



中华人民共和国国家标准

GB/T 25500.4—2010

可扩展商业报告语言(XBRL)技术规范 第4部分:版本

Extensible Business Reporting Language(XBRL) specification—
Part 4: Versioning

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 命名空间及其前缀 1

5 版本规范概述 2

 5.1 总体概述 2

 5.2 模块化 2

 5.3 背景 2

 5.4 URI 解析 3

 5.5 语法规义规则 3

6 版本报告的结构 3

 6.1 组成 3

 6.2 版本报告 4

 6.3 DTS 标识符 4

 6.4 任务 5

 6.5 任务类别 5

 6.6 行为 6

 6.7 事件 7

 6.8 标识符 7

 6.9 标签和引用 7

 6.10 相关版本报告 8

7 基础事件 8

 7.1 命名空间名称事件 8

 7.2 角色 URI 事件 9

8 标识符 9

 8.1 命名空间名称标识符 9

 8.2 角色 URI 标识符 9

 8.3 URI 引用 9

9 映射 10

 9.1 命名空间映射 10

 9.2 角色 URI 映射 10

10 概念基础 10

 10.1 概述 10

10.2 事件	10
10.3 概念标识符	11
10.4 概念映射	11
附录 A (规范性附录) 模式文件	13
参考文献	17
表 1 命名空间和命名空间前缀	1
表 2 <ver:report> 元素的 XML 表示	4
表 3 <ver:dts.identifier>元素的 XML 表示	4
表 4 <ver:assignment>元素的 XML 表示	5
表 5 标准的任务类别	5
表 6 <ver:category>元素的 XML 表示	6
表 7 <ver:action> 元素的 XML 表示	6
表 8 组件及对应元素的标签和引用定义	7
表 9 <ver:reportRef> 元素的 XML 表示	8
表 10 命名空间名称事件	8
表 11 角色 URI 事件	9
表 12 <ver:fromURI>和<ver:toURI>在 XML 中的表示	9
表 13 基础概念事件	11
表 14 XML 表现形式总结:vercb:fromE 和 vercb:toE	11

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4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 版本规范 1.0 版(基础和概念基础)(XBRL Versioning Specification 1.0, Base and Concept base)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。

本部分起草人:应唯、王颖、李红霞、黄敏、李敏敏、丁亮、覃东、杨海峰、朱健鹏、杨诚、臧梓伊、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,定义了 XBRL 分类标准版本报告的 XML 语法,提供了两个可发现分类标准集之间差异的结构化描述。

可扩展商业报告语言(XBRL)技术规范
第 4 部分:版本

1 范围

GB/T 25500 的本部分规定了 XBRL 分类标准版本报告的 XML 语法,提供了两个 DTS 之间差异的结构化描述。

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

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

GB/T 25500.2 可扩展商业报告语言(XBRL)技术规范 第 2 部分:维度

GB/T 25500.3 可扩展商业报告语言(XBRL)技术规范 第 3 部分:公式

3 术语和定义

GB/T 18793—2002、GB/T 25500.1、GB/T 25500.2、GB/T 25500.3 中界定的以及下列术语和定义适用于本文件。

4 命名空间及其前缀

元素或者属性的命名空间前缀使用以 ns:name 为例,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

前缀	命名空间 URI
ver	http://xbrl.org/2010/versioning-base
vere	http://xbrl.org/2010/versioning-base/error
xs	http://www.w3.org/2001/XMLSchema
gen	http://xbrl.org/2008/generic
xlink	http://www.w3.org/1999/xlink
vercb	http://xbrl.org/2010/versioning-concept-base
vercbe	http://xbrl.org/2010/versioning-concept-base/error

5 版本规范概述

5.1 总体概述

XBRL 基础规范以 XBRL 分类标准的形式使提取特定报告场景的结构化元数据成为可能。分类标准很少是固定对象,而是需要根据各种原因定期更新,这些原因包括:赖以构建分类标准的业务需求的变化(例如,会计准则的变化),分类标准技术设计的变化,或者是简单的错误更正。新版本分类标准的发布将对该分类标准的所有使用者产生影响,包括:

- a) 实例文档创建者,实例生成软件开发者;
- b) 该分类标准的扩展分类标准创建者,扩展分类标准生成软件开发者;
- c) 希望能够对依据不同版本分类标准编制的实例文档进行比较的用户。

本部分定义了 XBRL 版本报告的 XML 语法,其规范化模式文件见 A.1。版本报告用于允许分类标准创建者以一种结构化的格式传递关于两个分类标准版本之间的变动信息,以便将版本更新给分类标准使用者带来的影响降至最低。

版本报告不同于简单的差异报告,版本报告包括通过 DTS 自动比对所不能获取的信息(例如,它可以识别在两个 DTS 中等效但不同名的概念),并且传递关于版本变更背后动因的信息。正是这种传递额外信息的能力使得创建标准化的版本报告具有价值。

针对任何一对 DTS,都可以编制版本报告。

5.2 模块化

XBRL 版本规范是一个模块化的规范(本部分是其中一个部分),它允许版本报告的作者选择不同的粒度来记录版本的变化。这种模块化的方法允许把 XBRL 基础规范扩展(如 XBRL 维度规范)的版本信息作为扩展模块处理。

5.3 背景

XBRL 版本是对“DTS 变动相关信息”报告内容的标准化,以节省 DTS 使用者调整应用程序适应新版本 DTS 的时间。

本部分区分“技术差异”和“语义差异”。“技术差异”是通过比较待观察的一对信息项的属性,软件可以自动发现的差异。“语义差异”是版本报告作者对技术差异的解释。在给定两个 DTS 和 DTS 间的信息项匹配规则下,技术差异集是唯一的,而语义差异集是无限的,因为它们只是技术差异的解释。“语义差异”包含了“技术差异”以及人类可读文档,这对于使用者将应用程序从以前的 DTS 版本向下一个版本迁移,提供了有用的信息。图 1 展示了版本报告信息的不同层级。

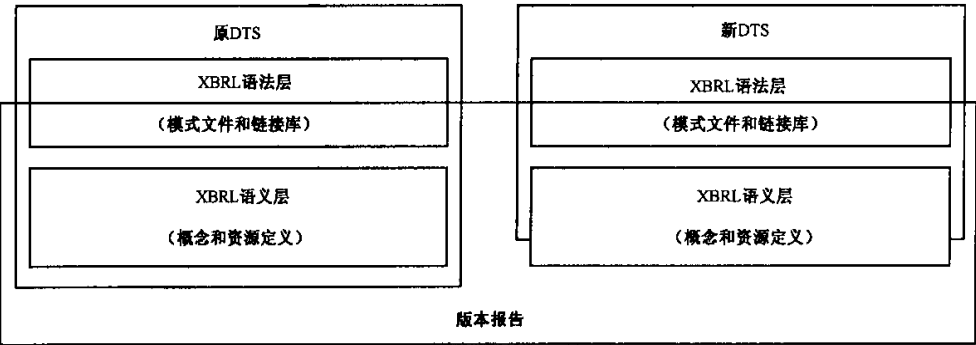


图 1 版本报告的信息源

在不影响使用 DTS 的应用程序的情况下, DTS 中的信息能够写入一个或多个文件中, 语义上等效但语法上不同的 DTS 见示例。

示例:

- a) 一个命名空间为 `http://foo` 的分类标准模式文件“A. XSD”, 定义了两个概念 `tx:One` 和 `tx:Two`, 它引用了一个列报链接库, 该链接库使用了 `http://www.xbrl.org/2003/role` 中的 `parent-child` 弧来连接这两个概念, 以表示 `tx:One` 是 `tx:Two` 的上一级概念。
- b) 命名空间同样为 `http://foo` 的另一个分类标准模式文件“B. XSD”, 定义了相同的概念(`tx:One` 和 `tx:Two`), 它有两个同样使用 `http://www.xbrl.org/2003/role` 角色的嵌入式列报链接库, 其中一个将 `tx:Two` 定义为 `tx:One` 的上一级概念, 另一个则禁止了上述关系并设定了另一个关系将 `tx:One` 定义为 `tx:Two` 的上一级概念。

从 DTS 用户的角度说, 以上两个 DTS 是等效的, 尽管模式文件“A. XSD”和“B. XSD”在语法上完全不同, 但它们都定义了同样的概念和同样结果的概念间关系。两个 DTS 对于处理 XBRL 元数据的应用程序来说, 都定义了相同的信息集, 除非 XBRL 分类标准的编辑者对于列报有更为严格的规定, 从而禁止使用者使用关系。

5.4 URI 解析

本部分中凡是相对的 URI 都需根据 XML Base 规范转换为绝对的 URI。

5.5 语法语义规则

版本报告是一个 XML 文档, 它根据本部分中的规则, 提供关于两个 DTS, 即 From-DTS 和 To-DTS, 它们之间如何从前者转换到后者的信息。

From-DTS 是形成版本报告中的源 DTS, 通常是版本较低或者发布时间较早的那个 DTS。

To-DTS 是作为版本报告中的目标 DTS, 通常是版本较高或者发布时间较晚的那个 DTS。

From-DTS 和 To-DTS 之间的每个独立的变化被称为事件。若干个事件组成的逻辑变化被称为行为。而依次再由若干个行为组成的业务级的变化被称为任务。

事件和行为的组合, 使得删除事件和新增事件可以被关联起来, 从而用于表达替代关系。

本部分允许为任务和行为与人可读文档之间关联起来, 同时也提供了任务类别的可扩展机制。任务、行为和事件的应用见示例。

示例:

From-DTS 中的一个概念可能被 To-DTS 中的一对概念所替代, 这可通过三个事件来描述:

- 从 From-DTS 中删除一个概念;
- 在 To-DTS 中增加两个概念。

为了使这个过程能更好地被理解为替代, 而不仅仅是一个概念的删除和两个概念的新增, 这三个事件被组合成一个行为。

这种一个概念被另外两个概念替代的行为可能是更大范围的、由涉及更多其他概念的变化所构成的、业务级的变化中的一小部分。该业务级变化可以通过很多行为来表达, 而这些行为还将进一步组合成一个任务。

6 版本报告的结构

6.1 组成

版本报告由以下几部分组成:

- a) From-DTS 标识符;
- b) To-DTS 标识符;

- c) 任务；
- d) 行为；
- e) 事件。

6.2 版本报告

6.2.1 XML 表述

版本报告用<ver:report> 元素来表示,如表 2 所示。

表 2 <ver:report> 元素的 XML 表示

<pre><ver:report id = xs:ID > Content; link;linkbaseRef *, ver:reportRef *, ver:fromDTS, ver:toDTS, ver:assignments *, ver:action * </ver:report></pre>	
属 性	含 义
{related reports}	通过<ver:report>元素的<ver:reportRef>子元素来标识的版本报告的集合
{from-DTS}	由<ver:report>元素的<ver:fromDTS>子元素来表示的 From-DTS 标识符
{to-DTS}	由<ver:report>元素的<ver:toDTS>子元素来表示的 To-DTS 标识符
{assignments}	由<ver:report>元素的<ver: assignment>子元素来表示的任务的集合
{actions}	由<ver:report>元素的<ver: action>子元素来表示的行为的集合
{events}	由<ver:report>元素的<ver: event>子元素来表示的事件的集合

6.2.2 有效性

仅当以下条件同时满足时,版本报告才是有效的:

- a) 根据规范的模式文件所定义的规则,见附录 A,它是有效的;
- b) 它符合本部分中定义的所有校验规则。

6.3 DTS 标识符

6.3.1 XML 表述

DTS 标识符是一个或多个 URI 的集合,每一个 URI 都是针对所识别的 DTS 发现的起始点。

DTS 标识符由<ver:dts.identifier>替换组中的元素来表示,如表 3 所示。

表 3 <ver:dts.identifier>元素的 XML 表示

<pre><ver:dts.identifier> Content;(link;linkbaseRef link:schemaRef) + </ver:dts.identifier></pre>	
属 性	含 义
{identified DTS}	通过从分别由<link;linkbaseRef>元素和<link:schemaRef>元素所表示的链接库和模式文件出发的 DTS 发现最终所形成的 DTS

6.3.2 校验规则

DTS 标识符应标识一个符合 XBRL 基础规范的有效 XBRL DTS。

⟨link;linkbaseRef⟩元素和⟨link;schemaRef⟩元素已在 GB/T 25500.1 中定义,它们的实例应符合 GB/T 25500.1 的规则和模式文件定义。

6.3.3 From-DTS 标识符

From-DTS 标识符是标识 From-DTS 的 DTS 标识符。

From-DTS 标识符通过⟨ver;fromDTS⟩元素来表示。

6.3.4 To-DTS 标识符

To-DTS 标识符是标识 To-DTS 的 DTS 标识符。

To-DTS 标识符通过⟨ver;toDTS⟩元素来表示。

6.4 任务

任务是一组相关的行为的组合。

包含在任务中的行为的组合使得通用文档或分类可与一组相关行为的集合相关联。

任务由⟨ver;assignment⟩元素来表示。如表 4 所示。

表 4 ⟨ver;assignment⟩元素的 XML 表示

<pre><ver;assignment id = xs:ID > Content:ver:category * </ver;assignment></pre> <p>ver:category 替换组中的任何元素都可以是 ver:assignment 的子元素。</p>	
属 性	含 义
{categories}	任务类别集是通过⟨ver:category⟩替换组中作为⟨ver:assignment⟩元素的子元素的元素来表示的
{actions}	含有包含了任务的⟨assignment⟩属性的版本控制报告中的行为的集合

6.5 任务类别

6.5.1 类别

本部分提供了任务归类的可扩展机制。该机制可被用来根据变化的诱因来将任务归类。

任务类别是可与任务相关联的类别。

本部分定义了 3 个标准的任务类别,如表 5 所示。

表 5 标准的任务类别

类别名	元 素	描 述
Business	⟨ver;businessCategory⟩	表明任务使分类标准更趋同于当前的业务要求
Technical	⟨ver;technicalCategory⟩	表明任务由技术要求驱动
Errata	⟨ver;errataCategory⟩	表明任务会修正分类标准中的错误

6.5.2 自定义任务类别

除了预定义的标准任务类别,版本报告还可使用自定义任务类别。

自定义任务类别由 XML 模式文件的元素声明来定义,定义是直接或间接地通过作为<ver:category>替换组的替换组附属的链来实现的。如果自定义类别包含属于另一种类别元素的替换组,则它表示该类别的子层级。如下给出了自定义任务类别的示例。

示例:

用<ver:category>的替换组来声明的元素会被认为是自定义任务类别,它和其他三种预定义的标准类别可在一起使用。用<ver:businessCategory>的替换组来声明的元素将标识特殊类型的业务变化。

6.5.3 XML 表述

任务类别由<ver:category>替换组中的一个元素来表示。如表 6 所示。

表 6 <ver:category>元素的 XML 表示

<pre><ver:category id = xs:ID > Content:none </ver:category></pre>
--

6.6 行为

6.6.1 概述

行为由一组相关事件组成,用来表达单独的、互不相关的分类标准的改变。

一组相关事件通过将 From-DTS 中被删除的组件和 To-DTS 中增加的组件组合起来表示替代关系。在适用的条件下,语义的理解建立在类似组件在单独事件类的定义。

6.6.2 XML 表示形式

一个<ver:action> 代表一个事件。如表 7 所示。

表 7 <ver:action>元素的 XML 表示

<pre><ver:action id = xs:ID > Content:ver:actionRef *, ver:event * </ver:action></pre>	
<pre><ver:actionRef ref = xs:IDREF > Content:none </ver:actionRef></pre>	
特 性	表 达 含 义
{任务}	由<ver:action>元素的若干个<ver:actionRef>子元素的 ref 属性标识的任务的集合
{事件}	由<ver:report>元素的若干个<ver:event>替换组元素子元素标识的事件的集合

6.6.3 校验规则

元素<ver;actionRef>中的 ref 属性应标识一个<ver;assignment> 元素。

6.7 事件

事件是 From-DTS 与 To-DTS 之间的个别差异。

本部分及其扩展模块定义了事件,其代表差异的特殊类型。一个事件通常包括一个或多个 From-DTS 或 To-DTS 中组件的标识符。

一个事件由<ver:event> 替换组元素表示。<ver:event> 元素拥有开放式内容模型;特殊事件由定义特殊内容模型的非抽象元素表示。

6.8 标识符

Identifier 标识符是 From-DTS 或 To-DTS 里的要件标识符;

From 标识符是 From-DTS 中要件标识符;

From 标识符是 To-DTS 中要件标识符;

单独标识符类应将指定自有的 XML 表示形式。

6.9 标签和引用

6.9.1 概述

本部分定义了组件的标签和引用语义,及与组件相对应并与通用标签和通用引用相关联的 XML 元素,见表 8 所示。也可为在版本报告中提供其他元素的标签和引用,但本部分中并未定义这些标签和引用。

表 8 组件及对应元素的标签和引用定义

组 件	元 素
Versioning Report	<ver;report>
Assignment	<ver;assignment>
Action	<ver;action>
Event	<ver:event>

6.9.2 标签和引用链接库的发现

一个版本报告可通过链接库引用对定义通用标签和通用引用的链接库进行引用。

6.9.3 链接库引用

6.9.3.1 链接库引用概述

链接库引用是一个连接到 XBRL 链接库的 XLink 简单链接。

6.9.3.2 XML 表示形式

链接库引用由<ver;report> 元素的若干个<link;linkbaseRef>子元素标识,<link;linkbaseRef>通过 XBRL 基础规范定义。

6.9.3.3 校验规则

一个链接库引用应遵循在 XBRL 基础规范中定义的规则。

6.10 相关版本报告

6.10.1 概述

一个相关报告是一个版本报告与另一个版本报告使用者可相关的报告。

一个版本报告可包含相关版本报告的引用,这种引用称为版本报告引用。

版本报告与相关版本报告的关系不存在任何限制。尤其对 From-DTS 与 To-DTS 之间的吻合或交叠不存在任何限制。以下给出了相关版本报告的示例。

示例:

一个扩展分类标准(基于“基础分类标准”扩展的)的版本报告是基础分类标准版本报告的相关版本报告。

版本报告引用是一个连接到 XBRL 相关管理报告的 XLink 简单链接。

6.10.2 XML 表示形式

版本报告引用用<ver:reportRef>元素表示,<ver:reportRef>是一个 XLink 规范简单链接。见表 9 所示。

表 9 <ver:reportRef> 元素的 XML 表示

<pre><ver:reportRef xlink:type = xs:string xlink:href = xs:anyURI xlink:arcrole = xs:anyURI xlink:role = xs:anyURI> Content:none </ver:reportRef></pre>	
特 性	表 达 含 义
{相关版本报告}	通过解析属性 xlink:href 的值来找到版本报告,相对 URLs 通过 XML Base 规范转换为绝对 URLs

6.10.3 校验规则

属性 xlink:type 的值应为“simple”。

相关版本报告应是一个本部分定义的有效版本报告。

属性 xlink:arcrole 应出现且值为“http://xbrl.org/arcrole/2010/versioning/related-report”。

7 基础事件

7.1 命名空间名称事件

命名空间名称事件如表 10 所示。

表 10 命名空间名称事件

代 码	元素	From 标识符	To 标识符
[NamespaceName]	<ver:namespaceMapping>	命名空间标识符	命名空间标识符

命名空间名称事件使 From-DTS 和 To-DTS 中的命名空间名称相关联。

7.2 角色 URI 事件

角色 URI 事件如表 11 所示。

表 11 角色 URI 事件

代码	元素	From 标识符	To 标识符
[RoleURI]	<ver:roleMapping>	角色 URI 标识符	角色 URI 标识符

角色 URI 事件使 From-DTS 和 To-DTS 中的角色 URI 相关联。

8 标识符

8.1 命名空间名称标识符

命名空间名称标识符是标识命名空间的 URI 引用。
URI 引用的 URI 属性应等于下列 DTS 中模式文件的目标命名空间：
——From-DTS, 如果命名空间名称标识符用于 From 标识符；
——To-DTS, 如果命名空间名称标识符用于 To 标识符。

8.2 角色 URI 标识符

一个角色 URI 标识符是定义角色 URI 的 URI 引用。
URI 引用的 URI 属性应等于下列 DTS 中定义的角色类型的 roleURI：
——From-DTS, 如果命名空间名称标识符用于 From 标识符；
——To-DTS, 如果命名空间名称标识符用于 To 标识符。

8.3 URI 引用

一个 URI 引用是 anyURI 的抽象标识符。
一个 URI 引用的表达可由下列两部分组成：
——一个<ver:fromURI> 元素, 如果用于 From 标识符；
——一个<ver:fromURI> 元素, 如果用于 To 标识符。
<ver:fromURI>和<ver:toURI>在 XML 中的表示如表 12 所示。

表 12 <ver:fromURI>和<ver:toURI>在 XML 中的表示

<pre><ver:fromURI value = xs:anyURI() Content:None </ver:fromURI></pre>	
<pre><ver:toURI value = xs:anyURI() Content:None </ver:toURI></pre>	
特 性	表 达 含 义
{URI}	<ver:fromURI>或<ver:toURI>元素的 value 属性的值

9 映射

9.1 命名空间映射

命名空间映射是 From 命名空间和相对应的 To 命名空间之间的关联。

一个 From 命名空间是 From-DTS 中的模式文件中的目标命名空间名称。

一个 To 命名空间是 To-DTS 中的模式文件中的目标命名空间名称。

一个命名空间映射由命名空间事件来定义。一个单独的命名空间名称可参与多个命名空间映射。

9.2 角色 URI 映射

角色 URI 映射是 From-Role-URI 和相对应的 To-Role-URI 之间的关联。

一个 From-Role-URI 是 From-DTS 中角色定义信息所定义的一个角色 URI。

一个 To-Role-URI 是 To-DTS 中角色定义信息所定义的一个角色 URI。

一个 To-Role-Mapping 是由一个角色 URI 事件所定义。一个单独角色 URI 可参与多个角色 URI 映射。

10 概念基础

10.1 概述

商业概念是商业信息的一个独立的抽象定义,一个 XBRL 概念是一个商业概念的具体实例化。XBRL 概念和商业概念的区别与本部分相关,这是由于一个单独的商业概念可能会在不同的 DTS 中由不同的 XBRL 概念表现。版本规范报告可以识别此类等价概念。A.2 的规范化模式文件对概念基础的语法进行了描述。

当两个 XBRL 概念代表相同的商业概念时,这两个 XBRL 概念为等价概念。事实报告使用任何一个概念来表达相同的信息都会被用户理解。概念等价性要求不会在概念的 XBRL 表现形式上强加特殊的限制。等价概念可能有不同的本地名称、命名空间、标签、引用或者数据类型。由于两个概念均代表同一个的商业概念,因此,两个概念的数据类型应该拥有相同的值空间,该值空间中所有商业概念的值均为有效值。在实际操作中,允许值空间不同。以下给出了数据类型不匹配的等价概念的示例。

示例:

在 From-DTS 的一个概念 A 中有一个数据类型字符串,而在其等价概念 To-DTS 中有一个带有限制形式的字符串数据类型更接近于商业概念的有效值。进一步讲,数据类型定义的错误意味着商业概念的有效值对另一个概念而言是无效的。但这并不能阻止两个概念成为等价概念。

本部分中数据类型变更的注释可以通过事件与相关概念关联。版本规范概念扩展规范提供了事件,这些事件允许专门为数据类型的变更附加注释,而不仅仅在概念层面提供注释。

当潜在商业概念的定义中有重叠时,两个 XBRL 概念被认为是相关概念。不同于等价概念,根据一个概念报告的事实未必和根据另一个概念报告的事实拥有相同的商业意义。相关概念可以作为一个 DTS 中没有等价概念的概念指明逻辑后继者。

10.2 事件

基础概念事件中共定义了三类事件。这三类事件与命名空间映射一起,生成了成对概念。成对概念的两个概念分别来源于 From-DTS 和 To-DTS。成对概念包括等价概念和相关概念。基础概念事件如表 13 所示。

表 13 基础概念事件

码	元 素	来源标识符	目标标识符
[ConceptDelete]	<vercb:conceptDelete>	Concept Identifier	—
[ConceptAdd]	<vercb:conceptAdd>	—	Concept Identifier
[ConceptName]	<vercb:conceptName>	Concept Identifier	Concept Identifier

10.3 概念标识符

10.3.1 概念标识符

概念标识符是 From-DTS 或者 To-DTS 中的一个概念的标识符。

10.3.2 XML 表现形式

- 一个概念标识符表现为：
- 作为来源标识符(From Identifier)使用时,表现为一个<vercb:fromE>元素；
 - 作为目标标识符(To Identifier)使用时,表现为一个<vercb:toE>元素。
- XML 表现形式总结如表 14 所示。

表 14 XML 表现形式总结：vercb:fromE 和 vercb:toE

<pre><vercb:fromE href = xs:anyURI Content:None </vercb:fromE></pre>	
<pre><vercb:toE href = xs:anyURI Content:None </vercb:toE></pre>	
特 性	表达含义
{Concept}	通过解析@href 属性的实际值来获得概念定义

10.3.3 概念校验规则

- 概念标识符的@href 属性必须解析为以下 DTS 中一个有效的概念定义：
- 当概念标识符作为来源标识符(From Identifier)使用时,概念定义在 From-DTS；
 - 当概念标识符作为目标标识符(To Identifier)使用时,概念定义在 To-DTS。

10.4 概念映射

- 概念映射信息通过以下版本报告获取：
- a) 当一个行为包含以下内容时, From DTS 中的概念 CF 和 To DTS 中的概念 CT 是相关的概念：
- 带有用于标识 CF 的来源标识符(From Identifier)的概念删除事件；
 - 带有用于标识 CT 的目标标识符(To Identifier)的概念增加事件。

- b) 当存在以下情况时,概念 CF 和概念 CT 为等价概念:
- 行为包含一个带有用于标识 CF 的来源标识符(From Identifier)和带有用于标识 CT 的目标标识符(To Identifier)的概念名称;
 - 不存在任何带有用于标识 CF 的来源标识符(From Identifier)的概念删除事件或概念名称事件,也不存在任何带有用于标识 CT 的目标标识符(To Identifier)的概念增加事件或概念名称事件,且 CF 和 CT 拥有相同的本地名称,存在命名空间映射为 From-Namespace 等于 CF 的命名空间名称及 To-Namespace 等于 CT 的命名空间名称。

附录 A

(规范性附录)

模式文件

A.1 版本——基础

```

<xs:schema xmlns:gen = "http://www.xbrl.org/2008/generic" xmlns:ver = "http://xbrl.org/2010/versioning-base"
xmlns:xs = "http://www.w3.org/2001/XMLSchema" xmlns:xl = "http://www.xbrl.org/2003/XLink"
xmlns:link = "http://www.xbrl.org/2003/linkbase" xmlns:xlink = "http://www.w3.org/1999/xlink"
targetNamespace = "http://xbrl.org/2010/versioning-base" elementFormDefault = "qualified">
  <xs:import namespace = "http://www.xbrl.org/2003/linkbase"
schemaLocation = "http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd"/>
  <xs:import namespace = "http://www.xbrl.org/2003/XLink"
schemaLocation = "http://www.xbrl.org/2003/xl-2003-12-31.xsd"/>
  <xs:import namespace = "http://www.w3.org/1999/xlink"
schemaLocation = "http://www.xbrl.org/2003/xlink-2003-12-31.xsd"/>
  <xs:annotation>
    <xs:appinfo>
      <link:arcroleType id = "versioning-label" arcroleURI = "http://xbrl.org/arcrole/2008/versioning/
label" cyclesAllowed = "none">
        <link:definition>
          has versioning label
        </link:definition>
        <link:usedOn>
          gen,arc
        </link:usedOn>
      </link:arcroleType>
      <link:arcroleType id = "versioning-reference" arcroleURI = "http://xbrl.org/arcrole/2008/
versioning/reference" cyclesAllowed = "none">
        <link:definition>
          has versioning reference
        </link:definition>
        <link:usedOn>
          gen,arc
        </link:usedOn>
      </link:arcroleType>
    </xs:appinfo>
  </xs:annotation>
  <!-- general nodes needed -->
  <xs:attributeGroup id = "xml-common.attributes" name = "common.attributes">
    <xs:attribute name = "id" type = "xs:ID" use = "optional"/>
    <xs:anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
  </xs:attributeGroup>
  <xs:attributeGroup id = "xml-required.common.attributes" name = "required.common.attributes">
    <xs:attribute name = "id" type = "xs:ID" use = "required"/>
    <xs:anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
  </xs:attributeGroup>
  <xs:complexType id = "xml-idref.type" name = "idref.type">
    <xs:attribute name = "ref" type = "xs:IDREF" use = "required"/>
    <xs:attributeGroup ref = "ver:common.attributes"/>
  </xs:complexType>

```

```

<!-- report node -->
<xs:element id = "xml-report" name = "report">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "link;linkbaseRef" minOccurs = "0" maxOccurs = "unbounded"/>
      <xs:element ref = "ver;reportRef" minOccurs = "0" maxOccurs = "unbounded"/>
      <xs:element ref = "ver;fromDTS" minOccurs = "1" maxOccurs = "1"/>
      <xs:element ref = "ver;toDTS" minOccurs = "1" maxOccurs = "1"/>
      <xs:element ref = "ver;assignments" minOccurs = "0" maxOccurs = "unbounded"/>
      <xs:element ref = "ver;action" minOccurs = "0" maxOccurs = "unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref = "ver;common.attributes"/>
  </xs:complexType>
</xs:element>
<xs:element id = "xml-reportref" name = "reportRef" substitutionGroup = "xl:simple">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base = "xl:simpleType">
        <xs:attribute ref = "xlink;arcrole" use = "required"/>
        <xs:anyAttribute namespace = "http://www.w3.org/XML/1998/namespace" processContents = "lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<!-- fromDTS and toDTS node -->
<xs:element id = "xml-from-dts" name = "fromDTS" type = "ver;dts.type" substitutionGroup = "ver;
dts.identifier"/>
<xs:element id = "xml-to-dts" name = "toDTS" type = "ver;dts.type" substitutionGroup = "ver;dts.identifier"/>
<xs:element id = "xml-dts-identifier" name = "dts.identifier" type = "ver;dts.type" abstract = "true"/>
<xs:complexType id = "xml-dts.type" name = "dts.type">
  <xs:choice minOccurs = "1" maxOccurs = "unbounded">
    <xs:element ref = "link;linkbaseRef"/>
    <xs:element ref = "link;schemaRef"/>
  </xs:choice>
</xs:complexType>
<!-- category node -->
<xs:element id = "xml-category" name = "category" type = "ver;category.type" abstract = "true"/>
<xs:complexType id = "xml-category.type" name = "category.type">
  <xs:attributeGroup ref = "ver;common.attributes"/>
</xs:complexType>
<!-- predefined categories -->
<xs:element id = "xml-errata.category" name = "errataCategory" type = "ver;category.type"
substitutionGroup = "ver;category"/>
<xs:element id = "xml-business.category" name = "businessCategory" type = "ver;category.type"
substitutionGroup = "ver;category"/>
<xs:element id = "xml-technical.category" name = "technicalCategory" type = "ver;category.type"
substitutionGroup = "ver;category"/>
<!-- assignment node -->
<xs:element id = "xml-assignments" name = "assignments" type = "ver;assignments.type"/>
<xs:complexType id = "xml-assignments.type" name = "assignments.type">
  <xs:sequence>
    <xs:element ref = "ver;assignment" minOccurs = "1" maxOccurs = "unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref = "ver;common.attributes"/>
</xs:complexType>
<xs:element id = "xml-assignment" name = "assignment">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element ref = "ver:category" minOccurs = "0" maxOccurs = "unbounded" />
  </xs:sequence>
  <xs:attributeGroup ref = "ver:common.attributes" />
</xs:complexType>
</xs:element>
<!-- action node -->
<xs:element id = "xml-action" name = "action">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "ver:actionRef" minOccurs = "0" maxOccurs = "unbounded" />
      <xs:element ref = "ver:event" minOccurs = "0" maxOccurs = "unbounded" />
    </xs:sequence>
    <xs:attributeGroup ref = "ver:common.attributes" />
  </xs:complexType>
</xs:element>
<xs:element id = "xml-actionref" name = "actionRef" type = "ver:idref.type" />
<!-- mapping nodes -->
<!-- arcroles are not served in versioning-base.xsd <xs:element id = "xml-arcrole.mapping" name = "
arcroleMapping" substitutionGroup = "ver:mapping" /> -->
<xs:element id = "xml-namespace.mapping" name = "namespaceMapping" substitutionGroup = "ver:mapping" />
<xs:element id = "xml-role.mapping" name = "roleMapping" substitutionGroup = "ver:mapping" />
<xs:element id = "xml-mapping" name = "mapping" type = "ver:mapping.type" substitutionGroup = "ver:event"
abstract = "true" />
<xs:complexType id = "xml-mapping.type" name = "mapping.type">
  <xs:complexContent>
    <xs:restriction base = "ver:event.type">
      <xs:sequence>
        <xs:element ref = "ver:fromURI" />
        <xs:element ref = "ver:toURI" />
      </xs:sequence>
      <xs:attributeGroup ref = "ver:common.attributes" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:element id = "xml-from.uri" name = "fromURI" type = "ver:uri.type" />
<xs:element id = "xml-to.uri" name = "toURI" type = "ver:uri.type" />
<xs:complexType id = "xml-uri.type" name = "uri.type">
  <xs:attribute name = "value" type = "xs:anyURI" use = "required" />
  <xs:attributeGroup ref = "ver:common.attributes" />
</xs:complexType>
<!-- event node -->
<xs:element id = "xml-event" name = "event" type = "ver:event.type" abstract = "true" />
<xs:complexType name = "event.type">
  <xs:complexContent>
    <xs:restriction base = "xs:anyType">
      <xs:sequence>
        <xs:any namespace = "##any" minOccurs = "0" maxOccurs = "unbounded" />
      </xs:sequence>
      <xs:attributeGroup ref = "ver:common.attributes" />
      <xs:anyAttribute processContents = "lax" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
</xs:schema>

```

A.2 版本——概念基础

```

<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"
xmlns:vercb = "http://xbrl.org/2010/versioning-concept-basic" xmlns:ver = "http://xbrl.org/2010/versioning-
base" targetNamespace = "http://xbrl.org/2010/versioning-concept-basic" elementFormDefault = "qualified">
  <xs:import namespace = "http://xbrl.org/2010/versioning-base" schemaLocation = "versioning-base.xsd"/>
  <!-- Parents of the events -->
  <xs:element name = "fromConcept" type = "ver:uri.type" id = "xml-from-e"/>
  <xs:element name = "toConcept" type = "ver:uri.type" id = "xml-to-e"/>
  <!-- complexTypes event nodes-->
  <xs:complexType name = "add.element.event.type" id = "xml-add.element.event.type">
    <xs:complexContent>
      <xs:restriction base = "ver:event.type">
        <xs:sequence>
          <xs:element ref = "vercb:toConcept"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name = "delete.element.event.type" id = "xml-delete.element.event.type">
    <xs:complexContent>
      <xs:restriction base = "ver:event.type">
        <xs:sequence>
          <xs:element ref = "vercb:fromConcept"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name = "change.element.event.type" id = "xml-change.element.event.type">
    <xs:complexContent>
      <xs:restriction base = "ver:event.type">
        <xs:sequence>
          <xs:element ref = "vercb:fromConcept"/>
          <xs:element ref = "vercb:toConcept"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <!-- Concept events -->
  <xs:element name = "conceptAdd" type = "vercb:add.element.event.type" substitutionGroup = "ver:event"
id = "xml-concept.add.event"/>
  <xs:element name = "conceptDelete" type = "vercb:delete.element.event.type" substitutionGroup = "ver:event"
id = "xml-concept.delete.event"/>
  <xs:element name = "conceptRename" type = "vercb:change.element.event.type" substitutionGroup = "ver:event"
id = "xml-concept.rename.event"/>
</xs:schema>

```

参 考 文 献

- [1] <http://www.w3.org/TR/xmlschema-0/>
 - [2] <http://www.w3.org/TR/xmlschema-1/>
 - [3] <http://www.w3.org/TR/xmlschema-2/>
 - [4] <http://www.iso.ch/iso/en/CatalogueDetailPageCatalogueDetail? CSNUMBER=16387>
 - [5] <http://www.ietf.org/rfc/rfc2396.txt>
 - [6] <http://www.ietf.org/rfc/rfc2732.txt>
 - [7] http://www.xbrl.org/resourcecenter/downloadsspecs.asp? rid=2&fname=XBRL_2.0a&demos=XBRL-2001-12-14231.doc
 - [8] <http://www.w3.org/TR/xhtml-modularization/>
 - [9] <http://www.w3.org/TR/xmlbase/>
 - [10] <http://www.w3.org/TR/xlink/>
 - [11] <http://www.w3.org/TR/xpath>
 - [12] <http://www.w3.org/TR/xptr-framework/>
 - [13] <http://www.xbrl.org/Specification/gnl/PR-2009-03-31/gnl-PR-2009-03-31.htm>
 - [14] <http://www.xbrl.org/Specification/genericReferences/REC-2009-06-22/genericReferences-REC-2009-06-22.html>
 - [15] <http://www.xbrl.org/Specification/genericLabels/REC-2009-06-22/genericLabels-REC-2009-06-22.html>
-