Extensible Business Reporting Language (XBRL) Specification

2000-07-31

Editors:

Walter Hamscher, eSprocket.com, formerly of PricewaterhouseCoopers LLP

David Vun Kannon, KPMG Consulting Inc.

XBRL.org

Specification Group

Chair:

David Vun Kannon, KPMG Consulting Inc.

This version:

http://www.xbrl.org/tr/2000-07-31/xbrl-2000-07-31.doc (in Word)

http://www.xbrl.org/tr/2000-07-31/xbrl-2000-07-31.pdf (in PDF)

with separate provision of the DTD and Schema described herein. All components, along with non-normative samples and certain auxiliary DTDs are available in a single Zip format archive.

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 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. Walter Hamscher, though no longer active in the shaping of the specification and this document, provided key insights and perspective.

Status of this document

Table of contents

1 Introduction

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 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.

1.1 Documentation Conventions

This document will eventually be produced using an [XML] DTD and an [XSLT] stylesheet.

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.
`<schema name="http://www.muzmo.com/XMLSchema/1.0/mySchema" >`
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: financial information preparers, intermediaries in the preparation and distribution process, and users of financial 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 financial information will generally have precedence over other needs when it is necessary to make specification design decisions that may be perceived as benefiting one community at the possible expense of another.

XBRL is intended to improve the financial statement product. It should only comply with, not change or set new, accounting standards. However, XBRL should facilitate possible changes in financial reporting over the long term.

XBRL will provide users with a standard format in which to *prepare* financial reports that can be subsequently presented in a variety of ways. XBRL will provide users with a standard format in which financial information can be *exchanged* between different software applications. XBRL will permit the automated, efficient and reliable *extraction* of financial 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 deprecate the use of bold, italics, and other stylistic techniques that may be used to distract from the true and fair presentation of financial results. Therefore, there is no functional requirement that XBRL documents need to support any particular text formatting conventions.

1.3 Relationship to Other Work

XBRL uses several World Wide Web consortium (w3c) recommendations, XML 1.0, XML Namespaces, and refers directly to XSL. It also relies extensively on the latest working draft of XML Schema.

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 includes financial reporting and contemplates extensive detail in the representation and use of accounting conventions, which distinguishes it from these other efforts. Also, the design of XBRL is deliberately drawn so as to allow the embedding of isolated XBRL items into other XML documents, which is key to future interoperability with other specifications.

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 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 Schema.
entity	A business entity, the subject of XBRL items. Where the XML/SGML concept of syntactic "entity" is meant, this will be pointed out.
group	Text containing a collection of items that concern one or more entities during one or more time periods.
period	An instant or a duration of time. In business reporting, financial numbers and other facts are reported "as of" an instant or for a period of a certain duration. Items that report on instants and durations are both common.
element	An XML element, but also a “fact” or piece of information described by this taxonomy. For example, the element with the name “nonCurrentAssets.propertyPlantAndEquipmentNet” is an element.
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. So would an HTML file that had various XBRL items embedded in it.
may	Conforming documents and consuming applications are permitted to but need not behave as described.
must	Conforming documents and consuming applications are required to behave as described; otherwise they are in error.
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 must not continue normal processing (*i.e.*, it must not continue to pass character data and information about the document's contents to the application in the normal way).
at user option	Conforming software may or must (depending on the modal verb in the sentence) behave as described; if it does, it must provide users a means to enable or disable the behavior described

described.

2 XBRL Framework

The main ideas in the XBRL conceptual framework are *items, groups, elements,* 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 context defined in such a way so that software applications can efficiently and reliably find, extract, and interpret relevant items in their appropriate context.

> **Tuples**. It is often the case that facts must be joined together to be understood. A tuple, like a row in a table, is a grouping of facts. For instance, the name, age and compensation of a director of a company must 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 that in fact can be interspersed among other text and elements in any XML document. 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. The intention of XBRL instance documents is just the transmission of some set of financial 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 financial 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.

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 corresponds exactly to 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, the concepts of "Accounts Receivable Trade, Gross", "Allowance for Doubtful Accounts", and "Accounts Receivable Trade, Net" are different parts of that particular taxonomy.

Although any given item can only refer to one taxonomy, within any given XML document any number of XBRL items can refer to any number of taxonomies, and taxonomies can be composed together to extend other taxonomies. Although the current release of the XBRL specification provides a particular taxonomy as an exemplar, any given XML document may refer to a taxonomy that defines additional terms and relationships.

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 Syntax of Instance Documents

The syntax chosen for XBRL instance documents corresponds to a recommended syntax for serializing graphs of data in XML [CANONICAL]. XBRL uses this canonical syntax to exploit the features of XML attributes, specifically their order independence, irredundancy, ability to accommodate enumerated types, and the ability to have default (#IMPLIED) values. In XBRL there are relatively few XML elements, but there is a rich set of attributes that are applicable to most elements.

The core syntax for statements is defined using an XML DTD. The elements defined there are the item, label, and group. The item and group elements have the same set of attributes, which are in some sense the more important part of the XBRL vocabulary. The set of attributes is defined as follows.

```
<!ENTITY % att_AttributeHolder  "
  id                  IDREF       #IMPLIED
  period              CDATA       #IMPLIED
  schemaLocation      CDATA       #IMPLIED
  scaleFactor         CDATA       #IMPLIED
  precision           CDATA       #IMPLIED
  type                CDATA       #IMPLIED
  unit                CDATA       #IMPLIED
  entity              CDATA       #IMPLIED
  decimalPattern      CDATA       #IMPLIED
  formatName          CDATA       #IMPLIED
">
```

Each attribute is described separately below.

3.1 id

This attribute is not required. The content must start with an alpha character. The id attribute can be used to attach a unique identifier to any element.

Examples:

```
C2424
```

3.2 period

Every item applies to a particular instant or duration. This attribute uses the ISO 8601 date representation. A duration is a pair of dates separated by a solidus (/). See http://www.iso.ch/markete/8601.pdf for authoritative definitions; see http://www.w3.org/TR/NOTE-datetime for a summary.

Example	Meaning
1999	The calendar year 1999
P1Y/1999-05-31	The full year ended 31 May 1999.
P3M/1999-05-31	The quarter ended 31 May 1999.
2000-04	The month of April 2000
1999-05-31	The day 31 May 1999
P3M/2000-03-25	The three months ended 25 March 2000.
2000-04-01/P91D	The thirteen weeks beginning 1 April 2000.
1999-12-29/2000-03-27	From December 29, 1999 through March 27, 2000.
2000-01/2000-03	The first three months (first quarter) of 2000.

Note: At http://www.w3.org/TR/xmlschema-2#timeDuration, XML Schema defines a timeDuration type. We have chosen not to follow their lead, but rather to retain a single concept of "period" and not to factor the period attribute into different attributes like "startDate" "endDate" and/or "duration" and thereby get (*e.g.*) duration="P13W" end="2000-04-01" to mean "the thirteen weeks ended 1 April 2000."

When dealing with financial information, all of the items can generally be categorized as "as of" or "point in time" measures, such as Assets, Cash, Liabilities; or are "for the period ended" measures of aggregate activity during a defined period, such as Revenues and Expenses.

Since any date by itself can be interpreted as a period (*e.g.*, "1999-04" refers to all the days in the entire month of April) this leads to some subtleties. If a financial report has its year end on April 30th 1999, a producing application that indicates only that the period is "1999" is conveying incorrect information to consuming applications; it should specify at least the period "P1Y/1999-04" which means "one year, whose end coincides with the end of April".

Robust consuming applications will *not* assume that the duration of an item is related to the date specified, in other words, just because period="2001-05" it does not mean that it is referring to a one-month period. Instead, an explicit duration should be expected when the semantics of the item warrant.

The period attribute is unique among XBRL item attributes in that it specifies a particular system of specifying dates, times and durations, rather than providing a framework (via Qnames) for specifying a system and a value within that system.

3.3 entity

An entity specifies a system for identifying business entities and a particular identifier within that system. A business entity does not have to be a full corporate entity; it could be a subsidiary, a division, even an individual: any reporting unit for which there is a financial statement. The entity is a QName so as to provide a framework for referencing naming authorities. It does not imply that the XBRL.org is a naming authority for business entities.

Example	Meaning
SAMP	Some entity known only as SAMP within the default namespace.
NASDAQ:SAMP	The company with NASDAQ ticker symbol SAMP.
DUNS:0236503276	The company or subsidiary with DUNS number 0236503276.
CUSIP:41009876AB	The entity with CUSIP number 41009876AB (e.g. a mutual fund).
URI:www.w3c.org	The non profit organization owning the URI www.w3c.org.

Each of the namespace prefixes in the above examples would have to be correctly declared previous to first use. XBRL makes no assumption about the ability of an application to resolve an identifier in any particular namespace that may appear as entity attribute content.

3.4 type

The type attribute provides the name of an element within a taxonomy. Its purpose is to specify the financial concept relevant to this particular measurement. A convention followed in the XBRL.org US GAAP C&I taxonomy is that the name of a type is is a dot-separated pair of camel-case identifiers representing a human readable name for the concept and its parent.

The reason for the "parent.child" naming convention is that within a taxonomy, it is necessary for an element name to be unique. A single name such as "NetIncome" is inadequate because it could appear at multiple points in a taxonomy. Adopting the (parent.child) naming convention helps, but still turns out to be no guarantee. An extreme solution, which was discarded early in the design, would be to use a number.

Example	Meaning
`ci:currentAssets.receivablesNet`	Receivables, in the “ci” namespace
`ci:identifiers.entityName`	Entity Name, etc.
`ci:notesToFinancialStatements.goingConcernNote`	A "Going Concern" note, etc.

Note that the type is a QName. The type attribute content must contain the correct namespace prefix, and all namespaces used in attributes must be declared in the document instance. However, it is still necessary to have a schemaLocation attribute to attach the namespaces to the right resources.

3.5 schemaLocation

The schemaLocation attribute is used to connect namespace URIs with actual, resolvable addresses for taxonomy resources.

Note that the way that namespaces are defined in XML, there is no guarantee that a URI with which a namespace is associated can be dereferenced to something useful. For example, the attribute content `xmlns:NASDAQ="http://www.nasdaq.com/XBRL/ticker"` does not imply that any such URI actually points to any service having to do with ticker symbol lookup available at the NASDAQ web site. XML Schema defines the schemaLocation attribute (http://www.w3.org/TR/xmlschema-1/#xsi:schemaLocation) that can be used in a document to provide hints as to the physical location of schema documents to be used for validation. There is further discussion of this in the XML Schema Primer [SCHEMA-0]. Because this is exactly the purpose intended for this attribute in XBRL, the same attribute name has been used. The semantics of the schemaLocation attribute in XBRL are exactly the same as those of the schemaLocation attribute in XSchema itself.

The content of the schemaLocation attribute is one or more pairs of space delimited strings. The first member of each pair is a namespace URI, the second is a resolvable address which points to a resource. A single schemaLocation attribute can contain several pairs, so it is possible to have only one instance of the schemaLocation attribute which will handle all of the namespaces.

Qualified names are used as the content of several attributes in the XBRL vocabulary. The schemaLocation attribute can be used to help resolve namespace resources for any of them. Namespaces used for XBRL taxonomies must resolve to valid XBRL taxonomies.

Example	Meaning

`schemaLocation=”`
`http://www.iasc.org/xbrl/airline/2000-07-07`
`http://www.iasc.org/xbrl/airline/2000-07-07-airline.xsd”`
	A taxonomy for the airline industry conforming to IASC guidelines.

`schemaLocation=”`
`http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31`
`http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd”`
	The XBRL.org’s US GAAP taxonomy for commercial and industrial companies.

`schemaLocation=”`
`http://www.xbrl.org/cica/canada/media/2000-06-02`
`http://www.xbrl.org/cica/canada/media/2000-06-02.xsd”`
	A Canadian Institute of Chartered Accountants' taxonomy for media companies.

By publishing a taxonomy structure for US GAAP, XBRL.org hopes to facilitate the analysis of data from many sources. However, creators of XBRL data may refer to other specific authoritative sources via the schemaLocation attribute. Business entities, governments, software vendors, standards bodies and auditors can all create taxonomic resources that are publicly referencable. The voluntary extension and refinement

of published taxonomies will allow for the flexibility in reporting concepts that most users of XBRL require, especially in the international arena.

3.6 unit

Unit specifies the standard which is relevant to the measurement. It is expected that most measurements will be monetary measurements. ISO 4217 standard currency designation is required for the units attribute in such a case. (http://www.iso.ch/cate/d23132.html) Pure numbers and counts of people, shares and the like can be specified as quantities. Enumerations (ENUM) depend on the taxonomy in force for the item's concept to specify the datatype of the element as an enumerated datatype, and to provide the allowable values.

Unit attribute content can be a Qname or QNames connected by * and /, and grouped with (). This allows for the creation of composite units such as ISO4217:USD/XBRL:shares, which could be used as the unit for an earnings per share (EPS) item.

Example	Meaning
ISO4217:GBP	Currency, UK Pounds.
SI:m2	Square meters
ISO:8601	Date, ISO format
US:ft2	Square feet
Moodys:rating	Credit rating (an enumeration)
USgaap:employees	Number of employees, a concept from a particular taxonomy

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

3.7 scaleFactor

An integer power of ten. If a scaleFactor value is not 0, the numeric value of the item content must have the proper multiplier applied to arrive at the actual value.

Example	Meaning
3	Thousands
6	Millions

3.8 precision

Precision is an integer intended to convey 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.

Examples:

Example	Meaning
9	Precision of nine decimal digits.

3.9 decimalPattern

decimalPattern is used to hold locale specific formatting for the item element content, precision, and scale attributes. It follows the usage of the XSLT Recommendation, Section 12.3 - Number Formatting. It

corresponds to the second argument of the XSLT number-format function. For more information see the source documents:

- http://java.sun.com/products/jdk/1.1/docs/api/java.text.DecimalFormat.html
- http://java.sun.com/docs/books/tutorial/i18n/format/decimalFormat.html

Example	Meaning
####.##	Typical US numbers, default treatment of negative with a leading minus sign.
#,###.##;(#,###.##)	Comma separator every three digits, negative numbers in parentheses and no minus sign.
#.###,##	Comma used as decimal separator. (when used in conjuction with a formatName attribute)

Inclusion of this and the following attribute in the specification is intended to allow for the use of XBRL in international settings. Although in the JDK mainly intended for output formatting, they have parsing implications as well. For example, in JDK 1.1.6 in order to parse numbers in a form such as "1,234.56" the decimal format "#,###.##" is needed; the decimal format "#.#" would incorrectly parse "1,234.56" as the integer "1".

The referenced XSLT Recommendation of the W3C itself refers to the JDK 1.1 specification for the details of constructing number formats. This reference to the JDK is not meant as requirement to use the JDK or Java in the implementation of applications that will use XBRL, rather, it merely references a widely available source of information.

3.10 formatName

formatName refers to an element from an XSLT namespace which is used to define a decimal format. It follows the usage of the XSLT Recommendation, Section 12.3 - Number Formatting. It corresponds to the third argument of the XSLT number-format function. If present, the document containing the item should also contain an decimal-format element from the XSLT namespace whose name matches the content of this attribute.

```
<xsl:decimal-format
  name = qname
  decimal-separator = char
  grouping-separator = char
  infinity = string
  minus-sign = char
  NaN = string
  percent = char
  per-mille = char
  zero-digit = char
  digit = char
  pattern-separator = char />
```

See http://www.w3.org/TR/xslt#format-number for more information.

Example	Meaning
<item type="cash" decimalPattern="99999v99" formatName="old:COBOLstyle">7</item> <xsl:decimal-format name="old:COBOLstyle" decimal-separator="v" digit="9" zero-digit="0"	A non-JDK standard decimalPattern with the necessary formatName attribute and XSLT element.

/>	

3.11 The item element

As discussed above, an *item* represents a single fact or business measurement. Although the content model of *item* allows parsed character data, the value is actually further restricted by the datatype given to the item type in the taxonomy. The latter constraint is not readily expressed using an XML DTD.

```
<!ELEMENT item (#PCDATA )>
<!ATTLIST item  %att_AttributeHolder; >
```

Example

```
<item entity="NASDAQ:SAMP" period="1998-12-31"
      xmlns="http://www.xbrl.org/core/2000-07-31/instance"
      xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-
31"
      xmlns:ISO4217=”http://www.ISO.org/4217”
      xmlns:NASDAQ=”http://www.NASDAQ.org/”
      schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-
2000-07-31 http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd“
      type="ci:capitalLeasedAssetsNet.capitalLeasedAssetsGross"
      unit="ISO4217:USD" scaleFactor="3" precision="3"
      decimalPattern="#,###" formatName="">727</item>
```

Meaning: The value of Gross Capital Leases at calendar year end 1998 for SAMP were $727,000.

Example

```
<item entity="NASDAQ:SAMP" period="1998-12-31"
      xmlns="http://www.xbrl.org/core/2000-07-31/instance"
      xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-
31"
      xmlns:NASDAQ=”http://www.NASDAQ.org/”
      schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-
2000-07-31 http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd"
      type="ci:notesToFinancialStatements.concentrationsNote">
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.
</item>
```

Meaning: The text of the Concentrations note at calendar year end 1998 for SAMP.

The item element is a leaf in the tree of XBRL elements within a given instance document. The content of an item cannot contain other markup, and in particular cannot contain other items.

As a side point, notice in these examples that the schemaLocation attribute does not contain URIs to help resolve the ISO4217 and NASDAQ namespaces. The examples assume that the applications which produced and will consume this instance will be able to resolve these namespace references without the help of the schemaLocation. (The URIs given are for example use only and do not reference actual resources of the ISO or NASDAQ.)

3.12 The label element

All elements within a given taxonomy should have a label assigned to them in one or more languages. The label element allows applications to override that label within an instance document.

```
<!ELEMENT label(#PCDATA)>
<!ATTLIST label
          href       CDATA       #IMPLIED
>
```

Note that the href attribute is CDATA, not IDREF. Producing applications should create documents that use Xpointer instead of IDREF. If the href content is not `href="xpointer(...)"` then a consuming application can try to interpret it as an IDREF to an item with an id attribute, but the DTD/XSL/DOM hooks will (probably) not work correctly, *e.g.*, it will not be possible to use the XPath id() function.

Note also that although the label element is legal in instance documents, it is really intended for use in taxonomy documents. Occurrence of label elements in the instance document is a last resort. If a company has a particular style of rendering a common accounting concept, that should be held in an extension taxonomy for that company. Labels in instance documents apply to that document only, which implies a very temporary usage.

Example

```
<group>
  <item type="ci:assets.currentAssets">657</item>
  <label
href="xpointer(//item[@type='ci:assets.currentAssets'])">Working Capital
Assets</label>
</group>
```

Meaning: Every instance of an item with type="ci:assets.currentAssets" gets the new label instead of the one in the taxonomy that would otherwise be providing it.

3.13 The group element

The group element is the generic container element of the XBRL vocabulary. It can also be used to aggregate items by attribute, so that attributes do not have to be given in full on each item. Items inherit the value of attributes from the closest parent with an explicit reference to the attribute value. The group element provides a convenient way to group similar items together, without forcing a particular hierarchy. Entity, period and type are all useful grouping attributes, and the specification allows each document to use them in whatever order is desired.

```
<!ELEMENT group ANY>
<!ATTLIST group
          %att_AttributeHolder;
>
```

Example

```
<group entity="SAMP"
       type="ci:propertyPlantandEquipmentNet.capitalLeasedAssetsNet"
       xmlns="http://www.xbrl.org/core/2000-07-31/instance"
       xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-
31"
       schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-
2000-07-31
        http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd"
       unit="ISO4217:USD" scaleFactor="3" precision="3"
       decimalPattern="#,###" formatName="">
 <item period="1998-12-31">727</item>
 <item period="1997-12-31">635</item>
 <item type="ci:notesToFinancialStatements.concentrationsNote" period="1998">
Concentration of credit risk with regard to short term investments is not
considered to be significant due to the Company's cash management policies.
 </item>
</group>
```

Meaning: Use of *group* with items varying by period and type.

Example

```
<group entity="SAMP"
       xmlns="http://www.xbrl.org/core/2000-07-31/instance"
       xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-
31"
       schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-
2000-07-31
        http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd"
       unit="ISO4217:USD" scaleFactor="3" precision="3"
       decimalPattern="#,###" formatName="">
 <group period="1998-12-31">
  <item
type="ci:propertyPlantandEquipmentNet.capitalLeasedAssetsNet">727</item>
  <item type="ci:notesToFinancialStatements.concentrationsNote">
Concentration of credit risk with regard to short term investments is not
considered to be significant due to the company's cash management policies.
  </item>
 </group>
 <group period="1997-12-31">
  <item
type="ci:propertyPlantandEquipmentNet.capitalLeasedAssetsNet">635</item>
 </group>
</group>
```

Meaning: Use of *group* with nesting by period, and items within the groups varying by type.

The significance of these two examples is that the "group" container element is designed to allow for more compact instance documents. It is *not* intended to convey presentation related information.

Applications should not produce instance documents with *group* elements expecting all consuming applications to respect those groupings, their type or other attributes if the only use of the group is to assign attribute values to all their contained elements. In such a case, the document can be flattened by pushing the attributes down into the items and eliminating the groups. Applications may find this useful in order to re-group later.

If the group element contains an attribute, and an item within that group contains the same attribute with a different value, then the group cannot be removed without loss of information. This situation can occur when groups are used to create the equivalent of table rows or tuples.

Example :

```
<group type="my:director" >
 <item type="my:director.directorName" >Velma Smith</item>
 <item type="my:director.directorAge" >55</item>
 <item type="my:director.directorSalary" >135000</item>
</group>
<group type="my:director" >
 <item type="my:director.directorName" >Tony Cruz</item>
 <item type="my:director.directorAge" >62</item>
 <item type="my:director.directorSalary" >142000</item>
</group>
```

Meaning: Use of *group* to create tuples of data.

3.14 Document Types

As a practical matter, many XBRL document instances will be structured in such a way that a group element is the outermost container. Consequently, the "document type" of an XBRL instance document will usually be "group". Non-namespace aware parsers will not validate properly given the use of namespace declarations, so generally a declaration such as the example below will be needed.

Example:

```
<!DOCTYPE group SYSTEM "http://www.xbrl.org/core/2000-07-31/xbrl-core-open-
2000-07-31.dtd"
[<!ATTLIST group
  xmlns:USGAAP  CDATA #IMPLIED
  xmlns:ISO4217 CDATA #IMPLIED
  xmlns:DUNS    CDATA #IMPLIED
>]>
```

Meaning: these are three namespaces that are used in other examples in this specification, declared internal to the document.

3.15 Additional attributes (Non-Normative)

The XBRL Core DTD allows the use of Dublin Core metadata attributes on all of the previous elements. Of particular interest is the keyword attribute. In principle, the keyword attribute might be used to include meta-information about items, such as "restated", "pro-forma", "budget", "actual", and "projected", as in

Example	Meaning
`<group dc:keyword="actual"/>`	Deprecated usage.

However, this usage is deprecated on the grounds that some distinctions, such as budget *vs*. actual, are common to many financial reporting situations and cross-application interoperability requires a common treatment while still satisfying internationalization requirements. Only lack of time has prevented this issue from being addressed in the current specification. It is not clear, for example, whether the introduction of contexts such as "budgeted" simultaneously introduces the need for a built-in versioning mechanism (*e.g.*, the budget as of 2000-02-01, the budget as of 2000-02-05) or whether versioning can be adequately dealt with by standards which would work generically for any XML document.

Hence, the working group expects a future revision of this specification to define one or more such attributes in the entity `%att_AttributeHolder` along with enumerated data types for their content. Implementers should suggest, and anticipate the inclusion of, additional context attributes. In the meantime, applications should use a form such as the following to extend the XBRL core definitions if necessary:

Example	Meaning

`<!ATTLIST item context`
`(actual | budget | projected | pro-forma | restated)`
`#IMPLIED>`
	Preferred usage.

with the understanding that the final specification will be informed but not constrained by early implementation.

3.16 #IMPLIED resolution

If an attribute (except ID) that is specified as #IMPLIED within the `att_AttributeHolder` XML entity is not present in an instance of an item, it must be available attached to a container element that is an ancestor (in the XPath sense) of the item element.

The XPath expression "`ancestor-or-self::*[@implied-attribute][1]/@implied-attribute`" finds the nearest value of an attribute, which may appear either attached to an element of higher up in the document tree.

The implication of this is that an XBRL item element is always fully specified in terms of all of its attributes, even if some of those attributes are not directly attached to the item element itself. The *id* attribute is the obvious exception; an item without an id does not inherit one from its containing parent.

There are no default values for any of the attributes that can appear on an item or group. This is a key requirement for full internationalization. Every XBRL instance document must specify, for all items, all relevant attributes—in particular, item types do *not* default to US dollars, US measurements, US number formatting conventions, US accounting principles, or US English for anything other than the names of elements and attributes.

3.17 Design Rationale (Non-normative)

Some of the features used in XBRL instance documents appear to be at odds with conventional definitions of XML document types. Order independence and the heavy use of attributes are relatively novel but offer crucial advantages to meet both defined current requirements and known future requirements.

3.17.1 Order independence

Although the ordering of financial information in presentation to a human is important, ordering is irrelevant insofar as the exchange of data between software applications is concerned. Therefore, in XBRL the ordering of *item* elements is unimportant and there is no document structure defined within the core specification. The main reason for this is that it greatly increases modularity. For one thing, it allows any XML document in the world that happens to describe financial information to include an XBRL item to describe it. Consider an HTML press release with an embedded XBRL item:

Example

```
<P>San Sushi, CA, 7 December 2002:  Sample Data Inc. today
announced net revenues of $<xbrl:item
xmlns:xbrl="http://www.xbrl.org/core/2000-07-31/instance"
xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31"
schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-
31
http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd"
type="ci:salesRevenueNet.salesRevenueGross"
period="2002-07-01/2002-10-01" entity="NASDAQ:SAMP"
precision="2" scaleFactor="6" decimalFormat="#.#"
units="ISO4217:USD">7.2</xbrl:item>m for the third quarter of 2002.</P>
```

XBRL embedded in HTML.

There are many reasons for wanting this embeddability feature; support for XML-aware search engines, and embedding XBRL items within the documents of other electronic commerce protocols are only two.

Order independence also simplifies the combination of financial information from different periods or entities, or even for the same entity under different reporting regimes, since in most cases an XBRL instance document can be created by concatenating other XBRL instance documents.

Finally, order independence makes it easier for individual reporting entities to define incremental new types, labels in different languages, *etc*. This level of extensibility is known to be a key requirement for meeting the reporting needs of a substantial number of business entities and has been an overriding consideration in the design of the language.

3.17.2 Use of Attributes

The interpretation of most items is embedded within attributes. Applications that process the information in an XBRL document, such as (say) rendering as HTML will process each item and perform a "lookup" into the appropriate taxonomy in order to extract properties such as its appropriate text label in a given human language, to determine the order in which it should be presented in a table relative to other items, *etc*. In principle, all of the attributes of the item element (except *id*) could have been done as an optional sub element. However, this would have sacrificed the ability to rely on the semantics of XML attributes, as well as making the document instance unnecessarily verbose.

The use of a general *group* construct to assign content to those attributes, means that pure XBRL documents resemble a database more than they resemble a presentation-oriented document. This is intentional, since future extensions of XBRL into the arena of internal reporting will in fact require XBRL to serve, in effect, as a neutral format for passing multidimensional data from one application to another.

4 Syntax of Taxonomies

Although only one XBRL taxonomy exists as of the release of this specification, there will be many. Each taxonomy consists of a list of new element definitions along with relations between these elements. The definition of a taxonomy is done using the XML XSchema vocabulary, extended with several elements that will be described here. The syntax of the XBRL extensions to XSchema is defined in a metamodel schema.

An XBRL taxonomy document is a valid instance of an XSchema document. Each taxonomy document must use the standard XSchema <import/> element to reference the XBRL metamodel.

Example

```
<schema targetNamespace="http://www.my.org/myTaxonomy"
      xmlns:xbrl="http://www.xbrl.org/core/2000-07-31/metamodel">
      <import namespace="http://www.xbrl.org/core/2000-07-31/metamodel"
            schemaLocation="http://www.xbrl.org/core/2000-07-31/xbrl-meta-
2000-07-31.xsd"/>
</schema>
```

Meaning: Defining the namespace alone would be sufficient for the use of XBRL elements within the <annotation> elements. The <import> allows XBRL datatypes to be used in the definition of elements.

A taxonomy can and should leverage other taxonomies as appropriate. Each publisher of a taxonomy is then responsible only for the concepts defined in their taxonomy.

Example

```
<schema targetNamespace="http://www.my.org/myTaxonomy"
      xmlns:xbrl="http://www.xbrl.org/core/2000-07-31/metamodel"
      xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31">
      <import namespace="http://www.xbrl.org/core/2000-07-31/metamodel"
            schemaLocation="http://www.xbrl.org/core/2000-07-31/xbrl-meta-
2000-07-31.xsd"/>
      <element name="goodwill.brandEquity" type="xbrl:monetary">
            <annotation><appinfo>
                  <xbrl:rollup to="ci:intangibles.goodwill"/>
            </appinfo></annotation>
      </element>
</schema>
```

Meaning: The brand equity concept is defined by my.org and related to the concept of goodwill defined by XBRL.org in the US GAAP Commercial and Industrial taxonomy.

4.1 The monetary and shares datatypes

The XBRL metamodel defines a datatype "monetary" that specializes the "decimal" type. Monetary strings are interpreted with respect the enclosing decimalPattern for any item where they appear. A taxonomy which includes numeric elements that are meant to be interpreted as monetary values should use this datatype rather than "string", which is the default.

The empty string "" is not a valid instance of the monetary datatype.

A negative number is a valid instance of the monetary datatype. Any item with datatype "monetary" can have a negative number as its value. The presentation of negative numbers (often in parentheses) is relatively rare in financial reporting; it is the responsibility of the producing application to ensure that the sign of the number indeed indicates a negative balance, *i.e.*, negative with respect to the normal balance for a given type.

If an instance item has type attribute content whose datatype is monetary, the unit attribute content of that instance item should be from the ISO 4217 currency designator namespace.

The XBRL metamodel defines a datatype "shares" that specializes the "decimal" type. Shares strings are interpreted with respect the enclosing decimalPattern for any item where they appear. A taxonomy which includes numeric elements that are meant to be interpreted as share values should use this datatype rather than "string", which is the default.

The empty string "" is not a valid instance of the shares datatype.

A negative number is a valid instance of the shares datatype.

4.2 element

An element has a name and data type. Because the US GAAP Commercial and Industrial Taxonomy contains nearly two thousand elements, a method was needed to prevent name clashes and this led to the convention of using names such as "marketableSecurities.availableForSale" containing both the colloquial name of the item as well as its immediate parent in the taxonomy. This convention is not a requirement of any taxonomy, although it is the case that all element names must be unique within a given taxonomy.

Examples

```
<element name="paymentOfDividends.preferredDividends"
         type="xbrl:monetary"/>
<element name="significantAccountingPoliciesNote.stockBasedCompensationPolicy"
         type="string"/>
```

Meaning: Typical element definitions, using both standard and XBRL datatypes.

The usage of the "element" element in an XBRL taxonomy is syntactically no different that its usage in XML Schema. For those familiar only with DTDs, these definitions are equivalent to use of a definition such as `<!ELEMENT paymentOfDividends.preferredDividends >`, with the additional power of Schema Constraints, in this case, data type constraints. This does *not* mean, however, that XBRL instance documents should be construed as containing forms such as `<paymentOfDividends.preferredDividends/>` as structural elements; they do not.

4.3 rollup

The rollup element defines how elements are related to one another in a parent-child relationship. The actual declaration within the XBRL metamodel defines a RollupType, with the rollup element being an element of that type.

```
<complexType name="RollupType">
  <attribute name="to" type="QName"/>
  <attribute name="weight" type="decimal" default="0"/>
  <attribute name="order" type="decimal" default="1"/>
</complexType>
```

There is one required attribute, *to* and two optional attributes, *weight*, and *order*.

Rollup elements are embedded within element definitions, using the standard XSchema constructs of <appinfo/> and <annotation/>.

```
<element name="assets.currentAssets" type="xbrl:monetary">
  <annotation>
    <appinfo>
      <xbrl:rollup to="balanceSheet.assets" weig ht="1" order="1"/>
    </appinfo>
  </annotation>
</element>
```

Meaning: The XBRL element <xbrl:rollup> appears in the <appinfo/> element, implying that it is intended to be used by applications.

4.3.1 to

A Qualified Name that indicates the parent element in the relation.

4.3.2 weight

Indicates the multiplier to be applied to an item value when accumulating numeric values upwards through the taxonomic hierarchy. 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.

4.3.3 order

A nonnegative decimal number indicating how sibling elements are normally ordered for presentation within their parent element. It defaults to "1". A consuming application is in principle free to ignore this attribute.

Examples

```
<!–from mandatorilyRedeemableSecurities.votingCharacteristics  -->
<xbrl:rollup
to="ci:liabilitiesandStockholdersEquity.mandatorilyRedeemableSecurities"
        weight=”0" order="3"/>
<!–from assets.currentAssets -->
<xbrl:rollup to="ci:balanceSheet.assets" weight=”1.0" order="1"/>
```

Meaning: Voting Characteristics is just a text string, and so the weight of any rollup from it is "0". Current assets, on the other hand, is monetary and therefore can be rolled up to (and from) with any weight.

4.4 label

One of the key internationalization features of XBRL is that although each taxonomy defines a single set of elements representing a coherent set of financial 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 set of financials could be presented by a single application in a language selected by the user (although recasting the underlying financials under a different set of national accounting principles is a far more complex matter).

Label elements appear inside <appinfo/> elements.

```
<complexType name="LabelType" content="textOnly">
  <attribute name="xml:lang" type="langu age"/>
</complexType>
```

Examples:

Example

```
<xbrl:label xml:lang="fr">Argent Comptant</label>
<xbrl:label xml:lang="en">Cash and Cash Equivalents</label>
```

Meaning : Consuming applications may display the item to speakers of English and French in the appropriate language.

Labels can be overridden in an instance document. The latter feature is important because individual companies routinely adjust the wording of an otherwise standard category, to reflect their particular circumstances.

4.5 reference

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

Reference elements appear inside <documentation/> elements.

```
   <complexType  name="ReferenceType ">
      <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>
      <attribute name="name" type="string"/>
      <attribute name="number" type="string"/>
      <attribute name="chapter" type="string"/>
      <attribute name="paragraph" type="string"/>
```

```
      <attribute name="subparagraph" type="string"/>
   </complexType>
      <element name="reference" type="self:ReferenceType"/>
```

Example

```
<xbrl: reference name="SAS" number="1" chapter="530" paragraph="3"/>
```

Meaning : Users can pursue the provided reference for more information about the concept.

4.5.1 name

A string, usually the acronym of a well known series of publications.

4.5.2 number

A string, usually a number but possibly of the form “2a”.

4.5.3 chapter

A string naming a top level section within the document.

4.5.4 paragraph

A string naming a particular span of text within a section.

4.5.5 subparagraph

A string naming a particular span of text within a paragraph.

4.6 Design Rationale (Non-normative)

There are several strengths displayed by the current design of XBRL taxonomies. First, it leverages existing standards, specifically, XML and those portions of XML Schema which are very unlikely to change.

Second, it supplies a commonly-used set of elements (in this case, Financial Reporting for Commercial and Industrial Companies, US GAAP) and therefore does not require every document instance to be designed "from scratch".

Third, it is fully extensible. Alternative approaches to extensibility were considered and failed to meet the requirements for complete independence from language, set of accounting principles, and document types. This required a novel use of XML Schema not merely as a replacement for DTDs, but as a kind of concept definition language.

5 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

5.1 Processing by consuming applications

While some consuming applications may be able to perform processing on an XBRL data file without referring to any of the 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, at a minimum 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 is an error if the underlying taxonomy cannot be found.

5.2 Validation

Validation of an instance document against the XBRL core DTD is expected but not required of any consuming application. Validation of an instance document against all of the taxonomies to which it refers by using a generalized validation is also possible, but this use of the word goes beyond the accepted syntactic sense of validation. Thorough validation of an instance document would, for example, require at least the following processing for each item:

- Retrieve the content of the item. Call this *itemContent.*
- Retrieve the value of the type attribute of the item, computing the value via inheritance from ancestors if necessary. Call this *itemType.*
- Retrieve the value of the schemaLocation attribute of the item, computing the value via inheritance from ancestors if necessary. Call this *itemTaxonomy.*
- Retrieve the contents of the elment in *itemTaxonomy* whose name attribute equals *itemType.* Call this *itemTypeElement.*
- Retrieve the content of the type attribute of *itemTypeElement.* Call this *itemDatatype.*
- Test that *itemContent* satisfies the *itemDatatype* schema constraint.

The following should pass validation as an XBRL document instance, and should be processable by any XBRL compliant application:

Example

```
<html><body>
<xbrl:group entity="SAMP"
       type="ci:propertyPlantandEquipmentNet.capitalLeasedAssetsNet"
       xmlns:xbrl="http://www.xbrl.org/core/2000-07-31/instance "
       xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31"
       schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31
http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd"
       unit="ISO4217:USD" scaleFactor="3" precision="3"
       decimalPattern="#,###" formatName="">
<h2>Capital Leases</h2>
<table><thead><tr>
<td width="30%"></td>
<td width="10%">1998</td>
<td width="10%">1997</td>
</tr></thead>
<tbody>
<tr><td>Net Capital Leases (in 000s)</td>
<td>$<xbrl:item period="1998-12-31">727</xbrl:item></td>
<td>$<xbrl:item period="1997-12-31">635</xbrl:item></td>
```

```
</tr>
</tbody>
</table>
<h2>Concentrations</h2>
<p>
<xbrl:item type="ci:notesToFinancialStatements.concentrationsNote"
          period="P1Y/1998-12-31">
Concentration of credit risk with regard to short term investments is
not considered to be significant due to the Company's cash management
policies.
</xbrl:item>
<p>
</xbrl:group>
</body></html>
```

which displays in a browser approximately as follows:

Capital Leases

	1998	1997
Net Capital Leases (in 000s)	$727	$635

Concentrations

Concentration of credit risk with regard to short term investments is not considered to be significant due to the Company's cash management policies.

The content of text items is highly constrained. No other markup (*e.g.*, presentation related HTML tags for bold, italics, images) may occur inside of text items, and in particular, occurrences of other XBRL elements within the content of an item element is an error. Any application claiming to be an XBRL validator should, at a minimum, detect the occurrence of XBRL elements inside the content of an XBRL item and signal an error.

This is not true of group elements; the content model of the group element allows for any embedded markup, including style information. Some consuming applications need styling information on their data. The problem is that not all producing applications can be relied upon to put it there. The compromise position that XBRL takes is that although items can't contain markup, the group element can. Therefore, the group element can be used to carry various kinds of presentation related information, leaving the item elements inside to remain easily accessible and parseable by software applications that only need access to the data, not the styling information.

5.3 The Parent-Child relationship

There is no nesting of XBRL items. Whatever structural relationships as might be desirable in an XBRL document instance are captured in rollups. Suppose a taxonomy contains the following rollups:

Example

```
<!—from “notestoFinancialStatements.businessCombinationsNote" -->
<xbrl:rollup to="ci:statements.notestoFinancialStatements" weight=”0”
        order="5"/>

<!—from "businessCombinationsNote.poolingofInterestsMethod"  -->
<xbrl:rollup to="ci:notestoFinancialStatements.businessCombinationsNote"
        weight=”0” order="1"/>

<!—from "businessCombinationsNote.purchaseAccounting"  -->
<xbrl:rollup to="ci:notestoFinancialStatements.businessCombinationsNote"
        weight=”0” order="2"/>
```

What this says is that a financial statement note concerning Business Combinations may include not only its own text, but may also include two different types of sections, one of which concerns pooling of interests and the other concerns purchase accounting. It may include any number of either type of subsection.

The XBRL instance document might contain the following, which shows a parent with two children of the same type:

Example

```
<item type="ci:notestoFinancialStatements.businessCombinationsNote">
During 1998, Sample Data Incorporated completed the acquisition of Small Fr y
Systems, a Delaware corporation.  Sample Data Incorporated also acquired
Exemplar Software Inc., also a Delaware corporation.
</item>
<item type="ci:businessCombinationsNote.purchaseAccounting">
The acquisition of Small Fry Systems for $8.7m, an excess o f $3.7m over its
book value of $5m, was accounted for with a charge to expenses for in -process
R&D of $2.5m and the remaining excess of purchase price over book value being
assigned to Goodwill.
</item>
<item type="ci:businessCombinationsNote.purchaseAccou nting">
The acquisition of Exemplar Software Inc. for $2.2m, an excess of $1.2m over
its book value of $1m, was accounted for with a charge to expenses for in -
process R&D of $1.1m and the remaining excess of purchase price over book value
being assigned to Goodwill.
</item>
```

(This example is only meant to illustrate a legal arrangement of items in a document instance, is not normative with respect to US GAAP or any other taxonomy, and is certainly not a realistic sample of the actual language of a financial statement note). Taxonomies are also free to include numeric items that roll up into non-numeric items, such as in the case of a tax reconciliation Note which has some numeric information that must then roll up into a note which is otherwise largely text.

5.4 Data Integrity and Confidentiality

There are many applications which require financial information to be transmitted securely, with a particular emphasis on data integrity (leading to the use of hash totals, etc., in financial data) and 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 financial 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.

6 Semantics of Taxonomies

Extensibility of taxonomies is a critical feature of XBRL. Taxonomies must be extensible to accommodate virtually any business entity's unique reporting requirements while maintaining some comparability across entities, or else XBRL will fail. These reporting requirements may be either external (the primary focus of this release) or internal (a goal for the near future). XBRL taxonomies may be constructed in such a way as to refer to other taxonomies.

The following short and financially unrealistic example demonstrates the relationships of multiple taxonomies to a single instance document. Two documents are included. The first is the instance document for a fictitious company, which is reporting two periods of data. The financial concepts reported on come mainly from the US GAAP Commercial and Industrial taxonomy, but one concept comes from the company's own taxonomy.

The second document is the taxonomy constructed by the company to define that financial concept. This taxonomy refers necessarily to the US GAAP taxonomy, also. For the sake of brevity, we do not present the US GAAP taxonomy itself.

Example balance_sheet.xml and brandEquity.xsd

```
<group
xmlns=”http://www.xbrl.org/core/2000-07-31/instance”
xmlns:ci="http://www.xbrl.org/us/gaap/ci/2000-07-31"
xmlns:FAB="http://www.fabulous.com/brandEquity"
xmlns:ISO4217=”http://www.iso.org/4217”
xmlns:DUNS=”http://www.Dun-and-Bradstreet.com/”
scale="6"
precision="9"
unit="ISO4217:USD"
entity="DUNS:1234567890"
decimalPattern=""
formatName=""
type="ci:statements.balanceSheet"
schemaLocation="http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31
http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.xsd
http://www.fabulous.com/brandEquity
http://www.fabulous.com/brandEquity.xsd">
      <group period="19981231">
            <item
type="ci:cashCashEquivalentsAndShortTermInvestments.cashAndCashEquivalents">18<
/item>
            <item type="ci:shortTermInvestments.marketableSecurities">2</item>
            <item type="ci:currentAssets.receivablesNet">14</item>
            <item type="ci:currentAssets.inventoriesNet">2</item>
            <item type="ci:currentAssets.prepaidExpenses">1</item>
            <item
type="ci:propertyPlantAndEquipmentNet.propertyPlantAndEquipmentGross">27</item>
            <item
type="ci:propertyPlantAndEquipmentNet.accumulatedDepreciationAndAmortization"> -
8</item>
            <item
type="ci:capitalLeasedAssetsNet.accumulatedAmortizationCapitalLeasedAssets"> -
13</item>
            <item type="ci:intangibleAssetsGross.goodwill">4</item>
            <item
type="FAB:intangibleAssetsGross.brandEquity">2</item>
            <item type="ci:noncurrentAssets.otherAssets">.5</item>
            <item type="ci:liabilities.currentLiabilities">7</item>
            <item type="ci:noncurrentLiabilities.longTermDebt">25</item>
            <item type="ci:stockholdersEquity.commonStock">10</item>
            <item
type="ci:stockholdersEquity.additionalPaidInCapital">7</item>
            <item type="ci:stockholdersEquity.preferredStock">.5</item>
      </group>
      <group period="19991231">
            <item
type="ci:cashCashEquivalentsAndShortTermInvestments.cashAndCashEquivalents">118
</item>
            <item type="ci:shortTermInvestments.marketableSecurities">2</item>
            <item type="ci:currentAssets.receivablesNet">14</item>
            <item type="ci:currentAssets.inventoriesNet">2</item>
            <item type="ci:currentAssets.prepaidExpenses">1</item>
            <item
type="ci:propertyPlantAndEquipmentNet.propertyPlantAndEquipmentGross">27</item>
            <item
type="ci:propertyPlantAndEquipmentNet.accumulatedDepreciationAndAmortization"> -
8</item>
            <item
type="ci:capitalLeasedAssetsNet.accumulatedAmortizationCapitalLeasedAssets"> -
13</item>
            <item type="ci:intangibleAssetsGross.goodwill">4</item>
            <item
```

```
 type="FAB:intangibleAssetsGross.brandEquity">2</item>
            <item type="ci:noncurrentAssets.otherAssets">.5</item>
            <item type="ci:liabilities.currentLiabilities">7</item>
            <item type="ci:noncurrentLiabilities.longTermDebt">125</item>
            <item type="ci:stockholdersEquity.commonStock">10</item>
            <item
type="ci:stockholdersEquity.additionalPaidInCapital">7</item>
            <item type="ci:stockholdersEquity.preferredStock">.5</item>
      </group>
      <item type="ci:significantAccountingPolicies.newAccountingStandardsNote">
      This balance sheet complies with SFAS 999....
      </item>
</group>


<schema xmlns:xbrl="http://www.xbrl.org/core/2000-07-31/metamodel"
xmlns:html="http://www.w3.org/1999/xhtml"
xmlns:ci="http://www.xbrl.org/us/2000-07-31/us-gaap-ci-2000-07-31"
targetNamespace="http://www.fabulous.com/brandEquity">

<import namespace="”http://www.xbrl.org/core/2000-07-31/metamodel”

schemaLocation="”http://www.xbrl.org/core/2000-07-31/xbrl-meta-2000-07-31.xsd”/>

      <element name="intangibleAssetsGross.brandEquity" type="xbrl:monetary">
                  <annotation><appinfo>

            <xbrl:rollup to="ci:intangibleAssetsNet.intangibleAssetsGross"
                  weight="1" order="4.5" />

            <xbrl:label xml:lang="en">Brand Equity</xbrl:label>

            </appinfo></annotation>

      </element>

</schema>
```

As the above example shows, the XBRL schemaLocation attribute used in instance documents has a different form than the schemaLocation attribute used in the XSchema <import/> element. This is an unfortunate ‘feature’ of XSchema itself. The XBRL schemaLocation is based on the XSchema *instance* form.

At every point, whether it is in the definition of a financial reporting concept or in the representation of a particular financial 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.

7 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.

8 Change Log

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

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

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

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

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

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

2000-03-29 [Hamscher] Miscellaneous typo corrections. Continuing repairs to text that concerns the fact that of 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.