ICS 03.100.60
A 11

中华人民共和国国家标准

GB/T 25500.2—2010

可扩展商业报告语言(XBRL)技术规范 第2部分:维度

Extensible Business Reporting Language (XBRL) specification—
Part 2:Dimensions

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

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2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XBRL维度规范1.0版(XBRL Dimensions Specification 1.0)基本一致。

本部分由中华人民共和国财政部归口。

本部分起草单位：中华人民共和国财政部。

本部分主要起草人：应唯、王颖、李红霞、李敏敏、黄敏、覃东、杨海峰、杨诚、丁亮、朱健鹏、童盼盼。

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 GB/T 25500.1《可扩展商业报告语言(XBRL)技术规范　第 1 部分:基础》的一个扩展规范,提供了一个定义维度元数据并且在 XBRL 实例文档中对其进行引用的通用机制,以维度化的方式处理分类标准中的元素定义问题,可达到同一元素不同背景环境下的复用。

可扩展商业报告语言(XBRL)技术规范 第2部分:维度

1 范围

GB/T 25500的本部分规定了定义XBRL维度的元数据,以及在可扩展商业报告语言(XBRL)实例文档中对其进行引用的通用机制。

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的相关开发与应用。

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

GB/T 25500.1 可扩展商业报告语言(XBRL)技术规范 第1部分:基础

3 术语和定义

GB/T 25500.1中界定的以及下列术语和定义适用于本文件。

3.1

源及目标[概念] source and target [concept(s)]

在一个扩展链接元素中,定位器元素的href属性值所表示的一对概念。label属性内容和弧的from属性内容相同的是源[概念],label属性内容和弧的to属性内容相同的是目标[概念]。

3.2

关系 relationship

用弧(arc)的xlink:arcrole属性以及其他属性来定义的源概念和目标概念之间的联系。

3.3

维度 dimension

xbrldt:dimensionItem替换组中的抽象元素,能表征事实的各个不同的方面。

3.4

域 domain

由若干成员组成的集合,可能为空集、有限集或者无限集。

注:一个维度可能有多重维度-域关系。

3.5

有效域 effective domain

所有相关的域的联合集。

3.6

域成员 domain-member

维度的域中所有可能的值。

注:明确域由域-成员关系来定义。域成员项包含在数据项替换组中。

3.7

明确维度 explicit dimension

域为有限集合的维度。

注：明确维度的成员可以明确地命名。

3.8

类型化维度 typed dimension

域成员不能逐个枚举的维度。

3.9

类型化维度元素 typed dimension element

在上下文的段或场景中作为维度标识符而使用的非 XBRL 元素。

3.10

空维度 empty dimension

不包含任何域的明确维度。

3.11

维度分类标准 dimensional taxonomy

包含附录 A 中描述的弧的分类标准。

3.12

超立方体 hypercube

一个维度的集合，是参与到 has-hypercube 关系和 hypercube-dimension 关系中的 hypercubeItem 替换组中的抽象元素。

3.13

空超立方体 empty hypercube

没有维度的超立方体。

3.14

模板分类标准 template taxonomy

定义超立方体以及超立方体与基础项之间关系的分类标准。

3.15

维度关系集 dimensional relationship set

遍历基础集内部及跨基础集的关系所组成的集合。

注：维度关系集中可含有不同角色的扩展链接关系和不同弧角色的关系。

3.16

维度处理器 dimensional processor

处理 XBRL 维度实例文档或维度分类标准，并根据本部分中的规则检查输入文档一致性的程序。

4 命名空间及前缀

元素或者属性的命名空间前缀使用如 ns:name，其中 ns 是命名空间的前缀，name 是本地名称。关于命名空间前缀到实际命名空间的映射，本部分与表 1 保持一致。表 1 中的前缀列是非规范性的，命名空间 URI 列是规范性的。

表 1 命名空间和命名空间前缀

命名空间前缀	命名空间 URI
xbrldt	http://xbrl.org/2005/xbrldt

表 1（续）

命名空间前缀	命名空间 URI
xbrldi	http://xbrl.org/2006/xbrldi
xbrldte	http://xbrl.org/2005/xbrldt/errors
xbrldie	http://xbrl.org/2005/xbrldi/errors
xbrli	http://www.xbrl.org/2003/instance
xs	http://www.w3.org/2001/XMLSchema
xlink	http://www.w3.org/1999/xlink
Link	http://www.xbrl.org/2003/linkbase

5 维度分类标准

5.1 架构

5.1.1 概述

在 XBRL 实例中，本部分所定义的元素按惯例使用前缀“xbrldi”，其命名空间为 http://xbrl.org/2006/xbrldi，这些元素仅在 scenario 元素和 segment 元素中出现。XBRL 实例根据类型化维度和明确维度所对应的语法约束来进行校验。类型化维度对应的语法约束仅要求通过 XML 模式文件的校验，而明确维度对应的语法约束要求校验每个成员要素的描述以及使用链接库的成员之间的关系。

维度分类标准的特点是使用一系列的弧角色（arc role）。这些弧角色以及它们的属性声明位于 XML 模式文件的 appinfo 部分。

维度模式文件的命名空间为 http://xbrl.org/2005/xbrldt。在文档中使用前缀“xbrldt”，联系到模式文件中定义的元素和属性上去。

维度分类标准可以导入 xbrldt 模式文件，但应通过模式文件校验。根据本部分定义的维度分类标准应符合 GB/T 25500.1。

A.2 xbrldi-2006.xsd 中所定义元素的 XBRL 实例应通过 XML 模式文件的校验。如果 XBRL 实例的 DTS 中包含维度分类标准，则应是符合 GB/T 25500.1 规定的有效实例。

图 1 用图解的方式展示了多种不同的关系、元素在源和目标的类型，以及使用这些元素的不同目的。它们或者是基础项，或者是明确域成员，或者是代表整个维度（类型化维度或者明确维度）的根项。这些关系可以不在同一个扩展类型链接元素中。xbrldt:targetRole 属性用来在多个扩展类型链接元素之间将从基础项到成员的不同片联系起来。{all,notAll} 表示有 all 和 notall 两个可能的关系。元素的其他属性（abstract，xbrldt：typedDomainRef）或弧（xbrldt：closed，xbrldt：usable 和 xbrldt：contextElement）及其类型将通过其可能出现的弧来体现。

基础项声明（Primary Item Declaration）是指在 XBRL 分类标准中所定义、属于 xbrli:item 替换组，但不属于 xbrldt:hypercubeItem 或 xbrldt:dimensionItem 替换组的元素。

只有在 xbrli:item 的替换组中定义的 XBRL 项才可以作为明确维度的成员使用。

连续关系（Consecutive Relationship）是指按照 5.1.2 中定义的规则而连接在一起的两个关系。

维度关系集（Dimensional Relationship Set，DRS）是指连续关系的集合，用于代表基础项声明及其多维元数据之间的关系。

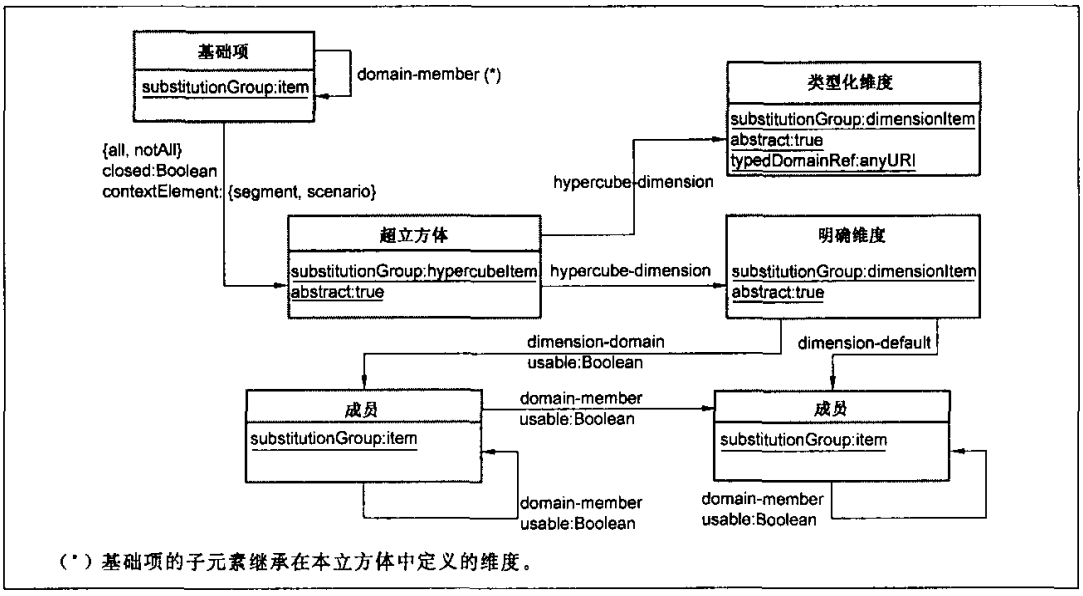


图 1 关于内容的约束与上下文含义的关系

5.1.2 连续关系

两个关系可以是连续的。一对连续关系包括一个起始关系和一个追随关系，两个关系之间满足以下条件即可构成连续关系：

a) 弧的用来描述起始关系的 xlink:arcrole 属性的值和用来描述追随关系的 xlink:arcrole 属性的值应符合表 2 中所列弧角色值的组合的某一项；

b) 由描述起始关系的弧的 xlink:to 属性所定义的定位器所指向的结点集，和由描述追随关系的弧的 xlink:from 属性所定义的定位器所指向的结点集，应是同一个集合。

表 2 可能的连续关系的弧角色值

起 始 弧	追 随 弧
All	hypercube-dimension
not-all	hypercube-dimension
hypercube-dimension	dimension-domain
dimension-domain	domain-member
domain-member	domain-member

5.2 超立方体

5.2.1 概述

超立方体声明(Hypercube Declaration)是 xbrldt:hypercubeItem 替换组的一个抽象的数据项声明。超立方体是一个维度的有序列表，由在维度关系集中通过 hypercube-dimension 关系而与超立方体连接起来的零个或多个维度声明的集合来定义，根据该关系的顺序属性来排序。hypercubeItem 元素的 XML 模式约束如下：

```
<xs:element
 name = "hypercubeItem"
 id = "xbrldt_hypercubeItem"
 abstract = "true"
 substitutionGroup = "xbrli:item"
 type = "xbrli:stringItemType"
 xbrli:periodType = "duration"/>
```

5.2.2 关于超立方体声明的约束

如果一个属于 xbrldt:hypercubeItem 替换组的元素不是抽象的，则维度处理器应能检查出这个错误并提示这个错误，错误用 xbrldte:HypercubeElementIsNotAbstractError 表示。

5.2.3 弧角色 http://xbrl.org/int/dim/arcrole/hypercube-dimension

hypercube-dimension 关系，将超立方体声明作为源，而将维度声明作为目标。

分类标准编辑工具中，用于分类标准表示目的的 hypercube-dimension 关系的顺序，由定义该关系的弧的顺序属性值来决定。

hypercube-dimension 关系角色不应含有任何定向的或非定向的环路。

hypercube-dimension 关系声明如下：

```
<arcroleType
 id = "hypercube-dimension"
 cyclesAllowed = "none"
 arcroleURI = "http://xbrl.org/int/dim/arcrole/hypercube-dimension">
 <definition>源(超立方体)包含目标(维度)</definition>
 <usedOn>definitionArc</usedOn>
</arcroleType>
```

以下示例展示了一个含有两种类型化维度(Team 和 Drink)的超立方体。该例表述的是一个在上下文的 segment 或 scenario 元素中对 Team 和 Drink 元素赋值的超立方体。

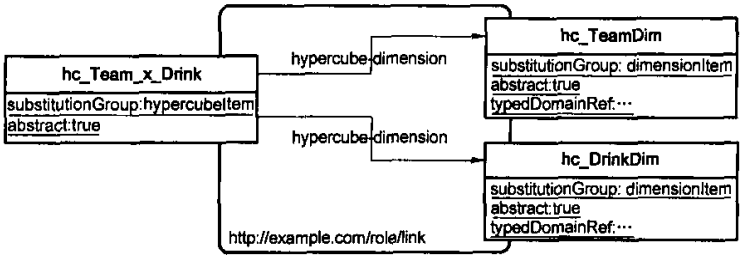
关于 hypercube-dimension 弧的源和目标有如下两条约束：

a) hypercube-dimension 弧的源应是超立方体声明。如果违反上述规则，维度处理器应通过 xbrldte:HypercubeDimensionSourceError 报错。

b) hypercube-dimension 弧的目标应是维度声明。如果违反上述规则，维度处理器应通过 xbrldte:HypercubeDimensionTargetErro 报错。

示例：

含有 Team 和 Drink 类型化维度的超立方体



5.3 基础项声明与超立方体

5.3.1 概述

为了约束可能出现于基础项的相关上下文集合，基础项声明可以与零个或多个超立方体相联系。

本部分中，未就其基础项声明与任何超立方体相联系的基础项来定义额外的约束。

超立方体集合可以通过“all”和“notAll”这两个操作符的组合来构成。操作符与其操作对象之间的关系可以使用 XLink 弧来描述，不同的操作员可以使用不同的弧角色。

下面两个弧角色，合称为 has-hypercube 关系：

a) http://xbrl.org/int/dim/arcrole/all；

b) http://xbrl.org/int/dim/arcrole/notAll。

这些关系可能在不同的基础集内。当 has-hypercube 关系在不同的基础集内时，在任何基础集内维度有效的基础项，就是维度有效的。

这些关系能接受在扩展分类标准中的禁用、覆盖或增加。

5.3.2 “all”和“notAll”弧角色

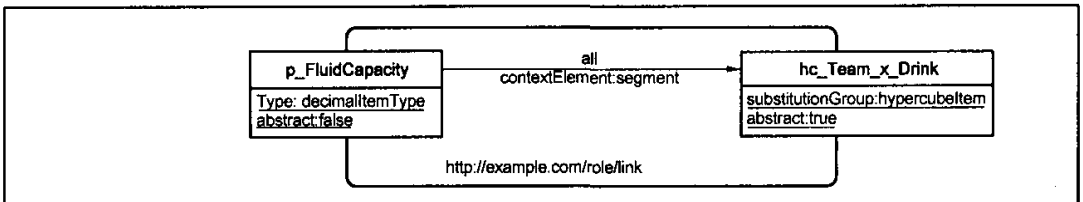
http://xbrl.org/int/dim/arcrole/all 关系中的维度关系集中的关系和实例校验是相关的。源和目标分别是基础项声明和超立方体声明。

“all”关系的否定是“notAll”关系，notAll 关系定义在 http://xbrl.org/int/dim/arcrole/notAll 中。

仅当针对所有联接的超立方体均为有效，实例文档中的基础项声明的实例化就超立方体联合而言才是维度有效的。如果同一个超立方体的非反定义是无效的，则反超立方体“notAll”是有效的。单一超立方体的联合就是超立方体本身，见示例 1、示例 2、示例 3。

示例 1：

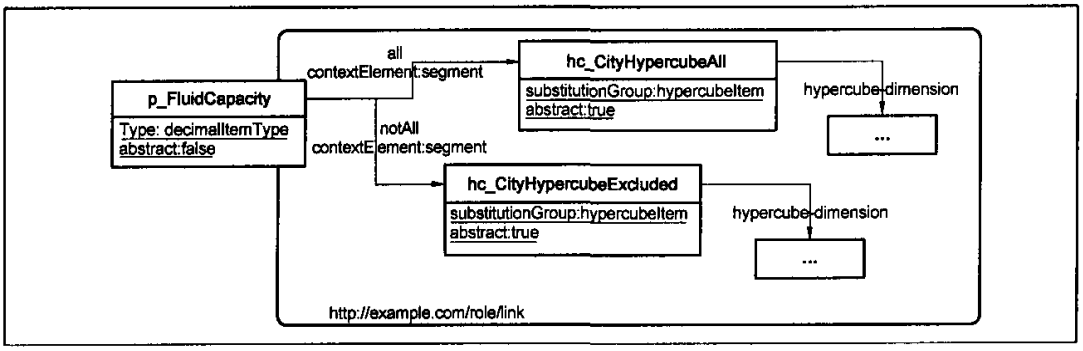
带单一超立方体的基础项声明



基础项声明 p_FluidCapacity 与一个超立方体相关联。仅当其引用了 Team 和 Drink 时，上下文就基础项而言才是维度有效的。

示例 2：

由“all”和“notAll”联合构成的两个超立方体的基础项声明



基础项声明 p_FluidCapacity 与属于同一基础集的两个超立方体的组合相关联。仅当其在 segment 中引用了 City 的相关内容时，上下文就基础项而言才是有效的，其中 segment 是 hc_CityHypercubeAll 的成员，且不是 hc_CityHypercubeExcluded 的成员。

在 http://xbrl.org/int/dim/arcrole/all 中的弧角色声明如下：

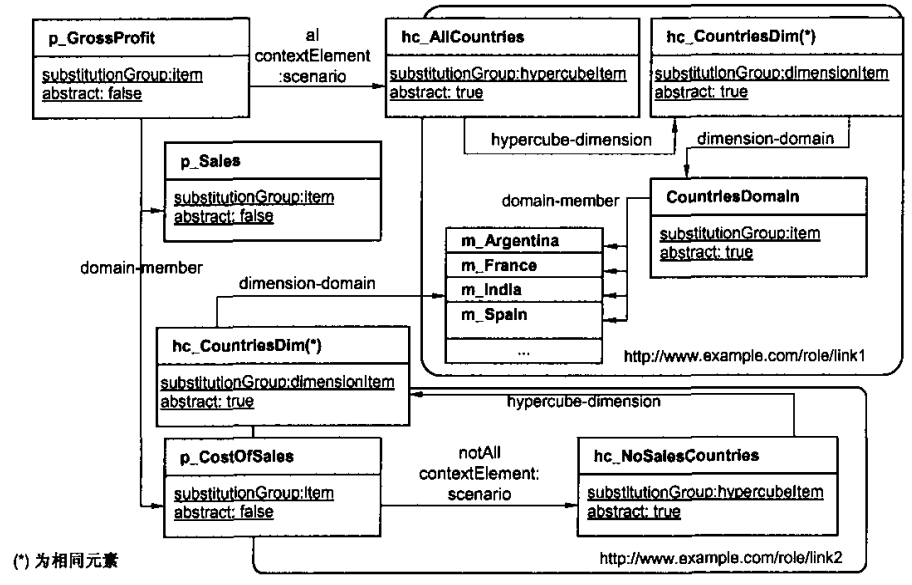
```
〈arcroleType
 id = "all"
 cyclesAllowed = "undirected"
 arcroleURI = "http://xbrl.org/int/dim/arcrole/all"〉
 〈definition〉源(基础项声明)要求目标(超立方体)的维度成员的组合出现在基础项的上下文里〈/definition〉
 〈usedOn〉definitionArc〈/usedOn〉
〈/arcroleType〉
```

在 http://xbrl.org/int/dim/arcrole/notAll 中的弧角色声明如下：

```
〈arcroleType
 id = "notAll"
 cyclesAllowed = "undirected"
 arcroleURI = "http://xbrl.org/int/dim/arcrole/notAll"〉
 〈definition〉源(基础项声明)要求目标(超立方体)的维度成员的组合不出现在基础项的上下文里〈/definition〉
 〈usedOn〉definitionArc〈/usedOn〉
〈/arcroleType〉
```

示例 3：

带有域成员和反超立方体的基础项，通过将 m_India 从域中移除来限制 p_CostOfSales 的 country 维度的值



基础项声明 p_GrossProfit 在 domain-member 关系网中有两个子项。对于 p_GrossProfit 和 p_Sales 来说，hc_CountriesDim 维度的有效的成员是{CountriesDomain | m_Argentina | m_France | m_India | m_Spain}，但是 p_CostOfSales 中关于 country 维度的有效成员仅为{CountriesDomain | m_Argentina | m_France | m_Spain}(因此 m_India 已从域中移除)

关于“all”或“notAll”弧的约束如下：

a) 如果“all”或“notAll”弧的源不是一个基础项声明，则维度处理器应通过 xbrldte:HasHypercubeSourceError 报错；

b) 如果“all”或“notAll”弧的目标不是一个超立方体声明，则维度处理器应通过 xbrldte:HasHypercubeTargetError 报错；

c) has-hypercube 弧应有一个 xbrldt:contextElement 属性。如果违反上述约束，则维度处理器应通过 xbrldte:HasHypercubeMissingContextElementAttributeError 报错。

5.3.3 “has-hypercube”弧中应有的“xbrldt:contextElement”属性

每个 has-hypercube 弧都应有 xbrldt:contextElement 属性。

```
<xs:simpleType name = "contextElementType">
 <xs:restriction base = "xs:token">
  <xs:enumeration value = "segment"/>
  <xs:enumeration value = "scenario"/>
      </xs:restriction>
</xs:simpleType>
<xs:attribute name = "contextElement" type = "xbrldt:contextElementType"/>
```

根据 XML 模式文件中关于 xbrldt 的规定，xbrldt:contextElement 属性应为 segment 值或 scenario 值的其中一项。

5.3.4 “has-hypercube”弧中可选的“xbrldt:closed”属性

可选的布尔类型属性 xbrldt:closed 可以出现在 has-hypercube 弧中。

```
<xs:attribute name = "closed" type = "xs:boolean" default = "false"/>
```

如果 has-hypercube 弧中的 xbrldt:closed 属性值为 true，且 xbrldt:contextElement 属性值为 segment，则超立方体对于基础集内 segment 元素是封闭的。

如果 has-hypercube 弧中的 xbrldt:closed 属性值为 true，且 xbrldt:contextElement 属性值为 scenario，则超立方体对于基础集内 scenario 元素是封闭的。

作为实例文档中的一个事实，当满足以下条件时，就超立方体而言，基础项声明的实例化过程才是维度有效的：

a) 基础项存在 has-hypercube 关系的超立方体的任意维度，满足以下任一条件：

1) has-hypercube 关系中的 xbrldt:contextElement = "segment"，且事实的上下文的 segment 维度容器对于这个维度有一个维度值；

2) has-hypercube 关系中的 xbrldt:contextElement = "scenario"，且事实的上下文的 scenario 维度容器对于这个维度有一个维度值；

3) 维度有缺省值。

b) 作为实例文档中的一个事实，当满足以下条件时，对封闭超立方体而言，基础项声明的实例化才是维度有效的：

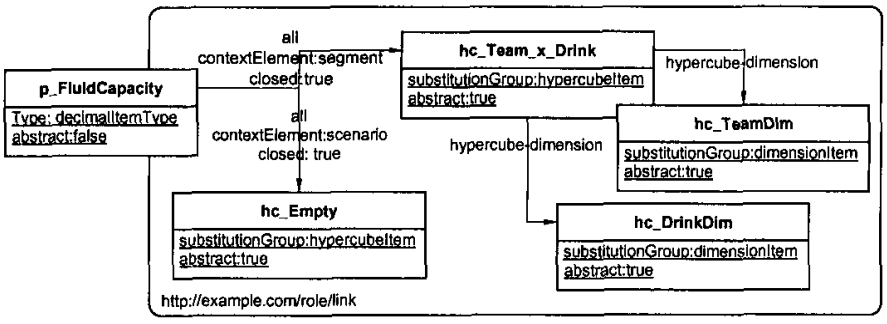
1） 关于超立方体的事实是维度有效的；

2） 关于事实的上下文的 segment 维度容器中的每个维度值，其维度即为封闭超立方体的维度；

3） 关于事实的上下文的 scenario 维度容器中的每个维度值，其维度即为封闭超立方体的维度。

示例：

两个封闭超立方体



如果弧的 segment 元素只含有 Team 和 Drink，scenario 元素为空，则弧的 xbrldt:closed="true"说明关于目标超立方体的上下文是有效的。需要注意的是，指向 hc_Team_x_Drink 的 all 弧的 xbrldt:contextElement 属性含有 segment 元素，且指向 hc_Empty 的 all 弧的 xbrldt:contextElement 属性含有 scenario 元素。

根据 XML 模式文件中关于 xbrldt 的规定，xbrldt:closed 属性一旦出现，则应有一个布尔值。

5.4 跨越不同基础集的维度关系集合的分割

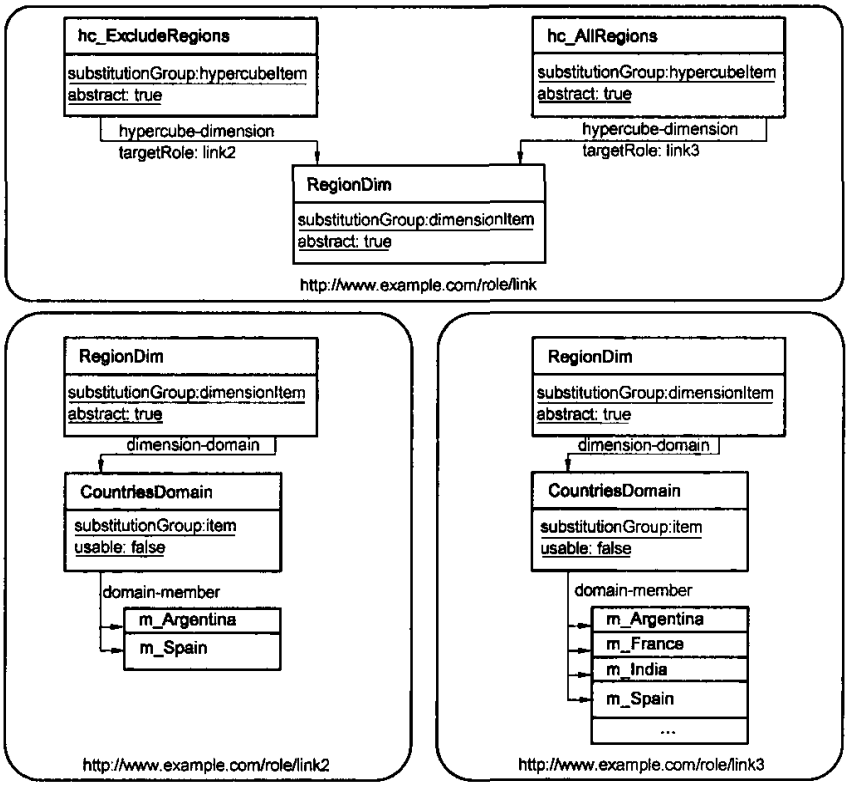
5.4.1 概述

分类标准的制定者可以利用扩展类型链接元素中的 xlink:role 属性将各种关系分割成不同的基础集。对计算链接库中的 Summation-Item 关系来说，对其进行分割，应保证不存在不相容的维度关系集合。分类标准的制定者可以指定对维度关系的不同基础集分别进行有效性检验。

此外，一个基础项声明集可以在多个 has-hypercube 关系的目标之间有相同的共享超立方体。同样，超立方体声明也可以在多个 hypercube-dimension 关系的目标之间有相同类型化维度。在后文中，新增的关系将会使关系网更加复杂，一些作为不同关系集的源的项目将用来定义不同的维度。

示例 1：

如果同一维度需要定义不同的域成员，则不同的扩展类型链接元素应进行分割，这里应使用一种能标示扩展链接流的机制。xbrldt:targetRole 正是用于此目的。



作为 hc_ExcludeRegions 立方体和 hc_AllRegions 立方体的一部分时，RegionDim 维度应含有不同的域成员。

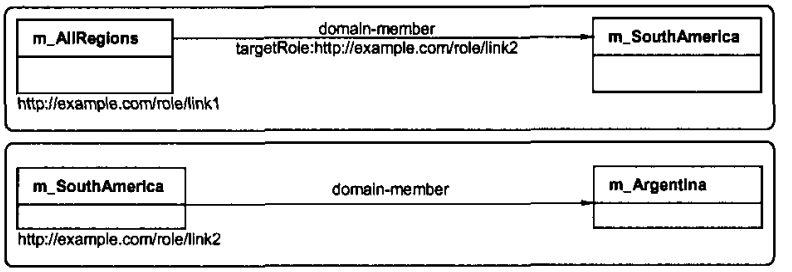
分类标准制定者可以使用弧的可选属性 xbrldt:targetRole，将存在于不同角色中的代表连续关系的两个弧连接在一起。xbrldt:targetRole 属性可以出现在定义弧中，它含有以下弧角色：all，notAll，hypercube-dimension，dimension-domain 和 domain-member。xbrldt:targetRole 属性有 anyURI 类型。URI 解析与 xml:base 属性是否出现无关，且其值应是一个绝对 URI。

```
<xs:attribute name = "targetRole" type = "xs:anyURI"/>
```

源角色(source role)是关系的基础集内 xlink:role 属性的内容。目标角色(target role)是弧本身的 targetRole 的属性的内容。在不同的扩展型链接元素中存在的、代表连续关系的两个弧，应使用 xbrldt:targetRole 属性将两者连接在一起。不这样做将使得连接无法建立并导致诸如空的超立方体、维度和域的出现。

示例 2：

基础集 link2 中的弧在基础集 link1 的弧的 DRS 中




```
〈definitionLink
 xlink:type = "extended" xlink:role = "http://example.com/role/link1" id = "link1"〉
  〈loc xlink:type = "locator"
   xlink:href = "m-2005-07-23.xsd # m_AllRegions"
   xlink:label = "AllRegions"/〉
  〈loc xlink:type = "locator"
   xlink:href = "m-2005-07-23.xsd # m_SouthAmerica"
   xlink:label = "SouthAmerica"/〉
  〈definitionArc xlink:type = "arc"
   xbrldt:targetRole = "http://example.com/role/link2"
   xlink:arcrole = "http://xbrl.org/int/dim/arcrole/domain-member"
   xlink:from = "AllRegions" xlink:to = "SouthAmerica" order = "1.0"/〉
〈/definitionLink〉
〈definitionLink xlink:type = "extended"
 xlink:role = "http://example.com/role/link2" id = "link2"〉
  〈loc xlink:type = "locator"
   xlink:href = "m-2005-07-23.xsd # m_SouthAmerica" xlink:label = "SouthAmerica"/〉
  〈loc xlink:type = "locator"
   xlink:href = "m-2005-07-23.xsd # m_Argentina" xlink:label = "Argentina"/〉
  〈definitionArc xlink:type = "arc"
   xlink:arcrole = "http://xbrl.org/int/dim/arcrole/domain-member"
   xlink:from = "SouthAmerica" xlink:to = "Argentina" order = "1.0"/〉
〈/definitionLink〉
```

xbrldt:targetRole 属性是可选的。如果它在一个关系中出现，任何表示源角色内连续关系的其他关系，均不得被视为是维度关系集的一部分。反而，表示目标角色中连续关系的关系，应被作为维度关系集的组成部分。表 3 给出了关系 A 与关系 B 的有效连续关系。

表 3　关系 A 与关系 B 的有效连续关系

关系 A 的弧角色	关系 B 的弧角色源于关系 A 的目标			
	all，notAll	hypercube-dimension	Dimension-domain	domain-member
all，notAll	False	role(B) ∈targetRole(A)	False	False
hypercube-dimension	False	False	role(B) ∈targetRole(A)	False
Dimension-domain	False	False	False	role(B) ∈targetRole(A)
Domain-member	False	False	False	role(B) ∈targetRole(A)

5.4.2　分割维度关系集对分类标准校验的影响

将维度关系集分割成多个角色将会影响 XBRL 分类标准的有效性。具有相同弧角色的连续关系

不得违反 cyclesAllowed 约束，该约束通常在基础集内适用。

xbrldt:targetRole 属性应包含一个经声明的角色。

5.4.3 分割维度关系集对实例校验的影响

将维度关系集分割成多个角色不会影响 XBRL 实例的有效性。无论维度关系集在一个还是多个角色中被定义，“基础项校验”所表达的含义是相同的。

5.4.4 对 xbrldt:targetRole 属性的约束

对 xbrldt:targetRole 属性的约束为：

a) 如果 definitionArc 的 xbrldt:targetRole 属性的值 V，不是一个标准的扩展链接角色(http://www.xbrl.org/2003/role/link)，则该 definitionArc 元素的祖先链接库元素应有一个 roleRef 子元素的 roleURI 属性的值为 V。如果这种约束不被满足，则维度处理器应报错 xbrldte:TargetRoleNotResolvedError。

b) 连续关系集(consecutive relationship set)，是始于“all”或“notAll”关系且后续连续关系紧随其后的关系集。维度处理器应对每个包含定向环路的连续关系集报错 xbrldte:DRSDirected-CycleError。

c) 根据 xbrldt 的 XML 模式文件，xbrldt:targetRole 属性的内容应是一个有效的 URI。

5.5 维度

5.5.1 概述

维度声明(dimensional declaration)是在 xbrldt:dimensionItem 替换组中的抽象的数据项声明。

```
<xs:element
name = "dimensionItem"
id = "xbrldt_dimensionItem"
abstract = "true"
substitutionGroup = "xbrli:item"
type = "xbrli:stringItemType"
xbrli:periodType = "duration"/>
```

维度项声明的 xbrli:balance、xbrli:periodType 和 nillable 属性没有意义。

本部分中有两个类型的维度：类型化维度和明确维度。维度声明通过使用一个 XBRL 实例的上下文元素中维度容器元素的 dimension 属性的 QName 来引用。维度容器元素的值，对明确维度而言，可以是 QName，对类型化维度而言，可以是复杂类型的 XML 元素。

一个非空的维度包含域，域中有成员。

对类型化维度而言，根据其 XML 模式文件定义而作的 XML 元素的实例化是域的成员，对于明确维度而言，其成员的 QName 为域的成员。

类型化维度的元素域是由一个 XML 模式文件中的全局元素定义来表示的。

明确维度的域是通过遍历连接维度与域、域与成员的弧而构成的。

5.5.2 对维度声明的约束

如果 xbrldt:dimensionItem 的替代组中的元素不是抽象的，则维度处理器应报错 xbrldte:DimensionElementIsNotAbstractError。

5.5.3 类型化维度

类型化维度(typed dimension)是一个维度声明,其成员组成的域是由另一个被 xbrldt:typedDomainRef 属性引用的 XML 元素来定义的。

类型化维度的 xbrldt:typedDomainRef 属性内容应为非空。

xbrldt:typedDomainRef 是定义维度域的 XML 模式文件中的对于元素声明的 xlink:href。

在以下的实例文档中,类型化维度值是有 dimension 属性的 xbrldi:typedMember 元素的子元素,而该 dimension 的值可用于定位类型化维度元素声明。

示例:

类型化维度元素及其域

tax.xsd 中的维度项声明	schema.xsd 中的域声明	instance.xbrl 中的域成员
<element name = "dCustomer" id = "tax_dCustomer" substitutionGroup = "xbrldt:dimensionItem" type = "xbrli:stringItemType" abstract = "true" xbrli:periodType = "duration" xbrldt:typedDomainRef = "schema.xsd#id_cust" />	<element name = "cust" id = "id_cust"> <simpleType> <restriction base = "string"> <pattern value = "[0-9][0-9][0-9][0-9][0-9]"/> </restriction> </simpleType> </element>	<xbrldi:typedMember dimension = "tax:dCustomer"> <cust>12345</cust> </xbrldi:typedMember> <xbrldi:typedMember dimension = "tax:dCustomer"> <cust>01742</cust> </xbrldi:typedMember>
<element name = "dPhone" id = "tax_dPhone" substitutionGroup = "xbrldt:dimensionItem" type = "xbrli:stringItemType" abstract = "true" xbrli:periodType = "duration" xbrldt:typedDomainRef = "schema.xsd#id_phone" />	<element name = "phone" id = "id_phone" xsi:nillable = "true"> <complexType> <sequence> <element name = "country" type = "integer"/> <element name = "city" type = "integer"/> <element name = "number" type = "integer"/> </sequence> </complexType> </element>	元素对于域类型有效,例如: <xbrldi:typedMember dimension = "tax:dPhone"> <phone> <country>7</country> <city>7</city> <number>5555555</number> </phone> </xbrldi:typedMember> <xbrldi:typedMember dimension = "tax:dPhone"> <phone xsi:nil = "true"/> </xbrldi:typedMember>

从在实例中实际出现的元素中分离维度项是必要的,因为 GB/T 25500.1 指明,定义链接库中的关系可能只有一个在 xbrli:item 或 xbrli:tuple 替换组中的目标,但对域本身而言,这样的限制既无必要,也不可取。

xbrldt:typedDomainRef 属性常用于在类型化维度中对在 XML 模式文件中定义类型化维度的内容的元素进行定位。

xbrldt:typedDomainRef 属性的值应是一个 URI 引用。xbrldt:typedDomainRef 的值应遵从本部分 XPointer 框架关于片段标识符的规定。

xbrldt:typedDomainRef 属性中引用的 URI 有一个 anyURI 类型。如果 URI 引用是相对的,则其绝对版本在使用之前应以 XML Base 规范的方法来进行计算。

```
<xs:attribute name = "typedDomainRef" type = "xs:anyURI"/>
```

由 xbrldt:typedDomainRef 指向的模式文件应是 DTS 的一部分。

xbrldt:typedDomainRef 属性的约束为：

a) 如果 xbrldt:typedDomainRef 属性指向的模式文件没有包含在分类标准的 DTS 中，则维度处理器将报错 xbrldte:OutOfDTSSchemaError。

b) 如果出现在的 XML 模式文件元素声明中的 xbrldt:typedDomainRef 属性不是维度声明，则维度处理器将报错 xbrldte:TypedDomainRefError。

c) 如果 xbrldt:typedDomainRef 属性发现以下情形(参照 xml:base 规范以及本部分中的相关定义)，则维度处理器将报错 xbrldte:TypedDimensionError：

1) 无；

2) 非全局 XML 模式文件元素声明 ；

3) 全局抽象元素声明。

d) 如果 xbrldt:typedDomainRef 属性没有包含片段标识符，则维度处理器将报错 xbrldte:TypedDimensionURIError。

e) 依照 xbrldt XML 模式文件，xbrldt:typedDomainRef 的属性内容应为一个有效的 URI。

5.5.4 明确维度

5.5.4.1 概述

明确维度(Explicit Dimension)是一个维度声明，其中不包含 xbrldt:typedDomainRef 属性，有连接零个或多个域成员声明的 dimension-domain 弧，其 QNames 构成维度域。

域成员声明(domain-member declaration)是分类标准中被定义在 xbrli:item 替换组中，而不在 xbrldt:hypercubeItem 或 xbrldt:dimensionItem 替换组中的元素。

明确维度的有效成员的域(domain of valid members of a explicit dimension)是源自同一个域成员的表示 domain-member 关系的维度关系集内所有可用元素的 QNames 的集合。这是去除被标记为不可用的元素的有效域。

域成员继承了所有 XBRL 项的特征，如多语言的标签、展示次序、引用、禁用、覆盖和增长等关系的扩展。这些域成员也可被安置在满足包含关系的需求的 domain-member 关系中。

明确维度的域由源为明确维度元素的 dimension-domain 关系的目标数据项来表示。域数据项的 QName 是一个有效的域成员。

依据图 1 定义的架构，基础项和明确维度域成员均在 xbrli:item 的替换组中。分类标准中定义的基础项能够在实例文件中扮演两个角色：一可作为其他项的上下文中的明确维度成员来使用；二可作为一个项使用。基础项的 QName 不应作为任何其明确维度的域成员。

示例：

明确维度元素及其域

分类标准模式文件中的维度数据项声明	分类标准模式文件中的域成员声明	XBRL 实例中的域成员
⟨xs:element name = "ShannxiDim" type = "xbrli:stringItemType" abstract = "true" substitutionGroup = "xbrldt:dimensionItem" nillable = "true" id = "geo_ShannxiDim" xbrli:periodType = "instant"/⟩	⟨xs:element name = "Xi'an" id = "geo_Xi'an" type = "xbrli:decimalItemType" substitutionGroup = "xbrli:item" nillable = "true" xbrli:periodType = "instant"/⟩ ⟨xs:element name = "Continent" id = "geo_Continent" type = "xbrli:decimalItemType" substitutionGroup = "xbrli:item" nillable = "true" xbrli:periodType = "instant"/⟩	QNames（假设 targetNamespace 的命名空间前缀是 geo，并且有一个从 Shannxi 到 Xi'an 的 domain-member 弧）： geo:Shannxi geo:Xi'an

5.5.4.2 弧角色 http://xbrl.org/int/dim/arcrole/dimension-domain

dimension-domain 关系将明确维度声明作为源，将域成员声明作为目标，并将维度和域绑定。

域声明的属性 xbrli:balance，xbrli:periodType 以及 nillable 并没有意义。

如 5.5.4.3 所示，弧可以具有非空的 xbrldt:usable 属性。

弧角色 http://xbrl.org/int/dim/arcrole/dimension-domain 声明如下：

```
<arcroleType
 id="dimension-domain"
 cyclesAllowed="none"
 arcroleURI="http://xbrl.org/int/dim/arcrole/dimension-domain">
 <definition>源(维度)只有目标(域)作为其域</definition>
 <usedOn>definitionArc</usedOn>
</arcroleType>
```

关于 dimension-domain 弧的约束为：

a) 如果弧的源不是一个明确维度声明，则维度处理器应报错 xbrldte:DimensionDomainSourceError。

b) 如果弧的目标不是一个域成员声明，则维度处理器应报错 xbrldte:DimensionDomainTargetError。

c) 如果存在一个环路，其中超立方体的基础项来源亦为有效成员域的其中一个成员，则维度处理器应报错 xbrldte:PrimaryItemPolymorphismError。

5.5.4.3 弧角色 http://xbrl.org/int/dim/arcrole/domain-member

domain-member 关系将域与其成员绑定。该关系的目的是创建明确域成员的集合。

明确维度的维度域(dimension-domain for explicit dimensions)是维度关系集的域成员声明的 Qnames 的集合，该维度关系集是基于 dimension-domain 弧的目标并和 domain-member 弧连接起来的。

domain-member 关系的基础集可以包含非定向的环路，但不应包含定向的环路。

弧角色 http://xbrl.org/int/dim/arcrole/domain-member 声明如下：

```
<arcroleType
 id="domain-member"
 cyclesAllowed="undirected"
 arcroleURI="http://xbrl.org/int/dim/arcrole/domain-member">
 <definition>源(域)包含目标(成员)</definition>
 <usedOn>definitionArc</usedOn>
</arcroleType>
```

关于 domain-member 弧的约束为：

a) 如果 domain-member 弧的源不是一个基础项声明，则维度处理器应报错 xbrldte:DomainMemberSourceError。

b) 如果 domain-member 弧的目标不是一个基础项声明，则维度处理器应报错 xbrldte:DomainMemberTargetError。

c) 如果存在这样一个环路：超立方体的基础项的来源同时是有效成员的域的一个成员，则维度处理器应报错 xbrldte: PrimaryItemPolymorphismError。

5.5.4.4 可选的 xbrldt:usable 属性

xbrldt:usable 属性用于标识不应在实例文档中被当作域值来使用的域成员。它允许将成员引入域成员层次中,以达到组织层次的目的。

xbrldt:usable 属性可以出现在弧 http://xbrl.org/int/dim/arcrole/dimension-domain 或者弧 http://xbrl.org/int/dim/arcrole/domain-member 中。

xbrldt:usable 属性的缺省值为 true。

如果某个弧的 xbrldt:usable 属性值为 false,则其目标将被排除在有效成员的域之外。

以上排除并不影响基于该等被排除的元素的 domain-member DRS 内的后继子元素。

维度的有效域(effective domain)是使用 dimension-domain 弧进行声明的所有维度域的并集,而该 dimension-domain 弧在维度关系集内是针对特定维度而存在的。

如果对在维度的有效域内进行的明确维度的有效成员的域进行赋值时,某成员在一个维度域中 xbrldt:usable=false 成立,且在另一维度域中 xbrldt:usable=true 成立,则应认为该成员被从明确维度的有效成员的域中有效排除。

对 xbrldt:usable 属性值的约束为,依据 xbrldt 的 XML 模式文件,xbrldt:usable 属性的值应为 Boolean。

5.6 domain-member 关系和继承

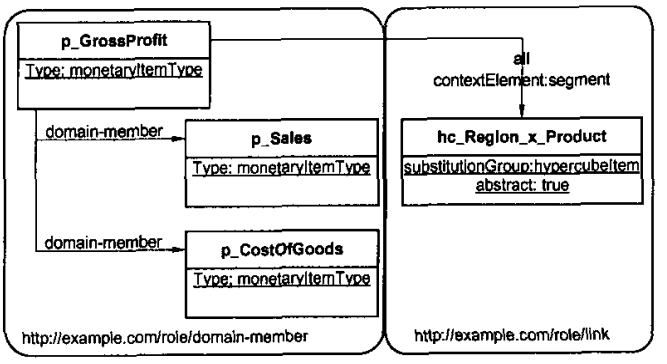
5.6.1 概述

基础项声明可以是 domain-member 关系的源。当基础项声明同时为 domain-member 关系和 has-hypercube 关系的源时,domain-member 关系的目标将继承源元素的 has-hypercube 关系。这种继承是可迁的。继承保留了基础集以及原始 has-hypercube 关系的 DRS,同时还保留了其 xbrldt:contextElement 属性和 xbrldt:closed 属性的值。

下例介绍了两个基础项声明拥有相同祖先,并均继承了超立方体的 all 关系。

示例:

继承超立方体的两个基础项声明



基础项声明 p_Sales 和 p_CostOfGoods 并不是任何 has-hypercube 关系的源;它们从基础项声明 p_GrossProfit 那里继承了超立方体 hc_Region_x_Product。

它们均为同一个维度基础集的根

没有直接的 has-hypercube 关系的基础项声明,可以从其 domain-member 网络的父级处继承任意

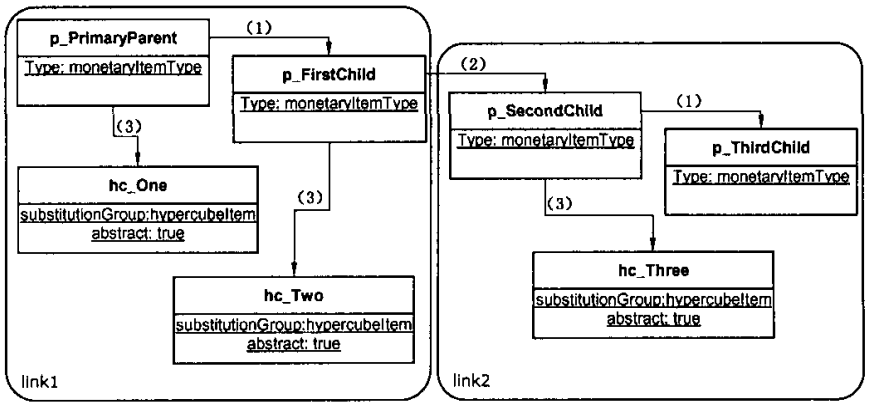
多的 has-hypercube 关系。has-hypercube 关系对实例校验的影响不变，是因为其关系已经得到继承。

5.6.2 多个 has-hypercube 弧的处理

使用基础分类标准的元素而产生的 domain-member 网络中可具有处在不同树层次中的多个 has-hypercube 关系。根据 6.2.2，只有在同一个基础集内的超立方体才能被考虑同时进行校验。

示例 1：

多个立方体的继承和处理



(1) domain-member

(2) domain-member(targetRole=link2)(注：链接 2 的超级立方体从链接 1 中继承而来)

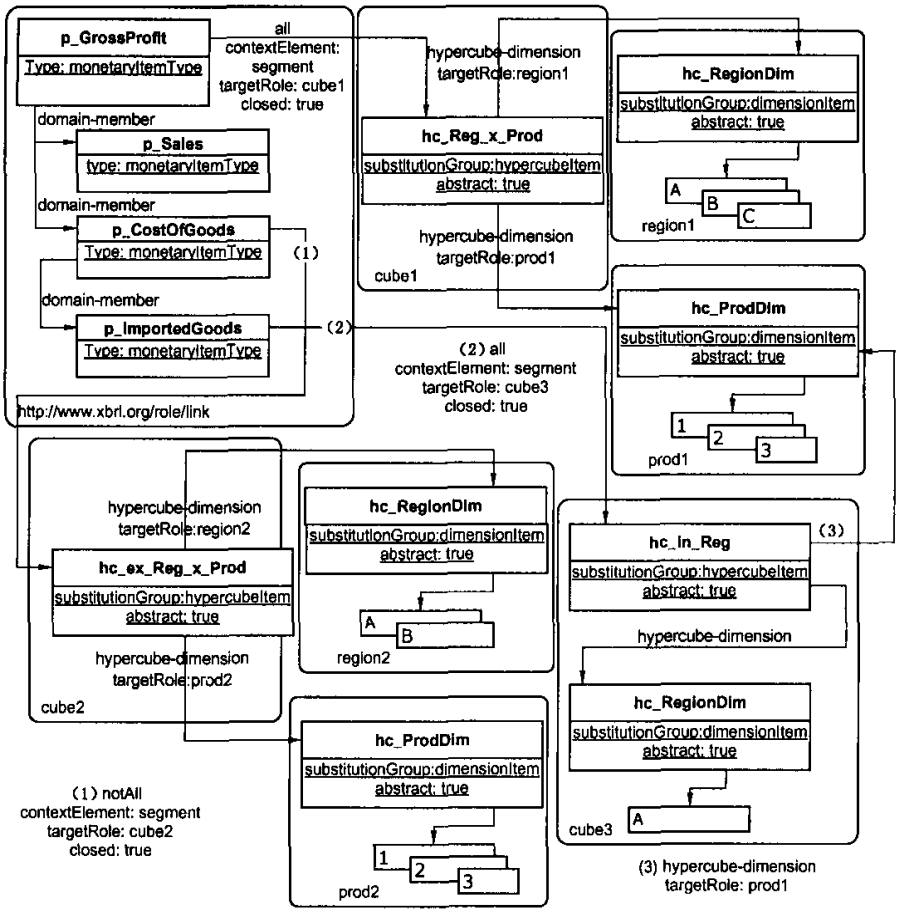
(3) all (has-hypercube 弧)

校验中的元素	扩展链接	超立方体
p_PrimaryParent	link1	Hc_One
p_FirstChild	link1	Hc_One 和 hc_Two
p_SecondChild	link1	Hc_One 和 hc_Two
p_ThirdChild	link1	Hc_One 和 hc_Two
p_PrimaryParent	link2	无
p_FirstChild	link2	无
p_SecondChild	link2	Hc_Three
p_ThirdChild	link2	Hc_Three

超立方体的继承遍布 domain-member 网络的整个维度关系集。

示例 2：

domain-member 网络中多个 all 属性超立方体



以下是各基础项的可能的值：

元素	产品维度	区域维度	解　　释
p_GrossProfit	1,2,3	A,B,C	在 cube1 中定义的直接连接的超立方体 hc_Reg_x_Prod
p_Sales	1,2,3	A,B,C	从父级 p_GrossProfit 处继承
p_CostOfGoods	1,2,3	C	从父级继承而来的是 cube1，但具有 notAll 属性的 cube2 将包含所有成员的 A，B 组合从产品维度中排除出去，并将元素 C 在与产品维度的所有成员相连接的区域维度中保留
p_ImportedGoods	1,2,3	无组合是一种可能出现的情况，如果发生这种情况，维度处理器会发出警告	只有 C 区域和产品域的所有成员的组合是从父级 p_CostOfGoods 处继承而来。cube3 不兼容，因为它仅包含区域 A。在 cube1 而非 cube2 中声明的超立方体或者在 cube3 中声明的超立方体将可能无效，因此超立方体是不兼容的，并且不可能为 p_ImportedGoods 创建维度有效的实例化

has-hypercube 弧的处理顺序与结果无关。

5.7 维度的缺省值

5.7.1 概述

维度允许有缺省值。

dimension-default 关系规定了哪个域成员担任维度的缺省成员。

维度的缺省值由维度处理器自动推理得出，不应在实例的上下文中出现。

对维度缺省值的自动推理可以被运用于分类标准中，使得 summation-item 或者特定事实间的其他关系，无需特别的维度即可报告。

示例：

关于 summation-item 关系的缺省值的自动推断

```
〈tx:GrossProfit contextRef = "ctx1" unitRef = "JPY"〉100000000〈/tx:GrossProfit〉
〈tx:GrossProfit contextRef = "ctx2" unitRef = "JPY"〉50000000〈/tx:GrossProfit〉
〈tx:Taxes contextRef = "ctx1" unitRef = "JPY" 〉35000000〈/tx:Taxes〉
〈tx:NetProfit contextRef = "ctx1" unitRef = "JPY"〉65000000〈/tx:NetProfit〉

〈context id = "ctx1"〉
 〈entity〉
  〈identifier scheme = "http://nic.net"〉例.com〈/identifier〉
 〈/entity〉
 〈period〉〈forever/〉〈/period〉
〈/context〉

〈context id = "ctx2"〉
 〈entity〉
   〈identifier scheme = "http://nic.net"〉例.com〈/identifier〉
   〈segment〉
    〈xbrldi:explicitMember dimension = "tp:ProductDim"〉p:Cars〈/xbrldi:explicitMember〉
   〈/segment〉
 〈/entity〉
 〈period〉〈forever/〉〈/period〉
〈/context〉
```

在 tx 分类标准中，只有 GrossProfit 这个基础概念有一个产品维度，产品维度中的缺省成员为 p:TotalProducts。在基础分类标准 tx 中存在计算网络去表达 NetProfit = GrossProfit - Taxes。TotalProducts (contextRef = "ctx1"的那一个)的概念 GrossProfit 是维度有效的(ProductDim = p:TotalProducts)，并和 Taxes 和 NetProfit 一起用于 summation-item 关系中。

5.7.2 弧角色 http://xbrl.org/int/dim/arcrole/dimension-default

dimension-default 弧角色从其源识别出维度元素，并从其目标识别出缺省成员。

在 dimension-default 弧的目标中标示缺省成员是全局的。如果一个 dimension-default 弧存在于一个扩展链接中，则其缺省成员被认为定义在维度被使用的所有扩展链接中，并且域中包含弧的目标所表示的域成员。

dimension-default 关系与 xbrldt:targetRole 属性无关。

dimension-default 弧的存在并不在任何源维度域内添加任何目标项。

http://xbrl.org/int/dim/arcrole/dimension-default 弧角色声明如下：

```
<arcroleType
 id = "dimension-default"
 cyclesAllowed = "none"
 arcroleURI = "http://xbrl.org/int/dim/arcrole/dimension-default">
 <definition>源(维度)声明,存在一个作为弧(成员)的目标的缺省成员</definition>
 <usedOn>definitionArc</usedOn>
</arcroleType>
```

5.7.3 对 dimension-default 弧的约束

对 xbrldt:targetRole 属性的约束为:

a) 如果弧的源不是一个明确维度声明,则维度处理器应报错 xbrldte:DimensionDefaultSourceError。

b) 如果弧的目标不是一个域成员声明,则维度处理器应报错 xbrldte:DimensionDefaultTargetError。

c) 一个维度不应含有超过一个缺省成员。如果违反该规则,则维度处理器应报错 xbrldte:TooManyDefaultMembersError。

5.7.4 有关维度缺省成员的实例文档的约束

维度的缺省成员不能出现在上下文中。当缺省成员出现时,维度处理器应报错 xbrldie:DefaultValueUsedInInstanceError。

6 实例文档里的维度

6.1 概述

基础分类标准定义了概念,以用于表示 XBRL 实例文档中的事实。

一个 DTS 包含依据本部分定义的维度关系的实例文档,应根据本部分定义的规则进行校验。

实例文档的校验应逐项进行。维度处理器应找到和文档 DTS 内的项目有关的超立方体,并且逐一校验。如果其所有维度都是有效的,则超立方体就是有效的。如在上下文中存在域的成员或者该维度包含缺省成员,则维度就是有效的。超立方体校验的结果应为那些使用特定运算符"all"、"not all"而在同一基础集内共存的超立方体而相连接。如果一个基础项不与任何超立方体关联,或者至少存在这样一个扩展链接角色,通过该角色基础项被关联到联合校验结果为有效的超立方体的组合上去,则该基础项是有效的。

6.2 基础项的校验

6.2.1 概述

每个拥有在实例文档的 DTS 中的超立方体的基础项,应根据至少一个定义超立方体的基础集而被判定为有效。

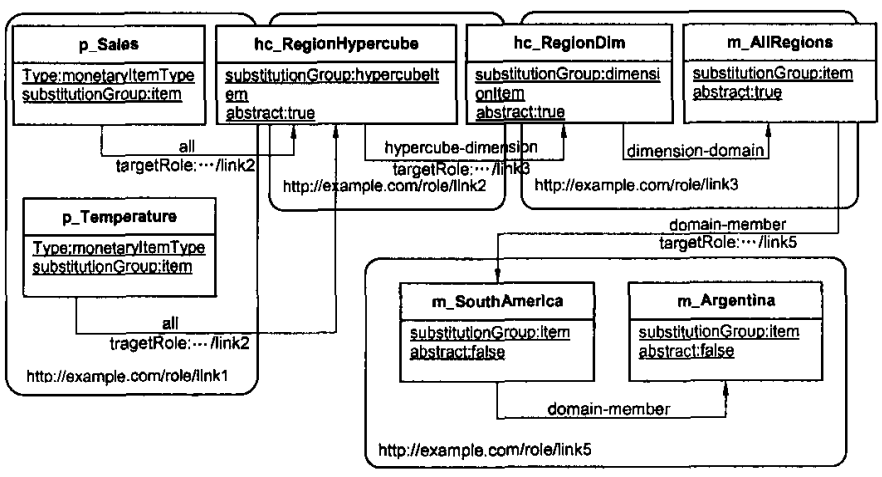
当基础集包含至少一个源为该基础项声明的 has-hypercube 关系时,基础项声明就是基础集内维度关系集的根。

当基础项声明是一个基础集的根的时候,其同时也是该基础集的维度关系集的根。

维度有效的基础项实例化应属于以下两种情况之一,它不是任何维度关系集的根,或者是上下文维度有效的维度关系集的根。

示例：

位于维度关系集的根的基础项



基础项声明 p_Sales 是维度关系集的根，包含基础集 http://example.com/role/link3 中的所有弧。

基础项声明是否在元组中出现，与基础项实例化和上下文是否维度有效无关。

6.2.2 关于基础项有效性的约束

如果超立方体至少在一个基础集被发现是相互有效的，则基础项就是维度有效的。如果在所有基础集内的超立方体组合是无效的，则维度处理器将报错 xbrldie：PrimaryItemDimensionallyInvalidError。

6.2.3 基础集内超立方体的互相有效性

如果将所有超立方体联结起来的操作能够被满足，则依据基础集内定义的超立方体来判断，基础项是有效的。

表 4 表示当根据超立方体以及在同一基础集内连接多个超立方体的操作做出判断，基础项为无效时，维度处理器将提出警告。

表 4 多个超立方体的连接和其运算结果

立方体数目	运算符	超立方体评价	基础项评价结果	警　告
1	all	有效	有效	无
1	notAll	无效	有效	无
2	all notAll	有效 无效	有效	无
1	all	无效	无效	仅有一个超立方体，其 all 运算无效
1	notAll	有效	无效	仅有一个超立方体，其 notAll 运算有效
2	all notAll	有效 有效	无效	基础集内已定义多个超立方体，其连接结果为无效

6.2.4 超立方体的个体有效性

超立方体有效性汇总表如表5所示。

表5 超立方体有效性汇总表

是否封闭	是否为空	维度	维度值	维度有效性	结果
是	是	不适用	无	不适用	有效
否	是	不适用	不适用	不适用	有效
是	是	不适用	找到维度值	不适用	无效
是	否	Dim1 和 Dim2	Dim1 和 Dim2 的维度值被找到	Dim1 有效，Dim2 有效	有效
是	否	Dim1 和 Dim2	Dim1、Dim2 和 Dim3 的维度值被找到(见注1)	不适用	无效
否	否	Dim1 和 Dim2	Dim1、Dim2 和 Dim3 的维度值被找到(见注2)	Dim1 有效，Dim2 有效，Dim3 不适用	有效
是/否	否	Dim1	Dim1 的维度值未被找到	不适用	无效
是/否	否	Dim1	Dim1 的维度值被找到	Dim1 无效	无效
注1：无效，因为超立方体已封闭并且 Dim3 并不需要。 注2：超立方体是开放的，所以 Dim3 是被允许的。					

超立方体中的所有维度应依据6.2.5中定义的规则来校验。

超立方体的个体有效性的约束为：

a) 一个封闭的超立方体不应包含不期望出现的维度值。如果封闭的超立方体中包含了任何不被期望出现的值，则它为无效。

b) 由 has-hypercube 弧的 xbrldt:contextElement 属性所标示的容器中应存在一个维度值。

c) 超立方体中定义的所有维度依据6.2.5规则而言是有效的。如果一个维度无效，则整个超立方体无效。

6.2.5 维度的有效性

维度值(dimension value)是指明确维度的明确维度容器的内容值的 QName，或是指作为类型化维度的类型化维度容器的第一个子元素的 XML 片段。维度值的存在是为一个或两个可能的上下文容器——segment 或 scenario 中的其中某个特定容器服务的。其默认值为所有可能的维度值，但它们并不包含在维度容器中。

维度容器(dimension container)是对类型化维度而言就是 xbrldi:typedMember 元素，对明确维度而言就是 xbrldi:explicitMember 元素。

维度值对其域应是有效的。

6.2.6 获取维度的维度值

维度处理器应能从 segment 或者 scenario 元素中获取类型化维度或明确维度的值。

类型化维度的维度值内容即为 xbrldi:typedMember 元素的内容，其 dimension 属性值为在用的维度的 QName 值。

明确维度的维度值内容即为 xbrldi:explicitMember 元素的内容，其维度属性为在用的维度的 QName 值，或者在该值未在实例中通报并且维度拥有缺省成员的情况下，即为缺省成员的 QName 值。

缺省值(default value)为缺省成员的 QName。

6.2.7 对维度值的约束

6.2.7.1 概述

对维度值的约束为：

a) 上下文中针对每个维度不应包含超过一个维度值 。如果此规则被违反，则校验器应报告错误 xbrldie:RepeatedDimensionInInstanceError。

b) 缺省值不能出现在实例文档中。如果出现缺省值，则应报告错误 xbrldie:DefaultValueUsed InInstanceError。

6.2.7.2 上下文及其维度有效性的示例

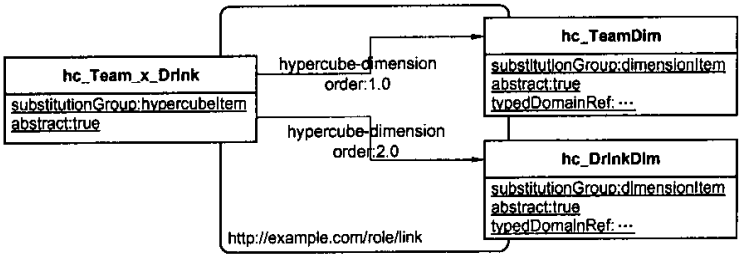
示例 1：

包含对相同维度的两个引用的维度无效的上下文

```
<context id="c3">
  <entity>
   <identifier scheme="http://nic.net">例.com</identifier>
   <segment>
    <xbrldi:typedMember dimension="tax:Team">
     <d:Team>Rams</d:Team>
    </xbrldi:typedMember>
   </segment>
  </entity>
  <period><forever/></period>
  <scenario>
   <xbrldi:typedMember dimension="tax:Team">
    <d:Team>Lakers</d:Team>
   </xbrldi:typedMember>
  </scenario>
</context>
```

示例 2：

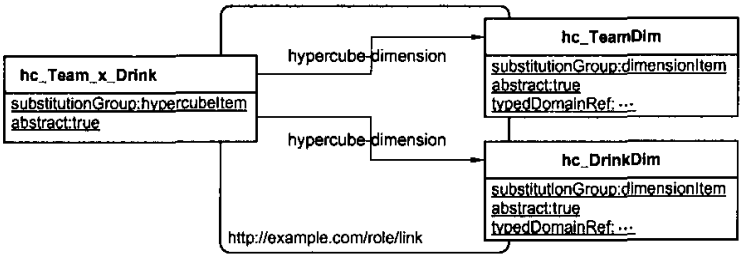
关于超立方体为有效的段



```
〈context id = "c4"〉
 〈entity〉
  〈identifier scheme = "http://nic.net"〉例.com〈/identifier〉
  〈segment〉
   〈xbrldi:typedMember dimension = "tax:TeamDim"〉
    〈d:Team〉Rams〈/d:Team〉
   〈/xbrldi:typedMember〉
   〈xbrldi:typedMember dimension = "tax:DrinkDim"〉
    〈d:Drink〉Coors〈/d:Drink〉
   〈/xbrldi:typedMember〉
  〈/segment〉
 〈/entity〉
 〈period〉〈forever/〉〈/period〉
〈/context〉
```

示例 3：

关于超立方体为维度无效的两个段



〈context id="c5"〉 〈entity〉 〈identifier scheme="http://nic.net"〉example.com〈/identifier〉 〈segment〉 〈xbrldi:typedMember dimension="tax:TeamDim"〉 〈d:Team〉Rams〈/d:Team〉 〈/xbrldi:typedMember〉 〈/segment〉 〈/entity〉 〈period〉〈forever/〉〈/period〉 〈/context〉	缺少对 Drink 维度的引用
〈context id="c7"〉 〈entity〉 〈identifier scheme="http://nic.net"〉example.com〈/identifier〉 〈segment〉 〈xbrldi:typedMember dimension="tax:DrinkDim"〉 〈d:Drink〉Coors〈/d:Drink〉 〈/xbrldi:typedMember〉 〈/segment〉 〈/entity〉 〈period〉〈forever/〉〈/period〉 〈/context〉	缺少对 Team 维度的引用

示例 4：

依据维度和域判断而得的有效和无效超立方体

维度	域	成员	结　果
RegionDim	r:Europe,r:Asia,r:Africa,r:SouthAmerica	r:Europe	有效（成员在域中）
RegionDim	r:Europe,r:Asia,r:Africa,r:SouthAmerica	r:Japan	无效(成员不在域中)
RegionDim 和 ProductDim	r:Europe,r:Asia,r:Africa,r:SouthAmerica p:Wine,p:Cars,p:Other	p:Wine	无效(Region 维度缺值) 应报告错误 xbrldie:primaryItemDimensionallyInvalidError
RegionDim 和 ProductDim	r:Europe,r:Asia,r:Africa,r:SouthAmerica p:Wine,p:Cars,p:Other	r:Japan	无效（Product 维度缺值） 应报告错误 xbrldie:primaryItemDimensionallyInvalidError
RegionDim 和 ProductDim	r:Europe,r:Asia,r:Africa,r:SouthAmerica p:Wine,p:Cars,p:Other	r:Africa p:Soja	无效（成员不在 Product 域中）
RegionDim 和 ProductDim	r:Europe,r:Asia,r:Africa,r:SouthAmerica p:Wine, p:Cars, p:Other	r:Africa p:Cars	有效（成员在 Region 和 Product 域中）

示例 5：

由于违反超立方体约束而导致基础项维度无效

```
〈unit   id = "eur"〉〈measure〉iso4217:CNY〈/measure〉〈/unit〉
〈context id = "c19"〉
 〈entity〉
  〈identifier scheme = "http://nic.net"〉例.com〈/identifier〉
  〈segment〉
   〈xbrldi:explicitMember dimension = "p:ProductDim"〉
     p:m_RedWine〈/xbrldi:explicitMember〉
    〈xbrldi:explicitMember dimension = "p:RegionDim"〉
     p:m_China〈/xbrldi:explicitMember〉
   〈/segment〉
  〈/entity〉
  〈period〉〈forever/〉〈/period〉
〈/context〉
〈p:p_GrossProfit
 contextRef = "c19" unitRef = "cny"
 decimals = "INF"〉10000〈/p:p_GrossProfit〉
〈p:p_CostOfGoods
  contextRef = "c19" unitRef = "cny"
  decimals = "INF"〉50000〈/p:p_CostOfGoods〉
〈p:p_Sales contextRef = "c19" unitRef = "cny"
 decimals = "INF"〉60000〈/p:p_Sales〉
```

假设上下文中使用了 5.6.1 和 6.2.7.4示例中定义的 DRS。p:m_China 不是 RegionDim 的域成员，尽管只有 p_Sales 有一个 hc_Product_x_Region 的明确“All”弧，但三个事实都违反了该约束

6.2.7.3　类型化维度

6.2.7.3.1　xbrldi:typedMember 元素

类型化维度的成员是元素的实例化，该等元素符合在类型化维度声明的 xbrldt:typedDomainRef 属性中被引用的元素。

xbrldi:typedMember 元素是 XML 元素，其内容应为另一个其模式声明位于 xbrldt:typedDomainRef 属性的元素。

```
<element name = "typedMember">
 <annotation>
  <documentation xml:lang = "en">该元素包含 anyType 的一个子项
  </documentation>
 </annotation>
 <complexType>
  <sequence>
   <any namespace = "##other"/>
  </sequence>
  <attribute name = "dimension" type = "QName" use = "required"/>
 </complexType>
</element>
```

xbrldi:typedMember 元素的 dimension 属性的内容应解析为在实例的 DTS 中的类型化维度声明的 QName。

如果根据在 xbrldt:typedDomainRef 属性中被引用 XML 模式文件声明，维度值是有效的，则类型化维度就是有效的。

示例 1：

两个在上下文的段中被引用的维度

```
<context id = "c1">
 <entity>
  <identifier scheme = "http://nic.net">example.com</identifier>
  <segment>
   <xbrldi:typedMember dimension = "tax:TeamDim">
    <d:Team>Lakers</d:dTeam>
   </xbrldi:typedMember>
   <xbrldi:typedMember dimension = "tax:DrinkDim">
    <d:Drink>Coors</d:Drink>
   </xbrldi:typedMember>
  </segment>
 </entity>
 <period><forever/></period>
</context>
```

示例 2：

两个在上下文的场景中被引用的维度

```
<context id = "c2">
 <entity>
  <identifier scheme = "http://nic.net">example.com</identifier>
 </entity>
 <period><forever/></period>
 <scenario>
  <xbrldi:typedMember dimension = "tax:TeamDim">
   <d:Team>Celtics</d:dTeam>
  </xbrldi:typedMember>
  <xbrldi:typedMember dimension = "tax:DrinkDim">
   <d:Drink>Sam Adams</d:Drink>
  </xbrldi:typedMember>
 </scenario>
</context>
```

并非所有出现在 segment 或者 scenario 中的元素都是维度元素；请参见本部分中关于“封闭”属性的部分。

6.2.7.3.2 关于 xbrldi:typedMember 元素的 dimension 属性的约束

xbrldi:typedMember 元素的 dimension 属性的内容应为在实例文档的 DTS 的模式文件中定义的类型化维度声明的 QName。如果 QName 出现在 xbrldi:typedMember 的 dimension 属性之中的元素被解析为类型化维度声明以外的其他内容，则维度处理器将报错 xbrldie:TypedMemberNotTypedDimensionError。

6.2.7.3.3 关于 xbrldi:typedMember 元素内容的约束

关于 xbrldi:typedMember 元素内容的约束为：

a) 依照 xbrldi XML 模式文件，xbrldi:typedMember 的每个实例应仅有一个子元素。

b) 类型化维度值依照在 xbrldt:typedDomainRef 属性被引用的元素的 XML 模式文件声明而言应是有效的。

c) 类型化维度内容应为在 xbrldi:typedMember 元素的 dimension 属性中指明的类型化维度的 xbrldt:typedDomainRef 所指向的元素的实例化。如果该规则被违反，维度处理器将报错 xbrldie:IllegalTypedDimensionContentError。

6.2.7.4 明确维度

6.2.7.4.1 xbrldi:explicitMember 元素

明确维度的成员为作为 xbrldi:explicitMember 元素的内容的 QName。它应为明确维度的一个有效成员。

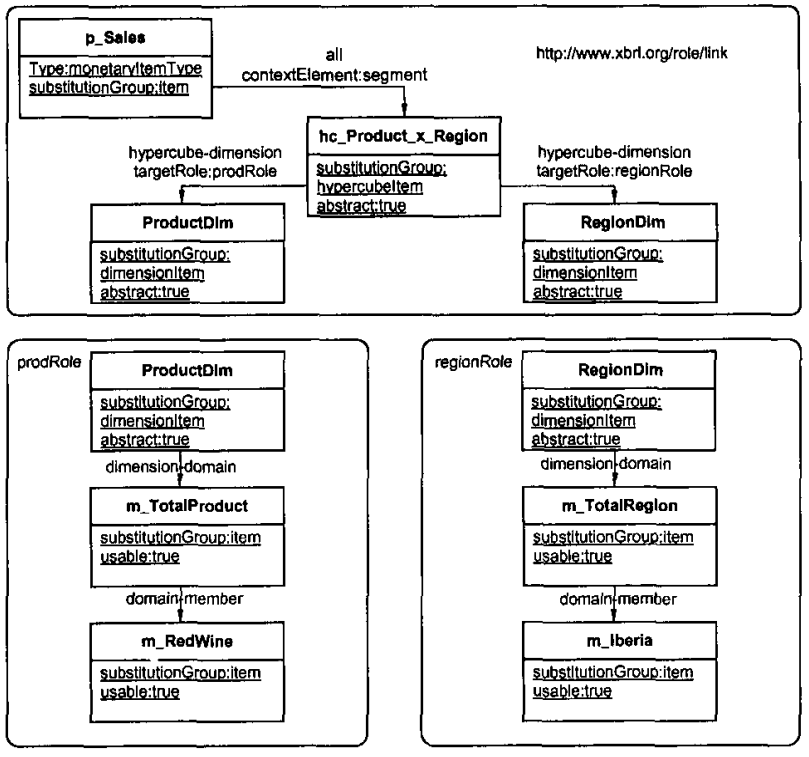
xbrldi:explicitMember 元素是一个内容应为 QName 的 XML 元素。

维度值(QName)是 dimension 属性引用的明确维度域的一个有效成员，则明确维度就是有效的。

```
<element name = "explicitMember">
 <annotation>
  <documentation xml:lang = "en">该元素包含某项的 QName，而该项是明确维度的成员
  </documentation>
 </annotation>
 <complexType>
  <simpleContent>
   <extension base = "QName">
    <attribute name = "dimension" type = "QName" use = "required"/>
   </extension>
  </simpleContent>
 </complexType>
</element>
```

示例：

关于超立方体和两个明确维度的维度有效的上下文




```
〈context id = "c16"〉
 〈entity〉
  〈identifier scheme = "http://nic.net〉example.com〈/identifier〉
   〈segment〉
    〈xbrldi:explicitMember dimension = "tax:RegionDim"〉
     r:m_Iberia〈/xbrldi:explicitMember〉
    〈xbrldi:explicitMember dimension = "tax:ProductDim"〉
     p:m_RedWine〈/xbrldi:explicitMember〉
   〈/segment〉
  〈/entity〉
  〈period〉〈forever〉〈/period〉
 〈/context〉
```

6.2.7.4.2 关于 xbrldi:explicitMember 元素的 dimension 属性的约束

xbrldi:explicitMember 元素的 dimension 属性的内容应为一个由实例文档的 DTS 中的模式文件中定义的明确维度元素的 QName。如果 QName 为出现在 xbrldi:explicitMember 的 dimension 属性中的元素，且被解析为明确维度声明以外的东西，则维度处理器将报错 xbrldie:ExplicitMemberNotExplicitDimensionError。

6.2.7.4.3 关于 xbrldi:explicitMember 元素的内容的约束

xbrldi:explicitMember 元素的内容应为一个 QName，该 QName 的全局元素定义可以在被 QName 的命名空间所引用的分类标准模式文件中找到。如果该规则被违反，则维度处理器将会报错 xbrldie:ExplicitMemberUndefinedQNameError。

6.3 维度等价事实的定义

维度等价(d-equal)指如果两个事实针对某个维度有相同的维度值,则这两个事实针对该维度而言就是维度等价的。

d-equal 相对于在 GB/T 25500.1 中定义的 c-equal、u-equal、s-equal 而言,是一个独立的运算。d-equal仅针对两个事实和一个维度而定义。如果称两个维度值满足 s-equal2,则这两个维度值是相同的。如果两个事实都拥有 dimension 属性的内容满足 s-equal2 条件的维度容器,并且两者均引用同一维度声明,则这两个事实拥有相同的维度。

这里所说的 s-equal2 运算和在 GB/T 25500.1 中已经定义的是同一个运算,并在 x-equal 的运算操作定义中用 XPath 2.0 替代了原来的 XPath 1.0,并且"XPath 1.0 兼容模式"属性在静态上下文中被设置为 false。

为了使 GB/T 25500.1 中定义的 summation-item 关系能够起作用,所有在上下文中的 segment 或者 scenario 容器中出现的维度元素应满足依照 GB/T 25500.1 所定义的 s-equal。

注:GB/T 25500.1 是基于 XPath 1.0 来制定的。依照 XPath 1.0 中所规定,属性的内容总是一个 string-value 而不是 QName。XBRL 应用程序接口实施维度应该注意这一点,并将出现在维度容器的 dimension 属性中和出现在实例文档的维度容器的内容中的 QNames 的前缀规范化。

出现在不同实例中的、和不同上下文相关的两个事实,对所有维度而言可以均满足 d-equal,而不管元素在 segment 或者 scenario 中的出现顺序。

示例:

多个上下文和 d-equal 运算的结果

上下文 A 的事实 Sales	上下文 B 的事实 Sales	d-equal(Sales, prodDim)
〈segment〉 〈xbrldi:explicitMember dimension = "tax:prodDim"〉 p:product〈/xbrldi:explicitMember〉 〈/segment〉	〈segment〉 〈xbrldi:explicitMember dimension = "tax:prodDim"〉p:product〈/xbrldi:explicitMember〉 〈/segment〉	满足。 请注意元素之间的空间是不同的
〈segment〉 〈xbrli:explicitMember dimension = "tax:prodDim"〉p:product〈/xbrldi:ExplicitMember〉 〈xbrli:explicitMember dimension = "tax:dept_dim"〉r:sales 〈/xbrldi:ExplicitMember〉 〈/segment〉	〈segment〉 〈xbrldi:explicitMember dimension = "tax:dept_dim"〉r:sales 〈/xbrldi:ExplicitMember〉 〈xbrldi:explicitMember dimension = "tax:prodDim"〉p:product 〈/xbrldi:ExplicitMember〉 〈/segment〉	满足。 请注意两个明确元素的顺序是不同的。 依照本节中的内容,这两个上下文应不在同一实例中出现
〈segment〉 〈xbrldi:explicitMember dimension = "tax:prodDim"〉p:product 〈/xbrldi:explicitMember〉 〈/segment〉	〈scenario〉 〈xbrldi:explicitMember dimension = "tax:prodDim"〉p:product 〈/xbrldi:explicitMember〉 〈/scenario〉	不满足 维度应通过相同的容器类型来报告

附 录 A
（规范性附录）
模 式 文 件

A.1 xbrldt-2005.xsd

```
<? xml version = "1.0" encoding = "utf-8"?>
<xs:schema
xmlns:xs = "http://www.w3.org/2001/XMLSchema" xmlns:xbrli = "http://www.xbrl.org/2003/instance"
xmlns = "http://www.xbrl.org/2003/linkbase"
xmlns:xlink = "http://www.w3.org/1999/xlink"
xmlns:xl = "http://www.xbrl.org/2003/XLink"
xmlns:xbrldt = "http://xbrl.org/2005/xbrldt" targetNamespace = "http://xbrl.org/2005/xbrldt"
elementFormDefault = "qualified"
attributeFormDefault = "unqualified">
 <xs:annotation>
  <xs:appinfo>
   <arcroleType
    id = "hypercube-dimension"
    cyclesAllowed = "none"
    arcroleURI = "http://xbrl.org/int/dim/arcrole/hypercube-dimension">
    <definition>源(超立方体)包含目标(维度)</definition>
    <usedOn>definitionArc</usedOn>
   </arcroleType>
   <arcroleType
    id = "dimension-domain"
    cyclesAllowed = "none"
    arcroleURI = "http://xbrl.org/int/dim/arcrole/dimension-domain">
    <definition>源(维度)只有目标(域)作为其域</definition>
    <usedOn>definitionArc</usedOn>
   </arcroleType>
   <arcroleType
    id = "domain-member"
    cyclesAllowed = "undirected"
    arcroleURI = "http://xbrl.org/int/dim/arcrole/domain-member">
    <definition>源(域)包含目标(成员)</definition>
    <usedOn>definitionArc</usedOn>
   </arcroleType>
   <arcroleType
    id = "all"
    cyclesAllowed = "undirected"
    arcroleURI = "http://xbrl.org/int/dim/arcrole/all">
    <definition>源(基础项声明)要求目标(超立方体)的维度成员的组合出现在基础项的上下文里</definition>
    <usedOn>definitionArc</usedOn>
   </arcroleType>
   <arcroleType
    id = "notAll"
    cyclesAllowed = "undirected"
    arcroleURI = "http://xbrl.org/int/dim/arcrole/notAll">
    <definition>源(基础项声明)要求目标(超立方体)的维度成员的组合不出现在基础项的上下文里</definition>
    <usedOn>definitionArc</usedOn>
```

```
    </arcroleType>
    <arcroleType
      id = "dimension-default"
      cyclesAllowed = "none"
      arcroleURI = "http://xbrl.org/int/dim/arcrole/dimension-default">
      <definition>源(维度)声明,存在一个作为弧(成员)的目标的缺省成员</definition>
      <usedOn>definitionArc</usedOn>
    </arcroleType>
  </xs:appinfo>
 </xs:annotation>
 <xs:import namespace = "http://www.xbrl.org/2003/instance" schemaLocation = "http://www.xbrl.org/2003/xbrl-
instance-2003-12-31.xsd"/>
 <xs:simpleType name = "contextElementType">
  <xs:restriction base = "xs:token">
   <xs:enumeration value = "segment"/>
   <xs:enumeration value = "scenario"/>
  </xs:restriction>
 </xs:simpleType>
 <xs:attribute name = "contextElement" type = "xbrldt:contextElementType"/>
 <xs:attribute name = "typedDomainRef" type = "xs:anyURI"/>
 <xs:attribute name = "closed" type = "xs:boolean" default = "false"/>
 <xs:attribute name = "usable" type = "xs:boolean" default = "true"/>
 <xs:attribute name = "targetRole" type = "xs:anyURI"/>
 <xs:element
  name = "hypercubeItem"
  id = "xbrldt_hypercubeItem"
  abstract = "true"
  substitutionGroup = "xbrli:item"
  type = "xbrli:stringItemType"
  xbrli:periodType = "duration"/>
 <xs:element
  name = "dimensionItem"
  id = "xbrldt_dimensionItem"
  abstract = "true"
  substitutionGroup = "xbrli:item"
  type = "xbrli:stringItemType"
  xbrli:periodType = "duration"/>
</xs:schema>
```

A.2 xbrldi-2006.xsd

```
<? xml version = "1.0" encoding = "utf-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
targetNamespace = "http://xbrl.org/2006/xbrldi"
elementFormDefault = "qualified"
attributeFormDefault = "unqualified">
 <annotation>
  <appinfo>
   <documentation xml:lang = "en">This schema is used by XBRL instances that use dimensions to define legal segment
and scenario element contents.</documentation>
  </appinfo>
 </annotation>
 <element name = "explicitMember">
  <annotation>
   <documentation xml:lang = "en">该元素包含某项的 QName,而该项是明确维度的成员
```

```
    </documentation>
   </annotation>
   <complexType>
    <simpleContent>
     <extension base = "QName">
      <attribute name = "dimension" type = "QName" use = "required"/>
     </extension>
    </simpleContent>
   </complexType>
  </element>
  <element name = "typedMember">
   <annotation>
    <documentation xml:lang = "en">该元素包含 anyType 的一个子项
    </documentation>
   </annotation>
   <complexType>
    <sequence>
     <any namespace = "##other"/>
    </sequence>
    <attribute name = "dimension" type = "QName" use = "required"/>
   </complexType>
  </element>
</schema>
```