FINANCIAL REPORTING TAXONOMY ARCHITECTURE (FRTA)

Version 1.5

This document is a sequel to the FRTA 1.0, it outlines the guidelines that govern best practices in Financial Reporting Taxonomy Architecture

Status: Public Working Draft

Date of Publication: 16 May, 2011

Index:

Contents

Document History

Version Number	Date	Author	Decription
1.0	28th February, 2011	Shweta Gupta, IRIS	First Draft
1.0.1	6th March, 2011	Roland Hommes, RHOCON	Comments
1.0.2	9th March, 2011	Shweta Gupta, IRIS	Revised based on discussion with the Working Group
1.0.3	16th March, 2011	Shweta Gupta, IRIS	Revised based on discussions with the Working Group
1.0.4	27th April, 2011	Hugh Wallis, XII	Remove all references to the TRP and editorial for publication as a PWD

Status

Circulation of this Public Working Draft is unrestricted. This document is normative. Other documents may supersede this document. Recipients are invited to submit comments to tapwg@xbrl.org, and to submit notification of any relevant patent rights of which they are aware and provide supporting documentation.

Background

XBRL International specifies this architecture 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 separately 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.

Language

The official language of XBRL International's own work products is English and the preferred spelling convention is UK English.

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 including related explanatory disclosures, 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.

Financial Reporting Layers

Following is how typically financial reporting taxonomies have been developed and are being developed and at various levels.

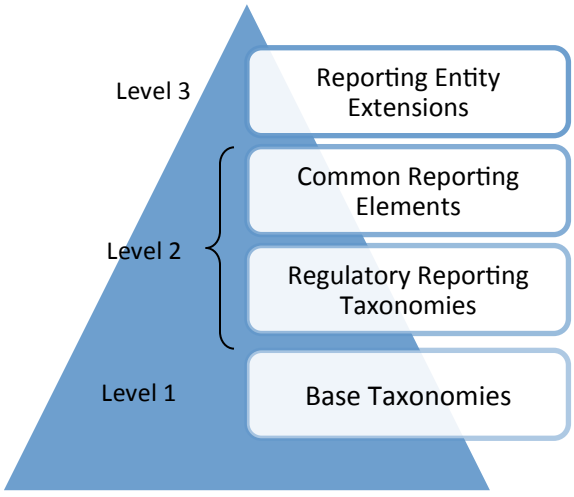


Figure 1: Typical Taxonomy Implementations

Level 1: Base Taxonomies

Base taxonomy for a country/jurisdiction/organization is the core first level taxonomy, which could be used directly to represent data, or extended to and then represents data or both. Primary characteristic of a base taxonomy is that it does not extend any taxonomy, but could serve as a base to be extended by other taxonomies.

Level 2: Regulatory & Common Reporting Elements

Taxonomies at these levels typically extend the base taxonomies to include elements for a specific regulator or a group of regulators to facilitate reporting to them in a format compliant with the regulator's regulations.

Level 3: Reporting Extensions Taxonomies

These taxonomies are created by specific entities reporting to a regulator, which may contain elements that are very specific to the entity's business, or that particular reporting period.

The scope of the document is limited to the Level 1 and Level 2 taxonomies. Rules describing will be a sequel to this document.

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.

FRTA 1.5 is a revision to FRTA 1.0, to remove any rules that have grown obsolete based on the various adoptions of XBRL in the Financial Reporting space, and the development and implementation of Dimensions 1.0 recommendation. In the event of any conflicts between this document and the Dimensions 1.0 recommendation, the Dimensions 1.0 prevails.

Goals of this document

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*
- XBRL software expertise *and* previous exposure to financial reporting concepts.

This document may also be useful to:

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

No part of this architecture requires any aspect of a taxonomy to have 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.

Organisation of this document

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

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

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

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

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.

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.

The Appendix 1 of this document outlines the changes that have been made to FRTA 1.0, for creation of FRTA 1.5. This table also maintains a link of FRTA 1.0 clauses for references.

Terminology and document conventions

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

Table 1. Terminology used in this document.

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 to development, use, maintenance or liquidation of taxonomies.
abstract element, ancestor, base set, 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 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 As defined in XBRL 2.1 specification.
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	As defined by the XML Linking Language [XLINK] . XBRL linkbases are made up of extended-type links.
FRTA	Financial Reporting Taxonomies Architecture: the set of rules described in this document.
FRTA compliant (FRTA-compliant)	An element, attribute, linkbase, schema or DTS satisfying all applicable mandatory ("must") rules in this document. Any of such artefacts that violates or ignores a recommended ("should") rule is inferior to one that obeys it and should not be emulated.
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, common practices, etc. [LLL]
LRR	Link Role Registry. An online listing of XLink role and arc role attribute values that MAY appear in taxonomies, along with structured information about their purpose, usage, and any intended impact on XBRL instance validation [LRR].
Module	An XBRL International recommendation that depends on XBRL and defines the syntax and semantics of additional elements, attributes, roles or arc roles that cannot be defined entirely within an XBRL valid taxonomy.
Persisting DTS (persisting extension)	A DTS whose purpose is to be stored as files to be referenced by instances of multiple entities and published in some fashion for users to examine. This contrasts with a DTS that is ephemeral—for example, dynamically created while processing instances, only to be discarded.
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.
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.
XBRL	XBRL 2.1 Recommendation, with corrected errata [XBRL]**Error! Reference source not found.**.
XBRL valid	XML instances and schemas that satisfy the syntax requirements of XBRL.

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.

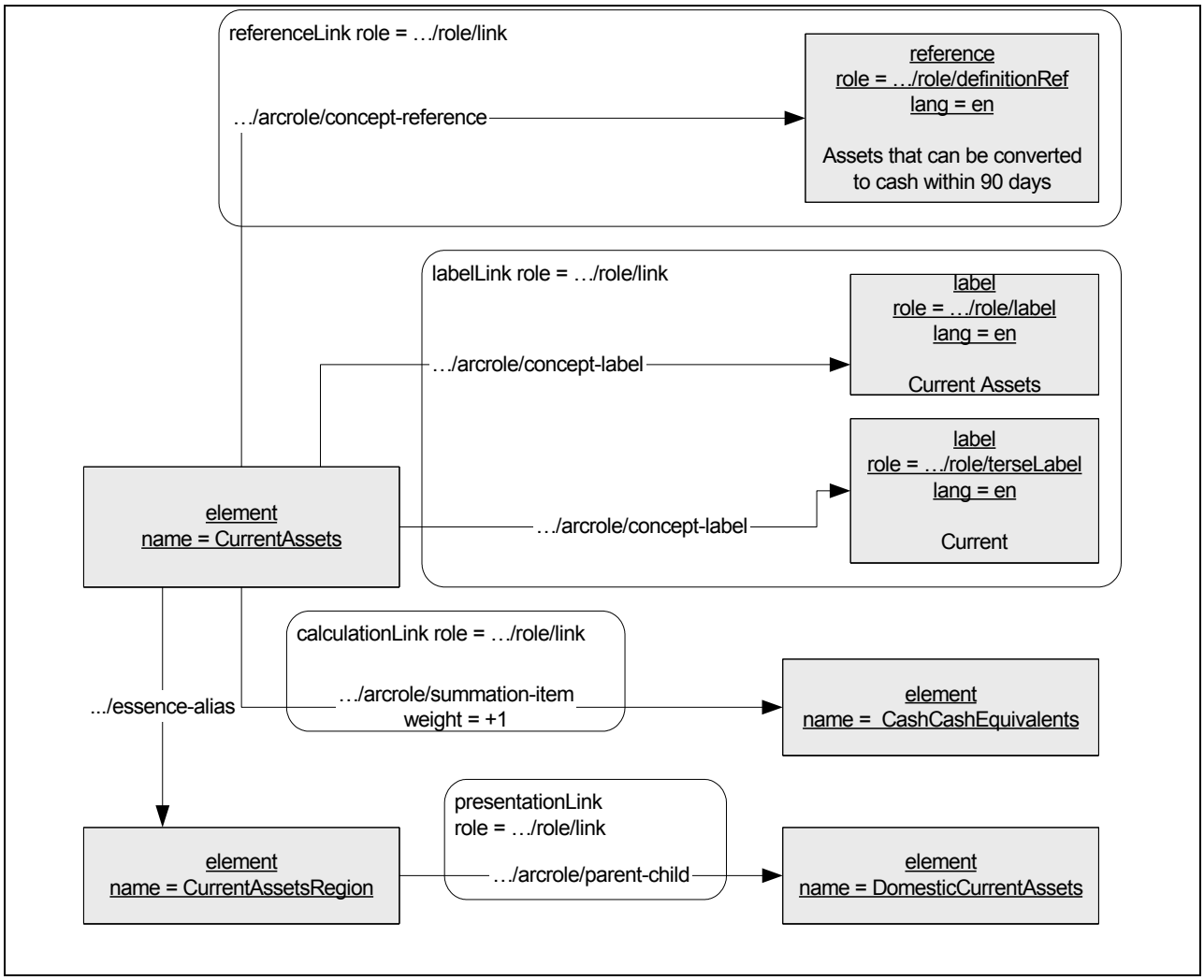
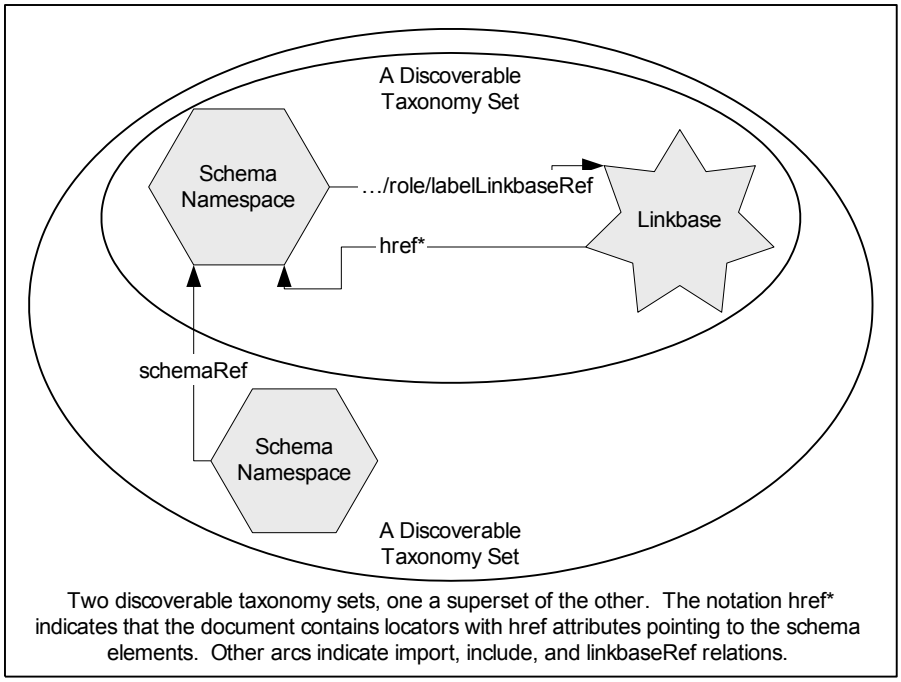


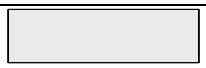
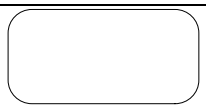
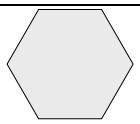
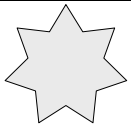
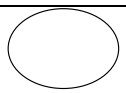
Figure 3. Legend of taxonomy schema and Linkbase drawing conventions.



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.

Table 2. Drawing notations in 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/this	`xlink:role` attribute value "`http://www.xbrl.org/2003/role/this`"
…/arcrole/that	`xlink:arcrole` attribute value "`http://www.xbrl.org/2003/arcrole/that`"

1.1 Rules for all concepts

The rules covering concepts apply to items and to tuples.

Abstract concepts are concepts (elements in the item or tuple substitution group) having their XML Schema `abstract` attribute equal to `true`. Abstract concepts cannot be used in XBRL instances. All rules applying to concepts apply also to abstract concepts unless otherwise indicated.

1.1.1 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 rules governing the "Discoverable Taxonomy Set Layer".

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	Change in Revenue Ratio	The first concept is the amount of revenue over a period of time, and the other is the change in revenue between one period of time to another period of time expressed as a ratio.
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 modeling in the specific situation.

1.1.1.1 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.

1.1.1.2 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.

1.1.1.3 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 modeled as the same concept in an XBRL taxonomy so long as the underlying definition of the concept accommodates both approaches to measurement.

1.1.1.4 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).

1.1.2 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 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.

1.1.3 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.

An additional reason to distinguish between `TelevisionResiduals` and `OilResiduals` in this example is that the distinction is useful should the Oil and Television tuples happen to be siblings in an instance. If both concepts had been represented by the same element (`Residuals`), then it would not be possible to define a calculation for the value of `TotalOilResiduals` as the sum of all `Residuals`. The interaction between calculation arcs and tuples is discussed further in section 2.3.4 below.

1.1.4 Concept names SHOULD adhere to the LC3 convention, where the labels are in non-graphical language.

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 not be underscore (_); this leaves only letters (as defined in XML) as valid first characters.
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. As a consequence of XML element name restrictions, all special characters must be omitted from the element name. Special characters include the following:

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

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

1.1.5 Element declarations for concepts MUST contain an "id" attribute whose value begins with the recommended namespace prefix of the taxonomy, followed by an underscore, followed by the element name.

The `id` attribute is not required (XBRL section 5.1.1) but it simplifies fragment identifiers in the URIs of linkbases. The recommended namespace prefix is defined according to rule 3.3.1 below.

Example 4. Required id attributes

English Label of Concept	Element Name	Namespace Prefix	id attribute
Cash in Bank	CashInBank	us-gaap-ci	us-gaap-ci_CashInBank
Gain (Loss)	AumentoPérdida	es-gaap	es-gaap_ AumentoPérdida
Cash in Bank	银行金钱	cn-csrc-ar	cn-csrc-ar_银行金钱

The resulting id MAY be longer than the 256 characters prescribed for the element name.

1.1.6 A concept MUST NOT prohibit the id attribute inherited from a base type.

All XBRL item types (section 4.3) define an optional id attribute, Concepts based on one of these XBRL types may use a restriction of that base type but MUST NOT prohibit the id attribute in doing so.

1.1.7 All documentation of a concept MUST be contained in XBRL linkbases.

Taxonomy element declarations MUST not use the XML Schema documentation element.

1.1.8 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.

1.1.9 All concepts within a taxonomy schema SHOULD have a unique label for the standard or verbose role in each language used in the DTS whose starting point is that schema.

Uniqueness within the scope of an entire DTS cannot be guaranteed by any single taxonomy author. Although the standard label for a concept need not be unique, for each value of `xml:lang` that appears on label resource elements in the DTS of the schema where a concept is defined, one of the following holds true:

1. No two concepts have the same content for the element containing their standard label (role `http://www.xbrl.org/2003/role/label`); or

2. Every concept has a verbose label (role http://www.xbrl.org/2003/role/verboseLabel) AND no two concepts have the same content for their verbose labels.

In practice, taxonomy authors will need to choose whether to make either the standard labels unique, or if this is not practical, use a complete set of verbose labels for this purpose.

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

The documentation MUST be provided in the following way:

1. label resource with the role `http://www.xbrl.org/2003/role/documentation`; or
2. reference resource with any reference role defined in the specification or in LRR

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.

The substance of the documentation of a concept should include:

- The meaning of the concept and any important distinctions from similar concepts;
- The reason for inclusion in the taxonomy, such as modelling a concept in the accounting literature, common practice within a jurisdiction, or structural elements believed to be necessary for technical reasons.

The substance of the documentation must include other considerations as indicated in rule 1.2.3 below. Also, while this rule allows alternative locations for documentation, text that is fully contained in the label linkbase will generally be more immediately accessible to all taxonomy users than anything linked to indirectly via a reference resource lacking its own URI part.

Exceptions:

The rule may not be applied to concepts that are dimensional in nature. This means that any concept taking part in a relationship having an arc that has been specified in the XBRL Dimensions 1.0 specification.

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

Human users are likely to be presented with a label, rather than the element name. This guidance is a consequence of rule 1.1.4.

1.1.12 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 0; label text SHOULD have meaning *only* to human users.

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

For example, “pro forma” MUST be used consistently rather than sometimes using “proforma” and sometimes “pro forma.” This rule should be interpreted as referring to labels all in a single language (see 3.2.6 below) and refers to root words only, for inflected languages such as German. This rule is advisory, rather than mandatory, for the role attributes listed in Table 3 below, “Roles indicating documentation”, because documentation may legitimately have reason to use variant spellings.

Table 3. Roles indicating documentation

`http://www.xbrl.org/2003/role/documentation`
`http://www.xbrl.org/2003/role/definitionGuidance`
`http://www.xbrl.org/2003/role/disclosureGuidance`
`http://www.xbrl.org/2003/role/presentationGuidance`
`http://www.xbrl.org/2003/role/measurementGuidance`
`http://www.xbrl.org/2003/role/commentaryGuidance`
`http://www.xbrl.org/2003/role/exampleGuidance`

1.1.14 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”. Note that “Treasury Shares, Ending Balance” could not be a standard label but rather is a period end label, so as this example implies, the rule of consistent phrasing applies across different roles.

1.1.15 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	Currency Translations, Net
terse label	F/X Net
verbose label	Foreign Currency Translations, Net Result
positive label	Currency Translations Gain
positive terse label	F/X Gain
positive verbose label	Foreign Currency Translations, Net Gain
negative label	Currency translations, Loss
negative terse label	F/X Loss
negative verbose label	Foreign Currency Translations, Net Loss
zero label	
zero terse label	
zero verbose label	
total label	Total Currency Translations, Net
	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.

1.1.16 A concept MUST NOT have more than one label in a base set for each combination of language and label role in the DTS whose starting point is the schema defining that concept.

The taxonomy author is free to define additional labels to existing concepts defined in the taxonomy schemas that are imported. However, it makes no sense for that author to create more than one label of any given combination of language and role.

Example 6 below shows three labels no two of which may both appear in a single linkbase. The scope of this rule applies only to the DTS discoverable from a taxonomy schema, since the taxonomy author can predict this, but cannot predict what DTS any instance will use.

If the DTS contains more than one label for a label role and language, then there should be a clear indication of which is the effective label, through use of priority and prohibited.

Example 6. Disallowed inconsistencies in labels

Concept	Role	Language	Label
CashInBank	standard label	En	Cash in Bank
CashInBank	standard label	En	Cash in Bank
CashInBank	standard label	En	Cash on Deposit

DTS does contain more than one label for the combination for label role and language. Then there should be clear indication of which is the effective label, by making use of priority and prohibited. Unless the word DTS, implies that an effective DTS is created after processing the priority attribute in the arcs.

1.1.17 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 pages or paragraphs in authoritative printed literature, or both.

Note that a consequence of the requirement that all components MUST be contained in reference parts means that although the XBRL schema does allow mixed content in the reference element (because it is a generic XLink resource-type element), FRTA compliant reference elements MUST NOT contain anything other than elements in the substitution group of part.

Example 7. Disallowed reference element with mixed content.

```
<link:reference
  xlink:type="resource"
  xlink:role="http://www.xbrl.org/2003/role/reference"
  xlink:label="ar_Revaluations_ref"
    xmlns:ref="http://www.xblr.org/2004/ref”>
    See page <ref:Page>27</ref:Page> of the <ref:Name>Revaluations</ref:Name> volume.
</link:reference>
```

1.1.18 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.”

Each part of the literature MUST be referenced in a consistent manner; for example, the same law should always be “IPA” and not sometimes “Investor Protection”, sometimes “Protection Act”, etc.

1.1.19 Reference part element definitions MUST provide a documentation element containing a human readable explanation.

Reference link bases may use reference parts that have been defined in a schema other than an XBRL International published schema wherever the reference part definition is found, a human readable text definition MUST appear within the element definition at the path `annotation/documentation`.

Example 8. A reference part definition.

```
<element name="Article" type="string"

         substitutionGroup="link:part" id="my_linkPart_Article">

  <annotation>

    <documentation>The title of an Article within a Law or other statutory
document.</documentation>

  </annotation>

</element>
```

1.2 Rules for items

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

1.2.1 XML Schema types SHOULD be used to constrain the content of items.

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: Data Types".

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.

1.2.2 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 9. 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.

1.2.3 A numeric item declaration without a balance attribute SHOULD have documentation for the item indicating its expected sign, and where the item represents a change in an underlying concept, increases MUST be represented as a positive number.

When a numeric item declaration has a `balance` attribute, the assignment of the value `credit` or `debit` to that attribute leaves no ambiguity as to the correct sign of any particular fact to be expressed using that item concept.

For other numeric items, such as those that appear in cash flow statements or movement analyses, the sign or polarity of the item in a taxonomy is to some extent an arbitrary choice, since the associated calculation arcs can subsequently be set with either positive or negative values as needed. For taxonomy designers, the sign of the item determines the sign of the weights; that is, when an instance contains a numeric fact, the correct sign of that fact MUST be determinable solely by the definition of the item without regard to the weights of adjacent calculation arcs or other parts of the taxonomy.

However, more than mere documentation is required when the goal is to enhance consistency in taxonomy design, remove ambiguity in instance document creation (particularly when there is a manual process), and enhance comparability among facts of the same item in a taxonomy. The following sub-rules apply:

1. The standard label of a numeric item SHOULD indicate the expected positive and (negative) sign of the fact values it will represent. See Example 10.

Example 10. Standard labels indicating expected positive and (negative) signs

Item	Standard Label
`CashFlowsFromUsedOperatingActivites`	Cash flows from (used in) operating activities
`IncreaseDecreaseTradeCreditors`	Increase (decrease) in Trade Creditors
`IncreaseDecreaseInventory`	Increase (decrease) in Inventory
`IncreaseDecreaseReceivables`	Increase (decrease) in Receivables

2. A fact that describes the “increase” or “upward movement” in value of an underlying item MUST have a positive value. See

3. Example 11.

Example 11. Facts indicating increases and decreases

Item	Value	Meaning
CashFlowsFromUsedOperatingActivities	200	Operations produced 200 cash.
IncreaseDecreaseTradeCreditors	-700	Trade creditors decreased 700.
IncreaseDecreaseInventory	-600	Inventory decreased 600.
IncreaseDecreaseReceivables	500	Receivables increased 500.

Note that the item-summation arc between item declarations in a taxonomy constructed in accordance with both sub-rules MUST have a `weight` attribute, but that attribute could be either -1 or 1. As noted earlier, in designing a taxonomy it is the sign of the item concept that determines arc weights, not the reverse. Illustration of its impact on calculation arc weights is shown in section 2.3 below, "Rules for calculation relationships".

This rule is optional for abstract numeric items.

1.2.4 Numeric items SHOULD NOT be percentages.

The notion of a "percentage" in numeric items of XBRL taxonomies introduces ambiguity, lack of comparability, and potential redundancy. Ratios are universal and unambiguous because there is never a question as to whether a factor of 100 has been applied to a figure such as ".8", which could be read as a 80% change or .8% change. Whether to present that ratio of .8 to a user as "80%" or as ".8" is up to a rendering application.

1.2.5 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.

Example 12. Related concepts measured at instants or over periods.

Concept	Period Type
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"

This is a consequence of specification section 5.1.1.1 [XBRL], which allows only `instant` and `duration` as values of the `periodType` attribute.

1.2.6 Sibling concepts in the content model of a tuple MAY have different values of the periodType attribute.

Tuples may reasonably associate elements that mix different period types.

Example 13. A tuple definition with children of different period types.

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

This is a consequence of specification section 4.9 [XBRL], which places no restrictions on the periodType of tuple children.

1.2.7 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).

Example 14. Numeric concepts requiring an instant period type.

Concept	Period Type
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.

1.2.8 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 a restated 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 restated values MUST be identified in instances using the `scenario` element.

1.3 Rules for tuples

Tuples are used to bind together, or associate, one or more items. Together, these concepts form a compound or complex concept. 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).

1.3.1 Tuples MUST be used to associate concepts that derive their meaning from each other.

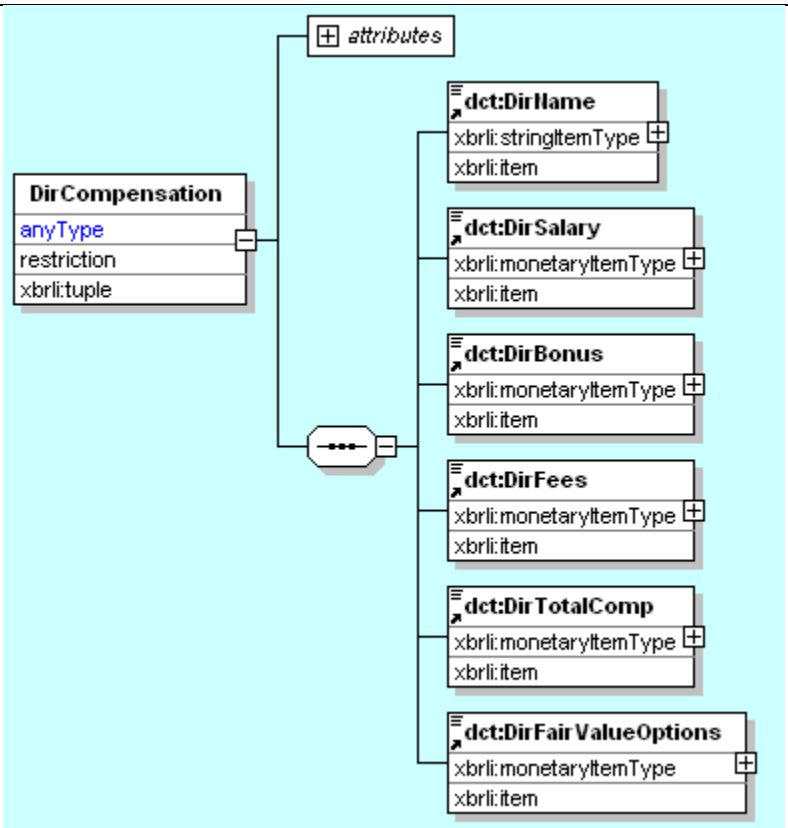
Tuples need to be used wherever it is necessary to convey a number of concepts that cannot be understood without being grouped together. For example, it would be common to list directors' names, salaries and options. To be understood, the entries need to be grouped together. Compare: there was a

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 15 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 15. A table in a financial statement modelled using a tuple.

Name of director	Salary	Bonus	Director fees	Total compensation	Fair Value of Options Granted
Horace Chang	0	0	60,000	60,000	0
Gerry Ferguson	879,639	1,213,486	0	2,093,125	569,000
Sally James	0	0	24,200	24,200	0
Ivan Chenokitov	0	0	57,000	57,000	0




```
<schema xmlns:link="http://www.xbrl.org/2003/linkbase"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2003/instance"
xmlns:dct="http://www.xbrl.org/int/fr/frta/dct/2005-04-04"
targetNamespace="http://www.xbrl.org/int/fr/frta/dct/2005-04-04"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <annotation>
    <appinfo>
      <link:linkbaseRef xlink:type="simple"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" xlink:href="dct-
2005-04-04-presentation.xml"
xlink:role="http://www.xbrl.org/2003/role/presentationLinkbaseRef"/>
      <link:linkbaseRef xlink:type="simple"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" xlink:href="dct-
2005-04-04-label-en.xml"
xlink:role="http://www.xbrl.org/2003/role/labelLinkbaseRef"/>
    </appinfo>
  </annotation>
  <import namespace="http://www.xbrl.org/2003/instance"
schemaLocation="http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd"/>
  <element name="DirCompensation" substitutionGroup="xbrli:tuple"
id="dct_DirCompensation">
    <complexType>
      <complexContent>
        <restriction base="anyType">
          <sequence>
            <element ref="dct:DirName"/>
            <element ref="dct:DirSalary"/>
            <element ref="dct:DirBonus"/>
```

```
            <element ref="dct:DirFees"/>
            <element ref="dct:DirTotalComp"/>
            <element ref="dct:DirFairValueOptions"/>
          </sequence>
          <attribute name="id" type="ID" use="optional"/>
        </restriction>
      </complexContent>
    </complexType>
  </element>
  <element name="DirName" type="xbrli:stringItemType" substitutionGroup="xbrli:item"
nillable="true" id="dct_DirName" xbrli:periodType="duration"/>
  <element name="DirSalary" type="xbrli:monetaryItemType"
substitutionGroup="xbrli:item" nillable="true" id="dct_DirSalary"
xbrli:periodType="duration"/>
  <element name="DirBonus" type="xbrli:monetaryItemType"
substitutionGroup="xbrli:item" nillable="true" id="dct_DirBonus"
xbrli:periodType="duration"/>
  <element name="DirFees" type="xbrli:monetaryItemType"
substitutionGroup="xbrli:item" nillable="true" id="dct_DirFees"
xbrli:periodType="duration"/>
  <element name="DirTotalComp" type="xbrli:monetaryItemType"
substitutionGroup="xbrli:item" nillable="true" id="dct_DirTotalComp"
xbrli:periodType="duration"/>
  <element name="DirFairValueOptions" type="xbrli:monetaryItemType"
substitutionGroup="xbrli:item" nillable="true" id="dct_DirFairValueOptions"
xbrli:periodType="instant"/>
</schema>
```

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

The first row of the compensation table shown above in Example 15 appears in an XBRL instance as shown in Example 16. 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 15, within the opening and closing tags of the tuple.

Example 16. XBRL Instance data containing the first row of a table.

```
<xbrl xmlns="http://www.xbrl.org/2003/instance" xmlns:link="http://www.xbrl.org/2003/linkbase" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:iso4217="http://www.xbrl.org/2003/iso4217" xmlns:dct="http://www.xbrl.org/int/fr/frta/dct/2005-04-04">
 <link:schemaRef xlink:type="simple" xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" xlink:href="../../dct/2005-04-04/dct-2005-04-04.xsd"/>
 <context id="sadv05p">
  <entity>
   <identifier scheme="dns:nic">www.StandardAdvantage.com</identifier>
  </entity>
  <period>
   <startDate>2005-01-01</startDate>
   <endDate>2005-12-31</endDate>
  </period>
 </context>
 <context id="sadv05e">
  <entity>
   <identifier scheme="dns:nic">www.StandardAdvantage.com</identifier>
  </entity>
  <period>
```

```
  <instant>2005-12-31</instant>
 </period>
</context>
<unit id="usd">
 <measure>iso4217:USD</measure>
</unit>
<dct:DirCompensation>
 <dct:DirName contextRef="sadv05p">Horace Chang</dct:DirName>
 <dct:DirSalary decimals="0" contextRef="sadv05p" unitRef="usd">0</dct:DirSalary>
 <dct:DirBonus decimals="0" contextRef="sadv05p" unitRef="usd">0</dct:DirBonus>
 <dct:DirFees decimals="0" contextRef="sadv05p" unitRef="usd">60000</dct:DirFees>
 <dct:DirTotalComp decimals="0" contextRef="sadv05p" unitRef="usd">60000</dct:DirTotalComp>
 <dct:DirFairValueOptions decimals="0" contextRef="sadv05e" unitRef="usd">0</dct:DirFairValueOptions>
</dct:DirCompensation>
</xbrl>
```

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.

As discussed earlier in 0 above, XBRL items and tuples are global, so an item such as `DirName` appearing *outside* of the tuple `DirCompensation` will inevitably be XBRL-valid even if the intent of the taxonomy author may have been to limit its use to being *inside* of it. Example 17 shows valid documentation along with an instance that violates that documentation.

Example 17. Disallowed use of a concept that must appear in a tuple.

```
<linkbase xmlns="http://www.xbrl.org/2003/linkbase" xmlns:xlink="http://www.w3.org/1999/xlink">
 <labelLink xlink:type="extended" xlink:role="http://www.xbrl.org/2003/role/link">
  <loc xlink:type="locator" xlink:href="../../dct/2005-04-04/dct-2005-04-04.xsd#dct_DirName" xlink:label="label_DirName_0"/>
  <labelArc xlink:type="arc" xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label" xlink:from="label_DirName_0"
xlink:to="label_DirName_1"/>
  <label xlink:type="resource" xlink:label="label_DirName_1" xlink:role="http://www.xbrl.org/2003/role/documentation"
xml:lang="en">This item must not appear at top level.</label>
 </labelLink>
</linkbase>
```

```
<xbrl xmlns="http://www.xbrl.org/2003/instance" xmlns:link="http://www.xbrl.org/2003/linkbase"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:iso4217="http://www.xbrl.org/2003/iso4217"
xmlns:dct="http://www.xbrl.org/int/fr/frta/dct/2005-04-04">
 <link:schemaRef xlink:type="simple" xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase" xlink:href="ex26-
2005-04-04.xsd"/>
 <context id="sadv05p">
  <entity>
   <identifier scheme="dns:nic">www.StandardAdvantage.com</identifier>
  </entity>
  <period>
   <startDate>2005-01-01</startDate>
   <endDate>2005-12-31</endDate>
  </period>
 </context>
 <dct:DirName contextRef="sadv05p">Horace Chang</dct:DirName>
</xbrl>
```

1.3.2 When instances may contain multiple values of the same item within the same context and having the same units, 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.

Although XBRL does not forbid instances to have facts that are both c-equal and u-equal, it is undesirable. The purpose of this rule is to prevent situations in which the *absence* of a tuple would force the instance author to create duplicate facts.

1.3.3 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 items "Street1", "Street2", and "Street3" does *not* violate this rule.

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

A "segment" means a line of business, geographical region, or other partitioning of an entity (see XBRL Specification 2.1 section 4.7.3.2 [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.

In general, 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.

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.

1.3.5 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.

1.3.6 Tuple content models MUST NOT use the "all" compositor.

The meaning of the content of an instance of a financial reporting taxonomy does not depend on the order in which the facts are expressed in the instance; the ordering is therefore arbitrary within tuples. Since the order does not matter, the taxonomy author loses no flexibility but processing software can be

somewhat simplified if, wherever the `all` compositor would be used, the `sequence` compositor is used instead.

1.3.7 Tuple content models MUST include an optional local attribute with name 'id' and type ID.

Tuples occurring in instances must be allowed to have an `id` attribute, to allow footnotes to tuples. See Example 15, which contains the defining element:

```
<attribute name="id" type="ID" use="optional" />
```

In that example the `DirCompensation` tuple in the instance has also been augmented with a value for the `id` attribute.

2 Relationships Layer

The relationship layer of the architecture describes how concepts (both items and tuples) and resource-type elements MAY be related to one another. The relationship layer also describes how these relationships SHOULD be modelled.

A relationship exists between a source concept and target concept (or resource) when there is an arc from the source to the target. A single XLink arc can create multiple relationships. This can occur when the values of the "from" and "to" attributes appear as XLink labels on more than one locator-type element in the same linkbase.

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. All relationships are captured in extended-type links. The extended-type links together make up linkbases.

When the scope of a rule about a taxonomy schema (or linkbase) is stated as "within a DTS" without further qualification, this should be understood to mean "the DTS whose starting point is the taxonomy schema (or linkbase) to which the rule is being applied." Rules 1.1.16 above and 2.1.5 below are examples of rules in which this scoping has been made explicit.

Section 3.5.3.9.7.3 [XBRL] deines a link base set, and rules governing relationships usually have a scope that only applies to relationships within the same base set.

2.1 Rules for all relationships

XBRL is an evolving set of standards and the set is always based on a particular version of the XBRL specification, currently 2.1. Additional members of this set of standards may include modules that are XBRL Recommendations and roles and arc roles which are approved and available in a link and role registry (LRR) hosted by XBRL International. Any specification, module, or role will be recommended or approved only when it has well established semantics.

2.1.1 An arc MUST have only its standard or LRR approved arc role.

A FRTA-compliant DTS MUST NOT use any arc roles except those documented in the XBRL Specification or approved in the LRR. 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.

The scope of this rule applies to linkbases defined in the XBRL 2.1 Specification.

2.1.2 The label and reference elements MUST have only their standard or LRR approved resource roles.

The set of label and reference roles defined in Sections 5.2.2.2 (Table 8) and 5.2.3.2.1 (Table 9) of the XBRL 2.1 Specification, and any label and reference roles defined in the LRR, are all that are allowed in `labelLink` and `referenceLink` elements.

2.1.3 The Taxonomy MUST not create new label roles which are similar to the standard or LRR (Link Role Registry) approved resource roles

The standard set of label roles are provided in the XBRL Specification 2.1, the LRR approved resource roles are available in the following link:

http://www.xbrl.org/lrr/lrr.xsd

The taxonomy authors MUST not create resource roles that are exactly the same or imply the same meaning as any of the roles already defined in the XBRL specification 2.1 or the LRR.

2.1.4 An extended-type link role MUST have no processing semantics other than specified by XBRL.

The only processing semantics that XBRL gives the `xlink:role` attribute on extended-type links is that the values partition the sets of arcs in a DTS into distinct sets called link base sets. This is the *only* semantics allowed for the `xlink:role`.

2.1.5 A schema MUST NOT define a role type that duplicates a definition in the DTS whose starting point is the schema defining that role type.

An equivalent formulation of this rule is that a schema-rooted DTS MUST NOT contain s-equal role types. Although a FRTA compliant taxonomy is constrained to use roles, the definitions of those roles MAY occur in various locations; this rule ensures that only one definition is used within a given DTS, because a taxonomy author can control this but not control the DTS of any instances. This rule also implies that the authoritative location of the role definition SHOULD be used.

2.1.6 Roles and arc roles from XBRL, XBRL modules, and the LRR SHOULD be used in preference to defining new roles.

This is a logical consequence of the fact that each of these sources has the status of an XBRL International recommendation.

2.1.7 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.

This is a consequence of specification section 3.5.3.3 [XBRL], which 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/link.`

2.1.8 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.

This is a consequence of specification sections 3.5.3.3 and 5.2 [XBRL], which define, respectively, the syntax and semantics of the extended-type link `role` attribute.

2.1.9 Any role type definition for an extended-type link in a persisting DTS SHOULD have a human-readable explanation in its definition element.

In addition to being good practice to document newly defined roles, the purpose of this rule is to ensure the availability of a human-readable "label" to appear in taxonomy tools. Users see "Balance Sheet, Order of Liquidity Format" rather than "`http://www.xbrl.org/2003/role/BalanceSheetLiquidity`".

Example 18. Role type definition with explanation.

```
<link:roleType id="BalanceSheetLiquidity"


roleURI="http://xbrl.iasb.org/int/fr/ifrs/gp/role/BalanceSheetLiquidity">

  <link:definition>Balance Sheet, Order of Liquidity Format</link:definition>

  <link:usedOn>link:presentationLink</link:usedOn>

  <link:usedOn>link:calculationLink</link:usedOn>

</link:roleType>
```

This is a role meant to identify a presentation link that contains arcs in which presentation siblings in a balance sheet are ordered by increasing liquidity.

```
<link:roleType id="endNote"

               roleURI="http://www.xbrl.org/int/fr/endNote">

  <link:definition>Indicates a note intended only to be rendered for presentation
at the end of a document.</link:definition>

  <link:usedOn>link:footnoteLink</link:usedOn>

</link:roleType>
```

This is a role meant to identify a footnote link containing notes intended only to be presented at the end of a document.

Additional description of the processing semantics SHOULD be provided in documentation.

2.1.10 The role URI in a roleType element SHOULD be an LRR approved role or begin with the same scheme and authority parts as the target namespace of the taxonomy schema where it appears.

This limits the potential for accidental merging of independently created networks of relationships. Only the scheme and authority [RFC2396] must be the same, not the entire path. When the URI is a URN [RFC2141]**Error! Reference source not found.**, this rule is interpreted to mean that the NID must be the same.

http://www.ffiec.gov/2003/xbrl/form031
This is a role URI meant to identify extended-type links relating to a particular regulatory form used by the government agency "FFIEC".
`urn:xbrl:taxonomy:gcd:2002-10-15`
This is a hypothetical URN identifying a dated version of a taxonomy published by XBRL International, with an NID of `xbrl`.

2.1.11 All arcs whose source and target both refer to concepts MUST specify an `order` attribute.

This rule universally applies to all arcs in all extended-type links in the calculation, definition and presentation linkbases, and applies to arcs with any arc role, whether standard or custom. This rule ensures that linkbases in taxonomies published conforming to FRTA have a common way of being presented in different tools. It is also meant to apply to any future XBRL modules that introduce new linkbases connecting concepts with each other; it does *not* apply to the label, reference (or footnote) linkbases. 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.

This rule applies to only Presentation, Calculation and Definition linkbases.

2.1.12 Two relationships defined by arcs in the same base set with the "use" attribute having the value "optional", having concepts as targets and sharing the same "from" concept SHOULD 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 prohibiting any arcs that introduce ambiguous ordering. This rule does not apply to relationships with the `use` attribute value of `prohibited`; it also does not apply to relationships between concepts and resource-type elements.

Note that this rule applies to relationships, not to arcs. Therefore, an arc with a "to" attribute value that is the XLink label of more that one concept would necessarily violate this rule since its 'order' attribute would then apply to siblings.

2.2 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.

For instances, it is the end user of that instance and their preferences as to level of detail, scope of the data, verbosity, language, currency, rounding, etc. that control how an application renders instance data. The presentation and label linkbases cannot, and SHOULD not control all of these aspects, but SHOULD provide useful data to rendering applications – for example, applications that render the taxonomy for its developers, reviewers and users.

That said it is nevertheless the case 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.

2.2.1 A concept meant to be ordered among its siblings MUST have a parent-child presentation relationship 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 arc ordering except as indicated in rule 2.2.4 below.

2.2.2 Presentation parent-child relationships having the same parent and child in extended links with the same role SHOULD provide preferred labels.

Although no pair of arcs in the same extended-type link can have the same "from" and "to" attributes [XLINK], it is still possible for separate extended-type links to have otherwise equivalent arcs. XBRL does allow undirected cycles in parent-child arc sets. But in addition to distinct values for the "order" attribute as suggested by 2.1.12 above, parent-child presentation arcs SHOULD indicate using the `preferredLabel` attribute which label an XBRL processor should use for the child concept depending on which parent concept it is being presented as a child of.

2.2.3 A DTS SHOULD provide parent-child presentation relationships intended for users of the taxonomy.

The base sets consisting of parent-child arcs in a DTS, taken in union, should provide enough information to display all the concepts that the DTS authors intend to be used in instances to be validated by that DTS. This does *not* mean that the base set has to provide all of the information needed to replicate or reconstruct printed financial statements or other standard displays. It also does not mean that presentation link bases must include all concepts in the DTS. If the standard role attribute value `http://www.xbrl.org/2003/role/link` is not used, then the documentation SHOULD specify the base sets (roles) whose union provides the intended coverage.

2.2.4 The DTS rooted at the schema where a tuple is defined SHOULD contain at least one tree of presentation parent-child relationships in which every concept that can appear as a descendant of the tuple in an instance appears as a descendant of the tuple in that presentation tree, and there SHOULD NOT exist any tree of presentation parent-child relationships in which a non-abstract concept that cannot appear as a descendant of the tuple in an instance appears as a descendant of the tuple in that presentation tree.

Tuple concepts MAY appear in presentation hierarchies and so all elements that could appear as descendants of a particular tuple in an instance SHOULD appear as descendants of that tuple in at least one such presentation hierarchy.

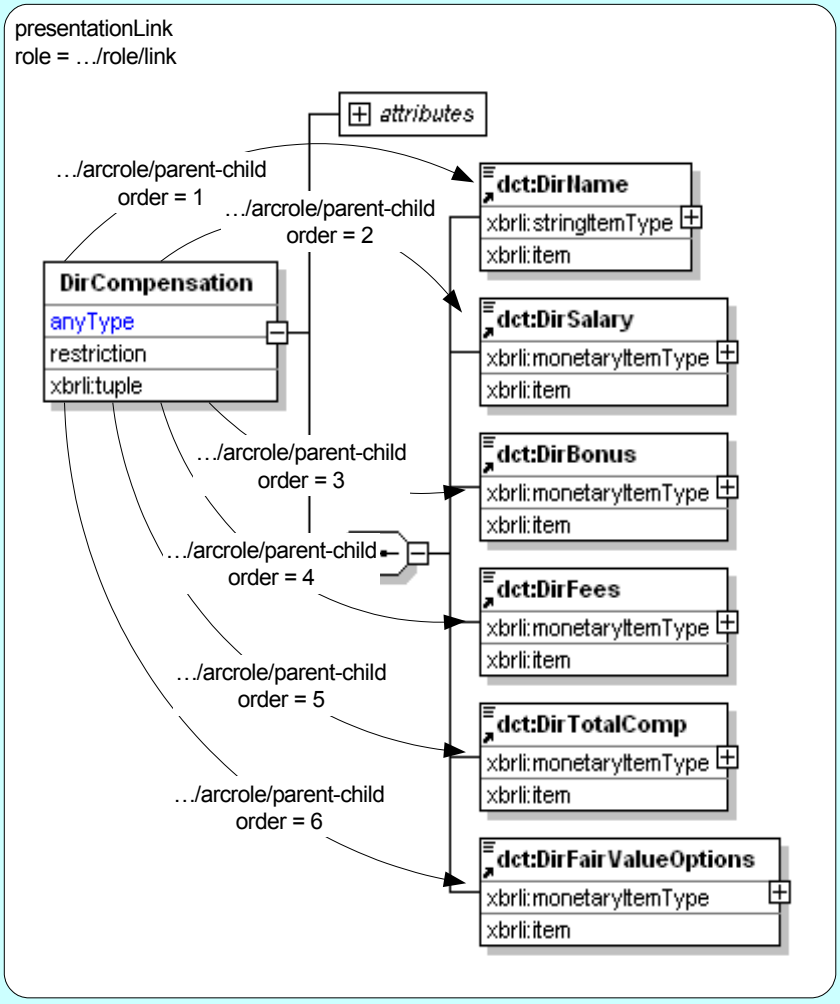
Other elements that do not appear as descendants anywhere in its content model SHOULD NOT appear as descendants anywhere in any of its presentation sub-trees.

Note that for this purpose, an element reference in a tuple content model with `maxOccurs="0"` is considered to be an element that "does not appear".

The `order` attribute is not constrained by the content model.

Example 19 shows presentation arcs added to Example 15 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`, and `choice`.

Example 19. Presentation parent-child arcs in a tuple.



Presentation arcs connecting a tuple to its sub-elements

For simplicity, this rule does not apply to concepts that might appear as descendants of tuples either

- as a result of being in the substitution group of an element in a the content model of a tuple, or
- as a result of matching a wildcard in the content model of a tuple.

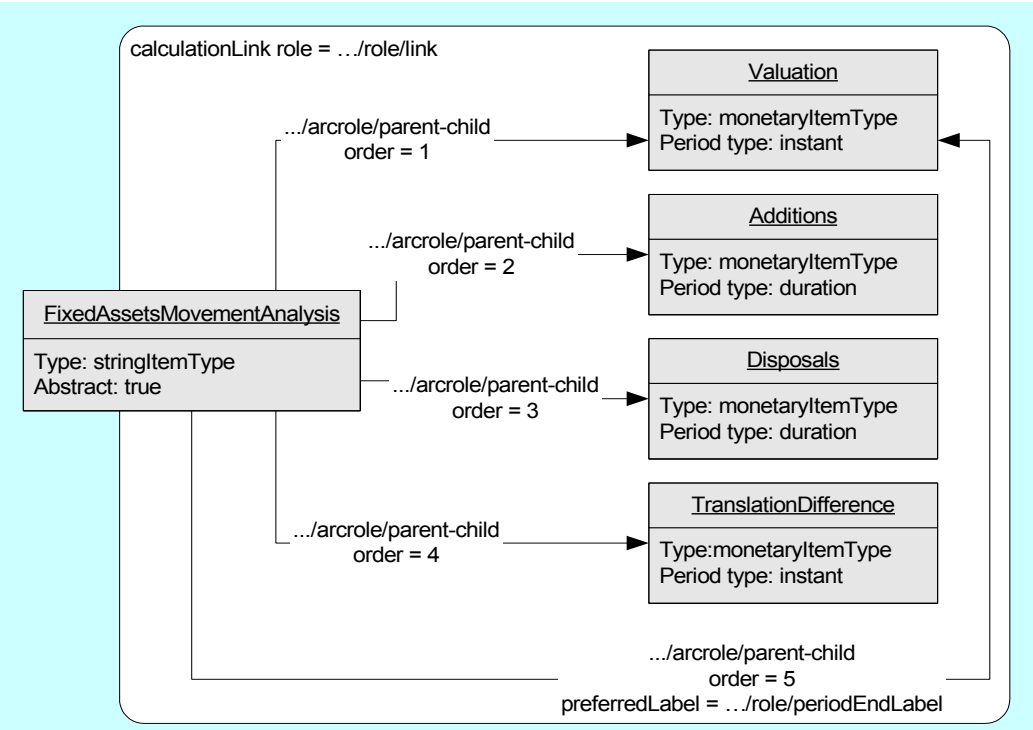
This exception means that strict application of the rule does not, therefore, achieve the complete intent of the rule in every possible circumstance where a taxonomy author has used complex features of XML schema. However, such complex uses are not expected to be commonplace, nor are they considered desirable.

2.2.5 The parent-child relationships 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 1.2.8, "The beginning balance, the ending balance, and any adjusted balances of an item for a period MUST be represented as a single item." Example 20 shows a movement analysis for fixed assets, showing reconciling items along the top, and a list of assets down the side.

Example 20. 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

2.3 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, extension taxonomy could express the new formula

A = B – C + D

With an additional `summation-item` relationship:

From A to D, weight 1.0.

The application of other rules may impact or constrain the way in which calculation arcs are used and their weights set. In particular, rule 1.2.3 states that "A numeric item declaration without a balance attribute SHOULD have documentation for the item indicating its expected sign, and where the item represents a change in an underlying concept, increases MUST be represented as a positive number." Once the "sign" of a numeric item has been selected in a taxonomy, the weights of the calculation arcs which connect that item to other items can be assigned.

Example 21 shows a fragment of a taxonomy where all but three of the calculation summation-item arcs have weight=1. The items in the example correspond to the items in the examples of rule 1.2.3, Example 10 and

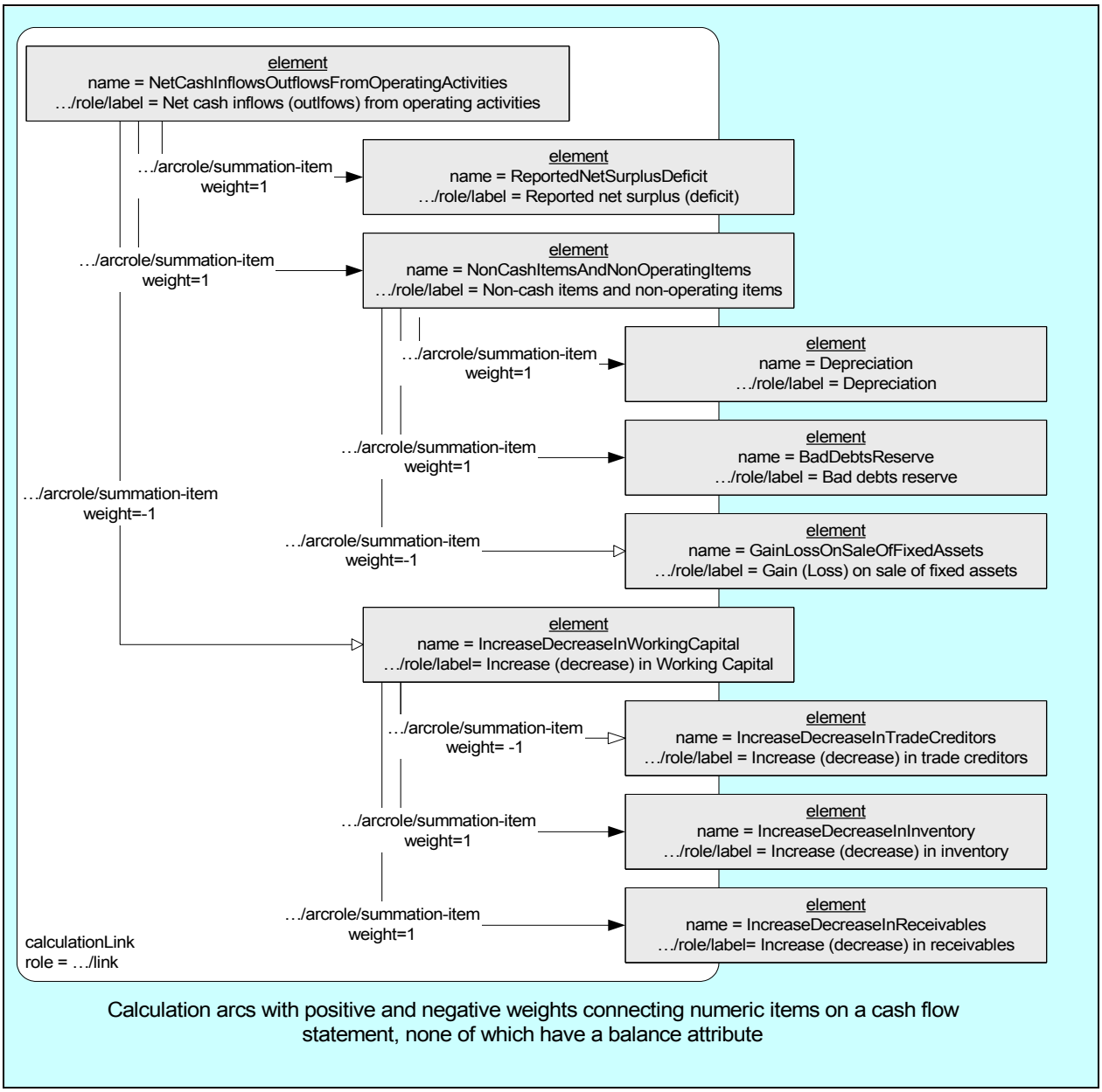
Example 11.

Example 22 shows a set of facts from a sample instance, along with an indication of the corresponding weight of the arc from that item to its parent in

Example 21.

23 then shows three different presentations of the same instance data (and implicitly, presenting the calculation or derivation of the data). In that example, it is assumed that some positive and negative terse and verbose labels (see Example 5) have been provided in the taxonomy.

Example 21. Calculation arcs in a cash flow statement



Calculation arcs with positive and negative weights connecting numeric items on a cash flow statement, none of which have a balance attribute

Example 22. Fact values of a cash flow statement in an instance

Standard Label	Fact	Weight of arc	Calculation
Net cash inflows (outflows) from operating activities	-440		+100+60-600
Reported net surplus (deficit)	100	+1	
Non-cash items and non-operating items	60	+1	+50+20-10
Depreciation	50	+1	

Standard Label	Fact	Weight of arc	Calculation
Bad debts reserve	20	+1	
Gain (loss) on sale of fixed assets	10	-1	
Increase (decrease) in working capital	600	-1	-(-700)+(-600)+500
Increase (decrease) in trade creditors	-700	-1	
Increase (decrease) in inventory	-600	+1	
Increase (decrease) in receivables	500	+1	

Example 23. Three alternative presentations of a single set of cash flow facts

Calculation relationships indicated by displayed values only:

Reported net surplus	100
Add (less) non-cash items and non-operating items:	
Depreciation	50
Bad debts reserve	20
(Gain) on sale of fixed assets	(10)
Movements in working capital:	
Change in trade creditors	(700)
Change in inventory	600
Change in receivables	(500)
Net cash flows from operating activities	**(440)**

Calculation relationships indicated by labels only:

Reported net surplus	100
non-cash items and non-operating items:	
Add:	
Depreciation	50
Bad debts reserve	20
Less:	
Gain on sale of fixed assets	10
Movement in working capital:	
Add:	
Decrease in inventory	600
Less:	
Decrease in trade creditors	700
Increase in receivables	500
Net cash outflows from operating activities	**440**

Calculation relationships indicated by a combination of label & displayed values:

Reported net surplus (deficit)	100
Add (less) non-cash items and non-operating	

	items:	
	Depreciation	50
	Bad debts reserve	20
	Loss (Gain) on sale of fixed assets	(10)
	Movements in working capital:	
	Increase (decrease) in trade creditors	(700)
	(Increase) decrease in inventory	600
	(Increase) decrease in receivables	(500)
	Net cash inflows (decrease) from operating activities	**(440)**

The examples above reinforce the point that calculation and presentation arcs do not necessarily correspond, and that the presentation of a particular fact value as positive (negative) could even depend on the sign of its parent and other factors.

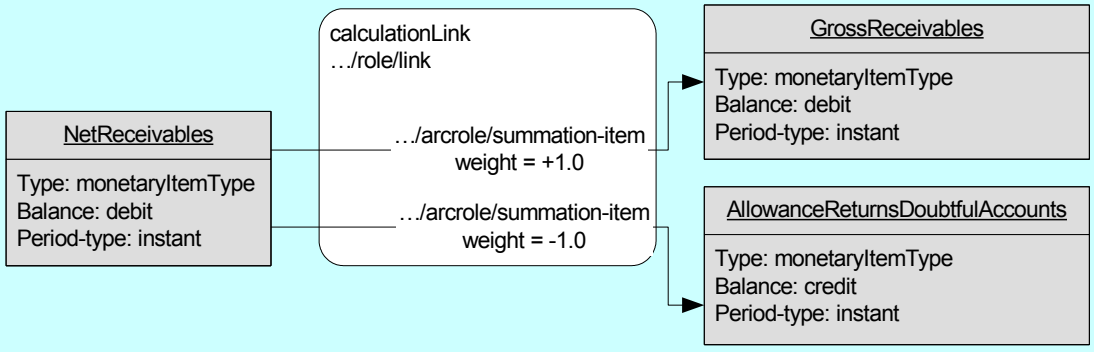
2.3.1 All concepts in a DTS which have an additive relationship in all equal contexts SHOULD have calculation relationships in that DTS.

Taxonomy authors SHOULD 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 24. A Net and Gross relationship

Accounts receivable, net of allowances, consists of the following as of the balance sheet date:	2001 €'000	2002 €'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 SHOULD be defined relating the gross, net and adjustment total concepts.

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

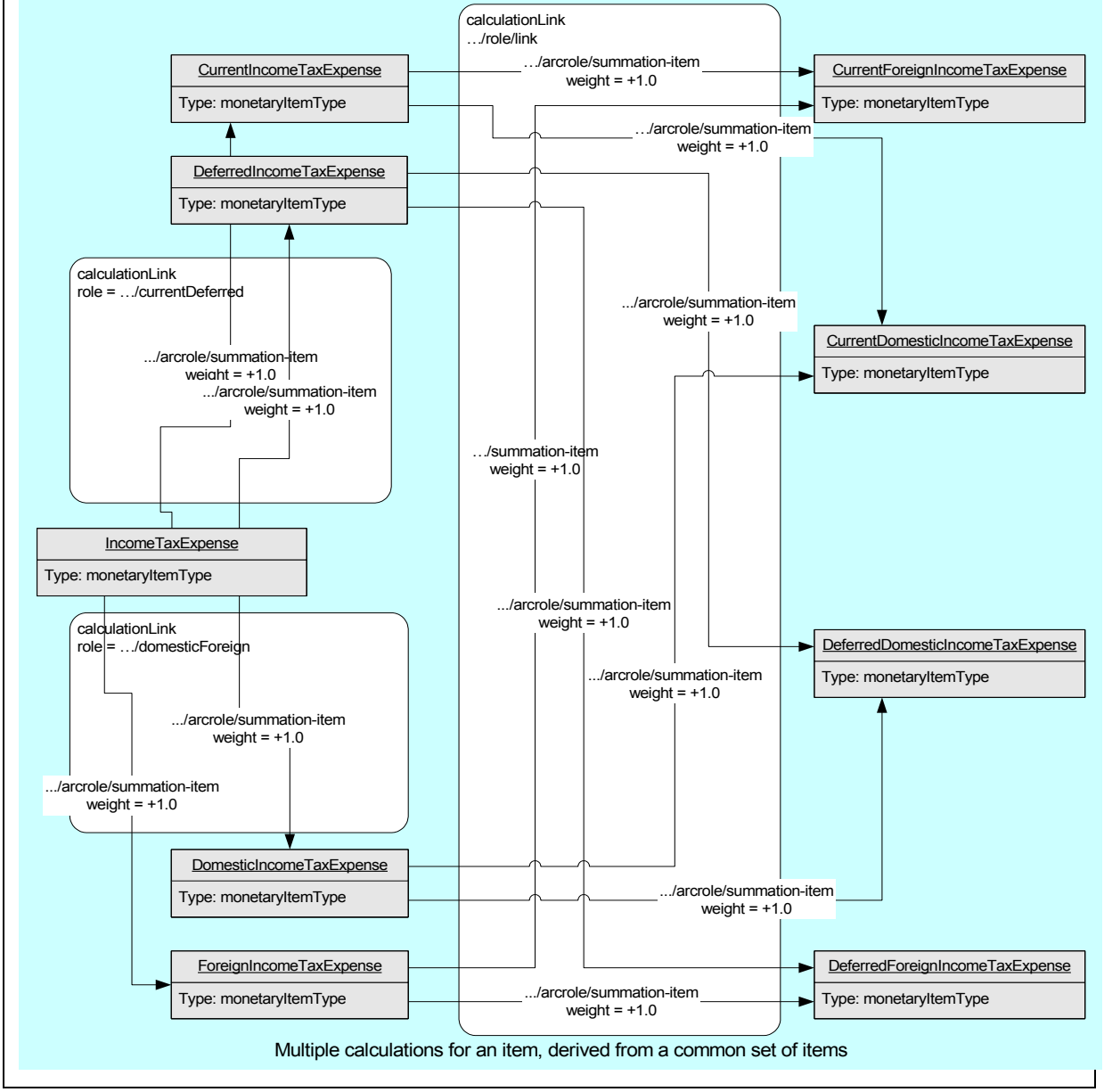
Double counting would result if two alternative ways of calculating an amount were to appear both in extended-type links with the same role. For example, total income tax expense might be calculated either by summing foreign and domestic taxes, summing current and deferred, or both. These calculations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

In Example 25, 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 2.1.9 and 2.1.10 above these would be based on some other namespace).

The `summation-item` arcs in Example 25 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 25. Two distinct summations in a financial report

The following is a summary of income tax expense:	2001	2002
	$'000	$'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	2002
	$'000	$'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



Specification section 5.2 [XBRL]**Error! Reference source not found.** details how the semantics embodied in extended link arcs is contingent on extended link arc role values, and forces independence on calculations in different base sets.

Compliance with this rule cannot be established in an entirely automated fashion because it is impossible to reliably determine the taxonomy authors' intent.

2.3.3 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 25, all of the items and relevant calculation arcs SHOULD be defined for cases where such alternatives are permitted.

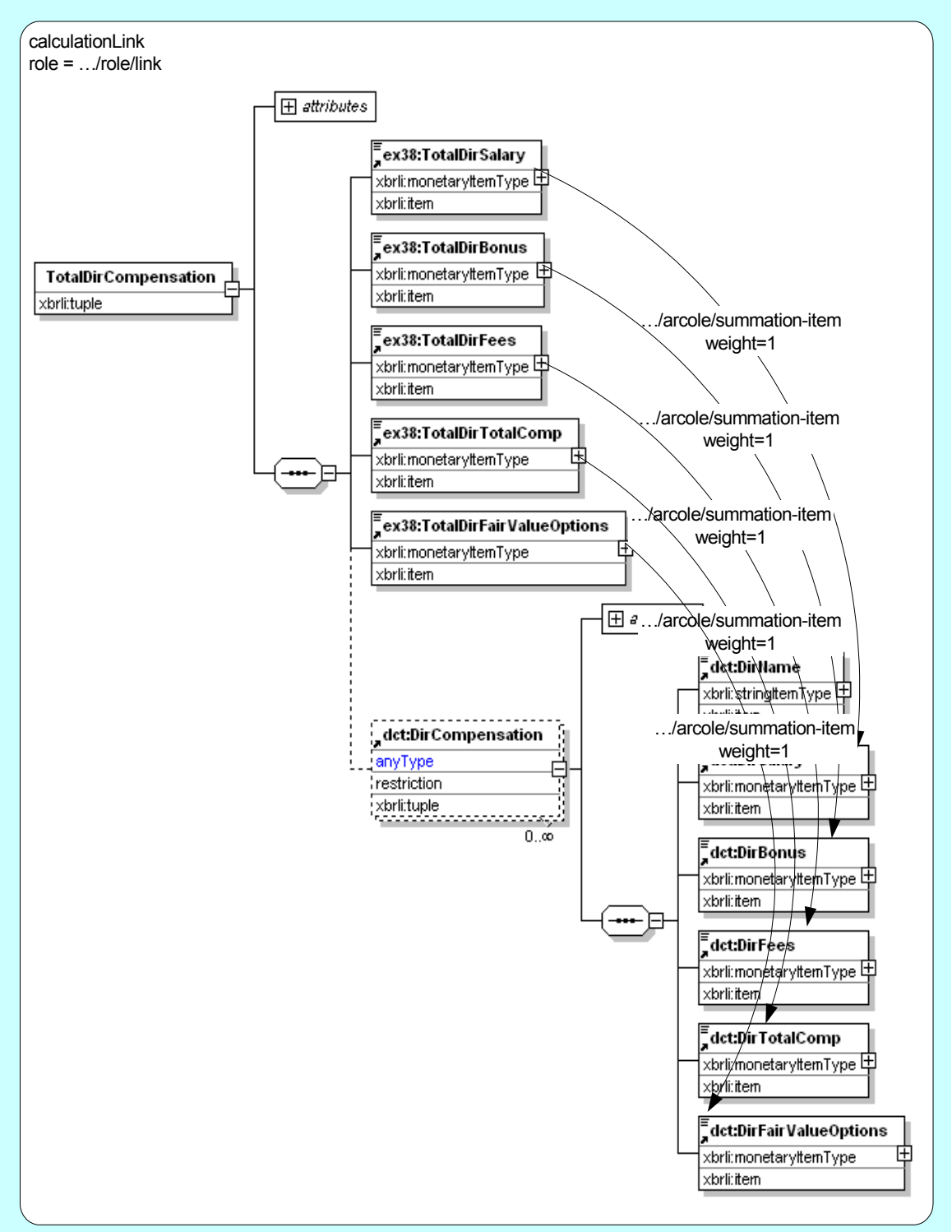
Multiple calculation hierarchies, summing a single set of concepts in multiple ways, occur in many guises in financial reporting. For example, in a classified balance sheet, assets and liabilities are totalled separately into current and non-current categories; while an unclassified balance sheet does 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.

2.3.4 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 26 is similar to Example 15 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 26. Table containing a summation across tuples.

Name of director	Salary	Bonus	Director fees	Total Salary, Bonus, and Director fees	Fair Value of Options Granted
Horace Chang	0	0	60,000	60,000	**0**
Gerry Ferguson	879,639	1,213,486	0	2,093,125	**569,0000**
Sally James	0	0	24,200	24,200	**0**
Ivan Chenokitov	0	0	57,000	57,000	**0**
Total	879,639	1,213,486	141,200	2,234,325	**569,000**



Calculation arcs connecting totals to items in tuples.

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.

2.3.5 The declarations of the source and target concepts of a summation-item relationship MUST have identical values of the periodType attribute.

For example, there MUST NOT be a calculation relationship between the items in Example 27, because the period types are different and therefore the items are in different contexts.

Example 27. Calculation links cannot cross period types

Item	Label Role	Item Label	periodType	Value
ChangeInCash	Standard	Change in Cash	Duration	-10
Cash	Period end	Cash, ending balance	Instant	90

2.3.6 The source and target concepts of a summation-item relationship MUST be distinct.

Summation-item arcs MUST NOT be used to describe relationships *if* the starting and ending balances are represented by the same numeric item but distinguished by different periods. Although XBRL section 5.2.5.2 allows all types of cycles in sets of summation-item arcs, this rule forbids the particular case of a cycle of a single item. There must not be any calculation relationships between the items in Example 28, because the two Cash items are the same concept.

Example 28. Calculation relationships cannot cross periods

Item	Label Role	Item Label	periodType	Value
Cash	Period start	Cash, beginning balance	instant	100
ChangeInCash	Standard	Change in Cash	duration	-10
Cash	Period end	Cash, ending balance	instant	90

Taken together, rules 2.3.5 and 2.3.6 mean that calculation relationships cannot associate the beginning balance, adjusted balance and ending balance in a movement analysis (see rule 1.2.8, "The beginning balance, the ending balance, and any adjusted balances of an item for a period MUST be represented as a single item."). Only the *presentation* of movement analyses can be represented using XBRL 2.1.

2.4 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 relationships 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 another. Definition relationships 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.

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

General-special relationships 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 specialisation of "contra-asset". The `general-special` relationship 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` relationships.

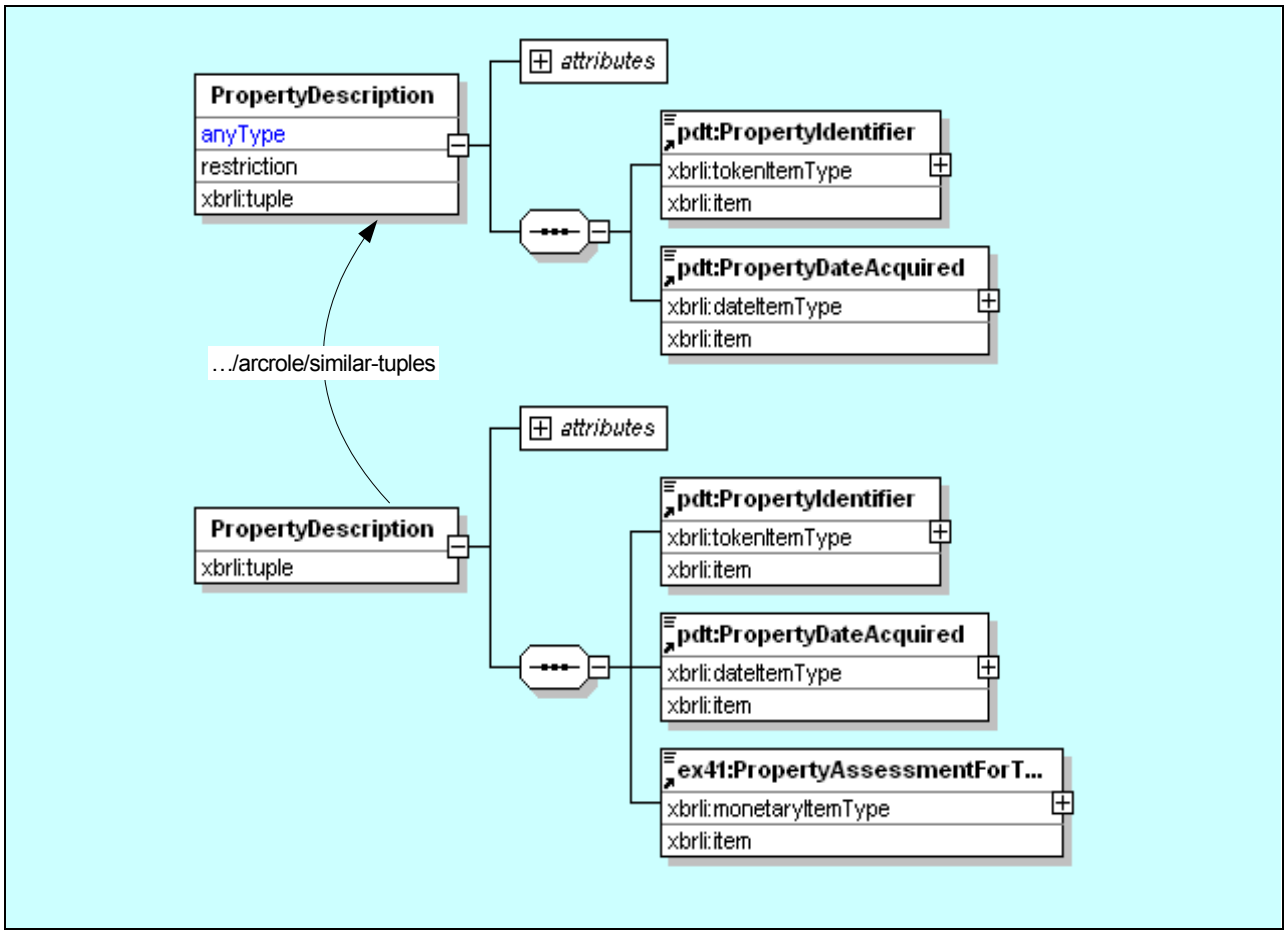
2.4.2 A tuple having the same reporting purpose as a tuple in a different taxonomy within the same DTS SHOULD have a similar-tuples relationship to that other tuple.

Extension taxonomies are meant to use `similar-tuple` definition relationships 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 29 shows two tuples:

- `pdt:PropertyDescription` having a content model of only two items `pdt:PropertyIdentifier` and `pdt:PropertyDateAcquired`, and below it,
- `ex41:PropertyDescription` having the same two items followed by a third item, `ex41:PropertyAssessmentForTaxes`.

Example 29. Similar-tuple relationship 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 using the XML Schema `redefine` construct that is prohibited by XBRL section 5.1.5.

2.4.3 The requires-element relationship SHOULD 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>
```

```
  <sequence>
    <element ref="TheElement">
    <element ref="TheElementThatIsRequired">
  </sequence>
  <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.

3 Discoverable taxonomy set layer

The DTS layer of the financial reporting taxonomy architecture encompasses the scope, syntax, naming and documentation relating to a DTS whose starting point is a given taxonomy schema.

3.1 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.

3.2 Rules for discoverable taxonomy set structure

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

3.2.1 A schema document MUST contain only declarations of reference parts OR declarations of concepts, roles and arc roles OR declarations that are not concepts and not reference parts.

A DTS must be organised with separate schema files so as to partition the entire set of element definitions into separate schema documents containing different types of definitions:

1. Reference parts;
2. Content for context segment and scenario elements (which are subject to the rules in [XBRL] sections 4.7.3.2 and 4.7.4);
3. Taxonomy concepts, custom roles and arc roles.

Consequences of this (taking [XBRL] section 5.1 into account) include the following:

- Any taxonomy schema that contains a declaration of an element substitutable for `xbrli:item` or `xbrli:tuple` MUST NOT contain declarations for any elements that are not substitutable for `xbrli:item` or `xbrli:tuple`.
- Any taxonomy schema that contains a declaration of an element substitutable for `link:part` MUST NOT contain declarations for any elements that are not substitutable for `link:part`. Note that `link:part` is a simple type and therefore a “part” would never have any child elements.
- If a schema contains `roleType` or `arcroleType` elements, any element declarations it contains MUST be for elements substitutable for `xbrli:item` or `xbrli:tuple`.

- Type declarations MAY appear in any schema.
- XML Schema `group`, `attributeGroup` and `attribute` declarations MAY appear in any schema.
- This rule imposes no constraints whatsoever on the contents of schemas that do not contain one of the following: a) concept declarations; b) reference part declarations; c) `roleType` elements; d) `arcroleType` elements.

3.2.2 Taxonomy schemas MUST be defined in XML documents in which the XML Schema "schema" element appears once only as the root element.

Taxonomy Schemas are XML Schemas, which are represented in XML as one or more `schema` elements. XML Schema [SCHEMA-1], [SCHEMA-2] allows these elements to appear anywhere in an XML document, but many Schema-validating XML parsers will only process `schema` elements that are the root element in the document in which they appear. To facilitate processing, this rule requires that every `schema` element is the root element in its containing XML document.

3.2.3 Taxonomy schemas MUST NOT contain embedded linkbases.

This is a consequence of 3.2.2 above to clarify that a linkbase is not considered a part of a taxonomy schema.

3.2.4 Taxonomy schemas MUST declare elementFormDefault to be "qualified," attributeFormDefault MUST have the value "unqualified", and the "form" attribute MUST NOT appear on element and attribute declarations.

This rule ensures consistent treatment of references to attributes and elements in element definitions. The XML Schema form attribute is disallowed because it could be used on individual attribute and element declarations to override the defaults specified on the schema element; "unqualified" is the default for attributeFormDefault [SCHEMA-0].

3.2.5 All extended-type links in a single linkbase MUST have the same namespace and local name.

Each linkbase can only contain one kind of extended link, such as `labelLink`, `referenceLink`, `definitionLink`, `calculationLink` or `presentationLink`.

3.2.6 A label linkbase SHOULD only contain labels defined in a single language.

A "single language" means a single ISO 639 language code [ISO]. For example, "`en`" and "`en-nz`" are distinct for this purpose.

3.2.7 A taxonomy schema SHOULD NOT contain import or include elements not strictly needed for XML Schema validity.

Many XBRL taxonomy schemas, even though they represent extensions of other taxonomies, will not need to import any schema other than the base XBRL schemas themselves. Identifying the taxonomy being extended is rarely needed, since the rules of DTS discovery will traverse the linkbase in question to gather all relevant taxonomy schemas. Furthermore, where such references are needed in order for

XBRL validation to perform correctly, `schemaRef` in instances is preferable to `import` and `include` in schemas.

3.2.8 A DTS SHOULD include scenario element definitions that are relevant to the reporting standard upon which it is based, unless such elements already exist in a recommended taxonomy.

Any context element that omits further detail in its scenario sub-element is left open to interpretation: is it a reported, verifiable fact, an estimate, a restatement of a prior period reported value? If these distinctions are important in the reporting standard, then they should be encoded as elements to appear in the `scenario` element. Applying the general principle that a FRTA compliant taxonomy MUST NOT redefine elements that already have recommended definitions, scenario elements appearing in XBRL International taxonomies that have achieved recommended status MUST be used if applicable.

3.2.9 Every schema in a DTS MUST define a non-empty targetNamespace attribute value.

The `targetNamespace` of a schema not only allows the definitions in the schema to be uniquely identified, but it also allows the entire schema to be referred to. Although a schema MAY be devoid of concept definitions (as for example if its only function is to group a set of linkbases and other schemas), it MUST still have a non-empty namespace.

3.3 Taxonomy naming rules

The conventions in this section relate to taxonomy (as opposed to element) naming and related rules.

3.3.1 Each unique taxonomy schema target namespace MUST have one and only one namespace prefix which will be its recommended namespace prefix.

The recommended namespace prefix SHOULD suggest the distinct scope and purpose of the concepts defined within that namespace.

Example 30. Recommended namespace prefixes.

Prefix	Meaning
`int-gcd`	International Global Common Document concepts
`ifrs-gp`	IFRS General Purpose concepts
`us-gaap-ci`	US GAAP Commercial and Industrial concepts
`au-ifrs-ci`	Australian IFRS extensions
`au-cd`	Australian Common Document concepts

The recommended namespace prefix MUST be the only prefix used in any importing FRTA compliant taxonomy schema.

3.3.2 A taxonomy that supersedes an existing version of itself SHOULD 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.

At this time there is no taxonomy element to express the linkage between two versions of taxonomy other than this naming convention.

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

Taxonomy file names SHOULD follow the pattern:

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

The {linkbasetype} must be one of label, reference, presentation, calculation, or definition. The {-qualifier} must not be used for any linkbase which is the "default" linkbase, as for example a presentation linkbase intended for use in presenting the taxonomy.

Example 31. Taxonomy file names with qualifiers.

File name	Meaning
`ifrs-gp-2004-08-15.xsd`	IFRS-GP schema
`us-gaap-ci-2004-08-15.xsd`	US-GAAP-CI schema
`us-gaap-ci-2004-08-15-label.xml`	US-GAAP-CI (default US English) labels linkbase
`us-gaap-ci-2004-12-25-label-es.xml`	US-GAAP-CI Spanish labels linkbase
`in-gaap-2005-12-31-reference-hi-np.xml`	Indian GAAP reference links to Hindi sources applicable to non-profit organisations.

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 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.

Appendix I

Following table, describes the changes that have been made to FRTA 1.0, to create FRTA 1.5.

Old Clause	New Clause	Description
2.1.1	1.1.1	No Change
2.1.2	1.1.2	No change
2.1.3	1.1.3	No Change
2.1.4	1.1.4	Added terms Non-graphical, as the rule will not apply to a graphical language name
2.1.5	1.1.5	No Change
2.1.6	NA	Deleted, this decision should be left to the taxonomy owners
2.1.7	NA	Deleted, all may rules will be deleted
2.1.8	1.1.6	No Change
2.1.9	1.1.7	No Change
2.1.10	1.1.8	No Change
2.1.11	1.1.9	No Change
2.1.12	1.1.10	Included Non-abstract concept, removed the third option of having documentation in the reference, and included an exception to this rule to cater to dimensional items
2.1.13	1.1.11	No Change
2.1.14	1.1.12	No Change
2.1.15	1.1.13	No Change
2.1.16	1.1.14	No Change
2.1.17	1.1.15	No Change
2.1.18	1.1.16	Included a sentence, If the DTS contains more than one label for a label role and language, then there should be a clear indication of which is the effective label, through use of priority and prohibited
2.1.19	1.1.17	No Change
2.1.20	1.1.18	No Change
2.1.21	NA	Deleted, already provided for in the specification
2.1.22	1.1.19	No Change
2.2.1	1.2.1	No Change
2.2.2	1.2.2	No Change
2.2.3	NA	Deleted, already provided for in the specification
2.2.5	1.2.4	No Change
2.2.6	NA	Deleted, already provided for in the specification
2.2.7	1.2.5	No Change
2.2.8	1.2.6	No Change
2.2.9	1.2.7	No Change
2.2.10	1.2.8	No Change
2.2.11	NA	Deleted

2.2.12	NA	Deleted
2.2.13	NA	Deleted
2.2.14	NA	Deleted
2.3.1	1.3.1	No Change
2.3.2	1.3.2	No Change
2.3.3	1.3.3	No Change
2.3.4	1.3.4	No Change
2.3.5	1.3.5	No Change
2.3.6	NA	Deleted
2.3.7	1.3.6	No Change
2.3.8	NA	Deleted, already provided for in the specification
3.1.1	NA	Deleted, already provided for in the specification
3.1.2	2.1.1	Changed, to define the scope which is limited to XBRL 2.1
3.1.3	2.1.2	No Change
NA	2.1.3	New rule
3.1.4	2.1.4	No Change
3.1.5	2.1.5	No Change
3.1.6	2.1.6	No Change
3.1.7	NA	Deleted, to cater to dimensional relationship
3.1.8	2.1.7	No Change
3.1.9	3.1.8	No Change
3.1.10	2.1.9	Changed from MUST to SHOULD
3.1.11	2.1.10	Changed from MUST to SHOULD
3.1.12	NA	Deleted
3.1.13	2.1.11	Changed, defined the scope of the rule
3.1.14	2.1.12	No Change
3.1.15	NA	Deleted, already provided for in the specification
3.1.16	NA	Deleted, already provided for in the specification
3.1.17	NA	Deleted, already provided for in the specification
3.2.1	NA	Deleted, already provided for in the specification
3.2.2	2.2.1	No Change
3.2.3	2.2.2	No Change
3.2.4	2.2.3	No Change
3.2.5	NA	Deleted, all may rules will be deleted
3.2.6	2.2.4	No Change
3.2.7	2.2.5	No Change
3.3.1	2.3.1	No Change
3.3.2	2.3.2	No Change
3.3.3	2.3.3	No Change
3.3.4	2.3.4	No Change
3.3.5	2.3.5	No Change

2.3.6	3.3.6	No Change
3.4.1	NA	Deleted
3.4.2	2.4.1	No Change
3.4.3	2.4.2	No Change
3.4.4	2.4.3	Changed from MUST to SHOULD
4.1	3.1	No Change
4.2	3.2	No Change
4.2.1	3.2.1	No Change
4.2.2	3.2.2	No Change
4.2.3	3.2.3	No Change
4.2.4	3.2.4	No Change
4.2.5	NA	Deleted
4.2.6	3.2.5	No Change
4.2.7	3.2.6	No Change
4.2.8	NA	Deleted, already provided for in the specification
4.2.9	3.2.7	No Change
4.2.10	3.2.8	No Change
4.2.11	3.2.9	No Change
4.3.1	NA	Deleted, obsolete
4.3.2	3.3.1	Changed, to remove the twelve character length limit in the prefix
4.3.3	3.3.2	Changed from MUST to SHOULD
4.3.4	3.3.3	No Change
5	NA	Deleted the section to limit the scope of the document to Level 1 and Level 2 taxonomies