Authored by
Taxonomy Architecture
Guidance Task Force

October 2014

for
XBRL International, Inc.

XBRL Taxonomy Guidance Document (XTGD)

Version 1.2

Output of the XBRL International, Inc. Best Practices Board's

Taxonomy Architecture Guidance Task Force (TAGTF)

ABSTRACT

The Taxonomy Architecture Guidance Task Force (TAGTF) is a project of the XBRL International, Inc. (XII) Best Practices Board (BPB) which publishes white papers and other resources to aid the market in the understanding and implementation of eXtensible Business Reporting Language (XBRL), the global, structured data standard for financial and non-financial reporting. The TAGTF was established in 2012 to develop guidance materials for XBRL taxonomy creators and owners to help them make informed decisions based on the XBRL taxonomy architectures identified.

This XBRL Taxonomy Guidance Document (XTGD) gives analysis and guidance on select taxonomy aspects that were chosen to cover a cross section of the challenges that taxonomy architects face, including:

- Architectural choices for filing programs
- Data modelling
- Application of taxonomy architecture features

All data is sourced from XBRL taxonomies in current production use, and guidance is provided in the XTGD based on analysis of this data and the collective taxonomy architecture and implementation experience and knowledge of the TAGTF members. The taxonomies analysed were predominantly used in filing programs concerned with the collection of financial statements and this focus is reflected in the XTGD. Future versions of the XTGD will respond to the continued need for objective guidance on other aspects of taxonomy architecture.

One way in which the TAGTF gathers market needs is through short online surveys. You can find information about participation in current surveys at https://www.xbrl.org/taxonomy-architecture/.

STATUS

This document is the final release of the XBRL Taxonomy Guidance Document v1 (XTGD). Circulation of this document is unrestricted. This version of the XTGD will only be updated in order to incorporate errata corrections.

Future efforts of the Taxonomy Architecture Guidance Task Force (TAGTF) will be focused on producing taxonomy guidance documentation covering a significantly expanded range of topics.

PROCESS TO SUBMIT COMMENTS:

Please send all comments by email to: TAGTF-feedback@xbrl.org.

GET INVOLVED:

If you are interested in contributing to the future versions of the XBRL Taxonomy Guidance Document, please contact us by emailing TAGTF@xbrl.org.

MEMBERS OF THE TAGTF

Members of the current TAGTF contributing to the development of the XTGD include:

Ben Russell (Chair): CoreFiling

Shilpa Dhobale: IRIS Business Services

Warwick Foster: Xtek Solutions

Paul Hulst: Deloitte Netherlands

Eric Jarry: Banque de France

Makoto Koizumi: Mercury and Earth

Chie Mitsui: Nomura Research

Brad Monterio (XII Best Practices Board Liaison): Colcomgroup

Trevor Pyman: IRIS Business Services

Pascal Rodrique: Banque Nationale de Belgique

Joel Vicente: CoreFiling

Paul Warren: XBRL International Inc**.**

Andromeda Wood (Vice Chair): IFRS Foundation

Former or non-active members of the TAGTF who contributed to the XTGD are:

Shweta Gupta (Former Chair)

Somnath Ray

Claudia Koschtial

Maya Nair

Ronald Schechter

CONTENTS

1. Introduction

XBRL today is a global, structured data standard widely used for compliance and business reporting. XBRL use is mandated by many governments, regulators and supervisory bodies. In most cases, the taxonomy architecture and other supporting decisions used by these overseers have been developed independently or follow patterns laid down by the technology being used to create the taxonomy. This has led to several, disparate schools of thought arising around XBRL taxonomy architecture that may benefit from the collective experiences of others organizations that have previously tried a particular approach or have been successful with other approaches.

Over the course of the last few years, the number of XBRL implementations around the world has reached critical mass - there are enough taxonomies to be able to study the practical application of XBRL taxonomy architectural decision-making and analyse empirical data taken from those taxonomies. This creates an opportunity for current and future custodians of taxonomies to learn from others in order to better achieve their business goals. By analysing a sample of taxonomies, this XTGD aims to increase the understanding around important taxonomy architecture choices as well as approaches designers can take.

The TAGTF has predominantly analysed taxonomies where the focus of the filing program is in the collection of financial reports. Even within this scope, the variability in solutions to reporting challenges has shown that the original goal of defining XBRL taxonomy profiles - the idea that groups of taxonomies with similar business requirements share common architectural points - is not reflected in the current set of taxonomies analysed by the TAGTF. Some taxonomies have very similar characteristics, but there is no compelling evidence that this is linked to their business requirements or goals.

What can be seen in the analysis is that, sometimes, there is a clear consensus of the approach to meet specific reporting requirements. In these cases, it is reasonable to assume that following this approach can be considered good practice. Where there are two or more approaches taken (i.e., where different mechanisms are used to achieve similar business goals or requirements), it is important for taxonomy developers to understand the available options, be able to evaluate the suitability of each option for their use case and be aware of potential consequences of the chosen option.

2. AUDIENCE AND USE

The XTGD is targeted at those needing to make decisions about an XBRL taxonomy architecture. While some parts of the XTGD are of general interest to those who are involved in developing an XBRL reporting program, it is primarily aimed at XBRL taxonomy architects who have a prior understanding of the technical XBRL specifications.

The document strives to report good practice and present options for meeting specific reporting goals, as well as benefits, challenges and examples of taxonomy architecture decisions. Users of the XTGD should understand that it is not meant to prescribe certain approaches, but rather inform architects of the possible impact of taxonomy architectural decisions. With this in mind, users should feel free to deviate from the practices described here if there is a good reason to do so – the TAGTF requests that they also provide feedback by email (TAGTF-feedback@xbrl.org) about their experiences to the TAGTF so that other architects may benefit.

3. Work to-date

In order to document prevailing taxonomy architecture, the TAGTF developed an observation document questionnaire that was used to gather information on a number of existing XBRL taxonomies in use around the world today (see "Appendix A – Observation document template"). The questionnaire was preferentially filled out by the taxonomy owner but TAGTF members with a detailed knowledge of the taxonomy also contributed to the observations.

Having captured observation data on these taxonomies, the TAGTF analysed the data for the benefit of the XBRL community. This proved to be a complex analysis process, requiring in-depth and specific knowledge of particular XBRL reporting programs as well as a practical, technical understanding of the XBRL specifications. The results of the initial investigations are included in this version of the XTGD.

Note: The architectural decisions surrounding taxonomies using data point modelling methodologies is not explicitly covered by this version of the XTGD - these types of taxonomies were not in production use at the time of analysis and writing.

4. THE TAGTF DEFINITION OF "GOOD PRACTICE"

To provide a guidance document that is useful as well as informative, the TAGTF considered a number of phrases to characterise its findings and recommendations.

- "Best practice" was considered inappropriate given the diversity of approaches found during the taxonomy analysis coupled with the lack of objective metrics to measure their success.
- "Common practice" implies some benefits in using mature tools, techniques, approaches and methodologies but should not be followed blindly since there are also some common mistakes.
- Therefore, throughout the XTGD, "**good practice**" has been used in the sense in which it is described below.

Good practice in taxonomy architecture can be said to be the design decisions that exploit the features that XBRL adds to XML and by doing so provides the intended users of a taxonomy with a more efficient and effective way to prepare and consume structured data content. In specific cases where an XBRL specification defines the intended business case that a taxonomy feature was designed to address, it is typically considered good practice to follow this usage.

Good practice aligns architectural approaches with business goals and helps communication of reporting requirements from the issuer to the users, helps the understanding of resultant reports, or both. In the XTGD, this is represented by the listing of pros and cons for approaches to the implementation of an XBRL feature or the realisation of a business goal. The pros and cons expose the impact of different approaches to the same architectural aspect, helping the reader to choose the best fit for their specific case.

5. General taxonomy points

This Chapter covers some of the more general aspects of XBRL taxonomy architecture. This includes enabling a taxonomy architect to define "why a taxonomy is being written" in a way that enables it to be classified or compared to other XBRL taxonomies. The practices identified in this Chapter are applicable for all taxonomy architects throughout the taxonomy lifetime and include:

- How to categorise a taxonomy (for example, taxonomy uses cases and profiles);
- What is included in the taxonomy (the scope of the taxonomy); and
- How the taxonomy will be used (for example, will it be extended).

5.1 Taxonomy use cases

As noted in the introduction to the XTGD, it is not yet clear how to describe profiles of XBRL taxonomy architectures that relate to specific taxonomy use cases. This section is included so that it is possible to refer to specific practices using these terms in the future and to give a view on how these use cases might be described.

Within the taxonomies analysed by the TAGTF, the most common taxonomy use case was for Generally Accepted Accounting Principles (GAAP). IFRS, Belgian GAAP, Spanish GAAP, US GAAP and UK GAAP are some of the examples of a GAAP use case and the basis for an XBRL taxonomy. There are, however, other use cases for building a taxonomy, for example:

- The Global Reporting Initiative (GRI) taxonomy that is based on and used for sustainability (financial *and* non-financial) reporting; and
- The Common Reporting (CoRep) taxonomy that is based on the capital requirements of amended Directives 2006/48/EC and 2006/49/EC.

Even within the set of taxonomies defined by the GAAP use-case, there are many different taxonomy architecture approaches used. The lack of similar choices made on the architecture for this set of taxonomies indicates that it is not a straightforward task to provide a description of a taxonomy that can be used in comparison to others. In order to address this issue, the TAGTF defines an approach that describes each taxonomy as being a combination of two factors:

- The **business domain** that it addresses; and
- The **final intended usage** of the taxonomy.

Examples of **business domains** include:

- Compliance reporting, including:
  - Tax/revenue collection;
  - Banking supervision;
  - Insurance supervision;
  - Capital markets reporting;
- Financial and non-financial reporting, including:
  - Sustainability/CSR reporting;
  - Integrated reporting;
  - Management reporting;
  - Annual/quarterly reports;
- Corporate governance;
- Corporate Actions;
- SBR; and
- Risk management.

Examples of the **final intended usage** of the taxonomy include:

- Define metadata for use with other XBRL taxonomies such as business rule and relations between report elements;
- Representing or replacing Excel-based/spreadsheet-based reporting;
- Formalising the reporting elements in a set of reporting principles;
- Providing a specialist taxonomy module to be reused within other taxonomies; and
- Providing information around data relationships and connectivity.

Throughout the XTGD, when taxonomies are described, reference may be made to the business domain, final intended usage or both. This was done to help the reader assess how closely the reviewed taxonomy matches the user's own situation and needs.

5.2 Scope of what needs to be represented in the taxonomy

As a general definition, taxonomies orderly classify something according to its presumed natural relationships. An XBRL taxonomy orderly classifies reporting concepts according to their presumed natural reporting relationships. These reporting concepts typically belong to a well-defined reporting framework. The principles, directives and laws found within this reporting framework normally define the scope of what needs to be represented in an XBRL taxonomy.

It is good practice to develop an understanding of the data being reported and its granularity in terms of both the end-use of the data and how the data is to be reported. Ideally, since XBRL taxonomies can capture and build a number of different relationships between pieces of information, it is important to have an understanding of individual data points (e.g., their type and form) and all the relationships between the pieces of data so that they can be reflected in the taxonomy architecture.

5.3 Using the XTGD in taxonomy design

5.3.1 Expected usage of the XTGD

The XTGD is intended to be used when:

- A requirement must be fulfilled and the reader is looking for guidance on good practice in architecture to implement this feature; and/or
- The reader is looking for guidance on good practices in taxonomy architecture to handle a specific business reporting need or requirement through XBRL.

The XTGD covers both of these in whichever of these ways is most natural to discuss good practice around a particular aspect of taxonomy architecture.

5.3.2 Content of each section

For each taxonomy aspect under consideration, specific business requirements or taxonomy aspects are taken and described. The following content can be expected within each section:

- A description of the business requirement or taxonomy aspect under investigation;
- A summary of approaches taken, including a description of each approach, and an indication as to whether approaches are used by all of the taxonomies analysed, some, or none;
- The TAGTF analysis, including benefits and challenges of each approach, or when each approach is relevant; and
- Conclusions, including good practices in taxonomy architecture.

5.3.3 Alignment of ordering of taxonomy aspects and design decisions

In the writing of the XTGD, is was considered how decisions made about one aspect of taxonomy architecture will have an impact on other decisions, making some choices more logical or preferable than others for the subsequent decision(s).

The XTGD strives to present taxonomy aspects in the order in which they are likely to be defined. In this version of the XTGD, the following taxonomy architectural decisions are covered in this order:

- Deciding whether or not to create a taxonomy designed to be extended;

- Deciding how to model tables and lists (e.g., whether a taxonomy that is designed to be extended should use explicit dimensions to which the reporter must add its own members); and
- Defining the purpose and use of labels – decisions made about the modelling approaches used can inform and affect the way in which labels are constructed.

6. Taxonomy extensions

A valuable feature of XBRL is the ability to extend taxonomies, that is, the ability to create a new taxonomy based on an existing taxonomy. Whether this feature is applicable to a particular taxonomy will, to a large extent, depend on the filing program (or "reporting system") for which the taxonomy is intended to be used.

6.1 Open vs closed reporting

XBRL is used to represent a wide variety of different report types. It is often helpful to divide these report types into "open" and "closed" report types.

In a closed reporting system, a report may only contain reporting items drawn from a fixed list. This is analogous to many paper forms, where reporters may only enter data into the boxes provided.

In an open reporting system, the set of permissible items is not (and often, cannot be) completely enumerated by the receiver, and instead, the reporting entity may introduce additional items as required. Open reporting systems are most commonly found where financial reports adhere to a principles-based accounting standard, as the exact items that must be reported in order to comply with the prescribed principles may vary from entity to entity.

Whether a reporting system is open or closed will be a key factor in determining whether it is appropriate to use taxonomy extensions, although there is not a hard and fast link between the two: it is possible to implement an open reporting system without extensions, and conversely, extensions may have a role to play in a closed reporting system.

It should be noted that there is no mechanism in XBRL to indicate whether a taxonomy is intended to be extended or not, and as such, it is not meaningful to describe the taxonomy itself as being "open" or "closed". However, whether a taxonomy is intended to be used with extensions may influence a number of taxonomy architecture decisions.

6.2 Should the taxonomy be designed for extensions?

The decision of whether or not to design a taxonomy for extensions should be driven by the business requirements of the reporting environment in which it is intended to be used.

In many cases, the decision will be straightforward, for example:

- Where a regulator is creating its own taxonomy in order to implement a closed reporting system, then a taxonomy without extensions would be the natural choice.
- Where a taxonomy represents an international reporting standard which is adopted with regional variations, the taxonomy should be designed for regulatory extensions to capture those variations.

In other cases, the decision will be more complicated, for example:

- Where a regulator is implementing an open filing system, a choice must be made between the completeness and granularity of tagging offered by filter extensions, and the relative simplicity of a fixed taxonomy.

As discussed below, there are a number of alternative approaches that can be adopted to implementing an open filing system without using filer extensions.

A number of specific use cases are analysed discussed below.

6.3 Regulatory extension vs preparer extension

In general, the extension of a taxonomy can be done both by regulators and preparers. A regulator may wish to extend an existing base standard taxonomy (for example, the IFRS Taxonomy). The regulator decides if this new taxonomy must be used by a preparer as is (not further extension allowed) or that the preparer must (or may) add further concepts and structures.

A preparer may be then be required to, or choose to, extend the base taxonomy specified within a filing system.

For example the IFRS Taxonomy is extended to produce the Danish IFRS Taxonomy. This can then be extended by preparers during filing.

In the following sections we have written the guidance to apply generally to both regulator and preparer extensions. We will explicitly state cases where the guidance given applies more to one or the other.

6.4 Analysis of extensions and use cases

In order to support the analysis, the following levels of XBRL taxonomy extension have been defined:

- **No extensions**: In the case of preparers: where the filing program has disallowed any taxonomy extensions, and XBRL instance documents must be written against the approved, in-use, published XBRL taxonomy. In the case of regulators: they have specified a base standard taxonomy supplied by another body to be used directly by their filers i.e. they have created no extensions themselves ;

- **Limited extensions**: Extensions contain either presentation changes that do not modify the original set of data points (e.g., translation of labels) or create new data points that go beyond the scope of the base taxonomy (e.g., sector-specific KPIs). While the criteria for allowed extensions could be different across implementations, the extension taxonomy continues to describe the complete set of data points defined in the base taxonomy; and
- **Unrestricted extensions**: Extensions contain additional concepts and linkbases that add to or override parts of the base taxonomy. These extension taxonomies do not describe the complete set of data points defined in the base taxonomy.

A summary analysis of where extensions are permitted by preparers can be found in Appendix C. The following observations are seen in that analysis:

- In some cases, the taxonomy is used to support an open filing program where unrestricted extensions are required in order to arrange or add to the concepts provided in the base taxonomy to meet the needs of the reporting program;
- In some cases, the taxonomy is only used as part of an extension taxonomy. In these cases the taxonomy is described as a definitional taxonomy, and unrestricted extensions must be created to define reports that reuse its content;
- In some cases, the taxonomy is used to support a closed filing program where limited extensions are allowed; and
- In some cases, the taxonomy is used to support a closed filing program that enforces no extensions, explicitly does not allow extensions, and only focuses on a defined data set.

6.4.1 Specific use cases

6.4.1.1 No extensions: fixed data points reporting use case

The decision to not plan for extensions is made when the taxonomy presents a fixed set of data points, and these are the only data points that are required and allowed. Allowing extensions in this case may make the data contained in the instance difficult or challenging to interpret.

An example from the taxonomies analysed is the SURFI taxonomy whereby specific data requirements for regulatory reporting are set out in regulations and legislation.

6.4.1.2 Limited extensions: international use case

Limited extensions are considered in the scope of some international taxonomies (e.g., those used for banking or insurance supervision in Europe). Extensions may be limited to introducing labels in specific languages (e.g., French labels in France), or to introduce specific assertions (e.g., to test that the monetary unit used corresponds to what is accepted - GBP in United Kingdom).

An example from the taxonomies analysed are the CoRep version 1 national taxonomies (released in 2006). This filing program is not considered open and only limited extensions (that do not add or remove data points) are allowed.

6.4.1.3 Unrestricted extensions: framework or definitional use case

Unrestricted extensions are planned for when the aim of the taxonomy is to provide a framework or a set of definitions that are not designed to be reported against directly. In this case, either the regulator or the preparer is required to create an extension.

An example from the taxonomies analysed is the IFRS Taxonomy – where general principles for what is to be reported are set out, and it is left to the user of the taxonomy to determine any specifics extensions that may apply to their business, industry, or sector.

When the user of the framework or definitional taxonomy creates an extension for their own purpose, then that user should also consider whether to plan for further extensions. For example, the Chile IFRS taxonomy is extended from the IFRS core taxonomy and allows further extensions.

6.4.1.4 No extensions: alternative approaches to open reporting systems

An overriding feature of open, principles-based reporting is that the preparer needs to apply judgement when deciding which information to disclose and how to present it. This would suggest that planning for extension by the preparer, as described in the preceding section, is appropriate. However, it has been observed that the requirements of some filing programs incorporating principles-based financial reports are met with no extensions.

Examples of how this has been done in the observed taxonomies are listed below.

- Use Inline XBRL (iXBRL) to provide both tagged (XBRL and HTML) and non-tagged (HTML only) data within an iXBRL instance document;
- Include reporting concepts that allow for additional information, for example, concepts that capture blocks of free text; and
- Use footnotes to add additional information to the XBRL instance document.

Whether a given approach is appropriate will depend on the specific mandate and business requirements of the implementation. These approaches, and the considerations surrounding their use, will be discussed in future versions of the XTGD.

6.4.2 Summary of specific use cases

The research data shows that there are relevant use cases for both planning for extensions or not and that the expected level of extensions is a function of the expected usage of the taxonomy rather than the business domain.

Different taxonomies were expected to support different types of extension leading to the definition of extension levels which depended on whether the base taxonomy's data structure was expected to be intact in extension taxonomies.

Use cases found in the taxonomies analysed show how expected levels of extension can be categorised and the way in which the decision to design for extensions must take into account the reporting program that the taxonomy supports.

6.5 Good practice when planning for extensions

It is good practice to plan for a level of extensibility that correctly supports the *filing program requirements* rather than attempt to define the *taxonomy* as being open or closed.

Once a decision has been made to design for taxonomy extensions, there are several good practices to consider:

- **Include strict, clear, consistent instructions on how taxonomy extensions should be created.** Specific features should be in place to allow for the taxonomy to be extended (e.g., taking a modular approach to different taxonomy sections, or, rather than define dimension members for an area that is known to be variable, provide an obvious place for extenders to add their own dimension members).
- **Only allow extensions that follow the rules described in the extension guidance document.** This means that any design features included and described for the base taxonomy continue to be valid and leveraged in the extension taxonomy for use by reporters or further extensions.
- **Enforce the extension rules at the reporting program level.** This helps maintain high quality data by making reports that do not follow the extension rules invalid for submission.
- **Provide empty schema and linkbases to hold the extension elements.** This helps the creator of the extension by giving clear guidance on where to put extension elements without the need for paper documentation.

- **Analyse and add extensions to the base taxonomy where appropriate.** Establish a taxonomy maintenance process to analyse and include common extension elements from corporate or regulator extension as best/common practice.

Where extensions are used, additional complexity in analysing such reports can be mitigated by the following good practices:

The creators of the extensions should:

- Not redefine existing data points already included in the core taxonomy;
- Maintain established taxonomy quality rules (e.g., namespaces from the base taxonomy should not be changed); and
- Taxonomy architecture of the extensions should be aligned with the base taxonomy, including the usage of taxonomy features.

These good practices aid in the understanding of the extension taxonomy, primarily by ensuring that the features and documentation of those features defined for the base taxonomy are reused in the extension and that consistency is achieved from base to extension taxonomy.

Good practices for creating extensions include:

- The extended taxonomy should refer to the base taxonomy using the canonical location of the base taxonomy. In this way, no changes are made to the base taxonomy files;
- The extension taxonomy adjustment should be contained in an identifiable and separate set of files;
- While exchanging the XBRL instance, the extended taxonomy should be readily available. Either hosted online or included as part of a file set containing both the instance and the extension; and
- The extended taxonomy should follow the taxonomy architecture defined for extensions where provided or take account of conventions used in the base taxonomy where no extension architecture is defined.

7. Modelling of tables or lists

Many reporting requirements require a breakdown of information by one or more dimensions, often represented as *tables* or *lists*. In this section, four types of tables will be presented that make up the most commonly used table structures in observed reporting requirements. Table 10 in Appendix C shows the use of these structures in the XBRL taxonomies that were analysed by the TAGTF. The TAGTF is aware of other structures being used (e.g. combining tuples and explicit dimensions); these will be analysed and added to this chapter in a future version.

7.1 Examples of table and list types

7.1.1 Example 1: one dimension – one or more members to be reported against, distinct members not known in advance

Reporting Requirement: Determine which countries/regions have sales in excess of 5% of total sales. Report for those countries/regions the following:

- Sales;
- Costs; and
- Number of employees.

A possible table representation could be:

Country/Region	Sales	Cost	Number of Employees
France	200	50	3
Spain	300	80	4
APAC	400	100	5

The characteristics of this structure are that

- The breakdown uses one dimension (country/region); and
- It is not known *in advance* how many items will be reported (one or more).

While the 5% requirement means that a maximum of 20 members will be required, the issue is whether the complete set of members can be practically enumerated by the taxonomy author.

7.1.2 Example 2: Unordered list of items

Reporting requirement: Report all awards received in the reporting period.

A possible table representation could be:

Award
Best employer of the year 2011 in the Netherlands
Company with the lowest level of injuries in the automotive industry in Europe for the period 2010 – 2011

In this case, the reporter must supply all values as specified in the requirement but the number of them is unknown. There is also no requirement to be able to identify them individually or order them. Overall the values are more likely to be variable and more like data than more stable meta-data.

7.1.3 Example 3: one dimension - one or more members to be reported, number of each member known in advance)

Reporting requirement: Report the total volume of water withdrawn in cubic meters per year (m3/year) by the following sources:

- Surface water, including water from wetlands, rivers, lakes, and oceans;
- Ground water;
- Rainwater collected directly and stored by the reporting organization;
- Waste water from another organization; and
- Municipal water supplies or other water utilities.

A possible table representation could be:

Water Source	m3/year
Surface water, including water from wetlands, rivers, lakes, and oceans	50
Ground water	0

Rainwater collected directly and stored by the reporting organization	100
Waste water from another organization	20
Municipal water supplies or other water utilities	300

The characteristics of this structure are that:

- The breakdown uses one dimension (in this example, water source); and
- It is known in advance how many items will be reported (in this example, five).

7.1.4 Example 4: two dimensions - one or more members to be reported for each dimension, number of each member not known in advance

Reporting requirement: provide revenue per operating segment and geographical market.

A possible table representation could be:

Revenue	Denmark	Sweden
Bikes	200	300
Cars	400	500

The characteristics of this structure are that:

- The breakdown uses two dimensions (in this example, operating segment and geographical market); and
- It is not known in advance how many items will be reported (one or more for each dimension), only the reporter will know which operating segments are used and what they define as a geographical market (i.e., it might be a country, or a group of countries such as the Nordic countries).

7.1.5 Example 5: two Dimensions – for one dimension, number of members known in advance; for second dimension, number of members not known in advance

Reporting requirement: Provide the number of employees broken down by gender and region.

A possible table representation could be:

Number of Employees	United Kingdom	Nordic Countries
Male	200	300
Female	400	500
Unknown	10	5

The characteristics of this structure are that:

- The breakdown uses two dimensions (gender and region);
- For one dimension - gender - the number of members is known in advance; and
- For the other dimension - region - it is not known in advance how many items will be reported.

Example 5 is a combination of examples 2 and 3 above and demonstrates how the each dimension could be considered separately in order to model multidimensional tables. In addition, the combination of dimensions should be considered as a whole to see if the resulting structure is usable.

7.1.6 Summary of Examples

In summary, each table can be characterized by:

- The number of dimensions used;
- Whether the reported members of those dimensions are known at the time of creation of the taxonomy (or are only known to the reporter); and
- The number of members in each dimension.

7.2 What XBRL offers to model common table structures

XBRL provides multiple ways to model table structures:

- **Tuples**; and
- Dimensions (two types):
  - **Explicit Dimensions**; and
  - **Typed Dimensions**.

In the observed taxonomies, a small number of taxonomies make use of tuples or typed dimensions. Explicit dimensions are a *common* structure, found in *most* of the taxonomies analysed. Taxonomies may use one, some or all of these approaches to correctly model the tables that are required.

The choice of data structure is one where there is little agreement across the taxonomies analysed. A typed dimension may seem a natural choice when the possible values cannot be defined in the taxonomy (e.g., asset or customer identification), or when the number of possible values is large (e.g., countries).

However, explicit dimensions were also used even for large numbers of value (e.g., countries, currencies) and some taxonomy architectures forbid the use of typed dimensions and request the preparers to define unknown *a priori* value through taxonomy extensions, defining explicit dimension members.

Other taxonomy architectures forbid the use of tuples requiring that only dimensional data structures be used.

Given the wide range of decisions made around which data structures to use for tables, the following sections analyse the pros and cons of the different XBRL options from a neutral viewpoint to allow decisions to be made with a view of the impact of those decisions.

7.2.1 TUPLES

Tuples are a hierarchical structure that groups a number of items and/or other tuples for use as an *unordered* list. An example of a tuple is a single address where a street name, a house number and a city name are grouped to give a full address. This grouping can then be reported multiple times to give an unordered list of addresses.

7.2.2 DIMENSIONS

Dimensions are defined in the XBRL Dimensions specification as being defined through an XBRL concept. Dimensions can be used to define disaggregation or breakdown of data (e.g., "country of sales", "type of customer") or characteristics of the data (e.g., "computation method", "before or after mitigation of risks").

Two types of dimensions are defined by the XBRL specification:

- **Explicit dimensions**: the possible values of the dimensions are defined in the taxonomy as *XBRL concepts* and called a "dimension member."
- **Typed dimensions**: the possible values are defined through an *XML type* (simple or complex), hence the name. Typically, the values of typed dimensions are not defined in the taxonomy, except when the type is an enumeration.

Another way of looking at the difference is that: with explicit dimensions, dimension members are defined in the taxonomy by the taxonomy author, whereas with typed dimensions, members are defined in the instance by the reporter.

7.3 Pros and cons of tuples and explicit/typed dimensions

TABLE 1: TUPLES

Tuples Pros	Tuples Cons
Have very clear item grouping, visible to a user (e.g., a list of addresses, list of Board members). Allow for data structures that do not require an understanding of dimensional data. Some reporting programs choose tuples because they can lead to smaller instance documents.	The structure of a tuple cannot be changed by an extension taxonomy. There is no natural unique ID for items in a tuple so it is not possible to identify a specific tuple or item unambiguously without referring to syntax-level content of an instance (e.g., in XII Formula error messages). The use of multiple level tuples leads to a hierarchical representation of data that is more appropriately handled by dimensions (e.g., financial data with multiple breakdowns).

TABLE 2: DIMENSIONS

Explicit or Typed Dimensions Pros	Explicit or Typed Dimensions Cons
Allows for a dimensional (rather than hierarchical) representation of data that is the same design as the dimensional data stores that are likely to be in place for both the reporter and the consumer of report. Non-duplicate facts in an instance can be unambiguously identified by the combination of its concept and its context (which contains the dimensional information).	Requires an understanding of multi-dimensional data. There is an additional constraint when using dimensions that repeating information requires data that makes each fact unique (e.g. implementing a list of awards using dimensions would require a dimension such as Award ID that gives uniqueness to each award). This approach can lead to additional data being added that carries no semantic meaning.

TABLE 3: EXPLICIT DIMENSIONS

Explicit Dimensions Pros	Explicit Dimensions Cons
This data model is one that feeds into a common understanding that something can either be broken down (as in sales-by-product) or '#tagged' (as in Twitter) using fixed options. A reader of an instance can identify sets of data for in one or more documents and be sure that they are comparing similar items (e.g., dimension member “namespace:Automobiles” in one breakdown has the same meaning as in any others in which it appears). When standard lists, such as ISO country lists, are used, this is equally true across different taxonomies. You know the complete set of possible data before you start to write business rules or other data-based logic, whereas with other data structures you do not. The structure of an explicit dimension can be changed by an extension taxonomy in a backwards-compatible manner (e.g., adding a new member to a breakdown). Allows assignment of a default dimension member which can provide additional functionality.	Explicit dimensions require the list of members to be known and stable over a time comparable to the release cycle of the taxonomy. The taxonomy creator must define a comprehensive list of the members that would be possible. This might be challenging, e.g., when defining the members for a dimension for “Region” the regions required by users may vary for example, “Asia-Oceania” and “Australasia”. There are interpretive issues for readers of instance. If someone does not appreciate that the combination of both the namespace and local name of a dimension member are required to tell that one dimension member is the same as another, they may find that they are not comparing like with like. (e.g. “namespace1:Cars” may not mean the same as “namespace2:Cars.” Maybe the definition in namespace2 also includes buses.). Since members have no 'start' and 'end' date on them to allow them viable in a dimension or not, the taxonomy must take care of this versioning aspect of dimensional content. This can be troublesome if the content is determined by third parties (like ISO).

TABLE 4: TYPED DIMENSIONS

Typed Dimensions Pros	Typed Dimensions Cons
Provides reporting flexibility - values are not constrained and can cover an infinite or extremely large number of unexpected	As the members will differ from instance to instance, they can’t be used for comparison without an analysis first to see if they are

values. No need to use extensions to capture information only the reporter can know (e.g., product lines). This reduces the need for taxonomy extension and thus can be used to capture information only the reporter can know (e.g., product lines). This means that software applications created for the reporting need can be simpler as only the instance has to be created, not a taxonomy as well. No need for the reporter to invest in software and knowledge for creating taxonomies.	the same. Flexible feature of XBRL that could be misused where it might be more appropriate to have other data structures.

7.4 Comparative view of the XBRL data model options

Aside from the suitability of a particular data structure from a modelling viewpoint, it is good practice to take into account the practicalities of choosing a particular model. A comparative view of the practical usage of the taxonomy is given in Table 5 to augment the pro/con analysis above. This focusses on particular expected usage of the chosen data model and gives specific considerations to help choose or implement a particular model.

TABLE 5: PRACTICAL CONSIDERATIONS WHEN CHOOSING SPECIFIC DATA MODELS

Expected use of the taxonomy	Considerations for tuples	Considerations for explicit dimensions	Considerations for typed dimensions
The data in the instance document is used in automated processing	XBRL does not provide a mechanism for uniquely identifying specific items in a tuple list so any operations should be considers covering all instances of a specific tuple	None identified	The constraints chosen for the dimension type should be chosen to help use typed members in automated processes
Additional data points need to be	No extensions are required. A tuple can	Extensions are required to add	No extensions are required. Additional

added beyond those defined by the taxonomy authors	be reported multiple times, creating additional data points	additional data points	dimension members are defined in the instance, creating additional data points
Data submitted in instance document should be comparable across instances (e.g. the value of a data point in one instance should be directly comparable to the equivalent data point in another instance)	The layout of the tuple remains constant between instances. This lends itself to some comparability at the tuple level rather than the individual fact level.	Data points are directly comparable between instances, due to explicit nature of dimension qualification.	The constraints chosen for the dimension type should be chosen to help comparability of data. Chosen and made known by the reporter (for comparing multiple reports from the same reporter) or the taxonomy author (for comparing reports across multiple reporters).
To understand what needs to be reported	A user must be able to identify the elements that make up a tuple	A user must be able to identify the hypercube, dimensions and dimensions members that make up the definition of the data structure	A user must be able to identify the hypercube, dimensions and dimension types that make up the definition of the data structure
Impact on taxonomy maintenance (Assuming other factors being constant)	Taxonomy will need to be updated when the structure of tuple has to undergo a change	Taxonomy will need to be updated when the list of domain members is to be changed/updated and when reporting items for the dimension undergo a change	Taxonomy will need to be updated when reporting items for the dimension change

7.5 Good practice when modelling tables and lists

It is important that clear decisions are made around which XBRL features to use for modelling data that can be described as a table or a list. It is also important to understand that good reasons can be formulated for using all or any of the described approaches for the same structure depending on the use case.

The following good practices can be applied when deciding on appropriate data models to use in the taxonomy:

- Decide whether or not the preparers must make extensions to capture the information, and whether tuples, typed dimensions or explicit dimensions are allowed to be used;
- Describe common structures (such as in the examples above), decide how to model them, and apply the same approach to each occurrence;
- Try not to be constrained by arbitrary decisions on whether or not to use particular models. Taxonomy creators should evaluate all three and adapt what suits the requirements.
- Document the decisions regarding the modelling of table structures in the taxonomy architecture guide.

7.5.1 Examples of good practice in data modelling

The table below shows how the various examples described above can be modelled using tuples, typed and/or explicit dimensions:

TABLE 6: GOOD PRACTICE IN DATA MODELLING

Examples (See Section 7.1 for full description)	Possible Approaches
Example 1 – table of one dimension, unknown members	Tuple approach allows for all the possible data to be reported: • **Tuple**: SalesCostEmployeesPerCountry • **Containing items**: CountryName, Sales, Cost and Number of Employees. Typed dimension approach allows for all the possible data to be reported: • **Hypercube:** SalesCostEmployeesPerCountry • **LineItems**: CountryName, Sales, Cost, Number of Employees • **Typed dimension**: country_identifier (string)

	Explicit dimension approach that allows for a limited set of data to be reported, under the condition that extensions are not possible: • **Hypercube**: SalesCostEmployeesPerCountry • **LineItems**: CountryName, Sales, Cost, Number of Employees • **Explicit dimension**: countries (string) • **Domain**: max_20_countries_domain • **Domain members**: country-01, country-02, .. , country-20
Example 2, a list of items	Tuple approach: • **Tuple**: AwardsReceived • **Containing items**: AwardDescription. Typed dimension approach: • **Hypercube** AwardsReveived • **LineItems**: AwardDescription • **Typed dimension**: Award_identifier (string)
Example 3, table of one dimension with known members	Create a hypercube with an explicit dimension: • **Hypercube**: WaterWithdrawalPerSource • **LineItems**: WaterWithdrawn • **Explicit dimension**: WaterSourcesDimension • **Domain**: WaterSourcesDomain • **Domain members**: SurfaceWaterMember, GroundWaterMember, RainWaterMember, WasteWaterMember, MunicipalWaterMember.
Example 4, multi-dimensional table with unknown members.	Use a typed dimensions or explicit dimensions as described in example 1. Do not use tuples for the reasons described in the “Tuples good practice” section.

7.5.2 Good practice for tuples

The following good practices cover the usage of **tuples**:

- Do not use nested tuples to define multi-dimensional table structures. It adds additional complexity that can be avoided, especially if the inner and outer tuple can appear multiple times (e.g., a list of people with roles each having multiple addresses). Since the data can most likely be represented in a different way that avoids this complexity, it is good practice to avoid this approach.
- Consider using a tuple for a one-dimensional table of unknown length (e.g., as seen in example 2) wherein the dimension would just be a meaningless identifier (e.g., sequence number). In this case, a tuple is the simplest solution available for the reporter as it requires no additional contexts in the XBRL instance.
- If the table has only one dimension and the amount of members is potentially unlimited (e.g., info on each car produced), consider using a tuple.

7.5.3 GOOD PRACTICE FOR TYPED DIMENSIONS

The following good practices cover the usage of **typed dimensions**:

- Add a concept to the table structure to capture a descriptive name for the typed dimension member (e.g., for a geographical typed dimension, define a line item country/region description). In that way the user of an instance does not have to decode the context to know for which country/region the data is reported.
- Ensure that no two members share the same description. This would be achieved by a general filing rule that prohibits inconsistent duplicates. Inconsistent duplicates are technically legal under XBRL v2.1.

  Example:
  Hypercube SalesCostEmployeesPerCountry
  LineItems: CountryName, Sales, Cost, Number of Employees
  Typed dimension: country_identifier (string)

  Context D2012_Typed_ID_1 uses country_identifier = 1 for the year 2012
  Context D2012_Typed_ID_2 uses country_identifier = 2 for the year 2012.

  Situation to avoid in the instance is:
  <CountryName context=” D2012_Typed_ID_1”>Belgium</CountryName>
  <CountryName context=” D2012_Typed_ID_2”>Belgium</CountryName>

  <CountryName context=” D2012_Typed_ID_1”>Belgium</CountryName>
  <CountryName context=” D2012_Typed_ID_1”>France</CountryName>

- Forbid complex types (XML structures) within a typed dimension since this leads to a more complex model. The alternative in this case is to use several simple typed dimensions rather than a single complex one.

- Consider using a more specific type than just number or string to guide filers. For example, use a XML format to force the string to be a car registration number or a restriction to limit the number of characters used.

7.5.4 Good practice when designing for extended taxonomies

The following good practices cover the usage of **taxonomy extensions**:

- As the preparer has to create an extension taxonomy, use explicit dimensions for structures that will have a limited number of members and let the preparer create the required members. In general explicit dimensions work best in situations where the members represent meta-data, e.g. the names of regions in which a company does business.
- Use a typed dimension where the number of members will be very large (e.g., list of cars sold with the exact price for each individual item) to avoid having a very large taxonomy.
- In general typed dimensions are more useful when the members represent something more like data, e.g. a list of car registrations.

8. Labels

Generally, the schema and linkbases (which define relationships between concepts as defined in schema) are defined in a manner that enables computer systems to read and process the associated files. Taxonomy labels provide the human readable interface for the taxonomy. It is good practice to define label usage as part of a taxonomy's architecture to allow for this.

The most basic labelling scheme is to use the label linkbase to define a single label for concepts using the *standard role*, this role is defined in the XBRL Standard with no specific meaning attributed to it. However, most taxonomies require more advanced labelling schemes to meet the business requirements. This section presents two alternatives to applying labels to a taxonomy, the label linkbase beyond the standard role and the generic linkbase. A section covering implementation detail of how to construct labels considers the construction of a standard role label but can also be applied when using the more advanced labelling schemes.

8.1 Label linkbase and labels beyond the standard role

The XBRL mechanism for defining labels beyond the standard role is label roles. XBRL label roles are used to assign a specific meaning to a label. An example of this is the documentation label role that can be used to document the reference or the meaning of a concept.

Use of additional label roles was observed in all taxonomies analysed, with most taxonomies using more than one label role for their concepts. The most commonly used label roles (not including the standard label role) were terse label, period start, period end, and total labels, with 13 of 24 taxonomies using these label roles (albeit not always to achieve the same purpose). In addition, 10 taxonomies have used the documentation label role.

Use cases for additional label roles are:

- The standard role is not used in favour of a more specific definition of what information the label is conveying;
- The concept is used in multiple ways and those differences must be reflected in the labels used (e.g., value at the start of the period and value at the end of period. The label roles to use here would be period-start and period-end);
- When there are different names for the same concept (e.g., Natural Capital vs. Resources); and
- To gather different parts of the complete metadata that surrounds a concept so that they can be viewed separately (e.g., the description of a concept, the use of a concept, the legislative instrument that supports collection of this data).

When one of these or another use case applies, and label roles beyond the standard role are required, good practice in choosing a label role is to follow the steps below, in order, until a suitable label role is defined:

1. Select a label role from the XBRL specification;
2. Select a label role from the Link Role Registry (LRR); and then
3. Create a new label role and register this with the LRR.

The first step is to determine whether the desired label usage is already accounted for in the XBRL specification. The XBRL specification includes pre-defined roles, and it is good practice to use these where they fit the taxonomy author's desired use. From the analysis of the taxonomies, it is observed that most of the taxonomies use the existing XBRL standard label roles. Given that a suitable role is not found, the next steps involve the use of the LRR (as discussed in the next Section).

8.1.1 The Link Role Registry (LRR) and new label roles

The XBRL specification has provided the flexibility to define new label roles to fit the requirements of taxonomy authors. New roles can be registered in the LRR[1] that includes many more label roles than the XBRL specification. If the label roles in the XBRL specification do not fit the desired usage, then it is good practice to reuse a label role from the LRR (which has therefore been defined for another taxonomy).

If the LRR does not contain a role matching the requirements, then a decision should be made as to whether to use an inexact role or create a new one. The pros and cons of creating new roles are given in the table below.

TABLE 7: PROS AND CONS OF CREATING NEW LABEL ROLES

Pros	Cons
New label roles help to represent the content in more appropriate manner.	Addition of label roles makes the taxonomy less standard.
If registered with LRR, the label roles are available for other taxonomy developers.	New roles may not be understood by consumers of the taxonomy or their software.
Inexact usage of existing label roles may be	

[1] Link role registry: http://www.xbrl.org/LRR

misleading.

In general, and given the pros and cons listed above, it is good practice to create a new label role if existing roles do not match the desired purpose.

8.1.2 Good practice for labels beyond the Standard Role

The following good practices cover labels beyond the standard role:

- Use label roles to define the specific meaning of a label;
- Favour the use of existing label roles, firstly from the XBRL specification, and secondly from the LRR;
- Define new roles only if the existing roles are not suitable to represent the information;
- Register any new label roles with the LRR; and
- Consider whether labels assigned to other label roles should have a specific construction as in the label construction section.

8.2 Generic labels

An alternative to the use of the label linkbase is the use of the Generic Links specification. Generic links is an XBRL specification for creating any type of linkbase - its goal is to provide support to the existing linkbases and aim to overcome the limitations of traditional linkbases.

The functioning of the generic label linkbase is similar to the label linkbase. The main technical difference between the two: while the label linkbase is limited to defining labels for concepts within a taxonomy, the generic linkbase can be used to create labels for any XML element. In the taxonomies analysed, generic labels were used to label extended link roles (ELRs), custom data types and enumerations. In particular, one of the most common uses was to define labels in multiple languages for ELRs.

Use cases for the generic label linkbase are:

- **As a replacement to the traditional label linkbase**: In this case, all the labels (whether for elements, ELRs or other artefacts) are defined using the generic linkbase. From a processing point of view, use of a single type of linkbase could be a consistent approach; however, since the generic linkbase is a relatively new addition to the XBRL suite of specifications (and not extensively used), support may not yet be available in all currently available software tools.
- **As an additional linkbase**: In this case, the taxonomy contains two types of linkbases for labels – the label linkbase for element labels, and the generic linkbase for defining labels that cannot be catered by the label linkbase. In this way, users of software that does not support generic labels will still have access to partial labelling information.

Pros and cons of whether the generic linkbase is a good fit for a taxonomy are given in the table below.

TABLE 8: PROS AND CONS OF USING THE GENERIC LINKBASE FOR LABELS

Pros	Cons
Generic labels can be defined for any XML element, which overcomes the limitation of the label linkbase.	If created along with the label linkbase, additional effort will be required to ensure that the labelling is coherent across the taxonomy and consistent in how they are presented in software using the taxonomy.

8.2.1 Good practice for Generic Labels

The following good practices apply to the use of generic labels:

- Use generic labels:
  - for taxonomy components that cannot be labelled using the label linkbase; and
  - where the label linkbase does not allow the flexibility required.
- Ensure that the usage of generic labels and the relationship to the label linkbase is well defined so that taxonomies can be made readable in software applications that use the taxonomy.

8.3 Good practice for "Standard" label construction

This Section covers standard label constructs whose primary purpose is to make the concept readable rather than any other purpose. Label construction refers to guidance or rules on the wording to be used for these items in order to make the report understandable once it has been rendered in an XBRL viewer. There are no significant technical limitations on how a label should be constructed so this can be considered a business-led decision.

A consistent approach is recommended when constructing labels. Consideration should be given to whether there is an existing label construction in the base of the taxonomy and whether labels can be unique. Labels should accurately represent the concept.

General good practice for standard label construction is described below:

- **Uniqueness**: In the same way that having unique names for concepts allows a computer to unambiguously identify a concept, having unique labels allows for people reading them to do the same. For example, instead of having multiple concepts labelled as "Other," with a unique labelling scheme, these could be labelled to more specifically indicate their usage (e.g., "Other deductible expense"). When labels are allowed to be duplicated, reading labels in isolation can lead to mistakes in the understanding of a taxonomy or report.

- **Usefulness**: Labels should describe a concept in such a way that it accurately describes the meaning of the concept.
- **Contextless**: You should not need to know the location of the concept in a linkbase to correctly understand the meaning of the concept.
- **Conciseness**: Labels should be constructed as concisely as possible to accurately represent the concept. This should be true of all labels.
- **Purpose**: This may be so that people can navigate a taxonomy more efficiently or that an instance document can be better understood.

Regarding the effort to create or maintain labels, different approaches to creating labels will require different amounts of time and effort to create and maintain.

The taxonomies analysed showed the following approaches used to construct labels:

- **Reflect non-XBRL usage**: use wording in existing documents/templates;
- **Create new labels**: follow guidelines as to the creation of a label; and
- **Reflect structured data**: reflect the label's position in a hierarchy of data items either in the taxonomy or some other structured data definition.

Using any of these approaches can be considered good practice once it has been decided that it fits the requirements of the reporting program. Documenting the approach followed and setting up a style guide helps in achieving consistency while creating, maintaining or extending the taxonomy and can also be considered good practice.

8.3.1 Approaches

This Section describes the approaches identified in the taxonomies for the creation of labels. The approaches identified are listed below:

- Reuse existing non-XBRL labelling
- Create new labels
- Use a formal structure

8.3.1.1 Reuse existing non-XBRL labelling

This approach is characterised by taking existing non-XBRL definitions and simply copying them into the appropriate XBRL labels. This leads to natural labels that may be easy to read but may be imprecise and not unique.

This approach makes use of the fact that many XBRL programs are replacing existing reporting mechanisms. Either in forms, accounting or regulatory standards, descriptions of what should be reported has already been thought through and decided upon. Often these descriptions are part of the common vocabulary of organizations that are reporting.

TABLE 9: PROS AND CONS OF REUSING EXISTING LABELLING

Pros	Cons
Labels already exist which may mean that minimal effort is required for labelling. As the labels are a copy of the framework it is capturing, there is a clear link between the taxonomy and the existing framework that may help in understanding the taxonomy.	Labels are context-dependent. Labels may not be unique. Labels do not represent the content of the data. Lack of control over the label construction may lead to issues in clarity, consistency or accuracy of the label. In some cases, the copying of labels from non-XBRL sources may mean that the text loses its meaning (e.g., references to pages, columns, rows or other aspects of the source material that is not carried into the XBRL). There is a lack of guidelines for creating labels for new concepts.

8.3.1.2 CREATE NEW LABELS

To ensure consistency in the structure and wording of labels, while maintaining readability, this approach starts from a natural description of the concept and applies a set of steps in order to make it appropriate for the taxonomy.

Steps may include converting to terms in common use for the underlying standard or rearrangement of the words in the label to ensure that it follows a particular pattern or removing parts of a label that are superfluous given the XBRL other information that XBRL allows.

TABLE 10: PROS AND CONS OF CREATING NEW LABELS

Pros	Cons
Labels are consistent across the taxonomy. Information that is implicitly or explicitly included in a concept by the use of XBRL is not repeated in the label (e.g., if the concept is of type monetary, then there is no need for the label to restate that the concept represents a monetary amount). Leads to concise labels through the explicit removal of implied words such as "Total" and "Amount of."	This is a labour-intensive way to create a label. The guidelines essentially create a taxonomy grammar that would need to be understood by the users of the taxonomy so that they can read the labels properly (e.g., "Risk, after mitigation" and "Profit, gross"). If rigidly applying transformation steps to a natural description, it may lead to a label that does not follow normal grammar (i.e., users of the taxonomy must understand the label construction in order to understand the labels). Label construction guidelines may not translate well between very different languages (e.g., English and Japanese).

8.3.1.3 USE A FORMAL STRUCTURE

To naturally achieve non-ambiguity in the labels, this approach starts from the basis of a formal structure of concept definitions and builds the label from parts of the hierarchy in this structure. In this way, you can create labels using other metadata, leading to a label that reflects the hierarchy and drill-down approach related to either the structure of the taxonomy or an external definition source.

As an example of referencing an external data source, the Australian SBR definitional taxonomy uses ISO 11179 meta-data standard for data element names. In this example, this is ObjectClass.Property.Classword.

As an example of referencing the taxonomy itself, hierarchical labels are used in the SURFI and Belgian GAAP taxonomies.

For example, the following hierarchy is defined in the business template used to define the report:

- Assets
    - Goodwill
    - Biological assets
    - Other

This gives the following labels, shown in a presentation hierarchy:

- Assets
    - Assets, Goodwill
    - Assets, Biological assets
    - Assets, Other than goodwill and biological assets

Table 11: PROS AND CONS OF USING A FORMAL LABELLING STRUCTURE

Pros	Cons
This approach delivers unique labels without having to consider the complete set of labels.	Information external to the taxonomy may be required to understand the labels.
Labels are applied consistently across the whole taxonomy.	Labels do not use natural language constructs (e.g., grammar, sentences).
Can be used to reference other global metadata standards (e.g., ISO 11179) for names of items.	Labels must be maintained in line with the taxonomy structure (which can mean an increase in taxonomy maintenance cost).
Parts of the label creation process can be automated.	Labels change when the position of the hierarchy changes.
	Could lead to longer labels.
	Prevents the reuse of concepts within the presentation tree as the standard label would not be able to represent both locations (preferred labels would have to be used in order to re-use concepts within the presentation tree).

8.3.2 Good practice for label construction

The following is a summary of good practice for label construction:

- Prefer unique labels throughout the taxonomy;
- Choose a label construction that matches the expected user of a label (e.g., are they for use by people who define data or people who write accounts);
- Choose a label construction that best meets the business needs using the pro/con analysis presented here as a guide; and
- Document the scheme chosen, including:
  - How the labels have been created;
  - Examples of label construction;
  - Specifically, if words are removed as part of label construction; and
  - Any semantic meanings contained in labels.

It is noted that the architectural choice of label construction is one that is particularly difficult to change once the taxonomy is in production, due to the reliance on them by users. This is particularly true if there has been a direct linkage between the label and the concept name.

Appendix A. TAXONOMY OBSERVATIONS TEMPLATE

The taxonomy observations template is embedded below. To obtain copies of filled in templates, please contact the TAGTF at the following address:

TAGTF-feedback@xbrl.org

TAGTF-ObservationDocumentVer1.02.doc

Appendix B. Glossary

This is a non-canonical index of terms used in the XTGD:

Term	Notes
Base taxonomy	A taxonomy that is extended to create an extension taxonomy.
Consumer / consuming party	A person or organisation that receives or retrieves XBRL instance documents for the purpose of collecting and/or analysing the data within them.
Data point	Definition of an item that can be reported in the corresponding instances. Usually given as a combination of concept, built-in dimensions and XDT dimensions.
Definitional taxonomy	Taxonomy used to describe elements, not to be used for reporting.
Extension taxonomy	A taxonomy that imports or otherwise extends a base taxonomy.
Hypercube/Table	A multi-dimensional definition of related data synonymous with a business intelligence or data warehousing "cube". At a technical level, an XBRL hypercube is defined by the XBRL Dimensions specification which can be found on the XBRL.org website.
Instance documents	An XBRL report that refers to (hence an instance of) an XBRL taxonomy. At a technical level, an XBRL instance is defined by the Extensible Business Reporting Language (XBRL) specification which can be found on the XBRL.org website.
Reporter	A person or organisation that creates XBRL instance documents referring to an XBRL taxonomy.
Reporting taxonomy	Taxonomy used for reporting.
Taxonomy architect	A person responsible for making design decisions that are applied across the whole taxonomy.
Taxonomy author	A person or organisation who is involved with the creation of concepts in a taxonomy. Does not have to be the consumer of the subsequent instances.

Appendix C. Taxonomy Architect Skills

"Taxonomy Architect" is used throughout this document. The taxonomy architect typically has a range of skills that could be the responsibility of a single individual but is often covered by a team of taxonomy design professionals given the diverse requirements needed to fulfil this key role.

A list of core skills recommended for a taxonomy architecture role is given below:

- **Analytical Skills** – Must have an organized, disciplined attention to detail, ability to analyse and articulate the business needs, determine the level of complexity required to meet these needs, and ability to translate these needs to the taxonomy architecture.
- **Data Modelling** – Must understand data modelling methodologies and techniques, relevant taxonomy design options available, potential uses of the data (including internal and external analytics) within the information domain, and be able to bring these together to create a data model that can be used to guide the taxonomy architecture.
- **Domain Knowledge** – Must have a broad theory and technical knowledge of the business domain, current legacy reporting processes, and how reporters *will* produce instances of the taxonomy and how consumers *will* consume them under the new XBRL filing program in order to inform the taxonomy architectural choices.
- **XBRL Technical Expertise** – Must have a solid technical understanding of both XBRL and XML, including how the two technologies differ, so that the taxonomy architecture choices fully leverage the unique features of XBRL. Must also have an understanding of the technical needs of the taxonomy producers/consumers so that appropriate taxonomy design decisions can be made to facilitate correct filings.

Appendix D. LISTS OF TAXONOMIES

Below is a list of taxonomies, their expected extension levels (as defined in Section 6.2) and their stated purpose.

TABLE 12: TAXONOMY EXTENSION LEVELS AND PURPOSE

Taxonomy	Expected extensions	Taxonomy purpose
Spanish GAAP 2007	None	Accounts
ES-BE-CB 2006	None	balance sheet data
UAE	Limited	capital markets
SBR Australia 2012	Unrestricted	definition/Forms
IFRS 2012	Unrestricted	definition/accounts
US-GAAP 2012	Unrestricted	definition/accounts
Belgian GAAP 2012	None	Accounts
UK GAAP 2009	Limited	Accounts
Global Reporting Initiative (GRI) 2012	Limited	Sustainability
Danish Commerce and Companies Agency (DCCA) 2012	None	Accounts
DGI GENERAL IDENTIFICATION DATA 2011	Unrestricted	Identification
FinRep 1.0 2006	Unrestricted	definition/accounts
CoRep 1.0 2006	Unrestricted	definition/prudential
Ministry of Corporate Affairs (MCA) of India 2011	None	Accounts
SURFI (Unified FInancial Reporting System) 2011	None	Prudential

Chile IFRS 2012	Unrestricted (domain members only)	Accounts
CONTAEPA 2012	None	Accounts

TABLE 13: TYPE OF DATA STRUCTURES USED

Taxonomy	Tuples	Explicit dimensions	Typed dimensions
ES-BE-CB 2006	Yes	Yes	No
Spanish GAAP 2007	Yes	Yes	No
UAE	No	Yes	No
SBR Australia 2012	Yes	Yes	Yes
IFRS 2012	No	Yes	No
US-GAAP 2012	No	Yes	No
Belgium GAAP 2012	Yes	No	No
UK GAAP 2009	Yes	Yes	No
Global Reporting Initiative (GRI) 2012	No	Yes	Yes
Danish Commerce and Companies Agency (DCCA) 2012	No	Yes	Yes
DGI GENERAL IDENTIFICATION DATA 2011	Yes	No	No
FinRep 1.0 2006	No	Yes	Yes
CoRep 1.0 2006	No	Yes	Yes
Ministry of Corporate Affairs (MCA) of India - 2010	Yes	No	No
Ministry of Corporate Affairs (MCA) of India - 2011	No	Yes	Yes

SURFI (Unified FInancial Reporting System)	No	Yes	Yes
Chile IFRS 2012	No	Yes	No
CONTAEPA 2012	Yes	No	No

Appendix E. Taxonomy Profiles

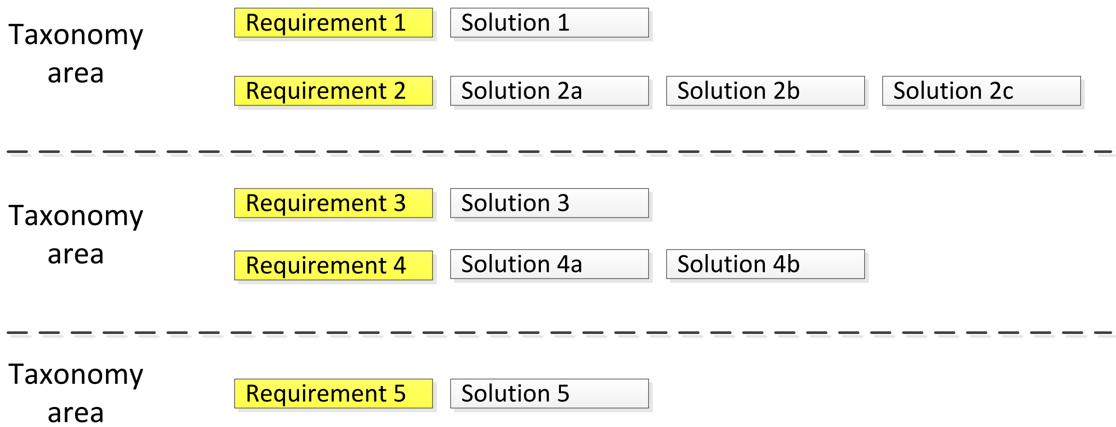
In preparing this XTGD, it has been noted by the TAGTF that there are almost as many taxonomy architecture solutions as there are business requirements. This implies that for each filing program, the decision making between defining a requirement and formulating a solution has been made as part of that filing program. These decisions represent a significant investment in testing, prototyping, design and discussion that add cost and risk to the filing program. It is a high-level goal of the TAGTF is that our work can be used to reduce the effort required to define the taxonomy architecture and give a better result. In addition, having clear guidance that leads to shared architecture decisions can be expected to make taxonomies easier to use, better leverage the efforts of others and allow reuse of software solutions.

The ability to define business requirements that the taxonomy architecture must meet and then select the appropriate solution has been condensed into the idea of "Taxonomy Profiles". Ideally, a realisation of taxonomy profiles would mean that with only a few high level characteristics of a reporting program, e.g. Prudential/Financial and Open/Closed, it would be possible to select a pre-defined taxonomy architecture that would automatically meet the known (or unknown) business requirements that the filing program has. However, it is the view of the TAGTF that there are enough differences between filing programs that the number of project specific decisions required would not be significantly reduced by attempting to define taxonomy architecture in this way. We believe that with this approach to profiles, taxonomy architects will still need to verify or adjust all parts of the generic architecture to ensure it meets their actual requirements.

Having considered the issues with the simple approach described above, the TAGTF has taken a pragmatic view of how to get as close to the ideal as possible, while acknowledging that every filing program is different. The TAGTF approach has been mapped out into three phases as follows:

Phase 1: Identification of good practice

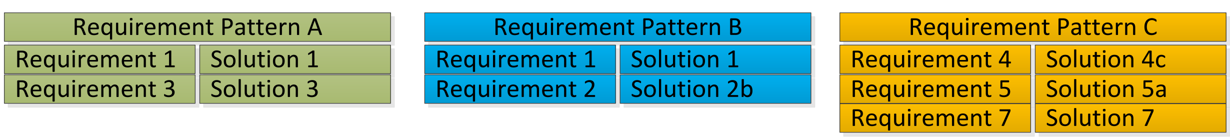
In this phase, taxonomy architecture solutions are critically evaluated looking for good practice solutions, which are linked to the business requirements and indexed.



The current XTGD architecture document shows how this can be carried out with a small number of taxonomy areas.

Phase 2: Identification of Requirements Patterns

In this phase, common groupings of requirements are identified as Requirement Patterns along with what is considered the best practice solution for requirements in this grouping. A Requirement Pattern takes into account that certain approaches have particular synergy and may only be useful when other parts of the pattern are included. A Requirement Pattern may be identified by a specific high level requirement that implies the other requirements in that grouping.



The analysis that has and will be carried out as part of Phase 1 is directly referenced from the Requirement Patterns. The Requirements Patterns will be identified using the existing and new taxonomy observations as well as those that are anticipated as being useful by the XBRL community.

Phase 3: Building Taxonomy Architecture Profiles

In this phase, good taxonomy architecture process is defined with a complete architectural profile being comprised of a number of requirement patterns, specific requirements that might not be attached to a pattern, as well as requirements that have not been previously documented. By bringing together a profile in this order, the number of project-specific decisions that need to be made is minimised.

Taxonomy profile	
Requirement Pattern A	
+ Requirement 4	Solution 4c
+ "New" requirement	Project discussion

It is expected that it will be the role of the filing program's taxonomy architect to carry out the definition of the profile. The role of the TAGTF will be to ensure that the presentation of the Requirement Patterns and the guidance for how to build a profile is made as clear and useable as possible.