the group. The dimension D_i is assigned to the union of the members of D_i of each hypercube in the group. In the previous example:

Group	Hypercube	Dimensions / members					
		D_1	D_2	D_3	D_i	…	D_n
G1	Fusion of H1 and H2	$d_{1,1}$,$d_{1,2}$	$d_{2,3}$	$d_{3,7}$,$d_{3,9}$	$d_{i,1}$, $d_{i,4}$	…	$d_{n,1}$
G2	H3	$d_{1,2}$	$d_{2,2}$	$d_{3,4}$	$d_{i,5}$	…	$d_{n,6}$
G3	Fusion of H4, H5 and H6	$d_{1,1}$,$d_{1,2}$,$d_{1,5}$	$d_{2,2}$, $d_{2,3}$	$d_{3,5}$	$d_{i,3}$, $d_{i,5}$, $d_{i,7}$	…	$d_{n,2}$

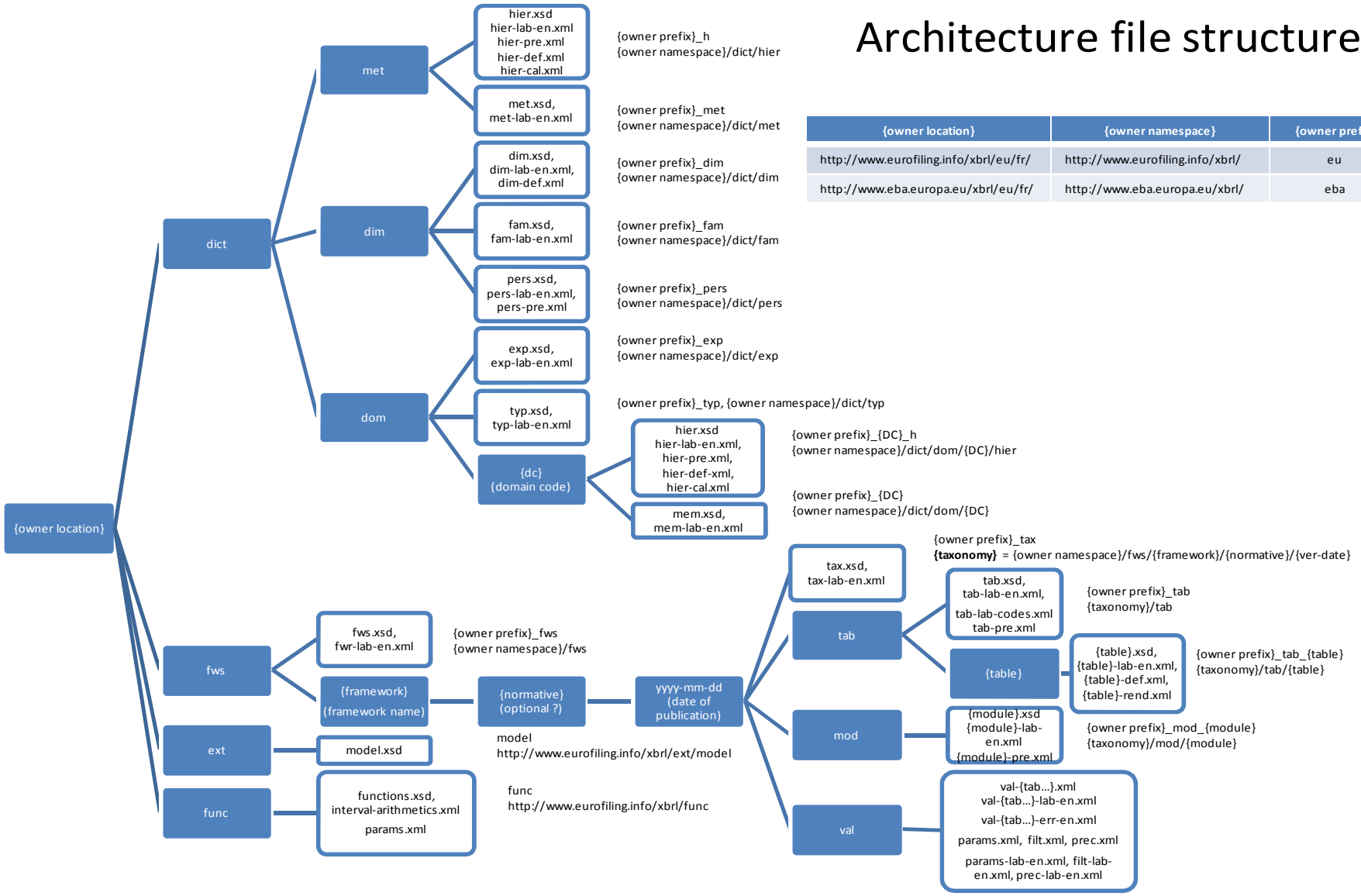
The final result of the algorithm is the result of the last iteration. The algorithm is not warranted to obtain the optimum number of hypercubes, but it will produce a similar result. The complexity of the

[15] The set of "possible values" for typed dimensions are "used" and "not used". It is not possible to establish dimensional constraints for specific values of a typed dimension according to the XDT specification.
[16] The algorithm works with multi-cellular hypercubes as well.
[17] In case of dimensions with the same number of members, the order is not relevant. In fact, in some cases the algorithm could produce better results following a different order of evaluation of its dimensions.

algorithm is O(n log n) with the number of dimensions and the number of members, so it should perform efficiently, even in interactive tools.

It is important to remark that XBRL hypercubes are validation artefacts and should not be used by external systems for the automatic creation of database structures. The hypercubes produced by this algorithm do not obey to any kind of 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.

Architecture file structure



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