Extensible Business Reporting Language (XBRL) 2.1

Public Working Draft of 2003-04-23

This version:

http://www.xbrl.org/2003/XBRL-WD-2003-04-23.doc (Word)
http://www.xbrl.org/2003/XBRL-WD-2003-04-23.pdf (Acrobat)

with separate provision of XML Schemas described herein. All components, along with non-normative samples and certain schemas are available in a single Zip format archive:

http://www.xbrl.org/2003/XBRL-WD-2003-04-23.zip

Editors

Name	Contact	Affiliation
Phillip Engel	phillipengel@kpmg.com	KPMG LLP
Walter Hamscher[1]	walter@hamscher.com	Standard Advantage
David vun Kannon	dvunkannon@kpmg.com	KPMG LLP
Hugh Wallis	hugh_wallis@hyperion.com	Hyperion Solutions Corporation

Contributors

Name	Contact	Affiliation
Charles Hoffman	charleshoffman@olywa.net	UBmatrix LLC
Campbell Pryde	cpryde@kpmg.com	KPMG LLP
Geoff Shuetrim	gshuetrim@kpmg.com.au	KPMG LLP

Status of this document

This document is a Public Working Draft. The process leading to the publication of this specification as a final recommendation of XBRL International is shown in the appendix on the last page. Comments should be directed to the editors. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

While excerpts from XBRL schemas are given throughout this document the complete versions of the schemas are available as separate .xsd files from www.xbrl.org, the XBRL International web site.

Abstract

XBRL is the specification for the eXtensible Business Reporting Language. XBRL allows software vendors, programmers, intermediaries in the preparation and distribution process and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information. Business reporting includes, but is not limited to, financial statements, financial information, non-financial information, general ledger transactions, and regulatory filings such as annual and quarterly financial statements.

This document defines XML elements and attributes that can be used to express information used in the creation, exchange, and comparison tasks of business reporting. XBRL consists of a core language of XML elements and attributes used in XBRL instances as well as a language used to define new elements and taxonomies of elements referred to in XBRL instances, and to express constraints among the contents of elements in those XBRL instances.

[1] Walter Hamscher is a consultant to PricewaterhouseCoopers LLP.

Table of contents

List of tables

List of examples

1 Introduction

XBRL is the specification for the eXtensible Business Reporting Language. XBRL allows software vendors, programmers and end users to enhance the creation, exchange, and comparison of business reporting information. Business reporting includes, but is not limited to, financial statements, financial information, non-financial information and regulatory filings such as annual and quarterly financial statements.

This document defines XML elements and attributes that can be used to express information used in the creation, exchange and comparison tasks of business reporting. XBRL consists of a core language of XML elements and attributes used in document instances. Abstract elements in this core language are replaced by concrete elements in XBRL instances. These abstract elements are defined in taxonomies. XBRL consists of a language used to define new elements and taxonomies of elements referred to in document instances and the relationships between taxonomy elements.

All parts of this document not explicitly identified as non-normative are normative. In the event of any conflict between normative parts of this document and schemas referenced herein that are published contemporaneously with this document, the document prevails.

1.1 Documentation conventions

The following highlighting is used to present technical material in this document:

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

Non-normative editorial comments are denoted by indentation and the prefix “Note”:

> Note: This is a non-normative editorial comment.

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

1.2 Purpose

The XBRL specification is intended to benefit four categories of users: 1) business information preparers, 2) intermediaries in the preparation and distribution process, 3) users of this information and 4) the vendors who supply software and services to one or more of these three types of user. The overall intention is to balance the needs of these groups creating a product that provides benefits to all groups.

The needs of end users of business information have generally had precedence over other needs when it has been necessary to make specification design decisions that might benefit one community at the expense of another.

A major goal of XBRL is to improve the business report product. It facilitates current practice; it does not change or set new accounting or other business domain standards. However, XBRL should facilitate changes in reporting over the long term.

XBRL provides users with a standard format in which to *prepare* reports that can subsequently be presented in a variety of ways. XBRL provides users with a standard format in which information can be *exchanged* between different software applications. XBRL permits the automated, efficient and reliable *extraction* of information by software applications. XBRL facilitates the automated *comparison* of financial and other business information, accounting policies, notes to financial statements between companies, and other items about which users may wish make comparisons that today are performed manually.

XBRL facilitates "drill down" to detailed information, authoritative literature, audit and accounting working papers. XBRL includes specifications for as much information about the reporting entity as may be relevant and useful to the process of financial and business reporting and the interpretation of the information.

XBRL supports international accounting and other standards as well as languages other than the various dialects of English.

XBRL is extensible by any adopter to increase its breadth of applicability, and its design encourages reuse via incremental extensions. XBRL specifies the format of information that would reasonably be expected in an electronic format for securities filings by public entities, for example. XBRL facilitates business reporting in general, and is not limited to financial and accounting reporting.

XBRL focuses on the genuine information needs of the user and adheres to the spirit of reporting standards that avoid the use of bold, italics, and other stylistic techniques that may distract from a true and fair presentation of results. Therefore, there is no functional requirement that XBRL documents support any particular text formatting conventions.

The purpose of XBRL instances is the transmission of a set of facts. There is no constraint on how much or how little they contain. A single fact can form the entire content of a valid XBRL document, for example, when the information being conveyed is limited to what “Cost of Goods Sold” was last quarter. An XBRL document can be a database dump, containing huge numbers of facts. It can be anything in between. This provides a great deal of flexibility and is meant specifically to achieve the goals of allowing XBRL to be reused within other specifications and for application software needing to extract data from otherwise arbitrarily formatted documents. It is expected that, for most uses of XBRL, many XML XBRL instances will be created that consist almost exclusively of facts.

1.3 Relationship to other work

XBRL uses several World Wide Web consortium (W3C) recommendations, XML 1.0, XML Namespaces, and refers directly to XML Linking. It also relies extensively on the XML Schema recommendation.

Discussions have taken place with other bodies issuing XML specifications in the financial arena, including OAG (Open Applications Group), OMG (Object Management Group), FpML (Financial Products Markup Language), finXML (Financial XML), OFX/IFX (Open Financial Exchange) and ebXML (e-Business XML). The scope of XBRL does not include transaction protocols. It includes financial reporting and contemplates extensive detail in the representation and use of accounting conventions, which distinguishes it from these other efforts.

1.4 Terminology

The terminology used in XBRL frequently overlaps with terminology from other fields, and the following short list is provided to reduce the possibility of ambiguity and confusion (see also the references in section 8 below).

Table 1. Terms and definitions.

Term	Definition
abstract element	An element that cannot be used in an XBRL instance. Instead, concrete elements that substitute for abstract elements must be used.
alias concept	The concept at the “from” end of an alias-essence definition arc or the concept at the “to” end of an essence-alias arc.
alias item	An item in an instance whose element is an alias concept.
concept	An XML Schema element definition that declares the element to be in the substitution group of either the XBRL item or XBRL tuple element (both of which are abstract elements defined in the XML Schema distributed with this specification).
concrete element	An XML element that appears in an XML instance.
c-equal	Context-equal: Items having the same item type in s-equal contexts.

CWA	"Closed World Assumption", a term used in logic and database theory. If the closed world assumption is true, the database or the portion of the XBRL instance for which it is stated to be true is understood to hold all the facts necessary to do further computations. The definitions for the computations MAY be found in one or more linkbases that MAY be part of a referenced taxonomy or directly referenced by the XBRL instance.
duplicate items	Both p-equal and c-equal: two occurrences in an XBRL instance of the same concept in the same context under the same parent.
duplicate tuples	Two occurrences of a tuple with all their descendants having the same content; more precisely: tuples that are p-equal, all of whose tuple children have a duplicate in the other tuple, and all of whose item children are v-equal to an item in the other tuple.
element	An XML element defined using XML Schema, but also a concept described by a taxonomy. For example, the element with the name "`landAndBuildingsNet`" is a concept.
entity	A business entity, the subject of XBRL items. Where the XML/SGML concept of syntactic "entity" is meant, this will be pointed out.
equal	Elements are *equal* when elements, their attributes and content share the same namespace and pairs of string values, that when compared using the XPath 1.0 "=" operator [XPATH], would return a value of "true."
error	A violation of the rules of this specification; results are undefined. Conforming software MAY detect and report an error and MAY recover from it.
essence concept	The concept at the "to" end of an alias-essence arc or the concept at the "from" end of an essence-alias arc in a definition linkbase.
essence item	An item in an instance whose element is an essence concept.
fatal error	An error, which a consuming application MUST detect and report. After encountering a fatal error, the application MAY continue processing the data to search for further errors and MAY report such errors. In order to support correction of errors, the processor MAY make unprocessed data from the document (with intermingled character data and mark-up) available to the application. Once a fatal error is detected, however, the processor MUST NOT continue normal processing (*i.e.*, it MUST NOT continue to pass character data and information about the document's contents to the application in the normal way).
instance namespace	The namespace of XBRL 2.1 instances, `http://www.xbrl.org/2003/instance`
item	An element derived from the abstract XML element "item" in XBRL. A concrete element of that item occurring in an XBRL instance represents a fact reported about a given business entity pertinent to a particular period (see definition below).
least common ancestor	In an instance, the element that is an ancestor of two elements and has no child that is also appears on the ancestor axis [XPATH] of those same two elements.
linkbase namespace	The namespace of XBRL 2.1 linkbases, `http://www.xbrl.org/2003/linkbase`
MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY, OPTIONAL	See [RFC2119] for definitions of these and other terms as used in this specification. These include, for example: MAY — Conforming documents and consuming applications are permitted to but need not behave as described. MUST — Conforming documents and consuming applications are required to behave as described; otherwise they are in error. at user option — Conforming software MAY or MUST (depending on the modal in the sentence) behave as described; if it does, it MUST provide users a means to enable or disable the behaviour described.
non-numeric item	An item whose content, if any, is not numeric. Dates, in particular, are not numeric.
numeric item	A fact whose simple content is derived by restriction from the XML Schema primitive types `decimal`, `float` or `double`, or complex content derived by restriction from the XBRL defined type `fractionItemType` (see Other data types, below).
child, parent, ancestor, sibling, grandparent, uncle	Relationships among elements in an XBRL instance: the path [XPATH] "*" from an element reaches its children, "`..`" its parent, "`../*`" its siblings, "`../..`" its grandparent, "`../../*`" its uncle, and "`ancestor::*`" its ancestors.

period	An instant or duration of time. In business reporting, financial numbers and other facts are reported "as of" an instant or for a period of certain duration. Facts about instants and durations are both common.
p-equal	Parent-equal: instance items or tuples having the same parent.
s-equal	Structure-equal: XML nodes that are either equal in the XML value space, or whose XBRL-relevant sub-elements and attributes are s-equal.
taxonomy	One or more XML Schemas that defines new XBRL elements each corresponding to a concept that can be referenced in XBRL documents, along with a set of XML Links that express relationships among the concepts and between the concepts and other information sources.
tuple	A term meaning a group of items that MUST be kept together to be understood. The XBRL tuple element is a container used to hold items together.
v-equal	Value-equal: c-equal items having either the same non-numeric value, or numeric values that are equal within some tolerance defined by the lesser of their respective precision or decimals attributes.
XBRL instance	An XML document containing concrete elements that together constitute one or more business reports. The financial statements of Apple, expressed in XBRL, would be an XBRL instance.

2 Changes from the previous published version

Changes from the previous, December 2001 version of XBRL (and the interim 2.0a "patch" release in November 2002) were driven by two factors. Several implementations of XML Schema required the removal of an ambiguous content model from the definition of contexts. This was done without changing the language recognized by the schema. Further implementation experience within the XBRL community, especially the publication of the XBRL General Ledger taxonomy, motivated many other changes. 2.1 A number of business requirements documented by the XBRL International Domain working group have been incorporated.

There is only one change in XBRL 2.1 relative to XBRL 2.0 that will prevent 2.1-compliant applications from being able to process 2.0 instances and taxonomies with no loss of information: a new required Boolean attribute, `instantaneous`, has been added to the element definition. The attribute specifies whether an item represents an instantaneous measurement or a measurement over a period of time. In every other case these changes involved restricting the syntax and clarifying processing semantics.

2.1 Changes in XBRL instances

The `group` element has been deprecated, but is still included for a transitional period. It may be removed in a future release of the specification. A new "`xbrl`" element has been added, which acts as the root element of an XBRL instance. A new set of profile attributes on the "`xbrl`" element allows the document author to indicate the level of processing capabilities needed to correctly process the content.

Duplicate items are prohibited in XBRL instances by default. The definition of a duplicate item has been changed to include reference to the content of any tuple structures that contain the items being compared. In addition, the prohibition of duplicate items has been extended to duplicate tuples.

To specify that numbers are stated exactly in an XBRL instance, two new types have been defined for use by the decimals and precision attributes. These types enable XBRL instances to specify that numbers are represented to an infinite number of significant figures or number of decimal places.

Guidance has been included on the entry of numerical quantities in XBRL instances for the common case of elements from accounting related taxonomies (elements using the optional "`balance`" attribute in their definition). The duration element has been deprecated. There is also additional guidance on entering data to define a period of time, and on appropriate use of the `cwa` attribute.

Additional mechanisms have been introduced to enable XBRL instance preparers to make statements about the numerical accuracy of the facts reported. The "precision" attribute on `numericContext` has been deprecated and its function has been provided by a "precision" attribute on items of numeric type instead. A new attribute "decimals" has been added to items of numeric type to indicate the number of decimal places to which a numerical fact is accurate. Rules for handling combined precision and decimals information have been provided.

Derivation of new item and tuple types from those defined by XBRL itself have been restricted so that item types may only be defined by extension from simple types in XML Schema or the fraction item type, and tuples may only be derived by restriction using the choice and sequence constructs of XML Schema.

2.2 Changes in XBRL taxonomies

All of the arcroles and roles previously *suggested* are now *normative*, in the sense that consuming applications may deem any other arcroles and roles encountered as application-specific, and ignore them. A means for declaring and documenting non-standard arcroles and roles has been added. The element-dimension relationship in the calculation linkbase has been superseded by the alias-essence relationship in the definition linkbase. The parent-child / child-parent relationship no longer exists in the calculation linkbase and has been replaced by summation-item / item-summation. The parent-child / child-parent relationship no longer exists in the definition linkbase and has been replaced by special-general / general-special and part-whole / whole-part. The general-special / special-general relationship now plays the role of establishing correspondences between items and tuples defined in extension taxonomies and how they relate to their precursors in the base taxonomy. Some relationships are no longer allowed to contain cycles. Tuples may now have a complex content model, but restricted only to the XML Schema `choice` and `sequence` constructs with `minOccurs` and `maxOccurs` attributes and only references to global elements [SCHEMA-1].

Calculation arcs may now specify the contexts to which their summation applies, and indicate how the contexts of the summing elements may differ from that of the summed element. This allows for a wider variety of calculation relationships such as that between the starting, ending, and change in a quantity over a given period. Calculations have been constrained to only apply within the scope of a tuple for items within a tuple.

The number of available item types has been expanded to include all the primitive data types of XML Schema [SCHEMA-2].

A new type for items has been defined to allow the specification of facts that are reported as fractions (such as 22.5/77.5). The fraction type is not among the built-in data types of XML Schema [SCHEMA-2]. Since fractions have two parts, denominator and numerator, it has complex content.

The suggested `xlink:role` attribute content that indicated the root element of a linkbase has been given more specific semantics.

3 XBRL framework

The main ideas in the XBRL conceptual framework are *items* and *taxonomies;* taxonomies in turn consist of *schemas* and *linkbases.* These terms are used in a precise way within XBRL.

Items. In the XBRL framework, the most fundamental concept is that of the *item.* An *item* is meant to correspond to a fact —often but not necessarily a numeric fact— that is being reported with respect to a given period of time about a given business entity. For example, the fact that the company whose ticker symbol is SAMP reported revenues of USD 7m for the year 1998 is an item. This is an example of a numeric item. An example of a non-numeric item would be a paragraph of text describing the principles of consolidation used to combine reports from the subsidiaries of SAMP. Although the latter is not numeric, this is nevertheless a fact being reported with respect to a given period of time (the year 1998) about a given business entity (SAMP).

XBRL defines a syntax in which many different kinds of facts can be represented and their contexts defined in such a way that software applications can efficiently and reliably find, extract, and interpret relevant facts in their appropriate contexts.

Tuples. It is often the case that facts must be joined together to be understood. A tuple, like a row in a database table, is a grouping of facts. For instance, the name, age and compensation of a director of a company should be grouped together to be correctly understood.

Root element. The root of an XBRL instance is the `xbrl` element (which replaces the deprecated `group` element). It is possible in principle to embed an XBRL item in *any* XML document. In this case, the `xbrl` element (or deprecated `group` element) is the container for the XBRL fragment.

Elements and Taxonomies. An equally important part of the XBRL framework is the concept of an *element* and its relationships to other elements within a *taxonomy.* In XBRL, the notion of a taxonomy element is represented by an element within an XML Schema [SCHEMA-1].

An example taxonomy for the purposes of the current specification is the particular taxonomy consisting of elements that correspond to well-defined concepts within the US Generally Accepted Accounting Principles (GAAP) when those principles are applied to Commercial and Industrial (C&I) companies. For example, concepts of "Accounts Receivable Trade, Gross", "Allowance for Doubtful Accounts", and "Accounts Receivable Trade, Net" may be associated with different elements in this particular taxonomy.

Although any given item can only be defined in a single schema, within any given XBRL instance any number of XBRL items can be elements from any number of schemas.

Linkbases. Relationships among the elements in a taxonomy, and relationships between the taxonomy and information outside of XBRL, are represented by linkbases using XML Linking Language [XLINK]. A linkbase consists of one or more “extended links,” which in turn consist of arcs, locators and resources; arcs with different “arcroles” have different semantics. An arc between two elements may, for example, indicate that XBRL instances may contain one of the elements as a sub-element of the other, or it may indicate that the values of the two elements in an instance must be equal; there are many such relationships all represented in the linkbases. A schema MAY refer to any number of linkbases to capture the semantics about, element labels, guidance for presenting the taxonomy to authors and editors, reference materials pointing to authoritative literature, definitions that group the elements, and calculations that applications use to check the consistency of the content of an XBRL instance. Collectively, schemas and their associated linkbases are referred to as taxonomies.

Individual XBRL taxonomies can also be used as “building blocks” to create larger, more sophisticated taxonomies. Users MAY compose groups of existing taxonomies or individual taxonomy schemas and linkbases into higher-level taxonomies and MAY selectively add elements and other relationships to taxonomies to specialize them: suppose, for example, that a significant portion of expenses is (in a hospital, for example) “physician salaries”. Because that term does not exist in the Financial Reporting for Commercial and Industrial Companies, US GAAP taxonomy as such, a new (small) taxonomy would be created to define an element for the concept “physician salaries” and new linkbases would relate this to the concept of “expenses” that already exists in the US GAAP taxonomy.

New linkbases can be used to augment existing taxonomies. A published taxonomy whose label linkbase provides only English labels could, for example, be augmented with a single linkbase providing Spanish labels. The author of the Spanish linkbase could publish it either standalone for XBRL applications to use, or could create an “extension taxonomy” consisting of a small Schema which simply imports the published taxonomy and includes a linkbase reference to the Spanish labels. XBRL instances and applications could then refer to the new extension taxonomy.

Specifications of additional linkbases may subsequently be published as XBRL recommendations as needs arise and are documented through the XBRL International process.

3.1 Processing

While some consuming applications might be able to perform processing on an XBRL data file without referring to any taxonomies that it references, normally, the interpretation and processing of any given XBRL item is relative to the contents of a named taxonomy.

For example, to correctly produce a table of values with rows corresponding to an ordered set of types and columns representing different periods, it is necessary to find the `label` elements corresponding to each item. The labels are found in a linkbase whose location is provided by a simple XLink in a taxonomy schema. That schema may import or include other schemas, and that collection of schemas may reference any number of

XBRL linkbases. When processing an instance, consuming applications should use any and all linkbases referenced directly or indirectly in this way. This is because publishers of XBRL instances and taxonomies may selectively reference other taxonomies and linkbases in order to convey meaning; for example, the same set of elements defined in a schema file might have Spanish and Portuguese literature references defined in different linkbases; the document publisher might specify either, both, or neither of these linkbases in order to specify which set of definitions they consider to be more appropriate.

Treatment of relative pathnames and caching of taxonomy files is implementation-dependent. For example, if a document instance contains a relative URL as the location of a `schemaLocation` attribute, it is up to the consuming application to dereference it; it is an error if the underlying taxonomy cannot be found. In particular, because of differences in the way that XML validation engines interpret the XML Schema specification [SCHEMA-1] and process the import element and `schemaLocation` attributes, normative XBRL schemas and linkbases might not validate in all applications.

3.2 Data integrity and confidentiality

There are many applications that require business information to be transmitted securely, with a particular emphasis on data integrity (leading to the use of hash totals, etc.) and with confidentiality (leading to the use of cryptographic means of protection). XBRL deliberately provides neither of these mechanisms, since its focus is on transmission of actual content in an agreed-upon format; it is assumed that like any other block of data, data integrity can be enhanced by adding redundant error correction bytes, by cryptographic hashing and signing with a private key, etc. These mechanisms are all outside the scope of XBRL.

An XBRL payload – `xbrl`, `group`, `linkbase`, and their children – does not have to be aware of whether all or some of it has been manipulated to be signed, encrypted, canonicalised, compressed, etc. By the time XBRL processing has to take place, all of those manipulations will have been unwound, and the XBRL payload will be free of any evidence of those operations.

3.3 Validation

XBRL imposes syntactic restrictions that are not captured in XML Schema. XBRL instances MUST validate against all of the taxonomy schemas and XML schemas to which it refers, and taxonomy schemas and linkbases MUST validate against the schemas to which they refer. An XBRL validating processor MUST signal a fatal error if the XBRL instance violates XBRL syntactic constraints. Consuming applications, prior to using XBRL data, SHOULD also perform XBRL validation.

A definition linkbase (see 5.5.7.15 below) describes constraints over the content of items. An XBRL validating processor MUST signal an error if the XBRL instance violates these constraints.

A calculation linkbase (see 5.5.7.11 below) describes semantic constraints over the content values of numeric items. Because the calculation links and values explicitly given in an XBRL instance can be used to check the consistency of other explicitly stated values in those specific contexts where the closed world assumption (see 4.4.5 below) holds, an XBRL validating processor MAY signal an error if the XBRL instance contains arithmetic inconsistencies with the semantic constraints imposed.

4 XBRL instances

An XBRL instance MUST comply with the general rules and structures specified herein. The syntax for XBRL instances is defined using an XML Schema. Example elements defined there include `xbrl`, `item`, `numericContext`, `nonNumericContext`, and `tuple`. All XBRL instances MUST be valid XML documents as defined by the XML Schema validation criteria in [SCHEMA-1] and the instance schema and MUST be recognised as such by any XML Schema minimally conforming processor.

The semantics of XBRL instances and their contents are specified only insofar as they impact the operation of software applications that use this specification. Refer to 3.1 above for more about the processing context assumed for XBRL instances.

4.1 The xbrl element

Expressing even a single fact in an XBRL instance requires multiple elements, at least one item and a context containing sub-elements. Therefore, a container element is necessary to serve as the root element of an XBRL instance. If multiple "data islands" of XBRL mark-up are included in a larger document, the `xbrl` element is the container for each. The schema definition for the `xbrl` element is set out below.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<complexType name="rootType" abstract="true">
   <choice minOccurs="0" maxOccurs="unbounded">
     <element ref="xbrli:item" minOccurs="0" maxOccurs="unbounded"/>
     <element ref="xbrli:tuple" minOccurs="0" maxOccurs="unbounded"/>
     <element ref="xbrli:nonNumericContext" minOccurs="0" maxOccurs="unbounded"/>
     <element ref="xbrli:numericContext" minOccurs="0" maxOccurs="unbounded"/>
     <element ref="link:linkbaseRef" minOccurs="0" maxOccurs="unbounded"/>
     <element ref="link:footnoteLink" minOccurs="0" maxOccurs="unbounded"/>
   </choice>
</complexType>

<element name="group" type="xbrli:rootType"/>
<element name="xbrl">
  <complexType>
    <complexContent>
      <extension base="xbrli:rootType">
          <attributeGroup ref="xbrli:profileAtts"/>
      </extension>
    </complexContent>
  </complexType>
</element>
```

Example 1. Use of xbrl as the root element.

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ci="http://www.xbrl.org/us/gaap/ci/2003/usfr-ci-2003"
      xsi:schemaLocation="
http://www.xbrl.org/us/fr/ci/2003/usfr-ci-2003
http://www.xbrl.org/us/fr/ci/2000-07-31/usfr-ci-2003.xsd">
  <ci:assets numericContext="c1">727</ci:assets>
  <ci:liabilities numericContext="c1">635</ci:liabilities>
  <numericContext id="c1" cwa="false"><!-- ... --></numericContext>
</xbrl>
```

Meaning: `xbrl` holds namespace prefix definitions and the `schemaLocation` attribute.

4.1.1 Profile attributes

The `xbrl` element has a set of attributes that allow a document author to indicate the level of processing capabilities needed to correctly process the content. Certain attribute values signal to the consuming application that specific usages do not appear. The restrictions expressed are those that cannot be adequately signaled by the taxonomy itself.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <attributeGroup name="linkbaseProfileAtts">
    <attribute name="noRoleTypes" type="boolean" default="false"/>
  </attributeGroup>
</schema>
```

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <attributeGroup name="profileAtts">
    <attributeGroup ref="link:linkbaseProfileAtts"/>
    <attribute name="noDeprecated" type="boolean" default="false"/>
    <attribute name="noDerivations" type="boolean" default="false"/>
    <attribute name="duplicatesLegal" type="boolean" default="false"/>
    <attribute name="profile" type="anyURI"/>
   </attributeGroup>
</schema>
```

4.1.1.1 The noDeprecated attribute

When `true`, deprecated elements and attributes appearing in this instance and in any referenced taxonomies or linkbases MUST be ignored. The default is `false`.

Example 2. The noDeprecated attribute

Example	Meaning

`<xbrl noDeprecated="true"`
`xmlns="http://www.xbrl.org/2003/instance"`
`xmlns:nl="http://www.xbrl-nl.org/2004/tax">`
	This XBRL instance is based on a taxonomy developed in 2004 and the instance, the taxonomy and all its linkbases are free of deprecated syntax. Applications compliant with current XBRL syntax MUST be able to process this instance.

4.1.1.2 The noDerivations attribute

When `true`, items whose type is other than an XBRL defined item type MUST be ignored. The default is `false`.

Example 3. The noDerivations attribute

Example	Meaning

`<xbrl`
`noDerivations="true"`
`xmlns:ma="http://www.xbrl-ma.org/2003/ma">`
	Applications that are not able to process anything other than XBRL defined item types should be able to process this instance.

4.1.1.3 The noRoleTypes attribute

The `noRoleTypes` attribute from the linkbase namespace MAY appear here. Attributes in the attribute group `linkbaseProfileAtts` are described in 5.5.2 below.

Example 4. The noRoleTypes attribute

Example	Meaning

`<xbrl`
`noRoleTypes="true"`
`xmlns:nv="http://www.xbrl-nv.org/2003/nv">`
	Processing of this instance requires nothing other than XBRL defined arc roles.

4.1.1.4 The duplicatesLegal attribute

When `true`, XBRL validating processors MUST NOT signal an error in the presence of duplicate items and tuples in the instance. The default is `false`. For example, in taxonomies and instances used to represent sets of individual accounting transactions as in XBRL GL, syntactically identical items and tuples represent distinct facts, and the `duplicatesLegal` attribute should be set to `true`.

Example 5. The duplicatesLegal attribute

Example	Meaning

`<xbrl duplicatesLegal="true"`
`      xmlns:nv="http://www.xbrl-nv.org/2003/nv">`
	Either this instance is already guaranteed to be duplicate-free, or, its duplicate items and tuples are to be taken as if they were distinct items.

4.1.1.5 The profile attribute

The profile attribute provides a simple extension mechanism for application profiles. Applications recognising only a subset of XBRL features may use a URI to uniquely identify that set of capabilities. The URI must be the online location of a textual description. Creating an XBRL instance with that URI in its `profile` attribute indicates to any consuming application what specified subset of features is needed. Applications MAY ignore this attribute.

Example 6. Values of the profile attribute

Example	Meaning
`<xbrl>`	Any XBRL construct may appear.
`<xbrl duplicatesLegal="true"/>`	It is not an error for this instance to contain duplicates.
`<xbrl link:noRoleTypes="true"/>`	All `xlink:arcrole` and `xlink:role` attribute values not defined by XBRL will be ignored. See 5.5.2.1 below.
`<xbrl profile="http://ftb.gov.nv/xbrl/"/>`	At URI "http://ftb.gov.nv/xbrl/" a document published by the Franchise Tax Board of Nirvana describes XBRL features of income tax filings that their system will ignore: arcs in extension taxonomies with `"use=prohibited", any scenario` elements, and calculation link bases. Such restrictions cannot be expressed within the FTB or any other taxonomy.

4.2 The (deprecated) group element

The (deprecated) `group` element was the generic container element of the XBRL vocabulary. It can be the root element of XBRL instances. Other than not allowing the `profile` attribute, its content model is identical to the `xbrl` element. Consuming applications MUST process occurrences of the (deprecated) `group` element not at top level (which were allowed in XBRL 2.0) as if their child elements appeared in their place.

4.3 The item element

As discussed in section 3 above, an *item* represents a single fact or business measurement. In the XML Schema for XBRL instances, item is defined as an abstract element. This means that it will never appear in the text of an XBRL instance. Therefore, all elements representing facts or business measurements defined in an XBRL taxonomy document and reported in an XBRL instance MUST be either (a) members of the substitution group *item*; or, (b) members of a substitution group originally based on *item*. XBRL taxonomies include XML Schema-compliant schema documents that contain such element definitions.

There is no nesting of `item` elements. Structural relationships necessary in an XBRL document instance MUST be captured only using `tuple` elements. The intellectual structure – the relationship of financial concepts to each other in a variety of senses – is captured by the link structure of taxonomy linkbases rather than by nesting of facts in XBRL instances.

The XML Schema definition of the item element and the data types for elements in the item substitution group are given below.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<attributeGroup name="numericItemAttrs">
  <attribute name="numericContext" type="IDREF" use="required"/>
  <attribute name="precision" type="xbrli:precisionType" use="optional"/>
  <attribute name="decimals" type="xbrli:decimalsType" use="optional"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</attributeGroup>

<attributeGroup name="nonNumericItemAttrs">
  <attribute name="nonNumericContext" type="IDREF" use="required"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</attributeGroup>

<!--   ********item types******         -->
<!--                monetaryItemType            -->
<complexType name="monetaryItemType">
  <simpleContent>
    <extension base="xbrli:monetary">
      <attributeGroup ref="xbrli:numericItemAttrs"/>
    </extension>
  </simpleContent>
</complexType>
<!--                sharesItemType            -->
<complexType name="sharesItemType">
  <simpleContent>
    <extension base="xbrli:shares">
      <attributeGroup ref="xbrli:numericItemAttrs"/>
    </extension>
  </simpleContent>
</complexType>
<!--                pureItemType            -->
<complexType name="pureItemType">
  <simpleContent>
    <extension base="xbrli:pure">
      <attributeGroup ref="xbrli:numericItemAttrs"/>
    </extension>
  </simpleContent>
</complexType>

<!-- nonZeroNonInfiniteFloat   -->
<simpleType name="nonZeroNonInfiniteFloat">
  <annotation>
    <documentation>
As the name implies this is a float value that can not take the value ±0 or ±INF - it is
used as the type for the denominator of a fractionItemType
    </documentation>
  </annotation>
  <union>
    <simpleType>
      <restriction base="float">
        <minExclusive value="0"/>
        <maxExclusive value="INF"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="float">
        <minExclusive value="-INF"/>
        <maxExclusive value="-0"/>
      </restriction>
    </simpleType>
  </union>
</simpleType>

<!--                decimalItemType            -->
<complexType name="decimalItemType">
  <simpleContent>
    <extension base="decimal">
      <attributeGroup ref="xbrli:numericItemAttrs"/>
      <anyAttribute namespace="##other" processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>

<!--                 fractionItemType        -->
<complexType name="fractionItemType">
  <sequence>
    <element name="numerator" type="float"/>
    <element name="denominator" type="xbrli:nonZeroNonInfiniteFloat"/>
  </sequence>
  <attribute name="numericContext" type="IDREF" use="required"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</complexType>

<!--                stringItemType            -->
<complexType name="stringItemType">
  <simpleContent>
    <extension base="string">
      <attribute name="nonNumericContext" type="IDREF" use="required"/>
      <anyAttribute namespace="##other" processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
<!— booleanItemType, hexBinaryItemType, base64BinaryItemType, anyURIItemType, uriItemType,
QNameItemType, NOTATIONItemType, durationItemType, dateTimeItemType, timeItemType,
dateItemType, gYearMonthItemType, gYearItemType, gMonthDayItemType, gDayItemType,
gMonthItemType, normalizedStringItemType, tokenItemType, languageItemType,
NMTOKENItemType, NMTOKENSItemType, NameItemType, NCNameItemType, IDItemType,
IDREFItemType, IDREFSItemType, ENTITYItemType, ENTITIESItemType -->
<!-- ... -->
<!--                item           -->
<element name="item" abstract="true"/>

</schema>
```

Example 7. A numeric fact with three significant digits.

```
<ci:capitalLeases numericContext="c1" precision="3">727000</ci:capitalLeases>
```

Meaning: The value of Capital Leases in the numeric context labelled c1 is 727000 accurate to 3 decimal places. Note that it will be necessary to consult the numeric context (defined below) in order to determine other details concerning the value such as unit, period, etc.

Example 8. A non-numeric item.

```
<ci:concentrationsNote nonNumericContext="c1">
Concentration of credit risk with regard to short term investments is not considered to be
significant due to the Company's cash management policies. These policies restrict
investments to low risk, highly liquid securities (that is, commercial paper, money market
instruments, etc.), outline issuer credit requirements, and limit the amount that may be
invested in any one issuer.
</ci:concentrationsNote>
```

Meaning: The text of the Concentrations note in the non-numeric context labelled c1.

The content of the abstract `item` element is derived from `anyType`. Each member of the substitution group of `item` is a defined XBRL item type. This allows each substitution for `item` in the instance to validate against its own data type. There is one defined XBRL item type derived from each of the primitive types of XML Schema, along with the `fraction` type. The complete list is in section 5.3 below. An item can never have complex content unless its item type is derived by restriction from `fractionItemType`. Applications using the `noDerivations` profile MAY ignore items not derived from an XBRL defined item type.

The `numericContext` or `nonNumericContext` attribute is an `IDREF` to the appropriate kind of context element (see below) that holds additional relevant metadata about the fact represented. An item MUST contain a context attribute that references a context element in the same XBRL instance. Note that an XBRL instance is an occurrence of the `xbrl` element, not the entire document. Items whose type is derived from an XML Schema primitive numeric type (`decimal`, `float` or `double`) or `fractionItemType` by restriction MUST use the `numericContext` attribute; all others MUST use the `nonNumericContext` attribute. Two optional attributes, `precision` and `decimals`, are available on numeric item elements (except those with type `fractionItemType`) to enable the XBRL instance creator to make statements about the accuracy of the facts represented. They are discussed below.

4.3.1 The precision attribute (optional)

The optional precision attribute MUST be a non-negative integer or the string "`INF`" (from the value space of the XML Schema "`float`" type) that conveys the arithmetic precision of a measurement, and, therefore, the utility of that measurement to further calculations. Different software packages may claim different levels of accuracy for the numbers they produce. The precision attribute allows any producer to state the precision of the output in the same way. If a numeric fact has a precision attribute that has the value "n" then it is correct to "n" significant figures. (See section 4.3.5.1 for the normative definition of 'correct to "n" significant figures') The precision attribute on an element MUST take precedence over any value specified for the (deprecated) precision attribute on the numericContext for the element. The precision attribute is legal only on elements that have a type that requires a `numericContext` attribute except those with a type of `fractionItemType`. An application SHOULD ignore any digits after the first "*n*" decimal digits, counting from the left, starting at the first non-zero digit in the lexical representation of any number for which the value of precision is specified or inferred to be *n*.

The meaning of `precision="INF"` is that the lexical representation of the number is the exact value of the fact being represented.

Example 9. Precision and lexical representation

Example	Meaning
`precision="9"`	Precision of nine digits. The first 9 digits, counting from the left, starting at the first non-zero digit in the lexical representation of the value of the numeric fact are known to be trustworthy for the purposes of computations to be performed using that numeric fact.

Precision	Example	Read as	Known to be GE	Known to be LT
INF	476.334	476.334	476.334	476.33400000000…1
3	205	205e0	204.5	205.5
4	2002000	2002e3	2001500	2002500
2	2012	20e2	1950	2050
2	2000	20e2	1950	2050
1	99	9e1	85	95
0	1234	1234	unknown	unknown

The simple type `precisionType` has been provided to define the value space for the value of the precision attribute. Its definition is as follows:

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<simpleType name="precisionType">
  <annotation>
    <documentation>
This type is used to specify the value of the (deprecated) precision attribute on
numericContext and the precision attribute on numeric items. It consists of the union of
nonNegativeInteger and "INF" (used to signify infinite precision or "exact value")
    </documentation>
  </annotation>
  <union memberTypes="nonNegativeInteger"><simpleType>
      <restriction base="float">
        <enumeration value="INF"/>
      </restriction>
    </simpleType>
  </union>
</simpleType>
</schema>
```

4.3.2 The decimals attribute (optional)

The optional `decimals` attribute MUST be an integer or the value "`INF`" (from the value space of the XML Schema “`float`” type) that informs the document consumer of the number of decimal places to which the value of the fact represented may be considered accurate, possibly as a result of rounding or truncation. If a numeric fact has a `decimals` attribute with the value “n” then it is known to be correct to “n” decimal places. (See section 4.3.5.2 for the normative definition of ‘correct to “n” decimal places’). The decimals attribute is legal only on elements that have a type that requires a numericContext attribute except those with a type of `fractionItemType`.

The meaning of `decimals="INF"` is that the lexical representation of the number is the exact value of the fact being represented.

Example 10. Decimals and lexical representation

Example	Meaning
`decimals="2"`	The value of the numeric fact is known to be correct to 2 decimal places.
`decimals="-2"`	The value of the numeric fact is known to be correct to –2 decimal places, i.e. all digits prior to the hundreds digit are accurate.

Decimals	Original	Read as	Known to be GE	Known to be LT
INF	436.749	436.749	436.749	436.74900000…1
2	10.00	10.00	9.995	10.005
2	10	10.00	9.995	10.005
2	10.000	10.00	9.995	10.005
2	10.009	10.00	9.995	10.005
0	10	10.	9.5	10.5
-1	10	10.	9.5	10.5
-1	11	10.	9.5	10.5
3	205	205.000	204.9995	205.0005
4	2002000	2002000.0000	2001999.99995	2002000.00005
-2	205	200.	150	250
-2	2002000	2002000.	2001950	2002050
-3	2002000	2002000.	2001500	2002500
-4	2002000	2000000.	1995000	2005000
-3	777000	777000	776500	777500

The simple type `decimalsType` has been provided to define the value space for the value of the `decimals` attribute. Its definition is as follows:

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli=http://www.xbrl.org/2003/instance
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<simpleType name="decimalsType">
  <annotation>
    <documentation>
This type is used to specify the value of the decimals attribute on numeric items.
It consists of the union of integer and "INF" (used to signify that a number is expressed
to an infinite number of decimal places or "exact value")
    </documentation>
  </annotation>
  <union memberTypes="integer">
    <simpleType>
      <restriction base="float">
        <enumeration value="INF"/>
      </restriction>
    </simpleType>
  </union>
</simpleType>
</schema>
```

4.3.3 Inferring accuracy from the lexical representation of a fact's value

XBRL instance producers may wish to use input data that they know may have varying precision (if, for example, they have provided input data rounded to a certain number of decimal places). Under such circumstances they would have to determine the value of precision on a fact-by-fact basis if they wished to make a statement about it. Additionally for most business information, the situation is somewhat symmetric with respect to statements about the number of decimal places of accuracy. This is because when numbers are known and rendered to a certain number of significant figures (i.e. precision), the "`decimals`" value would vary according to the magnitude of the number (contrast 1234.567 with 1,234,567). Accordingly the following inference rules are provided to enable XBRL instance consumers to infer minimal information about the accuracy of the facts represented from their lexical representation.

If the precision attribute is not present and there is no (deprecated) `precision` attribute on the `numericContext` for that fact, then the XBRL instance makes no explicit statement about the precision of representation of the numeric facts for the purposes of future calculation. In such a case a consuming application MAY infer that the numeric fact is known to be correct to *n* significant figures where *n* is equal to the maximum of 0 and the result of the following calculation:

> if there are non-zero digits to the left of the decimal point or implied decimal point if absent then *the number of digits excluding any leading zeros to the left of the decimal point (or implied decimal point if absent) in the lexical representation of the numerical fact*
>
> otherwise *the negative of the number of zeros between the decimal point and the first non-zero digit in the lexical representation of the numerical fact*
>
> ***plus***
>
> *the value of the exponent in the lexical representation of the numerical fact (if present)*
>
> ***plus***
>
> *the number of decimal places to which the numeric fact is known or inferred to be correct.*

Example 11. Lexical representation, precision and decimals

Lexical Representation	Value of the decimals attribute	Inferred value of the precision attribute
123	2	3+2=5
123.4567	2	3+2=5
123e5	-3	3+5+(-3)=5
0.1e-2	5	0+(-2)+5=3
0.001E-2	5	(-2)+(-2)+5=1
0.001e-3 (this is a pathological case)	4	(-2)+(-3)+4=-1 which is less than 0 and hence 0

Similarly the `decimals` attribute is optional. For any given numeric fact, where the decimals attribute is omitted, a consuming application MAY infer a value of its decimals attribute based on the lexical representation of the value of the numeric fact.

Table 2. Inferring a value for the decimals attribute.

Lexical Representation Used	Other condition(s)	Value of the decimals attribute	Example(s)	
			Lexical representation	Value of the decimals attribute
Mantissa followed by the letter 'E' or 'e' followed by an exponent	The mantissa has digits to the right of the decimal point	The negative of the exponent plus the number of digits to the right of the decimal point in the mantissa	123E4 123.0E4 123e-4 123.0e-4	-4 -4+1=-3 4 4+1=5

Lexical Representation Used	Other condition(s)	Value of the decimals attribute	Example(s)	
			Lexical representation	Value of the decimals attribute
Mantissa followed by the letter 'E' or 'e' followed by an exponent	The mantissa has no digits to the right of the decimal point	The negative of the exponent plus the negative of the number of consecutive zeros immediately to the left of the decimal point in the mantissa	1230E4 1230.e4 123E-4 1230e-4	-4-1=-5 -4-1=-5 4 4-1=3
Decimal representation (i.e. no exponent)	The representation has digits to the right of the decimal point	The number of digits to the right of the decimal point	123.456 123.40	3 2
Decimal representation (i.e. no exponent)	The representation has no digits to the right of the decimal point	The negative of the number of consecutive zeros immediately to the left of the decimal point	12300 123. 123 1230.	-2 0 0 -1

The inferences made are only about the accuracy of individual numeric facts and not about the numeric context of those facts. In the event neither precision nor decimals is specified then any inference about the numeric accuracy of the fact MUST be made by inferring a value for decimals first and then by inferring a value for precision, which will be based on the previously inferred value for decimals.

4.3.4 Effect of precision and decimals on item equality and inconsistency

It is an error if, given a specified value of `decimals` for that fact, the value that would be *inferred* for `precision` of that item is greater than the value of `precision` actually specified for that fact, if any.

For any value of the `precision` attribute, it is possible to derive the lexical representation of a number for any *lower* precision. Two numeric items are equal if their least precise common lexical representations are identical. If a value for the `decimals` attribute must be inferred because it is not present, that value should be inferred first, before inferring a value for `precision` and thus a lexical representation. This is relevant in determining equivalency for purposes of the alias-essence relationship discussed in 4.3.6 below.

Two numeric items **A** and **B** are v-equal if and only if they have the identical XBRL canonical lexical representation when rounded to **N** significant figures where **N** is the lowest of:

i. the specified precision for the **A** if precision is specified for **A**
ii. the specified precision for **B** if precision is specified for **B**
iii. the inferred precision for **A** if precision is not specified **A**
iv. the inferred precision for **B** if precision is not specified for **B**

The canonical lexical representation to be used for comparison of values from different XML Schema or XBRL defined types may be determined by reference to the following table. Except where otherwise stated the XBRL canonical representation for a value of a particular XML Schema type is identical to the XML Schema canonical representation of that type.

Table 3. XBRL canonical lexical representations for use in determining numeric equivalency

Value space (or subset thereof) for first item, **A**	Value space (or subset thereof) for second item, **B**	Canonical forms of lexical representation to be used for comparison to determine whether **A** and **B** have the same value
`integer`	`decimal`	Since integer is derived from decimal its value space is a subset of that for decimal. Therefore the XML Schema canonical representation for decimal as defined at http://www.w3.org/TR/xmlschema-2/#decimal - section 3.2.3.2
`decimal`	`float`	**A** SHALL be mapped to the value space of **B** according to http://www.w3.org/TR/xmlschema-2/#float and the XML Schema canonical representation for float as defined in section 3.2.4.2 therein
`decimal`	`double`	**A** SHALL be mapped to the value space of **B** according to http://www.w3.org/TR/xmlschema-2/#double and the XML Schema canonical representation for float as defined in section 3.2.5.2
`float`	`double`	For **A**, the XML Schema canonical representation for float according to section 3.2.4.2 at http://www.w3.org/TR/xmlschema-2/#float and for **B**, the XML Schema canonical representation for double according to section 3.2.5.2 at http://www.w3.org/TR/xmlschema-2/#double
`fraction`	`fraction`	The numerator and denominator of both **A** and **B** shall be converted into the XBRL canonical lexical representation for the fraction type as defined below. For **A** to be v-equal to **B** the resulting numerator of **A** must be identical to that of **B** and the resulting denominator of **A** must be identical to that of **B**. Note that no consideration need be given to precision since fractions are always considered to be of infinite precision.
`fraction`	any other type	The numerator and denominator of **A** SHALL be converted to the XBRL canonical lexical representation for the fractionItemType type as defined below. The numerator SHALL then be divided by the denominator according to the rules defined in [IEEE]. Since the resulting value may fall outside the value space for the XML Schema float type but, nevertheless, in the value space for the XML Schema double type, treat **A** as a double with this value and apply the above rules accordingly.

The XBRL canonical lexical representation of the numerator and denominator of `fractionItemType` items SHALL be obtained from the original lexical representation of the item as follows:

i. convert both the numerator and the denominator into their XML Schema canonical representations as floats according to section 3.2.4.2 at http://www.w3.org/TR/xmlschema-2/#float
ii. subtract N from the exponent of both the numerator and denominator where N is the lower of the two exponent values.
iii. express the resulting values in their XML Schema canonical representations as floats according to section 3.2.4.2 at http://www.w3.org/TR/xmlschema-2/#float

It should be noted that the above process does not necessarily conform to the rules or intent of floating-point arithmetic according to [IEEE], nor produce the same results, but is specified in this way because of the intended use for fractionItemType in XBRL. It is intended, for example, that 1.23E4/5.67E8 has the same value as 1.23E0/5.67E4 regardless of whether floating-point division would produce that result.

4.3.5 Definitions

The following definitions are provided for clarity regarding accuracy related features of this specification, i.e. `precision` and `decimals` attributes.

4.3.5.1 "Correct to *n* Significant Figures", "Rounding" and "Truncation"

If a number is said to be correct to *n* significant figures it means that the first "*n*" decimal digits, counting from the left, starting at the first non-zero digit in the lexical representation of the number are known to be accurate for the purposes of computations to be performed using that number. (Note: in the following it is assumed that all zeros to the left of the decimal point and to the left of the first non-zero digit in the decimal representation have been removed first).

More precisely: in the decimal representation of a number, a significant figure is any one of the digits 1, 2, 3... 9 that specify the magnitude of a number. Zero (0) is a significant figure except when it appears to the left of all non-zero digits or is used solely to fill the places of unknown or discarded digits (after truncation or rounding - see later). Thus, in the number "0.00263" there are three significant figures: 2, 6, and 3. The zeroes are not significant. In the number "3809" all four of the digits are significant. In the number "46300" the digits 4, 6, and 3 are known to be significant but it is not possible to conclude anything concerning the two zeroes as they are written. This ambiguity can be removed by writing the number in terms of powers of ten. If there are three significant figures the representation becomes 4.63×10^4; if there are four significant figures it becomes 4.630×10^4, etc.

It is often necessary to round significant figures following a calculation. This is known as **rounding**. To round a number to *n* significant figures, discard all digits to the right of the *n*th place. This step is known as **truncation**. Then, if the left-most discarded digit is less than 5, leave the *n*th digit unchanged; if the left-most discarded digit is greater than or equal to 5, add 1 to the nth digit (propagating any carries to digits further to the left according to the normal rules of arithmetic and removing the final 0 if necessary). For example:

Example 12. Rounding

Original	Rounded to *n* significant figures	
	n=2	*n=3*
3.5643	3.6	3.56
3.5673	3.6	3.57
0.49787	0.50	0.498
3.9999	4.0	4.00
9.999991	10	10.0
22.55	23	22.6
0.0019	0.0019	0.00190
0.00002	0.00020	0.0000200

The same procedure MAY be followed for any value of *n*, and we then say that a particular number is ***correct to n significant figures***.

4.3.5.2 "Correct to *n* Decimal Places"

If the representation of a number is ***correct to n decimal places*** then the absolute difference between the value of the number and its representation (known as the "absolute error" of the representation - e_{abs}) is less than or equal to $0.5 \text{ x } 10^{-n}$. *n* may be a positive or negative integer or zero.

Mathematically this may be expressed as follows:

For the number X, x is a representation of X correct to *n* decimal places if and only if

$$e_{abs} = |X-x| \le 0.5 \times 10^{-n}$$

or, because of rounding conventions,

$$-0.5 \times 10^{-n} \le x-X < 0.5 \times 10^{-n}$$

Rounding, as described earlier, can be used to make a number correct to exactly *n* decimal places. The following table shows the representations of the number 123456.789012 correct to various numbers of decimal places:

Example 13. Correct to *n* decimal places

123456.789012 correct to *n* decimal places				
n=-3	*n*=-2	*n*=0	*n*=3	*n*=6
123000	123500	123457	123456.789	123456.789012

4.3.6 Item equivalence

It is an error for two items in an XBRL instance to be duplicate items when the profile attribute `duplicatesLegal` is `false`. XBRL forbids duplicate items by default not only to avoid *redundancy* but also to avoid *inconsistency.* The concept of item equivalence for this purpose is one of *equal location*, not *equal content.* Attributes other than `numericContext, nonNumericContext, precision` and `decimals` are ignored for the purposes of this comparison. Additional “id” attributes or attributes from other namespaces do not distinguish otherwise equal items. Items may be c-equal, v-equal, both, or neither. Definitions of c-equal and v-equal appear in Table 1. Whether elements appear within a tuple impacts whether they are duplicated (see 4.5.3 below).

4.4 The context elements numericContext and nonNumericContext

The context elements `numericContext` and `nonNumericContext` are the holders of various metadata in attributes and element content that provide the necessary context for understanding a business fact captured in an XBRL item. The `nonNumericContext` element contains metadata and attributes relating to textual, date, URI, Boolean, string and other types of non-numeric facts; the `numericContext` provides similar elements for numeric facts.

The context elements MUST conform to the following definitions:

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!--               nonNumericContext          -->
  <element name="nonNumericContext">
    <complexType>
      <sequence>
        <element name="entity" type="xbrli:entityType"/>
        <element name="period" type="xbrli:periodType"/>
        <element name="unit" type="xbrli:unitType" minOccurs="0"/>
        <element ref="xbrli:scenario" minOccurs="0"/>
      </sequence>
      <attribute name="id" type="ID" use="required"/>
    </complexType>
  </element>
  <!--               numericContext          -->
  <element name="numericContext">
    <complexType>
      <sequence>
        <element name="entity" type="xbrli:entityType"/>
        <element name="period" type="xbrli:periodType"/>
        <element name="unit" type="xbrli:unitType"/>
        <element ref="xbrli:scenario" minOccurs="0"/>
      </sequence>
      <attribute name="id" type="ID" use="required"/>
      <attribute name="precision" type="xbrli:precisionType" use="optional"/>
      <attribute name="cwa" type="boolean" use="required"/>
    </complexType>
  </element>
</schema>
```

Each attribute is described separately below.

In the examples, the xsi:schemaLocation attribute does not contain URIs to resolve the ISO4217 and NASDAQ namespaces. The examples assume that the applications that produced and will consume this instance will be able to resolve these namespace references without the help of the xsi:schemaLocation. The URIs given are examples only; they do not reference actual resources of the ISO or NASDAQ.

4.4.1 The id attribute

Every context element MUST include this attribute. The content MUST be in accordance with the XML Schema rules for attributes with the ID type. The id attribute identifies the context (see 4.3 above) of the item.

Example 14. IDs

Example	id="C2424"	
Counterexample	id="42"	The ID type must not begin with a number.

4.4.2 The period element

The period element contains the instant or interval of time for reference by an item. The sub-elements of period are used to construct one of the allowed choices for representing date intervals.

Elements	Meaning
startDate, endDate	A period beginning and ending as specified.
instant	A point in time.
forever	An element to represent 'forever'.

The sub-elements of period MAY include the following deprecated alternatives that might be eliminated from future XBRL specifications.

Elements	Meaning
startDate, (deprecated) duration	A period beginning as specified, for the given length of time.
(deprecated) duration, endDate	A period of the given length of time, ending as specified.

Each of the period sub-elements uses a standard XML Schema representation of a date or duration.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <simpleType name="dateUnion">
    <union memberTypes="date dateTime"/>
  </simpleType>
  <!--                startDate            -->
  <element name="startDate" type="xbrli:dateUnion"/>
  <!--                endDate            -->
  <element name="endDate" type="xbrli:dateUnion"/>
  <!-- (deprecated) duration -->
  <element name="duration" type="duration"/>
  <!--                instant            -->
  <element name="instant" type="xbrli:dateUnion"/>
  <!--                forever            -->
  <element name="forever">
    <complexType/>
  </element>
  <!-- periodType            -->
  <complexType name="periodType">
    <choice>
      <sequence minOccurs="0">
        <element ref="xbrli:startDate"/>
        <choice>
          <element ref="xbrli:endDate"/>
          <element ref="xbrli:duration"/>
        </choice>
      </sequence>
      <sequence minOccurs="0">
```

```
        <element ref="xbrli:duration"/>
        <element ref="xbrli:endDate"/>
      </sequence>
      <element ref="xbrli:instant" minOccurs="0"/>
      <element ref="xbrli:forever" minOccurs="0"/>
    </choice>
  </complexType>
</schema>
```

Sub-element	XML Schema data type
instant	date or dateTime.
forever	empty
startDate	date or dateTime
endDate	date or dateTime
(deprecated) duration	duration

While the content of the instant, startDate, endDate and (deprecated) duration elements are defined to use the data representation defined by ISO 8601 (as restricted by XML Schema Part 2, Datatypes), XBRL adds further restrictions and constraints.

For an element with instantaneous="true", the period MUST contain an instant element.

For an element with instantaneous="false", the period MUST contain forever or a valid combination of startDate, endDate., and (deprecated) duration.

A (deprecated) duration element MUST NOT contain duration of length zero, such as P0Y, P0M, P0D, etc. It is a fatal error for duration to be negative.

Specifying a date (which has no time part) in the content of an instant or startDate elements is defined to be equivalent to specifying a dateTime of the same date, and T00:00:00 (midnight at the beginning of the day).

Specifying a date in the endDate element is defined to be equivalent to specifying a dateTime of the same date, and T24:00:00:00 (midnight at the end of the day).

It is a fatal error for the endDate to contain a value that is not later than the value of startDate.

4.4.3 The unit element

The unit element specifies the standard that is relevant to the measurement. Many measurements will be monetary measurements. ISO standard currency designation [ISO] MUST be used for the unit element in such a case. Other measurements based on shares MUST use the shares item type. Growth rates, percentages and ratios where the numerator and denominator are in the same units MUST use the pure item type. Counts of employees, square footage and the like MUST be specified as using a numeric item type, but SHOULD have a numeric context with a more specific unit measure such as “FTE” or “ft*ft” and MAY have the unit measure “decimal”.

The unit sub-element is required in numericContext.

The unit MUST contain a single measure or several measures combined with the multiply and divide operators.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!-- measure element             -->
  <element name="measure" type="QName"/>
  <!-- operatorNameEnum            -->
  <simpleType name="operatorNameEnum">
```

```
    <restriction base="string">
      <enumeration value="multiply"/>
      <enumeration value="divide"/>
    </restriction>
  </simpleType>
  <!-- operator element            -->
  <element name="operator">
    <complexType>
      <choice minOccurs="2" maxOccurs="2">
        <element ref="xbrli:measure"/>
        <element ref="xbrli:operator"/>
      </choice>
      <attribute name="name" type="xbrli:operatorNameEnum" use="required"/>
    </complexType>
  </element>
  <!-- unitType            -->
  <complexType name="unitType">
    <choice minOccurs="1" maxOccurs="1">
      <element ref="xbrli:measure"/>
      <element ref="xbrli:operator"/>
    </choice>
  </complexType>
</schema>
```

Example 15. Use of the unit element.

Example	Meaning
`<unit><measure>ISO4217:GBP</measure></unit>`	Currency, UK Pounds.
`<unit><measure>xbrli:pure</measure></unit>`	A pure number, such as % revenue change.
`<unit><measure>xbrli:shares</measure></unit>`	A number of shares.

`<unit>`
`    <operator name="divide">`
`       <measure>ISO4217:EUR</measure>`
`       <measure>xbrli:shares</measure>`
`    </operator>`
`</unit>`
	Earnings per share (EPS) measured in Euros per share

Since `measure` uses Qualified Names, the prefixes in these examples will have been previously defined in namespace declarations.	

4.4.4 The (deprecated) precision attribute on numericContext

The optional, deprecated `precision` attribute of `numericContext` has the same meaning as the `precision` attribute on numeric items as defined in 4.3.1 above. The `precision` attribute on an element MUST take precedence over any value specified for the (deprecated) `precision` attribute on the `numericContext` for the element.

4.4.5 The cwa attribute

The `cwa` attribute MUST be a Boolean value indicating the validity of the ‘closed world assumption’. If `cwa="true"` then the ‘closed world assumption’ MUST be valid over the set of items that refer to this context and consuming applications MAY assume that there is no information relevant to the calculation missing from the provided data. For a context with `cwa="false"` a consuming application MUST NOT attempt to calculate any new fact based on the information given.

Calculations that can take advantage of the `cwa` attribute are specified in various linkbases, which are referenced either directly or indirectly from the XBRL instance. An example is the calculation linkbase that may appear as part of the taxonomy from which items are drawn.

Example 16. The closed world assumption and calculations.

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:ci="http://www.xbrl.org/us/gaap/ci/2003/us-gaap-ci-2001">

  <ci:intangibles numericContext="c1">7</ci:intangibles>
  <ci:cash numericContext="c1">2</ci:cash>
  <ci:notesPayable numericContext="c1">5</ci:notesPayable>
  <ci:bondsPayable numericContext="c1">1</ci:bondsPayable>
  <ci:treasuryStock numericContext="c1">2</ci:treasuryStock>
  <ci:commonStock numericContext="c1">1</ci:commonStock>
  <numericContext id="c1" cwa="true">...</numericContext>
</xbrl>
```

Abbreviated XBRL instance.

```
<calculationArc
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:from="intangibles" xlink:to="assets" weight="1"/>
<calculationArc
  xlink:arcrole=”http://www.xbrl.org/2003/role#item-summation"
  xlink:from="cash" to="assets" weight="1"/>
<calculationArc
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:from="notesPayable" xlink:to="liabilities" weight="1"/>
<calculationArc
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:from="bondsPayable" xlink:to="liabilities" weight="1"/>
<calculationArc
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:from="treasuryStock" xlink:to="equity" weight="1"/>
<calculationArc
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:from="commonStock" xlink:to="equity" weight="1"/>
```

Arcs in the calculation linkbase of the taxonomy referred to by the `ci:` prefix in the previous XBRL instance.

```
<xbrl xmlns:ci="http://www.xbrl.org/us/gaap/ci/2003/us-gaap-ci-2001"
  <ci:assets numericContext="c1">9</ci:assets>
  <ci:liabilities numericContext="c1">6</ci:liabilities>
  <ci:equity numericContext="c1">3</ci:equity>
  <numericContext id="cl" cwa="true"/>
</xbrl>
```

New facts an XBRL processor SHOULD derive.

4.4.6 The entity sub-element

The `entity` element documents the organisation (business, government department, individual, etc.) that business fact describes. The `entity` element is required content of the context element. The `entity` element MUST contain an `identifier` element and MAY contain an optional `segment` element.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<!-- entityType -->
<complexType name="entityType">
  <sequence>
    <element name="identifier" minOccurs="1" maxOccurs="1">
      <complexType>
        <simpleContent>
          <extension base="token">
            <attribute name="scheme" type="anyURI" use="required"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
    <element ref="xbrli:segment" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
<!--                end of context/entityType           -->
</schema>
```

4.4.6.1 identifier

An identifier element specifies a scheme for identifying business entities. The required scheme attribute contains the namespace URI of the identification scheme, providing a framework for referencing naming authorities. The element content MUST be a token that is a valid identifier within the namespace referenced by the scheme attribute. XBRL International is not a naming authority for business entities. XBRL makes no assumption about the ability of an application to resolve an identifier that may appear as element content in any particular scheme.

Example 17. Entity identifiers.

Example	Meaning
`<identifier scheme="">SAMP</identifier>`	Some entity known only as SAMP within the default namespace.
`<identifier scheme="www.nasdaq.com">SAMP</identifier>`	The company with NASDAQ ticker symbol SAMP.
`<identifier scheme="www.dnb.com">121064880</identifier>`	The company or subsidiary with D-U-N-S® number 121064880.
`<identifier scheme="www.cusip.org">41009876AB</identifier>`	The entity with CUSIP number 41009876AB (e.g. a mutual fund).
`<identifier scheme="www.ietf.org/URI">www.w3c.org</identifier>`	The non-profit organisation owning the URI www.w3c.org.

4.4.6.2 segment

The segment element is an optional container for additional mark-up that the preparer of an XBRL instance MAY use to identify the business segment more completely. In general, the content of a segment will be application specific.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="segment">
    <complexType>
      <choice minOccurs="0" maxOccurs="unbounded">
        <any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
      </choice>
      <attribute name="name" type="token" use="optional"/>
    </complexType>
  </element>
</schema>
```

Example 18. Using the segment element.

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:my="http://www.someCompany.com/scenarios">

<nonNumericContext id="c1" >
  <entity>
    <!—required content -->
    <identifier scheme="www.dnb.com">121064880</identifier>
  <!-- optional content -->
    <segment>
      <my:stateProvince>MI</my:stateProvince>
    </segment>
  </entity>
</nonNumericContext>
</xbrl>
```

```
<!-- Company specific segment sub-element -->
<schema targetNamespace="http://www.someCompany.com/scenarios"
        xmlns:my="http://www.someCompany.com/scenarios"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <simpleType name="stateProvinceType">
    <restriction base="token">
      <enumeration value="MI"/>
      <enumeration value="ON"/>
    </restriction>
  </simpleType>

  <element name="stateProvince" type="my:stateProvinceType"/>
</schema>
```

Meaning: The preparer has used a `<segment>` to indicate that the business facts relate to operations in the state of Michigan. The company's own XML Schema defines the `stateProvince` element as including just Michigan and Ontario.

Creators of taxonomies should anticipate that XBRL instance creators will define elements to insert in the segment element to represent one or more dimensions of distinction such as:

- Organisational structure, such as a the corporate headquarters and individual subsidiaries of an entity;
- Regional decomposition, such as operations in Asia, Europe, and North America;
- Functional distinctions, such as results from continuing and discontinued operations;
- Product distinctions, such as operations relating to fishing, forestry and farming;
- Operational distinctions such as recurring vs. non-recurring revenues or new subscriptions vs. renewals.

It is up to the preparer of the document to provide the proper namespace support and `xsi:schemaLocation` hints necessary to ensure that the `segment` element is properly validated by an XML Schema validation process.

4.4.7 The scenario sub-element

Business facts can be reported as actual, budgeted, restated, pro forma, etc. For internal reporting purposes, there can be an even greater variety of additional metadata that preparers want to associate with items. The optional `scenario` element allows additional valid mark-up (see note above regarding segment) to be included for this purpose.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <!--              context/scenario element           -->
  <!--              scenarioItem           -->
  <element name="scenario">
    <complexType>
      <choice minOccurs="0" maxOccurs="unbounded">
        <any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
      </choice>
      <attribute name="name" type="string" use="optional"/>
    </complexType>
  </element>
  <!--              end of context/scenario element           -->
</schema>
```

Example 19. Use of the scenario element.

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:fid="http://www.someInsuranceCo.com/scenarios"
      xmlns:other="http://www.example.com">
<nonNumericContext id="c1" >
   <scenario>
     <other:bestEstimate/>
     <fid:dwSlice>
       <fid:residence>MA</fid:residence>
       <fid:nonSmoker>true</fid:nonSmoker>
       <fid:minAge>34</fid:minAge>
       <fid:maxAge>49</fid:maxAge>
     </fid:dwSlice>
   </scenario>
</nonNumericContext>
</xbrl>
```

Meaning: The preparer has used a <scenario> to indicate that the reported values relate to a "best estimate" scenario for non-smokers residing in Massachusetts of the specified age group.

It is up to the preparer of the instance to provide the proper namespace support and `xsi:schemaLocation` hints necessary to ensure that the `scenario` element is properly validated by an XML Schema validation process.

The scenario and segment sub-elements have exactly the same structure, but are used for two different purposes. Segment is used to specify some component of the business entity. Scenario is used to document the circumstances surrounding the measurement of a set of facts, and like the segment element, its content will be application specific.

Creators of business reporting taxonomies should anticipate that XBRL instance creators will define elements to insert in the `scenario` element to represent dimensions of distinction such as:

- Assuming certain valuations of assets or future revenue streams;
- Actual, adjusted, estimated, forecasted, or reported as of a certain date;
- Assuming a particular foreign currency exchange rate.

4.4.8 Context equivalence

It is an error for XBRL instances to contain duplicate items, the definition of which depends on detecting s-equal contexts:

- Structure-equal (s-equal) contexts have s-equal cwa attributes, period elements, entity elements, unit elements and scenario elements (with any missing `scenario` treated as s-equal to an empty `scenario` element and any missing `segment` treated as s-equal to an empty `segment` element).

Under the closed world assumption (see 4.4.5 above), applications that consume XBRL instances MAY use information expressed in linkbases to warrant certain inferences drawn from items appearing in those instances, when the sub-elements of their contexts are s-equal or match patterns expressed as relative or absolute contexts.

4.5 The tuple element

While most business facts can be independently understood, some facts are dependent on each other for proper understanding, especially if multiple occurrences of that fact are being reported. For example, in reporting the management of a company, each manager’s name has to be properly associated with the manager’s correct title. Such sets of facts (manager’s title/manager’s name) are called “tuples”.

Tuples have complex content and MAY contain both items and other tuples. Like the `item` element, the `tuple` element is abstract. All tuple elements MUST be members of the substitution group that has `tuple` as its head. Therefore, all item and tuple elements must be declared globally, because only global elements can be in a substitution group. Also, to be in the substitution group of tuple, the tuple elements must have a type that is derived from `tupleType`. The type `tupleType` is final with respect to extension, so that all tuples MUST be derived only by restriction directly from `tupleType` (*not* by restriction from any type derived from `tupleType`); tuples defined in a taxonomy cannot have mixed content, simple content, and any additional attributes must be drawn from a non-XBRL namespace. Tuple elements MUST NOT be declared `abstract`; sub-elements of the tuple MUST NOT contain any anonymous type declarations; and the `restriction` element of the tuple declaration MUST NOT contain elements other than `sequence`, `choice`, `any`, and `element`.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<complexType name="tupleType" final="extension">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="xbrli:item" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="xbrli:tuple" minOccurs="0" maxOccurs="unbounded"/>
  </choice>
  <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
<element name="tuple" type="xbrli:tupleType" abstract="true"/>
</schema>
```

Example 20. Defining a tuple element, a restriction of tupleType

An abbreviated example Taxonomy Schema document:

```
<schema targetNamespace="http://mycompany.com/xbrl/taxonomy"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xbrli="http://www.xbrl.org/2003/instance" >

  <element name="managementName" type="xbrli:tokenItemType"
           substitutionGroup="xbrli:item"/>
  <element name="managementTitle" type="xbrli:tokenItemType"
           substitutionGroup="xbrli:item"/>
  <element name="managementTitle" type="xbrli:nonNegativeIntegerItemType"
           substitutionGroup="xbrli:item"/>

  <element name="managementInformation" substitutionGroup="xbrli:tuple">
    <complexType>
      <complexContent>
        <restriction base="xbrli:tupleType">
          <sequence>
            <element ref="s:managementName"/>
            <element ref="s:managementTitle"/>
            <element ref="s:managementAge" minOccurs="0"/>
          </sequence>
        </restriction>
      </complexContent>
    </complexType>
  </element>
</schema>
```

An XBRL instance of the taxonomy (`nonNumericContext` elements and `linkbaseRef` elements not shown):

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:s="http://mycompany.com/xbrl/taxonomy">

  <s:managementInformation>
    <s:managementName nonNumericContext="c1">Haywood Chenokitov</s:managementName>
    <s:managementTitle nonNumericContext="c1">President</s:managementTitle>
    <s:managementAge nonNumericContext="n1">42</s:managementAge>
  </s:managementInformation>
  <s:managementInformation>
    <s:managementName nonNumericContext="c1">Miriam Minderbender</s:managementName>
    <s:managementTitle nonNumericContext="c1">CEO</s:managementTitle>
  </s:managementInformation>
</xbrl>
```

Sequence and choice constructs can appear in tuples. For example, consider information that is disclosed in tax filings regarding real estate and other properties:

Example 21. Elements describing business properties held and disposed.

Label	Element Name	Balance	Substitution Group
Property	`property`		tuple
Property description	`description`		item
Date property acquired	`dateAcquired`		item
Date property disposed of	`dateDisposedOf`		item
Property fair market value	`fairMarketValue`		item

Although the description and date acquired are relevant for any property, the property is either has a fair market value or has already been disposed of, but not both.

Example 22. Hierarchy in a tuple

```
property
   |
   +-- description
   |
   +-- dateAcquired
   |
   +-- [choice]
          |
          +-- dateDisposedOf
          |
          +-- fairMarketValue
```

Example: tuples associate concepts cannot be understood independently and repeat within an XBRL instance, usually with variations either within a taxonomy, with variations as a result of taxonomy extensions.

In general, tuple elements do not differ from item elements with respect to the linkbases presented elsewhere in this specification. Tuple elements can participate in definition and presentation links. Labels and references can be created for tuple elements.

One exception is the calculation linkbase; a tuple element cannot participate in a calculation link, because it does not have a numeric value. Another exception is the definition linkbase, as described below.

4.5.1 Tuple specialisation and inheritance

It is useful to specialise tuples and inherit properties to those specialisations, just as in XML Schema and other data languages define types and use restriction to specialise them. A tuple concept connected to another tuple concept by a definition arc with arc role `http://www.xbrl.org/2003/role#general-special` indicates a form of inheritance; the occurrence in an XBRL instance of the concept on the `special` end MAY be treated as an occurrence of the concept on the `general` end with respect to arcs in the label, reference and presentation linkbase. More precisely, if tuple **S** is a specialisation of tuple **G**, and arc(e,**G**,x) is an arc from (to) **G** in an extended-type link e, where e is a label, reference or presentation link, then an application processing MAY infer the presence of a corresponding arc(e,**S**,x) for any purpose other than the detection of cycles or duplicates.

Example 23. Interpretation of the general-special link for tuples

The concept `managementInformation`, a restriction of `tupleType`, has these arcs:

1. Presentation content-container to `companyInfo`
2. Presentation container-content to `managementName`
3. Presentation container-content to `managementTitle`
4. Presentation container-content to `managementAge`.

The concept `moreManagementInfo`, also a restriction of `tupleType`, has these arcs:

5. presentation container-content to `managementSalary`
6. definition special-general to `managementInfo`

For purposes of presentation, the presence of arc 6 means that the content of the concept `moreManagementInfo` includes all of the arcs 1 through 5.

Example: Specialising a tuple allows arcs to be inferred.

4.5.2 Tuple equivalence

No two tuples in an XBRL instance may have the same structure and content under a common parent. The parent is the `tuple`, `xbrl` or (deprecated) `group` element that is the immediate outer container of both tuple elements in the XBRL instance; in XPath terminology, the first and only node on the parent axis. This does not refer to any relationship expressed in a linkbase among taxonomy elements.

4.5.3 Duplicate contexts, items, and tuples

There are several different senses of "equal" that are relevant to duplication in XBRL instances: Structure equal (s-equal), Parent equal (p-equal), Value equal (v-equal), and Context equal (c-equal) as defined in Table 1. These different equality predicates are polymorphic and formally defined in a mutually recursive fashion.

Table 4. Equality predicate definitions.

Argument Types	Predicates	Definition
`node`	identical	Exactly the same XML node.
`sequence`	s-equal, v-equal, c-equal	Every node in the sequence is {s-equal, v-equal, c-equal} to the node in the same position in the other sequence.
`set`	s-equal, v-equal, c-equal	Every node in the set is {s-equal, v-equal, c-equal} to exactly one node in the other set.
`text`	s-equal	The XPATH '=' operator returns 'true'
`attribute`	s-equal	The XPATH '=' operator returns 'true'
`element`	s-equal	Not identical, and the text of their element names are s-equal, and the set of their attributes are s-equal, and the sequence of text and sub-element contents are s-equal.
`entity`	s-equal	`identifier` elements are s-equal, and `segment` elements are s-equal (with any missing `segment` treated as s-equal to an empty `segment` element).
`context`	s-equal	`cwa` attributes are equal, and `period` elements are equal, and `entity` elements are equal, and `unit` elements are equal, and `scenario` elements are s-equal (with any missing `scenario` treated as s-equal to an empty `scenario` element).
`item`	s-equal	`numericContext` attributes are s-equal, and `nonNumericContext` attributes are s-equal, and `precision` attributes are s-equal, and `decimals` attributes are s-equal, and text is s-equal.
`tuple`	s-equal	The set of immediate (`item` and `tuple`) children are s-equal.
`item`	p-equal	Nodes are children of the identical parent.
`tuple`	p-equal	Nodes are children of the identical parent.
`item`	c-equal	not s-equal, and the text of their `nonNumericContext` attributes identify `nonNumericContexts` that are identical or s-equal, and the text of their `numericContext` attributes identify `numericContexts` that are identical or s-equal.
`numeric item`	v-equal	Not identical, and c-equal, and applying the precision and decimals attributes to the lexical representation of one result in text that is s-equal to the other – see section 4.3.4 above.
`non numeric item`	v-equal	Not identical, and c-equal.

item	duplicate	s-equal element names, and p-equal, and c-equal.
tuple	duplicate	s-equal element names, and p-equal, and their sets of child tuples are duplicate tuples, and their sets of child items are v-equal.

The following extended example illustrates positive and negative examples of each of the above predicates

Example 24. Duplicate items, tuples and contexts

element	An XBRL instance containing three contexts (two of which are duplicates of each other) and doubly nested tuples. Several of the elements are named in the left column.

```
                <xbrl xmlns="http://www.xbrl.org/2003/instance"
                      xmlns:s="http://mycompany.com/xbrl/taxonomy"
                      xmlns:xbrli="http://www.xbrl.org/2003/instance"
                      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    a analysis  <s:analysis>
    b customer    <s:customer>
        b name      <s:name nonNumericContext="nnp3">Acme</s:name>
       b gross      <s:gross numericContext="np3" precision="4">3001</s:gross>
     b returns      <s:returns numericContext="np3">100</s:returns>
                    <s:net numericContext="np3">2900</s:net>
                  </s:customer>
    c customer    <s:customer>
        c name      <s:name nonNumericContext="Xnnp3X">Acme</s:name>
       c gross      <s:gross numericContext="np3" precision="3">3000</s:gross>
                    <s:returns numericContext="np3">100</s:returns>
                    <s:net numericContext="np3">2900</s:net>
                  </s:customer>
    d customer    <s:customer>
                    <s:name nonNumericContext="nnp3">Acme</s:name>
                    <s:gross numericContext="np3">3000</s:gross>
     d returns      <s:returns numericContext="np3">500</s:returns>
                    <s:net numericContext="np3">2500</s:net>
                  </s:customer>
    e customer    <s:customer>
        f name      <s:name nonNumericContext="nnp3">Bree</s:name>
        g name      <s:name nonNumericContext="Xnnp3X">Bree</s:name>
                    <s:gross numericContext="np3">3000</s:gross>
                    <s:returns numericContext="np3">200</s:returns>
                    <s:net numericContext="np3">2800</s:net>
                  </s:customer>
  h totalGross   <s:totalGross numericContext="np3">6000</s:totalGross>
                </s:analysis>

                <!-- Two Contexts chosen from {n,nn}{p,i}{2,3} -->
                <!-- i.e. {numeric,nonnumeric}{period,instant}{2002,2003} -->
                <!-- One Redundant Context Xnnp3X = nonnumeric,period,2003 -->
                <!-- without the empty scenario element -->
                <numericContext id="np3" cwa="true">
                  <entity>
                    <identifier scheme="www.nasdaq.com">SAMP</identifier>
                    <segment/>
                  </entity>
                  <period>
                    <startDate>2003-12-31</startDate>
                    <endDate>2003-12-31</endDate>
                  </period>
                  <unit><measure>ISO4217:USD</measure></unit>
                  <scenario/>
                </numericContext>
          nnp3  <nonNumericContext id="nnp3">
                  <entity>
```

```
            <identifier scheme="www.nasdaq.com">SAMP</identifier>
            <segment/>
          </entity>
          <period>
            <startDate>2003-01-01</startDate>
            <endDate>2003-12-31</endDate>
          </period>
          <scenario/>
        </nonNumericContext>
Xnnp3X  <nonNumericContext id="Xnnp3X">
          <entity>
            <identifier scheme="www.nasdaq.com">SAMP</identifier>
          </entity>
          <period>
            <startDate>2003-01-01</startDate>
            <endDate>2003-12-31</endDate>
          </period>
        </nonNumericContext>
        </xbrl>
```

Example 25. Predicates for detecting duplicates.

Node 1	Node 2	Type	Predicate	True	Reason
nnp3	Xnnp3X	context	identical	no	different nodes
nnp3	Xnnp3X	context	s-equal	yes	entity, period, cwa are equal and the empty scenario elements are ignored
f name	g name	item	s-equal	no	different context id's nnp3 and Xnnp3X
f name	g name	item	p-equal	yes	same parent element
f name	g name	item	c-equal	yes	equal contexts nnp3 and Xnnp3X
f name	g name	item	v-equal	yes	equal content "Bree"
f name	g name	item	duplicates	yes	p-equal and c-equal
b name	c name	item	s-equal	no	
b name	c name	item	p-equal	no	they are in different customer tuples
b name	c name	item	c-equal	yes	equal contexts nnp3 and Xnnp3X
b name	c name	item	v-equal	yes	they both have content "Acme"
b name	c name	item	duplicates	no	not p-equal, so v-equal doesn't matter
b gross	c gross	item	s-equal	no	
b gross	c gross	item	p-equal	no	different parents
b gross	c gross	item	c-equal	yes	they both have context np3
b gross	c gross	item	v-equal	yes	"3001" with precision 3 equals "3000"
b gross	c gross	item	duplicates	no	not p-equal, so v-equal doesn't matter
b customer	c customer	tuple	s-equal	no	different context id's nnp3 and Xnnp3X
b customer	c customer	tuple	p-equal	yes	same parent "a analysis"
b customer	c customer	tuple	c-equal	n/a	c-equal doesn't apply to tuples
b customer	c customer	tuple	v-equal	n/a	v-equal doesn't apply to tuples
b customer	c customer	tuple	duplicates	yes	p-equal, and child items name, gross, returns and net are all v-equal
b returns	d returns	item	s-equal	no	different values
b returns	d returns	item	p-equal	no	parents are b customer and d customer
b returns	d returns	item	c-equal	yes	both have context np3
b returns	d returns	item	v-equal	no	b value is 100, d value is 500
b returns	d returns	item	duplicates	no	not p-equal, so v-equal doesn't matter
b customer	d customer	tuple	s-equal	no	different values of returns and net

`b customer`	`d customer`	`tuple`	p-equal	yes	same parent “`a analysis`”
`b customer`	`d customer`	`tuple`	c-equal	n/a	c-equal doesn’t apply to tuples
`b customer`	`d customer`	`tuple`	v-equal	n/a	v-equal doesn’t apply to tuples
`b customer`	`d customer`	`tuple`	duplicates	no	p-equal, and child items `b name` and `b gross` are v-equal to `d name` and `d gross`, and child items `b returns` and `b net` are not v-equal to `b returns` and `b net`.

The XBRL definition of duplicate items and tuples encompasses many, but not all, inconsistent and redundant data items in an XBRL instance. Tuples that are not duplicates according to the XBRL definition may still have semantic inconsistencies. In the example above, customer elements “c” and “d” appear to contain data about the same customer, in the same context, but have inconsistent data; XBRL does not detect these as duplicate tuples even though to a human reader an element such as name indicates a “unique key” that is sufficient to determine that these two tuples are, in effect, c-equal (same context, different content).

4.6 Footnotes

While tuples deal with certain regularly structured associations between elements that might appear in an XBRL instance, many documents include irregularly structured associations between facts. For instance, several facts may all be linked to the sentence “Including the effects of the merger with Example.com.” To express these irregular linkages, XBRL uses XLink [XLINK]. XLink specifies attributes and elements within a separate namespace, which, when appearing in XBRL instance elements, indicate:

- the presence of a link;
- the type (simple, similar to HTML’s `<a href="">`, or extended);
- the specific resources or elements it connects;
- the directionality of any arcs within the link (e.g., from an item to a container, from an element to an alias for it);
- a token indicating any application-specific semantics of the link (e.g., the “part-whole” role).

Familiarity with XLink is presumed throughout the following text.

4.6.1 The footnoteLink extended-type link element

The footnote extended-link type element is similar to the extended-link type elements used in XBRL taxonomies. They contain locator elements, which point to the instance facts, footnote resource-type elements, which, like XBRL label elements, contain text, and `footnoteArc` arc-type elements.

4.6.1.1 xlink:type attribute

The `xlink:type` attribute has the fixed content “`extended`”. For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schema, this attribute MUST be given explicitly in every extended link-type element.

4.6.1.2 xlink:role attribute

The `xlink:role` attribute is an optional attribute which applications MAY use to differentiate the purpose or relevance of various footnotes.

4.6.2 The loc locator-type element

The loc element of the footnoteLink has the same syntax as the loc element used in the extended links of the label linkbase. See 5.5.4 below for details.

4.6.3 The footnote resource-type element

The footnote resource element has the same syntax as the label resource element. See 5.5.6.3 below for details.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="footnote" substitutionGroup="xl:resource">
    <complexType>
      <complexContent>
        <extension base="xl:resourceType">
          <attribute ref="xml:lang"/>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

4.6.4 The footnoteArc arc-type element

The footnoteArc element has exactly the same syntax as the labelArc element. See 5.5.7 for details.

4.6.4.1 xlink:arcrole attribute

The content MUST be a roleType whose value indicates the meaning of the arc.

xlink:arcrole	Meaning
http://www.xbrl.org/arc/fact-footnote	The arc is from fact to footnote.
http://www.xbrl.org/arc/footnote-fact	The arc is from footnote to fact.

4.6.4.2 xlink:title attribute

The content MUST be a string. The title content MAY be visible to users of presentation and XLink-enabled applications.

In the following example, notice that the xlink:title attribute has been used on the locators, the footnote resource and the footnoteArc elements.

Example 26. A footnote in an XBRL instance.

```
<?xml version="1.0"?>
<xbrl
  xmlns="http://www.xbrl.org/2003/instance"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:fr="http://www.xbrl-fr.org/xbrl/2003-02-29"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ISO4217="http://www.xbrl.org2003/2003/iso4217"
  xsi:schemaLocation="http://www.xbrl-fr.org/xbrl/2003-02-29 fr.xsd">

  <fr:propertyPlantEquipmentGross numericContext="c1">1200</fr:propertyPlantEquipmentGross>
  <fr:assetsTotal id="f1" numericContext="c1">2600</fr:assetsTotal>
  <fr:equityTotal id="f3" numericContext="c1">1100</fr:equityTotal>
  <fr:liabilitiesTotal id="f2" numericContext="c1">2600</fr:liabilitiesTotal>
  <link:footnoteLink
    xlink:type="extended" xlink:title="1">
    <link:footnote
      xlink:type="resource"
      xlink:label="footnote1"
      xlink:title="1"
      xlink:role="http://www.xbrl.org/2003/role#footnote"
      xml:lang="en">Including the effects of the merger.</link:footnote>
    <link:footnote
      xlink:type="resource"
      xlink:label="footnote1"
      xlink:title="1"
      xlink:role="http://www.xbrl.org/2003/role#footnote"
      xml:lang="fr">Y compris les effets de la fusion.</link:footnote>
    <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#f1"/>
    <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#f2"/>
    <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#f3"/>
    <link:footnoteArc
     xlink:from="fact1" xlink:to="footnote1"
     xlink:title="fact1-footnote1"
     xlink:arcrole="http://www.xbrl.org/arc#fact-footnote"
     xlink:show="replace" xlink:actuate="onRequest"/>
    <link:footnoteArc
      xlink:from="footnote1"
      xlink:to="fact1"
      xlink:title="footnote1-fact1"
      xlink:arcrole="http://www.xbrl.org/arc#footnote-fact"
      xlink:show="replace" xlink:actuate="onRequest"/>
  </link:footnoteLink>
  <numericContext id="c1" cwa="true">
    <entity>
      <identifier scheme="http://www.un.org/">Example plc</identifier>
      <segment/>
    </entity>
    <period>
      <instant>2001-08-16</instant>
    </period>
    <unit><measure>ISO4217:EUR</measure></unit>
    <scenario name="Actual values">
      <fr:scenarioType>actual</fr:scenarioType>
    </scenario>
  </numericContext>
</xbrl>
```

Footnotes in an XBRL instance are never considered to be duplicates of one another as defined in 4.5.3 above.

5 XBRL Taxonomies

A taxonomy MUST have a schema containing a list of element, attribute, or data type definitions. Additionally, there MAY be definitions of the relations between these elements themselves or between these elements and the elements of another taxonomy. A taxonomy is defined using the XML Schema vocabulary and elements that implement a set of XLink-compliant linkbases. An XBRL taxonomy is considered to be the combination of element definitions in XML Schema, along with zero or more links. If links are included in a taxonomy, the schema documents MUST contain a `linkbaseRef` element, an XLink simple link, which points to one or more linkbases, as documented in the XML Linking Recommendation of the W3C (Section 5.5.1 below).

An XBRL taxonomy schema is a valid instance of an XML Schema document. Each taxonomy schema document MUST use the XML Schema `import` element to import the XBRL instance schema either directly or through a chain of `import` elements in other schemas.

The XBRL instance schema defines the abstract elements `item` and `tuple`. These definitions are used in the substitution group declarations. The XBRL instance schema also defines the basic data types used in XBRL taxonomies. This schema also imports the other XBRL schemas.

Often, it will be necessary to provide namespace declarations for several other schemas, such as the schema for XML Schema itself and the namespace being defined.

Example 27. A skeletal taxonomy schema showing linkbase references.

```
<schema
  targetNamespace="http://www.xbrl.org/us/gaap/ci/2003/us-gaap-ci-2001"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:link="http://www.xbrl.org/XLink/xbrllinkbase"
  xmlns:ci="http://www.xbrl.org/us/gaap/ci/2003/us-gaap-ci-2001"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <annotation>
    <appinfo>
      <link:linkbaseRef
        xlink:type="simple"
        xlink:href="linkbase_presentation.xml"
        xlink:actuate="onRequest"
        xlink:role="http://www.xbrl.org/2003/role#presentationLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for presentation relationship</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef
        xlink:type="simple"
        xlink:href="linkbase_calculation.xml"
        xlink:actuate="onRequest"
        xlink:role="http://www.xbrl.org/2003/role#calculationLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for calculation relationship</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef
        xlink:type="simple"
        xlink:href="linkbase_definition.xml"
        xlink:actuate="onRequest"
        xlink:role="http://www.xbrl.org/2003/role#definitionLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for definition relationship</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef
        xlink:type="simple"
        xlink:href="linkbase_label.xml"
        xlink:actuate="onRequest"
        xlink:role="http://www.xbrl.org/2003/role#labelLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for labels</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef
        xlink:type="simple"
        xlink:href="linkbase_reference.xml"
        xlink:actuate="onRequest"
        xlink:role="http://www.xbrl.org/2003/role#referenceLinkbaseRef"
        xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for literature references</xhtml:p>
      </link:linkbaseRef>
    </appinfo>
  </annotation>
  <import
    namespace="http://www.xbrl.org/2003/instance"
    schemaLocation="xbrl-instance.xsd"/>
</schema>
```

XBRL taxonomies MAY be constructed in such a way as to refer to other taxonomies; this extensibility of taxonomies is a critical feature of XBRL. In order to realize the complete potential of the technology, taxonomies must be extensible to accommodate virtually any business entity's unique reporting requirements while maintaining significant comparability across entities.

XBRL taxonomies MAY import other taxonomy schemas and reference additional XBRL linkbases as appropriate.

5.1 The monetary, shares and pure data types

The XBRL schema defines the `monetary` data type, which specialises the XML Schema `decimal` type. All numeric elements in XBRL Taxonomies that represent monetary values MUST use the `monetaryItemType` data type or one derived from it. The `shares` data type represents share-based values and the `pure` data type represents growth rates, percentages, and other measures where an implicit numerator and denominator are the same units. The `fractionItemType` can be used where a numerator and denominator are explicit.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<!-- ******************* simple types************************** -->
  <!--  monetary  -->
  <simpleType name="monetary">
    <annotation>
      <documentation>This is the datatype for those financial concepts in a taxonomy that
denote units in a currency.  Instance items with this type should have a unit of measure
from the ISO4217 namespace of currencies.</documentation>
    </annotation>
    <restriction base="decimal"/>
  </simpleType>
  <!--  shares  -->
  <simpleType name="shares">
    <annotation>
      <documentation>
    This is the datatype for share-based financial concepts.
  </documentation>
    </annotation>
    <restriction base="decimal">
      <minInclusive value="0"/>
    </restriction>
  </simpleType>
  <!-- pure  -->
  <simpleType name="pure">
    <annotation>
      <documentation>This datatype serves as the type for dimensionless numbers such as
percentage change, growth rates, and other ratios where the numerator and denominator have
the same units.</documentation>
      </annotation>
    <restriction base="decimal"/>
  </simpleType>
</schema>
```

5.2 The fraction data type

The values of some facts that are to be reported may be known exactly but it may not be possible to represent them exactly using any of the primitive data types provided for in XML Schema. Examples are fractional values whose decimal representation contains recurring digits such as 1/3 (whose decimal representation is 0.333333…). To enable XBRL instances to report these exact values a new complex type “fractionItemType” is provided as defined in section 4.3 above. All values of `fractionItemType` are exact, regardless of the value of any (deprecated) `precision` attribute on their `numericContext`. The attributes `precision` and `decimals` are not permitted on elements having `fractionItemType`.

Example 28. Representing fractions

Fractional value	Representation

1/3	`<myTaxonomy:oneThird id="oneThird" numericContext="numC1">`
	`    <numerator>1</numerator>`
	`    <denominator>3</denominator>`
	`</myTaxonomy:oneThird>`

5.3 Other data types

All item types MUST be one of the types listed below or derived from one of them by restriction. This set of XBRL provided base types covers the entire set of XML Schema built-in types (both primitive and derived) [SCHEMA-2] as well as 3 types that have been identified as having particular relevance to the domain space addressed by XBRL (`monetaryItemType`, `sharesItemType`, `pureItemType` and `fractionItemType`) and hence explicitly defined in the XBRL namespace. All these types have simple content except for `fractionItemType`. Therefore, an item type in a taxonomy can never have complex content unless it is derived by restriction from `fractionItemType`.

Table 5. Defined item types.

XBRL Item Type	Base type	numeric Context
`decimalItemType`	`decimal`	yes
`floatItemType`	`float`	yes
`doubleItemType`	`double`	yes
The following numeric types are all based on the XML Schema built-in types that are derived by restriction from `decimal`.		
`integerItemType`	`integer`	yes
`nonPositiveIntegerItemType`	`nonPositiveInteger`	yes
`negativeIntegerItemType`	`negativeInteger`	yes
`longItemType`	`long`	yes
`intItemType`	`int`	yes
`shortItemType`	`short`	yes
`byteItemType`	`byte`	yes
`nonNegativeIntegerItemType`	`nonNegativeInteger`	yes
`unsignedLongItemType`	`unsignedLong`	yes
`unsignedIntItemType`	`unsignedInt`	yes
`unsignedShortItemType`	`unsignedShort`	yes
`unsignedByteItemType`	`unsignedByte`	yes
`positiveIntegerItemType`	`positiveInteger`	yes
The following numeric types are all types that have been identified as having particular relevance to the domain space addressed by XBRL and are hence included in addition to the built-in types from XML Schema.		
`monetaryItemType`	`xbrli:monetary`	yes
`sharesItemType`	`xbrli:shares`	yes
`pureItemType`	`xbrli:pure`	yes
`fractionItemType`	`complex type with the numerator being a float and the denominator being a non-zero, non-infinite float (xbrli:nonZeroNonInfiniteFloat)`	yes
`stringItemType`	`string`	no
`booleanItemType`	`Boolean`	no
`hexBinaryItemType`	`hexBinary`	no
`base64BinaryItemType`	`base64Binary`	no
`anyURIItemType`	`anyURI`	no
`uriItemType`	`anyURI`	no
`QNameItemType`	`QName`	no
`NOTATIONItemType`	`NOTATION`	no
`durationItemType`	`duration`	no
`dateTimeItemType`	`dateTime`	no

timeItemType	time	no
dateItemType	date	no
gYearMonthItemType	gYearMonth	no
gYearItemType	gYear	no
gMonthDayItemType	gMonthDay	no
gDayItemType	gDay	no
gMonthItemType	gMonth	no
The following non-numeric types are all based on the XML Schema built-in types that are derived by restriction (and/or list) from `string`.		
normalizedStringItemType	normalizedString	no
tokenItemType	token	no
languageItemType	language	no
NMTOKENItemType	NMTOKEN	no
NMTOKENSItemType	NMTOKENS	no
NameItemType	Name	no
NCNameItemType	NCName	no
IDItemType	ID	no
IDREFItemType	IDREF	no
IDREFSItemType	IDREFS	no
ENTITYItemType	ENTITY	no
ENTITIESItemType	ENTITIES	no

Some of these types, especially some of those that XML Schema has defined for backward compatibility with DTDs, may never be needed for any XBRL application, but all are provided by XBRL for completeness and compatibility with XML Schema.

Example 29. Deriving an enumerated item type

```
<schema targetNamespace="http://www.someCompany.com/taxonomy"
        xmlns:my="http://www.someCompany.com/taxonomy"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
       xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <complexType name="stateProvinceItemType">
    <simpleContent>
      <restriction base="xbrli:tokenItemType">
         <enumeration value="MI"/>
         <enumeration value="ON"/>
      </restriction>
    </simpleContent>
  </complexType>

  <element name="stateProvince" id="my_stateProvince"
           substitutionGroup="xbrli:item" type="my:stateProvinceItemType"/>
</schema>
```

Meaning: Deriving new item types by extension from the XBRL provided item types is the only allowed method for XBRL taxonomy schemas. Earlier, in Example 18, the `stateProvinceType` was defined and used to define a sub-element of `segment`. Here, instead we define an XBRL concept appearing in the company's own taxonomy; note that the previously defined simple type is not used.

5.4 Taxonomy elements

An element has a name, a substitution group, and a data type. All element names MUST be unique within a given taxonomy schema.

An element MAY have an optional `id` attribute. Providing an `id` attribute simplifies the content of the `xlink:href` attribute on linkbase locators (see 5.5.1.2 below).

Example 30. Typical element definitions in a taxonomy schema

```
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli="http://www.xbrl.org/2003/instance">
  <element
    id="ci_preferredDividends" name="preferredDividends"
    xbrli:instantaneous="false"
    type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/>
  <element
    id="ci_stockBasedCompensationPolicy" name="stockBasedCompensationPolicy"
    xbrli:instantaneous="false"
    type="xbrli:stringItemType" substitutionGroup="xbrli:item"/>
</schema>
```

Meaning: Two elements are defined with `id` attribute values with the prefix "`ci_`" so as to increase the likelihood that even if this schema is imported or included into other schemas, no XML Schema processor will correctly or incorrectly detect a violation of a uniqueness constraint such as those defined in [SCHEMA-1].

5.4.1 The instantaneous attribute

Some elements define measurable concepts at an instant in time, others a change over a duration of time. Likewise, some facts are only meaningful at an instant; others are meaningful over an arbitrarily long period. The required Boolean `instantaneous` attribute allows the taxonomy author to specify which of these choices applies to an element.

Example 31. Instantaneous and non-instantaneous element definitions.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2003/instance">
  <element id="a1" name="changeInRetainedEarnings"
           xbrli:instantaneous="false"
           type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/>
  <element id="a2" name="fixedAssets"
           xbrli:balance="debit"
           xbrli:instantaneous="true"
           type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/>
</schema>
```

It is a fatal error if an item in an XBRL instance has an "`instant`" content in the period of its context when the element definition in the taxonomy includes `instantaneous="false"`. It is also a fatal error if the period contains `"forever"` `"duration"` `"startDate"` or `"endDate"` while the element definition of the item includes `instantaneous="true"`. The `instantaneous` attribute is required for both numeric and non-numeric item types.

5.4.2 The balance attribute

An optional `balance` attribute MAY be added to the element definition. If the idea of debit/credit balance is appropriate to the element, it MAY be indicated using this attribute.

Example 32. Using the balance element to indicate normal debit and credit balances.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xbrli="http://www.xbrl.org/2003/instance">
  <element
    id="netIncome" name="netIncome" xbrli:balance="credit"
    xbrli:instantaneous="false"
    type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/>
  <element
    id="fixedAssets" name="fixedAssets" xbrli:balance="debit"
    xbrli:instantaneous="true"
    type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/>
</schema>
```

The `balance` attribute is important to applications that consume numbers related to accounting concepts such as asset, liability, equity, revenue and expense. The balance attribute (debit/credit) provides a definitive declaration of how values in XBRL instances are to be authored and interpreted when the debit/credit designation is provided.

Table 6. Correct signage in an XBRL instance.

Taxonomy element	Account balance	Sign of XBRL instance element value
`balance="credit"`	Credit	Positive or zero
`balance="credit"`	Debit	Negative or zero
`balance="debit"`	Debit	Positive or zero
`balance="debit"`	Credit	Negative or zero

The numeric representation of a debit or credit item will normally (that is, more often than not) be positive in an XBRL instance.

Example 33. A concept appearing with positive and negative values in an XBRL instance.

```
<xbrl xmlns:my="http://my.namespace.com">
  <my:netIncome numericContext="c1">500</my:netIncome>
  <my:netIncome numericContext="c2">-200</my:netIncome>
</xbrl>
```

Meaning: A profit of 500 and a loss of 200 in different contexts.

In addition, the assignment of balance attributes constrains the legal weights in `calculationArc` elements.

Table 7. Constraints among the balance attribute and calculation arc weights

`balance` attribute of "`from`" item	`balance` attribute of "`to`" item	illegal values of the `weight` attribute on `calculationArc`
`debit`	`debit`	Negative (< 0)
`debit`	`credit`	Positive (> 0)
`credit`	`debit`	Positive (> 0)
`credit`	`credit`	Negative (< 0)

It is an error to designate a balance as debit or credit if that element is not either a debit or a credit from an accounting perspective. For example, designating *cash* as a debit would be correct usage; designating *available seat miles* as either a debit or credit would be incorrect usage.

5.5 Linkbases

There are five kinds of extended link linkbases used in XBRL taxonomies.

- Relation links (calculation, definition, and presentation) manage the relations between taxonomy elements.

- Label links manage the text associated with taxonomy elements in various languages.

- Reference links manage the references to authoritative literature (either online or paper).

Each of these extended links MUST be held in an XLink document container. The document container MAY be a `linkbase` element located either:

1. at the path "`/schema/annotation/appinfo/*`" in the taxonomy schema document, or,

2. as the root element of a separate document.

5.5.1 The linkbaseRef element

The XLink specification provides for a standard way of finding linkbases [XLINK]http://www.w3.org/TR/xlink/ - xlg. The `linkbaseRef` element conforms to this standard by using a specific `xlink:arcrole` content value.

5.5.1.1 The xlink:type attribute

The `xlink:type` attribute has the fixed content “`simple`”. For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL schemas, this attribute MUST be given explicitly in every `linkbaseRef` element

5.5.1.2 The xlink:href attribute

The `xlink:href` attribute of the `linkbaseRef` element contains a URI. The URI points to a particular linkbase element that contains the appropriate extended-type links having the same the `xlink:role` attribute. If the URI reference is relative, its absolute version MUST be computed by the method of XML Base [XML Base] before use. Taxonomy authors placing an absolute URI in the `xlink:href` attribute forego the flexibility of a relative URI and SHOULD assume that consuming applications will employ a wide range of caching strategies, including no caching at all.

5.5.1.3 The xlink:role attribute

The optional `xlink:role` attribute of the `linkbaseRef` element identifies the kind of extended link contained in the linkbase. Listed below are role values for the standard kinds of XBRL extended links. Consuming applications MAY ignore any other values. It is an error for the `linkbaseRef` role to be specified when the element pointed to by the `linkbaseRef` `href` element contains other than the specified type of elements.

Table . Roles in the linkbaseRef element

Values of the `linkbaseRef` `xlink:role` attribute	Element pointed to by `xlink:href`…
`(unspecified)`	MAY contain any extended-type link elements
`http://www.xbrl.org/2003/role#calculationLinkbaseRef`	MUST contain only `calculationLink` elements
`http://www.xbrl.org/2003/role#definitionLinkbaseRef`	MUST contain only `definitionLink` elements
`http://www.xbrl.org/2003/role#labelLinkbaseRef`	MUST contain only `labelLink` elements
`http://www.xbrl.org/2003/role#presentationLinkbaseRef`	MUST contain only `presentationLink` elements
`http://www.xbrl.org/2003/role#referenceLinkbaseRef`	MUST contain only `referenceLink` elements

5.5.1.4 The xlink:arcrole attribute

The `xlink:arcrole` attribute on the `linkbaseRef` element has the XLink standard fixed content:

`http://www.w3.org/1999/xlink/properties/linkbase`

For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL schemas, this attribute MUST be given explicitly in every `linkbaseRef` element.

5.5.1.5 The xlink:actuate attribute

In XBRL linkbases, the `xlink:actuate` attribute has the fixed content “`onRequest`”. No link traversal is considered to be mandatory. No link needs to be traversed as soon as a file is loaded. Upon encountering the `onRequest` value of the `xlink:actuate` attribute an application SHOULD traverse from the starting resource to the ending resource only on a post-loading event triggered for the purpose of traversal [XLINK]. Such an event may never occur. For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schemas, this attribute MUST be given explicitly in every `linkbaseRef` element.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
      xmlns:xl="http://www.xbrl.org/2001/XLink"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
```

```
  <complexType name="simpleType">
    <complexContent>
      <restriction base="anyType">
        <attribute ref="xlink:type"/>
        <attribute ref="xlink:href"/>
        <attribute ref="xlink:role"/>
        <attribute ref="xlink:arcrole"/>
        <attribute ref="xlink:title"/>
        <attribute ref="xlink:show"/>
        <attribute ref="xlink:actuate"/>
      </restriction>
    </complexContent>
</complexType>
<element name="simple" type="xl:simpleType" abstract="true"/>
</schema>
```

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:xl="http://www.xbrl.org/2001/XLink"
        xmlns="http://www.w3.org/2001/XMLSchema">

  <element name="linkbaseRef" type="xl:simpleType"/>
</schema>
```

5.5.2 The linkbase element

The container element in an XBRL linkbase document, containing any number of extended-type links, is `linkbase`. The `profile` attribute indicates whether the linkbase can be processed by applications that support only the `noRoleTypes` profile, only the `noDeprecated` profile, or both. The attributes `xml:lang`, `xml:space` and `xml:base` [XML Base] MAY appear on the `linkbase` element.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:xl="http://www.xbrl.org/2001/XLink"
        xmlns:x="http://www.w3.org/XML/1998/namespace"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<element name="linkbase">
  <complexType>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="link:linkbaseRef" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="xl:extended" minOccurs="0" maxOccurs="unbounded"/>
    </choice>
    <attribute name="profile" type="anyURI"/>
    <attributeGroup ref="link:linkbaseProfileAtts"/>
    <attributeGroup ref="x:specialAttrs"/>
  </complexType>
</element>

<attributeGroup name="linkbaseProfileAtts">
  <attribute name="noRoleTypes" type="boolean" default="false"/>
</attributeGroup>
</schema>
```

For an XBRL processor to validate any resource-type or arc-type element it MUST resolve the URI portion of the `role` and `arcrole` attributes to a schema where it will find its definition. To aid the processor, the XML Schema instance `schemaLocation` attribute MAY be used to resolve the namespace. If the XBRL processor cannot resolve the namespace, only standard XBRL `arcrole` and `role` values can be validated; therefore, unless the profile attribute `noRoleTypes` is true, a non-standard `arcrole` or role value is an error.

Example 34. A skeletal linkbase

```
<linkbase
  xmlns="http://www.xbrl.org/2003/linkbase"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  xmlns:samp="http://www.xbrl.org/sample"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  noArcTypes="true"
  xsi:schemaLocation="http://www.xbrl.org/sample samp001.xsd">
  <calculationLink xlink:type="extended">
    <!-- ... -->
  </calculationLink>
</linkbase>
```

Meaning: Use of `linkbase` as the root element, holding namespace prefix definitions, profile attributes and the `schemaLocation` attribute. The "`xml:`" prefix cannot, and need not, be declared.

5.5.2.1 The noRoleTypes attribute

XBRL instances MAY directly or indirectly reference linkbases that define additional roles and arc roles that are not necessary or relevant to that instance. When the `noRoleTypes` attribute is `true`, resource-type elements having a `role`, and arc-type elements having an `arcrole`, that are not defined in this specification MUST be ignored. The default is `false`.

5.5.2.2 The profile attribute

The XBRL instance producer MAY specify a URI where a textual explanation of unused XBRL features. An XBRL linkbase processor MAY ignore this attribute.

5.5.3 Extended-type elements

There are six extended-type link elements.

Extended-type link elements	contain elements
`calculationLink`	`loc, aloc, calculationArc, absoluteContext` and `relativeContext`
`definitionLink`	`loc` and `definitionArc`
`footnoteLink`	`loc`, `footnote` and `footnoteArc`
`labelLink`	`loc`, `label` and `labelArc`
`presentationLink`	`loc` and `presentationArc`
`referenceLink`	`loc`, `reference`, and `referenceArc`

Extended-type link elements are containers for sub-elements of the `locator`, `arc`, or `resource` type; sub-elements of this type that are not direct children of an extended-type element have no XLink-specified meaning, and hence have no XBRL-specified meaning.

5.5.3.1 The xlink:type attribute

The `xlink:type` attribute has the fixed content "`extended`". For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL schemas, this attribute MUST be given explicitly in every extended link-type element.

5.5.3.2 The xlink:role attribute

The content of the `xlink:role` attribute is application-specific and MAY be used by applications to partition links into functional groups, for example so that consuming applications need not process all the links in a linkbase when only some are necessary. XBRL does not mandate any particular content for this attribute and XBRL processors MAY ignore it.

Example 35. Sample values of xlink:role for several referenceLink elements.

```
http://www.my.org/role#balanceSheet
http://www.my.org/role#incomeStatement
http://www.my.org/role#statementOfComprehensiveIncome
http://www.my.org/role#statementOfStockholdersEquity
http://www.my.org/role#cashFlows
```

Meaning: The taxonomy author has given a “role” to each extended `referenceLink` link so as to partition the links in an accounting-related taxonomy based on which part of a financial statement they belong to.

5.5.3.3 Extended-type link schema

The extended-type links are the container elements in the `linkbase` element for `definitionArc`, `labelArc`, `referenceArc`, and so on.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
      xmlns:xl="http://www.xbrl.org/2001/XLink"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <complexType name="extendedType">
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="xl:locator" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="xl:arc" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="xl:resource" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="xl:title"/>
    </choice>
    <attribute ref="xlink:type"/>
    <attribute ref="xlink:role"/>
    <attribute ref="xlink:title"/>
  </complexType>

  <element name="extended" type="xl:extendedType" abstract="true"/>
</schema>
```

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
      xmlns:link="http://www.xbrl.org/2003/linkbase"
      xmlns:xl="http://www.xbrl.org/2001/XLink"
      xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="presentationLink" type="xl:extendedType" substitutionGroup="xl:extended"/>
  <element name="definitionLink" type="xl:extendedType" substitutionGroup="xl:extended"/>
  <element name="labelLink" type="xl:extendedType" substitutionGroup="xl:extended"/>
  <element name="referenceLink" type="xl:extendedType" substitutionGroup="xl:extended"/>
  <element name="footnoteLink" type="xl:extendedType" substitutionGroup="xl:extended"/>
  <element name="calculationLink" type="xl:extendedType" substitutionGroup="xl:extended">
    <complexType>
      <complexContent>
        <extension base="xl:extendedType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="link:aloc" minOccurs="0" maxOccurs="unbounded"/>
            <element name="calculationArc" type="link:calculationArcType"
                     minOccurs="0" maxOccurs="unbounded"/>
            <element ref="link:relativeContext" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="link:absoluteContext" minOccurs="0" maxOccurs="unbounded"/>
          </choice>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="aloc" type="link:calculationLocatorType"/>
</schema>
```

5.5.4 The locator-type element loc

For consistency, the same locator-type elements are used in all of the extended link elements of XBRL taxonomy linkbases. The `calculationLink` element allows a different locator element, `aloc`, as well.

5.5.4.1 The xlink:type attribute

The `xlink:type` attribute has the fixed content “`locator`”. For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL schemas, this attribute MUST be given explicitly in every `locator-type` element.

5.5.4.2 The xlink:href attribute

The `xlink:href` attribute contains a URI. In the six kinds of extended links for XBRL taxonomies, the URI points to a concept definition in an XML Schema file. To point to a particular node (element in a schema, item in an instance) the URI will end in a fragment identifier. According to the XLink specification, XPointer syntax is allowed in the fragment identifier. XBRL restricts the allowed forms of the fragment identifier.

Fragment identifier	Notes
`#`*`id`*	The node pointed to MUST have an id attribute whose content is *id*.
`#xpointer(` *`XPath expression`* `)`	The expression MUST NOT use XPointer extension functions.

If the URI reference is relative, its absolute version must be computed by the method of XML Base [XML Base] before use. Taxonomy authors placing an absolute URI in the `xlink:href` attribute forgo the flexibility of a relative URI and SHOULD assume that consuming applications may employ a wide range of caching strategies, including no caching at all.

5.5.4.3 The xlink:label attribute

The `xlink:label` attribute of locator-type elements identifies the locator for reference by the arc-type element of the extended link. XLink allows more than one locator contained in an extended link to have the same `xlink:label` attribute value. In XBRL the `footnoteLink`, `labelLink` and `referenceLink` extended-type elements MAY contain multiple locator-type element with the same `label`; locators contained in the `definitionLink`, `presentationLink`, and `calculationLink` extended-type elements MUST NOT have more than one locator with the same `label`.

5.5.5 The locator-type element aloc

When processing an XBRL instance containing a particular concept, consuming applications MUST interpret the locator of that concept by default to refer to all occurrences of that concept as items in the XBRL instance. The `aloc` locator MUST appear only in the calculation linkbase and it is the only locator-type element that MAY possess an `absoluteContext` attribute.

5.5.5.1 The absoluteContext attribute

The purpose of the `absoluteContext` attribute is to restrict the scope of a locator in calculation linkbases so that it applies only to occurrences of that element that appear in matching contexts. For example, the absolute context may specify “all contexts whose end date is later than 2001-01-01,” in which case any facts in an XBRL instance whose context period ends earlier than 2001-01-01T24:00:00 would be ignored when this restriction appears in the locator of the element name in question.

Example 36. Using absolute contexts

```
<aloc
  xlink:type="locator" absoluteContext="#gullReef"
  xlink:href="gr001.xsd#cashAdjustement"
  xlink:title="element [cashAdjustment]"
    xlink:label="gr_cashAdjustment_gullReef"/>

<absoluteContext id="gullReef"
  <entityConstraint>
    <identifier scheme="www.dstc.com">Gull Reef Hotel</identifier>
  </entityConstraint>
</absoluteContext>

<calculationArc
  xlink:from="gr_cashAdjustment_gullReef" xlink:to="ci_cash" weight="-1.0"
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:type="arc" xlink:title="calculation: Go up"
  xlink:show="replace" xlink:actuate="onRequest"
  order="1"/>
```

Meaning: A calculation arc, from “cash adjustment” to “cash” is valid in all periods but only in contexts where the cash adjustment value is for the entity “Gull Reef Hotel.”

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:xbrli="http://www.xbrl.org/2003/instance"
      xmlns:s="http://www.xbrl.org/sample">

<numericContext id="np1" cwa="true">
  <entity><identifier scheme="www.dstc.com">Salty Dog Tours</identifier></entity>
  <period><instant>2001-04-01</instant></period>
  <unit><measure>ISO4217:USD</measure></unit>
</numericContext>
<numericContext id="np2" cwa="true">
  <entity><identifier scheme="www.dstc.com">Gull Reef Hotel</identifier></entity>
  <period><instant>2001-04-01</instant></period>
  <unit><measure>ISO4217:USD</measure></unit>
</numericContext>

<s:cashAdjustment numericContext="np1">5000</s:cashAdjustment>
<s:cashAdjustment numericContext="np2">5000</s:cashAdjustment>
</xbrl>
```

Meaning: The value of cash for entity Gull Reef Hotel will include the cash adjustment item, but the cash for entity Salty Dog Tours will not.

See 5.5.6.5 below for explanation of the `calculationArc` and another usage example.

5.5.5.2 Locator-type elements schema

When `aloc` appears in `calculationLink`, the `absoluteContext` attribute MAY be supplied as the URI of an `absoluteContext` element (see schema for `calculationLink` in 5.5.3.3 above).

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
      xmlns:xl="http://www.xbrl.org/2001/XLink"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <complexType name="locatorType">
    <attribute ref="xlink:type"/>
    <attribute ref="xlink:href"/>
    <attribute ref="xlink:role"/>
    <attribute ref="xlink:title"/>
    <attribute ref="xlink:label"/>
  </complexType>
</schema>
```

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
      xmlns:link="http://www.xbrl.org/2003/linkbase"
      xmlns:xl="http://www.xbrl.org/2001/XLink"
      xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="loc" substitutionGroup="xl:locator">
</schema>
```

5.5.6 Resource-type elements

The label, reference, and footnote linkbases all contain resource-type elements. Syntactically, all resource-type elements have the attributes `xlink:type` and `xlink:label`, but the element name and its semantics are specific to the linkbase they appear in. Only the calculation linkbase may contain elements other than resources, locators and arcs.

Extended-type element	MAY contain resource-type elements	MAY contain other elements
`definitionLink`	(none)	
`presentationLink`	(none)	
`labelLink`	`label`	
`referenceLink`	`reference`	
`footnoteLink`	`footnote`	
`calculationLink`		`absoluteContext`, `relativeContext`

5.5.6.1 The xlink:type attribute

The `xlink:type` attribute has the fixed content "`resource`". For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schemas, this attribute MUST be given explicitly in every resource-type element.

5.5.6.2 The xlink:label attribute

The `xlink:label` attribute identifies the resource for reference by the arc-type element of the extended link. Several resources in an extended link are allowed to have the same label.

Example 37. Label and reference resources

```
<label xlink:type="resource" xlink:role="http://www.xbrl.org/2003/role#label"
       xlink:label="ci_currentAssets_en" xlink:title="ci_currentAssets_en"
       xml:lang="en">Current Assets</label>
```

```
<reference
  xlink:type="resource" xlink:label="ci_propertyPlantAndEquipmentNet_APB"
  xlink:title="ci_propertyPlantAndEquipmentNet_APB">
  <ci:name>ABP</ci:name>
  <ci:page>42</ci:page>
</reference>
```

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:xl="http://www.xbrl.org/2001/XLink"
        xmlns:xml="http://www.w3.org/XML/1998/namespace"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!--                 label             -->
 <element name="label" substitutionGroup="xl:resource">
    <complexType>
      <complexContent mixed=”true”>
        <restriction base="xl:resourceType">
          <sequence>
            <any namespace="http://www.w3.org/1999/xhtml"
                 minOccurs="0" maxOccurs="unbounded" processContents="skip"/>
          </sequence>
          <attribute ref="xml:lang"/>
        </restriction>
      </complexContent>
    </complexType>
 </element>

  <!--                 part             -->
  <element name="part" type="string" abstract="true"/>
  <!--                 reference             -->
  <element name="reference" substitutionGroup="xl:resource">
    <complexType>
      <complexContent>
        <extension base="xl:resourceType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="link:part" minOccurs="0" maxOccurs="unbounded"/>
          </choice>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

5.5.6.3 The label resource

Although each taxonomy defines a single set of elements representing a set of business reporting concepts, the human readable XBRL documentation for those concepts, including labels (strings used as human readable names for each concept) and other explanatory documentation, is contained in a resource-type element in the label linkbase. The resource has an indication of the language used (via the XML standard `lang` attribute) and an optional classification of the purpose of the documentation (via a `role` attribute).

This ability to provide documentation in a variety of different languages enables XBRL concepts to be used in a multilingual environment because business reports can be presented to users in their language of choice.

Documentation of XBRL concepts MUST be contained in `label` elements of a label linkbase. Label elements MUST use the standard XML `lang` attribute, and they MUST appear inside `labelLink` extended link elements. Label content is mixed content containing a simple string, or a fragment of XHTML. The XHTML MUST be restricted to TEXT module elements [XHTML].

The label linkbase is designed mainly to use local resources (`label` elements). If instead, a linkbase author wishes to use remote content for labels, pointed to by `loc` (`locator-type`) elements, the following rules MUST be observed:

- The remote content MUST be in an XML or HTML document.
- In a `labelLink` element the remote content MUST be an XBRL `label` element or another (XML or HTML) element. If it is not an XBRL label element, then the label is defined to be the element content of

that element. The element MUST allow the attachment of the XBRL label element's `role` attribute as well as the XML `lang` attribute.

Example 38. Label material in remote sources

Example: Remote label content requires the `xlink:role` and `xml:lang` attributes.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
          "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xlink="xmlns:xlink="http://www.w3.org/1999/xlink"
  <head/>
  <body>
    <p id="netRevenue" xml:lang="en"
       xlink:role="http://www.xbrl.org/2003/role#label">Net Revenue</p>
  </body>
</html>
```

Linkbase reference:

```
<labelLink xlink:type="extended">
  <labelArc
    xlink:type="arc"
    xlink:arcrole="http://www.xbrl.org/2003/role#concept-label"
    xlink:title="Net Revenue is an en label for element netRevenue"
    xlink:show="replace" xlink:actuate="onRequest"
    xlink:from="L37689_NetRevenue"
    xlink:to="L37689_NetRevenue_en"/>
  <loc
    xlink:type="locator"
    xlink:href="samp001.xsd#netRevenue"
    xlink:title="element [netRevenue]"
    xlink:label="L37689_NetRevenue"/>
  <loc
    xlink:type="locator"
    xlink:href="http://www.dictionary.org/accounting.html#netRevenue"
    xlink:title="element [netRevenue]"
    xlink:label="L37689_NetRevenue_en"/>
</labelLink>
```

Meaning: The labels are stored in an XHTML file and referred to by the linkbase so long as they provide discriminating attributes.

Consuming applications are NOT REQUIRED to detect or display XBRL taxonomy element documentation that appears anywhere other than in a label linkbase resource.

Label elements MAY contain an optional `xlink:role` attribute, which SHOULD distinguish between `label` elements by the nature of the XBRL concept documentation that they provide. The table specifies all standard XBRL role attribute values and the meanings that they convey.

Table 8. Standard label role attribute values.

`label` resource `xlink:role` attribute value	Meaning
`(Omitted role attribute)`	Standard label for a concept
`http://www.xbrl.org/2003/role#label`	Standard label for a concept
`http://www.xbrl.org/2003/role#terseLabel`	Short label for a concept, often omitting text that should be inferable when the concept is reported in the context of other related concepts
`http://www.xbrl.org/2003/role#verboseLabel`	Extended label for a concept, making sure not to omit text that is required to enable the label to be understood on a stand alone basis

`label` resource `xlink:role` attribute value	Meaning

`http://www.xbrl.org/2003/role#positiveLabel`
`http://www.xbrl.org/2003/role#positiveTerseLabel`
`http://www.xbrl.org/2003/role#positiveVerboseLabel`
 	
`http://www.xbrl.org/2003/role#negativeLabel`
`http://www.xbrl.org/2003/role#positiveTerseLabel`
`http://www.xbrl.org/2003/role#negativeVerboseLabel`
 	
`http://www.xbrl.org/2003/role#zeroLabel`
`http://www.xbrl.org/2003/role#zeroTerseLabel`
`http://www.xbrl.org/2003/role#zeroVerboseLabel`
	Label for a concept, when the value being presented is positive (negative, zero). For example, the standard and standard positive labels might be “profit after tax” and the standard negative labels “loss after tax”, the terse label and terse positive labels might both be “profit”, while the negative terse label might be “loss”.

`http://www.xbrl.org/2003/role#totalLabel`	The label for a concept for use in presenting values associated with the concept when it is being reported as the total of a set of other values
`http://www.xbrl.org/2003/role#periodStartLabel` `http://www.xbrl.org/2003/role#periodEndLabel`	The label for a concept with `instantaneous="true"` for use in presenting values associated with the concept when it is being reported as a start (end) of period value.
`http://www.xbrl.org/2003/role#documentation`	Documentation of a concept, providing an explanation of its meaning and its appropriate usage and any other documentation deemed necessary
`http://www.xbrl.org/2003/role#definitionGuidance`	A precise definition of a concept, providing an explanation of its meaning and its appropriate usage.
`http://www.xbrl.org/2003/role#disclosureGuidance`	An explanation of the disclosure requirements relating to the concept. Indicates whether the disclosure is • mandatory (i.e. prescribed by authoritative literature); • recommended (i.e. encouraged by authoritative literature; • common practice (i.e. not prescribed by authoritative literature, but disclosure is common); • structural completeness. (i.e., included to complete the structure of the taxonomy).
`http://www.xbrl.org/2003/role#presentationGuidance`	An explanation of the rules guiding presentation (placement and/or labelling) of this concept in the context of other concepts in one or more specific types of business reports. For example, “Net Surplus should be disclosed on the face of the Profit and Loss statement”.
`http://www.xbrl.org/2003/role#placementGuidance`	An explanation of the rules guiding placement of this concept in the context of other concepts in one or more specific types of business reports
`http://www.xbrl.org/2003/role#measurementGuidance`	An explanation of the method(s) required to be used when measuring values associated with this concept in business reports
`http://www.xbrl.org/2003/role#commentaryGuidance`	Any other general commentary on the concept that assists in determining definition, disclosure, measurement, presentation or usage

`label` resource `xlink:role` attribute value	Meaning
`http://www.xbrl.org/2003/role#exampleGuidance`	An example of the type of information intended to be captured by the concept.

When an XBRL instance profile contains `noRoleTypes` the consuming application MAY ignore all resources that do not use the XBRL standard role types in the `role` attribute.

All `label` elements having a `role` attribute value prefixed by `http://www.xbrl.org/guidance` are to be regarded as providing all or part of the primary explanation (guidance) of the XBRL concept.

Example 39. Bi-directional arcs between a concept and one of its labels

```
<label
  xlink:type="resource"
  xlink:label="ci_currentAssets_en"
  xlink:title="ci_currentAssets_en"
  xlink:role="http://www.xbrl.org/2003/role#label"
  xml:lang="en">Current Assets</label>
<loc
  xlink:type="locator"
  xlink:href="us_bs_v2.xsd#assets.currentAsset"
  xlink:label="ci_currentAssets"
  xlink:title="ci_currentAssets"/>
<labelArc
  xlink:type="arc"
  xlink:from="ci_currentAssets"
  xlink:to="ci_currentAssets_en"
  xlink:show="embed" xlink:actuate="onRequest"
  xlink:arcrole="http://www.xbrl.org/2003/role#concept-label"
  xlink:title="Go to label of ci:assets.currentAssets_en"/>
<labelArc
  xlink:type="arc"
  xlink:from="ci_currentAssets_en"
  xlink:to="ci_currentAssets"
  xlink:show="replace" xlink:actuate="onRequest"
  xlink:arcrole="http://www.xbrl.org/2003/role#label-concept"
  xlink:title="Go to element ci:assets.currentAssets"/>
```

Meaning: The `label` element contains the text of the label and the two arc elements allow bi-directional navigation between element and label. The additional attributes of `show`, `actuate` and `title` allow XLink-aware software to navigate the linkbase.

5.5.6.4 The reference resource

Reference elements allow XBRL taxonomies to ground the definitions of reported concepts in authoritative statements in the published business, financial and accounting literature.

Reference elements SHOULD only provide links to reference materials that are relevant to understanding appropriate usage of the elements associated with the concept being defined. They MUST NOT contain the content of those reference materials themselves. Where textual content is required to complete the definition of an XBRL context, this MUST be contained in XBRL `label` elements with an XLink `role` attribute value; the standard values of the role attributes are shown in Table 9.

Reference elements MUST appear inside linkbases. Reference elements are composed of parts. Since the division of references into parts varies in every jurisdiction, `part` is an abstract element. Taxonomy writers MAY define elements that substitute for `part`, to be included inside reference elements.

Example 40. Bi-directional arcs between a concept and supporting references

```
<linkbase xmlns="http://www.xbrl.org/2003/linkbase"
          xmlns:ref="http://www.xbrl.org/2003/ref"
          xmlns:link="http://www.xbrl.org/2003/linkbase"
          xmlns:xlink="http://www.w3.org/1999/xlink"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <referenceLink xlink:type="extended">
```

```
    <!-- locator for element -->
    <loc xlink:type="locator"
         xlink:href="samp001.xsd#s_customerSales"
         xlink:title="element [customerSales]"
         xlink:label="s_customerSales"/>
    <!-- arcs -->
    <referenceArc
      xlink:type="arc" xlink:from="s_customerName_REF" xlink:to="s_customerName"
      xlink:arcrole="http://www.xbrl.org/2003/role#reference-concept"
      xlink:title="reference for [customerName]"
      xlink:show="replace" xlink:actuate="onRequest"/>
    <referenceArc
      xlink:type="arc" xlink:from="s_customerName" xlink:to="s_customerName_REF"
      xlink:arcrole="http://www.xbrl.org/2003/role#concept-reference"
      xlink:title="reference for [customerName]"
      xlink:show="replace" xlink:actuate="onRequest" />
      <!-- references all with the same xlink:label -->
    <reference
      xlink:type="resource"
      xlink:label="s_salesBycustomer_REF"
      xlink:role="http://www.xbrl.org/2003/role#definitionGuidance"
      xlink:title="Sales by Customer: Definition">
      <ref:name>Handbook of Business Reporting</ref:name>
      <ref:pages>5</ref:pages>
    </reference>
    <reference
      xlink:type="resource"
      xlink:label="s_salesBycustomer_REF"
      xlink:role="http://www.xbrl.org/2003/role#measurementGuidance"
      xlink:title="Sales by Customer: Measurement Guidance">
      <ref:name>Handbook of Business Reporting</ref:name>
      <ref:pages>45-50</ref:pages>
    </reference>
  </referenceLink>
</linkbase>
```

Meaning: The `reference` elements contains two literature citations, with different `xlink:role` attributes to distinguish them. The two arc-type elements allow bi-directional navigation between concept and reference. The additional attributes of `show`, `actuate` and `title` allow XLink aware software to navigate the linkbase. The elements `name` and `pages` are defined as members of the `part` substitution group in the taxonomy referred to by the `ref:` namespace prefix, as shown below:

```
<schema targetNamespace="http://www.xbrl.org/2003/ref"
        xmlns:ref="http://www.xbrl.org/2003/ref"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <import namespace="http://www.xbrl.org/2003/linkbase"
          schemaLocation="xbrl-linkbase.xsd"/>
  <element name="name" type="string" substitutionGroup="link:part"/>
  <element name="number" type="string" substitutionGroup="link:part"/>
  <element name="paragraph" type="string" substitutionGroup="link:part"/>
  <element name="subparagraph" type="string" substitutionGroup="link:part"/>
  <element name="clause" type="string" substitutionGroup="link:part"/>
  <element name="pages" type="string" substitutionGroup="link:part"/>
</schema>
```

5.5.6.4.1 The xlink:role attribute

Reference elements MAY contain an optional `xlink:role` attribute, which MUST distinguish between reference elements by the nature of the XBRL concept documentation that they make external reference to. The table specifies the defined role attribute values and the meanings that they convey, which are parallel to several of those for the `label` resource.

Table 9. Reference role attribute values.

`reference` resource `xlink:role` attribute value	Meaning
`(Omitted role attribute)`	Standard reference for a concept
`http://www.xbrl.org/2003/role#reference`	Standard reference for a concept
`http://www.xbrl.org/2003/role#definitionRef`	Reference to documentation that details a precise definition of the concept.

`http://www.xbrl.org/2003/role#disclosureRef`
`http://www.xbrl.org/2003/role#mandatoryDisclosureRef`
`http://www.xbrl.org/2003/role#recommendedDisclosureRef`
	Reference to documentation that details an explanation of the disclosure requirements relating to the concept. Specified categories include:
	• mandatory
	• recommended

`http://www.xbrl.org/2003/role#unspecifiedDisclosureRef`	Reference to documentation that details an explanation of the disclosure requirements relating to the concept. Unspecified categories include, but are not limited to: • common practice • structural completeness The latter categories do not reference documentation but are indicated in the link role to indicate why the concept has been included in the taxonomy.
`http://www.xbrl.org/2003/role#presentationRef`	Reference to documentation which details an explanation of the presentation or labelling of this concept in the context of other concepts in one or more specific types of business reports
`http://www.xbrl.org/2003/role#measurementRef`	Reference concerning the method(s) required to be used when measuring values associated with this concept in business reports
`http://www.xbrl.org/2003/role#commentaryRef`	Any other general commentary on the concept that assists in determining appropriate usage
`http://www.xbrl.org/2003/role#exampleRef`	Reference to documentation that illustrates by example the application of the concept that assists in determining appropriate usage.

All `reference` elements having a `role` attribute value prefixed by `http://www.xbrl.org/ref` are to be regarded as providing authoritative explanation (guidance) of the XBRL concept.

When an XBRL instance profile contains `noRoleTypes` the consuming application MAY ignore all resources that do not use the XBRL standard role types in the `role` attribute.

The `xlink:label` attribute identifies the resource for reference by the arc-type element of the extended link. Several resources in an extended link MAY have the same label.

```
<schema
  targetNamespace="http://www.xbrl.org/2003/linkbase"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xl="http://www.xbrl.org/2001/XLink"
  xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="part" type="string" abstract="true"/>

  <element name="reference" substitutionGroup="xl:resource">
    <complexType>
      <complexContent>
        <extension base="xl:resourceType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="link:part" minOccurs="0" maxOccurs="unbounded"/>
          </choice>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

5.5.6.5 The absoluteContext element

Absolute contexts define constraints that allow a taxonomy author to restrict the applicability of each `calculationArc` to certain numeric contexts. Only a calculation linkbase MAY contain elements to define absolute contexts. An `absoluteContext` element contains one sub-element for each of the types of constraints that can be expressed: entity, segment, scenario and period. If a sub-element does not appear in an `absoluteContext`, then the calculation is not constrained to match any value with regard to that aspect of its context. An empty or missing absolute context means that the `calculationArc` can be applied across all contexts having any entity, any period, and any scenario.

Example 41. Absolute contexts

Example	Matches

`<absoluteContext id="after2003">`
`  <periodConstraint>`
`    <startDateConstraint minimum="2003-01-01"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any context with a `startDate` on or after 1st January 2003.

`<absoluteContext id="payday">`
`  <periodConstraint>`
`    <instantConstraint day="15 30"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any context describing an instant falling on the 15th or 30th of any month.

`<absoluteContext id="any-year">`
`  <periodConstraint>`
`    <durationConstraint year="1"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any context with a period whose start date and end date differ by one year.

`<absoluteContext id="calendar-year">`
`  <periodConstraint>`
`    <startDateConstraint monthDay="01-01"/>`
`    <durationConstraint offset="P1Y"/>`
`    <endDateConstraint monthDay="12-31"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any context with a period whose start date and end date differs by one year and starts on 1 January and ends 31 December of that year.

`<absoluteContext id="birth">`
`  <entityConstraint>`
`    <identifier scheme="http://www.state.us">NM</identifier>`
`  </entityConstraint>`
`  <periodConstraint>`
`    <instantConstraint day="19 20" month="10" year="1998"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any context in which the state of New Mexico is the entity and the date is the 19th or 20th of October 1998.

`<absoluteContext>`
`  <scenarioConstraint><my:budget/></scenarioConstraint>`
`</absoluteContext>`
	Matches contexts in which the scenario element contains a "budget" element.

For more detail on usage and processing of the `absoluteContext` element in the calculation linkbase see 5.5.5 above and 5.5.7.11 below. `absoluteContext` is in the linkbase namespace:

```
<schema
  targetNamespace="http://www.xbrl.org/2003/linkbase"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  xmlns:xl="http://www.xbrl.org/2001/XLink"
  xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<complexType name="constraintType" abstract="true"/>
<element name="absoluteContext">
  <complexType>
    <sequence>
      <element ref="link:entityConstraint" minOccurs="0"/>
      <element ref="link:periodConstraint" minOccurs="0"/>
      <element ref="link:scenarioConstraint" minOccurs="0"/>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
  </complexType>
</element>
</schema>
```

The optional sub-elements of `absoluteContext` correspond to parts of `numericContexts`.

5.5.6.5.1 The entityConstraint element

An `entityConstraint` is used to restrict calculations to apply only to facts pertaining to a particular entity or entities. Multiple identifiers mean that more than one entity MAY match. Absence of an `entityConstraint` element means that all entities match.

Example 42. Entity constraints in absolute contexts

Example	Matches

`<entityConstraint>`
`  <identifier scheme="http://www.nasdaq.com">SADV</identifier>`
`</entityConstraint>`
	An entity with a specific name in a specific scheme.

`<entityConstraint>`
`  <identifier>SSI</identifier>`
`  <identifier>Streetwise Systems Inc.</identifier>`
`</entityConstraint>`
	An entity with either of two names, in any scheme.

`<entityConstraint>`
`  <identifier scheme="http://www.dnb.com" />`
`  <segmentConstraint><region:asia/></segmentConstraint>`
`</entityConstraint>`
	Only entities named using a specific scheme and having a specific segment element.

`entityConstraint` is defined in the linkbase namespace:

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
         xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<complexType name="entityConstraintType">
  <sequence>
    <element name="identifier" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <simpleContent>
          <extension base="token">
            <attribute name="scheme" type="anyURI"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
    <element ref="link:segmentConstraint" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>

<element name="entityConstraint" type="link:entityConstraintType"/>
</schema>
```

The `numericContext` elements in an XBRL instance matched by an `entityConstraint` MUST be the same as those selected by an XPATH expression whose general form is shown below:

Table 10. XPATH expression defining the entity constraint

```
numericContext[(entity/identifier='{identifier1}'
                and entity/identifier/@scheme='{scheme1}')
                ...
                or (entity/identifier='{identifierN}'
                    and entity/identifier/@scheme='{schemeN}')]
```

Where, for each of *1st* through *Nth* occurrence of `entityConstraint/identifier`, `{identifierN}` is the non-empty content of `entityConstraint/identifier`; and `{schemeN}` is the non-empty content of `entityConstraint/identifier/@scheme`

5.5.6.5.2 The segmentConstraint element

Any number of `segmentConstraint` elements can be used to further restrict applicability of the arc. More than one segment constraint means that either MAY match. Absence of the `segmentConstraint` element means that all segments match.

Example 43. Segment constraints in an absolute context

Example	Matches

`<entityConstraint>`
`  <segmentConstraint><region:asia/></segmentConstraint>`
`  <segmentConstraint><product:steel/></segmentConstraint>`
`</entityConstraint>`
	Only contexts whose entities have either the `region:asia` or `product:steel` segment elements will be used in the calculation.

`segmentConstraint` is defined in the linkbase namespace:

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!-- discriminator: segments and scenarios are context discriminators -->
  <complexType name="discriminatorType">
    <choice minOccurs="0" maxOccurs="unbounded">
      <any namespace="##other" processContents="strict"
           minOccurs="0" maxOccurs="unbounded"/>
    </choice>
    <attribute name="name" type="NCName" use="optional"/>
  </complexType>
  <element name="segmentConstraint" type="link:discriminatorType"/>
  <element name="scenarioConstraint" type="link:discriminatorType"/>
</schema>
```

The `numericContext` elements in an XBRL instance matched by a `segmentConstraint` MUST be the same as those selected by an XPATH expression whose general form is shown below:

Table 11. XPATH expression defining the segment constraint

```
numericContext[(entity/segment='{segment1}'
                ...
                or (entity/segment='{segmentN}']
```

Where, for each of *1st* through *Nth* occurrence of `entityConstraint/segment`, `{segmentN}` is the non-empty content of `entityConstraint/segment`;

5.5.6.5.3 The scenarioConstraint element

A `scenarioConstraint` element restricts applicability of the arc to contexts with specific scenario elements. Any empty portion is taken to match any segment.

Example 44. Scenario constraints in an absolute context

Example	Matches

`<scenarioConstraint>`
`  <my:budget/>`
`</scenarioConstraint>`
	Only contexts where element `my:budget` is in the scenario.

`scenarioConstraint` is defined in namespace `http://www.xbrl.org/2003/linkbase` and its syntax is identical to `segmentConstraint`.

The `numericContext` elements in an XBRL instance matched by a `segmentConstraint` MUST be the same as those selected by an XPATH expression whose general form is shown below:

Table 12. XPATH expression defining the segment constraint

`numericContext[(entity/segment='{scenario1}'`
`                ...`
`                or (entity/segment='{scenarioN}']`

Where, for each of *1st* through *Nth* occurrence of `scenarioConstraint` in the current `absoluteContext`, `{segment`*N*`}` is the non-empty content of `scenarioConstraint`.

5.5.6.5.4 The periodConstraint element

A `periodConstraint` element restricts applicability of an arc to contexts whose periods match specific features. Any empty portion of the `periodConstraint` is taken to match any period. When multiple elements appear, a context MAY match any or all of them.

The `periodConstraint` sub-elements have attributes that allow individual parts of the date to be constrained, and allow minimum and maximum date values to be set. Except for the `maximum` and `minimum` attributes, the attributes allow multiple values to be specified as a whitespace-separated list of numbers.

Example 45. Period constraints in absolute contexts

Example	Matches

`<absoluteContext id="monthStart">`
`  <periodConstraint>`
`    <instantConstraint day="01"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any context that is the beginning of a month.

`<absoluteContext id="thirteenMonths">`
`  <periodConstraint>`
`    <startDateConstraint day="22" year="1962" month="10"/>`
`    <endDateConstraint day="22" year="1963" month="11"/>`
`  </periodConstraint>`
`</absoluteContext>`
	The period of 13 months starting 22 October 1962.

`<absoluteContext id="afterPublicLaunchOfXBRL">`
`  <periodConstraint>`
`    <startDateConstraint minimum="2000-04-06"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any period starting on or after 6th April 2000.

`<absoluteContext id="endOfFiscalQuarter">`
`  <periodConstraint>`
`    <instantConstraint monthDay="03-31 06-30 09-30 12-31"/>`
`  </periodConstraint>`
`</absoluteContext>`
	Any instant falling at the end of any fiscal quarter.

`<absoluteContext id="idesOfMarch21stCentury">`
`  <periodConstraint>`
`    <instantConstraint`
`      yearMonth="2001-03 2002-03 2003-03" day="15"/>`
`  </periodConstraint>`
`</absoluteContext>`
	The Ides of March that have occurred in the current millennium, so far.

The element `periodConstraint` and all of its sub-elements, which rely on the date related data types in XML Schema [SCHEMA-2], are defined in the linkbase namespace:

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<simpleType name="gYearList"><list itemType="gYear"/></simpleType>
<simpleType name="gMonthList"><list itemType="gMonth"/></simpleType>
<simpleType name="gDayList"><list itemType="gDay"/></simpleType>
<simpleType name="gMonthDayList"><list itemType="gMonthDay"/></simpleType>
<simpleType name="gYearMonthList"><list itemType="gYearMonth"/></simpleType>
<attributeGroup name="absoluteDateTimeAttributes">
  <attribute name="minimum" type="date"/>
  <attribute name="maximum" type="date"/>
  <attribute name="year" type="link:gYearList"/>
  <attribute name="yearMonth" type="link:gYearMonthList"/>
  <attribute name="month" type="link:gMonthList"/>
  <attribute name="monthDay" type="link:gMonthDayList"/>
  <attribute name="day" type="link:gDayList"/>
  <attribute name="week" type="link:nonNegativeIntegerList"/>
  <attribute name="hour" type="link:nonNegativeIntegerList"/>
  <attribute name="minute" type="link:nonNegativeIntegerList"/>
  <attribute name="second" type="link:nonNegativeIntegerList"/>
</attributeGroup>

<!-- startDate -->
<element name="startDateConstraint">
  <complexType>
    <attributeGroup ref="link:absoluteDateTimeAttributes"/>
  </complexType>
</element>

<!-- duration -->
<element name="durationConstraint">
  <complexType>
    <attributeGroup ref="link:relativeDateTimeAttributes"/>
  </complexType>
</element>

<!-- endDate  -->
<element name="endDateConstraint">
  <complexType>
    <attributeGroup ref="link:absoluteDateTimeAttributes"/>
  </complexType>
</element>

<!-- instant -->
<element name="instantConstraint">
  <complexType>
    <attributeGroup ref="link:absoluteDateTimeAttributes"/>
  </complexType>
</element>

<!-- forever: the context must have duration "forever" -->
<element name="foreverConstraint"><complexType/></element>
```

Absolute context matching MUST ignore any occurrences of the (deprecated) `duration` element in the `period` element.

Unlike the entity, segment and scenario constraints, there is no simple transformation from a period constraint to an XPATH expression because the resulting expressions would require comparisons on XML Schema dates and durations, the arithmetic for which is not defined in XPATH 1.0. A numericContext will match a periodConstraint depending on the attributes appearing in the sub-elements of periodConstraint.

Table 13. Matching a numericContext period to different constraint attributes

period type:	instant	period	period	period
matched by:	instantConstraint	startDateConstraint	duration Constraint	endDateConstraint
when constraint has:				
offset	n/a; any instant matches	n/a; any period matches	startDate + duration EQ endDate	n/a; any period matches
minimum	instant is GE minimum	startDate is GE minimum	n/a; any period matches	endDate is GE minimum
maximum	instant is LE maximum	startDate is LE maximum	n/a; any period matches	endDate is LE maximum
second, minute, hour, day, month, year (list)	corresponding dateTime component of instant EQ a member of the list	corresponding dateTime component of startDate EQ a member of the list	n/a; any period matches	corresponding dateTime component of endDate EQ a member of the list
dayMonth (list)	day and month component both EQ a member of the list	both day and month dateTime components of startDate EQ a member of the list	n/a; any period matches	both day and month dateTime components of endDate EQ a member of the list
monthYear (list)	month and year component both EQ a member of the list	both year and month dateTime components of startDate EQ a member of the list	n/a; any period matches	both day year month dateTime components of endDate EQ a member of the list

5.5.6.6 The relativeContext element

Relative contexts express a relationship between the input and output (from and to) ends of a calculationArc. A relative context uses a numericContext as a base and produces a matching numericContext. The calculation linkbase MAY contain resources that define relative contexts. The content of the relativeContext element are sub-elements that express functions of the sub-elements and attributes of the item context. Any missing element or attribute is copied from the base context. Therefore, an empty relativeContext means that there are no differences between the base context and the output context; the two MUST always share s-equal contexts.

Example 46. Relative contexts

`<relativeContext id="same"/>`	An empty relative context means that the output context is s-equal to that of the input.

`<relativeContext id="period-end">`
`  <periodOffset>`
`    <instantOffset base="end"/>`
`  </periodOffset>`
`</relativeContext>`
	Given an input item's numericContext, this produces a matching context that is s-equal in all respects except that its period is the instant which marks the end of the input period.

`<relativeContext id="period-start">`
`  <periodOffset>`
`    <instantOffset base="start"/>`
`  </periodOffset>`
`</relativeContext>`
	Given an input item's numericContext, this produces a matching context that is s-equal in all respects except that its period is the instant which marks the start of the input period.

<relativeContext id="prior-year-end">
  <periodOffset>
    <instantOffset base="end" offset="-P1Y"/>
  </periodOffset>
</relativeContext>
	Produces that matching context which is s-equal in all respect except that its period is an instant whose year is 1 less than the endDate of the input period.

<relativeContext id="next-year">
  <periodOffset>
     <startDateOffset base="end" offset="-P1D"/>
     <endDateOffset base="end" year="P1D"/>
  </periodOffset>
</relativeContext>
	The output context has a period that starts on the day before the input period ends, and ends one year later.

relativeContext and its sub-elements periodOffset, instantOffset, foreverOffset, startDateOffset in the linkbase namespace:

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:xl="http://www.xbrl.org/2001/XLink"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<element name="relativeContext">
  <complexType>
    <sequence>
      <element ref="link:periodOffset" minOccurs="0"/>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
  </complexType>
</element>
</schema>
```

5.5.6.6.1 The periodOffset element

The content of the periodOffset element in relativeContext is similar to the element content of the period element in contexts. The sub-elements of period have attributes that allow individual parts of the relative period to be offset from portions of the base period. The sub-elements of relativeContextinclude are relativePeriod, instantOffset, startDateOffset, and endDateOffset.

Example 47. Period offsets in relative contexts

Examples of the periodOffset element	Base	Result

<relativeContext id="ex11">
  <periodOffset>
    <instantOffset base="start" offset="P1Y"/>
  </periodOffset>
</relativeContext>
	2002-01-01 to 2002-12-31	2003-01-01

<relativeContext id="ex12">
  <periodOffset>
    <instantOffset base="start" offset="-P3M"/>
  </periodOffset>
</relativeContext>
	2002-01-01	2001-10-01

<relativeContext id="ex13">
  <periodOffset>
    <startDateOffset base="start" offset="-P15D"/>
    <endDateOffset base="end" offset="P15D"/>
  </periodOffset>
</relativeContext>
	2002-10-01 to 2002-10-31	2002-09-16 to 2002-11-15

<relativeContext id="ex14">
  <periodOffset>
    <instantOffset offset="P91D" base="end"/>
  </periodOffset>
</relativeContext>
	2002-10-01 to 2002-10-31	2003-01-31

<relativeContext id="ex15">
  <periodOffset>
    <startDateOffset offset="P3M"/>
    <endDateOffset offset="P3M"/>
  </periodOffset>
	2002-10-01 to 2002-10-07	2003-01-01 to 2003-01-07

`</relativeContext>`		

`<relativeContext id="ex16">`
`  <periodOffset>`
`    <startDateOffset offset="P3M" base="start"/>`
`    <endDateOffset offset="P3M" base="end"/>`
`  </periodOffset>`
`</relativeContext>`
	forever	forever

`<relativeContext id="ex17">`
`  <periodOffset>`
`    <instantOffset offset="P1D" base="start"/>`
`  </periodOffset>`
`</relativeContext>`
	forever	Ambiguous; no result

```
<!--  period  -->
  <complexType name="periodType">
    <complexContent>
      <extension base="link:contextPartType">
        <sequence>
          <element ref="link:startDateOffset" minOccurs="0" maxOccurs="1"/>
          <element ref="link:endDateOffset" minOccurs="0" maxOccurs="1"/>
          <element ref="link:instantOffset" minOccurs="0" maxOccurs="1"/>
          <element ref="link:forever" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
<element name="period" type="link:periodType" substitutionGroup="link:contextPart"/>
```

5.5.6.6.2 Attributes of the periodOffset element

The elements `relativeInstant`, `endDateOffset`, and `startDateOffset` appearing in `relativeContext` have an optional attribute `offset` whose content is a `duration` that specifies a relative period, and an optional attribute `base` that specifies whether the period is relative to the start or end of the input context. The `offset` attribute defaults to the zero length duration `P0D`.

```
<simpleType name="startEndType">
    <restriction base="string">
      <enumeration value="start"/>
      <enumeration value="end"/>
    </restriction>
</simpleType>

<complexType name="relativeDateTimeType">
    <attribute name="base" type="link:startEndType"/>
    <attribute name="offset" type="duration"/>
</complexType>

<!-- startDate -->
  <element name="startDate" type="link:relativeDateTimeType"/>
  <!-- endDate -->
  <element name="endDate" type="link:relativeDateTimeType"/>
  <!-- instant -->
  <element name="instant" type="link:relativeDateTimeType"/>
  <!-- forever -->
  <element name="forever">
    <complexType/>
</element>
```

A `numericContext` with the period `forever` matches any output `numericContext`. The `base`, `relativeStartDate`, `endDateOffset` and `instantOffset` elements are all ignored in that case.

5.5.7 Arc-type elements

In presentation, calculation, and definition linkbases, arc-type elements enable XBRL concepts to be organised into networks of relationships wherein each concept can be associated with other concepts. In label and reference linkbases, arc-type elements relate a resource (label or reference) to an XBRL concept. In footnote linkbases, the arc-type elements relate the footnote resource to an XBRL fact (item or tuple).

Arc-type elements join the elements referenced in their "`from`" and "`to`" attributes. These two attributes contain labels of either locators or resources within the same extended link as the arc itself. Every arc-type element MUST have the following attributes: `from`, `to`, `type`, `show`, `actuate`, and `arcrole`, and MAY have the XLink attribute `title` and XBRL attributes `order`, `weight`, `priority`, and `relativeContext`.

Concepts that have no relationship in which they are the "`to`" element within a given arc role type within a given type of linkbase (presentation, calculation, or definition) are said to be root concepts within the scope of that arc role in that type of linkbase among a set of active linkbases. The presentation of root concepts, within the scope of a given type of linkbase, is application dependent.

The presentation, definition, or calculation linkbases are not required to specify the formatting of a report derived from a collection of XBRL instances. However, XBRL instance consuming applications are free to use the semantic information provided in a taxonomy to format such reports as they deem appropriate.

Each of these linkbases defines its own relations between concepts that taxonomy authors may or may not find useful to keep in some kind of correspondence. Some MUST be maintained free of undirected cycles; others MUST be maintained free of directed cycles. Some applications might choose to maintain linkbases free of cycles even if not syntactically required, and also may choose to maintain the contents of presentation, definition, and calculation linkbases in some kind of correspondence with one another. Fundamentally, however, these are genuinely different networks.

To take a simple example, consider the following concepts that might make up the elements of a taxonomy (note that the label would not be part of the element; labels are shown to provide clarity):

Example 48. Elements of a financial reporting taxonomy.

Label	Element Name	Balance	Substitution Group
Income Statement	`incomeStatement`		
… other taxonomy elements	`(various)`	(various)	(various)
Net Income Before Tax	`netIncomeBeforeTax`	credit	item
Taxes	`taxes`	debit	item
Net Income After Tax	`netIncomeAfterTax`	credit	item
Extraordinary Items	`extraordinaryItems`	debit	item
Net Income	`netIncome`	credit	item
Performance Measures	`performanceMeasures`		item

Suppose that the mathematical relations that exist between the concepts expressed as elements within the taxonomy as documented by some source are as follows:

1. `netIncomeAfterTax` = `netIncomeBeforeTax` – `taxes`
2. `netIncome` = `netIncomeAfterTax` – `extraordinaryItems`

The calculation linkbase might then contain calculation links to compute `netIncome`, `netIncomeBeforeTax`, `netIncomeAfterTax`, per the formulas above and expressed in a tree hierarchy in an application. The weight indicates the weight attribute value of the calculation link expressing how the element contributes to the calculation/summation.

Example 49. Hierarchy in a calculation linkbase

```
  netIncome
      |
      +--(-1)-- extraordinaryItems
      |
      +--(+1)-- netIncomeAfterTax
                     |
                     +--(-1)-- taxes
                     |
                     +--(+1)-- netIncomeBeforeTax
```

Example: Calculation hierarchy in which each item contributes to a summation.

Arcs are annotated with the numeric weight in parentheses.

The definition linkbase might also contain definition links that relate concepts to other concepts. In the case below, `performanceMeasures` is a element defined in the taxonomy and the types of performance measures are: `netIncome`, `netIncomeBeforeTax`, and `netIncomeAfterTax`. The `arcrole` of the link, a URI such as `http://www.xbrl.org/2003/role#special-general`, explains the type of definition relationship of the relation.

Example 50. Hierarchy in a definition linkbase

```
performanceMeasures
   |
   +--1-- netIncome
   |
  +--2-- netIncomeBeforeTax
  |
  +--3-- netIncomeAfterTax
```

Example: Definition hierarchy in which various concepts are defined to be “Performance Measures.”

Arcs are annotated with their “order” attribute used for presenting the hierarchy.

Presentation links are used to arrange taxonomy elements into a hierarchy and specific ordering. In general, different uses will require different sets of presentation links. There is one set of users – taxonomy developers and domain experts working with a taxonomy – whose presentation needs remain relevant throughout the entire lifecycle of a taxonomy. In some sense this view is “context free” as opposed to the presentation of instance data that is “context dependent.” When taxonomies are published they cannot contain all possible presentations but they may contain at least one “developer’s eye” view, which is “context free” in the sense that it does not need to take XBRL instance contexts into account. The presentation linkbase in this example could contain presentation links to organise concepts to look like line items in a financial statement. Another presentation linkbase could contain links to organise a subset of these same concepts into a data collection form.

Example 51. Hierarchy in a presentation linkbase

```
incomeStatement
   |
   +-- 1...13 -- other line items, e.g., revenue
   |
   +---- 14 ---- netIncomeBeforeTaxes
   |
   +---- 15 ---- taxes
   |
   +---- 16 ---- netIncomeAfterTaxes
   |
   +---- 17 ---- extraordinaryItems
   |
   +---- 18 ---- netIncome
```

Example: Presentation hierarchy that mimics the order in which line items might appear on an income statement.

This view might be used in applications to present taxonomies to users of the application. The arcs are annotated with their “order” attribute.

In these examples the three linkbases are trees, but they need not be strict trees at all. This is particularly true for the calculation linkbase. There are several ways to calculate movements in Equity, for example: one might net the issuing and retirement of common stock, net the issuing and retirement of preferred stock, and add those two – or one might add up all the issuance of stock whether common or preferred, and net it against the retirement of common and preferred. Although the calculations are hierarchical (that is, there are no loops), they do not form a tree.

5.5.7.1 The xlink:type attribute

The `xlink:type` attribute of arcs has the fixed content "`arc`". For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL schemas this attribute MUST be given explicitly in every arc-type element.

5.5.7.2 The xlink:show attribute

In XBRL linkbases, the `xlink:show` attribute has the content "`embed`" or "`replace`". Use "`embed`" in situations when a resource is linked to an element in a different file; use "`replace`" to link elements in the same file.

5.5.7.3 The xlink:actuate attribute

In XBRL linkbases, the `xlink:actuate` attribute has the fixed content "`onRequest`". No link traversal is considered to be mandatory. No link needs to be traversed as soon as a file is loaded. For the benefit of applications (such as some XSL and DOM based applications) that need this information, but cannot find it in the XBRL schemas this attribute MUST be given explicitly in every arc-type element.

5.5.7.4 The xlink:title attribute

In XBRL linkbases, the `xlink:title` attribute is used by XLink-aware applications to provide a human readable description of the arc traversal in the user interface, if one exists. This attribute is optional in every arc-type element.

5.5.7.5 The xlink:to attribute

The content MUST be an `NCName`. The content MUST appear as the content of the `xlink:label` attribute of at least one locator or resource contained in the same extended link-type element as the arc-type element.

5.5.7.6 The xlink:from attribute

The content MUST be an `NCName`. The content MUST appear as the content of the `xlink:label` attribute of at least one locator or resource contained in the same extended link-type element as the arc-type element.

5.5.7.7 The xlink:arcrole attribute

The content MUST be an absolute `URI` that identifies an arc role type. Arc role types disambiguate the meaning of an arc. Two pieces of information can be inferred from the arc role type:

1. The type of relationship the arc represents. Is this a "parent / child" display relationship? Is it a summation relationship? Is this a relationship between a container concept and a contained concept?

2. The direction of the arc. All arcs are unidirectional. Two arcs together represent a bi-directional relationship; two arcs MUST be used, one for each direction. The arc role type identifies how the "`to`" and "`from`" attributes of the `arc-type` element participate in the relationship. A set of arcs to show that A is a parent of B contains two arcs, one for "A is a parent of B" and the other for "B is a child of A". In the first case the arc would have an `arcrole` representing a parent/child relationship with a "`from`" attribute for A and a "`to`" attribute for B. In the second case the arc would have an `arcrole` representing a child/parent relationship with a "`from`" attribute for B and a "`to`" attribute for A.

Arc roles can be classified by the kinds of things that the arc relates. The valid values for the `arcrole` attribute are dependent on the classification. The classifications are:

1. Inter-concept arcs. These are arcs that represent a relationship between concepts. This includes the `definitionArc`, `presentationArc`, and `calculationArc`.

2. Arcs that relate concepts and resources. This includes the `labelArc` and `referenceArc`.
3. Arcs that relate items and resources. This includes the `footnoteArc`.

XBRL supplies a set of standard arc role types. For inter-concept arcs, the arc role types may also be defined in a taxonomy. These arcs require that either standard or taxonomy defined arc role types are used. Any other value in the `arcrole` attribute for inter-concept arcs is invalid.

All inter-concept arc role types follow a special URI format. The URI MUST be a valid namespace that identifies where the arc role type is defined. An additional fragment identifier is used to identify the specific arc role type that is defined in the namespace. The XBRL standard inter-concept arc role types are defined in the XBRL linkbase schema.

See 5.5.8 below for details on defining arc role types.

All other arc role types MAY use XBRL standard or non-XBRL standard arc role types. The non-standard arc role types are not predefined.

One or more arcs with the same `arcrole` that connect concepts can form a cycle (that is, an element may reach itself by following some number of arcs). Depending on the semantics of any given concept-to-concept relationship, different types of cycles may be legal and do not need to be tested, or may represent an error that XBRL validation should detect. Cycles occur when a node (concept or resource) can reach itself by traversing arcs having the same `arcrole` either with regard (directed) or without regard (undirected) to the direction of each individual arc. Table 14 through Table 18 indicate which arc role types MAY participate in such cycles by use of the word “Legal” in the “Dir. Cycles” and “Und. Cycles” columns and which MAY NOT participate in such cycles by use of the word “Error”.

When an XBRL instance has `noRoleTypes="true"` the consuming application MAY ignore all arcs that do not use the XBRL standard arc role types in the `arcrole` attribute.

5.5.7.7.1 Standard arc role types

XBRL defines a set of standard arc role types for each type of arc. These arc roles are symmetrically paired. Every pair of elements related by an arc within a given extended-type link MUST include the symmetric arc in that same extended-type link.

The `labelArc`: Consuming applications MAY ignore any association between locators of concept elements and remote locators or locators of `label` elements (no matter what `role` attribute a `label` element has) for an XBRL instance with the `noRoleTypes` profile when the arc has any other than one of the following `arcrole` values:

Table 14. Defined symmetric arc roles of the label arc type

`labelArc xlink:arcrole symmetric pair`	Meaning
`http://www.xbrl.org/2003/role#concept-label`	The arc is from a concept to its label.
`http://www.xbrl.org/2003/role#label-concept`	The arc is from a label to its concept.

The `referenceArc`: Consuming applications MAY ignore any `referenceArc` between locators of concept elements and remote locators or locators of `reference` elements (no matter what `role` attribute the `reference` element has) for an XBRL instance with the `noRoleTypes` profile that have any other than one of the following `arcrole` values:

Table 15. Defined symmetric arc roles of the reference arc type

`referenceArc xlink:arcrole symmetric pair`	Meaning
`http://www.xbrl.org/2003/role#reference-concept`	The arc is from a reference to a concept.
`http://www.xbrl.org/2003/role#concept-reference`	The arc is from a concept to a reference.

The `presentationArc`: Consuming applications MAY ignore any `presentationArc` between concept element and other concept elements for an XBRL instance with the `noRoleTypes` profile when the arc has any other than one of the following `arcrole` values:

Table 16. Defined symmetric arc roles of the presentation arc type

`presentationArc xlink:arcrole symmetric pair`	Meaning	Dir. Cycles	Und. Cycles
`http://www.xbrl.org/2003/role#child-parent`	The arc is from a tree child to the parent, i.e., toward the roots. In a hierarchical display of the taxonomy the child SHOULD be displayed as a child of the parent.	Error	Error
`http://www.xbrl.org/2003/role#parent-child`	The arc is from a tree parent to a child, i.e., away from the roots. In a hierarchical display of the taxonomy the child SHOULD be displayed as a child of the parent.	Error	Error
`http://www.xbrl.org/2003/role#content-container`	The arc is from a concept (item or tuple) to a tuple that MAY contain it. The element MAY appear as a child of the tuple container.	Legal	Legal
`http://www.xbrl.org/2003/role#container-content`	The arc is from a tuple to a concept (item or tuple) that it MAY contain. The content MAY appear as a child of the container.	Legal	Legal

The `calculationArc`: Consuming applications MAY ignore any `calculationArc` between a concept element and other concept elements for an XBRL instance with the `noRoleTypes` profile when the arc has any other than one of the following `arcrole` values:

Table 17. Defined symmetric arc roles for the calculation arc type

`calculationArc xlink:arcrole pair`	Meaning	Dir. Cycles	Und. Cycles
`http://www.xbrl.org/2003/role#item-summation`	The arc is from an item to a summation. The numeric value of the item is summed into the numeric value of the target provided that the result context (the base context modified by the relative context offsets) is s-equal to that of the summation. Validating processors SHOULD assume that calculation arcs with different relative contexts may form undirected cycles.	Error	Error

`http://www.xbrl.org/2003/role#summation-item`	The arc is from a summation item to another item. The numeric value of the target item is summed into the numeric value of the target provided that the result context (the base context modified by the relative context offsets) is s-equal to that of the summation. Validating processors SHOULD assume that calculation arcs with different relative contexts may form undirected cycles.	Error	Error

The `definitionArc`: Consuming applications MAY ignore any `definitionArc` between a concept element and other concept elements for an XBRL instance with the `noRoleTypes` profile when the arc has any other than one of the following `arcrole` values:

Table 18. Defined symmetric arc roles for the definition arc type

`definitionArc xlink:arcrole pair`	Meaning	Dir. Cycles	Und. Cycles
`http://www.xbrl.org/2003/role#special-general`	The arc is from a concept to a generalisation of it. A valid value of the special is a valid value of the general. Definition links created with the `child-parent` arc role in conformance with XBRL 2.0 MAY be interpreted as having this role.	Error	Legal
`http://www.xbrl.org/2003/role#general-special`	The arc is from a concept to a specialisation of it. A valid value of the special is a valid value of the general. Definition links created with the `parent-child` arc role in conformance with XBRL 2.0 MAY be interpreted as having this role.	Error	Legal

`definitionArc xlink:arcrole pair`	Meaning	Dir. Cycles	Und. Cycles
`http://www.xbrl.org/2003/role#alias-essence`	The arc is from an element that represents one perspective on this concept to an equivalent, but preferred, canonical or more generic element. The content of the alias and essence MUST be equivalent in s-equal contexts.	Error	Legal
`http://www.xbrl.org/2003/role#essence-alias`	The arc is from a preferred, canonical or generic element to an equivalent element that represents just one perspective on, or alias for, this concept. The content of the essence and alias MUST be equivalent in equivalent contexts.	Error	Legal

definitionArc xlink:arcrole pair	Meaning	Dir. Cycles	Und. Cycles
http://www.xbrl.org/2003/role#part-whole	The arc is from a concept to another concept of which it is only a part. When the use attribute of the arc equals required, then the whole MUST appear in any context where the part does.	Legal	Legal
http://www.xbrl.org/2003/role#whole-part	The arc is from a concept to another concept that forms one part of it. When the use attribute of the arc equals required, then the part MUST appear in any context where the whole does.	Legal	Legal

Applications MAY define the semantics of any other application-specific arcrole value and include them in linkbases. The roleType element (see 5.5.8 below) allows any taxonomy to extend the set of defined arcroles. An arcrole attribute MUST be a URI; hence content that is any URI will pass XML Schema validation. XBRL validating processors, on the other hand, MUST detect in linkbases all violations of the syntactic constraints as defined by the roleType element. The arcrole content MUST consist of an absolute URI that can be used to locate a schema, which in turn will contain the roleType element. Although XBRL validating processors MUST process them, XBRL applications are not required to process new arcrole values, and XBRL applications that conform to the noRoleTypes profile need not preserve them.

5.5.7.8 Arc equivalence

XLink does not define the interaction of arcs from multiple linkbases in any way. It only specifies that arc-type elements connecting the same “from” and “to” labels MUST appear in different extended-type link elements even if the arcrole attributes are equal [XLINK]. There are situations in XBRL taxonomy construction when a third party may want to edit the linkbase constructed by a previous author. This may be the addition of links, but it may also be the desire of the third party to override or negate links created by the original taxonomy author.

The concept of equivalence for XBRL arcs is based on having the same source, destination, and arc role, but does not take into account any other attributes.

Argument Types	Predicates	Definition
arc	s-equal	They have the same element name in the same namespace, and Their arcrole attributes have identical content, and Either of the extended-type links in which they are contained has an empty role, or the roles have identical content, and Their “from” attributes have the label of locator-type elements with identical href content, and Their “to” attributes have the label of locator-type elements that have identical href content, and Their relativeContext, weight, and other XBRL-specific attributes are equal.

5.5.7.9 The order attribute

The optional order attribute is a decimal number that indicates the order in which applications MUST display sibling elements when the links are being displayed, for example to a taxonomy author. It defaults to "1". If multiple siblings have the same order attribute value, the presentation order of those siblings is application dependent. The value of the order attribute is not restricted to integers, which is useful when there is a need to place a new sibling element in between two previously defined sibling elements.

In situations where it is necessary to present prohibited relationships to a user, the `order` attribute value of the prohibiting arcs dictates the ordering of siblings. The value of the `order` attribute for a prohibited (overridden) arc MUST be ignored.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!--                 useEnum             -->
  <simpleType name="useEnum">
    <restriction base="string">
      <enumeration value="optional"/>
      <enumeration value="required"/>
      <enumeration value="prohibited"/>
    </restriction>
  </simpleType>
  <!--                 basicArcType             -->
  <complexType name="basicArcType">
    <complexContent>
      <extension base="xl:arcType">
        <attribute name="use" type="link:useEnum"/>
        <attribute name="priority" type="decimal"/>
        <attribute name="order" type="decimal"/>
      </extension>
    </complexContent>
  </complexType>

  <element name="presentationArc" type="link:basicArcType" substitutionGroup="xl:arc"/>
  <element name="definitionArc" type="link:basicArcType" substitutionGroup="xl:arc"/>
  <element name="labelArc" type="link:basicArcType" substitutionGroup="xl:arc"/>
  <element name="footnoteArc" type="link:basicArcType" substitutionGroup="xl:arc"/>
  <element name="referenceArc" type="link:basicArcType" substitutionGroup="xl:arc"/>
</schema>
```

5.5.7.10 Overriding arcs

Since a third party will not have write permissions on the links created by the original taxonomy author, the only option available is to create the new, desired, link and to create another link negating the original link.

The ability to override arcs means that taxonomy authors will often wish to specify the way that their taxonomies are to be presented to human users, so as to communicate the relationships among elements and to ensure that new or overridden arcs are in some way visually apparent. The taxonomy author needs a way to specify how the elements of their taxonomy should be presented.

For an example of the use of overriding arcs, please see the non-normative examples that are distributed with the specification.

To address this situation XBRL adds two attributes to all arc-type elements. These attributes are `use` and `priority`.

5.5.7.10.1 The use attribute

The `use` attribute has an enumerated type of three values – “`optional`”, “`required`”, or “`prohibited`”.

Two of the standard values for `use` would be typically used by an original taxonomy author:

- `use="optional"` indicates that the arc MAY be traversed. This is the default value of the `use` attribute.

- use="required" appearing on a definitionArc having the whole-part or part-whole arcrole indicates that an XBRL validating processor MUST verify that the presence in the XBRL instance of an element is matched by the presence of the other element in the same context for the document.

For example, the data that is normally entered into a paper form could be represented electronically using XBRL instances. To represent the "required field" idea, the taxonomy author can create use="required" part-whole definition arcs. These arcs would link the elements representing the required fields and an element representing the concept of the form itself.

- use="prohibited" indicates that any equivalent (see 5.5.7.8 above) arc MUST NOT be traversed. Typically, only third parties will create links with this use value; consequently, overriding arcs MAY have arc roles whose namespace the author of the overriding arc does not control.

As a motivating example, consider the situation of a third party desiring to create a new "sub-total" item intervening between an item that already has summation-item arcs to other items. The creator of the new element will add links from the children to the new item and from the new item to the parent. There would then be two paths from some items to the summation item, one using the new arcs through the sub-total item, and the other using the original arcs direct to the parent. In the case of calculation links, this could result in the double counting of values. The creator of the new element SHOULD create a prohibiting arc to prevent this.

5.5.7.10.2 The priority attribute

The content of the priority attribute is an integer.

Given two arcs in two different linkbases, one of which allows the traversal from one element to the other and one of which prohibits the same traversal, the behaviour of an XBRL application MUST be based on the priority attributes of the two arcs. The arc with the numerically larger priority attribute MUST override the other arc. If the two arcs have the same priority value, behaviour is application dependent.

The default value of the priority attribute is "0".

Where there is the possibility that an extension linkbase might introduce an illegal cycle (see 5.5.7.7 above) the authors should take care to assign priority attributes in such a way as to prevent such cycles. Overridden arcs MUST NOT be taken into account when detecting cycles.

5.5.7.11 The calculationArc element

Calculation arcs between elements having numeric data types assert relationships between the content of those elements when they appear in XBRL instances. These relationships are all by default among elements with *equivalent* contexts, but optionally among elements whose contexts *match* in specified ways (see 4.4.8 above). For every fact in every numeric context in which cwa is true, XBRL calculation processors MUST indicate an error when the summation described by the links in the calculation linkbase relating to that fact, in conjunction with other facts in the same instance, would compute a result that is not equal to its contents.

Example 52. A calculation arc

```
<calculationArc
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  xlink:from="ci_prepaidExpenses"
  xlink:to="ci_currentAssets" weight="1.0" order="1"
  xlink:type="arc" xlink:title="calculation: Go up to ci:currentAssets"
  xlink:show="replace" xlink:actuate="onRequest" />
```

Meaning: Current assets are computed by summing prepaid expenses along with the values of the other elements that also have "current assets" as their parent. "Prepaid expenses" is a child of current assets and prepaid expenses contributes all of its value to current assets. This calculation is defined only when the numericContext attribute of the contributing elements are s-equal. The additional attributes of show, actuate and title tell XLink-aware software how to navigate the linkbase and what to display as the title of the arc. The order attribute tells software displaying the arc how to position it relative to others.

5.5.7.12 The relativeContext attribute

The `relativeContext` attribute is an optional reference to a `relativeContext` element that may appear when `arcrole="http://www.xbrl.org/2003/role#item-summation"`. Calculation linkbases MAY contain `relativeContext` elements that specify how to match contexts in situations where the contexts of the elements located by the "`from`" and "`to`" attributes need not be equivalent. The `relativeContext` element also provides the information for the calculation link processor to construct a correct context for the result.

Example 53. Using relative contexts with calculation arcs

```
<calculationArc
  xlink:type="arc" xlink:title="calculation: Go up"
  xlink:show="replace" xlink:actuate="onRequest"
  xlink:from="ci_commonStock"
  xlink:to="ci_commonStock" weight="1.0"
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  relativeContext="#next-year"
  order="1"/>

<relativeContext id="next-year"
  <periodOffset>
    <instantOffset offset="P1Y"/>
  </periodOffset>
</relativeContext>

<calculationArc
  xlink:type="arc" xlink:title="calculation: Go up"
  xlink:show="replace" xlink:actuate="onRequest"
  xlink:from="ci_commonStockAdditions"
  xlink:to="ci_commonStock" weight="1.0"
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  relativeContext="#period-end"
  order="1"/>
<calculationArc
  xlink:type="arc" xlink:title="calculation: Go up"
  xlink:show="replace" xlink:actuate="onRequest"
  xlink:from="ci_commonStockDisposals"
  xlink:to="ci_commonStock" weight="-1.0"
  xlink:arcrole="http://www.xbrl.org/2003/role#item-summation"
  relativeContext="#period-end"
  order="1"/>
<relativeContext id="period-end"
  <periodOffset>
      <instantOffset base="end"/>
   </periodOffset>
</relativeContext>
```

Meaning: Common stock at the end of a period is computed by summing:

1. The value of Common Stock at the beginning of an annual period, and
2. Adding the additions to Common Stock during an annual period, and
3. Subtracting disposals of Common Stock during that annual period.

The second and third calculation arcs are only defined when the `numericContext` attribute of the (not instantaneous) contributing elements are equivalent in all respects other than that their `endDate` needs to be identical to the date of the (instantaneous) Common Stock (the `absoluteContext` ensuring that only annual periods are used is not shown in this example).

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:xbrli="http://www.xbrl.org/2003/instance"
      xmlns:s="http://www.xbrl.org/sample">

<numericContext id="ni1" cwa="true">
  <entity><identifier scheme="www.hkex.com">SAMP</identifier></entity>
  <period><instant>2001-03-31</instant></period>
  <unit><measure>ISO4217:HKD</measure></unit>
</numericContext>
<numericContext id="np2" cwa="true">
  <entity><identifier scheme="www.hkex.com">SAMP</identifier></entity>
  <period><startDate>2001-04-01</startDate>
          <endDate>2002-03-31</endDate></period>
  <unit><measure>ISO4217:HKD</measure></unit>
</numericContext>
<numericContext id="ni2" cwa="true">
  <entity><identifier scheme="www.hkex.com">SAMP</identifier></entity>
  <period><instant>2002-03-31</instant></period>
  <unit><measure>ISO4217:HKD</measure></unit>

</numericContext>

<s:commonStock numericContext="ni1">5000</s:commonStock>
<s:commonStockDisposals numericContext="np2">1000</s:commonStockDisposals>
<s:commonStockAdditions numericContext="np2">500</s:commonStockAdditions>
<s:commonStock numericContext="ni2">4500</s:commonStock>
</xbrl>
```

Meaning: The instance is consistent, since 5000 (`commonStock` as of 2001-03-31) minus 1000 (`commonStockDisposals` during the year ending 2002-03-31) plus 500 (`commonStockAdditions` during the year ending 2002-03-01) equals 4500 (`commonStock` as of 2002-03-31).

The accuracy of a numerical result MUST be determined from the known facts about the accuracy of the participating facts in the calculation as determined from the various statements or inferences about precision or decimals.

5.5.7.13 The weight attribute

The `weight` attribute is a decimal that indicates the multiplier to be applied to an item value when accumulating numeric values from item elements to summation elements. A value of "1.0" means that 1.0 times the numeric value of the item is applied to the parent item. A weight of "-1.0" means that 1.0 times the numeric value is subtracted from the summation item. A weight of "0.0" indicates that the calculation arc exists only for documentation and has no impact on calculations.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="calculationArc" substitutionGroup="xl:arc">
    <complexType>
      <complexContent>
        <extension base="link:basicArcType">
          <attribute name="weight" type="decimal" use="required"/>
          <attribute name="relativeContext" type="anyURI"/>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

5.5.7.13.1 Calculation scoping within tuples

The taxonomy elements that are located by the "`from`" and "`to`" attributes of an `item-summation` calculation arc identify elements within equivalent contexts of an XBRL instance. However, calculations also take into account tuple structure in the XBRL instance. The "`from`" element MUST be a child of both elements' *least common ancestor,* and it MUST be a *sibling* item or *uncle* item (Table 1) of the "to" element, in order for the calculation to apply.

Example 54. XBRL instance fragment with nested tuples

There are three calculation arcs in the `calculationLink`:
from (item) `gross` to (summation) `net`, weight 1.0
from (item) `returns` to (summation) `net`, weight -1.0
from (item) `gross` to (summation) `totalGross`, weight 1.0
The following is a fragment of an XBRL instance. Note that all numeric items share a single context `c1`.

```
<analysis>
 <customer>
   <name nonNumericContext="c0">Acme</name>
   <gross numericContext="c1">3000</gross>
   <returns numericContext="c1">100</returns>
   <net numericContext="c1">2900</net>
 </customer>
 <customer>
   <name nonNumericContext="c0">Bree</name>
   <gross numericContext="c1">2000</gross>
   <returns numericContext="c1">200</returns>
   <net numericContext="c1">1800</net>
 </customer>
 <totalGross numericContext="c1">5000</totalGross>
</analysis>
```

calculation item ("from") path	calculation summation ("to") path	Match?	Reason
`analysis/customer[1]/gross`	`analysis/customer[1]/net`	Yes.	They are siblings.
`analysis/customer[2]/gross`	`analysis/customer[2]/net`	Yes.	They are siblings.
`analysis/customer[1]/returns`	`analysis/customer[1]/net`	Yes.	They are siblings.
`analysis/customer[2]/gross`	`analysis/customer[2]/net`	Yes.	They are siblings.
`analysis/customer[1]/gross`	`analysis/customer[2]/net`	No.	The “to” summation is not a sibling or uncle of the item.
`analysis/customer[2]/gross`	`analysis/customer[1]/net`	No.	The “to” summation is not a sibling or uncle of the item.
`analysis/customer[1]/gross`	`analysis/totalGross`	Yes.	`totalGross` is an uncle of the item under ancestor `analysis`.
`analysis/customer[2]/gross`	`analysis/totalGross`	Yes.	`totalGross` is an uncle of the item under ancestor `analysis`.

Taxonomy authors MUST use schema definitions to indicate the relationship of elements within tuples, and tuples within tuples, in such a way that calculation arcs among the elements will compute totals and other results within an appropriate scope.

5.5.7.14 The presentationArc element

The `presentationArc` element defines how elements relate to one another for presentation when the taxonomy is displayed as a parent-child hierarchy.

Example 55. A presentation arc

```
<presentationArc
  xlink:type="arc"
  xlink:from="ci_ prepaidExpenses"
  xlink:to="ci_currentAssets"
  xlink:show="replace" xlink:actuate="onRequest"
  xlink:title="presentation: Go up to ci: currentAssets"
  xlink:arcrole="http://www.xbrl.org/2003/role#child-parent"
  order="4"/>
```

Meaning: Current assets must be presented as the parent of prepaid expenses. The prepaid expense element appears after any children of current assets whose `order` is less than 4, and appears before any children of current assets whose `order` is more than 4. The additional attributes of `show`, `actuate` and `title` tell XLink aware software how to navigate the linkbase and what to display as the title of the arc.

A taxonomy author MAY choose to define abstract elements (Table 1) and create links to and from them, so as to allow taxonomy presentation applications to present elements that are related in groups even when they are not part of a tuple, calculation, or other grouping.

Example 56. An abstract concept definition.

```
<element name="balanceSheet" id="ci_balanceSheet" abstract="true"/>
```

Meaning: The `balanceSheet` element exists in the taxonomy only to organise other elements; it MUST NOT appear in an XBRL instance. It has no declared type or other attributes.

5.5.7.15 The definitionArc element

The `definitionArc` elements define how the concepts represented by elements are related to one another through generalisation and specialisation, part and whole, or through aliasing.

The `general-special` relationship means that a valid value for the “from” element is valid for the “to” element, but it is not necessarily the case that a valid value for the “to” element is a valid value for the “from” element.

Example 57. A definition arc

```
<definitionArc
  xlink:type="arc"
  xlink:from="ci_assets"
  xlink:to="ci_ currentAssets"
  xlink:show="replace" xlink:actuate="onRequest"
  xlink:title="definition: Go down to ci:currentAssets"
  xlink:arcrole="http://www.xbrl.org/2003/role#general-special"
  order="1"/>
```

Meaning: Assets is a generalisation of current assets. The `order` attribute indicates that when this link is displayed to a user, it appears after links with order less than 1, and before links with order greater than 1. The additional attributes of `show`, `actuate` and `title` tell Xlink-aware software how to navigate the linkbase and what to display as the title of the arc.

Taxonomy authors may wish to indicate, for example, that a certain set of elements are all part of a balance sheet, part of an electronic form, should not be displayed to the user according to their selected level of granularity, or are otherwise grouped in a way not captured by presentation, calculation, generalisation or other relationships available. A `definitionArc` element with the arc role "`http://www.xbrl.org/2003/role#part-whole`" MAY be used to indicate such a relationship among concepts.

A taxonomy might include a single concept viewed from different perspectives or as having several different dimensions. In the example below, the concepts of Cash by Branch Location, Cash by Account Type, and Cash by Availability are aliases of the essential concept, Cash. The three alias concepts do not sum up to Cash; each of them must be individually numerically equivalent to Cash.

Example 58. Cash, equivalent to cash as totalled by branch location and account type

Cash
- Cash by Branch Location
  - Cash in Domestic Branches
  - Cash in Foreign Branches
- Cash by Account Type
  - Cash in Interest Bearing Accounts
  - Cash in Non-interest Bearing Accounts
- Cash by Availability
  - Cash on Hand
  - Cash as Balances Due

More generally, it is often the case that it is necessary for applications to present the same facts in different ways, or calculate a value for an item in different ways. "Net income," for example, can appear both in an income statement and in a cash flow statement. Not only does the same figure appear in a different order (presentation), but also it is in fact derived differently (calculation) in these two different areas. It is appropriate in such cases for taxonomy authors to have a single "canonical best element" or "essence" for an item like "net income" and to create other "net income" items with different presentation and calculation links; use the alias-essence definition arc to indicate to XBRL validating processors and other XBRL instance consuming applications that the items MUST be consistent within any given numeric context.

A `definitionArc` element where `xlink:arcrole` is `http://www.xbrl.org/2003/role#alias-essence` denotes the relationship between two concepts, one of which is the essence (basic, primary) concept, and the other is an alias (alternative name) for the same concept. In the following text these arcs are referred to as `alias-essence` and `essence-alias` arcs respectively for brevity.

For definitions of "alias concept" "alias item" "essence concept" and "essence item" refer to Table 1. For any set of `essence-alias` or `alias-essence` arcs that have the same essence concept the term "alias concept set" means the set of alias concepts associated with the set of arcs and the term "alias item set" means a corresponding set of items in an s-equal context in an XBRL instance. The following conditions apply to definition arcs (that are not prohibited in any extension taxonomy) having one of these two arc roles, to the alias concepts and essence concepts of such arcs, and to their corresponding alias items and essence items.

1. An alias concept MAY be the essence concept of any number of definition arcs.

2. It is a fatal error for an element to be an alias concept of more than one essence concept.

3. Both the alias concept and essence concept of an arc MUST have the same item type.

4. An alias item and essence item in an XBRL instance that are c-equal MUST be v-equal regardless of whether `cwa` is `true` or `false` in those respective s-equal contexts.

5. For any non-numeric essence concept **E** for which the value is not supplied for an XBRL instance context **C**, an XBRL processor MAY infer a value for **E** that is v-equal to the values of all of the members of the alias item set **S** corresponding to all `alias-essence` arcs with **E** as their essence concept, regardless of whether `cwa` is `true` or `false` in **C**. All members of **S** MUST be v-equal. It is a fatal error if all members of **S** are not v-equal. If an application applies this rule and any member **M** of **S** does not have a value supplied but is an essence item in some set of `alias-essence` arcs, this rule MUST be applied recursively to infer the value of **M** before inferring the value of **E**.

   Example 59. Inference of values for non-numeric items with concepts connected by essence-alias arcs

   In an XBRL instance there is a context `c1` with `cwa=false`. The concepts D and E are string item types connected by an essence-alias arc, with E being the essence and D being an alias. E has the value "Bert" in context c1 while D has the value "Ernie" in context c1. This is an error.

6. For any numeric essence concept **E** for which the value is not supplied for an XBRL instance context **C**, an XBRL processor MAY at user option infer a value for **E** that is v-equal to the values of all of the members of the alias item set **S** corresponding to all `alias-essence` arcs with **E** as their essence concept, at the

greatest values of precision and decimals for which this is possible (see 4.3.4 above) regardless of whether cwa is true or false in **C**.

Example 60. Inference of values for numeric items with concepts connected by essence-alias arcs

The concepts A, B and C are connected by essence-alias arcs, with A being the essence and B and C being aliases. In an XBRL instance, B has the value 110 with precision=2 and C has the value 99 with precision=2, all in the same context.

No inference can be made for A at precision=2.

Rounding B to precision=1 gives the result 100
Rounding C to precision=1 gives the result 100

Since these two values are the same, a value of 100 at precision=1 MAY be inferred for A.

5.5.7.16 The labelArc and referenceArc elements

These arc-type elements join label and reference resources to concepts.

5.5.8 Defining roles

Taxonomy authors MAY define new roles in a taxonomy schema by adding element children to an element located at /schema/annotation/appinfo. They MUST NOT redefine roles defined by this specification.

Example 61. Defining a new role type

Example: The role type definition of a role: "http://www.mycomp.com/role#endnote" to indicate those footnotes in an XBRL instance that ought to be presented only at the end of a document.

```
<schema targetNamespace="http://www.mycomp.com/role"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<annotation>
  <appinfo>
    <link:roleType usedOn="footnote" name="endnote">
      <link:description>A footnote that should only be displayed at the end of a
document</link:description>
    </link:roleType>
  </appinfo>
</annotation>
</schema>
```

This roleType describes a role that could be used as follows:

```
<link:footnote xlink:role="http://www.mycomp.com/role#endnote"
              xlink:type="resource" xlink:label="endnote1">
Excluding the effects of the merger and contingent liabilities.
</link:footnote>
```

The schema definition for the roleType element and its sub-elements is set out below. The roleType elements appear in taxonomy schemas as children of /schema/annotation/appinfo.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<!-- role types -->
  <complexType name="roleTypeType">
    <sequence>
      <element ref="link:description"/>
    </sequence>
    <attribute name="name" type="NCName" use="required"/>
    <attribute name="usedOn" use="required">
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="label"/>
          <enumeration value="reference"/>
          <enumeration value="footnote"/>
          <enumeration value="definitionLink"/>
          <enumeration value="calculationLink"/>
          <enumeration value="presentationLink"/>
          <enumeration value="labelLink"/>
          <enumeration value="referenceLink"/>
          <enumeration value="footnoteLink"/>
          <enumeration value="linkbaseRef"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>
  <element name="roleType" type="link:roleTypeType"/>
</schema>
```

An XBRL validating processor MUST ignore any `role` not defined as specified.

Taxonomy authors MUST declare any new role value in the resource-type elements `label`, `reference`, and `footnote` intended fro application-dependent processing. Note that if the profile attribute `noRoleTypes="true" in an instance, then` an XBRL validating processor MUST ignore any resource whose role value is not standard.

Annotations are validated in “lax” mode so that the definitions above will be used and `roleType` definitions validated only if the schema of the linkbase namespace is available. It is an error for a schema to redefine a `role` previously defined in the same schema or an imported schema.

5.5.8.1 The roleType element

The `roleType` element is the element containing the role type definition. The `roleType` element describes the relationship by defining the type of resource it applies to, declaring the name, and associating a description string with it.

5.5.8.2 The usedOn attribute

The `usedOn` attribute is a required attribute that identifies what element MAY use this role type. The only valid values are `reference`, `footnote`, `definitionLink`, `calculationLink`, `presentationLink`, `labelLink`, `referenceLink`, `footnoteLink`, or `linkbaseRef`. Each role type defined MUST appear within only one element.

5.5.8.3 The name attribute

The `name` attribute contains the fragment id of the role being defined. The URI of the role is the concatenation of the `targetNamespace` attribute of the current schema with name attribute used as the fragment identifier.

5.5.8.4 The description element

The content of the `roleType` element `description` sub-element is its description string.

5.5.9 Defining arcroles

Taxonomy authors MAY define new arcroles in a taxonomy schema by adding element children to an element located at `/schema/annotation/appinfo`. They MUST NOT redefine arcroles defined by this specification.

Example 62. Defining a new arc role type

Example: The definition of a pair of arc role values: "`http://www.mycomp.com/role#part-total`" and "`http://www.mycomp.com/role#total-part`" that connect items in the calculation linkbase.

```
<schema targetNamespace="http://www.mycomp.com/role"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<annotation>
  <appinfo>
    <link:arcroleType
      arcType="calculationArc" bidirectional="true" connector="-"
      undirectedCycleLegal="false" directedCycleLegal="true">
      <link:description>A calculation of numeric items.</link:description>
      <link:sideA name="total">
        <link:description>The total of all the items that participate on the other side of
the relationship.</link:description>
      </link:sideA>
      <link:sideB name="part"
        <link:description>A numeric item that will sum up to a total.</link:description>
      </link:sideB>
    </link:arcroleType>
  </appinfo>
</annotation>
</schema>
```

This `arcroleType` describes two arc roles for a bi-directional relationship that would be used in calculation linkbases as follows:

```
<calculationArc xlink:arcrole="http://www.mycomp.com/role#total-part" .../>
<calculationArc xlink:arcrole="http://www.mycomp.com/role#part-total" .../>
```

The schema definition for the `arcroleType` element and its sub-elements is set out below. The `arcroleType` elements appear in taxonomy schemas as children of `/schema/annotation/appinfo`.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<!--  arcroleTypeType -->
  <complexType name="arcroleTypeType">
    <sequence>
      <element ref="link:description" minOccurs="0"/>
      <element name="sideA" type="link:arcroleSideType"/>
      <element name="sideB" type="link:arcroleSideType"/>
    </sequence>
   <attribute name="bidirectional" type="boolean" use="optional" default="true"/>
   <attribute name="connector" type="string" use="optional" default="-"/>
   <attribute name="directedCyclesLegal" type="boolean" use="optional" default="true"/>
   <attribute name="undirectedCyclesLegal" type="boolean" use="optional" default="true"/>
   <attribute name="arcType" use="required">
     <simpleType>
       <restriction base="NMTOKEN">
         <enumeration value="labelArc"/>
         <enumeration value="referenceArc"/>
         <enumeration value="definitionArc"/>
         <enumeration value="calculationArc"/>
         <enumeration value="presentationArc"/>
         <enumeration value="footnoteArc"/>
       </restriction>
     </simpleType>
    </attribute>
  </complexType>
  <!--  arcrole types -->
  <complexType name="arcroleSideType">
    <sequence>
      <element ref="link:description" minOccurs="0"/>
    </sequence>
    <attribute name="name" type="NCName" use="required"/>
  </complexType>
  <!-- arcrole type elements -->
  <element name="description" type="string"/>
  <element name="arcroleType" type="link:arcroleTypeType"/>
</schema>
```

An XBRL validating processor MUST ignore any `arcrole` not defined as specified.

Taxonomy authors MAY use any new role value in the resource-type elements label, reference, and footnote for application-dependent processing without any additional declarations; however, if the profile attribute `noRoleTypes="true"` an XBRL validating processor MUST ignore any resource whose role value is not specified.

Annotations are validated in "lax" mode so that the definitions above will be used and `arcroleType` definitions validated only if the schema references the linkbase namespace. It is an error for a schema to redefine an `arcrole` previously defined in the same schema or an imported schema.

5.5.9.1 The arcroleType element

The `arcroleType` element is the root element of the arc role type definition. It is a container for the rest of the definition. The `arcroleType` element describes the relationship by defining the two sides of the relationship: side A and side B. These sides represent the valid end points of the relationship.

5.5.9.2 The arcType attribute

The `arcType` attribute is a required attribute that identifies what arc-type element MAY use this arc role type. The only valid values are "`labelArc`", "`referenceArc`", "`presentationArc`", "`definitionArc`", and "`calculationArc`". Each arc role type defined MUST appear within only one arc-type element.

5.5.9.3 The bidirectional attribute

The bidirectional attribute is an optional attribute that indicates whether the relation defined by the arcrole is bi-directional or unidirectional. This is a Boolean attribute that can contain either "true" or "false". If it is "true" than the arc role type defines two arc roles, one for each direction. If it is "false" than the arc role type defines only one arc role. It is defaulted to "true" if it is not present.

5.5.9.4 The connector attribute

The connector attribute is an optional attribute that indicates what string is used in forming the arc role names that will appear in the arcrole attribute of the arc. It defaults to a dash ("-") if it is not present. It is a string that MUST contain only characters that can legally appear in a URI fragment identifier. When interpreting an arc role name, the first occurrence of the connector string is used to separate the side A name and the side B name.

5.5.9.5 The directedCycleLegal attribute

The directedCycleLegal attribute is an optional attribute that defaults to true. It indicates whether it is legal for a set of linkbases to contain non-prohibited arcs having this arcrole that form a directed cycle. When the noRoleTypes profile attribute is false, and the directedCycleLegal attribute is false, an XBRL validating processor MUST detect directed cycles having this arc role.

5.5.9.6 The undirectedCycleLegal attribute

The undirectedCycleLegal attribute is an optional attribute that defaults to true. It indicates whether it is legal for a set of linkbases to contain non-prohibited arcs having this arcrole that form an undirected cycle. When the noRoleTypes profile attribute is false, and the undirectedCycleLegal attribute is false, an XBRL validating processor MUST detect undirected cycles having this arc role. If the undirectedCycleLegal attribute is false then the directedCycleLegal attribute MUST be false.

5.5.9.7 The description element

The description element is an optional string used to document the arc role type. If it is used it MUST be the first child element of the arcroleType element, the sideA element, or the sideB element.

5.5.9.8 The sideA and sideB elements

The sideA element and the sideB element describe the concepts that MAY participate in an arc that uses this arc role type. They each represent one side of the relationship. They MAY contain a description element for documentation purposes. The sideA element MUST come before the sideB element.

For purposes of hierarchical presentation *only*, sideA MUST be treated as the parent, with sideB the child.

When defining a labelArc or referenceArc arcrole between a concept and a resource-type element, sideA MUST be treated as the concept side and sideB MUST be treated as the resource side. When defining a footnoteArc arcrole, sideA MUST be treated as the fact side and sideB the footnote resource side.

5.5.9.9 The name attribute

The name attribute is used to form the derived arc role name that is used in the arcrole attribute of the arc. The arcrole name is derived from the name attributes on the sideA element and the sideB element. The rules for deriving the arcroles names are:

1. The first arcrole name is always derived by combining in order the value of the `name` attribute on the `sideA` element with the value of the `connector` attribute on the `arcroleType` element, with the value of the `name` attribute on the `sideB` element.

2. If the `bidirectional` attribute of the `arcroleType` element has the value "`true`" then the second arc role name is derived by combining in order the value of the `name` attribute on the `sideB` element with the value of the `connector` attribute on the `arcroleType` element, with the value of the `name` attribute on the `sideA` element. If the `bidirectional` attribute has a value of "`false`" then there is no second arc role name.

In the example given, since the `bidirection` attribute is "`true`", there are two derived arc role names. The first is "`total-part`" and the second is "`part-total`". The names are formed by connecting the values of the `name` attributes for each side with the value of the `connector` attribute, which is a dash. The value of an arc role is a URI that concatenates the target namespace of the current schema with a fragment identifier that is the arc role name. Using the previous example, the namespace portion is `"http://www.mycomp.com/role"` so that a `calculationArc` element would use the attribute `xlink:arcrole="http://www.mycomp.com/role#total-part"`.

6 Linkbase, XLink, XL and other schemas

In order to allow validation of linkbase documents, the XBRL linkbase namespace (`http://www.xbrl.org/2003/linkbase)` MUST be used with other schemas that implement the XLink specification and provide certain attributes in the XML namespace. These are provided with XBRL schemas and sample files. These schemas that implement the XLink specification and define the namespaces `http://www.w3.org/1999/xlink` and `http://www.w3.org/XML/1998/namespace` are not official documents of the W3C. It is the intention of XBRL International to integrate with the official schemas for XLink should they become available.

A schema for namespace `http://www.xbrl.org/2003/role` contains all the arc roles defined in XBRL itself and MAY be used by processors that do not have these definitions built in.

A schema for namespace `http://www.xbrl.org/2003/iso4217` defines elements for international currencies and MUST be used by instances using ISO4217 currency codes in the `unit` element. Each currency code is an element so as to define a `QName` for validation. The simple type `currency` is also defined for use in taxonomies and instances.

```
<schema targetNamespace="http://www.xbrl.org/2003/iso4217"
        xmlns="http://www.w3.org/2001/XMLSchema"
  <element name="EUR">
  <!-- ... 181 other currencies ... -->
  <simpleType name="currency">
    <restriction base="NMTOKEN">
      <enumeration value="EUR"/>
      <!-- ... 181 other currencies ... -->
    </restriction>
  </simpleType>
</schema>
```

Only fragments relating to the Euro (EUR) are shown above, not all 182 currencies.

7 Document types and MIME types

Proper detection of an XBRL document depends on the XBRL namespace. The namespace for the current version is `http://www.xbrl.org/2003/instance` for instances, `http://www.xbrl.org/2003/linkbase` for linkbases, and `http://www.xbrl.org/2003/role` for standard roles and arc roles. A MIME type allows applications to detect an XBRL document without needing to open the document.

It is the intention of XBRL International to register MIME types for each of the common kinds of XBRL document. These MIME types will be:

```
application/xbrl-instance+xml
application/xbrl-schema+xml
application/xbrl-linkbase+xml
```

8 References

[IEEE] IEEE
IEEE Standard for Binary Floating Point Arithmetic
http://standards.ieee.org/reading/ieee/std_public/description/busarch/754-1985_desc.html

[ISO] International Standards Organisation.
ISO 4217 Currency codes, ISO 639 Language codes, ISO 3166 Country codes.
http://www.iso.ch/

[RFC2119] Scott Bradner.
RFC 2119
http://www.ietf.org/rfc/rfc2119.txt

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

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

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

[URI] Tim Berners-Lee, Roy Fielding, and Larry Masinter
RFC 2396: Uniform Resource Identifiers. Internet Engineering Task Force, 1995.
http://www.ietf.org/rfc/rfc2396.txt

[XBRL] David vun Kannon and Luther Hampton
Extensible Business Reporting Language (XBRL) 2.0 Specification.
http://www.xbrl.org/TR/2001/xbrl-2002-02-04.doc

[XHTML] Murray Altheim et al.
Modularization of HTML
http://www.w3.org/TR/xhtml-modularization/

[XML] Tim Bray, Jean Paoli, C.M. Sperberg-McQueen
Extensible Markup Language (XML) 1.0 (Second Edition).
http://www.w3.org/TR/rec-xml

[XML Base] Jonathan Marsh
XML Base.
http://www.w3.org/TR/xmlbase/

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

[XPATH] James Clark and Steve DeRose
XML Path Language (XPATH) 1.0 Specification.
http://www.w3.org/TR/xpath/

[XPTR] Paul Grasso, Eve Maler, Jonathan Marsh, and Norman Walsh, editors
XML Pointer Language (XPointer Framework) V1.0.
http://www.w3.org/TR/xptr-framework/

Document history and acknowledgments (non-normative)

This specification could not have been written without the contribution of many people. The participants in the XBRL Specification Working Group, public commentators, and personal advisors have all played a significant role. The XBRL International Specification Group is chaired by Masatomo Goto, Fujitsu Laboratories of USA, and its vice chair is Hugh Wallis of Hyperion Solutions Corporation. The XBRL International Domain Working Group also produced and refined many issue drafts and final requirements documents that defined the scope and guided the priorities of this version of the specification. The XBRL International Domain working group is chaired by Mark Schnitzer of Morgan Stanley and vice chaired by John Turner of KPMG. In alphabetical order and in addition to those individuals already credited as editors, Peter Calvert of ICAEW, Eric E. Cohen of PricewaterhouseCoopers, Josef MacDonald of Ernst & Young, Manabu Mizutani of PCS, David Prather of IASCF, and Eiichi Watanabe of TSR, all contributed to the authoring and refinement of those requirements.

2003-04-23 [Hamscher] Edits to incorporate name of release as the name of specification, updated status to Public Working Draft. Updated list of editors, contributors and Acknowledgements. Corrected numerous typographical and style errors caught by Charles Hoffman, Campbell Pryde and Hugh Wallis.

2003-04-21 [Wallis] Finalised changes required to present to Domain Working Group as a candidate for submition to the ISC for approval as Public Working Draft. Incorporated minor corrections from Charles Hoffman. Added detailed text to define v-equal for numeric items of different types in a complete and unambiguous way. Various minor formatting and grammatical updates.

2003-04-20 [Hamscher] Changed the relative context specifiers to use the XML Schema duration type; provided tables detailing the matching rules for absolute contexts; removed proposed absolute and relative context filters; provided an example of an absolute context in use. Consolidated all roles and arc roles as fragments under the http://www.xbrl.org/2003/role namespace URI. Added footnote linkbase material in several places per suggestion of Phillip Engel.

2003-04-17 [Hamscher] Edited arc role material to incorporate distinction between directed and undirected arcs, adding attributes to the arc role definition material, along with changes to schema. Removed composition linkbase material, and rewrote the tuple related material, moving composition linkbase functionality relating to extensions into the definition linkbase, and defining the legal schema constructs appearing in restrictions of the tuple type. Clarified text relating to equality testing in the presence of the precision attribute. Added note clarifying that items may only refer to a context ID that is within the scope of the enclosing `xbrl` element. Added note clarifying that the general-special arc role has the same semantic intent as 2.0's definition parent-child arc.

2003-04-14 [Hamscher] Updated material on arc roles and equality definitions. Updated schemas accordingly. Made the symmetry of arc roles more explicit and made explicit the requirement that arcs be symmetric. Added standard "zero" label roles. Added table captions and table of tables. Generalised c-equal to not require identical element names so as to use it in alias-essence definitions. Removed unused references. Changed the `absoluteContext` and `relativeContext` types to `anyURI` so as to allow for remote context definitions.

2003-04-08 [Hamscher] Typo, schema, and reference fixes in preparation for internal release.

2003-04-06 [Hamscher] Fixed example text based on suggestions of Rene van Egmond and Don Dwiggins of UBmatrix. Section 5.3 on derived types changed to mandate the derivation of item types by restriction from a provided item type. Corrected miscellaneous typos in examples and schemas detected by Charles Hoffman. Added more to Example 8. Began converting to use of upper case modals. Weakened directions for use of the `balance` attribute from "MUST" to "MAY" at direction of DWG. Incorporated comments from David vun

Kannon and Geoff Shuetrim, adding the "/positive" label role, defining "linkbase namespace" and "instance namespace", clarifying the role of XBRL validation, moving MIME type note to the end possibly to be removed, changed the profile description to use a set of Boolean attributes while removing the nopointers profile, adding the pure type and item type, created the ISO4217 namespace and schema, rearranged description of order attribute, made fixes to the absolute and relative context examples. Removed conceptMatch attribute and generalised the arcRole definition mechanism to cover any arc role with concomitant changes to the schema. Replaced occurrences of must, shall and may with MUST and MAY. Added notes regarding the impact of combining schemas with different name spaces on phenomena such as arc overrides and arc role definitions. Rewrote sections relating to equivalence and duplications to provide precise definitions of various notions of equality. Changed the relativecontext and absoluteContext to normal elements instead of resources, and restricted the use of the relativeContext and absoluteContext attributes only within the calculationLink element. Added a calculation linkbase example using relative contexts. Updated the label and reference linkbase role tables to reflect most recent changes from Josef MacDonald. Updated schemas.

2003-03-30 [Hamscher] Added clarifications and other edits from Hugh Wallis, Eric E. Cohen, and others. Revised the four introductory linkbase examples using material provided by Charles Hoffman. Incorporated arcroleType material from Phillip Engel and propagated arcrole syntax changes throughout. Distinguished between XBRL validation and optional calculation linkbase validation. Changed baseProfile to profile as list of tokens and propagated changes throughout. Revised schemas. Fixed typos, replaced "instance document" and variations with "XBRL instance" throughout. Added example captions. Changed the use="required" statement to apply only to the part-whole arc role. Expanded the examples of duplicates and equivalence. Removed sections 6 and 7 (semantics) since this material is now integrated into sections 4 and 5.

2003-03-23 [Hamscher] Added acknowledgement of Domain working group members. Defined the numericItemAttrs attribute group, rootType complex type that disallows nested group elements, disallowed nested segment elements, and otherwise brought consistency to other Schema changes throughout the text. Cleaned up text relating to allowed item types. Defined equality for numeric items in the face of differing values of precision and decimals. Clarified that equality of items is *not* affected by adding "ID" attributes. Removed the optional unit sub-element in nonNumericContext and multiple segment sub-elements in the entity type. Moved the bulk of the tuple definition material to the linkbase section as a placeholder. Changed arcroles to remove linkprops path element. Added text about arc cycles. Shortened the footnote example. Used the newly DWG approved debit/credit material. Specified the two legal locations for linkbase elements. Added the linkbase element syntax. Provided an example of remote label content and moved this material to the label resource section. Tentatively restricted the linkbaseRef element to empty content. Included schema fragments for every defined element. Removed linkprops component from all defined role and arcrole values. Tentatively added three negative label roles pending DWG approval. Added a tentative table of reference resource roles. Added mention of XML Base in three places and note regarding absolute URI usage in two. Incorporated material from Geoff Shuetrim into the composition linkbase, which includes the tuple arc, sequence resource, and choice resource. Removed element-dimension from the calculation linkbase and incorporated text into the definition linkbase for the alias-essence relationship.

2003-03-11 [Hamscher] Began revisions to relativeContext and absoluteContext and miscellaneous fixes to schema material.

2003-03-11 [Shuetrim] Added a section proposing a variant on the calculation link processing model that is sensitive to calculation link role attribute values. Introduced a number of smaller edits and queries regarding the approach in relation to tuples and other areas of significant change since the previous draft.

2003-03-10 [Hamscher] Added relative contexts to the calculation linkbase and the relativeContext element and all its paraphernalia. Tentatively added absolute contexts. Redefined equivalence so as to ignore non-XBRL attributes and rely only on tuple elements. Added example of tuple scoping for calculation arcs. Removed the stock-flow and flow-stock arcroles. Added additional explanatory text to the abstract. Separated the explanation of linkbases from taxonomies and schemas. Added table of primitive and derived types and item types. Tightened up language around the href attribute of linkbaseRef. More formatting tweaks particularly to non-normative examples.

2003-03-07 [Hamscher] Changed the `baseProfile` attribute to a URI. Added "0.0" as a legal value for the `weight` attribute on `calculationArc`. Added additional material regarding schemaLocation. Added list of legal item types.

2003-03-06 [Hamscher] Changed `stockFlow` to `instantaneous` to generalise. Added example of Spanish and Portuguese labels to reinforce the point that schemas and linkbases can be mixed and matched by any given schema. Defined "identical" "equivalent" and in some cases, "matching," and used these to rewrite context processing and duplicate items. Defined "inconsistency" of `decimals` and `precision` attributes. Changed `xbrlPrecision` to `precisionType`, etc. Added the `baseProfile` attribute and noted inline where it impacts the scope of XBRL syntax recognised. Moved the `order` attribute to appear on all arc-type elements. Yet more formatting changes, small fixes to examples and schema fragments but these still need to be finalised with published schemas.

2003-02-18 [Hamscher] Responded to comments from Hugh Wallis and Geoff Shuetrim, in most cases by editing the text as requested, and noted areas requiring further resolution. Tried to increase the consistency of formatting, in particular to indicate all normative material as unshaded even when appearing inside a table.

2003-02-08 [Wallis] Numerous editorial changes and comments added. Changed, deleted and added sections about precision and decimals. Added definitions section. Added a fractionItemType data type.

2003-01-27 [Hamscher] Added normative text relating to arcroles. Removed the reference-actual and actual-reference arcroles to conform with Linkbase clarity issues. Revised the section on arcrole to conform to linkbase clarity requirements insofar as they are currently defined. Described the definitionArc as a "specialisation / generalisation" arc. When used to define a tuple, the relationship is actually a part-whole relationship, as noted when defining the constraint that children of a tuple definition must not appear in XBRL instances except when wrapped by the parent. Added placeholders for numeric precision and decimal sections. Removed anySimpleType from the schema. Changed references to 2.1 to Tulip. Reformatted entire document based on more recent XBRL International documents. Changed example uses of `<group>` to `<xbrl>`.

2003-01-22 [vun Kannon] Added material clarifying the syntax and semantics of tuples.

2003-01-19 [Shuetrim] Added material relating to linkbase clarity, and all new roles for `label` resources.

2002-09-05 [vun Kannon] Released as internal working draft of 2.1 specification. Included `stockFlow and balance` attributes and XML Schema primitive data types.

2002-06-12 [vun Kannon] Began 2.1 changes. Eliminated reference to the group element. Added xbrl root element. Changed definition of duplicate items to allow duplicates in separate tuples. Added prohibition of duplicate tuples.

2002-01-09 [vun Kannon] Corrected the discussion of the datatype of item to refer to anySimpleType.

2001-12-13 [vun Kannon] Added additional explanatory text relating to concept equivalency. Eliminated references to "draft" status.

2001-11-21 [Hamscher] Added additional explanatory text relating to links and linkbases and their intended uses, reformatted examples and callouts for readability, applied "code" and "code block" styles as appropriate, corrected minor typos.

2001-11-15 [Matherne] Edited for consistency and readability. Added "example" and "suggested" label to several illustrations for clarity. In the example at section 4.4, changed the link pointing to a file on the web site. Change the page footer to XBRL Specification v2, 2001-11-14. Added text at "Status of This Document".

2001-11-15 [vun Kannon] Added wording on MIME types, priority deadlock in overriding arcs.

2001-10-16 [vun Kannon/Wang] Edited for consistency and readability. Modified examples to make namespaces consistent. Incorporated commentary from discussion groups and added explanatory material.

2001-08-24 [Hampton] Edited for consistency and readability. Modified examples to make namespaces consistent. Incorporated commentary from discussion groups and added explanatory material.

2001-06-21 [vun Kannon] First Draft of enhanced version. Modified examples to reflect use of substitution groups and other features of XML Schema. Modified taxonomy section to reflect use of XML Linking structures.

2000-07-31 [vun Kannon] Final review. Added namespace prefix to many examples.

2000-07-20 [vun Kannon] changed sense={add, subtract, none} to numeric weight.

2000-06-27 [vun Kannon] Corrected schemaLocation attribute examples and explanation. Corrected typos and namespace references.

2000-04-12 [Hoffman] Made corrections to reference to public discussion group, changed xfrml-public to xbrl-public. Changed the links pointing to this document on the web site from 00-04-04 version to 00-04-06 version. Removed a link in section 1.2 of this document to a document (March 3rd, 2000 version of SPEC) in the private eGroups vault. Updated PDF version and HTML versions for all of these changes.

2000-04-06 [Hamscher] Made corrections to the SAMP and IMA examples. Remaining text did not change.

2000-04-02 [Hamscher] In the taxonomy, eliminated "total" from element names or changed them to "gross" as appropriate. In the taxonomy, changed "cash flow" to "cash flows". In the taxonomy, changed "intangible assets" in long term assets to "intangibles". Added additional examples of the period attribute. Deleted the [Instance Rationale] note, since the design rationale discussion covers all the necessary points. Removed the [Style Everywhere] note, since we have a current compromise which allows the group element to contain elements other than items. Added section discussing the meaning of "period" and why a specific date and duration is a good idea. Added section discussing prior period balances and how that interacts with taxonomies. Added note on alternate breakdowns. Added cautionary note about applications assuming duration. Fixed all the capitalization problems in the examples to agree with 00-04-04 release of the files.

2000-03-29 [Hamscher] Miscellaneous typo corrections. Continuing repairs to text that concerns the fact that markup is forbidden inside items. Changed all "CamelCase" names to "camelCase". Added an additional paragraph explaining the "sense" attribute. Checked for references to "footnote" that should have been references to Notes. Added the [Long Names] note.

2000-03-28 [Hamscher] Added the "pure" datatype, deleted the [unit examples] issue. Reverted to original explanation of the item tag disallowing embedded markup. Changed wording of the paragraph contrasting namespaces with the schemaLocation attribute. Added [Instance Includes] suggestion raised by David vun Kannon. Added explanation of parsing implications of decimalPattern. Got rid of the [Time Duration] issue and changed to an explanation that we are differing from XML Schema convention. Miscellaneous typo corrections.

2000-03-24 [Hamscher] Changed text references to "taxonomy attribute" to schemaLocation. Fixed typo in example of 3.12. Fixed the period definition with a better reference for ISO 8601 than the incomplete summary given in the W3C material. Miscellaneous typo corrections.

2000-03-23 [Hamscher] Added change log. Changed "taxonomy" to schemaLocation. Repaired broken definition of period attribute, raised new timeDuration issue. Included new "unique elements" issue. Raised issue of deleting "links". Added XML Schema: Primer reference. Changed text of the Unit Examples text, fixing the Moody's example and removing the PURE example. Added issue regarding label processing. Got rid of the Parents Required issue, left the discussion. Added historical notes regarding the fundamental decisions agreed to at the Chicago meeting. Changed scalefactor to scaleFactor. Changed taxonomy to schemaLocation. Added distinction between financial presentation and accounting, in the context of order independence. Similar distinction with respect to negative balances. Added discussion of the unique naming issue. Fixed the non-negative-integer datatype of order. Added taxonomy extensions issue, from Eric Cohen. Miscellaneous typo corrections.

2000-03-19 [Hamscher] First released version.

Intellectual property status (non-normative)

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to XBRL International or XBRL organizations, except as required to translate it into languages other than English. Members of XBRL International agree to grant certain licenses under the XBRL International Intellectual Property Policy (www.xbrl.org/legal).

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

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

Appendix: Approval process (non-normative)

The approval process follows that described in xbrl-processes-REC-2002-04-02. This section will be removed from the final Recommendation.

	Stage (* - Current)	Party responsible for decision	Next step	Revisions needed	Target date for stage completion
1	Internal WD	Spec. WG + Domain WG	Recommend for Stage 2	Stay in Stage 1	2003-04-08
2 *	Internal WD pending publication.	ISC	Approve for Stage 3	Return to Stage 1	2003-04-22
3	Public WD under 60 day review	WD Editor(s)	Minor revisions – to Stage 4	Major revisions, Restart Stage 1	2003-06-22
4	Draft Recommendation	Spec WG + Domain WG	Recommend for Stage 5	Restart Stage 3	2003-06-30
5	Recommendation pending publication	ISC	Approve for Stage 6	Restart Stage 4	2003-07-15
6	Recommendation	Done			