Extensible Business Reporting Language (XBRL) 2.1

2003-07-31 Update to the Public Working Draft of 2003-04-23

This version:

XBRL-WD-2003-07-31.doc

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:

XBRL-WD-2003-07-31.zip

Editors

Name	Contact	Affiliation
Phillip Engel	phillipengel@kpmg.com	KPMG LLP
Walter Hamscher[1]	walter@hamscher.com	Standard Advantage
Geoffrey Shuetrim	gshuetrim@kpmg.com.au	KPMG LLP
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
Campbell Pryde	cpryde@kpmg.com	KPMG LLP

Status of this document

This document is an XBRL International Internal update to the Public Working Draft that was published on 2003-04-23. It contains all updates that have been approved by the Specification Working Group up to and including 2003-07-31. 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 reports.

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 the English language text of this document and/or schema fragments included in the main body of this document and the normative schemas contained herein, the schemas SHALL prevail. The schemas and other documents published separately and contemporaneously with the specification are non-normative and are provided for the convenience of users of this specification.

1.1   Documentation conventions

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

The following highlighting is used for non-normative commentary in this document:
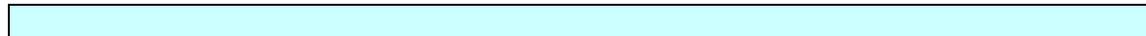
Non-normative editorial comments are denoted by indentation and the prefix "Note":

> **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 standard that benefits to all four groups.

The needs of end users of business information have generally taken 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. For example, XBRL specifies the format of information that would reasonably be expected in an electronic format for securities filings by public entities. 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 or an XBRL document can be a database dump, containing huge numbers of facts. It can also 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 list is provided to reduce the possibility of ambiguity and confusion (see also the references in section 6 below).

Table 1. Terms and definitions.

Term	Definition
abstract element	An element for which the attribute `abstract` in its XML schema declaration has the value "`true`" and which, therefore, cannot be used in an XBRL instance.
alias concept	The concept at the "to" end of a definition arc with arc role `http://www.xbrl.org/2003/role#essence`-alias.
alias item	An item in an instance whose element is an alias concept.

Term	Definition
arc	Arcs relate concepts to each other by associating their locators. Arcs also associate concepts with resources by connecting the concept locators to the resources themselves. Arcs are also used to connect fact locators to footnote resources in footnote extended links. Arcs have a set of attributes that document the nature of the relationships expressed in extended links. Importantly all arcs have an `xlink:arcrole` attribute that determines the semantics of the relationship they describe.
c-equal	Context-equal: Items or sets or sequences of items having the same item type in s-equal contexts. For a formal definition, see Section 4.8 below.
child, parent, sibling, grandparent, uncle, ancestor	Relationships among elements in an XBRL instance: using the terminology of [XPATH], for any element **E**, another element **F** is its: • child if and only if **F** appears on the `child` axis of **E** • parent if and only if **F** appears on the `parent` axis of **E** • sibling if and only if **F** appears on the `child` axis of the parent of **E** and is not **E** itself • grandparent if and only if **F** is the parent of the parent of **E** • uncle if and only if **F** is a sibling of the parent of **E** • ancestor if and only if **F** appears on the `ancestor` axis of **E**
concept	Concepts are defined in two equivalent ways. In a syntactic sense, a concept is an XML Schema element definition, defining the element to be in the `item` element substitution group or in the `tuple` element substitution group. At a semantic level, a concept is a definition of kind of fact that can be reported about the activities or nature of a business activity.
concrete element	An element for which the attribute `abstract` in its XML schema declaration has the value "`false`" and which, therefore, may appear in an XML instance.
context	Contexts are XML fragments that occur as children of the root element in XBRL instances. They document the entity, the period and the scenario that gives appropriate context for understanding the values of items. Each context element is only capable of documenting a single combination of entity, period and scenario.
Discoverable Taxonomy Set (DTS)	A DTS is a collection of taxonomy schemas and linkbases. The bounds of a DTS are such that the DTS includes all taxonomy schemas and linkbases that can be discovered by following links or references in the taxonomy schemas and linkbases included in the DTS. At least one taxonomy schema in a DTS must import the *xbrl-instance-2003-07-31.xsd (normative)* schema. See Section 3 for details on the discovery process.
duplicate items	Two items that are p-equal and c-equal: two occurrences, in an XBRL instance document, of the same concept in the same context under the same parent. For a formal definition see duplicate item in 4.8 "Equality predicates relevant to detecting duplicate" below.
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. For a formal definition see duplicate tuple in 4.8 "Equality predicates relevant to detecting duplicate" below.
element	An XML element defined using XML Schema.
entity	A business entity, the subject of XBRL items. Where the XML/[SGML] concept of syntactic "entity" is meant, this will be pointed out.

Term	Definition
essence concept	The concept at the "from" end of a definition arc with arc role `http://www.xbrl.org/2003/role#essence`-alias.
essence item	An item in an instance whose element is an essence concept.
extended link	An extended link is an element identified as an extended link using the syntax defined in the XML Linking Language [XLINK]. Extended links define a set of relationships between information that they contain and information contained in third party documents. See Section 3.5.3 for more details.
fact	Facts can be simple, in which case their values must be expressed as simple content (except in the case of simple facts whose values are expressed as a ratio), and facts can be compound, in which case their value is made up from other simple and/or compound facts. Simple facts are expressed using items (and are referred to as items in this specification) and compound facts are expressed using tuples (and are referred to as tuples in this specification).
instance namespace	The namespace used for XBRL 2.1 instances, `http://www.xbrl.org/2003/instance`
item	An item is an element in the substitution group for the XBRL `item` element. It contains the value of the simple fact and a reference to the context (and unit for numeric items) needed to correctly interpret that fact. When items occur as children of a tuple, they must also be interpreted in light of the other items and tuples that are children of the same tuple. There are numeric items and non-numeric items, with numeric items being required to document their measurement accuracy and units of measurement.
least common ancestor	In an instance, the element that is an ancestor of two elements and has no child that also appears on the `ancestor` axis [XPATH] of those same two elements.
linkbase	A linkbase is a collection of XML Linking Language [XLINK] extended links that document the semantics of concepts in a taxonomy.
linkbase namespace	The namespace of XBRL 2.1 linkbases, `http://www.xbrl.org/2003/linkbase`
locator	Locators supply an XPointer [XPTR] reference to the taxonomy schema element definitions that uniquely identify each concept. They provide an anchor for extended link arcs. See Section 3.5.3.6 for more details.

Term	Definition
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:

	MUST REQUIRED SHALL	The definition is an absolute requirement of the specification.
	MUST NOT SHALL NOT	The definition is an absolute prohibition of the specification.
	SHOULD RECOMMENDED	There may be valid reasons in particular circumstances to ignore a particular feature, but the full implications must be understood and carefully weighed before choosing a different course.
	SHOULD NOT NOT RECOMMENDED	There may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing an behaviour described with this label.
	MAY OPTIONAL	A feature is truly optional. One vendor may choose to include the feature because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same feature. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein and implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

Term	Definition
non-numeric item	An item that is not a numeric item as defined below. Dates, in particular, are not numeric.
numeric item	An item 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 Section 5.1.1.3 for details on item types).
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. For a formal definition, see Section 4.8 below.
resource	Resources are XML fragments, contained within extended links that provide additional information about concepts or items. See Section 3.5.3.7 for details.
root of an XBRL instance	The root of an XBRL instance is the `xbrl` element. In principle, it is possible to embed an XBRL instance in *any* XML document. In this case, the `xbrl` element is the container for the XBRL instance.
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. For a formal definition, see Section 4.8 below.

Term	Definition
taxonomy	A taxonomy is an XML schema and the set of XBRL linkbases that it references using `linkbaseRef` elements and the linkbases that are nested within it.
tuple	A tuple is an element in the substitution group for the XBRL `tuple` element. Tuples are used to bind together the parts of a compound fact. Those constituent parts are themselves, facts but they must be interpreted in light of each-other. For example, the name, age and compensation of a director of a company need to be grouped together to be correctly understood.
unit	Units are XML fragments that occur as children of the root element in XBRL instances. They document the unit of measure for numeric items. Each unit element is only capable of documenting a single unit of measurement.
u-equal	Unit-equal. u-equal numeric items having the same units of measurement. For a formal definition, see Section 4.8 below.
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, implied precision or decimals attributes. For a formal definition see Section 4.8 below.
taxonomy schema	A taxonomy schema is an XML Schema [SCHEMA-1] compliant XML document. A large part of many taxonomy schemas is given over to the definition of the syntax for the concepts in that taxonomy.
XBRL instance	XBRL instances are XML fragments with root element, `xbrl`. XBRL instances contain business report facts, with each fact corresponding to a concept defined in their supporting DTS. XBRL instances also contain contexts and units that provide additional information needed to interpret the facts in the instance.
x-equal	[XPATH]-equal: The XPath "=" operator returns the value `true`. For a formal definition, see Section 4.8 below.

1.5 Levels of conformance

This specification describes two levels of conformance for XBRL aware processors. The first is required of all XBRL processors. Support for the other level of conformance will depend on the purpose of the processor.

Minimally conforming XBRL processors MUST completely and correctly implement all of the syntactic restrictions embodied in this specification.

Fully conforming XBRL processors MUST be minimally conforming and, in addition, they MUST completely and correctly implement all of the semantic restrictions relating to linkbases and XBRL instance documents.

All restrictions embodied in this specification apply to minimally conforming processors unless otherwise stated.

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 recognised by the schema. Further implementation experience within the XBRL community, especially the publication of the XBRL General Ledger taxonomy, motivated many other changes. A number of business requirements documented by the XBRL International Domain working group have been incorporated.

There are only two changes 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. Firstly, a new required enumerated attribute, `periodType`, has been added to the element definition. The attribute specifies whether an item represents an instantaneous measurement or a measurement over a duration. Secondly, the parent-child arc in definition linkbases has been replaced less ambiguous relationships which distinguish between the different types of relationships that parent-child formerly represented. In every other case changes involved restricting the syntax and clarifying processing semantics and automated conversion of 2.0 compliant documents to conform to the 2.1 standard is possible.

2.1 Changes in XBRL instances

The `group` element has been eliminated. It has been replaced with the `<xbrl>` element, which acts as the root element of an XBRL instance.

The set of taxonomy schemas and linkbases supporting an XBRL instance has been formally defined (as a Discoverable Taxonomy Set (DTS)). XBRL instances now identify their supporting DTS using a new `schemaRef` element, which points to supporting taxonomy schemas and using the existing `linkbaseRef` element, which points to supporting linkbases. The XML Schema Instance `schemaLocation` attribute is no longer required in the DTS discovery process.

The `schemaRef` elements must now appear first in an XBRL instance. The `linkbaseRef` elements must appear after the `schemaRef` elements and before all other elements in an XBRL instance.

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 eliminated from context periods so durations now have to be represented using `startDate` and `endDate`. There is also additional guidance on entering data to define a period of time. The `precision` attribute on `numericContext` has been eliminated in favour of more detailed documentation at the level of the numeric items. The `CWA` attribute on the `numericContext` element has been eliminated. The `unit` element has been separated from the `numericContext` element to enable numeric and non-numeric items to use the same context structures. The `numericContext` element and the `nonNumericContext` element have been replaced with a `context` element that documents only entity, period and scenario.

An additional mechanism has been introduced to enable XBRL instance preparers to make statements about the numerical accuracy of the facts reported. Specifically, a new `decimals` attribute has been allowed on items of numeric type to provide an alternative way to document accuracy in terms of the number of decimal places to which a numerical fact is accurate. Rules for handling `precision` and `decimals` attributes have been provided.

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.

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.

2.2 Changes in XBRL taxonomies

Some of the arc role values and role values previously *suggested* are now *normative* and additional arc role values and role values have been defined. Some of the previously suggested arc role values have been removed. A new mechanism to define custom arc role values and role values has been added. The element-dimension relationship in the calculation extended link has been superseded by the essence-alias arc in definition extended links. The parent-child arc no longer exists in the calculation extended link and has been replaced by summation-item arc. The parent-child arc no longer exists in the definition extended link and has been replaced by the general-special arc and by the XML Schema approach to content modelling for tuples. Because the parent-child arc in definition extended links has two possible replacements, this is one area where complete backward compatibility with 2.0 has not been

achieved. Some manual intervention may be required when converting these relationships expressed in 2.0 taxonomies to 2.1. Some networks of relationships are no longer allowed to contain directed or undirected cycles.

Tuples may now have a complex content model, but MUST only use the XML Schema `ref`, `choice`, `all` and `sequence` elements with `minOccurs` and `maxOccurs` attributes to describe this content model. Tuple content model definitions MUST NOT permit descendant elements for the tuple that are not in the item substitution group or in the tuple substitution group. This implies that the declarations of the descendant elements for tuples MUST be references to globally declared elements [SCHEMA-1].

Calculations have been constrained to apply only within the scope of a tuple for items within a tuple.

The number of available item types has been expanded to include all of the appropriate built-in 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.

Derivation of new item and tuple types from those defined by XBRL itself has been limited so that item types MUST be defined by restriction from the set of item types provided by XBRL. This set contains item types that are derived by extension from all the appropriate built-in simple types of XML Schema and a special purpose type with complex content, the `fractionItemType`.

The suggested `xlink:role` attribute on extended link locators, that indicated the root element of a relationship hierarchy, has been eliminated.

Clarity has been provided around the possibility for linkbases to be contained in taxonomy schemas.

A mandatory `periodType` attribute has been added to concept definitions to constrain the type of period that can be attached to items based on concepts.

3   XBRL framework

XBRL defines a syntax in which a fact can be reported as the value of a well defined reporting concept within a particular context. The syntax enables software to efficiently and reliably find, extract and interpret those facts. The XBRL framework splits business reporting information into two components: XBRL instances and taxonomies.

XBRL instances contain the facts being reported while the taxonomies define the concepts being communicated by the facts. The combination of an XBRL instance and its supporting taxonomies, and additional linkbases constitute an XBRL business report.

3.1   Overview of XBRL taxonomies

A taxonomy is comprised of an XML Schema [SCHEMA-1] and all of the linkbases contained in that schema or directly referenced by that schema. The XML schema is known as a taxonomy schema.

In XBRL terminology, a concept is a definition of a reporting term. Concepts manifest as XML Schema [SCHEMA-1] element definitions. In the taxonomy schema a concept is given a concrete name and a type. The type defines the kind of data types allowed for facts measured according to the concept definition. For example, a "cash" concept would typically have a monetary type. This declares that when cash is reported, its value will be monetary. In contrast, a "accountingPoliciesNote" concept would typically have a string type so that, when the "accountingPoliciesNote" is reported in an XBRL instance, its value would be interpreted as a string of characters. Additional constraints on how concepts can be used are documented by

additional XBRL attributes on the XML Schema [SCHEMA-1] element definitions that correspond to the concepts. See Section 5.1.1 for details.

The linkbases in a taxonomy further document the meaning of the concepts by expressing relationships between concepts (inter-concept relationships) and by relating concepts to their documentation. See Section 5.2 for details.

A linkbase is a collection of extended links. There are five different kind of extended links used in taxonomies to document concepts: definition, calculation, presentation, label and reference. The first three types of extended link express inter-concept relationships, and the last two express relationships between concept and their documentation.

The linkbases MAY be contained in a separate document from the taxonomy schema, and they MAY be embedded in the taxonomy schema. When a linkbase is not embedded in a taxonomy schema, the taxonomy schema MUST contain a `linkbaseRef` to point to the linkbase document if the linkbase is to be part of the taxonomy built around the taxonomy schema.

3.2 Overview of XBRL instances

While a taxonomy defines reporting concepts, it does not contain the actual values of facts based on the defined concepts. The fact values are contained in XBRL instances and are referred to as "facts". Besides the actual value of a fact, such as "cash is 500,000", the XBRL instance provides contextual information necessary for interpreting the fact values. For numeric facts, the XBRL instance also documents measurement accuracy and measurement units.

An instance document can be supported by more than one taxonomy. Also, taxonomies can be interconnected, extending and modifying each other in various ways. Generally, it is necessary to consider multiple related taxonomies together when interpreting an XBRL instance. The set of related taxonomies supporting an XBRL instance is called a Discoverable Taxonomy Set (DTS). A DTS is a collection of taxonomy schemas and linkbases. At least one taxonomy schema in a DTS must import the *xbrl-instance-2003-07-31.xsd (normative)* schema. The bounds of a DTS are determined by starting from some set of documents (instance, taxonomy schema, or linkbase) and following DTS discovery rules. Although an instance document can be the starting point for DTS discovery, the instance document itself is not part of the DTS.

DTS rules of discovery:

Taxonomy schemas in the DTS are those:

1. referenced directly from an instance document using the `schemaRef` element. The `xlink:href` attribute on the `schemaRef` element contains the URL of the taxonomy schema that is discovered. Every taxonomy schema that is referenced by the `schemaRef` element MUST be discovered.

2. referenced from a discovered taxonomy schema via an XML Schema `import` or `include` element. Every taxonomy schema that is referenced by an `import` or `include` element in a discovered taxonomy schema MUST be discovered.

3. referenced from a discovered linkbase document via a `loc` element. Every taxonomy schema that is referenced by an `xlink:href` attribute on a `loc` element in a discovered linkbase MUST be discovered.

Linkbase documents in the DTS are those:

1. referenced directly from an XBRL instance via the `linkbaseRef` element. The `xlink:href` attribute contains the URL of the linkbase document being discovered. Every linkbase that is referenced by the `linkbaseRef` element MUST be discovered.

2. referenced from a discovered taxonomy schema via the `linkbaseRef` element. The `xlink:href` attribute contains the URL of the linkbase being discovered.

Every linkbase that is referenced by the `linkbaseRef` element MUST be discovered.

3. that occur at the XPath `"schema/annotation/appinfo/*"` in a discovered taxonomy schema (Throughout this specification, `schema`, `annotation` and `appinfo` are all elements defined in the XML Schema namespace).

For example, the "Financial Reporting for Commercial and Industrial Companies, US GAAP DTS" consists of well-defined concepts within the US Generally Accepted Accounting Principles (GAAP) when those principles are applied to Commercial and Industrial (C&I) companies. This DTS contains an "expense" concept.

A hospital instance document may use these concepts from the US GAAP C&I DTS as well as an additional concept "physician salaries" that is defined in a separate taxonomy. This taxonomy would include linkbases that relate the "physician salaries" concept to the "expense" concept in the US GAAP C&I DTS. The hospital instance document would have a `schemaRef` element pointing to the hospital taxonomy. This instance document would be the starting place for determining the DTS that supports the instance document. The discovery starts by following the `schemaRef` element to the hospital taxonomy. In the hospital taxonomy there would be a `linkbaseRef` element pointing to its linkbases. One of the linkbases contains a `loc` element pointing to the "expense" concept in one the US GAAP C&I taxonomies. The taxonomy that contains the "expense" concept would point to the other taxonomies in the US GAAP C&I DTS. Following this discovery process, all necessary taxonomies would be discovered and the result would be DTS that includes the US GAAP C&I DTS and the hospital specific taxonomy.

As this example shows, DTSs can also be used as "building blocks" to create larger, more sophisticated DTSs. Users MAY compose groups of existing DTSs into higher-level DTSs and MAY selectively add concepts and concept relationships via extension taxonomies.

While some consuming applications might be able to perform processing on an XBRL data file without referring to a DTS, normally, the interpretation and processing of any given XBRL fact is relative to the contents of a DTS.

For example, given an XBRL instance, to correctly produce a list of facts with the entries in the list corresponding to an ordered set of concepts, it is necessary to find the label corresponding to each fact being listed. The labels are contained in label extended links. The locations of the label extended links may be specified by `linkbaseRef` elements in the taxonomy schemas that have been identified as supporting the facts being presented. The label extended link locations may also be specified by `linkbaseRef` elements in the XBRL instance itself.

When processing an XBRL instance, consuming applications MUST use all of the linkbases referenced directly or indirectly in this way, if they are relevant to the processing activities. All references to taxonomy schemas and linkbases MUST be resolved when determining the DTS supporting an XBRL instance.

3.3 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 document 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.

3.4 Validation

XBRL instances, XBRL linkbases and XBRL taxonomy schemas MUST comply with the syntax requirements imposed in this specification. Many of these syntax requirements are expressed using XML Schemas so a part of the validation process can be performed using XML Schema validation software. Some of these syntax requirements are not or cannot be expressed using XML Schemas and so, MUST be handled using other validation technologies.

Consuming applications MAY also check that the data in an XBRL instance is consistent with the semantics expressed in the DTS supporting the instance. Semantic inconsistencies do not invalidate the XBRL documents in which they occur. However, this specification identifies the semantic inconsistencies that can be tested for by fully conformant XBRL processors.

3.5 XLink in XBRL

Links between XML fragments occur in many forms in XBRL. There are links between XBRL instances and their supporting DTS. There are links between XBRL instance facts and the footnotes that can connect them. There are links between concept syntax definitions and their semantics, defined in linkbases. The semantics themselves are expressed in the networks of links that constitute the linkbases. XBRL expresses all of these links using the syntax defined in the XLink standard [XLINK]. XBRL uses both the simple links and the extended links defined in the XLink standard.

The XLink standard specifies the syntax and semantics for a set of attributes in the XLink namespace, http://www.w3.org/1999/xlink. These attributes can then be used on elements defined in another namespace, to document various kinds of links between XML fragments. Many of these attributes are used extensively in XBRL. Others have no semantics that are relevant to the links defined by XBRL. These attributes are permitted by the XML Schema syntax constraints but they are not documented or given any specific semantics in this specification. Examples include the `xlink:show` and the `xlink:actuate` attributes.

This section documents the generic forms of the simple links and the extended links used in XBRL. Specific elements that use the simple link or extended link syntax are documented in detail in the relevant sections of this specification dealing with the syntax of XBRL instances or the syntax of XBRL taxonomies.

The syntax of the generic XLink structures used by XBRL is constrained by two XML Schemas: the *xlink.xsd (normative)* that defines the syntax for the XLink attributes; and the *xl.xsd (normative)* that defines the content models for the various kinds of link-related elements defined by this specification.

3.5.1 Simple links

A simple link is a link from one resource to another [XLINK] http://www.w3.org/TR/xlink/#simple-links. XBRL uses simple links to point to linkbases from XBRL instances and from taxonomy schemas (See Section 4.3). XBRL also uses simple links to point to taxonomy schemas from XBRL instances (See Section 4.2). XBRL uses simple links only to connect XML resources.

The XML Schema constraints on the simple links used by XBRL are shown below.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <complexType name="simpleType">
    <annotation>
      <documentation>
      Type for the simple links defined in XBRL
```

```
        </documentation>
     </annotation>
     <complexContent>
       <restriction base="anyType">
         <attribute ref="xlink:type" use="required">
           <annotation>
             <documentation>
             This attribute must have a value of "simple".
             </documentation>
           </annotation>
         </attribute>
         <attribute ref="xlink:href" use="required" />
         <attribute ref="xlink:arcrole" use="optional" />
         <attribute ref="xlink:role" use="optional" />
         <attribute ref="xlink:title" use="optional" />
         <attribute ref="xlink:show" use="optional" />
         <attribute ref="xlink:actuate" use="optional" />
       </restriction>
     </complexContent>
   </complexType>
   <element name="simple" type="xl:simpleType" abstract="true">
     <annotation>
       <documentation>
       The abstract element at the head of the simple link substitution group.
       </documentation>
     </annotation>
   </element>

</schema>
```

3.5.1.1 The xlink:type attribute on simple links

The `xlink:type` attribute MUST occur and MUST have the fixed content "`simple`".

3.5.1.2 The xlink:href attribute on simple links

The `xlink:href` attribute of simple link MUST occur and MUST contain a URI. The URI MUST point to an XML document or to an XML fragment within an XML document. If the URI is relative, it MUST be resolved to obtain an absolute URI as specified in XML Base specification [XML Base].

To point to a particular XML element, the URI MUST end in a fragment identifier. According to the XLink specification, XPointer [XPTR] syntax is allowed in the fragment identifier. However, XBRL restricts the allowed forms of the fragment identifier.

Fragment identifier	Notes
#*id*	The node pointed to MUST be an element with an id attribute whose content is *id*.
#xpointer(*[XPATH]* *expression*)	The expression MUST NOT use XPointer [XPTR] extension functions.

3.5.1.3 The xlink:role attribute on simple links (optional)

The optional `xlink:role` attribute MUST take URI values. If it is provided, the `xlink:role` attribute MUST NOT be empty.

3.5.1.4 The xlink:arcrole attribute on simple links (optional)

If it occurs, the `xlink:arcrole` attribute MUST NOT be an empty string. For simple links that point to linkbases (See Section 4.3), the `xlink:arcrole` attribute MUST have the XLink standard fixed content:

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

3.5.2 Linkbases

The XLink standard defines linkbases as "documents containing collections of inbound and third-party links are called link databases, or linkbases". XBRL uses linkbases to define the semantics of concepts whose syntax is governed by taxonomy schemas. Linkbases are XML elements that are extended links or that contain extended links. Linkbases can also contain `documentation` elements.

The `linkbase` element is defined as a specific linkbase container. The XML Schema constraints on the `linkbase` element are shown below.

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

  <element name="linkbase">
    <annotation>
      <documentation>
      Definition of the linkbase element. Used to
      contain a set of zero or more extended link elements.
      </documentation>
    </annotation>
    <complexType>
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="link:documentation"/>
        <element ref="xl:extended"/>
      </choice>
      <attribute name="id" type="IDREF" use="optional" />
      <attribute ref="xml:base"/>
    </complexType>
  </element>

</schema>
```

Example 1. A skeletal linkbase
```
<linkbase
  xmlns="http://www.xbrl.org/2003/linkbase"
  xmlns:samp="http://www.xbrl.org/sample"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.xbrl.org/sample samp001.xsd"
  xml:base="http://www.xbrl.org/sample">

  <calculationLink xlink:type="extended">
    <!-- ... -->
  </calculationLink>

</linkbase>
```
Meaning: Use of `linkbase` as the root element, holding namespace prefix definitions and the `schemaLocation` attribute. The "`xml:`" prefix need not be declared. One extended link element, the `calculationLink`, is contained in the linkbase.

3.5.2.1 The id attribute on the xbrl element

The `linkbase` element MAY have an `id` attribute. The value of the `id` attribute MUST conform to the [XML] rules for attributes with the ID type (http://www.w3.org/TR/REC-xml#NT-TokenizedType).

3.5.2.2 The xml:base attribute on the linkbase element (optional)

The `xml:base` attribute [XML Base] MAY appear on the `linkbase` element, participating in the resolution of relative URIs in the contained extended links.

3.5.2.3 Documentation elements in extended links (optional)

All `linkbase` elements MAY also contain `documentation` elements.

The XML Schema constraints on the documentation element are shown below.

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

  <element name="documentation" substitutionGroup="xl:documentation">
    <annotation>
      <documentation>
      Concrete element to use for documentation of
      extended links and linkbases.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:documentationType">
          <attribute ref="xml:lang" use="required" />
          <anyAttribute namespace="##other" processContents="lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

The `documentation` element MUST have string content and MAY contain an xml:lang attribute, indicating the language used for the documentation. The documentation element MAY also contain any attribute that is not defined in the XBRL linkbase namespace, `http://www.xbrl.org/2003/linkbase`.

3.5.3 Extended links

Extended links are XLink annotated XML fragments that document a set of relationships between resources. XBRL extended links document relationships between resources that are XML fragments.

The generic XML Schema constraints on the extended links used by XBRL are shown below.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <complexType name="extendedType">
    <annotation>
      <documentation>
      Generic extended link type
      </documentation>
    </annotation>
    <complexContent>
```

```
      <restriction base="anyType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title" />
          <element ref="xl:documentation" />
          <element ref="xl:locator" />
          <element ref="xl:arc" />
          <element ref="xl:resource" />
        </choice>
        <attribute ref="xlink:type" use="required">
          <annotation>
            <documentation>
            This attribute must have a value of "extended".
            </documentation>
          </annotation>
        </attribute>
        <attribute ref="xlink:role" use="required" />
        <attribute ref="xlink:title" use="optional" />
        <attribute name="id" type="ID" use="optional" />
      </restriction>
    </complexContent>
  </complexType>
  <element name="extended" type="xl:extendedType" abstract="true">
    <annotation>
      <documentation>
      Abstract extended link element at head of extended link substitution group.
      </documentation>
    </annotation>
  </element>

</schema>
```

XBRL extended links MAY contain five different types of child elements:

- `documentation` elements;

- `title` elements (titles);

- `locator` elements (locators);

- `resource` elements (resources); and

- `arc` elements (arcs).

The `documentation` element is for XBRL documentation purposes only and has no XLink specific semantics. Titles, locators, resources and arcs are identified by specific XLink attributes. If the titles, locators, resources and arcs are not direct children of an extended element, then they have no [XLINK] specified meaning, and hence have no XBRL-specified meaning.

Extended links are decorated by a range of different attributes. These are described below.

3.5.3.1 The id attribute (optional)

Extended links MAY have an `id` attribute. The value of the `id` attribute MUST conform to the [XML] rules for attributes with the ID type (See http://www.w3.org/TR/REC-xml#NT-TokenizedType for details). The `id` attribute identifies an extended link (see Section 4.6) so that it may be referenced directly by simple links.

3.5.3.2 The xlink:type attribute on extended links

The `xlink:type` attribute MUST occur on extended links and MUST have the fixed content "extended".

3.5.3.3 The xlink:role attribute on extended links (optional)

The content of the `xlink:role` attribute is referred to as the extended link role value. The extended link role value MUST be used by applications to partition extended links into separate networks of relationships. See Section 5.1.2 for details on how the semantics embodied in extended link arcs is contingent on extended link arc role values. If the attribute is missing then its default value, for the purposes of extended link partitioning, is:

```
http://www.xbrl.org/2003/role#link
```

3.5.3.4 Documentation elements in extended links (optional)

All XBRL extended links MAY contain `documentation` elements.

The `documentation` elements in extended links conform to the same syntax requirements that apply to `documentation` elements in `linkbase` elements. See Section 3.5.2.1 for details.

3.5.3.5 Titles in extended links (optional)

All XBRL extended links MAY contain titles. Titles may be used to document extended links, as an alternative to the more limited `xlink:title` attributes. They are particularly useful where information needs to be provided in multiple languages. Titles have no XBRL specified semantics. To use a title, it is necessary to define a new element that is in the substitution group for the abstract `title` element.

The XML Schema constraints on the titles are shown below.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <complexType name="titleType">
    <annotation>
      <documentation>
      Type for the abstract title element -
      used as a title element template.
      </documentation>
    </annotation>
    <complexContent>
      <restriction base="anyType">
        <attribute ref="xlink:type" use="required">
          <annotation>
            <documentation>
            This attribute must have a value of "title".
            </documentation>
          </annotation>
        </attribute>
      </restriction>
    </complexContent>
  </complexType>
  <element name="title" type="xl:titleType" abstract="true">
    <annotation>
      <documentation>
      Generic title element for use in extended link documentation.
      Used on extended links, arcs, locators.
      See http://www.w3.org/TR/xlink/#title-element for details.
      </documentation>
    </annotation>
  </element>
```

```
</schema>
```

3.5.3.5.1 The xlink:type attribute on titles

The `xlink:type` attribute must occur on all titles and MUST have the fixed content "`title`".

3.5.3.6 Locators

Locators are child elements of an extended link that point to resources external to the extended link itself. All XBRL extended links MAY contain locators.

The XML Schema constraints on generic locators are shown below.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <complexType name="locatorType">
    <annotation>
      <documentation>
      Generic locator type.
      </documentation>
    </annotation>
    <complexContent>
      <restriction base="anyType">
        <sequence>
          <element ref="xl:title" minOccurs="0" maxOccurs="unbounded" />
        </sequence>
        <attribute ref="xlink:type" use="required">
          <annotation>
            <documentation>
            This attribute must have a value of "locator".
            </documentation>
          </annotation>
        </attribute>
        <attribute ref="xlink:href" use="required" />
        <attribute ref="xlink:label" use="required" />
        <attribute ref="xlink:role" use="optional" />
        <attribute ref="xlink:title" use="optional" />
      </restriction>
    </complexContent>
  </complexType>
  <element name="locator" type="xl:locatorType" abstract="true">
    <annotation>
      <documentation>
      Abstract locator element to be used as head of locator substitution group
      for all extended link locators in XBRL.
      </documentation>
    </annotation>
  </element>

</schema>
```

For consistency, the `loc` element is the only locator defined for use in XBRL extended links. The `loc` element is a concrete version of the generic locator. The XML Schema syntax constraints on the `loc` element are shown below.

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

```
  xmlns:xl="http://www.xbrl.org/2001/XLink"
  elementFormDefault="qualified">

  <element name="loc" type="xl:locatorType" substitutionGroup="xl:locator">
    <annotation>
      <documentation>
      Concrete locator element. The loc element is the
      XLink locator element for all extended links in XBRL.
      </documentation>
    </annotation>
  </element>

</schema>
```

3.5.3.6.1 The xlink:type attribute on locators

The `xlink:type` attribute must occur on all locators and MUST have the fixed content "`locator`".

3.5.3.6.2 The xlink:href attribute on locators

The `xlink:href` attribute on locators conforms to the same requirements applying to the `xlink:href` attribute on simple links. See Section 3.5.1.2 for details.

3.5.3.6.3 The xlink:label attribute on locators

The `xlink:label` attribute on a locator identifies the locator so that it can be referenced by arcs in the same extended link. Multiple locators and resources in an extended link MAY have the same `xlink:label` attribute value. The `xlink:label` attribute value MUST be an NCName [XML] (http://www.w3.org/TR/REC-xml-names/#NT-NCName). This requirement means that `xlink:label` attributes MUST begin with a letter or an underscore.

3.5.3.6.4 Titles on locators (optional)

Locators MAY contain titles. Title children of locators MUST conform to the same restrictions applying to title children of extended links. See Section 3.5.3.5 for details.

3.5.3.7 Resources

Some XBRL extended links MAY contain resources. A resource is an XML fragment in an extended link that is related to other resources in the extended link and to resources outside of the extended link.

The XML Schema constraints on generic resources are shown below.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <complexType name="resourceType">
   <annotation>
     <documentation>
     Generic type for the resource type element
     </documentation>
   </annotation>
   <complexContent>
     <restriction base="anyType">
       <attribute ref="xlink:type" use="required">
```

```
        <annotation>
          <documentation>
          This attribute must have a value of "resource".
          </documentation>
        </annotation>
      </attribute>
      <attribute ref="xlink:label" use="required" />
      <attribute ref="xlink:role" use="optional" />
      <attribute name="id" type="ID" use="optional" />
    </restriction>
  </complexContent>
</complexType>
<element name="resource" type="xl:resourceType" abstract="true">
  <annotation>
    <documentation>
    Abstract element to use as head of resource element substitution group.
    </documentation>
  </annotation>
</element>

</schema>
```

The content of generic resources is very loosely constrained. More specific constraints are applied by this specification for specific kinds of resources in specific kinds of extended links.

3.5.3.7.1 The xlink:type attribute on resources

The `xlink:type` attribute MUST occur on all resources and MUST have the fixed content "`resource`".

3.5.3.7.2 The xlink:label attribute on resources

The `xlink:label` attribute on a resource identifies the resource so that it can be referenced by arcs in the same extended link. The `xlink:label` attribute on resources conforms to the same requirements applying to the `xlink:label` attribute on locators. See Section 3.5.3.6.3 for details. Several resources in an extended link MAY have the same label.

3.5.3.7.3 The xlink:role attribute on resources (optional)

The optional `xlink:role` attribute on a resource is referred to as the resource role.

Resources MAY contain an `xlink:role` attribute, which SHOULD distinguish between resources based on the nature of the information that they contain. Some of the resources defined in this specification have a set of standard resource role values defined for them.

3.5.3.7.4 The id attribute on resources (optional)

The `id` attribute MAY occur on all resources in XBRL extended links. The value of the `id` attribute MUST conform to the [XML] rules for attributes with the ID type (http://www.w3.org/TR/REC-xml#NT-TokenizedType). The `id` attribute identifies the resource so that it may be referenced by `locators` in other extended links for the purposes of arc prohibition (See Section 3.5.3.8.5).

3.5.3.8 Arcs

All XBRL extended links MAY contain `arcs`. Arcs document relationships between resources identified by locators in extended links or occurring as resources in extended links.

Arcs join the elements referenced in their XLink attributes: `xlink:from` and `xlink:to`. These two attributes contain the `xlink:label` attribute values of either locators or resources within the same extended link as the arc itself. The relationship documented by an arc is a

relationship running from the resource or locator identified by the `xlink:from` attribute to the resource or locator identified by the `xlink:to` attribute.

The XML Schema constraints on generic arcs are shown below.

```xml
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <simpleType name="useEnum">
    <annotation>
      <documentation>
      Enumerated values for the use attribute  on extended link arcs.
      </documentation>
    </annotation>
    <restriction base="NMTOKEN">
      <enumeration value="optional" />
      <enumeration value="prohibited" />
    </restriction>
  </simpleType>
  <complexType name="arcType">
    <annotation>
      <documentation>
      basic extended link  arc  type - extended  where necessary  for  specific arcs
      Extends  the  generic  arc  type by  adding use,  priority and order attributes.
      </documentation>
    </annotation>
    <complexContent>
      <restriction base="anyType">
        <sequence>
          <element ref="xl:title" minOccurs="0" maxOccurs="unbounded" />
        </sequence>
        <attribute ref="xlink:type" use="required">
          <annotation>
            <documentation>
            This attribute must have a value of "arc".
            </documentation>
          </annotation>
        </attribute>
        <attribute ref="xlink:from" use="required" />
        <attribute ref="xlink:to" use="required" />
        <attribute ref="xlink:arcrole" use="required" />
        <attribute ref="xlink:title" use="optional" />
        <attribute ref="xlink:show" use="optional" />
        <attribute ref="xlink:actuate" use="optional" />
        <attribute name="order" type="decimal" use="optional" />
        <attribute name="use" type="xl:useEnum" use="optional" />
        <attribute name="priority" type="integer" use="optional" />
        <anyAttribute namespace="##other" processContents="lax" />
      </restriction>
    </complexContent>
  </complexType>
  <element name="arc" type="xl:arcType" abstract="true">
    <annotation>
      <documentation>
      Abstract element to use as head of arc element substitution group.
      </documentation>
    </annotation>
  </element>
```

```
</schema>
```

Example 2. Correct use of arcs according to [XLINK]

[XLINK] forbids duplicate arcs within a single extended link and ignores `arcrole` in determining duplicates so the following example is invalid (See Section 5.2.6 for details of `definitionLink` extended links):

```
<definitionLink xlink:type="extended">
  <definitionArc from="a" to="b"
arcrole="http://www.xbrl.org/2003/role#general-special" />
  <definitionArc from="a" to="b"

  arcrole="http://www.xbrl.org/2003/role#requires-target"/>
</definitionLink>
```

instead, the following construction MUST be used:

```
<definitionLink xlink:type="extended">
  <definitionArc from="a" to="b"
arcrole="http://www.xbrl.org/2003/role#general-special" />
</definitionLink>
<definitionLink>
  <definitionArc from="a" to="b"

  arcrole="http://www.xbrl.org/2003/role#requires-target"/>
</definitionLink>
```

3.5.3.8.1 The xlink:type attribute on arcs

The `xlink:type` attribute MUST occur on all arcs and MUST have the fixed content "`arc`".

3.5.3.8.2 The xlink:from attribute

The `xlink:from` attribute on an arc MUST be equal to the value of an `xlink:label` attribute of at least one locator or resource in the same extended link element as the arc element itself.

The `xlink:from` attribute value MUST be an NCName [XML] (http://www.w3.org/TR/REC-xml-names/#NT-NCName). This requirement means that `xlink:from` attributes MUST begin with a letter or an underscore.

3.5.3.8.3 The xlink:to attribute

The `xlink:to` attribute on an arc MUST be equal to the value of an `xlink:label` attribute of at least one locator or resource in the same extended link element as the arc element itself.

The `xlink:to` attribute value MUST be an NCName [XML] (http://www.w3.org/TR/REC-xml-names/#NT-NCName). This requirement means that `xlink:to` attributes MUST begin with a letter or an underscore.

3.5.3.8.4 The xlink:arcrole attribute

The `xlink:arcrole` attribute documents the specific kind of relationship being expressed by the arc. Its value is referred to as an arc role value. Its value MUST be an absolute URI. A set of standard arc role values are defined and given specific meaning in this specification for each arc element. These are documented in the sections describing the specific XBRL arc elements (`labelArc`, `referenceArc`, `calculationArc`, `definitionArc`, `presentationArc`, and `footnoteArc`) on which they are to be used.

New arc role values MAY be defined in taxonomy schemas. The semantics for new arc role values are defined using the `arcroleType` element (see Section 5.1.4). An arc role value MUST be an absolute URI that can be resolved to address the fragment of an XML Schema document containing the `roleType` element.

All arc role values in XBRL extended links follow a special URI format. The arc role value MUST identify the `arcroleType` element (see Section 5.1.4) where the semantics and syntax of the arc role value is defined. A fragment identifier in the arc role value is used to identify the specific `arcroleType` element within the document identified by the URI. The XBRL standard arc role values are defined in the XBRL role schema (See *xbrl-role-2003-07-31.xsd (normative)*).

3.5.3.8.5 Prohibiting arcs

A taxonomy author will generally not have write permissions on linkbases created by other taxonomy authors. In situations where a taxonomy author needs to modify the relationships expressed in linkbases that they cannot alter directly, they may create new linkbases that override specific arcs in the linkbases to be modified.

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

The only option available to prohibit relationships in the third party's extended link is to create a new equivalent arc in a new extended link that negates the original arc in the original extended link.

The syntax for prohibiting arcs is provided by two attributes, `use` and `priority`, that are available on all arc elements defined in this specification.

3.5.3.8.5.1 Arc equivalence

XLink does not define the interaction of arcs from multiple linkbases in any way. It only specifies that arc elements connecting the same "from" and "to" labels MUST appear in different extended 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 formally defined as follows:

Two arcs are equivalent if and only if all of the following conditions are true:

    i.   They have s-equal element names, and
    ii.   Their containing extended links have element names that are s-equal, and
    iii.   Their arc role attributes have s-equal content, and
    iv.   The extended links in which they are contained have roles with s-equal content, and
    v.   Their `xlink:from` attributes have the label of locator elements with s-equal `href` content, and
    vi.   For arcs relating concepts, their `xlink:to` attributes have the `xlink:label` of locator elements that have s-equal `href` content, and
    vii.   For arcs relating concepts to an extended link resource, both arcs MUST have `xlink:to` attributes that are either equal to the `xlink:label` attribute values on locators in their own extended links that point to the resource or, if the arcs are in the extended link that also contains the resource, that are equal to the `xlink:label` attribute value on the resource itself.

3.5.3.8.5.2 The use attribute (optional)

The optional `use` attribute MUST take one of two possible values – "`optional`", or "`prohibited`".

`use="optional"` indicates that the arc MAY be traversed. This is the default value that MUST be inferred for the `use` attribute if the `use` attribute is not specified.

`use="prohibited"` indicates that any equivalent (See Section 3.5.3.8.5.1) arc with a lower priority (See Section3.5.3.8.5.3) MUST NOT be traversed because it is a prohibited arc.

As a motivating example, consider the situation of a third party desiring to create a new "sub-total" concept intervening between "children" concepts that already have summation-item arcs to the "total" concept (See Section 5.2.5.2 for details about summation-item arcs and calculation relationships in extended links). The creator of the sub-total concept will add arcs from the sub-total concept to the children concepts and from the total concept to the sub-total concept. There would then be two paths from the children concepts to the total concept, one using the new arcs through the sub-total concept, and the other using the original arcs direct from the summation concept. In the case of calculation links, this could result in the double counting of values. The creator of the sub-total concept SHOULD create prohibiting arcs to prevent this, preventing traversal of the arcs going directly from the total concept to the children concepts.

Example 3. Using arc prohibition to insert a new sub-total into a calculation network
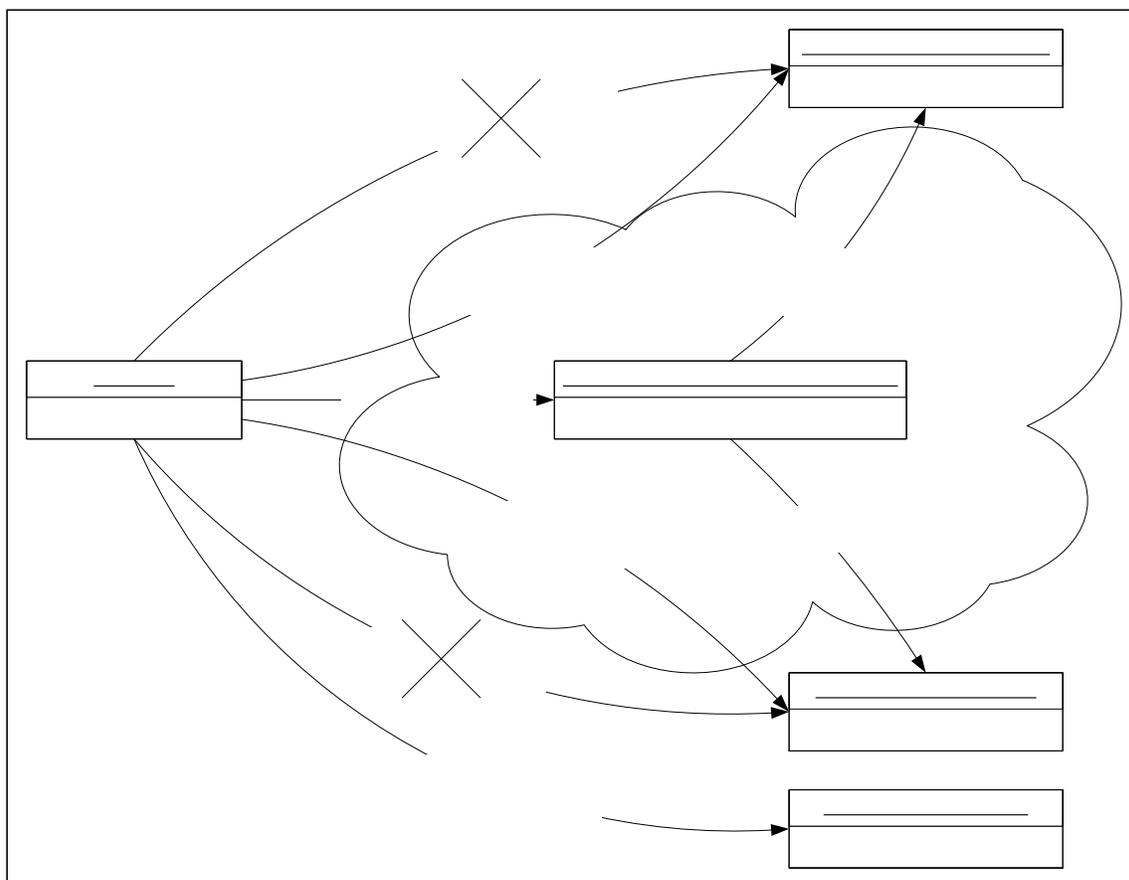
Arc prohibition is also defined for arcs from concepts to resources, such as `label` and `reference` elements. Prohibition of an arc to a resource implies that the resource itself has been disassociated from the concept. To prohibit an arc to a resource element in another extended link, a prohibiting arc MUST connect to a locator that points to the resource participating in the relationship being prohibited.

3.5.3.8.5.3 The priority attribute (optional)

The content of the `priority` attribute is an integer. Given two equivalent arcs, the behaviour of an XBRL application MUST be based on the values of the priority attributes of the two arcs. The arc with the numerically larger `priority` attribute value MUST override the other arc, being the only one of the two arcs that is traversed. 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 Section 5.2) 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.

3.5.3.8.6 The order attribute (optional)

The optional `order` attribute MUST have a decimal value that that indicates the order in which applications MUST display siblings when hierarchical networks of arcs are being displayed. If missing, the `order` attribute value MUST default to "1". If multiple siblings in the hierarchy 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 in between two previously defined siblings.

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

3.5.3.8.7 Titles on arcs (optional)

Arcs MAY contain titles. Title children of arcs MUST conform to the same restrictions applying to title children of extended links. See Section 3.5.3.5 for details.

4   XBRL instances (the xbrl element)

An overview of XBRL instances is provided in Section 3.2.

XBRL instances are XML fragments with root element, `xbrl`. XBRL instances contain facts, with each fact corresponding to a concept defined in their supporting DTS. XBRL instances also contain `context` and `unit` elements that provide additional information needed to interpret the facts in the instance.

Facts can be simple, in which case their values are expressed as simple content (except in the case of simple facts whose values are expressed as a ratio), and facts can be compound, in which case their values are made up from other simple and/or compound facts. Simple facts are expressed using items (and are referred to as items in this specification) and compound facts are expressed using tuples (and are referred to as tuples in this specification).

Although the syntax for any given tuple or item can only be defined in a single taxonomy schema, an XBRL instance MAY contain XBRL items and tuples from any number of taxonomy schemas.

XBRL instances identify the taxonomy schemas and XBRL linkbases that make up the starting points for discovery of the DTS that supports them. Section 3.2 documents how the DTS supporting an XBRL instance is to be determined.

The taxonomy schemas and the linkbases used as starting points in DTS discovery are identified via the `schemaRef` elements `linkbaseRef` elements in XBRL instances respectively. This enables XBRL instances to exert some control over the interpretation of the information that they report.

For example, the same set of elements defined in a taxonomy schema might have Spanish and Portuguese literature references defined in different linkbases (that are not referenced directly from that schema). An instance might provide access to both or neither of these linkbases in order to specify which set of references they consider to be more appropriate.

An XBRL instance MUST comply with the rules specified herein. The syntax for XBRL instances is defined using a set of XML Schemas. Example elements defined in the XBRL instance schema, *xbrl-instance-2003-07-31.xsd (normative)*, include `xbrl`, `item`, `context`, `unit`, and `tuple`. All XBRL instances MUST be valid XML documents as defined by XML Schema [SCHEMA-1].

The semantics of XBRL instances and their contents are specified only insofar as they impact the operation of software applications that use this specification.

Expressing even a single fact in an XBRL instance requires multiple elements, at least one item element (see Section 4.1.1) and a context element containing sub-elements (see Section 4.5 below). Therefore, a container element is necessary to serve as the root element of an XBRL instance. This container is the xbrl element. If multiple "data islands" of XBRL mark-up are included in a larger document, the xbrl element is the container for each. The XML Schema definition for the xbrl element is set out below.

The XML Schema constraints on the xbrl element are shown below.

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

  <element name="xbrl">
    <annotation>
      <documentation>
      XBRL instance root element.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="link:schemaRef" minOccurs="1" maxOccurs="unbounded" />
        <element ref="link:linkbaseRef" minOccurs="0" maxOccurs="unbounded" />
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xbrli:item" />
          <element ref="xbrli:tuple" />
          <element ref="xbrli:context" />
          <element ref="xbrli:unit" />
          <element ref="link:footnoteLink" />
        </choice>
      </sequence>
      <attribute name="id" type="ID" use="optional" />
      <attribute ref="xml:base" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
    </complexType>
  </element>

</schema>
```

Example 4. Use of xbrl as the root element

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:xlink="http://www.xbrl.org/2001/XLink"
 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">
  <schemaRef xlink:type="simple"
             xlink:href="http://www.xbrl.org/us/fr/ci/2000-07-31/usfr-ci-2003.xsd"
             xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
  <ci:assets contextRef="c1">727</ci:assets>
  <ci:liabilities contextRef="c1">635</ci:liabilities>
  <context id="c1"><!-- ... --></context>
</xbrl>
```

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

4.1.1 The id attribute on the xbrl element (optional)

The `xbrl` element MAY have an `id` attribute. The value of the `id` attribute MUST conform to the [XML] rules for attributes with the ID type (http://www.w3.org/TR/REC-xml#NT-TokenizedType).

4.1.2 The xml:base attribute on the xbrl element (optional)

The `xbrl` element MAY have an `xml:base` attribute. The `xml:base` attribute [XML Base] MAY appear on the `xbrl` element, participating in the resolution of relative URIs in the XBRL instance.

4.2 The schemaRef element

Every XBRL instance MUST contain at least one `schemaRef` element. The `schemaRef` element is a simple link, as defined in Section 3.5.1. The `schemaRef` element MAY occur as a child element of an `xbrl` element. All `schemaRef` elements in an XBRL instance MUST occur before other children of the `xbrl` element, in document order.

In an XBRL instance document, the `schemaRef` element points to a taxonomy schema that becomes part of the DTS supporting that XBRL instance.

The XML Schema definition of the `schemaRef` element is shown below.

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
        xmlns:xl="http://www.xbrl.org/2001/XLink"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        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">

  <element name="schemaRef" type="link:simpleType"/>

</schema>
```

4.2.1 The xlink:type attribute on schemaRef elements

The `xlink:type` attribute MUST occur and MUST have the fixed content "`simple`".

4.2.2 The xlink:href attribute on schemaRef elements

The `xlink:href` attribute MUST contain a URI. The URI MUST point to an XML Schema. If the URI reference is relative, its absolute version MUST be determined as specified in XML Base [XML Base] before use.

4.2.3 The xlink:arcrole attribute on schemaRef elements (optional)

The `xlink:arcrole` attribute May be used on the `schemaRef` element. It is given no semantics by this specification

4.2.4 The xlink:role attribute on schemaRef elements (optional)

The `xlink:role` attribute May be used on the `schemaRef` element. No semantics are defined for the `xlink:role` attribute when it occurs on the `schemaRef` element.

4.3 The linkbaseRef element in XBRL instances

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 (See Section 4.3.3).

One or more `linkbaseRef` elements MAY occur as children of the `xbrl` element (They MAY also occur in taxonomy schemas. See Section 0 for details). If `linkbaseRef` elements occur as children of xbrl elements, they MUST follow the schemaRef elements and precede all other elements, in document order.

In an XBRL instance document, the `linkbaseRef` element identifies a linkbase that becomes part of the DTS supporting that XBRL instance.

The XML Schema constraints applying to the `linkbaseRef` element are shown below.

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

  <element name="linkbaseRef" substitutionGroup="xl:simple">
    <annotation>
      <documentation>
      Definition of the linkbaseRef element - used
      to link to XBRL taxonomy extended links from
      taxonomy schema documents and from XBRL
      instance documents.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:simpleType">
          <attribute ref="xlink:arcrole" use="required">
            <annotation>
              <documentation>
              This attribute must have the value:
              http://www.w3.org/1999/xlink/properties/linkbase
              </documentation>
            </annotation>
          </attribute>
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

4.3.1 The xlink:type attribute on linkbaseRef elements

The `xlink:type` attribute MUST occur and MUST have the fixed content "`simple`".

4.3.2 The xlink:href attribute on linkbaseRef elements

The `xlink:href` attribute of the `linkbaseRef` element MUST contain a URI. The URI MUST point to a linkbase (as defined in Section 3.5.2) that contains the appropriate extended links, as determined by the value of the `xlink:role` attribute. If the URI reference is relative, its absolute version MUST be determined as specified in XML Base [XML Base] before use.

4.3.3 The xlink:arcrole attribute

To indicate that the linkbaseRef element points to a linkbase, the `xlink:arcrole` attribute on the `linkbaseRef` element MUST have the XLink standard fixed content:

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

4.3.4 The xlink:role attribute on linkbaseRef elements (optional)

The optional `xlink:role` attribute constrains the kinds of extended links that are permitted within the linkbase identified by the `linkbaseRef` element. Table 2 sets out the standard `xlink:role` attribute values for the `xlink:role` attribute when it occurs on the `linkbaseRef` element. Table 2 also documents which kinds of extended links:

- MUST be contained by the linkbase connected to by a `linkbaseRef` element with each of the standard `xlink:role` attribute values; and

- MUST NOT be contained by the linkbase connected to by a `linkbaseRef` element with each of the standard `xlink:role` attribute values.

If a `linkbaseRef` element connects to a linkbase containing an extended link that has not been defined in this specification, then a non-standard value of the `xlink:role` attribute MAY be used or the `xlink:role` attribute MAY be omitted.

Table 2. Roles in the linkbaseRef element

Values of the `linkbaseRef xlink:role` **attribute**	Element pointed to by `xlink:href`
(unspecified)	MAY contain any extended 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

4.4 Items

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 its own right in an XBRL instance. Therefore, all elements representing single 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 taxonomy schemas that contain such element definitions.

`item` elements MUST NOT be descendants of other `item` elements. Structural relationships necessary in an XBRL document instance MUST be captured only using tuples (see Section 4.7). 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 xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
```

```
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  targetNamespace="http://www.xbrl.org/2003/instance"
  elementFormDefault="qualified">

  <attributeGroup name="numericItemAttrs">
    <annotation>
      <documentation>
      Group of attributes for numeric items
      </documentation>
    </annotation>
    <attribute name="contextRef" type="IDREF" use="required" />
    <attribute name="unitRef" type="IDREF" use="required" />
    <attribute name="precision" type="xbrli:precisionType" use="optional" />
    <attribute name="decimals" type="xbrli:decimalsType" use="optional" />
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="##other" processContents="lax" />
  </attributeGroup>
  <attributeGroup name="nonNumericItemAttrs">
    <annotation>
      <documentation>
      Group of attributes for non-numeric items
      </documentation>
    </annotation>
    <attribute name="contextRef" type="IDREF" use="required" />
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="##other" processContents="lax" />
  </attributeGroup>

  <annotation>
    <documentation>
    XBRL domain numeric item types - for use on concept element definitions
    The following 4 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.
    </documentation>
  </annotation>
  <complexType name="monetaryItemType" final="extension">
    <simpleContent>
      <extension base="xbrli:monetary">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="sharesItemType" final="extension">
    <simpleContent>
      <extension base="xbrli:shares">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="pureItemType" final="extension">
    <simpleContent>
      <extension base="xbrli:pure">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="fractionItemType" final="extension">
    <sequence>
      <element name="numerator" type="decimal" />
      <element name="denominator" type="xbrli:nonZeroDecimal" />
    </sequence>
    <attribute name="contextRef" type="IDREF" use="required" />
```

```
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="##other" processContents="lax" />
  </complexType>

  <complexType name="stringItemType" final="extension">
    <simpleContent>
      <extension base="string">
        <attributeGroup ref="xbrli:nonNumericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
<!--
booleanItemType, hexBinaryItemType, base64BinaryItemType, anyURIItemType,
uriItemType, QNameItemType, durationItemType, dateTimeItemType, timeItemType,
dateItemType, gYearMonthItemType, gYearItemType, gMonthDayItemType, gDayItemType,
gMonthItemType, normalizedStringItemType, tokenItemType, languageItemType,
NameItemType,  ...
-->

  <element name="item" type="anyType" abstract="true">
    <annotation>
      <documentation>
      Abstract item element used as head of item substitution group
      </documentation>
    </annotation>
  </element>

</schema>
```

Example 5. A numeric fact with three significant digit

```
<ci:capitalLeases contextRef="c1" unitRef="u1"
precision="3">727000</ci:capitalLeases>
```

Meaning: The value of Capital Leases in the numeric context labelled c1 is 727000 accurate to 3 significant figures. Note that it will be necessary to consult the context (defined below) in order to determine other details concerning the value such as entity, period, etc. and it will be necessary to consult the referenced `unit` element to determine the relevant unit information.

Example 6. A non-numeric item

```
<ci:concentrationsNote contextRef="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` must have 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 appropriate built-in types of XML Schema, along with the `fractionItemType` type. The complete list is in Section 5.1.1.3. An item MUST NOT have complex content unless its item type is derived by restriction from `fractionItemType`.

The `contextRef` attribute is an `IDREF` to the `context` element (see Section 4.5) that holds additional relevant information about the fact represented. An item MUST contain a `contextRef` 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 content is derived from an XML Schema built-in numeric type (`decimal`, `float` or double or a built-in type derived from one of them) or `fractionItemType` by restriction MUST use the `contextRef` attribute and the `unitRef` attribute; all others MUST use the `contextRef` attribute.

The `unitRef` attribute is an `IDREF` to the `unit` element (see Section 4.6) that holds information about units in which numeric facts have been measured. The `unitRef` attribute MUST NOT occur in non-numeric items. The `unitRef` attribute MUST occur in numeric items, referencing a `unit` element in the same XBRL instance.

Two optional attributes, `precision` and `decimals`, are available on numeric items (except those with type `fractionItemType`) to enable the XBRL instance creator to make statements about the accuracy of the facts represented. They are discussed in the following sections.

4.4.1 Usage of precision and decimals attributes

A numeric item MUST have either a `precision` attribute or a `decimals` attribute unless it is of the `fractionItemType` or of a type that is derived by restriction from `fractionItemType` or has a nil value, in which case, it MUST NOT have either a `precision` attribute or a `decimals` attribute.

A numeric item MUST NOT have both a `precision` attribute and a `decimals` attribute.

A non-numeric item MUST NOT have either a `precision` or a `decimals` attribute.

When determining whether two numeric items are v-equal (a predicate that is used in the definition of various other equality type predicates) it is necessary to take into consideration the values of `precision` (or the precision inferred from the value of the `decimals` attribute) for the two numeric items. The formal definition of v-equal for two numeric items is given in Section 4.8.

4.4.2 The precision attribute (optional)

The precision attribute MUST be a non-negative integer or the string "`INF`" 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 4.4.1 for the normative definition of 'correct to "n" significant figures'). 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 7. 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 xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  targetNamespace="http://www.xbrl.org/2003/instance"
  elementFormDefault="qualified">

  <simpleType name="precisionType">
    <annotation>
      <documentation>
      This type is used to specify the value of 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="string">
          <enumeration value="INF" />
        </restriction>
      </simpleType>
    </union>
  </simpleType>

</schema>
```

4.4.3 The decimals attribute (optional)

The decimals attribute MUST be an integer or the value "INF" 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 4.4.5.2 for the normative definition of 'correct to "n" decimal places').

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

Example 8. 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.	5	15
-1	11	10.	5	15
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

Decimals	Original	Read as	Known to be GE	Known to be LT
-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 xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  targetNamespace="http://www.xbrl.org/2003/instance"
  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="string">
          <enumeration value="INF" />
        </restriction>
      </simpleType>
    </union>
  </simpleType>

</schema>
```

4.4.4 Inferring precision from decimals

The following inference rules are provided to enable XBRL instance consumers to infer `precision` from the `decimals` attribute value.

If, on a numeric item, the `decimals` attribute is present rather than the `precision` attribute, then a consuming application MUST infer the precision of that numeric fact if it is to be used in calculations or searches for duplicate in XBRL instances.

Given the value of the `decimals` attribute, the precision of a numeric item is equal to *n*, 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 9. 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
123.45e5	-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

4.4.5 Definitions pertaining to accuracy

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

4.4.5.1 "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 10. 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	0.000020	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**.

4.4.5.2 "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 0.5×10^{-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

$$e_{abs} = |X-x| \leq 0.5 \times 10^{-n}$$
$$\text{or, because of rounding conventions,}$$
$$-0.5 \times 10^{-n} \leq x-X < 0.5 \times 10^{-n}$$

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 11. 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

4.5 The context element

The `context` element contains information about the entity being described, the reporting period and the reporting scenario, all of which are necessary for understanding a business fact captured as an XBRL item.

The `context` element MUST conform to the following XML Schema constraints:

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

  <element name="context">
    <annotation>
      <documentation>
      Used for an island of context to which facts can be related.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="entity" type="xbrli:contextEntityType" />
        <element name="period" type="xbrli:contextPeriodType" />
        <element name="scenario" type="xbrli:contextScenarioType" minOccurs="0" />
      </sequence>
      <attribute name="id" type="ID" use="required" />
    </complexType>
  </element>

</schema>
```

In the examples, the `xsi:schemaLocation` attribute does not contain URIs to resolve the ISO4217 and NASDAQ namespaces. In the case of NASDAQ the examples assume that the applications that produced and will consume this instance will be able to resolve this namespace reference without the help of the `xsi:schemaLocation`. The ISO4217 namespace does not refer to an XML Schema that can be used for validation of the XBRL instances shown in the examples. The ISO4217 and NASDAQ URIs do not reference actual resources of the ISO or NASDAQ.

4.5.1 The id attribute

Every `context` element MUST include the `id` attribute. The content of the `id` attribute MUST conform to the [XML] rules for attributes with the ID type (http://www.w3.org/TR/REC-xml#NT-TokenizedType). The `id` attribute identifies the context (see Section 4.5) so that it may be referenced by item elements.

Example 12. IDs

Example	`id="C2424"`	
Counterexample	`id="42"`	Content of the ID type must not begin with a number.

4.5.2 The period element

The period element contains the instant or interval of time for reference by an `item` element. 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'.

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

The XML Schema constraints on the `period` element are shown below.

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

  <simpleType name="dateUnion">
    <annotation>
      <documentation>
      The union of the date and dateTime simple types.
      </documentation>
    </annotation>
    <union memberTypes="date dateTime " />
  </simpleType>
  <complexType name="contextPeriodType">
    <annotation>
      <documentation>
      The type for the period element, used to describe the reporting date info.
      </documentation>
    </annotation>
    <choice>
      <sequence>
        <element name="startDate" type="xbrli:dateUnion" />
        <element name="endDate" type="xbrli:dateUnion" />
      </sequence>
      <element name="instant" type="xbrli:dateUnion" />
      <element name="forever">
        <complexType />
      </element>
    </choice>
  </complexType>

</schema>
```

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

While the content of the instant, startDate and endDate elements are defined to use the data representation defined by ISO 8601 (as restricted by [SCHEMA-2]), XBRL adds further restrictions and constraints.

For an item element with periodType="instant" (See Section 5.1.1.1), the period MUST contain an instant element.

For an item element with periodType="duration", the period MUST contain forever or a valid sequence of startDate and endDate.

A date, with no time part, in the content of an startDate element is defined to be equivalent to specifying a dateTime of the same date, and T00:00:00 (midnight at the end of the day).

A date, with no time part, in the endDate or instant element is defined to be equivalent to specifying a dateTime of the same date plus P1D and T00:00:00:00 (midnight at the end of the day but ISO8601 requires that this be specified as midnight at the start of the following day).

If supplied, the endDate MUST imply a point in time that is later than implied point in time of the startDate.

4.5.3 The entity 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 a segment element.

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

  <complexType name="contextEntityType">
    <annotation>
      <documentation>
      The type for the entity element, used to describe the reporting entity.
      Note that the scheme attribute is required and cannot be empty.
      </documentation>
    </annotation>
    <sequence>
      <element name="identifier">
        <complexType>
          <simpleContent>
            <extension base="token">
              <attribute name="scheme" use="required">
                <simpleType>
                  <restriction base="anyURI">
                    <minLength value="1" />
                  </restriction>
                </simpleType>
              </attribute>
            </extension>
          </simpleContent>
        </complexType>
      </element>
      <element ref="xbrli:segment" minOccurs="0" />
    </sequence>
  </complexType>

</schema>
```

4.5.3.1 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 13. Entity identifiers

Example	Meaning
`<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.

4.5.3.2 The segment element (optional)

The `segment` element is an optional container for additional mark-up that the preparer of an XBRL instance SHOULD use to identify the business segment more completely in cases where the entity identifier is insufficient. In general, the content of a `segment` will be specific to the purpose of the XBRL instance document. Elements contained by the `segment` element MUST NOT be defined in the `http://www.xbrl.org/2003/instance` namespace. Also, they MUST NOT be in the substitution group for elements defined in the `http://www.xbrl.org/2003/instance` namespace. The `segment` element MUST NOT be empty.

The XML Schema restrictions on the `segment` element are shown below.

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

  <element name="segment">
    <complexType>
      <sequence>
        <any namespace="##other" processContents="lax" minOccurs="1"
            maxOccurs="unbounded" />
      </sequence>
    </complexType>
  </element>

</schema>
```

Example 14. Using the segment element

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:my="http://www.someCompany.com/segment">
<context id="c1">
  <entity>
    <!—required content -->
    <identifier scheme="www.dnb.com">121064880</identifier>
  <!-- optional content -->
    <segment>
      <my:stateProvince>MI</my:stateProvince>
    </segment>
  </entity>
  <period><instant>2002-12-01</instant></period>
</context>
</xbrl>
```

```
<!-- Company specific segment sub-element -->
<schema targetNamespace="http://www.someCompany.com/segment"
        xmlns:my="http://www.someCompany.com/segment"
        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.

4.5.4 The scenario element (optional)

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.

Elements contained by the `scenario` element MUST NOT be defined in the `http://www.xbrl.org/2003/instance` namespace. Also, they MUST NOT be in the substitution group for elements defined in the `http://www.xbrl.org/2003/instance` namespace. The `scenario` element MUST NOT be empty.

The XML Schema restrictions on the `scenario` element are shown below.

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

  <complexType name="contextScenarioType">
    <annotation>
      <documentation>
      Used for the scenario under which fact have been reported.
      </documentation>
    </annotation>
    <sequence>
      <any namespace="##other" processContents="lax"
        minOccurs="1" maxOccurs="unbounded" />
    </sequence>
  </complexType>
</schema>
```

Example 15. 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">
<context id="c1">
   <entity>
     <identifier scheme="http://www.example.com"></identifier>
   </entity>
   <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>
</context>
</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.

4.6   The unit element

The `unit` element specifies the units in which a numeric item has been measured. The content of the unit element MUST be either a simple unit of measure expressed with a single `measure` element or a complex unit of measure based on a mathematical combination of `measure` elements where the mathematical operations are expressed with the `operator` element.

Some examples of simple units of measure are EUR (Euros), meters, kilograms and FTE (Full Time Equivalents). Some examples of complex units of measures are Earnings per Share and Square Feet.

The XML Schema restrictions on the `unit` element are shown below.

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

  <annotation>
    <documentation>
    XML Schema components contributing to the unit element
    </documentation>
  </annotation>
  <element name="measure" type="QName" />
  <complexType name="measuresType">
    <annotation>
      <documentation>
      A collection of sibling measure elements
      </documentation>
    </annotation>
    <sequence>
      <element ref="xbrli:measure" minOccurs="1" maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <element name="divide">
    <annotation>
      <documentation>
      Element used to represent division in units
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="unitNumerator" type="xbrli:measuresType" />
        <element name="unitDenominator" type="xbrli:measuresType" />
      </sequence>
    </complexType>
  </element>
  <element name="unit">
    <annotation>
      <documentation>
      Element used to represent units information about numeric items
      </documentation>
    </annotation>
    <complexType>
      <choice>
        <element ref="xbrli:measure" minOccurs="1" maxOccurs="unbounded" />
        <element ref="xbrli:divide" />
      </choice>
      <attribute name="id" type="ID" use="required" />
    </complexType>
  </element>

</schema>
```

4.6.1 The id attribute

Every `unit` element MUST include an `id` attribute. The value of the `id` attribute MUST conform to the [XML] rules for attributes with the ID type (http://www.w3.org/TR/REC-xml#NT-TokenizedType). The `id` attribute identifies the unit (see Section 4.6) so that it may be referenced by `item` elements.

4.6.2 The measure element

The `measure` element is used to describe a basic unit of measure. The content of the `measure` element MUST be a `QName` comprised of a namespace prefix and a local part separated by a colon (":"). The namespace prefix MUST resolve to a namespace that is in scope for the `measure` element. The local part MUST be an `NCName`.

Some facts have restrictions on the content of the `unit` element and the value of the `measure` element that is a consequence of the type of concept they represent. These restrictions are set out in the following table.

Table 3. Unit restrictions based on item types.

Item type	implies `unit` **MUST contain**
`monetaryItemType` or derived from `monetaryItemType`	A single `measure` element where The local part of the `measure` MUST be an ISO 4217 currency designation [ISO] that was valid during the time designated by the period element of the item's context. The namespace prefix MUST resolve to: `http://www.xbrl.org/2003/iso4217`
`sharesItemType` or derived from `sharesItemType`	A single `measure` element. The local part of the `measure` MUST be `"shares"` and the namespace prefix that MUST resolve to `http://www.xbrl.org/2003/instance`

For facts that are rates, percentages or ratios where the numerator and the denominator are in the same units, the context MUST have a `unit` element with a single `measure` element. The local part of the `measure` MUST be `"pure"` with a namespace prefix that resolves to a namespace of `"http://ww.xbrl.org/2003/instance"`. Rates, percentages or ratios must be reported using decimal or scientific notation rather in percentages where the value has been multiplied by 100.

A complex unit of measure can expressed by showing the mathematical relationships between other units of measure using a sequence of sibling `measure` elements (which imply a multiplication of those measure elements) and a single `divide` element (which implies division of a numerator by a denominator).

4.6.3 The divide element

The `divide` element MUST contain a `numerator` element followed by a `denominator` element.

4.6.4 The unitNumerator and unitDenominator elements

The `unitNumerator` element and the `unitDenominator` element must both contain one or more `measure` elements.

Units MUST be expressed in their simplest possible form. The `divide` element MUST not contain any `measure` elements in its `unitNumerator` that are s-equal to `measure` elements in its `unitDenominator`.

Some examples of the `unit` element are shown in the following example.

Example 16. Use of the unit element

Example	Meaning
`<unit id="u1"><measure xmlns:ISO4217="http://www.xbrl.org/2003/iso4217" >ISO4217:GBP</measure></unit>`	Currency, UK Pounds.
`<unit id="u2"><measure xmlns:ISO4217="http://www.xbrl.org/2003/iso4217" >ISO4217:gbp</measure></unit>`	Incorrect lower case currency designator.

`<unit id="u1"><measure>xbrli:pure</measure></unit>`	A pure number, such as % revenue change.

`<unit id="u3">`
`    <measure>myuom:feet</measure>`
`    <measure>myuom:feet</measure>`
`</unit>`
	Square feet – feet multiplied by feet.

`<unit id="u4"><measure>xbrli:shares</measure></unit>`	A number of shares.
`<unit id="u5"><measure>myuom:FTE</measure></unit>`	A head count (number of Full Time Equivalents).

`<unit id="u6">`
`    <divide>`
`        <unitNumerator>`
`          <measure>ISO4217:EUR</measure>`
`        </unitNumerator>`
`        <unitDenominator>`
`          <measure>xbrli:shares</measure>`
`        </unitDenominator>`
`    </divide>`
`</unit>`
	Earnings per share (EPS) measured in Euros per share.

`<unit id="u6">`
`    <divide>`
`        <unitNumerator>`
`          <measure>ISO4217:EUR</measure>`
`        </unitNumerator>`
`        <unidDenominator>`
`          <measure>ISO4217:EUR</measure>`
`        </unitDenominator>`
`    </divide>`
`</unit>`
	Illegal because the same measure occurs in both the `numerator` and the `denominator` of the `divide` element.

The `"ISO4217"` namespace prefix used in these examples must resolve to `"http://www.xbrl.org/2003/iso4217"`. The `"xbrli"` namespace prefix used in these examples must resolve to `"http://www.xbrl.org/2003/instance"`. The `"myuom"` namespace prefix is not defined by the XBRL specification, but it must resolve to a namespace that is in scope for the `measure` element. This namespace may be a URL that identifies a resource that describes the units of measure that are contained by the namespace. Although there are no XBRL semantics on how to interpret this information, it may provide assistance to creators of instance documents. For example, if the `myuom` namespace prefix resolves to "http://www.mycomp.com/myuom" then this namespace could be a URL that contains an HTML document that lists the available units of measure.	

Some complex units of measure MAY be expressed as a simple unit of measure. For example, square feet may be expressed as a complex unit of measure showing a multiplication of two basic measures of feet as shown in the following example. It is at the discretion of the instance document creator to use a `unit` element that describes the unit of measure to the appropriate degree.

Example 17. Simple and complex unit of measure comparison

Simple Unit of Measure	Complex Unit of Measure

`<unit id="u1">`
`    <measure>myuom:sqrft</measure>`
`</unit>`
	`<unit id="u4">`
	`    <measure>myuom:feet</measure>`
	`    <measure>myuom:feet</measure>`
	`</unit>`

Note: The namespace prefix `myuom` must resolve to a valid namespace. It should be understood that the measures in this example `"sqrft"`, and `"feet"` are contained in this namespace.	

A `measure` element with a namespace prefix that resolves to the `"http://www.xbrl.org/2003/instance"` namespace MUST have a local part of either `"pure"` or `"share"`. A `measure` element with a namespace prefix that resolves to the `"http://www.xbrl.org/2003/iso4217"` namespace MUST have a local part that is a valid

ISO4217 currency code [ISO] that was valid during the time designated by the `period` element of the item's `context`.

4.7 Tuples

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. The following rules apply to tuples:

- All tuples MUST be members of the substitution group that has `tuple` as its head. Therefore, all item and tuples must be declared globally, because only global elements can be in a substitution group.

- All tuple syntax definitions MUST NOT include a `periodType` or `balance` attribute (See Sections 5.1.1.1 and 5.1.1.2 respectively);

- To be in the substitution group of tuple, a tuple must have a type that is derived from `tupleType`. The type `tupleType` is final with respect to extension, so all tuples MUST be derived only by restriction from `tupleType`.

- Tuples cannot have mixed content, or simple content, and any additional attributes must be drawn from a non-XBRL namespace.

- Tuples MUST NOT be declared `abstract`.

- Descendant elements of a tuple MUST NOT contain any anonymous type declarations.

- The `restriction` element of the tuple declaration MUST NOT contain elements other than `sequence`, `choice`, `all`, `any`, `anyAttribute` and `element`.

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

  <complexType name="tupleType" final="extension">
    <annotation>
      <documentation>
      Tuple element type used for head of tuple substitution group
      </documentation>
    </annotation>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="xbrli:item" maxOccurs="unbounded" />
      <element ref="xbrli:tuple" maxOccurs="unbounded" />
    </choice>
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="##other" processContents="lax" />
  </complexType>
  <element name="tuple" type="xbrli:tupleType" abstract="true">
    <annotation>
      <documentation>
      Abstract tuple element used as head of tuple substitution group
      </documentation>
    </annotation>
  </element>

</schema>
```

Example 18. Defining a tuple as a restriction of tupleType

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

  <element name="managementName" type="xbrli:tokenItemType"
xbrli:periodType="instant"
          substitutionGroup="xbrli:item"/>
  <element name="managementTitle" type="xbrli:tokenItemType"
xbrli:periodType="instant"
          substitutionGroup="xbrli:item"/>
  <element name="managementAge" type="xbrli:nonNegativeIntegerItemType"
          xbrli:periodType="instant" 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 (context **and** unit **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 contextRef="c1">Haywood Chenokitov</s:managementName>
    <s:managementTitle contextRef="c1">President</s:managementTitle>
    <s:managementAge contextRef="n1" unitRef="u1">42</s:managementAge>
  </s:managementInformation>
  <s:managementInformation>
    <s:managementName contextRef="c1">Miriam Minderbender</s:managementName>
    <s:managementTitle contextRef="c1">CEO</s:managementTitle>
  </s:managementInformation>

</xbrl>
```

The all, sequence and choice elements MAY appear in tuples. For example, consider information that is disclosed in tax filings regarding real estate and other properties:

Example 19. 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 either has a fair market value or has already been disposed of, but not both.

Example 20. Hierarchy in a tuple

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

Example: tuples associate concepts that cannot be understood independently and repeat within an XBRL instance. Multiple occurrences of a tuple within an XBRL instance are distinguished by having different content and contexts.

The content models for tuples can be defined using only XML Schema. Content models for tuples are not defined or modified by any of the XBRL linkbases.

4.8 Equality predicates relevant to detecting duplicate items and tuples

There are several different senses of "equal" that are relevant to detection of duplicates in XBRL instances: Identical, Structure equal (s-equal), Parent equal (p-equal), Value equal (v-equal), [XPATH] equal (x-equal), Context equal (c-equal) and Unit equal (u-equal). These different equality predicates are polymorphic and formally defined in a 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 one 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 one set can be paired with a different node in the other set to which it is {s-equal, v-equal, c-equal} and no nodes in either set are left un-paired.
any XML object	x-equal	An XML object **A** is x-equal to an XML object **B** if the [XPATH] expression **A** = **B** returns the value true (see www.w3.org/TR/xpath.html#booleans)
text	s-equal	The two text strings are x-equal
attribute	s-equal	The two attributes have local names and namespaces that are s-equal and have values that are x-equal
Element (except those handled separately in this list)	s-equal	Not identical, and their element local names and namespaces are both 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).
startDate	s-equal	The implied date/time is equal, according to the rules set out in Section 4.5.2
endDate	s-equal	The implied date/time is equal, according to the rules set out in Section 4.5.2
instant	s-equal	The implied date/time is equal, according to the rules set out in Section 4.5.2

Argument Types	Predicates	Definition
period	s-equal	One of the following conditions applies: 1. both elements have a child `forever` element, or 2. their child `instant` elements are s-equal, or 3. their child `startDate` elements are s-equal and their child `endDate` elements are s-equal
measure	s-equal	The namespace prefix in the content of the two measure elements resolves to the same namespace and the local names in the content of the two measure elements are s-equal.
operator	s-equal	The first child elements of the operator element are s-equal and the second child elements of the operator element are s-equal.
unit	s-equal	The descendant `operator` and `measure` elements are s-equal.
context	s-equal	`period` elements are s-equal, and `entity` elements are x-equal, and `scenario` elements are s-equal.
item	s-equal	they are c-equal, and they are u-equal, and `precision` attributes are s-equal, and `decimals` attributes are s-equal, and the text of their contents is s-equal after converting any values of numeric items to a decimal representation.
tuple	s-equal	The sets of (`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 their `contextRef` attributes identify `contexts` that are identical or s-equal
item	u-equal	not s-equal, and their `unitRef` attributes identify units of measurement that are identical or s-equal.
numeric items not of type `fractionItemType` or a type derived from `fractionItemType` by restriction	v-equal	**A** and **B** are v-equal if and only if all the following conditions apply: i. **A** and **B** are c-equal ii. the numeric values **AN** and **BN** are x-equal where **AN** is obtained by rounding the content of **A** to **N** significant figures and **BN** is obtained by rounding the content of **B** to **N** significant figures where **N** is the lowest of: a. the specified or inferred precision for **A** and b. the specified or inferred precision for **B**

Argument Types	Predicates	Definition

numeric items of type `fractionItemType` or a type derived from `fractionItemType` by restriction	v-equal	**A** and **B** are v-equal if and only if all the following conditions apply:
		i. **A** and **B** are c-equal
		ii. **AN** is x-equal to **BN** and **AD** is x-equal to **BD** **where:**
		    a. **AN** is the numerator and **AD** is the denominator of the normal form (defined below) of **A** and
		    b. **BN** is the numerator and **BD** is the denominator of the normal form of **B.**
		
		For any item **F** of type `fractionItemType` or a type derived from `fractionItemType` by restriction, the normal form has numerator F_N and denominator F_D such that F_N and F_D are integers and have no integer common factor and there exists a number **H** such that multiplying F_N by **H** gives the numerator of **F** and multiplying F_D by **H** gives the denominator of **F.**

numeric items, `one of which is of type fractionItemType` or a type derived from `fractionItemType` by restriction and the other of which is not	v-equal	v-equal is always false for such combinations of numeric items
`non-numeric item`	v-equal	**A** and **B** are v-equal if and only if all the following conditions apply i. **A** and **B** are c-equal ii. [XPATH] **normalize-space(AC) = normalize-space(BC)** where **AC** is the content of **A** and **BC** is the content of **B.**
`item`	duplicate	The items are not identical and they have element local names that are s-equal and their elements are defined in the same namespace and the items are p-equal, and the items are c-equal.
`tuple`	duplicate	The tuple elements are not identical and they have element local names that are s-equal and their elements are defined in the same namespace and they are 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 21. 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 contextRef="np3">Acme</s:name>`
b gross	`    <s:gross unitRef="u3" contextRef="np3" precision="4">3001</s:gross>`
b returns	`    <s:returns unitRef="u3" contextRef="np3"`
	`              precision="3">100</s:returns>`
	`    <s:net unitRef="u3" contextRef="np3" precision="4">2900</s:net>`
	`  </s:customer>`
c customer	`  <s:customer>`
c name	`    <s:name contextRef="Xnnp3X">Acme</s:name>`
c gross	`    <s:gross unitRef="u3" contextRef="np3" precision="3">3000</s:gross>`
	`    <s:returns unitRef="u3" contextRef="np3"`
	`              precision="3">100</s:returns>`
	`    <s:net unitRef="u3" contextRef="np3" precision="4">2900</s:net>`
	`  </s:customer>`
d customer	`  <s:customer>`
	`    <s:name contextRef="np3">Acme</s:name>`
	`    <s:gross unitRef="u3" contextRef="np3" precision="4">3000</s:gross>`
d returns	`    <s:returns unitRef="u3" contextRef="np3"`
	`precision="3">500</s:returns>`
	`    <s:net unitRef="u3" contextRef="np3" precision="4">2500</s:net>`
	`  </s:customer>`
e customer	`  <s:customer>`
f name	`    <s:name contextRef="np3">Bree</s:name>`
g name	`    <s:name contextRef="Xnnp3X">Bree</s:name>`
	`    <s:gross unitRef="u3" contextRef="np3" precision="4">3000</s:gross>`
	`    <s:returns unitRef="u3" contextRef="np3"`
	`              precision="3">200</s:returns>`
	`    <s:net unitRef="u3" contextRef="np3" precision="4">2800</s:net>`
	`  </s:customer>`
h totalGross	`  <s:totalGross unitRef="u3" contextRef="np3"`
	`              precision="3">12000</s:totalGross>`
	`</s:analysis>`
	`<!-- One Redundant Context Xnnp3X = period,2003 -->`
	`<!-- without the empty scenario element -->`
np3	`<context id="np3">`
	`  <entity>`
	`    <identifier scheme="www.nasdaq.com">SAMP</identifier>`
	`    <segment/>`
	`  </entity>`
	`  <period>`
	`    <startDate>2003-01-01</startDate>`
	`    <endDate>2003-12-31</endDate>`
	`  </period>`
	`  <scenario/>`
	`</context>`
u3	`<unit id="u3"><measure>ISO4217:USD</measure></unit>`
Xnnp3X	`<context id="Xnnp3X">`
	`  <entity>`
	`    <identifier scheme="www.nasdaq.com">SAMP</identifier>`
	`  </entity>`
	`  <period>`
	`    <startDate>2003-01-01</startDate>`

```
          <endDate>2003-12-31</endDate>
        </period>
      </context>
      </xbrl>
```

Note that, notwithstanding the lack of a calculation linkbase in this example, the total of 12000 in "h totalGross" is the most precise value that can be derived from sum of the values of gross for the 4 customers (3001+3000+3000+3000=12001 but the most precise value can be correct to only 3 significant figures because c gross has `precision="3"` and is hence 12000)

Example 22. Predicates for detecting duplicates

Node 1	Node 2	Type	Predicate	True	Reason
np3	Xnnp3X	context	Identical	no	different nodes
np3	Xnnp3X	context	s-equal	yes	`entity` and `period` are s-equal and the empty `scenario` elements are ignored
f name	g name	item	s-equal	no	different context id's `np3` and `Xnnp3X`
f name	g name	item	p-equal	yes	same parent element
f name	g name	item	c-equal	yes	equal contexts `np3` 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 `np3` 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 and unit u3
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 ids np3 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 and unit u3`
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

Node 1	Node 2	Type	Predicate	True	Reason
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 equality predicates in the definition of duplicate items are ones of *equal location*, not *equal content.* In addition, it should be noted that attributes other `contextRef`, `unitRef`, `precision` and `decimals` MUST be ignored for the purposes of this comparison (a consequence of the definition of s-equality for items). For example, additional `id` attributes do not distinguish otherwise equal items. Whether items appear within a tuple or not also impacts on whether they are duplicates, because the definition of duplicate items also carries the proviso that they have the same parent (i.e. are p-equal).

When determining whether two numeric items are v-equal (a predicate that is used in the definition of various other equality type predicates) it is necessary to take into consideration the values of `precision` for the two numeric items. If `precision` has not been specified for either of the two numeric items it is necessary to infer a value for it according to the rules in Section 4.4.4.

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 might 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).

4.9 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 the `footnoteLink` element to describe these irregularly structured associations between facts in an XBRL instance.

4.9.1 The footnoteLink element

The `footnoteLink` element is an extended link. Its generic syntax is documented in Section 3.5.3. It contains locators, resources and arcs that describe irregular relationships between facts in an XBRL instance.

The XML Schema constraints on the `footnoteLink` element are shown below.

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

  <element name="footnoteLink">
    <annotation>
      <documentation>
```

```
      footnote extended link element definition
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:extendedType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xl:title"/>
            <element ref="link:documentation"/>
            <element ref="link:loc"/>
            <element ref="link:footnoteArc"/>
            <element ref="link:footnote"/>
          </choice>
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

Example 23. 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 unitRef="u1"
contextRef="c1">1200</fr:propertyPlantEquipmentGross>
  <fr:assetsTotal id="f1" unitRef="u1" contextRef="c1">2600</fr:assetsTotal>
  <fr:equityTotal id="f3" unitRef="u1" contextRef="c1">1100</fr:equityTotal>
  <fr:liabilitiesTotal id="f2" unitRef="u1"
contextRef="c1">2600</fr:liabilitiesTotal>

  <link:footnoteLink
    xlink:type="extended" xlink:title="1">
    <link:footnote
      xlink:type="resource"
      xlink:label="footnote1"
      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: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:type="arc"
     xlink:from="fact1" xlink:to="footnote1"
      xlink:title="view explanatory footnote"
     xlink:arcrole="http://www.xbrl.org/arc#fact-footnote"/>
  </link:footnoteLink>
  <context id="c1">
    <entity>
      <identifier scheme="http://www.un.org/">Example plc</identifier>
    </entity>
```

```
    <period>
      <instant>2001-08-16</instant>
    </period>
    <scenario name="Actual values">
      <fr:scenarioType>actual</fr:scenarioType>
    </scenario>
  </context>
  <unit id="u1"><measure>ISO4217:EUR</measure></unit>
</xbrl>
```

Meaning: The one `footnoteArc` connects three facts to two footnotes. The two footnotes are in different languages. The `xlink:title` attribute has been used on the `footnoteArc` element to document the nature of the resource being made accessible from the facts.

4.9.1.1 Locators in footnoteLink elements

`footnoteLink` elements MUST NOT contain locators that are not `loc` elements. `loc` elements are documented in detail in Section 3.5.3.6.. The `loc` element, when used in a `footnoteLink`, MUST only point to items or tuples in the same XBRL instance that contains the `loc` element itself.

4.9.1.2 The footnote element

The `footnote` element is the only resource allowed in `footnoteLink` elements. Generic resources are documented in detail in Section 3.5.3.7. The content of `footnote` resources is restricted relative to generic resources. Specifically, `footnote` resources MUST be mixed content containing a simple string, or a fragment of XHTML or a mixture of both.

The XML Schema constraints on the `footnote` element are shown below.

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

  <element name="footnote" substitutionGroup="xl:resource">
    <annotation>
      <documentation>
      Definition of the reference  resource element
      </documentation>
    </annotation>
    <complexType mixed="true">
      <complexContent mixed="true">
        <extension base="xl:resourceType">
          <sequence>
            <any namespace="http://www.w3.org/1999/xhtml" processContents="skip"
              minOccurs="0" maxOccurs="unbounded" />
          </sequence>
          <attribute ref="xml:lang" use="required" />
        </extension>
      </complexContent>
    </complexType>
  </element>

</schema>
```

4.9.1.2.1 The xml:lang attribute on footnote elements

All `footnote` resources MUST have an `xml:lang` attribute identifying the language used for the content of the footnote. The value of the `xml:lang` attribute MUST conform to [XML] rules. (See http://www.w3.org/TR/2000/REC-xml-20001006#sec-lang-tag for details).

4.9.1.3 The footnoteArc element

The `footnoteArc` element has the same syntax as generic extended link arcs. See Section 3.5.3.8 for details.

The XML Schema constraints on the `footnoteArc` element are shown below.

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

  <element name="footnoteArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in footnote extended links.
      </documentation>
    </annotation>
  </element>

</schema>
```

4.9.1.3.1 xlink:arcrole attributes on footnoteArc elements

The value of the `xlink:arcrole` attribute MUST be URI that indicates the meaning of the arc.

One standard arc role value has been defined for arc role values on `footnoteArc` elements. Its value is:

http://www.xbrl.org/arc/fact-footnote

This arc role value is for use on a `footnoteArc` from item or tuple locators to `footnote` resources and it indicates that the `footnote` conveys human-readable information about the fact or facts.

4.9.1.3.2 xlink:title attribute (optional)

The `xlink:title` attribute MAY be used to convey information about the relationship between facts and related footnotes to users navigating between those facts and footnotes. The content of the `xlink:title` attribute MUST be a string. The `xlink:title` attribute content MAY be made visible to users of XLink-enabled applications.

If the `xlink:title` attribute is insufficient for this purpose (for example, if the information needs to be provided in several languages), then titles, as defined in Section 3.5.3.8.7, MAY be used.

5 XBRL Taxonomies

Section 3.1 provides an overview of XBRL taxonomies.

A taxonomy is defined as an XML Schema [SCHEMA-1] and the set of directly referenced extended links (via the `linkbaseRef` element; see Section 5.1.2) and any extended links that are nested within the XML Schema. The XML Schemas in taxonomies are referred to, in this specification, as "taxonomy schemas".

5.1 Taxonomy schemas

A taxonomy MUST include a taxonomy schema. A taxonomy schema MUST be a valid instance of an XML Schema.

If extended links are included in a taxonomy, the taxonomy schema MUST contain `linkbaseRef` elements that point to their linkbases (See Section 5.1.2) or the extended links MUST be nested in linkbases contained in the taxonomy schema itself.

Taxonomy schemas MUST import the XBRL instance schema if they define concepts (elements in the item or tuple substitution groups) because the XBRL instance schema defines the abstract elements `item` and `tuple`. However, taxonomy schemas do not need to import the XBRL instance schema (for example, if their only purpose is to define syntax for segments and scenarios in contexts).

It will be necessary to include namespace declarations for several other schemas in taxonomy schemas, such as the namespace for XML Schema itself.

Example 24. A skeletal taxonomy schema showing linkbase references

```
<schema
   targetNamespace="http://www.mycompany.com/taxonomy/2003-10-19"
   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.mycompany.com/taxonomy/2003-10-19"
   xmlns:xlink="http://www.w3.org/1999/xlink">
   <annotation>
     <appinfo>
       <link:linkbaseRef
         xlink:type="simple"
         xlink:href="linkbase_presentation.xml"
         xlink:role="http://www.xbrl.org/2003/role#presentationLinkbaseRef"
         xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
       <link:linkbaseRef
         xlink:type="simple"
         xlink:href="linkbase_calculation.xml"
         xlink:role="http://www.xbrl.org/2003/role#calculationLinkbaseRef"
         xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
       <link:linkbaseRef
         xlink:type="simple"
         xlink:href="linkbase_definition.xml"
         xlink:role="http://www.xbrl.org/2003/role#definitionLinkbaseRef"
         xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
       <link:linkbaseRef
         xlink:type="simple"
         xlink:href="linkbase_label.xml"
         xlink:role="http://www.xbrl.org/2003/role#labelLinkbaseRef"
         xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
       <link:linkbaseRef
         xlink:type="simple"
         xlink:href="linkbase_reference.xml"
         xlink:role="http://www.xbrl.org/2003/role#referenceLinkbaseRef"
         xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>
     </appinfo>
   </annotation>
   <import
     namespace="http://www.xbrl.org/2003/instance"
     schemaLocation="xbrl-instance.xsd"/>
</schema>
```

XBRL taxonomies MAY be constructed to refer to other taxonomies; this extensibility of taxonomies is a critical feature of XBRL. In order to realise 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 taxonomy schemas MAY import other taxonomy schemas and reference additional XBRL linkbases as appropriate to achieve this extensibility.

Taxonomy schemas MAY also define custom role values and custom arc role values for use in linkbases. See Section 5.1.2 and 5.1.4 for details.

5.1.1 Concept definitions

Concepts are defined in taxonomy schemas. Each concept defined in a taxonomy schema is uniquely identified by an element's syntax definition in the taxonomy schema. To correspond to a concept definition, an XML Schema element definition has to specify the element's name, a substitution group, and type. All element names MUST be unique within a given taxonomy schema. The substitution group MUST imply that the element is in the substitution group for the XBRL `item` or `tuple` element. The element MAY also include any of the other XML Schema attributes that can be used on an element syntax definitions, including `abstract` and `nillable`.

An element defining the syntax for a concept SHOULD also have an `id` attribute. Providing an `id` attribute simplifies the content of the `xlink:href` attribute on linkbase `loc` elements (See Section 3.5.1.2). Note that some XML Schema validators require uniqueness of all `id` attribute values in a taxonomy schema and in all XML schemas that it imports or includes, directly or indirectly. To increase robustness to such interpretations of the XML Schema specification [SCHEMA-2], care SHOULD be taken to limit the extent to which `id` attributes values are likely to clash with `id` attribute values in related schemas. In the example below, this has been done by prefixing the element name with an additional string, "`ci_`" when choosing the value of the `id` attribute for each element.

Example 25. 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:periodType="duration"
    type="xbrli:monetaryItemType"
    substitutionGroup="xbrli:item" nillable="true"/>
  <element
    id="ci_stockBasedCompensationPolicy"
    name="stockBasedCompensationPolicy"
    xbrli:periodType="duration"
    type="xbrli:stringItemType"
    substitutionGroup="xbrli:item" nillable="true"/>
</schema>
```

Meaning: Two concepts have been defined, one associated with the `preferredDividends` element and the other associated with the `stockBasedCompensationPolicy` element. Both concepts can be represented by nil-value items in instance documents. The `preferredDividends` concept is required to appear in XBRL instances as a numeric item with a duration period in its context and the `stockBasedCompensationPolicy` concept is to appear in XBRL instances as a non-numeric item with an instant period in its context.

XBRL also defines two attributes, `periodType` and `balance`, that MAY be used on the element syntax definitions. These attributes facilitate greater control over the way that the concepts. Their definitions are provided in the following sections.

5.1.1.1 The periodType attribute

Some elements are associated with concepts that are measurable at an instant in time while others measure change over a period of time.

The XML Schema constraints on the `periodType` attribute are shown below.

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

  <attribute name="periodType">
    <annotation>
      <documentation>
      The periodType attribute (restricting the period for XBRL items)
      </documentation>
    </annotation>
    <simpleType>
      <restriction base="token">
        <enumeration value="instant" />
        <enumeration value="duration" />
      </restriction>
    </simpleType>
  </attribute>

</schema>
```

The `periodType` attribute MUST be used on elements in the substitution group for the `item` element. A value of `instant` for the periodType attribute indicates that the element, when used in an XBRL instance, MUST always be associated with a context in which the period is an instant. A value of `duration` indicates that the element, when used in an XBRL instance, MUST always be associated with a context in which the period is a duration, expressed using the `startDate` and `endDate` elements or expressed using the `forever` element.

Example 26. Instant and duration concept definitions

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

5.1.1.2 The balance attribute (optional)

An optional `balance` attribute MAY be added to the definition of an element if its type is `monetaryItemType` or derived from `monetaryItemType`. The `balance` attribute MUST NOT be used on items that do not have type equal to the `monetaryItemType` or to a type that is derived from `monetaryItemType`.

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

The XML Schema constraints on the `balance` attribute are shown below.

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

  <attribute name="balance">
    <annotation>
```

```
        <documentation>
        The balance attribute (imposes calculation relationship restrictions)
        </documentation>
      </annotation>
      <simpleType>
        <restriction base="token">
          <enumeration value="debit" />
          <enumeration value="credit" />
        </restriction>
      </simpleType>
    </attribute>

</schema>
```

Example 27. 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:periodType="duration"
    type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/>
  <element
    id="fixedAssets" name="fixedAssets" xbrli:balance="debit"
    xbrli:periodType="instant"
    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 5. 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 28. A concept appearing with positive and negative values in an XBRL instance

```
<xbrl xmlns:my="http://my.namespace.com">
  <my:netIncome contextRef="c1">500</my:netIncome>
  <my:netIncome contextRef="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 6. 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)

5.1.1.3 Item 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 appropriate subset of XML Schema built-in types (both primitive and derived) [SCHEMA-2] as well as 4 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`.

The [SCHEMA-1] mechanism that enables the explicit assertion of the type of an element in an instance document (http://www.w3.org/TR/xmlschema-1/index.html#xsi_type) MUST NOT be applied to any `item` or `tuple` in an XBRL instance. The type of `items` and `tuples` MUST be specified in the appropriate taxonomy schema instead.

Table 7. Defined item types

XBRL Item Type	Base type	unitRef attribute
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
durationItemType	duration	no
dateTimeItemType	dateTime	no
timeItemType	time	no

XBRL Item Type	Base type	unitRef attribute
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
NameItemType	Name	no
NCNameItemType	NCName	no

Some of these types, especially some of those that XML Schema has defined for backward compatibility with Document Type Definitions ("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" xbrli:periodType="instant"
           substitutionGroup="xbrli:item" type="my:stateProvinceItemType"/>
</schema>
```

Meaning: Deriving new item types by restriction from the XBRL provided item types is the only allowed method for XBRL taxonomy schemas. Earlier, in Example 14, 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.

5.1.1.3.1 The monetary, shares and pure data types

The XBRL instance 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 expressed in the same units. The fractionItemType can be used where a numerator and denominator are explicit.

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

```
  <simpleType name="monetary">
    <annotation>
      <documentation>
      the monetary type serves as the datatype for those financial
      concepts in a taxonomy which denote units in a currency.
      Instance items with this type must have a unit of measure
      from the ISO 4217 namespace of currencies.
      </documentation>
    </annotation>
    <restriction base="decimal" />
  </simpleType>
  <simpleType name="shares">
    <annotation>
      <documentation>
      This datatype serves as the datatype for share based
      financial concepts.
      </documentation>
    </annotation>
    <restriction base="decimal" />
  </simpleType>
  <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>
```

5.1.1.3.2 The fractionItemType 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 built-in 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 complex type, `fractionItemType`, is provided. All values of `fractionItemType` are exact. The attributes, `precision` and `decimals`, MUST not occur on items with the `fractionItemType`.

The XML Schema constraints on the `fractionItemType` are shown below.

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

  <complexType name="fractionItemType" final="extension">
    <sequence>
      <element name="numerator" type="decimal" />
      <element name="denominator" type="xbrli:nonZeroDecimal" />
    </sequence>
    <attribute name="contextRef" type="IDREF" use="required" />
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="##other" processContents="lax" />
  </complexType>

</schema>
```

Example 30. Representing fractions

Fractional value	Representation

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

The `numerator` element MUST contain numeric values. The denominator element MUST contain a numeric value that is non-zero and finite.

5.1.2 The linkbaseRef element

The `linkbaseRef` element MAY occur at the XPath [XPATH] path "schema/annotation/appinfo/*" in a taxonomy schema. In a taxonomy schema, the `linkbaseRef` element identifies a linkbase that MUST always participate in a DTS if that taxonomy schema participates in the DTS.

The syntax of the `linkbaseRef` element in taxonomy schemas is identical to the syntax of the `linkbaseRef` element in XBRL instances. For more details, see Section 4.3.

5.1.3 The roleType element

Role types constrain the allowed values for the `xlink:role` attribute on the [XLINK] simple link, extended link and resource elements. Role types MUST be either a standard role type, defined by this specification, or be a custom role type, defined in a taxonomy schema using the `roleType` element. The `roleType` element MUST be located at the [XPATH] path "schema/annotation/appinfo". A custom role type for a given element and with a given name MUST be defined only once within a taxonomy schema with a given target namespace. The role types that are defined by this specification MUST NOT be redefined using the `roleType` element.

The value of the `xlink:role` attribute MUST be an absolute URI. For the `definitionLink`, `calculationLink`, `presentationLink`, `labelLink`, `referenceLink`, `footnoteLink`, `label`, `reference`, and `footnote` elements, the value of the `xlink:role` attribute MUST be composed of a URI with a fragment identifier and MUST be either a standard role type defined by this specification or a custom role type. For custom defined role types, the URI portion of the value of the `xlink:role` attribute value MUST be the taxonomy schema namespace where the value of the `name` attribute of the `roleType` element MUST match the value of the fragment identifier and the value of the `usedOn` attribute of the `roleType` element MUST match the element name of the parent element of the `xlink:role` attribute.

Example 31. 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="link:footnote" name="endnote">
      <link:definition>
      A footnote that should be displayed
      only at the end of a document
      </link:definition>
    </link:roleType>
  </appinfo>
</annotation>
</schema>
```

This `roleType` element defines 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 XML Schema constraints on the `roleType` element and its sub-elements are set out below.

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

  <element name="definition" type="string">
    <annotation>
      <documentation>
      The element to use for human readable definition
      of custom roles and arc roles.
      </documentation>
    </annotation>
  </element>
  <element name="roleType">
    <annotation>
      <documentation>
      The roleType element definition - used to define custom
      role values in XBRL extended links.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="link:definition" minOccurs="0" />
      </sequence>
      <attribute name="name" type="NCName" use="required" />
      <attribute name="usedOn" type="QName" use="required" />
    </complexType>
  </element>

</schema>
```

5.1.3.1 The roleType element

The `roleType` element is the element containing the role type definition. The `roleType` element describes the role type by defining the name of the role type, declaring the element that the role type may be used on, and providing a human-readable definition of the role type.

5.1.3.2 The name attribute on roleType elements

The `name` attribute is a required attribute that contains the value of the fragment ID for the role type being defined. When the custom role type is used, the `xlink:role` attribute contains an absolute URI with a fragment identifier. This fragment identifier MUST match the value of the `name` attribute on the `roleType` element.

5.1.3.3 The usedOn attribute on roleType elements

The `usedOn` attribute is a required attribute that identifies the element that MAY use this role type as the value of its `xlink:role` attribute. The `usedOn` attribute MUST have a QName value. Elements that do not have the local-name and the namespace implied by the QName MUST NOT use this custom role type as their role value.

5.1.3.4 The definition element on roleType elements (optional)

The `definition` element is an optional sub element of the `roleType` element. Its content is simple content that contains a string giving meaning to the role type.

5.1.4 The arcroleType element

Arc role values are used on the [XLINK] arc elements as value of the `xlink:arcrole` attribute. Arc role values MUST be either a standard arc role value defined by this specification or be a custom defined arc role value. Custom defined arc role values MUST be defined using the `arcroleType` element. The `arcroleType` element MUST be located at the [XPATH] path "`schema/annotation/appinfo`". An arc role value for a given arc and with a given name MUST be defined only once within a namespace. The arc role values that are defined by this specification MUST NOT be redefined using the `arcroleType` element.

The value of the `xlink:arcrole` attribute MUST be a URI. For the `definitionArc`, `calculationArc`, `presentationArc`, `labelArc`, `referenceArc` and `footnoteArc` the value of the `xlink:arcrole` attribute MUST be composed of a URI with a fragment identifier and MUST be either a standard arc role value defined by this specification or a taxonomy defined arc role value. For taxonomy defined arc role values, the URI portion of the value of the `xlink:arcrole` attribute MUST be the taxonomy schema namespace where the value of the `name` attribute of the `arcroleType` element MUST match the value of the fragment identifier and the value of the `usedOn` attribute of the `arcroleType` element MUST match the element name of the parent element of the `xlink:arcrole` attribute.

Example 32. Defining a new arc role value

Example: The definition of an arc role value: `"http://www.mycomp.com/role#average-item"` that connects items in the calculation linkbase

<pre><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 name="average-item" usedOn="link:calculationArc"
      cyclesAllowed="none">
    </link:arcroleType>
  </appinfo>
</annotation>
</schema></pre>

This `arcroleType` describes an arc role value for a calculation relationship. It would appear on an arc as `<calculationArc xlink:arcrole="http://www.mycomp.com/role#average-item" .../>`

The XML Schema constraints on the `arcroleType` element and its sub-elements are set out below.

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

  <element name="definition" type="string">
    <annotation>
      <documentation>
      The element to use for human readable definition
      of custom roles and arc roles.
      </documentation>
    </annotation>
  </element>
  <element name="arcroleType">
    <annotation>
      <documentation>
      The  arcroleType  element  definition – used to define custom
      arc role values in XBRL extended links.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="link:definition" minOccurs="0" />
      </sequence>
      <attribute name="name" type="NCName" use="required" />
      <attribute name="usedOn" type="QName" use="required" />
      <attribute name="cyclesAllowed" use="required">
        <simpleType>
          <restriction base="NMTOKEN">
            <enumeration value="any" />
            <enumeration value="undirected" />
            <enumeration value="none" />
          </restriction>
        </simpleType>
      </attribute>
    </complexType>
  </element>

</schema>
```

5.1.4.1 The arcroleType element

The `arcroleType` element is the element containing the definition of the custom arc role type.
The `arcroleType` element describes the arc role value by defining the name of the arc role
value, declaring the element that the arc role type may be used on, declaring the type of
cycles that are allowed for a network of arcs using the arc role type, and providing a human-
readable definition of the meaning of the arc role type.

5.1.4.2 The name attribute on arcroleType elements

The `name` attribute is a required attribute that contains the value of the fragment ID for the arc
role type being defined. When the taxonomy defined arc role type is used, the `xlink:arcrole`
attribute contains a URI with a fragment identifier. This fragment identifier MUST match the
value of the `name` attribute on the `arcroleType` element.

5.1.4.3 The usedOn attribute on arcroleType elements

The `usedOn` attribute is a required attribute that identifies the element that MAY use this arc
role type as the value of its `xlink:arcrole` attribute. The `usedOn` attribute MUST have a QName

value. Elements that do not have the local-name and the namespace implied by the QName MUST NOT use this custom arc role type as their arc role value.

5.1.4.4 The cyclesAllowed attribute

The `cyclesAllowed` attribute is a required attribute that identifies the type of cycles that are allowed in a network of arcs as defined in Section 5.2. Fully conformant XBRL processors must detect and signal networks of arcs with custom arc role types that violate the requirements documented with this attribute.

The valid values of the `cyclesAllowed` attribute are:

Value	Notes
any	A network that allows any cycles does not limit the kinds of relationships that can be expressed.
undirected	A network that allows undirected cycles MAY have any number of directed paths between any two nodes in the network, but MUST NOT allow a node to have a path back to itself. A path is a set of arcs between nodes.
none	A network that allows no cycles MUST have only one direct path between any two nodes.

5.1.4.5 The definition element on arcroleType elements (optional)

The `definition` element is an optional sub element of the `arcroleType` element. Its content is simple content that contains a string giving human-readable meaning to the arc role type.

5.2 Taxonomy linkbases

The extended links in a taxonomy provide additional information about concepts by expressing relationships between concepts (inter-concept relationships) or associating concepts with documentation about their meaning. The extended links in a taxonomy are grouped into linkbases, as defined in Section 3.5.2. Taxonomies currently use five different kind of extended links: definition, calculation, presentation, label and reference. The first three types of extended link express inter-concept relationships, while the last two relate concepts to their documentation.

An example of an inter-concept relationship is a calculation linkbase that expresses a relationship between "cash" and "current assets" where "cash" sums up to "current assets". An example of a relationship between a concept and additional documentation is a label linkbase that expresses a relationship between the concept "cash" and a human readable label in English, such as "Cash" and additional labels for cash in other languages. Also, the label linkbase may contain additional labels for different purposes, such as a label of "Opening Cash Balance", "Closing Cash Balance" and "Total Cash". Although the concept is always referred to as "cash" the labels provided multiple ways of tagging the concept for display purposes.

The linkbases MAY exist in a separate document from the taxonomy schema, although they MAY alternatively be embedded in the taxonomy schema at the XPath [XPATH] path "schema/annotation/appinfo/link:linkbase/*". When a linkbase in a taxonomy is not embedded in the taxonomy schema document, the taxonomy schema MUST contain a `linkbaseRef` to point to the document containing the linkbase.

There are five kinds of extended links 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 XLink document container MUST be a `linkbase` element located either:

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

2. as the root element of a separate document.

In the presentation, calculation, and definition extended links in a DTS, arcs organise XBRL concepts into networks of relationships that associate each concept with other concepts. In label and reference extended links, arcs associate a resource (label or reference) with an XBRL concept.

The presentation, definition, and calculation extended links 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 DTS to format such reports, as they deem appropriate.

A network of arcs is defined, for a DTS, as the set of defined and non-prohibited arcs (See Section 3.5.3.8.5 for a definition of arc prohibition) in the DTS that:

- have same the local name, namespace and `xlink:arcrole` attribute value; and

- are contained in extended link elements with the same local name, namespace, and `xlink:role` attribute value.

If a network of arcs has no arc that identifies a particular concept by the `xlink:to` attribute, then that concept is a root concept in that network. The presentation of root concepts is application dependent.

Taxonomy authors may or may not find it useful to keep the networks that can be expressed using presentation, calculation and definition extended links in some kind of correspondence.

One or more arcs in a network can form a cycle (that is, an element may reach itself by following some number of arcs). Depending on the semantics of the relationships in a network, different types of cycles may be semantically coherent, or they may represent a semantic inconsistency that processing applications MAY choose to detect. Cycles occur when a node in the network of arcs can reach itself by traversing arcs either with regard (directed) or without regard (undirected) to the direction of each individual arc. Fully conformant XBRL processors MUST detect cycles that constitute semantic inconsistencies. Semantically inconsistent cycles are identified for each network that is given semantic meaning in this specification.

Example 33. Types of cycles
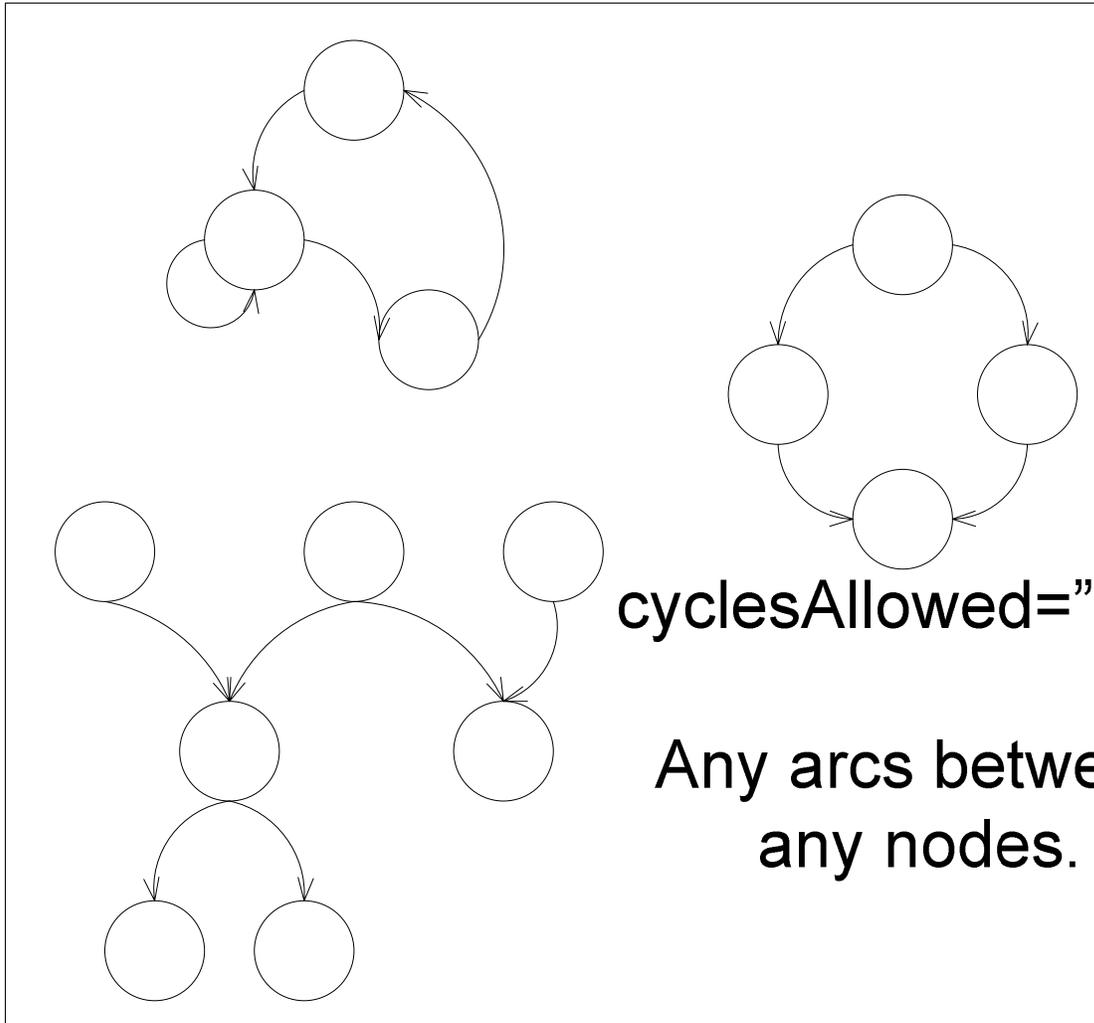
To illustrate networks of relationships between concepts, consider the following concepts that might be defined in a taxonomy (note that the label would not be part of the element; labels are just shown to provide clarity):

Example 34. 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

D

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 extended links to facilitate computation of `netIncome`, `netIncomeBeforeTax`, `netIncomeAfterTax`, per the formulae above and expressed in a tree hierarchy in an application.

Example 35. Hierarchy in a calculation linkbase

<pre>netIncome
    |
    +--(-1)-- extraordinaryItems
    |
    +--(+1)-- netIncomeAfterTax
                 |
                 +--(-1)-- taxes
                 |
                 +--(+1)-- netIncomeBeforeTax</pre>
	Example: Calculation hierarchy in which each item contributes to a summation.
	
	Arcs are annotated with the numeric weight in parentheses. The weight indicates the weight attribute value of the calculation link expressing how the element contributes to the calculation/summation.

The definition linkbase might also contain definition extended links that relate concepts to other concepts. In the case below, `performanceMeasures` is an element defined in the taxonomy and the types of performance measures are: `netIncome`, `netIncomeBeforeTax`, and `netIncomeAfterTax`. The `xlink:arcrole` of the link, an absolute URI such as `http://www.xbrl.org/2003/role#general-special`, explains the type of definition relationship of the relation. See Section 3.5.3.8.4 for details.

Example 36. Hierarchy of general-special arcs in a definition linkbase

<pre>performanceMeasures
    |
    +--1-- netIncome
    |
    +--2-- netIncomeBeforeTax
    |
    +--3-- netIncomeAfterTax</pre>
	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 37. Hierarchy in a presentation linkbase

<table>
<tr>
<td>

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

</td>
<td>

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.

</td>
</tr>
</table>

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.

5.2.1 The linkbase element

The linkbase element is fully documented in Section 3.5.2.

5.2.2 The labelLink element

The labelLink element is an extended link. Its generic syntax is documented in Section 3.5.3. It is intended to contain relationships between concepts and textual documentation and labels for those concepts.

The XML Schema constraints on the labelLink element are shown below.

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

<element name="labelLink">
  <annotation>
    <documentation>
    label extended link element definition
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title" />
          <element ref="link:documentation" />
          <element ref="link:loc" />
          <element ref="link:labelArc" />
          <element ref="link:label" />
        </choice>
      </restriction>
    </complexContent>
  </complexType>
</element>
```

```
</schema>
```

5.2.2.1 Locators in labelLink elements

`labelLink` elements MUST NOT contain locators that are not `loc` elements. `loc` elements are documented in detail in Section 3.5.3.6. The `loc` element, when used in a `labelLink`, MUST only point to concepts in taxonomy schemas.

5.2.2.2 The label element

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 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 in `labelLink` extended links. The `label` element is an XLink resource. Its generic syntax is documented in Section 3.5.3.7. The `label` element MUST have the standard `xml:lang` attribute, and it MUST appear inside a `labelLink` element. This content of the `label` element is mixed, allowing a simple string, a fragment of XHTML or a combination of both.

XBRL processors are NOT REQUIRED to detect or display concept documentation that appears anywhere other than in a `label` element.

The XML Schema constraints on the `label` element are shown below.

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

  <element name="label" substitutionGroup="xl:resource">
    <annotation>
      <documentation>
      Definition of the label  resource element.
      </documentation>
    </annotation>
    <complexType mixed="true">
      <complexContent mixed="true">
        <extension base="xl:resourceType">
          <sequence>
            <any namespace="http://www.w3.org/1999/xhtml" processContents="skip"
              minOccurs="0" maxOccurs="unbounded" />
          </sequence>
          <attribute ref="xml:lang" use="required" />
        </extension>
      </complexContent>
    </complexType>
  </element>

</schema>
```

Example 38. Label resource

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

5.2.2.2.1 The xml:lang attribute on label elements

All `label` resources MUST have an `xml:lang` attribute identifying the language used for the content of the label. The value of the `xml:lang` attribute MUST conform to [XML] rules. (See http://www.w3.org/TR/2000/REC-xml-20001006#sec-lang-tag for details).

5.2.2.2.2 The xlink:role attribute on label elements (optional)

Label resources MAY contain an `xlink:role` attribute, which SHOULD distinguish between `label` resources by the nature of the XBRL concept documentation that they provide. Table 8 specifies all standard `xlink:role` attribute values and their meanings for label resources.

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.
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#negativeTerseLabel 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 `periodType="instant"` 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.

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

Example 39. Arc between a concept and one of its labels

```
<label
  xlink:type="resource"
  xlink:label="A"
  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="B"/>
<labelArc
  xlink:type="arc"
  xlink:from="B"
  xlink:to="A"
  xlink:arcrole="http://www.xbrl.org/2003/role#concept-label"/>
```

Meaning: The `label` resource contains the text of the label and the arc element associates the concept with the label.

5.2.2.3 The labelArc element

The `labelArc` element is an XLink arc. Its generic syntax is defined in Section 3.5.3.8. In `labelLink` elements, it connects concepts with `label` resources.

The XML Schema constraints on the `labelArc` element are shown below.

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

  <element name="labelArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in label extended links.
      </documentation>
    </annotation>
  </element>

</schema>
```

One standard arc role value is defined for `labelArc` elements. Its value is:

> http://www.xbrl.org/2003/role#concept-label

This arc role value is for use on a `labelArc` from a concept locator (`loc` element) to a `label` element and it indicates that the label conveys human-readable information about the concept.

`labelArc` elements cannot describe cyclic relationships between concepts because they only relate concepts to `label` resources, not other concepts. For this reason, no restrictions on cyclic `labelArc` networks need be prescribed.

5.2.3 The referenceLink element

The `referenceLink` element is an extended link. Its generic syntax is documented in Section 3.5.3. It is intended to contain relationships between concepts and references to authoritative statements in the published business, financial and accounting literature that give meaning to the concepts.

The XML Schema constraints on the `referenceLink` element are shown below.

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

  <element name="referenceLink">
    <annotation>
      <documentation>
      reference extended link element definition
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:extendedType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xl:title" />
            <element ref="link:documentation" />
            <element ref="link:loc" />
            <element ref="link:referenceArc" />
            <element ref="link:reference" />
          </choice>
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

Example 40. 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 has given a "role" to each `referenceLink extended` link to partition the extended links in an accounting-related taxonomy based on which part of a financial statement they relate to.

5.2.3.1 Locators in referenceLink elements

`referenceLink` elements MUST NOT contain locators that are not `loc` elements. `loc` elements are documented in detail in Section 3.5.3.6. The `loc` element, when used in a `referenceLink`, MUST only point to concepts in taxonomy schemas.

5.2.3.2 The reference element

The `reference` element enables XBRL taxonomies to ground the definitions of concepts in authoritative statements in the published business, financial and accounting literature. The `reference` element SHOULD only provide information necessary to find the reference materials that are relevant to understanding appropriate usage of the concept being defined. They MUST NOT contain the content of those reference materials themselves. Where textual documentation is required to complete the definition of an XBRL context, this MUST be contained in XBRL `label` elements as documented in Section 5.2.2.2.

The `reference` element is an XLink resource. Its generic syntax is documented in Section 3.5.3.7. The `reference` element MUST appear inside a `referenceLink` element.

The XML Schema constraints on the `reference` element are shown below.

```
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
```

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

  <element name="part" type="string" abstract="true">
    <annotation>
      <documentation>
      Definition of the reference  part element - for use in reference  resources.
      </documentation>
    </annotation>
  </element>
  <element name="reference" substitutionGroup="xl:resource">
    <annotation>
      <documentation>
      Definition of the reference  resource element.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="xl:resourceType">
          <sequence>
            <element ref="link:part" minOccurs="0" maxOccurs="unbounded" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>

</schema>
```

Reference elements are composed of parts. Since the division of references into parts varies in every jurisdiction, part is an abstract element defined in this specification. Taxonomies MAY define elements that substitute for part, allowing them to be included inside reference elements.

Example 41. Arc 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:label="s_customerSales"/>
    <!-- arcs -->
    <referenceArc
      xlink:type="arc" xlink:from="s_customerName" xlink:to="s_customerName_REF"
      xlink:arcrole="http://www.xbrl.org/2003/role#concept-reference"/>
      <!-- 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">
      <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"
      <ref:name>Handbook of Business Reporting</ref:name>
      <ref:pages>45-50</ref:pages>
    </reference>
  </referenceLink>
</linkbase>
```

Meaning: The `reference` elements contain two literature citations, with different `xlink:role` attributes to distinguish them. The arc relates the concept at to both references. 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>
```

Example 42. Reference resource

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

5.2.3.2.1 The xlink:role attribute on reference elements (optional)

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. Table 9 specifies the standard `xlink:role` attribute values and their meanings for reference resources. These parallel the standard `xlink:role` attribute values for `label` resources.

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

reference **resource** xlink:role **attribute value**	Meaning
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.

5.2.3.3 The referenceArc element

The referenceArc element is an XLink arc. Its generic syntax is defined in Section 3.5.3.8. In referenceLink elements, it connects concepts with reference resources.

The XML Schema constraints on the referenceArc element are shown below.

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

  <element name="referenceArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in reference extended links.
      </documentation>
    </annotation>
  </element>

</schema>
```

One standard arc role value is defined for `referenceArc` elements. Its value is:

```
http://www.xbrl.org/2003/role#concept-reference
```

This arc role value is for use on a `referenceArc` from a concept locator (`loc` element) to a `reference` resource and it indicates that the reference is to materials documenting the meaning of the concept.

`referenceArc` elements cannot describe cyclic relationships between concepts because they only relate concepts to `reference` resources, not to other concepts. For this reason, no restrictions on cyclic `referenceArc` networks need be prescribed.

5.2.4 The presentationLink element

The `presentationLink` element is an extended link. Its generic syntax is documented in Section 3.5.3. It is intended to describe presentational relationships between concepts in taxonomies. The `presentationLink` element MUST NOT contain XLink resources.

The XML Schema constraints on the `presentationLink` element are shown below.

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

  <element name="presentationLink" substitutionGroup="xl:extended">
    <annotation>
      <documentation>
      presentation extended link element definition.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:extendedType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xl:title" />
            <element ref="link:documentation" />
            <element ref="link:loc" />
            <element ref="link:presentationArc" />
          </choice>
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

5.2.4.1 Locators in presentationLink elements

`presentationLink` elements MUST NOT contain locators that are not `loc` elements. `loc` elements are documented in detail in Section 3.5.3.6. The `loc` element, when used in a `presentationLink`, MUST only point to concepts in taxonomy schemas.

5.2.4.2 The presentationArc element

The `presentationArc` element is an XLink arc. Its generic syntax is defined in Section 3.5.3.8. The `presentationArc` element defines how concepts relate to one another for presentation.

The XML Schema constraints on the syntax for `presentationArc` elements are shown below.

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

  <element name="presentationArc" substitutionGroup="xl:arc">
    <complexType>
      <annotation>
        <documentation>
        Extension of the extended link arc type for presentation arcs.
        Adds a preferredLabel attribute that documents the role attribute
        value of preferred labels (as they occur in label extended links).
        </documentation>
      </annotation>
      <complexContent>
        <extension base="xl:arcType">
          <attribute name="preferredLabel" use="optional">
            <simpleType>
              <restriction base="anyURI">
                <minLength value="1" />
              </restriction>
            </simpleType>
          </attribute>
        </extension>
      </complexContent>
    </complexType>
  </element>

</schema>
```

Example 43. A presentation arc

```
<presentationArc
  xlink:type="arc"
  xlink:from="ci_currentAssets"
  xlink:to="ci_prepaidExpenses"
  xlink:arcrole="http://www.xbrl.org/2003/role#parent-child"
  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.

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

Example 44. 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.

One standard arc role value is defined for presentationArc elements. Its value is:

    http://www.xbrl.org/2003/role#parent-child

Such relationships are referred to as "parent-child" arcs. Parent-child arcs go from a parent concept to a child concept and indicate that, in a hierarchical view of XBRL information, it is appropriate to show the child concept as a child of the parent concept. Parent-child arcs MUST only relate concepts that are in the item or tuple substitution groups.

Because a network of parent-child arcs describes a hierarchy of concepts, it makes no sense for such a network to document that a concept is its own descendant. For this reason, directed

cycles are not allowed in networks of parent-child arcs. Fully conformant XBRL processors MUST detect and signal undirected cycles in networks of parent-child relationships.

5.2.4.2.1 The preferredLabel attribute

The `preferredLabel` attribute is a URI that MAY be supplied on a parent-child arc to indicate the most appropriate kind of label to use when presenting the arc's child concept. If supplied, the value of the `preferredLabel` attribute MUST be equal to an `xlink:role` attribute value on a `label` resource (in a `labelLink` extended link) that is the target of a concept-label arc from the `presentationArc` element's child concept.

XBRL processors MAY use the value of the `preferredLabel` attribute to choose between different labels that have been associated with the one concept. This can be particularly useful when a given concept is used in a variety of ways within a DTS. For example, cash can be used in the balance sheet and as the starting and ending balances in a cash flow statement. Each appearance of the concept in a set of presentation links MAY use this feature to indicate a different preferred label.

5.2.5 The calculationLink element

The `calculationLink` element is an extended link. Its generic syntax is documented in Section 3.5.3. It describes additive calculation relationships between concepts in taxonomies. The `calculationLink` element MUST NOT contain XLink resources.

The XML Schema constraints on the `calculationLink` element are shown below.

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

  <element name="calculationLink">
    <annotation>
      <documentation>
      calculation  extended link element definition
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:extendedType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xl:title" />
            <element ref="link:documentation" />
            <element ref="link:loc" />
            <element ref="link:calculationArc" />
          </choice>
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

5.2.5.1 Locators in calculationLink elements

`calculationLink` elements MUST NOT contain locators that are not `loc` elements. `loc` elements are documented in detail in Section 3.5.3.6. The `loc` element, when used in a `calculationLink`, MUST only point to concepts in taxonomy schemas.

5.2.5.2 The calculationArc element

The `calculationArc` element is an XLink arc. Its generic syntax is defined in Section 3.5.3.8. The `calculationArc` element defines how concepts relate to one another for calculation purposes.

The XML Schema constraints on the syntax for `calculationArc` elements are shown below.

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

  <element name="calculationArc" substitutionGroup="xl:arc">
    <complexType>
      <annotation>
        <documentation>
        Extension of the extended link arc type for calculation arcs.
        Adds a weight attribute to track weights on contributions to
        summations.
        </documentation>
      </annotation>
      <complexContent>
        <extension base="xl:arcType">
          <attribute name="weight" type="decimal" use="required" />
        </extension>
      </complexContent>
    </complexType>
  </element>

</schema>
```

One standard arc role value is defined for `calculationArc` elements. Its value is:

```
http://www.xbrl.org/2003/role#summation-item
```

Such relationships are referred to as "summation-item" arcs. Summation-item arcs MUST only associate concepts that are in the `item` substitution group. They document aggregation relationships between concepts. They connect from one concept, referred to as the summation concept, to another concept, referred to as the contributing concept.

A complete summation-item arc set for a given summation concept is defined in the context of the DTS supporting an XBRL instance. It is the set of all summation-item arcs, defined in `calculationLink` extended links with the same `xlink:role` attribute value that associate contributing concepts to the given summation concept.

A summation item is an occurrence of a summation concept in an XBRL instance. A contributing item is an occurrence of a contributing concept in an XBRL instance.

A calculation binds for a summation item if there is exactly one contributing item in its scope (see calculation "scoping" below), with a context and unit that is s-equal to the context and unit of the summation item, for each calculationArc in the complete `calculationArc` set defined for the summation item. Items with nil values do not participate in calculation bindings.

A binding calculation is defined to be consistent if the value of its summation item is equal to the sum of the values of the contributing items in the binding, each multiplied by the value of the `weight` attribute on its associated `calculationArc` element.

An XBRL instance is consistent with the semantics of the calculation linkbases in its supporting DTS if all binding calculations for the XBRL instance are consistent.

Fully conformant XBRL processors MUST detect and signal inconsistencies, as defined above, between an XBRL instance and the summation-item arcs of calculation linkbases in its supporting DTS.

No cycles are allowed in networks of summation-item arcs. Fully conformant XBRL processors MUST detect any cycles in networks of summation-item arcs.

Example 45. Syntax of a `calculationArc`

```
<calculationArc
  xlink:type="arc"
  xlink:arcrole="http://www.xbrl.org/2003/role#summation-item"
  xlink:from="currentAssets"
  xlink:to="prepaidExpenses"
  weight="1.0"
  order="1"/>
```

A taxonomy might include a single concept viewed from different perspectives or as having several different dimensions. In the example below, the cash concept can be broken down by branch location, by account type, and by availability.

Example 46. 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

Cash in domestic branches and cash in foreign branches adds to cash. Cash in interest bearing accounts and cash in non-interest bearing accounts also adds to cash. Cash on hand and cash as balances due also adds to cash. To ensure that the calculation relationships between all of these disaggregate cash concepts and the cash concept itself do not cause double or triple counting, the three pairs of summation-item arcs would be grouped into extended links with different extended link role values.

Thus, the summation-item arcs from cash to cash in domestic branches and to cash in foreign branches could be defined in extended links with the extended link role value:

> http://www.mytaxonomy.com/calcLinks/cashByBranchLocation

Likewise, the summation-item arcs from cash to cash in interest bearing accounts and cash in non-interest bearing accounts could be defined in extended links with the extended link role value:

> http://www.mytaxonomy.com/calcLinks/cashByAccountType

Finally, the summation-item arcs from cash to cash on hand and cash as balances due could be defined in extended links with the extended link role value:

> http://www.mytaxonomy.com/calcLinks/cashByAvailability

The different extended link role values avoid double or triple counting in this example by ensuring that the pairs of summation-item arcs are not all processed together.

5.2.5.2.1 The weight attribute

The `weight` attribute MUST appear on `calculationArc` elements. The `weight` attribute MUST have a non-zero decimal value. For summation-item arcs, the `weight` attribute 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.

5.2.5.2.2 Calculation scoping

A summation-item `calculationArc` applies when the taxonomy concepts that are located by the "`from`" and "`to`" attributes of a `summation-item` calculation arc identify c-equal and u-equal items (i.e. they are within equivalent contexts and units in an XBRL instance). However, calculations also take into account tuple structure in the XBRL instance. The "`from`" item element MUST be a child of the least common ancestor of both the "from" and "to" item elements for the calculation relationships to bind. A consequence of this scoping is that items inside duplicate tuples cannot participate in calculations.

Example 47. 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 contextRef="c1">Acme</name>
   <gross contextRef="c1">3000</gross>
   <returns contextRef="c1">100</returns>
   <net contextRef="c1">2900</net>
 </customer>
 <customer>
   <name contextRef="c1">Bree</name>
   <gross contextRef="c1">2000</gross>
   <returns contextRef="c1">200</returns>
   <net contextRef="c1">1800</net>
 </customer>
 <totalGross contextRef="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.

calculation item ("from") path	calculation summation ("to") path	Match ?	Reason
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 XML Schema element definitions to describe 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.

5.2.6 The definitionLink element

The definitionLink element is an extended link. Its generic syntax is documented in Section 3.5.3. It is intended to describe a variety of miscellaneous relationships between concepts in taxonomies. The definitionLink element MUST NOT contain XLink resources.

The XML Schema constraints on the definitionLink element are shown below.

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

  <element name="definitionLink">
    <annotation>
      <documentation>
      definition extended link element definition
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:extendedType">
          <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xl:title" />
            <element ref="link:documentation" />
            <element ref="link:loc" />
            <element ref="link:definitionArc" />
          </choice>
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

5.2.6.1 Locators in calculationLink elements

definitionLink elements MUST NOT contain locators that are not loc elements. loc elements are documented in detail in Section 3.5.3.6. The loc element, when used in a definitionLink, MUST only point to concepts in taxonomy schemas.

5.2.6.2 The definitionArc element

The `definitionArc` element is an XLink arc. Its generic syntax is defined in Section 3.5.3.8. The `definitionArc` elements define various kinds of relationships between concepts.

The XML Schema constraints on the syntax for `definitionArc` elements are shown below.

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

  <element name="definitionArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in definition extended links.
      </documentation>
    </annotation>
  </element>

</schema>
```

Four standard arc role values are defined for `definitionArc` elements. The first standard value is:

> http://www.xbrl.org/2003/role#general-special

Such relationships are referred to as "general-special" relationships. General-special arcs MUST only associate source and target concepts that are in the item substitution group.

General-special arcs connect from a generalisation concept locator to a specialisation concept locator. A generalisation item is an occurrence of a generalisation concept in an XBRL instance. A specialisation item is an occurrence of a specialisation concept in an XBRL instance. A valid value for a specialisation item is a valid value of its generalisation item (if both items are c-equal and u-equal). However, a valid value for a generalisation item is not necessarily a valid value for its specialisation item, even if they are c-equal and s-equal.

Only undirected cycles are allowed in networks of general-special arcs. Fully conformant XBRL processors MUST detect and signal directed cycles in networks of general-special arcs.

Example 48. A general-special arc

```
<definitionArc
  xlink:type="arc"
  xlink:from="postalCode"
  xlink:to="zipCode"
  xlink:arcrole="http://www.xbrl.org/2003/role#general-special"
  order="1"/>
```
Meaning: `postalCode` is a generalisation of `zipCode`. 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 second standard arc role value for `definitionArc` elements is:

> http://www.xbrl.org/2003/role#essence-alias

Such relationships are referred to as "essence-alias" arcs. `definitionArc` elements with this arc role value MUST only associate concepts that are in the item substitution group.

This arc role value is for use on a `definitionArc` from an essence concept locator to an alias concept locator.

An essence item is an occurrence of an essence concept in an XBRL instance. An alias item is an occurrence of an alias concept in an XBRL instance. If the values of associated essence and

alias items, with s-equal contexts and units, are not equal, then those items are not consistent with the semantics of the definition extended links that have associated them.

Only undirected cycles are allowed in networks of essence-alias arcs. Fully conformant XBRL processors MUST detect any directed cycles in networks of essence-alias arcs.

It is often the case that particular concepts have been defined more than once in a single taxonomy or in a set of taxonomies. It is appropriate, in such cases, for taxonomy authors to have a single "canonical best element" or "essence" for one of the concepts and to associate it with the other "alias" concepts using the `essence-alias` definition arc to indicate to XBRL validating processors and other XBRL instance consuming applications that the items MUST be consistent within a given context.

A `definitionArc` element where the `xlink:arcrole` is `http://www.xbrl.org/2003/role#essence-alias` denotes a relationship between two concepts, from the essence (basic, primary) concept, to the other alias (alternative name) concept.

For definitions of "alias concept" "alias item" "essence concept" and "essence item" refer to Table 1. For any set of essence-alias 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 (See Section 3.5.3.8.5 for details on prohibited arcs) 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 other alias concepts.

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

3. If an alias item and an essence item in an XBRL instance that are c-equal are not v-equal in those respective s-equal contexts and units, then the two items are not consistent with the semantics of the definition links in the DTS supporting the XBRL instance. This requirement only applies if both items do not have nil values. Only fully conformant XBRL processors MUST detect such inconsistencies.

4. 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 (non nil valued) members of the alias item set **S** corresponding to all essence-alias arcs with **E** as their essence concept. If all (non nil valued) members of **S** are not v-equal, then the XBRL instance is not consistent with the definition link semantics expressed in its DTS and fully conformant XBRL processors MUST detect and signal such inconsistencies. If an application applies this rule and any member **M** of **S** does not have a value supplied or has a nil value, but is an essence item in some set of essence-alias arcs, this rule MUST be applied recursively to infer the value of **M** before inferring the value of **E**.

XBRL processors are not required to infer the values of alias items from the values of essence items.

Example 49. Inference of values for non-numeric items with concepts connected by essence-alias arcs

In an XBRL instance there is a context `c1`. The concepts D and E are string item types connected by an essence-alias `definitionArc`, with E being the essence concept and D being the alias concept. E has the value "Bert" in context c1 while D has the value "Ernie" in context c1. These values are inconsistent with the `definitionArc` semantics that have been expressed.

5. For any 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 (non nil valued) alias item set **S** corresponding to all essence-alias arcs with **E** as their essence concept, at the greatest values of `precision` and `decimals` for which this is possible (see 4.4.1 above). If all (non nil valued) members of **S**

are not v-equal, then the XBRL instance is not consistent with the definition link semantics expressed in its DTS and fully conformant XBRL processors MUST detect such inconsistencies. If an application applies this rule and any member **M** of **S** does not have a value supplied or has a nil value, but is an essence item in some set of essence-alias arcs, this rule MUST be applied recursively to infer the value of **M** before inferring the value of **E**.

Example 50. Inference of values for numeric items with concepts connected by essence-alias arcs

Case 1

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. A, B and C are c-equal.

The values of B and C are inconsistent at their specified precision of 2. As a result, no inference can be made for A.

Case 2

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=1 and C has the value 99 with precision=1. A, B and C are c-equal.

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 can be inferred for A.

The third standard arc role value for `definitionArc` elements is:

> `http://www.xbrl.org/2003/role#similar-tuples`

Such relationships are referred to as "similar-tuples" arcs. Both the source and target concepts of similar-tuples arcs MUST be in the tuple substitution group. The similar-tuples arcs associate tuple concepts that have equivalent definitions (as provided in the labels and references for those tuples) even when they have different XML content models.

For example, this kind of relationship would be appropriate to use between two different tuple concepts that are both designed to describe mailing addresses.

The semantics of similar-tuples arcs are symmetric. It does not matter which tuple the arc goes from and which tuple the arc goes to.

Any cycles can be semantically sensible in networks of `definitionArc` elements with the `http://www.xbrl.org/2003/role#similar-tuples` arc role value because the relationship between concepts being described by these relationships is symmetric.

The fourth standard arc role value for `definitionArc` elements is:

> `http://www.xbrl.org/2003/role#requires-element`

Such relationships are referred to as "requires-element" arcs. `definitionArc` elements with this arc role value MAY associate concepts that are in the tuple or item substitution groups.

If an instance of the concept at the source of the arc occurs in an XBRL instance then a c-equal instance of the arc's target concept MUST also occur in the XBRL instance. Fully conformant XBRL processors MUST detect and signal instances in which this relationship is violated.

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 a `definitionArc` with the `http://www.xbrl.org/2003/role#requires-element` arc role value. This arc would link the concepts representing the required fields and an element representing the concept of the form itself.

Cycles are allowed in networks of requires-element arcs. The source and target concepts for requires-element arcs MUST be in the tuple or item substitution groups.

6 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 Radner
            RFC2119
            http://www.ietf.org/rfc/rfc2119.txt

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

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

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

[SGML]   International Standards Organisation
            Information Processing – Text and office systems – Standard Generalized Markup Language (SGML)
            http://www.iso.org/iso/en/CatalogueDetailPage.CatalogDetail?CSNUMBER=16387

[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, editors
            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   Jonathan Marsh
Base]
            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/

A. Schemas

The following are the versions of the XML schemas provided as part of this specification. These are all normative except for iso4217.xsd for which no normative version is supplied (See Section 4.6.2). Non-normative versions (which should be identical to these except for comments indicating their non-normative status) are also provided as separate files for convenience of users of the specification.

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.

The 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.

xbrl-instance-2003-07-31.xsd (normative)

```
<?xml version="1.0" ?>
<!-- (c) XBRL International. See www.xbrl.org/legal -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  targetNamespace="http://www.xbrl.org/2003/instance"
  elementFormDefault="qualified">
  <annotation>
    <documentation>
    Taxonomy schema for XBRL. This schema defines syntax relating to
    XBRL instance documents.
    </documentation>
  </annotation>
  <import namespace="http://www.xbrl.org/2003/linkbase"
    schemaLocation="xbrl-linkbase-2003-07-31.xsd" />
  <import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="xml-2001-03.xsd" />
  <annotation>
    <documentation>
    Define the attributes to be used on XBRL concept definitions
    </documentation>
  </annotation>
  <attribute name="periodType">
    <annotation>
      <documentation>
      The periodType attribute (restricting the period for XBRL items)
      </documentation>
    </annotation>
    <simpleType>
     <restriction base="token">
       <enumeration value="instant" />
       <enumeration value="duration" />
     </restriction>
    </simpleType>
  </attribute>
  <attribute name="balance">
    <annotation>
      <documentation>
```

```
        The balance attribute (imposes calculation relationship restrictions)
      </documentation>
    </annotation>
    <simpleType>
      <restriction base="token">
        <enumeration value="debit" />
        <enumeration value="credit" />
      </restriction>
    </simpleType>
  </attribute>
  <annotation>
    <documentation>
    Define the simple types used as a base for for item types
    </documentation>
  </annotation>
  <simpleType name="monetary">
    <annotation>
      <documentation>
      the monetary type serves as the datatype for those financial
      concepts in a taxonomy which denote units in a currency.
      Instance items with this type must have a unit of measure
      from the ISO 4217 namespace of currencies.
      </documentation>
    </annotation>
    <restriction base="decimal" />
  </simpleType>
  <simpleType name="shares">
    <annotation>
      <documentation>
      This datatype serves as the datatype for share based
      financial concepts.
      </documentation>
    </annotation>
    <restriction base="decimal" />
  </simpleType>
  <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>
  <!-- nonZeroDecimal   -->
  <simpleType name="nonZeroDecimal">
    <annotation>
      <documentation>
      As the name implies this is a decimal value that can not take
      the value 0 - it is used as the type for the denominator of a
      fractionItemType.
      </documentation>
    </annotation>
    <union>
      <simpleType>
        <restriction base="decimal">
          <minExclusive value="0" />
        </restriction>
      </simpleType>
      <simpleType>
        <restriction base="decimal">
          <maxExclusive value="0" />
```

```
        </restriction>
      </simpleType>
    </union>
  </simpleType>
  <simpleType name="nonZeroNonInfiniteFloat">
    <annotation>
      <documentation>
      As the name implies this is a decimal value that can not take
      the value 0 - it is used as the type for the denominator of a
      fractionItemType item
      </documentation>
    </annotation>
    <union>
      <simpleType>
        <restriction base="decimal">
          <minExclusive value="0" />
        </restriction>
      </simpleType>
      <simpleType>
        <restriction base="decimal">
          <maxExclusive value="0" />
        </restriction>
      </simpleType>
    </union>
  </simpleType>
  <simpleType name="precisionType">
    <annotation>
      <documentation>
      This type is used to specify the value of 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="string">
          <enumeration value="INF" />
        </restriction>
      </simpleType>
    </union>
  </simpleType>
  <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="string">
          <enumeration value="INF" />
        </restriction>
      </simpleType>
    </union>
  </simpleType>
  <attributeGroup name="numericItemAttrs">
    <annotation>
      <documentation>
      Group of attributes for numeric items
```

```
        </documentation>
      </annotation>
      <attribute name="contextRef" type="IDREF" use="required" />
      <attribute name="unitRef" type="IDREF" use="required" />
      <attribute name="precision" type="xbrli:precisionType" use="optional" />
      <attribute name="decimals" type="xbrli:decimalsType" use="optional" />
      <attribute name="id" type="ID" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
  </attributeGroup>
  <attributeGroup name="nonNumericItemAttrs">
    <annotation>
      <documentation>
    Group of attributes for non-numeric items
    </documentation>
    </annotation>
      <attribute name="contextRef" type="IDREF" use="required" />
      <attribute name="id" type="ID" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
  </attributeGroup>
  <annotation>
    <documentation>
General numeric item types - for use on concept element definitions
The following 3 numeric types are all based on the built-in
data types of XML Schema.
</documentation>
  </annotation>
  <complexType name="decimalItemType" final="extension">
    <simpleContent>
      <extension base="decimal">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="floatItemType" final="extension">
    <simpleContent>
      <extension base="float">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="doubleItemType" final="extension">
    <simpleContent>
      <extension base="double">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <annotation>
    <documentation>
    XBRL domain numeric item types - for use on concept element definitions
    The following 4 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.
    </documentation>
  </annotation>
  <complexType name="monetaryItemType" final="extension">
    <simpleContent>
      <extension base="xbrli:monetary">
        <attributeGroup ref="xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="sharesItemType" final="extension">
```

```
      <simpleContent>
        <extension base="xbrli:shares">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="pureItemType" final="extension">
      <simpleContent>
        <extension base="xbrli:pure">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="fractionItemType" final="extension">
      <sequence>
        <element name="numerator" type="decimal" />
        <element name="denominator" type="xbrli:nonZeroDecimal" />
      </sequence>
      <attribute name="contextRef" type="IDREF" use="required" />
      <attribute name="id" type="ID" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
    </complexType>
    <annotation>
      <documentation>
      The following 13 numeric types are all based on the XML Schema
      built-in types that are derived by restriction from decimal.
      </documentation>
    </annotation>
    <complexType name="integerItemType" final="extension">
      <simpleContent>
        <extension base="integer">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="nonPositiveIntegerItemType" final="extension">
      <simpleContent>
        <extension base="nonPositiveInteger">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="negativeIntegerItemType" final="extension">
      <simpleContent>
        <extension base="negativeInteger">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="longItemType" final="extension">
      <simpleContent>
        <extension base="long">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="intItemType" final="extension">
      <simpleContent>
        <extension base="int">
          <attributeGroup ref="xbrli:numericItemAttrs" />
        </extension>
      </simpleContent>
    </complexType>
```

```
<complexType name="shortItemType" final="extension">
  <simpleContent>
    <extension base="short">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="byteItemType" final="extension">
  <simpleContent>
    <extension base="byte">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="nonNegativeIntegerItemType" final="extension">
  <simpleContent>
    <extension base="nonNegativeInteger">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="unsignedLongItemType" final="extension">
  <simpleContent>
    <extension base="unsignedLong">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="unsignedIntItemType" final="extension">
  <simpleContent>
    <extension base="unsignedInt">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="unsignedShortItemType" final="extension">
  <simpleContent>
    <extension base="unsignedShort">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="unsignedByteItemType" final="extension">
  <simpleContent>
    <extension base="unsignedByte">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<complexType name="positiveIntegerItemType" final="extension">
  <simpleContent>
    <extension base="positiveInteger">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
<annotation>
  <documentation>
  The following 17 non-numeric types are all based on the primitive built-in
  data types of XML Schema.
  </documentation>
</annotation>
<complexType name="stringItemType" final="extension">
```

```
          <simpleContent>
            <extension base="string">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="booleanItemType" final="extension">
          <simpleContent>
            <extension base="boolean">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="hexBinaryItemType" final="extension">
          <simpleContent>
            <extension base="hexBinary">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="base64BinaryItemType" final="extension">
          <simpleContent>
            <extension base="base64Binary">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="anyURIItemType" final="extension">
          <simpleContent>
            <extension base="anyURI">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="uriItemType" final="extension">
          <simpleContent>
            <extension base="anyURI">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="QNameItemType" final="extension">
          <simpleContent>
            <extension base="QName">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="durationItemType" final="extension">
          <simpleContent>
            <extension base="duration">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="dateTimeItemType" final="extension">
          <simpleContent>
            <extension base="xbrli:dateUnion">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
    </complexType>
    <complexType name="timeItemType" final="extension">
```

```
        <simpleContent>
          <extension base="time">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="dateItemType" final="extension">
        <simpleContent>
          <extension base="date">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="gYearMonthItemType" final="extension">
        <simpleContent>
          <extension base="gYearMonth">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="gYearItemType" final="extension">
        <simpleContent>
          <extension base="gYear">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="gMonthDayItemType" final="extension">
        <simpleContent>
          <extension base="gMonthDay">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="gDayItemType" final="extension">
        <simpleContent>
          <extension base="gDay">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="gMonthItemType" final="extension">
        <simpleContent>
          <extension base="gMonth">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <annotation>
        <documentation>
        The following 4 non-numeric types are all based on the XML Schema
        built-in types that are derived by restriction and/or list from string.
        </documentation>
      </annotation>
      <complexType name="normalizedStringItemType" final="extension">
        <simpleContent>
          <extension base="normalizedString">
            <attributeGroup ref="xbrli:nonNumericItemAttrs" />
          </extension>
        </simpleContent>
      </complexType>
      <complexType name="tokenItemType" final="extension">
        <simpleContent>
```

```
            <extension base="token">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
        </complexType>
        <complexType name="languageItemType" final="extension">
          <simpleContent>
            <extension base="language">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
        </complexType>
        <complexType name="NameItemType" final="extension">
          <simpleContent>
            <extension base="Name">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
        </complexType>
        <complexType name="NCNameItemType" final="extension">
          <simpleContent>
            <extension base="NCName">
              <attributeGroup ref="xbrli:nonNumericItemAttrs" />
            </extension>
          </simpleContent>
        </complexType>
        <annotation>
          <documentation>
          XML Schema components contributing to the context element
          </documentation>
        </annotation>
        <element name="segment">
          <complexType>
            <sequence>
              <any namespace="##other" processContents="lax" minOccurs="1"
maxOccurs="unbounded" />
            </sequence>
          </complexType>
        </element>

        <complexType name="contextEntityType">
          <annotation>
            <documentation>
            The type for the entity element, used to describe the reporting entity.
            Note that the scheme attribute is required and cannot be empty.
            </documentation>
          </annotation>
          <sequence>
            <element name="identifier">
              <complexType>
                <simpleContent>
                  <extension base="token">
                    <attribute name="scheme" use="required">
                      <simpleType>
                        <restriction base="anyURI">
                          <minLength value="1" />
                        </restriction>
                      </simpleType>
                    </attribute>
                  </extension>
                </simpleContent>
              </complexType>
            </element>
```

```
          <element ref="xbrli:segment" minOccurs="0" />
      </sequence>
  </complexType>
  <simpleType name="dateUnion">
    <annotation>
      <documentation>
      The union of the date and dateTime simple types.
      </documentation>
    </annotation>
    <union memberTypes="date dateTime " />
  </simpleType>
  <complexType name="contextPeriodType">
    <annotation>
      <documentation>
      The type for the period element, used to describe the reporting date info.
      </documentation>
    </annotation>
    <choice>
      <sequence>
        <element name="startDate" type="xbrli:dateUnion" />
        <element name="endDate" type="xbrli:dateUnion" />
      </sequence>
      <element name="instant" type="xbrli:dateUnion" />
      <element name="forever">
        <complexType />
      </element>
    </choice>
  </complexType>
  <complexType name="contextScenarioType">
    <annotation>
      <documentation>
      Used for the scenario under which fact have been reported.
      </documentation>
    </annotation>
    <sequence>
      <any namespace="##other" processContents="lax" minOccurs="1" maxOccurs="unbounded"
/>
    </sequence>
  </complexType>
  <element name="context">
    <annotation>
      <documentation>
      Used for an island of context to which facts can be related.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="entity" type="xbrli:contextEntityType" />
        <element name="period" type="xbrli:contextPeriodType" />
        <element name="scenario" type="xbrli:contextScenarioType" minOccurs="0" />
      </sequence>
      <attribute name="id" type="ID" use="required" />
    </complexType>
  </element>
  <annotation>
    <documentation>
    XML Schema components contributing to the unit element
    </documentation>
  </annotation>
  <element name="measure" type="QName" />
  <complexType name="measuresType">
    <annotation>
      <documentation>
```

```
      A collection of sibling measure elements
      </documentation>
    </annotation>
    <sequence>
      <element ref="xbrli:measure" minOccurs="1" maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <element name="divide">
    <annotation>
      <documentation>
      Element used to represent division in units
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="unitNumerator" type="xbrli:measuresType" />
        <element name="unitDenominator" type="xbrli:measuresType" />
      </sequence>
    </complexType>
  </element>
  <element name="unit">
    <annotation>
      <documentation>
      Element used to represent units information about numeric items
      </documentation>
    </annotation>
    <complexType>
      <choice>
        <element ref="xbrli:measure" minOccurs="1" maxOccurs="unbounded" />
        <element ref="xbrli:divide" />
      </choice>
      <attribute name="id" type="ID" use="required" />
    </complexType>
  </element>
  <annotation>
    <documentation>
    Elements to use for facts in instances
    </documentation>
  </annotation>
  <element name="item" type="anyType" abstract="true">
    <annotation>
      <documentation>
      Abstract item element used as head of item substitution group
      </documentation>
    </annotation>
  </element>
  <complexType name="tupleType" final="extension">
    <annotation>
      <documentation>
      Tuple element type used for head of tuple substitution group
      </documentation>
    </annotation>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="xbrli:item" maxOccurs="unbounded" />
      <element ref="xbrli:tuple" maxOccurs="unbounded" />
    </choice>
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="##other" processContents="lax" />
  </complexType>
  <element name="tuple" type="xbrli:tupleType" abstract="true">
    <annotation>
      <documentation>
      Abstract tuple element used as head of tuple substitution group
```

```
      </documentation>
    </annotation>
  </element>
  <element name="xbrl">
    <annotation>
      <documentation>
      XBRL instance root element.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="link:schemaRef" minOccurs="1" maxOccurs="unbounded" />
        <element ref="link:linkbaseRef" minOccurs="0" maxOccurs="unbounded" />
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xbrli:item"/>
          <element ref="xbrli:tuple"/>
          <element ref="xbrli:context"/>
          <element ref="xbrli:unit"/>
          <element ref="link:footnoteLink"/>
        </choice>
      </sequence>
      <attribute name="id" type="ID" use="optional" />
      <attribute ref="xml:base" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
    </complexType>
  </element>
</schema>
```

xbrl-linkbase-2003-07-31.xsd (normative)

```
<?xml version="1.0" ?>
<!-- (c) XBRL International. See www.xbrl.org/legal -->
<schema targetNamespace="http://www.xbrl.org/2003/linkbase"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xl="http://www.xbrl.org/2001/XLink"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
    XBRL simple and extended link schema constructs
    </documentation>
  </annotation>

  <import namespace="http://www.xbrl.org/2001/XLink"
    schemaLocation="xl.xsd" />

  <import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="xlink.xsd"/>

  <import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="xml-2001-03.xsd" />

  <element name="documentation" substitutionGroup="xl:documentation">
    <annotation>
      <documentation>
      Concrete element to use for documentation of
      extended links and linkbases.
      </documentation>
    </annotation>
    <complexType>
```

```
      <complexContent>
        <restriction base="xl:documentationType">
          <attribute ref="xml:lang" use="required" />
          <anyAttribute namespace="##other" processContents="lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

  <element name="loc" type="xl:locatorType" substitutionGroup="xl:locator">
    <annotation>
      <documentation>
      Concrete locator element. The loc element is the
      XLink locator element for all extended links in XBRL.
      </documentation>
    </annotation>
  </element>

  <element name="labelArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in label extended links.
      </documentation>
    </annotation>
  </element>
  <element name="referenceArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in reference extended links.
      </documentation>
    </annotation>
  </element>
  <element name="definitionArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in definition extended links.
      </documentation>
    </annotation>
  </element>
  <element name="presentationArc" substitutionGroup="xl:arc">
    <complexType>
      <annotation>
        <documentation>
        Extension of the extended link arc type for presentation arcs.
        Adds a preferredLabel attribute that documents the role attribute
        value of preferred labels (as they occur in label extended links).
        </documentation>
      </annotation>
      <complexContent>
        <extension base="xl:arcType">
          <attribute name="preferredLabel" use="optional">
            <simpleType>
              <restriction base="anyURI">
                <minLength value="1" />
              </restriction>
            </simpleType>
          </attribute>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="calculationArc" substitutionGroup="xl:arc">
    <complexType>
```

```xml
      <annotation>
        <documentation>
        Extension of the extended link arc type for calculation arcs.
        Adds a weight attribute to track weights on contributions to
        summations.
        </documentation>
      </annotation>
      <complexContent>
        <extension base="xl:arcType">
          <attribute name="weight" type="decimal" use="required" />
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="footnoteArc" type="xl:arcType" substitutionGroup="xl:arc">
    <annotation>
      <documentation>
      Concrete arc for use in footnote extended links.
      </documentation>
    </annotation>
  </element>

  <element name="label" substitutionGroup="xl:resource">
    <annotation>
      <documentation>
      Definition of the label  resource element.
      </documentation>
    </annotation>
    <complexType mixed="true">
      <complexContent mixed="true">
        <extension base="xl:resourceType">
          <sequence>
            <any namespace="http://www.w3.org/1999/xhtml" processContents="skip"
              minOccurs="0" maxOccurs="unbounded" />
          </sequence>
          <attribute ref="xml:lang" use="required" />
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="part" type="string" abstract="true">
    <annotation>
      <documentation>
      Definition of the reference  part element - for use in reference  resources.
      </documentation>
    </annotation>
  </element>
  <element name="reference" substitutionGroup="xl:resource">
    <annotation>
      <documentation>
      Definition of the reference  resource element.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="xl:resourceType">
          <sequence>
            <element ref="link:part" minOccurs="0" maxOccurs="unbounded" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
```

```
<element name="footnote" substitutionGroup="xl:resource">
  <annotation>
    <documentation>
    Definition of the reference  resource element
    </documentation>
  </annotation>
  <complexType mixed="true">
    <complexContent mixed="true">
      <extension base="xl:resourceType">
        <sequence>
          <any namespace="http://www.w3.org/1999/xhtml" processContents="skip"
            minOccurs="0" maxOccurs="unbounded" />
        </sequence>
        <attribute ref="xml:lang" use="required" />
      </extension>
    </complexContent>
  </complexType>
</element>

<element name="presentationLink" substitutionGroup="xl:extended">
  <annotation>
    <documentation>
    presentation extended link element definition.
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title" />
          <element ref="link:documentation" />
          <element ref="link:loc" />
          <element ref="link:presentationArc" />
        </choice>
      </restriction>
    </complexContent>
  </complexType>
</element>
<element name="definitionLink">
  <annotation>
    <documentation>
    definition extended link element definition
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title" />
          <element ref="link:documentation" />
          <element ref="link:loc" />
          <element ref="link:definitionArc" />
        </choice>
      </restriction>
    </complexContent>
  </complexType>
</element>
<element name="calculationLink">
  <annotation>
    <documentation>
    calculation  extended link element definition
    </documentation>
  </annotation>
```

```
        <complexType>
          <complexContent>
            <restriction base="xl:extendedType">
              <choice minOccurs="0" maxOccurs="unbounded">
                <element ref="xl:title" />
                <element ref="link:documentation" />
                <element ref="link:loc" />
                <element ref="link:calculationArc" />
              </choice>
            </restriction>
          </complexContent>
        </complexType>
    </element>
    <element name="labelLink">
      <annotation>
        <documentation>
        label extended link element definition
        </documentation>
      </annotation>
      <complexType>
        <complexContent>
          <restriction base="xl:extendedType">
            <choice minOccurs="0" maxOccurs="unbounded">
              <element ref="xl:title" />
              <element ref="link:documentation" />
              <element ref="link:loc" />
              <element ref="link:labelArc" />
              <element ref="link:label" />
            </choice>
          </restriction>
        </complexContent>
      </complexType>
    </element>
    <element name="referenceLink">
      <annotation>
        <documentation>
        reference extended link element definition
        </documentation>
      </annotation>
      <complexType>
        <complexContent>
          <restriction base="xl:extendedType">
            <choice minOccurs="0" maxOccurs="unbounded">
              <element ref="xl:title" />
              <element ref="link:documentation" />
              <element ref="link:loc" />
              <element ref="link:referenceArc" />
              <element ref="link:reference" />
            </choice>
          </restriction>
        </complexContent>
      </complexType>
    </element>
    <element name="footnoteLink">
      <annotation>
        <documentation>
        footnote extended link element definition
        </documentation>
      </annotation>
      <complexType>
        <complexContent>
          <restriction base="xl:extendedType">
            <choice minOccurs="0" maxOccurs="unbounded">
```

```
            <element ref="xl:title" />
            <element ref="link:documentation" />
            <element ref="link:loc" />
            <element ref="link:footnoteArc" />
            <element ref="link:footnote" />
          </choice>
        </restriction>
      </complexContent>
    </complexType>
  </element>

  <element name="linkbase">
    <annotation>
      <documentation>
      Definition of the linkbase element. Used to
      contain a set of zero or more extended link elements.
      </documentation>
    </annotation>
    <complexType>
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="link:documentation"/>
        <element ref="xl:extended"/>
      </choice>
      <attribute name="id" type="ID" use="optional" />
      <attribute ref="xml:base" />
    </complexType>
  </element>

  <element name="linkbaseRef" substitutionGroup="xl:simple">
    <annotation>
      <documentation>
      Definition of the linkbaseRef element - used
      to link to XBRL taxonomy extended links from
      taxonomy schema documents and from XBRL
      instance documents.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base="xl:simpleType">
          <attribute ref="xlink:arcrole" use="required">
            <annotation>
              <documentation>
              This attribute must have the value:
              http://www.w3.org/1999/xlink/properties/linkbase
              </documentation>
            </annotation>
          </attribute>
        </restriction>
      </complexContent>
    </complexType>
  </element>
  <element name="schemaRef" type="xl:simpleType" substitutionGroup="xl:simple">
    <annotation>
      <documentation>
      Definition of the schemaRef element - used
      to link to XBRL taxonomy schemas from
      XBRL instance documents.
      </documentation>
    </annotation>
  </element>

  <element name="definition" type="string">
```

```
        <annotation>
          <documentation>
          The element to use for human readable definition
          of custom roles and arc roles.
          </documentation>
        </annotation>
    </element>
    <element name="roleType">
      <annotation>
        <documentation>
        The roleType element definition - used to define custom
        role values in XBRL extended links.
        </documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="link:definition" minOccurs="0" />
        </sequence>
        <attribute name="name" type="NCName" use="required" />
        <attribute name="usedOn" type="QName" use="required" />
      </complexType>
    </element>
    <element name="arcroleType">
      <annotation>
        <documentation>
        The  arcroleType  element  definition - used to define custom
        arc role values in XBRL extended links.
        </documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="link:definition" minOccurs="0" />
        </sequence>
        <attribute name="name" type="NCName" use="required" />
        <attribute name="usedOn" type="QName" use="required" />
        <attribute name="cyclesAllowed" use="required">
          <simpleType>
            <restriction base="NMTOKEN">
              <enumeration value="any" />
              <enumeration value="undirected" />
              <enumeration value="none" />
            </restriction>
          </simpleType>
        </attribute>
      </complexType>
    </element>
</schema>
```

xbrl-role-2003-07-31.xsd (normative)

```
<?xml version="1.0"?>
<!-- (c) XBRL International.  See www.xbrl.org/legal -->
<schema targetNamespace="http://www.xbrl.org/2003/role"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:role="http://www.xbrl.org/2003/role"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  elementFormDefault="qualified">

  <import namespace="http://www.xbrl.org/2003/linkbase"
    schemaLocation="xbrl-linkbase-2003-07-31.xsd"/>

  <annotation>
```

```
    <documentation>
    linkbaseRef role type definitions
    </documentation>
    <appinfo>
      <link:roleType name="calculationLinkbaseRef" usedOn="link:linkbaseRef">
        <link:definition>Must contain only calculationLink elements</link:definition>
      </link:roleType>
      <link:roleType name="definitionLinkbaseRef" usedOn="link:linkbaseRef">
        <link:definition>Must contain only definitionLink elements</link:definition>
      </link:roleType>
      <link:roleType name="labelLinkbaseRef" usedOn="link:linkbaseRef">
        <link:definition>Must contain only labelLink elements</link:definition>
      </link:roleType>
      <link:roleType name="presentationLinkbaseRef" usedOn="link:linkbaseRef">
        <link:definition>Must contain only presentationLink elements</link:definition>
      </link:roleType>
      <link:roleType name="referenceLinkbaseRef" usedOn="link:linkbaseRef">
        <link:definition>Must contain only referenceLink elements</link:definition>
      </link:roleType>
      <link:roleType name="footnoteLinkbaseRef" usedOn="link:linkbaseRef">
        <link:definition>Must contain only footnoteLink elements</link:definition>
      </link:roleType>
    </appinfo>
  </annotation>

  <annotation>
    <documentation>
    default extended link role type definitions
    </documentation>
    <appinfo>
      <link:roleType name="link" usedOn="link:calculationLink">
        <link:definition>The default standard extended link role for calculation
links</link:definition>
      </link:roleType>
      <link:roleType name="link" usedOn="link:definitionLink">
        <link:definition>The default standard extended link role for definition
links</link:definition>
      </link:roleType>
      <link:roleType name="link" usedOn="link:presentationLink">
        <link:definition>The default standard extended link role for presentation
links</link:definition>
      </link:roleType>
      <link:roleType name="link" usedOn="link:labelLink">
        <link:definition>The default standard extended link role for label
links</link:definition>
      </link:roleType>
      <link:roleType name="link" usedOn="link:referenceLink">
        <link:definition>The default standard extended link role for reference
links</link:definition>
      </link:roleType>
      <link:roleType name="link" usedOn="link:footnoteLink">
        <link:definition>MuThe default standard extended link role for footnote
links</link:definition>
      </link:roleType>
    </appinfo>
  </annotation>

  <annotation>
    <documentation>
    label resource role type definitions
    </documentation>
    <appinfo>
      <link:roleType name="label" usedOn="link:label">
```

```
      <link:definition>
      Standard label for a concept.
      </link:definition>
    </link:roleType>
    <link:roleType name="terseLabel" usedOn="link:label">
      <link:definition>
      Short label for a concept, often omitting text that should be
      inferable when the concept is reported in the context of other
      related concepts.
      </link:definition>
    </link:roleType>
    <link:roleType name="verboseLabel" usedOn="link:label">
      <link:definition>
      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.
      </link:definition>
    </link:roleType>
    <link:roleType name="positiveLabel" usedOn="link:label">
      <link:definition>
      Standard label for a concept when the value of the concept is positive.
      </link:definition>
    </link:roleType>
    <link:roleType name="positiveTerseLabel" usedOn="link:label">
      <link:definition>
      Terse label for a concept when the value of the concept is positive.
      </link:definition>
    </link:roleType>
    <link:roleType name="positiveVerboseLabel" usedOn="link:label">
      <link:definition>
      Verbose label for a concept when the value of the concept is positive.
      </link:definition>
    </link:roleType>
    <link:roleType name="zeroLabel" usedOn="link:label">
      <link:definition>
      Standard label of a concept when the value of the concept is negative.
      </link:definition>
    </link:roleType>
    <link:roleType name="zeroTerseLabel" usedOn="link:label">
      <link:definition>
      Terse label of a concept when the value of the concept is negative.
      </link:definition>
    </link:roleType>
    <link:roleType name="zeroVerboseLabel" usedOn="link:label">
      <link:definition>
      Verbose label of a concept when the value of the concept is negative.
      </link:definition>
    </link:roleType>
    <link:roleType name="negativeLabel" usedOn="link:label">
      <link:definition>
      Standard label of a concept when the value of the concept is negative.
      </link:definition>
    </link:roleType>
    <link:roleType name="negativeTerseLabel" usedOn="link:label">
      <link:definition>
      Terse label of a concept when the value of the concept is negative.
      </link:definition>
    </link:roleType>
    <link:roleType name="negativeVerboseLabel" usedOn="link:label">
      <link:definition>
      Verbose label of a concept when the value of the concept is negative.
      </link:definition>
    </link:roleType>
    <link:roleType name="totalLabel" usedOn="link:label">
```

```
      <link:definition>
      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.
      </link:definition>
    </link:roleType>
    <link:roleType name="periodStartLabel" usedOn="link:label">
      <link:definition>
      The label for a concept with instantaneous="true" for use in presenting
      values associated with the concept when it is being report as a beginning
      of period value.
      </link:definition>
    </link:roleType>
    <link:roleType name="periodEndLabel" usedOn="link:label">
      <link:definition>
      The label for a concept with instantaneous="true" for use in presenting
      values associated with the concept when it is being reported as an
      end of period value.
      </link:definition>
    </link:roleType>
    <link:roleType name="documentation" usedOn="link:label">
      <link:definition>
      Documentation of a concept, providing an explanation of its meaning and
      its appropriate usage and any other documentation deemed necessary.
      </link:definition>
    </link:roleType>
    <link:roleType name="definitionGuidance" usedOn="link:label">
      <link:definition>
      A precise definition of a concept, providing an explanation of its meaning
      and its appropriate usage.
      </link:definition>
    </link:roleType>
    <link:roleType name="disclosureGuidance" usedOn="link:label">
      <link:definition>
      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 place), or structural completeness (i.e. merely
      included to complete the structure of the taxonomy).
      </link:definition>
    </link:roleType>
    <link:roleType name="presentationGuidance" usedOn="link:label">
      <link:definition>
      An explanation of the rules guiding presentation (placement and/or labeling)
      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".
      </link:definition>
    </link:roleType>
    <link:roleType name="placementGuidance" usedOn="link:label">
      <link:definition>
      An explanation of the rules guiding placement of this concept in the context
      of other concepts in one or more specific types of business reporting.
      </link:definition>
    </link:roleType>
    <link:roleType name="measurementGuidance" usedOn="link:label">
      <link:definition>
      An explanation of the method(s) required to be used when measuring values
      associated with this concept in business reports.
      </link:definition>
    </link:roleType>
    <link:roleType name="commentary" usedOn="link:label">
```

```
      <link:definition>
      Any other general commentary on the concept that assists in determining
      definition, disclosure, measurement, presentation or usage.
      </link:definition>
    </link:roleType>
    <link:roleType name="example" usedOn="link:label">
      <link:definition>
      An example of the type of information intended to be captured by the concept.
      </link:definition>
    </link:roleType>
  </appinfo>
</annotation>

<annotation>
  <documentation>
  Reference Resource Role Types
  </documentation>
  <appinfo>
    <link:roleType name="reference" usedOn="link:reference">
      <link:definition>Standard reference for a concept</link:definition>
    </link:roleType>
    <link:roleType name="definitionRef" usedOn="link:reference">
      <link:definition>
      Reference to documentation that details a precise definition of the concept.
      </link:definition>
    </link:roleType>
    <link:roleType name="disclosureGuidanceRef" usedOn="link:reference">
      <link:definition>
      Reference to documentation that details an explanation of the disclosure
      requirements relating to the concept.
      </link:definition>
    </link:roleType>
    <link:roleType name="mandatoryDisclosureGuidanceRef" usedOn="link:reference">
      <link:definition>
      Reference to documentation that details an explanation of the mandatory
      disclosure requirements relating to the concept.
      </link:definition>
    </link:roleType>
    <link:roleType name="recommendedDisclosureGuidanceRef" usedOn="link:reference">
      <link:definition>
      Reference to documentation that details an explanation of the recommended
      disclosure requirements relating to the concept.
      </link:definition>
    </link:roleType>
    <link:roleType name="unspecifiedDisclosureGuidanceRef" usedOn="link:reference">
      <link:definition>
      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 and 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.
      </link:definition>
    </link:roleType>
    <link:roleType name="presentationRef" usedOn="link:reference">
      <link:definition>
      Reference to documentation which details an explanation of the presentation or
      labeling of this concept in the context of other concepts in one or more
      specific types of financial reports.
      </link:definition>
    </link:roleType>
    <link:roleType name="measurementRef" usedOn="link:reference">
      <link:definition>
      Reference concerning the method(s) required to be used when measuring values
```

```
          associated with this concept in business reports.
          </link:definition>
        </link:roleType>
        <link:roleType name="commentaryRef" usedOn="link:reference">
          <link:definition>
          Any other general commentary on the concept that assist in determining
          appropriate usage.
          </link:definition>
        </link:roleType>
        <link:roleType name="exampleRef" usedOn="link:reference">
          <link:definition>
          Reference to documentation that illustrates by example the application of
          the concept that assists in determining appropriate usage.
          </link:definition>
        </link:roleType>
      </appinfo>
    </annotation>

    <annotation>
      <documentation>
      arc role type definitions
      </documentation>
      <appinfo>
        <link:arcroleType name="parent-child" usedOn="link:presentationArc"
cyclesAllowed="undirected">
          <link:definition>
          This arc role value is used to form a tree. It contains end points
          that represent a parent and the related child.
          </link:definition>
        </link:arcroleType>
        <link:arcroleType name="summation-item" usedOn="link:calculationArc"
cyclesAllowed="none">
          <link:definition>
          This arc role value is used to show how an item participates in a summation.
          It consists of an item end point and a summation end point where the item is
          summed into the summation.
          </link:definition>
        </link:arcroleType>
        <link:arcroleType name="general-special" usedOn="link:definitionArc"
cyclesAllowed="undirected" >
          <link:definition>
          This arc role value is used to show a definitional relationship between a
          generalised concept and its related specialised concept.
          </link:definition>
        </link:arcroleType>
        <link:arcroleType name="essence-alias" usedOn="link:definitionArc"
cyclesAllowed="undirected" >
          <link:definition>
          This arc role value is used to show a definitional relationship between a
          concept that represents a different perspective of another concept.
          </link:definition>
        </link:arcroleType>
        <link:arcroleType name="similar-tuples" usedOn="link:definitionArc"
cyclesAllowed="any" >
          <link:definition>
          This arc role value is used to relate tuples that are similar at a definition
level
          despite differences at a content model level.
          </link:definition>
        </link:arcroleType>
        <link:arcroleType name="requires-element" usedOn="link:definitionArc"
cyclesAllowed="any" >
          <link:definition>
```

```
            If an instance of the concept at the source of the arc occurs in an XBRL
instance
        then a c-equal instance of the arc's target concept MUST also occur in the XBRL
        instance. Fully conformant XBRL processors MUST detect and signal instances in
        which this relationship is violated.
        </link:definition>
      </link:arcroleType>
      <link:arcroleType name="concept-label" usedOn="link:labelArc"
cyclesAllowed="none">
        <link:definition>
        This arc role value is used to connect from a concept to a label.
        </link:definition>
      </link:arcroleType>
      <link:arcroleType name="concept-reference" usedOn="link:referenceArc"
cyclesAllowed="none">
        <link:definition>
        This arc role is used to connect from a concept to a reference.
        </link:definition>
      </link:arcroleType>
      <link:arcroleType name="fact-footnote" usedOn="link:footnoteArc"
cyclesAllowed="none">
        <link:definition>
        This arc role is used to connect from a facts in an XBRL instance to a footnote.
        </link:definition>
      </link:arcroleType>
    </appinfo>
  </annotation>
</schema>
```

xbrl-ref-2003-07-31.xsd (normative)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- (c) XBRL International. See www.xbrl.org/legal -->
<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-2003-07-31.xsd"/>

  <element name="name" type="string" substitutionGroup="link:part" id="name"/>
  <element name="number" type="string" substitutionGroup="link:part" id="number"/>
  <element name="paragraph" type="string" substitutionGroup="link:part" id="paragraph"/>
  <element name="subparagraph" type="string" substitutionGroup="link:part"
id="subparagraph"/>
  <element name="clause" type="string" substitutionGroup="link:part" id="clause"/>
  <element name="pages" type="string" substitutionGroup="link:part" id="pages"/>

</schema>
```

xml-2001-03.xsd (normative)

```
<?xml version="1.0"?>
<!-- DOCTYPE xs:schema PUBLIC "-//W3C//DTD XMLSCHEMA 200102//EN" "XMLSchema.dtd" -->
<xs:schema targetNamespace="http://www.w3.org/XML/1998/namespace"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xml:lang="en">
```

```
<xs:annotation>
  <xs:documentation>
  See http://www.w3.org/XML/1998/namespace.html and
  http://www.w3.org/TR/REC-xml for information about this namespace.
  </xs:documentation>
</xs:annotation>

<xs:annotation>
  <xs:documentation>
  This schema defines attributes and an attribute group
  suitable for use by schemas wishing to allow xml:base,
  xml:lang or xml:space attributes on elements they define.
  To enable this, such a schema must import this schema
  for the XML namespace, e.g. as follows:
  &lt;schema .  .>
  .  .
  &lt;import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>

  Subsequently, qualified reference to any of the attributes
  or the group defined below will have the desired effect, e.g.

  &lt;type .  .>
  .  .
  &lt;attributeGroup ref="xml:specialAttrs"/>

  will define a type which will schema-validate an instance
  element with any of those attributes
  </xs:documentation>
</xs:annotation>

<xs:annotation>
  <xs:documentation>
  In keeping with the XML Schema WG's standard versioning
  policy, this schema document will persist at
  http://www.w3.org/2001/03/xml.xsd.
  At the date of issue it can also be found at
  http://www.w3.org/2001/xml.xsd.
  The schema document at that URI may however change in the future,
  in order to remain compatible with the latest version of XML Schema
  itself. In other words, if the XML Schema namespace changes, the version
  of this document at
  http://www.w3.org/2001/xml.xsd will change
  accordingly; the version at
  http://www.w3.org/2001/03/xml.xsd will not change.
  </xs:documentation>
</xs:annotation>

<xs:attribute name="lang" type="xs:language">
  <xs:annotation>
    <xs:documentation>
    In due course, we should install the relevant ISO 2- and 3-letter
    codes as the enumerated possible values .  .
    </xs:documentation>
  </xs:annotation>
</xs:attribute>

<xs:attribute name="space" default="preserve">
  <xs:simpleType>
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="default"/>
      <xs:enumeration value="preserve"/>
    </xs:restriction>
```

```
      </xs:simpleType>
    </xs:attribute>

    <xs:attribute name="base" type="xs:anyURI">
      <xs:annotation>
        <xs:documentation>
        See http://www.w3.org/TR/xmlbase/ for
        information about this attribute.
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>

    <xs:attributeGroup name="specialAttrs">
      <xs:attribute ref="xml:base"/>
      <xs:attribute ref="xml:lang"/>
      <xs:attribute ref="xml:space"/>
    </xs:attributeGroup>

</xs:schema>
```

xlink.xsd (normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) XBRL International. See www.xbrl.org/legal -->
<schema targetNamespace="http://www.w3.org/1999/xlink"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
    XLink attribute specification
    </documentation>
  </annotation>

  <simpleType name="nonEmptyURI">
    <annotation>
      <documentation>
      A URI type with a minimum length of 1 character.
      Used on role and arcrole and href elements.
      </documentation>
    </annotation>
    <restriction base="anyURI">
      <minLength value="1"/>
    </restriction>
  </simpleType>

  <simpleType name="typeEnum">
    <annotation>
      <documentation>
    Enumeration of values for the type attribute
    </documentation>
    </annotation>
    <restriction base="string">
      <enumeration value="simple"/>
      <enumeration value="extended"/>
      <enumeration value="locator"/>
      <enumeration value="arc"/>
      <enumeration value="resource"/>
      <enumeration value="title"/>
    </restriction>
  </simpleType>
```

```
<simpleType name="showEnum">
  <annotation>
    <documentation>
    Enumeration of values for the show attribute
    </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="new"/>
    <enumeration value="replace"/>
    <enumeration value="embed"/>
    <enumeration value="other"/>
    <enumeration value="none"/>
  </restriction>
</simpleType>

<simpleType name="actuateEnum">
  <annotation>
    <documentation>
    Enumeration of values for the actuate attribute
    </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="onLoad"/>
    <enumeration value="onRequest"/>
    <enumeration value="other"/>
    <enumeration value="none"/>
  </restriction>
</simpleType>

<attribute name="type" type="xlink:typeEnum"/>
<attribute name="role" type="xlink:nonEmptyURI"/>
<attribute name="arcrole" type="xlink:nonEmptyURI"/>
<attribute name="title" type="string"/>
<attribute name="show" type="xlink:showEnum"/>
<attribute name="actuate" type="xlink:actuateEnum"/>
<attribute name="label" type="NCName"/>
<attribute name="from" type="NCName"/>
<attribute name="to" type="NCName"/>
<attribute name="href" type="anyURI"/>
```

xl.xsd (normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) XBRL International. See www.xbrl.org/legal -->
<schema targetNamespace="http://www.xbrl.org/2001/XLink"
  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"
  attributeFormDefault="unqualified">

  <import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="xlink.xsd"/>

  <complexType name="documentationType">
    <annotation>
      <documentation>
      Element type to use for documentation of
      extended links and linkbases.
      </documentation>
    </annotation>
```

```
          <simpleContent>
            <extension base="string">
              <anyAttribute/>
            </extension>
          </simpleContent>
      </complexType>
      <element name="documentation" type="xl:documentationType" abstract="true">
        <annotation>
          <documentation>
          Abstract element to use for documentation of
          extended links and linkbases.
          </documentation>
        </annotation>
      </element>

      <annotation>
        <documentation>
        XBRL simple and extended link schema constructs
        </documentation>
      </annotation>

      <complexType name="titleType">
        <annotation>
          <documentation>
          Type for the abstract title element -
          used as a title element template.
          </documentation>
        </annotation>
        <complexContent>
          <restriction base="anyType">
            <attribute ref="xlink:type" use="required">
              <annotation>
                <documentation>
                This attribute must have a value of "title".
                </documentation>
              </annotation>
            </attribute>
          </restriction>
        </complexContent>
      </complexType>
      <element name="title" type="xl:titleType" abstract="true">
        <annotation>
          <documentation>
          Generic title element for use in extended link documentation.
          Used on extended links, arcs, locators.
          See http://www.w3.org/TR/xlink/#title-element for details.
          </documentation>
        </annotation>
      </element>

      <complexType name="locatorType">
        <annotation>
          <documentation>
          Generic locator type.
          </documentation>
        </annotation>
        <complexContent>
          <restriction base="anyType">
            <sequence>
              <element ref="xl:title" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
            <attribute ref="xlink:type" use="required">
              <annotation>
```

```
          <documentation>
          This attribute must have a value of "locator".
          </documentation>
        </annotation>
      </attribute>
      <attribute ref="xlink:href" use="required" />
      <attribute ref="xlink:label" use="required" />
      <attribute ref="xlink:role" use="optional" />
      <attribute ref="xlink:title" use="optional" />
    </restriction>
  </complexContent>
</complexType>
<element name="locator" type="xl:locatorType" abstract="true">
  <annotation>
    <documentation>
    Abstract locator element to be used as head of locator substitution group
    for all extended link locators in XBRL.
    </documentation>
  </annotation>
</element>
<simpleType name="useEnum">
  <annotation>
    <documentation>
    Enumerated values for the use attribute  on extended link arcs.
    </documentation>
  </annotation>
  <restriction base="NMTOKEN">
    <enumeration value="optional" />
    <enumeration value="prohibited" />
  </restriction>
</simpleType>
<complexType name="arcType">
  <annotation>
    <documentation>
    basic extended link  arc  type - extended  where necessary  for  specific arcs
    Extends  the  generic  arc  type by  adding use,  priority and order attributes.
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <sequence>
        <element ref="xl:title" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
      <attribute ref="xlink:type" use="required">
        <annotation>
          <documentation>
          This attribute must have a value of "arc".
          </documentation>
        </annotation>
      </attribute>
      <attribute ref="xlink:from" use="required" />
      <attribute ref="xlink:to" use="required" />
      <attribute ref="xlink:arcrole" use="required" />
      <attribute ref="xlink:title" use="optional" />
      <attribute ref="xlink:show" use="optional" />
      <attribute ref="xlink:actuate" use="optional" />
      <attribute name="order" type="decimal" use="optional" />
      <attribute name="use" type="xl:useEnum" use="optional" />
      <attribute name="priority" type="integer" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
    </restriction>
  </complexContent>
</complexType>
```

```xml
<element name="arc" type="xl:arcType" abstract="true">
  <annotation>
    <documentation>
    Abstract element to use as head of arc element substitution group.
    </documentation>
  </annotation>
</element>
<complexType name="resourceType">
  <annotation>
    <documentation>
    Generic type for the resource type element
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <attribute ref="xlink:type" use="required">
        <annotation>
          <documentation>
          This attribute must have a value of "resource".
          </documentation>
        </annotation>
      </attribute>
      <attribute ref="xlink:label" use="required" />
      <attribute ref="xlink:role" use="optional" />
      <attribute name="id" type="ID" use="optional" />
    </restriction>
  </complexContent>
</complexType>
<element name="resource" type="xl:resourceType" abstract="true">
  <annotation>
    <documentation>
    Abstract element to use as head of resource element substitution group.
    </documentation>
  </annotation>
</element>
<complexType name="extendedType">
  <annotation>
    <documentation>
    Generic extended link type
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xl:title" />
        <element ref="xl:documentation" />
        <element ref="xl:locator" />
        <element ref="xl:arc" />
        <element ref="xl:resource" />
      </choice>
      <attribute ref="xlink:type" use="required">
        <annotation>
          <documentation>
          This attribute must have a value of "extended".
          </documentation>
        </annotation>
      </attribute>
      <attribute ref="xlink:role" use="required" />
      <attribute ref="xlink:title" use="optional" />
      <attribute name="id" type="ID" use="optional" />
    </restriction>
  </complexContent>
</complexType>
```

```
<element name="extended" type="xl:extendedType" abstract="true">
  <annotation>
    <documentation>
    Abstract extended link element at head of extended link substitution group.
    </documentation>
  </annotation>
</element>
<complexType name="simpleType">
  <annotation>
    <documentation>
    Type for the simple links defined in XBRL
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <attribute ref="xlink:type" use="required">
        <annotation>
          <documentation>
          This attribute must have a value of "simple".
          </documentation>
        </annotation>
      </attribute>
      <attribute ref="xlink:href" use="required" />
      <attribute ref="xlink:arcrole" use="optional" />
      <attribute ref="xlink:role" use="optional" />
      <attribute ref="xlink:title" use="optional" />
      <attribute ref="xlink:show" use="optional" />
      <attribute ref="xlink:actuate" use="optional" />
    </restriction>
  </complexContent>
</complexType>
<element name="simple" type="xl:simpleType" abstract="true">
  <annotation>
    <documentation>
    The abstract element at the head of the simple link substitution group.
    </documentation>
  </annotation>
</element>
```

B. 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, Don Dwiggins of UBmatrix, Charles Hoffman of UBmatrix, Josef MacDonald of Ernst & Young, Manabu Mizutani of PCS, David Prather of IASCF, Campbell Pryde of KPMG, Noboyuki Sambuichi of Hitachi, and Eiichi Watanabe of TSR, all contributed to the authoring and refinement of requirements and reviewing of the specification.

2003-07-31 [Shuetrim] Made the usedOn attribute a QName and eliminated the enumeration restriction on it. Changed the schema dates to 2003-07-31 from 2003-07-28. Corrected the definition of arc equivalence to cover prohibition of arcs to resources. Introduced the requirement that a DTS must include a taxonomy schema that imports the XBRL-instance schema. Prohibited values of zero for the `weight` attribute on `calculationArc` elements. Eliminated the XHTML content in simple links.

2003-07-30 [Shuetrim] Introduced the section on XLink and XBRL. Reorganised the sections on extended links, linkbases and simple links to refer to the new section on XLink and XBRL. Reorganised the section on taxonomy extended links to bring together all materials for each type of extended link into the one sub-section. Reorganised the section on XBRL instances to bring together the various sections dealing with syntax in taxonomy schemas. Clarified the definition of arc equivalence to make the definition no longer contingent on extraneous attribute values. Added the requirement that the balance attribute only be used on items with a monetaryItemType or a type derived therefrom. Clarified the interpretation of tupleTypes being final with respect to extension. Changed the font to Verdana from Times New Roman. Modified restrictions on parent-child arcs to allow undirected cycles. Updated the xbrl-role.xsd schema to reflect the new syntax. Updated the schema appendix to reflect current syntax.

2003-07-29 [Shuetrim] Removed the items types: NOTATIONitemType, NMTOKENItemType, NMTOKENItemType, NMTOKENSItemType, IDItemType, IDREFItemType, IDREFSItemType, ENTITYItemType and ENTITIESItemType. Changed the content model for the xbrl element to require a schemaRef element and to require that the schemaRef elements occur first, followed by linkbaseRef elements, followed by the other possible children in any order. Also introduced the requirement that at least one schemaRef element occurs as a child of the xbrl element. Changed the name of the numerator and denominator child elements of the divide element to be called unitNumerator and unitDenominator to avoid a naming clash with the fractionItemType children. Added the documentation element to be used for documentation on linkbase elements and extended link elements. Added ID attributes to the linkbase and extended link elements. Changed the DTS discovery model to ensure linkbases contained in discovered schemas are also discovered. Clarified the treatment of linkbases that are nested within taxonomy schemas. Modified the definition of taxonomy schemas to allow XML Schemas that do not import the XBRL instance schema.

2003-07-28 [Shuetrim] Modified the schemas to tighten the XML Schema constraints on extended links and their content. Clarified the definition of arc equivalence to cover arcs from concepts to resources instead of just concepts to other concepts. Modified the calculationLink XML Schema content model to allow flexible ordering of children.

2003-07-23 [Shuetrim] Changed the instantaneous attribute to be called the periodType attribute. Added the schemaRef element. Rearranged the standard arc role value sections, merging them with the descriptions of each of the specific arc elements. Separated the unit element from the context element. Changed the numericContext and nonNumericContext to a

single context element and modified the attributes on items to reference unit and context elements using contextRef and unitRef attributes.

2003-07-22. [Shuetrim] References to MIME type have been removed from the specification. Moved the section on the linkbase related schemas to the appendix listing the text of the various schema documents supporting this specification. Modified the syntax for the unit element to eliminate the `multiply` element. Added the section on levels of conformance of XBRL processors.

2003-07-20. [Shuetrim] Removed profile attributes. Removed the references to deprecated syntax, eliminating the syntax instead. Removed the aloc, absoluteContext and relativeContext elements from the calculationLink element, removing the capacity for expressing cross-context calculations using the calculationLink. Removed the references to an ability to associate concepts to remote labels Removed the CWA attribute. Changed all rules expressed in terms of processing errors or fatal errors into rules expressed in terms of MUST and MUST NOT style requirements.

2003-06-09. [Wallis] Numerous editorial changes, clarifications etc. Incorporated changes pursuant to the resolution of comments 025 (no change needed), 030 (no change required), 032, 034, 036, 037, 045, 055

2003-05-16. [Wallis] Incorporated changes pursuant to the resolution of comments 003, 004, 005, 006, 007, 008, 009, 010, 011, 013, 014, 015, 018, 019, 020, 021, 022, 026, 028

2003-04-29. [Wallis] Formatting, table headings (bolding and repeating on new pages), prevent table cells splitting across pages where appropriate, font, pagination, hyperlinking and typographical changes.

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 node 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 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.

C. 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).

D. 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	2003-06-25
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			