Financial Reporting Taxonomies Architecture 1.0

Public Working DRAFT of Thursday, 30 October 2003

Corresponding to XBRL 2.1 Candidate Recommendation 2

Editor

Name	Contact	Affiliation
Walter Hamscher[1]	walter@hamscher.com	Standard Advantage

Authors

Name	Contact	Affiliation
Charles Hoffman	charleshoffman@olywa.net	UBmatrix
Josef MacDonald	jmacdonald@iasb.org.uk	IASCF
Jeffrey Naumann	jnaumann@aicpa.org	AICPA
Geoff Shuetrim	gshuetrim@kpmg.com	KPMG

Abstract

This document describes the architecture of financial reporting taxonomies and their corresponding instances using the eXtensible Business Reporting Language [XBRL]. The recommended architecture establishes rules and conventions that assist in comprehension, usage and performance among different financial reporting taxonomies. "Financial reporting" encompasses disclosures derived from authoritative financial reporting standards and/or applicable generally accepted accounting practice/principles, regulatory reports whose subject matter is primarily financial position and performance, and data sets used in the collection of financial statistics; it excludes transaction- or journal-level reporting, reports that primarily consist of narrative (for example, internal controls assessments) and non-financial quantitative reports (for example, air pollution measurements). This document assumes use of the XBRL 2.1 Specification.

Status

This is a public working draft whose circulation is unrestricted. It may change and is not appropriate to reference from public documents.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

[1] Walter Hamscher is a consultant to PricewaterhouseCoopers.

Table of Contents

Table of Tables

Table of Figures

Table of Examples

1. Introduction

XBRL International specifies this architecture[i] to enhance consistency among the XBRL taxonomies used for financial reporting. An important design goal for financial reporting taxonomies is to maximise the usability of the taxonomy to the non-technical (from a computer science perspective) users and experts of the reporting domain, while not compromising the ability of the taxonomy to describe reporting requirements and possibilities in an accurate and XBRL-compliant manner. Where these goals conflict, the architecture is biased in favour of comprehensibility over implementation ease for software designed to support the architecture. The financial reporting taxonomy architecture addresses several areas of consistency:

- **Representation**: Taxonomies should use similar XBRL structures to represent similar relationships among concepts. For example, financial reporting concepts that are measured the same, aggregated the same, and disclosed the same are represented using the same shared XBRL element. Distinctions such as period, entity, or units that are meant to be captured using XBRL contexts are not reflected in the taxonomy itself. The different levels of equivalency allowed within the architecture are a critical aspect of its design.

- **Modularity**: Taxonomies should have a common approach to grouping taxonomy content at a file level. For example, language-specific labels and references are placed in separate linkbase files; jurisdiction-specific references are placed in separate linkbase files; sets of logically related elements that are unlikely to change are placed in the same schema files.

- **Evolution**: Taxonomies built to the architecture set out in this document can be extended or revised using similar approaches.

Consistency among financial reporting taxonomies is important because lack of consistency may lead to additional effort being required to consume, use, compare and extend financial facts reported using these taxonomies.

Taxonomies are meant to be long-lived and broadly used across a business reporting supply-chain. In practice this means they are developed in collaboration among several parties. In recognition of this, the needs of those reviewing and maintaining the financial reporting taxonomies have also influenced this document.

1.1. Scope of the architecture

In this document, "financial reporting" encompasses authoritative financial reporting standards and financial reporting best practices (or GAAP), regulatory reports whose subject matter is primarily financial position and performance, and data sets used in the collection of financial statistics; it excludes transaction- or journal-level reporting, primarily narrative reports (for example, internal controls assessments) and non-financial quantitative reports (for example, air pollution measurements).

This architecture is NOT itself a set of financial reporting standards. For example, FAS and IFRS are financial reporting standards. FRTA—the Financial Reporting Taxonomy Architecture—provides the means by which disclosures made pursuant to those financial reporting standards, GAAP, and so forth can be captured using XBRL. This architecture improves the consistency with which such standards are expressed in the XBRL financial reports that are based on them. The architecture does NOT require that preparers of XBRL instances disclose any more information than they currently do in a non-XBRL environment.

1.2. Relationship to other work

This financial reporting taxonomy architecture assumes XBRL 2.1 [XBRL]. Parts of this document reiterate for expository clarity certain syntactic and semantic restrictions imposed by the XBRL Specification, but this document does not modify the XBRL Specification. In the event of any conflicts between this document and the XBRL 2.1 Specification, the XBRL 2.1 Specification prevails. This document does place additional restrictions above and beyond those prescribed by the XBRL Specification. The purpose of these additional restrictions is to maximize XBRL instance comparability of external financial reports where a large number of extension taxonomies are expected.

The IFRS and USFR taxonomy frameworks [IFRS] [USFR] will provide examples of this architecture once they have achieved Approved status [Processes]. In the event of any conflict between this document and any current version of IFRS and USFR taxonomy frameworks, this document prevails.

1.3. Goals of this document

A financial reporting taxonomy or extension of the USFR or IFRS taxonomy that receives Approved status from XBRL International MUST conform to this architecture. This document is normative with respect to such taxonomies. The architecture MUST be used during XBRL International's review of taxonomies that are candidates for Approved status [Processes]. All parts of this document not explicitly identified as non-normative are normative.

This document should be used by *taxonomy developers*, that is, those who already have some familiarity with XBRL usage, syntax and semantics and who are contributing to or responsible for a financial reporting taxonomy, *either with:*

- financial reporting domain expertise and previous exposure to XBRL technology, *or*
- software expertise and previous exposure to financial reporting concepts.

This document may also be useful to:

1. Anyone creating a financial reporting taxonomy who wishes to follow a broadly used set of practices;
2. Any company wishing to create an extension taxonomy to support their financial statements using XBRL using custom concepts and relationships; and
3. Application developers who support development or use taxonomies that meet the requirements set out in this document.

No part of this architecture requires an English translation. Any rule which depends on a feature present in English but not in another language, may be ignored for taxonomy content in that other language. During the process of reviewing taxonomies for proposed approved status, XBRL International will rely on qualified fluent speakers to review any relevant text in languages other than English.

1.4. Organisation of this document

This document describes the architecture in layers from the bottom up. Overall, the architecture comprises:

1. a *Concept* layer describing rules governing XBRL representation structures such as elements, concepts, and links;

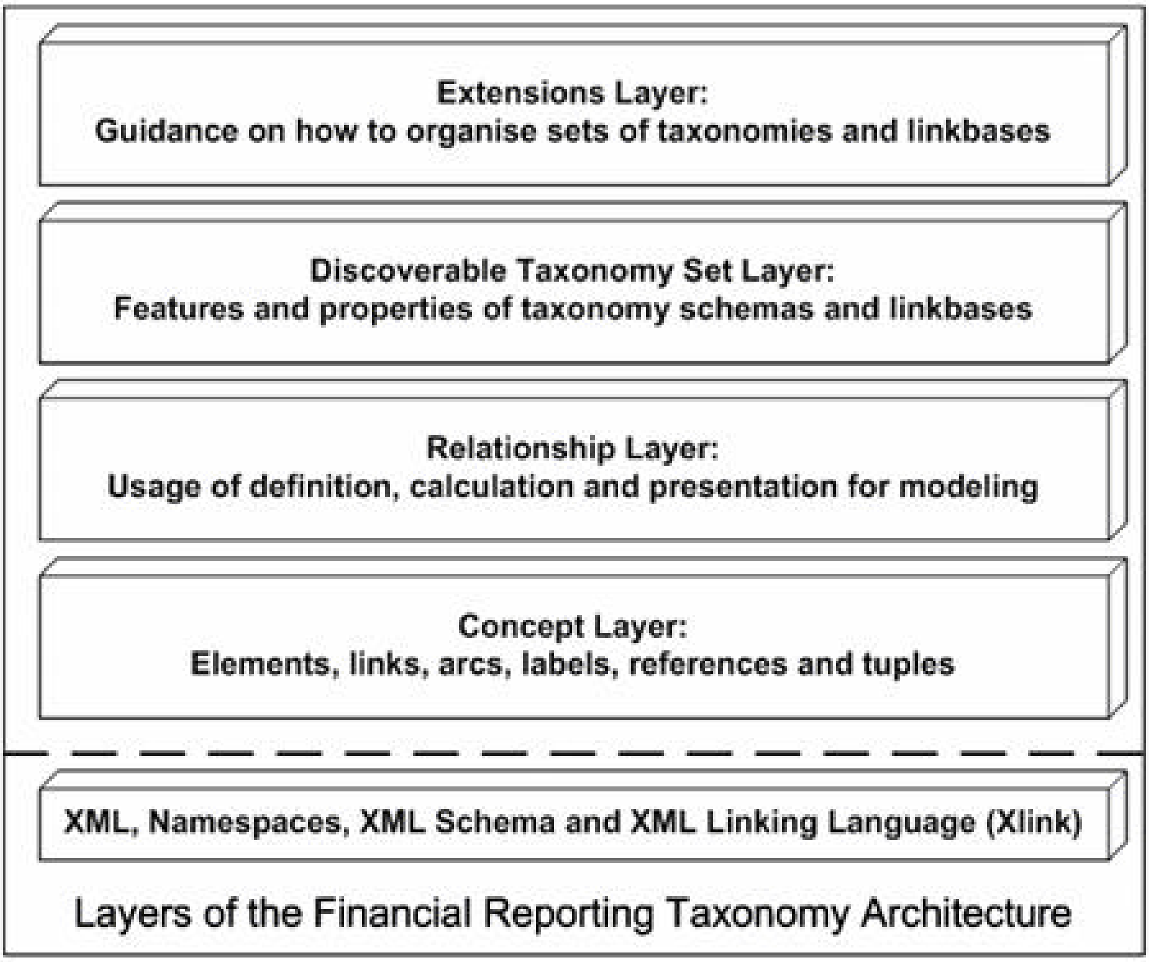
2. a *Relationship* layer describing rules of link usage and how relationships are captured using link types such as definition, calculation and presentation;

3. a *Discoverable Taxonomy Set* layer defining the rules of the organisation of individual files to form discoverable taxonomy sets; and

4. an *Extensions* layer dealing with rules by which taxonomy extensions are to be created and general principles governing modularity.

XBRL is implicitly a part of this architecture although much of what is covered in the XBRL Specification is not repeated in this document. XML Schema and XML Linking Language are also implicitly part of the architecture because they are building blocks for XBRL, however they are not covered explicitly in this document either.

Figure 1. Layers of the FR taxonomy architecture.


Extensions Layer:
Guidance on how to organise sets of taxonomies and linkbases
Discoverable Taxonomy Set Layer:
Features and properties of taxonomy schemas and linkbases
Relationship Layer:
Usage of definition, calculation and presentation for modeling
Concept Layer:
Elements, links, arcs, labels, references and tuples
XML, Namespaces, XML Schema and XML Linking Language (Xlink)
Layers of the Financial Reporting Taxonomy Architecture

Many taxonomy development errors result from a lack of understanding the consequences for XBRL instances; hence there are examples and discussion relating to instances even though that is not the focus of this document.

1.5. Terminology and document conventions

Terminology used in XBRL frequently overlaps with terminology from other fields.

abstract element, ancestor, bind, child, concept, concrete element, context, duplicate items, duplicate tuples, element, entity, essence concept, fact, fully conforming, grandparent, instance, item, least common ancestor, linkbase, minimally conforming, parent, period, sibling, taxonomy, taxonomy schema, tuple, uncle, unit	As defined in the XBRL 2.1 specification.
MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY, OPTIONAL	See [RFC2119] for definitions of these and other terms. These include, in particular: SHOULD Conforming documents and applications are encouraged to behave as described. MUST Conforming documents and consuming applications are required to behave as described; otherwise they are in error.
DTS (Discoverable Taxonomy Set)	A DTS is a set of taxonomy schema and linkbase files. It includes all taxonomy schemas and linkbases that are discovered by following links or references in the taxonomy schemas and linkbases included in the DTS.
base DTS extension DTS	An extension DTS is a DTS that is a proper superset of a base DTS. Because an extension must be a proper superset, a DTS is not an extension of itself.
extended-type link	A set of arcs and other elements that relate a set of concepts to each other. XBRL linkbases are made up of extended-type links.
FRTA	Financial Reporting Taxonomies Architecture: the set of rules described in this document.
GAAP	Generally Accepted Accounting Practice/ Principles: Term used to describe broadly the body of principles that governs the accounting for financial transactions underlying the preparation of a set of financial statements. Generally accepted principles are derived from a variety of sources, including promulgations of Accounting Standards Boards, together with the general body of accounting literature consisting of textbooks, articles, papers, *etc.* [LLLL]

source	The *source* of an arc is the element indicated by the "`from`" attribute.
target	The *target* of an arc is the element indicated by the "`to`" attribute.
Taxonomy status (see [Processes]): Acknowledged Approved Recommended	Acknowledged: XBRL International recognizes that the taxonomy is technically in compliance with all appropriate specifications. Approved: In addition to being Acknowledged, XBRL International warrants that the taxonomy was developed in an open fashion and it complies with all best practices for compatibility. Recommended: In addition to being approved, XBRL International singles out a Recommended taxonomy as being the one preferred for a given type of reporting. Financial reporting taxonomies are not expected to achieve this status from XBRL International since it is not the custodian of the financial reporting standards themselves.
version control	A version control system maintains an organized set of all the versions of files that are made over time. Version control systems allow people to go back to previous revisions of individual files, and to compare any two revisions to view the changes between them.

The following highlighting is used for non-normative examples in this document:

Non-normative editorial comments are denoted as follows:

> WH**:** This highlighting is used to indicate editorial comments about the current draft, prefixed by the editor's initials. All such comments must be removed from candidate recommendations.

Italics are used for rhetorical emphasis only and do *not* convey any special normative meaning.

Figure 2 illustrates drawing conventions followed in figures showing taxonomy fragments and taxonomies.

Figure 2. Legend of drawing conventions for taxonomy fragments

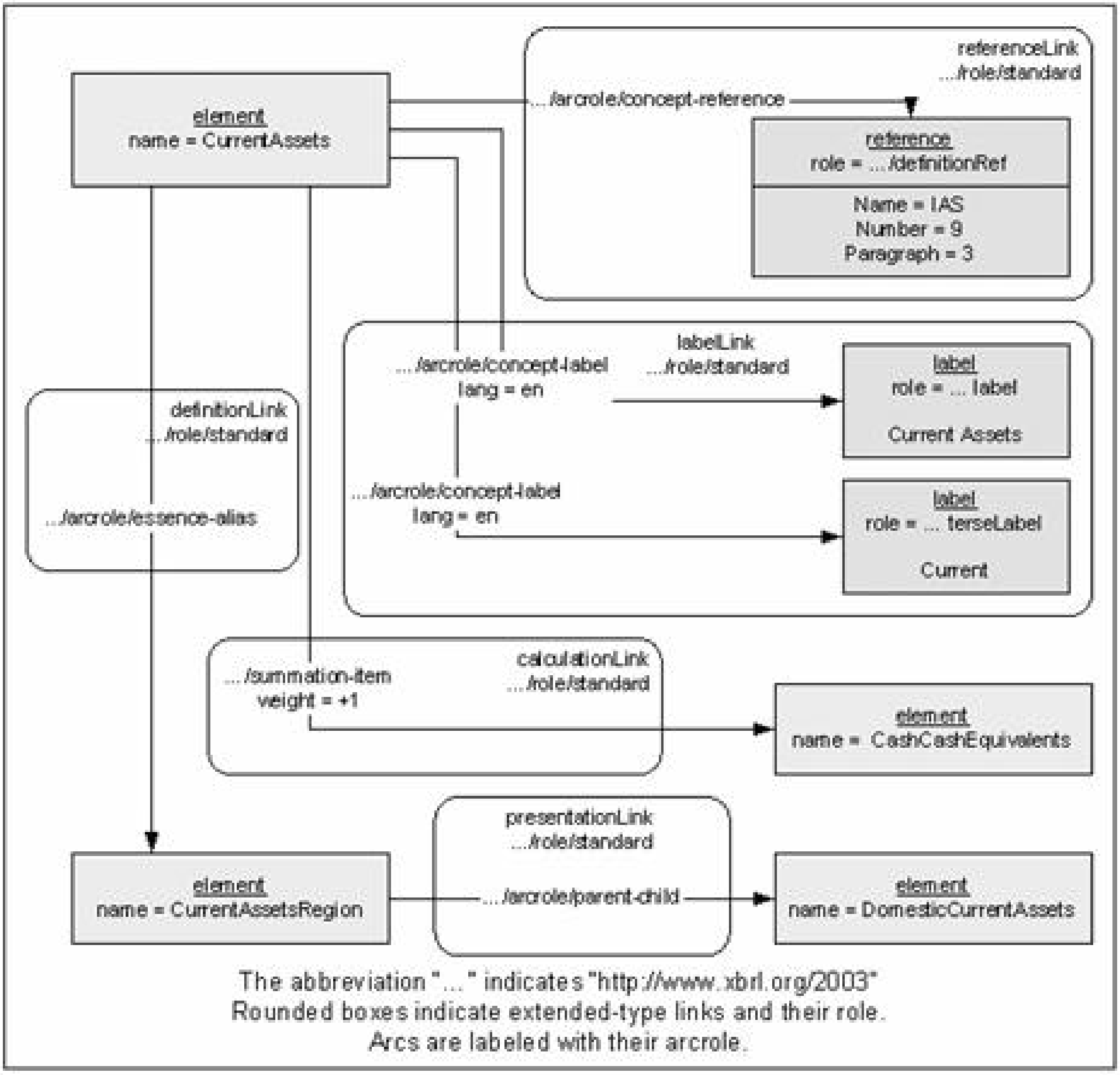
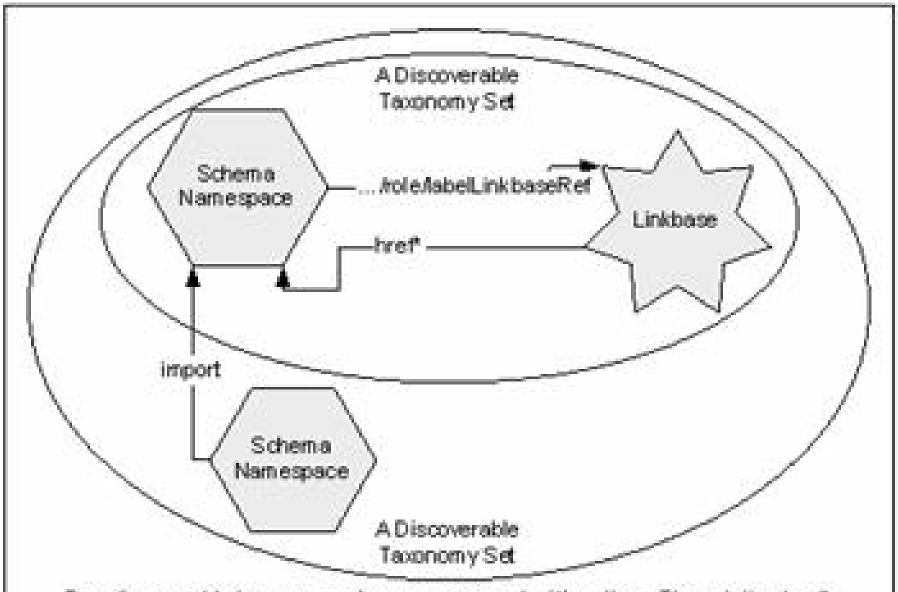


Figure 3. Legend of taxonomy schema and Linkbase drawing conventions.


A Discoverable Taxonomy Set
Schema Namespace
.../role/labelLinkbaseRef
Linkbase
href*
import
Schema Namespace
A Discoverable Taxonomy Set

Two discoverable taxonomy sets, one a superset of the other. The notation href* indicates that the document contains locators with href attributes pointing to the schema elements. Other arcs indicate import, include, and linkbaseRef relations.

The following table summarizes the notation used in the diagrams of this document.

⟶	A "from-to" arc *from* a source element (end of line with no arrow), *to* a target element (end of line with arrow).
▭	A concept element
▢	An extended-type link element
⬡	Taxonomy schema
✶	Linkbase
⬭	Discoverable taxonomy set
summation-item	summation-item arc role
weight = +1	Weight of 1 relative to parent (on summation-item arc)
parent-child	parent-child arc role
essence-alias	essence-alias arc role
documentation	documentation role
terseLabel	Label link, terse role
lang = en	`xml:lang` attribute value "`en`"
…	Abbreviation for `http://www.xbrl.org/2003`
…/role/standard	`xlink:role` attribute value "`http://www.xbrl.org/2003/role/standard`"
…/arcrole/standard	`xlink:arcrole` attribute value "`http://www.xbrl.org/2003/arcrole/standard`"

2. Concept Layer

In a syntactic sense, a concept is an XML Schema element definition, defining the element to be in the `item` element substitution group or in the `tuple` element substitution group. At a semantic level, a concept is a definition of kind of fact that can be reported about the activities or nature of a business activity. Taxonomies contain XBRL concepts represented by XML Schema element definitions. Concepts are meant to represent a type of fact, that is, *data*. The *presentation* of the concept in any given situation is described by other XBRL constructs, and that distinction between data and presentation is fundamental to XBRL.

2.1. Rules for all concepts

The rules covering concepts apply to items and to tuples.

2.1.1. A taxonomy schema MUST define only one concept for each separately defined class of facts.

Having one concept per definition of how a class of facts is to be measured simplifies applications that must extract, compare and combine information from XBRL instances. Two facts fall into the same "class" in this sense if for any context the two values would always be the same in an instance. For example, "Cash Balance in Bank" would, theoretically, have only one element to express this concept, and XBRL instances would use different contexts to report the value for this element for different periods, different entities, etc. Similarly, concepts that have multiple uses within financial reporting (for example, in primary financial statements and in explanatory notes to financial statements) MUST be defined only once.

The uniqueness requirement only applies to sets of concepts defined within a single taxonomy schema and does not extend to discoverable taxonomy sets. Where duplicate concepts are identified, taxonomy authors SHOULD recognise such equivalencies using `essence-alias` relationships in definition extended-type links. For rules governing these relationships see chapter 4, "Discoverable taxonomy set layer" and chapter 5, "Taxonomy Extensions".

The equivalency of two concepts must be assessed at the semantic level, by comparing the set of possible values that are valid to report using the syntax for those concepts. This requires a comparison of the labels, references and inter-concept relationships associated with the two concepts in the linkbases.

Example 1. Identical concepts

Concept	Concept	Explanation
Net Profit	Net Loss	These are not distinct concepts because an entity cannot have both a profit and loss in the same period. Concepts such as `NetProfit` and `NetLoss` are redundant and SHOULD be represented a single element such as `NetProfitLoss`.

Example 2. Distinct concepts

Concept	Related but distinct concept	Explanation
Cash Balance	Change in Cash Balance	The first concept is the amount of cash at a specific instant; the other is the change in the cash balance between one instant and

		another.
Revenue	Percent Change in Revenue	The first concept is the amount of revenue over a period of time, and the other is the percent change in revenue between one period of time to another period of time.
Inventory (measured using the LIFO or FIFO method)	LIFO Inventory	The 2nd concept is measured using the LIFO method only.
	FIFO Inventory	The 2nd concept is measured using the FIFO method only.
	Inventory Measurement Policy	Text describing how the inventory is measured.
Trade Receivables, Net	Trade Receivables, Gross	These concepts are different because they are calculated differently; one nets out "Allowances for Bad Debts" and the other does not.
Deferred Tax Assets	Deferred Tax Liabilities	These concepts are distinct because they are disclosed separately; that is, unlike net income which can only be a profit or loss, an entity may have both deferred tax assets and liabilities that do not offset.

Equivalence of concepts is affected by four factors affecting the set of valid values for a concept: *measurement*, *aggregation*, *materiality*, and *disclosure*. These are discussed below and should be taken into account when determining whether two concepts are duplicates. Naturally, concepts should be examined on a case-by-case basis to determine appropriate modelling in the specific situation.

2.1.1.1. Measurement

Concepts that are measured differently MAY be represented by a single concept if that concept has a broad enough definition provided by its labels and references and by its calculation and definition extended-type link relationships to other concepts.

For example, LIFO and FIFO inventory both value inventory, but are measured differently. An inventory concept that allowed both measurement approaches could validly be defined to contain inventory facts measured using either approach.

In contrast an inventory concept that only allowed measurement using one approach SHOULD NOT be used to contain inventory facts measured using the other approach.

2.1.1.2. Aggregation

Concepts that are aggregated or calculated the same way MAY be equivalent and represented by a single concept.

Concepts MAY also be considered equivalent even if their values are calculated slightly differently, so long as their underlying definitions permit both kinds of calculations. However, in general, the calculation relationships describing how the values for one concept can be derived from the values of others provide a good guide to concept equivalencies: if they are calculated differently they are probably distinct.

Aggregation can also be a good guide to concept identification for non-numeric concepts. For example, notes can be provided as a single block of text or they can be provided as a series of separate facts whose text values can be combined to constitute the combined value of the non-numeric concept with the broader, more aggregated definition.

For example, a concept could be defined to validly contain a comprehensive description of all accounting policies. Alternatively a set of concepts could be defined so that each can only validly contain text about a particular kind of accounting policy. Depending on the granularity of reporting that specific instances are intended to achieve, either the aggregated single concept or the disaggregated set of concepts could appear in an instance.

To allow different levels of granularity in reporting, taxonomies MAY define both the single concept and the set of concepts and MAY represent the associations between the aggregate concept and the disaggregated concepts using presentation extended-type link parent-child relationships.

2.1.1.3. Materiality

Materiality guidelines generally call for disaggregating reported items down to some relative materiality, which differs from entity to entity depending on factors such as management discretion. For example, "Cash" under some standards includes postage stamps and under others do not, but it is unlikely in the general case that the total "Cash" amount disclosed would be materially different; hence these MAY be modelled as the same concept in an XBRL taxonomy so long as the underlying definition of the concept accommodates both approaches to measurement.

2.1.1.4. Disclosure

Reporting standards frequently mandate qualitative disclosures that nevertheless do not warrant separate XBRL items. For example, an "Inventory" monetary figure must be disclosed, but it may be neither necessary nor desirable to have different inventory items to distinguish every possible distinction (for example, perishable vs. durable). Such disclosures can be made in a text description provided with a separate concept.

XBRL does not provide an extended-type link relation between the numeric item and the non-numeric item that provides textual detail. The distinctions that can be captured in the disclosure description (text) concept MUST NOT be part of the concept definitions determining valid values for the concept whose disclosure is being described in additional detail. Returning to the Inventory example above, define either (a) an Inventory item and an Inventory Policy item, or (b) a LIFO Inventory and FIFO Inventory item, but not both (a) and (b).

2.1.2. Contextual and measurement information in XBRL instances MUST NOT result in different elements in a taxonomy.

For example, a concept definition MUST NOT specify that the concept is only to be used for facts about company XYZ or for facts that are true as at the end of a financial year.

XBRL instances contain facts that are instances of of concepts. Facts can contain content values that should meet the semantic requirements associated with the concepts that they are instances of. Besides the value of a fact, such as "the value of cash is 500,000", the XBRL instance provides contextual information necessary to correctly interpret each fact. This context includes:

- the entity that the value of the fact describes;
- a period for which or over which the fact is true; and
- the scenario under which the value of the fact has been measured.

Because only facts have a period associated with them, there is no such thing as "the period over which a concept applies." Hence (for example) "cash," "cash at the beginning of a period," and "cash at the end of a period" are not distinct concepts. There is only one concept in this case: cash, and it is measured at an instant.

For numeric facts, XBRL instances also provide information relating to measurement accuracy and measurement units.

2.1.3. Concepts' meanings MUST NOT depend on their position within an instance.

A single item or tuple can appear within many different tuples because all items and tuples are defined globally. For example, the item `Residuals` may appear within different tuples only if it has the same meaning in both places. Therefore, if one tuple relates to payments received for each rerun after an initial showing of a TV show, while another tuple relates to the value of oil not yet extracted from beneath leased property, two different items (for example, `TelevisionResiduals` and `OilResiduals`) should be defined.

2.1.4. Abstract concepts MUST be defined to be in the item substitution group and MUST NOT have a Schema type attribute.

Abstract concepts are concept definitions with the XML Schema `abstract` attribute equal to `true`. Abstract concepts cannot be used in XBRL instances. Instead, their role is limited to organisation (grouping) of other concepts defined in taxonomies. Abstract concepts MUST conform to the element naming and id attribute value requirements set out in this document, and MAY have labels and references.

2.1.5. Concept names SHOULD adhere to the LC3 convention.

LC3 means Label CamelCase Concatenation (LC3). LC3 rules require that:

1. Element names MUST be based on an appropriate presentation label for the element. A label SHOULD be a natural language expression that is meaningful to experts in the domain covered by a taxonomy (for example, "Revaluo Propio", "Restatement of Fixed Assets"), for a given item.

2. If multiple labels exist for a concept, then any one of those labels MAY be used as the basis for construction of the element name. Furthermore, if the element name is originally based on a label and in a subsequent version of the taxonomy the label changes, the element name MUST NOT be changed merely to maintain agreement.

3. The first character of the element name MUST be alphabetic.

4. The first character of the element name MUST be capitalised.

5. Connective words in the label MAY be omitted from the element name to make names shorter. Examples of English connective words include (but are not limited to) the following:

   - the, and, for, which, of, a

6. All special characters MUST be omitted from the element name. Special characters include the following (and are limited to these, for English labels):

( ) * + [ ] ? \ / ^ { } | @ # % ^ = ~ ` “ ‘ ; : , < > & $ £ €

7. Element names MUST be limited to 256 characters or fewer.

8. Words in a label from which an element name is derived MAY be abbreviated when used in the element name. A list of standard abbreviations and rules for substitution (for example, “Property Plant and Equipment” in a label is always replaced by “PPE” in the element name) SHOULD be maintained by the taxonomy author(s). When standard abbreviations are used, they SHOULD be applied consistently throughout the taxonomy.

9. If two or more elements share the same element name and the element name is less that 256 characters long, then uniqueness may be accomplished by one of the following means:

   - appending a distinguishing suffix;
   - adding a distinguishing prefix;
   - appending the first duplicate name with a number suffix, beginning with 1 and incrementing by 1 for each element with a common name.

   The distinguishing suffix or prefix MAY be derived from the label of one or more ancestor elements. If two or more elements share the same name and the element prefix takes the name length beyond 256 characters, sufficient characters from the end of the element name MUST be dropped and rule number 9 MUST be applied.

The following is an example of element names based on the naming conventions described above. The table shows a concept label and the corresponding element name, based on the LC3 naming conventions.

Example 3. Sample LC3 element names.

English Label of Concept	Element Name
Assets	Assets
Cash & Marketable Securities	CashMarketableSecurities
Notes to Financial Statements	NotesFinancialStatements
Statement of Compliance	StatementCompliance
1st Time Application of US-GAAP	FirstTimeApplicationUSGAAP
First-Time application of US-GAAP	FirstTimeApplicationUSGAAP
Impact on Net Profit (Loss) for Each Period Presented for Change in Classification in Significant Foreign Operation	ImpactNetProfitLossEachPeriodPresentedChangeClassificationSignificantForeignOperation
Arm's length disposals of Excess of nominated proceeds from PRT1(Part2) Sterling Value £	ArmsLengthDisposalsExcessNominatedProceedsPRT1Part2SterlingValueUKPound

2.1.6. Element definitions for concepts MUST contain an “id” attribute whose value is the concatenation of the recommended namespace prefix of the taxonomy and the “name” attribute of the element.

The recommended namespace prefix is supplied in the documentation supporting a taxonomy (see 4.3.2).

Example 4. Sample id attribute

English Label of Concept	Element Name	Recommended Namespace Prefix	id attribute
Cash in Bank	CashInBank	us-gaap-ci	us-gaap-ci_CashInBank

This convention helps to avoid problems with certain XML Schema processors when importing one schema into another. The resulting id MAY be longer than the 256 characters prescribed for the element name.

2.1.7. The default value of the XML Schema "nillable" attribute is true for items.

XBRL instances can include items with nil values to indicate that the value of the item is not known. An example of where this is useful is in instances that are produced as the result of database queries that return incomplete results. The only use of nillable="false" (which is the XML Schema default) on items SHOULD be cases where the financial reporting standard itself mandates that a particular value cannot be left unspecified. The value nillable="true" has no effect on items for which abstract="true".

2.1.8. An "element" element MAY include any of the other XML Schema attributes that can be used on a global element syntax definition.

Specification section 3.11 reads "The element MAY also include any of the other XML Schema attributes that can be used on an element's syntax definition, including abstract and nillable."

2.1.9. All documentation of a concept that constrains the set of valid values for that concept MUST be contained in XBRL linkbases.

Taxonomies MAY but SHOULD NOT duplicate some or all of their documentation using the XML Schema documentation element.

2.1.10. A concept MUST have a label with the standard label role.

The standard label role is http://www.xbrl.org/2003/role/label.

Understanding the precise meaning of concepts within a financial reporting taxonomy is critical. The meaning of a concept is provided by a combination of documentation provided in the form of text in the label linkbase (using the "documentation" role) and/or references to other documentation provided external to the actual taxonomy, such as a paper volume of accounting standards.

This label must be in an extended-type link that is discoverable from the taxonomy schema in which the concept is defined.

2.1.11. All concepts within a taxonomy schema SHOULD have a unique label.

Uniqueness within the scope of an entire DTS cannot be guaranteed by any single taxonomy author. Also, the standard label for a concept need not be unique. However, at least one label role (such as http://www.xbrl.org/2003/role/verboseLabel) should provide a distinct label for each concept.

2.1.12. Each concept MUST have documentation in either the label or reference linkbase.

The documentation MUST be provided in at least one of these three ways:

1. documentation label with the role `http://www.xbrl.org/2003/role/documentation`;

2. definition label with the role `http://www.xbrl.org/2003/role/definitionGuidance`; or

3. reference with the reference role `http://www.xbrl.org/2003/role/reference`.

A concept MAY have many different labels, each distinguished by the role assigned to that label and by the language that the label is expressed in. A concept may also have many different references to other literature that sheds light on the meaning of that concept. These references are distinguished using reference roles.

This documentation must be in label or reference extended-type links that are directly discoverable from the taxonomy schemas in which the concepts are defined.

2.1.13. Labels SHOULD have a correspondence to the meaning of the element.

Human users are likely to be presented with the label, rather than the element name. This guidance is a consequence of 2.1.5, "Concept names SHOULD adhere to the LC3 convention."

2.1.14. There MUST NOT be internal structure in label text that requires software to draw inferences about the meaning of the label.

This is the dual of rule 2.1.13; label text should have meaning *only* to human users.

2.1.15. Words MUST be spelled consistently throughout the labels in a linkbase.

For example, "pro forma" should be used consistently rather than sometimes using "proforma" and sometimes "pro forma." This rule should be interpreted as referring to root words only, for inflected languages such as German.

2.1.16. Labels SHOULD have a consistent style of phrasing.

For example, "treasury shares, ending balance", "treasury shares, changes", and "treasury shares, beginning balance" are consistent phrasings. Inconsistent phrasings would be "final treasury shares," "treasury shares, changes" and "beginning of period treasury shares".

2.1.17. Non-alphabetic characters, if used, should be used consistently in labels.

For example, if a comma is used to separate parts of a label, as in "treasury shares, ending balance", then commas should be used in other labels in the taxonomy for the same purpose -- not mixed with dashes and brackets.

The following are example labels for each of the label roles:

Example 5. Labels

Role	Label for item NetResultForeignCurrencyTranslations (period type = duration)
standard label	foreign currency translations, net
terse label	F/X net
verbose label	net result of foreign currency translations
positive label	foreign currency translations gain
positive terse label	F/X gain
positive verbose label	net gain on foreign currency translations
negative label	foreign currency translations loss
negative terse label	F/X loss
negative verbose label	net loss on foreign currency translations
zero label	
zero terse label	
zero verbose label	
total label	total, net foreign currency translations
	Label for item FinishedGoodsInventory (period type = instant)
period start label	finished goods inventory, beginning of period
period end label	finished goods inventory, end of period

Labelling guidelines for languages other than English are the responsibility of individual XBRL jurisdictions and, when they exist, MUST be followed in any labelling linkbase in the relevant language.

2.1.18. All components of references to authoritative literature documenting concepts MUST be contained in appropriately defined reference parts.

References documenting a concept may consist of a hyperlink to web-based reference material or to specific pages or paragraphs in authoritative literature, or both.

2.1.19. Reference parts SHOULD include the name of the standard or other enactment, and sections, clauses or paragraphs as appropriate.

The reference parts point to other materials. Note that specification section 5.2.3.2 says that reference parts "MUST NOT contain the content of those reference materials themselves."

2.1.20. References MUST use elements in the substitution group of the XBRL linkbase "part" element from the namespace http://www.xbrl.org/2003/ref.

Reference parts have been defined by XBRL International with default namespace prefix "ref", where those elements are appropriate for the documentation that they are referencing. For example, page numbers may be irrelevant for some references.

```
ref:Name
ref:Number
ref:Paragraph
ref:Subparagraph
ref:Clause
ref:Subclause
ref:Pages
```

Any subset or combination of these parts MAY be used in any given reference. Only one of each reference part MAY be used in any given reference, and they may appear as sub-elements in any order, since their contents will usually be displayed in an application-specific fashion. Reference elements MAY use additional reference part elements they deem appropriate for their documentation.

Example 6. Reference contents.

IAS 1 75 (e)	
`ref:Name`	IAS
`ref:Number`	1
`ref:Paragraph`	75
`ref:Subparagraph`	e

Section 12(2)(a) Securities Act 1983	
`ref:Name`	Securities Act
`ref:Year`	1983
`ref:Section`	12
`Ref:Subsection`	2
`ref:Paragraph`	a

Reference elements MUST contain an `xlink:role` attribute that MUST distinguish between reference elements by the nature of the XBRL concept documentation that they make external reference to. Example 7 below provides an example of some standard `xlink:role` attribute values and their meanings for reference resources. In the Balance sheet of the IFRS taxonomy, there is an element whose label is "Onerous Contracts Provision, Non Current". The example has multiple references linked to the element `OnerousContractsProvisionNonCurrent`.

Example 7. Reference role usage.

`xlink:role` attribute value for `reference` (Definition from XBRL 2.1 Section 5.2.2.2)	**Reference example** (actual text)	
`http://www.xbrl.org/2003/role/definitionRef`	Name: Number: Paragraph:	IAS 37 10
Reference to documentation that details a precise definition of the concept.	*An <u>onerous contract</u> is a contract in which the unavoidable costs of meeting the obligations under the contract exceed the economic benefits expected to be received under it.*	
`http://www.xbrl.org/2003/role/presentationRef`	Name: Number: Paragraph:	IAS 1 73 d
Reference to documentation which details an explanation of the presentation, placement or labelling of this concept in the context of other concepts in one or more specific types of business reports	*… provisions are analysed showing separately provisions for employee benefit costs and any other items classified in a manner appropriate to the enterprise's operations*	
`http://www.xbrl.org/2003/role/measurementRef`	Name: Number: Paragraph:	IAS 37 66
Reference concerning the method(s) required to be used when measuring values associated with this concept in business reports	*If an enterprise has a contract that is onerous, the present obligation under the contract should be recognised and measured as a provision.*	

`xlink:role` **attribute value for** `reference` (Definition from XBRL 2.1 Section 5.2.2.2)	**Reference example** (actual text)
`http://www.xbrl.org/2003/role/commentaryRef`	Name: IAS Number: 37 Paragraph: 67
Any other general commentary on the concept that assists in determining appropriate usage	*Many contracts (for example, some routine purchase orders) can be cancelled without paying compensation to the other party, and therefore there is no obligation. Other contracts establish both rights and obligations for each of the contracting parties. Where events make such a contract onerous, the contract falls within the scope of this Standard and a liability exists which is recognised. Executory contracts that are not onerous fall outside the scope of this Standard.*
	Name: IAS Number: 37 Paragraph: 68
	This Standard defines an onerous contract as a contract in which the unavoidable costs of meeting the obligations under the contract exceed the economic benefits expected to be received under it. The unavoidable costs under a contract reflect the least net cost of exiting from the contract, which is the lower of the cost of fulfilling it and any compensation or penalties arising from failure to fulfil it.
	Name: IAS Number: 37 Paragraph: 69
	Before a separate provision for an onerous contract is established, an enterprise recognises any impairment loss that has occurred on assets dedicated to that contract (see IAS 36 Impairment of Assets).
`http://www.xbrl.org/2003/role/exampleRef`	Name: IAS Number: 37 Paragraph: Appendix C Example 8
Reference to documentation that illustrates by example the application of the concept that assists in determining appropriate usage.	*An enterprise operates profitably from a factory that it has leased under an operating lease. During December 2000 the enterprise relocates its operations to a new factory. The lease on the old factory continues for the next four years, it cannot be cancelled and the factory cannot be re-let to another user … Conclusion - A provision is recognised for the best estimate of the unavoidable lease payments.*

2.2. Implications of the concept rules on instances

2.2.1. When different occurrences of a concept in an instance are distinguished by measurement or aggregation, labels MUST NOT be used to encode these distinctions.

For example, labels must not be used to distinguish valuation at *cost* from valuation at *market value* while using the same concept to report these two valuations in the same instance. Furthermore, if the definition of the concept is broad enough to accommodate both measurement approaches, then the labels associated with that concept MUST NOT indicate that a particular measurement approach has been chosen.

2.3. Rules for items

This section documents syntax rules for concepts in the item substitution group.

2.3.1. The XML Schema type attribute SHOULD be used to enable XML Schema testing of constraints on valid concept values.

XML Schema offers a number of ways to provide constraining facets, all of which restrict the values allowed for elements. For example, enumerated lists, the minimum or maximum length of the string representation of a fact value, a certain pattern for a value, may all be used. These restrictions are documented in XML Schema Part 2: Datatypes [SCHEMA-2].

Taxonomies SHOULD use these XML Schema restrictions as far as possible to enable XML Schema checking of compliance with the constraints on valid values for concepts, insofar as the constraints hold universally. Constraints such as "revenues can have no more than 12 decimal digits" are too application-specific.

For example, item types whose content is restricted to enumerations are encouraged in financial reporting taxonomies when there are a finite number of valid values for an instance of a concept. For example, if "`FixedRate`" or "`VariableRate`" are the ONLY options, and exactly one value is required, an enumeration with the values of "`FixedRate`" and "`VariableRate`" as a restriction of `token` should be used as the data type of which the concept's item type is an extension.

2.3.2. Different values for an item MUST NOT result in different elements.

Concepts MUST NOT constrain the set of valid values for their instances on the basis of any of these limitations:

- the period over which a fact is measured;
- the entities or entity segments that the fact describes;
- the scenarios under which the fact is applicable; or
- the allowed units of measurement (for example, "in US Dollars") unless specific units are literally and specifically required by the reporting standards underpinning the taxonomy.

Example 8. Concepts and facts

Concept	Fact	Explanation
Intangible Assets	Intangible Assets as of December 31, 2003	The one concept is used to represent facts in instances each with a different context. This context is for a particular point in time.
	Intangible Assets as of December 31, 2004	This context is for a different point in time as the previous fact.
	Intangible Assets as of December 31, 2003 for the East Asian Division	This context is for a different entity.
	Budgeted Intangible Assets as of December 31, 2003	This is a different measurement context.

2.3.3. Monetary concepts corresponding to accounting trial balances (asset, liability, equity, revenue, expenses) MUST use the balance attribute.

The `balance` attribute must have the value `credit` or `debit`. Section 5.1.1.2 of the XBRL 2.1 Specification is explicit that the `balance` attribute MUST NOT be used on items that do not have type equal to the `monetaryItemType` or to a type that is derived from `monetaryItemType`.

2.3.4. Each item that is not abstract MUST only be asserted over either a period or at an instant in time.

The context of a fact includes the instant (`periodType="instant"`) or the period of time (`periodType="duration"`) over which that fact is asserted to be true.

2.3.5. Variations on the same concept that can be measured either over a period or at an instant in time MUST be represented by separate concepts.

Cash and cash equivalents	periodType="instant"
Change in cash and cash equivalents	periodType="duration"
Number of Shares at the End of the Period	periodType="instant"
Number of Shares Average of the Period	periodType="duration"

2.3.6. Abstract elements and Tuples MUST NOT have the periodType attribute.

This is enforced by Section 5.1.1.1 of the XBRL 2.1 Specification. Abstract concepts cannot appear in instances as facts.

2.3.7. Sibling concepts in a tuple MAY have different values of the periodType attribute.

Tuples may reasonably associate elements that mix different period types.

Director Information (tuple)
• Director Name (item)
• Compensation (item)
• Shares Held (item)
	periodType="duration"
	periodType="duration"
	periodType="instant"

2.3.8. Numeric concepts representing a balance or to be captured at a specific point in time MUST have a periodType of instant.

Taken as a whole, financial statements are traditionally stated either historically (for example, for the period ended 31 December 2002) or prospectively (for example, for the period ending 31 December 2010). However, balances in the balance sheet, notes and other components of financial statements are stated "as at" or "as of" a specified date (for example, as at 31 December 2002).

Current assets	periodType="instant"
Bank overdraft	periodType="instant"

The XBRL specification enforces the distinction between `periodType="duration"` and `periodType="instant"` at the level of the taxonomy so as to provide additional syntactic constraints on instances that are useful to application software that must consume instances efficiently. Also, applications that must consume and interpret instances using taxonomies that they have never before encountered can still process, present and interpret the taxonomy if more basic properties such as this are known.

2.3.9. The beginning balance, the ending balance, and any adjusted balances of an item for a period MUST be represented as a single item.

Financial reports often include a reconciliation where a beginning balance is shown (an instantaneous value), changes to that balance are shown (a value for the period which is a duration) reconciled to an ending balance (instant, but in a different period than the beginning balance). This is commonly called a "movement analysis". Sometimes there is an "originally stated" beginning balance and adjustments to that beginning balance and possibly an adjusted balance. Distinctions between the beginning and ending balances of a given item MUST be identified in instances using the `period` element; distinctions between originally stated and adjusted values MUST be identified in instances using the `scenario` element.

2.3.10. Numeric concepts not measurable at a point in time MUST have a periodType of "duration".

All other numeric concepts, including those representing movements in balances over a period, revenue and expense items, etc., will have a `periodType` of `"duration"`.This holds regardless of whether the numeric concept appears in primary financial statements, notes or elsewhere.

Net Income	`periodType="duration"`
Change in provision for doubtful debts	`periodType="duration"`
Movements in asset revaluation reserve	`periodType="duration"`
Earnings per share	`periodType="duration"`

Determining whether a concept is measurable at a point in time may require examining its components. For example, EPS is often stated "as at" or "as of" a particular date, usually balance date. However, a closer look at the components of the EPS equation suggest otherwise. The numerator (Earnings) is clearly a duration; the denominator (Number of shares) may either be an instant (for example, shares at balance date) – or, more commonly, as a duration (for example, weighted average number of shares across the period). Therefore the EPS calculation is more likely than not to have been constructed from the division of two durations and should be represented as a duration by default.

2.3.11. Non-numeric concepts that are stated as at a specified date, but apply to an entire period, MUST have a periodType of "duration".

While there is consensus over which of the instant or duration alternatives apply to components of the primary financial statements, the components and concepts that make up the notes and accounting policies often are not addressed directly by accounting literature. Because the XBRL 2.1 Specification requires that all concepts have their `periodType` specified, general principles and rules assist consistency in taxonomy building and comparability of instance documents.

With regard to financial statement concepts, facts that are true over an entire period are clearly durations and those that are true at a specific date are clearly instants. The problem is that most note disclosures and accounting policies are true over the entire period, despite being stated as at the end of a period. Based on the logic that the duration includes the instant, it has been decided that note disclosures and accounting policies should be captured as periodType="duration".

An example of a numeric item is the depreciation expense of buildings. An example of a non-numeric item is a paragraph of text that explains an accounting policy. Accounting policies are stated as at a specified date, but apply to an entire period.

Accounting policy for inventory	periodType="duration"
Measurement basis	periodType="duration"
Changes in accounting policies	periodType="duration"

2.3.12. Non-numeric concepts that are only true "as of" or "as at" a specific date, MUST have a periodType of "instant".

This holds regardless of whether the numeric concept appears in primary financial statements, notes or elsewhere.

Subsequent events	periodType="instant"
Assets held for sale disclosures	periodType="instant"

Although by definition "subsequent events" occur after the balance date and before the financial statements are finalised, they form part of the financial statements of the period. Therefore they should be stated "as of" or "as at" the balance date. This is supported by the fact that subsequent events affect conditions at the stated balance date and reflect on measurements of balance sheet items.

2.3.13. All other non-numeric concepts, such as accounting policies and disclosures, MUST have periodType of "duration", whether or not they relate to balances or to a period.

Because financial statements are generally stated for a period, it follows that an assignment of periodType="duration" will usually be correct.

CertificationDisclosureControlsProcedures	periodType="duration"
OmittedFactsIndependentAuditNotCompleted	periodType="duration"

2.3.14. Where it is unclear what the period type is that should be assigned to a concept, the default assignment MUST be periodType of "duration".

This is a consequence of rules 2.3.10 and 2.3.13.

2.4. Implications of item rules on instances

2.4.1. Any context without a scenario element MUST be treated as if it is an assertion of an actual, verifiable fact.

Implicit in the overall representation of facts in XBRL is that taken together they constitute a business report for a given period; therefore, any context element that omits further detail in its scenario sub-element MUST refer to an actual measurement.

2.4.2. Facts relating to events or concepts MUST NOT be assigned to any date outside the period unless necessary to reflect accurately the occurrence of the concept.

For facts that are true over a period of time that extends beyond the period that a financial report is documenting, Rule 2.3.12 applies and the instant specified in the contexts for those facts MUST fall within the period documented by the report.

Some elements of financial statements are themselves dates. The *content* of the date element (for example, the date of the Accountant's report) is different from the *context* date to which it refers (in this case, the context is the period covered by the accountant's report).

a) Subsequent Events should be reported as at the balance date to which they are subsequent, with `periodType="instant"`.

b) (Exception) Directors' names and signatures to the accounts should be reported as at the date of signing financial statements, subsequent to the balance date, when known.

There may be other exceptions.

Note that accounting policies relate to the period of the stated financial statements and to any prior periods caught by requisite comparatives.

Restated figures should be `periodType="instant"`. By default, they should be assigned the date at the start of the period captured in the financial statements.

2.4.3. Facts relating to a financial statement for a period MUST NOT have any context that is any longer than the period being reported.

A financial statement covering a year may have facts in it whose duration is longer than that particular year. Nevertheless, the facts should be "broken up" into separate contexts each one covering no more than one year, so as to simplify the task of applications that extract information.

2.5. Rules for tuples

Tuples are used to bind together, or associate, one or more items. Together, these concepts form a compound or complex fact. Examples include lists and tables in financial statements. Sets of tuples are also the only mechanism in XBRL that allows repeated occurrences of a concept to appear in an instance document in the same context (for instance, a list of subsidiary companies as of a point in time).

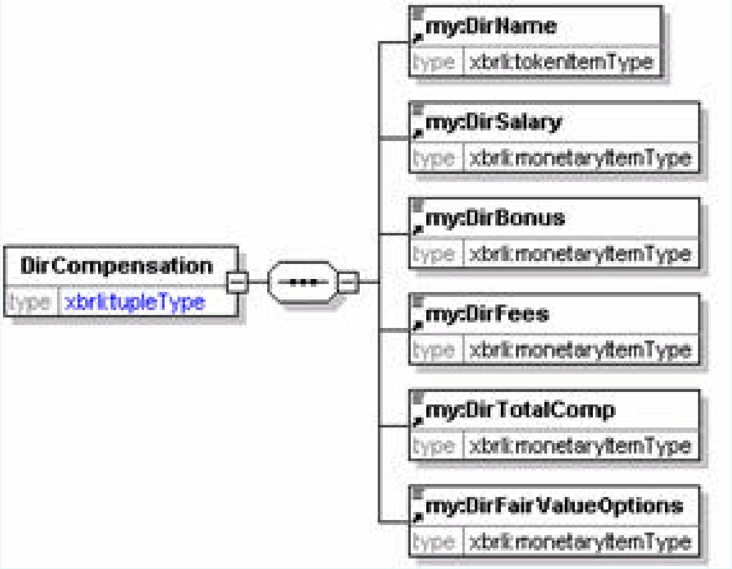
2.5.1. Tuples MUST be used to associate facts that derive their meaning from each other.

Tuples need to be used wherever it is necessary to convey a number of concepts that cannot be understood without being grouped together. For example, it would be common to list directors' names, salaries and options. To be understood, the entries need to be grouped together. Compare: there was a director named "Jane Smith," there was a director that earned "$10,000" and there was a director granted "$50,000" in options, *versus* the fact that "Jane Smith" earned "$10,000" and was granted "$50,000" in options. If an XBRL instance is only composed of element name and value pairs inside atomic items, it is impossible to determine these fact groupings. Tuples associate the name and title pairs by nesting those items within the tuple of director's remuneration in an instance.

Example 9 shows a table of compensation for directors of a company. For each director, the name of the director, salary, bonus, director's fees paid, total compensation paid, and fair value of stock options granted are presented. This is a two dimensional table with (in this presentation) the groups of related facts displayed in rows, and the taxonomy concepts contained in columns. This information can be presented for any number of directors. While there is variation at the level of each group (row) of fact values, the concepts are set by the taxonomy. The schema diagram shows how this would be encoded using XBRL. The element `DirCompensation` is a tuple that contains six items. Each column of the table corresponds to one of the items.

Example 9. A table in a financial statement modelled using a tuple.

Name of director	Salary	Bonus	Director fees	Fair Value of Options Granted
Ho Ching	0	0	60,000	0
Boon Swan Foo	879,639	1,213,486	0	569,0000
Tan Guong Ching	0	0	24,200	0
Ng Kee Choe	0	0	57,000	0



```
<element name="DirName" type="xbrli:tokenItemType"
  substitutionGroup="xbrli:item" id="s_DirName" xbrli:periodType="period"/>
<element name="DirSalary" type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" id="s_DirSalary" xbrli:periodType="period"/>
<element name="DirBonus" type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" id="s_DirBonus" xbrli:periodType="period"/>
<element name="DirFees" type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" id="s_DirFees" xbrli:periodType="period"/>
<element name="DirTotalComp" type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" id="s_DirTotalComp" xbrli:periodType="period"/>
<element name="DirFairValueOptions" type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" id="s_DirFairValueOptions"
  xbrli:periodType="instant"/>
<element name="DirCompensation" substitutionGroup="xbrli:tuple"
id="s_DirCompensation">
  <complexType>
    <complexContent>
      <restriction base="anyType">
        <sequence>
          <element ref="my:DirName"/>
          <element ref="my:DirSalary"/>
          <element ref="my:DirBonus"/>
          <element ref="my:DirFees"/>
          <element ref="my:DirTotalComp"/>
          <element ref="my:DirFairValueOptions"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
</element>
```

In an XBRL instance, each row in the table will be a separate occurrence of the tuple.

2.5.2. When instances may contain multiple values of the same element within the same context, a tuple MUST be used.

For example, a single entity, during a single period, may have any number of subsidiaries. Therefore an item such as `SubsidiaryName` must appear within a tuple.

2.5.3. Numbered sequences of items to accommodate multiple values of the same item MUST NOT be used.

Items should not be created such as "Address1, City1, State1" and "Address2, City2, State2" simply to allow for two distinct addresses.

Accommodating three lines of street address with "Street1, Street2, Street3" does *not* violate this rule.

2.5.4. Tuples SHOULD NOT be used to represent segments.

A "segment" means a line of business, geographical region, or other partitioning of an entity (see XBRL Specification example 15 [XBRL]). Segments should be represented as one or more `segment` sub-elements of `context` elements. Using tuples to model segments can make it more difficult to compare data in different instances, because it allows instance creators too much flexibility to invent new and different segments from those used by other instances.

This rule is not expressed as an absolute prohibition because there may be situations in which the nature of the reporting standards in fact indicates that tuples are appropriate.

2.5.5. Tuples SHOULD NOT be used to represent units, entities, periods or scenarios.

Data that has multiple values within an instance depending on units, entities, periods or scenarios do not require tuples to model. This is a more general case than that specific to segments, but the rationale is the same. If the same item has different values when it appears in different contexts, then it is not necessary to use a tuple. Using tuples to embed these different dimensions of variation into a tuple can make it more difficult to compare data in different instances, because it allows instance creators too much flexibility to invent new and different segments from those used by other instances.

2.5.6. Tuple content models MUST enforce the constraints on their contents that are expressed in their labels and references.

For example, if a tuple is documented (in its label or reference linkbases) as the remuneration of a director, then its content model (in the schema) cannot contain more than one director name and one remuneration value.

2.5.7. Items in tuples MUST have `maxOccurs` equal to 1.

For example, if a tuple reports the remuneration of a director, then it cannot contain the names of two directors and their remunerations. Nor can it contain the name of one director and several remuneration amounts, even when those remuneration amounts have different contexts. Without this constraint, there is no mechanism for avoiding duplication, without additional application-specific validation. The content model MUST therefore enforce this by requiring only one occurrence of the remuneration item and one occurrence of the concept or concepts capturing the director’s name in any one instance of this tuple.

Note that this rule is in contrast to tuple elements appearing inside other tuples; these SHOULD have `maxOccurs` equal to `unbounded`.

2.6. Impact of tuple rules on instances

The first row of the compensation table shown above in Example 9 appears in an XBRL instance as shown in Example 10. Each row of facts is grouped together by the nested tuple element, in this case `my:DirCompensation`, with each item contained, according to the sequence requirements set out in the content model of Example 9, within the opening and closing tags of the tuple.

Example 10. XBRL Instance data containing the first row of a table.

```
<unit id="hkd"><measure>iso:HKD</measure></unit>
<context id="c3">
  <entity>
    <identifier scheme="http://www.hkex.com">SADV</identifier>
  </entity>
  <period>
    <startDate>2001-01-01</startDate>
    <endDate>2001-12-31</endDate>
  </period>
</context>
<ex:DirCompensation>
  <ex:DirName contextRef="c3">Ho Ching</ex:DirName>
  <ex:DirSalary contextRef="c3" unitRef="hkd" decimals="0">0</ex:DirSalary>
  <ex:DirBonus contextRef="c3" unitRef="hkd" decimals="0">0</ex:DirBonus>
  <ex:DirFees contextRef="c3" unitRef="hkd" decimals="0">60000</ex:DirFees>
  <ex:DirTotalComp contextRef="c3" unitRef="hkd" decimals="0">60000</ex:DirTotalComp>
  <ex:DirFairValueOptions
    contextRef="c3" unitRef="hkd" decimals="0">0</ex:DirFairValueOptions>

</ex:DirCompensation>
```

In general, if one visualises the instance data as a multidimensional table, each “cell” in the table will appear as a separate item in the XBRL instance.

3. Relationships Layer

The relationship layer of the architecture describes how concepts (both items and tuples) MUST be related to one another through presentation, calculation and definition relationships. Presentation, calculation and definition relationships are all captured in extended-type links. The extended-type links together make up linkbases. The relationship layer also describes how these relationships SHOULD be modelled.

Where noted, the rules for this layer of the architecture also apply to the extended-type links and arcs that make up label and reference linkbases.

3.1. Rules for all relationships

Section 5.2 of the XBRL 2.1 Specification [XBRL] describes how relationships are modelled by arcs (arc-type elements) that appear within extended-type links. Every arc has an arc role. Every extended-type link has a role, and MAY contain one or more arcs.

3.1.1. Each extended-type link MUST have a nonempty `role` attribute.

XBRL processors treat extended-type links separately when they have different values for the `role` attribute. Section 3.5.5.3 of the XBRL 2.1 Specification [XBRL] indicates that the `role` attribute must not be empty and that the standard value for the `role` attribute is `http://www.xbrl.org/2003/role/standard`.

3.1.2. All arcs within an extended-type link MUST have the same arc role.

The XML Linking Language [XLINK] forbids duplicate arcs within a given extended-type link, even when the arcs in question have different arc roles. Conforming XBRL processors detect violations of this syntax constraint. Accidental violations can be minimised by forcing each extended-type link to have only a single arc role on all the arcs that the extended-type link contains. In practice, this is most relevant to definition extended-type links, which have four standard arc roles defined:

```
http://www.xbrl.org/2003/role/general-special
http://www.xbrl.org/2003/role/essence-alias
http://www.xbrl.org/2003/role/similar-tuples
http://www.xbrl.org/2003/role/requires-element
```

even though there are additional restrictions on which definition arcs may apply to which element pairs. The other extended-type links in XBRL each have only one standard arc role defined in each.

3.1.3. Extended-type links that are not necessarily processed together by consuming applications MUST have distinct role values.

Typical reasons that extended-type links are not be processed together are that the links may be incompatible (such as two alternative presentation formats that cannot be mixed), or that the links may be redundant.

3.1.4. Any role on an extended-type link other than the standard role MUST use a namespace owned by the taxonomy author.

This limits the potential for accidental merging of independently created networks of relationships.

http://www.ffiec.gov/2003/xbrl/form031
This is a role meant to identify extended-type links relating to a particular regulatory form used by the government agency "FFIEC".

3.1.5. Any role on an extended-type link other than the standard role SHOULD use the taxonomy namespace followed by "role", the linkbase type, and a human-readable name.

http://www.xbrl.org/taxonomy/int/fr/ifrs/ci/2003-10-15/role/calculation/ByFunction
This is a role meant to identify an extended-type link that contains arcs having a non-standard set of arc roles related to summation "by function".

3.1.6. Within a DTS the arcs MUST have only their standard arc roles.

A FRTA-compliant DTS MUST NOT use custom arc roles, but MUST only use the arc roles documented in the XBRL Specification. This does not prevent the publication of an additional set of schemas, role definitions and linkbases that constitute a non-FRTA compliant *superset* of a FRTA-compliant DTS.

3.1.7. All arcs MUST specify an order attribute.

This rule universally applies to all arcs in all extended-type links in all linkbases, and applies to arcs with any arc role, whether standard or custom. This rule ensures that taxonomies published conforming to FRTA have a common way of being presented in different tools. Section 3.5.3.9.6 of XBRL 2.1 Specification indicates that the order attribute is optional, but the order attribute is required in FRTA-compliant taxonomies.

Note that each sub-network of relationships and the way it is displayed to a user MAY bear no resemblance to any other sub-network. For example, a display in which the definition essence-alias arcs show each essence item as the parent of a list of alias items need bear no relationship to presentation parent-child or calculation summation-item arcs.

3.1.8. A DTS SHOULD ensure that two arcs to the same parent having the same arc type and arc role within extended-type links having the same role, have distinct values for the order attribute.

It is desirable for a DTS to have a deterministic ordering among siblings when displayed. This is always possible to ensure even for a DTS that imports two otherwise incompatible DTS's, by overriding arcs that introduce non-deterministic ordering.

3.1.9. All arc-type elements MAY have use and priority attribute values.

XBRL processors interpret the use and priority attributes as detailed in Section 3.5. to 3.9.5 of the XBRL 2.1 Specification.

3.1.10. All extended-type, locator-type, arc-type, and resource-type elements MAY have a title attribute.

XBRL processing ignores the title attribute. The title attribute is intended for use by XLink processors.

3.1.11. Taxonomy creators MAY provide `show` and `actuate` attribute values on linkbase arcs.

XBRL processing ignores the `show` and `actuate` attributes. These attributes are intended for use by XLink processors.

3.2. Rules for presentation relationships

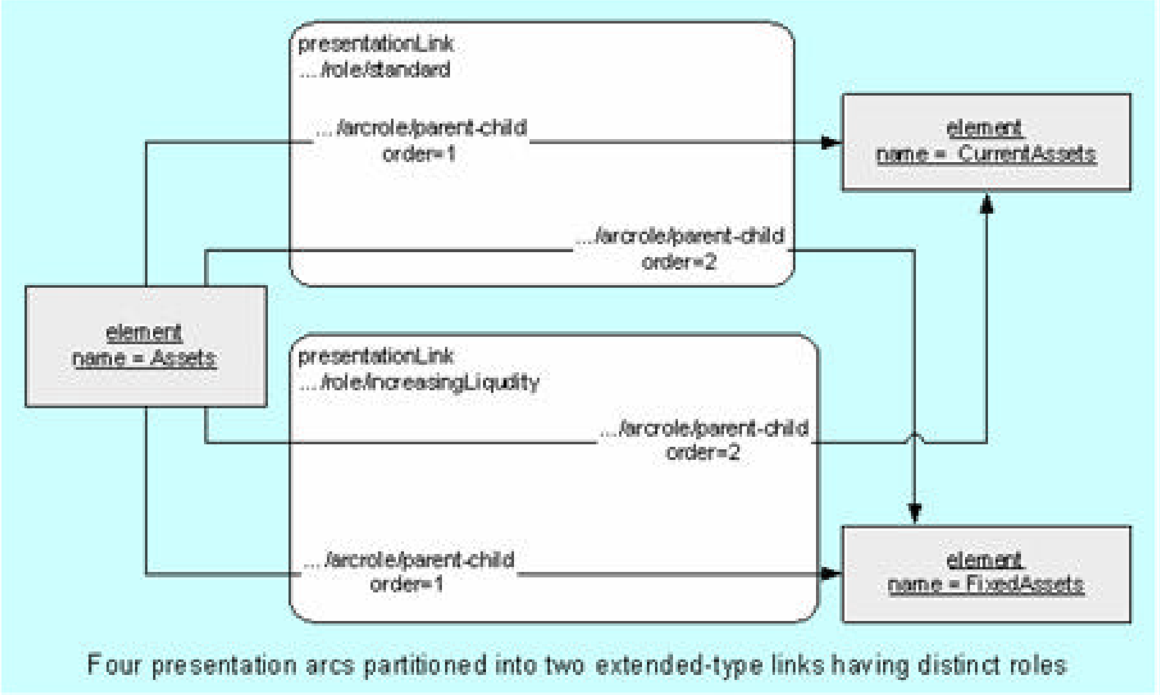
Presentation relationships are used to arrange taxonomy concepts into hierarchies with specific orderings for siblings. The usual purpose of a presentation linkbase is to show taxonomy elements in a hierarchical structure that is broadly familiar from printed reports or other standard displays. This helps users to find, identify and distinguish concepts.

In general, different reporting purposes will require different hierarchies. For example, one set of extended-type links and arcs might contain relationships that organise concepts into line items for a financial statement; another might organise the same set of concepts or a subset of these same concepts into a data collection form.

3.2.1. A DTS MAY contain any number of sets of extended-type links partitioned by role.

Any given DTS has a (possibly empty) set of presentation extended-type links that is partitioned according to the values of their `role` attributes. It is that partitioning—*not* the partitioning into files—of extended-type links within a DTS is what determines which extended-type links are processed together. Example 11 shows a simple example in which one extended-type link shows the children of an "Assets" elements in the standard way with decreasing liquidity, while a different extended-type link shows them ordered by increasing liquidity.

Example 11. Presentation extended-type links, roles, arcs and arc roles



Four presentation arcs partitioned into two extended-type links having distinct roles

```
<linkbase
xmlns="http://www.xbrl.org/2003/linkbase"
xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
  <presentationLink
    xlink:role="http://www.xbrl.org/2003/role/standard" xlink:type="extended">
    <loc xlink:label="my_Assets"
         xlink:href="my.xsd#my_Assets" xlink:type="locator"/>
    <loc xlink:label="my_CurrentAssets"
         xlink:href="my.xsd#my_CurrentAssets" xlink:type="locator"/>
    <loc xlink:label="my_FixedAssets"
         xlink:href="my.xsd#my_FixedAssets" xlink:type="locator"/>
    <presentationArc
       xlink:arcrole="http://www.xbrl.org/2003/role/parent-child"
       order="1" xlink:from="my_Assets" xlink:to="my_CurrentAssets"
       xlink:type="arc"/>
    <presentationArc
         xlink:arcrole="http://www.xbrl.org/2003/role/parent-child"
         order="2" xlink:from="my_Assets" xlink:to="my_FixedAssets"
         xlink:type="arc"/>
  </presentationLink>

  <presentationLink
    xlink:role="http://www.xbrl.org/taxonomy/int/fr/ifrs/ci/2003-12-
25/role/presentation/increasingLiquidity" xlink:type="extended">
    <loc xlink:label="my_Assets"
         xlink:href="my.xsd#my_Assets" xlink:type="locator"/>
    <loc xlink:label="my_CurrentAssets"
         xlink:href="my.xsd#my_CurrentAssets" xlink:type="locator"/>
    <loc xlink:label="my_FixedAssets"
         xlink:href="my.xsd#my_FixedAssets" xlink:type="locator"/>
    <presentationArc
       xlink:arcrole="http://www.xbrl.org/2003/role/parent-child"
       order="2" xlink:from="my_Assets" xlink:to="my_CurrentAssets"
       xlink:type="arc"/>
    <presentationArc
         xlink:arcrole="http://www.xbrl.org/2003/role/parent-child"
         order="1" xlink:from="my_Assets" xlink:to="my_FixedAssets"
         xlink:type="arc"/>
  </presentationLink>
</linkbase>
```

Note that in Example 11 it does not matter whether the four arcs had been placed in four separate presentation links instead of two, so long as they remained within a presentation link having the same `role` value as before.

3.2.2. A role attribute used on one or more `presentationLink` elements MUST indicate the purpose of that set of presentation links.

This is a consequence of rules 3.1.1, 3.1.3 and 3.1.5 and the fact that different presentation relationship hierarchies MUST be defined in extended-type links with different role values.

http://www.xbrl.org/taxonomy/int/fr/ifrs/ci/2003-10-15/role/presentation/increasingLiquidity
This is a role meant to identify a presentation link that contains arcs in which presentation siblings are ordered by increasing liquidity.

3.2.3. A concept meant to be ordered among its siblings MUST have a parent-child presentation arc from its parent concept.

This rule applies to concepts whether they are items or tuples. The XML Schema content model of a tuple does not constrain the presentation arcs in any way.

3.2.4. Presentation arcs SHOULD provide a preferred label for each concept that is the target of more than one parent-child arc.

The preferred label is used to distinguish which label an XBRL processor should use for a concept depending on which parent concept it is being presented as a child of.

3.2.5. A DTS MUST have at least one set of presentation links intended for users of the taxonomy, called the *default presentation link set*.

The default presentation link set might not provide all of the information which is necessary to exactly replicate or reconstruct the printed financial statement or other standard display. Nevertheless, a DTS submitted to XBRL International for approval must have at least a default presentation link set.

3.2.6. The default presentation link set SHOULD use the standard value of the role attribute.

It is desirable, though not required, to use `http://www.xbrl.org/2003/role/standard` as the role attribute value in the default presentation link set. One situation that might warrant a different value would be that if an imported DTS has many links that are inappropriate for the current DTS, then using a different role will be easier than overriding a large number of arcs.

3.2.7. The default presentation link set MUST NOT contain cycles of any kind.

Although XBRL 2.1 forbids directed cycles and allows undirected cycles in presentation arcs, default presentation link sets MUST form strict hierarchies.

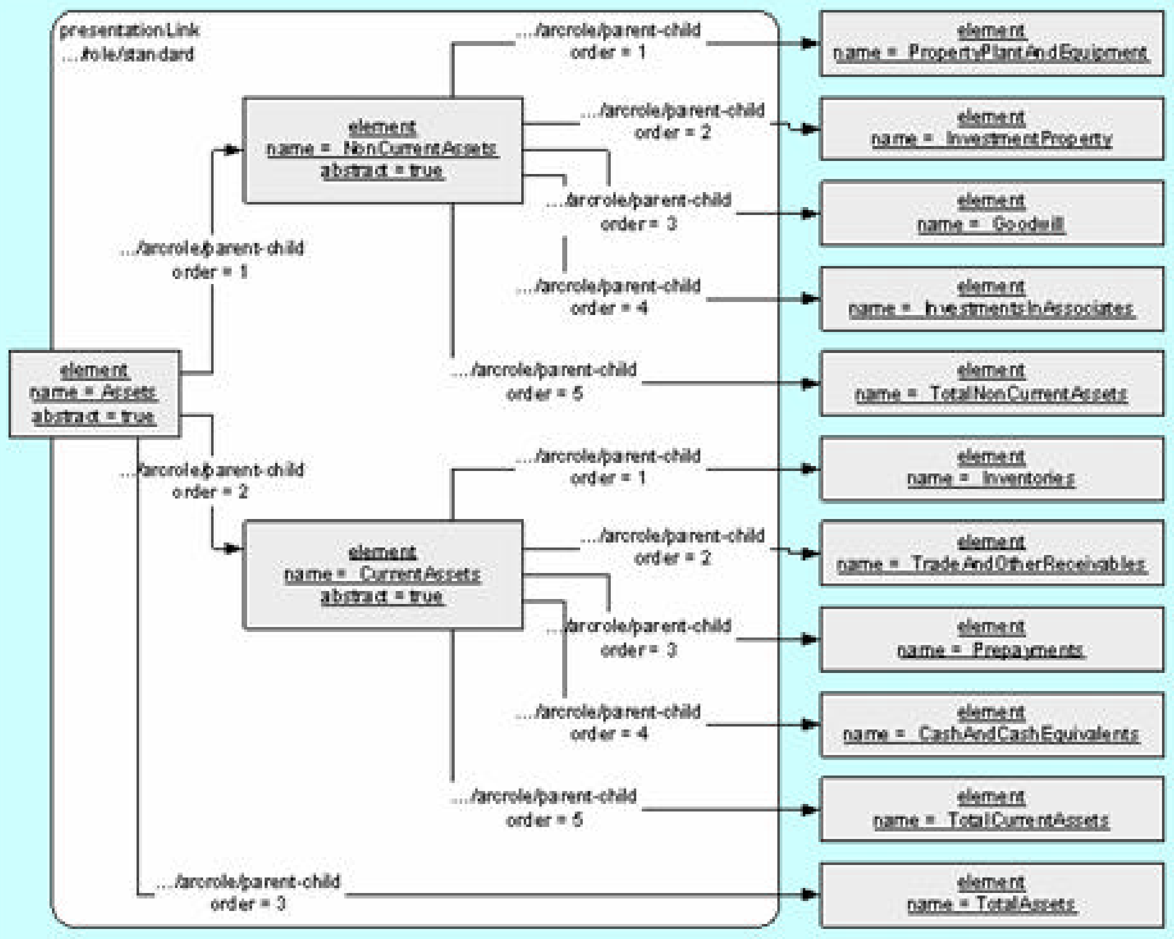
3.2.8. Abstract elements MAY be used as a heading to group other concepts for presentation.

Related financial data items and tuples are often presented together grouped under a heading or section. If the headings do not have to be tuples because each data item can stand on its own, and if there is no data item reported specifically for that heading, then an abstract element MAY be used to organize the presentation relationships. In Example 12, Assets is a heading; Non Current Assets and Current Assets are sub-headings, and there are line items beneath the sub headings. All three headings in the example are abstract elements, and the entire presentation link happens to have the standard value for `role`. See additional rules in 2.1 that apply to abstract elements.

Example 12. Grouped headings and data items.

	As of December 31,	
	2003	2002
	€’000	€’000
ASSETS		
Non Current Assets		
Property, Plant and Equipment	540,000	400,000

Investment Property	150,000	150,000
Goodwill	140,000	140,000
Investments in Associates	60,000	60,000
Total Non Current Assets	890,000	760,000
Current Assets		
Inventories	350,000	175,000
Trade and Other Receivables	490,000	590,000
Prepayments	5,000	5,000
Cash and Cash Equivalents	849,000	547,000
Total Current Assets	1,694,000	1,317,000
Total Assets	2,584,000	2,077,000



presentationLink
…/role/standard
element name = Assets abstract = true
…/arcrole/parent-child order = 1
element name = NonCurrentAssets abstract = true
…/arcrole/parent-child order = 1
element name = PropertyPlantAndEquipment
…/arcrole/parent-child order = 2
element name = InvestmentProperty
…/arcrole/parent-child order = 3
element name = Goodwill
…/arcrole/parent-child order = 4
element name = InvestmentsInAssociates
…/arcrole/parent-child order = 5
element name = TotalNonCurrentAssets
…/arcrole/parent-child order = 2
element name = CurrentAssets abstract = true
…/arcrole/parent-child order = 1
element name = Inventories
…/arcrole/parent-child order = 2
element name = TradeAndOtherReceivables
…/arcrole/parent-child order = 3
element name = Prepayments
…/arcrole/parent-child order = 4
element name = CashAndCashEquivalents
…/arcrole/parent-child order = 5
element name = TotalCurrentAssets
…/arcrole/parent-child order = 3
element name = TotalAssets

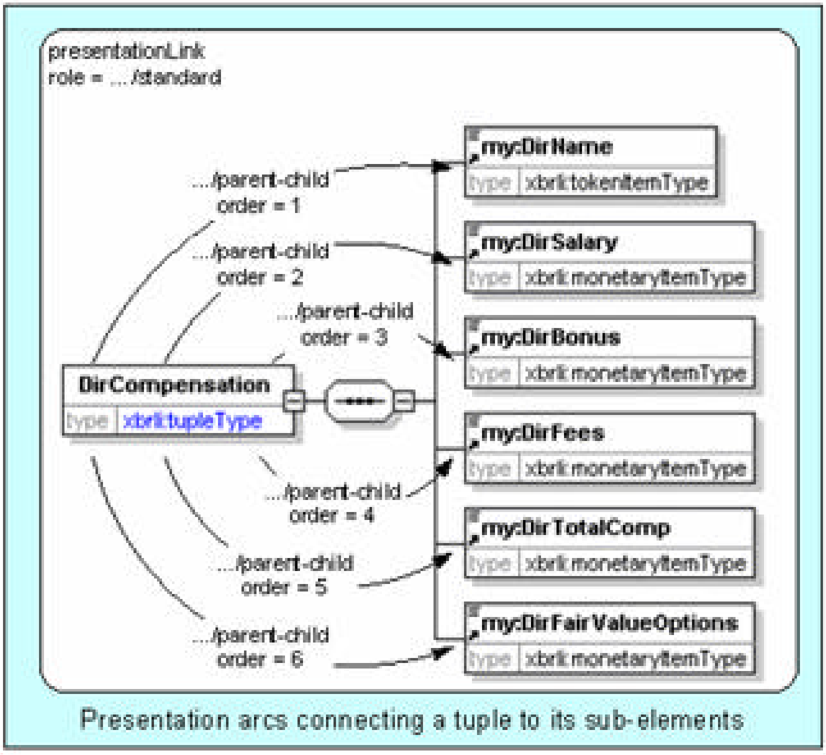
Presentation arcs connecting to abstract elements so as to create a hierarchy in a presentation link

3.2.9. The presentation children of a tuple MUST include all concepts appearing in its content model, and only those concepts.

Tuple concepts MAY appear in presentation hierarchies and the underlying structure of a tuple expressed using presentation links MUST parallel that of the XML Schema content model for that tuple concept. Its children MUST contain all elements that could appear as children of the tuple in an instance, and it MUST NOT contain other elements that do not appear in its content model. The order attribute is not constrained in any way by the content model.

Example 13 shows presentation arcs added to Example 9 above; the arcs connect the elements in the tuple to the tuple element. Presentation arcs, because they appear separately from the tuple definition itself and can exist in extended-type links with different role values, are more flexible than the tuple definition itself, which controls only the arrangement of facts within instances. Presentation arcs impose their presentation order without any regard to the nesting or arrangement of XML Schema constructs such as `sequence`, `choice`, and `all`.

Example 13. Presentation parent-child arcs in a tuple.



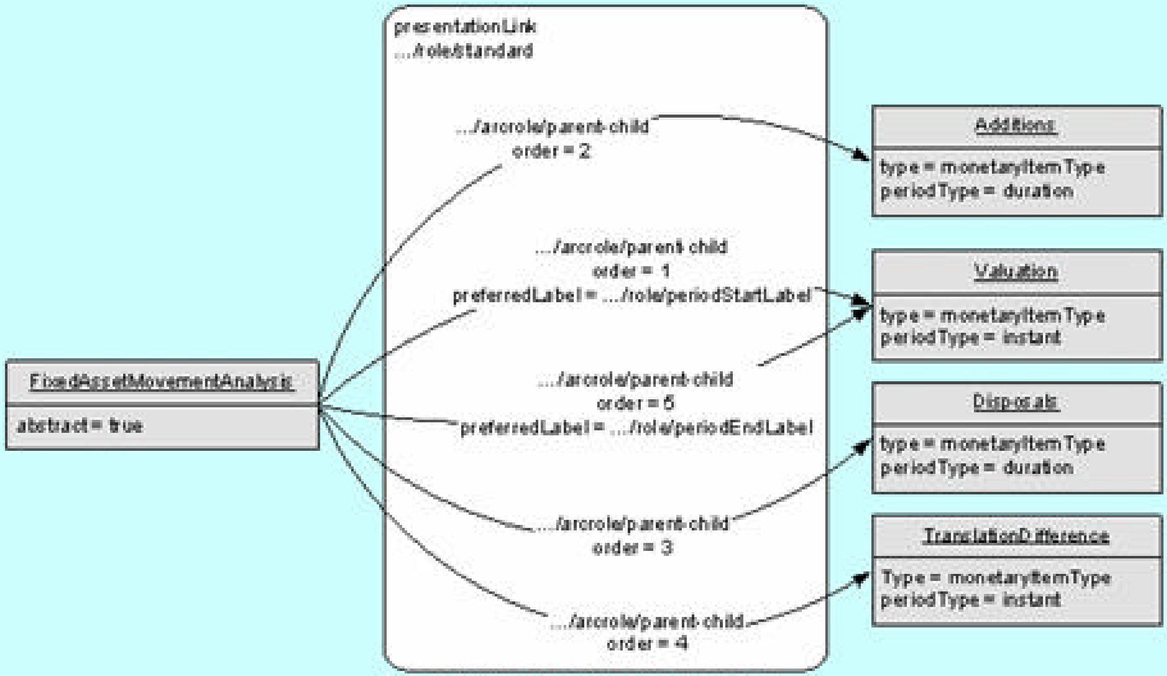
Presentation arcs connecting a tuple to its sub-elements

3.2.10. The parent-child arcs of a movement analysis MUST refer to a single item for the beginning, adjusted and ending balance values, each with a different preferred label.

Examples of movement analysis in financial reporting include the statement of changes in shareholders equity, the movement analysis for property, plant and equipment, and depreciation schedules in income tax reporting. As stated in rule 2.3.9, "The beginning balance, the ending balance, and any adjusted balances of an item for a period MUST be represented as a single item." Example 14 shows a movement analysis for fixed assets, showing reconciling items along the top, and a list of assets down the side.

Example 14. Movement analysis for fixed assets.

	Valuation/Cost				
	As at 1.1.2003	Additions	Disposals	Translation difference	As at 31.12.2003
	€'000	€'000	€'000	€'000	€'000
Land and Buildings	244,508	109,659	(193)	12,401	366,375
Furniture and Fixtures	34,457	0	0	0	34,457
Other	6,702	7,100	(262)	(7,487)	6,053
Total	285,667	116,759	(455)	4,914	406,885



Presentation of a movement analysis

3.3. Rules for calculation relationships

Calculation relationships, expressed using `summation-item` arcs in calculation extended-type links, allow taxonomy authors to document the meaning of items in terms of other items representing their mathematical components. Where the calculation relationships are sufficiently restricted that they can be expressed entirely within a single context (same period, same entity, same scenario), fully conforming XBRL processors will also use the calculation links as constraints on the consistency of instances. In general, a formula involving items A, B and C that is expressed as:

A = B - C

Is represented by two `summation-item` relationships:

- From A to B, weight 1.0;
- From A to C, weight -1.0.

Calculation arcs are designed so that taxonomy extensions can add new concepts to existing formulas without restating the parts of the formula that they are not altering. Therefore, an extension taxonomy could express the new formula

A = B – C + D

With an additional `summation-item` relationship:

- From A to D, weight 1.0;

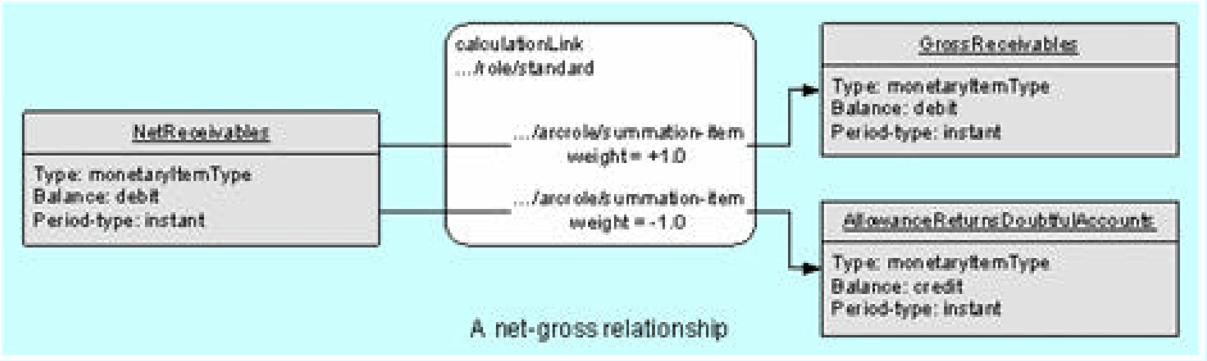
3.3.1. All concepts in a DTS which have an additive relationship in all equal contexts MUST have calculation relationships in that DTS.

Taxonomy authors MUST supply a calculation relationship for any two concepts in the same DTS, whenever it is the case that in *any* context, one item is a mathematical component of the other.

For example, suppose that a DTS encompasses the concepts "Gross receivables", "Net receivables" and the adjustment "Allowance for returns and doubtful accounts", and further suppose that the documented definitions of these concepts indicate that the relationship is a total ("Gross") with two items "Net" and "Adjustment". Mathematically this is identical to the "A = B – C" example illustrated above and so the calculation links are structured identically.

Example 15. A Net and Gross relationship

Accounts receivable, net of allowances, consists of the following as of the balance sheet date:	2001	2002
	€'000	€'000
Gross accounts receivable	18,280	13,472
Less allowance for returns and doubtful accounts	(5,687)	(4,682)
Net accounts receivable	12,593	8,790



A net-gross relationship

In this case, calculation relationships MUST be defined relating the gross, net and adjustment total concepts.

3.3.2. Calculation relationships that represent alternative summations for the same item MUST be in extended-type links with distinct roles.

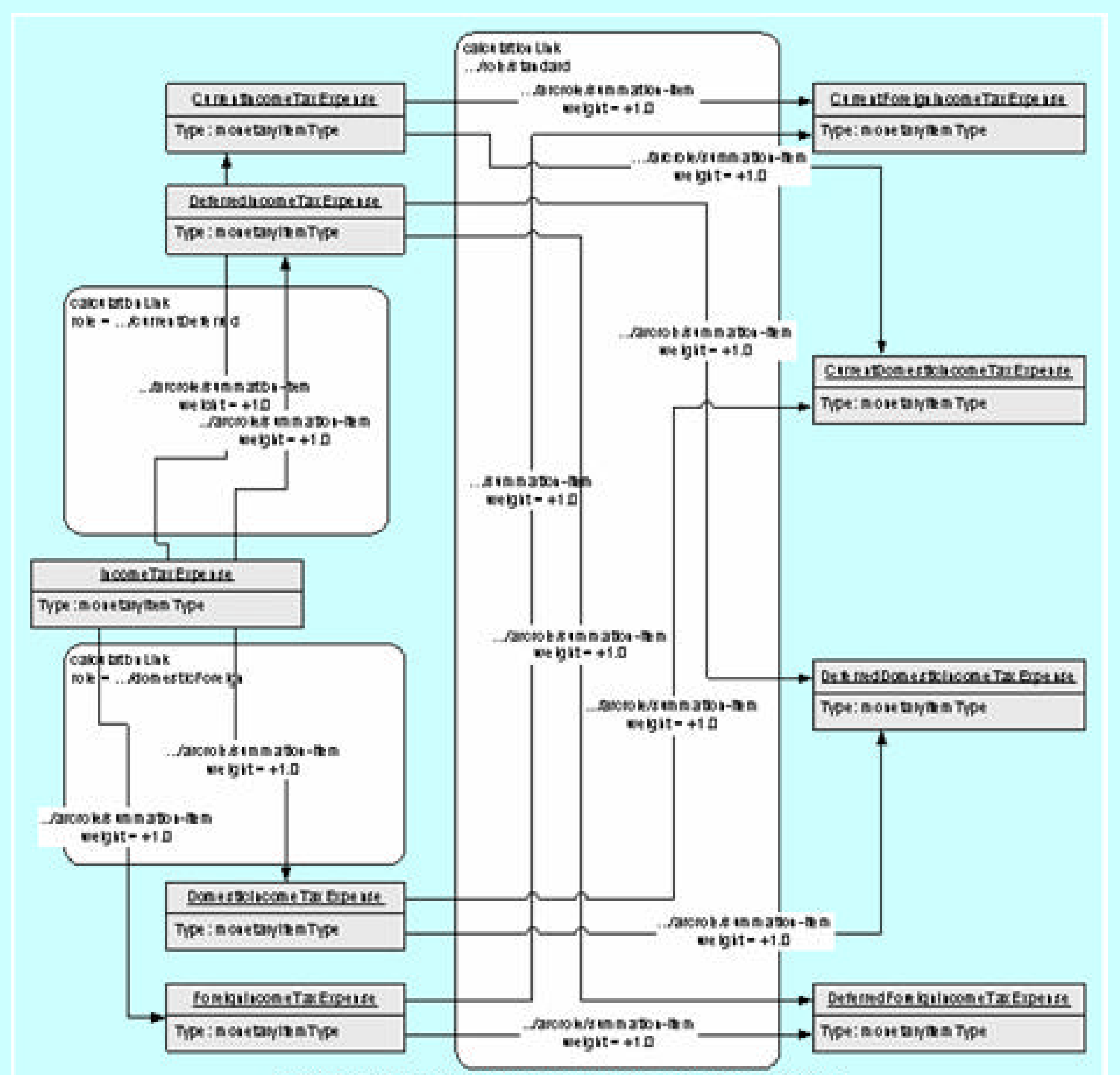
Double counting would result if two alternative ways of calculating an amount were to appear both in extended-type links with the same role. For example, total income tax expense might be calculated either by summing foreign and domestic taxes, summing current and deferred, or both. These calculations must appear in extended-type links with distinct roles.

In Example 16, three extended-type links are shown, one with the standard role value, one with role value `http://www.xbrl.org/2003/role/currentDeferrred`, and one with role value `http://www.xbrl.org/2003/role/foreignDomestic` (these are example roles; to conform with rules 3.1.4 and 3.1.5 these would be based on some other namespace).

The `summation-item` arcs in Example 16 all have weight equal to 1.0, and all of the concepts have `balance="credit"` and `periodType="duration"` since they are all expenses that are measured over a period of time.

Example 16. Two distinct summations in a financial report

The following is a summary of income tax expense:		2001 $'000	2002 $'000
Current income tax expense			
Foreign		5,408	1,994
Domestic		7,972	1,426
	Total current	13,380	3,420
Deferred income tax expense			
Foreign		6,046	838
Domestic		(90)	0
	Total deferred	5,956	838
	Total Income Tax Expense	19,336	4,258
The following is a summary of income tax expense:		2001 $'000	2002 $'000
Foreign income tax expense			
Current		5,408	1,994
Deferred		6,046	838
	Total foreign	11,454	2,832
Domestic income tax expense			
Current		7,972	1,426
Deferred		(90)	0
	Total domestic	7,882	1,426
	Total Income Tax Expense	19,336	4,258



Multiple calculations for an item, derived from a common set of items

3.3.3. Taxonomies SHOULD define an extensive set of subtotal concepts to limit the extent to which XBRL instances requiring such sub-totals need to create report-specific extensions.

Just as in Example 16, all of the items and relevant calculation arcs SHOULD be defined for cases where such alternatives are permitted.

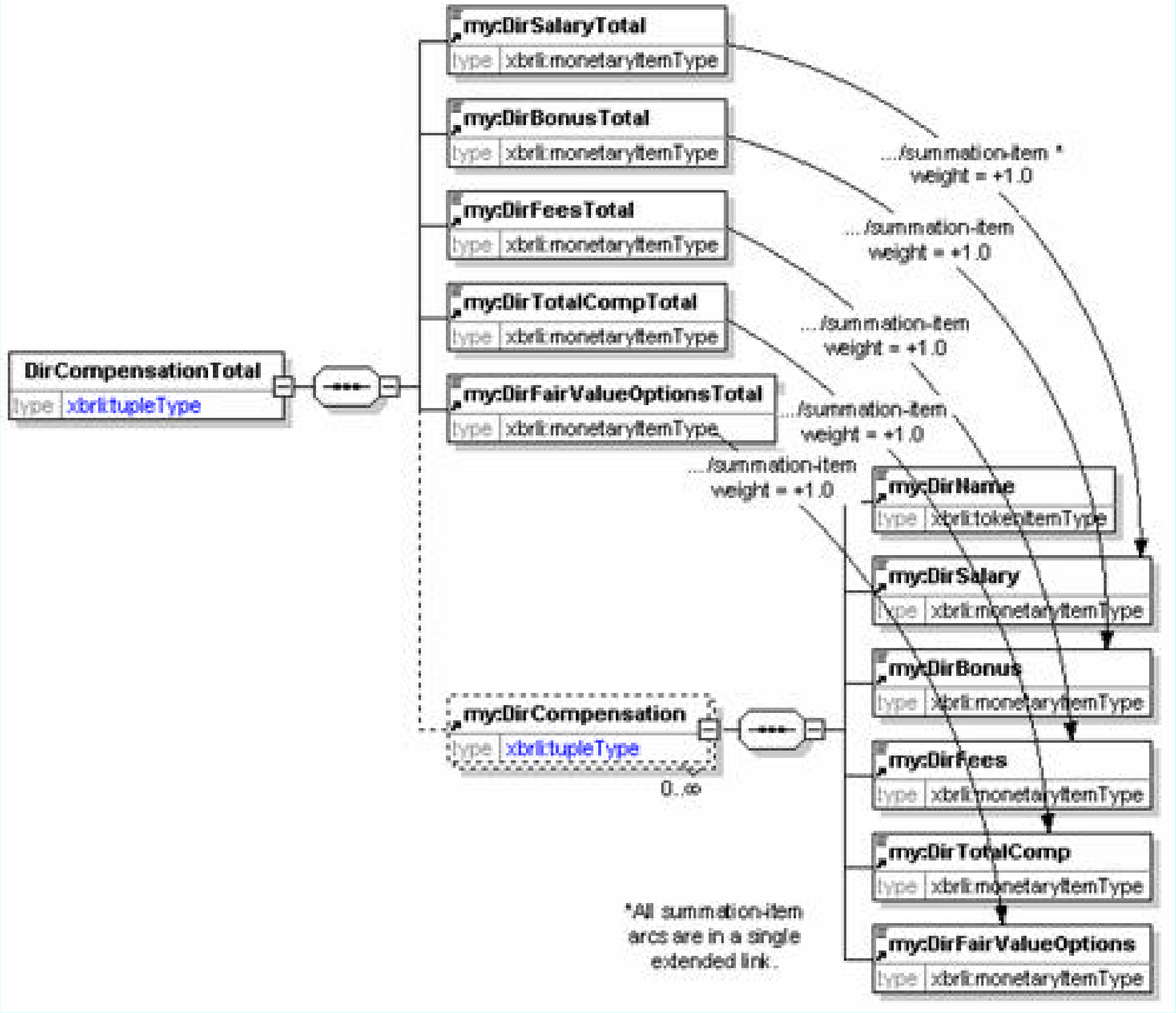
Multiple calculation hierarchies, summing a single set of concepts in multiple ways, occur in many guises in financial reporting. For example, in a classified balance sheet, assets and liabilities are totalled separately into current and non-current categories; while an unclassified balance sheet does not make the current versus non-current distinction. Classified balance sheets may also be presented as "assets = liabilities + equity," as "net assets = assets – liabilities = equity," as "net assets = assets – liabilities – minority interests = equity," and so on. These relationships MUST be defined in calculation links having different roles.

3.3.4. Calculation relationships MUST be defined between items being totalled in a tuple.

Financial reporting tables often show totals for one or more of the columns. Calculation relationships MUST be defined between the items being totalled within the table and the item that represents the total itself where such calculation relationships hold within a single context. Example 17 is similar to Example 9 except for the item "Total Salary, Bonus, and Director Fees". This is a total *within* a tuple. The total *across* the tuples is the "Total" at the bottom of the table. Each such total is a child of the enclosing tuple, here called `DirCompensationTotal`. The relationships are shown below.

Example 17. Table containing a summation across tuples.

Name of director	Salary	Bonus	Director fees	Total Salary, Bonus, and Director fees	Fair Value of Options Granted
Ho Ching	0	0	60,000	60,000	0
Boon Swan Foo	879,639	1,213,486	0	2,093,125	569,0000
Tan Guong Ching	0	0	24,200	24,200	0
Ng Kee Choe	0	0	57,000	57,000	0
Total	879,639	1,213,486	141,200	2,234,325	569,000



It is up to XBRL instance creators to ensure that their XBRL instances present the various instances of the concepts in a way that enables the calculation relationships to bind. Generally, a total item SHOULD be a sibling of the tuples that contain the items whose values aggregate to the value of the total item.

3.3.5. Calculation relationships MUST NOT be defined if the items involved in the constraint would have to be in different contexts.

Calculation relationships MUST NOT be used to describe relationships such as starting and ending balances in movement analyses *if* the starting and ending balances are represented by the same item but distinguished by different contexts. For example, there MUST NOT be any calculation relationships among the items in Example 18, because the period types are different and therefore the items are in different contexts.

Example 18. Calculation links cannot cross contexts

Item Label	periodType	Value
Cash, beginning balance	instant	100
Change in Cash	duration	-10
Cash, ending balance	instant	90

Calculation relationships cannot associate the beginning balance, adjusted balance or ending balance (see rule 2.3.9, "The beginning balance, the ending balance, and any adjusted balances of an item for a period MUST be represented as a single item."). Calculation relationships do not bind across contexts. Only the *presentation* of movement analyses can be represented using XBRL 2.1.

3.4. Rules for definition relationships

XBRL represents relationships among concepts that influence each others' values or presentation. Definition relationships allow the taxonomy author to represent relationships that are not expressed by presentation or calculation relationships. Consuming applications MAY use these definition relationships to draw inferences about the concepts.

Definition arcs are not sensitive to any portions of any context element in an instance. XBRL 2.1 provides no way to distinguish between definition arcs that should only apply to one entity in an instance and not the other. Definition links are a "blunt instrument" and because of the variety of situations in which they might be used, none of the rules that govern their use are phrased as mandatory ("MUST") rules.

3.4.1. Items in different taxonomy schemas that are equivalent SHOULD be indicated by essence-alias arcs.

Section 5.2.6.2 of the XBRL 2.1 Specification [XBRL] imposes the constraint that items connected by an essence-alias arc must have the same item type and must have identical values within the same context in an instance. Also, rule 2.1.1 ("A taxonomy schema MUST define only one concept for each separately defined class of facts.") means that each taxonomy schema MUST use unique elements to express unique concepts.

Therefore, the intended use of the essence-alias arc is to map equivalence between taxonomies. In fact, because of rule 2.1.1, this rule is relevant only for arcs where the source and target are in different taxonomy schemas. There are no requirements governing which concept is chosen as the essence (source) and which the alias (target) in the relationship.

3.4.2. Items that fall into the same category or family SHOULD be related using the general-special arc.

General-special arcs provide the user of the taxonomy assistance in identifying what a particular concept means by helping classify the concept, and can help end users to identify appropriate elements to select when mapping their own data models or terminology to a taxonomy. For example, "fixed assets" are a specialisation of "assets"; "profit" is a specialisation of "business measure"; "accumulated depreciation" is a specalisation of "contra-asset". The `general-special` arc suggests its meaning to a human observer, but conforming XBRL processors do not draw any particular inferences from the presence or absence of `general-special` arcs.

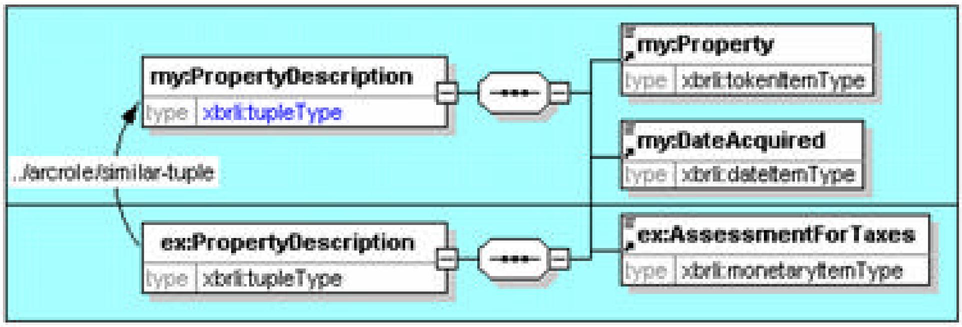
3.4.3. A tuple having the same reporting purpose as a tuple in a different taxonomy within the same DTS SHOULD have a similar-tuple arc to that other tuple.

Extension taxonomies are meant to use `similar-tuple` definition links to relate a new tuple to an existing tuple in the taxonomy that is being extended, where the new tuple had the same reporting purpose. Example 19 shows two tuples:

- `my:PropertyDescription` having a content model of only two items `my:Property` and `my:DateAcquired`, and below it,

- `ex:PropertyDescription` having the same two items followed by a third item, `ex:AssessmentForTaxes`.

Example 19. Similar-tuple documents relations between old and new tuples.



In a strict sense, "similar" tuples are tuples with similar meanings but different content models. The `similar-tuples` arc role is used to indicate that two different tuple concepts are both designed to serve the same purpose, for example, to relate two mailing address tuples with different address structures. This arc role is for the documentation of relationships between tuples and a conforming XBRL processor draws no inferences from it. The most common circumstance contemplated is where a new tuple has been added to a DTS via an extension taxonomy. This provides a mechanism for documenting relationships between a new tuple and its predecessor, without encouraging the use of the XML Schema `redefine` construct.

3.4.4. The `requires-element` arc MUST NOT be used when a tuple construct can adequately represent the same constraint.

As stated in 5.2.6.2 [XBRL], "If an instance of the concept at the source of the arc occurs in an XBRL instance then an instance of the arc's target concept MUST also occur in the

XBRL instance." A conforming XBRL processor will enforce this constraint on instances. A similar effect can be achieved with the following tuple content model:

```
<choice>
  <all>
    <element ref="TheElement">
    <element ref="TheElementThatIsRequired">
  </all>
  <element ref="TheElementThatIsRequired" minOccurs="0"/>
</choice>
```

However, the intent of the reporting standard being expressed by the taxonomy may be more or less restrictive than that. 5.2.6.2 [XBRL] also points out that "this requirement does not impose requirements on relative locations of the concept instances in tuples." Therefore, if the intent of the taxonomy to require one element if another appears, irrespective of content, irrespective of where the element appears in the instance, and irrespective of usage by other taxonomies, that is the *only* appropriate use of the `requires-element` arc.

4. Discoverable taxonomy set layer

The DTS layer of the financial reporting taxonomy architecture encompasses the scope, syntax, naming and documentation relating to a DTS rooted at a given taxonomy schema.

4.1. Scope of discoverable taxonomy sets for financial reporting

For financial reporting, a DTS should include the concept definitions and documentation and relationships that describe:

1. Required financial reporting disclosures; and
2. Common practices in financial reporting.

The goal of a financial reporting DTS should be to provide users of that DTS with what is commonly contained within financial reported information within the jurisdiction and industry in which an entity operates.

It is up to entities reporting using a specified financial reporting DTS to extend that DTS for specific disclosures which are material to that entity, but are not covered by the DTS.

4.2. Rules for discoverable taxonomy set structure

The DTS rules governing the process of discovering all the files of a DTS are documented in the XBRL Specification section 3.2 [XBRL]. The rules in this section cover appropriate usage and syntactic constraints on the files in a DTS.

4.2.1. A DTS MUST contain only schemas and linkbase documents containing definitions depending on the XBRL specification.

Specifically, a DTS must contain only:

- taxonomy schemas that define XBRL concepts only;
- taxonomy schemas that define customised XBRL item types;
- taxonomy schemas that define the content of context segments and scenarios;
- taxonomy schemas that define reference parts;
- taxonomy schemas that define custom roles and arc roles;
- the taxonomy schemas defined as part of the XBRL specification; and
- linkbase documents.

4.2.2. Taxonomy schemas MUST be defined in XML documents that have the XML Schema “schema” element as their root element.

All valid taxonomy schemas are therefore also valid XML Schemas [SCHEMA-1], [SCHEMA-2].

4.2.3. Taxonomy schemas MUST contain only one taxonomy schema.

This follows from rule 4.2.2.

4.2.4. Taxonomy schemas MUST NOT contain mark-up that is not part of that taxonomy schema.

Naturally, the taxonomy schemas contain `linkbaseRef` elements within its annotation elements, and its `element` elements will contain XBRL-specific attributes.

4.2.5. Taxonomy schemas MUST NOT contain embedded linkbases.

This is a consequence of 4.2.4 to clarify that a linkbase is not considered a part of a taxonomy schema.

4.2.6. Taxonomy schemas MUST declare elementFormDefault to be “qualified” and attributeFormDefault to be “unqualified”.

This rule ensures consistent treatment of references to attributes and elements in element definitions.

4.2.7. A linkbaseRef element MUST NOT have a null role value.

Although Table 2 in the specification [XBRL] allows an empty role value, this rule forces the value to be one of the five specified values corresponding to the type of the target linkbase.

4.2.8. Extended-type links MUST be defined in linkbase documents that have the linkbase element as their root.

As a consequence, linkbase documents will not contain any elements that are not part of a linkbase of extended-type links defined by XBRL.

4.2.9. Each linkbase element MUST contain only one type of XBRL extended-type link.

Each linkbase (and, by rule 4.2.7, each linkbase document) only contains one of the types `labelLink`, `referenceLink`, `definitionLink`, `calculationLink` or `presentationLink` elements.

4.2.10. A label linkbase MUST only contain labels defined in a single language.

The `xml:lang` attribute of all `label` elements within a linkbase must therefore be identical; for example, `en` and `en-uk` are not considered a single language.

4.2.11. Any number of taxonomy schemas MAY contain links to select schemas and linkbases to enable discovery of unique DTS’s.

A DTS MAY be defined in such a way that it includes other DTS’s acknowledged by XBRL International. To ensure discovery of specific taxonomy components from a given starting document, that starting document simply provides physical links to those other documents.

A taxonomy schema used as a starting document MAY therefore contain only `import`, `include` and `linkbaseRef` elements without any `element`, `complexType` or any other XML Schema elements. This allows for controlled discovery of certain taxonomies for specific reporting purposes and MAY be distributed as part of a DTS for financial reporting.

In Example 20, there are three DTS's:

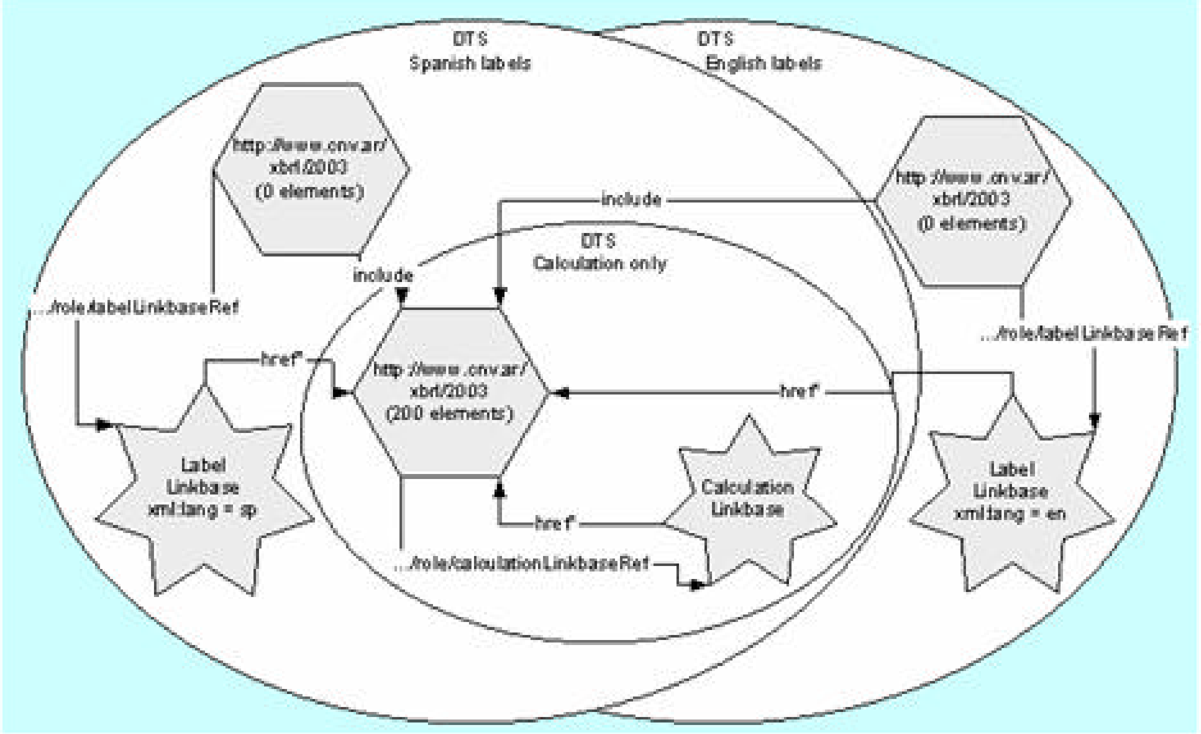
1. A taxonomy schema of 200 elements and an associated calculation linkbase;

2. A taxonomy schema of zero elements that includes that first schema and a reference to a linkbase of Spanish labels;

3. A third formed from a different empty Schema consisting only of an import and a link to a different (English labels) linkbase.

Note that only the 2nd and 3rd discoverable taxonomy sets are FRTA-compliant; the first, lacking labels, violates rule 2.1.10.

Example 20. Three distinct discoverable taxonomy sets.


DTS
Spanish labels
DTS
English labels
http://www.cnv.ar/ xbrl/2003 (0 elements)
include
http://www.cnv.ar/ xbrl/2003 (0 elements)
DTS
Calculation only
include
.../role/labelLinkbaseRef
.../role/labelLinkbaseRef
href"
http://www.cnv.ar/ xbrl/2003 (200 elements)
href"
Label Linkbase xml:lang = sp
Calculation Linkbase
Label Linkbase xml:lang = en
href"
.../role/calculationLinkbaseRef

```
<!-- Schema element definitions and link reference only to calculations -->
<schema targetNamespace="http://www.cnv.ar/xbrl/2003"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema">
  <annotation>
    <appinfo>
      <link:linkbaseRef xlink:href="cnv-calculation.xml"
        xlink:type="simple"
        xlink:role="http://www.xbrl.org/2003/role#calculationLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
      </appinfo>
    </annotation>
  <import namespace="http://www.xbrl.org/2003/instance" schemaLocation="xbrl.xsd"/>
  <!-- 200 elements go here -->
</schema>
```

```
<!-- Schema with reference to Spanish labels -->
<schema targetNamespace="http://www.cnv.ar/xbrl/2003"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema">
  <annotation>
```

```
    <appinfo>
      <link:linkbaseRef xlink:href="cnv-label.xml"
        xlink:type="simple"
        xlink:role="http://www.xbrl.org/2003/role#labelLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
      </appinfo>
    </annotation>
  <import namespace="http://www.cnv.ar/xbrl/2003" schemaLocation="cnv.xsd"/>
</schema>
```

```
<!-- Schema with reference to English labels -->
<schema targetNamespace="http://www.cnv.ar/xbrl/2003"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema">
  <annotation>
    <appinfo>
      <link:linkbaseRef
        xlink:type="simple" xlink:href="cnv-label-en.xml"
        xlink:role="http://www.xbrl.org/2003/role#labelLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
      </appinfo>
    </annotation>
  <import namespace="http://www.cnv.ar/xbrl/2003" schemaLocation="cnv.xsd"/>
</schema>
```

Such additional taxonomy schemas, with or without XBRL `item`, `tuple` or other definitions, are not mandatory; the decision is up to DTS designers and SHOULD be driven by considerations of modularity (see 5.3, "Modularity goals") and control over the strength of the association between semantics and the syntax for XBRL concepts.

4.3. Taxonomy naming rules

As noted in section 1.3, "Goals of this document," a financial reporting taxonomy or extension of the USFR or IFRS taxonomy that receives Approved status [Processes] from XBRL International MUST conform to this architecture. The conventions in this section relate to taxonomy (as opposed to element) naming and related rules.

4.3.1. Taxonomy owners MUST have an XBRL International URI for all final versions of their taxonomies.

XBRL International uses this URI naming convention:

```
http://www.xbrl.org/taxonomy/{jurisdiction}/{reportingType}/{accountingType}/{industry}/{qualifier/}*{versionDate}
```

XBRL jurisdictions choosing to host taxonomies SHOULD use this taxonomy URI naming convention:

```
http://www.{jurisdiction}.org/taxonomy/{jurisdiction}/{reportingType}/{accountingType}/{industry}/{qualifier/}*{versionDate}
```

For example, the following are the URIs for IFRS-CI and US-GAAP-CI Taxonomies.

Example	Meaning
http://www.xbrl.org/taxonomy/int/fr/ifrs/ci/2002-11-15	IFRS-CI files
http://www.xbrl.org/taxonomy/us/fr/gaap/ci/2002-10-15	US-GAAP-CI files

The components are defined as follows.

Component	Definition

`http://www.xbrl.org/taxonomy/`
or
`http://xbrl-{jurisdiction}/taxonomy/`
	This is the root location of all taxonomies hosted by XBRL International. Alternatively, the root location should be the root URL of the hosting site of the jurisdiction at its "taxonomy" URL.

`jurisdiction`	Indicates the jurisdiction abbreviation. Typically, jurisdictions SHOULD be the IANA country code (http://www.iana.org/cctld/cctld-whois.htm) of the jurisdiction. However, this practice cannot be used in all cases. Current jurisdictional abbreviations include: • int – International • us – United States • de – Germany • nz – New Zealand
`reportingType`	Represents the report type. Current report types: • br – Business Reporting • fr – Financial Reporting
`accountingType`	Represents the type of accounting, currently: • ifrs – International Financial Reporting Standards • gaap – Generally Accepted Accounting Standards • tax – Tax based reporting
`industry`	Indicates the industry code for the industry of the taxonomy, currently: • ci – Commercial and Industrial entities • basi – Banking and Savings Institutions
`qualifier`	Indicates any other qualifier, such as a language code, regulatory form identifier, etc.
`versionDate`	The release date of the taxonomy in the following format: YYYY-MM-DD. For example, 2003-06-30.

4.3.2. Each unique taxonomy schema target namespace MUST have a recommended default namespace prefix of four to ten characters.

The recommended default namespace prefix should suggest the distinct scope and purpose of the concepts defined within that namespace. For example:

Example	Meaning
`ifrs-ci`	IFRS-CI elements
`us-gaap-ci`	US-GAAP-CI elements

4.3.3. A taxonomy that supersedes an existing version of itself MUST use the date portion of its namespace URI to identify the new version.

The date may be the date of anticipated publication, date of the end of the comment period, or any other significant date which disambiguates the version in question from prior and subsequent versions.

4.3.4. Taxonomy file names SHOULD use the default namespace prefix and identifying date in their names.

Taxonomy file names SHOULD follow the pattern:

Schema files	`{defaultNamespacePrefix}-{date}.xsd`
Linkbase files	`{defaultNamespacePrefix}-{date}-{linkbasetype}{-qualifier}*.xml`
Label Linkbase files	`{defaultNamespacePrefix}-{date}-label{-language}{-qualifier}*.xml`

The {-qualifier} MUST NOT be used for any linkbase which is the "default" linkbase, as for example the default presentation linkbase intended for use in presenting the taxonomy.

Example	Meaning
ifrs-ci-2003-07-15.xsd	IFRS-CI schema
us-gaap-ci-2002-10-15.xsd	US-GAAP-CI schema
Us-gaap-ci-2002-10-15-labels.xml	US-GAAP-CI (default US English) labels linkbase
us-gaap-ci-2003-12-25-labels-sp.xml	US-GAAP-CI Spanish labels linkbase

A consequence of rule 4.2.11 is that a linkbase MAY have an existence distinct from the other taxonomy schemas and linkbases in its DTS. For example, the Spanish labels linkbase of a US-GAAP-CI taxonomy, for example, may have an independent publication date from the schemas it refers to, and new versions of the Spanish labels may be published at any time. The DTS whose starting point is that Spanish labels linkbase should nevertheless have a file name following the convention described in this rule.

4.3.5. The authoritative copy of all files of an approved DTS MUST be hosted by XBRL International.

This location MUST be reflected in the document URL's specified in the various taxonomy schemas and extended-type links in the DTS. XBRL International will use the path naming convention for all files comprising the taxonomy:

```
{taxonomyURI}/{taxonomyFileName}
```

Example	Meaning
http://www.xbrl.org/int/fr/ifrs/ci/2003-07-15/ifrs-ci-2003-07-15.xsd	Location of IFRS-CI schema file
http://www.xbrl.org/us/fr/gaap/ci/2002-10-15/us-gaap-ci-2002-10-15.xsd	Location of US-GAAP-CI schema file

4.4. Discoverable taxonomy set documentation rules

DTS documentation refers to the documentation provided with a DTS to explain the DTS. The documentation described in these rules MUST be provided in order for XBRL International to review and acknowledge the DTS.

4.4.1. A DTS MUST provide one page of summary information and pointers to other documentation related to that DTS.

Summary information shown in Table 1 MUST be provided by taxonomy authors on a summary page.

Table 1. Required taxonomy summary information

Name	Description
Status	Status of the DTS, as defined by XBRL International Processes REC 2002-04-20. Valid status values are: recommendation, internal working draft, public working draft
Release Type	Indicates the release type of the DTS, as defined by XBRL International Processes REC 2002-04-20. Valid values are: acknowledged, approved.
Date Issued	Date the DTS was issued for the taxonomy with the status, as defined above.
Issued by	The organisation that is issuing the DTS; preferably but not necessarily a financial reporting standards body such as FASB, IASB, Canadian Auditing Standards Committee.

Name	Description
Name	The official name of the DTS.
Description	A one to two paragraph description of the DTS.
Identifiers	The official `targetNamespaces` of the DTS.
Recommended namespace prefixes	The namespace prefixes which the DTS authors recommend to be used with this DTS. This namespace prefix helps users of the DTS understand which taxonomies are being used.
Relation to other DTS's	A brief description of the relation this DTS has to other DTS that have been released.
Incompatible taxonomies	Identify any taxonomy schemas or linkbases that must not be added to the DTS because of known semantic inconsistencies.
Physical Location of DTS Package	This is a list of hypertext pointers to the actual files of the DTS. This should include all schema files and linkbases used by the DTS.
Explanatory Notes	Links to explanatory notes to the taxonomy. Explanatory notes in HTML, PDF, and Microsoft Word formats are recommended.
Printouts of DTS Elements	Links to printouts (PDF or HTML) of DTS elements.
Samples	Sample instances using this DTS.
Errors	Summary of errors discovered in the DTS and workarounds to fix the errors.

4.4.2. A DTS must have narrative Explanatory Notes that explain the purpose of the taxonomy.

Explanatory notes to the DTS are intended to explain aspects of the DTS in a narrative form meant for the consumption of humans. The following are specific topics that should be covered in this narrative, either explicitly or by reference to other documentation:

Table 2. Required taxonomy explanatory notes

Name	Description
Summary information	Document should refer to a copy of summary information for the DTS, as defined above.
List of Editors and Contributors	A listing of editors and other contributors who participated in the creation of the DTS.
Abstract	A brief abstract that explains this taxonomy.
Terminology	Definition of any terminology that may be required to understand the documentation.
Table of Contents	A table of contents for the explanatory documentation.
Overview	Explanation of who should use this DTS.
Authority	Summary of the authority for issuing the DTS.
DTS Status	Explanation of the status of the DTS.
Scope of DTS	Discussion of the scope of the DTS as intended by its authors.
Overview of DTS	Any narrative information the authors deems appropriate for explaining the DTS to intended users.
Samples	Explanation of the samples provided.
Feedback	Where to send feedback relating to this DTS.
Intellectual Property	Statement granting a royalty-free license to all users in compliance with the IP policy of XBRL International at www.xbrl.org/legal.
Acknowledgements	Any acknowledgements the DTS authors wish to make.

4.4.3. DTS documentation MUST provide a report of DTS concepts viewed alphabetically and viewed by arc role.

Required reports include:

1. Dictionary – An alphabetical listing of elements in the taxonomy, by element name (NOT by label) showing the documentation required in rule 2.1.12.
2. Calculation links – `summation-item` view.
3. Presentation links – `parent-child` views.
4. Definition links – `general-special`, `essence-alias`, `similar-tuples`, and `requires-element` views.

MS Excel SHOULD be used to deliver the reports since that facilitates analysis by reviewers.

4.4.4. DTS documentation MUST include sample instances.

Sample instances MUST be provided to explain and help understand how to use the taxonomy. To achieve XBRL International acknowledgement, five different samples are required. Samples should provide:

1. Links to sample instances.
2. Links to sample taxonomy extensions.
3. Links to original reports that samples have been derived from.
4. Links to other explanatory materials that facilitate use of the samples.

XBRL International requires sample instances to be provided with any taxonomy achieving a recommended status. These samples should illustrate all aspects of the taxonomy and collectively cover all elements defined.

5. Taxonomy Extensions

Taxonomy extensions add concepts and modify the relationships among the concepts in the *base taxonomies* that they extend. Extension taxonomies will commonly be created to support specialised reporting requirements in specific accounting jurisdictions, in specific industries, or for specific companies. Taxonomy extensions consist of a set of taxonomy schemas and/or linkbases that augment a DTS that includes the base taxonomies. Rules relating to extensions include rules of syntax and rules of documentation.

5.1. Rules for extension taxonomies

5.1.1. An extension MUST NOT modify the meaning of concepts in the base.

More precisely, extensions MUST NOT modify the meaning of concepts as documented in the base in ways that are inconsistent with the meaning of those concepts defined by any DTS that includes the base but excludes the extension.

5.1.2. Word choice in the labels of an extension SHOULD be consistent with the terminology used in its base.

This rule, while seemingly self evident, does allow for the possibility that a legitimate purpose for an extension might be to apply a different, but consistent terminology, as for industry specific terminology applied to a concept known by a more generic term in the base.

5.1.3. An extension that defines new concepts MUST have its own target namespace distinct from the namespaces of its base taxonomies.

Concepts added to an extension taxonomy MUST reside within their own namespace(s), distinct from the namespace(s) of the base taxonomy and MAY have relationships with concepts in the base taxonomies and other concepts in the extension taxonomies.

Note that in Example 20 the use of `include` allows both the "Spanish labels" and "English labels" extension taxonomies to share the same namespace, because neither extension defined any new concepts.

5.1.4. Extensions MUST NOT change the content model of tuples in the base.

The XML Schema `redefine` construct is not allowed in FRTA-compliant taxonomies even though the XBRL 2.1 Specification does not forbid its use. For example, an extension cannot add a seventh concept to the tuple in Example 9. The extension MAY define a new tuple that contains each of the old items as well as any new items that are needed. The `similar-tuple` relation documents the relationship between the old and new tuples.

5.1.5. An extension needing a tuple that is consistent with the meaning of an existing tuple in the base MUST be defined in the extension taxonomy schema.

If a new content model is required to report content that is consistent with the meaning of another tuple that has already been defined then a new tuple MUST be created in the extension taxonomy to represent the new content model. Rule 3.4.3 also indicates that the `similar-tuples` arc should be used to document the relationship thus established.

5.1.6. An extension SHOULD NOT add new concepts that would be equivalent to concepts in the base.

Every concept defined in a taxonomy MAY be used in another taxonomy simply by importing the schema that defines it.

A fundamental goal of XBRL is "to enhance the creation, exchange, and comparison of business reporting information" [XBRL]. Comparability of instances is enhanced when the same concept is represented by the same element.

Rule 2.1.1 states that "A taxonomy schema MUST define only one concept for each separately defined class of facts." While impractical to enforce at the level of a DTS, it is nevertheless the underpinning of the current rule.

Prior to adding a new element in the extension, consideration should be given to the use of an existing concept in the base. Where such a concept exists in the base it SHOULD be imported and referenced by the extension. In these cases a new concept SHOULD NOT be created.

5.1.7. An extension that defines a concept equivalent to a concept in a Recommended XBRL International taxonomy MUST indicate such equivalence through a definition link.

The "recommended" state means that XBRL International singles out this taxonomy as the most appropriate one to use for the set of concepts it contains [Processes]. For example, the Global Common Document taxonomy issued on 15 November 2002 is the recommended 2.0 version taxonomy for concepts such as "company name" and "postal address." It follows that extension taxonomies SHOULD NOT create new concepts to cover these terms in the GCD, but if they do, they MUST provide an `essence-alias` or `similar-tuples` arc, as appropriate.

5.1.8. An extension MUST NOT prohibit element-label, element-reference, essence-alias, general-special or tuple-similar arcs involving an existing concept in the base.

References and labels are provided on elements in base taxonomies to assist in defining the concept. Removing these links in extension taxonomy may alter the intended meaning of the concept from the base taxonomy and is therefore not permitted. See rule 5.1.11 for the preferred method.

If these existing link(s) need to be prohibited then a new concept should be defined.

5.1.9. An extension MAY prohibit requires-element, parent-child, and summation-item arcs involving an existing concept drawn from the base.

5.1.10. An extension MAY augment an existing concept in the base with links having any role and arcs having any arc role.

Adding new arcs (as opposed to prohibiting existing ones) in an extension does not necessarily alter the original intent or meaning of the concept in the base taxonomy. Table 3 summarises rules 5.1.8, 5.1.9 and 5.1.10.

New labels or references MUST NOT modify the set of valid values for those concepts.

Table 3. Summary of extension rules.

Arc role	Extension MAY prohibit	Extension MAY augment
`element-label`	No	Yes
`element-reference`	No	Yes
`essence-alias`	No	Yes
`general-special`	No	Yes
`tuple-similar`	No	Yes
`requires-element`	Yes	Yes
`parent-child`	Yes	Yes
`summation-item`	Yes	Yes

5.1.11. An arc that augments an existing arc in the base SHOULD have a higher priority.

Although arcs such as `element-label` and `element-reference` must not be prohibited, applications can use the `priority` attribute to indicate which one should be used. For example, two labels for a given concept, both of which have the same arc role, same language, and extended-type link role, should have different values for the `priority` attribute.

5.1.12. For any existing concept in the base that will not be used, an extension SHOULD prohibit requires-element, parent-child, and summation-item arcs involving it.

XBRL 2.1 does not provide any way to eliminate a concept from a taxonomy. If an extension has specific reporting purposes that only use a subset of concepts in a DTS, then that extension taxonomy SHOULD prohibit the presentation, calculation and definition links from the base taxonomy that are not relevant to the reporting purpose of the extension.

5.1.13. Any value of href in an extension where the intent is for that href to be equivalent to a prior use of href in the base MUST resolve to an identical absolute URI.

This rule guarantees the proper matching of `href` attribute values when they are intended to form matching pairs, as for example in arcs with `use="prohibited"`. Section 3.5.1.2 of the XBRL 2.1 Specification allows both `#id` and `#xpointer` syntax for fragment identifiers, and Section 3.5.1.5 notes that the `xml:base` attribute MUST be used to resolve a relative URI to an absolute URI.

5.2. Documentation rules for extensions

Extension taxonomies have additional documentation requirements in addition to those of a normal DTS.

5.2.1. Extension documentation MUST provide a report of concepts added.

The content to be reported for each added concept is described in rule 4.4.3.

5.2.2. Extension documentation MUST provide a report of concepts existing in the base that are not to be used.

Concepts in a base taxonomy are deemed removed in the extension taxonomy by prohibiting `parent-child`, `summation-item` and `requires-element` arcs defined in the base taxonomy relevant to that concept. Note that this is not the same as a list of all concepts in the extension that have no such arcs, since those arcs could have already been prohibited in the base DTS.

5.3. Modularity goals

Modularisation decisions reflect the trade-off within a variety of goals, some of which are complementary and others of which are conflicting. The relative importance of the various goals to the taxonomy users and developers governs modularisation decisions. Here, there are nine goals listed in *descending* order of importance from the standpoint of promoting XBRL adoption. These are not stated as rules, but rather as goals, because the decision of XBRL International to accept or reject a taxonomy is not dependent on the degree of conformance to these suggestions.

5.3.1. Modules SHOULD correspond to the reporting standards and rules that they are based upon.

In many cases, XBRL taxonomies are based on a set of reporting requirements that are documented by standard-setting bodies or regulators. Adoption of such taxonomies can depend significantly on getting support from the body responsible for defining those reporting requirements. This support in turn depends crucially on the accuracy with which taxonomy content describes the reporting requirements that have been defined.

Taxonomy modularisation may be able to simplify this verification process by:

- Group concepts in a way that aligns with the grouping of definitions in the reporting standards – for example, defining a separate taxonomy schema for each standard.

- Treat the analysis of taxonomy content in the same way that a reporting purpose is treated, supporting it with structures that enable presentation of the taxonomy concept in a way that is closely aligned to the standards.

5.3.2. Modules SHOULD facilitate independent development and use.

Taxonomy modularisation can influence the speed with which a taxonomy reaches the market. By breaking a taxonomy content into modules such that subsets of them can have a stand alone role in business reporting, it becomes possible to get those modules developed, tested and to market faster than if all content has to progress at the speed of the slowest modules.

For example, the IFRS taxonomies covering the primary financial statements [IFRS] have reached market much faster than taxonomies covering the much larger set of content embodied in the extended notes and disclosures.

This motivates modularisation along *reporting purpose lines*, with particular focus on the sub-components, or divisible standalone sections of reports. That is, group concepts into taxonomy schemas where those concepts are all relevant to a specific kind of reporting purpose – for example a quarterly earnings statement or a five year summary.

This also motivates modularisation of the supporting linkbases into groups of relationships and documentation that support specific reporting purposes.

5.3.3. Modules SHOULD be comprehensible to domain experts.

Taxonomy modularisation can influence the comprehensibility of the DTS's that can be constructed from the taxonomy files. If a taxonomy schema were to contain concepts with few or no relationships to each other and no common reporting purpose then their DTS's would provide very little insight into the purpose of the taxonomy, restricting its usefulness.

Ways to achieve comprehensibility include:

- Design taxonomy modules along reporting purpose lines so that the reporting purpose binds together the concepts in the module;

- Ensure that files include discovery paths so that DTS's will not encompass incoherent or misleading sets of files;

- Avoid including discovery paths between files that define taxonomy content having no common reporting purpose.

5.3.4. Modules SHOULD allow distributed taxonomy development.

Taxonomy modularisation can influence the ease with which taxonomy development can be distributed among individuals. By carving taxonomy content into modules, an entire taxonomy structure can be parcelled out to a variety of individuals working independently.

Steps that facilitate distributed taxonomy development include:

- Design the taxonomy modules along narrow reporting purpose lines allowing modules to document all aspects of the taxonomy content related to that reporting purpose.

- Avoid partitioning into modules that mutually depend on one another.

5.3.5. Modules SHOULD ease version control.

Version control is made most difficult if all concepts are defined in a single schema and the discovery of every potentially relevant linkbase is made discoverable from that schema. To make updating easier:

- Partition taxonomy concepts into groups/modules that are more likely to be updated together (for example, group according to accounting standard);

- Limit to the bare essentials those linkbase references appearing in taxonomy schemas that define concepts. This makes it possible to update linkbase files by simply creating new linkbases and incorporating them in new DTS's that do not modify the underlying schemas and linkbases.

Of course, the difficulty of version control is fundamentally dependent on the degree of expected volatility as well as the number of different situations the taxonomy will be used; branching versions needed to support different use cases are difficult to manage no matter what modularisation is used.

5.3.6. Modules SHOULD ease taxonomy extension.

Taxonomy extension to support a particular reporting purpose is made easier when the organisation of a base taxonomy allows one to:

- Draw upon the parts of a base DTS that are needed;

- Ignore parts of a base DTS that are *not* needed.

This is another reason to limit linkbase references appearing in schemas that define concepts, and to group concepts together when they are likely to be used or modified together.

For example, modularising a general financial reporting DTS with partitioning of concepts along lines governed by differences in industry requirements would assist in the creation of industry taxonomy extensions.

5.3.7. Modules SHOULD minimise the number of redundant concepts defined in DTS's supporting specific reporting purposes.

Irrelevant and distracting concepts make it difficult to map a large taxonomy to existing data.

- Modularising taxonomy schemas along reporting purpose lines means that a supporting DTS will not define many extraneous concepts not required for that type of reporting.

5.3.8. Modules SHOULD minimise the number of files required to express taxonomy content.

The larger the number of files in a DTS, the more opportunities there are for the URL-based XLink relationships in the DTS to break down as a result of document unavailability. Larger numbers of documents also add slightly to the complexity of the DTS discovery process.

Although other modularisation goals take priority, the number of documents in the DTS supporting an XBRL instance should be kept to a minimum.

5.3.9. Modules SHOULD minimise the number of namespaces that have to be defined for XBRL concepts.

Superfluous namespaces complicate use of XBRL by increasing the number of namespaces that have to be declared in XBRL instances.

While namespaces are a valuable means of limiting the potential for namespace clashes between XBRL concepts, these clashes are most threatening between taxonomies owned by different organisations and between old and new versions of the same taxonomies. New namespaces should be introduced only in situations where there is:

- Risk of namespace clashes because of difficulties in ensuring uniqueness of element names; or

- Desire to re-use the local names for a set of concepts when defining new concepts.

Intellectual Property Status (non-normative)

This document and the information contained herein is provided on an "AS IS" basis and XBRL INTERNATIONAL DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The attention of users of this document is directed to the possibility that compliance with or adoption of XBRL International specifications may require use of an invention covered by patent rights. XBRL International shall not be responsible for identifying patents for which a license may be required by any XBRL International specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. XBRL International specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents. XBRL International takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Members of XBRL International agree to grant certain licenses under the XBRL International Intellectual Property Policy (www.xbrl.org/legal).

References (non-normative)

[IEEE]
IEEE Standard 610.12-1990
IEEE Standard Glossary of Software Engineering Terminology
IEEE, 1990. http://www.ieee.org

[IFRS]
IFRS Taxonomy Framework
http://www.xbrl.org/taxonomy/int/fr/ias/

[IFRS-CI]
IFRS-CI, International Financial Reporting Standards (IFRS), Financial Reporting for Commercial and Industrial Entities (CI)
http://www.xbrl.org/taxonomy/int/fr/ifrs/ci/2003-07-15/

[LLLL]
The 'Lectric Law Library.
http://www.lectlaw.com/

[USFR]
US Financial Reporting Taxonomy Framework
http://www.xbrl.org/Taxonomy/us/TaxonomyFrameworkOverview.htm

[US-GAAP-CI]
US-GAAP Commercial and Industrial Taxonomy
http://www.xbrl.org/taxonomy/us/fr/gaap/ci/2002-10-15/

[Processes]
Walter Hamscher
XBRL International Processes 1.0, 4 April 2002
http://www.xbrl.org/resourcecenter

[RFC2119]
Scott Bradner
Key words for use in RFCs to Indicate Requirement Levels, March 1997
http://www.ietf.org/rfc/rfc2119.txt

[SCHEMA-0]
World Wide Web Consortium.
XML Schema Part 0: Primer.
http://www.w3.org/TR/xmlschema-0/

[SCHEMA-1]
World Wide Web Consortium.
XML Schema Part 1: Structures.
http://www.w3.org/TR/xmlschema-1/

[SCHEMA-2]
World Wide Web Consortium.
XML Schema Part 2: Datatypes.
http://www.w3.org/TR/xmlschema-2/

[XBRL]
Phillip Engel, Walter Hamscher, Geoff Shuetrim, David vun Kannon, Hugh Wallis.
Extensible Business Reporting Language (XBRL) 2.1 Specification
http://www.xbrl.org/resourcecenter

[XLINK]
Steve DeRose, Eve Maler, David Orchard
XML Linking Language (XLink) Version 1.0.
http://www.w3.org/TR/xlink/

Document history and acknowledgments (non-normative)

This document could not have been written without the contribution of many people. The participants in the XBRL Domain Working Group, public commentators, and personal advisors have all played a significant role. The XBRL International domain working group is chaired by Mark Schnitzer, Morgan Stanley and vice chaired by John Turner, KPMG. Contributing members of the working group include Peter Calvert, ICAEW; Eric E. Cohen, PricewaterhouseCoopers; Tom Egan, Deloitte; George Farkas, XBI Software; Yuji Furusho, Fujitsu; Jason Golieb, PricewaterhouseCoopers; Brad Homer, AICPA; Josef Macdonald, Ernst & Young; Jeff Naumann, AICPA; Yossef Newman, Deloitte; David Prather, IASCF; Campbell Pryde, KPMG; Geoff Shuetrim, KPMG; Matt Slavin, Ernst & Young; Tom Taylor, CICA; Alan Teixeira, ICANZ; Phil Walenga, FDIC; Hugh Wallis, Hyperion Solutions; Gary Wicklund, Eagle Technologies.

2003-03-26	Hoffman	First draft of document prepared, preliminary summary of discussions.
2003-03-27	Naumann	Edits
2003-03-29	Hamscher	Edited all sections to indicate the proposed normative status of this architecture and follow XBRL International documentation conventions.
2003-04-03	Schnitzer and Turner	Edits incorporated.
2003-04-11	Hoffman	Built out details for sections.
2003-05-04	Hamscher	Used existing material from Jeff Naumann and Charles Hoffman, and reorganised entire document into the four architecture layers, one per section. Added examples, new figures, tables, tables of contents and cross referencing. Added cross-references to Peter Calvert's Domain Topics list and interspersed notes on pending issues. The extensions layer material borrows selectively but crucially from unpublished papers by Charles Hoffman, Josef MacDonald, Jeffrey Naumann, Alan Teixiera, and the KPMG Global Services Team.
2003-05-12	Hamscher	Clarified audience intent, role of outer tuple in the first table example. Moved observations about Specification constraints back into XBRL 2.1 Draft comments. Changed all figures to use consistent arc direction. Rewrote section on contexts to generalise the point that concepts must not depend on specific entity names, periods, etc., but without writing the rule as if it depended on the element name. Changed example showing use of general-special arc to link concepts that differ in measurement, to one which shows alternative calculation methods. Added the sample use of a discriminator in a tuple and footnote regarding primary keys. Added footnote about undirected cycles. Added figures showing minimal and maximal schemas and the relationships of taxonomies, encompassing both IAS-style and NAFR-style inheritance. Incorporated suggested edits from George Farkas, Charlie Hoffman, Jeff Naumann and Trevor Pyman.
2003-06-09	Hamscher	After a round of internal feedback, made many small editorial fixes. Added a definition of the scope of financial reporting. Added the classified and unclassified balance sheet example.

		Added improved movement analysis examples from Charlie Hoffman and added movement analysis with adjustment. Incorporated a large number of comments from Peter Calvert. Copied in the LC3 element name convention as normative, and changed all figures and examples to conform to the naming convention. Edited the tuples description to include explanation of the mechanisms; also changed textual examples showing instances into figures generated from a schema development tool. Added an example of extending a tuple. Raised the profile of the general principle that XBRL taxonomies should not provide more granularity than the accounting standards they represent. Reordered the set of modularity guidelines and rewrote to indicate the need to recognise the need for compromise in any given situation. Added suggestion about arc priorities from David vun Kannon. Added suggestion about modularity and other issues from Charlie Hoffman. Added material detailing the URI components. Added disclaimers about scenario and entity identifiers to the material about contexts.
2003-07-07	Hoffman	Fixed typos. Added various comments to document as issues which need to be discussed. Incorporated Peter Calvert edits to the previous version of the document. Incorporated George Farkas comments on the previous version into this document. Removed references to Excel spreadsheet.
2003-08-27	Shuetrim	Thorough editorial review. Deleted all material relating to constructs now removed from the XBRL 2.1 Draft. Removed all criteria for assessing appropriate modularizations. Reduced all criteria for appropriate usage. Added new prohibition against having tuples as presentation parents. Removed all criteria for assessing different approaches to adding and removing calculation and other arcs.
2003-09-01	Hamscher	Edits for clarity in sections relating to concepts; repairs to diagrams, removal of new constraints pending wider discussion.
2003-10-13	Hamscher	Reorganised the introductory chapter to separate goals, scope and approach. Rewrote chapter 2 and part of chapter 3 to be a series of rules. Included section on modularity guidelines. Redrew examples to take the extended-type link roles into account. Included section on period types.
2003-10-16	Hamscher	Rewrote the Relationship layer rules incorporating work team feedback and factored rules. Promoted rules and text applicable to all linkbases into the first set of Relationship Layer rules. Rewrote the DTS and Extensions Layer chapters reorganising material into rules and promoting the more general rules to the early parts of the chapter and in some cases to earlier chapters. Collapsed several Extensions Layer rules into three rules and a table. Incorporated material from Josef MacDonald and Geoff Shuetrim relating to modularity goals for the Extensions Layer. Created new figures for DTS examples. Added rule pointing to the GCD. Added rule indicating the need to use the priority attribute to disambiguate alternative labels, references, etc. Added rule to indicate how the Date portion of the URI is to be used. Added rule that the assumed scenario value is "actual." Changed default namespace prefix rule to allow four-character

		names. Incorporated fixes and suggestions from reviews by Eric E. Cohen and Charles Hoffman, with introduction of several cross-references, new section on implications for instances of the concept rules, removal of spurious rule relating to label capitalisation, movement of period type discussion section into individual rules, repair of figures to accommodate change in standard role and arc role names.
2003-10-20	Hamscher	Incorporated feedback from Josef MacDonald and Charles Hoffman. Changed document name. Added IEEE reference and explanation of the term "architecture" versus "engineering" in the context of software systems as an end note. Removed rule relating to dateTime items, removed one exception from the non-numeric text period type rule, and folded rules regarding dateTime elements into a related rule. Added new instance guidance regarding the maximum duration of contexts, in lieu of a more lengthy discussion of context proliferation and contiguous periods. Clarified wording of rule requiring use of tuples instead of elements with numeric suffixes. Edited wording in instance guidance relating to the "actual" interpretation in the absence of a scenario. Removed term "extended link" in favour of "extended-type link" where generic, or replaced with the name of the specific link type in context. Reformatted reference example provided by Josef MacDonald. Reduced depth of table of contents to be more readable. Added additional definitions and used icons for definitions. Converted pasted bitmaps showing financial statements into regular tables and made other formatting changes to keep relevant materials together. Fixed example that had incorrect figure pasted in. Promoted footnotes into inline text where still relevant. Removed editorial notes. Fixed bug in Legend. Removed obsolete references and changed references to unpublished documents into acknowledgments of authors' contributions. Fixed bug in "…/role/standard" occurrences. Removed word "semantics" in favour of "meaning", with explanation in context. Fixed bug in example of label usage. Rewrote the rule regarding reference parts to eliminate redundancy with XBRL 2.1 specification. Added examples of non-numeric concepts from the SEC-CERT taxonomy. Clarified wording of the tuple presentation parent-child arcs rule. Resolved inconsistencies relating to prohibitions on duplicate concepts and use of the essence-alias arc. Fixed rule that indicated qualified or unqualified values for taxonomy schema elements and attributes. Relaxed the wording of rule regarding the GCD and other recommended taxonomies, to allow essence-alias relations.
2003-10-30	Hamscher	Incorporation of George Farkas' comments and final polishing for release as public working draft.

Approval process (non-normative)

This section will be removed from the final recommendation. DWG = Domain Working Group; ISC = International Steering Committee.

	Stage (* - Current)	Party responsible for decision	Next step	Revisions needed	Target date for stage completion
1	Internal WD	DWG	Recommend for Stage 2	Stay in Stage 1	2003-10-27
2	Internal WD pending publication	ISC	Approve for Stage 3	Return to Stage 1	2002-11-03
3*	Public WD under 45 day review	WD Editors	Minor revisions – to Stage 4	Major revisions, Restart Stage 1	2003-12-18
4	Draft Candidate Recommendation	DWG	Recommend for Stage 5	Restart Stage 3	2003-12-22
5	Candidate Recommendation	ISC	Approve for Stage 6	Restart Stage 4	2004-01-13
6	Recommendation	Done			

[i] Architecture: the fundamental organization of a system embodied by its components, their relationships to each other and to the environment and the principles guiding its design and evolution. This definition may just as usefully be applied to technical architecture" [IEEE]. This document describes in the form of design rules the organization of financial reporting taxonomies embodied by schemas, linkbases, concepts, links, and other components, their relationships to each other and to financial reporting standards, and principles that justify the design rules both for base taxonomies and for the extensions that will inevitably follow. Contrast this with the IEEE definition of Software Engineering: "A systematic approach to developing, using, maintaining and liquidating systems;" this document does not cover approaches do development, use, maintenance or liquidation of taxonomies.