

Extensible Business Reporting Language (XBRL) 2.0 Specification

2001-12-14

Comments should be posted to XBRL-public@yahoogroups.com by 14 December 2001

Editors:

Luther Hampton, e-Numerate
David vun Kannon, KPMG LLP

XBRL.org

Specification Group

Chairs:

Luther Hampton, e-Numerate
David vun Kannon, KPMG LLP

This version:

<http://www.xbrl.org/tr/2001/xbrl-2001-12-14.doc> (in Word)

<http://www.xbrl.org/tr/2001/xbrl-2001-12-14.pdf> (in PDF)

with separate provision of [Schemas](#) described herein. All components, along with non-normative samples and certain schemas are available in a single Zip format archive.

[Copyright](#) ©2001 [xbrl.org](#)® All Rights Reserved. XBRL liability, [trademark](#), [document use](#) and [software licensing](#) rules apply.

Abstract

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

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

Acknowledgements

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. Yufei Wang has done most of the work creating examples, writing stylesheets, and providing important technical support. Walter Hamscher provided key insights, wordsmithing, and perspective.

Status of this document

This document is final.

This document is a revision of the draft dated 14 November 2001 and has incorporated suggestions received during last call review, comments, and further deliberations of the XBRL Specification Working Group.

The comment period was open through 14 December 2001. All comments continue to be welcome and should be posted to XBRL-public@yahogroups.com.

While excerpts from the XBRL schema are given throughout this document the authoritative version of the XBRL schema is that version available as a separate .xsd file from the XBRL.org web site.

Table of contents	Page
Extensible Business Reporting Language (XBRL) 2.0 Specification	1
2001-12-14	1
Abstract	1
Acknowledgements	1
Status of this document	2
Table of contents Page	3
1 Introduction	5
1.1 Documentation Conventions	5
1.2 Purpose	5
1.3 Relationship to Other Work	6
1.4 Terminology	6
2 Changes from the previous published version	7
2.1 Changes in the Instance Document	7
2.2 Changes in the Taxonomy Document	8
2.3 Substantive changes from the previous draft (November 14, 2001)	8
3 XBRL Framework	9
3.1 Document Types and MIME types	10
4 Syntax of Instance Documents	10
4.1 The item element	10
4.2 The context elements (<i>numericContext</i> , <i>nonNumericContext</i>)	12
4.2.1 id	13
4.2.2 period	13
4.2.3 unit	15
4.2.4 precision (<i>numericContext</i> only)	16
4.2.5 cwa (<i>numericContext</i> only)	16
4.2.6 The entity sub-element	17
4.2.7 The scenario sub-element	19
4.3 The tuple element	20
4.4 The group element	21
4.5 Footnotes	22
4.5.1 footnoteLink extended-type link element	22
4.5.2 loc locator-type element	22
4.5.3 footnote resource-type element	23
4.5.4 footnoteArc arc-type element	23

5	Syntax of Taxonomies	24
5.1	<i>The monetary and shares datatypes</i>	25
5.2	<i>element</i>	26
5.2.1	balance attribute	26
5.3	<i>Linkbases</i>	27
5.3.1	linkbaseRef element	27
5.3.2	Extended Link-type elements	28
5.3.3	Locator-type elements (loc)	29
5.3.4	Resource-type elements (label and reference)	30
5.3.5	Arc-type elements	31
5.3.6	Linkbase schema	36
6	Semantics of Instance Documents	39
6.1	<i>Processing by consuming applications</i>	39
6.2	<i>Validation</i>	39
6.3	<i>The Parent-Child relationship</i>	39
6.4	<i>Data Integrity and Confidentiality</i>	39
7	Semantics of Taxonomies	40
8	References (Non-normative)	40
9	Change Log	40

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 instance documents. These concrete 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.

1.1 Documentation Conventions

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

XML Declarations

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

Example
A non-normative example illustrating use of the XBRL language, or a related instance.
<code><schema name="http://www.example.com/XMLSchema/1.0/mySchema" /></code>
And an explanation of the example.

NOTE: General comments directed to all readers.

1.2 Purpose

The XBRL specification is meant to maximize benefits to all stakeholders that use it. The specification is intended to benefit three categories of users: business information preparers, intermediaries in the preparation and distribution process, and users of this information. There is also a fourth category of beneficiary, the vendors who supply software and services to one or more of these three types of user. The overall intention is to balance the needs of these groups creating a product that provides benefits to all groups.

The needs of end users of business information will generally have precedence over other needs when it is 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 standards. However, XBRL should facilitate possible changes in reporting over the long term.

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

XBRL should facilitate "drill down" to detailed information, authoritative literature, audit and accounting working papers. XBRL should include 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 should support international accounting standards and languages other than the American dialect of English.

XBRL should be extensible by any adopter to increase its breadth of applicability, and its design should encourage reuse via incremental extensions. XBRL should specify the format of information that would be reasonably expected in an electronic format for securities filings by public entities. XBRL should facilitate business reporting in the long term, and should not be 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 be used to distract from the true and fair presentation of results. Therefore, there is no functional requirement that XBRL documents need to support any particular text formatting conventions.

The intention of XBRL instance documents is just the transmission of some set of facts. There is no constraint on how much or how little they contain. A single item can be a valid XBRL document, for example, when the information being conveyed is limited to, for example, what Cost of Goods Sold was last quarter. An XBRL document can be a database dump. It can be anything in between. This provides a great deal of flexibility and is meant specifically to achieve the goals of allowing XBRL to be reused within other specifications and for application software to be able to most easily extract data from otherwise arbitrarily formatted documents. It is expected that for most uses of XBRL, many instance documents will be created that consist almost exclusively of items.

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 is not transactional. It includes financial reporting and contemplates extensive detail in the representation and use of accounting conventions, which distinguishes it from these other efforts.

1.4 Terminology

The terminology used in XBRL frequently overlaps with terminology from other fields, and the following short list is provided to reduce the possibility of ambiguity and confusion.

item	A fact reported within a given period of time about a given business entity. Corresponds to an abstract XML element "item" in XBRL.
taxonomy	An XML Schema that defines new elements each corresponding to a concept that can be referenced in XBRL documents. XBRL taxonomies can be regarded as extensions of XML Schemas, including XML Link-based information.
entity	A business entity, the subject of XBRL items. Where the XML/SGML concept of syntactic "entity" is meant, this will be pointed out.
group	The root element of an XBRL instance document. Groups may also serve as generic containers for other elements inside an XBRL instance document.
tuple	A term meaning a group of items which must be kept together to be understood. The XBRL tuple element is a container used to hold items together.
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. Items that report on instants and durations are both common.
CWA	"Closed World Assumption", a term used in logic and database theory. If the closed world assumption is true, the database or XBRL document is understood to hold all the facts necessary to do further computations.
element	An XML element, but also a concept described by a taxonomy. For example, the element with the name "nonCurrentAssets.propertyPlantAndEquipmentNet" is a concept.
instance	An XML document containing XBRL elements that together constitute one or more statements. The financial statements of IBM, expressed in XBRL, would be an instance.
may	Conforming documents and consuming applications are permitted to but need not behave as described.
shall	Conforming documents and consuming applications are required to behave as described; otherwise they are in error.
error	A violation of the rules of this specification; results are undefined. Conforming software may detect and report an error and may recover from it.
fatal error	An error, which a consuming application must detect and report. After encountering a fatal error, the application may continue processing the data to search for further errors and may report such errors. In order to support correction of errors, the processor may make unprocessed data from the document (with intermingled character data and markup) available to the application. Once a fatal error is detected, however, the processor shall not continue normal processing (<i>i.e.</i> , it shall not continue to pass character data and information about the document's contents to the application in the normal way).
at user option	Conforming software may or shall (depending on the modal verb in the sentence) behave as described; if it does, it shall provide users a means to enable or disable the behavior described.

2 Changes from the previous published version

Changes from the previous, July 2000 version of XBRL were driven by two factors. New technology that became available, such as XML Schema and XML Linking, was one factor. The other factor was implementation experience. Implementations revealed that the July 2000 version needed more flexible metadata and easier metadata addressing.

2.1 Changes in the Instance Document

The primary change to the XBRL instance document has been to adopt XML Schema validation as a primary technical goal. The benefits of this validation were seen as compelling, when compared to the level of validation available from the DTD for the July 2000 version of XBRL. These include:

- Validity of namespaces
- Validity of element names within namespaces
- Type checking of instance data
- Instance document structure

Further, the 'inherited attribute' style of the July 2000 version has been abandoned in favor of a context element, which collects all the metadata for an item in a single element. This change was driven by the change in markup for segment and scenario, as well as improved readability for XPath expressions involving XBRL documents. The segment sub-element of context allows for finer definition of the business entity, while the scenario sub-element allows for the definition of additional, proprietary metadata markup.

The `cwa` attribute has been added. This attribute documents the validity of the “closed world assumption” for a set of items.

Adherence to XML Schema also modified the set of metadata attributes. XBRL previously had a `schemaLocation` attribute that provided the functionality now contained in the `xsi:schemaLocation` attribute defined in XML Schema. XBRL now relies directly on `xsi:schemaLocation` for this functionality. The XBRL monetary datatype previously permitted alternate lexical representations of numbers, documented by the `decimalPattern` and `formatName` attributes. Since XML Schema specifies the lexical representation of decimal numbers, these attributes are no longer necessary. This decision also means that thousands separators are no longer allowed.

The tuple concept has been separated from the group concept. The label element as a separate element in the instance document has been eliminated due to changes in the structure of taxonomy documents. Overriding labels can now be segregated into separate linkbase documents.

NOTE: A consequence of choosing XML Schema validation with sophisticated features such as substitution groups is that it is no longer possible to write an all-encompassing DTD. This is because certain information can only be expressed using the schema; it cannot be expressed in the DTD and is therefore not available to DTD-based application systems. For instance, it is possible to state the default for an attribute in the schema and the schema validation process will respect it, in the sense that the PSVI (Post Schema Validation Infoset) will include the appropriate item. That doesn't mean anything to the current crop of SAX, DOM and XSLT based systems, which are not based on XML Schema. For these systems, the attribute never appeared, and since there is no DTD at hand, there is nothing to do. For such systems new XBRL documents are not 'valid', they are only well formed. Giving up DTD-based validity in favor of XML Schema-based validity puts XBRL ahead of the tools vendors and other standards efforts in this regard, but it was felt that the gains in functionality offered by XML Schema outweighed the loss of the DTD as an all-encompassing document validator.

2.2 Changes in the Taxonomy Document

Substitution groups based on the `item` and `tuple` elements are now the basis for defining the unique elements of a taxonomy and the syntactic constraints they must obey. Taxonomy elements that substitute for `item` represent concepts, and so can legally appear anywhere in an instance document that an `item` element could. Taxonomy elements that substitute for `tuple` represent containers, which have no content of their own, and can legally appear anywhere in an instance document that a `tuple` element could. Element definitions should contain the appropriate substitution group: `item` or `tuple`.

Rollups have been elaborated into three separate elements (definition, calculation, and presentation). In addition, rollup, label and reference information that was previously embedded in the XML Schema element definition has been factored out into an adjunct file or files. These files are structured as XLink-compliant linkbases. These changes allow taxonomy authors to document the intellectual structure more accurately, as well as allow for the third-party emendation of taxonomies in a manner that was not available in the July 2000 version.

2.3 Substantive changes from the previous draft (November 14, 2001)

The specification previously included restrictions on extended link structures. The following paragraph has been REMOVED from section 5.3.2.

“XBRL extended link-type elements are restricted in certain ways, compared to the general XLink syntax. `LabelLink` can link a label resource with only one element. Only one locator can be referenced by the `from` and `to` attributes of `calculationArc`, `definitionArc`, and `presentationArc`.”

The following sentence has been REMOVED from section 5.2.

“For the sake of easier integration with XLink, all element elements shall have an id attribute, with content equal to the content of the name attribute.”

A Note has been added to section 5.3 describing the use of remote content for label and reference links

Added arcroles in section 5.3.5.7 to support the idea of concept equivalency. Concept equivalency allows taxonomies to include a concept and variants of the concept which look at that concept as having a particular breakdown. See section 5.3.5.7.1.

3 XBRL Framework

The main ideas in the XBRL conceptual framework are *items* and *taxonomies*. This section bears careful reading, since these terms are used in a precise way within XBRL.

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

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

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

Groups. In XBRL, a *group* is a set of related *items* that can appear in any order and can be interspersed among other text and elements in any XML document. The root of an XBRL Instance Document is a group, which may, in turn, contain other groups. There is, therefore, no "XBRL document type" as such. It is possible in principle to embed an XBRL item in *any* document, such as a press release that is otherwise formatted in HTML.

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

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

Although any given item can only refer to a single taxonomy, within any given XBRL document any number of XBRL items can refer to any number of taxonomies.

Individual XBRL taxonomies can also be used as “building blocks” to create larger, more sophisticated taxonomies. Users may compose groups of existing taxonomies into higher-level taxonomies and may selectively add segments to taxonomies to specialize them: Suppose, for example, that a significant portion of expenses is (in a hospital, for example) "physician salaries". Because that term does not exist in the Financial Reporting for Commercial and Industrial Companies, US GAAP taxonomy as such, a new (small) taxonomy would be defined which defined the term "physician salaries" and referred to the US GAAP taxonomy so as to relate this to the concept of "expenses" that already exists there.

3.1 Document Types and MIME types

Proper detection of an XBRL document depends on the XBRL namespace. The namespace for the current version is <http://www.xbrl.org/2001/instance>. A [MIME](#) type allows applications to detect an XBRL document without needing to open the document.

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

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

Registration will occur in conjunction with the final publication of this specification.

4 Syntax of Instance Documents

An XBRL instance document shall comply with the general rules and structures specified herein. In addition, all XBRL Instance documents shall be valid documents as defined by the XML Schema validation criteria in [[SCHEMA-1](#)] and shall be recognized as such by any XML Schema validation compliant system.

The core syntax for instance documents is defined using an XML Schema. The elements defined there are *item*, *context*, *tuple* and *group*.

4.1 The *item* element

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

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >
<!-- *****item types***** -->
<!-- itemType -->
<complexType name="itemType">
  <simpleContent>
    <extension base="anySimpleType">
      <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
<!-- monetaryItemType -->
<complexType name="monetaryItemType">
  <simpleContent>
    <extension base="xbrli:monetary">
      <attribute name="numericContext" type="IDREF"
use="required"/>
    <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
```

```

</complexType>
<!--          sharesItemType          -->
<complexType name="sharesItemType">
  <simpleContent>
    <extension base="xbrli:shares">
      <attribute name="numericContext" type="IDREF"
use="required"/>
      <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
<!--          decimalItemType          -->
<complexType name="decimalItemType">
  <simpleContent>
    <extension base="decimal">
      <attribute name="numericContext" type="IDREF"
use="required"/>
      <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
<!--          stringItemType          -->
<complexType name="stringItemType">
  <simpleContent>
    <extension base="string">
      <attribute name="nonNumericContext" type="IDREF"
use="required"/>
      <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
<!--          uriItemType          -->
<complexType name="uriItemType">
  <simpleContent>
    <extension base="anyURI">
      <attribute name="nonNumericContext" type="IDREF"
use="required"/>
      <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
<!--          dateTimeItemType          -->
<complexType name="dateTimeItemType">
  <simpleContent>
    <extension base="xbrli:dateUnion">
      <attribute name="nonNumericContext" type="IDREF"
use="required"/>
      <anyAttribute namespace="##any"
processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>

```

```

<!--          item          -->
<element name="item" type="xbrli:itemType" abstract="true"/>

</schema>

```

Example

```

< ci:capitalLeasedAssetsNet.capitalLeasedAssetsGross numericContext="c1">727
</ci:capitalLeasedAssetsNet.capitalLeasedAssetsGross>

```

Meaning: The value of Gross Capital Leases in the numeric context labeled c1 is 727. Note that it will be necessary to consult the context (defined below) in order to determine other details concerning the value such as unit, period, precision, etc.

Example

```

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

```

Meaning: The text of the Concentrations note in the string context labeled c1.

The item element is defined with a datatype based on anyType. Each member of the substitution group can be defined with the appropriate type. This allows each substitution for item in the instance to validate against its own datatype, which may be the monetary type provided by XBRL or a more complex type, including other markup.

The numericContext or nonNumericContext attribute is an [IDREF](#) to the appropriate kind of context element (see below) that holds the relevant metadata. An item shall always contain a context attribute that references a context element.

No two item elements in an XBRL document shall be identical. Two item elements shall be considered identical if their content, attributes and attribute content, and the content, attributes and attribute content of their corresponding contexts are identical; that is, they share the same namespace and identical string values.

4.2 The context elements (numericContext, nonNumericContext)

The context elements, numericContext and nonNumericContext, are the holders of various metadata in attributes and element content that provide the necessary context for understanding a financial fact captured in an XBRL item. The nonNumericContext element contains metadata and attributes relating to textual facts; the numericContext provides similar elements for numeric facts. The context elements shall conform to the following definitions:

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >
<!-- context: nonNumericContext/numericContext elements -->
<!--          nonNumericContext          -->
<element name="nonNumericContext">
  <complexType>
    <sequence>
      <element name="entity" type="xbrli:entityType"/>
      <element name="period" type="xbrli:periodType"/>
      <element name="unit" type="xbrli:unitType"/>
    </sequence>
  </complexType>
</element>
minOccurs="0"/>

```

```

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

```

Each attribute is described separately below.

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

4.2.1 id

Every context element shall include this attribute. The content shall be in accordance with the XML Schema rules for attributes with the ID type. The `id` attribute shall identify the context (see §4.2 above) of the item.

Example
<code>id="C2424"</code>

4.2.2 period

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

Example	Meaning
<code>instant</code>	A specific point in time
<code>forever</code>	An element to represent 'forever'.
<code>duration options</code>	
<code>startDate, duration</code>	A period beginning as specified, for the given length of time.
<code>duration, endDate</code>	A period of the given length of time, ending as specified.

startDate, endDate	A period beginning and ending as specified.
--------------------	---

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

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >

  <!--*****context: period elements *****-->
  <simpleType name="dateUnion">
    <union memberTypes="date dateTime"/>
  </simpleType>
  <!--          startDate          -->
  <element name="startDate" type="xbrli:dateUnion"/>
  <!--          endDate          -->
  <element name="endDate" type="xbrli:dateUnion"/>
  <!--          duration          -->
  <element name="duration" type="duration"/>
  <!--          instant          -->
  <element name="instant" type="xbrli:dateUnion"/>
  <!--          forever          -->
  <element name="forever">
    <complexType/>
  </element>
  <!--          context/periodType          -->
  <complexType name="periodType">
    <choice>
      <sequence minOccurs="0">
        <element ref="xbrli:startDate"/>
        <element ref="xbrli:endDate"/>
      </sequence>
      <sequence minOccurs="0">
        <element ref="xbrli:startDate"/>
        <element ref="xbrli:duration"/>
      </sequence>
      <sequence minOccurs="0">
        <element ref="xbrli:duration"/>
        <element ref="xbrli:endDate"/>
      </sequence>
      <element ref="xbrli:instant" minOccurs="0"/>
      <element ref="xbrli:forever" minOccurs="0"/>
    </choice>
  </complexType>
  <!--          end of context/periodType          -->
</schema>

```

Sub-element	XML Schema Datatype
instant	date or dateTime.
forever	empty
startDate	date or dateTime

endDate	date or dateTime
duration	duration

4.2.3 unit

The unit element shall specify the standard that is relevant to the measurement. It is expected that most measurements will be monetary measurements. ISO 4217 standard currency designation shall be used for the unit element in such a case. (ISO 4217) Pure numbers and counts of people, shares and the like shall be specified as quantities.

Enumerations depend on the taxonomy in force to specify the datatype of the element as an enumerated datatype, and to provide the allowable values.

The unit sub-element is required in the numericContext. The unit sub-element is optional in the nonNumericContext element so that items with values drawn from enumerations can document the enumeration as the unit of measure.

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

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >
<!-- ***** context: unit element ***** --
>
<!-- _____context:measure element -->
<element name="measure" type="QName"/>
<!-- _____context:operatorNameEnum -->
<simpleType name="operatorNameEnum">
  <restriction base="string">
    <enumeration value="multiply"/>
    <enumeration value="divide"/>
  </restriction>
</simpleType>
<!-- _____context:operator element -->
<element name="operator">
  <complexType>
    <choice minOccurs="2" maxOccurs="2">
      <element ref="xbrli:measure"/>
      <element ref="xbrli:operator"/>
    </choice>
    <attribute name="name" type="xbrli:operatorNameEnum"
use="required"/>
  </complexType>
</element>
<!-- _____context:unitType -->
<complexType name="unitType">
  <choice>
    <element ref="xbrli:measure"/>
    <element ref="xbrli:operator"/>
  </choice>
</complexType>
</schema>
```

Example	Meaning
<code><unit><measure>ISO4217:GBP</measure></unit></code>	Currency, UK Pounds.
<pre> <unit> <operator name="divide"> <measure>ISO4217:EUR</measure> <measure>xbrli:shares</measure> </operator> </unit> </pre>	Earnings per share (EPS) measured in Euros per share

Note: Since measure uses Qualified Names, the prefixes in the above examples must have been previously defined in namespace declarations.

4.2.4 precision (numericContext only)

The precision attribute shall be an integer 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.

Example	Meaning
<code>precision="9"</code>	Precision of nine significant decimal digits.

4.2.5 cwa (numericContext only)

The cwa attribute shall be a boolean value indicating the validity of the ‘closed world assumption’. If cwa="true" then the ‘closed world assumption’ shall be valid over the set of items which refer to this context and consuming applications can assume that there is no information missing from the provided data. If cwa="false" a consuming application shall not attempt to calculate any new value based on the information given.

An abbreviated example instance document:

```

<group xmlns="http://www.xbrl.org/2001/instance"
xmlns:ci="http://www.xbrl.org/us/gaap/ci/2001/us-gaap-ci-2001">
<ci:intangibles numericContext="c1">7</ci:intangibles>
<ci:cash numericContext="c1">2</ci:cash>
<ci:notesPayable numericContext="c1">5</ci:notesPayable>
<ci:bondsPayable numericContext="c1">1</ci:bondsPayable>
<ci:treasuryStock numericContext="c1">2</ci:treasuryStock>
<ci:commonStock numericContext="c1">1</ci:commonStock>
<numericContext id="c1" cwa="true"/>
</group>

```

In the taxonomy referred to by the ci: prefix there are the following links:

```

<calculationArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
xlink:from="intangibles" xlink:to="assets" weight="1"/>
<calculationArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"

```

```
xlink:from="cash" to="assets" weight="1"/>
<calculationArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
xlink:from="notesPayable" xlink:to="liabilities" weight="1"/>
<calculationArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
xlink:from="bondsPayable" xlink:to="liabilities" weight="1"/>
<calculationArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
xlink:from="treasuryStock" xlink:to="equity" weight="1"/>
<calculationArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
xlink:from="commonStock" xlink:to="equity" weight="1"/>
```

The above abbreviated example would allow software to deduce/assume/calculate:

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

4.2.6 The entity sub-element

The entity element documents the organization (business, government department, individual, etc.) for which the financial fact is true. The entity element shall be required content of the context element. The entity element shall contain an identifier element and may contain multiple optional segment elements.

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

4.2.6.1 identifier

An identifier element specifies a system for identifying business entities. The scheme attribute contains the namespace URI of the identification scheme, providing a framework for referencing naming authorities.

The element content shall be a [string](#) that is a valid identifier within the namespace referenced by the scheme attribute. XBRL.org 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	Meaning
<pre><identifier scheme="">SAMP</identifier> <identifier scheme="www.nasdaq.com">SAMP </identifier></pre>	<p>Some entity known only as SAMP within the default namespace.</p> <p>The company with NASDAQ ticker symbol SAMP.</p>
<pre><identifier scheme="www.dnb.com">0236503276 </identifier></pre>	<p>The company or subsidiary with number 0236503276 (not a real D-U-N-S® number).</p>
<pre><identifier scheme="www.cusip.org">41009876 AB</identifier></pre>	<p>The entity with CUSIP number 41009876AB (e.g. a mutual fund).</p>
<pre><identifier scheme="www.ietf.org/URI">www.w 3c.org</identifier></pre>	<p>The non-profit organization owning the URI www.w3c.org.</p>

4.2.6.2 segment

The segment element is an optional container for additional markup that the preparer of an instance document may use to identify the business segment more completely.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >
  <element name="segment">
    <complexType>
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xbrli:segment" minOccurs="0"
maxOccurs="unbounded"/>
        <any namespace="##any" minOccurs="0"
maxOccurs="unbounded"/>
      </choice>
      <attribute name="name" type="string" use="optional"/>
    </complexType>
  </element>
</schema>
```

```
<group xmlns="http://www.xbrl.org/2001/instance"
xmlns:my="http://www.someCompany.com/scenarios"
xmlns:other="http://www.example.com/">
<nonNumericContext id="c1" >
  <entity>
    <!--required content -->
    <identifier scheme="www.dnb.com">1234567890</identifier>
    <!-- optional content -->
```

```

        <segment>
            <my:state>NJ</my:state>
        </segment>
    </entity>
    <!-- optional content -->
    <scenario>
        <other:budgeted/>
    </scenario>
</nonNumericContext>
</group>

```

Meaning: The preparer has used a <segment> to indicate that the financial facts relate to operations in the state of New Jersey. Note that the namespace "other" need not be explicitly defined.

NOTE: In general, the content of a segment will be application specific. Therefore, it is up to the preparer of the document to provide the proper namespace support and xsi:schemaLocation hints necessary to insure that the segment element is properly validated by an XML Schema validation process.

4.2.7 The scenario sub-element

Financial 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 associated with items. The optional scenario element allows additional valid markup (see note above) to be included for this purpose.

```

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

```

```

<group xmlns="http://www.xbrl.org/2001/instance"
xmlns:fid="http://www.someInsuranceCo.com/scenarios"
xmlns:other="http://www.example.com" >
<nonNumericContext id="c1" >
    <scenario>
        <other:bestEstimate/>
        <fid:dwSlice>
            <fid:residence>MA</fid:residence>
            < fid:nonSmoker>true</fid:nonSmoker>
            < fid:minAge>34</fid:minAge>
            < fid:maxAge>49</fid:maxAge>
        </fid:dwSlice>
    </scenario>
</nonNumericContext>
</group>

```

```

    </scenario>
</ nonNumericContext>
</group>

```

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. Note again that the namespace "other" need not be explicitly defined.

NOTE: 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.

4.3 The tuple element

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

The tuple element is abstract: Therefore, all XBRL elements of this type shall be substitution groups having initial origin in tuple. Specific elements from other taxonomies may be in the tuple substitution group. Tuples may contain both items and other tuples.

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >
  <complexType name="tupleType">
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="xbrli:item" minOccurs="0"
maxOccurs="unbounded" />
      <element ref="xbrli:tuple" minOccurs="0"
maxOccurs="unbounded" />
    </choice>
  </complexType>
  <element name="tuple" type="xbrli:tupleType" abstract="true" />
</schema>

```

An abbreviated example schema document:

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance" >
  <element name="managementInformation" substitutionGroup="xbrli:tuple" />

  <element name="managementName" type="xbrli:stringItemType"
substitutionGroup="xbrli:item" />

  <element name="managementTitle" type="xbrli:stringItemType"
substitutionGroup="xbrli:item" />
</schema>

```

In the linkbase are the following links:

```

<linkbase xmlns="http://www.xbrl.org/2001/XLink/xbrllinkbase "
xmlns:xlink="http://www.w3.org/1999/xlink" >
  <definitionArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent "

```

```

xlink:from="managementName" xlink:to="managementInformation"/>
<definitionArc xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
xlink:from="managementTitle" xlink:to="managementInformation"/>
</linkbase>
The abbreviated instance document:
<group xmlns="http://www.xbrl.org/2001/instance"
xmlns:ci="http://www.xbrl.org/us/gaap/ci/2001/us-gaap-ci-2001">
<ci:managementInformation>
    <ci:managementName nonNumericContext="c1">Bill Ballmer
    </ci:managementName>
    <ci:managementTitle nonNumericContext="c1">Big Kahuna
    </ci:managementTitle>
</ci:managementInformation>
<ci:managementInformation>
    <ci:managementName nonNumericContext="c2">Steve Gates</ci:managementName>
    <ci:managementTitle nonNumericContext="c2">Kahuna Emeritus
    </ci:managementTitle>
</ci:managementInformation>
</group>

```

4.4 The group element

The group element is the generic container element of the XBRL vocabulary. It shall be the root element of XBRL instance documents.

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2001/instance">
<element name="group">
    <complexType>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xbrli:group" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="xbrli:item" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="xbrli:tuple" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="xbrli:nonNumericContext" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="xbrli:numericContext" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="link:linkbaseRef" minOccurs="0"
maxOccurs="unbounded"/>
            <element ref="link:footnoteLink" minOccurs="0"
maxOccurs="unbounded"/>
        </choice>
    </complexType>
</element>
</schema>

```

Example

```
<group xmlns="http://www.xbrl.org/2001/instance"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:ci="http://www.xbrl.org/us/gaap/ci/2001/us-gaap-ci-2001"
       xsi:schemaLocation="
http://www.xbrl.org/us/gaap/ci/2001/us-gaap-ci-2001
http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31\_v2.xsd">
  <ci:assets numericContext="c1">727</ci:assets >
  <ci:liabilities numericContext="c1">635</ci:liabilities >
  <numericContext id="c1" cwa="false"/>
</group>
```

Meaning: Use of group as the root element, holding namespace prefix definitions and the xsi:schemaLocation attribute.

4.5 Footnotes

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

- the presence of a link;
- the type (simple, similar to HTML’s , or extended);
- the specific resources or elements it connects;
- the directionality of any arcs within the link (e.g., from parent to child or from child to parent);
- a token indicating any application-specific semantics of the link (e.g., the “parent-of” role).

Familiarity with XLink is presumed throughout the following text.

4.5.1 footnoteLink extended-type link element

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

4.5.1.1 xlink:type attribute

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

4.5.1.2 xlink:role attribute

The xlink:role attribute is an optional attribute that can be used to differentiate the purpose or relevance of various footnotes.

4.5.2 loc locator-type element

The loc element of the footnoteLink has exactly the same syntax as the loc element used in the extended links for taxonomies. See 5.3.3 for details.

4.5.3 footnote resource-type element

The footnote resource element has exactly the same syntax as the label resource element. See 5.3.4 and 5.3.4.2 for details.

4.5.4 footnoteArc arc-type element

The footnoteArc element has exactly the same syntax as the labelArc element. See 5.3.5 for details

4.5.4.1 xlink:arcrole attribute

The content shall be a string.

In order to allow bi-directional navigation, there will often be arcs from A to B as well as B to A. Therefore the to and from attributes of the arc-type element cannot be consistently understood as “from the child, to the parent”, which was the kind of arc expressed by the previous version of XBRL’s rollup element. Instead, the arcrole attribute shall be used to disambiguate whether the arc has the semantic meaning of “parent-to-child” or “child-to-parent”.

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

4.5.4.2 xlink:title attribute

The content shall be a string.

The title content may be visible to users of presentation and XLink-enabled applications

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

Example of a footnote in an instance document.
<pre><?xml version="1.0"?> <group xmlns="http://www.xbrl.org/2001/instance" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ias="http://www.iasb.org.uk/xbrl/2001-08-16/" xmlns:link="http://www.xbrl.org/2001/XLink/xbrllinkbase" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:iso4217="http://www.iso.org/4217" xsi:schemaLocation="http://www.iasb.org.uk/xbrl/2001-08-16/ ias.xsd"> <ias:bs> <ias:asset-cce-net numericContext="c1">100</ias:asset-cce-net> <ias:asset-rec-net numericContext="c1">700</ias:asset-rec-net> <ias:asset-cur-tot numericContext="c1">800</ias:asset-cur-tot> <ias:asset-ppe-cost-gross numericContext="c1">1200</ias:asset-ppe-cost- gross> <ias:asset-intan-cost-mastheads numericContext="c1">600</ias:asset- intan-cost-mastheads> <ias:asset-nonCur-total numericContext="c1">1800</ias:asset-nonCur- total> <ias:asset-total id="f1" numericContext="c1">2600</ias:asset-total> <ias:liab-overdraft numericContext="c1">500</ias:liab-overdraft> <ias:liab-cur-total numericContext="c1">500</ias:liab-cur-total> <ias:liab-pay-bondsPay numericContext="c1">1000</ias:liab-pay-bondsPay> <ias:liab-nonCur-total numericContext="c1">1000</ias:liab-nonCur-total> <ias:eq-cap-apic numericContext="c1">900</ias:eq-cap-apic> <ias:eq-re numericContext="c1">200</ias:eq-re> <ias:eq-total id="f3" numericContext="c1">1100</ias:eq-total></pre>

```

        <ias:liab-total id="f2" numericContext="c1">2600</ias:liab-total>
</ias:bs>
<link:footnoteLink xlink:type="extended" xlink:role="Merger" xlink:title="1">
    <link:footnote xlink:type="resource" xlink:label="footnotel"
xlink:title="1" xlink:role="standard" xml:lang="en">Including the effects
of the merger.</link:footnote>
    <link:footnote xlink:type="resource" xlink:label="footnotel"
xlink:title="1" xlink:role="standard" xml:lang="fr">Y compris les effets de
la fusion.</link:footnote>
    <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#f1"/>
    <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#f2"/>
    <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#f3"/>
    <link:footnoteArc xlink:from="fact1" xlink:to="footnotel"
xlink:title="fact1-footnotel"
xlink:arcrole="http://www.xbrl.org/linkprops/arc/fact-footnote"
xlink:show="replace" xlink:actuate="onRequest"/>
    <link:footnoteArc xlink:from="footnotel" xlink:to="fact1"
xlink:title="footnotel-fact1"
xlink:arcrole="http://www.xbrl.org/linkprops/arc/footnote-fact"
xlink:show="replace" xlink:actuate="onRequest"/>
</link:footnoteLink>
<numericContext id="c1" precision="18" cwa="true">
    <entity>
        <identifier scheme="http://www.un.org/">Example plc</identifier>
        <segment/>
    </entity>
    <period>
        <instant>2001-08-16</instant>
    </period>
    <unit><measure>iso4217:EUR</measure></unit>
    <scenario name="Actual values">
        <ias:scenarioType>actual</ias:scenarioType>
    </scenario>
</numericContext>
</group>

```

5 Syntax of Taxonomies

A taxonomy shall consist of a list of element, attribute, or datatype definitions. Additionally, there may be definitions of the relations between these elements themselves or between these elements and the elements of another taxonomy. A taxonomy is defined using the XML Schema vocabulary and a series of elements that implement a set of Xlink-compliant linkbases. No matter how many physical files are involved, an XBRL taxonomy is considered to be the combination of element definitions and links, which are authored as a unit. If links are included in a taxonomy, the schema document shall contain an XLink simple link, which points to the linkbase, as documented in the XML Linking Recommendation of the W3C (Section 5.1.5 Locating Linkbases).

An XBRL taxonomy document is a valid instance of an XML Schema document. Each taxonomy document shall use the standard XML Schema import element to reference the XBRL schemas. Every XBRL Taxonomy shall import the XBRL instance document schema.

The XBRL instance document schema defines the abstract elements item and tuple. These definitions are used in the substitution group declarations. The instance document schema also defines the basic datatypes used in XBRL taxonomies. (This schema imports the XBRL linkbase schema.)

In general, several other schemas will be referred to with namespace declarations, such as the schema for XML Schema and a self-reference to the schema under construction.

Example
<pre> <schema targetNamespace="http://www.xbrl.org/us/gaap/ci/2001/us-gaap-ci-2001" xmlns="http://www.w3.org/2001/XMLSchema" </pre>

```

xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:xbrli="http://www.xbrl.org/2001/instance"
xmlns:link="http://www.xbrl.org/XLink/xbrllinkbase"
xmlns:ci="http://www.xbrl.org/us/gaap/ci/2001/us-gaap-ci-2001"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <annotation>
    <appinfo>
      <link:linkbaseRef xlink:type="simple"
xlink:href="linkbase_presentation.xml" xlink:actuate="onRequest"
xlink:role="http://www.xbrl.org/linkprops/linkRef/presentation"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for presentation relationship</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef xlink:type="simple" xlink:href="linkbase
_calculation.xml" xlink:actuate="onRequest"
xlink:role="http://www.xbrl.org/linkprops/linkRef/calculation"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for calculation relationship</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef xlink:type="simple" xlink:href="linkbase
_definition.xml" xlink:actuate="onRequest"
xlink:role="http://www.xbrl.org/linkprops/linkRef/definition"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for definition relationship</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef xlink:type="simple" xlink:href="linkbase
_label.xml" xlink:actuate="onRequest"
xlink:role="http://www.xbrl.org/linkprops/linkRef/label"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for labels</xhtml:p>
      </link:linkbaseRef>
      <link:linkbaseRef xlink:type="simple" xlink:href="linkbase
_reference.xml" xlink:actuate="onRequest"
xlink:role="http://www.xbrl.org/linkprops/linkRef/reference"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase">
        <xhtml:p>Links for literature references</xhtml:p>
      </link:linkbaseRef>
    </appinfo>
  </annotation>
  <import namespace="http://www.xbrl.org/2001/instance"
schemaLocation="core\xbrl-instance.xsd"/>
</schema>

```

A template for a taxonomy containing appropriate namespaces, imports and links.

XBRL Taxonomies can and should include other taxonomies as appropriate. Each taxonomy publisher is then responsible only for the new concepts defined in the published taxonomy, not those imported from existing taxonomies.

5.1 The monetary and shares datatypes

The XBRL schema defines the monetary datatype, which specializes the XML Schema decimal type. All numeric elements in XBRL Taxonomies that represent monetary values shall use this datatype.

```

<schema xmlns="http://www.w3.org/2001/XMLSchema" >
  <!-- ***** simple types***** -->
    <!-- monetary -->

```

```

    <simpleType name="monetary">
      <annotation>
        <documentation>
          This is the datatype for those financial concepts in a
          taxonomy that denote units in a currency.
          Instance items with this type should have a unit of measure
          from the ISO4217 namespace of currencies.
        </documentation>
      </annotation>
      <restriction base="decimal"/>
    </simpleType>
    <!--          shares          -->
    <simpleType name="shares">
      <annotation>
        <documentation>
          This is the datatype for share-based financial concepts.
        </documentation>
      </annotation>
      <restriction base="decimal">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </schema>

```

5.2 element

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

An element may have an optional id attribute.

Examples
<pre> <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xbrli="http://www.xbrl.org/2001/instance"> <element id="paymentOfDividends.preferredDividends" name="paymentOfDividends.preferredDividends" type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/> <element id="significantAccountingPoliciesNote.stockBasedCompensationPolicy" name="significantAccountingPoliciesNote.stockBasedCompensationPolicy" type="xbrli:stringItemType" substitutionGroup="xbrli:item"/> </schema> </pre>
Meaning: Typical element definitions.

5.2.1 balance attribute

An optional attribute may be added to the element definition. If the idea of debit/credit balance is appropriate to the element, it can be documented using this attribute.

Examples
<pre> <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xbrli="http://www.xbrl.org/2001/instance"> <element id="myCredit" name="myCredit" xbrli:balance="credit" type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/> <element id="myDebit" name="myDebit" xbrli:balance="debit" type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"/> </pre>

</schema>

Meaning: Debit and credit element definitions.

5.3 Linkbases

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

- Relation links (calculation, definition, and presentation) manage the relations between taxonomy elements.
- Label links manage the text associated with taxonomy elements in various languages.
- Literature links manage the references to authoritative literature (either online or paper).

These extended links shall be held in an XLink document container. The document container can be the XML Schema document of the taxonomy or a separate document which contains an XBRL linkbase element as the root element.

NOTE: Literature and label linkbases are designed with the assumption that local resources (the reference and label elements, respectively) will be used. If instead, a linkbase author wishes to use remote content pointed to by loc (locator-type) elements, the following rules should be observed:

- The remote content shall be in an XML or HTML document.
- In a labelLink, the remote content shall be an XBRL label element or another (XML or HTML) element. If it is not an XBRL label element, then the label is defined to be the element content of that element. The element shall allow the attachment of the XBRL label element's role attribute as well as the XML lang attribute.
- In a referenceLink, the remote content shall be an XBRL reference element or another (XML or HTML) element. If it is not an XBRL reference element, then the element content shall be understood as the actual authoritative literature itself, NOT a citation of the literature. All literature citations shall be contained in XBRL reference elements exactly the same as reference elements used as local resources.

5.3.1 linkbaseRef element

The XLink specification provides for a standard way of [finding linkbases](#). The linkbaseRef element conforms to this standard by using a specific xlink:arcrole content value.

5.3.1.1 xlink:type attribute

The xlink:type attribute has the fixed content "simple". For the benefit of applications (such as current XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schema (xbrl-linkbase.xsd), this attribute shall be given explicitly in every linkbaseRef element.

5.3.1.2 xlink:href attribute

The xlink:href attribute contains a URI. The URI points to a particular linkbase.

5.3.1.3 xlink:role attribute

The xlink:role attribute identifies the kind of extended link contained in the linkbase. Listed below are role values for the standard kinds of XBRL extended links.

Suggested values for linkbaseRef xlink:role

<http://www.xbrl.org/linkprops/linkRef/presentation>

```

http://www.xbrl.org/linkprops/linkRef/calculation
http://www.xbrl.org/linkprops/linkRef/definition
http://www.xbrl.org/linkprops/linkRef/label
http://www.xbrl.org/linkprops/linkRef/reference

```

5.3.1.4 xlink:arcrole attribute

The xlink:arcrole attribute has the XLink standard fixed content “http://www.w3.org/1999/xlink/properties/linkbase”. For the benefit of applications (such as current XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schema (xbrl-linkbase.xsd), this attribute shall be given explicitly in every linkbaseRef element.

5.3.1.5 xlink:actuate attribute

In XBRL linkbases, the xlink:actuate attribute has the fixed content “onRequest”. No link traversal is considered to be mandatory. No link needs to be traversed as soon as a file is loaded. For the benefit of applications (such as current XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schema (xbrl-linkbase.xsd), this attribute shall be given explicitly in every linkbaseRef element.

5.3.2 Extended Link-type elements

These are the five extended type link elements of XBRL taxonomies.

Standard extended-type link elements
calculationLink – contain locator and calculationArc elements
definitionLink - contain locator and definitionArc elements
presentationLink – contain locator and presentationArc elements
labelLink – contain locator, label and labelArc elements
referenceLink – contain locator, reference, and referenceArc elements

5.3.2.1 xlink:type attribute

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

5.3.2.2 xlink:role attribute

The table below lists several standard role values. These are intended to be useful to writers of accounting related taxonomies. If the taxonomy writer divides the links into functional groups, application developers will not be burdened with reading all the links when only some are necessary. The values given are suggestions. Other values can be created as the taxonomy author deems necessary or useful.

Suggested
Extended link xlink:role
http://www.xbrl.org/linkprops/extended/balanceSheet

```

http://www.xbrl.org/linkprops/extended/incomeStatement
http://www.xbrl.org/linkprops/extended/statementOfComprehensiveIncome
http://www.xbrl.org/linkprops/extended/statementOfStockholdersEquity
http://www.xbrl.org/linkprops/extended/cashFlows

```

5.3.3 Locator-type elements (loc)

For consistency, the same locator element is used in each of the five extended link elements of XBRL taxonomy linkbases.

5.3.3.1 xlink:type attribute

The xlink:type attribute has the fixed content "locator". For the benefit of applications (such as current XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schema (xbrl-linkbase.xsd), this attribute shall be given explicitly in every locator element.

5.3.3.2 xlink:href attribute

The xlink:href attribute contains a URI. In the five kinds of extended links for XBRL taxonomies, the URI points to an element in an XML Schema file.

To point to a particular node (element in a schema, item in an instance) the URI will end in a fragment identifier. According to the XLink specification, XPointer syntax is allowed in the fragment identifier. XBRL restricts the allowed forms of the fragment identifier.

Fragment identifier	Notes
# <i>id</i>	The node pointed to must have an id attribute whose content is <i>id</i> .
#xpointer(<i>XPath expression</i>)	The XPath expression is restricted to what is commonly understood by DOM and XSLT processors. XPointer extension functions are not allowed.

5.3.3.3 xlink:label attribute

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

5.3.3.4 xlink:role attribute

There is one distinguished functional role for locator-type elements, to point to the root of a tree of elements in a taxonomy. Many taxonomies will contain one or more such trees of elements, and each kind of link will have its own set of trees. Distinguishing the root of a tree eases the burden on applications to discover the information on their own.

Suggested
locator xlink:role
http://www.xbrl.org/linkprops/locator/root

5.3.4 Resource-type elements (label and reference)

5.3.4.1 xlink:type attribute

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

5.3.4.2 xlink:label attribute

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

5.3.4.3 Label resources

One of the key internationalization features of XBRL is that although each taxonomy defines a single set of elements representing a coherent set of business reporting concepts, the label—a string used to present the name of that concept—is declared separately with an indication of the language using the XML standard lang attribute. Thus, a given document could be presented by a single application in a language selected by the user (Recasting the underlying business concepts under a different set of national accounting principles is a far more complex matter).

Label elements shall use the standard XML lang attribute, and they shall appear inside labelLink extended link elements. Label content is mixed content containing a simple string, or a fragment of XHTML. The XHTML shall be restricted to TEXT module elements (see [Modularization of XHTML](#) for details).

5.3.4.4 xlink:role attribute

Label elements may contain an optional role attribute, which can disambiguate multiple labels in the same language by their intended use. The table below provides two standard roles for labels.

Suggested
Label resource xlink:role
http://www.xbrl.org/linkprops/label/standard
http://www.xbrl.org/linkprops/label/total

Example
<pre><label xlink:type="resource" xlink:label="ci_assets.currentAssets_en" xlink:title=" ci_assets.currentAssets_en" xlink:role="http://www.xbrl.org/linkprops/label/standard" xml:lang="en"> Current Assets </label> <loc xlink:type="locator" xlink:href="us_bs_v2.xsd#assets.currentAsset" xlink:label="ci_assets.currentAssets " xlink:title="ci_assets.currentAssets"/> <labelArc xlink:type="arc" xlink:from="ci_assets.currentAssets" xlink:to="ci_assets.currentAssets_en" xlink:show="embed" xlink:arcrole="http://www.xbrl.org/linkprops/arc/element-label" xlink:actuate="onRequest" xlink:title="Go to label of ci_assets.currentAssets_en"/> <labelArc xlink:type="arc" xlink:from="ci_assets.currentAssets_en" xlink:to="ci_assets.currentAssets" xlink:show="replace" xlink:arcrole="http://www.xbrl.org/linkprops/arc/label-element" xlink:actuate="onRequest" xlink:title="Go to element ci_assets.currentAssets"/></pre>
Meaning: The label element contains the text of the label and the two arc elements allow bi-directional

navigation between element and label. The additional attributes of show, actuate and title allow XLink aware software to navigate the linkbase.

5.3.4.5 Reference resources

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

Reference elements shall appear inside linkbases. Reference elements shall be composed of parts. Since the division of metadata into parts varies in every jurisdiction, part is an abstract element. Taxonomy writers are free to create elements that substitute for part, to be included inside reference elements.

Example

```
<reference xlink:type="resource"
xlink:label="ci_noncurrentAssets.propertyPlantAndEquipmentNet_APB"
xlink:title="ci_noncurrentAssets.propertyPlantAndEquipmentNet_APB">
<ci:name>APB</ci:name>
<ci:number>12</ci:number>
<ci:paragraph>15</ci:paragraph>
</reference>
<loc xlink:type="locator"
xlink:href="us_bs_v2.xsd#noncurrentAssets.propertyPlantAndEquipmentNet"
xlink:label="ci_noncurrentAssets.propertyPlantAndEquipmentNet"
xlink:title="ci_noncurrentAssets.propertyPlantAndEquipmentNet"/>
<referenceArc xlink:type="arc"
xlink:arcrole="http://www.xbrl.org/linkprops/arc/element-reference"
xlink:from="ci_noncurrentAssets.propertyPlantAndEquipmentNet"
xlink:to="ci_noncurrentAssets.propertyPlantAndEquipmentNet_APB"
xlink:show="embed" xlink:actuate="onRequest" xlink:title="Go to reference of
ci_noncurrentAssets.propertyPlantAndEquipmentNet_APB"/>
<referenceArc xlink:type="arc"
xlink:arcrole="http://www.xbrl.org/linkprops/arc/reference-element"
xlink:from="ci_noncurrentAssets.propertyPlantAndEquipmentNet_APB"
xlink:to="ci_noncurrentAssets.propertyPlantAndEquipmentNet"
xlink:show="replace" xlink:actuate="onRequest" xlink:title="Go to element
ci_noncurrentAssets.propertyPlantAndEquipmentNet"/>
```

Meaning: The reference element contains the literature citation and the two arc elements allow bi-directional navigation between element and reference. The additional attributes of show, actuate and title allow XLink aware software to navigate the linkbase. The elements name, number and paragraph must be defined as members of the part substitution group in the taxonomy referred to by the ci: namespace prefix.

5.3.5 Arc-type elements

Arc-type elements join the resources referenced in their from and to attributes. These attributes contain labels of either locators or resources within the same extended link as the arc. Every arc-type element has the following attributes.

5.3.5.1 xlink:type attribute

The xlink:type attribute has the fixed content "arc". For the benefit of applications (such as current XSL and DOM based applications) that need this information, but cannot find it in the XBRL linkbase schema (xbrl-linkbase.xsd), this attribute shall be given explicitly in every arc-type element.

5.3.5.2 xlink:show attribute

In XBRL linkbases, the xlink:show attribute has the content “embed” or “replace”. Use “embed” in situations when a resource is linked to an element in a different file, such as a label and an element. Use “replace” to link elements in the same file.

5.3.5.3 xlink:actuate attribute

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

5.3.5.4 xlink:title attribute

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

5.3.5.5 xlink:to attribute

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

5.3.5.6 xlink:from attribute

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

5.3.5.7 xlink:arcrole attribute

The content shall be a string.

In order to allow bi-directional navigation, there will often be arcs from A to B as well as B to A. Therefore the to and from attributes of the arc-type element cannot be consistently understood as “from the child, to the parent”, which was the kind of link expressed by the previous version of XBRL’s rollup element. Instead, the arcrole attribute shall be used to disambiguate whether the arc has the semantic meaning of “parent-to-child” or “child-to-parent”.

Suggested	
calculationArc, definitionArc, presentationArc xlink:arcrole	Meaning
http://www.xbrl.org/linkprops/arc/child-parent	The arc is from child to parent. For definition arcs, the equivalent of ‘is-a’.
http://www.xbrl.org/linkprops/arc/parent-child	The arc is from parent to child. For definition arcs, the equivalent of ‘has-a’.
http://www.xbrl.org/linkprops/arc/dimension-element	The arc is from an element to an equivalent element that represents one perspective on this concept. For definition arcs, the equivalent of ‘same-as’.
http://www.xbrl.org/linkprops/arc/element-dimension	The arc is from an element to an equivalent element that represents one perspective on this concept. For definition arcs, the equivalent of ‘same-as’.

5.3.5.7.1 Concept equivalency

Sometimes a taxonomy might want to include a single concept viewed from different perspectives or as having several different dimensions.

<p>Cash</p> <ul style="list-style-type: none"> ■ Cash by Branch Location <ul style="list-style-type: none"> ■ Cash in Domestic Branches ■ Cash in Foreign Branches ■ Cash by Account Type <ul style="list-style-type: none"> ■ Cash in Interest Bearing Accounts ■ Cash in Non-interest Bearing Accounts ■ Cash by Availability <ul style="list-style-type: none"> ■ Cash on Hand ■ Cash as Balances Due
--

In the example above, the concept Cash is equivalent to Cash by Branch Location, Cash by Account Type, and Cash by Availability. The three equivalent concepts do not sum up to Cash, each of them is individually equal to Cash. For this reason, calculationArcs that join a concept to the set of its equivalents are treated specially.

Suggested	
labelArc xlink:arcrole	Meaning
http://www.xbrl.org/linkprops/arc/label-element	The arc is from a label to an element.
http://www.xbrl.org/linkprops/arc/element-label	The arc is from an element to a label.

Suggested	
referenceArc xlink:arcrole	Meaning
http://www.xbrl.org/linkprops/arc/reference-element	The arc is from a reference to an element.
http://www.xbrl.org/linkprops/arc/element-reference	The arc is from an element to a reference.
http://www.xbrl.org/linkprops/arc/actual-element	The arc is from actual literature content to an element.
http://www.xbrl.org/linkprops/arc/element-actual	The arc is from an element to actual literature content.

5.3.5.8 Overriding arcs

Note: XLink does not define the interaction of multiple linkbases from multiple authors in any way. There are situations in XBRL taxonomy construction when a third party may want to edit the set of links

constructed by a taxonomy 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.

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

As a motivating example, consider the situation of a third party desiring to create a new “sub-total” between a parent element and some of its children. It is not enough for the creator of the sub-total element to simply add links from the children to the sub-total and from the sub-total to the parent. If that was all that was done, there would be two paths from the child to the parent, one using the new links through the sub-total, and the other using the original link direct to the parent. In the case of calculation links, this could result in the double counting of values.

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

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

5.3.5.8.1 use attribute

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

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

- use = “required” shall indicate that the presence in the instance document of an element will require the presence of the other element for the document to be semantically valid.

For instance, the data which normally was filled in to a paper form could be represented electronically using XBRL instance documents. To represent the “required field” idea, the taxonomy author can create use=“required” definition arcs. These arcs would link the elements representing the required fields and an element representing the idea of the form itself.

- use = “optional” shall indicate that the link is a candidate for traversal. This is the default value of the use attribute.
- use=“prohibited” shall indicate that the link should not be traversed. Typically, links with this use value will only be written by third parties.

5.3.5.8.2 priority attribute

The content of the priority attribute is an integer.

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

The default value of the priority attribute is 0.

5.3.5.9 calculationArc

Calculation elements define elements that take part in summation.

Suggested
<pre><calculationArc xlink:type="arc" xlink:from="ci_currentAssets.prepaidExpenses" xlink:to="ci_assets.currentAssets" xlink:show="replace" xlink:actuate="onRequest" xlink:title="calculation: Go up to ci_assets.currentAssets"</pre>

```
xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent" weight="1"/>
```

Meaning: There is a summation relationship between current assets and prepaid expenses. Prepaid expenses contributes all of its value to current assets. The additional attributes of show, actuate and title tell XLink aware software how to navigate the linkbase and what to display as the title of the arc.

5.3.5.9.1 weight attribute

Indicates the multiplier to be applied to an item value when accumulating numeric values from child elements to parent elements. A value of "1.0" means that 100% of the numeric value of the item is applied to the parent item. A weight of "-1.0" means that 100% of the numeric value is subtracted from the parent item.

NOTE: If the xlink:arcrole is dimension-element, then the weight attribute is ignored. Instead, one of the values of the equivalent, sibling dimensions is assigned to the element. If the xlink:arcrole is element-dimension, then the weight attribute is ignored. Instead, the value of the element is assigned to each of the sibling dimensions.

5.3.5.10 presentationArc

The presentation element defines how elements relate to one another for presentation in typical parent-child hierarchy.

Suggested

```
<presentationArc xlink:type="arc"
xlink:from="ci_currentAssets.prepaidExpenses"
xlink:to="ci_assets.currentAssets" xlink:show="replace"
xlink:actuate="onRequest" xlink:title="calculation: Go up to
ci_assets.currentAssets"
xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent" order="4"/>
```

Meaning: There is a presentational relationship between current assets and prepaid expenses. Prepaid expenses is the fourth child of current assets. The additional attributes of show, actuate and title tell XLink aware software how to navigate the linkbase and what to display as the title of the arc.

5.3.5.10.1 order

A non-negative decimal number that indicates the presentation order of sibling elements. It defaults to "1". If multiple siblings have the same order attribute value, the presentation order of those siblings is application dependent. The value of the order attribute is not restricted to integers, which is useful when there is a need to place a new sibling element in between two previously defined sibling elements.

5.3.5.11 definitionArc

DefinitionArc elements define how elements relate to one another in an abstract, concept-to-concept sense.

Example

```
<definitionArc xlink:type="arc" xlink:from="ci_balanceSheet.assets"
xlink:to="ci_assets.currentAssets" xlink:show="replace"
xlink:actuate="onRequest" xlink:title="definition: Go down to
ci_assets.currentAssets" xlink:arcrole="
http://www.xbrl.org/linkprops/arc/parent-child"/>
```

Meaning: There is a definitional relationship between assets and current. The additional attributes of show, actuate and title tell XLink-aware software how to navigate the linkbase and what to display as the title of the arc.

5.3.5.12 labelArc, referenceArc

These arc-type elements join label and reference resources to elements defined in XML Schema documents. Because they do not link two elements, the arcoles given above are not relevant to these kinds of arcs.

5.3.6 Linkbase schema

In order to allow validation of linkbase documents, the following schema needs to be used with other schema that implement the XLink specification. These are provided with XBRL schemas and sample files.

NOTE: These schemas which implement the XLink specification are not official documents of the W3C. It is the intention of XBRL.org to integrate with the official schemas for XLink as they become available.

```
<schema targetNamespace="http://www.xbrl.org/2001/XLink/xbrllinkbase"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xl="http://www.xbrl.org/2001/XLink"
xmlns:self="http://www.xbrl.org/2001/XLink/xbrllinkbase"
xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <import namespace="http://www.xbrl.org/2001/XLink"
schemaLocation="xl.xsd"/>
  <import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="xml.xsd"/>
  <!-- useEnum -->
  <simpleType name="useEnum">
    <restriction base="string">
      <enumeration value="optional"/>
      <enumeration value="required"/>
      <enumeration value="prohibited"/>
    </restriction>
  </simpleType>
  <!-- xbrlArcType -->
  <complexType name="xbrlArcType">
    <complexContent>
      <extension base="xl:arcType">
        <attribute name="use" type="self:useEnum"/>
        <attribute name="priority" type="decimal"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- calculationLink -->
  <element name="calculationLink" type="xl:extendedType"
substitutionGroup="xl:extended">
</element>
  <!-- presentationLink -->
  <element name="presentationLink" type="xl:extendedType"
substitutionGroup="xl:extended"/>
  <!-- definitionLink -->
  <element name="definitionLink" type="xl:extendedType"
substitutionGroup="xl:extended"/>
  <!-- labelLink -->
  <element name="labelLink" type="xl:extendedType"
substitutionGroup="xl:extended"/>
  <!-- footnoteLink -->
  <element name="footnoteLink" type="xl:extendedType"
```

```

substitutionGroup="xl:extended"/>
<!--          referenceLink          -->
<element name="referenceLink" type="xl:extendedType"
substitutionGroup="xl:extended"/>
<!--          extendedLink          -->
<element name="extendedLink" type="xl:extendedType"
substitutionGroup="xl:extended"/>
<!--          calculationArc          -->
<element name="calculationArc" substitutionGroup="xl:arc">
  <complexType>
    <complexContent>
      <extension base="self:xbrlArcType">
        <attribute name="weight" type="decimal"
use="required"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<!--          presentationArc          -->
<element name="presentationArc" substitutionGroup="xl:arc">
  <complexType>
    <complexContent>
      <extension base="self:xbrlArcType">
        <attribute name="order" type="decimal"
use="required"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<!--          definitionArc          -->
<element name="definitionArc" type="self:xbrlArcType"
substitutionGroup="xl:arc"/>
<!--          labelArc          -->
<element name="labelArc" type="self:xbrlArcType"
substitutionGroup="xl:arc"/>
<!--          footnoteArc          -->
<element name="footnoteArc" type="self:xbrlArcType"
substitutionGroup="xl:arc"/>
<!--          referenceArc          -->
<element name="referenceArc" type="self:xbrlArcType"
substitutionGroup="xl:arc"/>
<!--          arc          -->
<element name="arc" type="xl:arcType" substitutionGroup="xl:arc"/>
<!--          loc          -->
<element name="loc" type="xl:locatorType"
substitutionGroup="xl:locator"/>
<!--          linkbase          -->
<element name="linkbase">
  <complexType>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="self:linkbaseRef" minOccurs="0"
maxOccurs="unbounded"/>
      <element ref="xl:extended" minOccurs="0"
maxOccurs="unbounded"/>
    </choice>
  </complexType>
</element>

```

```

<!--          linkbaseRef          -->
<element name="linkbaseRef">
  <complexType>
    <sequence>
      <element name="p" type="string" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute ref="xlink:type"/>
    <attribute ref="xlink:href"/>
    <attribute ref="xlink:role"/>
    <attribute ref="xlink:arcrole"/>
    <attribute ref="xlink:title"/>
    <attribute ref="xlink:show"/>
    <attribute ref="xlink:actuate"/>
  </complexType>
</element>
<!--          label          -->
<element name="label" substitutionGroup="xl:resource">
  <complexType>
    <complexContent mixed="true">
      <restriction base="xl:resourceType">
        <sequence>
          <any
namespace="http://www.w3.org/1999/xhtml" minOccurs="0"
maxOccurs="unbounded" processContents="skip"/>
        </sequence>
        <attribute ref="xml:lang"/>
      </restriction>
    </complexContent>
  </complexType>
</element>
<!--          footnote          -->
<element name="footnote" substitutionGroup="xl:resource">
  <complexType>
    <complexContent>
      <extension base="xl:resourceType">
        <attribute ref="xml:lang"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<!--          part          -->
<element name="part" type="string" abstract="true"/>
<!--          reference          -->
<element name="reference" substitutionGroup="xl:resource">
  <annotation>
    <documentation>
      This datatype defines the reference to authoritative literature
      that may appear for a financial concept.
      These are references to published documents, not online resources.
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="xl:resourceType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="self:part" minOccurs="0"

```

```
maxOccurs="unbounded" />
        </choice>
    </extension>
</complexContent>
</complexType>
</element>
</schema>
```

6 Semantics of Instance Documents

The semantics of instance documents and their contents are specified here only insofar as they impact the operation of software applications that use this specification. The primary topics in this regard are:

- Processing by consuming applications
- Validation
- The parent-child relationship

6.1 Processing by consuming applications

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

For example, to correctly produce a table of values with rows corresponding to an ordered set of types and columns representing different periods, it is necessary to dereference the appropriate `schemaLocation` attribute in order to find the `label` elements and the `order` attributes corresponding to each item type. This is similar to a relational database join, where the document instance contains an "item" table, some of whose columns (e.g. type) are used as foreign keys into a table representing the taxonomy.

Treatment of relative pathnames and caching of the taxonomy file is implementation-dependent. For example, if a document instance contains a relative URL as the location of a `schemaLocation` attribute, it is up to the consuming application to dereference it; it shall be an error if the underlying taxonomy cannot be found.

6.2 Validation

Validation of an instance document against the XBRL core schema is expected but not required of any consuming application. XML Schema validation of an instance document against all of the taxonomies to which it refers shall be required.

The content of text items is not highly constrained. Other markup (e.g., presentation related HTML tags for bold, italics, images) may occur inside of text items, if the proper redefinition of `itemType` occurs in the schema.

6.3 The Parent-Child relationship

There is no nesting of XBRL item elements. Whatever structural relationships as might be necessary in an XBRL document instance shall be captured in tuple elements. The intellectual structure – the relationship of financial concepts to each other in a variety of senses – is captured in the link structure of the taxonomies.

6.4 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.

7 Semantics of Taxonomies

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

At every point, whether it is in the definition of a reporting concept or in the representation of a particular fact, it is always clear, through the use of namespaces, who is responsible for the definition of a concept, a fact, an entity, or a measure.

8 References (Non-normative)

This is a partial list of key references.

[CANONICAL] Bosworth, A., A. Layman and M. Rys. Serializing Graphs of Data in XML. BizTalk.org Library, Microsoft Corporation, 1999.

[SCHEMA-0] World Wide Web Consortium. XML Schema Part 0: Primer.

[SCHEMA-1] World Wide Web Consortium. XML Schema Part 1: Structures.

[SCHEMA-2] World Wide Web Consortium. XML Schema Part 2: Datatypes.

9 Change Log

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 3^d, 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.