Eurofiling Framework Taxonomy Architecture

Version: 0.1 (Public Draft)

Publication date: 2009-01-04

Abstract

The Eurofiling Framework Taxonomy Architecture describes methods of translating information requirements into a formal representation of XBRL taxonomy.

The purpose of this document is the definition of the architecture used for all taxonomies of the EUROFILING project.

The aim of the architecture is to impose constraints and rules on defining and modelling of taxonomies in order to improve their consistency, maintainability and readability and facilitate processing by consuming applications and simplify their design.

Status

This document is a public working draft. Its circulation is unrestricted.

The architecture proposed in this document is subject to review. It is expected that it is changed or updated following the evaluation of different approaches, for example for classification of primary items and dimensions, visual representation of the information requirements using presentation link or generic link, etc.

This project has been made possible with the financial assistance of the European Union

This project can under no circumstances be regarded as reflecting the policies of the European Union

Index

1

Introduction

This document describes the methods of translating information requirements into a formal representation of XBRL taxonomy.

1.1 Authority

The EUROFILING project is a joint venture of the Committee of European Banking Supervisors and its national members, XBRL specification team as well as industry and commercial banks.

1.2 Goals of this document

The purpose of this document is the definition of the architecture used for all taxonomies of the EUROFILING project.

The aim of the architecture is to impose constraints and rules on defining and modelling of taxonomies in order to improve their consistency, maintainability and readability and facilitate processing by consuming applications and simplify their design.

1.3 Organization of this document

This document starts with the identification and description of the principles guiding the definition of the architecture. Decisions and alternative approaches must be related to these principles and only those aligned or maximising fulfilment of the selected criteria shall be included in the architecture.

The following chapter describes the characteristics of information requirements that needs to be translated into XBRL taxonomies and introduces two products of the EUROFILING project which are the Data Point Structure and the Matrix Schema. It also presents main considerations for translation of information requirements into the XBRL data model.

The last chapter describes the framework, common structures and links between the components of the framework, modularization of the framework as well as relations to external taxonomies. It also presents the approaches and solutions applied for definition of primary items, dimensions, domain members, modelling of domains, sub-domains and tables. At the end, it summarizes the open issues.

1.4 Intellectual Property Policy

The Best Practices on “Intellectual Property Policies for XBRL Taxonomies”, as published by XBRL International on www.xbrl.org, are the guide for EUROFILING taxonomies. For this purpose, the following disclaimer is to be included in all the Eurofiling XBRL documents:

“Copyright © CEBS Secretariat Ltd., Tower 42, 25 Old Broad Street, London, EC2N 1HQ, United Kingdom. See details at www.eurofiling.info/legal.html”

1.5 Terminology and document conventions

…

2 Principles

Following principles guide the definition of the architecture:

1. **Simplicity of the reporting process**: the architecture must focus on simplification of instance document creation.[1]
2. **Stability**: the application of the architecture must minimize the impact of changes resulting from amendments of information requirements on consuming systems.
3. **Consistency**: the framework under the architecture must be consistent in design and the taxonomies must be coherent and explicit.
4. **Compliance with specifications, best practices and related taxonomies** (eg IFRS): the architecture must maximally conform to approaches applied in other related projects.
5. **Maintainability**: the architecture must allow for the framework be easy to maintain by supervisors.
6. **Performance**: the application of the architecture should result in other technical advantages including reduced size of instance documents, better performance in processing (eg DTS loading), etc.

These principles need to be taken into account when evaluating approaches and justifying decisions.

Selected approaches and decisions made must be in agreement with these principles or at least try to maximise their fulfilments. Exceptions/Departures from the principles must be documented and justified.

These principles are by definition software independent. This documentation as well as the specifications of the XBRL standard are publically available and free of any license fees. Therefore the solutions are expected to emerge as a result of mechanisms of a free market (demand of reporting entities/receiving institutions and supply of software vendors). However, there are commercial and open source solutions already available and their functionality may be taken into account when evaluating approaches or decisions against certain criteria.

[1] It is assumed that instance documents are created by the means of mapping and automatic extracting of data from internal systems of reporting institutions.

3 Information requirement and the data model

The process of exchange of information is a component of an information supply chain where a reporting entity must be able to produce the required information and a receiving institution must be able to store, analyze, publish and further transmit the received data.

XBRL taxonomy is a formal representation of information requirements. It identifies all data that needs to be transmitted from a reporting entity to a receiving institution.

The form of the description of information requirements independent of any specific implementations or protocols is an information model. In case of the EUROFILING project, these requirements has been model on less abstract level that already has some degree of specificity towards the use of the XBRL standard (eg the need of split into primary items and dimensions, application of domains and sub-domains, etc). This level is closer to the concept of a data model which in the EUROFILING projects is represented by the Data Points Structures and the Matrix Schemas. XBRL taxonomy is a formal representation of a data model (reflecting information requirements).

3.1 Sources of information requirements

Information requirements are usually expressed in a textual form (standards, guidelines, regulations, etc). Additionally, textual requirements can be interpreted in a tabular form.

EUROFILING project is originally based on information requirements of COREP and FINREP defined by the Committee of European Banking Supervisors. Information requirements of FINREP are defined in the following documents:

- International Financial Reporting Standards (IFRS) adopted by the International Accounting Standards Board (IASB),
- Regulation (EC) No 1606/2002 of the European Parliament and of the Council (19 July 2002) on the application of international accounting standards,
- Directive 2006/49/EC of the European Parliament and of the Council (14 June 2006) on the capital adequacy of investment firms and credit institutions,
- Guideline of the European Central Bank (30 July 2002) concerning certain statistical information requirements of the European Central Bank and the procedures for reporting by the national central banks of statistical information in the field of money and banking statistics (ECB/2002/5) (2002/656/EC),
- Guidelines on Financial Reporting defined by the Committee of European Banking Supervisors (CEBS),

while COREP is based on:

- Basel II, International Convergence of Capital Measurement and Capital Standards: A Revised Framework (Comprehensive Version) published by the Basel Committee on Banking Supervision (June 2006),
- Directive 2006/48/EC of the European Parliament and of the Council (14 June 2006) relating to the taking up and pursuit of the business of credit institutions (recast),

- Directive 2006/49/EC of the European Parliament and of the Council (14 June 2006) on the capital adequacy of investment firms and credit institutions,
- Guideline of the European Central Bank (30 July 2002) concerning certain statistical information requirements of the European Central Bank and the procedures for reporting by the national central banks of statistical information in the field of money and banking statistics (ECB/2002/5) (2002/656/EC),
- Guidelines on Common Reporting defined by the Committee of European Banking Supervisors (CEBS).

An integral part of the Guidelines on Common Reporting (COREP) and the Guidelines on Financial Reporting (FINREP) defined by the CEBS is a set of tables and supporting documentation explicitly defining the scope and the content of reportable information. These guidelines are the sole information requirements for the FINREP and COREP taxonomies. They also define the character and scope of changes that can be imposed on information requirements by national banking supervisors in the EU member countries.

In the long term, the EUROFILING project is expected to cover not only the banking industry but also capital market, insurance sector and other areas of reporting following the legislative and organizational changes in the financial markets supervisions and reducing the reporting burden in the EU.

3.2 Data Points Structure

Data Points Structure is a form of representation of information requirements by identification of reportable information as data points that have specified nature and can be characterized using consistently applied breakdowns. This approach has been developed for the purposes of the EUROFILING project.

A data point as a financial concept is characterized by defining its basic financial meaning (nature) and specifying information of breakdowns in which it is described in different tables or paragraphs of the documentation. An outcome of this process is a complete set of data points that are required to be reported. It explicitly describes all characteristics and allows identifying relations between data points located in different tables or paragraphs of documentation.

Data Point Structure as a data model introduces a very initial distinction into primary items (basic financial meaning) and dimension (breakdown) and applies differentiates the primary items basing on the period type property (stocks/flows). Application of the Data Point Structure on the formal representation of information requirements taxonomy) may assume merging basic financial meaning with some breakdowns (i.e. primary items are defined as concatenation of the nature of a financial term with components of a breakdown). This merging must be applied consistently with regard to the nature of a financial concept and the breakdown.

Data Points Structure has been described and explained in the document “Data Points Structure - Explanatory Documentation”[2].

[2] reference to a PDF

3.3 Matrix Schema

The Matrix Schema is another form of a data model. In represents the content and structure of a taxonomy. It contains much more specific information than it is defined in the Data Point Structure (which is less XBRL oriented). In particular, the matrix schema specifies primary items, dimensions and domains members and presents the multidimensional combinations resembling the actual modelling of the taxonomy. In particular, it introduces following concepts:

1. Domains.
   The domain of a dimension is the set of all possible values (domain members) that can be assigned to a certain dimension. The domain of a dimension is a global concept, that is, the domain of a dimension is unique to the taxonomy.

2. Sub-domains.
   A sub-domain of a dimension is the set of domain members that can be assigned to a certain dimension in the context of one or more primary items. A sub-domain of a dimension is, thus, a subset of the members of the domain of that dimension. A dimension has at least one sub-domain (the domain is to be considered a special case of sub-domain), but can have more.

3. Homogeneous sections
   A homogeneous section (or a section) is a region of the dimensional space defined by a taxonomy, where every primary item is combined with a common set of dimensions. The sub-domain of each dimension is the same for each primary item in the region but there could be some exceptions. The *sub-domain of a dimension in a section* is the superset of all the sub-domains of that dimension for each primary item in the section.

The description and explanation of the Matrix Schemas has been presented in the document “Matrix schemas for COREP and FINREP taxonomies”[3].

3.4 Modelling considerations

A taxonomy as a formal representation of the data model should explicitly and consistently represent information requirements. Due to the technical reasons (resulting from the architecture of the XBRL standard) a taxonomy must distinguish between primary items, dimensions and domain members. Additionally, it is required that the taxonomy corresponds to the original tabular view of the information requirements (rendering), defines constraints on the data (validation) and is able to represent different versions of information requirements in time (versioning).

3.4.1 Primary items and dimensions

There is no clear definition of primary item, dimension and domain member in any of the XBRL specification that considers these concepts from the business perspective. In fact these constructs are only described in abstract on syntactical level. As a result of that, it is a decision of a taxonomy developer to identify the difference, justify the decision and apply the reasoning consistently across the taxonomy.

[3] reference to a PDF

As a result of discussions among the members of the EUROFILING project initiative it was agreed, that for the first draft release of the architecture documentation supported by the draft of the FINREP taxonomy, the split into primary items, dimensions and domain members is aligned with the definitions of basic accounting meaning (nature) and breakdowns applied in the Data Points Structure. This decision has been motivated by the deadline for the release of the products (end of 2009) which didn't allow for the preparation and thorough analysis of alternative approaches. It is expected that the selected approach will be evaluated in the next phases of the project.

3.4.2 Corresponding to the tabular view

It is important that identification of information requirements takes into account views and structures applying to these requirements.

In particular this identification shall allow reflecting the split of information requirements in tables (as for example defined in the guidelines of the CEBS). This split is used in many countries to differentiate between information requirements basing on frequency (monthly, quarterly, semi-annually, and yearly), type of report (solo, consolidated, etc) and type of reporting entity.

Information requirements expressed in form of tables in guidelines (for example in the MS Excel format) can be flexibly modelled in order to express the preferred layout. Data modelling in general (and in XBRL in particular) focuses on explicit and coherent identification of information requirements. The mechanisms for presenting these information requirements in views that resemble the initial layout of tables defined in guidelines are currently under development as XBRL specifications. Due to the lack of a standard solution, this architecture makes assumptions on modelling of sections in the definition linkbase of a taxonomy (as described in section 3.3 of this document) that as a group correspond to sets of data defined in the tables in guidelines. Additionally, an attempt has been made in order to customize the presentation link for rendering purposes. This is a temporary approach and needs further development in the next phases of the project.

In addition to that, taxonomies contain references to underlying standards/guidelines/regulations that are global (refer to the generic definition independent from the context of any table or type of reporting) and local (eg applying only to a certain table). Similar distinction applies to labels. There are cases for which reflecting this information in taxonomy using standard XBRL mechanism for labels and references may be problematic due to the split into primary items and dimensional constructs. This situation requires a special approach for modelling or a mechanism that allows to link primary item together with applicable dimensional information to one or more resources.

...

3.4.3 Constrains on data

Instance documents are validated with regard to different criteria. Some criteria are out of scope of the taxonomy architecture. For example constraints related to the reporting process model (eg structure and content of instance documents, relation between type of a report and a reporting entity, etc.) must be applied at application level. Structural constraints (related to the way the information is serialized in an electronic format to be sent to the receiving institution) are performed according the XBRL 2.1, XDT 1.0 and other specifications and are generic with respect to taxonomies.

Apart from these two types of constraints, there are three groups that may be considered for the taxonomy architecture:

1. Constrains on/definition of required data (rejecting filings which include data that has not been required or doesn't include information that was explicitly requested).[4]
2. Constraints on the value of a single fact (check whether a reported fact has a valid value, independently of the value of any other fact).
3. Constraints on the value of more than one fact (logical and mathematical relationships between different facts reported taking into account the error due to the accumulation of rounding margins of the figures involved; given that there is not a common agreement on the minimum precision required at European level nor there is a single currency, those constraints must be designed in a flexible way).

The EUROFILING taxonomies shall predominantly contain constrains on the value of a single fact or on the value of more than one fact in order to check the data quality as well as for documentation purposes (mathematical/logical relations may also carry semantic meaning). As a whole, constraints can be considered to be global (applied to the definition of a concept irrespective of the table) however in most of the cases they are associated to one or more than one tables.

This documentation contains a proposal for definition of these kinds of constraints. It shall be evaluated prior to release of a final product.

3.4.4 Changes in information requirements

Another aspect that needs to be taken into consideration for the taxonomy architecture is changes of information requirements. Standards, regulations and guidelines are subject to modifications which need to be reflected in the data model and later in the formal representation that is a taxonomy. The taxonomy must include information about the version of information requirements that it expresses. This topic has been addressed in this taxonomy architecture.

[4] There is not a clear difference between this kind of constraints and the ones related to the reporting process. For example, different supervisors may require specific subsets of data for certain institutions with different frequency. There is neither a clear difference with XDT structural constraints: dimensional structural constraints check that reported information is in the dimensional space defined, whereas constrains on required data check whether the information is expected (no matter if it is valid or not).

4 Framework and modularization

The EUROFILING project is expected to cover information requirements and serve reporting purposes of different industries and sectors in Europe. At present, it is limited only to the banking sector and the guidelines and requirement of the Committee of European Banking Supervisors.

4.1 Components of the framework

A framework of the EUROFILING project is split in components. Each component of the framework relates to separate information requirements (of different purpose or nature). For example FINREP and COREP are two different components of the framework of which the first one relates to IFRS based reports while the other is focused on Basel II Accord requirements.

Modularization of the framework relates to the arrangement of and relations between its components. Apart from components reflecting specific (hence defined separately) information requirements, there are also components containing common definitions and structures reused across the framework or applied for an industry/sector. It is expected that all components of the framework are aligned in terms of the architectural design.

4.1.1 Location and base URIs

Due to the fact, that the EUROFILING project originates and is currently limited only to the banking supervision, all taxonomy files of the framework are hosted on the Internet location in the CEBS domain: *http://www.c-ebs.org/eu/fr/esrs* which it the official URI of the framework.

This URI is a root for the hierarchy of the framework URIs used for location of files and on XBRL technical constructs (namespaces, roles, etc).

Each component of the framework (eg COREP, FINREP) is represented at first level in the hierarchy (one level below the root). The base URI of a framework component is constructed as an extension of the root URI of the framework with the name of a component:

{root URI}/{framework component}

Common definitions and constructs reused across framework components are defined in the component named COMMON as it shall only include structures that are independent from standards/guidelines/regulations..

Table 1 presents currently used based URIs for components of the EUROFILING framework.

Table 1

Description	URI
COMMON	http://www.c-ebs.org/eu/fr/esrs/common
FINREP	http://www.c-ebs.org/eu/fr/esrs/finrep
COREP	http://www.c-ebs.org/eu/fr/esrs/corep

4.1.2 Versions

The EUROFILING framework as a whole can evolve it time and include more components. Additionally, as in any other project, the life cycle of information requirements for each component involves corrections and revisions of the releases.

A taxonomy that is a formal representation of the information requirements of a framework component at a moment in time includes information required to identify its version. The codification used for this identification takes into account the different natures of the changes of a framework component along its life cycle, i.e. changes in the information requirements and changes due to architecture changes or correction of bugs in the formal representation of the information requirements (not the information requirement themselves).

This version information is included in the absolute URI of the official Internet location of each file and consists of two components:

A - year and month of publication of information or IT requirements (eg *2009-12* for requirements published in December 2009), and

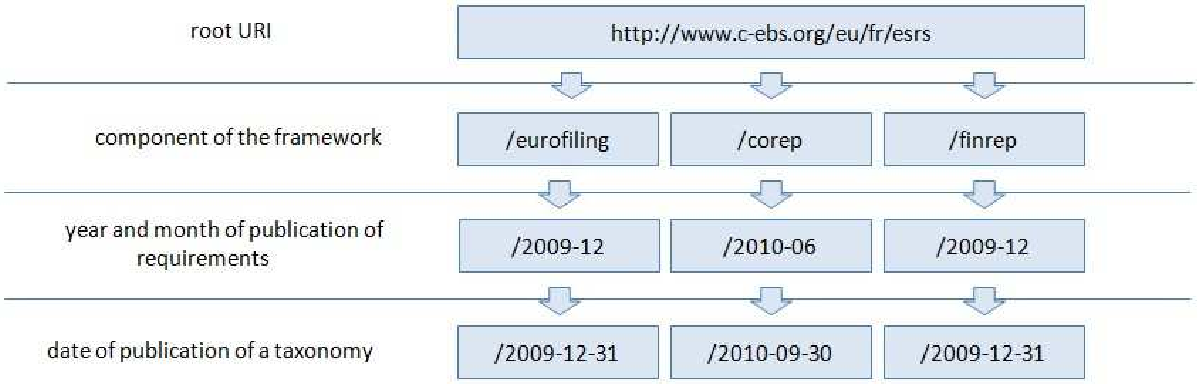
B - date of publication of a taxonomy (eg *2009-12-31* for a taxonomy released on 31st December 2009).

This version information is combined with the base URIs of the framework components according to the following pattern:

{root URI}/{framework component}/{year and month of publication of requirements}/{date of publication of a taxonomy}

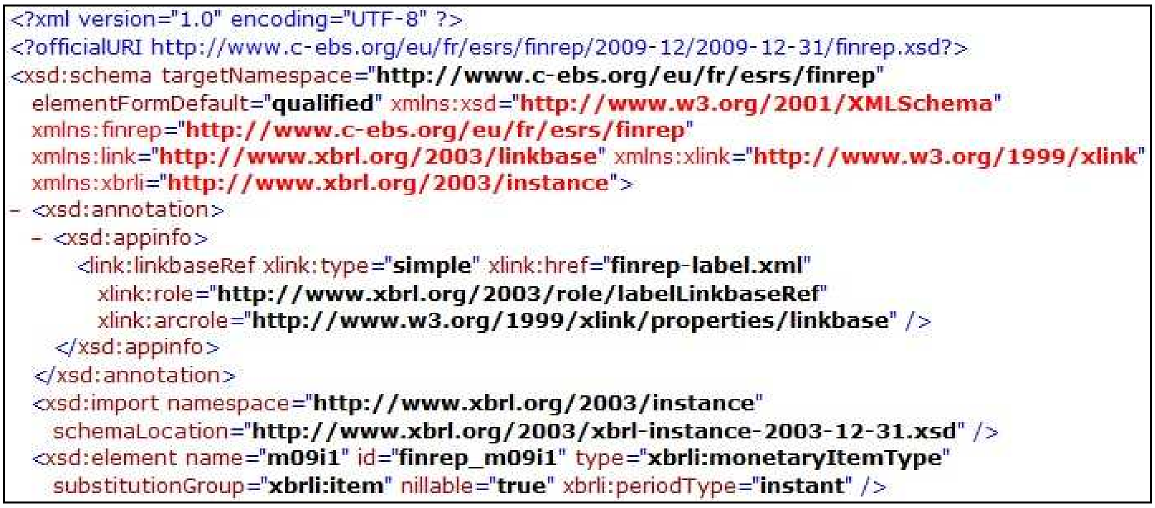
Examples illustrating application of the general pattern for versions of framework components are presented on Fig 1.

Fig 1


root URI
http://www.c-ebs.org/eu/fr/esrs
component of the framework
/eurofiling
/corep
/finrep
year and month of publication of requirements
/2009-12
/2010-06
/2009-12
date of publication of a taxonomy
/2009-12-31
/2010-09-30
/2009-12-31

The version information is also reflected in the processing instruction named *officialURI* that is included in each file of the framework and contains the address of the official Internet location of a resource. An example of the processing instruction embedded in a schema file is presented on Fig 2.

Fig 2



```
<?xml version="1.0" encoding="UTF-8" ?>
<?officialURI http://www.c-ebs.org/eu/fr/esrs/finrep/2009-12/2009-12-31/finrep.xsd?>
<xsd:schema targetNamespace="http://www.c-ebs.org/eu/fr/esrs/finrep"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:finrep="http://www.c-ebs.org/eu/fr/esrs/finrep"
  xmlns:link="http://www.xbrl.org/2003/linkbase" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xbrli="http://www.xbrl.org/2003/instance">
- <xsd:annotation>
  - <xsd:appinfo>
      <link:linkbaseRef xlink:type="simple" xlink:href="finrep-label.xml"
        xlink:role="http://www.xbrl.org/2003/role/labelLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" />
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://www.xbrl.org/2003/instance"
    schemaLocation="http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd" />
  <xsd:element name="m09i1" id="finrep_m09i1" type="xbrli:monetaryItemType"
    substitutionGroup="xbrli:item" nillable="true" xbrli:periodType="instant" />
```

Version information is NOT defined on any of the namespaces, role URIs or file names.

4.1.3 Version numbering

The issuing of a new edition of the taxonomies can arise from a number of reasons, including the following:

- Error or errors in the current edition of the taxonomy.

- A change in the business requirements.

- An improvement or change in the XBRL specification.

Categorising the effect (degree of impact) of such changes is always a matter for debate. In this document a key consideration, when categorising the degree of impact, is if the XBRL instance documents from different editions are compatible.

The identification of the effect of different versions is based in this XML instruction placed at the very beginning of each taxonomy file:

<?taxonomy-version 9.9.9?> where 9.9.9 is the version and release number in three levels. e.g.:

<?taxonomy-version 1.2.5?>

It is recommended that an equivalent arrangement is used in the XBRL instance documents, according to the features of the instance generator. The proposed identification of different releases is non normative for XBRL and, in general, it will be ignored by XBRL software. It is used for informative purposes and as an *ad hoc* private optional arrangement for XBRL practitioners.

Third level number: If the instance XBRL documents are backward compatible then the change gives rise to a new ***release*** of the taxonomy. The decision about backward compatibility is to be taken by the taxonomy owner. New releases would arise from non functional changes, as wording amendments (include e.g. amendments to references and other improvements) or the correction of typographical and other errors, all of them provided backward compatibility at XBRL level. Therefore, if the instance XBRL documents are seen as compatible, then the change gives rise to a new release of the taxonomy and this is reflected in a change in the third level of the numbering system. [i.e. *1.2.**5*** to *1.2.**6***]

If the instance XBRL documents are not compatible, then the change gives rise to a new version of the taxonomy, and this is reflected in a change in the second or first level in the numbering system.

Second level number: Minor functional ***version*** would include limited amendments to templates and guidelines for implementation, as well as non backwards compatible XBRL taxonomy improvements, including the correction of functional errors. (i.e. *1.**2.5*** to *1.**3.0***)

First level number: Major functional ***version*** would include material amendments to templates and guidelines for implementation, as well as new architectures or versions of the XBRL Specification. (i.e. ***1.3.0*** to ***2.0.0***)

4.1.4 Ensuring access to prior releases of taxonomies

This requirement arises from the reality that national supervisors may decide not to "upgrade" to a new release of a particular taxonomy. Supervisors might decide not to "upgrade" to the latest release of a particular taxonomy for several reasons: sometimes the corrected items do not affect their reporting process, or sometimes they might prefer to make the corrections at local level, or simply saving the regression tests to move from one release to other. So, the process defined in this document should keep this possibility in mind.

The production of a new taxonomy, depending on the extent of the changes, may necessitate a substantial re-mapping effort for the supervisor who may well decide to reflect the changes by way of a further extension of its existing taxonomy, a decision which should require less remapping. The common practice in the XBRL arena is to associate new namespaces to all new editions, regardless of being functional releases or maintenance releases. This approach has an important impact on the XBRL reporting chain. For instance, a bank that has established a mapping between its internal model and its corresponding XBRL representation would be forced to redefine this mapping even if there is no change in the concepts represented!

All the taxonomy releases are kept available to the national supervisors. Moreover, national supervisors who have extended taxonomies should be free to stay at a specific release if they so wish. For instance, the corrected errors may have no impact on a particular extension, and so they should be able to avoid taking the risks of adopting the new release of the taxonomy.

As a solution, every taxonomy release will be available on the official Internet location. The content of a taxonomy release MUST NOT be modified. Any change to a release MUST be implemented by editing a new release. New releases of the same version should include a documented list of the changes made to the previous release/s.

Each file associated with a taxonomy ***release*** will be located at a fixed and absolute URI of the official Internet location of each file and consists of two components:

A - year and month of publication of information or IT requirements (eg 2009-12 for requirements published in December 2009), and

B - date of publication of a taxonomy (eg 2009-12-31 for a taxonomy released on 31st December 2009).

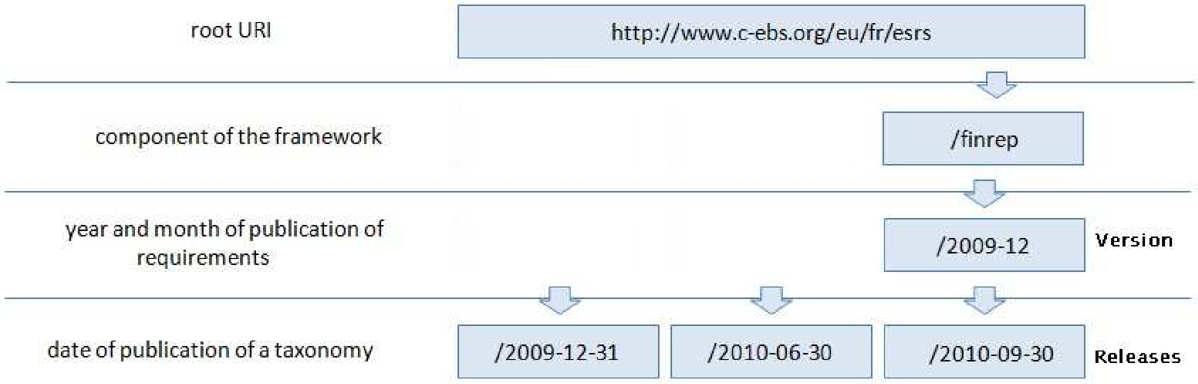
Each file associated with the ***latest release*** of a taxonomy ***version*** will be **ALSO** located at a fixed and absolute URI of the official Internet location of each file and consists of one component:

A - year and month of publication of information or IT requirements (eg 2009-12 for requirements published in December 2009)

Note that, as opposed to the former ***release*** URI, the ***version*** URI does not include the release date. This location can be thought of as a link to the latest official ***release*** of a functional ***version***. So, a taxonomy pointed by this URI will always point to the last maintenance release available of a specific version. A change of version is considered a change on IT requirements and therefore forces a change in the *year and month of publication of information or IT requirements*.

New editions will be published for reviewing under its new and unique corresponding ***release*** URI.

Once the release is approved, in the version do not change, then the files in the official ***version*** URI will be removed and replaced by the new ***release***; otherwise, a new official ***version*** URI will be created with the same files than in the ***release*** URI.


root URI
http://www.c-ebs.org/eu/fr/esrs
component of the framework
/finrep
year and month of publication of requirements
/2009-12
Version
date of publication of a taxonomy
/2009-12-31
/2010-06-30
/2010-09-30
Releases

Supervisors have two choices for consecutive releases of the same version:

- Link to a specific taxonomy release using the ***release*** URI. This option freezes the taxonomy files. No further changes to the taxonomy are to be applied here. New releases of the same version should include a documented list of the changes made to the previous release/s. Bug fixing and any other change implies the use of a different URL. Each URL change usually implies a change in the instance document generator implementation, to be properly tested. This is the equivalent to the traditional XBRL mechanism.

- Link to the functional ***version*** URI, which will always have the latest release of such version. Each set of modifications (bug fixing and any other changes) is coded as a new release replacing the content of the files of the official version. Each release has been previously tested to prove being backwards compatible with the previous release(s) of the same official version. Each taxonomy release is oriented to be propagated, downloaded and applied in all and each one of the instance document generators, without mandatory testing. All the instance documents of the same taxonomy version, independently of which particular taxonomy release was used at generation time, will be processed with the last taxonomy release of such taxonomy version. This method is applicable only INSIDE a specific version: the change of a version forces a new URI.

4.2 Structure of a framework component

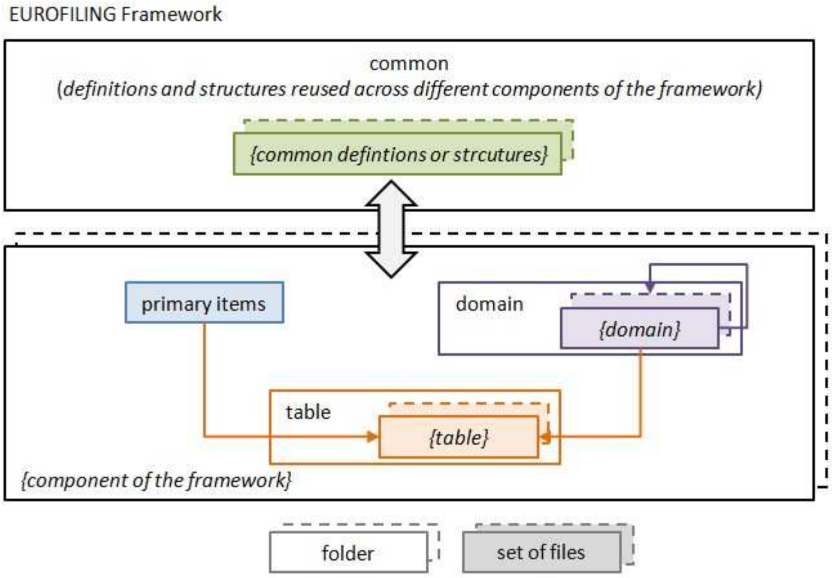
Framework components reflect information requirements. They consist of XBRL files containing:

1. definitions of primary items, dimensions and members and declarations of domains and sub-domains,
2. constructions of tables (expressing and communicating information requirements).

COMMON component of the framework contains technical constructs (eg related to syntactical matters, roles used on extended links or resources, etc) reused across different framework components.

The generic overview of the framework including the generic structures of framework components is presented on Fig 3.

Fig 3


EUROFILING Framework
common
(definitions and structures reused across different components of the framework)
{common defintions or strcutures}
primary items
domain
{domain}
table
{table}
{component of the framework}
folder
set of files

Information requirements are described using different constructs. These constructs are in particular primary items, dimensions and domain members. The full identification of a financial concept required to be reported is achieved by interpretation of the primary item in the context of associated dimensional information. Due to the fact, that a primary item can be described using different dimensional information and the opposite (that the same dimensional information can illustrate many primary items), the constructs representing primary items and dimensional information are defined in XBRL as separate and independent components that are eventually link to each other using another component defined for this purpose (tables).

Primary items, dimensions and domain members must by uniquely identified with a local name within a namespace.

4.2.1 Primary items

According to section 3.4.1 of this document primary items reflect basic accounting meaning (nature) as defined in the Data Points Structure. Definition of a primary item requires specifying its financial and technical characteristics. The financial properties comprise of: data type, period type and balance attribute. Technical characteristics include: substitution group, nillable attribute and local name.

Abstract items are very likely to be defined only in these components of the taxonomy that link primary items with applicable dimensions and domains in order to represent the required reportable information for a table. Nevertheless, in order not to constrain the use of abstract items, they are allowed to be defined together with non-abstract primary items if it is needed (eg for the purpose of presentation of a general hierarchy of primary items).

4.2.1.1 Folders, files, namespaces

For each component of the framework, primary items are defined in a single schema file. This schema file is defined in the root folder of a particular version a framework component. The name of this schema file, recommended prefix and the target namespace are derived from the name and the official location of the framework component for which the primary items are defined. Examples of schema file names, recommended prefixes and target namespaces for different framework components are presented in Table 2.

Table 2

Framework component	1. schema file location and name 2. recommended prefix 3. target namespace
FINREP	1. http://www.c-ebs.org/eu/fr/esrs/finrep/2009-12/2009-12-31/finrep.xsd 2. finrep 3. http://www.c-ebs.org/eu/fr/esrs/finrep
COREP	1. http://www.c-ebs.org/eu/fr/esrs/corep/2010-01/2010-06-01/corep.xsd 2. corep 3. http://www.c-ebs.org/eu/fr/esrs/corep
COMMON	1. http://www.c-ebs.org/eu/fr/esrs/common/2010-01/2012-06-01/common.xsd 2. common 3. http://www.c-ebs.org/eu/fr/esrs/common

Schema file defining primary items may refer to linkbase files:

- containing label link with standard role and different labels (as defined in the XBRL 2.1 specification); name of this linkbase file is created using a following pattern: *{framework component}-label.xml* (eg *finrep-label.xml*);
- containing reference link with standard role and different references (as defined in the XBRL 2.1 specification) that use standard (XII) parts or custom parts defined in the COMMON framework component; name of this linkbase file is created using a following pattern: *{framework component}*-reference.xml (eg *finrep-reference.xml*);
- other links, for example general hierarchy in the presentation linkbase, general assertions, etc; names of these files are defined according to the following pattern: *{framework component}-{type of extended link}.xml.*

Information in linkbases referred from the primary items schema files must contain only the general information on primary items that is related to their global definitions.

4.2.1.2 Data type and constrains

Information requirements express constrains on the type of requested data. This architecture assumes that only the following basic data types are used for definition of primary items:

- monetary item type: applied for primary items that are represented by a number at certain precision and refer to a unit which measure is ISO4217 currency code,
- decimal item type: applied for primary items that are represented by a number at certain precision and refer to a unit,
- date item type: applied for primary items that if present in the instance document must be represented by a valid XML Schema date format,
- string item type: applied for primary items that if present in the instance document must be represented by a set of characters.

Additional constraints regarding allowed ranges for numbers and dates, patterns and enumerations for strings, etc as well as references to specified units are imposed using assertions. These are general assertions as explained in section 4.2.3.6 of this document.

For consistency reason, the data type for abstracts defined in the schema file for primary items is string (this information has no semantic meaning).

4.2.1.3 Period type

Period type attribute is used for classification of information requirements according the distinction set in the XBRL 2.1 specification, i.e.:

- primary items that are reported at a point of time (as of a specified date) are instant,
- primary items that are reported for a period (between specified dates or forever) are duration.

It is possible that a financial term is represented by two primary items of which one is instant and the other is duration (eg *Allowances* and *Change in allowances*).

For consistency reason, the period type for all abstracts is duration (this information has no semantic meaning).

Application of a custom dimension for the purpose of identification of the duration information and the length of a period needs be analysed and evaluated.

4.2.1.4 Balance attribute

Balance attribute is not used on the definitions of primary items. All information regarding positive or negative values for primary items is expressed using assertions.

4.2.1.5 Substitution group

All primary items must be defined in the *xbrli:item* substitution group. *xbrli:tuple* as well as other custom substitution groups are not allowed on definitions of primary items.

The use of custom substitution for the purpose of filtering in formula linkbase groups needs be analysed and evaluated.

4.2.1.6 Nillable

Nillable attribute on all primary items must be true. Nil elements are often used when correcting reports for removing previously send facts. In required, it is possible to prohibit nil elements from being reported by application of an assertion.

4.2.1.7 Local name

Local name of a primary item is constructed as a concatenation of four components:

- a small letter reflecting the data type: *m* – monetary item type, *c* – decimal item type, *d* – date item type, *s* – string item type or *a* for abstract items,
- two digits reflecting last two positions identifying year when the item is created, eg *11* for 2011,
- a small letter reflecting the period type: *i* – instant, *d* – duration or . *(dot)* for abstract item,
- a sequential number starting from *1* and unique within a namespace.

Two primary items representing the same information but defined for different period type (eg *Allowances* and *Changes in allowances*) must share the same sequential number. Examples of local names are presented in Table 3.

Table 3

prefix: local name	description
finrep:a11.1	abstract item defined in 2011 for the first time
finrep:m11d2	monetary item, instant period type
finrep:m11i2	monetary item, duration period type, counterpart of the primary item finrep:m11d2
finrep:d11d3	date item, duration period type

4.2.1.8 Label

Primary items must have labels that reflect human readable name of a concept represented by the primary items. These must be general labels independent from the placement of a primary item in the structure of information requirements.

Labels are defined in the label linkbase file referred from the schema file defining primary items. They are defined in the standard extended link role. At least one label must be defined in the standard label role.

...

4.2.1.9 Reference

Primary items should have references that indicate fragments of the authoritative literature describing the meaning and application of primary items. These must be general references independent from the placement of a primary item in the structure of information requirements.

References are defined in the reference linkbase file referred from the schema file defining primary items. They are defined in the standard extended link role.

...

4.2.2 Dimensions, domain members, domains and sub-domains

Each component of the framework may contain definition of structures used to represent dimensional information. Dimensions and domain members gathered in domains and sub-domains are defined in order to represent these structures. The interpretation of a domain and subdomain is has been described in section 3.3 of this document.

Physically, a dimension is defined together with a domain that it refers to. It is possible, that there is more than one dimension defined with the domain. Is such a case, all dimensions defined together with a domain must apply to refer to exactly the same domain. A domain is interpreted in the context of a dimension to which it is applied. Domains can refer to each other. As a result of that, a dimension may refer to a domain that reuses domain members from other domains. In this situation the domain is interpreted with regard to applicable members (reused or locally defined).

4.2.2.1 Folders, files, namespaces

For each component of the framework there is a folder dedicated to contain dimensional constructs. The name of this folder is 'domain'. It has a representation in the official internet location of the taxonomy and as a component is added to namespaces in order to signal dimensional character of defined information.

Each domain is defined a separate set of files in the 'domain' folder and has a short but meaningful name describing its content. This name is used as a file name, a recommended prefix and additional component to the namespace differentiating it from other domains. Examples of file names, recommended prefixes and namespaces of domains are presented in Table 4.

Table 4

Domains	1. schema file location and name 2. recommended prefix 3. target namespace
instrument	1. http://www.c-ebs.org/eu/fr/esrs/finrep/2009-12/2009-12-31/domain/instrument.xsd 2. instrument 3. http://www.c-ebs.org/eu/fr/esrs/finrep/domain/instrument
residence	1. http://www.c-ebs.org/eu/fr/esrs/finrep/2009-12/2009-12-31/domain/residence.xsd 2. residence 3. http://www.c-ebs.org/eu/fr/esrs/finrep/domain/residence

Schema files for domains contain definition of dimensions and domain members may refer to linkbase files:

- containing label link with standard role and different labels (as defined in the XBRL 2.1 specification); name of this linkbase file is created using a following pattern: *{domain name}-label.xml* (eg *instruments-label.xml*);
- containing reference link with standard role and different references (as defined in the XBRL 2.1 specification) that use standard (XII) parts or custom parts defined in the COMMON framework component; name of this linkbase file is created using a following pattern: *{domain name}*-reference.xml (eg *instruments-reference.xml*);
- containing definition link representing the structure of a domain and each sub-domain separately (in a separate extended link role); names of these files are defined according to the following pattern: *{domain name}-definition.xml* (eg *instrument-label.xml*);

Information in label and reference links referred from the schema files for domains must contain only general information on dimension and domain members related to their definitions (and not table specific).

4.2.2.2 Dimensions

Dimensions are defined in the schema file of the domain that they refer to.

According to the XBRL 1.0 Dimensions Specification, items representing dimensions are defined in the xbrldt:dimensionItem substitution group and are abstract. Other attributes on declaration of a dimension have no semantic meaning. Nevertheless, for consistency reasons all dimension items are: string item type, duration period type and nillable.

Local names for dimensions are created as a concatenation of four components:

- two letters reflecting a type of dimension:
  - *xd* for explicit dimension,
  - *xt* for typed dimension
- two digits reflecting last two positions identifying year when the dimension is created, eg *11* for 2011,
- a single character . *(dot)*,
- a sequential number starting from *0* and unique within a namespace.

There may be two dimensions representing the same breakdown of which one is explicit and the other is typed. Consequently, two domains, explicit and typed, that represent the same breakdown must be defined under a single namespace. Examples of local names of dimensions are presented in Table 5.

Table 5

prefix: local name	description
residence:xd10.0	explicit dimension defined in 2010 for residence domain
residence:xt10.0	typed dimension counterpart of residence:xd10.1
residence:xd10.1	explicit dimension defined in 2010 for residence domain but different to residence:xd10.1
instrument:xd11.0	explicit dimension

If there is more than one dimension of a type (explicit or typed) defined in for a domain, then they must reused exactly the same domain (including its structure). These dimensions may refer to different sub-domains.

4.2.2.3 Domains

The domain of a dimension is the set of all possible values (domain members) that can be assigned to a certain dimension. The domain of a dimension is a global concept, that is, the domain of a dimension is unique to the taxonomy.

There are two types of domains:

- explicit: whose set of possible values is explicitly enumerated in the taxonomy in the form of domain members,
- typed: domain whose set of possible values is defined using an XML data type.

4.2.2.4 Explicit domain members

Explicit domain members are defined in xbrli:item substitution group. They are of custom data type which is xbrldty:domainMemberItemType[5]. Domain members must not be reused as primary items and vice versa. Thus all domain members are abstract items. Other attributes on declaration of domain members have no semantic meaning. Nevertheless, for consistency reasons all domain members are: duration period type and nillable.

Local names for domain members are created as a concatenation of four components:

- a letter *x*,
- two digits reflecting last two positions identifying year when the domain member is created, eg *11* for 2011,
- a single character . (dot),
- a sequential number starting from *1* and unique within a namespace; 0 (zero) is reserved for members representing a total of a domain.

Examples of local names of domain members are presented in Table 6.

Table 6

prefix: local name	description
instrument:x11.1	member of domain instruments, defined in 2011
instrument:x10.0	member representing a total for instruments, defined in 2010
residence:x09.1	member of domain residence

4.2.2.5 Typed domain

Typed domain is an XML construct that is referenced from the declaration of a typed dimension. Only simple constructs (restrictions on data types) are allowed for typed mains.

The local name of the element that defines the content model of the domain for typed dimensions is *domain*, unless the domain is defined in a schema external to the EUROFILING framework (eg ISO codes). Examples of local names of typed domains are presented in Table 7

Table 7

prefix: local name	description
securityCode:domain	security code typed domain
obligorGrade:domain	obligor grade typed domain

4.2.2.6 Labels

Dimensions and domain members must have labels that reflect their human readable meaning. Labels are defined in the label linkbase file referred from the schema file defining a domain. They are defined in the standard extended link role. Domain members that are reused by different domains may have different labels depending on the domain. In such situation, the label applicable for a particular domain is defined in the extended link role of that domain. At least one label must be

[5] This is similar approach to the IFRS taxonomy for 2010. At present this data type is defined in the DTS of the EUROFILING taxonomies in COMMON component of the framework but it is expected to become recognized and recommended by the XII and referred from official XII schema files.

defined in the standard label role. These must be general labels independent from the placement of a dimension or domain member in the structure of information requirements.

4.2.2.7 References

Dimensions and domain members may have references that indicate fragments of the authoritative literature describing their meaning and application. References are defined in the reference linkbase file referred from the schema file defining a domain. They are defined in the standard extended link role. Domain members that are reused by different domains may have different references depending on the domain. In such situation, the reference applicable for a particular domain is defined in the extended link role of that domain. These must be general references independent from the application of dimension in the structure of information requirements.

4.2.2.8 Role types and representation of a domain

The whole set of domains is represented as relationships in a single definition linkbase file referred to from the schema file defining a domain. Domain and each of the sub-domains is defined in a separate extended link role with a root element being a dimension.

Role types used for extended links are declared in the schema file defining a domain. Their URI is constructed according to the following pattern:

{root URI}/{framework component}/role/domain/{domain name}/{sequential number}

The sequential number is omitted for a domain. This role type may be used on definition, label and reference links.

Role types for sub-domains contain the sequential number which starts with *1*. These role types are used on definition link.

Definition on a role type declaration consists of a domain name in the English language plus a sequential number (for sub-domain). Human description of extended link roles in other languages shall be provided using generic labels.

Examples of role type declarations are presented in Table 8.

Table 8

Definition	Role URI	Description
Instrument	http://www.c-ebs.org/eu/fr/esrs/finrep/role/domain/instrument	Instrument domain
Instrument 1	http://www.c-ebs.org/eu/fr/esrs/finrep/role/domain/instrument/1	Sub-domain of instrument

4.2.2.9 Dimension default member

Explicit dimension may have a default member. Default member must be defined in a domain referred by a dimension. In most cases default member is a total of a domain, a cross-section member of a domain or an assumed standard value for a domain.

'dimension-default' relation must be defined in an custom extended link role which role type is defined in the COMMON component of the framework and its role URI is 'http://www.c-ebs.org/eu/fr/esrs/common/role/dimension-defaults'.

4.2.3 Tables

Information requirements expressed in the tables of guidelines/standards/regulations are reflected in taxonomies in the form of multidimensional combinations as described in section 3.3 of this document.

These combinations are created independently form the definition of primary items, dimensions, domain members and structures of domains and subdomain.

4.2.3.1 Folders, files, namespaces

Each table is reflected as a separate set of files defined in the 'tables' folder. This folder is placed in the location of a version of a component of the framework. Each table has a short name that reflects its number and possibly also the content. This name is used as a file name, a recommended prefix and additional component to the namespace. Examples are presented in Table 9.

Table 9

Table	1. schema file location and name, 2. recommended prefix 3. target namespace
Table 1.1	1. http://www.c-ebs.org/eu/fr/esrs/finrep/2009-12/2009-12-31/table/table1_1.xsd 2. table1_1 3. http://www.c-ebs.org/eu/fr/esrs/finrep/table1_1
Table 13B	1. http://www.c-ebs.org/eu/fr/esrs/finrep/2009-12/2009-12-31/table/table13b.xsd 2. table13b 3. http://www.c-ebs.org/eu/fr/esrs/finrep/table13b

Schema files defining tables may refer to linkbase files:

- containing label link with standard and/or table specific role and different labels; name of this linkbase file is created using a following pattern: *{table name}-label.xml* (eg *table5a-label.xml*);
- containing definition link with table specific role or roles and relations linking primary items with dimensions (applying multidimensional combinations); name of this linkbase file is created using a following pattern: *{table name}-definition.xml* (eg *table5a-definition.xml*);
- containing presentation link with table specific role and relations aiming at visual representation of the content of the table; name of this linkbase file is created using a following pattern: *{table name}-presentation.xml* (eg *table5a-presentation.xml*);

Information in label and reference linkbases referred from the schema files of a table contains table specific information on primary items, dimensions and domain members as well as standard labels and references on abstract items and hypercubes defined in the schema file for a table.

Schema file for a table imports schema files defining primary items and schema files defining applicable in a table domains.

4.2.3.2 Abstract items and hypercubes

A schema file for a table may contain definitions of items that are needed to represent the information requirements and are not primary items, dimensions and domain members. In particular, these items are:

- an abstract item used as a root in extended links,

- other abstract items used in order to structure the hierarchy of primary items,
- one or more hypercubes used for linking primary items with applicable dimensional combinations or exclusions.

Local names for these items are created as a concatenation of three components:

- a letter *a* for abstract items and *h* for abstract hypercube items,
- a single character . (dot),
- a sequential number starting from *1* and unique within a namespace.

All items defined in the schema file for a table must be abstract. Abstract items are defined in the xbrli:item substitution group while hypercubes are defined in the xbrldt:hypercubeItem substitution group. Other attributes on declarations of abstract items or hypercubes have no semantic meaning. Nevertheless, for consistency reasons all items defined in the schema file for a table are: string item type, duration period type and nillable.

Abstract items and hypercubes may have general (standard labels).

4.2.3.3 Role types

Schema file for a table must define role types used on extended links referred by linkbase files associated with this schema.

Each role type declared in the schema file for a table may be used on label, reference, definition, presentation and generic links.

Construction of a base role URI for a table follows the pattern:

{root URI}/{framework component}/role/table/{table name}

Structures defined within extended link role using this role type refer explicitly to the content of the table identified by its name.

Due to the reason that the tables defined in information requirement may require a split into several sections in order to apply dimensional combinations (as explained in section 3.3 of this document), several additional role types may be defined. They are created according to the following pattern:

{root URI}/{framework component}/role/table/{table name}/{sequential number}

The link between a structure of information requirements and the sections is established through the role types naming convention that reuses the base role URI for a table.

Definitions of a role type declaration consist of a table name in the English language plus a sequential number (for sections). Human description of role types in other languages shall be provided using generic labels.

Table 10

Definition	Role URI	Description
Table 5a	http://www.c-ebs.org/eu/fr/esrs/finrep/role/table/5a	Contains information on Table 5a
Table 6 1	http://www.c-ebs.org/eu/fr/esrs/finrep/role/table/6/1	Contains information on a section of Table 6

4.2.3.4 Applying multidimensional combinations

Schema file for a table must refer to a single linkbase file containing definition links. The name of this linkbase file is *{table name}-definition.xml*.

A table may be reflected in one or more definition links. Factors that determine the number of extended links needed to represent a table result from its content, and in particular:

- existence of mutually exclusive dimensions (dimensions that cannot be combined together),
- number of excluded combinations of members of different dimensions.

These factors result in a split of the data set defined in a single table in information requirements into several extended links.

A root node of each extended link is an abstract item defined in the schema file for a table that groups primary items through 'domain-member' relationships. The set of primary items can be arranged hierarchically to improve its maintainability. This arrangement is facilitated by other abstract item defined in the schema file for a table.

A hypercube is linked to the root node of the extended link role through an 'all' arc. It links all primary items with the dimensions required for a section. Dimensions are connected to the hypercube via 'hypercube-dimension' arc. This arc contains xbrldt:targetRole attribute indicating an extended link role of applicable for this dimension domain or sub-domain.

Those primary items that have a more limited dimensional space are linked to another hypercube or hypercubes through one or more 'notAll' arcs to discard unwanted combinations. Again, these exclusions are defined by indicating appropriate sub-domains.

4.2.3.5 Visualisation

The purpose of the presentation link is merely to present the content of the taxonomy. Standard application of the presentation link is not sufficient to reflect the layout of information requirements that is expressed by tables defined in guidelines.

It is possible to customize the presentation link and by this mean introduce a proprietary rendering mechanism. Alternatively a generic link solution may be used for this purpose. It is important that the approach is able to facilitate table specific labels and references.

Different approaches for the visualisation are currently evaluated including WGWD of the rendering specification.

4.2.3.6 Assertions and formulas

Constraints on data content are performed by assertions in the formula linkbase. These assertions can take advantage of additional information at schema level (eg user attributes, substitution groups), by standard relationships of other linkbases (eg 'domain-member' relationships of the definition link) or by user specific relationships.

The use of XML Schema validation in this the architecture of the EUROFILING framework has been restricted only to very basic data types used for definition of primary items. Instead, assertions with filters based on schema level attributes are used. The advantage of this approach compared to schema validation is its flexibility and extensibility, i.e.:

– schema validations apply with no exceptions whereas assertions can be refined with additional filters or preconditions, so that the constraint can be ignored under certain conditions (for instance, for a certain value of a dimension).
– schema validations cannot be modified by extensions - if that would be the case, the only solution is the creation of a new concept to replace the existing one, which entails incompatibility issues (XBRL Formula is based on XLink and thus, can be easily extended).

Assertion sets are used to group and to classify assertions regarding the scope where they are to be applied. The following scopes are defined:

1. General assertions, which apply to the whole set of data.
2. Individual table assertions where each evaluation of such assertions involves just facts in a single table (such assertions can be applied to more than one table, but individual evaluations involve data in one table or another).
3. Cross tables assertions that involve facts represented more than one table for each evaluation of such assertions (thus, in order to be verified, those tables must be included in a single instance document).

Assertions are expressed in terms of concepts and are independent of a table or any other view of the conceptual layer. But, in the EUROFILING framework, they are applied on the level of tables. The reason for that is the assumption that non reported data is equivalent to zero. Input variables can be assigned to a default value (zero), should the data be absent on the input instance document. If an individual or crossed statement assertion is evaluated on an instance document where such statement is not reported, it could raise errors because of these default values. Therefore, assertion sets are associated with tables.

4.2.4 Common

Definitions of common constructs for the framework, appears at the level of the COMMON component of the framework. These constructs are independent from a specific purpose or meaning of any of the formwork components.

Common constructs may include:

- role types used on extended links and resources,
- arcrole types,
- generic link extended links, resources and arcs,
- other items used for custom modelling,
- etc.

Common constructs are defined in schema files and possibly also in linkbase files. Namespaces, role URIs, arcrole URIs, etc must base on the URI defined for the COMMON component of the framework in section 4.1.1 of this document.

4.2.5 Other remarks

4.2.5.1 ID attributes

The id attribute has no semantic meaning. It is used only for the purpose of referencing in XLink. Although in this framework ids are created consistently (eg FRTA pattern is applied for declarations:

{recommended prefix}_{local name}), at no means this should this be perceived as a rule for consuming applications.

...

4.3 Relation between components of the framework and to external taxonomies

It is assumed that there may be relations between different components of the framework between the components of the framework and external taxonomies. These relations may include matching of definition. This matching is performed using XLink mechanisms.

In particular, there is an assumption that the link between the FINREP and the IFRS taxonomy is maintained by the mean of a formula linkbase defining a set of transformation rules allowing to convert the FINREP based instance document into IFRS based instance document and vice versa in the scope of the matching definitions.

Matching definitions of the FINREP and IFRSs taxonomy (version 2009) has been identified and sample transformation rules has been defined for the first draft of the FINREP taxonomy in order to evaluate this solution.

4.4 Open issues

...