ICS 03.100.60
A 11

中华人民共和国国家标准

GB/T 25500.1—2010

可扩展商业报告语言(XBRL)技术规范 第1部分:基础

Extensible Business Reporting Language (XBRL) specification—Part 1: Base

2010-10-18 发布　　2011-01-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　次

前言 …… Ⅲ
引言 …… Ⅳ
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 1
4　缩略语 …… 5
5　XBRL 处理器应满足的要求及其分级 …… 5
6　命名空间前缀惯例说明 …… 6
7　XBRL 架构 …… 6
7.1　XBRL 分类标准概述 …… 6
7.2　XBRL 实例概述 …… 6
7.3　数据完整性和安全性 …… 7
7.4　校验 …… 7
7.5　XBRL 中的 XLink …… 7
8　XBRL 实例 …… 25
8.1　概述 …… 25
8.2　根元素 xbrl …… 25
8.3　schemaRef 元素 …… 27
8.4　linkbaseRef 元素 …… 28
8.5　roleRef 元素 …… 29
8.6　arcroleRef 元素 …… 30
8.7　item 元素 …… 30
8.8　数据类型定义 …… 30
8.9　context 元素 …… 38
8.10　元组的应用 …… 47
8.11　等价性谓词表 …… 49
8.12　脚注 …… 55
9　XBRL 分类标准 …… 59
9.1　分类标准模式 …… 59
9.2　分类标准链接库 …… 74
附录 A（规范性附录）　模式文件 …… 96
A.1　xbrl-instance-2003-12-31.xsd …… 96
A.2　xbrl-linkbase-2003-12-31.xsd …… 109
A.3　xlink-2003-12-31.xsd …… 118
A.4　xl-2003-12-31.xsd …… 120

附录 B（资料性附录） 元素表 …… 125
附录 C（资料性附录） 属性表 …… 127
附录 D（资料性附录） 数据类型表 …… 128
参考文献 …… 131

表 1 命名空间前缀惯例对照表 …… 6
表 2 简单链接元素的属性表 …… 8
表 3 linkbase 元素的属性及子元素表 …… 9
表 4 roleRef 元素的属性表 …… 11
表 5 arcroleRef 元素的属性表 …… 12
表 6 扩展链接元素的属性表 …… 13
表 7 扩展链接中的标题元素的属性表 …… 15
表 8 定位器元素的属性表 …… 16
表 9 Resource 的属性表 …… 17
表 10 arc 元素的属性表 …… 18
表 11 禁止和覆盖关系的弧元素的属性表 …… 22
表 12 XBRL 实例的根元素 xbrl 的属性表 …… 25
表 13 模式引用元素的属性表 …… 27
表 14 linkbaseRef 元素的属性表 …… 28
表 15 linkbaseRef 元素中的角色 …… 29
表 16 数据项的属性表 …… 33
表 17 period 的子元素含义对照表 …… 39
表 18 子元素数据类型对照表 …… 40
表 19 entity 元素表 …… 40
表 20 基于数据项类型的单位约束 …… 45
表 21 等价谓词的解释 …… 50
表 22 footnoteArc 元素的属性表 …… 58
表 23 概念的属性表 …… 61
表 24 实例中的正确符号 …… 63
表 25 balance 属性和计算弧权值的限制 …… 63
表 26 已定义的数据项类型 …… 63
表 27 数据类型参照表 …… 67
表 28 roleType 元素的属性和子元素表 …… 69
表 29 arcroleType 元素的属性表 …… 71
表 30 cyclesAllowed 属性值注释表 …… 74
表 31 label 元素的属性表 …… 79
表 32 标准标签角色属性值 …… 80
表 33 参考角色属性值 …… 85
表 34 数据项含义表 …… 91

前　言

GB/T 25500《可扩展商业报告语言(XBRL)技术规范》分为四个部分：

——第1部分：基础；

——第2部分：维度；

——第3部分：公式；

——第4部分：版本。

本部分为GB/T 25500的第1部分。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分技术内容与XBRL国际组织制定的可扩展商业报告语言技术规范2.1版(Extensible Business Reporting Language Specification 2.1)(2008年7月2日的勘误修订版)基本一致。

本部分由中华人民共和国财政部归口。

本部分主要起草单位：中国证券监督管理委员会信息中心，中华人民共和国财政部，上海证券交易所，深圳证券交易所，中国科学院研究生院计算与通信工程学院。

本部分主要起草人：杨淑琴、应唯、白硕、邹胜、王颖、刘世平、初壮、黄敏、杨海峰、程立、郑斌祥、苏梅、俞志刚、赵国强、周晓明、赵伟、廖伟东、王开云、蒋小娜、梁炜、贾霄鹏、陈明哲。

引　言

可扩展商业报告语言(Extensible Business Reporting Language,XBRL)是一种基于可扩展置标语言(Extensible Markup Language,XML)的开放性业务报告技术标准。它通过给财务会计报告等业务报告中的数据增加特定标记、定义相互关系,使计算机能够"读懂"这些报告,并进行符合业务逻辑的处理。

XBRL的构想最早由美国注册会计师查尔斯·霍夫曼在1998年提出。随后,在美国注册会计师协会(AICPA)赞助下提出了第一个XBRL原型。XBRL技术广泛适用于财务会计报告、上市公司年报、金融机构监管报告、税务报告等领域,目前在美国、英国、日本、澳大利亚等很多国家中都已投入实际应用。在我国,XBRL已应用于上市公司信息披露和基金信息披露领域,取得良好效果。

XBRL技术的应用,可以避免报告数据的重复性录入、报送、传输、转换、比对等人工操作,减少差错率,提高数据生成、传递、使用效率和信息化水平。因此,推进XBRL在我国的应用,有利于促进财务会计报告等业务报告信息的深度分析利用,提高监管效能。XBRL技术规范,是各项XBRL应用所需共同遵循的底层技术标准。制定XBRL技术规范,是推进XBRL在我国应用的基础性工作。目前,国际上均遵循XBRL国际组织制定的技术规范。

本部分是XBRL技术规范的核心,规定了XBRL的技术架构,定义了XBRL分类标准和实例文档中使用的XML元素和属性。

可扩展商业报告语言(XBRL)技术规范 第1部分:基础

1 范围

GB/T 25500的本部分规定了XBRL的技术架构,定义了XBRL分类标准和实例文档中使用的XML元素和属性。

本部分适用于XBRL分类标准的制定、实例文档的编制或使用,以及XBRL相关开发与应用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 18793—2002 信息技术 可扩展置标语言(XML)1.0

3 术语和定义

下列术语和定义适用于本文件。

3.1

分类标准 taxonomy

XML模式文件和XBRL链接库的组合。其中,链接库可通过linkbaseRef元素被模式文件引用或者内嵌到模式文件中。

3.2

分类标准模式 taxonomy schema

定义XBRL概念语法的XML模式文件。

3.3

链接库 linkbase

若干个用于描述分类标准中概念的语义的XLink扩展链接的集合。

3.4

可发现分类标准集 Discoverable Taxonomy Set;DTS

通过引用形成的一组分类标准模式和链接库的集合。DTS的范围是可通过遍历分类标准模式和链接库中的链接或引用来发现的所有分类标准模式和链接库。DTS中至少有一个分类标准模式应引用xbrl-instance-2003-12-31.xsd模式文件。

3.5

事实 fact

具体业务内容,分简单事实simple fact与复合事实compound fact两种类型。复合事实是简单事实组合而成。简单事实用数据项来表达,而复合事实用元组表达。

3.6

元素 element

XML模式文件中定义的XML元素。

3.7

实体 entity

XBRL 数据项描述的业务主体。

注：当实体指代 XML 规范或标准通用标记语言(Standard Generalized Markup Language，SGML)规范中的语法实体概念时，将专门指出。

3.8

XBRL 实例 XBRL instance

以 xbrl 为根元素的 XML 片段。

注：XBRL 实例包含商业报告事实，其中每个事实都与 DTS 中已定义的概念相对应，也包含上下文和单位等额外信息来解释实例中的事实。XBRL 实例的根元素是“xbrl”元素。原则上，可在任何 XML 文档中嵌入 XBRL 实例，“xbrl”元素是 XBRL 实例的容器。

3.9

抽象元素 abstract element

对应业务概念在逻辑上没有具体值、不能在 XBRL 实例中出现的元素，其 abstract 属性值为 true。

注：列报链接库可利用抽象元素对概念进行分组展示。例如资产负债表“balanceSheet”元素仅在分类标准中定义，用于组织其他元素，不能出现在 XBRL 实例中。

3.10

具体元素 concrete element

对应业务概念在逻辑上有具体值、可在 XBRL 实例中出现的元素，其 abstract 属性值为 false。

3.11

上下文 context

XBRL 实例中根元素的子元素，用来说明实体、时期和场景。

注：上下文有助于准确地理解数据项的值。

3.12

数据项 item

XBRL 数据项替换组中的元素，包含简单事实的值、帮助理解该事实所需的上下文以及数值型数据项的单位。当数据项作为一个元组的子元素出现时，应结合该元组子元素中的其他数据项和元组来理解。数据项包括数值型数据项和非数值型数据项两种，对于数值型数据项，需要指出精度及度量单位。

3.13

元组 tuple

多个数据项捆绑在一起而形成的一个新的元素，用来表达一个复杂概念。多个元组也可以捆绑起来形成一个更复杂的元组。

注：XBRL 元组替换组中的元素，是一个有序的元素列表。

3.14

定位器 locator

定位分类标准模式中元素的 XML 指针，为扩展链接弧提供了锚。

3.15

弧 arc

通过定位器把不同的概念链接到一起的关系定义。弧可连接概念、资源和脚注。弧都有 xlink:arcrole 属性，用来定义弧关系的种类。

3.16

本名与别名 essence and alias

用于表达同一概念的一组名称，用弧来表示，弧的“from”端的概念为本名，“to”端的概念为别名。

注：别名概念的有效值同时对本名概念也是有效值。

3.17

XPATH 轴　XPATH Axis

所选节点与当前节点之间的路径关系。

注：应用 XPATH 中的轴来描述 XBRL 实例中元素的关系，对于任意元素 E，另外一个元素 F：

——当且仅当 F 出现在 E 的祖先轴上时，F 是 E 的祖先；

——当且仅当 F 出现在 E 的子轴上时，F 是 E 的子；

——当且仅当 F 出现在 E 的子孙轴上时，F 是 E 的子孙；

——当且仅当 F 出现在 E 的父轴上时，F 是 E 的父；

——当且仅当 F 是 E 的父元素的父元素时，F 是 E 的祖父；

——当且仅当 F 出现在 E 的父元素的子轴上，并且 F 不是 E 本身时，F 是 E 的兄弟；

——当且仅当 F 是 E 的父元素的兄弟时，F 是 E 的叔父。

3.18

父元素等价　p-equal

实例数据项或者元组有相同父元素。

3.19

结构等价　s-equal

XML 节点在 XML 值空间相同，或者它的 XBRL 相关子元素和属性结构等价。

3.20

上下文等价　c-equal

数据项、集合或数据项序列有相同的数据项类型，且其上下文结构等价。

3.21

单位等价　u-equal

数值型数据项有相同的度量单位。

3.22

值等价　v-equal

数据项上下文等价，并且对于非数值数据项，值相同；对于数值型数据项，在其精度、推导精度或者小数属性中较小精度所确定的容错范围内值相等。

3.23

路径等价　x-equal

XPATH 等价，XPath“=”操作符返回值为“true”。

3.24

概念　concept

关于商业活动或其本质可供报告的事实，在语法上被定义为 XML 模式文件中的元素。

3.25

标准弧元素　standard arc element

本文件已定义的，xl:arc 派生出来的元素。只包括以下五个类型：

——link:presentationArc；

——link:calculationArc；

——link:labelArc；

——link:referenceArc；

——link:definitionArc。

3.26

自定义弧元素　custom arc element

本文件未予定义的，从 xl:arc 派生的弧元素。因此自定义弧元素不包括以下五种类型：

——link:presentationArc；

——link:calculationArc;
——link:labelArc;
——link:referenceArc;
——link:definitionArc。

3.27

标准扩展链接元素 standard extended link element

从 xl:link 派生出来的、本部分定义的元素。

注：标准扩展链接元素包含如下类型：

——link:presentationLink;
——link:calculationLink;
——link:labelLink;
——link:referenceLink;
——link:definitionLink。

3.28

自定义扩展链接元素 custom extend link element

从 xl:link 派生出来的、本部分未预定义的扩展链接元素。

注：自定义扩展链接元素不包括以下类型：

——link:presentationLink;
——link:calculationLink;
——link:labelLink;
——link:referenceLink;
——link:definitionLink。

3.29

标准资源元素 standard resource element

从 xl:resource 派生出来的、本部分定义的元素。

注：标准资源元素包含如下类型：

——link:label;
——link:reference;
——link:footnote。

3.30

自定义资源元素 custom resource element

从 xl:resource 派生出来的、本部分未预定义的资源元素。

注：自定义资源元素不包括：

——link:label;
——link:reference;
——link:footnote。

3.31

重复数据项 duplicate items

具有相同的父元素、相同的上下文和相同概念的两个数据项。

3.32

重复元组 duplicate tuples

所有子孙元素都有相同内容的两个元组。

3.33

扩展链接 extended link

利用 XML 链接语言语法定义的、描述其所含信息与第三方文档信息之间关系的元素。

3.34

最小公共祖先 least common ancestor

两个元素的共同祖先元素，并且该祖先元素的所有子元素都不再是这两个元素的共同祖先元素。

3.35

数值型数据项 numeric item

反映数值的数据项。其简单类型可为 decimal、float、double，复杂类型为 fractionItemType。

3.36

非数值型数据项 non-numeric item

除数值型数据项以外的数据项。

注：日期类型不属于数值型。

3.37

期间 period

一个时间点或者一个时间段。

注：在商业报告中，财务数据和其他事实通常是以时间点或者以某个特定时间段来报告的。

3.38

资源 resource

概念或数据项的额外信息的扩展链接中所包含的 XML 片段。

3.39

单位 unit

在 XBRL 实例中作为根元素的子元素出现的、以说明数值型数据项度量单位的 XML 片段。一个单位元素只能用于说明一个特定的度量单位。

4 缩略语

下列缩略语适用于本文件。

DTS	可发现分类标准集	Discoverable Taxonomy Set
GAAP	公认会计原则	Generally Accepted Accounting Principles
ISO	国际标准化组织	International Organization for Standardization
URI	通用资源标识符	Universal Resource Identifier
URL	统一资源定位符	Uniform Resource Locator
XBRL	可扩展商业报告语言	eXtensible Business Reporting Language
XLink	XML 链接语言	XML Linking Language
XML	可扩展置标语言	eXtensible Markup Language
Xpath	XML 路径语言	XML Path Language
XPointer	XML 指针语言	XML Pointer Language
XQuery	XML 查询语言	XML Query Language

5 XBRL 处理器应满足的要求及其分级

本部分对 XBRL 处理器设定了最低符合要求和完全符合要求，其中：

——最低符合要求：XBRL 处理器应完整并且正确无误地符合本部分中的全部语法要求；

——完全符合要求：取决于处理器的使用目的，应首先达到最低要求，另外，它应完整并且正确无误地符合与链接库和 XBRL 实例相关的全部语义要求。

除非特别指出，本部分中的全部要求适用于所有满足最低要求的处理器。

6 命名空间前缀惯例说明

本部分在描述元素和属性时采用一系列命名空间前缀，命名空间前缀惯例应用如表 1 所示。

表 1 命名空间前缀惯例对照表

命名空间前缀	命名空间 URI
link	http://www.xbrl.org/2003/linkbase
xbrli	http://www.xbrl.org/2003/instance
xl	http://www.xbrl.org/2003/XLink
xlink	http://www.w3.org/1999/xlink
xml	http://www.w3.org/XML/1998/namespace
xsi	http://www.w3.org/2001/XMLSchema-instance
xsd	http://www.w3.org/2001/XMLSchema

本部分中的某些元素和属性的定义没有使用命名空间前缀或命名空间。本部分中定义的所有元素和属性的规范化命名空间是由附录 A 中规范模式文件决定的。本部分以 GB/T 18793—2002 为基础。

7 XBRL 架构

7.1 XBRL 分类标准概述

XBRL 分类标准由一个 XML 模式以及该模式内包含的或其直接引用的全部链接库组成。该 XML 模式被称作分类标准模式。

在 XBRL 术语中，概念是指对一个报告术语的定义，概念以 XML 模式元素定义的形式出现。分类标准模式应当为概念设置具体的名称和类型。根据概念定义，用类型说明所度量的事实可采用的数据类型。

链接库通过表达概念之间的关系（概念间关系）以及概念与其文档说明间的关系说明概念的含义。

链接库是指扩展链接的集合。在 XBRL 分类标准中有五种扩展链接用以说明定义、计算、展示、标签和引用这几个概念。其中，前三种类型表达概念间的关系，后两种则是用来表达概念与其文档说明间的关系。

链接库可嵌入在分类标准模式中；也可包含在分类标准模式之外的一个单独的文档中，此时分类标准模式应包括一个 linkbaseRef（参见附录 B）来指向该链接库文档。

7.2 XBRL 实例概述

XBRL 分类标准定义了报告概念，但并不包括以这些概念为基础的事实的实际值。事实的值包含在 XBRL 实例中，被称为“事实”。除了事实的实际值外，XBRL 实例还提供必要的上下文信息来解释该实际值。

一个 XBRL 实例可由一个或多个分类标准来支持，分类标准之间也能以不同的方式相互链接、扩展和修改。通常，在解释 XBRL 实例时需要综合考虑多个相关的分类标准。这种相互关联的分类标准的集合被称为可发现分类标准集，简称 DTS。确定 DTS 的范围时，应先从某个文档集（实例、分类标准

模式或者链接库)开始并遵循 DTS 发现规则。DTS 是分类标准模式和链接库的集合。

DTS 发现规则:

DTS 中包含的分类标准模式如下:

a) XBRL 实例通过 schemaRef(模式引用)、roleRef(角色引用)、arcroleRef(弧角色引用)或者 linkbaseRef(链接库引用)等元素直接引用的分类标准模式。schemaRef、roleRef、arcroleRef 或 linkbaseRef 元素的 xlink:href(参见附录 C)属性值包含了被发现的分类标准模式的 URL。

b) 在一个已发现的分类标准模式中通过 XML 模式的 import(导入)或者 include(包括)元素引用的分类标准模式。

c) 在一个已发现的链接库文档中通过 loc(定位)元素引用的分类标准模式。在已发现的链接库中,每个被定位元素上的 xlink:href 属性引用的分类标准模式应被发现。

d) 在一个已发现的链接库文档中通过角色引用元素引用的分类标准模式。在已发现的链接库中,每个被角色引用元素上的 xlink:href 属性引用的分类标准模式应被发现。

e) 在一个已发现的链接库文档中通过弧角色引用元素引用的分类标准模式。在已发现的链接库中,每个被弧角色引用元素上的 xlink:href 属性引用的分类标准模式应被发现。

f) 在一个已发现的分类标准模式中通过链接库引用元素引用的分类标准模式。在已发现的分类标准模式中,每个被链接库引用元素上的 xlink:href 属性引用的链接库文档所涉及的分类标准模式应被发现。

DTS 中包含的链接库如下:

a) XBRL 实例通过 linkbaseRef 元素直接引用的链接库,其 xlink:href 属性包括被发现的链接库的 URL。

b) 在一个已发现的分类标准模式中通过 linkbaseRef 元素引用的链接库。其 xlink:href 属性包括被发现的链接库的 URL。

c) 在已发现的分类标准模式中,通过 XPath 路径"//xsd:schema/xsd:annotation/xsd:appinfo/*"识别的节点集中的链接库[本部分中 schema(模式),annotation(注释)和 appinfo(应用程序信息)都是在 XML 模式命名空间中定义的元素]。

d) 在一个已发现的链接库文档中通过定位"loc"元素引用的链接库。已发现的链接库通过 loc 元素的 xlink:href 属性引用资源,每个包含这种资源的链接库应被发现。

7.3 数据完整性和安全性

XBRL 技术规范不提供传输加密、身份认证等信息安全机制。

7.4 校验

XBRL 实例、XBRL 链接库以及 XBRL 分类标准模式都应遵守本部分中的语法要求。其中很多语法要求都是通过 XML 模式表示的,所以校验程序的部分工作可利用 XML 模式的校验软件完成。语法要求中还有一些不是用 XML 模式或不能用 XML 模式表达,则应使用其他校验技术进行处理。

也可对 XBRL 实例中的数据与支持该实例的 DTS 中表达的语义是否相一致进行检查,而语义矛盾并不能使 XBRL 实例无法通过校验。本部分已指明那些通过完全符合要求的 XBRL 处理器能够识别出的语义矛盾。

7.5 XBRL 中的 XLink

7.5.1 概述

XML 片段之间的链接在 XBRL 中能以多种形式出现。XBRL 采用了 XLink 规范中定义的简单链

接和扩展链接来描述各种链接，主要包括：XBRL 实例和相应的 DTS 之间的链接，XBRL 实例中事实和描述这些事实的脚注之间的链接，以及链接库中概念语法定义和语义之间的链接。

XLink 规范确立了在 XLink 命名空间 http://www.w3.org/1999/xlink 中的一组属性的语法和语义。这些属性可被其他命名空间定义的元素使用，以记录 XML 片段间的各种不同种类的链接。这些属性在 XBRL 中被广泛地应用。其中许多属性都在 XBRL 中得以广泛运用，其他属性不含与 XBRL 中定义的链接相关的语义。这些其他的属性在 XML schema 语法要求中是被允许的，但是本部分中并未予以记录或设置特定的语义，例如 xlink:show 属性和 xlink:actuate 属性。

XBRL 中使用的 XLINK 中一般结构的语法被 xlink-2003-12-31.xsd 和 xl-2003-12-31.xsd 这两个 XML 模式所约束：前者定义了 XLINK 中属性的语法，后者定义了本部分中各种与链接相关的元素的内容模型。

7.5.2 简单链接

简单链接是指由一个资源指向另外一个资源的链接，其属性如表 2 所示。

表 2 简单链接元素的属性表

属性名称	选择性	描　述
xlink:type	必备	应有固定的属性值“simple”
xlink:href	必备	此属性应是一个 URI。URI 应指向一个 XML 文档或者 XML 文档中的一个 XML 片段。相对 URI 在使用前应解析为绝对 URI
xlink:role	可选	应具有 URI 值。如果提供了该属性，该属性一定不能为空
xlink:arcrole	可选	如果存在 xlink:arcrole 属性，该属性不能为空字符串
xml:base	可选	xml:base 属性可出现在简单链接中，用于解析 xlink:href 属性指定的相对 URI

XML 模式对于 XBRL 中的简单链接的约束如下所示：

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

  <complexType name = "simpleType">
    <annotation>
      <documentation>
      XBRL 中定义的简单链接的类型
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <attributeGroup ref = "xlink:simpleType"/>
        <attribute ref = "xlink:href" use = "required" />
        <attribute ref = "xlink:arcrole" use = "optional" />
        <attribute ref = "xlink:role" use = "optional" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute ref = "xlink:show" use = "optional" />
        <attribute ref = "xlink:actuate" use = "optional" />
```

```
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
          processContents = "lax" /〉
      〈/restriction〉
    〈/complexContent〉
  〈/complexType〉
  〈element name = "simple" type = "xl:simpleType" abstract = "true"〉
    〈annotation〉
      〈documentation〉
      抽象元素作为简单链接替换组中的头元素。
      〈/documentation〉
    〈/annotation〉
  〈/element〉

〈/schema〉
```

7.5.3 linkbase 元素

7.5.3.1 概述

按照 XLink 规范，包含一系列内部链接和第三方链接的文档被称为链接库。链接库是扩展链接或包含扩展链接的元素，也可包含 documentation 元素。linkbase 元素可作为扩展链接的容器，其属性及子元素如表 3 所示。

表 3 linkbase 元素的属性及子元素表

属性/子元素名称	选择性	描　述
id 属性	可选	链接库元素可包含一个 id 属性。此属性的值应遵循 XML 规范中对 id 类型(http://www.w3.org/TR/REC-xml#NT-TokenizedType)的规定
xml:base 属性	可选	xml:base 属性 XML 规范可出现在 linkbase 元素中，用于将扩展链接的相对 URI 解析为绝对 URI
documentation 元素	可选	所有 linkbase 元素中也可包括 documentation 元素。documentation 元素应含有字符串内容。documentation 元素可包含 XBRL 链接库命名空间(http://www.xbrl.org/2003/linkbase)中没有定义的某种属性

XML 模式对 linkbase 元素的约束如下所示：

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

  〈element name = "linkbase"〉
    〈annotation〉
      〈documentation〉
      linkbase 元素的定义——用于包含一组零个或多个扩展链接元素
      〈/documentation〉
    〈/annotation〉
```

```
    〈complexType〉
      〈choice minOccurs = "0" maxOccurs = "unbounded"〉
        〈element ref = "link:documentation"/〉
        〈element ref = "link:roleRef"/〉
        〈element ref = "link:arcroleRef"/〉
        〈element ref = "xl:extended"/〉
      〈/choice〉
      〈attribute name = "id" type = "ID" use = "optional"/〉
      〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
        processContents = "lax" /〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

示例:链接库框架

```
〈linkbase
   xmlns = "http://www.xbrl.org/2003/linkbase"
   xmlns:samp = "http://www.xbrl.org/sample"
   xmlns:xlink = "http://www.w3.org/1999/xlink"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation = "http://www.xbrl.org/sample samp001.xsd"
   xml:base = "http://www.xbrl.org/sample"〉

〈calculationLink
     xlink:role = "http://www.xbrl.org/2003/role/link"
     xlink:type = "extended"〉
     〈! --……-->
  〈/calculationLink〉

〈/linkbase〉
```

含义:将 linkbase 作为根元素,其中包括命名空间前缀定义和 schemaLocation 属性。"xml:"前缀不需要声明。calculationLink作为一个扩展链接元素包含在该链接库中。

XML 模式对 documentation 元素的约束如下所示:

```
〈schema targetNamespace = "http://www.xbrl.org/2003/XLink"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xl = "http://www.xbrl.org/2003/XLink"
  xmlns = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified"
  attributeFormDefault = "unqualified"〉

  〈complexType name = "documentationType"〉
    〈annotation〉
      〈documentation〉
      链接库和扩展链接文件中使用的元素类型。
      〈/documentation〉
    〈/annotation〉
    〈simpleContent〉
      〈extension base = "string"〉
        〈anyAttribute namespace = "##other" processContents = "lax"/〉
```

```
      </extension>
    </simpleContent>
  </complexType>

  <element name = "documentation" type = "xl:documentationType" abstract = "true">
    <annotation>
      <documentation>
      链接库和扩展链接文件中使用的抽象元素。
      </documentation>
    </annotation>
  </element>

</schema>

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

  <element name = "documentation"
    type = "xl:documentationType" substitutionGroup = "xl:documentation">
    <annotation>
      <documentation>
      链接库和扩展链接文件中使用的具体元素。
      </documentation>
    </annotation>
  </element>

</schema>
```

7.5.3.2 roleRef 元素

roleRef 元素用于解析在链接库或 XBRL 实例中用到的自定义的 xlink:role 的值。roleRef 元素是一个简单链接，指向分类标准模式文档中的 roleType 元素，roleType 元素声明了 xlink:role 的属性值，其属性如表 4 所示。

表 4 roleRef 元素的属性表

属性名称	选择性	描　述
xlink:type	必备	其值应为“simple”
xlink:href	必备	xlink:href 属性应是 URI。URI 应指向一个分类标准模式文档中的 roleType 元素。相对 URI 在使用前应解析为绝对 URI。无论 roleRef 在哪里出现，其引用的文档应作为 DTS 的一部分被发现
xlink:arcrole	可选	该属性在 roleRef 元素中没有语义定义
xlink:role	可选	可选属性 xlink:role 应为 URI 值。如果选用，则该属性不能为空。该属性在 roleRef 元素中没有语义定义
roleURI	必备	roleURI 属性用于确定 roleRef 元素所指向的 XML 资源中定义的 xlink:role 的属性值。此属性的值应和 roleRef 元素所指向的 roleType 元素中的 roleURI 属性值相匹配。在同一个链接库或者 XBRL 实例中，有相同 roleURI 属性值的 roleRef 元素最多只能有一个

XML 模式对 roleRef 元素的约束如下所示：

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

  <element name = "roleRef" substitutionGroup = "xl:simple">
    <annotation>
      <documentation>
      roleRef 元素定义——用于链接 xlink :role 属性解析值与 roleType 元素的声明
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base = "xl:simpleType">
          <attribute name = "roleURI" type = "xlink:nonEmptyURI" use = "required">
            <annotation>
              <documentation>
                该属性包含了一个角色名称。
              </documentation>
            </annotation>
          </attribute>
        </extension>
      </complexContent>
    </complexType>
  </element>

</schema>
```

7.5.3.3 arcroleRef 元素

arcroleRef 元素用于解析在链接库或 XBRL 实例中用到的自定义的 xlink:arcrole 的值。arcroleRef 元素是一个简单链接，指向声明了 xlink:arcrole 属性值的分类标准模式文档中的 arcroleType 元素，其属性如表 5 所示。

表 5 arcroleRef 元素的属性表

属性名称	选择性	描　述
xlink:type	必备	其值应为“simple”
xlink:href	必备	xlink:href 属性应是 URI。URI 应指向一个分类标准模式文档中的 arcroleType 元素。相对 URI 在使用前应解析为绝对 URI。所有被 xlink:href 属性引用的文档应作为 DTS 的一部分被发现，而不考虑 arcroleRef 元素出现在哪个链接库中
xlink:arcrole	可选	该属性在 arcroleRef 元素中没有语义定义
xlink:role	可选	应为 URI 值。如果选用，则该属性不能为空。该属性在 arcroleRef 元素中没有语义定义
arcroleURI	必备	arcroleURI 属性用于确定 arcroleRef 元素所指向的 XML 资源中定义的 xlink:arcrole 的属性值。此属性的值应和 arcroleRef 元素所指向的 arcroleType 元素中的 arcroleURI 属性值相匹配。在一个链接库或者一个 XBRL 实例中，有相同 arcroleURI 属性值的 arcroleRef 元素最多一个

XML 模式对 arcroleRef 元素的定义如下所示：

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

  〈element name = "arcroleRef" substitutionGroup = "xl:simple"〉
    〈annotation〉
      〈documentation〉
      arcroleRef 元素的定义：用于链接 xlink:arcrole 属性的解析值以及 arcroleType 元素的声明
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈complexContent〉
        〈extension base = "xl:simpleType"〉
          〈attribute name = "arcroleURI" type = "xlink:nonEmptyURI" use = "required"〉
            〈annotation〉
              〈documentation〉
                该属性包含了一个 arcrole 名称
              〈/documentation〉
            〈/annotation〉
          〈/attribute〉
        〈/extension〉
      〈/complexContent〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

7.5.4 扩展链接的使用

7.5.4.1 概述

扩展链接是有注解的 XML 片段。XBRL 扩展链接说明了以 XML 片段形式表示的资源间的关系，其属性如表 6 所示。

表 6 扩展链接元素的属性表

属性名称	选择性	描　述
id	可选	扩展链接可有一个 id 属性。id 属性的值应和 XML 规范中 ID 类型中属性的规则相一致。id 属性标识为一个扩展链接，以便于它可被简单链接直接引用
xlink:type	必备	其值应是“extended”
xlink:role	必备	xlink:role 属性应出现在标准扩展链接上。xlink:role 属性的内容被称为扩展链接的角色值。扩展链接角色值在应用中用来区分不同的关系。本部分定义的标准扩展链接角色遵循 http://www.xbrl.org/2003/role/link。 标准扩展链接可在使用这种角色时，无需使用 roleType 和 roleRef
xml:base	可选	用于处理扩展链接的相对 URI

XML 模式对 XBRL 使用的扩展链接的一般约束如下所示：

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

  <complexType name = "extendedType">
    <annotation>
      <documentation>
      一般扩展链接类型
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <choice minOccurs = "0" maxOccurs = "unbounded">
          <element ref = "xl:title" />
          <element ref = "xl:documentation" />
          <element ref = "xl:locator" />
          <element ref = "xl:arc" />
          <element ref = "xl:resource" />
        </choice>
        <attributeGroup ref = "xlink:extendedType"/>
        <attribute ref = "xlink:role" use = "required" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute name = "id" type = "ID" use = "optional" />
        <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
          processContents = "lax" />
      </restriction>
    </complexContent>
  </complexType>
  <element name = "extended" type = "xl:extendedType" abstract = "true">
    <annotation>
      <documentation>
      抽象扩展链接元素作为扩展链接替换组中的头元素。
      </documentation>
    </annotation>
  </element>

</schema>
```

XBRL 扩展链接可包括 5 种不同类型的子元素：documentation（备注）元素、title（标题）元素、locator（定位器）元素、resource（资源）元素和 arc（弧）元素。

documentation 元素仅用于 XBRL 注释，没有 Xlink 特定的语义。title，locator，resource 以及 arc 元素可由特定的 Xlink 属性确定。如果 title，locator，resource 和 arc 元素不是一个扩展链接元素的直接子元素，那么它们在 XLink 规范中就没有特定的意义，在 XBRL 中也就没有特定的意义。

7.5.4.2 documentation 元素

所有 XBRL 扩展链接都可包含 documentation 元素，其与 linkbase 元素中使用的 documentation 元素遵循相同的语法规则。

7.5.4.3 title 元素

所有的 XBRL 扩展链接都可包含标题。相对于 xlink:title 属性的有限表达能力，title 元素可用于说明扩展链接。在需要用多种语言提供信息时，尤其需要使用 title 元素。标题没有 XBRL 特定语义。若要在扩展链接中使用一个标题，就应为抽象的标题元素在替换组中定义一个新的元素，其属性如表 7 所示。

表 7 扩展链接中的标题元素的属性表

属性名称	选择性	描 述
xlink:type	必备	其值应为“title”

XML 模式对 title 元素的约束如下所示：

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

  <complexType name = "titleType">
    <annotation>
      <documentation>
      抽象 title 元素的类型· ···用作一个 title 元素模板。
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <attributeGroup ref = "xlink:titleType"/>
      </restriction>
    </complexContent>
  </complexType>
  <element name = "title" type = "xl:titleType" abstract = "true">
    <annotation>
      <documentation>
      在扩展链接 documentation 中使用的一般 title 元素。
      用于扩展链接，弧，定位器。
      详见 http://www.w3.org/TR/xlink/#title-element。
      </documentation>
    </annotation>
  </element>

</schema>
```

7.5.4.4 定位器(loc 元素)

定位器是扩展链接的子元素，指向扩展链接本身之外的资源。所有 XBRL 扩展链接都可包括定位器，其属性如表 8 所示。

表 8　定位器元素的属性表

属性名称	选择性	描　述
xlink:type	必备	其值应为“locator”
xlink:href	必备	此属性应是一个 URI。URI 应指向一个 XML 文档或者 XML 文档中的一个或多个 XML 片段。相对 URI 在使用前应解析为绝对 URI。所有被 xlink:href 属性引用的文档应作为 DTS 的一部分被发现，而不考虑该定位器出现在哪个链接库中
xlink:label	必备	在定位器中的 xlink:label 属性标识了该定位器，使得在同一扩展链接中的弧可引用它。在一个扩展链接中的多个定位器和资源可有相同的 xlink:label 属性值。xlink:label 属性值一定为一个 NCName（http://www.w3.org/TR/REC-xml-names/#NT-NCNAME），这要求 xlink:label 属性应以一个字母或者下划线开头

XML 模式对定位器元素的约束如下所示：

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

  <complexType name = "locatorType">
    <annotation>
      <documentation>
      通用定位器类型。
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <sequence>
          <element ref = "xl:title" minOccurs = "0" maxOccurs = "unbounded" />
        </sequence>
        <attributeGroup ref = "xlink:locatorType"/>
        <attribute ref = "xlink:href" use = "required" />
        <attribute ref = "xlink:label" use = "required" />
        <attribute ref = "xlink:role" use = "optional" />
        <attribute ref = "xlink:title" use = "optional" />
      </restriction>
    </complexContent>
  </complexType>
  <element name = "locator" type = "xl:locatorType" abstract = "true">
    <annotation>
      <documentation>
      抽象定位器元素作为定位器替换组中的头元素
      用于 XBRL 中的所有扩展链接定位器。
      </documentation>
    </annotation>
  </element>

</schema>
```

为了满足一致性，loc 元素是为了应用于 XBRL 扩展链接而定义的唯一定位器。loc 元素是通用定位器的一个具体形式。

XML 模式对 loc 元素的约束如下所示：

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

  〈element name = "loc" type = "xl:locatorType" substitutionGroup = "xl:locator"〉
   〈annotation〉
     〈documentation〉
     具体定位器元素。loc 元素是 XBRL 中的所有扩展链接的 XLink 定位器元素。
     〈/documentation〉
    〈/annotation〉
  〈/element〉

〈/schema〉
```

定位器可包括标题，应和扩展链接的子标题遵循相同的要求。

7.5.4.5 资源（resource 元素）

资源是扩展链接中的一个 XML 片段，通过扩展链接既可与扩展链接中的资源有联系，也可与扩展链接外部的资源有联系，其属性如表 9 所示。

表 9 Resource 的属性表

属性名称	选择性	描　述
xlink:type	必备	其值一定为“resource”
xlink:label	必备	在资源中的 xlink:label 属性标识了该资源，使得在相同扩展链接中的弧可引用它。资源中的 xlink:label 属性应和定位器中的 xlink:label 属性遵循相同的规范。在一个扩展链接的多个资源可有相同的标签
xlink:role	可选	资源的可选属性 xlink:role 作为资源的角色值被引用。 资源中可包含 xlink:role 属性，根据资源所含信息的本质，xlink:role 属性被用来区分资源。本部分中定义的一些资源有一组针对它们定义的标准资源角色值。自定义引用角色可使用 roleType 来定义
id	可选	id 属性可在 XBRL 扩展链接的所有资源中存在。id 属性值应遵守 XML 规范中关于带有 ID 类型的属性的规则（http://www.w3.org/TR/REC-xml # NT TokenizedType）。id 属性标识了资源，使得它能够被其他弧禁止的扩展链接中的定位器引用

XML 模式对 Resource 的约束如下所示：

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

  <complexType name = "resourceType">
    <annotation>
      <documentation>
      资源类型元素的一般类型。
      </documentation>
    </annotation>
    <complexContent mixed = "true">
      <restriction base = "anyType">
        <attributeGroup ref = "xlink:resourceType"/>
        <attribute ref = "xlink:label" use = "required" />
        <attribute ref = "xlink:role" use = "optional" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute name = "id" type = "ID" use = "optional" />
      </restriction>
    </complexContent>
  </complexType>
  <element name = "resource" type = "xl:resourceType" abstract = "true">
    <annotation>
      <documentation>
     抽象元素作为资源元素替换组的头元素使用。
      </documentation>
    </annotation>
  </element>

</schema>
```

7.5.4.6 arc 元素

7.5.4.6.1 概述

所有 XBRL 扩展链接都可包含 arc 元素。弧表示通过扩展链接中的定位器标识的资源间的关系，或者扩展链接中资源间的关系，其属性如表 10 所示。

表 10 arc 元素的属性表

属性名称	选择性	描　述
xlink:type	必备	其值应是“arc”
xlink:from	必备	弧中的 xlink:from 属性应和相同扩展链接元素中的至少一个定位器或资源的 xlink:label 属性值相等。 xlink:from 属性值应是一个 NCName，即 xlink:from 属性应以一个字母或者一个下划线开头
xlink:to	必备	一个弧的 xlink:to 属性应和相同扩展链接元素中的至少一个定位器或资源的 xlink:label 属性值相等。 xlink:to 属性值应是一个 NCName，即 xlink:to 属性应以一个字母或者一个下划线开头

表 10 arc 元素的属性表（续）

属性名称	选择性	描　述
xlink:arcrole	必备	xlink:arcrole 属性用于记录弧所表达的特定类型的关系。它的值称为弧角色值。 本部分定义了一组标准的 arcrole 值，自定义 arcrole 值可使用 arcroleTypes 来定义。
order 属性	可选	order 属性是可选择的，它应有一个小数值；当关系的等级网络展示时，该小数值可表明应用软件在正确展示兄弟概念时的次序。如不选择该属性，order 属性必为默认值“1”。如果在等级网络中的多个兄弟有相同的 order 属性值，那么这些兄弟的展示顺序由应用软件决定。order 属性值不一定是整数，这样可方便在已经定义好的两个兄弟中间添加一个新的兄弟

arc 元素的 XML 模式约束如下所示：

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

  <simpleType name = "useEnum">
    <annotation>
      <documentation>
      扩展链接弧中 use 属性的枚举值。
      </documentation>
    </annotation>
    <restriction base = "NMTOKEN">
      <enumeration value = "optional" />
      <enumeration value = "prohibited" />
    </restriction>
  </simpleType>

  <complexType name = "arcType">
    <annotation>
      <documentation>
      基本扩展链接弧类型——在必要时对特定弧进行扩展，通过对一般弧类型增加 use，priority 和 order 属性进行扩展
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <sequence>
          <element ref = "xl:title" minOccurs = "0" maxOccurs = "unbounded" />
        </sequence>
        <attributeGroup ref = "xlink:arcType"/>
        <attribute ref = "xlink:from" use = "required" />
        <attribute ref = "xlink:to" use = "required" />
        <attribute ref = "xlink:arcrole" use = "required" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute ref = "xlink:show" use = "optional" />
        <attribute ref = "xlink:actuate" use = "optional" />
        <attribute name = "order" type = "decimal" use = "optional" />
```

```
        〈attribute name = "use" type = "xl:useEnum" use = "optional" /〉
        〈attribute name = "priority" type = "integer" use = "optional" /〉
        〈anyAttribute namespace = "##other" processContents = "lax" /〉
      〈/restriction〉
    〈/complexContent〉
  〈/complexType〉
  〈element name = "arc" type = "xl:arcType" abstract = "true"〉
    〈annotation〉
      〈documentation〉
    抽象元素作为弧元素替换组中的头元素使用
      〈/documentation〉
    〈/annotation〉
  〈/element〉

〈/schema〉
```

弧展现了 XML 片段之间的关系，见示例 1、示例 2、示例 3，其 XML 片段是通过弧的两个属性 xlink:from 和 xlink:to 引用的。xlink:from 和 xlink:to 两个属性分别代表了弧的两边。这两个属性包含了定位器和资源的 xlink:label 属性值，该定位器和资源与弧本身在同一个扩展链接中。对于一个定位器，被引用的 XML 片段由定位器的 xlink:href 属性标识的一组 XML 元素组成。对于一个资源，被引用的 XML 片段是资源元素本身。当扩展链接中的多个定位器和/或资源具有相同的 xlink:label 属性值，并且该属性值是靠弧的某端识别的，弧可分别在两端同时引用多个 XML 片段。

示例 1：

一对一的弧关系

该列报链接包含一个弧，它把一个 XBRL 概念和另一个 XBRL 概念链接起来。在"from"端的 XML 片段为概念 A 元素在 example.xsd 分类标准模式中的定义。在"to"端的 XML 片段为概念 B 元素在 example.xsd 分类标准模式中的定义。

```
〈presentationLink xlink:type = "extended"
  xlink:role = "http://www.xbrl.org/2003/role/link"〉
  〈loc xlink:type = "locator" xlink:label = "a" xlink:href = "example.xsd#conceptA"/〉
  〈loc xlink:type = "locator" xlink:label = "b" xlink:href = "example.xsd#conceptB"/〉
  〈presentationArc xlink:type = "arc" xlink:from = "a" xlink:to = "b"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/parent-child" order = "1"/〉
〈/presentationLink〉
```

示例 2：

一对多的弧关系

该标签链接包含单一的弧，它把一个 XBRL 概念和两个 XBRL 标签相链接。这是通过给每个标签资源赋予相同的 xlink:label 属性值而实现的，该属性值和弧中的 xlink:to 属性值也是相同的。该弧代表了两种关系，一种是概念 A 和标准标签("Concept A")之间的联系，另外一种是概念 A 和合计标签("Total of Concept A")之间的关系。

```
〈labelLink xlink:type = "extended"
xlink:role = "http://www.xbrl.org/2003/role/link"〉
〈loc xlink:type = "locator" xlink:label = "a" xlink:href = "example.xsd#conceptA"/〉
〈label xlink:type = "resource" xml:lang = "en" xlink:label = "lab_a"
xlink:role = "http://www.xbrl.org/2003/role/label"〉Concept A〈/label〉
〈label xlink:type = "resource" xml:lang = "en" xlink:label = "lab_a"
xlink:role = "http://www.xbrl.org/2003/role/totalLabel"〉Total of Concept A〈/label〉
〈labelArc xlink:type = "arc" xlink:from = "a" xlink:to = "lab_a"
  xlink:arcrole = "http://www.xbrl.org/2003/arcrole/concept-label"/〉
〈/labelLink〉
```

该扩展链接也可表达相同的两种关系，但是需要分别列出各标签和两个弧的 xlink:label 属性值。

```
〈labelLink xlink:type = "extended"
 xlink:role = "http://www.xbrl.org/2003/role/link"〉
 〈loc xlink:type = "locator" xlink:label = "a" xlink:href = "example.xsd#conceptA"/〉
 〈label xlink:type = "resource" xml:lang = "en" xlink:label = "lab_a_standard"
xlink:role = "http://www.xbrl.org/2003/role/label"〉Concept A〈/label〉
 〈label xlink:type = "resource" xml:lang = "en" xlink:label = "lab_a_total"
xlink:role = "http://www.xbrl.org/2003/role/totalLabel"〉Total of Concept A〈/label〉
 〈labelArc xlink:type = "arc" xlink:from = "a" xlink:to = "lab_a_standard"
  xlink:arcrole = "http://www.xbrl.org/2003/arcrole/concept-label"/〉
 〈labelArc xlink:type = "arc" xlink:from = "a" xlink:to = "lab_a_total"
  xlink:arcrole = "http://www.xbrl.org/2003/arcrole/concept-label"/〉
〈/labelLink〉
```

从语义上讲，这两个扩展链接表现的是概念及其标签之间相同的关系集。

示例 3：

多对多的弧关系

该标签链接包含单一的弧，它将两个概念和两个标签联系起来。这是通过给每个概念的定位器赋予相同的 xlink:label 属性值实现的，该属性值和弧中的 xlink:from 属性值是相同的，并且，还给每个标签资源赋予相同的 xlink:label 属性值，该属性值和 xlink:to 属性值是相同的。

```
〈labelLink xlink:type = "extended"
  xlink:role = "http://www.xbrl.org/2003/role/link"〉
  〈loc xlink:type = "locator" xlink:label = "ab" xlink:href = "example.xsd#conceptA"/〉
  〈loc xlink:type = "locator" xlink:label = "ab" xlink:href = "example.xsd#conceptB"/〉
  〈label xlink:type = "resource" xml:lang = "en" xlink:label = "lab_ab"
xlink:role = "http://www.xbrl.org/2003/role/label"〉Concept A or B〈/label〉
  〈label xlink:type = "resource" xml:lang = "en" xlink:label = "lab_ab"
xlink:role = "http://www.xbrl.org/2003/role/totalLabel"〉Total of Concept A or B〈/label〉
  〈labelArc xlink:type = "arc" xlink:from = "ab" xlink:to = "lab_ab"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/concept-label"/〉
〈/labelLink〉
```

弧表示以下四种关系：

a) 概念 A 和标签资源“Concept A or B”之间的关系；

b) 概念 A 和标签资源“Total of Concept A or B”之间的关系；

c) 概念 B 和标签资源“Concept A or B”之间的关系；

d) 概念 B 和标签资源“Total of Concept A or B”之间的关系。

正如一对多的例子，该扩展链接可用 4 个一对一的弧重写，这四个弧的每个定位器和资源有一个唯一的 xlink:label 属性值。它也可用两个一对二的弧重写，这两个弧的标签资源有相同的 xlink:label 属性值，定位器有唯一的 xlink:label 属性，反之亦然。

在一个扩展链接中，不允许存在任何 XLink 重复的弧。XLink 重复的弧是指一个扩展链接中有相同的 xlink:from 以及 xlink:to 属性值的弧。见示例 4。

示例 4：

依照 XLink 规范正确使用弧

根据 XLINK 规范，在单个扩展链接中禁止出现重复弧，并且在决定是否重复时应忽略 arcrole，下面是一个错误的例子：

```
〈definitionLink xlink:type = "extended"
  xlink:role = "http://www.xbrl.org/2003/role/link"〉
  〈loc xlink:type = "locator" xlink:label = "a" xlink:href = "example.xsd#conceptA"/〉
  〈loc xlink:type = "locator" xlink:label = "b" xlink:href = "example.xsd#conceptB"/〉
  〈definitionArc xlink:type = "arc" xlink:from = "a" xlink:to = "b"
```

```
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/generalspecial" />
  <definitionArc xlink:type = "arc" xlink:from = "a" xlink:to = "b"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/requires-element"/>
</definitionLink>
```

相反，根据 XLink 规范，应使用一个合法的替代构造，例如：

```
<definitionLink xlink:type = "extended"
xlink:role = "http://www.xbrl.org/2003/role/link">
  <loc xlink:type = "locator" xlink:label = "a" xlink:href = "example.xsd#conceptA"/>
  <loc xlink:type = "locator" xlink:label = "b" xlink:href = "example.xsd#conceptB"/>
  <definitionArc xlink:type = "arc" xlink:from = "a" xlink:to = "b"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/generalspecial" />
</definitionLink>
<definitionLink xlink:type = "extended"
xlink:role = "http://www.xbrl.org/2003/role/link">
  <loc xlink:type = "locator" xlink:label = "a" xlink:href = "example.xsd#conceptA"/>
  <loc xlink:type = "locator" xlink:label = "b" xlink:href = "example.xsd#conceptB"/>
  <definitionArc xlink:type = "arc" xlink:from = "a" xlink:to = "b"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/requires-element"/>
</definitionLink>
```

7.5.4.6.2 title 元素

弧元素可包含标题，应与扩展链接的子标题遵循相同的要求。

7.5.4.6.3 禁止和覆盖关系

7.5.4.6.3.1 概述

一个分类标准创建者通常不能修改其他分类标准创建者创建的链接库。如果他需要修改这些链接库所表达的关系，那么可以建立一个新的链接库，以此体现对特定关系进行禁止或者覆盖的关系。无论是要覆盖还是禁止一个现有的关系，都需要创建一个新的弧。

禁止弧是表达一个或者一组禁止关系的弧。禁止关系是用于否定其他关系的关系。覆盖弧是表达一个或者一组覆盖关系的弧。覆盖关系是用于取代其他关系的关系。当确定一个 DTS 中的关系网内的不同关系时，禁止和覆盖是相关的。

用于表达禁止和覆盖关系的弧受 use 和 priority 两个属性控制，本部分中定义的所有弧元素都有这两个属性，其属性如表 11 所示。

表 11 禁止和覆盖关系的弧元素的属性表

属性名称	选择性	描　述
use	可选	取值应是“optional” 和 “prohibited”两者之一。 其中，若 use=“optional”，则表示该弧代表的一个或者一组关系可参与到 DTS 中的关系网中。当 use 属性不明确时，其默认值为“optional”。 若 use=“prohibited”，则表示该弧代表的一个或者一组关系禁止其本身及其他等价关系参与到 DTS 中的关系网络中。这种关系被称为是禁止关系
priority	可选	priority 属性的内容应是一个整数，其默认值为“0”。在关系网中应用禁止以及覆盖规则时，应使用此属性。每个关系都有一个优先权，它的值等于用于表达此关系的弧的 priority 属性值

7.5.4.6.3.2 DTS 中的关系网

DTS 中的扩展链接所表示的弧描述了 XML 片段之间的关系网。

单独地来说，每个弧描述一个或者多个关系。但是，在一个 DTS 的内部，仅仅其中的一些关系参与到 DTS 所描述的关系网中。

DTS 的所有关系都是 DTS 所描述的内部关系网的候选关系。但是，有一些被其他关系禁止或者覆盖的关系是被 DTS 所描述的关系网排斥在外的。

DTS 中的所有弧被分为组织到弧的几个基础集中。在一个弧基础集中的全部弧需要满足以下条件：

a) 有相同的本地名称、命名空间以及 xlink:arcrole 属性值；

b) 被包含在有相同的本地名称、命名空间以及 xlink:role 属性值的扩展链接元素中。

DTS 中的每个弧的每个基础集都是表达内部关系网候选关系的集合。对于 DTS 中的弧的每个弧基础集，关系禁止和覆盖的规则决定了参与到 DTS 中内部相应关系网的基础集中的关系的子集。

7.5.4.6.3.3 等价关系

应用关系禁止和覆盖规则时，基础集中的弧表达的每个关系都要和该基础集中的弧表达的所有其他关系进行比较。

在下列情形下认为给定的基础集中的弧所表达的两种关系是等价的：

a) 下面的条件需要满足后模式验证信息集(post-schema-validation infoset)：

——弧有非豁免属性(non-exempt attributes)的同样数字；

——第一弧的每个非豁免属性，在第二弧有一个相对应的结构等价的非豁免属性(弧的 use 和 priority 两个属性是豁免属性，以下命名空间中的其他属性也都是豁免属性)：

- http://www.w3.org/2000/xmlns/；
- http://www.w3.org/1999/xlink；

——所有其他属性是非豁免属性。

b) XML 片段中关系的“from”端相同。

c) XML 片段中关系的“to”端相同。

7.5.4.6.3.4 禁止和覆盖关系规则

对于一个基础集中的弧所表达的每个关系，如果该关系包含在该弧的基础集的关系网中，禁止和覆盖关系规则由弧的 use 和 priority 属性以及关系等价概念来决定。

禁止和覆盖规则按下列方式应用在每个等价关系集合上，每个等价关系集合是由基础集中的弧表示：

a) 基础集中的弧表达的关系网不包含该集合中的任何禁止关系。

b) 如果只有一个关系有最高优先级，而且该关系不是禁止关系，那么该关系是覆盖关系，应包含在基础集的关系网中。弧基础集中的所有其他等价关系都不包含在关系网中。

c) 如果不止一个关系有最高优先级，而且都不是禁止关系，那么这些最高优先级关系有且仅有一个应包含在弧基础集的关系网中。被选中的关系应当是覆盖关系。所有其他等价关系(即“被覆盖关系”)不允许包含在弧基础集的关系网中。具体选择哪个关系包含在弧基础集的关系网中，由应用程序决定。

d) 如果不止一个关系有最高优先级，并且这些关系中至少有一个是禁止关系的，那么这些等价的关系都不能被包含在弧的基础集的关系网中(这些不是禁止等价关系，而是被禁止了)。

禁止和覆盖关系规则的具体示例见示例 5。

示例 5：

禁止和覆盖关系

以下这组示例包含了一些很少能够遇到的情况，并且示范了如何根据禁止和覆盖关系规则处理这些情况。这些例子给出了一系列扩展分类标准，这些扩展分类标准可以是由分类标准创建者以外的人创建。

如果弧基础集中的两个弧表现的是一组等价关系（如下所示），那么这两个关系都没有被包含在与该弧基础集相关联的关系网中。

- 弧 A 用 use="optional"并且 priority="1"表示关系 A；
- 弧 B 用 use="prohibited"并且 priority="2"表示关系 B。

弧 B 的优先级较高，并且表达一种禁止关系。因此关系 A 被关系 B 排除在与弧基础集相关的关系网之外。由于关系 B 禁止了关系 A，因此，根据定义，它不被包括在与弧的基础集相关的关系网中。

如果后续有另外一个弧按照以下形式介入到弧的基础集中：

- 弧 C 用 use="prohibited"和 priority="3"表示关系 C。

并且，关系 C 与关系 A、B 是等价的，因为关系 C 有最高优先级，它是一个禁止关系。因此，关系 A 被关系 C 排除在与弧的基础集相关联的关系网之外。关系 B 和 C 是禁止关系，因此，根据定义，它们不被包含在与此弧基础集相关联的关系网中。

如果后续有另外一个弧按照以下形式介入到弧的基础集中：

- 弧 D 用 use="optional"和 priority="4"表示关系 D。

并且关系 D 与关系 A、B、C 是等价的，因为关系 D 有最高优先级，所以，它是一个覆盖关系。因此，关系 A、B 和 C 不被包含在与此弧基础集相关联的关系网中。从而，该关系 D 实际地覆盖了禁止关系 B 和 C 的效果，因此被包含在与弧基础集相关联的关系网中。

如果后续有另外一个弧按照以下形式介入到弧的基础集中：

- 弧 E 用 use="optional"和 priority="4"表示关系 E。

并且，关系 E 和关系 A、B、C、D 等价，因为关系 E 和关系 D 的优先级相等，所以，关系 E 和关系 D 究竟哪一个是覆盖关系，需要由应用程序决定。关系 A、B 和 C 仍然不被包含在与此弧基础集相关联的关系网中。因为这几个关系是等价的，那么，关系 E 和关系 D 究竟哪一个是覆盖关系要由应用程序决定的事实，变得不重要了，因为任选其中的哪一个都不会影响所要表达的语义。

如果后续有另外一个弧按照以下形式介入到弧的基础集组中：

- 弧 F 用 use="prohibited"和 priority="4"表示关系 F。

并且，关系 F 和关系 A、B、C、D、E 都是等价的，因为关系 F 是有最高优先级的关系之中的一个，而且关系 F 是一个禁止关系，因此关系 A、B、C、D、E、F 都不被包含在与此弧基础集相关联的关系网中。

将一个 DTS 中所有被发现的弧划分为基础集并在一系列关系网中的禁止和覆盖规则的应用过程产生了一系列的关系网，每个关系网中包括的关系如下：

——用具有相同本地名称、命名空间以及 xlink:arcrole 属性值的 arcType 元素的弧表达；

——用包含在有相同本地名字、命名空间以及 xlink:role 属性值的 extendedType 元素中的弧表达；

——不是被禁止关系、禁止关系或者覆盖关系。

7.5.5 在 URI 片段标识符中使用 XPointer

为了能够指向一个特定的 XML 元素，在 XLink 规范中多个链接（href）中用到的 URI 应以一个片段标识符结束。根据 XLink 规范，XPointer 语法允许在片段标识符中使用。片段标识符的格式应符合快捷指针的要求（http://www.w3.org/TR/xptr-framework/#shorthand），或者符合一个基于架构的

指针的要求(http://www.w3.org/TR/xptr framework/#scheme)。在 XBRL 链接中唯一允许的基于架构的指针是元素模式。

示例:

xlink:href 值示例

示例	含　义
#f1	当前文档片段有值为"f1"的 id 属性
us_bs_v21.xsd#currentAssets	us_bs_v21.xsd 文档有 id 属性值为"currentAssets"的元素
us_bs_v21.xsd#element(/1/14)	us_bs_v21.xsd 文档的该元素是根元素的第 14 子元素(根据文档顺序)
us_bs_v21.xsd#element(currentAssets)	us_bs_v21.xsd 文档的该元素有一个值为"currentAssets"的 id 属性

8 XBRL 实例

8.1 概述

XBRL 实例是以"xbrl"为根元素的 XML 片段。XBRL 实例包含若干事实,每个事实都对应着一个概念,概念在 DTS 中被定义。XBRL 实例还包含 context 和 unit 元素,它们为解释实例中的事实提供所需的额外信息。

事实可是简单的,在这种情况下事实值将被表示成简单的内容(那些以比率表示其值的简单事实除外);事实也可是复合的,在这种情况下事实值由其他的简单事实或者复合事实的值复合而成。简单事实用数据项来表达,而复合事实用元组表达。

虽然任何给定元组或数据项的语法仅能在单个分类标准模式中进行定义,XBRL 实例中出现的数据项或者元组却可取自任意个分类标准模式。

XBRL 实例可识别一些分类标准模式和 XBRL 链接库,并且以此作为起点进而发现支持这些实例的 DTS。在获取 DTS 的过程中作为起点的分类标准模式与链接库由 XBRL 实例中的 schemaRef 元素和 linkbaseRef 元素分别识别。这使得 XBRL 实例能够对其所报告信息的解释加以控制。

一份 XBRL 实例应遵循以下规则:XBRL 实例的语法通过一组 XML 模式进行约束;所有的 XBRL 实例应是符合 XML 模式定义的有效的 XML 文档。

至于 XBRL 实例的语义及其内容,仅在其会对使用本部分的软件应用程序产生影响时予以说明。

8.2 根元素 xbrl

每个 XBRL 实例,都至少包括一个名称为 xbrl 的元素,作为封装 XBRL 实例的根元素。元素 xbrl 不能为空,至少包括一个数据项和本身包含子元素的上下文元素。如果一个更大的文档中包含多个由 XBRL 标记的数据块,每个数据块都要有一个 xbrl 元素作为容器。因此,一个 XBRL 实例的根元素应是容器元素,其属性如表 12 所示。

表 12 XBRL 实例的根元素 xbrl 的属性表

属性名称	选择性	描　述
id	可选	id 属性的值应满足 XML 规范中关于 ID 类型属性的规定(http://www.w3.org/TR/REC xml#NT TokenizedType)
xml:base	可选	xbrl 元素可有一个 xml:base 属性,用于解析 XBRL 实例中的相对 URI

XML 模式对 xbrl 元素的约束如下所示：

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

  <element name = "xbrl">
    <annotation>
      <documentation>
      XBRL 实例的根元素。
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref = "link:schemaRef" minOccurs = "1" maxOccurs = "unbounded" />
        <element ref = "link:linkbaseRef" minOccurs = "0" maxOccurs = "unbounded" />
        <element ref = "link:roleRef" minOccurs = "0" maxOccurs = "unbounded" />
        <element ref = "link:arcroleRef" minOccurs = "0" maxOccurs = "unbounded" />
        <choice minOccurs = "0" maxOccurs = "unbounded">
          <element ref = "xbrli:item"/>
          <element ref = "xbrli:tuple"/>
          <element ref = "xbrli:context"/>
          <element ref = "xbrli:unit"/>
          <element ref = "link:footnoteLink"/>
        </choice>
      </sequence>
      <attribute name = "id" type = "ID" use = "optional" />
      <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
        processContents = "lax" />
    </complexType>
  </element>

</schema>
```

示例：

使用 xbrl 作为根元素

```
<xbrl xmlns = "http://www.xbrl.org/2003/instance"
          xmlns:xlink = "http:// www.w3.org/1999/xlink"
          xmlns:link = "http://www.xbrl.org/2003/linkbase"
          xmlns:xsi = "http://www.w3.org/2001/XML Schema-instance"
          xmlns:ci = "http://www.xbrl.org/us/gaap/ci/2003/usfr-ci-2003"
          xsi:schemaLocation = "
    http://www.xbrl.org/us/fr/ci/2003/usfr-ci-2003
    http://www.xbrl.org/us/fr/ci/2000-07-31/usfr-ci-2003.xsd">
      <link:schemaRef xlink:type = "simple"
                xlink:href = "http://www.xbrl.org/us/fr/ci/2000-07-31/usfr-ci-2003.xsd"/>
      <ci:assets precision = "3" unitRef = "u1" contextRef = "c1">727</ci:assets>
      <ci:liabilities precision = "3" unitRef = "u1" contextRef = "c1">635</ci:liabilities>
      <context id = "c1"><! --… --></context>
      <unit id = "u1"><! --… --></unit>
    </xbrl>
```

含义：xbrl 保留命名空间前缀定义和 schemaLocation 属性。

8.3 schemaRef 元素

每个 XBRL 实例应包含至少一个 schemaRef(模式引用)元素。它是一个简单链接,应作为 xbrl 元素的子元素出现。XBRL 实例中所有的 schemaRef 元素在文档中都应先于 xbrl 元素的其他子元素出现。

在 XBRL 实例中,schemaRef 元素指向一个分类标准模式,该分类标准模式是支持该 XBRL 实例的 DTS 的一部分,其属性如表 13 所示。

表 13 模式引用元素的属性表

属性名称	选择性	描 述
xlink:type	必备	其值为"simple"
xlink:href	必备	xlink:href 属性应是 URI。URI 应指向某个 XML 模式。相对 URI 在使用前应解析为绝对 URI
xlink:arcrole	可选	本部分没有赋予该属性任何语义。xlink:arcrole 属性值应是符合 XLink 规范定义的 URI
xlink:role	可选	xlink:role 属性可用于 schemaRef 元素。本部分没有对出现在 schemaRef 元素上的 xlink:role 的语义作出定义。xlink:role 属性值应是符合 XLink 规范定义的 URI
xml:base	可选	可出现在 schemaRef 元素中,参与解析 xlink:href 属性中的相对 URI
注:同一 XBRL 实例中,若 schemaRef 元素与 schemaLocation 属性不符,则 XBRL 处理器可能无法正确解析。		

xlink:schemaRef 的 XML 模式定义如下所示:

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

  <complexType name = "simpleType">
    <annotation>
      <documentation>
      XBRL 中定义的简单链接类型。
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <attributeGroup ref = "xlink:simpleType"/>
        <attribute ref = "xlink:href" use = "required" />
        <attribute ref = "xlink:arcrole" use = "optional" />
        <attribute ref = "xlink:role" use = "optional" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute ref = "xlink:show" use = "optional" />
        <attribute ref = "xlink:actuate" use = "optional" />
        <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
          processContents = "lax" />
      </restriction>
    </complexContent>
```

```
  </complexType>

</schema>

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

<element name = "schemaRef" type = "xl:simpleType" substitutionGroup = "xl:simple">
  <annotation>
    <documentation>
    定义 schemaRef 元素     用于链接 XBRL 实例和 XBRL 分类标准模式。
    </documentation>
  </annotation>
</element>

</schema>
```

8.4 linkbaseRef 元素

XLink 规范提出了一种查找链接库的标准方法。linkbaseRef(链接库引用)元素通过使用某个特定的 xlink:arcrole 值来遵循这一标准。

xbrl 元素允许一个或多个 linkbaseRef 元素作为其子元素出现。如果 linkbaseRef 元素在 xbrl 元素中作为子元素出现,它们在文档中的顺序应紧随 schemaRef 元素之后,并且在其他任何元素之前。

linkbaseRef 元素标识 XBRL 实例的一个链接库,该链接库是支持该 XBRL 实例的 DTS 的一部分,其属性如表 14 所示,角色如表 15 所示。

表 14 linkbaseRef 元素的属性表

属性名称	选择性	描　述
xlink:type	必备	其值为"simple"
xlink:href	必备	xlink:href 属性应是一个 URI,该 URI 应指向一个链接库。该链接库包含适当的扩展链接,扩展链接的类型由 xlink:role 属性的值决定。相对 URI 在使用前应解析为绝对 URI
xlink:arcrole	必备	linkbaseRef 元素的属性 xlink:arcrole 应设置为 XLink 规范中规定的恒值"http://www.w3.org/1999/xlink/properties/linkbase"
xlink:role	可选	属性限定了由 linkbaseRef 元素所标识的链接库中所允许的扩展链接的种类。扩展链接的种类(计算、定义、标签、展示和参考链接): ——应被具有标准 xlink:role 属性值的 linkbaseRef 元素指向的链接库所包含; ——应不被具有标准 xlink:role 属性值的 linkbaseRef 元素指向的链接库所包含。 如果一个 linkbaseRef 元素关联到某个包含扩展链接的链接库,而且该扩展链接尚未在本部分中定义,那么可使用 xlink:role 的非标准值,或者忽略 xlink:role 该属性。 linkbaseRef 元素中的角色值如表 15 所示
xlink:base	可选	属性可出现在 linkbaseRef 元素中,参与解析在 xlink:href 属性中的相对 URI

XML 模式对链接库引用元素的约束如下所示：

```
〈schema targetNamespace = "http://www.xbrl.org/2003/linkbase"
  xmlns = "http://www.w3.org/2001/XML Schema"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
  xmlns:xl = "http://www.xbrl.org/2003/XLink"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  elementFormDefault = "qualified"〉

  〈element name = "linkbaseRef" substitutionGroup = "xl:simple"〉
    〈annotation〉
      〈documentation〉
  定义了链接库引用元素：该元素用于从分类标准模式和 XBRL 实例链接到 XBRL 分类标准的扩展链接。
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈complexContent〉
        〈restriction base = "xl:simpleType"〉
          〈attribute ref = "xlink:arcrole" use = "required"〉
            〈annotation〉
              〈documentation〉
              该属性的值为：
              http://www.w3.org/1999/xlink/properties/linkbase
              〈/documentation〉
            〈/annotation〉
          〈/attribute〉
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
processContents = "lax" /〉
        〈/restriction〉
      〈/complexContent〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

表 15　linkbaseRef 元素中的角色

linkbaseRef 中的 xlink:role 属性值	xlink:href 所指向的元素
未指定	可包含任何扩展链接元素
http://www.xbrl.org/2003/role/calculationLinkbaseRef	应且仅能包含 calculationLink 元素
http://www.xbrl.org/2003/role/definitionLinkbaseRef	应且仅能包含 definitionLink 元素
http://www.xbrl.org/2003/role/labelLinkbaseRef	应且仅能包含 labelLink 元素
http://www.xbrl.org/2003/role/presentationLinkbaseRef	应且仅能包含 presentationLink 元素
http://www.xbrl.org/2003/role/referenceLinkbaseRef	应且仅能包含 referenceLink 元素

8.5　roleRef 元素

XBRL 实例中可使用一个或多个 roleRef(角色引用)元素。如果用了 roleRef 元素，它们在 XBRL 实例文档中出现的顺序应紧随 linkbaseRef 元素之后。XBRL 实例使用 roleRef 元素来引用任何自定义的 xlink:role 属性值。这些属性值用于 XBRL 实例的脚注链接。

8.6 arcroleRef 元素

XBRL 实例中可使用一个或多个 arcroleRef(弧角色引用)元素。如果用了 arcroleRef 元素,它们在 XBRL 实例文档中出现的顺序应紧随 roleRef 元素之后。XBRL 实例使用 arcroleRef 元素来引用任何自定义的 xlink:arcrole 属性值。这些属性值用于 XBRL 实例的脚注链接。

8.7 item 元素

一个数据项表示单一事实或业务度量。在 XBRL 实例的 XML 模式中,数据项被定义为一个抽象的元素 item,即其本身不会出现在 XBRL 实例中。

数据项不允许自身嵌套。如果一个 XBRL 实例需要表达结构化的关系,应借助元组来表示。智能化的结构——财务概念在不同角度上的相互关联——应借助分类标准链接库的链接结构来体现,而不是将 XBRL 实例中的事实嵌套在一起。

8.8 数据类型定义

8.8.1 概述

数据项以及数据项替换组元素的数据类型在 XML 模式中的定义如下:

```
〈schema targetNamespace = "http://www.xbrl.org/2003/instance"
  xmlns = "http://www.w3.org/2001/XML Schema"
  xmlns:xbrli = "http://www.xbrl.org/2003/instance"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
  elementFormDefault = "qualified"〉

  〈attributeGroup name = "factAttrs"〉
    〈annotation〉
      〈documentation〉
      所有数据项和元组元素的属性。
      〈/documentation〉
    〈/annotation〉
    〈attribute name = "id" type = "ID" use = "optional" /〉
    〈anyAttribute namespace = "# ##other" processContents = "lax" /〉
  〈/attributeGroup〉

  〈attributeGroup name = "tupleAttrs"〉
    〈annotation〉
      〈documentation〉
      元组元素的属性组。
      〈/documentation〉
    〈/annotation〉
    〈attributeGroup ref = "xbrli:factAttrs" /〉
  〈/attributeGroup〉

  〈attributeGroup name = "itemAttrs"〉
    〈annotation〉
      〈documentation〉
      所有数据项的属性。
      〈/documentation〉
    〈/annotation〉
    〈attributeGroup ref = "xbrli:factAttrs" /〉
    〈attribute name = "contextRef" type = "IDREF" use = "required" /〉
```

```
〈/attributeGroup〉

  〈attributeGroup name = "essentialNumericItemAttrs"〉
    〈annotation〉
      〈documentation〉
      所有数值型数据项(分数型和非分数型)的属性。
      〈/documentation〉
    〈/annotation〉
    〈attributeGroup ref = "xbrli:itemAttrs" /〉
    〈attribute name = "unitRef" type = "IDREF" use = "required" /〉
〈/attributeGroup〉

〈attributeGroup name = "numericItemAttrs"〉
  〈annotation〉
    〈documentation〉
    非分数的数值型数据项的属性组。
    〈/documentation〉
  〈/annotation〉
    〈attributeGroup ref = "xbrli:essentialNumericItemAttrs" /〉
  〈attribute name = "precision" type = "xbrli:precisionType" use = "optional" /〉
  〈attribute name = "decimals" type = "xbrli:decimalsType" use = "optional" /〉
〈/attributeGroup〉

〈attributeGroup name = "nonNumericItemAttrs"〉
  〈annotation〉
    〈documentation〉
    非数值型数据项的属性组。
    〈/documentation〉
  〈/annotation〉
  〈attributeGroup ref = "xbrli:itemAttrs" /〉
〈/attributeGroup〉

〈annotation〉
  〈documentation〉
  XBRL 中的域数值型数据项类型:用于概念元素的定义。
  一共定义了下列 4 种数值类型,与 XBRL 的域空间有特别关联,因此和 XML 模式的内建类型一起被包括在内。
  〈/documentation〉
〈/annotation〉

〈complexType name = "monetaryItemType" final = "extension"〉
  〈simpleContent〉
    〈extension base = "xbrli:monetary"〉
      〈attributeGroup ref = "xbrli:numericItemAttrs" /〉
    〈/extension〉
  〈/simpleContent〉
〈/complexType〉

〈complexType name = "sharesItemType" final = "extension"〉
  〈simpleContent〉
    〈extension base = "xbrli:shares"〉
      〈attributeGroup ref = "xbrli:numericItemAttrs" /〉
    〈/extension〉
  〈/simpleContent〉
〈/complexType〉
```

```
    〈complexType name = "pureItemType" final = "extension"〉
      〈simpleContent〉
        〈extension base = "xbrli:pure"〉
          〈attributeGroup ref = "xbrli:numericItemAttrs" /〉
        〈/extension〉
      〈/simpleContent〉
    〈/complexType〉

    〈element name = "numerator" type = "decimal" /〉
    〈element name = "denominator" type = "xbrli:nonZeroDecimal" /〉
    〈complexType name = "fractionItemType" final = "extension"〉
      〈sequence〉
        〈element ref = "xbrli:numerator" /〉
        〈element ref = "xbrli:denominator" /〉
      〈/sequence〉
      〈attributeGroup ref = "xbrli:essentialNumericItemAttrs" /〉
    〈/complexType〉

    〈complexType name = "stringItemType" final = "extension"〉
      〈simpleContent〉
        〈extension base = "string"〉
          〈attributeGroup ref = "xbrli:nonNumericItemAttrs" /〉
        〈/extension〉
      〈/simpleContent〉
    〈/complexType〉

〈! --
booleanItemType, hexBinaryItemType, base64BinaryItemType, anyURIItemType, QNameItemType, durationItemType,
dateTimeItemType, timeItemType, dateItemType, gYearMonthItemType, gYearItemType, gMonthDayItemType, gDayItemType,
gMonthItemType, normalizedStringItemType, tokenItemType, languageItemType, NameItemType,…
  --〉

    〈element name = "item" type = "anyType" abstract = "true"〉
      〈annotation〉
        〈documentation〉
        作为数据项替换组头元素的抽象数据项。
        〈/documentation〉
      〈/annotation〉
    〈/element〉

  〈/schema〉
```

示例 1：

一个具有三位有效数字的数值型事实

```
〈ci:capitalLeases contextRef="c1" unitRef="u1" precision="3"〉727432〈/ci:capitalLeases〉
```

含义：在上下文 c1 下的融资租赁协议的值如果精确到三位有效数字，应为 727000。注意：有必要参考上下文以获取关于该数值的其他细节信息，如实体、期间等。同样有必要参考引用的 unit 元素以获取相关的度量单位信息。

示例 2：

非数值型数据项

〈ci:concentrationsNote contextRef="c1"〉 因为公司现金管理策略，所以短期投资信用风险的集中被认为是不重要的。公司的这些策略限制投资资金用在低风险、高流动性的有价证券上，概括描述了发行者的信用需求，并且限制了可投资在任何一个发行者的现金数量。 〈/ci:concentrationsNote〉
含义：上下文 c1 的专门注释文本。

抽象 item 元素的内容可派生于 anyType。数据项替换组的每个成员应拥有一个已定义的 XBRL 数据项类型。这将允许实例中每个数据项替换组校验自身的数据类型。对于 XML 模式中定义的每个合适的内建类型，都为之定义一个派生于该类型的 XBRL 数据项类型，fractionItemType 类型也是如此。某一个数据项除非其类型通过约束 fractionItemType 派生而来，否则不能包含复合内容。

contextRef 属性是指向 context(上下文)元素的一个 IDREF，context 元素包含所表示事实的额外信息。一个数据项应包含一个 contextRef 属性，该 contextRef 属性引用同一 XBRL 实例的某个 context 元素。XBRL 实例是 xbrl 元素的一次出现，而非整个文档。如果数据项的数据类型派生于 XML 模式的内建数值类型(包括 decimal、float、double 或派生于它们的内建类型)或 fractionItemType 约束，数据项应同时使用 contextRef 属性和 unitRef 属性；其他的数据项应使用 contextRef 属性。

unitRef 属性是指向 unit(单位)元素的 IDREF。它包含数值型事实度量单位的信息。unitRef 属性不能出现在非数值型数据项中。unitRef 属性应出现在数值型数据项中，引用同一 XBRL 实例的某个 unit 元素。

数值型数据项还提供了两个可选属性：precision 和小数 decimals (fractionItemType 除外)，从而使得 XBRL 实例创建者能够说明所表示事实的精度。其属性如表 16 所示。

表 16　数据项的属性表

属性名称	选择性	描　述
contextRef		所有的数据项都应具备相应的上下文，但所有的元组都不允许包含相应的上下文。数据项使用 contextRef 属性识别对应的 context 元素。该 context 元素与包含 contextRef 属性的数据项关联。 contextRef 属性值应与 XBRL 实例中某个 context 元素的 id 属性值相等，该实例同样包含出现 contextRef 属性的数据项中
unitRef		所有的数值型数据项应指明度量单位。一切元组和非数值型数据项均不能指定度量单位。数值型数据项通过 unitRef 属性识别其度量单位，该 unit 元素与出现 unitRef 属性的那个数据项相关联。 unitRef 属性值应与 XBRL 实例中 unit 元素的 id 属性值相等，该实例同样包含出现 unitRef 属性的数据项
precision	可选	precision 属性应为非负整数或“INF”字符串。precision 属性反映计量精度，从而反映在此基础上参与下一步计算的有效位数这种计量方法对于后续计算的作用。不同的软件包可提供不同精度的数字。precision 属性使任意厂商能够以相同的方式声明产品的精度。对于任何精度指定或推断为 n 的数值，应用程序应忽略 n 位有效数字之后的任何数位。n 位有效数字起始于该数左边第一位不为 0 的数字。 若 precision =“INF”，即该数的词法表示就是所表示事实的精确数值。 注：本部分中的定义表明 precision 和 decimals 属性表示一个范围，XBRL 实例中表示的数值所对应的事实的实际值就处在该范围中

表 16 数据项的属性表（续）

属性名称	选择性	描　述
decimals	可选	decimals 属性值须为整数或“INF”。decimals 属性决定了所表示事实数值所能精确到的小数点后的位数，可看作是舍入或截断的结果。如果某个数据事实包含一个值为 n 的 decimals 属性，就意味着能够精确到小数点后 n 位。 decimals = "INF"，即该数的词法表示就是所表示事实的精确数值

8.8.2 precision 和 decimals 属性

一个数值型数据项应拥有一个 precision（精度）属性或 decimals（小数）属性，除非它的数据类型为 fractionItemType 或经由 fractionItemType 限制而派生的类型，或者包含某个空值。在这种情况下，不能包括 precision 属性或 decimals 属性。

数值型数据项不能同时包含 precision 属性和 decimals 属性。

非数值型数据项不能包含 precision 属性或 decimals 属性。

当需要判断两个数值型数据项是否值等价时，应考虑这两个数值型数据项的 precision 属性值。

precision 表示见示例 1；decimals 表示见示例 2。

示例 1：

精度和词法表示

示例	含　义			
precision = "9"	精度为 9 位数字。从数值型事实的数值左边第一位不为 0 的数字开始的 9 位数字被认为是精确的，可参与该数值型事实的相关运算			
精度	XBRL 实例词法表示示例	显示（省略或清除所有多余数字后）	大于或等于	小于
INF	476.334	476.334	476.334	476.33400000000…1
3	205	205e0	204.5	205.5
4	2002000	2002e3	2001500	2002500
4	−2002000	−2002e3	−2002500	−2001500
2	2012	20e2	1950	2050
2	2000	20e2	1950	2050
1	99	9e1	85	95
0	1234	1234	未知	未知

简单类型 precisionType 定义了精度属性值的值空间。定义如下：

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

  <simpleType name = "precisionType">
    <annotation>
```

```
      〈documentation〉
      该类型用于规定数值型数据项的 precision 属性值，它包括
      非负整数和 INF(表示无限精确或准确值)。
      〈/documentation〉
    〈/annotation〉
    〈union memberTypes = "nonNegativeInteger"〉
      〈simpleType〉
        〈restriction base = "string"〉
          〈enumeration value = "INF" /〉
        〈/restriction〉
      〈/simpleType〉
    〈/union〉
  〈/simpleType〉

〈/schema〉
```

示例 2：

小数和词法表示

示例		含义		
decimals ="2"		该数值型事实的值精确到小数点后两位		
decimals ="2"		该数值型事实的值精确到小数点前两位，百位左边的数字都是精确的		
decimals 属性	XBRL 实例词法表示示例	显示(省略或清除所有多余数字后)	大于或等于	小于
INF	436.749	436.749	436.749	436.749 000 00…1
2	10.00	10.00	9.995	10.005
2	10	10.00	9.995	10.005
2	10.000	10.00	9.995	10.005
2	10.009	10.00	9.995	10.005
0	10	10	9.5	10.5
−1	10	10	5	15
−1	11	10	5	15
3	205	205.000	204.999 5	205.000 5
4	2 002 000	2 002 000.000 0	2 001 999.999 95	2 002 000.000 05
−2	−205	−200	−250	−150
−2	205	200	150	250
−2	2 002 000	2 002 000	2 001 950	2 002 050
−3	2 002 000	2 002 000	2 001 500	2 002 500
−4	2 002 000	2 000 000	1 995 000	2 005 000
−3	777 000	777 000	776 500	777 500

简单类型 decimalsType 定义了 decimals 属性的合法取值。其 XML 模式文件定义如下：

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

  <simpleType name = "decimalsType">
    <annotation>
      <documentation>
       该类型用于规定数值型数据项的 decimals 属性值。它包括
       整数和 INF(用来表示无穷小数或"准确值")。
      </documentation>
    </annotation>
    <union memberTypes = "integer ">
      <simpleType>
        <restriction base = "string">
          <enumeration value = "INF" />
        </restriction>
      </simpleType>
    </union>
  </simpleType>

</schema>
```

8.8.3 推导精度

XBRL 实例的用户能够在数值型数据项的 precision 属性值缺失的情况下按以下规则推测其值：

对于 fractionItemType 及其受限派生类型的数值项，如果运算中需要使用精度信息，应用程序应推测出其精度值为"INF"。

如果一个数值型数据项仅有 decimals 属性而无 precision 属性，而计算或者在 XBRL 实例中查找重复数据项时又需要使用精度信息，相应的应用程序应能够推测出数值型事实的精度。

如果已知 decimals 属性的值，假定数据项的 precision 属性值为 n，n 等于 0 和下列计算结果中的较大值：

——如果小数点(如果没有的话，则指默认的小数点位置)左边有非零数字，取该数值型事实的词法表示中小数点左边第一位非零数字开始的整数部分的位数；

——否则，如果小数点右边有非零数字，先计算出小数点和该数值型事实词法表示中第一个不为 0 的数位之间 0 的个数，取前者的相反数；

——否则取 0，加上：

该数值型事实词法表示的指数值(如果有的话)；

加上：

该数值型事实所能精确到的小数点后位数。

示例：

词法表示，精度和小数

词法表示	小数属性的值	精确度属性的推断值
123	2	3+2=5
123.4567	2	3+2=5
123e5	−3	3+5+(−3)=5
123.45e5	−3	3+5+(−3)=5
0.1e−2	5	0+(−2)+5=3
0.001E-2	5	(−2)+(−2)+5=1
0.001e-3	4	(−2)+(−3)+4= −1 小于 0,值为 0

8.8.4 准确性

在字长受到限制的前提下,一个数值类型数据项的取值,还需一种规则来描述如何舍弃次要的字节,并确保数值的准确性。本部分定义了如下两个规则:

a) "n 位有效数字"

如果词法表示某个数值保留 n 位有效数字,即从该数的词法表示中最左边的非零数字开始的 n 位数字(包括最左边的非零数字本身)是准确的,可参与该数的计算。

b) "舍入"

通常在计算之后需要进行舍入。为了舍入一个有 n 位有效数字的数,将第 n 位之后的所有数字均丢弃,一般情况下按"四舍五入"的原则进行。

示例:

舍入

原始数值	舍入为 n 位有效数字	
	$n=2$	$n=3$
3.564 3	3.6	3.56
3.567 3	3.6	3.57
0.497 87	0.50	0.498
3.999 9	4.0	4.00
9.999 991	10	10.0
22.55	23	22.6
0.001 9	0.001 9	0.001 90
0.000 02	0.000 020	0.000 020 0

若同样的过程适用于 n 取任意值的情况,则说某个词法表示的数值保留 n 位有效数字。这项技术用于生成 XBRL 实例中 precision 属性值为 n 的事实的词法表示。

c) 精确到小数点后 n 位

如果某个数精确到小数点后 n 位,那么该数与其真值的差(也称作绝对差 $-e_{abs}$)小于或等于 0.5×10^{-n}。n 可取正整数,也可取负整数或零。

其数学表达形式如下:

对于任意实数 X,当且仅当满足以下条件时,x 表示精确到小数点后 n 位的 X,

$$e_{abs}=|X-x|\leqslant 0.5\times10^{-n}$$

或者,根据舍入约定

$$-0.5\times10^{-n}\leqslant x-X<0.5\times10^{-n}$$

示例：

精确到小数点后 n 位

123 456.789 012 精确到小数点后 n 位				
$n=-3$	$n=-2$	$n=0$	$n=3$	$n=6$
123 000	123 500	123 457	123 456.789	123 456.789 012

8.9 context 元素

8.9.1 概述

context(上下文)元素包含所描述实体的信息、报告期间和报告场景，对于理解可作为 XBRL 数据项的业务事实都是必要的。

上下文元素的 XML 模式约束如下所示：

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

  〈element name = "context"〉
    〈annotation〉
      〈documentation〉
      用于联系事实的上下文。
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈sequence〉
        〈element name = "entity" type = "xbrli:contextEntityType" /〉
        〈element name = "period" type = "xbrli:contextPeriodType" /〉
        〈element name = "scenario" type = "xbrli:contextscenarioType"
                  minOccurs = "0" /〉
      〈/sequence〉
      〈attribute name = "id" type = "ID" use = "required" /〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

8.9.2 id 属性

每个 context 元素应包含 id 属性。id 属性值应遵循 XML 规范中有关 ID 类型属性的规定(http://www.w3.org/TR/REC-xml#NT-TokenizedType)。id 属性标识上下文元素并确保它可被数据项所引用。

示例：ID 属性示例

正例	id="C2424"	
反例	id="42"	ID 类型的内容不能以数字开头

8.9.3 period 元素

period(时间类型)元素包含 item 元素需要引用的时间点或时间间隔。period 的子元素用于构造若干可选择的日期间隔表达中的其中一个，其含义如表 17 所示，数据类型如表 18 所示。

表 17 period 的子元素含义对照表

元 素	含 义
起始日期(startDate)，结束日期(endDate)	具有指定起点和终点的一段时间
时间点(instant)	某个时间点
永久(forever)	表示“永久”的元素

每个时间子元素均使用日期的标准 XML 模式表示。

时间元素的 XML 模式约束如下所示：

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

  <simpleType name = "dateUnion">
    <annotation>
      <documentation>
      日期和日期与时间的结合的联合体
      </documentation>
    </annotation>
    <union memberTypes = "date dateTime " />
  </simpleType>

  <complexType name = "contextPeriodType">
    <annotation>
      <documentation>
      时间元素的类型，用于描述报告日期信息
      </documentation>
    </annotation>
    <choice>
      <sequence>
        <element name = "startDate" type = "xbrli:dateUnion" />
        <element name = "endDate" type = "xbrli:dateUnion" />
      </sequence>
      <element name = "instant" type = "xbrli:dateUnion" />
      <element name = "forever">
        <complexType />
      </element>
    </choice>
  </complexType>

</schema>
```

表 18 子元素数据类型对照表

子元素	XML 数据类型
instant(时间点)	date 或 dateTime
forever(永久)	Empty
startDate(起始日期)	date 或 dateTime
endDate(结束日期)	date 或 dateTime

instant、startDate 和 endDate 元素的内容使用 ISO 8601 中定义的数据表示形式，同时 XBRL 增添了进一步的约束与限制：

——对于 periodType="instant"的数据项，period 应包含时间点元素。

——对于 periodType="duration"的数据项，时间应包含 forever 元素或一个由 startDate 和 endDate 元素构成的有效序列。

若 startDate 元素的内容为一个没有具体时间的日期，默认其值等同于一个具有相同日期并且具体时间为 T00:00:00(一天开始时的午夜时分)的日期和时间类型值。

若 endDate 元素或 instant 元素的内容为一个没有具体时间的日期，默认其值等同于一个具有相同日期，附加上 P1D，具体时间为 T00：00：00 的 dateTime 类型值。此处代表一天结束时的午夜时分。将其定义为第二天开始时的午夜时分的原因在于：SCHEMA 规范规定时间的小时字段不允许取值 24。

注：P1D 为 ISO 8601 的一个专用标签。

8.9.4 entity 元素

entity(实体)元素说明事实所描述的实体(包括商业、政府部门、个人等)。实体是上下文元素的必需内容。实体应包含标识符元素，同时可包含片段元素。其元素如表 19 所示。

表 19 entity 元素表

元素名称	选择性	描　述
identifier (标识符)	必备	identifier 元素指定了验证业务实体所需的 scheme 属性。必需的 scheme 属性包含了校验 scheme 所属命名空间的 URI，为引用命名机构提供了框架。元素的内容应是一个表征字符串类型的值，它是 scheme 属性所引用命名空间内的一个有效标识符。XBRL 并未作出如下假设：应用程序能够解析任何特定模式属性内以元素内容形式出现的标识符
segment (片段)	可选	segment 元素是可选的容器元素。当实体的标识符不足以帮助 XBRL 实例更为完整地标识商业部门时，segment 元素可作为额外标记供 XBRL 实例编制者需要使用。一般而言，segment 的具体内容与 XBRL 实例的目的相关。segment 元素中包含的子元素不能在 http://www.xbrl.org/2003/instance 命名空间中定义，同样不能是该命名空间中已定义元素的替换元素。segment 元素也不能为空

entity 元素的 XML 模式约束如下所示：

```
<schema targetNamespace = "http://www.xbrl.org/2003/instance"
  xmlns = "http://www.w3.org/2001/XML Schema"
  xmlns:xbrli = "http://www.xbrl.org/2003/instance"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
```

```
  elementFormDefault = "qualified">

  <complexType name = "contextEntityType">
    <annotation>
      <documentation>
      实体元素的类型用来描述报告实体
      请注意,模式属性是必要和不能为空
      </documentation>
    </annotation>
    <sequence>
      <element name = "identifier">
        <complexType>
          <simpleContent>
            <extension base = "token">
              <attribute name = "scheme" use = "required">
                <simpleType>
                  <restriction base = "anyURI">
                    <minLength value = "1" />
                  </restriction>
                </simpleType>
              </attribute>
            </extension>
          </simpleContent>
        </complexType>
      </element>
      <element ref = "xbrli:segment" minOccurs = "0" />
    </sequence>
  </complexType>

</schema>
```

示例：

实体标识符

示　例	含　义
<identifier scheme="http://www.sse.com.cn">SINA</identifier>	上证所股票代码为 SINA 的公司。
<identifier scheme="http://www.dnb.com">121064880</identifier>	D U N S® 号为 121064880 的公司或其附属子公司
<identifier scheme="http://www.cusip.org">G81477104</identifier>	SINA 公司的 CUSIP 为 G81477104
<identifier scheme="http://www.nacao.org.cn">12345678-9</identifier>	某非盈利组织的国家机构代码为 12345678-9

8.9.5 segment 元素

当实体标识符无法完全满足需要时，XBRL 实例的编制者可使用 segment（片段）元素更完整的描述业务片段信息。通常情况下，片段的内容根据 XBRL 实例的目的确定。segment 元素包含的子元素不能在 http://www.xbrl.org/2003/instance 命名空间中定义，同样不能作为该命名空间中已定义元素的替换元素。segment 元素不能为空。

segment 元素的 XML 模式约束如下所示：

```
〈schema targetNamespace = "http://www.xbrl.org/2003/instance"
  xmlns = "http://www.w3.org/2001/XML Schema"
  xmlns:xbrli = "http://www.xbrl.org/2003/instance"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
  elementFormDefault = "qualified"〉

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

〈/schema〉
```

示例：

segment 元素的用法

```
〈xbrl xmlns = "http://www.xbrl.org/2003/instance"
      xmlns:my = http://www.someCompany.com/segment
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation = "http://www.someCompany.com/segment
           http://www.someCompany.com/segment/segment-schema.xsd"〉
〈! --…此处至少有一个 link:schemaRef 元素…--〉
〈! --…此处为包含事实数据值的元素…--〉
    〈context id = "c1"〉
      〈entity〉
        〈! --required content --〉
        〈identifier scheme = "http://www.dnb.com"〉121064880〈/identifier〉
      〈! -- optional content --〉
        〈segment〉
          〈my:Province〉BeiJing〈/my:Province〉
        〈/segment〉
      〈/entity〉
      〈period〉〈instant〉2002-12-01〈/instant〉〈/period〉
    〈/context〉
    〈/xbrl〉
```

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

  〈simpleType name = "ProvinceType"〉
    〈restriction base = "token"〉
      〈enumeration value = "BeiJing"/〉
      〈enumeration value = "ON"/〉
    〈/restriction〉
  〈/simpleType〉

  〈element name = "Province" type = "my:ProvinceType"/〉
    〈/schema〉
```

含义：报告编制者使用了一个 segment 元素来表述此业务数据与北京相关。该公司的 XML 模式文件定义了 Province 元素，其包含中国除台湾以外的 31 个省、自治区和直辖市名称。

分类标准的创建者应预料到 XBRL 实例创建者将会定义新的元素并插入片段元素中以表达的一个或多个维度差别，例如：

- 组织结构，比如一个公司的总部和附属子公司；
- 区域划分，比如在亚洲、欧洲和北美经营业务；
- 功能差别，比如连续和非连续操作得到的结果；
- 产品差别，比如渔业林业、畜牧业和农业的业务；
- 操作差别，例如是经常性收入还是非经常性收入，是首次签约还是到续签。

文档的制定者可决定提供命名空间方面的适当支持和 xsi:schemaLocation 提示，从而确保 segment 元素能够通过 XML 模式校验。

8.9.6 scenario 元素

业务事实可通过四种形式报告：决算、预算、重估和预估。就内部报告而言，编制者希望将更多的元数据与数据项关联，为此可用 scenario(场景)元素来添加额外标记。

scenario 元素中包含的子元素不能在 http://www.xbrl.org/2003/instance 命名空间中定义，同样不能作为该命名空间中已定义元素的替换元素。scenario 元素不能为空。

scenario 元素的 XML 模式约束如下所示：

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

  <complexType name = "contextscenarioType">
    <annotation>
      <documentation>在事实被报告的情况下用于场景</documentation>
    </annotation>
    <sequence>
      <any namespace = "##other" processContents = "lax"
        minOccurs = "1" maxOccurs = "unbounded" />
    </sequence>
  </complexType>

</schema>
```

示例：

scenario 元素的使用

```
<xbrl xmlns = "http://www.xbrl.org/2003/instance"
      xmlns:fid = "http://www.someInsuranceCo.com/scenarios"
      xmlns:other = "http://www.example.com"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.someInsuranceCo.com/scenarios
          http://www.someInsuranceCo.com/scenarios/scenarios-schema.xsd
          http://www.example.com http://www.example.com/other/other-schema.xsd">
<! --…此处至少有一个 link:schemaRef 元素… -->
```

```
<!—…此处为包含事实数据值的元素…—>
    <context id = "c1">
      <entity>
    <identifier scheme = "http://www.example.com">someInsuranceCo</identifier>
      </entity>
      <scenario>
        <other:bestEstimate/>
        <fid:dwSlice>
          <fid:residence>BeiJing</fid:residence>
          <fid:nonSmoker>true</fid:nonSmoker>
          <fid:minAge>34</fid:minAge>
          <fid:maxAge>49</fid:maxAge>
        </fid:dwSlice>
      </scenario>
    </context>
    </xbrl>
```

含义：报告编制者使用了一个 scenario 元素来表述此业务数据与北京特定年龄组不吸烟人群相关。

实例的生成者还需提供对命名空间和 xsi:schemaLocation 的支持，确保 scenario 元素可通过 XML 模式验证环节的验证。

scenario 子元素和 segment 子元素的结构完全一致，但用途不同。片段用于说明业务实体的某些组成部分，而场景用于说明一组事实的度量环境。与 segment 元素相同，scenario 元素的内容也是与特定应用相关的。

8.9.7 unit 元素

unit(单位)元素指定了数值型数据项的度量单位，其内容可是由单一的 measure 元素表示的简单度量单位，也可是通过 divide 元素表示的比率单位。其中，分子单位和分母单位均由 measure 元素表示。

unit 元素的 XML 模式约束如下所示：

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

  <annotation>
    <documentation>
    对 unit 元素的起作用的 XML 模式组件
    </documentation>
  </annotation>

  <element name = "measure" type = "QName" />

  <complexType name = "measuresType">
    <annotation>
      <documentation>
      兄弟度量元素的收集
      </documentation>
    </annotation>
    <sequence>
```

```
      〈element ref = "xbrli:measure" minOccurs = "1" maxOccurs = "unbounded" /〉
    〈/sequence〉
  〈/complexType〉

  〈element name = "divide"〉
    〈annotation〉
      〈documentation〉
      用于描述比率单位
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈sequence〉
        〈element name = "unitNumerator" type = "xbrli:measuresType" /〉
        〈element name = "unitDenominator" type = "xbrli:measuresType" /〉
      〈/sequence〉
    〈/complexType〉
  〈/element〉

  〈element name = "unit"〉
    〈annotation〉
      〈documentation〉
      用来描述数值型数据项的单位信息
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈choice〉
        〈element ref = "xbrli:measure" minOccurs = "1" maxOccurs = "unbounded" /〉
        〈element ref = "xbrli:divide" /〉
      〈/choice〉
      〈attribute name = "id" type = "ID" use = "required" /〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

a) id 属性

每个 unit 元素应包含 id 属性，其取值应遵循 XML 规范中关于 ID 类型属性的规定(http://www.w3.org/TR/REC-xml#NT-TokenizedType)。id 属性能够标识单位，这样该单位就可被 item 元素引用。

b) measure 元素

measure(度量)元素的类型为 xsd:QName。

数据项对应的类型约束了 unit 元素的内容和 measure 元素的值，如表 20 所示。

表 20 基于数据项类型的单位约束

数据项类型	要求单位应包含
monetaryItemType 及其派生类型	单一的 xbrli:measure 元素的 xsd:QName 内容被约束如下：xsd:QName 的本地部分应是 ISO 4217 指定的“货币”，并且在数据项上下文中的期间元素所指定的时间范围内有效。xsd:QName 的命名空间应是 http://www.xbrl.org/2003/iso4217
sharesItemType 及其派生类型	单一的 xbrli:measure 元素的 xsd:QName 内容被约束如下：xsd:QName 的本地部分应是“shares”，并且 xsd:QName 的命名空间名称应是 http://www.xbrl.org/2003/instance

为了表示分子和分母具有相同单位的百分数或者比率，事实应使用 unitRef 属性关联到某个包含唯一 measure 子元素的单位元素。

复杂的度量单位可通过其他度量单位间的数学关系来表达，比如若干并列的兄弟 measure 元素表示乘积单位，单一的 divide 元素表示比率单位。

c) divide 元素

divide(除)元素应包含 unitNumerator 元素和 unitDenominator 元素。

d) uintNumerator 和 uniteDnominator 元素

unitNumerator(分子单位)元素和 unitDenominator(分母单位)元素应包含一个或多个 measure 元素。

单位的表示要求尽可简单。divide 元素中的 unitNumerator 元素和 unitDenominator 元素中的 measure 元素不能出现结构等价(s-equal)的情况。

示例 1：

使用单位元素

示　例	含　义
〈unit id = "u1"〉〈measure xmlns:ISO4217 = "http://www.xbrl.org/2003/iso4217"〉ISO4217:CNY〈/measure〉〈/unit〉	货币，人民币元
〈unit id = "u2"〉〈measure xmlns:ISO4217 = "http://www.xbrl.org/2003/iso4217"〉ISO4217:cny〈/measure〉〈/unit〉	货币指定采用了不正确的小写形式
〈unit id = "u1"〉〈measure〉xbrli:pure〈/measure〉〈/unit〉	一个纯小数
〈unit id = "u3"〉 〈measure〉meter〈/measure〉 〈measure〉meter〈/measure〉 〈/unit〉	平方米 - 一米乘以米
〈unit id = "u4"〉〈measure〉xbrli:shares〈/measure〉〈/unit〉	股份数
〈unit id = "u5"〉〈measure〉FTE〈/measure〉〈/unit〉	工作强度（正式员工的数量）
〈unit id = "u6"〉 〈divide〉 〈unitNumerator〉 〈measure〉ISO 4217:CNY〈/measure〉 〈/unitNumerator〉 〈unitDenominator〉 〈measure〉xbrli:shares〈/measure〉 〈/unitDenominator〉 〈/divide〉 〈/unit〉	每股所得收益(EPS)，以人民币元/每股度量

示　例	含　义
⟨unit id = "u6"⟩ ⟨divide⟩ ⟨unitNumerator⟩ ⟨measure⟩ISO 4217:CNY⟨/measure⟩ ⟨/unitNumerator⟩ ⟨unitDenominator⟩ ⟨measure⟩ISO 4217:CNY⟨/measure⟩ ⟨/unitDenominator⟩ ⟨/divide⟩ ⟨/unit⟩	不规范，divide 元素中的分子和分母中出现了相同的度量元素

示例 2：

简单度量单位和复杂度量单位

简单度量单位	复杂度量单位
⟨unit id = "u1"⟩ ⟨measure⟩meter⟨/measure⟩ ⟨/unit⟩	⟨unit id = "u4"⟩ ⟨measure⟩ meter ⟨/measure⟩ ⟨measure⟩ meter ⟨/measure⟩ ⟨/unit⟩

8.10　元组的应用

元组有复杂的内容，可包含数据项和其他的元组。tuple(元组)是抽象元素。以下规则适用于元组和元组在分类标准模式中的声明：

a) 所有元组应是以 tuple 为头元素的替换组的成员。因此，应声明元组为全局元素，因为只有全局元素才能出现在替换组中。

b) 元组在分类标准模式中的声明一定不能包含 periodType 属性和 balance 属性。

c) 可从他处(如脚注)引用元组。因此，所有元组在分类标准模式中的声明都应当(但不要求)包含一个可选的本地属性的声明，且属性的名字 id 标识符类型为 xsd:ID。扩展分类标准的创建者在限制元组的数据类型时，不应禁止已存在的 id 属性(如果分类标准的创建者并未定义或者禁止一个元组的 id 属性，那么该元组将不能通过缩写的 XPointer 被引用)。

d) 在元组声明中，属性应使用 SCHEMA 规范中定义属性，不能引用下列命名空间中定义的属性：

——http://www.xbrl.org/2003/instance;

——http://www.xbrl.org/2003/linkbase;

——http://www.xbrl.org/2003/XLink;

——http://www.w3.org/1999/xlink;

e) 元组和元组在分类标准模式中的声明均不应含有混合内容或简单内容。

f) 除"id"属性以外，元组在分类标准模式中的声明不需要明确说明本地属性。

g) 实例中元组的孩子应是元素，并且这些元素在一个以数据项或者元组为头元素的替换组里。

h) 在元组的声明中一般不定义非概念性的元组，即元组的子元素应是全局元素(分类标准的创建者不宜将非概念性元素归入元组声明的内容模型中)，在元组在分类标准模式中的声明中，该元组子元素的声明应参照替换组中的全局性元素的声明，且该替换组是以数据项或者元组为头元素。

示例 1：

将元组定义为一个“tuple”替换组成员

分类标准模式的简短示例

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

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

<element name = "managementInformation" substitutionGroup = "xbrli:tuple">
  <complexType>
    <complexContent>
      <restriction base = "anyType">
        <sequence>
          <element ref = "s:managementName"/>
          <element ref = "s:managementTitle"/>
          <element ref = "s:managementAge" minOccurs = "0"/>
        </sequence>
        <attribute name = "id" type = "ID" use = "optional"/>
      </restriction>
    </complexContent>
  </complexType>
 </element>
</schema>
```

示例 2：

分类标准的一个 XBRL 实例(未显示上下文和单位元素和链接库引用元素)

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

<! —…one link:schemaRef element MUST exist here referencing previous taxonomy…—>

  <s:managementInformation>
    <s:managementName contextRef = "c1">张三</s:managementName>
    <s:managementTitle contextRef = "c1">COB</s:managementTitle>
    <s:managementAge precision = "2" contextRef = "n1" unitRef = "u1">42</s:managementAge>
  </s:managementInformation>
  <s:managementInformation>
    <s:managementName contextRef = "c1">李四</s:managementName>
    <s:managementTitle contextRef = "c1">CEO</s:managementTitle>
  </s:managementInformation>

<! —…上下文应在此定义…—>
<! —…单位应在此定义…—>

</xbrl>
```

元组中可出现 all、sequence 和 choice 元素。

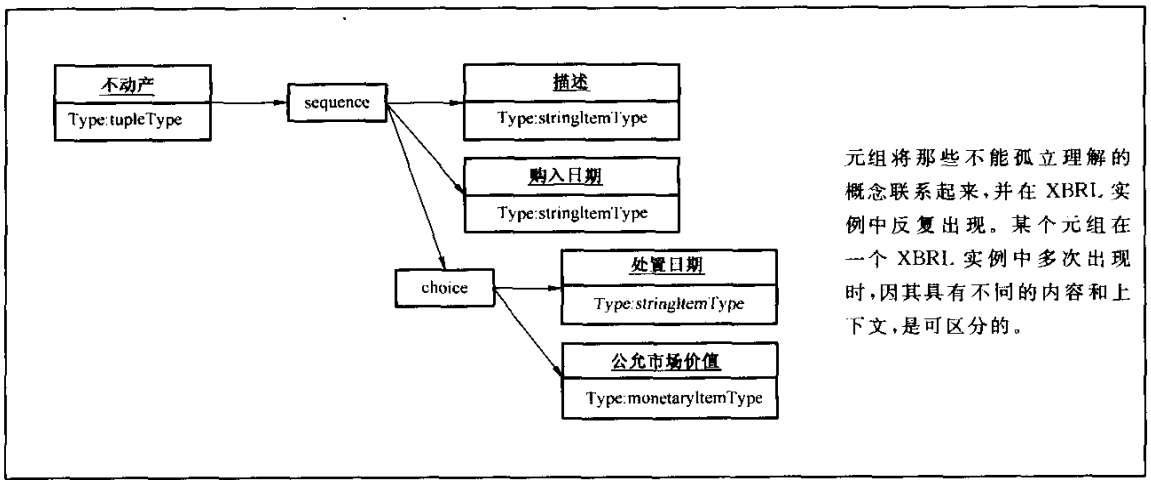
示例 3：

描述所持有及处置的商业不动产的元素

标签	元素名称	平衡	替换组
Property(不动产)	Property		元组
Property description (不动产描述)	Description		数据项
Date property acquired (不动产购入日期)	dateAcquired		数据项
Date property disposed of (不动产处置日期)	dateDisposedOf		数据项
Property fair market value (不动产的公允市场价值)	fairMarketValue		数据项

描述内容和购入日期对于任何不动产都是适用的，但公允市场价值和被处置的情况不可同时出现，即不动产或含有公允市场价值或已经被处置。

示例 4：元组中的层级结构



元组的内容模型仅能通过 XML 模式予以定义。任何 XBRL 链接库都不能定义或修改元组的内容模型。

8.11 等价性谓词表

等价性谓词表是检查数据项和元组是否是重复的谓词词表。

“等价”一词在 XBRL 实例中与重复性的检查相关。等价词表里都是对称的谓词，不同的等价谓词形式也不同，并且用递归方法定义，如表 21 所示。

表 21 等价谓词的解释

参数类型	谓 词	定 义
节点(node)	identical(同一)	完全相同的 XML 节点
序列(sequence)	s-equal(结构等价) v-equal(值等价) c-equal(上下文等价) u-equal(单位等价)	序列中的每个节点均与另一序列中处于相同位置的节点{结构等价,值等价,上下文等价,单位等价}
集合(set)	identical(同一) s-equal(结构等价) v-equal(值等价) c-equal(上下文等价) u-equal(单位等价)	集合 X 与集合 Y 满足{同一,结构等价,值等价,上下文等价,单位等价},当集合 X 里的每个节点都能在集合 Y 中找到某个节点与之{同一,结构等价,值等价,上下文等价,单位等价},并且两个集合中的元素数目相同
任何 XML 对象(any XML object)	x-equal (XML 路径等价)	XML 对象 A 与 XML 对象 B XML 路径等价,当 XPATH 表达式 A = B 返回 true
文本(text)	s-equal(结构等价)	两个文本字符串
属性(attribute)	s-equal(结构等价)	两个属性的本地名字和命名空间是结构等价的,数值是 XML 路径等价的
元素(element) (本表所列的单独处理的元素除外)	s-equal(结构等价)	不同一,但它们的元素本地名称和命名空间是结构等价的,而且其属性集是结构等价的,文本和子元素的内容也是结构等价的
实体(entity)	s-equal(结构等价)	identifier 元素是结构等价的,segment 元素是结构等价的(如果没有 segment 元素,则认为是与空的 segment 元素结构等价)
起始日期(startDate)	s-equal(结构等价)	包含的日期/时间是相等的
结束日期(endDate)	s-equal(结构等价)	包含的日期/时间是相等的
时间点(instant)	s-equal(结构等价)	包含的日期/时间是相等的
时间(period)	s-equal(结构等价)	当下列条件之一成立时: a) 两个元素均包含 forever 子元素; b) 它们的 instant 子元素是结构等价的; c) 它们的 startDate 子元素和 endDate 子元素是结构等价的
单位(unit)	s-equal(结构等价)	divide 子元素和 measure 元素集是结构等价的
除(divide)	s-equal(结构等价)	unitNumerator 和 unitDenominator 元素是结构等价的
单位分子 (unitNumerator)	s-equal(结构等价)	measure 子元素集是结构等价的
单位分母(unitDenominator)	s-equal(结构等价)	measure 子元素集是结构等价的
度量(measure)	s-equal(结构等价)	两个度量元素内容的命名空间前缀指向同一个命名空间,并且这两个度量元素内容的本地名字是结构等价的
上下文(context)	s-equal(结构等价)	period 元素、entity 元素、scenario 元素均是结构等价的

表 21 等价谓词的解释（续）

参数类型	谓 词	定 义
数据项(item)	s-equal(结构等价)	它们是上下文等价和单位等价的，它们的 precision 属性和 decimals 是结构等价的，将数值型数据项的值转换成十进制表示之后，其内容文本是结构等价的
元组(tuple)	s-equal(结构等价)	(数据项和元组)子元素的集合是结构等价的
可用性(used()n)	s-equal(结构等价)	两个 used()n 元素内容的命名空间前缀指向同一命名空间，并且这两个 used()n 元素内容的本地名字是结构等价的
数据项(item)	p-equal(父元素等价)	节点具有相同的父元素
元组(tuple)	pequal(父元素等价)	节点具有相同的父元素
数据项(item)	c-equal(上下文等价)	它们的 contextRef 属性标识同一或结构等价的上下文
任何一对数值型数据项 (any pair of numeric items)	u-equal(单位等价)	当且仅当下列所有条件成立，数值型数据项 X 和 Y 是单位等价的： a) UX 的子孙元素 unitNumerator 的集合与 UY 的子孙元素 unitNumerator 的集合是结构等价的； b) UX 的子孙元素 unitDenominator 的集合与 UY 的子孙元素 unitDenominator 的集合是结构等价的； c) UX 的子元素 measure 的集合与 UY 的子元素 measure 的集合是结构等价的。 UX 代表 X 的 unitRef 属性引用的 unit 元素，UY 代表 Y 的 unitRef 属性引用的 unit 元素
任何一对非数值型数据项 (any pair of non-numeric items)	u-equal(单位等价)	真
一个数值型数据项和一个非数值型数据项 (one numeric item and one non-numeric items)	u-equal(单位等价)	假
数值型数据项，但类型不是 fractionItemType 类型或其受到某种限制后派生出来的类型 (numeric items not of type fractionItemType or a type derived from fractionItemType by restriction)	v-equal(值等价)	当且仅当下列所有条件成立，A 和 B 是值等价的： a) A 和 B 既是上下文等价的也是单位等价的； b) 数值 AN 和 BN 是 XML 路径等价的，A 的数值保留 N 位有效数字后舍入得到 AN，B 的数值保留 N 位有效数字后舍入得到 BN。 其中，N 是以下两个数中的较小值： 1) A 的指定精度或推断精度； 2) B 的指定精度或推断精度。
数值型数据项，类型为 fractionItemType 或其受到某种限制后派生出来的类型 (numeric items of type fractionItemType or a type derived from fractionItemType by restriction)	v equal(值等价)	当且仅当以下的所有条件成立时，A 和 B 是值等价的： a) A 和 B 既是上下文等价的又是单位等价的； b) AN 与 BN 和 AD 与 BD 都是 XML 路径等价的。 其中 AN 是 A 的正规式的分子，AD 是分母，BN 是 B 的正规式的分子，BD 是分母。 对于类型为 fractionItemType 或其受到限值派生出来的类型的任意数值型数据项 F，其正规式的分子 FN 和分母 FD 均为整数，并且没有公因子。此外，存在整数 H，将 FN 乘以 H 得到 F 的分子，将 FD 乘以 H 得到 F 的分母

表 21 等价谓词的解释（续）

参数类型	谓 词	定 义
数值型数据项，其中之一的类型为 fractionItemType 或其受限派生类型，而另一数值型数据项则不是 (numeric items, one of which is of type fractionItemType or a type derived from fractionItem Type by restriction and the other of which is not)	v-equal(值等价)	对于数值型数据项的这样一种结合，值等价的值通常为假
非数值型数据项(non-numeric items)	v-equal(值等价)	当且仅当下列所有条件都成立时，A 和 B 是值等价的： a) A 和 B 是上下文等价的 b) 〔XPATH〕标准化空间(AC) = 标准化空间(BC) 其中，AC 代表 A 的内容，BC 代表 B 的内容
数据项(item)	duplicate(重复)	当且仅当下列所有条件均成立时，数据项 X 和 Y 是重复的： a) X 和 Y 不同一； b) X 元素的本地名字与 Y 元素的本地名字是结构等价的； c) X 和 Y 定义在同一命名空间； d) X 和 Y 是父元素等价的； e) X 和 Y 是上下文等价的； f) X 和 Y 是单位等价的
元组(tuple)	duplicate(重复)	当且仅当下列所有条件均满足时，元组 X 和 Y 是重复的： a) X 和 Y 不同一； b) X 元素的本地名字与 Y 元素的本地名字结构等价； c) X 和 Y 定义于同一个命名空间； d) X 和 Y 是父元素等价的； e) 对于 X 的子元组组成的集合中的每个节点 A，都能在 Y 的子元组组成的集合中找到节点 B，并且 A 和 B 满足除父元素等价以外的重复元组的所有需求； f) X 和 Y 有着相同数目的子元组； g) 对于 X 的子数据项组成的集合中的每个节点 A，都能在 Y 的子数据项组成的集合中找到节点 B，并且 A 和 B 满足除父元素等价外的重复数据项的所有需求； h) X 和 Y 有着相同数目的子数据项

以下的扩展示例针对每个谓词列示了正例和反例。

示例 1：

重复的数据项，元组和上下文

元　素	一个包含两个结构等价的上下文元素和双重嵌套元组的 XBRL 实例 （左列列举了若干元素）
	⟨xbrl xmlns = "http://www.xbrl.org/2003/instance" xmlns:s = "http://mycompany.com/xbrl/Taxonomy" xmlns:xbrli = "http://www.xbrl.org/2003/instance" xmlns:xsi = "http://www.w3.org/2001/XML Schema - instance"⟩
a analysis	⟨s:analysis⟩
b customer	⟨s:customer⟩
b name	⟨s:name contextRef = "np3"⟩SINA⟨/s:name⟩
b gross	⟨s:gross unitRef = "u3" contextRef = "np3" precision = "4"⟩3001⟨/s:gross⟩
b returns	⟨s:returns unitRef = "u3" contextRef = "np3" precision = "3"⟩100⟨/s:returns⟩
	⟨s:net unitRef = "u3" contextRef = "np3" precision = "4"⟩2900⟨/s:net⟩
	⟨/s:customer⟩
	⟨s:customer⟩
c customer	⟨s:name contextRef = "Xnnp3X"⟩SINA⟨/s:name⟩
c name	⟨s:gross unitRef = "u3" contextRef = "np3" precision = "3"⟩3000⟨/s:gross⟩
c gross	⟨s:returns unitRef = "u3" contextRef = "np3" precision = "3"⟩100⟨/s:returns⟩
	⟨s:net unitRef = "u3" contextRef = "np3" precision = "4"⟩2900⟨/s:net⟩
	⟨/s:customer⟩
	⟨s:customer⟩
	⟨s:name contextRef = "np3"⟩SINA⟨/s:name⟩
d customer	⟨s:gross unitRef = "u3" contextRef = "np3" precision = "4"⟩3000⟨/s:gross⟩
	⟨s:returns unitRef = "u3" contextRef = "np3" precision = "3"⟩500⟨/s:returns⟩
	⟨s:net unitRef = "u3" contextRef = "np3" precision = "4"⟩2500⟨/s:net⟩
d returns	⟨/s:customer⟩
	⟨s:customer⟩
	⟨s:name contextRef = "np3"⟩Bree⟨/s:name⟩
	⟨s:name contextRef = "Xnnp3X"⟩Bree⟨/s:name⟩
	⟨s:gross unitRef = "u3" contextRef = "np3" precision = "4"⟩3000⟨/s:gross⟩
e customer	⟨s:returns unitRef = "u3" contextRef = "np3" precision = "3"⟩200⟨/s:returns⟩
f name	⟨s:net unitRef = "u3" contextRef = "np3" precision = "4"⟩2800⟨/s:net⟩
g name	⟨/s:customer⟩
	⟨s:totalGross unitRef = "u3" contextRef = "np3" precision = "3"⟩12000⟨/s:totalGross⟩
	⟨/s:analysis⟩
	⟨! --个冗余的上下文 Xnnp3X = period,2003 --⟩
	⟨context id = "np3"⟩
h totalGross	⟨entity⟩
	⟨identifier scheme = "http://www.nasdaq.com"⟩SAMP⟨/identifier⟩
	⟨/entity⟩
	⟨period⟩
np3	⟨startDate⟩2003-01-01⟨/startDate⟩
	⟨endDate⟩2003-12-31⟨/endDate⟩
	⟨/period⟩
	⟨/context⟩
	⟨unit id = "u3"⟩⟨measure⟩ISO4217:USD⟨/measure⟩⟨/unit⟩

元　素	一个包含两个结构等价的上下文元素和双重嵌套元组的 XBRL 实例 （左列列举了若干元素）
u3 Xnnp3X	⟨context id = "Xnnp3X"⟩ ⟨entity⟩ ⟨identifier scheme = "http://www.nasdaq.com"⟩SAMP⟨/identifier⟩ ⟨/entity⟩ ⟨period⟩ ⟨startDate⟩2003-01-01⟨/startDate⟩ ⟨endDate⟩2003-12-31⟨/endDate⟩ ⟨/period⟩ ⟨/context⟩ ⟨/xbrl⟩
注：尽管本例缺少计算链接库，但“h totalGross”的总和 12 000 仍然是最精确的数值，其计算方法是四位顾客的总额之和(3 001＋3 000＋3 000＋3 000＝12 001，但由于 c 的 precision 为 3，该数据仅能保留三位有效数字，为 12 000)。	

示例 2：

检测重复的谓词

节点 1	节点 2	类型	谓词	是否为真	原　因
np3	Xnnp3X	上下文	Identical	否	不同的节点
np3	Xnnp3X	上下文	s-equal	是	entity 和 period 是结构等价的
f name	g name	数据项	s-equal	是	尽管结构等价，但具有不同上下文标识符 np3 和 Xnnp3X
f name	g name	数据项	p-equal	是	相同的父元素
f name	g name	数据项	c-equal	是	相同内容的上下文元素 np3 和 Xnnp3X
f name	g name	数据项	v-equal	是	相同的内容“Bree”
f name	g name	数据项	duplicates	是	父元素等价和上下文等价
b name	c name	数据项	s-equal	是	尽管结构等价，但具有不同上下文标识符 np3 和 Xnnp3X
b name	c name	数据项	p-equal	否	它们位于不同的顾客元组中
b name	c name	数据项	c-equal	是	相同内容的上下文元素 np3 和 Xnnp3X
b name	c name	数据项	v-equal	是	相同的“SINA”内容
b name	c name	数据项	duplicates	否	不是父元素等价的，所以，即使值等价也没有关系
b gross	c gross	数据项	s equal	否	
b gross	c gross	数据项	p-equal	否	不同的父元素
b gross	c gross	数据项	c-equal	是	它们都有上下文 np3 和 u3 单位

节点 1	节点 2	类型	谓词	是否为真	原　因
b gross	c gross	数据项	v-equal	是	"3 001"保留三位精度，等于"3 000"
b gross	c gross	数据项	duplicates	否	不是父元素等价，所以，即使值等价也没有关系
b ustomer	c customer	元组	s-equal	否	不同的上下文标识符：np3 和 Xnnp3X
b customer	c customer	元组	p-equal	是	相同的父元素"a analysis"
b customer	c customer	元组	c-equal	不适用	上下文等价并不适用于元组
b customer	c customer	元组	v equal	不适用	值等价并不适用于元组
b customer	c customer	元组	duplicates	是	是父元素等价的，并且子数据项 name，gross，returns 和 net 互相是值等价的
b returns	d returns	数据项	s-equal	否	不同的值
b returns	d returns	数据项	p-equal	否	父元素是 b customer 和 d customer
b returns	d returns	数据项	c-equal	是	均包含上下文 np3 和 u3 单位
b returns	d returns	数据项	v-equal	否	b 的值为 100，d 值为 500
b returns	d returns	数据项	duplicates	否	不是父元素等价的，所以，即使值等价也没有关系
b customer	d customer	元组	s-equal	否	returns 和 net 不同值
b customer	d customer	元组	p-equal	是	相同的父元素"a analysis"
b customer	d customer	元组	c-equal	不适用	上下文等价不适用于元组
b customer	d customer	元组	v-equal	不适用	值等价不适用于元组
b customer	d customer	元组	duplicates	否	是父元素等价的，并且子数据项 b name 和 b gross 与 d name 和 d gross 是值等价的，但子数据项 b returns 和 b net 与 d returns 和 d net 并非值等价

重复数据项定义中的等价谓词属于等价定位，而不是等价内容。就此进行比较时，contextRef、unitRef、precision 和 decimals 之外的属性都应忽略。

判断两个数值型数据项是否值等价（该谓词在多种其他等价类型的谓词定义中出现和使用）时，应将两个数值型数据项的 precision 值考虑在内。

8.12　脚注

XBRL 使用 footnoteLink（脚注链接）元素来描述 XBRL 实例中事实之间的不规则结构化的关联。

footnoteLink 元素是一个扩展链接，它包含定位器、资源和描述 XBRL 实例中事实之间的不规则关系的弧。

footnoteLink 元素的 XML 模式约束如示例 1 和示例 2 所示：

示例 1：

XBRL 实例中的脚注

```
<schema targetNamespace = "http://www.xbrl.org/2003/linkbase"
  xmlns = "http://www.w3.org/2001/XML Schema"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
  xmlns:xl = "http://www.xbrl.org/2003/XLink"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  elementFormDefault = "qualified">
  <element name = "footnoteLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
      footnootLink 链接元素的定义
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base = "xl:extendedType">
          <choice minOccurs = "0" maxOccurs = "unbounded">
            <element ref = "xl:title"/>
            <element ref = "link:documentation"/>
            <element ref = "link:loc"/>
            <element ref = "link:footnoteArc"/>
            <element ref = "link:footnote"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>
</schema>
```

示例 2：

XBRL 实例中的脚注

```
<? xml version = "1.0" encoding = "UTF-8"?>
<xbrl
  xmlns = "http://www.xbrl.org/2003/instance"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xmlns:fr = "http://www.xbrl-fr.org/xbrl/2003-02-29"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:ISO4217 = "http://www.xbrl.org2003/2003/iso4217"
  xsi:schemaLocation = "http://www.xbrl-fr.org/xbrl/2003-02-29 fr.xsd">

  <link:schemaRef xlink:type = "simple" xlink:href = "fr.xsd"/>

  <fr:propertyPlantEquipmentGross precision = "4" unitRef = "u1" contextRef = "c1">1200</fr:propertyPlantE-
quipmentGross>
  <fr:assetsTotal id = "f1" precision = "4" unitRef = "u1" contextRef = "c1">2600</fr:assetsTotal>
  <fr:equityTotal id = "f3" precision = "4" unitRef = "u1" contextRef = "c1">1100</fr:equityTotal>
  <fr:liabilitiesTotal id = "f2" precision = "4" unitRef = "u1" contextRef = "c1">2600</fr:liabilitiesTotal>

  <link:footnoteLink
    xlink:type = "extended" xlink:title = "1"
   xlink:role = "http://www.xbrl.org/2003/role/link">
   <link:footnote
```

```
      xlink:type = "resource"
      xlink:label = "footnote1"
      xlink:role = "http://www.xbrl.org/2003/role/footnote"
      xml:lang = "en">Including the effects of the merger.</link:footnote>
    <link:footnote
      xlink:type = "resource"
      xlink:label = "footnote1"
      xlink:role = "http://www.xbrl.org/2003/role/footnote"
      xml:lang = "fr">Y compris les effets de la fusion.</link:footnote>
    <link:loc xlink:type = "locator" xlink:label = "fact1" xlink:href = "#f1"/>
    <link:loc xlink:type = "locator" xlink:label = "fact1" xlink:href = "#f2"/>
    <link:loc xlink:type = "locator" xlink:label = "fact1" xlink:href = "#f3"/>
    <link:footnoteArc
     xlink:type = "arc"
     xlink:from = "fact1" xlink:to = "footnote1"
     xlink:title = "view explanatory footnote"
     xlink:arcrole = "http://www.xbrl.org/2003/arcrole/fact-footnote"/>
  </link:footnoteLink>
  <context id = "c1">
    <entity>
      <identifier scheme = "http://www.un.org/">Example plc</identifier>
    </entity>
    <period>
      <instant>2001-08-16</instant>
    </period>
    <scenario name = "Actual values">
      <fr:scenarioType>actual</fr:scenarioType>
    </scenario>
  </context>
  <unit id = "u1"><measure>ISO4217:EUR</measure></unit>
</xbrl>
```

含义：一个 footnoteArc 将三个事实链接到两个脚注。这两个脚注使用不同的语言。*footnoteArc* 元素使用 *xlink:title* 属性来说明通过该事实可访问的资源的本质。

a) footnoteArc 元素的定位器

footnoteArc 元素包含的定位器应是 loc 元素。当在 footnoteLink 元素中使用 loc 元素时，它应且仅能指向同一 XBRL 实例中包含该 loc 元素的数据项或元组。

b) 脚注元素(footnote)

footnote 元素是 footnoteLink 元素中允许出现的唯一资源。footnote 资源的内容相对于普通资源来说是受限的。尤其是 footnote 资源可有混合内容，其中包含一个简单的字符串或一个 XHTML 片段，或者二者兼而有之。

footnote 元素的 XML 模式约束如示例 3 所示：

示例 3：

footnote 元素的 XML 模式约束

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

```
  elementFormDefault = "qualified">

  <element name = "footnote" substitutionGroup = "xl:resource">
    <annotation>
      <documentation>
      引用资源元素的定义
      </documentation>
    </annotation>
    <complexType mixed = "true">
      <complexContent mixed = "true">
        <extension base = "xl:resourceType">
          <sequence>
            <any namespace = "http://www.w3.org/1999/xhtml"
              processContents = "skip" minOccurs = "0" maxOccurs = "unbounded"/>
          </sequence>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
            processContents = "lax" />
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

所有 footnote 资源应包含 xml:lang 属性，该属性用于校验脚注内容所使用的语言。xml:lang 属性的值应遵循相应的 XML 规范。

c) 脚注弧(footnoteArc)元素

footnoteArc 元素具有与普通扩展链接弧相同的语法，其属性如表 22 所示：

表 22 footnoteArc 元素的属性表

属性名称	选择性	描 述
xlink:arcrole		xlink:arcrole 属性的值应是一个能够表明弧含义的 URI。 该弧角色值被从数据项或元组定位器到脚注资源的某个 footnoteArc 元素所用，旨在表明该 footnote 传递了关于事实的可读信息
xlink:title	可选	xlink:title 属性用于向那些在事实和脚注之间来回转换的用户传达关于事实和相关脚注之间关系的信息，其内容应是一个字符串，并且对于使用 XLINK 规范的应用程序的用户是可视的

footnoteArc 元素的 XML 模式约束如示例 4 所示：

示例 4：

footnoteArc 元素的 XML 模式约束

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

  <element name = "footnoteArc" type = "xl:arcType" substitutionGroup = "xl:arc">
```

```
    〈annotation〉
      〈documentation〉
      在 footnootLink 链接中具体使用的弧
      〈/documentation〉
    〈/annotation〉
  〈/element〉

〈/schema〉
```

9 XBRL 分类标准

9.1 分类标准模式

9.1.1 概述

本部分中，分类标准中出现的 XML 模式文件被称为“分类标准模式”。分类标准应包含分类标准模式。分类标准模式应是某个 XML 模式的有效实例。

如果一个分类标准中包含可扩展链接，该分类标准模式应包含一个指向相应链接库的 linkbaseRef 元素，或者该可扩展链接应嵌入到分类标准模式本身所包含的链接库中。

分类标准模式不一定非要导入 XBRL 实例模式文件(例如，在上下文中，仅仅为了定义 segment 或 scenario 的语法)。然而，如果分类标准模式要定义概念(在数据项或者元组替换组中的元素)，它应引入 XBRL 实例模式文件 xbrl-instance-2003-12-31.xsd，这是因为 XBRL 实例模式文件定义了抽象元素数据项和元组。

分类标准模式应该指定一个目标命名空间，如果一个目标命名空间值被指定，其目标命名空间的值不能为空。当创建分类标准模式时，应包含其他模式的命名空间声明，比如 XML 模式文件本身的命名空间。

示例：

一个框架分类标准模式显示对链接库的引用

```
〈schema
  targetNamespace = "http://www.mycompany.com/taxonomy/2003-10-19"
  xmlns = "http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli = "http://www.xbrl.org/2003/instance"
  xmlns:link = "http://www.xbrl.org/2003/linkbase"
  xmlns:ci = "http://www.mycompany.com/taxonomy/2003-10-19"
  xmlns:xlink = "http://www.w3.org/1999/xlink"〉
    〈annotation〉
    〈appinfo〉
      〈link:linkbaseRef
        xlink:type = "simple"
        xlink:href = "linkbase_presentation.xml"
        xlink:role = "http://www.xbrl.org/2003/role/presentationLinkbaseRef"
        xlink:arcrole = "http://www.w3.org/1999/xlink/properties/linkbase"/〉
      〈link:linkbaseRef
        xlink:type = "simple"
        xlink:href = "linkbase_calculation.xml"
        xlink:role = "http://www.xbrl.org/2003/role/calculationLinkbaseRef"
        xlink:arcrole = "http://www.w3.org/1999/xlink/properties/linkbase"/〉
      〈link:linkbaseRef
```

```
        xlink:type = "simple"
        xlink:href = "linkbase_definition.xml"
        xlink:role = "http://www.xbrl.org/2003/role/definitionLinkbaseRef"
        xlink:arcrole = "http://www.w3.org/1999/xlink/properties/linkbase"/>
      <link:linkbaseRef
        xlink:type = "simple"
        xlink:href = "linkbase_label.xml"
        xlink:role = "http://www.xbrl.org/2003/role/labelLinkbaseRef"
        xlink:arcrole = "http://www.w3.org/1999/xlink/properties/linkbase"/>
      <link:linkbaseRef
        xlink:type = "simple"
        xlink:href = "linkbase_reference.xml"
        xlink:role = "http://www.xbrl.org/2003/role/referenceLinkbaseRef"
        xlink:arcrole = "http://www.w3.org/1999/xlink/properties/linkbase"/>
    </appinfo>
  </annotation>
<import
  namespace = "http://www.xbrl.org/2003/instance"
  schemaLocation = "http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd"/>

<! --…taxonomy elements declaration starts here…-->

</schema>
```

创建的 XBRL 分类标准可引用其他的分类标准，可扩展性是 XBRL 的关键特色。XBRL 分类标准模式可引用其他分类标准模式和附加 XBRL 链接库，以达到其可扩展性。在引用链接库时，分类标准模式也可自定义角色值以及弧角色值。

9.1.2 关于概念

在分类标准模式文件中，元素是用来描述概念的。为了和概念定义保持一致，模式文件的元素定义应指定元素的名称、替换组以及类型。在一个分类标准模式中的元素名称都应是唯一的，元素应是数据项或元组替换组的成员。

元素在语法上应有 id 属性，以简化链接库 loc 元素中 xlink:href 属性的内容。一些 XML 模式文件验证器需要所有 id 属性的唯一性，这些 id 属性是出现在分类标准模式以及所有直接或间接引入或包含的 XML 模式中。为了提高对 XML 模式文件解析的鲁棒性，应该注意限制 id 属性值与相关模式中 id 属性值发生冲突的可性。在下面的例子中，通过在元素名称前面加一个附加字符串“ci_”来避免冲突。

示例 1：

在分类标准模式中典型的元素定义

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

<! -…in this the example unused namespaces declarations
        are missing at root element…-->

<! -…linkbases and imports go here…->

  <element
```

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

定义了两个概念，一个与 preferredDividends(优先股息) 元素相关，另一个与 stockBasedCompensationPolicy(股权激励政策)元素相关。在 XBRL 实例中这两个值都可用空值来表达。在 XBRL 实例中优先股息概念在它的上下文中应表现为一个时间段的数值型数据项(numeric item)，股权激励政策概念在它的上下文中应表现为一个时间点的非数值型数据项。

XBRL 还定义了 periodType(时期类型)和 balance(借贷方向)属性，可用于元素的语法定义，如表 23 所示。

表 23 概念的属性表

属性名称	选择性	描　述
periodType (时间类型)		时间类型属性应用在数据项替换组的元素上。时间类型属性的值为时间点时，表示当该元素用在一个 XBRL 实例时应与一个时间是时间点的上下文相关。值为时间段时，表示当该元素用在一个 XBRL 实例时应与一个时间是时间段的上下文相关，并使用起始日期和结束日期元素或者永久元素来表达
balance (借贷方向)	可选	balance 属性只能在货币项类型或者货币项类型的派生类型的元素中进行定义。 balance 属性在消费数目与资产、负债、权益、收入、费用等会计概念相关的应用中非常重要。当提供了借方/贷方指标时，balance 属性为如何授权和解释 XBRL 实例中的值提供了权威性的说明

XML 模式文件约束了时间类型属性，如下所示：

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

  <attribute name = "periodType">
    <annotation>
      <documentation>
      时间类型属性(限制 XBRL 数据项的时间)
      </documentation>
    </annotation>
    <simpleType>
      <restriction base = "token">
```

```
        〈enumeration value = "instant" /〉
        〈enumeration value = "duration" /〉
      〈/restriction〉
    〈/simpleType〉
  〈/attribute〉

〈/schema〉
```

示例 2：

时间点和时间段概念定义

```
〈element id = "a1" name = "changeInRetainedEarnings"
         xbrli:periodType = "duration"
           type = "xbrli:monetaryItemType" substitutionGroup = "xbrli:item"/〉
〈element id = "a2" name = "fixedAssets"
         xbrli:balance = "debit"
         xbrli:periodType = "instant"
         type = "xbrli:monetaryItemType" substitutionGroup = "xbrli:item"/〉
```

balance 属性的 XML 模式约束如下所示：

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

  〈attribute name = "balance"〉
    〈annotation〉
      〈documentation〉
      balance 属性(约束计算关系)
      〈/documentation〉
    〈/annotation〉
    〈simpleType〉
      〈restriction base = "token"〉
        〈enumeration value = "debit" /〉
        〈enumeration value = "credit" /〉
      〈/restriction〉
    〈/simpleType〉
  〈/attribute〉

〈/schema〉
```

示例 3：

应用 balance 元素来表示标准的借方和贷方余额

```
      〈element
      id = "netIncome" name = "netIncome" xbrli:balance = "credit"
      xbrli:periodType = "duration"
      type = "xbrli:monetaryItemType" substitutionGroup = "xbrli:item"/〉
    〈element
      id = "fixedAssets" name = "fixedAssets" xbrli:balance = "debit"
      xbrli:periodType = "instant"
      type = "xbrli:monetaryItemType" substitutionGroup = "xbrli:item"/〉
```

表 24 实例中的正确符号

分类标准元素	账户类型	XBRL 实例元素符号
balance="credit"	贷方	正或零
balance="credit"	借方	负或零
balance="debit"	借方	正或零
balance="debit"	贷方	负或零

在 XBRL 实例中，一个借方/贷方数据项的数字表达通常是正的。

示例 4：

在一个 XBRL 实例中有正值概念和负值概念

```
<my:netIncome precision = "3" unitRef = "u1"
      contextRef = "c1">500</my:netIncome>
<my:netIncome precision = "3" unitRef = "u1"
      contextRef = "c2">-200</my:netIncome>
```

在不同的上下文中，一个利润是 500，一个亏损是 200。

在计算弧元素中，对 balance 属性的赋值限制了合法的权值，如表 25 所示。

表 25 balance 属性和计算弧权值的限制

"from" 项的 balance 属性	"to" 项的余额属性	计算弧上权值属性的非法值
借方	借方	负(< 0)
借方	贷方	正(> 0)
贷方	借方	正(> 0)
贷方	贷方	负(< 0)

所有的数据项类型应是以下所列的类型或者是由其通过约束派生的类型(参见附录 D)。XBRL 所提供的基本类型覆盖 XML 模式文件内嵌类型的子集 SCHEMA 规范(基类型或者派生类型)以及 XBRL 指定 4 种类型与领域空间相关的(货币项类型，股份项类型，纯数据项类型，分数项类型)，因此明确地定义了 XBRL 的命名空间。除了分数项类型外，所有类型都有简单的内容，所以在一个分类标准中不可包含复杂内容，除非它是通过分数项类型来约束。

数据项类型或元组类型应指定在适当的分类标准模式中，如表 26 所示，数据类型如表 27 所示。

表 26 已定义的数据项类型

XBRL 数据项类型	基本类型	*unitRef*(单位引用)属性	数值类型说明
decimalItemType	Decimal	yes	
floatItemType	Float	yes	
doubleItemType	Double	yes	
integerItemType	Integer	yes	基于从十进制受限派生的 XML 模式的内置类型

表 26 已定义的数据项类型（续）

XBRL 数据项类型	基本类型	unitRef(单位引用)属性	数值类型说明
nonPositiveIntegerItemType	nonPositiveInteger	yes	基于从十进制受限派生的 XML 模式的内置类型
negativeIntegerItemType	negativeInteger	yes	基于从十进制受限派生的 XML 模式的内置类型
longItemType	Long	yes	基于从十进制受限派生的 XML 模式的内置类型
intItemType	Int	yes	基于从十进制受限派生的 XML 模式的内置类型
shortItemType	Short	yes	基于从十进制受限派生的 XML 模式的内置类型
byteItemType	Byte	yes	基于从十进制受限派生的 XML 模式的内置类型
nonNegativeIntegerItemType	nonNegativeInteger	yes	基于从十进制受限派生的 XML 模式的内置类型
unsignedLongItemType	unsignedLong	yes	基于从十进制受限派生的 XML 模式的内置类型
unsignedIntItemType	unsignedInt	yes	基于从十进制受限派生的 XML 模式的内置类型
unsignedShortItemType	unsignedShort	yes	基于从十进制受限派生的 XML 模式的内置类型
unsignedByteItemType	unsignedByte	yes	基于从十进制受限派生的 XML 模式的内置类型
positiveIntegerItemType	positiveInteger	yes	基于从十进制受限派生的 XML 模式的内置类型
monetaryItemType	xbrli:monetary	yes	与 XBRL 指定的领域空间特别相关，因此，除了 XML schema 内置的类型，该数值类型也被包含在内
sharesItemType	xbrli:shares	yes	与 XBRL 指定的领域空间特别相关，因此，除了 XML schema 内置的类型，该数值类型也被包含在内
pureItemType	xbrli:pure	yes	与 XBRL 指定的领域空间特别相关，因此，除了 XML schema 内置的类型，该数值类型也被包含在内

表 26 已定义的数据项类型（续）

XBRL 数据项类型	基本类型	unitRef(单位引用)属性	数值类型说明
fractionItemType	具有小数分子和非零小数分母的复杂类型(xbrli:nonZeroDecimal)	yes	与 XBRL 指定的领域空间特别相关，因此，除了 XML schema 内置的类型，该数值类型也被包含在内
stringItemType	String	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
booleanItemType	Boolean	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
hexBinaryItemType	hexBinary	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
base64BinaryItemType	base64Binary	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
anyURIItemType	anyURI	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
QNameItemType	Qname	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
durationItemType	Duration	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
dateTimeItemType	xbrli:dateUnion	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
timeItemType	Time	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
dateItemType	Date	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来

表 26 已定义的数据项类型（续）

XBRL 数据项类型	基本类型	unitRef(单位引用)属性	数值类型说明
gYearMonthItemType	gYearMonth	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
gYearItemType	gYear	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
gMonthDayItemType	gMonthDay	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
gDayItemType	gDay	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
gMonthItemType	gMonth	no	该类型基于 XML 模式的内置类型，而这些内置类型不是由十进制或者字符串派生而来
normalizedStringItemType	normalizedString	no	基于由字符串受限派生而来的 XMLschema 的内置类型
tokenItemType	Token	no	基于由字符串受限派生而来的 XMLschema 的内置类型
languageItemType	Language	no	基于由字符串受限派生而来的 XMLschema 的内置类型
NameItemType	Name	no	基于由字符串受限派生而来的 XMLschema 的内置类型
NCNameItemType	NCName	no	基于由字符串受限派生而来的 XMLschema 的内置类型

示例 5：

派生一个枚举数据项类型

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

    <import
      namespace = "http://www.xbrl.org/2003/instance"
```

```
    schemaLocation = "http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd"/>

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

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

在分类标准模式中唯一允许的方法是从 XBRL 提供的数据项类型中受限派生新的数据项类型。示例 5 定义了 stateProvinceType 类型，并用来定义一个 segment 的子元素。这里，定义一个在公司自己分类标准中的 XBRL 概念。注意，没有使用前面定义的简单类型。

表 27　数据类型参照表

数据类型	内　容
monetary(货币)，*shares*(股份)和 pure(纯)数据类型	XBRL 实例模式文件定义了货币数据类型，它专指 XML 模式文件的十进制类型。XBRL 分类标准中表示所有货币值的数值元素应使用货币项类型或者由它派生的数据类型；股份类型表示基于股份的值；纯数据表示增长率、百分比以及其他量度。此处隐含地认为分子与分母具有相同的单位
fractionItemType (分数项)数据类型	XBRL 为了能够表达精确的值，提供了一种分数项数据类型。分数项数据类型中的数据值应是精确的。即分数项的分子和分母不能是根据精度推导规则推导出，而应在分数式中以原精度直接、完整表达的值，不能出现精度和小数属性

XML 模式文件对货币、股份和纯数据类型的定义如下所示：

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

    <annotation>
      <documentation>
      定义简单类型作为数据项类型的基类
      </documentation>
    </annotation>

    <simpleType name = "monetary">
      <annotation>
        <documentation>
        在表示货币单位的分类标准中，货币类型被用来表示财务概念的数据类型
```

```
            这种类型的实例数据项的度量单位应出自 ISO 4217 货币名空间
            </documentation>
          </annotation>
          <restriction base = "decimal" />
        </simpleType>

        <simpleType name = "shares">
          <annotation>
            <documentation>
            股份类型用于基于股份的财务概念
            </documentation>
          </annotation>
          <restriction base = "decimal" />
        </simpleType>

        <simpleType name = "pure">
          <annotation>
            <documentation>
            纯数据类型用来作为无计量单位的类型,例如百分比变化,增长率和其他一些比率
            </documentation>
          </annotation>
          <restriction base = "decimal" />
        </simpleType>

    </schema>
```

fractionItemType 的 XML 模式约束如下所示：

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

  <element name = "numerator" type = "decimal" />
  <element name = "denominator" type = "xbrli:nonZeroDecimal" />
  <complexType name = "fractionItemType" final = "extension">
    <sequence>
      <element ref = "xbrli:numerator"/>
      <element ref = "xbrli:denominator"/>
    </sequence>
    <attributeGroup ref = "xbrli:essentialNumericItemAttrs" />
  </complexType>

</schema>
```

示例 6：

表达小数

分　数	表　示
1/3	<myTaxonomy:oneThird id = "oneThird" unitRef = "u1" contextRef = "numC1"> 　<numerator>1</numerator> 　<denominator>3</denominator> </myTaxonomy:oneThird>

分子元素应包含数字值，分母元素应包含有限且非零的数字值。

9.1.3 linkbaseRef 元素

在分类标准模式中，linkbaseRef(链接库引用)元素用于识别一个链接库。如果分类标准模式在一个 DTS 中，该链接库也应在该 DTS 中。

在分类标准模式中，链接库引用元素的语法与 XBRL 实例中链接库引用元素的语法是相同的。

9.1.4 自定义 roleType 元素

roleType(角色类型)元素包含一个自定义角色类型。roleType 元素使用 roleURI 定义角色类型，列出可使用该角色的元素，并提供该角色类型的说明。

角色类型为 XLINK 可扩展链接和资源元素的 xlink:role 属性定义自定义值。角色类型元素应放在 XPATH 路径(http://xsd:schema/xsd:annotation/xsd:appinfo/*)所标识的节点集合中。本部分(作为标准的角色属性值)定义的角色值不能对角色类型元素作出重复定义。

一个分类标准模式中不可有多于一个角色类型元素有同一个 roleURI 属性值。在一个 DTS 中，可有不止一个的角色类型元素具有相同 roleURI 属性值。然而，具有相同 roleURI 属性值的所有角色类型元素应是结构等价的。

roleURI 属性值标识了要定义的 xlink:role 属性值。自定义角色类型子元素的值被用来指定允许哪个元素使用自定义角色类型。由于在使用自定义角色的链接库时通过角色引用指向角色类型元素，因此角色类型元素可有一个 id 属性，其属性以及子元素如表 28 所示。

表 28 roleType 元素的属性和子元素表

属性和子元素名称	选择性	描　述
roleURI 属性	必备	应包含已经定义的角色值。当使用自定义角色类型时，xlink:role 属性值与 roleURI 值相匹配
Id 属性	可选	id 属性的值应遵循具有 id 类型的属性的 XML 规范
definition 元素	可选	roleType 元素可包含一个 definition 元素，其内容应是一个赋予角色类型意义的字符串
usedOn 元素	可选	roleType 元素可包含一个或多个 usedOn 元素，列出可使用该角色类型的元素。在一个 roleType 元素中，不能包含结构等价的 usedOn 元素。标准扩展链接元素和资源元素应由 usedOn 元素来标识，而用户自定义扩展链接元素和资源元素则不受此条件限制

示例：

定义一个自定义角色类型

示例一个角色的角色类型定义：http://www.mycomp.com/role/endnote，以用来展示仅在文档末尾表达的一个 XBRL 实例中的脚注。

```
<schema targetNamespace = "http://www.mycomp.com/mytaxonomy"
        xmlns:link = "http://www.xbrl.org/2003/linkbase"
        xmlns = "http://www.w3.org/2001/XMLSchema" elementFormDefault = "qualified">
<annotation>
<appinfo>
    <link:roleType roleURI = "http://www.mycompany.com/role/endnote"
```

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

该 roleType 元素定义了一个角色。

```
<link:roleRef xlink:type = "simple"
  xlink:href = "mycomproles.xsd # endnote"
  roleURI = "http://www.mycomp.com/role/endnote"/>

...

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

xlink:role 值通过查找具有与 xlink:role 值匹配的 roleURI 属性值的角色引用元素来追溯到 roleType 元素。在分类标准模式 mycomproles.xsd 中角色引用元素的 xlink:href 属性直接指向(通过片段标识器)id 属性等于"endnote"的 roleType 元素。该 roleType 元素有一个匹配的 roleURI 属性值。

roleType 元素及其子元素的 XML 模式约束如下所示：

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

  <element name = "definition" type = "string">
    <annotation>
      <documentation>
      该元素用来表示自定义角色和角色弧的可读定义
      </documentation>
    </annotation>
  </element>

  <element name = "usedOn" type = "QName">
    <annotation>
      <documentation>
      usedOn 元素的定义——用来标识哪些元素可使用分类标准所定义的角色或者角色弧的值
      </documentation>
    </annotation>
  </element>

  <element name = "roleType">
    <annotation>
```

```
        〈documentation〉
        roleType 元素的定义    用来定义扩展链接中自定义角色的值
        〈/documentation〉
      〈/annotation〉
      〈complexType〉
        〈sequence〉
          〈element ref = "link:definition" minOccurs = "0"/〉
          〈element ref = "link:usedOn" maxOccurs = "unbounded"/〉
        〈/sequence〉
        〈attribute name = "roleURI" type = "xlink:nonEmptyURI" use = "required"/〉
        〈attribute name = "id" type = "ID"/〉
      〈/complexType〉
    〈/element〉

  〈/schema〉
```

9.1.5 自定义 arcroleType 元素

arcroleType(弧角色类型)元素包含一个自定义弧角色的定义。arcroleType 元素通过声明弧角色值来描述自定义 arcroleType,阐明 arcroleType 可用于的元素,通过允许使用 arcroleType 的关系网络的循环类型来描述自定义弧角色的类型,并提供一个便于读者阅读的 arcroleType 意义的定义。

arcroleType 元素应存在 XPATH 路径(http://xsd:schema/xsd:annotation/xsd:appinfo/*)所标识的节点集合中。在本部分中定义的弧角色值(作为标准的弧角色值)不能使用 arcroleType 元素再次定义。其属性如表 29 所示。

表 29 arcroleType 元素的属性表

属性名称	选择性	描　述
arcroleURI	必备	应包含已定义的弧角色的值。当使用已定义的 arcroleType 时,xlink:arcrole 的属性值应与 arcroleURI 的值相匹配
id	可选	该 id 属性的值应遵循具有 id 类型的属性的 XML 规范。它提供了一个 id 属性从而简化了 arcroleType 元素 xlink:href 属性的内容
cyclesAllowed	必备	此属性用来鉴定在一个关系网络中允许出现的循环类型。具有完全一致性的 XBRL 处理器应能检测并标明那些违反该属性限制的自定义角色类型关系网络,因为自定义角色关系类型的网络出现在标准扩展链接的标准弧上。需要注明的是,自定义弧元素的网络不受此约束条件的限制,出现在自定义的扩展链接中标准弧元素的网络也不受此限制

在一个分类标准模式中不同的 arcroleType 元素的 arcroleURI 属性值不同。在一个 DTS 中,可有多个 arcroleType 元素具有相同的 arcroleURI 值。然而,具有相同 arcroleURI 值的所有的 arcroleType 元素应是结构等价的。

arcroleURI 属性值标识了那些已被定义的 xlink:arcrole 属性值。自定义角色类型子元素的值可标识哪些弧可使用该 arcroleType。由于在使用自定义弧角色值的链接库中是由弧角色引用元素来指向 arcroleType 元素的,因此 arcroleType 元素可有一个 id 属性。

示例 1:

定义一个自定义弧角色值

定义计算链接库中链接数据项的弧角色值"http://www.mycomp.com/arcrole/average-item"

```
<schema targetNamespace = "http://www.mycomp.com/mytaxonomy"
        xmlns:link = "http://www.xbrl.org/2003/linkbase"
        xmlns = http://www.w3.org/2001/XMLSchema elementFormDefault = "qualified">
<annotation>
<appinfo>
    <link:arcroleType arcroleURI = "http://www.mycomp.com/arcrole/average-item"
      id = "average-item"
      cyclesAllowed = "none">
      <link:usedOn>link:calculationArc</link:usedOn>
    </link:arcroleType>
  </appinfo>
</annotation>
</schema>
```

```
<link:arcroleRef xlink:type = "simple"
  xlink:href = "mycomparcroles.xsd # average-item"
  arcroleURI = "http://www.mycomp.com/arcrole/average-item"/>

...

<link:calculationArc xlink:arcrole = "http://www.mycomp.com/arcrole/average-item"
                     xlink:type = "arc"
                     xlink:from = "salesAverage" xlink:to = "salesDetail"
                     link:weight = "1"/>
```

xlink:arcrole 值是通过查找具有与 xlink:arcrole 值相匹配的 arcroleURI 属性值的弧角色引用元素来追溯 arcroleType 元素的。在分类标准模式 mycomparcroles.xsd 中，弧角色引用元素的 xlink:href 属性直接指向(通过片段标识器)某个 id 属性等于"average item"的 arcroleType 元素。该 arcroleType 元素有一个匹配的 arcroleURI 属性值。

arcroleType 元素及其子元素的 XML 模式约束如下所示：

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

  <element name = "definition" type = "string">
    <annotation>
      <documentation>
      该元素用来表示自定义角色和角色弧的可读取定义
      </documentation>
    </annotation>
  </element>

  <element name = "usedOn" type = "QName">
    <annotation>
      <documentation>
      使用自定义角色类型元素的定义，
      用来标识哪些元素可使用分类标准来定义角色或者角色弧的值
      </documentation>
```

```
    </annotation>
  </element>

  <element name = "arcroleType">
    <annotation>
      <documentation>
      arcroleType 元素的定义，
      用来定义 XBRL 扩展链接中自定义弧角色的值
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref = "link:definition" minOccurs = "0"/>
        <element ref = "link:usedOn" maxOccurs = "unbounded"/>
      </sequence>
      <attribute name = "arcroleURI" type = "xlink:nonEmptyURI" use = "required"/>
      <attribute name = "id" type = "ID"/>
      <attribute name = "cyclesAllowed" use = "required">
        <simpleType>
          <restriction base = "NMTOKEN">
            <enumeration value = "any"/>
            <enumeration value = "undirected"/>
            <enumeration value = "none"/>
          </restriction>
        </simpleType>
      </attribute>
    </complexType>
  </element>

</schema>
```

在 XBRL 中的关系网络形成了连通图。由于 XPTR 中 XPointer 在 XBRL 中的应用，图中的节点将总会同 XML 的元素相对应。就 9.2 中所指定的关系来说，这些节点也会与概念或者资源相对应。在网络中的每一个关系都会同图中的一条有向的边相对应，形成了一对有序的节点对〈u，v〉。

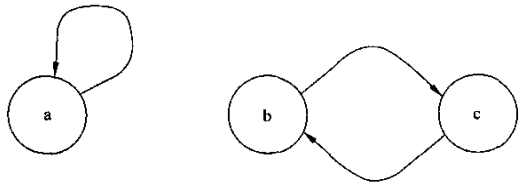
一条路径即为节点的一条顺序序列。〈V_0，V_1，…，V_{n-1}，V_n〉

当考虑了边的方向，若存在任意一个结点到其自身的路径，则称一个有向图包含一个有向的环。即，当存在一个节点的有序序列〈V_0，V_1，…，V_{n-1}，V_n〉，且 $V_0 = V_n$，则对于任意一个 V_i，当 $0 \leqslant i < n$，就存在一条有向的边（V_i，V_{i+1}）。

示例 2：

有向环

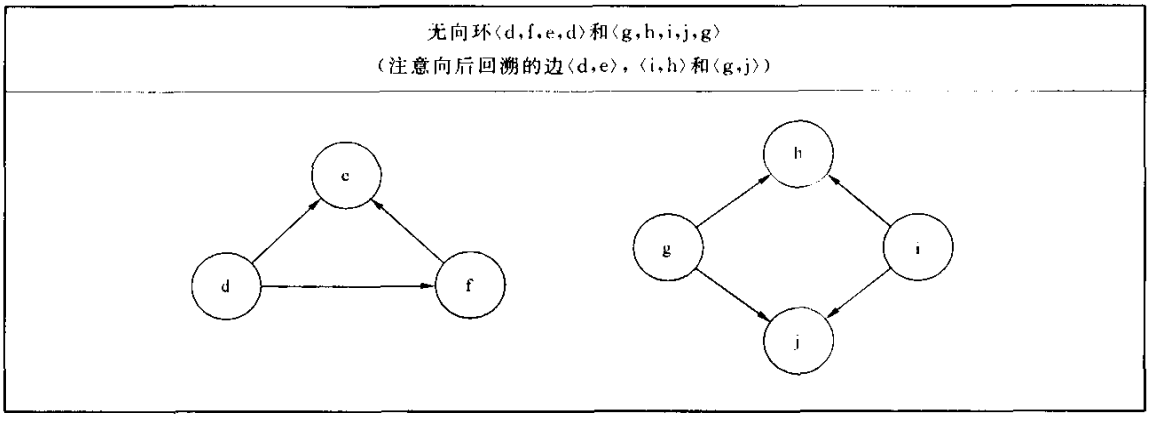
有向环〈a，a〉和〈b，c，b〉



当忽略边的方向性，若存在任意一个结点到其自身的路径，则称一个有向图包含一个无向环。即，当存在一个节点的有序序列$\langle V_0, V_1, \cdots, V_{n-1}, V_n \rangle$，且$V_0 = V_n$，则对于任意一个$V_i$，当$0 \leqslant i < n$，就存在一条有向边$(V_i, V_{i+1})$或$(V_{i+1}, V_i)$，其不同于之前的所有边。

示例 3：

无向环



cyclesAllowed 属性应具有下列值之一，如表 30 所示。

表 30　cyclesAllowed 属性值注释表

值	注　释
any	一张图可含有任何数目的有向环和无向环
undirected	一张图可含有任何数目的无向环，但不允许含有任意一条有向环
none	一张图不允许含有任何有向环和无向环

a) definition 元素

arcroleType 元素可含有一个 definition 元素，该元素应含有一个对 arcroleType 元素进行解释性定义的字符串。

b) usedOn 元素

arcroleType 元素可包含一个或多个 usedOn 类型元素。该 usedOn 类型元素可识别哪些元素可使用已定义的 arcroleType。在一个 arcroleType 元素中不能有结构等价的 usedOn 类型元素。使用已定义 arcroleType 的标准弧元素应在 arcroleType 元素中用 usedOn 元素来标识，无论其何时出现在标准扩展链接中。

重定义结构会让引用定位器的链接库中的链接目标模糊不清，因此禁止使用。

9.2　分类标准链接库

9.2.1　概述

在 XBRL 分类标准中，使用扩展链接来表达概念之间的关系以及对这些概念的进一步说明。分类标准包含五种不同的扩展链接，分别是：定义、计算、列报、标签和参考。其中，定义、计算和列报链接表达的是分类标准中概念之间的关系；标签和参考链接库表达的是概念与描述其含义的文档之间的关系；标签链接管理与多种语言表示的分类标准元素相关的文本；参考链接管理对权威文献的引用(在线或纸质形式)。

在分类标准中，如果一个链接库没有被嵌入在分类标准模式文档里，该分类标准模式应包含一个指向包含该链接库的文件的链接库引用元素。

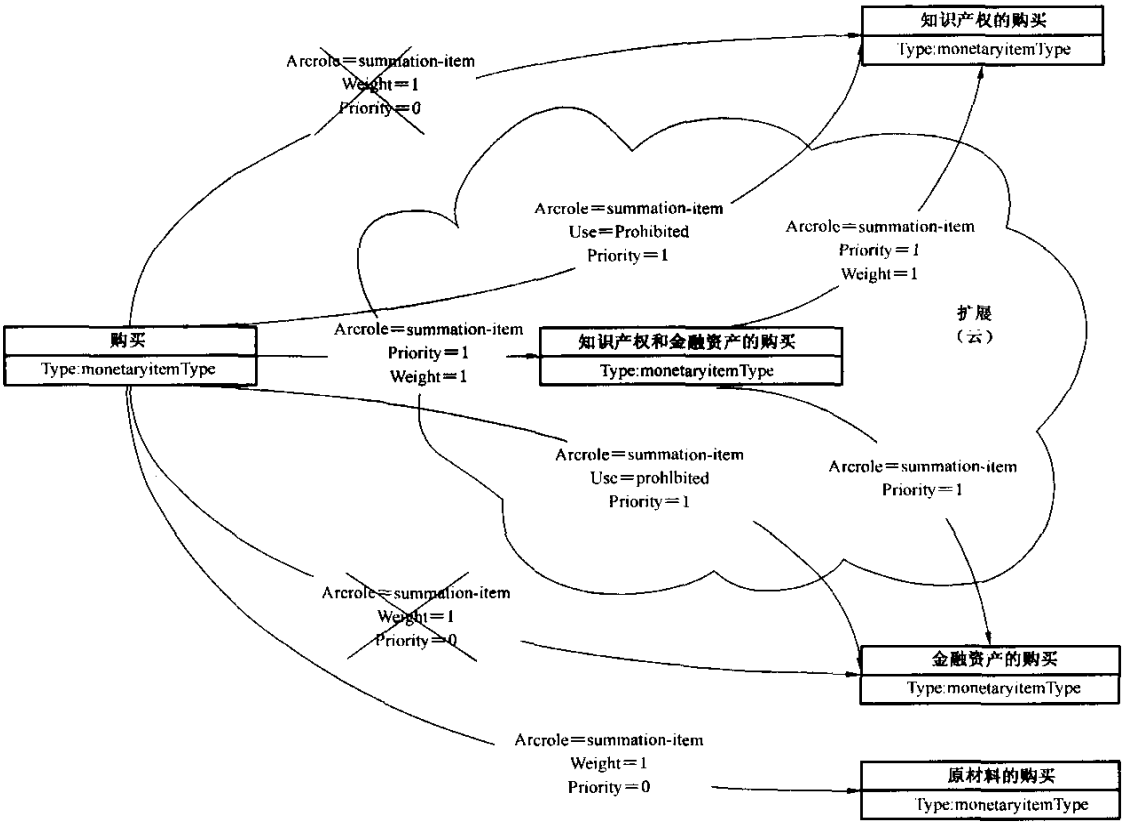
在一个 DTS 的列报、计算和定义扩展链接中，弧将 XBRL 概念组织成关系网络，以便把每个概念与其他概念关联起来。在标签和引用扩展链接中，弧表示其中的概念和它们文档间的关系网络。

DTS 中的每个概念内部之间的关系网络可包含若干根概念。对于一个给定的关系网络，根概念不是该网络中任何关系的"to"侧的 XML 片段。一个概念在一个关系网络中是根概念，但可在另一个关系网络中不是根概念。如果一个概念既不是"from"，又不是"to"侧的 XML 片段，则它是根概念。

概念间的关系以及概念与描述它们的资源间的关系可被代替或禁止，如下是一个关于禁止的例子。若考虑第三方想要在"子"概念间创建一个新的"小计"概念中介，而原来在"子"概念间已经存在对"合计"概念的合计-项弧。该"小计"概念的创建者将会从小计概念到子概念以及合计概念到小计概念各增加一条弧。那从子概念到合计概念就会有两条路径，一条使用经过小计概念的新弧，另一条使用原来的直接来自合计概念的弧。在计算链接的情况下，这将导致值的两次计算。小计概念的创建者应设立禁止弧来阻止这种情况的发生，在计算时有效地移去关系网络中直接从合计概念到子概念的弧。

示例 1：

使用关系禁止在计算网络中插入一个新的小计

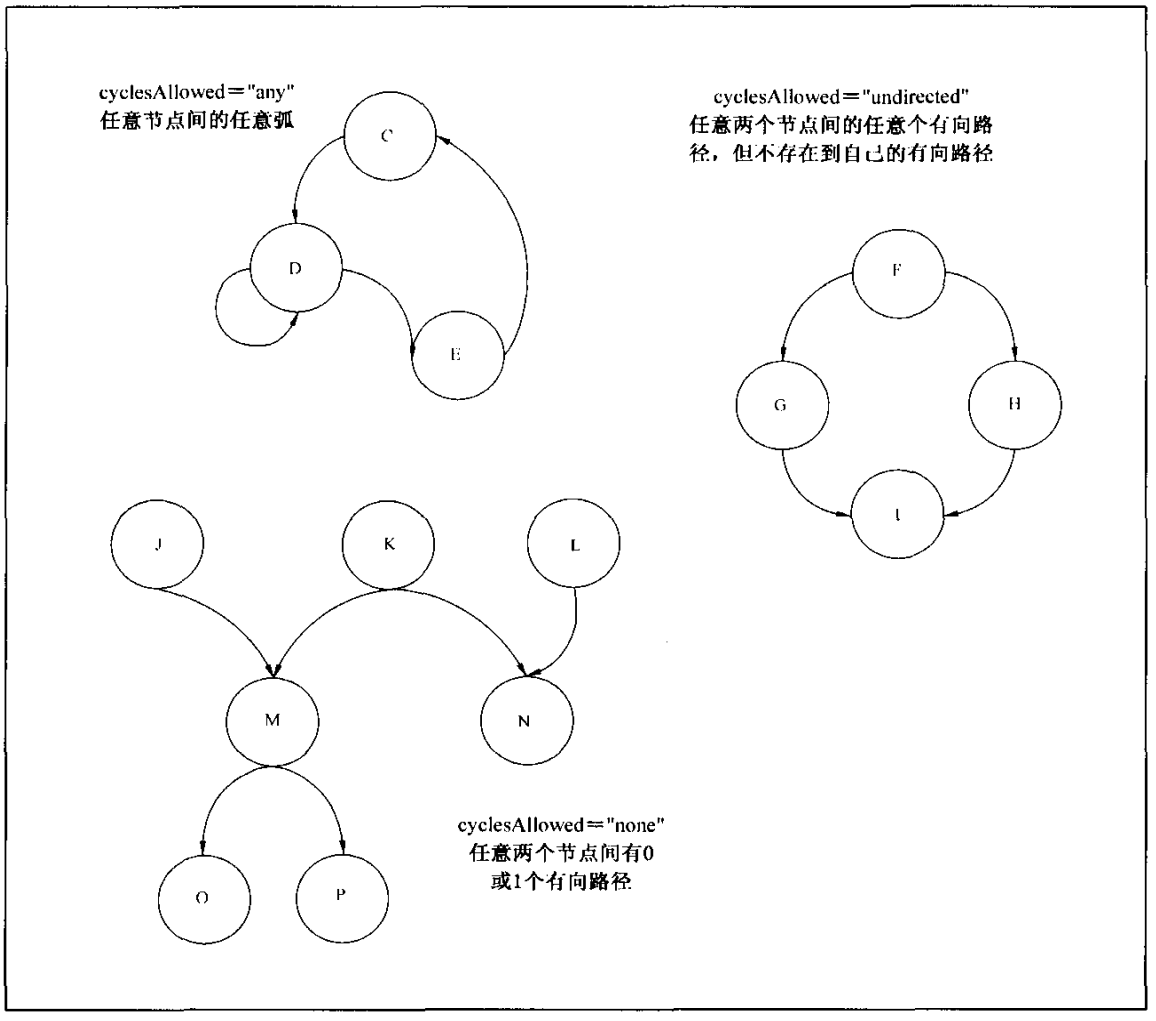


在一个关系网络中的一个或多个关系可形成一个环（即，在网络中可存在一条路径从一个 XML 片段到同一个 XML 片段，而不需要多次涉及其他的任何关系）。在一个网络中，根据关系的语义的不同，不同类型的循环在语义上可是一致的也可是不一致的，这种不一致性可被应用程序检测出来。

具有完全一致性的 XBRL 处理器应能够检测构成语义矛盾的循环。在本部分中每个给定语义意义的网络语义矛盾的循环都已被标记。

示例 2：

循环的类型



为了阐明概念间的关系网络，考虑以下可被定义到分类标准中的概念。

示例 3：

财务报表分类标准中的元素

标　签	元素名	balance	替换组
利润表	incomeStatement		
税前净收益	netIncomeBeforeTax	贷方	数据项
税	taxes	借方	数据项
税后净收益	netIncomeAfterTax	贷方	数据项
特别项目	extraordinaryItems	借方	数据项
净收益	netIncome	贷方	数据项
业绩计量	performanceMeasures		数据项

假设将分类标准中表达的元素的概念间的数学关系表示如下：

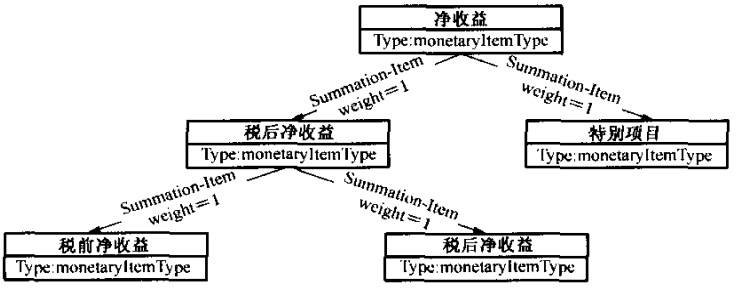
税后净收益＝税前净收益－税

净收益＝税后净收益－特别项目

计算链接库可包含计算扩展链接以便计算净收益、税前净收益、税后净收益。在一个应用中，上面的每个公式都被表示为一个树型层次结构。

示例 4：

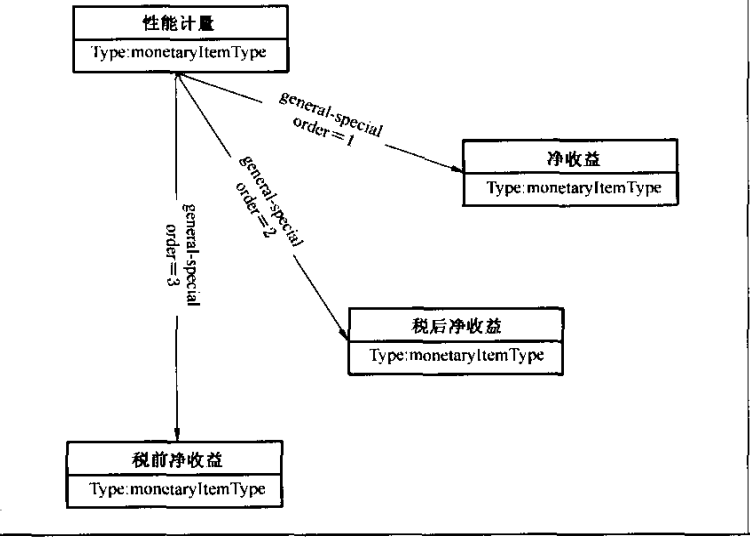
计算链接库中的层次



定义链接库中也可包含一个概念与其他概念相关联的定义扩展链接。在下例中，业绩计量（performanceMeasures）是分类标准中定义的一个元素，业绩计量的类型是：净收益、税前净收益和税后净收益。链接的 xlink:arcrole 是一个 URI，解释了对关系定义的联系类型。

示例 5：

定义链接库中从一般到特殊的弧的层次



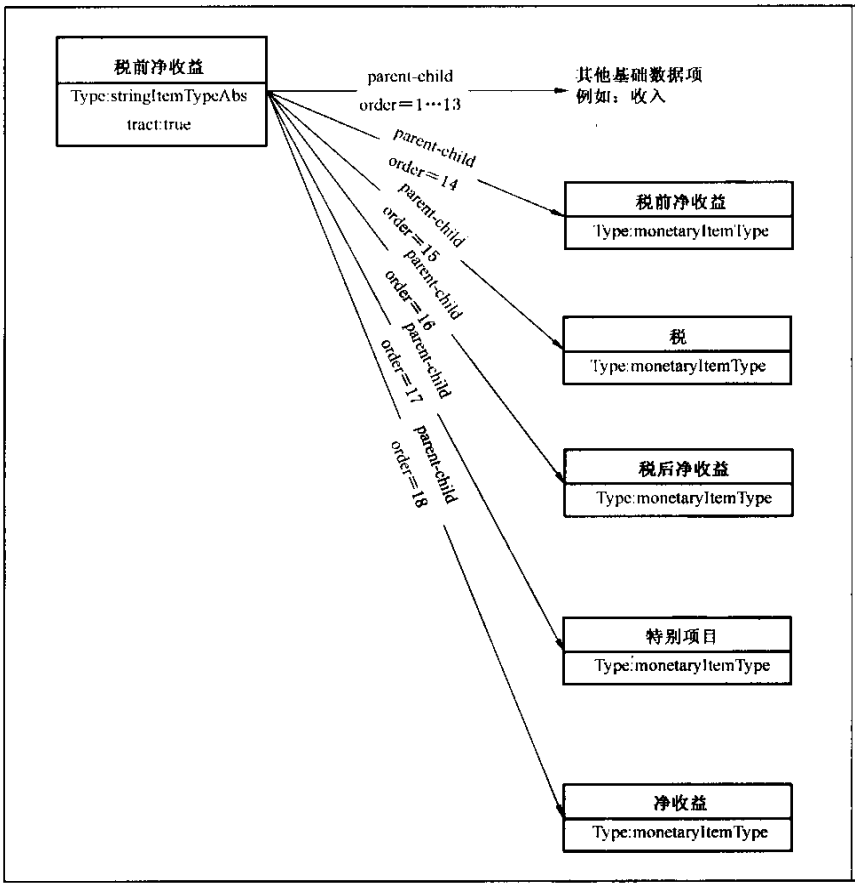
列报链接用于组织分类标准元素之间的层次关系并为之排序。

不同的应用需要不同的列报链接。有一组用户的集合——分类标准开发者和使用分类标准的领域专家，他们的列报需要在分类标准的整个生命时间相关。当分类标准发布时，它们不可包含所有可能的列报，但是可包含至少一个“开发者角度”的视图，该视图与上下文无关，它不必考虑 XBRL 实例上下文。在该例子中的列报链接库可包含一个列报链接来组织概念，从而使它们看起来像在财务报表中的

行项目。其他列报链接库可包含某一链接以将那些相同的概念子集组成一个数据集合。

示例 6：

列报链接库中的层次



模仿在一个利润表中可出现的行项目顺序的列报层次。

该视图可用于向用户展现分类标准。弧用“次序”属性来注释。

9.2.2 labelLink 元素

labelLink(标签链接)元素是一个扩展链接，其目的是包含概念和原文文档及这些概念标签的关系。

labelLink 元素的 XML 模式约束如下：

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

  <element name = "labelLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
      标签扩展链接元素的定义
      </documentation>
    </annotation>
    <complexType>
```

```
      〈complexContent〉
        〈restriction base = "xl:extendedType"〉
          〈choice minOccurs = "0" maxOccurs = "unbounded"〉
            〈element ref = "xl:title"/〉
            〈element ref = "link:documentation"/〉
            〈element ref = "link:loc"/〉
            〈element ref = "link:labelArc"/〉
            〈element ref = "link:label"/〉
          〈/choice〉
          〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
processContents = "lax" /〉
        〈/restriction〉
      〈/complexContent〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

a) 定位器

labelLink 元素不能包含非 loc 元素的定位器。

b) label 元素

尽管每个分类标准都定义了一个表示一系列业务报告概念的单独元素集，但是对于其他概念的可阅读 XBRL 文档，包括标签(字符串，其能让每个概念都有一个可理解的名字)和其他说明文档，均被放置在标签链接库的资源元素中。该资源元素用 xml:lang 属性来指定所使用的语言(通过 XML 的标准语言属性)和一个对文件目的可选的分类标准(通过角色属性)，其属性如表 31 所示。

表 31 label 元素的属性表

属性名称	选择性	描　述
xml:lang	必备	用于识别标签内容所使用的语言，xml:lang 属性值应遵循 XML 规范
xlink:role	可选	该属性应通过标签资源所提供的 XBRL 概念文档的性质来在标签资源之间进行区分。标准标签的角色属性值如表 32 所示

XBRL 概念文档应被包含在标签链接的扩展链接标签元素中。label 元素是一个 XLINK 资源，应有标准 xml:lang 属性，而且应出现在一个标签链接元素的内部。label 元素的内容是混合的，可是一个简单的字符串，一个 XHTML 规范片段或者两者的组合。

XBRL 处理器没有必要检测或者显示出现在标签元素以外的概念文档。

lable 元素的 XML 模式约束如下：

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

  〈element name = "label" substitutionGroup = "xl:resource"〉
    〈annotation〉
```

```
      〈documentation〉
          标签资源元素的定义
      〈/documentation〉
    〈/annotation〉
    〈complexType mixed = "true"〉
      〈complexContent mixed = "true"〉
        〈extension base = "xl:resourceType"〉
          〈sequence〉
            〈any namespace = "http://www.w3.org/1999/xhtml"
              processContents = "skip" minOccurs = "0" maxOccurs = "unbounded"/〉
          〈/sequence〉
          〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
            processContents = "lax" /〉
        〈/extension〉
      〈/complexContent〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

示例 1：

标签资源示例

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

```
〈label xlink:type = "resource" xlink:role = "http://www.xbrl.org/2003/role/label"
     xmlns:xhtml = "http://www.w3.org/1999/xhtml" xlink:label = "ci_netIncome_en"
    xml:lang = "en"〉〈xhtml:b〉Net Income〈/xhtml:b〉 (Loss)〈/label〉
```

表 32　标准标签角色属性值

标签资源 xlink:role 属性值	含　义
无 xlink:role 属性值	概念的标准标签
http://www.xbrl.org/2003/role/label	概念的标准标签
http://www.xbrl.org/2003/role/terseLabel	概念的短标签，当该概念在其他相关概念的上下文中使用时，通常省略能推导出的内容
http://www.xbrl.org/2003/role/verboseLabel	概念的扩展标签，确保该标签在单独使用时不忽略该概念被理解的信息
http://www.xbrl.org/2003/role/positiveLabel http://www.xbrl.org/2003/role/positiveTerseLabel http://www.xbrl.org/2003/role/positiveVerboseLabel http://www.xbrl.org/2003/role/negativeLabel http://www.xbrl.org/2003/role/negativeTerseLabel http://www.xbrl.org/2003/role/negativeVerboseLabel http://www.xbrl.org/2003/role/zeroLabel http://www.xbrl.org/2003/role/zeroTerseLabel http://www.xbrl.org/2003/role/zeroVerboseLabel	当概念要表达的值是正（负或零）时使用的标签。例如，标准标签和标准的正标签可是“税后利润”，而标准的负标签是“税后亏损”，短标签和正的短标签可都是“利润”，而负的短标签可是“亏损”

表 32 标准标签角色属性值（续）

标签资源 xlink:role 属性值	含　义
http://www.xbrl.org/2003/role/totalLabel	当该概念被用于表示一组其他值的集合的总数时使用的标签
http://www.xbrl.org/2003/role/periodStartLabel http://www.xbrl.org/2003/role/periodEndLabel	时间类型="时间点"概念的标签，当它在一段时间值的开始(结束)点上出现时，即被用于表示与概念相关的值
http://www.xbrl.org/2003/role/documentation	概念的文档，它提供了对该概念的含义，适当的用法和其他必要文档的解释
http://www.xbrl.org/2003/role/definitionGuidance	概念的精确定义，提供了对该概念的含义和适当用法的解释
http://www.xbrl.org/2003/role/disclosureGuidance	对与该概念相关的披露需求的解释，指出该披露是否是： 1) 强制性的(例如：在权威文献上描述的)； 2) 建议性的(例如：权威文献上提倡的)； 3) 普遍实践性的(例如：不是权威文献所描述的，但披露是普遍的)； 4) 结构完整的(例如：包含在完备的分类标准结构中)
http://www.xbrl.org/2003/role/presentationGuidance	在一个或多个具体类型的业务报告的其他概念的上下文中对该概念的定向表达(布置或者标签)的规则进行解释(例如：净余额应能在损益表中显示出来)
http://www.xbrl.org/2003/role/measurementGuidance	对在业务报告中与该概念相关的度量值所需要的度量方法的解释
http://www.xbrl.org/2003/role/commentaryGuidance	任何其他的有助于对概念的解释、发布、度量、展示或者使用的通用注释
http://www.xbrl.org/2003/role/exampleGuidance	一个被概念捕获的信息类型的例子

示例 2：

概念和它的某个标签之间的弧

```
<label
  xlink:type = "resource"
xlink:label = "A"
xlink:role = "http://www.xbrl.org/2003/role/label"
xml:lang = "en">Current Assets</label>
<loc
xlink:type = "locator"
xlink:href = "us_bs_v2.xsd#currentAssets"
xlink:label = "B"/>
<labelArc
xlink:type = "arc"
xlink:from = "B"
xlink:to = "A"
xlink:arcrole = "http://www.xbrl.org/2003/arcrole/concept-label"/>
```

c) labelArc 元素

labelArc(标签弧)元素是一个 XLINK 弧。在 labelLink 元素中,它将概念和标签资源链接起来。

labelArc 元素的 XML 模式约束如下:

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

  <element name = "labelArc" type = "xl:arcType" substitutionGroup = "xl:arc">
    <annotation>
      <documentation>
          在标签扩展链接中对弧的具体使用
      </documentation>
    </annotation>
  </element>

</schema>
```

labelArc 元素不能描述概念间的循环关系,因为它只是把概念与标签资源相关联,而没把概念同其他概念相关联,所以没有对循环标签弧网络的约束进行描述。

当标签弧上的可用性属性是"禁止"时,label 元素可是 XLINK 本地资源或者是 XLINK 的远程资源;当标签弧上的可用性属性不是"禁止"时,参与 labelArc 元素描述关系的标签元素应是 XLINK 本地资源。

9.2.3 referenceLink 元素

referenceLink(参考链接)元素是一个扩展链接,其目的是包含概念和相关法律法规(条款)间的引用关系。

referenceLink 元素的 XML 模式约束如下:

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

  <element name = "referenceLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
          参考扩展链接元素的定义
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base = "xl:extendedType">
          <choice minOccurs = "0" maxOccurs = "unbounded">
            <element ref = "xl:title"/>
            <element ref = "link:documentation"/>
```

```
            <element ref = "link:loc"/>
            <element ref = "link:referenceArc"/>
            <element ref = "link:reference"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
        processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

示例 1：

几个参考链接元素的 xlink:role 值的样本

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

分类标准给每个引用链接的扩展链接一个"角色"来划分一个会计相关的分类标准中的扩展链接，基于该会计相关的分类标准来划分与之相关的财务报告

a) 定位器

referenceLink 元素不能包含非 loc 元素的定位器。

b) reference 元素

XBRL 分类标准使用 reference(参考)元素对已公开出版的商业、经济、会计文献中概念定义的权威陈述进行引用。reference 元素只提供必要信息来查找和理解与已定义概念的适当用法相关的引用资料，不包含这些参考资料本身的内容。当需要文本文档来完成对一个 XBRL 内容的定义时，它应包含在标签元素之中。

reference 元素是一个 XLINK 资源，应出现在 referenceLink 元素中。

参考元素的 XML 模式约束如下：

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

  <element name = "part" type = "anySimpleType" abstract = "true">
    <annotation>
      <documentation>
      定义参考部分元素，为了在参考资源中使用
      </documentation>
    </annotation>
  </element>

  <element name = "reference" substitutionGroup = "xl:resource">
    <annotation>
```

```
      〈documentation〉
        参考资源元素的定义
      〈/documentation〉
    〈/annotation〉
    〈complexType mixed = "true"〉
      〈complexContent mixed = "true"〉
        〈extension base = "xl:resourceType"〉
          〈sequence〉
            〈element ref = "link:part" minOccurs = "0" maxOccurs = "unbounded"/〉
          〈/sequence〉
        〈/extension〉
      〈/complexContent〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

reference 元素由很多部分组成。由于各个领域对参考文献的划分均不同，因此本文件中定义了一个 part 抽象元素。分类标准可定义元素来替代 part 元素，以包含在 reference 元素中。

示例 2：

概念和支撑参考之间的弧

```
〈linkbase xmlns = "http://www.xbrl.org/2003/linkbase"
          xmlns:ref = "http://www.xbrl.org/2003/ref"
          xmlns:link = "http://www.xbrl.org/2003/linkbase"
          xmlns:xlink = "http://www.w3.org/1999/xlink"
          xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"〉
  〈referenceLink xlink:type = "extended"
    xlink:role = "http://www.xbrl.org/2003/role/link"〉
    〈! -- locator for element --〉
    〈loc xlink:type = "locator"
         xlink:href = "samp001.xsd # s_customerName"
         xlink:label = "s_customerName"/〉
    〈! -- arcs --〉
    〈referenceArc
      xlink:type = "arc" xlink:from = "s_customerName" xlink:to = "s_customerName_REF"
      xlink:arcrole = "http://www.xbrl.org/2003/arcrole/concept-reference"/〉
      〈! -- references all with the same xlink:label --〉
    〈reference
      xlink:type = "resource"
      xlink:label = "s_customerName_REF"
      xlink:role = "http://www.xbrl.org/2003/role/definitionRef"〉
      〈ref:name〉Handbook of Business Reporting〈/ref:name〉
      〈ref:pages〉5〈/ref:pages〉
    〈/reference〉
    〈reference
      xlink:type = "resource"
      xlink:label = "s_customerName_REF"
      xlink:role = "http://www.xbrl.org/2003/role/measurementRef"〉
      〈ref:name〉Handbook of Business Reporting〈/ref:name〉
      〈ref:pages〉45-50〈/ref:pages〉
    〈/reference〉
  〈/referenceLink〉
〈/linkbase〉
```

该参考元素包含两个文献引用，它们以不同的 xlink:role 属性来进行区分。弧将概念和两个参考相关联。元素名称和页码被定义为部分替换组成员，在分类标准中通过"ref:"命名空间前缀进行引用，如下所示：

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

示例 3：

参考资源

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

reference 元素可包含 xlink:role 属性，用于区分 reference 元素的类型。表 33 说明了标准 xlink:role 属性的值以及参考资源的含义，与标签资源的标准 xlink:role 属性值类似。

表 33　参考角色属性值

参考资源 xlink:role 属性值	含　义
无 xlink:role 属性值	概念的标准引用
http://www.xbrl.org/2003/role/reference	概念的标准引用
http://www.xbrl.org/2003/role/definitionRef	对概念精确定义的文档的引用
http://www.xbrl.org/2003/role/disclosureRef http://www.xbrl.org/2003/role/mandatoryDisclosureRef http://www.xbrl.org/2003/role/recommendedDisclosureRef	对与概念相关的发布需求解释文档的引用。指定种类包括：强制性，建议性
http://www.xbrl.org/2003/role/unspecifiedDisclosureRef	对与概念相关的发布需求解释细节文档的引用。非指定种类包括，但不局限于：一般实践，结构完整性。后者没有引用文档，但是隐含在链接角色中来暗示为什么该概念被包含在分类标准中
http://www.xbrl.org/2003/role/presentationRef	对文档的引用，此文档详细解释了在一个或多个具体类型的财务报告的其他概念的上下文中的该概念的展示、位置和标签

表 33 参考角色属性值（续）

参考资源 xlink:role 属性值	含 义
http://www.xbrl.org/2003/role/measurementRef	在财务报告中对与该概念相关的度量值所用方法的引用
http://www.xbrl.org/2003/role/commentaryRef	其他的对该概念的一般解释，这些解释有助于对其恰当的使用
http://www.xbrl.org/2003/role/exampleRef	对该概念应用实例的阐述文档的引用，这有助于其恰当的应用

c) referenceArc 元素

referenceArc(参考弧)元素是一个 XLINK 弧。在 referenceLink 元素中，referenceArc 元素链接概念和参考资源。

referenceArc 元素的 XML 模式约束如下：

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

  <element name = "referenceArc" type = "xl:arcType"
substitutionGroup = "xl:arc">
    <annotation>
      <documentation>
          在参考列报链接中弧的具体使用
      </documentation>
    </annotation>
  </element>

</schema>
```

referenceArc 值是从一个概念 loc 元素到参考资源的引用弧，指向描述概念含义的相关资料。

referenceArc 元素不能描述概念间的循环关系，因为它只是表达概念和引用资源间的关系，而不是概念和其他概念间的关系。所以没有对循环参考弧网络的约束进行描述。

当 referenceArc 上的可用性属性是“禁止”时，引用元素可是 XLINK 本地资源或者是 XLINK 的远程资源；当 referenceArc 上的可用性属性不是“禁止”时，参与 referenceArc 元素描述关系的引用元素应是 XLINK 本地资源。

9.2.4 presentationLink 元素

presentationLink(列报链接)元素是一个扩展链接，描述分类标准中概念间的列示关系。presentationLink 元素不能包含 XLINK 资源。

presentationLink 元素的 XML 模式约束如下：

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

  <element name = "presentationLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
      列报扩展链接元素的定义
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base = "xl:extendedType">
          <choice minOccurs = "0" maxOccurs = "unbounded">
            <element ref = "xl:title"/>
            <element ref = "link:documentation"/>
            <element ref = "link:loc"/>
            <element ref = "link:presentationArc"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

a) 定位器

presentationLink 元素中不能包含非 loc 元素的定位器。当 loc 元素用于一个列报链接时，其只能指向分类标准模式中的概念。

b) presentationArc 元素

presentationArc(展示弧)元素是一个 XLINK 弧。presentationArc 元素定义了一个概念在展示时如何与其他概念相关联。

presentationArc 元素语法的 XML 模式约束如下：

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

  <element name = "presentationArc" substitutionGroup = "xl:arc">
    <complexType>
      <annotation>
        <documentation>
    为展示弧而对列报链接弧类型的扩充。增加一个 preferredLabel 属性来记录首选标签的角色属性值(当它们在
标签扩展链接中出现时)
```

```
        </documentation>
      </annotation>
      <complexContent>
        <extension base = "xl:arcType">
          <attribute name = "preferredLabel" use = "optional">
            <simpleType>
              <restriction base = "anyURI">
                <minLength value = "1"/>
              </restriction>
            </simpleType>
          </attribute>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

示例 1：

展示弧

```
<presentationArc
  xlink:type = "arc"
  xlink:from = "ci_currentAssets"
  xlink:to = "ci_prepaidExpenses"
  xlink:arcrole = "http://www.xbrl.org/2003/arcrole/parent-child"
  order = "4"/>
```

Current assets(流动资产)应展示为 prepaidExpenses(预收费用)的父元素。预收费用应出现在任何次序小于 4 的流动资产的子元素的后面，出现在任何次序大于 4 的流动资产的子元素的前面

分类标准可定义抽象元素并创建展示关系到这些元素或者创建源于它们的展示关系，允许分类标准展示应用程序来表达概念组，即使这些概念在其他任何方式上均无关联。抽象元素应出现在抽象的 XBRL 数据项的替换组中。

示例 2：

抽象概念定义

```
<element name = "balanceSheet" id = "ci_balanceSheet"
    type = "xbrli:stringItemType"
    substitutionGroup = "xbrli:item" abstract = "true"
    xbrli:periodType = "instant"/>
```

存在于分类标准中的 balanceSheet (资产负债表)元素仅用来组织其他元素；它不能出现在一个 XBRL 实例中。它有任意类型的属性，这些任意的属性不增加任何语义信息

preferredLabel 属性是一个 URI，它可被用在一个“父—子”弧中来指出当展示一个弧的子元素时应采用的最恰当的标签类型。如果使用 preferredLabel 属性，其值应与标签资源(在标签链接的扩展链接中)的 xlink:role 属性值相等，该标签资源是源于展示弧元素的子概念的“概念—标签”弧的目标。

XBRL 处理器可使用 preferredLabel 属性值来在与某个概念相关的不同标签间进行选择，特别是当一个给定的概念在 DTS 中被用于多种不同的方式时。

包含首选标签的标签扩展链接的 xlink:role 属性值与包含展示弧元素的展示扩展链接的 xlink:role 属性值不必相等。

9.2.5 calculationLink 元素

calculationLink(计算链接)元素是一个扩展链接,描述分类标准中概念间的计算关系。计算链接元素不能包含 XLINK 资源。

calculationLink 元素的 XML 模式约束如下:

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

  <element name = "calculationLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
    计算扩展链接元素的定义
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base = "xl:extendedType">
          <choice minOccurs = "0" maxOccurs = "unbounded">
            <element ref = "xl:title"/>
            <element ref = "link:documentation"/>
            <element ref = "link:loc"/>
            <element ref = "link:calculationArc"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

a) 定位器

calculationLink 元素不能包含非 loc 元素的定位器,该定位器应且仅能指向分类标准模式中的概念。

b) calculationArc(计算弧)元素

calculationArc 是一个 XLINK 弧表达概念间的计算关系。

calculationArc 元素语法的 XML 模式约束如下:

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

  <element name = "calculationArc" substitutionGroup = "xl:arc">
```

```
    〈complexType〉
      〈annotation〉
        〈documentation〉
      为计算弧而对扩展链接弧类型的扩充。
      增加一个 weight(权值)属性来追踪权值以便求和。
        〈/documentation〉
      〈/annotation〉
      〈complexContent〉
        〈extension base = "xl:arcType"〉
          〈attribute name = "weight" type = "decimal" use = "required"/〉
        〈/extension〉
      〈/complexContent〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

计算仅仅是检查在实例中已经明确提供的信息，不考虑通过本名-别名关系推断出的不同项的和值。如果几个数据项(它们都对应于一个概念)不是重复的，则可被绑定到一个和项上。这与通过元组计算处理的上下文相关，并且这意味着重复检测的方法对 XBRL 实例中的双重计算问题并不是一个充分的测试。

绑定计算的和被定义为，在绑定中的贡献数值型数据项经过四舍五入之后的和，每个数值型数据项都乘以与该数据项相关的计算弧的权值属性值。该乘法在任何必要的舍入之后进行。数值型数据项的舍入值由该数据项的值根据它的精度或者推理精度舍入后得到。如果合计项的舍入值与该合计项的总数的舍入精度或者推理精度相同，则将该绑定计算定义为一致的。

如果 XBRL 实例中所有的绑定计算都是一致的，则该 XBRL 实例与它所支撑 DTS 中的计算链接库的语义一致。

示例 1：

涉及小数和精度的计算

假设数值型数据项(numeirc item)a 是数值型数据项 b 与数值型数据项 c(权值为 1.0)的和，并在该实例中存在着一个 id 为"c1"的上下文和一个 id 为"u1"的单位，因此该总和被绑定。为了进行计算，首先将 984.8 舍入成精度为 3 的 985，然后将 582.334973 舍入为推理精度为 4 的 582.3，因而，合计为 1567.3。合计舍入精度为 2 时(和数据项 a 的精度)1600 与总和项相等，因而该计算具有一致性

```
〈a contextRef = "c1" unitRef = "u1" precision = "2"〉1559〈/a〉
〈b contextRef = "c1" unitRef = "u1" precision = "3"〉984.8〈/b〉
〈c contextRef = "c1" unitRef = "u1" decimals = "1"〉582.334973〈/c〉
```

该计算不具有一致性，因为合计的精度是 2，仍然是 1 600，但是总和项到精度为 2 时的值却为 1 500

```
〈a contextRef = "c1" unitRef = "u1" precision = "2"〉1527〈/a〉
〈b contextRef = "c1" unitRef = "u1" precision = "3"〉984.8〈/b〉
〈c contextRef = "c1" unitRef = "u1" decimals = "1"〉582.334973〈/c〉
```

示例 2：

计算弧的语法

```
〈calculationArc
    xlink:type = "arc"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/summation-item"
```

```
    xlink:from = "currentAssets"
    xlink:to = "prepaidExpenses"
    weight = "1.0"
    order = "1"/>
```

一个 DTS 可包含一个不同视角的单个元素或者有几个不同的维度。在下面的例子中，现金概念可按照分支机构的位置，账户类型和可用性划分。

示例 3：

现金，现金等价物通过分支机构位置和账户类型进行计算

现金
- 由分支结构位置划分的现金
  - 国内分支机构的现金
  - 国外分支机构的现金
- 由账户类型划分的现金
  - 带息账户现金
  - 无息账户现金
- 通过认可方式划分的现金
  - 库存现金
  - 结欠余额现金

表 34　数据项含义表

数据项	含　义
weight 属性	weight 属性应出现在计算弧元素上，其属性应是一个非零小数值
计算范围	在 XBRL 实例中的计算要考虑元组的结构。"from"项应是计算关系绑定的"from"和"to"项的最小共同祖先的一个 *f* 项。这种范围致使重复元组内的数据项不能同时参与计算

示例 4：

具有嵌套元组的 XBRL 实例片段

在该计算链接中有三个计算弧：
从(合计项)净额到(项)总额，权值＝1.0
从(合计项)净额到(项)利润，权值＝－1.0
从(合计项)总额合计到(项)总额，权值＝1.0
下面是一个 XBRL 实例片段。注意所有数值型数据项(numeric item)共享一个单独的上下文 c1

```
<analysis>
    <customer>
      <name contextRef = "c1">SINA</name>
      <gross precision = "4" unitRef = "u1" contextRef = "c1">3000</gross>
      <returns precision = "3" unitRef = "u1" contextRef = "c1">100</returns>
      <net precision = "4" unitRef = "u1" contextRef = "c1">2900</net>
    </customer>
    <customer>
      <name contextRef = "c1">Bree</name>
```

```
    〈gross precision = "4" unitRef = "u1" contextRef = "c1"〉2000〈/gross〉
    〈returns precision = "3" unitRef = "u1" contextRef = "c1"〉200〈/returns〉
    〈net precision = "4" unitRef = "u1" contextRef = "c1"〉1800〈/net〉
  〈/customer〉
  〈totalGross precision = "4" unitRef = "u1" contextRef = "c1"〉5000〈/totalGross〉
  〈/analysis〉
```

计算数据项"to"的路径	计算数据项"from"的路径	匹配	理由
analysis/customer[1]/gross	analysis/customer[1]/net	是	它们是兄弟
analysis/customer[2]/gross	analysis/customer[2]/net	是	它们是兄弟
analysis/customer[1]/returns	analysis/customer[1]/net	是	它们是兄弟
analysis/customer[2]/gross	analysis/customer[2]/net	是	它们是兄弟
analysis/customer[1]/gross	analysis/customer[2]/net	否	"to"侧的和不是项的兄弟或者叔叔
analysis/customer[2]/gross	analysis/customer[1]/net	否	"to"侧的和不是项的兄弟或者叔叔
analysis/customer[1]/gross	analysis/totalGross	是	在祖先 analysis 下 totalGross 是项的叔叔
analysis/customer[2]/gross	analysis/totalGross	是	在祖先 analysis 下 totalGross 是项的叔叔

9.2.6 definitionLink 元素

9.2.6.1 概述

definitionLink(定义链接)元素是一个扩展链接,包含分类标准中概念间多种复杂的关系。definitionLink 元素不能包含 XLINK 资源。

definitionLink 元素的 XML 模式约束如下:

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

  〈element name = "definitionLink" substitutionGroup = "xl:extended"〉
    〈annotation〉
      〈documentation〉
      定义扩展链接元素的定义
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈complexContent〉
        〈restriction base = "xl:extendedType"〉
          〈choice minOccurs = "0" maxOccurs = "unbounded"〉
            〈element ref = "xl:title"/〉
            〈element ref = "link:documentation"/〉
```

```
            <element ref = "link:loc"/>
            <element ref = "link:definitionArc"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
        processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

</schema>
```

9.2.6.2 定位器

definitionLink 元素不能包含非 loc 元素的定位器，且只能指向分类标准模式中的概念。

9.2.6.3 definitionArc

9.2.6.3.1 概述

definitionArc(定义弧)元素是一个 XLINK 弧，定义了概念间的多种关系。

定义弧元素语法的 XML 模式约束如下：

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

  <element name = "definitionArc" type = "xl:arcType" substitutionGroup = "xl:arc">
    <annotation>
      <documentation>
      在定义扩展链接中弧的具体使用
      </documentation>
    </annotation>
  </element>

</schema>
```

定义弧元素定义了“通用—特殊”(general-special)弧、“本名—别名”(essence-alias)弧、“相似元组”(similar-tuples)弧、“必需元素”(requires-element)弧等四个标准弧角色值。

9.2.6.3.2 通用—特殊弧

“通用—特殊”(general-special)弧角色值定义的弧元素应表示仅仅存在于数据项替换组中的概念间的关系。

“通用—特殊”弧链接一个通用的概念定位器和一个特殊化的概念定位器。在 XBRL 实例中，通用数据项是通用概念的体现，特殊化数据项是特殊化概念的体现。特殊化数据项的有效值即为它的通用数据项的有效值(如果两者是上下文等价并且单位等价的)。但是，通用数据项的有效值不一定是它的特殊化数据项的有效值，即使它们是上下文等价并且单位等价的。

在“通用—特殊”弧的网络中只允许存在无向环。具有完全一致性的 XBRL 处理器应能检测和报告“通用—特殊”弧网络中的有向环。

示例:“通用—特殊”弧

```
<definitionArc
    xlink:type = "arc"
    xlink:from = "postalCode"
    xlink:to = "zipCode"
    xlink:arcrole = "http://www.xbrl.org/2003/arcrole/general-special"
    order = "1"/>
```

postalCode(地区编码)是 zipCode(邮政编码)的通用形式。次序属性指出当该链接展现给用户时,它出现在次序小于 1 的链接之后,次序大于 1 的链接之前。

9.2.6.3.3 “本名—别名”弧

“本名—别名”(essence-alias)弧角色值定义的弧元素应表达仅存在于数据项替换组中的概念之间的关系。

该弧角色值用于一个从本名概念定位器到别名概念定位器的定义弧。

在“本名—别名”弧的网络中只允许出现无向环。具有完全一致性的 XBRL 处理器应能检测在“本名—别名”弧网络中存在的任何有向环。

通常的情况是,在一个单独的分类标准或一个分类标准集中,特定的概念会不止一次地被定义。在这种情况下,分类标准的制定者对一个概念有一个单独的“规范的最好元素”或“本名”并将它用“本名—别名”定义弧与其他“别名”概念相关联,并指定给 XBRL 验证器和其他 XBRL 实例应用程序,该数据项应与下面的定义一致。

“本名—别名”弧表示两个概念间的关系,从本名(基本的,主要的)概念到其他别名(选择性名称)概念。

在一个 XBRL 实例里的结构等价的上下文中,对于具有相同本名概念的“本名—别名”弧的任何集合,术语“别名概念集合”意味着与该弧集合相关的别名概念集合,术语“别名项集合”则意味着相关数据项的集合。下面的情形可用于定义弧——其在具有这种弧角色的任何扩展分类标准中没有被禁止,也可用于定义这种弧的别名概念和本名概念,以及它们相应的别名项和本名项。

a) 别名概念可是任意个其他别名概念的本名概念。

b) 弧的别名概念和本名概念应有相同的数据项类型并且 periodType 属性应有相同的值。同样,如果一个弧的别名概念和本名概念都出现了 balance 属性,则两个概念应有相同的值。如果这两个概念中有一个没有 balance 属性,则就没有类似的要求。

c) 如果在 XBRL 实例中的结构等价的上下文中别名数据项和本名数据项是上下文等价且是父元素等价的,但非值等价或非单位等价,那么这两个数据项在支撑该实例的 DTS 中语义冲突。只有在这两个数据项都不是空值时才需要进行判断,也只有完全遵照 XBRL 规范的处理器才要求应检测出这种冲突。

d) 对任何一个非数值的本名概念 E,E 在 XBRL 实例上下文 C 中没有相关的其父元素是 P 的本名数据项 EI,一个 XBRL 处理器可推导出存在一个数据项 EI,该 EI 有一个与非空别名项集 S 的所有成员值(非空值)是值等价的,与集合 S 相应的本名别名弧是以 E 作为本名概念、P 为双亲。如果 S 集合的所有成员(非空值的)不是值等价的,那么该 XBRL 实例与在 DTS 中所表达的定义链接语义是不一致的,具有完全一致性的处理器应能检测并通告这种不一致性。如果一个应用软件程序采用了该规则,并且 S 的任意成员 M 没有提供一个值,或者有一个空值,但是本名数据项存在于“本名—别名”弧的某些集合中,该规则应被递归地应用于在推导 E 值之

前推导 M 的值。

示例 1：

对由本名—别名弧链接的概念的非数值型数据项值的推理

在 XBRL 实例中，存在一个上下文 c1。概念 D 和 E 是由一个“本名—别名”定义弧链接的字符串数据项类型，E 是本名概念，D 是别名概念。在上下文 c1 下，E 的值为“Bert”，D 的值为“Ernie”。这些值与已经表达的定义弧语义不一致

e) 对任何数值本名概念 E，E 在 XBRL 实例上下文 C 中没有其父元素为 P 的相关本名数据项 EI，一个 XBRL 处理器可推断出存在一个数据项 EI，EI 有一个与非空别名项集 S 的所有成员值（非空值）值等价并取可取的最大精度和小数值，与集合 S 相应的本名别名弧是以 E 作为本名概念、P 为双亲。如果所有 S 的所有成员（非空）不是值等价的，那么该 XBRL 实例与它在 DTS 中所表达的定义链接语义是不一致的，那么具有完全一致性的 XBRL 处理器应能够检测这种不一致性。如果一个应用软件程序采用这条规则，并且集合 S 的任意成员 M 没有提供一个值，或者有一个空值，但是本名数据项存在于“本名—别名”弧的某些集合中，那么该规则应被递归地应用在推理 E 的值之前推理 M 值。

XBRL 处理器不需要从本名数据项的值去推断别名数据项的值，而且在本部分中也没有定义这样的规则。

示例 2：

对由“本名—别名”弧链接的数值型数据项概念的值的推断

案例 1

概念 A，B，C 由本名—别名弧链接，A 是本名，B，C 是别名。在 XBRL 实例中，B 有一个精度为 2 的值 110，C 有一个精度为 2 的值 99。A，B，C 是上下文等价的。

B，C 的值在它们指定的精度 2 上是不一致的。所以，对 A 不能进行推断。

案例 2

概念 A，B，C 由本名—别名弧链接，A 是本名，B，C 是别名。在 XBRL 实例中，B 有一个精度为 1 的值 110，C 有一个精度为 1 的值 99。A，B，C 是上下文等价的。

四舍五入 B 的精度到 1 的结果是 100。

四舍五入 C 的精度到 1 的结果是 100。

这两个值是相等的，可推断 A 在精度是 1 时，值为 100。

9.2.6.3.4 “相似元组”弧

“相似元组”（similar-tuples）弧角色值定义的弧元素只能表达存在于元组替换组中概念间的关系。

“相似元组”弧表达了具有等价定义的元组概念间的关系，即使它们有不同的 XML 内容模型。

“相似元组”弧的语义关系是对称的。弧源于哪个元组与弧到哪个元组去是没有关系的。

9.2.6.3.5 “必需元素”弧

“必需元素”（requires-element）弧角色值定义的弧元素只能表达概念（根据定义，这些概念是在元组或数据项替换组中的）间的关系。

如果在一个 XBRL 实例中出现了弧的来源的概念实例，那么在该 XBRL 实例中也应出现弧的目标概念实例。当测试该需求时，对这些概念实例的上下文等价和单位等价没有要求。同样，该需求也没有在元组概念实例的相对位置上加以要求。具有完全一致性的 XBRL 处理器应检测和通告违反了该关系的实例。

在“必需元素”弧的网络中，允许环的存在。

附 录 A
（规范性附录）
模 式 文 件

A.1 xbrl-instance-2003-12-31.xsd

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

  <annotation>
    <documentation>
    Taxonomy schema for XBRL. This schema defines syntax relating to
    XBRL instances.
    </documentation>
  </annotation>

  <import namespace = "http://www.xbrl.org/2003/linkbase"
    schemaLocation = "xbrl-linkbase-2003-12-31.xsd" />

  <annotation>
    <documentation>
    Define the attributes to be used on XBRL concept definitions
    </documentation>
  </annotation>

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

  <attribute name = "balance">
    <annotation>
      <documentation>
      The balance attribute (imposes calculation relationship restrictions)
      </documentation>
    </annotation>
    <simpleType>
```

```
    〈restriction base = "token"〉
      〈enumeration value = "debit" /〉
      〈enumeration value = "credit" /〉
    〈/restriction〉
  〈/simpleType〉
〈/attribute〉

〈annotation〉
  〈documentation〉
  Define the simple types used as a base for for item types
  〈/documentation〉
〈/annotation〉

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

〈simpleType name = "shares"〉
  〈annotation〉
    〈documentation〉
    This datatype serves as the datatype for share based
    financial concepts.
    〈/documentation〉
  〈/annotation〉
  〈restriction base = "decimal" /〉
〈/simpleType〉

〈simpleType name = "pure"〉
  〈annotation〉
    〈documentation〉
    This datatype serves as the type for dimensionless numbers
    such as percentage change, growth rates, and other ratios
    where the numerator and denominator have the same units.
    〈/documentation〉
  〈/annotation〉
  〈restriction base = "decimal" /〉
〈/simpleType〉

〈simpleType name = "nonZeroDecimal"〉
  〈annotation〉
    〈documentation〉
    As the name implies this is a decimal value that can not take
    the value 0-it is used as the type for the denominator of a
    fractionItemType.
    〈/documentation〉
  〈/annotation〉
  〈union〉
    〈simpleType〉
```

```
        〈restriction base = "decimal"〉
          〈minExclusive value = "0" /〉
        〈/restriction〉
      〈/simpleType〉
      〈simpleType〉
        〈restriction base = "decimal"〉
          〈maxExclusive value = "0" /〉
        〈/restriction〉
      〈/simpleType〉
    〈/union〉
  〈/simpleType〉

  〈simpleType name = "precisionType"〉
    〈annotation〉
      〈documentation〉
      This type is used to specify the value of the
      precision attribute on numeric items.  It consists
      of the union of nonNegativeInteger and "INF" (used
      to signify infinite precision or "exact value").
      〈/documentation〉
    〈/annotation〉
    〈union memberTypes = "nonNegativeInteger"〉
      〈simpleType〉
        〈restriction base = "string"〉
          〈enumeration value = "INF" /〉
        〈/restriction〉
      〈/simpleType〉
    〈/union〉
  〈/simpleType〉

  〈simpleType name = "decimalsType"〉
    〈annotation〉
      〈documentation〉
      This type is used to specify the value of the decimals attribute
      on numeric items.  It consists of the union of integer and "INF"
      (used to signify that a number is expressed to an infinite number
      of decimal places or "exact value").
      〈/documentation〉
    〈/annotation〉
    〈union memberTypes = "integer "〉
      〈simpleType〉
        〈restriction base = "string"〉
          〈enumeration value = "INF" /〉
        〈/restriction〉
      〈/simpleType〉
    〈/union〉
  〈/simpleType〉

  〈attributeGroup name = "numericItemAttrs"〉
    〈annotation〉
      〈documentation〉
      Group of attributes for numeric items
      〈/documentation〉
    〈/annotation〉
    〈attribute name = "contextRef" type = "IDREF" use = "required" /〉
    〈attribute name = "unitRef" type = "IDREF" use = "required" /〉
```

```
  <attribute name = "precision" type = "xbrli:precisionType" use = "optional" />
  <attribute name = "decimals" type = "xbrli:decimalsType" use = "optional" />
  <attribute name = "id" type = "ID" use = "optional" />
</attributeGroup>

<attributeGroup name = "nonNumericItemAttrs">
  <annotation>
    <documentation>
    Group of attributes for non-numeric items
    </documentation>
  </annotation>
  <attribute name = "contextRef" type = "IDREF" use = "required" />
  <attribute name = "id" type = "ID" use = "optional" />
</attributeGroup>

<annotation>
  <documentation>
  General numeric item types-for use on concept element definitions
  The following 3 numeric types are all based on the built-in
  data types of XML Schema.
  </documentation>
</annotation>

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

<complexType name = "floatItemType" final = "extension">
  <simpleContent>
    <extension base = "float">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = " # # other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "doubleItemType" final = "extension">
  <simpleContent>
    <extension base = "double">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = " # # other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<annotation>
  <documentation>
  XBRL domain numeric item types-for use on concept element definitions
  The following 4 numeric types are all types that have been identified as
  having particular relevance to the domain space addressed by XBRL and are
  hence included in addition to the built-in types from XML Schema.
```

```
    </documentation>
  </annotation>

  <complexType name = "monetaryItemType" final = "extension">
    <simpleContent>
      <extension base = "xbrli:monetary">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "sharesItemType" final = "extension">
    <simpleContent>
      <extension base = "xbrli:shares">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "pureItemType" final = "extension">
    <simpleContent>
      <extension base = "xbrli:pure">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

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

  <annotation>
    <documentation>
    The following 13 numeric types are all based on the XML Schema
    built-in types that are derived by restriction from decimal.
    </documentation>
  </annotation>

  <complexType name = "integerItemType" final = "extension">
    <simpleContent>
      <extension base = "integer">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>
```

```
<complexType name = "nonPositiveIntegerItemType" final = "extension">
  <simpleContent>
    <extension base = "nonPositiveInteger">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "negativeIntegerItemType" final = "extension">
  <simpleContent>
    <extension base = "negativeInteger">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "longItemType" final = "extension">
  <simpleContent>
    <extension base = "long">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "intItemType" final = "extension">
  <simpleContent>
    <extension base = "int">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "shortItemType" final = "extension">
  <simpleContent>
    <extension base = "short">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "byteItemType" final = "extension">
  <simpleContent>
    <extension base = "byte">
      <attributeGroup ref = "xbrli:numericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "nonNegativeIntegerItemType" final = "extension">
```

```
    <simpleContent>
      <extension base = "nonNegativeInteger">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "unsignedLongItemType" final = "extension">
    <simpleContent>
      <extension base = "unsignedLong">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "unsignedIntItemType" final = "extension">
    <simpleContent>
      <extension base = "unsignedInt">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "unsignedShortItemType" final = "extension">
    <simpleContent>
      <extension base = "unsignedShort">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "unsignedByteItemType" final = "extension">
    <simpleContent>
      <extension base = "unsignedByte">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "positiveIntegerItemType" final = "extension">
    <simpleContent>
      <extension base = "positiveInteger">
        <attributeGroup ref = "xbrli:numericItemAttrs" />
      </extension>
    </simpleContent>
  </complexType>

  <annotation>
    <documentation>
    The following 17 non-numeric types are all based on the primitive built-in
    data types of XML Schema.
```

```
    </documentation>
  </annotation>

  <complexType name = "stringItemType" final = "extension">
    <simpleContent>
      <extension base = "string">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "booleanItemType" final = "extension">
    <simpleContent>
      <extension base = "boolean">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "hexBinaryItemType" final = "extension">
    <simpleContent>
      <extension base = "hexBinary">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "base64BinaryItemType" final = "extension">
    <simpleContent>
      <extension base = "base64Binary">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "anyURIItemType" final = "extension">
    <simpleContent>
      <extension base = "anyURI">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "QNameItemType" final = "extension">
    <simpleContent>
      <extension base = "QName">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = "##other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>
```

```
  <complexType name = "durationItemType" final = "extension">
    <simpleContent>
      <extension base = "duration">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = " # # other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "dateTimeItemType" final = "extension">
    <simpleContent>
      <extension base = "xbrli:dateUnion">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = " # # other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "timeItemType" final = "extension">
    <simpleContent>
      <extension base = "time">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = " # # other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "dateItemType" final = "extension">
    <simpleContent>
      <extension base = "date">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = " # # other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "gYearMonthItemType" final = "extension">
    <simpleContent>
      <extension base = "gYearMonth">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = " # # other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "gYearItemType" final = "extension">
    <simpleContent>
      <extension base = "gYear">
        <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
        <anyAttribute namespace = " # # other" processContents = "lax" />
      </extension>
    </simpleContent>
  </complexType>

  <complexType name = "gMonthDayItemType" final = "extension">
```

```
  <simpleContent>
    <extension base = "gMonthDay">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "gDayItemType" final = "extension">
  <simpleContent>
    <extension base = "gDay">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "gMonthItemType" final = "extension">
  <simpleContent>
    <extension base = "gMonth">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
      <anyAttribute namespace = "##other" processContents = "lax" />
    </extension>
  </simpleContent>
</complexType>

<annotation>
  <documentation>
  The following 4 non-numeric types are all based on the XML Schema
  built-in types that are derived by restriction and/or list from string.
  </documentation>
</annotation>

<complexType name = "normalizedStringItemType" final = "extension">
  <simpleContent>
    <extension base = "normalizedString">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "tokenItemType" final = "extension">
  <simpleContent>
    <extension base = "token">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "languageItemType" final = "extension">
  <simpleContent>
    <extension base = "language">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

```
<complexType name = "NameItemType" final = "extension">
  <simpleContent>
    <extension base = "Name">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

<complexType name = "NCNameItemType" final = "extension">
  <simpleContent>
    <extension base = "NCName">
      <attributeGroup ref = "xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

<annotation>
  <documentation>
  XML Schema components contributing to the context element
  </documentation>
</annotation>

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

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

```
  </sequence>
</complexType>

<simpleType name = "dateUnion">
  <annotation>
    <documentation>
    The union of the date and dateTime simple types.
    </documentation>
  </annotation>
  <union memberTypes = "date dateTime " />
</simpleType>

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

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

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

```
  </element>

  <annotation>
    <documentation>
    XML Schema components contributing to the unit element
    </documentation>
  </annotation>

  <element name = "measure" type = "QName" />

  <complexType name = "measuresType">
    <annotation>
      <documentation>
      A collection of sibling measure elements
      </documentation>
    </annotation>
    <sequence>
      <element ref = "xbrli:measure" minOccurs = "1" maxOccurs = "unbounded" />
    </sequence>
  </complexType>

  <element name = "divide">
    <annotation>
      <documentation>
      Element used to represent division in units
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name = "unitNumerator" type = "xbrli:measuresType" />
        <element name = "unitDenominator" type = "xbrli:measuresType" />
      </sequence>
    </complexType>
  </element>

  <element name = "unit">
    <annotation>
      <documentation>
      Element used to represent units information about numeric items
      </documentation>
    </annotation>
    <complexType>
      <choice>
        <element ref = "xbrli:measure" minOccurs = "1" maxOccurs = "unbounded" />
        <element ref = "xbrli:divide" />
      </choice>
      <attribute name = "id" type = "ID" use = "required" />
    </complexType>
  </element>

  <annotation>
    <documentation>
    Elements to use for facts in instances
    </documentation>
  </annotation>
```

```
  〈element name = "item" type = "anyType" abstract = "true"〉
    〈annotation〉
      〈documentation〉
      Abstract item element used as head of item substitution
      〈/documentation〉
    〈/annotation〉
  〈/element〉

  〈element name = "tuple" type = "anyType" abstract = "true"〉
    〈annotation〉
      〈documentation〉
      Abstract tuple element used as head of tuple substitution
      〈/documentation〉
    〈/annotation〉
  〈/element〉

  〈element name = "xbrl"〉
    〈annotation〉
      〈documentation〉
      XBRL instance root element.
      〈/documentation〉
    〈/annotation〉
    〈complexType〉
      〈sequence〉
        〈element ref = "link:schemaRef" minOccurs = "1" maxOccurs = "unbounded" /〉
        〈element ref = "link:linkbaseRef" minOccurs = "0" maxOccurs = "unbounded" /〉
        〈element ref = "link:roleRef" minOccurs = "0" maxOccurs = "unbounded" /〉
        〈element ref = "link:arcroleRef" minOccurs = "0" maxOccurs = "unbounded" /〉
        〈choice minOccurs = "0" maxOccurs = "unbounded"〉
          〈element ref = "xbrli:item"/〉
          〈element ref = "xbrli:tuple"/〉
          〈element ref = "xbrli:context"/〉
          〈element ref = "xbrli:unit"/〉
          〈element ref = "link:footnoteLink"/〉
        〈/choice〉
      〈/sequence〉
      〈attribute name = "id" type = "ID" use = "optional" /〉
      〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" /〉
    〈/complexType〉
  〈/element〉

〈/schema〉
```

A.2 xbrl-linkbase-2003-12-31.xsd

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

  〈annotation〉
```

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

  <import namespace = "http://www.xbrl.org/2003/XLink"
    schemaLocation = "xl-2003-12-31.xsd"/>

  <import namespace = "http://www.w3.org/1999/xlink"
    schemaLocation = "xlink-2003-12-31.xsd"/>

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

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

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

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

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

  <element name = "presentationArc" substitutionGroup = "xl:arc">
```

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

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

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

〈element name = "label" substitutionGroup = "xl:resource"〉
  〈annotation〉
    〈documentation〉
    Definition of the label  resource element.
    〈/documentation〉
  〈/annotation〉
  〈complexType mixed = "true"〉
    〈complexContent mixed = "true"〉
      〈extension base = "xl:resourceType"〉
```

```
            <sequence>
              <any namespace = "http://www.w3.org/1999/xhtml" processContents = "skip" minOccurs = "0"
maxOccurs = "unbounded"/>
            </sequence>
            <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
          </extension>
        </complexContent>
      </complexType>
    </element>

    <element name = "part" type = "string" abstract = "true">
      <annotation>
        <documentation>
        Definition of the reference  part element-for use in reference  resources.
        </documentation>
      </annotation>
    </element>

    <element name = "reference" substitutionGroup = "xl:resource">
      <annotation>
        <documentation>
        Definition of the reference  resource element.
        </documentation>
      </annotation>
      <complexType mixed = "true">
        <complexContent mixed = "true">
          <extension base = "xl:resourceType">
            <sequence>
              <element ref = "link:part" minOccurs = "0" maxOccurs = "unbounded"/>
            </sequence>
          </extension>
        </complexContent>
      </complexType>
    </element>

    <element name = "footnote" substitutionGroup = "xl:resource">
      <annotation>
        <documentation>
        Definition of the reference  resource element
        </documentation>
      </annotation>
      <complexType mixed = "true">
        <complexContent mixed = "true">
          <extension base = "xl:resourceType">
            <sequence>
              <any namespace = "http://www.w3.org/1999/xhtml" processContents = "skip" minOccurs = "0"
maxOccurs = "unbounded"/>
            </sequence>
            <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
          </extension>
        </complexContent>
      </complexType>
    </element>

    <element name = "presentationLink" substitutionGroup = "xl:extended">
      <annotation>
```

```
    〈documentation〉
    presentation extended link element definition.
    〈/documentation〉
  〈/annotation〉
  〈complexType〉
    〈complexContent〉
      〈restriction base = "xl:extendedType"〉
        〈choice minOccurs = "0" maxOccurs = "unbounded"〉
          〈element ref = "xl:title"/〉
          〈element ref = "link:documentation"/〉
          〈element ref = "link:loc"/〉
          〈element ref = "link:presentationArc"/〉
        〈/choice〉
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" /〉
      〈/restriction〉
    〈/complexContent〉
  〈/complexType〉
〈/element〉

〈element name = "definitionLink" substitutionGroup = "xl:extended"〉
  〈annotation〉
    〈documentation〉
    definition extended link element definition
    〈/documentation〉
  〈/annotation〉
  〈complexType〉
    〈complexContent〉
      〈restriction base = "xl:extendedType"〉
        〈choice minOccurs = "0" maxOccurs = "unbounded"〉
          〈element ref = "xl:title"/〉
          〈element ref = "link:documentation"/〉
          〈element ref = "link:loc"/〉
          〈element ref = "link:definitionArc"/〉
        〈/choice〉
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" /〉
      〈/restriction〉
    〈/complexContent〉
  〈/complexType〉
〈/element〉

〈element name = "calculationLink" substitutionGroup = "xl:extended"〉
  〈annotation〉
    〈documentation〉
    calculation  extended link element definition
    〈/documentation〉
  〈/annotation〉
  〈complexType〉
    〈complexContent〉
      〈restriction base = "xl:extendedType"〉
        〈choice minOccurs = "0" maxOccurs = "unbounded"〉
          〈element ref = "xl:title"/〉
          〈element ref = "link:documentation"/〉
          〈element ref = "link:loc"/〉
          〈element ref = "link:calculationArc"/〉
        〈/choice〉
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" /〉
```

```
        </restriction>
      </complexContent>
    </complexType>
  </element>

  <element name = "labelLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
      label extended link element definition
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base = "xl:extendedType">
          <choice minOccurs = "0" maxOccurs = "unbounded">
            <element ref = "xl:title"/>
            <element ref = "link:documentation"/>
            <element ref = "link:loc"/>
            <element ref = "link:labelArc"/>
            <element ref = "link:label"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

  <element name = "referenceLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
      reference extended link element definition
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <restriction base = "xl:extendedType">
          <choice minOccurs = "0" maxOccurs = "unbounded">
            <element ref = "xl:title"/>
            <element ref = "link:documentation"/>
            <element ref = "link:loc"/>
            <element ref = "link:referenceArc"/>
            <element ref = "link:reference"/>
          </choice>
          <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" />
        </restriction>
      </complexContent>
    </complexType>
  </element>

  <element name = "footnoteLink" substitutionGroup = "xl:extended">
    <annotation>
      <documentation>
      footnote extended link element definition
      </documentation>
    </annotation>
    <complexType>
```

```
    〈complexContent〉
      〈restriction base = "xl:extendedType"〉
        〈choice minOccurs = "0" maxOccurs = "unbounded"〉
          〈element ref = "xl:title"/〉
          〈element ref = "link:documentation"/〉
          〈element ref = "link:loc"/〉
          〈element ref = "link:footnoteArc"/〉
          〈element ref = "link:footnote"/〉
        〈/choice〉
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" /〉
      〈/restriction〉
    〈/complexContent〉
  〈/complexType〉
〈/element〉

〈element name = "linkbase"〉
  〈annotation〉
    〈documentation〉
    Definition of the linkbase element.  Used to
    contain a set of zero or more extended link elements.
    〈/documentation〉
  〈/annotation〉
  〈complexType〉
    〈choice minOccurs = "0" maxOccurs = "unbounded"〉
      〈element ref = "link:documentation"/〉
      〈element ref = "link:roleRef"/〉
      〈element ref = "link:arcroleRef"/〉
      〈element ref = "xl:extended"/〉
    〈/choice〉
    〈attribute name = "id" type = "ID" use = "optional"/〉
    〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/〉
  〈/complexType〉
〈/element〉

〈element name = "linkbaseRef" substitutionGroup = "xl:simple"〉
  〈annotation〉
    〈documentation〉
    Definition of the linkbaseRef element-used
    to link to XBRL taxonomy extended links from
    taxonomy schema documents and from XBRL
    instances.
    〈/documentation〉
  〈/annotation〉
  〈complexType〉
    〈complexContent〉
      〈restriction base = "xl:simpleType"〉
        〈attribute ref = "xlink:arcrole" use = "required"〉
          〈annotation〉
            〈documentation〉
            This attribute must have the value:
            http://www.w3.org/1999/xlink/properties/linkbase
            〈/documentation〉
          〈/annotation〉
        〈/attribute〉
        〈anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax" /〉
      〈/restriction〉
```

```
      </complexContent>
    </complexType>
  </element>

  <element name = "schemaRef" type = "xl:simpleType" substitutionGroup = "xl:simple">
    <annotation>
      <documentation>
      Definition of the schemaRef element-used
      to link to XBRL taxonomy schemas from
      XBRL instances.
      </documentation>
    </annotation>
  </element>

  <element name = "roleRef" substitutionGroup = "xl:simple">
    <annotation>
      <documentation>
      Definition of the roleRef element-used
      to link to resolve xlink:role attribute values to
      the roleType element declaration.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base = "xl:simpleType">
          <attribute name = "roleURI" type = "xlink:nonEmptyURI" use = "required">
            <annotation>
              <documentation>
                This attribute contains the role name.
              </documentation>
            </annotation>
          </attribute>
        </extension>
      </complexContent>
    </complexType>
  </element>

  <element name = "arcroleRef" substitutionGroup = "xl:simple">
    <annotation>
      <documentation>
      Definition of the roleRef element-used
      to link to resolve xlink:arcrole attribute values to
      the arcroleType element declaration.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base = "xl:simpleType">
          <attribute name = "arcroleURI" type = "xlink:nonEmptyURI" use = "required">
            <annotation>
              <documentation>
                This attribute contains the arc role name.
              </documentation>
            </annotation>
          </attribute>
        </extension>
```

```
    </complexContent>
  </complexType>
</element>

<element name = "definition" type = "string">
  <annotation>
    <documentation>
    The element to use for human-readable definition
    of custom roles and arc roles.
    </documentation>
  </annotation>
</element>

<element name = "usedOn" type = "QName">
  <annotation>
    <documentation>
    Definition of the usedOn element-used
    to identify what elements may use a
    taxonomy defined role or arc role value.
    </documentation>
  </annotation>
</element>

<element name = "roleType">
  <annotation>
    <documentation>
    The roleType element definition-used to define custom
    role values in XBRL extended links.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref = "link:definition" minOccurs = "0"/>
      <element ref = "link:usedOn" maxOccurs = "unbounded"/>
    </sequence>
    <attribute name = "roleURI" type = "xlink:nonEmptyURI" use = "required"/>
    <attribute name = "id" type = "ID"/>
  </complexType>
</element>

<element name = "arcroleType">
  <annotation>
    <documentation>
    The  arcroleType element definition-used to define custom
    arc role values in XBRL extended links.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref = "link:definition" minOccurs = "0"/>
      <element ref = "link:usedOn" maxOccurs = "unbounded"/>
    </sequence>
    <attribute name = "arcroleURI" type = "xlink:nonEmptyURI" use = "required"/>
    <attribute name = "id" type = "ID"/>
    <attribute name = "cyclesAllowed" use = "required">
      <simpleType>
```

```
            <restriction base = "NMTOKEN">
              <enumeration value = "any"/>
              <enumeration value = "undirected"/>
              <enumeration value = "none"/>
            </restriction>
          </simpleType>
        </attribute>
      </complexType>
    </element>

</schema>
```

A.3 xlink-2003-12-31.xsd

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

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

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

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

```
<attributeGroup name = "simpleType">
  <attribute name = "type" use = "required">
    <simpleType>
      <restriction base = "token">
        <enumeration value = "simple"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>

<attributeGroup name = "extendedType">
  <attribute name = "type" use = "required">
    <simpleType>
      <restriction base = "token">
        <enumeration value = "extended"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>

<attributeGroup name = "locatorType">
  <attribute name = "type" use = "required">
    <simpleType>
      <restriction base = "token">
        <enumeration value = "locator"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>

<attributeGroup name = "arcType">
  <attribute name = "type" use = "required">
    <simpleType>
      <restriction base = "token">
        <enumeration value = "arc"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>

<attributeGroup name = "resourceType">
  <attribute name = "type" use = "required">
    <simpleType>
      <restriction base = "token">
        <enumeration value = "resource"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>

<attributeGroup name = "titleType">
  <attribute name = "type" use = "required">
    <simpleType>
      <restriction base = "token">
        <enumeration value = "title"/>
      </restriction>
```

```
      </simpleType>
    </attribute>
  </attributeGroup>

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

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

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

</schema>
```

A.4 xl-2003-12-31.xsd

```
<? xml version = "1.0" encoding = "UTF-8"?>
<! -- (c) XBRL International.  See www.xbrl.org/legal -->
<schema targetNamespace = "http://www.xbrl.org/2003/XLink"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xl = "http://www.xbrl.org/2003/XLink"
  xmlns = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified"
```

```
attributeFormDefault = "unqualified">

<import namespace = "http://www.w3.org/1999/xlink" schemaLocation = "xlink-2003-12-31.xsd"/>

<complexType name = "documentationType">
  <annotation>
    <documentation>
    Element type to use for documentation of
    extended links and linkbases.
    </documentation>
  </annotation>
  <simpleContent>
    <extension base = "string">
      <anyAttribute namespace = "##other" processContents = "lax"/>
    </extension>
  </simpleContent>
</complexType>

<element name = "documentation" type = "xl:documentationType" abstract = "true">
  <annotation>
    <documentation>
    Abstract element to use for documentation of
    extended links and linkbases.
    </documentation>
  </annotation>
</element>

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

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

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

<simpleType name = "useEnum">
  <annotation>
    <documentation>
    Enumerated values for the use attribute on extended link arcs.
    </documentation>
  </annotation>
  <restriction base = "NMTOKEN">
    <enumeration value = "optional" />
    <enumeration value = "prohibited" />
  </restriction>
</simpleType>

<complexType name = "arcType">
  <annotation>
    <documentation>
    basic extended link arc type-extended where necessary for specific arcs
    Extends the generic arc type by adding use, priority and order attributes.
    </documentation>
  </annotation>
  <complexContent>
    <restriction base = "anyType">
      <sequence>
        <element ref = "xl:title" minOccurs = "0" maxOccurs = "unbounded" />
      </sequence>
      <attributeGroup ref = "xlink:arcType"/>
      <attribute ref = "xlink:from" use = "required" />
      <attribute ref = "xlink:to" use = "required" />
      <attribute ref = "xlink:arcrole" use = "required" />
```

```
      〈attribute ref = "xlink:title" use = "optional" /〉
      〈attribute ref = "xlink:show" use = "optional" /〉
      〈attribute ref = "xlink:actuate" use = "optional" /〉
      〈attribute name = "order" type = "decimal" use = "optional" /〉
      〈attribute name = "use" type = "xl:useEnum" use = "optional" /〉
      〈attribute name = "priority" type = "integer" use = "optional" /〉
      〈anyAttribute namespace = "# ##other" processContents = "lax" /〉
    〈/restriction〉
  〈/complexContent〉
〈/complexType〉
〈element name = "arc" type = "xl:arcType" abstract = "true"〉
  〈annotation〉
    〈documentation〉
    Abstract element to use as head of arc element substitution.
    〈/documentation〉
  〈/annotation〉
〈/element〉

〈complexType name = "resourceType"〉
  〈annotation〉
    〈documentation〉
    Generic type for the resource type element
    〈/documentation〉
  〈/annotation〉
  〈complexContent mixed = "true"〉
    〈restriction base = "anyType"〉
      〈attributeGroup ref = "xlink:resourceType"/〉
      〈attribute ref = "xlink:label" use = "required" /〉
      〈attribute ref = "xlink:role" use = "optional" /〉
      〈attribute ref = "xlink:title" use = "optional" /〉
      〈attribute name = "id" type = "ID" use = "optional" /〉
    〈/restriction〉
  〈/complexContent〉
〈/complexType〉
〈element name = "resource" type = "xl:resourceType" abstract = "true"〉
  〈annotation〉
    〈documentation〉
    Abstract element to use as head of resource element substitution.
    〈/documentation〉
  〈/annotation〉
〈/element〉

〈complexType name = "extendedType"〉
  〈annotation〉
    〈documentation〉
    Generic extended link type
    〈/documentation〉
  〈/annotation〉
  〈complexContent〉
    〈restriction base = "anyType"〉
      〈choice minOccurs = "0" maxOccurs = "unbounded"〉
        〈element ref = "xl:title" /〉
        〈element ref = "xl:documentation" /〉
        〈element ref = "xl:locator" /〉
        〈element ref = "xl:arc" /〉
        〈element ref = "xl:resource" /〉
```

```
        </choice>
        <attributeGroup ref = "xlink:extendedType"/>
        <attribute ref = "xlink:role" use = "required" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute name = "id" type = "ID" use = "optional" />
        <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
      </restriction>
    </complexContent>
  </complexType>
  <element name = "extended" type = "xl:extendedType" abstract = "true">
    <annotation>
      <documentation>
      Abstract extended link element at head of extended link substitution.
      </documentation>
    </annotation>
  </element>

  <complexType name = "simpleType">
    <annotation>
      <documentation>
      Type for the simple links defined in XBRL
      </documentation>
    </annotation>
    <complexContent>
      <restriction base = "anyType">
        <attributeGroup ref = "xlink:simpleType"/>
        <attribute ref = "xlink:href" use = "required" />
        <attribute ref = "xlink:arcrole" use = "optional" />
        <attribute ref = "xlink:role" use = "optional" />
        <attribute ref = "xlink:title" use = "optional" />
        <attribute ref = "xlink:show" use = "optional" />
        <attribute ref = "xlink:actuate" use = "optional" />
        <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
      </restriction>
    </complexContent>
  </complexType>
  <element name = "simple" type = "xl:simpleType" abstract = "true">
    <annotation>
      <documentation>
      The abstract element at the head of the simple link substitution.
      </documentation>
    </annotation>
  </element>

</schema>
```

附 录 B
（资料性附录）
元 素 表

XBRL 元素	说 明
Attribute	属性
Divide	除
Documentation	注释
essence-alias	本名—别名关系
footnoteArc	脚注弧
footnoteLink	脚注链接
Instance	实例
Label	标签
labelArc	标签弧
labelLink	标签链接
Link	链接
Linkbase	链接库
linkbaseRef	链接库引用
Loc	定位
Measure	度量
Namespace	命名空间
Presentation	展示
presentationArc	展示弧
presentationLink	列报链接
Reference	引用
referenceArc	参考弧
referenceLink	引用链接
requirement-element	必需元素关系
Role	角色
roleRef	角色引用
Scenario	场景
Schema	模式文件
Schemaref	模式引用
Segment	段
similar-tuple	相似元组关系
substitution group	替换组
summation-item	加总关系

表（续）

XBRL 元素	说　明
Unit	单位
unitDenominator	单位分母
unitNumerator	单位分子
usedOn	使用于

附 录 C
（资料性附录）
属 性 表

XBRL 属性名称	释 义
Arcrole	弧角色
arcroleType	弧角色类型
arcroleURI	弧角色通用资源标识符
Balance	余额
Base	基准路径
contextRef	上下文引用
cyclesAllowed	允许出现环路
Decimals	小数位数
Href	超文本引用
Ref	引用
id,identifier	标识符
Name	名称
periodType	期间类型
Precision	整数位数
Priority	优先权
Role	角色
roleURI	角色通用资源标识符
Title	标题
Type	类型
unitRef	单位引用
Use	可用性
Weight	权重

附 录 D
（资料性附录）
数据类型表

数据类型	说 明
anyURI	任何通用资源标识符
anyURIItemType	任何通用资源标识符项类型
arcroleURI	弧角色通用资源标识符
base64Binary	基于 64 位的二进制数据
base64BinaryItemType	基于 64 位的二进制数据项类型
Boolean	布尔
booleanItemType	布尔项类型
Byte	字节
byteItemType	字节项类型
Date	日期
dateItemType	日期项类型
dateTime	日期与时间
dateTimeItemType	日期与时间项类型
decimalItemType	小数项类型
Double	双精度浮点数
doubleItemType	双精度浮点数项类型
Duration	时间段
durationItemType	时间段项类型
endDate	结束日期
Float	浮点数
floatItemType	浮点数项类型
fractionItemType	分数项类型
gDay	日
gDayItemType	日项类型
gMonth	月份
gMonthDay	月与天
gMonthDayItemType	月与天项类型
gMonthItemType	月份项类型
gYear	年份
gYearItemType	年份项类型
gYearMonth	年与月

表（续）

数据类型	说　明
gYearMonthItemType	年与月项类型
hexBinary	十六进制的二进制数据
hexBinaryItemType	十六进制的二进制数据项类型
Identical	同一的
IDREF	唯一标识符引用
Instant	时间点
Int	整型
Integer	整数
integerItemType	整数项类型
intItemType	整型项类型
languageItemType	语言项类型
Long	长整型
longItemType	长整型项类型
Measure	度量
Monetary	货币
monetaryItemType	货币项类型
NameItemType	姓名项类型
NCName	无前缀名称
NCNameItemType	无前缀名称项类型
negativeInteger	负整数
negativeIntegerItemType	负整数项类型
NMTOKEN	名称符号
nonNegativeInteger	非负整数
nonNegativeIntegerItemType	非负整数项类型
non-numeric item	非数值型数据项
nonPositiveInteger	非正整数
nonPositiveIntegerItemType	非正整数项类型
normalizedString	标准化字符串
normalizedStringItemType	标准化字符串项类型
numeric item	数值型数据项
positiveInteger	正整数
positiveIntegerItemType	正整数项类型
pureItemType	纯数项类型
Qname	带限定名名称

表（续）

数据类型	说　明
QNameItemType	带限定名名称项类型
roleType	角色类型
sharesItemType	股份项类型
Short	短整型
shortItemType	短整型项类型
startDate	起始日期
String	字符串
stringItemType	字符串项类型
Time	时间
timeItemType	时间项类型
Token	表征字符串
tokenItemType	表征字符串项类型
unsignedByte	无符号字节
unsignedByteItemType	无符号字节项类型
unsignedInt	无符号整型
unsignedIntItemType	无符号整型项类型
unsignedLong	无符号长整型
unsignedLongItemType	无符号长整型项类型
unsignedShort	无符号短整型
unsignedShortItemType	无符号短整型项类型

参 考 文 献

[1] http://www.w3.org/TR/REC-xml-names/
[2] http://www.ietf.org/rfc/rfc2119.txt
[3] http://www.w3.org/TR/xmlschema-0/
[4] http://www.w3.org/TR/xmlschema-1/
[5] http://www.w3.org/TR/xmlschema-2/
[6] http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=16387
[7] http://www.ietf.org/rfc/rfc2396.txt
[8] http://www.ietf.org/rfc/rfc2732.txt
[9] http://www.w3.org/TR/xhtml-modularization/
[10] http://www.w3.org/TR/xmlbase/
[11] http://www.w3.org/TR/xlink/
[12] http://www.w3.org/TR/xpath
[13] http://www.w3.org/TR/xptr-framework/